ANSYS

Theory Reference

Release 5.6

We have shipped this manual unassembled to prevent unnecessary damage to the pages and binder rings during shipping.

To assemble the manual:

1. Remove the shrink–wrap packaging.
2. Insert all pages onto the binder rings. We suggest you keep the heavy, blank pages on the front and back of the manual.
3. Insert the plastic sheet lifter in *front* of the manual to help keep the pages from getting caught under the curved portion of the rings.

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
Phone: (724) 746–3304
Fax: (724) 514–9494
Internet: ansysinfo@ansys.com

Theory Reference

Release 5.6

Edited by
Peter Kohnke, Ph.D.

ANSYS, Inc.
Southpointe
275 Technology Drive
Canonsburg, PA 15317
Phone: (724) 746–3304
Fax: (724) 514–9494
Internet: ansysinfo@ansys.com

ANSYS, Inc. is a UL registered **ISO 9001: 1994** Company

Revision History

DN–R300:50–4	Upd0	ANSYS Revision 5.0	December 23, 1992
DN–R300:50–4	Upd0	ANSYS Revision 5.0	November 1, 1993
DN–R300:50–4	Upd0	ANSYS Revision 5.0	November 19, 1993
DN–R300:50–4	Upd0	ANSYS Revision 5.0	February 15, 1994
DN–R300:51–4	Upd0	ANSYS Revision 5.1	September 30, 1994
000450	Upd0	ANSYS Revision 5.2	August 31, 1995
000656	Seventh Edition	ANSYS Release 5.3	June 1996
000855	Eighth Edition	ANSYS Release 5.4	September 1997
001099	Tenth Edition	ANSYS Release 5.5	September 1998
001242	Eleventh Edition	ANSYS Release 5.6	November 1999

Registered Trademarks

Disclaimer Notice

This document has been reviewed and approved in accordance with ANSYS, Inc. Documentation Review and Approval Procedures.

Acknowledgement

The contributions of the technical staff at ANSYS, Inc. are gratefully acknowledged for writing most of the modifications required to update the Theory volume of the User's Manual from the previous edition. Production help came from many sides, but special mention must be made of Jackie Williamson, who considers this "her" manual for good reason. Further, many people throughout ANSYS, Inc. were involved in checking of the manuscript and this is also appreciated.

Note

Most equations are shown in a different font than the rest of the text. For example, the equation may use ϱ, but the text uses ρ. This discrepancy will be removed at a later edition. The editor apologizes for any inconvenience that this causes.

Future Editions

The ANSYS program is continually being improved and expanded, so future editions of this manual are planned. It is requested that all suggestions for improvements to the manual, or to the theory itself, be forwarded to the editor so this manual and the ANSYS program can be made more useful.

Theory Reference
Table of Contents

The Phenomena

5 Electromagnetics

6 Heat Flow

7 Fluid Flow

8 Acoustics

9 This chapter intentionally omitted

10 This chapter intentionally omitted

11 Coupling

The Elements

12 Shape Functions

13 Element Tools

14 Element Library

The Solvers

15 Analysis Tools

16 This chapter intentionally omitted

17 Analysis Procedures

The Data Handlers

18 Pre and Postprocessing Tools

19 Postprocessing

20 Design Optimization

Chapter 1
Introduction

ANSYS Theory Reference
Chapter 1 – Table of Contents

1.1 Introduction

Welcome to the *ANSYS Theory* Reference . This manual presents theoretical descriptions of all ANSYS elements, as well as of many procedures and commands. It is included in the basic documentation set because every ANSYS user should understand how the program uses the input data to calculate the output. In addition, this manual is indispensable for its explanations of how to interpret certain element and command results. In addition, the *ANSYS Theory Reference* describes the relationship between input data and output results produced by the program, and is essential for a thorough understanding of how the program functions.

1.1.1 Purpose of the ANSYS Theory Reference

The purpose of the *ANSYS Theory Reference* is to inform you of the theoretical basis of the ANSYS program. By understanding the underlying theory, you can use the ANSYS program more intelligently and with greater confidence, making better use of its capabilities while being aware of its limitations. Of course, you are not expected to study the whole of this volume; you need only to refer to sections of it as required for specific elements and procedures.

This manual does not, and cannot, present all theory relating to finite element analysis. If you need the theory behind the basic finite element method, you should obtain one of the many references available on the topic. If you need theory or information that goes beyond that presented here, you should (as applicable) consult the indicated reference, run a simple test problem to try the feature of interest, or contact your ANSYS Support Distributor for more information.

The theory behind the basic analysis disciplines is presented in Chapters 2 through 11. Chapter 2 covers structural theory, with Chapters 3 and 4 adding geometric and structural material nonlinearities. Chapter 5 discusses electromagnetics, Chapter 6 deals with heat flow, Chapter 7 handles fluid flow and Chapter 8 deals with acoustics. Chapters 9 and 10 are reserved for future topics. Coupled effects are treated in Chapter 11.

Element theory is examined in Chapters 12, 13, and 14. Shape functions are presented in Chapter 12, information about element tools (integration point locations, matrix information, and other topics) is discussed in Chapter 13, and theoretical details of each ANSYS element are presented in Chapter 14.

Chapter 15 examines a number of analysis tools (acceleration effect, damping, element reordering, and many other features). Chapter 16 is reserved for a future topic. Chapter 17 discusses the theory behind the different analysis types used in the ANSYS program.

Numerical processors used in preproprocessing and postprocessing are covered in Chapter 18. Chapter 19 goes into a number of features from the general postprocessor (POST1) and the time–history postprocessor (POST26). Chapter 20 deals with design optimization.

An index of keywords and commands has been compiled to give you handy access to the topic or command of interest.

1.1.2 Notation

The notation defined below is a partial list of the notation used throughout the manual. There are also some tables of definitions given in Section 11.0 — Coupled Effects, and Section 4.1 — Rate–Independent Plasticity following sections:

Due to the wide variety of topics covered in this manual, some exceptions will exist.

A. General Terms

Term	**Meaning**
[B]	strain–displacement matrix
[C]	damping matrix
$[C^t]$	specific heat matrix
[D]	elasticity matrix
E	Young's modulus
{F}	force vector
[I]	identity matrix
{I}	current vector, associated with electrical potential DOFs
{J}	current vector, associated with magnetic potential DOFs
[K]	stiffness matrix
$[K^t]$	conductivity matrix
[M]	mass matrix
[O]	null matrix
P, {P}	pressure (vector)
{Q}	heat flow vector
[S]	stress stiffness matrix
{T}	temperature vector
t	time, thickness
$[T_R]$	local to global conversion matrix
u, v, w, {u}	displacement, displacement vector
{V}	electric potential vector
δU	virtual internal work
δV	virtual external work

Term	Meaning
{W}	fluid flow vector
x, y, z	element coordinate
X, Y, Z	nodal coordinates (usually global Cartesian)
α	coefficient of thermal expansion
ε	strain
ν	Poisson's ratio
σ	stress

B. Superscripts and Subscripts

Below is a partial list of superscripts and subscripts used on [K], [M], [C], [S], {u}, {T}, and/or {F}. See also Section 11.0. The absence of a subscript on the above terms implies the total matrix in final form, ready for solution.

Term	Meaning
ac	nodal effects caused by an acceleration field
c	convection surface
cr	creep
e	based on element in global coordinates
el	elastic
g	internal heat generation
i	equilibrium iteration number
ℓ	based on element in element coordinates
m	master
n	substep number (time step)
nd	effects applied directly to node
pl	plasticity
pr	pressure
s	slave
sw	swelling
t, th	thermal
^	(flex over term) reduced matrices and vectors
·	(dot over term) time derivative

1.1.3 Applicable ANSYS Products

This manual applies to the following ANSYS products (described fully in the preface to the *ANSYS Basic Analysis Procedures Guide*):

ANSYS/Multiphysics
ANSYS/Mechanical
ANSYS/Structural
ANSYS/Mechanical with the electromagnetics add–on (“Emag”)
ANSYS/Mechanical with the FLOTRAN CFD add–on (“FLOTRAN”)
ANSYS/LinearPlus (“Linear” or “LinearPlus”)
ANSYS/Thermal (“Thermal”)
ANSYS/Emag 3–D (“Emag” or “Emag 3–D”)
ANSYS/Emag 2–D (“Emag” or “Emag 2–D”)
ANSYS/FLOTRAN (“FLOTRAN”)
ANSYS/PrepPost (“PrepPost”)
ANSYS/ED (“ED”)

Some command arguments and element KEYOPT settings have defaults in the derived products that are different from those in the full ANSYS product. These cases are clearly documented under the “Product Restrictions” section of the affected commands and elements. If you plan to use your derived product input file in the ANSYS/Multiphysics product, you should explicitly input these settings in the derived product, rather than letting them default; otherwise, behavior in the full ANSYS product will be different.

Chapter 2
Structures

ANSYS Theory Reference
Chapter 2 – Table of Contents

2.1 Structural Fundamentals

2.1.1 Stress–Strain Relationships

This section discusses material relationships for linear materials. Nonlinear materials are discussed in Chapter 4. The stress is related to the strains by:

$$\{\sigma\} = [D] \{\varepsilon^{el}\} \qquad (2.1\text{–}1)$$

where: $\{\sigma\}$ = stress vector = $\lfloor \sigma_x \; \sigma_y \; \sigma_z \; \sigma_{xy} \; \sigma_{yz} \; \sigma_{xz} \rfloor^T$ (also output quantity S)

$[D]$ = elasticity or elastic stiffness matrix (defined in equations (2.1–19) through (2.1–24) or inverse defined in equations (2.1–4) and (2.1–5) or, for a few anistropic elements, defined with input from the **TBDATA** commands with **TB**,ANEL.)

$\{\varepsilon^{el}\}$ = $\{\varepsilon\} - \{\varepsilon^{th}\}$ = elastic strain vector (output quantity EPEL)

$\{\varepsilon\}$ = total strain vector = $\lfloor \varepsilon_x \; \varepsilon_y \; \varepsilon_z \; \varepsilon_{xy} \; \varepsilon_{yz} \; \varepsilon_{xz} \rfloor^T$

$\{\varepsilon^{th}\}$ = thermal strain vector (defined in equation (2.1–3)) (output quantity EPTH)

The output quantity EPEL is the strain that causes stress ($\{\varepsilon\} - \{\varepsilon^{th}\}$).

Note: A related quantity used in POST1 labelled “component total strain” (output quantity EPTO) is described in Section 4.0.

The stress vector is shown in Figure 2.1–1. The sign convention for direct stresses and strains used throughout the ANSYS program is that tension is positive and compression is negative. For shears, positive is when the two applicable positive axes rotate toward each other. Shear strains are engineering shear strains, not tensor shear strains.

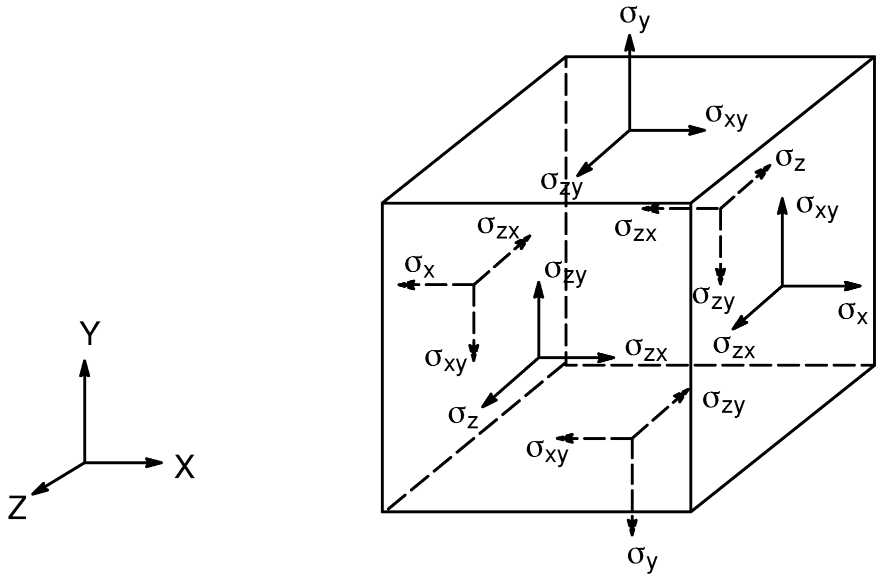


Figure 2.1–1 Stress Vector Definition

Equation (2.1–1) may also be inverted to:

$$\{\varepsilon\} = \{\varepsilon^{th}\} + [D]^{-1}\{\sigma\} \tag{2.1–2}$$

For the 3–D case, the thermal strain vector is:

$$\{\varepsilon^{th}\} = \Delta T \lfloor \alpha_x \quad \alpha_y \quad \alpha_z \quad 0 \quad 0 \quad 0 \rfloor^T \tag{2.1–3}$$

where: α_x = thermal coefficient of expansion in the x direction (input as ALPX on **MP** command)

ΔT = $T - T_{REF}$

T = current temperature at the point in question

T_{REF} = reference (strain–free) temperature (input on **TREF** command)

The flexibility or compliance matrix, $[D]^{-1}$, in “column normalized” format, is:

$$[D]^{-1} = \begin{bmatrix} 1/E_x & -\nu_{xy}/E_y & -\nu_{xz}/E_z & 0 & 0 & 0 \\ -\nu_{yx}/E_x & 1/E_y & -\nu_{yz}/E_z & 0 & 0 & 0 \\ -\nu_{zx}/E_x & -\nu_{zy}/E_y & 1/E_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{yz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{xz} \end{bmatrix} \qquad (2.1\text{–}4)$$

or in “row–normalized” format, is:

$$[D]^{-1} = \begin{bmatrix} 1/E_x & -\bar{\nu}_{xy}/E_x & -\bar{\nu}_{xz}/E_x & 0 & 0 & 0 \\ -\bar{\nu}_{yx}/E_y & 1/E_y & -\bar{\nu}_{yz}/E_y & 0 & 0 & 0 \\ -\bar{\nu}_{zx}/E_z & -\bar{\nu}_{zy}/E_z & 1/E_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{yz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{xz} \end{bmatrix} \qquad (2.1\text{–}5)$$

where typical terms are:

E_x = Young’s modulus in the x direction (input as EX on **MP** command)

ν_{xy} = minor Poisson’s ratio (input as NUXY on **MP** command)

$\bar{\nu}_{xy}$ = major Poisson’s ratio (input as PRXY on **MP** command)

G_{xy} = shear modulus in the xy plane (input as GXY on **MP** command)

The $[D]^{-1}$ matrix must be positive definite (see Section 13.5). Also, the $[D]^{-1}$ matrix is presumed to be symmetric, so that for orthotropic materials:

$$\frac{\nu_{yx}}{E_x} = \frac{\nu_{xy}}{E_y} \qquad (2.1\text{–}6)$$

$$\frac{\nu_{zx}}{E_x} = \frac{\nu_{xz}}{E_z} \qquad (2.1\text{–}7)$$

$$\frac{\nu_{zy}}{E_y} = \frac{\nu_{yz}}{E_z} \qquad (2.1\text{–}8)$$

or

$$\frac{\bar{\nu}_{yx}}{E_y} = \frac{\bar{\nu}_{xy}}{E_x} \qquad (2.1\text{–}9)$$

$$\frac{\bar{\nu}_{zx}}{E_z} = \frac{\bar{\nu}_{xz}}{E_x} \qquad (2.1\text{–}10)$$

$$\frac{\bar{\nu}_{zy}}{E_z} = \frac{\bar{\nu}_{yz}}{E_y} \qquad (2.1\text{–}11)$$

Because of the above six relationships, ν_{yx}, ν_{zy}, ν_{zx}, $\bar{\nu}_{yx}$, $\bar{\nu}_{zy}$, and $\bar{\nu}_{zx}$ are dependent quantities and are therefore not input quantities.

The use of Poisson's ratios (input as either PRXY, PRYZ, and PRXZ, or NUXY, NUYZ, and NUXZ) for orthotropic materials sometimes causes confusion, so that care should be taken in their use. Note that the right hand sides of equations (2.1–4) and (2.1–5) must be exactly identical to each other. Therefore, looking at the terms in the first row and second column, it may be seen that:

$$\frac{\nu_{xy}}{E_y} = \frac{\bar{\nu}_{xy}}{E_x} \qquad (2.1\text{–}12$$

Assuming that E_x is larger than E_y, $\bar{\nu}_{xy}$ (PRXY) is larger than ν_{xy} (NUXY). Hence, $\bar{\nu}_{xy}$ is commonly referred to as the “major Poisson's ratio”, because it is larger than ν_{xy} ($=\bar{\nu}_{yx}$) which is commonly referred to as the “minor” Poisson's ratio. When describing Poisson's ratios, row–normalized format, $\bar{\nu}$–notation, “major” form, and PR–notation are all synonymous, while column–normalized format, ν–notation, “minor” form, and NU–notation are also all synonymous. For simplicity, only ν_{xy}, ν_{yz}, ν_{xz}, (and not $\bar{\nu}_{xy}$, $\bar{\nu}_{yz}$, and $\bar{\nu}_{xz}$) will be used in the rest of this section. For isotropic materials ($E_x = E_y = E_z$ and $\nu_{xy} = \nu_{yz} = \nu_{xz}$), it makes no difference which type of input is used. For orthotropic materials, the user needs to inquire of the source of the material property data as to which type of input is appropriate. In practice, orthotropic material data are most often supplied in the PR notation form.

Expanding equation (2.1–2) with equations (2.1–3), (2.1–4), and (2.1–6) thru (2.1–8) and writing out the six equations explicitly,

$$\varepsilon_x = \alpha_x \, \Delta T + \frac{\sigma_x}{E_x} - \frac{\nu_{xy}\,\sigma_y}{E_y} - \frac{\nu_{xz}\,\sigma_z}{E_z} \qquad (2.1\text{–}13)$$

$$\varepsilon_y = \alpha_y \, \Delta T - \frac{\nu_{xy} \sigma_x}{E_y} + \frac{\sigma_y}{E_y} - \frac{\nu_{yz} \sigma_z}{E_z} \qquad (2.1\text{–}14)$$

$$\varepsilon_z = \alpha_z \, \Delta T - \frac{\nu_{xz} \sigma_x}{E_z} - \frac{\nu_{yz} \sigma_y}{E_z} + \frac{\sigma_z}{E_z} \qquad (2.1\text{–}15)$$

$$\varepsilon_{xy} = \frac{\sigma_{xy}}{G_{xy}} \qquad (2.1\text{–}16)$$

$$\varepsilon_{yz} = \frac{\sigma_{yz}}{G_{yz}} \qquad (2.1\text{–}17)$$

$$\varepsilon_{xz} = \frac{\sigma_{xz}}{G_{xz}} \qquad (2.1\text{–}18)$$

where typical terms are:

ε_x	=	direct strain in the x direction
ε_{xy}	=	shear strain in the x–y plane
σ_x	=	direct stress in the x direction
σ_{xy}	=	shear stress on the x–y plane

Alternatively, equation (2.1–1) may be expanded by first inverting equation (2.1–4) and then combining that result with equations (2.1–3) and (2.1–6) thru (2.1–8) to give six explicit equations:

$$\sigma_x = \frac{E_x}{h} \left(1 - \left(\nu_{yz} \right)^2 \frac{E_y}{E_z} \right) \left(\varepsilon_x - \alpha_x \, \Delta T \right) + \frac{E_x}{h} \left(\nu_{xy} + \nu_{xz} \nu_{yz} \frac{E_y}{E_z} \right)$$

$$\left(\varepsilon_y - \alpha_y \, \Delta T \right) + \frac{E_x}{h} \left(\nu_{xz} + \nu_{yz} \nu_{xy} \right) \left(\varepsilon_z - \alpha_z \, \Delta T \right) \qquad (2.1\text{–}19)$$

$$\sigma_y = \frac{E_x}{h} \left(\nu_{xy} + \nu_{xz} \nu_{yz} \frac{E_y}{E_z} \right) \left(\varepsilon_x - \alpha_x \, \Delta T \right) + \frac{E_y}{h} \left(1 - \left(\nu_{xz} \right)^2 \frac{E_x}{E_z} \right)$$

$$\left(\varepsilon_y - \alpha_y \, \Delta T \right) + \frac{E_y}{h} \left(\nu_{yz} + \nu_{xz} \nu_{xy} \frac{E_x}{E_y} \right) \left(\varepsilon_z - \alpha_z \, \Delta T \right) \qquad (2.1\text{–}20)$$

$$\sigma_z = \frac{E_x}{h}\left(\nu_{xz} + \nu_{yz}\nu_{xy}\right)\left(\varepsilon_x - \alpha_x\,\Delta T\right) + \frac{E_y}{h}\left(\nu_{yz} + \nu_{xz}\nu_{xy}\frac{E_x}{E_y}\right)$$

$$\left(\varepsilon_y - \alpha_y\,\Delta T\right) + \frac{E_z}{h}\left(1 - \left(\nu_{xy}\right)^2\frac{E_x}{E_y}\right)\left(\varepsilon_z - \alpha_z\,\Delta T\right) \qquad (2.1\text{–}21)$$

$$\sigma_{xy} = G_{xy}\ \epsilon_{xy} \qquad (2.1\text{–}22)$$

$$\sigma_{yz} = G_{yz}\ \epsilon_{yz} \qquad (2.1\text{–}23)$$

$$\sigma_{xz} = G_{xz}\ \epsilon_{xz} \qquad (2.1\text{–}24)$$

where:

$$h = 1 - \left(\nu_{xy}\right)^2\frac{E_x}{E_y} - \left(\nu_{yz}\right)^2\frac{E_y}{E_z} - (\nu_{xz})^2\frac{E_x}{E_z} - 2\ \nu_{xy}\ \nu_{yz}\ \nu_{xz}\frac{E_x}{E_z}$$

If the shear moduli G_{xy}, G_{yz}, and G_{xz} are not input for isotropic materials, they are computed as:

$$G_{xy} = G_{yz} = G_{xz} = \frac{E_x}{2\ (1 + \nu_{xy})} \qquad (2.1\text{–}25)$$

For orthotropic materials, the user needs to inquire of the source of the material property data as to the correct values of the shear moduli, as there are no defaults provided by the program.

The [D] matrix must be positive definite. The program checks each material property as used by each active element type to ensure that [D] is indeed positive definite. Positive definite matrices are defined in Section 13.5. In the case of temperature dependent material properties, the evaluation is done at the uniform temperature (**BFUNIF** command) for the first load step. The material is always positive definite if the material is isotropic or if ν_{xy}, ν_{yz}, and ν_{xz} are all zero. An example of a material that is not positive definite is one that has E_y less than or equal to $E_x(\nu_{xy})^2$.

2.1.2 Orthotropic Material Transformation for Axisymmetric Models

The transformation of material property data from the R–θ–Z cylindrical system to the x–y–z system used for the input requires special care. The conversion of the Young's moduli is fairly direct, whereas the correct method of conversion of the Poisson's ratios is not obvious. Consider first how the Young's moduli transform from the global cylindrical system to the global Cartesian as used by the axisymmetric elements for a disc:

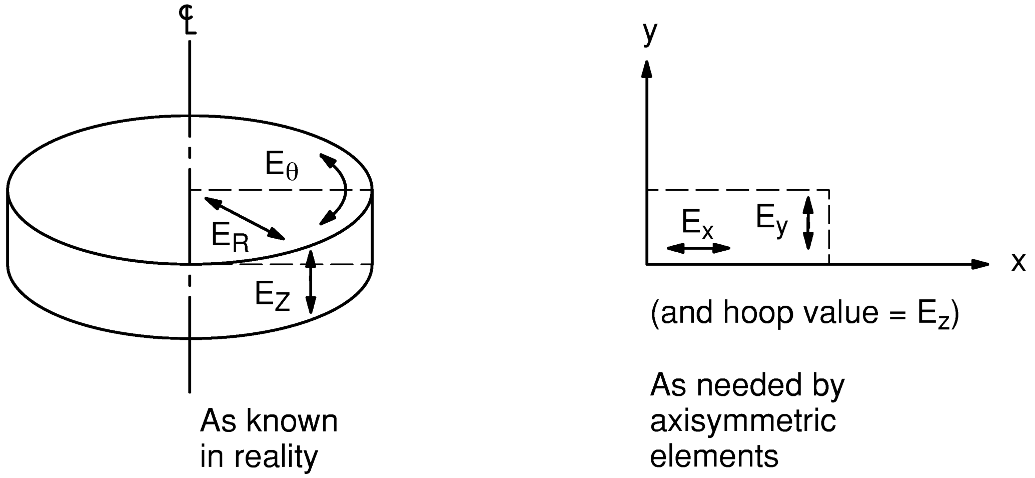


Figure 2.1–2 Material Coordinate Systems

Thus, $E_R \rightarrow E_x$, $E_\theta \rightarrow E_z$, and $E_Z \rightarrow E_y$. Starting with the global Cartesian system, the input for x–y–z using input labels NUXY, NUYZ, and NUXZ on the **MP** command gives the following column–normalized compliance matrix for the non–shear terms:

$$[D_{x-y-z}]^{-1} = \begin{bmatrix} 1/E_x & -\nu_{xy}/E_y & -\nu_{xz}/E_z \\ -\nu_{yx}/E_x & 1/E_y & -\nu_{yz}/E_z \\ -\nu_{zx}/E_x & -\nu_{zy}/E_y & 1/E_z \end{bmatrix} \qquad (2.1\text{–}26)$$

Assuming that the R–θ–Z Poisson's ratio data is also column–normalized, the compliance matrix is the same as equation (2.1–26). Rearranging so that the R–θ–Z axes match the x–y–z axes (i.e., x → R, y → Z, z → θ):

$$[D_{R-\theta-Z}]^{-1} = \begin{bmatrix} 1/E_R & -\nu_{RZ}/E_Z & -\nu_{R\theta}/E_\theta \\ -\nu_{ZR}/E_R & 1/E_Z & -\nu_{Z\theta}/E_\theta \\ -\nu_{\theta R}/E_R & -\nu_{\theta Z}/E_Z & 1/E_\theta \end{bmatrix} \qquad (2.1\text{–}27)$$

Comparing equation (2.1–26) and (2.1–27) term–by–term gives:

$$E_x = E_R \qquad (2.1\text{–}28)$$

$$E_y = E_Z \tag{2.1–29}$$

$$E_z = E_\theta \tag{2.1–30}$$

$$\nu_{xy} = \nu_{RZ} \tag{2.1–31}$$

$$\nu_{yz} = \nu_{Z\theta} = \frac{E_\theta}{E_Z} \nu_{\theta Z} \tag{2.1–32}$$

$$\nu_{xz} = \nu_{R\theta} \tag{2.1–33}$$

where the left–hand side of the above equations represents input (using input labels NUXY, NUYZ, and NUXZ) and the right–hand side represents the given data in column–normalized format.

If the R–θ–Z Poisson's ratio data are in row–normalized form, the compliance matrix in the rearranged form is:

$$\left[D_{R-\theta-Z}\right]^{-1} = \begin{bmatrix} 1/E_R & -\bar{\nu}_{RZ}/E_R & -\bar{\nu}_{R\theta}/E_R \\ -\bar{\nu}_{ZR}/E_Z & 1/E_Z & -\bar{\nu}_{Z\theta}/E_Z \\ -\bar{\nu}_{\theta R}/E_\theta & -\bar{\nu}_{\theta Z}/E_\theta & 1/E_\theta \end{bmatrix} \tag{2.1–34}$$

Comparing equation (2.1–26) and (2.1–34) gives:

$$E_x = E_R \tag{2.1–35}$$

$$E_y = E_Z \tag{2.1–36}$$

$$E_z = E_\theta \tag{2.1–37}$$

$$\nu_{xy} = \frac{E_Z}{E_R} \bar{\nu}_{RZ} \tag{2.1–38}$$

$$\nu_{yz} = \frac{E_\theta}{E_Z} \bar{\nu}_{Z\theta} = \bar{\nu}_{\theta Z} \tag{2.1–39}$$

$$\nu_{xz} = \frac{E_\theta}{E_R} \bar{\nu}_{R\theta} \tag{2.1–40}$$

where, again, the left–hand side of the above equations represents input (using input labels NUXY, NUYZ, and NUXZ) and the right–hand side represents the given data in row–normalized format.

The inputs shown in equations (2.1–28) through (2.1–33) and equations (2.1–35) through (2.1–40) could also be given using input labels PRXY, PRYZ, and PRXZ with the appropriate additional transformations.

2.1.3 Temperature–Dependent Coefficient of Thermal Expansion

Considering a typical component, the thermal strain from equation (2.1–3) is:

$$\epsilon^{th} = \alpha \left(T - T_{ref}\right) \tag{2.1–41}$$

This assumes that α is itself not a function of temperature. If α is a function of temperature, equation (2.1–41) becomes:

$$\epsilon^{th} = \int_{T_{ref}}^{T} \alpha_{inst}(T)\ dT \tag{2.1–42}$$

where: $\alpha_{inst}(T)$ = instantaneous coefficient of thermal expansion

The program, however, uses a mean or weighted–average value of α, similar in form to equation (2.1–41):

$$\epsilon^{th} = \overline{\alpha}(T) \left(T - T_{ref}\right) \tag{2.1–43}$$

where: $\overline{\alpha}(T) = \dfrac{\int_{T_{ref}}^{T} \alpha_{inst}\ dT}{T - T_{ref}}$

= mean value of coefficient of thermal expansion (input as ALPX, ALPY, or ALPZ on the **MP** command)

Data is commonly supplied as a mean value, and this data is correctly used by the program, as long as T_{ref} is not only the definition temperature about which the data is supplied but also the reference temperature at which zero strains exist. If this condition is not true, an adjustment must be made. Consider:

$$\epsilon_o^{th} = \bar{\alpha}^o(T)\ (T - T_o) = \int_{T_o}^{T} \alpha_{inst}\ dT \tag{2.1–44}$$

$$\epsilon_r^{th} = \bar{\alpha}^r(T)\ \left(T - T_{ref}\right) = \int_{T_{ref}}^{T} \alpha_{inst}\ dT \tag{2.1–45}$$

Equations (2.1–44) and (2.1–45) represent the thermal strain at a temperature T for two different starting points, T_o and T_{ref}. Now let T_o be the temperature about which the data has been generated (definition temperature), and T_{ref} be the temperature at which all strains are zero (reference temperature). Thus, $\bar{\alpha}^o(T)$ is the supplied data, and $\bar{\alpha}^r(T)$ is what is needed as program input.

The right–hand side of equation (2.1–44) may be expanded as:

$$\int_{T_o}^{T} \alpha_{inst}\ dT = \int_{T_o}^{T_{ref}} \alpha_{inst}\ dT + \int_{T_{ref}}^{T} \alpha_{inst}\ dT \tag{2.1–46}$$

also,

$$\int_{T_o}^{T_{ref}} \alpha_{inst}\ dT = \bar{\alpha}^o(T_{ref})\ (T_{ref} - T_o) \tag{2.1–47}$$

or

$$\int_{T_o}^{T_{ref}} \alpha_{inst}\ dT = \bar{\alpha}^r(T_o)\ (T_{ref} - T_o) \tag{2.1–48}$$

Combining equations (2.1–44) through (2.1–47),

$$\bar{\alpha}^r(T) = \bar{\alpha}^o(T) + \frac{T_{ref} - T_o}{T - T_{ref}}\ \left(\bar{\alpha}^o(T) - \bar{\alpha}^o(T_{ref})\right) \tag{2.1–49}$$

Thus, equation (2.1–49) must be accounted for when making an adjustment for the definition temperature being different from the strain–free temperature. This adjustment is automatically made using the **MPAMOD** command.

Note that if $T_{ref} = T_o$, equation (2.1–49) is trivial.

Also, note that if $T=T_{ref}$, equation (2.1–49) is undefined. The values of T as used here are the temperatures defined on the **MPTEMP** command. Thus, when using the **MPAMOD** command, it is recommended to avoid defining a T value to be the same as $T=T_{ref}$ (to a tolerance of one degree). If a T value is the same as T_{ref}, and

– The T value is at either end of the input range, then the new α value is simply the same as the new α value of the nearest adjacent point.

– The T value is not at either end of the input range, then the new α value is the average of the two adjacent new α values.

2.2 Derivation of Structural Matrices

The principle of virtual work states that a virtual (very small) change of the internal strain energy must be offset by an identical change in external work due to the applied loads, or:

$$\delta U = \delta V \tag{2.2–1}$$

where:
U = strain energy (internal work) = $U_1 + U_2$
V = external work = $V_1 + V_2 + V_3$
δ = virtual operator

The virtual strain energy is:

$$\delta U_1 = \int_{vol} \{\delta\varepsilon\}^T \{\sigma\} \, d(vol) \tag{2.2–2}$$

where:
$\{\varepsilon\}$ = strain vector
$\{\sigma\}$ = stress vector
vol = volume of element

Continuing the derivation assuming linear materials and geometry, equations (2.1–1) and (2.2–2) are combined to give:

$$\delta U_1 = \int_{vol} \left(\{\delta\varepsilon\}^T [D] \{\varepsilon\} - \{\delta\varepsilon\}^T [D] \{\varepsilon^{th}\}\right) \, d(vol) \tag{2.2–3}$$

The strains may be related to the nodal displacements by:

$$\{\varepsilon\} = [B]\{u\} \tag{2.2–4}$$

where:
[B] = strain–displacement matrix, based on the element shape functions
{u} = nodal displacement vector

It will be assumed that all effects are in the global Cartesian system. Combining equation (2.2–4) with equation (2.2–3), and noting that {u} does not vary over the volume:

$$\delta U_1 = \{\delta u\}^T \int_{vol} [B]^T [D] [B] \, d(vol) \, \{u\}$$
$$- \{\delta u\}^T \int_{vol} [B]^T [D] \{\varepsilon^{th}\} \, d(vol) \tag{2.2–5}$$

Another form of virtual strain energy is when a surface moves against a distributed resistance, as in a foundation stiffness. This may be written as:

$$\delta U_2 = \int_{area_f} \{\delta w_n\}^T \{\sigma\} \, d(area_f) \tag{2.2–6}$$

where:

$\{w_n\}$ = motion normal to the surface

$\{\sigma\}$ = stress carried by the surface

$area_f$ = area of the distributed resistance

Both $\{w_n\}$ and $\{\sigma\}$ will usually have only one non–zero component. The point–wise normal displacement is related to the nodal displacements by:

$$\{w_n\} = [N_n] \{u\} \tag{2.2–7}$$

where: $[N_n]$ = matrix of shape functions for normal motions at the surface

The stress, $\{\sigma\}$, is

$$\{\sigma\} = k \{w_n\} \tag{2.2–8}$$

where: k = the foundation stiffness in units of force per length per unit area

Combining equations (2.2–6) thru (2.2–8), and assuming that k is constant over the area,

$$\delta U_2 = \{\delta u\}^T k \int_{area_f} [N_n]^T [N_n] \, d(area_f) \, \{u\} \tag{2.2–9}$$

Next, the external virtual work will be considered. The inertial effects will be studied first:

$$\delta V_1 = - \int_{vol} \{\delta w\}^T \frac{\{F^a\}}{vol} \, d(vol) \tag{2.2–10}$$

where: $\{w\}$ = vector of displacements of a general point

$\{F^a\}$ = acceleration (D'Alembert) force vector

According to Newton's second law:

$$\frac{\{F^a\}}{vol} = \varrho \frac{\partial^2}{\partial t^2} \{w\} \qquad (2.2–11)$$

where: ϱ = density (input as DENS on **MP** command)

t = time

The displacements within the element are related to the nodal displacements by:

$$\{w\} = [N] \{u\} \qquad (2.2–12)$$

where [N] = matrix of shape functions. Combining equations (2.2–10), (2.2–11), and (2.2–12) and assuming that ϱ is constant over the volume,

$$\delta V_1 = -\{\delta u\}^T \varrho \int_{vol} [N]^T [N]\, d(vol) \frac{\partial^2}{\partial t^2} \{u\} \qquad (2.2–13)$$

The pressure force vector formulation starts with:

$$\delta V_2 = \int_{area_p} \{\delta w_n\}^T \{P\} d(area_p) \qquad (2.2–14)$$

where: $\{P\}$ = the applied pressure vector (normally contains only one non–zero component)

$area_p$ = area over which pressure acts

Combining equations (2.2–12) and (2.2–14),

$$\delta V_2 = \{\delta u\}^T \int_{area_p} [N_n]^T \{P\}\, d(area_p) \qquad (2.2–15)$$

Unless otherwise noted, pressures are applied to the outside surface of each element and are normal to curved surfaces, if applicable.

Nodal forces applied to the element can be accounted for by:

$$\delta V_3 = \{\delta u\}^T \left\{F_e^{nd}\right\} \qquad (2.2–16)$$

where: $\left\{F_e^{nd}\right\}$ = nodal forces applied to the element

All material properties for stress analysis elements are evaluated at the average temperature of each element. Finally, equations (2.2–1), (2.2–5), (2.2–9), (2.2–13), (2.2–15) and (2.2–16) may be combined to give:

$$\begin{aligned}
&\{\delta u\}^T \int_{vol} [B]^T[D][B]\; d(vol)\;\; \{u\} - \{\delta u\}^T \int_{vol} [B]^T[D]\{e^{th}\}\; d(vol) \\
&\qquad + \{\delta u\}^T k \int_{area_f} [N_n]^T[N_n]\; d\left(area_f\right)\; \{u\} \qquad (2.2\text{–}17) \\
&= -\{\delta u\}^T \rho \int_{vol} [N]^T[N]\; d(vol)\, \frac{\partial^2}{\partial t^2}\; \{u\} \;+\; \{\delta u\}^T \int_{area_p} [N_n]^T\{P\}\; d(area_p) \;+\; \{\delta u\}^T \left\{F_e^{nd}\right\}
\end{aligned}$$

Noting that the $\{\delta u\}^T$ vector is a set of arbitrary virtual displacements common in all of the above terms, the condition required to satisfy equation (2.2–17) reduces to:

$$\left([K_e] + \left[K_e^f\right]\right) \{u\} - \left\{F_e^{th}\right\} = [M_e]\,\{\ddot{u}\} + \left\{F_e^{pr}\right\} + \left\{F_e^{nd}\right\} \qquad (2.2\text{–}18)$$

where:

$$[K_e] = \int_{vol} [B]^T[D][B]\; d(vol) = \text{element stiffness matrix}$$

$$\left[K_e^f\right] = k \int_{area_f} [N_n]^T[N_n]\; d\left(area_f\right) = \text{element foundation stiffness matrix}$$

$$\left\{F_e^{th}\right\} = \int_{vol} [B]^T[D]\{\varepsilon^{th}\}\; d(vol) = \text{element thermal load vector}$$

$$[M_e] = \rho \int_{vol} [N]^T[N]\; d(vol) = \text{element mass matrix}$$

$$\{\ddot{u}\} = \frac{\partial^2}{\partial t^2}\{u\} = \text{acceleration vector (such as gravity effects)}$$

$$\left\{F_e^{pr}\right\} = \int_{area_p} [N_n]^T\{P\}\; d\left(area_p\right) = \text{element pressure vector}$$

Equation (2.2–18) represents the equilibrium equation on a one element basis.

The above matrices and load vectors were developed as “consistent.” Other formulations are possible. For example, if only diagonal terms for the mass matrix are requested (**LUMPM**,ON), the matrix is called “lumped” (see Section 13.2). For most lumped mass matrices, the rotational degrees of freedom (DOFs) are removed. If the rotational DOFs are requested to be removed (**KEYOPT** commands with certain elements), the matrix or load vector is called “reduced.” Thus, use of the reduced pressure load vector does not generate moments as part of the pressure load vector. Use of the consistent pressure load vector can cause erroneous internal moments in a structure. An example of this would be a thin circular cylinder under internal pressure modelled with irregular shaped shell elements. As suggested by Figure 2.2–1, the consistent pressure loading generates an erroneous moment for two adjacent elements of dissimilar size.

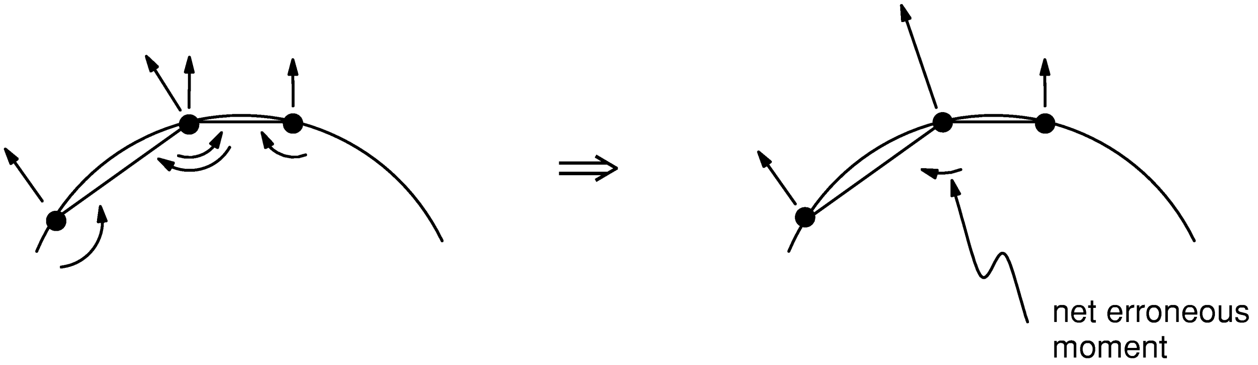


Figure 2.2–1 Effects of Consistent Pressure Loading

The “CENTROID, MASS, AND MASS MOMENT OF INERTIA” output does not use any mass matrices. Rather, it simply lumps the mass of each element at the center of that element. The element “centroid” is not the mass center but rather the origin of the element coordinate system:

$$X_c = \{N_o\}^T \{X_n\} \tag{2.2–19}$$

$$Y_c = \{N_o\}^T \{Y_n\} \tag{2.2–20}$$

$$Z_c = \{N_o\}^T \{Z_n\} \tag{2.2–21}$$

where:
X_c = output quantity XC, etc.
$\{N_o\}$ = vector of shape functions, evaluated at the origin of the element coordinate system
$\{X_n\}$ = vector of nodal locations.

As an example, consider a curved PLANE82 element in Figure 2.2–2.

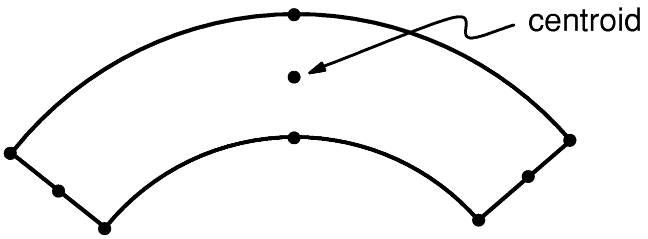


Figure 2.2–2 Centroid Location

2.3 Structural Strain and Stress Evaluations

The element integration point strains and stresses are computed by combining equations (2.1–1) and (2.2–4) to get:

$$\{\varepsilon^{el}\} = [B]\ \{u\} - \{\varepsilon^{th}\} \tag{2.3–1}$$

$$\{\sigma\} = [D]\ \{\varepsilon^{el}\} \tag{2.3–2}$$

where:
$\{\varepsilon^{el}\}$ = strains that cause stresses (output quantity EPEL)
$[B]$ = strain–displacement matrix evaluated at integration point
$\{u\}$ = nodal displacement vector
$\{\varepsilon^{th}\}$ = thermal strain vector
$\{\sigma\}$ = stress vector (output quantity S)
$[D]$ = elasticity matrix

Nodal and centroidal stresses are available from the integration point stresses as described in Section 13.6.

2.3.1 Combined Strains

The principal strains are calculated from the strain components by the cubic equation:

$$\begin{vmatrix} \varepsilon_x - \varepsilon_o & \frac{1}{2}\varepsilon_{xy} & \frac{1}{2}\varepsilon_{xz} \\ \frac{1}{2}\varepsilon_{xy} & \varepsilon_y - \varepsilon_o & \frac{1}{2}\varepsilon_{yz} \\ \frac{1}{2}\varepsilon_{xz} & \frac{1}{2}\varepsilon_{yz} & \varepsilon_z - \varepsilon_o \end{vmatrix} = 0 \tag{2.3–3}$$

where:
ε_o = principal strain (3 values)

The three principal strains are labeled ε_1, ε_2, and ε_3 (output quantities 1, 2, and 3 with strain items such as EPEL). The principal strains are ordered so that ε_1 is the most positive and ε_3 is the most negative.

The strain intensity ε_I (output quantity INT) is the largest of the absolute values of $\varepsilon_1 - \varepsilon_2$, $\varepsilon_2 - \varepsilon_3$, or $\varepsilon_3 - \varepsilon_1$. That is:

$$\varepsilon_I = \text{MAX}\left(|\varepsilon_1 - \varepsilon_2|, |\varepsilon_2 - \varepsilon_3|, |\varepsilon_3 - \varepsilon_1|\right) \tag{2.3–4}$$

The von Mises or equivalent strain ε_e (output quantity EQV) is computed as:

$$\varepsilon_e = \frac{1}{1+\nu'}\left(\frac{1}{2}\left[(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2\right]\right)^{\frac{1}{2}} \tag{2.3–5}$$

where: ν' = effective Poisson's ratio

$$= \begin{cases} \text{input as EFFNU on } \textbf{AVPRIN} \text{ command (POST1 only)} \\ \text{0.0 for all other cases} \end{cases}$$

2.3.2 Combined Stresses

The principal stresses (σ_1, σ_2, σ_3) are calculated from the stress components by the cubic equation:

$$\begin{vmatrix} \sigma_x - \sigma_o & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y - \sigma_o & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z - \sigma_o \end{vmatrix} = 0 \tag{2.3–6}$$

where: σ_o = principal stress (3 values)

The three principal stresses are labeled σ_1, σ_2, and σ_3 (output quantities S1, S2, and S3). The principal stresses are ordered so that σ_1 is the most positive and σ_3 is the most negative.

The stress intensity σ_I (output quantity SINT) is the largest of the absolute values of $\sigma_1 - \sigma_2$, $\sigma_2 - \sigma_3$, or $\sigma_3 - \sigma_1$. That is:

$$\sigma_I = \text{MAX}\left(|\sigma_1 - \sigma_2|, |\sigma_2 - \sigma_3|, |\sigma_3 - \sigma_1|\right) \tag{2.3–7}$$

The von Mises or equivalent stress σ_e (output quantity SEQV) is computed as:

$$\sigma_e = \left(\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]\right)^{\frac{1}{2}} \tag{2.3–8}$$

When $\nu' = \nu$ (input as PRXY or NUXY on **MP** command), the equivalent stress is related to the equivalent strain through

$$\sigma_e = E\, \epsilon_e \tag{2.3–9}$$

where: E = Young's modulus (input as EX on **MP** command)

2.3.3 Surface Stresses

Surface stress output may be requested on "free" faces of 2–D and 3–D elements. "Free" means not connected to other elements as well as not having any imposed displacements or nodal forces normal to the surface. The following steps are executed at each surface Gauss point to evaluate the surface stresses. The integration points used are the same as for an applied pressure to that surface.

1. Compute the in–plane strains of the surface at an integration point using:

$$\{\varepsilon'\} = [B']\{u'\} - \left\{(\varepsilon^{th'})\right\} \tag{2.3–10}$$

   Hence, ε'_x, ε'_y and ε'_{xy} are known. The prime (′) represents the surface coordinate system, with z being normal to the surface.

2. A each point, set:

$$\sigma_z{}' = -P \tag{2.3–11}$$

$$\sigma'_{xz} = 0 \tag{2.3–12}$$

$$\sigma'_{yz} = 0 \tag{2.3–13}$$

   where P is the applied pressure. Equations (2.3–12) and (2.3–13) are valid, as the surface for which stresses are computed is presumed to be a free surface.

3. At each point, use the six material property equations represented by:

$$\{\sigma'\} = [D']\, \{\varepsilon'\} \tag{2.3–14}$$

   to compute the remaining strain and stress components (ε'_z, ε'_{xz}, ε'_{yz}, σ'_x, σ'_y, and σ'_{xy}).

4. Repeat and average the results across all integration points.

2.3.4 Shell Element Output

For elastic shell elements, the forces and moments per unit length (using shell nomenclature) are computed as:

$$T_x = \frac{t\left(\sigma_{x,top} + 4\sigma_{x,mid} + \sigma_{x,bot}\right)}{6} \qquad (2.3\text{–}15)$$

$$T_y = \frac{t\left(\sigma_{y,top} + 4\sigma_{y,mid} + \sigma_{y,bot}\right)}{6} \qquad (2.3\text{–}16)$$

$$T_{xy} = \frac{t\left(\sigma_{xy,top} + 4\sigma_{xy,mid} + \sigma_{xy,bot}\right)}{6} \qquad (2.3\text{–}17)$$

$$M_x = \frac{t^2\left(\sigma_{x,top} - \sigma_{x,bot}\right)}{12} \qquad (2.3\text{–}18)$$

$$M_y = \frac{t^2\left(\sigma_{y,top} - \sigma_{y,bot}\right)}{12} \qquad (2.3\text{–}19)$$

$$M_{xy} = \frac{t^2\left(\sigma_{xy,top} - \sigma_{xy,bot}\right)}{12} \qquad (2.3\text{–}20)$$

$$N_x = \frac{t\left(\sigma_{xz,top} + 4\sigma_{xz,mid} + \sigma_{xz,bot}\right)}{6} \qquad (2.3\text{–}21)$$

$$N_y = \frac{t\left(\sigma_{yz,top} + 4\sigma_{yz,mid} + \sigma_{yz,bot}\right)}{6} \qquad (2.3\text{–}22)$$

where: T_x, T_y, T_{xy} = in–plane forces per unit length (output quantities TX, TY, and TXY)

M_x, M_y, M_{xy} = bending moments per unit length (output quantities MX, MY, and MXY)

N_x, N_y = transverse shear forces per unit length (output quantities NX and NY)

t = thickness at midpoint of element, computed normal to center plane

σ_x, etc. = direct stress (output quantity SX, etc.)

σ_{xy}, etc. = shear stress (output quantity SXY, etc.)

It should be noted that the shell nomenclature and the nodal moment conventions are in apparent conflict with each other. For example, a cantilever beam located along the x axis and consisting of shell elements in the x–y plane that deforms in the z direction under a pure bending load with coupled nodes at the free end, has the following relationship:

$$M_x b = F_{MY} \tag{2.3–23}$$

where: b = width of beam

F_{MY} = input as *VALUE* on **F** command with *Lab* = MY (not MX) applied at the free end

The shape functions of the shell element result in constant transverse strains and stresses through the thickness. Some shell elements adjust these values so that they will peak at the midsurface with 3/2 of the constant value and be zero at both surfaces, as noted in the element discussions in Chapter 14.

The thru–thickness stress (σ_z) is set equal to the negative of the applied pressure at the surfaces of the shell elements, and linearly interpolated in between.

Chapter 3
Structures with Geometric Nonlinearities

ANSYS Theory Reference
Chapter 3 – Table of Contents

3.0 Introduction to Geometric Nonlinearities

This chapter discusses the different geometrically nonlinear options within the ANSYS program including large strain, large deflection, stress stiffening, and spin softening. Only elements with displacements degrees of freedom (DOFs) are applicable. Not included in this section are the multi–status elements (such as LINK10, CONTAC12, COMBIN40, and CONTAC52, discussed in Chapter 14) and the eigenvalue buckling capability (discussed in Section 17.5).

Geometric nonlinearities refer to the nonlinearities in the structure or component due to the changing geometry as it deflects. That is, the stiffness [K] is a function of the displacements {u}. The stiffness changes because the shape changes and/or the material rotates. The program can account for four types of geometric nonlinearities:

1. *Large strain* assumes that the strains are no longer infinitesmal (they are finite). Shape changes (e.g. area, thickness, etc.) are also accounted for. Deflections and rotations may be arbitrarily large.

2. *Large rotation* assumes that the rotations are large but the mechanical strains (those that cause stresses) are evaluated using linearized expressions. The structure is assumed not to change shape except for rigid body motions. The elements of this class refer to the original configuration.

3. *Stress stiffening* assumes that both strains and rotations are small. A 1st order approximation to the rotations is used to capture some nonlinear rotation effects.

4. *Spin softening* also assumes that both strains and rotations are small. This option accounts for the radial motion of a body's structural mass as it is subjected to an angular velocity. Hence it is a type of large deflection but small rotation approximation.

All elements support the spin softening capability, while only some of the elements support the other options. Table 3.0–1 lists the elements that have large strain, large deflection and/or stress stiffening capability.

Table 3.0–1 Elements Having Nonlinear Geometric Capability

Label	Name	NLGEOM=1	Stress Stiffening
LINK1	2–D Spar (or Truss)	LR	√
PLANE2	2–D 6–Node Triangular Structural Solid	LS	√
BEAM3	2–D Elastic Beam	LR	√
BEAM4	3–D Elastic Beam	LR	√
SOLID5	3–D Coupled–Field Solid	–	√
COMBIN7	Revolute Joint	LR	√
LINK8	3–D Spar (or Truss)	LR	√
LINK10	Tension–Only (Chain)	LR	√
LINK11	Linear Actuator	LR	√
PLANE13	2–D Coupled–Field Solid		√
COMBIN14	Spring–Damper	LR	√
PIPE16	Elastic Straight Pipe	LR	√
PIPE17	Elastic Pipe Tee	LR	√
PIPE18	Elastic Curved Pipe (Elbow)	LR	–
PIPE20	Plastic Straight Pipe	LR	√
MASS21	Structural Mass	LR	
BEAM23	2–D Plastic Beam	LR	√
BEAM24	3–D Thin–Walled Plastic Beam	LR	√
PLANE25	4–Node Axisymmetric–Harmonic Structural Solid	–	√
CONTAC26	2–D Point–to–Ground Contact	SC	–
SHELL28	Shear/Twist Panel	–	√
COMBIN37	Control		–
FLUID38	Dynamic Fluid Coupling		–
COMBIN39	Nonlinear Spring	LR	√
COMBIN40	Combination		–
SHELL41	Membrane Shell	LR	√
PLANE42	2–D Structural Solid	LS	√
SHELL43	Plastic Shell	LS	√
BEAM44	3–D Elastic Tapered Unsymmetric Beam	LR	√
SOLID45	3–D Structural Solid	LS	√
SOLID46	3–D Layered Structural Shell	LR	√
CONTAC48	2–D Point–to–Surface Contact	SC	–
CONTAC49	3–D Point–to–Surface Contact	SC	–
MATRIX50	Substructure Matrix	LR	–
SHELL51	Axisymmetric Structural Shell	LR	√

Label	Name	NLGEOM=1	Stress Stiffening
BEAM54	2–D Elastic Tapered Unsymmetric Beam	LR	√
HYPER56	2–D Mixed U–P Hyperelastic Solid	LS	AN
HYPER58	3–D Mixed U–P Hyperelastic Solid	LS	AN
PIPE59	Immersed Pipe or Cable	LR	√
PIPE60	Plastic Curved Pipe (Elbow)	LR	–
SHELL61	Axisymmetric–Harmonic Structural Shell	–	√
SOLID62	3–D Coupled Magnetic–Structural Solid	LS	√
SHELL63	Elastic Shell	LR	√
SOLID64	3–D Anisotropic Structural Solid	LR	√
SOLID65	3–D Reinforced Concrete Solid	LS	√
SOLID72	3–D Tet with Rotations	–	√
SOLID73	3–D Solid with Rotations	–	√
HYPER74	2–D 8–Node Mixed U–P Hyperelastic Solid	LS	AN
PLANE82	2–D 8–Node Structural Solid	LS	√
PLANE83	8–Node Axisymmetric–Harmonic Structural Solid	–	√
HYPER84	2–D Hyperelastic Solid	LS	AN
HYPER86	3–D Hyperelastic Solid	LS	AN
VISCO88	2–D 8–Node Viscoelastic Solid	LS	√
VISCO89	3–D 20–Node Viscoelastic Solid	LS	√
SHELL91	16–Layer Structural Shell	LS	√
SOLID92	3–D 10–Node Tetrahedral Structural Solid	LS	√
SHELL93	8–Node Structural Solid	LS	√
SOLID95	20–Node Structural Solid	LS	√
SOLID98	Tetrahedral Coupled–Field Solid	–	√
SHELL99	100–Layer Structural Shell	LR	√
VISCO106	2–D Large Strain Solid	LS	AN
VISCO107	3–D Large Strain Solid	LS	AN
VISCO108	2–D 8–Node Large Strain Solid	LS	AN
SHELL143	Plastic Shell	LR	√
SHELL150	Structural Shell p–element	–	–
SURF151	2–D Thermal Surface	SC	–
SURF152	3–D Thermal Surface	SC	–
SURF153	2–D Structural Surface	SC	√
SURF154	2–D Structural Surface	SC	√
HYPER158	3–D Hyperelastic Tetrahedral Solid	LS	AN
TARGE169	2–D Target	SC	–
TARGE170	3–D Target	SC	–

Label	Name	NLGEOM=1	Stress Stiffening
CONTA171	2–D 2–Node Surface–to–Surface	SC	–
CONTA172	2–D 3–Node Surface–to–Surface	SC	–
CONTA173	3–D General, 4–Node Contact	SC	–
CONTA174	3–D General, 8–Node Contact	SC	–
LINK180	3–D Finite Strain Spar (or Truss)	LS	AB
SHELL181	4–Node Structural Shell	LS	AB
PLANE182	Continuum 2–D Structural Solid, 4–Node	LS	AB
PLANE183	2–D 8–Node Structural Solid	LS	AB
SOLID185	3–D Structural Solid	LS	AB
SOLID186	20–Node Structural Solid	LS	AB
SOLID187	3–D 10–Node Tetrahedral Structural Solid	LS	AB
BEAM188	3–D Linear Beam	LS	AB
BEAM189	3–D Quadratic Beam	LS	AB

Codes associated with NLGEOM=1:

LS = large strain element

LR = Element that can do a rigid body roation. The NLGEOM=1 provides only a rigid body rotation. Strains, if any, are linear.

SC = surface or contact element. The element follows the underlying element.

Codes associated with stress stiffening:

√ = has option of computing stress stiffness matrix

AN = if NLGEOM=1, stress stiffening is automatically included. However, the element is not capable of linear buckling using ANTYPE,BUCKLE.

AB = if NLGEOM = 1, stress stiffening is automatically included, and the element is also capable of linear buckling using ANTYPE,BUCKLE.

3.1 Large Strain

When the strains in a material exceed more than a few percent, the changing geometry due to this deformation can no longer be neglected. Analyses which include this effect are called large strain, or finite strain, analyses. A large strain analysis is performed in a static (**ANTYPE**,STATIC) or transient (**ANTYPE**,TRANS) analysis with **NLGEOM**,ON when the appropriate element type(s) is used, see Table 3.0–1.

The remainder of this section addresses the large strain formulation for elastic–plastic elements. These elements use a hypoelastic formulation so that they are restricted to small elastic strains (but allow for arbitrarily large plastic strains). Section 4.5 addresses the large strain formulation for hyperelastic elements, which allow arbitrarily large elastic strains.

3.1.1 Theory

The theory of large strain computations can be addressed by defining a few basic physical quantities (motion and deformation) and the corresponding mathematical relationship. The applied loads acting on a body make it move from one position to another. This motion can be defined by studying a position vector in the “deformed” and “undeformed” configuration. Say the position vectors in the “deformed” and “undeformed” state are represented by {x} and {X} respectively, then the motion (displacement) vector {u} is computed by (see Figure 3.1–1):

$$\{u\} = \{x\} - \{X\} \tag{3.1–1}$$

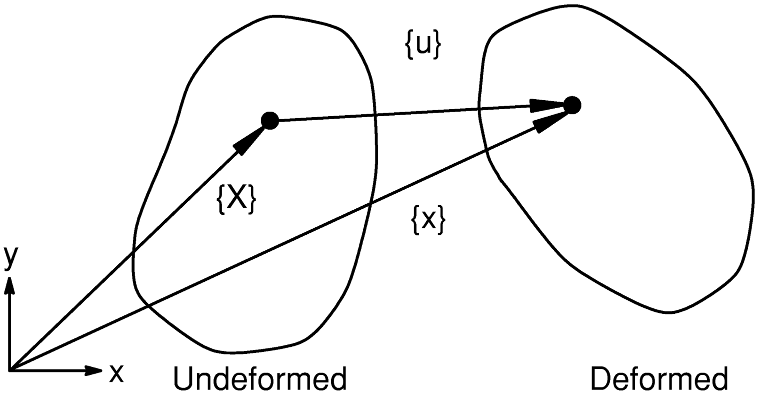


Figure 3.1–1 Position Vectors and Motion of a Deforming Body

The deformation gradient is defined as:

$$[F] = \frac{\partial\{x\}}{\partial\{X\}} \tag{3.1–2}$$

which can be written in terms of the displacement of the point via equation (3.1–1) as:

$$[F] = [I] + \frac{\partial\{u\}}{\partial\{X\}} \tag{3.1–3}$$

where: $[I]$ = identity matrix

The information contained in the deformation gradient [F] includes the volume change, the rotation and the shape change of the deforming body. The volume change at a point is

$$\frac{dV}{dV_o} = \det\ [F] \tag{3.1–4}$$

where:
V_o = original volume
V = current volume
$\det[\cdot]$ = determinant of the matrix

The deformation gradient can be separated into a rotation and a shape change using the right polar decomposition theorem:

$$[F] = [R]\ [U] \tag{3.1–5}$$

where:
$[R]$ = rotation matrix ($[R]^T[R] = [I]$)
$[U]$ = right stretch (shape change) matrix

Once the stretch matrix is known, a logarithmic or Hencky strain measure is defined as:

$$[\varepsilon] = \ell n\ [U] \tag{3.1–6}$$

($[\varepsilon]$ is in tensor (matrix) form here, as opposed to the usual vector form $\{\varepsilon\}$). Since [U] is a 2nd order tensor (matrix), equation (3.1–6) is determined through the spectral decomposition of [U]:

$$[\varepsilon] = \sum_{i=1}^{3} \ell n\ (\lambda_i)\ \{e_i\}\ \{e_i\}^T \tag{3.1–7}$$

where:
λ_i = eigenvalues of [U] (principal stretches)
$\{e_i\}$ = eigenvectors of [U] (principal directions)

The polar decomposition theorem (equation (3.1–5)) extracts a rotation [R] that represents the average rotation of the material at a point. Material lines initially orthogonal will not, in general, be orthogonal after deformation (because of shearing), see Figure 3.1–2. The polar decomposition of this deformation, however, will indicate that they will remain orthogonal (lines x–y′ in Figure 3.1–2). For this reason, non–isotropic behavior (e.g. orthotropic elasticity or kinematic hardening plasticity) should be used with care with large strains, especially if large shearing deformation occurs.

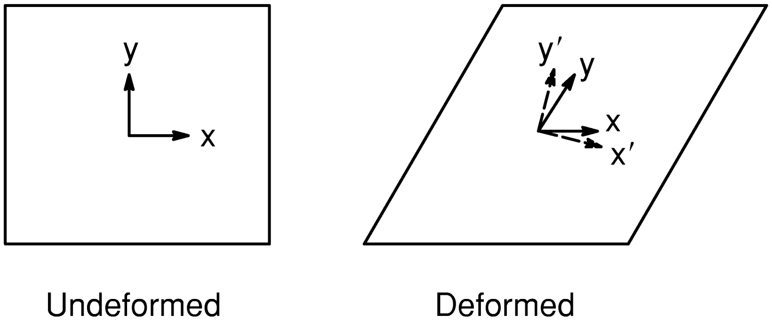


Figure 3.1–2 Polar Decomposition of a Shearing Deformation

Implementation

Computationally, the evaluation of equation (3.1–6) is performed by one of two methods using the incremental approximation (since, in an elastic–plastic analysis, we are using an incremental solution procedure):

$$[\varepsilon] = \int d[\varepsilon] \approx \sum [\Delta\varepsilon_n] \tag{3.1–8}$$

with

$$[\Delta\varepsilon_n] = \ell n\ [\Delta U_n] \tag{3.1–9}$$

where $[\Delta U_n]$ is the increment of the stretch matrix computed from the incremental deformation gradient:

$$[\Delta F_n] = [\Delta R_n]\ [\Delta U_n] \tag{3.1–10}$$

where $[\Delta F_n]$ is:

$$[\Delta F_n] = [F_n]\ [F_{n-1}]^{-1} \tag{3.1–11}$$

$[F_n]$ is the deformation gradient at the current time step and $[F_{n-1}]$ is at the previous time step. Two methods are employed for evaluating equation (3.1–9).

Method 1 (Weber, et al.(127)) uses the idea of equation (3.1–7):

$$[\Delta\varepsilon_n] = \sum_{i=1}^{3} \ell n\ (\lambda_i)\ \{e_i\}\ \{e_i\}^T \tag{3.1–12}$$

where λ_i and $\{e_i\}$ are the eigenvalue and eigenvector for the ith principal stretch increment of the incremental stretch matrix $[\Delta U_n]$, equation (3.1–10). This is the method employed by the large strain solids VISCO106, VISCO107 and VISCO108.

Method 2 (Hughes(156)) uses the approximate 2nd order accurate calculation:

$$[\Delta\varepsilon_n] = [R_{1/2}]^T[\Delta\tilde{\varepsilon}_n][R_{1/2}] \tag{3.1–13}$$

where $[R_{1/2}]$ is the rotation matrix computed from the polar decomposition of the deformation gradient evaluated at the midpoint configuration:

$$[F_{1/2}] = [R_{1/2}][U_{1/2}] \tag{3.1–14}$$

where $[F_{1/2}]$ is (using equation (3.1–3)):

$$[F_{1/2}] = [I] + \frac{\partial\{u_{1/2}\}}{\partial\{X\}} \tag{3.1–15}$$

and the midpoint displacement is:

$$\{u_{1/2}\} = \frac{1}{2}\left(\{u_n\} + \{u_{n-1}\}\right) \tag{3.1–16}$$

$\{u_n\}$ is the current displacement and $\{u_{n-1}\}$ is the displacement at the previous time step. $[\Delta\varepsilon_n]$ is the “rotation–neutralized” strain increment over the time step. The strain increment $[\Delta\tilde{\varepsilon}_n]$ is also computed from the midpoint configuration:

$$\{\Delta\tilde{\varepsilon}_n\} = [B_{1/2}]\{\Delta u_n\} \tag{3.1–17}$$

$\{\Delta u_n\}$ is the displacement increment over the time step and $[B_{1/2}]$ is the strain–displacement relationship evaluated at the midpoint geometry:

$$\{X_{1/2}\} = \frac{1}{2}\left(\{X_n\} + \{X_{n-1}\}\right) \tag{3.1–18}$$

This method is an excellent approximation to the logarthmic strain if the strain steps are less than ~10%. This method is used by the standard 2–D and 3–D solid and shell elements.

The computed strain increment $[\Delta\varepsilon_n]$ (or equivalently $\{\Delta\varepsilon_n\}$) can then be added to the previous strain $\{\varepsilon_{n-1}\}$ to obtain the current total Hencky strain:

$$\{\varepsilon_n\} = \{\varepsilon_{n-1}\} + \{\Delta\varepsilon_n\} \tag{3.1–19}$$

This strain can then be used in the stress updating procedures, see Sections 4.1 and 4.2 for discussions of the rate–independent and rate–dependent procedures respectively.

3.1.2 Definition of Thermal Strains

According to Callen(243), the coefficient of thermal expansion is defined as the fractional increase in the length per unit increase in the temperature. Mathematically,

$$\alpha = \frac{1}{\ell}\frac{d\ell}{dT} \tag{3.1–20}$$

where:
- α = coefficient of thermal expansion
- ℓ = current length
- T = temperature

Rearranging equation (3.1–20) gives:

$$\frac{d\ell}{\ell} = \alpha\ dT \tag{3.1–21}$$

On the other hand, the logarithmic strain is defined as:

$$\epsilon^{\ell} = \ell n\left(\frac{\ell}{\ell_o}\right) \tag{3.1–22}$$

where:
- ε^{ℓ} = logrithmic strain
- ℓ_o = initial length

Differential of equation (3.1–22) yields:

$$d\epsilon^{\ell} = \frac{d\ell}{\ell} \tag{3.1–23}$$

Comparison of equations (3.1–21) and (3.1–23) gives:

$$d\epsilon^{\ell} = \alpha dT \tag{3.1–24}$$

Integration of equation (3.1–24) yields:

$$\epsilon^{\ell} - \epsilon_o^{\ell} = \alpha \; (T - T_o) \qquad (3.1\text{–}25)$$

where: ϵ_o^{ℓ} = initial (reference) strain at temperature T_o

T_o = reference temperature

In the absence of initial strain ($\epsilon_o^{\ell} = 0$), then equation (3.1–25) reduces to:

$$\epsilon^{\ell} = \alpha \; (T - T_o) \qquad (3.1\text{–}26)$$

The thermal strain corresponds to the logrithmic strain. As an example problem, consider a line element of a material with a constant coefficient of thermal expansion α. If the length of the line is ℓ_o at temperature T_o, then the length after the temperature increases to T is:

$$\ell = \ell_o \exp \epsilon^{\ell} = \ell_o \exp \; [\alpha \; (T - T_o)] \qquad (3.1\text{–}27)$$

Now if one interpreted the thermal strain as the engineering (or nominal) strain, then the final length would be different.

$$\epsilon^{e} = \alpha \; (T - T_o) \qquad (3.1\text{–}28)$$

where: ε^{e} = engineering strain

The final length is then:

$$\ell = \ell_o \; (1 + \epsilon^{e}) = \ell_o \; [1 + \alpha \; (T - T_o)] \qquad (3.1\text{–}29)$$

However, the difference should be very small as long as:

$$\alpha \mid T - T_o \mid << 1 \qquad (3.1\text{–}30)$$

because

$$\exp \; [\alpha \; (T - T_o)] \approx 1 + \alpha \; (T - T_o) \qquad (3.1\text{–}31)$$

3.1.3 Element Formulation

The element matrices and load vectors are derived using an updated Lagrangian formulation. This produces equations of the form:

$$[\overline{K}_i] \; \Delta u_i = \{F^{app}\} - \left\{F_i^{nr}\right\} \qquad (3.1\text{–}32)$$

where the tangent matrix $[\overline{K}_i]$ has the form:

$$[\overline{K}_i] = [K_i] + [S_i] \tag{3.1–33}$$

$[K_i]$ is the usual stiffness matrix:

$$[K_i] = \int [B_i]^T[D_i][B_i] \; d(vol) \tag{3.1–34}$$

$[B_i]$ is the strain–displacement matrix in terms of the current geometry $\{X_n\}$ and $[D_i]$ is the current stress–strain matrix.

$[S_i]$ is the stress stiffness (or geometric stiffness) contribution, written symbolically as:

$$[S_i] = \int [G_i]^T[\tau_i][G_i] \; d(vol) \tag{3.1–35}$$

where $[G_i]$ is a matrix of shape function derivatives and $[\tau_i]$ is a matrix of the current Cauchy (true) stresses $\{\sigma_i\}$ in the global Cartesian system. The Newton–Raphson restoring force is:

$$\left[F_i^{nr}\right] = \int [B_i]^T\{\sigma_i\} \; d(vol) \tag{3.1–36}$$

All of the plane stress and shell elements account for the thickness changes due to the out–of–plane strain ε_z (Hughes and Carnoy(157)). Shells, however, do not update their reference plane (as might be required in a large strain out–of–plane bending deformation); the thickness change is assumed to be constant through the thickness.

3.1.4 Applicable Input

NLGEOM,ON activates large strain computations in those elements which support it. **SSTIF**,ON activates the stress–stiffening contribution to the tangent matrix.

3.1.5 Applicable Output

For elements which have large strain capability, stresses (output quantity S) are true (Cauchy) stresses in the rotated element coordinate system (the element coordinate system follows the material as it rotates). Strains (output quantities EPEL, EPPL, etc.) are the logarithmic or Hencky strains, also in the rotated element coordinate system.

An exception is for the hyperelastic elements. For these elements, stress and strain components maintain their original orientations and some of these elements use other strain measures.

3.2 Large Rotation

If the rotations are large but the mechanical strains (those that cause stresses) are small, then a large rotation procedure can be used. A large rotation analysis is performed in a static (**ANTYPE**,STATIC) or transient (**ANTYPE**,TRANS) analysis with **NLGEOM**,ON when the appropriate element type is used, see Table 3.0–1. Note that all large strain elements also support this capability, since both options account for the large rotations and for small strains, the logarithmic strain measure and the engineering strain measure coincide.

3.2.1 Theory

Section 3.1 presented the theory for general motion of a material point. Large rotation theory follows a similar development, except that the logarithmic strain measure (equation (3.1–6)) is replaced by the Biot, or small (engineering) strain measure:

$$[\varepsilon] = [U] - [I] \qquad (3.2–1)$$

where: $[U]$ = stretch matrix

$[I]$ = 3 x 3 identity matrix

3.2.2 Implementation

A corotational (or convected coordinate) approach is used in solving large rotation/small strain problems (Rankin and Brogan(66)). The nonlinearities are contained in the strain–displacement relationship which for this algorithm takes on the special form:

$$[B_n] = [B_v]\,[T_n] \qquad (3.2–2)$$

where: $[B_v]$ = usual small strain–displacement relationship in the original (virgin) element coordinate system

$[T_n]$ = orthogonal transformation relating the original element coordinates to the convected (or rotated) element coordinates

The convected element coordinate frame differs from the original element coordinate frame by the amount of rigid body rotation. Hence $[T_n]$ is computed by separating the rigid body rotation from the total deformation $\{u_n\}$ using the polar decomposition theorem, equation (3.1–5). From equation (3.2–2), the element tangent stiffness matrix has the form:

$$[K_e] = \int_{vol} [T_n]^T \; [B_v]^T \; [D] \; [B_v] \; [T_n] \; d(vol) \tag{3.2–3}$$

and the element restoring force is:

$$\left\{F_e^{nr}\right\} = \int_{vol} [T_n]^T \; [B_v]^T \; [D] \; \left\{\epsilon_n^{el}\right\} \; d(vol) \tag{3.2–4}$$

where the elastic strain is computed from:

$$\left\{\epsilon_n^{el}\right\} = [B_v]\left\{u_n^d\right\} \tag{3.2–5}$$

$\left\{u_n^d\right\}$ is the element deformation which causes straining as described in a subsequent subsection.

The large rotation process can be summarized as a three step process for each element:

1. Determine the updated transformation matrix $[T_n]$ for the element.
2. Extract the deformational displacement $\left\{u_n^d\right\}$ from the total element displacement $\{u_n\}$ for computing the stresses as well as the restoring force $\left\{F_e^{nr}\right\}$.
3. After the rotational increments in $\{\Delta u\}$ are computed, update the node rotations appropriately. All three steps require the concept of a rotational pseudovector in order to be efficiently implemented.

3.2.3 Pseudovector Representation

Finite rotations, unlike displacements, are not linear vectors and cannot be treated as such. Thus the rotational components of multiple rotations cannot simply be added to form the final rotation. For that reason the rotation vector is referred to as a pseudovector. The following decribes the formation of the rotational pseudovector and the associated transformation matrix needed to characterize the large rotation motion.

Argyris(67) defines the rotational pseudovector as:

$$\{\theta\} = \left\{\theta_1 \; \theta_2 \; \theta_3\right\}^T = \theta \; \{e\} \tag{3.2–6}$$

where: $\theta_1, \; \theta_2, \; \theta_3$ = rotational components

$$\theta = \text{magnitude of rotation}$$

$$= \left(\theta_1^{\;2} + \theta_2^{\;2} + \theta_3^{\;2}\right)^{\frac{1}{2}}$$

$$\{e\} = \text{unit vector along axis of rotation}$$

It is convenient to normalize the pseudovector, such that:

$$\{\varpi\} = \omega \; \{e\} \tag{3.2–7}$$

where $\{\varpi\}$ is also a pseudovector with magnitude ω and components ω_1, ω_2, and ω_3. Adopting the normalization of Rankin and Brogan(66),

$$\omega = 2 \sin\frac{\theta}{2} \tag{3.2–8}$$

establishes the relationship between θ and ω. Combining (3.2–7) and (3.2–8),

$$\{\varpi\} = 2 \sin\frac{\theta}{2} \; \{e\} \tag{3.2–9}$$

Following Rankin and Brogan(66), the transformation matrix associated with rotation $\{\varpi\}$ (or $\{\theta\}$) is given by:

$$[T] = [I] + \sqrt{1 - (\omega/2)^2} \; [\Omega] + \frac{1}{2} \, [\Omega]^2 \tag{3.2–10}$$

where [I] is the 3 x 3 identity matrix and $[\Omega]$ is the skew–symmetric matrix representation of $\{\varpi\}$, namely,

$$[\Omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

Thus, once a pseudovector is formed and normalized, the transformation connected to the finite rotation is determined from equation (3.2–10). Conversely, knowing the components of [T], $[\Omega]$ and $\{\varpi\}$ can be determined from:

$$[\Omega] = \frac{1}{\sqrt{1 + \gamma}} \left([T] - [T]^T\right) \tag{3.2–11}$$

where γ is the trace of [T]:

$$\gamma = T_{1,1} + T_{2,2} + T_{3,3} \tag{3.2–12}$$

Turning attention to pseudovector updating, the goal is to determine the present rotation vector $\{\theta_{n+1}\}$ from the previous $\{\theta_n\}$ and $\{\Delta\theta\}$, in which $\{\Delta\theta\}$ represents the rotational increment from $\{\theta_n\}$ to $\{\theta_{n+1}\}$. Argyris(67) derives a method of pseudovector updating that uses the mathematics of quarternions. A quarternion is defined as the sum of a scalar and a vector and is written as:

$$< q > \; = \; a + \{b\} \tag{3.2–13}$$

where the brackets (<>) denote a quarternion. The scalar and vector parts of <q> are a and {b}, respectively.

Two useful quarternion operations are multiplication and norm calculations. The norm of <q> is expressed by:

$$|q| \; = \; a^2 + b_1{}^2 + b_2{}^2 + b_3{}^2 \tag{3.2–14}$$

where b_1, b_2, and b_3 are the components of {b}. If |q| = 1, then |q| is termed a unit quarternion. Consider two quarternions given by:

$$< q_1 > \; = \; a + \{b\} \tag{3.2–15}$$

$$< q_2 > \; = \; c + \{d\} \tag{3.2–16}$$

Multiplication is defined as:

$$\begin{aligned} < q_{21} > \; &= \; < q_2 > < q_1 > \\ &= \; ac - \{b\}^T\{d\} + c\{b\} + a\{d\} - \{b\} \times \{d\} \end{aligned} \tag{3.2–17}$$

It follows that scalar and vector components of $<q_{21}>$ can be written as:

$$S(q_{21}) \; = \; ac - \{b\}^T\{d\} \tag{3.2–18}$$

$$\{V(q_{21})\} \; = \; c\{b\} + a\{d\} - \{b\} \times \{d\} \tag{3.2–19}$$

In Argyris(67), the rotational pseudovector is presented in the form of a quarternion, in order to circumvent any limitation on the amount of rotation:

$$< q > \; = \; \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\,\{e\} \tag{3.2–20}$$

It is clear that <q> above is a unit quarternion. By equation (3.2–9), the above can be rewritten as:

$$< q > \; = \; \cos\frac{\theta}{2} \; + \; \frac{1}{2}\{\omega\} \qquad (3.2\text{–}21)$$

The pseudovector updating considers sequential rotations $\{\theta_n\}$ and $\{\Delta\theta\}$ in the form:

$$< q_n > \; = \; \cos\frac{\theta_n}{2} \; + \; \frac{1}{2}\{\omega_n\} \qquad (3.2\text{–}22)$$

$$< \Delta q > \; = \; \cos\frac{\Delta\theta}{2} \; + \; \frac{1}{2}\{\Delta\omega\} \qquad (3.2\text{–}23)$$

where $\{\omega_n\}$ is the normalized $\{\theta_n\}$ and $\{\Delta\omega_n\}$ is the normalized $\{\Delta\theta_n\}$, see equation (3.2–9).

Updating to form $<q_{n+1}>$ uses the multiplication template of equation (3.2–17). Thus,

$$\begin{aligned} < q_{n+1} > \; &= \; < \Delta q > < q_n > \\ &= \cos\frac{\theta_n}{2}\cos\frac{\Delta\theta}{2} - \frac{1}{4}\{\omega_n\}^T\{\Delta\omega\} \qquad (3.2\text{–}24) \\ &+ \frac{1}{2}\cos\frac{\Delta\theta}{2}\{\omega_n\} + \frac{1}{2}\cos\frac{\theta_n}{2}\{\Delta\omega\} - \frac{1}{4}\{\omega_n\}\times\{\Delta\omega\} \end{aligned}$$

Since both $<q_n>$ and $<\Delta q>$ are unit quarternions, their product is also a unit quarternion. The pseudovector for the updated quarternion is (from equation (3.2–21)):

$$< q_{n+1} > \; = \; \cos\frac{\theta_{n+1}}{2} \; + \; \frac{1}{2}\{\omega_{n+1}\} \qquad (3.2\text{–}25)$$

or using the definition of equation (3.2.8):

$$< q_{n+1} > \; = \; \cos\frac{\theta_{n+1}}{2} \; + \; \sin\frac{\theta_{n+1}}{2}\{e_{n+1}\} \qquad (3.2\text{–}26)$$

Equating scalar and vector parts between equations (3.2–24) and (3.2–25) leads to:

$$S(q_{n+1}) \; = \; \cos\frac{\theta_{n+1}}{2} \; = \; \cos\frac{\theta_n}{2}\cos\frac{\Delta\theta_n}{2} - \frac{1}{4}\{\omega_n\}^T\{\Delta\omega\} \qquad (3.2\text{–}27)$$

$$\{V(q_{n+1})\} = \frac{1}{2}\{\omega_{n+1}\} = \frac{1}{2}\cos\frac{\Delta\theta}{2}\{\omega_n\} + \frac{1}{2}\cos\frac{\theta_n}{2}\{\Delta\omega\} - \frac{1}{4}\{\omega_n\}\times\{\Delta\omega\} \qquad (3.2\text{–}28)$$

t is clear that θ_{n+1} can be extracted from equation (3.2–27), while the components of $\{\omega_{n+1}\}$ are available from equation (3.2–28).

It is important to mention that θ is restricted to the range $\pm\ \pi$. This restriction is due to the nonuniqueness of the sine term of equation (3.2–26) in the vicinity of $\theta = \pi$ (i.e., $\sin\ (\pi + \phi)/2 = \sin\ (\pi - \phi)/2$). Thus, when $|\theta|$ exceeds π, the pseudovector is shifted as follows:

$$\theta' = \begin{cases} \theta \text{ if } |\theta| \leq \pi \\ 2\pi - \theta \text{ if } |\theta| > \pi \end{cases} \qquad (3.2\text{–}29)$$

The updated quarternion of equation (3.2–24) or (3.2–25) is adjusted accordingly (θ replaced with θ'). This shift of the magnitude of the pseudovector allows unlimited rotation (> 180°).

3.2.4 Element Transformation

The updated transformation matrix $[T_n]$ relates the current element coordinate system to the global Cartesian coordinate system as shown in Figure 3.2–1.

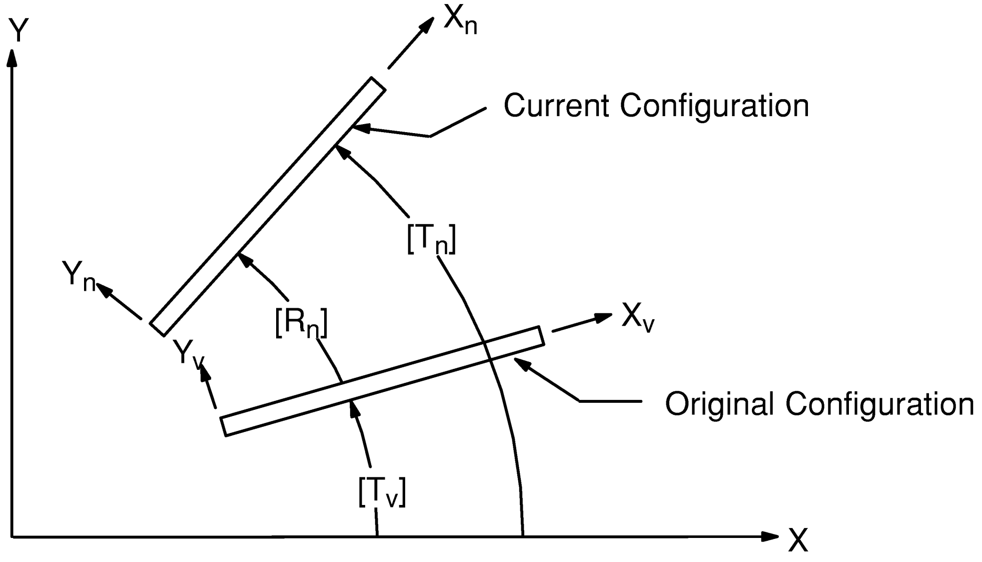


Figure 3.2–1 Element Transformation Definitions

$[T_n]$ can be computed directly or the rotation of the element coordinate system $[R_n]$ can be computed and related to $[T_n]$ by

$$[T_n] = [T_v][R_n] \qquad (3.2\text{–}30)$$

where $[T_v]$ is the original transformation matrix. The determination of $[T_n]$ is unique to the type of element involved, whether it is a solid element, shell element, beam element, or spar element.

A. Solid Elements. The rotation matrix $[R_n]$ for these elements is extracted from the displacement field using the deformation gradient coupled with the polar decomposition theorem (see Malvern(87)).

B. Shell Elements. The updated normal direction (element z direction) is computed directly from the updated coordinates. The computation of the element normal is given in Chapter 14 for each particular shell element. The extraction procedure outlined for solid elements is used coupled with the information on the normal direction to compute the rotation matrix $[R_n]$.

C. Beam Elements. The nodal rotation increments from $\{\Delta u\}$ are averaged to determine the average rotation of the element and equation (3.2–28) is used to update the average element rotation. Equation (3.2–10) then yields the rotation matrix $[R_n]$ from the updated average element rotation. In special cases where the average rotation of the element computed in the above way differs significantly from the average rotation of the element computed from nodal translations, the quality of the results will be degraded.

D. Link Elements. The updated transformation $[T_n]$ is computed directly from the updated coordinates.

E. Generalized Mass Element (MASS21). The nodal rotation increment from $\{\Delta u\}$ is used in equation (3.2–28) to update the average element rotation. Equation (3.2–10) then yields the rotation matrix $[R_n]$ from the updated element rotation.

3.2.5 Deformational Displacements

The displacement field can be decomposed into a rigid body translation, a rigid body rotation, and a component which causes strains:

$$\{u\} = \{u^r\} + \{u^d\} \tag{3.2–31}$$

where: $\{u^r\}$ = rigid body motion

$\{u^d\}$ = deformational displacements which cause strains

$\{u^d\}$ contains both translational as well as rotational DOF.

The translational component of the deformational displacement can be extracted from the displacement field by

$$\left\{u_t^d\right\} = [R_n] \left(\{x_v\} + \{u\}\right) - \{x_v\} \qquad (3.2–32)$$

where: $\left\{u_t^d\right\}$ = translational component of the deformational displacement

$[R_n]$ = current element rotation matrix

$\{x_v\}$ = original element coordinates in the global coordinate system

$\{u\}$ = element displacement vector in global coordinates

$\{u^d\}$ is in the global coordinate system.

For elements with rotational DOFs, the rotational components of the deformational displacement must be computed. The rotational components are extracted by essentially “subtracting” the nodal rotations {u} from the element rotation given by $\{u^r\}$. In terms of the pseudovectors this operation is performed as follows for each node:

1. Compute a transformation matrix from the nodal pseudovector $\{\theta^n\}$ using equation (3.2–10) yielding $[T_n]$.
2. Compute the relative rotation $[T^d]$ between $[R_n]$ and $[T_n]$:

   $$[T^d] = [R_n]\,[T_n]^T \qquad (3.2–33)$$

   This relative rotation contains the rotational deformations of that node as shown in Figure 3.2–2.
3. Extract the nodal rotational deformations $\{u^d\}$ from $[T^d]$ using equation (3.2–11).

Because of the definition of the pseudovector (equation (3.2–9)), the deformational rotations extracted in step 3 are limited to less than 30°, since $2\sin(\theta/2)$ no longer approximates θ itself above 30°. This limitation only applies to the rotational distortion (i.e. bending) within a single element.

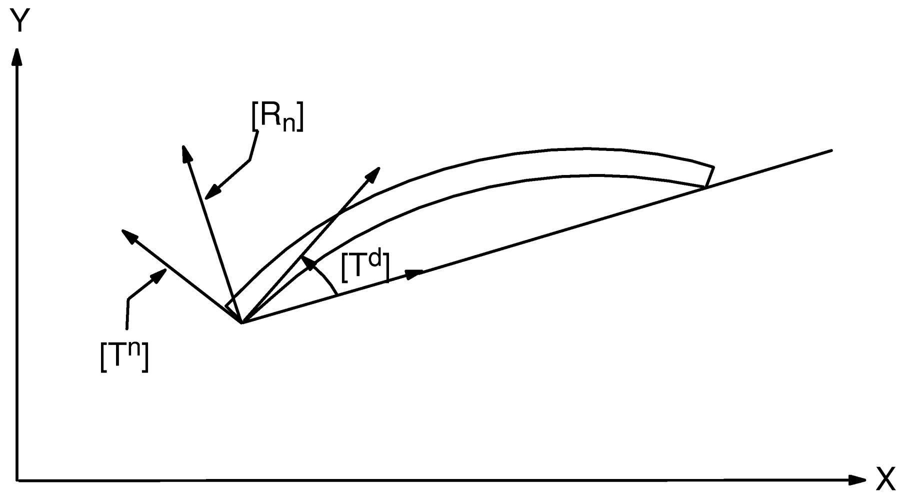


Figure 3.2–2 Definition of Deformational Rotations

3.2.6 Updating Rotations

Once the transformation [T] and deformational displacements $\{u^d\}$ are determined, the element matrices (equation (3.2–3)) and restoring force (equation (3.2–4)) can be determined. The solution of the system of equations yields a displacement increment $\{\Delta u\}$. The nodal rotations at the element level are updated using equation (3.2–24) with the rotational components of $\{\Delta u\}$. The global rotations (in the output and on the results file) are not updated with the pseudovector approach, but are simply added to the previous rotation in $\{u_{n-1}\}$.

3.2.7 Applicable Input

NLGEOM,ON activates the large rotation computations in those elements which support it. **SSTIF**,ON activates the stress–stiffening contribution to the tangent stiffness matrix (which may be required for structures weak in bending resistance).

3.2.8 Applicable Output

Stresses (output quantity S) are engineering stresses in the rotated element coordinate system (the element coordinate system follows the material as it rotates). Strains (output quantity EPEL, EPPL, etc.) are engineering strains, also in the rotated element coordinate system. This applies to element types that do not have large strain capability. For element types that have large strain capability, see Section 3.1.

3.2.9 Consistent Tangent Stiffness Matrix and Finite Rotation

It has been found in many situations that the use of consistent tangent stiffness in a nonlinear analysis can speed up the rate of convergence greatly. It normally results in a quadratic rate of convergence. A consistent tangent stiffness matrix is derived from the discretized finite element equilibrium equations without the introduction of various approximations. The terminology of finite rotation in the context of geometrical nonlinearity implies that rotations can be arbitrarily large and can be updated accurately. A consistent tangent stiffness accounting for finite rotations derived by Nour–Omid and Rankin(175) for beam/shell elements is used. The technology of consistent tangent matrix and finite rotation makes the buckling and postbuckling analysis a relatively easy task. KEYOPT(2) = 1 implemented in BEAM4 and SHELL63 uses this technology. The theory of finite rotation representation and update has been described in Section 3.2 (Pseudovector Representation) using a pseudovector representation. The following will outline the derivations of a consistent tangent stiffness matrix used for the corotational approach.

The nonlinear static finite element equations solved can be characterized by at the element level by:

$$\sum_{e=1}^{N} \left([T_n]^T \left\{ F_e^{int} \right\} - \left\{ F_e^{a} \right\} \right) = 0 \qquad (3.2–34)$$

where:

N = number of total elements

$\left\{ F_e^{int} \right\}$ = element internal force vector in the element coordinate system, generally see equation (3.2–35)

$[T_n]^T$ = transform matrix transferring the local internal force vector into the global coordinate system

$\left\{ F_e^{a} \right\}$ = applied load vector at the element level in the global coordinate system

$$\left\{ F_e^{int} \right\} = \int [B_v]^T \{\sigma_e\} \; d(vol) \qquad (3.2–35)$$

Hereafter, we shall focus on the derivation of the consistent tangent matrix at the element level without introducing an approximation. The consistent tangent matrix is obtained by differentiating equation (3.2–34) with respect to displacement variables $\{u_e\}$:

$$\left[K_e^T\right]_{consistent} = [T_n]^T \frac{\partial\left\{F_e^{int}\right\}}{\partial\{u_e\}} + \frac{\partial[T_n]^T}{\partial\{u_e\}} \{F_e^{int}\}$$

$$= \underbrace{[T_n]^T \int_e [B_v]^T \frac{\partial\{\sigma_e\}}{\partial\{u_e\}} \, d(vol)}_{I} + \underbrace{[T_n]^T \int_e \frac{\partial[B_v]^T}{\partial\{u_e\}} \{\sigma_e\} \, d(vol)}_{II} \qquad (3.2\text{–}36)$$

$$+ \underbrace{\frac{\partial[T_v]^T}{\partial\{u_e\}} \left\{F_e^{int}\right\}}_{III}$$

It can be seen that Part I is the main tangent matrix (equation (3.2–3)) and Part II is the stress stiffening matrix (equation (3.1–35), (3.3–14) or (3.3–17)). Part III is another part of the stress stiffening matrix (see Nour–Omid and Rankin(175)) traditionally neglected in the past. However, many numerical experiments have shown that Part III of $\left[K_e^T\right]$ is essential to the faster rate of convergence. KEYOPT(2) = 1 implemented in BEAM4 and SHELL63 allows the use of $\left[K_e^T\right]$ as shown in equation (3.2–36). In some cases, Part III of $\left[K_e^T\right]$ is unsymmetric; when this occurs, a procedure of symmetrizing $\left[K_e^T\right]$ is invoked.

As Part III of the consistent tangent matrix utilizes the internal force vector $\left\{F_e^{int}\right\}$ to form the matrix, it is required that the internal vector $\left\{F_e^{int}\right\}$ not be so large as to dominate the main tangent matrix (Part I). This can normally be guaranteed if the realistic material and geometry are used, that is, the element is not used as a rigid link and the actual thicknesses are input.

It is also noted that the consistent tangent matrix (3.2–36) is very suitable for use with the arc–length solution method.

3.3 Stress Stiffening

3.3.1 Overview and Usage

Stress stiffening (also called geometric stiffening, incremental stiffening, initial stress stiffening, or differential stiffening by other authors) is the stiffening (or weakening) of a structure due to its stress state. This stiffening effect normally needs to be considered for thin structures with bending stiffness very small compared to axial stiffness, such as cables, thin beams, and shells and couples the in–plane and transverse discplacements. This effect also augments the regular nonlinear stiffness matrix produced by large strain or large deflection effects (**NLGEOM**,ON). The effect of stress stiffening is accounted for by generating and then using an additional stiffness matrix, hereinafter called the "stress stiffness matrix". The stress stiffness matrix is added to the regular stiffness matrix in order to give the total stiffness using the **SSTIF**,ON command. Stress stiffening may be used for static (**ANTYPE**,STATIC) or transient (**ANTYPE**,TRANS) analyses. Working with the stress stiffness matrix is the pressure load stiffness, discussed in Section 3.3.4.

The stress stiffness matrix is computed based on the stress state of the previous equilibrium iteration. Thus, to generate a valid stress–stiffened problem, at least two iterations are normally required, with the first iteration being used to determine the stress state that will be used to generate the stress stiffness matrix of the second iteration. If this additional stiffness affects the stresses, more iterations need to be done to obtain a converged solution.

In some linear analyses, the static (or initial) stress state may be large enough that the additional stiffness effects must be included for accuracy. Modal (**ANTYPE**,MODAL), reduced harmonic (**ANTYPE**,HARMIC with *Method* = REDUC on the **HROPT** command), reduced transient (**ANTYPE**,TRANS with *Method* = REDUC on the **TRNOPT** command) and substructure (**ANTYPE**,SUBSTR) analyses are linear analyses for which the prestressing effects can be included using the **PSTRES**,ON command. Note that in these cases the stress stiffness matrix is constant, so that the stresses computed in the analysis (e.g. the transient or harmonic stresses) are assumed small compared to the prestress stress.

If membrane stresses should become compressive rather than tensile, then terms in the stress stiffness matrix may "cancel" the positive terms in the regular stiffness matrix and therefore yield a nonpositive–definite total stiffness matrix, which indicates the onset of buckling. If this happens, it is indicated with the message: *"Large negative pivot value ___, at node ___ may be because buckling load has been exceeded"*. It must be noted that a stress stiffened model with insufficient boundary conditions to prevent rigid body motion may yield the same message.

The linear buckling load can be calculated directly by adding an unknown multiplier of the stress stiffness matrix to the regular stiffness matrix and performing an eigenvalue buckling problem (**ANTYPE**,BUCKLE) to calculate the value of the unknown multiplier. This is discussed in more detail in Section 17.5.

3.3.2 Theory

The strain–displacement equations for the general motion of a differential length fiber are derived below. Two different results have been obtained and these are both discussed below. Consider the motion of a differential fiber, originally at dS, and then at ds after deformation.

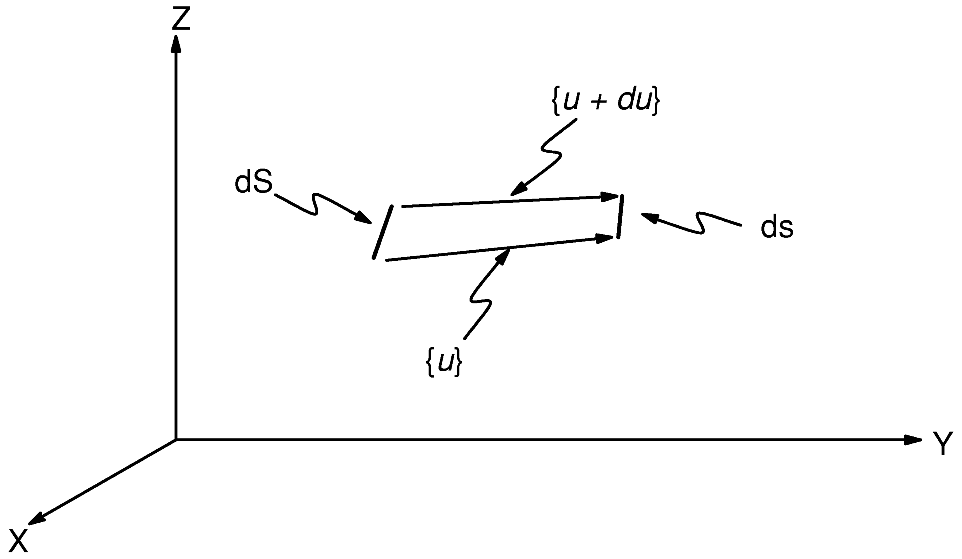


Figure 3.3–1 General Motion of a Fiber

One end moves {*u*}, and the other end moves {*u* + d*u*}, as shown in Figure 3.3–1. The motion of one end with the rigid body translation removed is {*u* + d*u*} – {*u*} = {d*u*}. {d*u*} may be expanded as

$$\{\mathrm{d}u\} = \begin{Bmatrix} \mathrm{du} \\ \mathrm{dv} \\ \mathrm{dw} \end{Bmatrix} \qquad (3.3\text{–}1)$$

where u is the displacement parallel to the original orientation of the fiber. This is shown in Figure 3.3–2. Note that X, Y, and Z represent global Cartesian axes, and x, y, and z represent axes based on the original orientation of the fiber. By the Pythagorean theorem,

$$\mathrm{ds} = \sqrt{(\mathrm{dS} + \mathrm{du})^2 + (\mathrm{dv})^2 + (\mathrm{dw})^2} \qquad (3.3\text{–}2)$$

The stretch, Λ, is given by dividing ds by the original length dS:

$$\Lambda = \frac{ds}{dS} = \sqrt{\left(1 + \frac{du}{dS}\right)^2 + \left(\frac{dv}{dS}\right)^2 + \left(\frac{dw}{dS}\right)^2} \qquad (3.3\text{–}3)$$

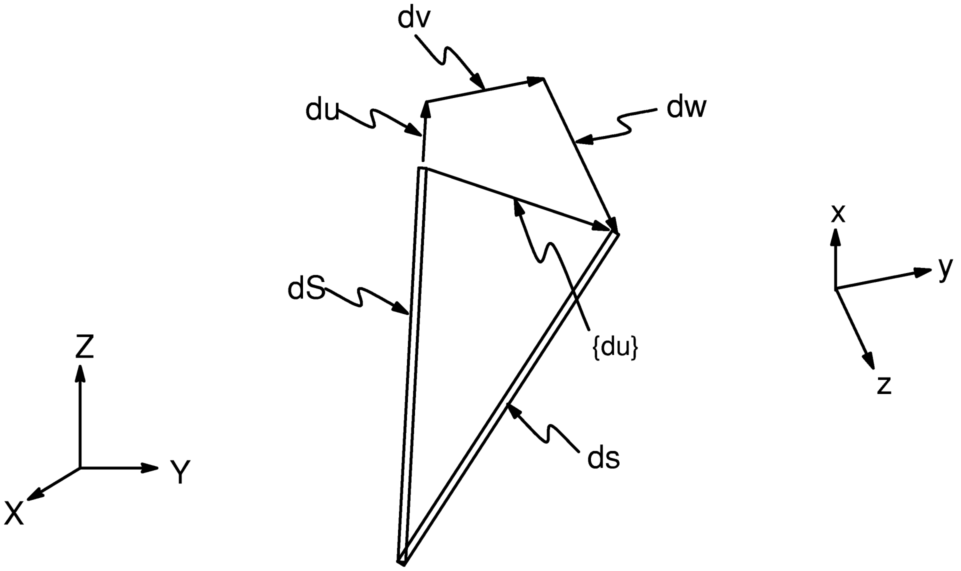


Figure 3.3–2 Motion of a Fiber with Rigid Body Motion Removed

As dS is along the local x axis,

$$\Lambda = \sqrt{\left(1 + \frac{du}{dx}\right)^2 + \left(\frac{dv}{dx}\right)^2 + \left(\frac{dw}{dx}\right)^2} \qquad (3.3\text{–}4)$$

Next, Λ is expanded and converted to partial notation:

$$\Lambda = \sqrt{1 + 2\frac{\partial u}{\partial x} + \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2} \qquad (3.3\text{–}5)$$

The binominal theorem states that:

$$\sqrt{1 + A} = 1 + \frac{A}{2} - \frac{A^2}{8} + \frac{A^3}{16} \ldots \qquad (3.3\text{–}6)$$

when $A^2 < 1$. One should be aware that using a limited number of terms of this series may restrict its applicability to small rotations and small strains. If the first two terms of the series in equation (3.3–6) are used to expand equation (3.3–5),

$$\Lambda = 1 + \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \qquad (3.3–7)$$

The resultant strain (same as extension since strains are assumed to be small) is then

$$\varepsilon_x = \Lambda - 1 = \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \qquad (3.3–8)$$

If, more accurately, the first <u>three</u> terms of equation (3.3–6) are used and displacement derivatives of the third order and above are dropped, equation (3.3–6) reduces to:

$$\Lambda = 1 + \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \qquad (3.3–9)$$

The resultant strain is:

$$\varepsilon_x = \Lambda - 1 = \frac{\partial u}{\partial x} + \frac{1}{2}\left(\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial x}\right)^2\right) \qquad (3.3–10)$$

For most 2–D and 3–D elements, equation (3.3–8) is more convenient to use as no account of the loaded direction has to be considered. The error associated with this is small as the strains were assumed to be small. For 1–D structures, and some 2–D elements, equation (3.3–10) is used for its greater accuracy and causes no difficulty in its implementation.

3.3.3 Implementation

The stress–stiffness matrices are derived based on equation (3.1–35), but using the nonlinear strain–displacement relationships given in equation (3.3–8) or (3.3–10) (Cook(5)).

For a spar such as LINK8 the stress–stiffness matrix is given as:

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \qquad (3.3–11)$$

The stress stiffness matrix for a 2–D beam (BEAM3) is given in equation (3.3–12), which is the same as reported by Przemieniecki(28). All beam and straight pipe elements use the same type of matrix. The 3–D beam and straight elements do not account for twist buckling. Forces used by straight pipe elements are based on not only the effect of axial stress with pipe wall, but also internal and external pressures on the 'end–caps' of each element. This force is sometimes referred to as effective tension.

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 0 & & & & & \text{Symmetric} \\ 0 & \frac{6}{5} & & & & \\ 0 & \frac{1}{10}L & \frac{2}{15}L^2 & & & \\ 0 & 0 & 0 & 0 & & \\ 0 & -\frac{6}{5} & -\frac{1}{10}L & 0 & \frac{6}{5} & \\ 0 & \frac{1}{10}L & -\frac{1}{30}L^2 & 0 & -\frac{1}{10}L & \frac{2}{15}L^2 \end{bmatrix} \qquad (3.3–12)$$

where: F = force in member
L = length of member

The stress stiffness matrix for 2–D and 3–D solid elements is generated by the use of numerical integration. A 3–D solid element (SOLID45) is used here as an example:

$$[S_\ell] = \begin{bmatrix} [S_o] & [0] & [0] \\ [0] & [S_o] & [0] \\ [0] & [0] & [S_o] \end{bmatrix} \qquad (3.3–13)$$

where the matrices shown in equation (3.3–13) have been reordered so that first all x–direction DOF are given, then y, and then z. $[S_o]$ is an 8 by 8 matrix given by:

$$[S_o] = \int_{vol} [S_g]^T[S_m]\,[S_g]\; d(vol) \qquad (3.3–14)$$

The matrices used by this equation are:

$$[S_m] = \begin{bmatrix} \sigma_x & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_y & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z \end{bmatrix} \qquad (3.3–15)$$

where σ_x, σ_{xy} etc. are stress based on the displacements of the previous iteration, and,

$$\left[S_g\right] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x} & \cdots\cdots & \frac{\partial N_8}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y} & \cdots\cdots & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \cdots\cdots & \frac{\partial N_8}{\partial z} \end{bmatrix} \tag{3.3–16}$$

where N_i represents the ith shape function. This is the stress stiffness matrix for small strain analyses (**NLGEOM**,OFF or **NLGEOM**,ON with large deflection elements). For large strain elements in a large strain analysis (**NLGEOM**,ON), the stress stiffening contribution is computed using the actual strain–displacement relationship (equation (3.1–6)).

One further case requires some explanation: axisymmetric structures with non–axisymmetric deformations. As any stiffening effects may only be axisymmetric, only MODE = 0 cases are used for the prestress case. Then, any subsequent load steps with any value of MODE (including 0 itself) uses that same stress state, until another, more recent, MODE = 0 case is available. Also, torsional stresses are not incorporated into any stress stiffening effects.

Specializing this to SHELL61 (Axisymmetric–Harmonic Structural Shell), only two stresses are used for prestressing: σ_s, σ_θ, the meridional and hoop stresses, respectively. The element stress stiffness matrix is:

$$\left[S_\ell\right] = \int_{vol} \left[S_g\right]^T \left[S_m\right] \left[S_g\right] d(vol) \tag{3.3–17}$$

$$\left[S_m\right] = \begin{bmatrix} \sigma_s & 0 & 0 & 0 \\ 0 & \sigma_s & 0 & 0 \\ 0 & 0 & \sigma_\theta & 0 \\ 0 & 0 & 0 & \sigma_\theta \end{bmatrix}$$

$$\left[S_g\right] = \left[A_s\right]\left[N\right]$$

where $[A_s]$ is defined below and [N] is defined by the element shape functions. $[A_s]$ is an operator matrix and its terms are:

$$[A_s] = \begin{bmatrix} 0 & 0 & \frac{\partial}{\partial s} \\ 0 & C\left(-\frac{\partial}{\partial s} - \frac{\sin\theta}{R}\right) & 0 \\ 0 & C\,\frac{\cos\theta}{R} & \frac{\partial}{R\partial\theta} \\ -\frac{\partial}{R\partial\theta} & 0 & 0 \end{bmatrix} \qquad (3.3–18)$$

where: $C = \begin{cases} 0.0 \text{ if MODE} = 0 \\ 1.0 \text{ if MODE} > 0 \end{cases}$

The three columns of the $[A_s]$ matrix refer to u, v, and w motions, respectively. As suggested by the definition for $[S_m]$, the first two rows of $[A_s]$ relate to σ_s and the second two rows relate to σ_θ. The first row of $[A_s]$ is for motion normal to the shell varying in the s direction and the second row is for hoop motions varying in the s direction. Similarly, the third row is for normal motions varying in the hoop direction. Thus equation (3.3–10), rather than equation (3.3–8), is the type of nonlinear strain–displacement expression that has been used to develop equation (3.3–18).

3.3.4 Pressure Load Stiffness

The effect of change of direction and/or area of an applied pressure is responsible for the pressure load stiffness matrix ($[S^{pr}]$) (Bonet and Wood(236), Section 6.5.2). It is used either for a large deflection analysis (NLGEOM,ON), regardless of the setting of the **SSTIF** command, for an eigenvalue buckling analysis, or for a dynamic analysis using the **PSTRES,ON** command.

The need of $[S^{pr}]$ is most dramatically seen when modelling the collapse of a ring due to external pressure using eigenvalue buckling. The expected answer is:

$$P_{cr} = \frac{CEI}{R^3} \qquad (3.3–19)$$

where:

P_{cr} = critical buckling load
E = Young’s modulus
I = moment of inertia
R = radius of the ring
C = 3.0

This value of C = 3.0 is achieved when using $[S^{pr}]$, but when it is missing, C = 4.0, a 33% error.

$[S^{pr}]$ is available only for SURF154, SHELL181, BEAM188, and BEAM189.

3.3.5 Applicable Input

Table 3.0–1 indicates the elements which compute a stress stiffness matrix.

In a nonlinear analysis (**ANTYPE**,STATIC or **ANTYPE**,TRANS), **SSTIF**,ON activates the stress stiffness contribution to the stiffness matrix. With **NLGEOM**,OFF, the rotations are presumed to be small and the additional stiffness induced by the stress state is included. With **NLGEOM**,ON, the stress stiffness augments the tangent matrix, affecting the rate of convergence but not the final converged solution.

In a prestressed linear analysis, **PSTRES**,ON activates the stress stiffness contribution in the prestressed analysis and directs the preceding analysis to save the stress state.

3.3.6 Applicable Output

In a small deflection/small strain analysis (**NLGEOM**,OFF), the 2–D and 3–D elements compute their strains using equation (3.3–8). The strains (output quantities EPEL, EPPL, etc.) therefore include the higher–order terms (e.g. $\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2$) in the strain computation. Also, nodal and reaction loads (output quantities F and M) will reflect the stress stiffness contribution, so that moment and force equilibrium include the higher order (small rotation) effects.

3.4 Spin Softening

The vibration of a spinning body will cause relative circumferential motions, which will change the direction of the centrifugal load which, in turn, will tend to destabilize the structure. As a small deflection analysis cannot directly account for changes in geometry, the effect can be accounted for by an adjustment of the stiffness matrix, called spin softening. Spin softening is activated with the input quantity *KSPIN* on the **OMEGA** command, and is intended for use only with modal (**ANTYPE**,MODAL), harmonic response (**ANTYPE**,HARMIC), reduced transient (**ANTYPE**,TRANS, with **TRNOPT**,REDUC) or substructure (**ANTYPE**,SUBSTR) analyses. When doing a static (**ANTYPE**,STATIC) or a full transient (**ANTYPE**,TRANS with **TRNOPT**,FULL) analysis, this effect is more accurately accounted for by large deflections (**NLGEOM**,ON).

Consider a simple spring–mass system, with the spring oriented radially with respect to the axis of rotation, as shown in Figure 3.4–1. Equilibrium of the spring and centrifugal forces on the mass using small deflection logic requires:

$$Ku = \omega_s^2 M r \tag{3.4–1}$$

where:

u = radial displacement of the mass from the rest position

r = radial rest position of the mass with respect to the axis of rotation

ω_s = angular velocity of rotation

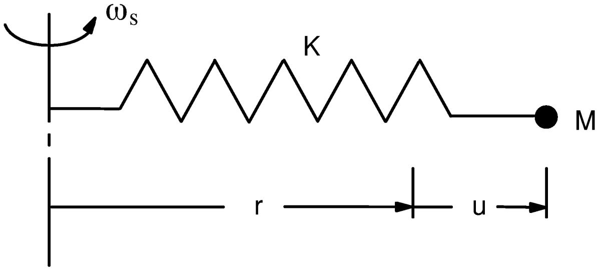


Figure 3.4–1 Spinning Spring–Mass System

However, to account for large deflection effects, equation (3.4–1) must be expanded to:

$$Ku = \omega_s^2 M (r + u) \tag{3.4–2}$$

Rearranging terms,

$$\left(K - \omega_s^2 M\right) u = \omega_s^2 M r \tag{3.4–3}$$

Defining:

$$\overline{K} = K - \omega_s^2 M \tag{3.4–4}$$

and

$$\overline{F} = \omega_s^2 M r \tag{3.4–5}$$

equation (3.4–3) becomes simply,

$$\overline{K} u = \overline{F} \tag{3.4–6}$$

$\overline{K}$ is the stiffness needed in a small deflection solution to account for large deflection effects. $\overline{F}$ is the same as that derived from small deflection logic. Thus, the large deflection effects are included in a small deflection solution. This decrease in the effective stiffness matrix is called spin (or centrifugal) softening. See also Carnegie(104) for additional development.

Extension of equation (3.4–4) into three dimensions gives:

$$\overline{K}_{xx} = K_{xx} - \left(\omega_y^2 + \omega_z^2\right) M_{xx} \tag{3.4–7}$$

$$\overline{K}_{yy} = K_{yy} - \left(\omega_x^2 + \omega_z^2\right) M_{yy} \tag{3.4–8}$$

$$\overline{K}_{zz} = K_{zz} - \left(\omega_x^2 + \omega_y^2\right) M_{zz} \tag{3.4–9}$$

where: K_{xx}, K_{yy}, K_{zz} = the x, y, and z components of stiffness as computed by the element

$\overline{K}_{xx}, \overline{K}_{yy}, \overline{K}_{zz}$ = the x, y, and z components of stiffness adjusted for spin softening

M_{xx}, M_{yy}, M_{zz} = the x, y, and z components of mass

$\omega_x, \omega_y, \omega_z$ = angular velocities of rotation about the x, y, and z axes

There are no modifications to the cross terms:

$$\overline{K}_{xy} = K_{xy} \tag{3.4–10}$$

$$\overline{K}_{yz} = K_{yz} \tag{3.4–11}$$

$$\overline{K}_{zx} = K_{zx} \tag{3.4–12}$$

From equations (3.4–7) thru (3.4–9), it may be seen that there are spin softening effects only in the plane of rotation, not normal to the plane of rotation. Using the example of a modal analysis, equation (3.4–4) can be combined with equation (17.3–4) to give:

$$\left| \left[\overline{\mathrm{K}}\right] - \omega^2 [\mathrm{M}] \right| = 0 \tag{3.4–13}$$

or

$$\left| \left([\mathrm{K}] - \omega_s^2 [\mathrm{M}]\right) - \omega^2 [\mathrm{M}] \right| = 0 \tag{3.4–14}$$

where: ω = the natural circular frequencies of the rotating body.

If stress stiffening is added to equation (3.4–14), the resulting equation is:

$$\left| \left([\mathrm{K}] + [\mathrm{S}] - \omega_s^2 [\mathrm{M}]\right) - \omega^2 [\mathrm{M}] \right| = 0 \tag{3.4–15}$$

Stress stiffening is normally applied whenever spin softening is activated, even though they are independent theoretically. The modal analysis of a thin fan blade is shown in Figure 3.4–2.

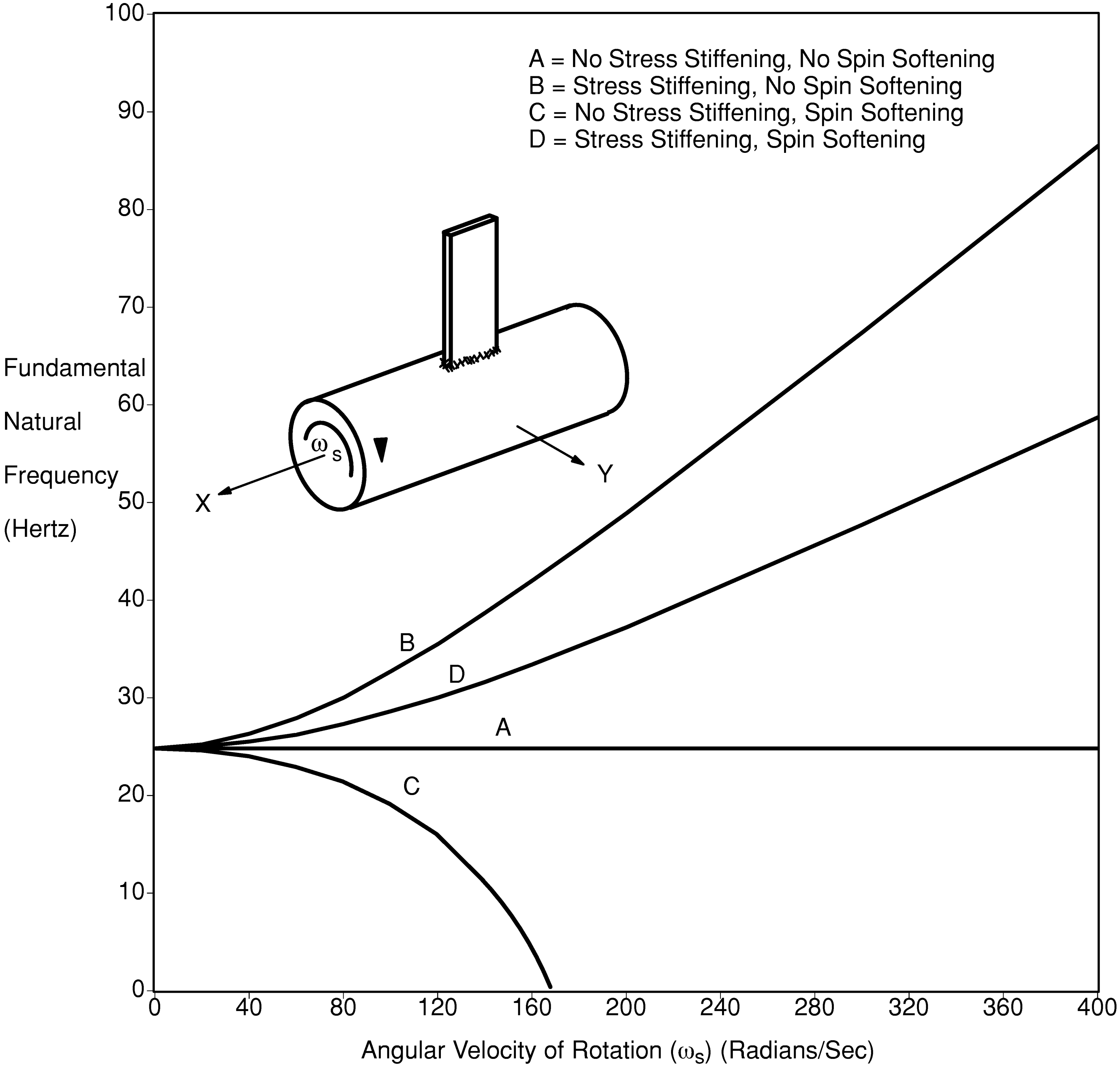


Figure 3.4–2   Effects of Spin Softening and Stress Stiffening on Fan Blade Natural Frequencies

Chapter 4
Structures with Material Nonlinearities

ANSYS Theory Reference
Chapter 4 – Table of Contents

4.0 Introduction to Material Nonlinearities

This chapter discusses the structural material nonlinearities of plasticity, creep, nonlinear elasticity, hyperelasticity, viscoelasticity, concrete and swelling. Not included in this section are the slider, frictional, or other nonlinear elements (such as COMBIN7, COMBIN40, CONTAC12, CONTAC48, etc. discussed in Chapter 14) that can represent other nonlinear material behavior.

Material nonlinearities are due to the nonlinear relationship between stress and strain, that is, the stress is a nonlinear function of the strain. The relationship is also path dependent (except for the case of nonlinear elasticity and hyperelasticity), so that the stress depends on the strain history as well as the strain itself.

The program can account for eight types of material nonlinearities:

1. *Rate–independent plasticity* is characterized by the irreversible instantaneous straining that occurs in a material.
2. *Rate–dependent plasticity* allows the plastic–strains to develop over a time interval. This is also termed *viscoplasticity*.
3. *Creep* is also an irreversible straining that occurs in a material and is rate–dependent so that the strains develop over time. The time frame for creep is usually much larger than that for rate–dependent plasticity.
4. *Nonlinear elasticity* allows a nonlinear stress–strain relationship to be specified. All straining is reversible.
5. *Hyperelasticity* is defined by a strain energy density potential that characterizes elastomeric and foam–type materials. All straining is reversible.
6. *Viscoelasticity* is a rate–dependent material characterization that includes a viscous contribution to the elastic straining.
7. *Concrete* materials include cracking and crushing capability.
8. *Swelling* allows materials to enlarge in the presence of neutron flux.

Only the concrete element (SOLID65) supports the concrete model and only the viscoelastic elements (VISCO88, VISCO89) support the viscoelastic material model. The elements which support the other options are listed in Table 4.0–1. Note that also listed in this table are how many stress and strain components are involved. One component uses x (e.g. SX, EPELX, etc.), four components use X, Y, Z, XY and six components use X, Y, Z, XY, YZ, XZ.

The plastic pipe elements (PIPE20 and PIPE60) have four components, so that the nonlinear torsional and pressure effects may be considered. If only one component is available, only the nonlinear stretching and bending effects could be considered. This is relevant, for instance, to the 3–D thin–walled beam (BEAM24) which has only one component. Thus linear torsional effects are included, but nonlinear torsional effects are not.

Table 4.0–1 Elements Having Nonlinear Material Capability

Name		Number of Stress Components	Rate–independent plasticity	Rate–dependent plasticity	Creep	Nonlinear Elasticity	Hyperelasticity	Swelling
LINK1	2–D Spar (or Truss)	1	X		X	X		X
PLANE2	2–D 6–Node Triangular Structural Solid	4	X		X	X		X
LINK8	3–D Spar (or Truss)	1	X		X	X		X
PIPE20	Plastic Straight Pipe	4	X		X	X		X
BEAM23	2–D Plastic Beam	1	X		X	X		X
BEAM24	3–D Thin–Walled Beam	1	X		X	X		X
PLANE42	2–D Structural Solid	4	X		X	X		X
SHELL43	Plastic Shell	6	X		X	X		X
SOLID45	3–D Structural Solid	6	X		X	X		X
SHELL51	Axisymmetric Structural Shell	4	X		X	X		X
HYPER56	2–D 4–Node Mixed U–P Hyperelastic Solid	4,6†					X	
HYPER58	3–D 8–Node Mixed U–P Hyperelastic Solid	6					X	
PIPE60	Plastic Curved Pipe (Elbow)	4	X		X	X		X
SOLID62	3–D Coupled Magnetic–Structural Solid	6	X		X	X		X
SOLID65	3–D Reinforced Concrete Solid	1,6*	X		X	X		
HYPER74	2–D 8–Node Mixed U–P Hyperelastic Solid	4,6†					X	
PLANE82	2–D 8–Node Structural Solid	4	X		X	X		X
HYPER84	2–D 8–Node Hyperelastic Solid	4,6†					X	
HYPER86	3–D 8–Node Hyperelastic Solid	6					X	
SHELL91	16–Layer Structural Shell	6	X			X		
SOLID92	3–D 10–Node Tetrahedral Structural Solid	6	X		X	X		X
SHELL93	8–Node Structural Shell	6	X			X		
SOLID95	20–Node Structural Solid	6	X		X	X		X
VISCO106	2–D Large Strain Solid	4,6†	X	X				
VISCO107	3–D Large Strain Solid	6	X	X				
VISCO108	2–D 8–Node Large Strain Solid	4,6†	X	X				
HYPER158	3–D 10–Node Tetrahedral Mixed U–P Hyperelastic Solid						X	
LINK180	3–D Finite Strain Spar (or Truss)		X		X			

Name		Number of Stress Components	Rate–independent plasticity	Rate–dependent plasticity	Creep	Nonlinear Elasticity	Hyperelasticity	Swelling
SHELL181	4–Node Structural Shell		X		X		X	
PLANE182	2–D Structural Solid		X		X			
PLANE183	2–D 8–Node Structural Solid		X		X			
SOLID185	3–D Structural Solid		X		X			
SOLID186	20–Node Structural Solid		X		X			
SOLID187	3–D 10–Node Tetrahedral Structural Solid		X		X			
BEAM188	3–D Finite Strain Linear Beam		X		X			
BEAM189	3–D Finite Strain Quadratic Beam		X		X			

* The rebar in SOLID65 uses one component, the bulk solid uses six components.

† Four components if plane strain, six if axisymmetric (includes torsion)

4.0.1 Strain Definitions

For the case of nonlinear materials, the definition of elastic strain given with equation (2.1–1) has the form of:

$$\{\epsilon^{el}\} = \{\epsilon\} - \{\epsilon^{th}\} - \{\epsilon^{pl}\} - \{\epsilon^{cr}\} - \{\epsilon^{sw}\} \tag{4.0–1}$$

where: ϵ^{el} = elastic strain vector (output quantity EPEL)

ϵ = total strain vector

ϵ^{th} = thermal strain vector (output quantity EPTH)

ϵ^{pl} = plastic strain vector (output quantity EPPL)

ϵ^{cr} = creep strain vector (output quantity EPCR)

ϵ^{sw} = swelling strain vector (output quantity EPSW)

In this case, $\{\epsilon\}$ is the strain measured by a strain gauge. (Equation (4.0–1) is only intended to show the relationships between the terms. See subsequent sections for more detail).

In POST1, total strain is reported as:

$$\{\epsilon^{tot}\} = \{\epsilon^{el}\} + \{\epsilon^{pl}\} + \{\epsilon^{cr}\} \tag{4.0–2}$$

where: ϵ^{tot} = component total strain (output quantity EPTO)

Comparing the last two equations,

$$\{\epsilon^{\mathrm{tot}}\} = \{\epsilon\} - \{\epsilon^{\mathrm{th}}\} - \{\epsilon^{\mathrm{sw}}\} \tag{4.0–3}$$

The difference between these two “total” strains stems from the different usages: $\{\epsilon\}$ can be used to compare strain gauge results and ϵ^{tot} can be used to plot nonlinear stress–strain curves.

4.1 Rate–Independent Plasticity

Rate–independent plasticity is characterized by the irreversible straining that occurs in a material once a certain level of stress is reached. The plastic strains are assumed to develop instantaneously, that is, independent of time. The ANSYS program provides seven options to characterize different types of material behaviors. These options are:

TB *Lab*	Material Behavior Option
BISO	Bilinear Isotropic Hardening
MISO	Multilinear Isotropic Hardening
NLISO	Nonlinear Isotropic Hardening
BKIN	Classical Bilinear Kinematic Hardening
MKIN/KINH	Multilinear Kinematic Hardening
CHAB	Nonlinear Kinematic Hardening
ANISO	Anisotropic
DP	Drucker–Prager
USER	User Specified Behavior (see Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide*)

Except for **TB**,USER, each of these is explained in greater detail later in this chapter. Figure 4.1–1 represents the stress–strain behavior of each of the options. The TB label (*Lab* field of the **TB** command) is used throughout this section to identify which type of plasticity is being discussed. The actual values are input on the **TBDATA** or **TBPT** commands. Table 4.0–1 indicates which elements support the rate–independent options.

4.1.1 Theory

Plasticity theory provides a mathematical relationship that characterizes the elasto–plastic response of materials. There are three ingredients in the rate–independent plasticity theory: the yield criterion, flow rule and the hardening rule. These will be discussed in detail subsequently. Table 4.1–1 summarizes the notation used in the remainder of this chapter.

4.1.2 Yield Criterion

The yield criterion determines the stress level at which yielding is initiated. For multi–component stresses, this is represented as a function of the individual components, f({σ}), which can be interpreted as an equivalent stress σ_e:

$$\sigma_e = f(\{\sigma\}) \qquad (4.1–1)$$

where: {σ} = stress vector

Table 4.1–1 Notation

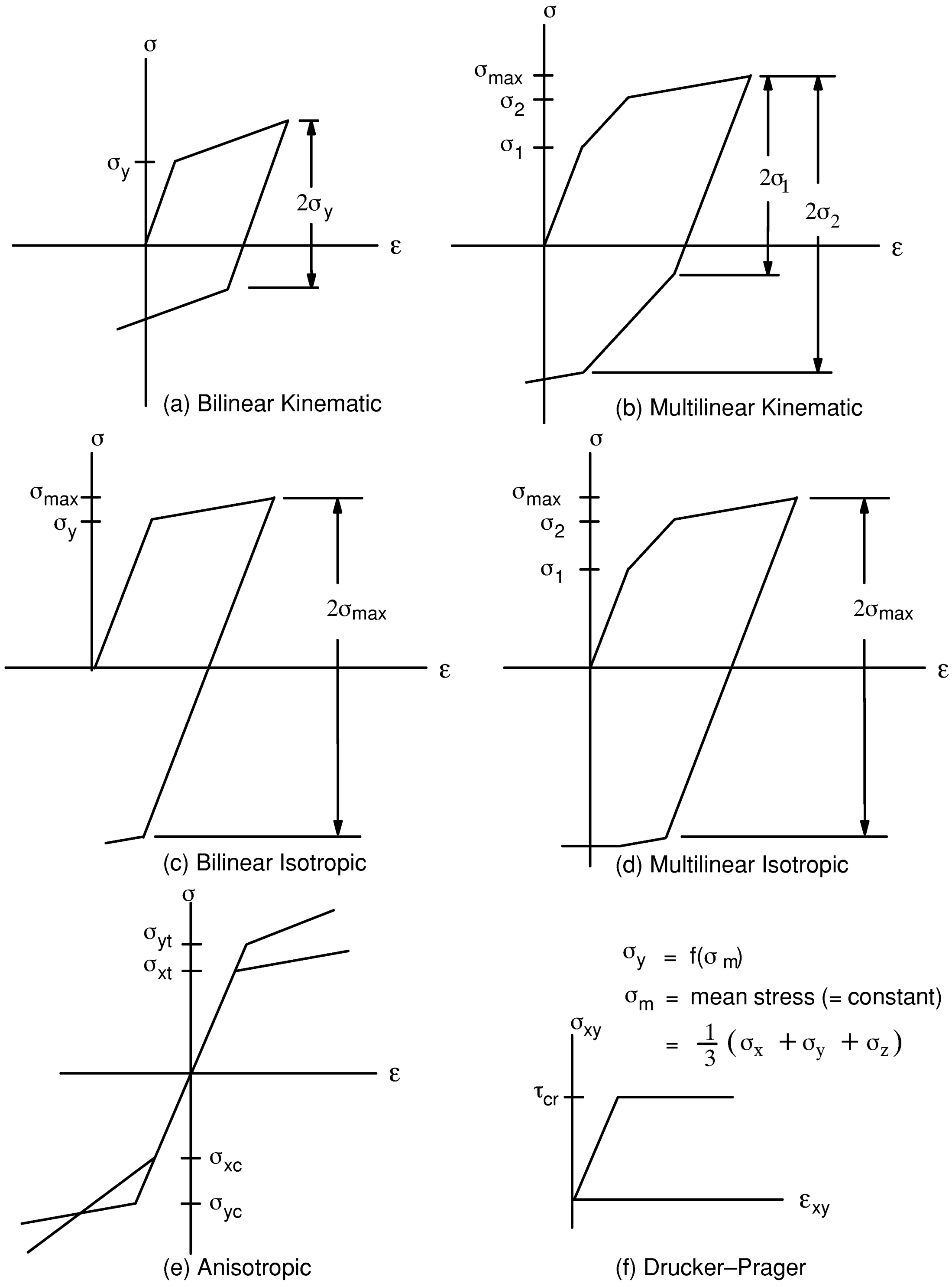
Variable	ANSYS Output Label	Definition
$\{\varepsilon^{el}\}$	EPEL	elastic strains
$\{\varepsilon^{pl}\}$	EPPL	plastic strains
$\{\varepsilon^{tr}\}$		trial strain
$\hat{\epsilon}^{pl}$	EPEQ*	equivalent plastic strain
$\{\sigma\}$	S	stresses
σ_e		equivalent stress
σ_y		material yield parameter
σ_m	HPRES	mean or hydrostatic stress
$\hat{\sigma}^{pl}_e$	SEPL	equivalent stress parameter
λ		plastic multiplier
$\{\alpha\}$		yield surface translation
$\varkappa$		plastic work
C		translation multiplier
[D]		stress–strain matrix
E_T		tangent modulus
F		yield criterion
N	SRAT	stress ratio
Q		plastic potential
{S}		deviatoric stress

*In the large strain solids VISCO106, VISCO107, and VISCO108, EPEQ is labeled as PSV.

When the equivalent stress is equal to a material yield parameter σ_y,

$$f(\{\sigma\}) = \sigma_y \qquad (4.1–2)$$

the material will develop plastic strains. If σ_e is less than σ_y, the material is elastic and the stresses will develop according to the elastic stress–strain relations. Note that the equivalent stress can never exceed the material yield since in this case plastic strains would develop instantaneously, thereby reducing the stress to the material yield. Equation (4.1–2) can be plotted in stress space as shown in Figure 4.1–2 for some of the plasticity options. The surfaces in Figure 4.1–2 are known as the yield surfaces and any stress state inside the surface is elastic, that is, they do not cause plastic strains.



(a) Bilinear Kinematic

(b) Multilinear Kinematic

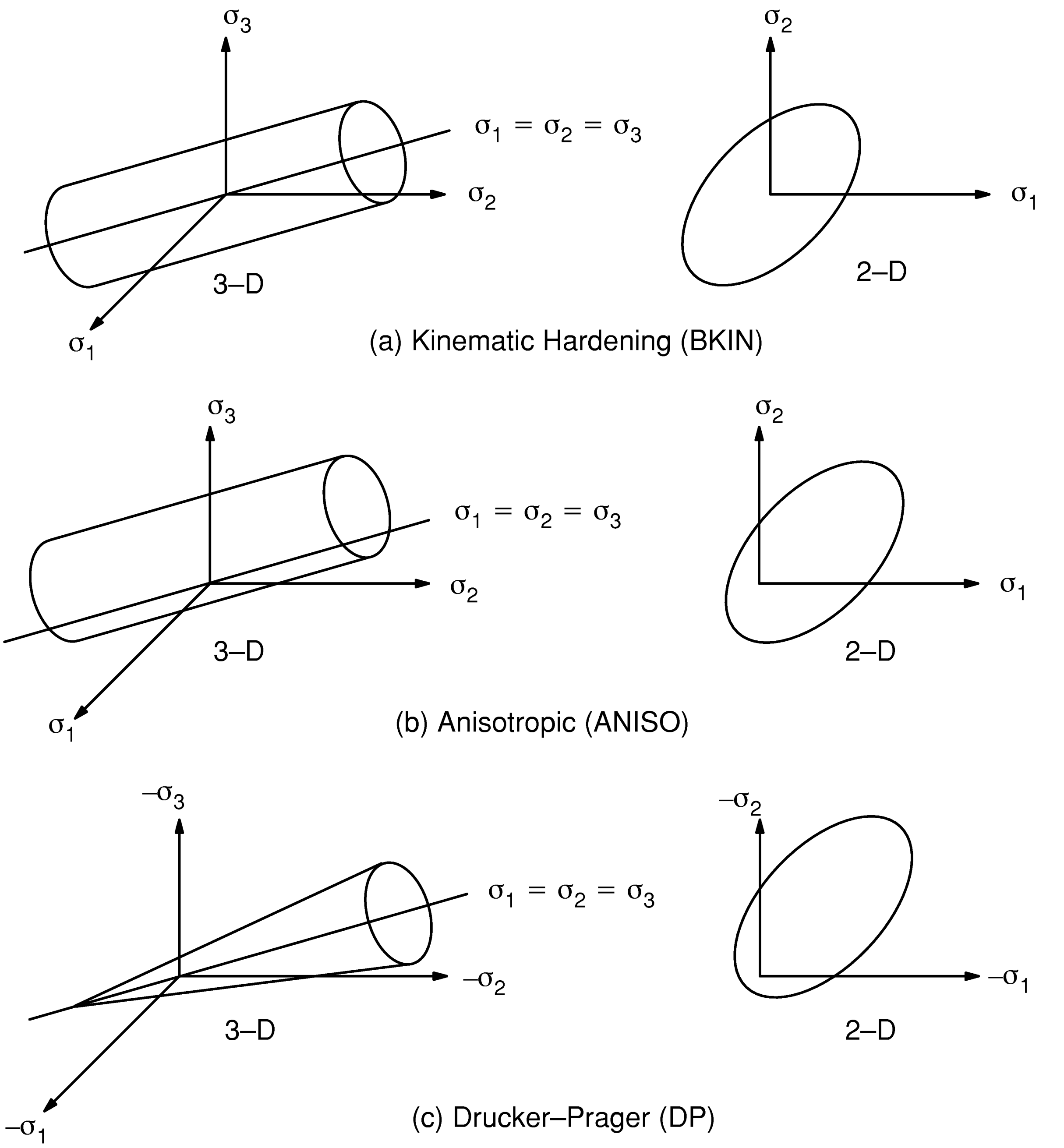
(c) Bilinear Isotropic

(d) Multilinear Isotropic

(e) Anisotropic

(f) Drucker–Prager

Figure 4.1–1 Stress–Strain Behavior of Each of the Plasticity Options



(a) Kinematic Hardening (BKIN)

(b) Anisotropic (ANISO)

(c) Drucker–Prager (DP)

Figure 4.1–2 Yield Surfaces for BKIN, ANISO and DP Options

4.1.3 Flow Rule

The flow rule determines the direction of plastic straining and is given as:

$$\left\{d\epsilon^{pl}\right\} = \lambda\left\{\frac{\partial Q}{\partial \sigma}\right\} \tag{4.1–3}$$

where: λ = plastic multiplier (which determines the amount of plastic straining)

Q = function of stress termed the plastic potential (which determines the direction of plastic straining)

If Q is the yield function (as is normally assumed), the flow rule is termed associative and the plastic strains occur in a direction normal to the yield surface.

4.1.4 Hardening Rule

The hardening rule describes the changing of the yield surface with progressive yielding, so that the conditions (i.e. stress states) for subsequent yielding can be established. Two hardening rules are available: work (or isotropic) hardening and kinematic hardening. In work hardening, the yield surface remains centered about its initial centerline and expands in size as the plastic strains develop. For materials with isotropic plastic behavior this is termed isotropic hardening and is shown in Figure 4.1–3a. Kinematic hardening assumes that the yield surface remains constant in size and the surface translates in stress space with progressive yielding, as shown in Figure 4.1–3b.

The yield criterion, flow rule and hardening rule for each option are summarized in Table 4.1–2 and are discussed in detail later in this chapter.

4.1.5 Plastic Strain Increment

If the equivalent stress computed using elastic properties exceeds the material yield, then plastic straining must occur. Plastic strains reduce the stress state so that it satisfies the yield criterion, equation (4.1–2). Based on the theory presented in the previous section, the plastic strain increment is readily calculated.

The hardening rule states that the yield criterion changes with work hardening and/or with kinematic hardening. Incorporating these dependencies into equation (4.1–2), and recasting it into the following form:

$$F\left(\{\sigma\},\ \kappa,\ \{\alpha\}\right) = 0 \qquad (4.1\text{–}4)$$

where: κ = plastic work

$\{\alpha\}$ = translation of yield surface

$\varkappa$ and $\{\alpha\}$ are termed internal or state variables. Specifically, the plastic work is the sum of the plastic work done over the history of loading:

$$\kappa = \int \{\sigma\}^T\ [M]\left\{d\epsilon^{pl}\right\} \qquad (4.1\text{–}5)$$

and translation (or shift) of the yield surface is also history dependent and is given as:

$$\{\alpha\} = \int C\left\{d\epsilon^{pl}\right\} \tag{4.1–6}$$

where: C = material parameter

{α} = back stress (location of the center of the yield surface)

Equation (4.1–4) can be differentiated so that the consistency condition is:

$$dF = \left\{\frac{\partial F}{\partial \sigma}\right\}^T [M]\{d\sigma\} + \frac{\partial F}{\partial \kappa}\, d\kappa + \left\{\frac{\partial F}{\partial \alpha}\right\}^T [M]\{d\alpha\} = 0 \tag{4.1–7}$$

where:

$$[M] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Noting from equation (4.1–5) that

$$d\kappa = \{\sigma\}^T [M]\left\{d\epsilon^{pl}\right\} \tag{4.1–8}$$

and from (4.1–6) that

$$\{d\alpha\} = C\left\{d\epsilon^{pl}\right\} \tag{4.1–9}$$

equation (4.1–7) becomes

$$\left\{\frac{\partial F}{\partial \sigma}\right\}^T [M]\{d\sigma\} + \frac{\partial F}{\partial \kappa}\, \{\sigma\}^T [M]\left\{d\epsilon^{pl}\right\} + C\left\{\frac{\partial F}{\partial \alpha}\right\}^T [M]\left\{d\epsilon^{pl}\right\} = 0 \tag{4.1–10}$$

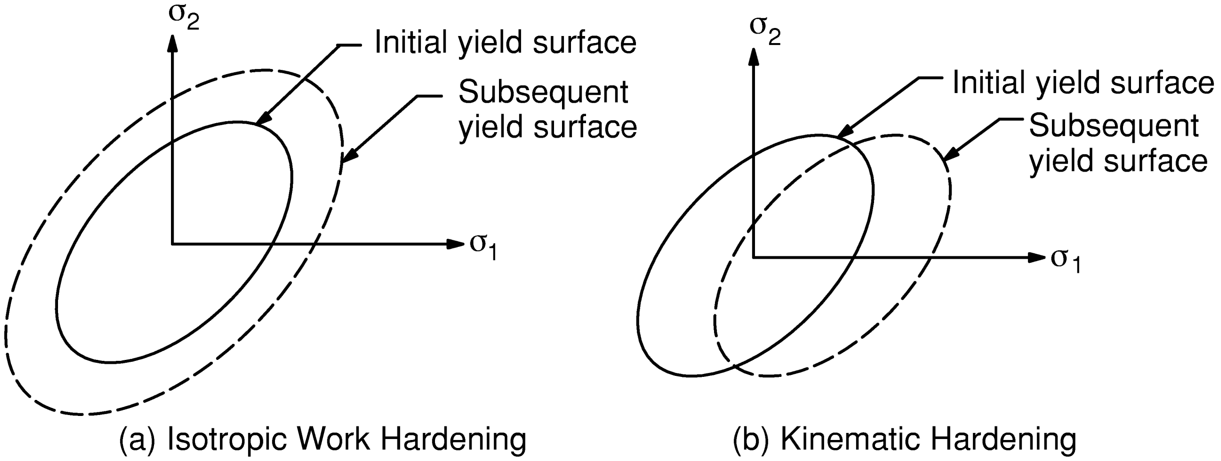


Figure 4.1–3   Hardening Rules

Table 4.1–2   Summary of Plasticity Options

Name	TB Lab	Yield Criterion	Flow Rule	Hardening Rule	Material Response
Bilinear Isotropic Hardening	BISO	von Mises	associative	work hardening	bilinear
Multilinear Isotropic Hardening	MISO	von Mises	associative	work hardening	multilinear
Nonlinear Isotropic Hardening	NLISO	von Mises	associative	work hardening	nonlinear
Classical Bilinear Kinematic Hardening	BKIN	von Mises	associative (Prandtl–Reuss equations)	kinematic hardening	bilinear
Multilinear Kinematic Hardening	MKIN/ KINH	von Mises	associative	kinematic hardening	multilinear
Nonlinear Kinematic Hardening	CHAB	von Mises	associative	kinematic hardening	nonlinear
Anisotropic	ANISO	modified von Mises	associative	work hardening	bilinear, each direction and tension and compression different
Drucker–Prager	DP	von Mises with dependence on hydrostatic stress	associative or nonassociative	none	elastic–perfectly plastic

The stress increment can be computed via the elastic stress–strain relations

$$\{d\sigma\} = [D]\left\{d\epsilon^{el}\right\} \tag{4.1–11}$$

where: [D] = strain–displacement matrix

with

$$\left\{d\epsilon^{el}\right\} = \{d\epsilon\} - \left\{d\epsilon^{pl}\right\} \tag{4.1–12}$$

since the total strain increment can be divided into an elastic and plastic part. Substituting equation (4.1–3) into equations (4.1–10) and (4.1–12) and combining equations (4.1–10), (4.1–11), and (4.1–12) yields

$$\lambda = \frac{\left\{\frac{\partial F}{\partial \sigma}\right\}^T [M][D]\{d\epsilon\}}{-\frac{\partial F}{\partial \kappa}\ \{\sigma\}^T[M]\left\{\frac{\partial Q}{\partial \sigma}\right\} - C\ \left\{\frac{\partial F}{\partial \alpha}\right\}^T [M]\left\{\frac{\partial Q}{\partial \sigma}\right\} + \left\{\frac{\partial F}{\partial \sigma}\right\}^T [M][D]\left\{\frac{\partial Q}{\partial \sigma}\right\}} \tag{4.1–13}$$

The size of the plastic strain increment is therefore related to the total increment in strain, the current stress state, and the specific forms of the yield and potential surfaces. The plastic strain increment is then computed using equation (4.1–3):

$$\left\{d\epsilon^{pl}\right\} = \lambda \left\{\frac{\partial Q}{\partial \sigma}\right\} \tag{4.1–14}$$

4.1.6 Implementation

An Euler backward scheme is used to enforce the consistency condition (equation (4.1–7)). This ensures that the updated stress, strains and internal variables are on the yield surface. The algorithm proceeds as follows:

1. The material parameter σ_y (equation (4.1–2)) is determined for this time step (e.g., the yield stress at the current temperature).
2. The stresses are computed based on the trial strain $\{\varepsilon^{tr}\}$, which is the total strain minus the plastic strain from the previous time point (thermal and other effects are ignored):

$$\left\{\epsilon_n^{tr}\right\} = \{\epsilon_n\} - \left\{\epsilon_{n-1}^{pl}\right\} \tag{4.1–15}$$

where the superscripts are described with Notation (Section 1.1.2) and subscripts refer to the time point. Where all terms refer to the current time point, the subscript is dropped. The trial stress is then

$$\left\{\sigma^{tr}\right\} = [D]\left\{\epsilon^{tr}\right\} \tag{4.1–16}$$

3. The equivalent stress σ_e is evaluated at this stress level by equation (4.1–1). If σ_e is less than σ_y the material is elastic and no plastic strain increment is computed.
4. If the stress exceeds the material yield, the plastic multiplier λ is determined by a local Newton–Raphson iteration procedure (Simo and Taylor(155)).
5. $\{\Delta\varepsilon^{pl}\}$ is computed via equation (4.1–14).
6. The current plastic strain is updated

$$\left\{\epsilon_n^{pl}\right\} = \left\{\epsilon_{n-1}^{pl}\right\} + \left\{\Delta\epsilon^{pl}\right\} \tag{4.1–17}$$

where: $\left\{\epsilon_n^{pl}\right\}$ = current plastic strains (output quantity EPPL)

and the elastic strain computed

$$\left\{\epsilon^{el}\right\} = \left\{\epsilon^{tr}\right\} - \left\{\Delta\epsilon^{pl}\right\} \tag{4.1–18}$$

where: $\left\{\epsilon^{el}\right\}$ = elastic strains (output quantity EPEL)

The stress vector is:

$$\{\sigma\} = [D]\left\{\epsilon^{el}\right\} \tag{4.1–19}$$

where: $\{\sigma\}$ = stresses (output quantity S)

7. The increments in the plastic work $\Delta\varkappa$ and the center of the yield surface $\{\Delta\alpha\}$ are computed via equations (4.1–8) and (4.1–9) and the current values updated

$$\kappa_n = \kappa_{n-1} + \Delta\kappa \tag{4.1–20}$$

and

$$\{\alpha_n\} = \left\{\alpha_{n-1}\right\} + \{\Delta\alpha\} \tag{4.1–21}$$

where the subscript n–1 refers to the values at the previous time point.

8. For output purposes, an equivalent plastic strain $\hat{\epsilon}^{pl}$ (output quantity EPEQ), equivalent plastic strain increment $\Delta\hat{\epsilon}^{pl}$ (output with the label "MAX PLASTIC STRAIN STEP"), equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output quantity SEPL) and stress ratio N (output quantity SRAT) are computed. The stress ratio is given as

$$N = \frac{\sigma_e}{\sigma_y} \tag{4.1–22}$$

where σ_e is evaluated using the trial stress $\{\sigma^{tr}\}$. N is therefore greater than or equal to one when yielding is occurring and less than one when the stress state is elastic. The equivalent plastic strain increment is given as:

$$\Delta\hat{\epsilon}^{pl} = \left(\frac{2}{3}\left\{\Delta\epsilon^{pl}\right\}^T[M]\left\{\Delta\epsilon^{pl}\right\}\right)^{\frac{1}{2}} \tag{4.1–23}$$

The equivalent plastic strain and equivalent stress parameters are developed for each option in the next sections.

Note that the Euler backward integration scheme in step 4 is the radial return algorithm (Krieg(46)) for the von Mises yield criterion.

4.1.7 Elasto–Plastic Stress–Strain Matrix

The tangent or elasto–plastic stress–strain matrix is derived from the local Newton–Raphson iteration scheme used in step 4 above (Simo and Taylor(155)). It is therefore the consistent (or algorithmic) tangent. If the flow rule is nonassociative ($F \neq Q$), then the tangent is unsymmetric. To preserve the symmetry of the matrix, for analyses with a nonassociative flow rule (Drucker–Prager only), the matrix is evaluated using F only and again with Q only and the two matrices averaged.

4.1.8 Specialization for Multilinear Isotropic Hardening (MISO) and Bilinear Isotropic Hardening (BISO)

These options use the von Mises yield criterion with the associated flow rule and isotropic (work) hardening.

The equivalent stress (equation (4.1–1)) is:

$$\sigma_e = \left[\frac{3}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} \tag{4.1–24}$$

where {s} is the deviatoric stress (equation (4.1–32)). When σ_e is equal to the current yield stress σ_k the material is assumed to yield. The yield criterion is:

$$F = \left[\frac{3}{2}\ \{s\}^T[M]\{s\}\right]^{\frac{1}{2}} - \sigma_k = 0 \tag{4.1–25}$$

For work hardening, σ_k is a function of the amount of plastic work done. For the case of isotropic plasticity assumed here, σ_k can be determined directly from the equivalent plastic strain $\hat{\epsilon}^{pl}$ of equation (4.1–38) (output quantity EPEQ) and the uniaxial stress–strain curve as depicted in Figure 4.1–4. σ_k is output as the equivalent stress parameter, output quantity SEPL. For temperature–dependent curves with the MISO option, σ_k is determined by temperature interpolation of the input curves after they have been converted to stress–plastic strain format.

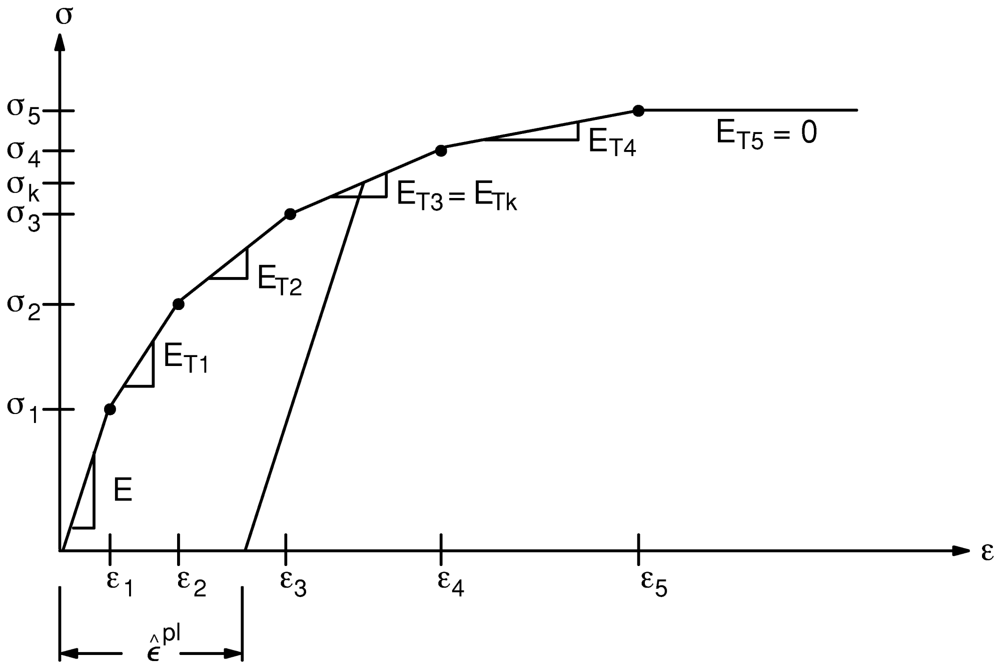


Figure 4.1–4 Uniaxial Behavior for MISO and σ_k Determination

4.1.9 Specification for Nonlinear Isotropic Hardening

In addition to the bilinear and multilinear isotropic hardening options, ANSYS also provides another nonlinear isotropic hardening option, which is also called the Voce(NO TAG) hardening law. The isotropic hardening behavior of materials is specified by an equation:

$$R = k + R_o\,\hat{\epsilon}^{pl} + R_\infty\left(1 - e^{-b\hat{\epsilon}^{pl}}\right) \tag{4.1–26}$$

where: k = elastic limit

R_o, R_∞, b = material parameters characterizing the isotropic hardening behavior of materials

$\hat{\epsilon}^{pl}$ = equivalent plastic strain

The constitutive equations are based on linear isotropic elasticity, the von Mises yield function and the associated flow rule. The yield function is:

$$F = \left[\frac{3}{2}\ \{s\}^T[M]\ \{s\}\right]^{\frac{1}{2}} - R = 0 \tag{4.1–27}$$

The plastic strain increment is:

$$\left\{\Delta\epsilon^{pl}\right\} = \lambda\left\{\frac{\partial Q}{\partial\sigma}\right\} = \lambda\left\{\frac{\partial F}{\partial\sigma}\right\} = \frac{3}{2}\ \lambda\ \frac{\{s\}}{\sigma_e} \tag{4.1–28}$$

where: λ = plastic multiplier

The equivalent plastic strain increment is then:

$$\Delta\ \hat{\epsilon}^{pl} = \sqrt{\frac{2}{3}\ \left\{\Delta\epsilon^{pl}\right\}^T\ [M]\ \left\{\Delta\epsilon^{pl}\right\}} = \lambda \tag{4.1–29}$$

The accumulated equivalent plastic strain is:

$$\hat{\epsilon}^{pl} = \sum \Delta\hat{\epsilon}^{pl} \tag{4.1–30}$$

4.1.10 Specialization for Bilinear Kinematic Hardening (BKIN)

This option uses the von Mises yield criterion with the associated flow rule and kinematic hardening.

The equivalent stress (equation (4.1–1)) is therefore

$$\sigma_e = \left[\frac{3}{2}\ (\{s\}-\{\alpha\})^T[M](\{s\}-\{\alpha\})\right]^{\frac{1}{2}} \tag{4.1–31}$$

where: $\{s\}$ = deviatoric stress vector

$$\{s\} = \{\sigma\} - \sigma_m\lfloor\ 1\quad 1\quad 1\quad 0\quad 0\quad 0\rfloor^T \tag{4.1–32}$$

where: σ_m = mean or hydrostatic stress = $\frac{1}{3}$ $(\sigma_x + \sigma_y + \sigma_z)$

$\{\alpha\}$ = yield surface translation vector (equation (4.1–6))

Note that since equation (4.1–31) is dependent on the deviatoric stress, yielding is independent of the hydrostatic stress state. When σ_e is equal to the uniaxial yield stress, σ_y, the material is assumed to yield. The yield criterion (equation (4.1–4)) is therefore,

$$F = \left[\frac{3}{2}\ (\{s\}-\{\alpha\})^T\,[M](\{s\}-\{\alpha\})\right]^{\frac{1}{2}} - \sigma_y = 0 \qquad (4.1\text{–}33)$$

The associated flow rule yields

$$\left\{\frac{\partial Q}{\partial \sigma}\right\} = \left\{\frac{\partial F}{\partial \sigma}\right\} = \frac{3}{2\sigma_e}\ (\{s\} - \{\alpha\}) \qquad (4.1\text{–}34)$$

so that the increment in plastic strain is normal to the yield surface. The associated flow rule with the von Mises yield criterion is known as the Prandtl–Reuss flow equation.

The yield surface translation is defined as:

$$\{\alpha\} = 2G\left\{\epsilon^{sh}\right\} \qquad (4.1\text{–}35)$$

where: G = shear modulus = E / (2 (1+ν))

E = Young's modulus (input as EX on **MP** command)

ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

The shift strain is computed analogously to equation (4.1–21):

$$\left\{\epsilon^{sh}_n\right\} = \left\{\epsilon^{sh}_{n-1}\right\} + \left\{\Delta\epsilon^{sh}\right\} \qquad (4.1\text{–}36)$$

where: $\left\{\Delta\epsilon^{sh}\right\} = \frac{C}{2G}\left\{\Delta\epsilon^{pl}\right\}$

$$C = \frac{2}{3}\ \frac{EE_T}{E - E_T} \qquad (4.1\text{–}37)$$

where: E = Young's modulus (input as EX on **MP** command)

E_T = tangent modulus from the bilinear uniaxial stress–strain curve

The yield surface translation $\{\varepsilon^{sh}\}$ is initially zero and changes with subsequent plastic straining.

The equivalent plastic strain is dependent on the loading history and is defined to be:

$$\hat{\epsilon}^{pl}_n = \hat{\epsilon}^{pl}_{n-1} + \Delta\hat{\epsilon}^{pl} \qquad (4.1\text{–}38)$$

where: $\hat{\epsilon}^{pl}_n$ = equivalent plastic strain for this time point (output quantity EPEQ)

$\hat{\epsilon}^{pl}_{n-1}$ = equivalent plastic strain from the previous time point

The equivalent stress parameter is defined to be:

$$\hat{\sigma}^{pl}_e = \sigma_y + \frac{EE_T}{E - E_T}\,\hat{\epsilon}^{pl}_n \qquad (4.1\text{–}39)$$

where: $\hat{\sigma}^{pl}_n$ = equivalent stress parameter (output quantity SEPL)

Note that if there is no plastic straining ($\hat{\epsilon}^{pl} = 0$), then $\hat{\sigma}^{pl}_e$ is equal to the yield stress. $\hat{\sigma}^{pl}_e$ only has meaning during the initial, monotonically increasing portion of the load history. If the load were to be reversed after plastic loading, the stresses and therefore σ_e would fall below yield but $\hat{\sigma}^{pl}_e$ would register above yield (since $\hat{\epsilon}^{pl}$ is non–zero).

4.1.11 Specialization for Multilinear Kinematic Hardening (MKIN/KINH)

This option uses the Besseling(53) model also called the sublayer or overlay model (Zienkiewicz(54)) to characterize the material behavior. The material behavior is assumed to be composed of various portions (or subvolumes), all subjected to the same total strain, but each subvolume having a different yield strength. (For a plane stress analysis, the material can be thought to be made up of a number of different layers, each with a different thickness and yield stress.) Each subvolume has a simple stress–strain response but when combined the model can represent complex behavior. This allows a multilinear stress–strain curve that exhibits the Bauschinger (kinematic hardening) effect (Figure 4.1–1b).

The following steps are performed in the plasticity calculations:

1. The portion of total volume for each subvolume and its corresponding yield strength are determined.
2. The increment in plastic strain is determined for each subvolume assuming each subvolume is subjected to the same total strain.
3. The individual increments in plastic strain are summed using the weighting factors determined in step 1 to compute the overall or apparent increment in plastic strain.
4. The plastic strain is updated and the elastic strain is computed.

The portion of total volume (the weighting factor) and yield stress for each subvolume is determined by matching the material response to the uniaxial stress–strain curve. A perfectly plastic von Mises material is assumed and this yields for the weighting factor for subvolume k

$$w_k = \frac{E - E_{Tk}}{E - \frac{1-2\nu}{3} E_{Tk}} - \sum_{i=1}^{k-1} w_i \tag{4.1–40}$$

where:

w_k = the weighting factor (portion of total volume) for subvolume k and is evaluated sequentially from 1 to the number of subvolumes

E_{Tk} = the slope of the kth segment of the stress–strain curve (see Figure 4.1–5)

Σw_i = the sum of the weighting factors for the previously evaluated subvolumes

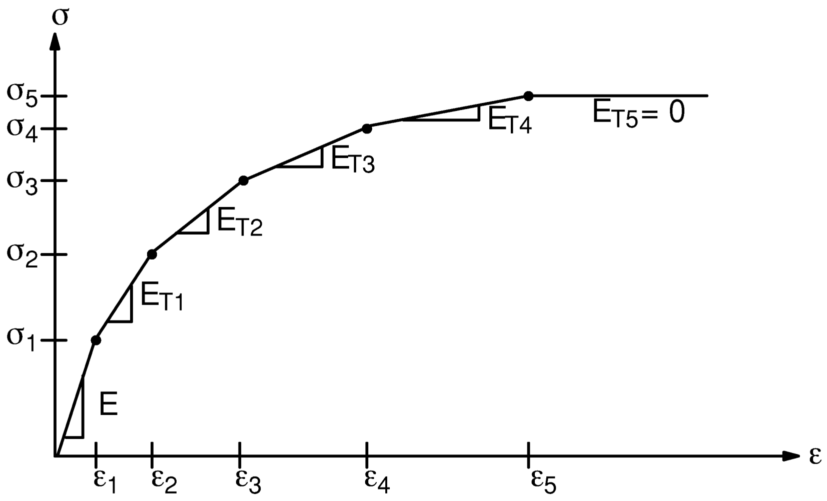


Figure 4.1–5 Uniaxial Behavior for MKIN

The yield stress for each subvolume is given by

$$\sigma_{yk} = \frac{1}{2(1+\nu)} \left(3E\epsilon_k - (1 - 2\nu)\sigma_k \right) \tag{4.1–41}$$

where (ε_k, σ_k) is the breakpoint in the stress–strain curve. The number of subvolumes corresponds to the number of breakpoints specified.

The increment in plastic strain $\left\{\Delta\epsilon_k^{pl}\right\}$ for each subvolume is computed using a von Mises yield criterion with the associated flow rule. The section on specialization for bilinear kinematic hardening is followed but since each subvolume is elastic–perfectly plastic, C and therefore {α} is zero.

The plastic strain increment for the entire volume is the sum of the subvolume increments:

$$\left\{\Delta\epsilon^{pl}\right\} = \sum_{i=1}^{N_{sv}} w_i \left\{\Delta\epsilon_i^{pl}\right\} \qquad (4.1\text{–}42)$$

where: N_{sv} = number of subvolumes

The current plastic strain and elastic strain can then be computed for the entire volume via equations (4.1–17) and (4.1–18).

The equivalent plastic strain $\hat{\epsilon}^{pl}$ (output quantity EPEQ) is defined by equation (4.1–38) and equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output quantity SEPL) is computed by evaluating the input stress–strain curve at $\hat{\epsilon}^{pl}$ (after adjusting the curve for the elastic strain component).

4.1.12 Specialization for Nonlinar Kinematic Hardening (CHAB)

The material model considered is a rate–independent version of the nonlinear kinematic hardening model proposed by Chaboche(244, 245). The constitutive equations are based on linear isotropic elasticity, a von Mises yield function and the associated flow rule. Like the bilinear and multilinear kinematic hardening options, the model can be used to simulate the monotonic hardening and the Bauschinger effect. The model is also applicable to simulate the ratcheting effect of materials. In addition, the model allows the superposition of several kinematic models as well as isotropic hardening models. It is thus able to model the complicated cyclic plastic behavior of materials, such as cyclic hardening or softening and ratcheting or shakedown.

The model uses the von Mises yield criterion with the associated flow rule, the yield function is:

$$F = \left[\frac{3}{2}\ (\{s\} - \{\alpha\})^T[M]\ (\{s\} - \{\alpha\})\right]^{\frac{1}{2}} - R = 0 \qquad (4.1\text{–}43)$$

where: R = isotropic hardening variable

According to the normality rule, the flow rule is written:

$$\left\{\Delta\epsilon^{pl}\right\} = \lambda \left\{\frac{\partial Q}{\partial \sigma}\right\} \qquad (4.1\text{–}44)$$

where: λ = plastic multiplier

The back stress {α} is superposition of several kinematic models as:

$$\{\alpha\} = \sum_{i=1}^{n} \{\alpha\}_i \qquad (4.1\text{–}45)$$

where: n = number of kinematic models to be superposed.

The evolution of the back stress (the kinematic hardening rule) for each component is defined as:

$$\{\Delta\alpha\}_i = \frac{2}{3} C_i \left\{\Delta\epsilon^{pl}\right\} - \gamma_i \{\alpha\}_i \lambda \qquad (4.1\text{–}46)$$

where:

$C_i,\ \gamma_i,\ i = 1,2,...n$ = material constants for kinematic hardening

The associated flow rule yields:

$$\left\{\frac{\partial Q}{\partial \sigma}\right\} = \left\{\frac{\partial F}{\partial \sigma}\right\} = \frac{3}{2}\,\frac{\{s\} - \{\alpha\}}{\sigma_e} \qquad (4.1\text{–}47)$$

The plastic strain increment, equation (4.1–44) is rewritten as:

$$\left\{\Delta\epsilon^{pl}\right\} = \frac{3}{2}\,\lambda\,\frac{\{s\} - \{\alpha\}}{\sigma_e} \qquad (4.1\text{–}48$$

The equivalent plastic strain increment is then:

$$\hat{\epsilon}^{pl} = \sqrt{\frac{2}{3}\left\{\Delta\epsilon^{pl}\right\}^T [M] \left\{\Delta\epsilon^{pl}\right\}} = \lambda \qquad (4.1\text{–}49)$$

The accumulated equivalent plastic strain is:

$$\hat{\epsilon}^{pl} = \sum \Delta\hat{\epsilon}^{pl} \qquad (4.1\text{–}50)$$

The isotropic hardening variable, R, can be defined by:

$$R = k + R_o\,\hat{\epsilon}^{pl} + R_\infty\left(1 - e^{-b\hat{\epsilon}^{pl}}\right) \qquad (4.1\text{–}51)$$

where: k = elastic limit

R_o, R_∞, b = material constants characterizing the material isotropic hardening behavior.

The material hardening behavior, R, in equation (4.1–43) can also be defined through bilinear or multilinear isotropic hardening options, which have been discussed early in Section 4.1.8.

The return mapping approach with consistent elastoplastic tangent moduli that was proposed by Simo and Hughes(252) is used for numerical integration of the constitutive equation described above.

4.1.13 Specialization for Anisotropic Plasticity (ANISO)

The anisotropic option uses Hill's(50) yield criterion, which accounts for differences in yield strengths in orthogonal directions, as modified by Shih and Lee(51) accounting for differences in yield strength in tension and compression. An associated flow rule is assumed and work hardening as presented by Valliappan et al(52) is used to update the yield criterion. The yield surface is therefore a distorted circular cylinder that is initially shifted in stress space which expands in size with plastic straining as shown in Figure 4.1–2b.

The equivalent stress for this option is redefined to be:

$$\sigma_e = \frac{1}{3}\{\sigma\}^T[M]\{\sigma\} - \frac{1}{3}\{\sigma\}^T\{L\} \qquad (4.1\text{–}52)$$

where [M] is a matrix which describes the variation of the yield stress with orientation and {L} accounts for the difference between tension and compression yield strengths. {L} can be related to the yield surface translation {α} of equation (4.1–31) (Shih and Lee(51)) and hence the equivalent stress function can be interpreted as having an initial translation or shift. When σ_e is equal to a material parameter K, the material is assumed to yield. The yield criterion (equation (4.1–4)) is then

$$3F = \{\sigma\}^T[M]\{\sigma\} - \{\sigma\}^T\{L\} - K = 0 \qquad (4.1\text{–}53)$$

The material is assumed to have three orthogonal planes of symmetry. The plastic behavior can then be characterized by the stress–strain behavior in the three element coordinate directions and the corresponding shear stress–shear strain behavior. Therefore [M] has the form:

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 & 0 \\ M_{12} & M_{22} & M_{23} & 0 & 0 & 0 \\ M_{13} & M_{23} & M_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & M_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & M_{66} \end{bmatrix} \qquad (4.1\text{–}54)$$

By evaluating the yield criterion (equation (4.1–53)) for all the possible uniaxial stress conditions the individual terms of [M] can be identified:

$$M_{jj} = \frac{K}{\sigma_{+j}\sigma_{-j}} \ , \ j = 1 \text{ to } 6 \qquad (4.1–55)$$

where σ_{+j} and σ_{-j} are the tensile and compressive yield strengths in direction j (j = x, y, z, xy, yz, xz). The compressive yield stress is handled as a positive number here. For the shear yields, $\sigma_{+j} = \sigma_{-j}$. Letting $M_{11} = 1$ defines K to be

$$K = \sigma_{+x}\sigma_{-x} \qquad (4.1–56)$$

The strength differential vector {L} has the form

$$\{L\} = \lfloor L_1 \quad L_2 \quad L_3 \quad 0 \quad 0 \quad 0 \ ^T \rfloor \qquad (4.1–57)$$

and from the uniaxial conditions {L} is defined as

$$L_j = M_{jj}\left(\sigma_{+j} - \sigma_{-j}\right), \ j = 1 \text{ to } 3 \qquad (4.1–58)$$

Assuming plastic incompressibility (i.e. no increase in material volume due to plastic straining) yields the following relationships

$$\begin{aligned} M_{11} + M_{12} + M_{13} &= 0 \\ M_{12} + M_{22} + M_{23} &= 0 \\ M_{13} + M_{23} + M_{33} &= 0 \end{aligned} \qquad (4.1–59)$$

and

$$L_1 + L_2 + L_3 = 0 \qquad (4.1–60)$$

The off–diagonals of [M] are therefore

$$\begin{aligned} M_{12} &= -\frac{1}{2}\left(M_{11} + M_{22} - M_{33}\right) \\ M_{13} &= -\frac{1}{2}\left(M_{11} - M_{22} + M_{33}\right) \\ M_{23} &= -\frac{1}{2}\left(-M_{11} + M_{22} + M_{33}\right) \end{aligned} \qquad (4.1–61)$$

Note that equation (4.1–60) (by means of equations (4.1–55) and (4.1–58)) yields the consistency equation

$$\frac{\sigma_{+x} - \sigma_{-x}}{\sigma_{+x}\sigma_{-x}} + \frac{\sigma_{+y} - \sigma_{-y}}{\sigma_{+y}\sigma_{-y}} + \frac{\sigma_{+z} - \sigma_{-z}}{\sigma_{+z}\sigma_{-z}} = 0. \qquad (4.1\text{–}62)$$

that must be satisfied due to the requirement of plastic incompressibility. Therefore the uniaxial yield strengths are not completely independent.

The yield strengths must also define a closed yield surface, that is, elliptical in cross section. An elliptical yield surface is defined if the following criterion is met:

$$M_{11}^2 + M_{22}^2 + M_{33}^2 - 2\left(M_{11}M_{22} + M_{22}M_{33} + M_{11}M_{33}\right) < 0 \qquad (4.1\text{–}63)$$

Otherwise, the following message is output: “THE DATA TABLE DOES NOT REPRESENT A CLOSED YIELD SURFACE. THE YIELD STRESSES OR SLOPES MUST BE MADE MORE EQUAL”. This further restricts the independence of the uniaxial yield strengths. Since the yield strengths change with plastic straining (a consequence of work hardening), this condition must be satisfied throughout the history of loading. The program checks this condition through an equivalent plastic strain level of 20% (.20).

For an isotropic material,

$$M_{11} = M_{22} = M_{33} = 1$$

$$M_{12} = M_{13} = M_{23} = -1/2$$

$$M_{44} = M_{55} = M_{66} = 3$$

and

$$L_1 = L_2 = L_3 = 0$$

and the yield criterion (equation (4.1–53)) reduces down to the von Mises yield criterion (equation (4.1–33) with {α} = 0).

Work hardening is used for the hardening rule so that the subsequent yield strengths increase with increasing total plastic work done on the material. The total plastic work is defined by equation (4.1–20) where the increment in plastic work is

$$\Delta\kappa = \{\overset{*}{\sigma}\}^T\left\{\Delta\epsilon^{pl}\right\} \qquad (4.1\text{–}64)$$

where: $\{\overset{*}{\sigma}\}$ = average stress over the increment

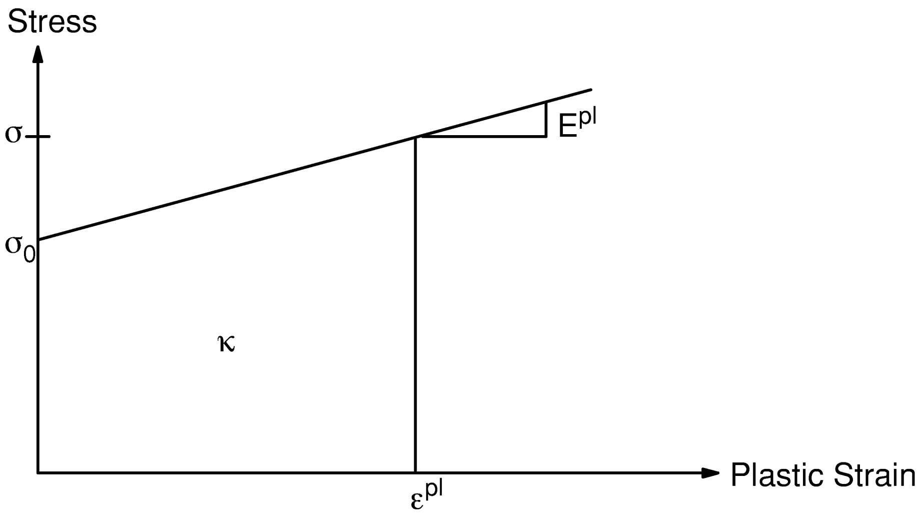


Figure 4.1–6 Plastic Work for a Uniaxial Case

For the uniaxial case the total plastic work is simply

$$\kappa = \frac{1}{2}\,\epsilon^{pl}\,(\sigma_o + \sigma) \qquad (4.1\text{–}65)$$

where the terms are defined as shown in Figure 4.1–6.

For bilinear stress–strain behavior,

$$\sigma = \sigma_o + E^{pl}\,\epsilon^{pl} \qquad (4.1\text{–}66)$$

where the plastic slope E^{pl} is related to the elastic and tangent moduli by

$$E^{pl} = \frac{EE_T}{E - E_T} \qquad (4.1\text{–}67$$

Combining equation (4.1–66) with (4.1–65) and solving for the updated yield stress σ:

$$\sigma = \left\{2E^{pl}\,\kappa + \sigma_o^2\right\}^{\frac{1}{2}} \qquad (4.1\text{–}68)$$

Extending this result to the anisotropic case gives,

$$\sigma_j = \left\{2E_j^{pl}\,\kappa + \sigma_{oj}^2\right\}^{\frac{1}{2}} \qquad (4.1\text{–}69)$$

where j refers to each of the input stress–strain curves. Equation (4.1–69) determines the updated yield stresses by equating the amount of plastic work done on the material to an equivalent amount of plastic work in each of the directions.

The parameters [M] and {L} can then be updated from their definitions (equations (4.1–55) and (4.1–58)) and the new values of the yield stresses. For isotropic materials, this hardening rule reduces to the case of isotropic hardening.

The equivalent plastic strain $\hat{\epsilon}^{pl}$ (output quantity EPEQ) is computed using the tensile x direction as the reference axis by substituting equation (4.1–66) into equation (4.1–65):

$$\hat{\epsilon}^{pl} = \frac{-\sigma_{+x} + \left(\sigma^2_{+x} + 2\kappa\, E^{pl}_{+x}\right)^{\frac{1}{2}}}{E^{pl}_{+x}} \qquad (4.1–70)$$

where the yield stress in the tensile x direction σ_{+x} refers to the initial (not updated) yield stress. The equivalent stress parameter σ^{pl}_{e} (output quantity SEPL) is defined as

$$\hat{\sigma}^{pl}_{e} = \sigma^{pl}_{+x} + \hat{E}^{pl}_{+x}\, \epsilon \qquad (4.1–71)$$

where again σ_{+x} is the initial yield stress.

4.1.14 Specialization for Drucker–Prager (DP)

This option uses the Drucker–Prager yield criterion with either an associated or nonassociated flow rule. The yield surface does not change with progressive yielding, hence there is no hardening rule and the material is elastic– perfectly plastic (Figure 4.1–1f). The equivalent stress for Drucker–Prager is

$$\sigma_e = 3\beta\sigma_m + \left[\frac{1}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} \qquad (4.1–72)$$

where: σ_m = the mean or hydrostatic stress $= \frac{1}{3}\left(\sigma_x + \sigma_y + \sigma_z\right)$

{s} = the deviatoric stress (equation (4.1–32))

β = material constant

[M] = as defined with equation (4.1–31)

This is a modification of the von Mises yield criterion (equation (4.1–31) with {α} = {0}) that accounts for the influence of the hydrostatic stress component: the higher the hydrostatic stress (confinement pressure) the higher the yield strength. β is a material constant which is given as

$$\beta = \frac{2\sin\phi}{\sqrt{3}\ (3 - \sin\phi)} \tag{4.1–73}$$

where: ϕ = the input angle of internal friction

The material yield parameter is defined as

$$\sigma_y = \frac{6c\ \cos\phi}{\sqrt{3}\ (3 - \sin\phi)} \tag{4.1–74}$$

where: c = the input cohesion value

The yield criterion (equation (4.1–4)) is then

$$F = 3\beta\sigma_m + \left[\frac{1}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}} - \sigma_y = 0 \tag{4.1–75}$$

This yield surface is a circular cone (Figure 4.1–2c) with the material parameters (equations (4.1–73) and (4.1–74)) chosen such that it corresponds to the outer aspices of the hexagonal Mohr–Coulomb yield surface, Figure 4.1–7 .

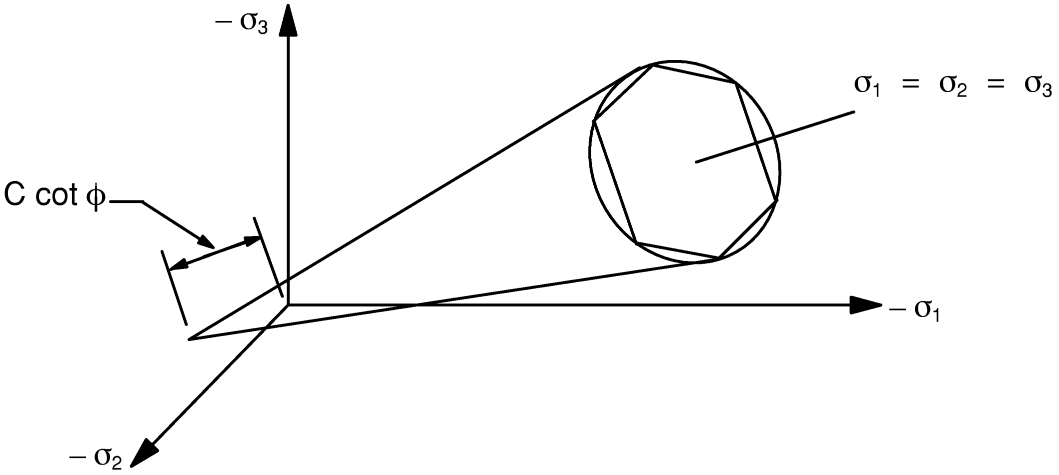


Figure 4.1–7 Drucker–Prager and Mohr–Coulomb Yield Surfaces

$\left\{\frac{\partial F}{\partial \sigma}\right\}$ is readily computed as

$$\left\{\frac{\partial F}{\partial \sigma}\right\} = \beta \lfloor 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \rfloor^T + \frac{1}{\left[\frac{1}{2}\{s\}^T[M]\{s\}\right]^{\frac{1}{2}}}\{s\} \tag{4.1–76}$$

$\left\{\frac{\partial Q}{\partial \sigma}\right\}$ is similar, however β is evaluated using ϕ_f (the input “dilatancy” constant). When $\phi_f = \phi$, the flow rule is associated and plastic straining occurs normal to the yield surface and there will be a volumetric expansion of the material with plastic strains. If ϕ_f is less than ϕ there will be less volumetric expansion and if ϕ_f is zero, there will be no volumetric expansion.

The equivalent plastic strain $\hat{\epsilon}^{pl}$ (output quantity EPEQ) is defined by equation (4.1–38) and the equivalent stress parameter $\hat{\sigma}_e^{pl}$ (output quantity SEPL) is defined as

$$\hat{\sigma}_e^{pl} = \sqrt{3}\,\left(\sigma_y - 3\beta\sigma_m\right) \tag{4.1–77}$$

The equivalent stress parameter is interpreted as the von Mises equivalent stress at yield at the current hydrostatic stress level. Therefore for any integration point undergoing yielding (stress ratio SRAT > 1), $\hat{\sigma}_e^{pl}$ (SEPL) should be close to the actual von Mises equivalent stress (SIGE) at the converged solution.

4.2 Rate–Dependent Plasticity

Rate–dependent plasticity (or viscoplasticity) is characterized by the irreversible straining that occurs in a material over time. The plastic strains are assumed to develop as a function of the strain rate. The ANSYS program provides two options to characterize different types of rate–dependent material behaviors, and they are only available with the large strain solids VISCO106, VISCO107, and VISCO108. These options are:

TB *Lab*	Material Behavior Option
ANAND	Anand's Model
USER	User Specified Behavior (see Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide*)

(Note that a rate–dependent model may be incorporated in the USER option of the rate–independent plasticity).

4.2.1 Overview

For metals, it has long been recognized that the notion of rate–independence of plastic response is only a convenient aproximation at low temperature. Although in reality the plastic flow due to dislocation even at low temperature is not truly rate–independent, the use of rate–independent plastic models is quite common. Here we present a rate–dependent plasticity model as proposed by Anand(159) and Brown et al.(147). This rate–dependent model differs from the rate–independent model in that there is no explicit yield condition, and no loading/unloading criterion is used. Instead, plastic flow is assumed to take place at all non–zero stress values, although at low stresses the rate of plastic flow may be immeasurably small. Further, the equivalent plastic strain rate, which is determined by the consistency condition in the rate–independent model, needs to be prescribed by an appropriate constitutive function in the rate–dependent model.

4.2.2 Theory

There are two basic features in Anand's model applicable to isotropic rate–dependent constitutive model for metals. First, there is no explicit yield surface, rather the instantaneous response of the material is dependent on its current state. Secondly, a single scalar internal variable "s", called the *deformation resistance*, is used to represent the isotropic resistance to inelastic flow of the material. The specifics of this constitutive equation are the flow equation:

$$d^p = A\, e^{-\frac{Q}{R\theta}} \left[\sinh\left(\xi\, \frac{\sigma}{s}\right)\right]^{\frac{1}{m}} \tag{4.2–1}$$

and the evolution equation:

$$\dot{s} = \left\{ h_o \quad (|\, B\, |)^a\, \frac{B}{|B|} \right\} d^p \tag{4.2–2}$$

Equation (4.2–2) allows modelling not only strain hardening, but also strain softening.

where: $B = 1 - \frac{s}{s^*}$

with

$$s^* = \hat{s} \left[\frac{d^p}{A}\, e^{\frac{Q}{R\theta}} \right]^n \tag{4.2–3}$$

where: d^p = effective inelastic deformation rate
σ = effective Cauchy stress
s = deformation resistance
s^* = saturation value of deformation resistance
$\dot{s}$ = time derivative of deformation resistance
θ = absolute temperature

The remaining terms are defined in Table 4.2–1. All terms must be positive, except constant a, which must be 1.0 or greater. The inelastic strain rate in Anand's definition of material is temperature and stress dependent as well as dependent on the rate of loading. Determination of the material parameters is performed by curve–fitting a series of the stress–strain data at various temperatures and strain rates as in Anand(159) or Brown et al.(147).

4.2.3 Implementation

A consistent stress update procedure which is equivalent to the Euler backward scheme used to enforce the consistency condition and the evolution equation (4.2–2) at the end of the time step. The consistency condition in rate–dependent plasticity requires that the stress and strain values are consistent via equations (4.1–11), (4.1–12) and the rate–dependent counterpart to equation (4.1–14).

The deformation resistance s is output as output quantity PSV. The accumulated plastic work (see equation (4.1–20)) is output as output quantity PLWK.

Table 4.2–1 Material Parameter Units for Anand Model

TBDATA Constant	Parameter	Meaning	Units
1	s_o	initial value of deformation resistance	stress e.g. psi, MPa
2	Q/R	Q = activation energy	energy / volume e.g. kJ / mole
		R = universal gas constant	energy / (volume temperature) e.g. kJ / (mole – °K)
3	A	pre–exponential factor	1 / time e.g. 1 / second
4	ξ	multiplier of stress	dimensionless
5	m	strain rate sensitivity of stress	dimensionless
6	h_o	hardening / softening constant	stress e.g. psi, MPa
7	$\hat{s}$	coefficient for deformation resistance saturation value	stress e.g. psi, MPa
8	n	strain rate sensitivity of saturation (deformation resistance) value	dimensionless
9	a	strain rate sensitivity of hardening or softening	dimensionless

kJ = kilojoules, °K = degrees Kelvin

4.3 Creep

4.3.1 Definition and Limitations

Creep is defined as material deforming under load over time in such a way as to tend to relieve the stress. Creep may also be a function of temperature and neutron flux level. The term “relaxation” has also been used interchangeably with creep. The material is assumed to be isotropic and the basic solution technique used is the initial–stiffness Newton–Raphson method.

The options available for creep are described in Section 2.5.7 of the *ANSYS Elements Reference.* Four different types of creep are available and the effects of the first three may be added together except as noted:

Primary creep is accessed with C_6 (C_i values refer to the ith value given in the **TBDATA** command with **TB**,CREEP). The creep calculations are bypassed if $C_1 = 0$.

Secondary creep is accessed with C_{12}. These creep calculations are bypassed if $C_7 = 0$. They are also bypassed if a primary creep strain was computed using the option $C_6 = 9$, 10, 11, 13, 14, or 15, since they include secondary creep in their formulations.

Irradiation induced creep is accessed with C_{66}.

User–specified creep may be accessed with $C_6 = 100$. See Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide* for more details.

The creep calculations are also bypassed if:

1. (change of time) $\leq 10^{-6}$
2. (input temperature + T_{off}) ≤ 0 where T_{off} = input on **TOFFST** command.
3. For $C_6 = 0$ case: A special effective strain based on ε^e and ε^{cr} is computed. A bypass occurs if it is equal to zero.

4.3.2 Calculation of Creep

The creep equations are integrated with an explicit Euler forward algorithm, which is efficient for problems having small amounts of contained creep strains. A modified total strain is computed:

$$\left\{\epsilon_n^{'}\right\} = \{\epsilon_n\} - \left\{\epsilon_n^{pl}\right\} - \left\{\epsilon_n^{th}\right\} - \left\{\epsilon_{n-1}^{cr}\right\} \tag{4.3–1}$$

This equation is analogous to equation (4.1–15) for plasticity. The superscripts are described with Notation (Section 1.1.2) and subscripts refer to the time point n. An equivalent modified total strain is defined as:

$$\epsilon_{et} = \frac{1}{\sqrt{2}} \left[\left(\epsilon_x^{'} - \epsilon_y^{'}\right)^2 + \left(\epsilon_y^{'} - \epsilon_z^{'}\right)^2 + \left(\epsilon_z^{'} - \epsilon_x^{'}\right)^2 + \frac{3}{2}\left(\gamma_{xy}^{'}\right)^2 + \frac{3}{2}\left(\gamma_{yz}^{'}\right)^2 + \frac{3}{2}\left(\gamma_{zx}^{'}\right)^2 \right]^{\frac{1}{2}} \tag{4.3–2}$$

Also an equivalent stress is defined by:

$$\sigma_e = 2G\,\epsilon_{et} \tag{4.3–3}$$

where: G = shear modulus = E / (2 (1+ν))
E = Young’s modulus (input as EX on **MP** command)
ν = Poisson’s ratio (input as PRXY or NUXY on **MP** command)

The equivalent creep strain increment $(\Delta\epsilon^{cr})$ is computed as a scalar quantity from the relations given in Section 2.5.7 of the *ANSYS Elements Reference* and is normally positive. If C_{11} = 1, a decaying creep rate is used rather than a rate that is constant over the time interval. This option is normally not recommended, as it can seriously underestimate the total creep strain where primary stresses dominate. The modified equivalent creep strain increment $(\Delta\epsilon_m^{cr})$, which would be used in place of the equivalent creep strain increment $(\Delta\epsilon^{cr})$ if C_{11} = 1, is computed as:

$$\Delta\epsilon_m^{cr} = \epsilon_{et}\left(1 - \frac{1}{e^A}\right) \tag{4.3–4}$$

where: e = 2.718281828 (base of natural logarithms)
A = $\Delta\varepsilon^{cr} / \varepsilon_{et}$

Next, the creep ratio (a measure of the increment of creep strain) for this integration point (C_s) is computed as:

$$C_s = \frac{\Delta\epsilon^{cr}}{\epsilon_{et}} \tag{4.3–5}$$

The largest value of C_s for all elements at all integration points for this iteration is called C_{max} and is output with the label “CREEP RATIO”.

The creep strain increment is then converted to a full strain tensor. N_c is the number of strain components for a particular type of element. If N_c = 1,

$$\Delta\epsilon_x^{cr} = \Delta\epsilon^{cr}\left(\frac{\epsilon_x'}{\epsilon_{et}}\right) \tag{4.3–6}$$

Note that the term in brackets is either +1 or –1. If $N_c = 4$,

$$\Delta\epsilon_x^{cr} = \frac{\Delta\epsilon^{cr}}{\epsilon_{et}}\ \frac{\left(2\epsilon_x' - \epsilon_y' - \epsilon_z'\right)}{2\ (1 + \nu)} \tag{4.3–7}$$

$$\Delta\epsilon_y^{cr} = \frac{\Delta\epsilon^{cr}}{\epsilon_{et}}\ \frac{\left(2\epsilon_y' - \epsilon_z' - \epsilon_x'\right)}{2\ (1 + \nu)} \tag{4.3–8}$$

$$\Delta\epsilon_z^{cr} = -\ \Delta\epsilon_x^{cr} - \Delta\epsilon_y^{cr} \tag{4.3–9}$$

$$\Delta\epsilon_{xy}^{cr} = \frac{\Delta\epsilon^{cr}}{\epsilon_{et}}\ \frac{3}{2(1 + \nu)}\ \gamma_{xy}' \tag{4.3–10}$$

The first three components are the three normal strain components, and the fourth component is the shear component. If $N_c = 6$, components 1 through 4 are the same as above, and the two additional shear components are:

$$\Delta\epsilon_{yz}^{cr} = \frac{\Delta\epsilon^{cr}}{\epsilon_{et}}\ \frac{3}{2(1 + \nu)}\ \gamma_{yz}' \tag{4.3–11}$$

$$\Delta\epsilon_{xz}^{cr} = \frac{\Delta\epsilon^{cr}}{\epsilon_{et}}\ \frac{3}{2(1 + \nu)}\ \gamma_{xz}' \tag{4.3–12}$$

Next, the elastic strains and the total creep strains are calculated as follows, using the example of the x–component:

$$\left(\epsilon_x^{el}\right)_n = \left(\epsilon_x'\right)_n - \Delta\epsilon_x^{cr} \tag{4.3–13}$$

$$(\epsilon_x^{cr})_n = (\epsilon_x^{cr})_{n-1} + \Delta\epsilon_x^{cr} \tag{4.3–14}$$

Stresses are based on $\left(\epsilon_x'\right)_n$. This gives the correct stresses for imposed force problems and the maximum stresses during the time step for imposed displacement problems.

4.3.3 Time Step Size

A stability limit is placed on the time step size (Zienkiewicz and Cormeau(154)). This is because an explicit integration procedure is used in which the stresses and strains are referred to time t_{n-1} (however, the temperature is at time t_n). The creep strain rate is calculated using time t_n. It is recommended to use a time step such that the creep ratio C_{max} is less than 0.10. If the creep ratio exceeds 0.25, the run terminates with the message: “CREEP RATIO OF . . . EXCEEDS STABILITY LIMIT OF .25.” Section 15.6 discusses the automatic time stepping algorithm which may be used with creep in order to increase or decrease the time step as needed for an accurate yet efficient solution.

4.4 Nonlinear Elasticity

4.4.1 Overview and Guidelines for Use

The ANSYS program provides a capability to model nonlinear (multilinear) elastic materials. Unlike plasticity, no energy is lost (the process is conservative).

The capability is accessed with **TBPT** commands with **TB**,MELAS and is available with the elements indicated in Figure 4.4–1. Figure 4.4–1 represents the stress–strain behavior of this option. Note that the material unloads along the same curve, so that no permanent inelastic strains are induced.

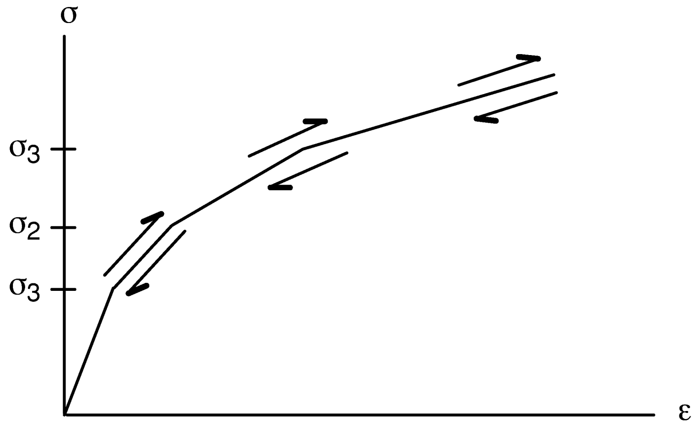


Figure 4.4–1 Stress–Strain Behavior for Nonlinear Elasticity

The total strain components $\{\varepsilon_n\}$ are used to compute an equivalent total strain measure:

$$\epsilon^t_e = \frac{1}{\sqrt{2}(1 + \nu)} \left[\left(\epsilon_x - \epsilon_y\right)^2 + \left(\epsilon_y - \epsilon_z\right)^2 + (\epsilon_z - \epsilon_x)^2 + \frac{3}{2}\left(\epsilon_{xy}\right)^2 + \frac{3}{2}\left(\epsilon_{yz}\right)^2 + \frac{3}{2}(\epsilon_{xz})^2 \right]^{\frac{1}{2}} \tag{4.4–1}$$

ϵ^t_e is used with the input stress–strain curve to get an equivalent value of stress σ_e.

The elastic (linear) component of strain can then be computed:

$$\left\{\epsilon_n^{el}\right\} = \frac{\sigma_e}{E\ \epsilon_e^t} \quad \{\epsilon_n\} \tag{4.4–2}$$

and the “plastic” or nonlinear portion is therefore:

$$\left\{\epsilon_n^{pl}\right\} = \{\epsilon_n\} - \left\{\epsilon_n^{el}\right\} \tag{4.4–3}$$

In order to avoid an unsymmetric matrix, only the symmetric portion of the tangent stress–strain matrix is used:

$$\left[D_{ep}\right] = \frac{\sigma_e}{E\epsilon_e} \ [D] \tag{4.4–4}$$

which is the secant stress–strain matrix.

4.5 Hyperelasticity

Introduction

Hyperelasticity refers to materials which can experience finite elastic deformation that is completely recoverable. Rubber and many other polymer materials fall into this category. The stresses for these materials are usually derived from strain energy density functions. The ANSYS program provides three options to characterize this type of behavior.

- The Mooney–Rivlin option (***MOONEY** command or **TBDATA** commands with **TB**,MOONEY) is a material law suitable for incompressible materials.
- The Blatz–Ko option (KEYOPT(2) = 1 on HYPER84 and HYPER86) is applicable to compressible foam or foam–type materials.
- The Arruda–Boyce option (**TBDATA** commands with **TB**,BOYCE) is a micromechanics–based material model for compressible rubbers and elastic polymers

Table 4.0–1 indicates which elements support the hyperelastic option. Basic element formulations are found in Sections 14.58 and 14.86. Because of the type of subject matter, the component form of tensor notation will be used in this section.

Definitions

A material is said to be hyperelastic if there exists an elastic potential function W (or strain energy density function) which is a scalar function of one of the strain or deformation tensors, whose derivative with respect to a strain component determines the corresponding stress component. This can be expressed by:

$$S_{ij} = \frac{\partial W}{\partial E_{ij}} \equiv 2\frac{\partial W}{\partial C_{ij}} \qquad (4.5\text{–}1)$$

where:
S_{ij} = components of the second Piola–Kirchhoff stress tensor
W = strain energy function per unit undeformed volume
E_{ij} = components of the Lagrangian strain tensor
C_{ij} = components of the right Cauchy–Green deformation tensor

The Lagrangian strain may be expressed as follows:

$$E_{ij} = \frac{1}{2}\left(C_{ij} - \delta_{ij}\right) \qquad (4.5\text{–}2)$$

where: $\delta_{ij} = \begin{cases} 1.0 & i = j \\ 0.0 & i \neq j \end{cases}$

The deformation tensor C_{ij} is comprised of the products of the deformation gradients f_{ij}

$$C_{ij} = f_{ik} f_{kj} = \text{component of the Cauchy–Green deformation tensor} \qquad (4.5–3)$$

where: $f_{ij} = \frac{\partial x_i}{\partial X_j}$

X_i = undeformed position of a point in direction i

x_i = $X_i + u_i$ = deformed position of a point in direction i

u_i = displacement of a point in direction i

The theory employed for the family of hyperelastic elements is limited to isotropic materials However, no limits are placed on the magnitude of the strain and the results are path independent.

4.5.1 Mooney–Rivlin and Blatz–Ko Models

Presently HYPER84 and HYPER86 elements offer both the Mooney–Rivlin and Blatz–Ko models, but HYPER56, HYPER58, HYPER74, HYPER158 and SHELL181 elements use only the Mooney–Rivlin material law.

The Mooney–Rivlin constitutive law is a reasonable model for representing the stress–strain behavior of some nearly incompressible natural rubbers (Rivlin(89), Mooney(91)). The Mooney–Rivlin strain energy density function, for HYPER84 and HYPER86, has the following form:

$$W = a_{10}\left(\bar{I}_1 - 3\right) + a_{01}\left(\bar{I}_2 - 3\right) + \beta\left(\bar{I}_3^{\,2} - \bar{I}_3^{\,-2}\right)^2 \qquad (4.5–4)$$

where: $\bar{I}_i$ = reduced in the i^{th} direction which are given by:

$$\bar{I}_1 = I_1 I_3^{-1/3}$$

$$\bar{I}_2 = I_2 I_3^{-2/3}$$

$$\bar{I}_3 = I_3^{1/2}$$

a_{10}, a_{01} = Mooney–Rivlin material constants (input using **TBDATA** command with **TB**,MOONEY)

$\beta = \frac{(1 + \nu)}{(1 - 2\nu)} \frac{a_{10} + a_{01}}{24}$

ν = Poisson's ratio (input as PRXY or NUXY on **MP** command, must be less than 0.50)

I_i = invariants of the right Cauchy–Green deformation tensor C_{ij} given as:

$$I_1 = C_{ii} \tag{4.5–5}$$

$$I_2 = \frac{1}{2}\left(I_1^2 - C_{ij}\ C_{ij}\right) \tag{4.5–6}$$

$$I_3 = \det C_{ij} \ = \text{volume change ratio} \tag{4.5–7}$$

Note that for small strains, $2(a_{10}+a_{01})$ represents the shear modulus and $6(a_{10}+a_{01})$ represents the Young's modulus.

For the elements HYPER56, HYPER58, HYPER74, HYPER158 and SHELL181 the two, five and nine parameter Mooney–Rivlin models are available. The strain energy density function for these elements is given in polynomial form by:

$$W = \sum_{k+\ell=1}^{N} a_{k\ell}\left(\bar{I}_1 - 3\right)^k \left(\bar{I}_2 - 3\right)^\ell + 1/2\ \kappa \left(\bar{I}_3 - 1\right)^2 \tag{4.5–8}$$

where: $a_{k\ell}$ = constants of the nine–parameter cubic Mooney–Rivlin relationship (input with **TB**,MOONEY using the **TBDATA** command or the ***MOONEY** command with experimental data input on STRS and STRN fields)

$\varkappa$ = bulk modulus = $\dfrac{2\ (a_{10} + a_{01})}{(1 - 2\nu)}$

Setting N=1 in equation (4.5–8), we obtain the strain energy density function for the two parameter Mooney–Rivlin model:

$$W = a_{10}\left(\bar{I}_1 - 3\right) + a_{01}\left(\bar{I}_2 - 3\right) + 1/2\ \kappa \left(\bar{I}_3 - 1\right)^2 \tag{4.5–9}$$

Likewise, setting N = 2 and N = 3 in equation (4.5–8), we obtain analogous strain energy density functions for the five (equation (4.5–10)) and nine (equation (4.5–11)) parameter Mooney–Rivlin models:

$$\begin{aligned} W = {} & a_{10}\left(\bar{I}_1 - 3\right) + a_{01}\left(\bar{I}_2 - 3\right) + a_{20}\left(\bar{I}_1 - 3\right)^2 + a_{11}\left(\bar{I}_1 - 3\right)\left(\bar{I}_2 - 3\right) \\ & + a_{02}\left(\bar{I}_2 - 3\right)^2 + 1/2\ \kappa \left(\bar{I}_3 - 1\right)^2 \end{aligned} \tag{4.5–10}$$

$$
\begin{aligned}
W = {} & a_{10}\left(\bar{I}_1 - 3\right) + a_{01}\left(\bar{I}_2 - 3\right) + a_{20}\left(\bar{I}_1 - 3\right)^2 + a_{11}\left(\bar{I}_1 - 3\right)\left(\bar{I}_2 - 3\right) \\
& + a_{02}\left(\bar{I}_2 - 3\right)^2 + a_{30}\left(\bar{I}_1 - 3\right)^3 + a_{21}\left(\bar{I}_1 - 3\right)^2\left(\bar{I}_2 - 3\right) \qquad (4.5\text{–}11) \\
& + a_{12}\left(\bar{I}_1 - 3\right)\left(\bar{I}_2 - 3\right)^2 + a_{03}\left(\bar{I}_2 - 3\right)^3 + 1/2\,\kappa\left(\bar{I}_3 - 1\right)^2
\end{aligned}
$$

Note that the last term in equations (4.5–8) through (4.5–11) always represents the hydrostatic (volumetric) work. For SHELL181 incompressibility is enforced if ν is equal or almost equal to 0.5.

The Blatz–Ko constitutive law is used to model compressible foam–type polyurethane rubbers (Kao(90), Blatz(92)). The Blatz–Ko strain energy density, as used in HYPER84 and HYPER86, is given as follows:

$$W = \frac{G}{2}\left(\frac{I_2}{I_3} + 2\sqrt{I_3} - 5\right) \qquad (4.5\text{–}12)$$

where: $G = \frac{E}{2(1+\nu)}$

E = Young's modulus (input quantity EX on **MP** command)

Output Quantities

Stresses (output quantities S) are true (Cauchy) stresses in the element coordinate system. They are computed from the second Piola–Kirchoff stresses using:

$$\sigma_{ij} = \frac{\rho}{\rho_o} f_{ik}\, S_{kl}\, f_{jl} = \frac{1}{\sqrt{I_3}} f_{ik}\, S_{kl}\, f_{jl} \qquad (4.5\text{–}13)$$

where: ϱ, ϱ_o = mass densities in the current and initial configurations

Strains (output quantity EPEL) are the Hencky (logarithmic) strains (see equation (3.1–6)). They are in the element coordinate system.

Determining Mooney–Rivlin Material Constants

The hyperelastic constants in the strain energy density function of a material determine its mechanical response. Therefore, in order to obtain successful results during a hyperelastic analysis, it is necessary to accurately assess the Mooney–Rivlin constants of the materials being examined. Mooney–Rivlin constants are generally derived for a material using experimental stress–strain data. It is recommended that this test data be taken from several modes of deformation over a wide range of strain values. In fact, it has been observed that to achieve stability, the Mooney–Rivlin constants should be fit using test data in at least as many deformation states as will be experienced in the analysis.

For hyperelastic materials, simple deformation tests (consisting of six deformation modes) can be used to accurately characterize the Mooney–Rivlin constants. These constants are determined using the ***MOONEY** command. All the available laboratory test data will be used to determine the Mooney–Rivlin hyperelastic material constants. The six different deformation modes are graphically illustrated in Figure 4.5–1. Combinations of data from multiple tests will enhance the characterization of the hyperelastic behavior of a material.

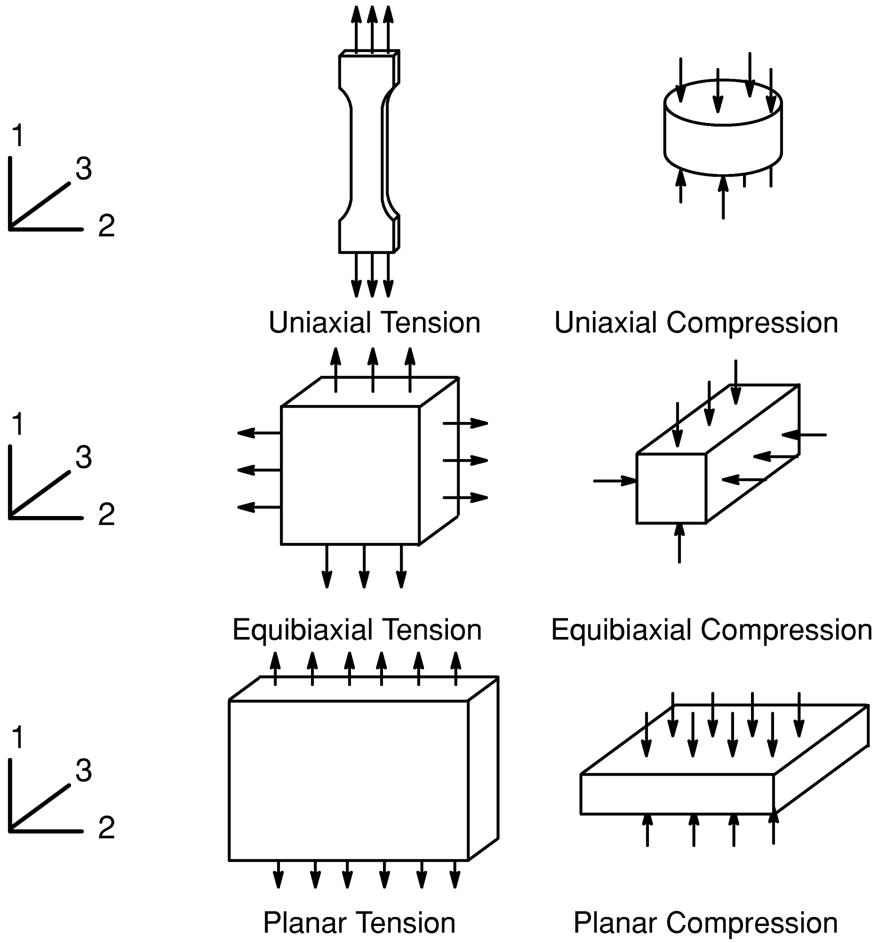


Figure 4.5–1 Illustration of Deformation Modes

Although the algorithm accepts up to six different deformation states, it can be shown that apparently different loading conditions have identical deformations, and are thus equivalent. Superposition of tensile or compressive hydrostatic stresses on a loaded incompressible body results in different stresses, but does not alter deformation of a material. As depicted in Figure 4.5–2, we find that upon the addition of hydrostatic stresses, the following modes of deformation are identical:

1. Uniaxial Tension and Equibiaxial Compression.

2. Uniaxial Compression and Equibiaxial Tension.
3. Planar Tension and Planar Compression.

With several equivalent modes of testing, we are left with only three independent deformation states for which one can obtain experimental data.

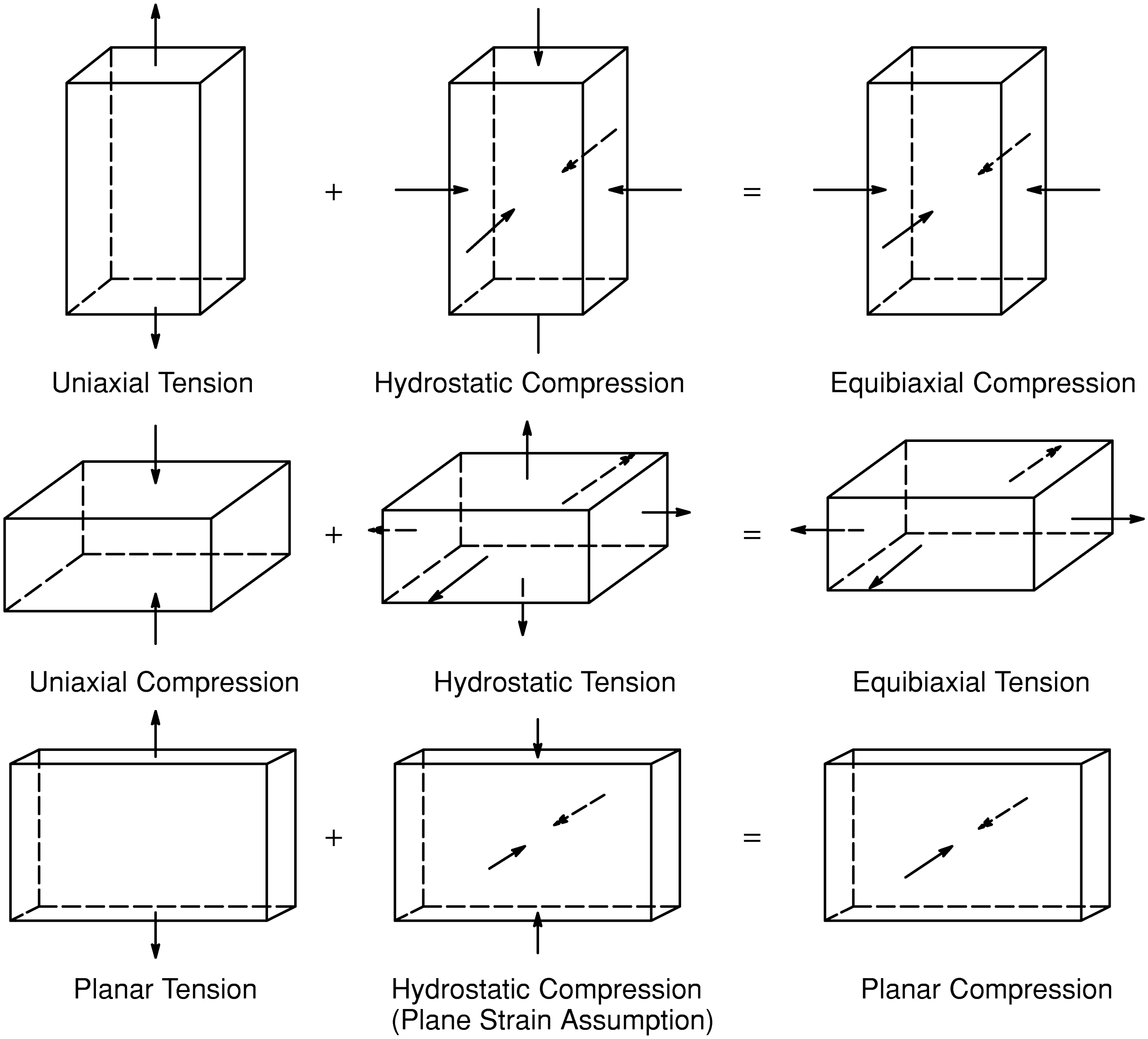


Figure 4.5–2   Equivalent Deformation Modes

The following sections outline the development of hyperelastic stress relationships for each independent testing mode. In the analyses, the coordinate system is chosen to coincide with the principal directions of deformation. Thus, the right Cauchy–Green strain tensor can be written in matrix form by:

$$[C] = \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \qquad (4.5–14)$$

where: λ_i = $1 + \varepsilon_i \equiv$ principal stretch ratio in the i^{th} direction

ε_i = principal value of the engineering strain tensor in the ith direction

The principal invariants of C_{ij}, equations (4.5–5) through (4.5–7), then become:

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 \qquad (4.5–15)$$

$$I_2 = \lambda_1^2\lambda_2^2 + \lambda_1^2\lambda_3^2 + \lambda_2^2\lambda_3^2 \qquad (4.5–16)$$

$$I_3 = \lambda_1^2\lambda_2^2\lambda_3^2 \qquad (4.5–17)$$

For each mode of deformation, fully incompressible material behavior is also assumed so that third principal invariant, I_3, is identically one:

$$\lambda_1^2\lambda_2^2\lambda_3^2 = 1 \qquad (4.5–18)$$

Finally, the hyperelastic Piolla–Kirchoff stress tensor, equation (4.5–1) can be algebratically manipulated to determine components of the Cauchy (true) stress tensor. In terms of the principal invariants of the right Cauchy–Green strain tensor, the Cauchy stress components, as determined from equation (4.5–1), can be shown to be:

$$\sigma_{ij} = -\,p\;\delta_{ij} + 2\;\partial W/\partial I_1\;C_{ij} - 2\partial W/\partial I_2\;C_{ij}^{-1} \qquad (4.5–19)$$

where: p = pressure

Uniaxial Tension (Equivalently, Equibiaxial Compression)

As shown in Figure 4.5–1, a hyperelastic specimen is loaded along one of its axis during a uniaxial tension test. For this deformation state, the principal stretch ratios in the directions orthogonal to the ‘pulling’ axis will be identical. Therefore, during uniaxial tension, the principal stretches, λ_i, are given by:

$$\lambda_1 = \text{stretch in direction being loaded} \qquad (4.5–20)$$

$$\lambda_2 = \lambda_3 = \text{stretch in directions not being loaded} \tag{4.5–21}$$

Due to incompressibility (equation (4.5–18)):

$$\lambda_2 \lambda_3 = \lambda_1^{-1} \tag{4.5–22}$$

and with equation (4.5–21),

$$\lambda_2 = \lambda_3 = \lambda_1^{-1/2} \tag{4.5–23}$$

For uniaxial tension, the first and second strain invariants then become:

$$I_1 = \lambda_1^2 + 2\lambda_1^{-1} \tag{4.5–24}$$

and

$$I_2 = 2\lambda_1 + \lambda_1^{-2} \tag{4.5–25}$$

Substituting the uniaxial tension principal stretch ratio values into the Cauchy stress equation (4.5–19), we obtain the following stresses in the 1 and 2 directions:

$$\sigma_{11} = -\, p + 2\ \partial W/\partial I_1\ \lambda_1^2 - 2\partial W/\partial I_2\ \lambda_1^{-2} \tag{4.5–26}$$

and

$$\sigma_{22} = -\, p + 2\ \partial W/\partial I_1\ \lambda_1^{-1} - 2\partial W/\partial I_2\ \lambda_1 = 0 \tag{4.5–27}$$

Subtracting equation (4.5–27) from equation (4.5–26), we obtain the principal true stress for uniaxial tension:

$$\sigma_{11} = 2 \left(\lambda_1^2 - \lambda_1^{-1}\right) \left[\partial W/\partial I_1 + \lambda_1^{-1}\ \partial W/\partial I_2\right] \tag{4.5–28}$$

Equibiaxial Tension (Equivalently, Uniaxial Compression)

During an equibiaxial tension test, a hyperelastic specimen is equally loaded along two of its axes, as shown in Figure 4.5–1. For this case, the principal stretch ratios in the directions being loaded are identical. Hence, for equibiaxial tension, the principal stretches, λ_i, are given by:

$$\lambda_1 = \lambda_2 = \text{stretch ratio in direction being loaded} \qquad (4.5\text{–}29)$$

$$\lambda_3 = \text{stretch in direction not being loaded} \qquad (4.5\text{–}30)$$

Utilizing incompressibility (equation (4.5–18)), we find:

$$\lambda_3 = \lambda_1^{-2} \qquad (4.5\text{–}31)$$

For equibiaxial tension, the first and second strain invariants then become:

$$I_1 = 2\lambda_1^2 + \lambda_1^{-4} \qquad (4.5\text{–}32)$$

and

$$I_2 = \lambda_1^4 + 2\lambda_1^{-2} \qquad (4.5\text{–}33)$$

Substituting the principal stretch ratio values for equibiaxial tension into the Cauchy stress equation (4.5–19), we obtain the stresses in the 1 and 3 directions:

$$\sigma_{11} = -\, p + 2\ \partial W/\partial I_1\ \lambda_1^2 - 2\partial W/\partial I_2\ \lambda_1^{-2} \qquad (4.5\text{–}34)$$

and

$$\sigma_{33} = -\, p + 2\ \partial W/\partial I_1\ \lambda_1^{-4} - 2\partial W/\partial I_2\ \lambda_1^4 = 0 \qquad (4.5\text{–}35)$$

Subtracting equation (4.5–34) from equation (4.5–35), we obtain the principal true stress for equibiaxial tension:

$$\sigma_{11} = 2\left(\lambda_1^2 - \lambda_1^{-4}\right)\left[\partial W/\partial I_1 + \lambda_1^2\ \partial W/\partial I_2\right] \qquad (4.5\text{–}36)$$

Pure Shear (Uniaxial Tension and Uniaxial Compression in Orthogonal Directions)

Pure shear deformation experiments on hyperelastic materials are generally performed by loading thin, short and wide rectangular specimens, as shown in Figure 4.5–3. For pure shear, plane strain is generally assumed so that there is no deformation in the 'wide' direction of the specimen: $\lambda_2 = 1$.

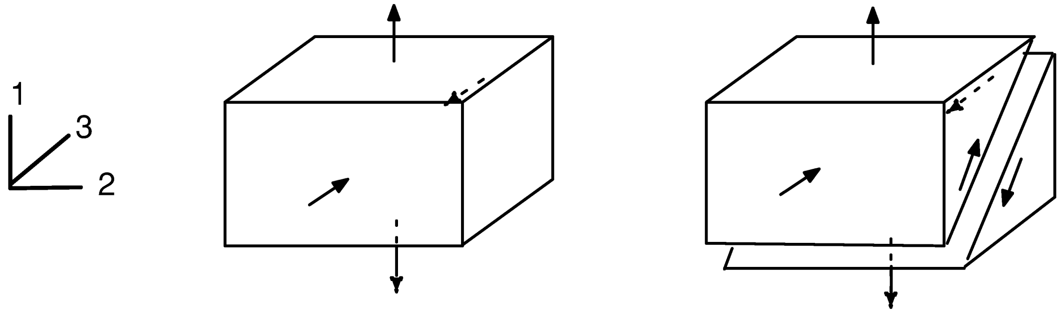


Figure 4.5–3 Pure Shear

Due to incompressibility (equation (4.5–18)), it is found that:

$$\lambda_3 = \lambda_1^{-1} \qquad (4.5–37)$$

For pure shear, the first and second strain invariants are:

$$I_1 = \lambda_1^2 + \lambda_1^{-2} + 1 \qquad (4.5–38)$$

and

$$I_2 = \lambda_1^2 + \lambda_1^{-2} + 1 \qquad (4.5–39)$$

Substituting the principal stretch ratio values for pure shear into the Cauchy stress equation (4.5–19), we obtain the following stresses in the 1 and 3 directions:

$$\sigma_{11} = -p + 2\ \partial W/\partial I_1\ \lambda_1{}^2 - 2\ \partial W/\partial I_2\ \lambda_1^{-2} \qquad (4.5–40)$$

and

$$\sigma_{33} = -p + 2\ \partial W/\partial I_1\ \lambda_1{}^2 - 2\ \partial W/\partial I_2\ \lambda_1^{-2} = 0 \qquad (4.5–41)$$

Subtracting equation (4.5–38) from equation (4.5–39), we obtain the principal pure shear true stress equation:

$$\sigma_{11} = 2\left(\lambda_1^{\,2} - \lambda_1^{-2}\right)\left[\partial W/\partial I_1 + \partial W/\partial I_2\right] \tag{4.5–42}$$

Least Square Fit Analysis

By performing a least squares fit analysis the Mooney–Rivlin constants can be determined from experimental stress–strain data and equations (4.5–28), (4.5–36), and (4.5–42). Briefly, the least squares fit minimizes the sum of squared error between experimental and Cauchy predicted stress values. The sum of the squared error is defined by:

$$E^2 = \sum_{i=1}^{n}\left(\sigma_i - \tilde{\sigma}_i\,(a_{ij})\right)^2 \tag{4.5–43}$$

where:

E = relative error

σ_i = experimental stress values

$\tilde{\sigma}_i(a_{ij})$ = Cauchy stress values (function of Mooney–Rivlin constants)

n = number of experimental data points

Equation (4.5–43) is minimized by setting the variation of the squared error to zero: $\delta E^2 = 0$. This yields a set of simultaneous equations which can be used to solve for the Mooney–Rivlin constants:

$$\begin{aligned} &\partial E^2/\partial a_{10} = 0 \\ &\partial E^2/\partial a_{01} = 0 \\ &\quad\vdots \\ &\quad \text{etc.} \end{aligned} \tag{4.5–44}$$

It should be noted that for the pure shear case, the Mooney–Rivlin constants cannot be uniquely determined from equation (4.5–42). In this case, the shear data must by supplemented by either or both of the other two types of test data to determine the constants.

Material Stability Check

Stability checks are provided for the Mooney–Rivlin hyperelastic materials. A nonlinear material is stable if the secondary work required for an arbitrary change in the deformation is always positive. Mathematically, this is equivalent to:

$$d\sigma_{ij}\ d\epsilon_{ij}\ >\ 0 \qquad (4.5–45)$$

where $d\sigma$ = change in the Cauchy stress tensor corresponding to a change in the logarithmic strain

Since the change in stress is related to the change in strain through the material stiffness tensor, checking for stability of a material can be more conveniently accomplished by checking for the positive definiteness of the material stiffness.

The material stability checks are done at two levels. The first stability check occurs at the end of preprocessing but before an analysis actually begins. At that time, the program checks for the loss of stability for six typical stress paths (uniaxial tension and compression, equibiaxial tension and compression, and planar tension and compression). The range of the stretch ratio over which the stability is checked is chosen from 0.1 to 10. If the material is stable over the range then no message will appear. Otherwise, a warning message appears that lists the Mooney–Rivlin constants and the critical values of the nominal strains where the material first becomes unstable. The second stability check is optional and occurs during the analysis. This check will be made for each element for each iteration. This optional stability check is available for all hyperelastic elements with mixed U–P formulation (HYPER56, HYPER58, HYPER74, HYPER158) by setting KEYOPT(8) = 1.

4.5.2 Arruda–Boyce Model

The Arruda–Boyce model is a physics–based hyperelastic model for modeling the large stretch behavior of rubber elastic materials. The model was first proposed by Arruda and Boyce(270) in 1993 for modeling incompressible materials and later extended by Bergstrom and Boyce(271) in 1998 to compressible materials. The material model is based on an eight–chain representation of the underlying macromolecular network structure of the rubber and non–Gaussian behavior of the individual chains in the proposed network. It has been shown that the model is superior when compared with the existing rubber elasticity models in the ability to capture the response of data in the literature(270).

TB,BOYCE accesses this material model, which is supported by SOLID185, SOLID186, and SOLID187 (8–node 3–D brick, 20–node 3–D brick, and 10–node 3–D tetrahedron elements, respectively).

For a hyperelastic material, the Cauchy stress tensor is a function of the left Cauchy–Green deformation tensor as:

$$\sigma_{ij}\ =\ \sigma_{ij}\ (B_{kl}) \qquad (4.5–46)$$

where: σ_{ij} = components of the Cauchy stress tensor

(B_{kl}) = components of the left Cauchy–Green deformation tensor

Components of the deformation tensor, B_{kl} is related to the components of the deformation gradient, f_{ij} by:

$$B_{kl} = f_{km}\, f_{lm} \tag{4.5–47}$$

The Cauchy stress tensor is decomposed into the pressure and deviatoric parts as:

$$\sigma_{ij} = -P\delta_{ij} + S_{ij} \tag{4.5–48}$$

where:
- P = pressure in the material (defined in equation (4.5–49))
- δ_{ij} = Kronecker delta
- S_{ij} = components of the stress deviator (defined in equation (4.5–50))

In the Arruda–Boyce material model, the pressure P and stress deviator S_{ij} are related to the deformation tensor through

$$P = -\frac{B}{2}\, \ell n\, (I_3) \tag{4.5–49}$$

$$S_{ij} = G_e\, B^d_{ij} \tag{4.5–50}$$

where:
- B = bulk modulus
- I_3 = det B_{ij}
- G_e = $C_R \dfrac{L^{-1}(r)}{r}$ = effective shear modulus
- C_R = initial modulus
- $L^{-1}(r)$ = inverse of the Langevin function defined in equation (4.5–51))
- r = $\dfrac{\sqrt{\frac{1}{3} I_1}}{\lambda_L}$
- I_1 = B_{ii}
- λ_L = limit stretch of the material
- B^d_{ij} = $B_{ij} - \frac{1}{3} I_1\, \delta_{ij}$

Parameter r in the above equation denotes the ratio of the averaged stretch in a material point and the limit stretch of the material. The Langevin function used above is defined by:

$$L(x) \equiv \coth(x) - \frac{1}{x} \tag{4.5–51}$$

The three constants for the material model, namely, the bulk modulus B, initial modulus C_R and the limit stretch λ_L are defined in Table 4.5–1. All constants must be positive.

Table 4.5–1 Material Parameter for Boyce Model

TBDATA constant	Parameter	Meaning	Units
1	C_R	initial modulus	stress, e.g. psi, mpa
2	λ_L	limit stretch	dimensionless
3	B	bulk modulus	stress, e.g. psi, mpa

The bulk modulus B of the material is determined by applying a hydrostatic stress state to the material and measuring the volume change. The initial modulus C_R and the limit stretch λ_L can be determined by a simple uniaxial compression test (see Appendix of Bergstrom and Boyce(271) for details). In the case where the bulk modulus is unknown and the material is known to be almost incompressible, B can be taken as a large number, for example, B = $500C_R$.

The output strain is the logarithmic strain

$$E_{ij} = \sum_{k=1}^{3} \ln \lambda_k \, n_i^k \, n_j^k \tag{4.5–52}$$

where:
E_{ij} = components of the logarithmic strain tensor
λ_k = principal stretches
n_i^k = components of the principal stretch directions

$\lambda_k \; n_i^k$ are extracted from the deformation tensor B_{ij} (defined in equation (4.5–47) by the following spectral decomposition:

$$B_{ij} = \sum_{k=1}^{3} \lambda_k^2 \, n_i^k \, n_j^k \tag{4.5–53}$$

4.6 Viscoelasticity

A material is said to be viscoelastic if the material has an elastic (recoverable) part as well as a viscous (non–recoverable) part. Upon application of a load, the elastic deformation is instantaneous while the viscous part occurs over time.

The viscoelastic model incorporated in the program usually depicts the deformation behavior of glass or glass–like materials and may simulate cooling and heating sequences of such material. These materials at high temperatures turn into viscous fluids and at low temperatures behave as solids. Further, the material is restricted to be thermorheologically simple, which assumes the material response to a load at a high temperature over a short duration is identical to that at a lower temperature but over a longer duration. The material model is available with the viscoelastic elements VISCO88 and VISCO89.

4.6.1 Basic Theory

The viscoelastic implementation uses a quasi–static boundary value approach to solving linear viscoelastic solids undergoing thermal and mechanical deformations. A given thermal history is presumed. The basic constitutive relationship is similar to that used in Markovsky et al.(108), Scherer and Rekhson(109), Narayanaswamy(110), and Zienkiewicz et al.(111) as the effects of time and temperatures are characterized by “fictive” temperature (also called reduced or pseudo time). The algorithmic details of the integration are identical to those proposed in Taylor et al.(112). All the necessary input data can be addressed by using **TBDATA** commands with **TB**,EVISC. See Section 2.5.3 of the *ANSYS Elements Reference* for more details.

The material properties are expressed in integral form using the kernel function of the generalized Maxwell elements as:

$$G(\xi) = \sum_{i=1}^{I} G_i e^{(-\xi/\lambda_i)} + G(\infty) \tag{4.6–1}$$

where:

$G(\xi)$ = current value of material property (output quantities EFF BULK MOD and EFF SHEAR MOD)

I = number of Maxwell elements used to approximate the material relaxation modulus (input constants 50 and 71). The input constant numbers refer to the Viscoelastic Material Data Table in section 2.5.3 of the *ANSYS Elements Reference*

G_i = $C_i\,(G(0) - G(\infty))$

C_i = constants associated with the instantaneous response (input constants 51 thru 60 and 76 thru 85)

G(0) = initial modulus (input constants 46 and 48)

G(∞) = final modulus (input constants 47 and 49

ξ = reduced or pseudo time (defined below)

λ_i = constants associated with a discrete relaxation spectrum (input constants 61 thru 70 and 86 thru 95)

The stress can be related to strain at any time by the convolution integral:

$$\sigma(t) = \int_{-0}^{t} G(\xi - \xi') \frac{d\epsilon(t')}{dt'} dt' \tag{4.6–2}$$

where:

σ = stress

ε = total strain (includes thermal strain)

t = time

ξ = pseudo time

The lower limit of the integral (–0) in equation (4.6–2) accounts for instantaneous loads at t=0.

Reduced (or pseudo) time defined here as:

$$\xi = \int_{0}^{t} \phi \, dt' \tag{4.6–3}$$

where:

ξ = reduced time

ϕ = shift factor = e^A

The exponent A on the shift factor is defined as:

$$A = \frac{H}{R}\left(\frac{1}{T_{ref}} - \frac{x}{T(t')} - \frac{(1 - x)}{T_f(t')}\right) \tag{4.6–4}$$

where:

$\frac{H}{R}$ = activation energy divided by the ideal gas constant (input constant 1)

T_{ref} = reference temperature (input on **TREF** command)

x = input constant 2

T(t′) = temperature at time t′

$T_f(t')$ = fictive temperature at time t′ (see below for more information)

Initial fictive temperature is defined using input constants 6 thru 15 and 36 thru 45. Subsequent fictive temperatures use input constants 16 thru 25. These usages are explained in reference (108, 110). The fictive temperature is given as output quantity FICT TEMP.

The incremental change in growth (volumetric) strain follows the relationship:

$$\Delta\epsilon^{gr} = \alpha(T)_g \, \Delta T + \left[\alpha(T_f)_\ell - \alpha(T)_g\right] \Delta T_f \tag{4.6–5}$$

where:

$\Delta\varepsilon^{gr}$ = incremental change in growth strain

$\alpha(T)_g$ = coefficient of thermal expansion for the glass state, which is a function of the actual temperature T (input constants 31 thru 35)

ΔT = change of actual temperature

$\alpha(T_f)_\ell$ = coefficient of thermal expansion for the liquid state, which is a function of the fictive temperature T_f (input constants 26 thru 30)

ΔT_f = change of fictive temperature

The incremental changes in growth strain are summed to give:

$$\epsilon^{gr} = \sum_{j=1}^{N_t} \left(\Delta\epsilon^{gr}\right) \tag{4.6–6}$$

where:

ε^{gr} = summed total strains (output quantity GR STRAIN)

N_t = total number of time steps up to and including the current time point

4.7 Concrete

The concrete material model predicts the failure of brittle materials. Both cracking and crushing failure modes are accounted for. **TB**,CONCR accesses this material model, which is available with the reinforced concrete element SOLID65.

The criterion for failure of concrete due to a multiaxial stress state can be expressed in the form

$$\frac{F}{f_c} - S \geq 0 \qquad (4.7\text{–}1)$$

where: F = a function (to be discussed) of the principal stress state (σ_{xp}, σ_{yp}, σ_{zp})

S = failure surface (to be discussed) expressed in terms of principal stresses and five input parameters f_t, f_c, f_{cb}, f_1 and f_2 defined in Table 4.7–1

f_c = uniaxial crushing strength

$\sigma_{xp}, \sigma_{yp}, \sigma_{zp}$ = principal stresses in principal directions

If equation (4.7–1) is not satisfied, there is no attendant cracking or crushing. Otherwise, the material will crack if any principal stress is tensile while crushing will occur if all principal stresses are compressive.

A total of five input strength parameters (each of which can be temperature dependent) are needed to define the failure surface as well as an ambient hydrostatic stress state. These are presented in Table 4.7–1.

Table 4.7–1 Concrete Material Table
(Input on TBDATA Commands with TB,CONCR)

Label	Description	Constant
f_t	Ultimate uniaxial tensile strength	3
f_c	Ultimate uniaxial compressive strength	4
f_{cb}	Ultimate biaxial compressive strength	5
σ_h^a	Ambient hydrostatic stress state	6

Label	Description	Constant
f_1	Ultimate compressive strength for a state of biaxial compression superimposed on hydrostatic stress state σ_h^a	7
f_2	Ultimate compressive strength for a state of uniaxial compression superimposed on hydrostatic stress state σ_h^a	8

However, the failure surface can be specified with a minimum of two constants, f_t and f_c. The other three constants default to (William and Warnke(37)):

$$f_{cb} = 1.2\ f_c \tag{4.7–2}$$

$$f_1 = 1.45\ f_c \tag{4.7–3}$$

$$f_2 = 1.725\ f_c \tag{4.7–4}$$

However, these default values are valid only for stress states where the condition

$$|\sigma_h| \quad \leq \sqrt{3}\ \ f_c \tag{4.7–5}$$

$$\left(\sigma_h = \text{hydrostatic stress state} = \frac{1}{3}\left(\sigma_{xp} + \sigma_{yp} + \sigma_{zp}\right)\right) \tag{4.7–6}$$

is satisfied. Thus condition (4.7–5) applies to stress situations with a low hydrostatic stress component. All five failure parameters should be specified when a large hydrostatic stress component is expected. If condition (4.7–5) is not satisfied and the default values shown in equations (4.7–2) thru (4.7–4) are assumed, the strength of the concrete material may be incorrectly evaluated.

Both the function F and the failure surface S are expressed in terms of principal stresses denoted as σ_1, σ_2, and σ_3 where:

$$\sigma_1 = \max\left(\sigma_{xp},\ \sigma_{yp},\ \sigma_{zp}\right) \tag{4.7–7}$$

$$\sigma_3 = \min\left(\sigma_{xp},\ \sigma_{yp},\ \sigma_{zp}\right) \tag{4.7–8}$$

and $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The failure of concrete is categorized into four domains:

1. $0 \geq \sigma_1 \geq \sigma_2 \geq \sigma_3$ (compression – compression – compression)
2. $\sigma_1 \geq 0 \geq \sigma_2 \geq \sigma_3$ (tensile – compression – compression)

3. $\sigma_1 \geq \sigma_2 \geq 0 \geq \sigma_3$ (tensile – tensile – compression)
4. $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ (tensile – tensile – tensile)

In each domain, independent functions describe F and the failure surface S. The four functions describing the general function F are denoted as F_1, F_2, F_3, and F_4 while the functions describing S are denoted as S_1, S_2, S_3, and S_4 . The functions S_i (i = 1,4) have the properties that the surface they describe is continuous while the surface gradients are not continuous when any one of the principal stresses changes sign. The surface will be shown in Figure 4.7–1 and Figure 4.7–3. These functions are discussed in detail below for each domain.

4.7.1 The Domain $0 \geq \sigma_1 \geq \sigma_2 \geq \sigma_3$

In the compression – compression – compression regime, the failure criterion of William and Warnke(37) is implemented. In this case, F takes the form

$$F = F_1 = \frac{1}{\sqrt{15}} \left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 \right]^{\frac{1}{2}} \tag{4.7–9}$$

and S is defined as

$$S = S_1 = \frac{2r_2(r_2^2 - r_1^2)\cos\eta + r_2(2r_1 - r_2)\left[4(r_2^2 - r_1^2)\cos^2\eta + 5r_1^2 - 4r_1r_2\right]^{\frac{1}{2}}}{4(r_2^2 - r_1^2)\cos^2\eta + (r_2 - 2r_1)^2} \tag{4.7–10}$$

Terms used to define S are:

$$\cos\eta = \frac{2\sigma_1 - \sigma_2 - \sigma_3}{\sqrt{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]^{\frac{1}{2}}} \tag{4.7–11}$$

$$r_1 = a_0 + a_1\xi + a_2\xi^2 \tag{4.7–12}$$

$$r_2 = b_0 + b_1\xi + b_2\xi^2 \tag{4.7–13}$$

$$\xi = \frac{\sigma_h}{f_c}$$

σ_h is defined by equation (4.7–6) and the undetermined coefficients a_0, a_1, a_2, b_0, b_1, and b_2 are discussed below.

This failure surface is shown as Figure 4.7–1. The angle of similarity η describes the relative magnitudes of the principal stresses. From equation (4.7–11), $\eta = 0°$ refers to any stress state such that $\sigma_3 = \sigma_2 > \sigma_1$ (e.g. uniaxial compression, biaxial tension) while $\xi = 60°$for any stress state where $\sigma_3 > \sigma_2 = \sigma_1$ (e.g. uniaxial tension, biaxial compression). All other multiaxial stress states have angles of similarity such that $0° \leq \eta \leq 60°$. When $\eta = 0°$, S_1 (equation (4.7–10)) equals r_1 while if $\eta = 60°$, S_1 equals r_2. Therefore, the function r_1 represents the failure surface of all stress states with $\eta = 0°$. The functions r_1, r_2 and the angle η are depicted on Figure 4.7–1.

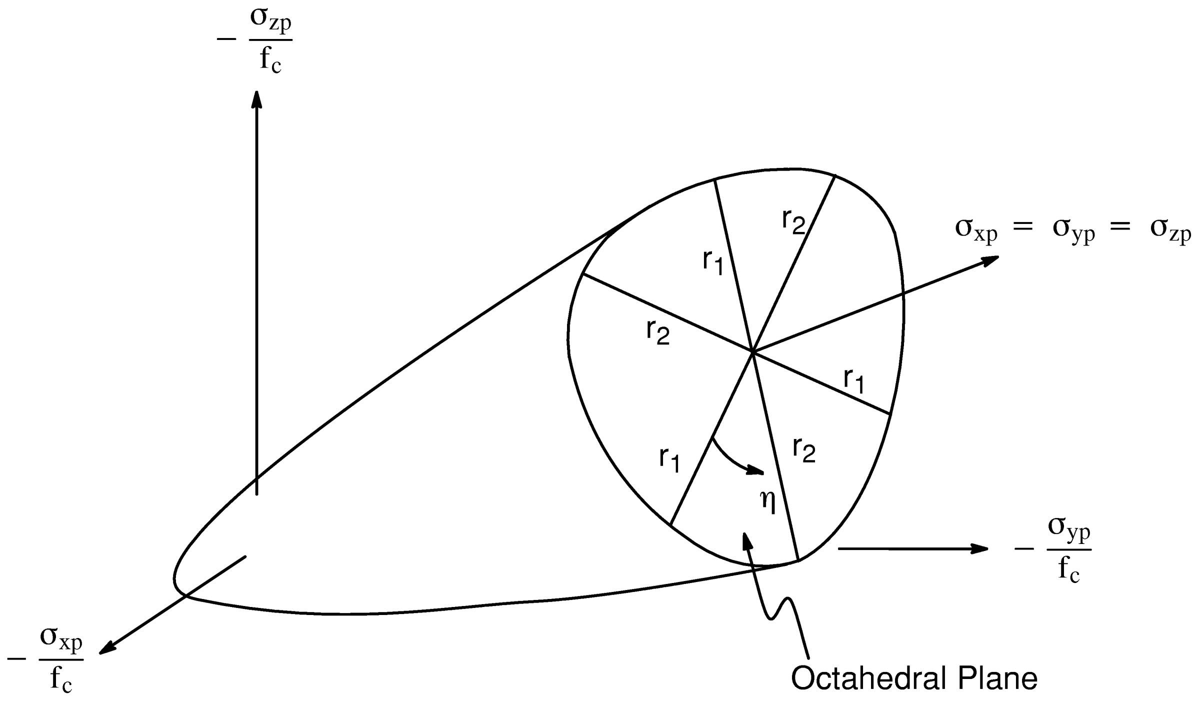

$-\frac{\sigma_{zp}}{f_c}$
$\sigma_{xp} = \sigma_{yp} = \sigma_{zp}$
r_1
r_2
r_2
r_1
r_1
r_2
η
$-\frac{\sigma_{yp}}{f_c}$
$-\frac{\sigma_{xp}}{f_c}$
Octahedral Plane

Figure 4.7–1 3–D Failure Surface in Principal Stress Space

It may be seen that the cross–section of the failure plane has cyclic symmetry about each 120° sector of the octahedral plane due to the range $0° \leq \eta \leq 60°$of the angle of similitude. The function r_1 is determined by adjusting a_0, a_1, and a_2 such that f_t, f_{cb}, and f_1 all lie on the failure surface. The proper values for these coefficients are determined through solution of the simultaneous equations:

$$\begin{Bmatrix} \frac{F_1}{f_c}\left(\sigma_1 = f_t,\ \sigma_2 = \sigma_3 = 0\right) \\ \frac{F_1}{f_c}\left(\sigma_1 = 0,\ \sigma_2 = \sigma_3 = -f_{cb}\right) \\ \frac{F_1}{f_c}\left(\sigma_1 = -\sigma_h^a,\ \sigma_2 = \sigma_3 = -\sigma_h^a - f_1\right) \end{Bmatrix} = \begin{bmatrix} 1 & \xi_t & \xi_t^2 \\ 1 & \xi_{cb} & \xi_{cb}^2 \\ 1 & \xi_1 & \xi_1^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \end{Bmatrix} \qquad (4.7\text{–}14)$$

with

$$\xi_t = \frac{f_t}{3f_c},\ \xi_{cb} = -\frac{2f_{cb}}{3f_c},\ \xi_1 = -\frac{\sigma_h^a}{f_c} - \frac{2f_1}{3f_c} \qquad (4.7\text{–}15)$$

The function r_2 is calculated by adjusting b_0, b_1, and b_2 to satisfy the conditions:

$$\begin{Bmatrix} \frac{F_1}{f_c}\left(\sigma_1 = \sigma_2 = 0,\ \sigma_3 = -f\right) \\ \frac{F_1}{f_c}\left(\sigma_1 = \sigma_2 = -\sigma_h^a,\ \sigma_3 = -\sigma_h^a - \frac{f_2}{f_c}\right) \\ 0 \end{Bmatrix} = \begin{bmatrix} 1 & -\frac{1}{3} & \frac{1}{9} \\ 1 & \xi_2 & \xi_2^2 \\ 1 & \xi_0 & \xi_0^2 \end{bmatrix} \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \end{Bmatrix} \qquad (4.7\text{–}16)$$

ξ_2 is defined by:

$$\xi_2 = -\frac{\sigma_h^a}{f_c} - \frac{f_2}{3f_c} \qquad (4.7\text{–}17)$$

and ξ_0 is the positive root of the equation

$$r_2(\xi_0) = a_0 + a_1\xi_0 + a_2\xi_0^2 = 0 \qquad (4.7\text{–}18)$$

where a_0, a_1, and a_2 are evaluated by equation (4.7–14).

Since the failure surface must remain convex, the ratio r_1 / r_2 is restricted to the range

$$.5 < r_1/r_2 < 1.25 \qquad (4.7\text{–}19)$$

although the upper bound is not considered to be restrictive since $r_1 / r_2 < 1$ for most materials (William(36)). Also, the coefficients a_0, a_1, a_2, b_0, b_1, and b_2 must satisfy the conditions (William and Warnke(37)):

$$a_0 > 0 \ , \ a_1 \leq 0 \ , \ a_2 \leq 0 \tag{4.7–20}$$

$$b_0 > 0 \ , \ b_1 \leq 0 \ , \ b_2 \leq 0 \tag{4.7–21}$$

Therefore, the failure surface is closed and predicts failure under high hydrostatic pressure ($\xi > \xi_2$). This closure of the failure surface has not been verified experimentally and it has been suggested that a von Mises type cylinder is a more valid failure surface for large compressive σ_h values (William(36)). Consequently, it is recommended that values of f_1 and f_2 are selected at a hydrostatic stress level $\left(\sigma_h^a\right)$ in the vicinity of or above the expected maximum hydrostatic stress encountered in the structure.

Equation (4.7–18) expresses the condition that the failure surface has an apex at $\xi = \xi_0$. A profile of r_1 and r_2 as a function of ξ is shown in Figure 4.7–2.

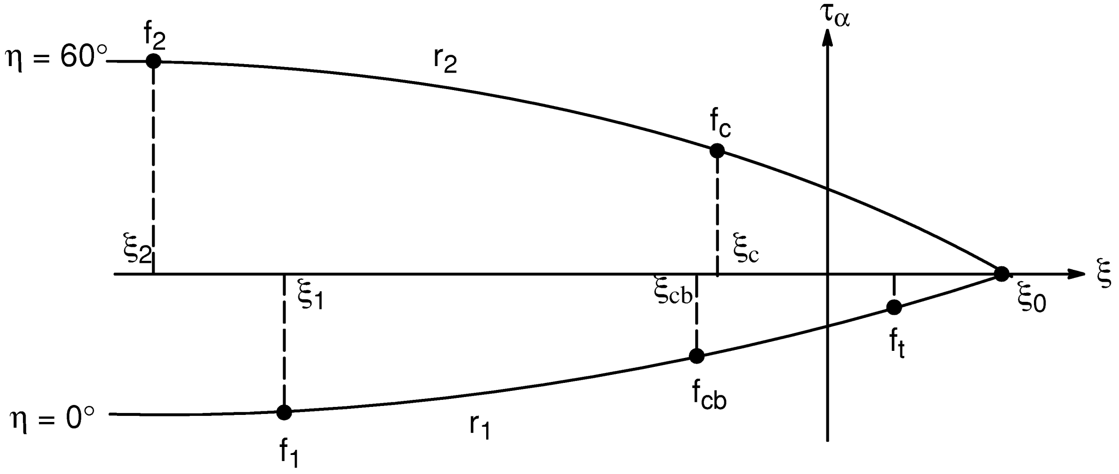


Figure 4.7–2 A Profile of the Failure Surface as a Function of ξ_α

The lower curve represents all stress states such that η = 0° while the upper curve represents stress states such that η = 60°. If the failure criterion is satisfied, the material is assumed to crush.

4.7.2 The Domain $\sigma_1 \geq 0 \geq \sigma_2 \geq \sigma_3$

In the tension – compression – compression regime, F takes the form

$$F = F_2 = \frac{1}{\sqrt{15}} \left[\left(\sigma_2 - \sigma_3\right)^2 + \sigma_2^2 + \sigma_3^2 \right]^{\frac{1}{2}} \tag{4.7–22}$$

and S is defined as

$$S = S_2 = \left(1 - \frac{\sigma_1}{f_t}\right)\frac{2p_2\left(p_2^2 - p_1^2\right)\cos\,\eta + p_2(2p_1 - p_2)\left[4\left(p_2^2 - p_1^2\right)\cos^2\eta + 5p_1^2 - 4p_1p_2\right]^{\frac{1}{2}}}{4\left(p_2^2 - p_1^2\right)\cos^2\eta + \left(p_2 - 2p_1\right)^2} \tag{4.7–23}$$

where cos η is defined by equation (4.7–11) and

$$p_1 = a_0 + a_1\chi + a_2\chi^2 \tag{4.7–24}$$

$$p_2 = b_0 + b_1\chi + b_2\chi^2 \tag{4.7–25}$$

The coefficients a_0, a_1, a_2, b_0, b_1, b_2 are defined by equations (4.7–14) and (4.7–16) while

$$\chi = \frac{1}{3}\left(\sigma_2 + \sigma_3\right) \tag{4.7–26}$$

If the failure criterion is satisfied, cracking occurs in the plane perpendicular to principal stress σ_1.

4.7.3 The Domain $\sigma_1 \geq \sigma_2 \geq 0 \geq \sigma_3$

In the tension – tension – compression regime, F takes the form

$$F = F_3 = \sigma_i \quad ; \quad i = 1,\ 2 \tag{4.7–27}$$

and S is defined as

$$S = S_3 = \frac{f_t}{f_c}\left(1 + \frac{\sigma_3}{S_2\left(\sigma_i\ , 0, \sigma_3\right)}\right) \quad ; \quad i = 1,\ 2 \tag{4.7–28}$$

If the failure criterion for both i = 1, 2 is satisfied, cracking occurs in the planes perpendicular to principal stresses σ_1 and σ_2. If the failure criterion is satisfied only for i = 1, cracking occurs only in the plane perpendicular to principal stress σ_1.

4.7.4 The Domain $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$

In the tension – tension – tension regimes, F takes the form

$$F = F_4 = \sigma_i \quad ; \quad i = 1,\ 2,\ 3 \tag{4.7–29}$$

and S is defined as

$$S = S_4 = \frac{f_t}{f_c} \tag{4.7–30}$$

If the failure criterion is satisfied in directions 1, 2, and 3, cracking occurs in the planes perpendicular to principal stresses σ_1, σ_2, and σ_3.

If the failure criterion is satisfied in directions 1 and 2, cracking occurs in the plane perpendicular to principal stresses σ_1 and σ_2.

If the failure criterion is satisfied only in direction 1, cracking occurs in the plane perpendicular to principal stress σ_1.

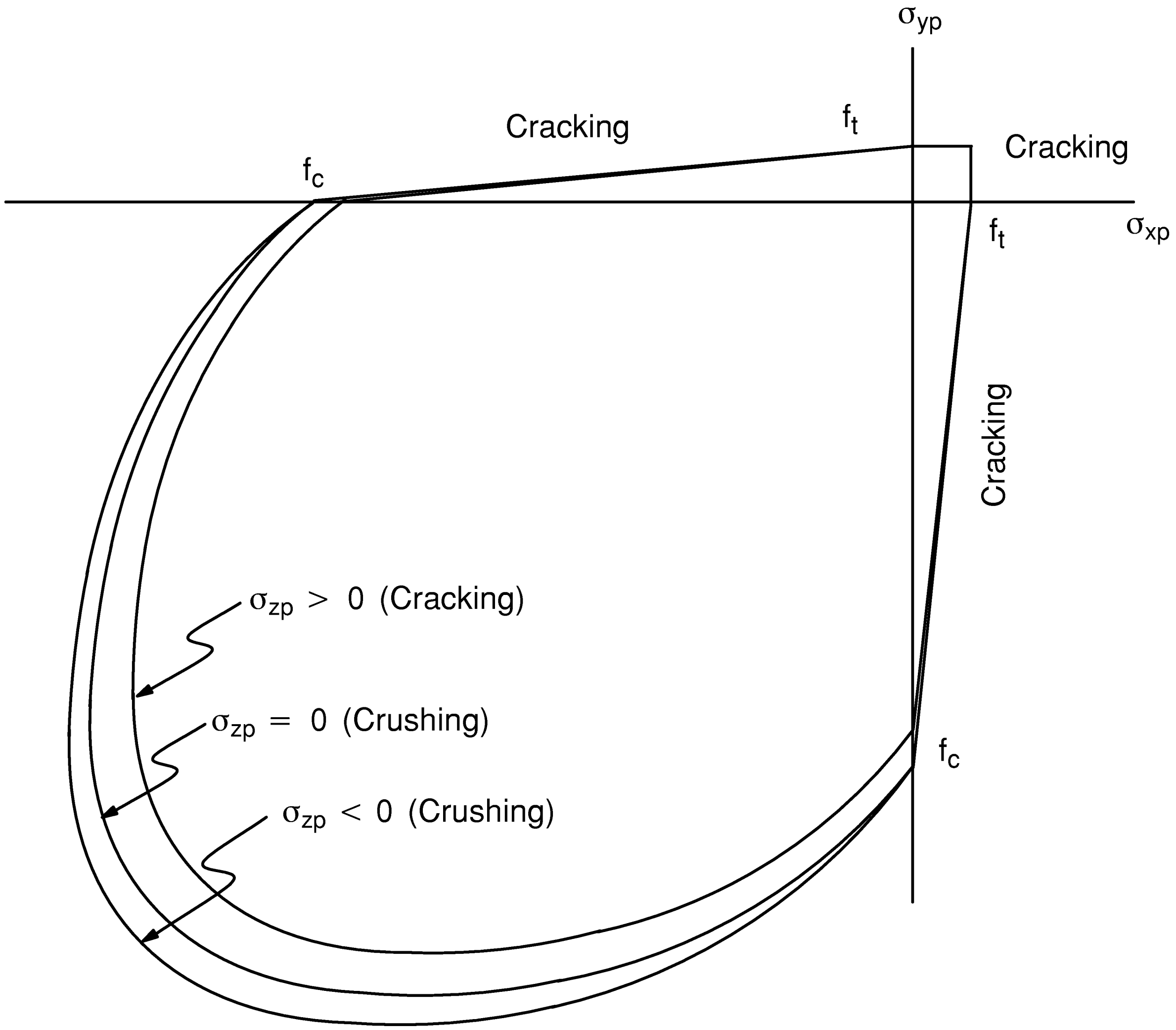

σ_{yp}
Cracking
f_t
Cracking
f_c
f_t
σ_{xp}
Cracking
$\sigma_{zp} > 0$ (Cracking)
$\sigma_{zp} = 0$ (Crushing)
f_c
$\sigma_{zp} < 0$ (Crushing)

Figure 4.7–3 Failure Surface in Principal Stress Space σ_{zp} Close to Zero

Figure 4.7–3 represents the 3–D failure surface for states of stress that are biaxial or nearly biaxial. If the most significant nonzero principal stresses are in the σ_{xp} and σ_{yp} directions, the three surfaces presented are for σ_{zp} slightly greater than zero, σ_{zp} equal to zero, and σ_{zp} slightly less than zero. Although the three surfaces, shown as projections on the $\sigma_{xp} - \sigma_{yp}$ plane, are nearly equivalent and the 3–D failure surface is continuous, the mode of material failure is a function of the sign of σ_{zp}. For example, if σ_{xp} and σ_{yp} are both negative and σ_{zp} is slightly positive, cracking would be predicted in a direction perpendicular to the σ_{zp} direction. However, if σ_{zp} is zero or slightly negative, the material is assumed to crush.

4.8 Swelling

The ANSYS program provides a capability of irradiation induced swelling. Swelling is defined as a material enlarging volumetrically in the presence of neutron flux. The amount of swelling may also be a function of temperature. The material is assumed to be isotropic and the basic solution technique used is the initial stress method. Swelling calculations are available only through the user swelling subroutine. See Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide* and the *Programmer's* manual for more details. Input is with the **TBDATA** commands, with **TB**,SWELL, in which C_{72} must be set to 10. Constants C_{67} through C_{71} are used together with fluence and temperature, as well as possibly strain, stress and time, to develop an expression for swelling rate.

Any of the following three conditions cause the swelling calculations to be bypassed:

1. If $C_{67} \leq 0.$ and $C_{68} \leq 0.$
2. If (input temperature + T_{off}) ≤ 0, where T_{off} = input on **TOFFST** command.
3. If $\text{Fluence}_n \leq \text{Fluence}_{n-1}$ (n refers to current time step).

The total swelling strain is computed in subroutine USERSW as:

$$\epsilon_n^{sw} = \epsilon_{n-1}^{sw} + \Delta\epsilon^{sw} \tag{4.8–1}$$

where:
ϵ_n^{sw} = swelling strain at end of substep n
$\Delta\epsilon^{sw}$ = $r\Delta f$ = swelling strain increment
r = swelling rate
Δf = $f_n - f_{n-1}$ = change of fluence
f_n = fluence at end of substep n (input quantity VAL1, etc. on the **BFE**,,FLUE command)

For a solid element, the swelling strain vector is simply:

$$\{\epsilon^{sw}\} = \lfloor \epsilon_n^{sw} \quad \epsilon_n^{sw} \quad \epsilon_n^{sw} \quad 0 \quad 0 \quad 0 \rfloor^T \tag{4.8–2}$$

It is seen that the swelling strains are handled in a manner totally analogous to temperature strains in an isotropic medium and that shearing strains are not used.

Chapter 5
Electromagnetics

ANSYS Theory Reference
Chapter 5 – Table of Contents

5.1 Electromagnetic Field Fundamentals

Electromagnetic fields are governed by the following Maxwell's equations:

$$\nabla \times \{H\} = \{J\} + \left\{\frac{\partial D}{\partial t}\right\} = \{J_s\} + \{J_e\} + \{J_v\} + \left\{\frac{\partial D}{\partial t}\right\} \tag{5.1–1}$$

$$\nabla \times \{E\} = -\left\{\frac{\partial B}{\partial t}\right\} \tag{5.1–2}$$

$$\nabla \cdot \{B\} = 0 \tag{5.1–3}$$

$$\nabla \cdot \{D\} = \rho \tag{5.1–4}$$

where:

$\nabla \times$ = curl operator

$\nabla \cdot$ = divergence operator

$\{H\}$ = magnetic field intensity vector

$\{J\}$ = total current density vector

$\{J_s\}$ = applied source current density vector

$\{J_e\}$ = induced eddy current density vector

$\{J_v\}$ = velocity current density vector

$\{D\}$ = electric flux density vector (Maxwell referred to this as the displacement vector, but to avoid misunderstanding with mechanical displacement, the name electric flux density is used here.)

t = time

$\{E\}$ = electric field intensity vector

$\{B\}$ = magnetic flux density vector

ρ = electric charge density

The continuity equation, $\nabla \cdot \left[\{J\} + \left\{\frac{\partial D}{\partial t}\right\}\right] = 0$, follows from (5.1–1) after taking the divergence of both sides. The continuity equation must be satisfied for the proper setting of Maxwell's equations. Users should prescribe J_S taking this into account.

The above field equations are supplemented by the constitutive relation that describes the behavior of electromagnetic materials. For problems considering saturable material without permanent magnets, the constitutive relation for the magnetic fields is:

$$\{B\} = [\mu]\{H\} \tag{5.1–5}$$

where: $[\mu]$ = magnetic permeability matrix, in general a function of {H}

The magnetic permeability matrix [μ] may be input either as a function of temperature or field. Specifically, if [μ] is only a function of temperature,

$$[\mu] = \mu_o \begin{bmatrix} \mu_{rx} & 0 & 0 \\ 0 & \mu_{ry} & 0 \\ 0 & 0 & \mu_{rz} \end{bmatrix} \tag{5.1–6}$$

where: μ_o = permeability of free space (input on **EMUNIT** command)

μ_{rx} = relative permeability in the x–direction (input as MURX on **MP** command)

If [μ] is only a function of field,

$$[\mu] = \mu_h \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.1–7}$$

where: μ_h = permeability derived from the input B versus H curve on the **TBPT** commands with **TB**, BH.

Mixed usage is also permitted, e.g.:

$$[\mu] = \begin{bmatrix} \mu_h & 0 & 0 \\ 0 & \mu_o\mu_{ry} & 0 \\ 0 & 0 & \mu_h \end{bmatrix} \tag{5.1–8}$$

When permanent magnets are considered, the constitutive relation becomes:

$$\{B\} = [\mu]\ \{H\} + \mu_o\ \{M_o\} \tag{5.1–9}$$

where: $\{M_o\}$ = remanent intrinsic magnetization vector

Rewriting the general constitutive equation in terms of reluctivity it becomes:

$$\{H\} = [\nu]\{B\} - \frac{1}{\nu_o}\ [\nu]\ \{M_o\} \tag{5.1–10}$$

where: $[\nu]$ = reluctivity matrix = $[\mu]^{-1}$

ν_o = reluctivity of free space = $\frac{1}{\mu_o}$

The constitutive relations for the related electric fields are:

$$\{J\} = [\sigma]\ [\{E\} + \{v\} \times \{B\}] \tag{5.1–11}$$

$$\{D\} = [\epsilon]\ \{E\} \tag{5.1–12}$$

where: $[\sigma]$ = electrical conductivity matrix

$$= \begin{bmatrix} \sigma_{xx} & 0 & 0 \\ 0 & \sigma_{yy} & 0 \\ 0 & 0 & \sigma_{zz} \end{bmatrix}$$

$[\epsilon]$ = permittivity matrix

$$= \begin{bmatrix} \epsilon_{xx} & 0 & 0 \\ 0 & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz} \end{bmatrix}$$

$\{v\}$ = velocity vector

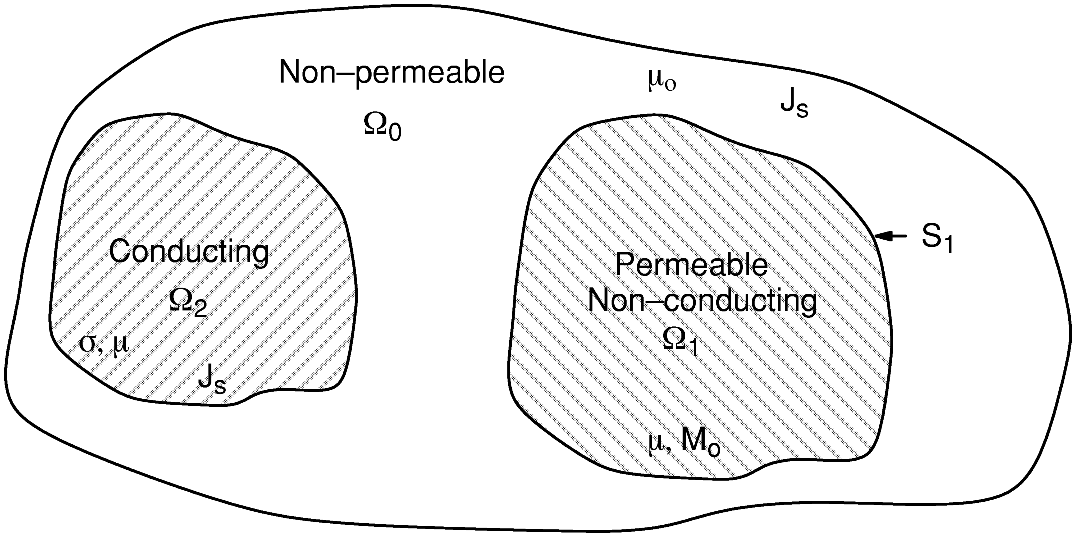
$$= \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix}$$

σ_{xx} = conductivity in the x–direction (input as inverse of RSVX on **MP** command)

ε_{xx} = permittivity in the x–direction (input as PERX on **MP** command)

The solution of magnetic field problems is commonly obtained using potential functions. Two kinds of potential functions, the magnetic vector potential and the magnetic scalar potential are used depending on the problem to be solved. Factors affecting the choice of potential include: field dynamics, field dimensionality, source current configuration, domain size and discretization.

The applicable regions are shown in Figure 5.1–1. These will be referred to with each solution procedure discussed below.



where:			
	Ω_0	=	free space region
	Ω_1	=	non–conducting permeable region
	Ω_2	=	conducting region
	μ	=	permeability of iron
	μ_o	=	permeability of air
	M_o	=	permanent magnets
	S_1	=	boundary of Ω_1
	σ	=	conductivity
	Ω	=	$\Omega_1 + \Omega_2 + \Omega_0$

Figure 5.1–1 Electromagnetic Field Regions

5.1.1 Magnetic Scalar Potential

The scalar potential method as implemented in SOLID5, SOLID96 and SOLID98 for 3–D magnetostatic fields is discussed in this section. Magnetostatics means that time varying effects are ignored. This reduces Maxwell's equations for magnetic fields to:

$$\nabla \text{ x } \{H\} = \{J_s\} \tag{5.1–13}$$

$$\nabla \cdot \{B\} = 0 \tag{5.1–14}$$

Solution Strategies

In the domain Ω_0 and Ω_1 of a magnetostatic field problem (Ω_2 is not considered for magnetostatics) a solution is sought which satisfies the relevant Maxwell's equations

(5.1–13) and (5.1–14) and the constitutive relation (5.1–9) in the following form (Gyimesi(141) and Gyimesi(149):

$$\{H\} = \{H_g\} - \nabla \phi_g \tag{5.1–15}$$

$$\nabla \cdot [\mu] \nabla \phi_g - \nabla \cdot [\mu] \{H_g\} - \nabla \cdot \mu_o \{M_o\} = \{0\} \tag{5.1–16}$$

where: $\{H_g\}$ = preliminary or “guess” magnetic field
ϕ_g = generalized potential

The development of $\{H_g\}$ varies depending on the problem and the formulation. Basically, $\{H_g\}$ must satisfy Ampere’s law (equation (5.1–13)) so that the remaining part of the field can be derived as the gradient of the generalized scalar potential ϕ_g. This ensures that ϕ_g is singly valued. Additionally, the absolute value of $\{H_g\}$ must be greater than that of $\nabla \phi_g$. In other words, $\{H_g\}$ should be a good approximation of the total field. This avoids difficulties with cancellation errors (Gyimesi(149)).

This framework allows for a variety of scalar potential formulation to be used. The appropriate formulation depends on the characteristics of the problem to be solved. The process of obtaining a final solution may involve several steps controlled by the **MAGOPT** solution option.

As mentioned above, the selection of $\{H_g\}$ is essential to the development of any of the following scalar potential strategies. The development of $\{H_g\}$ always involves the Biot–Savart field $\{H_s\}$ which satisfies Ampere’s law and is a function of source current $\{J_s\}$. $\{H_s\}$ is obtained by evaluating the integral:

$$\{H_s\} = \frac{1}{4\pi} \int_{volc} \frac{\{J_s\} \text{ x } \{r\}}{|\{r\}|^3} \, d(volc) \tag{5.1–17}$$

where: $\{J_s\}$ = current source density vector at d(volc)
$\{r\}$ = position vector from current source to node point
volc = volume of current source

The above volume integral can be reduced to the following surface integral (Gyimesi et al.(173))

$$\{H_s\} = \frac{1}{4\pi} \int_{surfc} \frac{\{J_s\}}{|\{r\}|} \times d(surfc) \tag{5.1–18}$$

where: surfc = surface of the current source

Evaluation of this integral is automatically performed upon initial solution execution or explicitly controlled by the **BIOT** command. The values of $\{J_s\}$ are obtained either directly as input by:

SOURC36 – Current Source

or indirectly calculated by electric field calculation using:

SOLID5 – 3–D Coupled–Field Solid

LINK68 – Coupled Thermal–Electric Line

SOLID69 – Coupled Thermal–Electric Solid

SOLID98 – Tetrahedral Coupled–Field Solid

Depending upon the current configuration, the integral given in equation (5.1–18) is evaluated in a closed form and/or a numerical fashion (Smythe(150)).

Three different solution strategies emerge from the general framework discussed above:

A. Reduced Scalar Potential (RSP) Strategy

B. Difference Scalar Potential (DSP) Strategy

C. General Scalar Potential (GSP) Strategy

A. RSP Strategy

Applicability

If there are no current sources ($\{J_s\}$ = 0) the RSP strategy is applicable. Also, in general, if there are current sources and there is no iron ($[\mu] = [\mu_o]$) within the problem domain, the RSP strategy is also applicable. This formulation is developed by Zienkiewicz(75).

Procedure

The RSP strategy uses a one–step procedure (**MAGOPT**,0). Equations (5.1–15) and (5.1–16) are solved making the following substitution:

$$\{H_g\} = \{H_s\} \qquad \text{in } \Omega_o \text{ and } \Omega_1 \tag{5.1–19}$$

Saturation is considered if the magnetic material is nonlinear. Permanent magnets are also considered.

B. DSP Strategy

Applicability

The DSP strategy is applicable when current sources and singly connected iron regions exist within the problem domain ($\{J_s\} \neq \{0\}$) and ($[\mu] \neq [\mu_o]$). A singly connected iron region does not enclose a current. In other words a contour integral of {H} through the iron must approach zero as u → ∞ :

$$\oint \{H\} \cdot \{d\ell\} \rightarrow \{0\} \qquad \text{in } \Omega_1 \text{ as } u \rightarrow \infty \qquad (5.1–20)$$

This formulation is developed by Mayergoyz(119).

Procedure

The DSP strategy uses a two–step solution procedure. The first step (**MAGOPT**,2) makes the following substitution into equations (5.1–15) and (5.1–16):

$$\{H_g\} = \{H_s\} \qquad \text{in } \Omega_o \text{ and } \Omega_1 \qquad (5.1–21)$$

subject to:

$$\{n\} \text{ x } \{H_g\} = \{0\} \qquad \text{on } S_1 \qquad (5.1–22)$$

This boundary condition is satisfied by using a very large value of permeability in the iron (internally set by the program). Saturation and permanent magnets are not considered. This step produces a near zero field in the iron region which is subsequently taken to be zero according to:

$$\{H_1\} = \{0\} \qquad \text{in } \Omega_1 \qquad (5.1–23)$$

and in the air region:

$$\{H_o\} = \{H_s\} - \nabla\phi_g \qquad \text{in } \Omega_o \qquad (5.1–24)$$

The second step (**MAGOPT**,3) uses the fields calculated on the first step as the preliminary field for equations (5.1–15) and (5.1–16):

$$\{H_g\} = \{0\} \qquad \text{in } \Omega_1 \qquad (5.1–25)$$

$$\{H_g\} = \{H_o\} \qquad \text{in } \Omega_o \qquad (5.1–26)$$

Here saturation and permanent magnets are considered. This step produces the following fields:

$$\{H_1\} = -\nabla\phi_g \qquad \text{in } \Omega_1 \qquad (5.1–27)$$

and

$$\{H_o\} = \{H_g\} - \nabla\phi_g \qquad \text{in } \Omega_o \tag{5.1–28}$$

which are the final results to the applicable problems.

C. GSP Strategy

Applicability

The GSP strategy is applicable when current sources ($\{J_s \neq \{0\}\}$) in conjunction with a multiply connected iron ($[\mu] \neq [\mu_o]$) region exist within the problem domain. A multiply connected iron region encloses some current source. This means that a contour integral of {H} through the iron region is not zero:

$$\oint \{H\} \cdot \{d\ell\} \neq \{0\} \qquad \text{in } \Omega_1 \tag{5.1–29}$$

This formulation is developed by Gyimesi(141,149, 201).

Procedure

The GSP strategy uses a three–step solution procedure. The first step (**MAGOPT**,1) performs a solution only in the iron with the following substitution into equations (5.1–15) and (5.1–16):

$$\{H_g\} = \{H_s\} \qquad \text{in } \Omega_1 \tag{5.1–30}$$

subject to:

$$\{n\} \cdot [\mu] \left(\{H_g\} - \nabla\phi_g\right) = 0 \qquad \text{on } S_1 \tag{5.1–31}$$

Here S_1 is the surface of the iron air interface. Saturation can optimally be considered for an improved approximation of the generalized field but permanent magnets are not. The resulting field is:

$$\{H_1\} = \{H_s\} - \nabla\phi_g \tag{5.1–32}$$

The second step (**MAGOPT**,2) performs a solution only in the air with the following substitution into equations (5.1–15) and (5.1–16):

$$\{H_g\} = \{H_s\} \qquad \text{in } \Omega_o \tag{5.1–33}$$

subject to:

$$\{n\} \text{ x } \{H_g\} = \{n\} \text{ x } \{H_1\} \qquad \text{in } S_1 \tag{5.1–34}$$

This boundary condition is satisfied by automatically constraining the potential solution ϕ_g at the surface of the iron to be what it was on the first step (**MAGOPT**,1). This step produces the following field:

$$\{H_o\} = \{H_s\} - \nabla\phi_g \qquad \text{in } \Omega_o \tag{5.1–35}$$

Saturation or permanent magnets are of no consequence since this step obtains a solution only in air.

The third step (**MAGOPT**,3) uses the fields calculated on the first two steps as the preliminary field for equations (5.1–15) and (5.1–16):

$$\{H_g\} = \{H_1\} \qquad \text{in } \Omega_1 \tag{5.1–36}$$

$$\{H_g\} = \{H_o\} \qquad \text{in } \Omega_o \tag{5.1–37}$$

Here saturation and permanent magnets are considered. The final step allows for the total field to be computed throughout the domain as:

$$\{H\} = \{H_g\} - \nabla\phi_g \qquad \text{in } \Omega \tag{5.1–38}$$

5.1.2 Magnetic Vector Potential

The vector potential method as implemented in PLANE13, PLANE53 and SOLID97 for both 2–D and 3–D electromagnetic fields is discussed in this section. Considering static and dynamic fields and neglecting displacement currents (quasi–stationary limit), the following subset of Maxwell's equations apply:

$$\nabla \text{ x } \{H\} = \{J\} \tag{5.1–39}$$

$$\nabla \text{ x } \{E\} = -\frac{\partial B}{\partial t} \tag{5.1–40}$$

$$\nabla \cdot \{B\} = 0 \tag{5.1–41}$$

The usual constitutive equation for magnetic and electric field apply as described by equations (5.1–10) and (5.1–11). Although some restriction on anistropy and nonlinearity do occur in the formulations mentioned below.

In the entire domain, Ω, of an electromagnetic field problem a solution is sought which satisfies the relevant Maxwell's equations (5.1–39) thru (5.1–40). See Figure 5.1–1 for a representation of the problem domain Ω.

A solution can be obtained by introducing potentials which allow the magnetic field {B} and the electric field {E} to be expressed as (Biro(120)):

$$\{B\} = \nabla \times \{A\} \tag{5.1–42}$$

$$\{E\} = - \left\{\frac{\partial A}{\partial t}\right\} - \nabla V \tag{5.1–43}$$

where:
{A} = magnetic vector potential
V = electric scalar potential

These specifications ensure the satisfaction of two of Maxwell's equations, (5.1–40) and (5.1–41). What remains to be solved is Ampere's law, equation (5.1–39) in conjunction with the constitutive relations, equation (5.1–10), and the divergence free property of current density. Additionally, to ensure uniqueness of the vector potential, the Coulomb gauge condition is employed. The resulting differential equations are:

$$\begin{aligned} \nabla \times [\nu]\, \nabla \times \{A\} - \nabla\, \nu_e\, \nabla \cdot \{A\} + [\sigma] \left\{\frac{\partial A}{\partial t}\right\} + [\sigma]\, \nabla V \\ - \{v\} \times [\sigma]\, \nabla \times \{A\} = \{0\} \qquad \text{in } \Omega_2 \end{aligned} \tag{5.1–44}$$

$$\nabla \cdot \left(- [\sigma] \left\{\frac{\partial A}{\partial t}\right\} - [\sigma]\, \nabla V + \{v\} \times [\sigma]\, \nabla \times \{A\} \right) = \{0\} \qquad \text{in } \Omega_2 \tag{5.1–45}$$

$$\nabla \times [\nu]\, \nabla \times \{A\} - \nabla\, \nu_e\, \nabla \cdot \{A\} = \{J_s\} + \nabla \times \frac{1}{\nu_o} [\nu]\, \{M_o\} \qquad \text{in } \Omega_o + \Omega_1 \tag{5.1–46}$$

where: $\nu_e = \frac{1}{3}\,\text{tr}\,[\nu] = \frac{1}{3}\,(\nu(1,1) + \nu(2,2) + \nu(3,3))$

Of course these equations are subject to the appropriate boundary conditions.

This system of simplified Maxwell's equations with the introduction of potential functions has been used for the solutions of 2–D and 3–D, static and dynamic fields. Silvester(72) presents a 2–D static formulation and Demerdash(151) develops the 3–D static formulation. Chari(69), Brauer(70) and Tandon(71) discuss the 2–D eddy current problem and Weiss(94) and Garg(95) discuss 2–D eddy current problems which allow for skin effects (eddy currents present in the source conductor). The development of 3–D eddy current problems is found in Biro(120). In many of these references the important issues of appropriate boundary conditions, gauging and uniqueness are discussed. The edge–flux formulation with tree gauging (Gyimesi and Ostergaard(202),(221), Ostergaard and Gyimesi(222),(223)) is discussed in section 14.117.2.

For models containing materials with different permeabilities, the 3–D vector potential formulation is not recommended. The solution has been found to be incorrect when the normal component of the vector potential is significant at the interface between elements of different permeability. A further discussion on this limitation is found in Biro et al(200).

5.1.3 Electric Scalar Potential

The electric scalar potential method as implemented in PLANE121, SOLID122, and SOLID123 for both 2–D and 3–D electrostatic fields is discussed in this section. Electrostatic fields satisfy the following subset of Maxwell's equations:

$$\nabla \times \{E\} = 0 \tag{5.1–47}$$

$$\nabla \cdot \{D\} = \rho \tag{5.1–48}$$

where: ϱ = free charge density

Repeating equation (5.1–12), the constitutive relation for the electric fields is:

$$\{D\} = [\epsilon]\ \{E\} \tag{5.1–49}$$

where: $[\varepsilon]$ = permittivity matrix

$$[\epsilon] = \begin{bmatrix} \epsilon_{xx} & 0 & 0 \\ 0 & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz} \end{bmatrix} \tag{5.1–50}$$

The relevant conditions for {E} and {D} on a dielectric interface are:

$$E_{t1} - E_{t2} = 0 \tag{5.1–51}$$

$$D_{n1} - D_{n2} = \rho_s \tag{5.1–52}$$

where: E_{t1}, E_{t2} = tangential components of {E} on both sides of the interface

D_{n1}, D_{n2} = normal components of {D} on both sides of the interface

ϱ_s = surface charge density

In the entire domain, Ω, of an electrostatic field problem, a solution is sought which satisfies the relevant Maxwell's equations (5.1–47) and (5.1–48), the constitutive relation (5.1–49), and the constraints (5.1–51) and (5.1–52).

A solution can be obtained by introducing the electric scalar potential, V, which allows the electric field to be expressed as:

$$\{E\} = -\nabla V \tag{5.1–53}$$

This specification ensures that the static electric field is irrotational. What remains to be solved is the divergence equation (5.1–48) and the constitutive relation (5.1–49). The corresponding differential equation is:

$$- \nabla \cdot [\epsilon] \nabla V = \rho \tag{5.1–54}$$

subject to the interface conditions, (5.1–51) and (5.1–52), as well as the appropriate boundary conditions.

5.1.4 Edge Flux Degrees of Freedom

Biro et al(200) and Preis et al(203) observed inaccuracies in the finite element analysis of 3–D magnetic field problems with the nodal based continuous vector potential, A, in the presence of inhomogeneous medium. This theoretical shortcomings of the nodal vector potential, A, has been demonstrated by Gyimesi and Ostergaard(201),(221), Ostergaard and Gyimesi(222), (223).

The shortcomings of the nodal based continuous vector potential formulation is demonstrated below. These shortcomings can be eliminated by the edge element method. The edge element formulation constitutes the theoretical foundation of low–frequency electromagnetic element, SOLID117. Section 12.9 describes the pertinent edge shape functions. Section 14.117.2 discusses topics related to the matrix formulation and gauging. Section 5.6 presents the details of the high frequency edge formulation.

To eliminate these inaccuracies, edge elements with a discontinuous normal component have been proposed. Sections 14.117 and 12.9 are devoted to discuss this topic.

Limitation of the Nodal Vector Potential

Consider a volume bounded by planes, x = ± –1, y = ± 1, and z = ± 1. See Figure 5.1–2. Subdivide the volume into four elements by planes, x = 0 and y = 0. The element numbers are set according to the space quadrant they occupy. The permeability, μ, of the elements is μ_1, μ_2, μ_3, and μ_4, respectively. Denote unit vectors by 1_x, 1_y, and 1_z. Consider a patch test with a known field, $H_k = 1_z$, $B_k = \mu H_k$ changes in the volume according to μ.

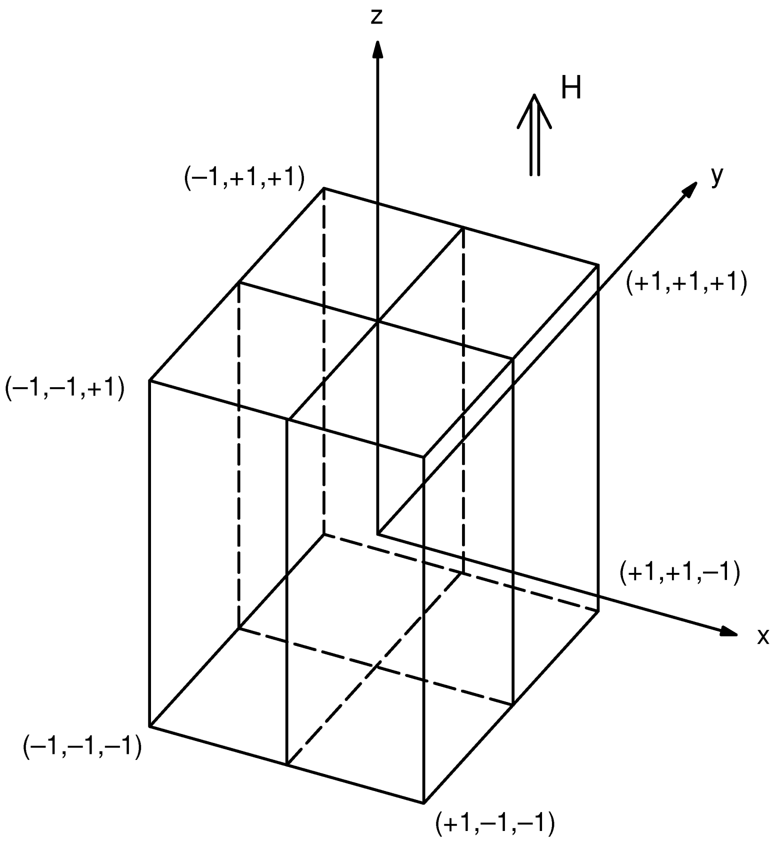


Figure 5.1–2 Patch Test Geometry

Since B_k is constant within the elements, one would expect that even a first order element could pass the patch test. This is really the case with edge element but not with nodal elements. For example, $A = \mu x 1_y$ provides a perfect edge solution but not a nodal one because the normal component of A in not continuous.

The underlying reason is that the partials of a continuous A do not exist; not even in a piece–wise manner. To prove this statement, assume that they exist. Denote the partials at the origin by:

$$\begin{aligned} A_x^+ &= \frac{\partial}{\partial y} A_x \text{ for } y > 0; \quad A_x^- = \frac{\partial}{\partial y} A_x \text{ for } y < 0; \\ A_y^+ &= \frac{\partial}{\partial x} A_y \text{ for } x > 0; \quad A_y^- = \frac{\partial}{\partial x} A_x \text{ for } x < 0; \end{aligned} \tag{5.1–55}$$

Note that there are only four independent partials because of A continuity. The following equations follow from B_k = *curl* A.

$$\begin{aligned} A_y^+ - A_x^+ &= \mu_1; \quad A_y^- - A_x^+ = \mu_2 \\ A_y^- - A_x^- &= \mu_3; \quad A_y^+ - A_x^- = \mu_4 \end{aligned} \tag{5.1–56}$$

Since the equation system, (5.1–56) is singular, a solution does not exist for arbitrary μ. This contradiction concludes the proof.

5.1.5 Harmonic Analysis Using Complex Formalism

In a general dynamic problem, any field quantity, q(r,t) depends on the space, r, and time, t, variables. In a harmonic analysis, the time dependence can be described by periodic functions:

$$q(r,t) = a(r) \cos (\omega t + \phi(r)) \tag{5.1–57}$$

or

$$q(r,t) = c(r) \cos (\omega t) - s(r) \sin (\omega t) \tag{5.1–58}$$

where:
r = location vector in space
t = time
ω = angular frequency of time change.
a(r) = amplitude (peak)
ϕ(r) = phase angle
c(r) = measureable field at $\omega \underline{\underline{t}}$ = 0 degrees
s(r) = measureable field at ωt = –90 degrees

In an electromagnetic analysis, q(r,t) can be the flux density, B, the magnetic field, H, the electric field, E, the current density, J, the vector potential, A, or the scalar potential, V. Note, however, that q(r,t) can not be the Joule heat, Q^j, the magnetic energy, W, or the force, F^{jb}, because they include a time–constant term.

The quantities in equations (5.1–57) and (5.1–58) are related by

$$c(r) = a(r) \cos (\phi(r)) \tag{5.1–59}$$

$$s(r) = a(r) \sin (\phi(r)) \tag{5.1–60}$$

$$a^2(r) = c^2(r) + s^2(r) \tag{5.1–61}$$

$$\tan (\phi(r)) = s(r) \; / \; c(r) \tag{5.1–62}$$

In equation (5.1–57) a(r), ϕ(r), c(r) and s(r) depend on space coordinates but not on time. This separation of space and time is taken advantage of to minimize the computational cost. The originally 4 (3 space + 1 time) dimensional real problem can be reduced to a 3 (space) dimensional complex problem. This can be achieved by the complex formalism.

The measurable quantity, q(r,t), is described as the real part of a complex function:

$$q(r,t) = \mathrm{Re}\ \{Q(r)\ \exp\ (j\omega t)\} \qquad (5.1\text{–}63)$$

where: Q(r) = $Q_r(r) + j\ Q_i\ (r)$ (5.1–64)

j = imaginary unit

Re { } = denotes real part of a complex quantity

$Q_r(r)$ and $Q_i(r)$ = real and imaginary parts of Q(r). Note that Q depends only on the space coordinates.

The complex exponential in (5.1–63) can be expressed by sine and cosine as

$$\exp\ (j\omega t) = \cos\ (\omega t) + j\ \sin\ (\omega t) \qquad (5.1\text{–}65)$$

Subsituting (5.1–65) into (5.1–63) and (5.1–64) provides

$$q(r,t) = Q_r(r)\ \cos\ (\omega t) - Q_i(r)\ \sin\ (\omega t) \qquad (5.1\text{–}66)$$

Comparing (5.1–57) with (5.1–66) reveals:

$$c(r) = Q_r(r) \qquad (5.1\text{–}67)$$

$$s(r) = Q_i(r) \qquad (5.1\text{–}68)$$

In words, the complex real, $Q_r(r)$, and imaginary, $Q_i(r)$, parts are the same as the measurable cosine, c(r), and sine, s(r), amplitudes.

A harmonic analysis provides two sets of solution: the real and imaginary components of a complex solution. According to (5.1–57), and (5.1–67) the magnitude of the real and imaginary sets describe the measurable field at t=0 and at ωt = –90 degrees, respectively. Comparing (5.1–58) and (5.1–67) provides:

$$a(r)^2 = Q_r(r)^2 + Q_i(r)^2 \qquad (5.1\text{–}69)$$

$$\tan\ (\phi(r)) = Q_i(r)\ /\ Q_r(r) \qquad (5.1\text{–}70)$$

(5.1–69) expresses the amplitude (peak) and phase angle of the measurable harmonic field quantities by the complex real and imaginary parts.

The time average of harmonic fields such as A, E, B, H, J or V is zero at point r. This is not the case for P, W, or F because they are quadratic functions of B, H or J. To derive the time dependence of a quadratic function – for the sake of simplicity – we deal only with a Lorentz force, F, which is product of J and B. (This is a cross product; but components are not shown to simplify writing. The space dependence is also omitted.)

$$F^{jb}(t) = J(t)\ B(t) = (J_r \cos(\omega t) - J_i \sin(\omega t))\ (B_r \cos(\omega t) - B_i \sin(\omega t))$$
$$= J_r\ B_r \cos(\omega t)^2 + J_i\ B_i \sin(\omega t)^2 - (J_i\ B_r + J_r\ B_i) \sin(\omega t) \cos(\omega t) \qquad (5.1\text{–}71)$$

where: F^{jb} = Lorentz Force density (output as FMAG on **PRESOL** command)

The time average of $\cos^2$ and $\sin^2$ terms is 1/2 whereas that of the sin cos term is zero. Therefore, the time average force is:

$$F^{jb} = 1/2\ (J_r\ B_r + J_i\ B_i) \qquad (5.1\text{–}72)$$

Thus, the force can be obtained as the sum of “real” and “imaginary” forces. In a similar manner the time averaged Joule power density, Q^j, and magnetic energy density, W, can be obtained as:

$$Q^j = 1/2\ (J_r\ E_r + J_i\ E_i) \qquad (5.1\text{–}73)$$

$$W = 1/4\ (B_r\ H_r + B_i\ H_i) \qquad (5.1\text{–}74)$$

where: W = magnetic energy density (output as SENE on **PRESOL** command)

Q^j = Joule Power density heating per unit volume (output as JHEAT on **PRESOL** command)

The time average values of these quadratic quantities can be obtained as the sum of real and imaginary set solutions.

The element returns the integrated value of F^{jb} is output as FJB and W is output as SENE. Q^j is the average element Joule heating and is output as JHEAT. For F and Q^j the 1/2 time averaging factor is taken into account at printout. For W the 1/2 time factor is ignored to preserve the printout of the real and imaginary energy values as the instantaeous stored magnetic energy at t=0 and at ωt = –90 degrees, respectively. The element force, F, is distributed among nodes to prepare a magneto–structural coupling. The average Joule heat can be directly applied to thermo–electric coupling.

5.1.6 Nonlinear Time–Harmonic Magnetic Analysis

Many electromagnetic devices operate with a time–harmonic source at a typical power frequency. Although the power source is time–harmonic, numerical modeling of such devices can not be assumed as a linear harmonic magnetic field problem in general, since the magnetic materials used in these devices have nonlinear B–H curves. A time–stepping procedure should be used instead. This nonlinear transient procedure provides correct solutions for electromagnetic field distribution and waveforms, as well as global quantities such as force and torque. The only problem is that the procedure is often computationally intensive. In a typical case, it takes about 4–5 time cycles to reach a sinusoidal steady state. Since in each cycle, at least 10 time steps should be used, the analysis would require 40–50 nonlinear solution steps.

In many cases, an analyst is often more interested in obtaining global electromagnetic torque and power losses in a magnetic device at sinusoidal steady state, but less concerned with the actual flux density waveform. Under such circumstances, an approximate time–harmonic analysis procedure may be pursued. If posed properly, this procedure can predict the time–averaged torque and power losses with good accuracy, and yet at much reduced computational cost.

The basic principle of the present nonlinear time–harmonic analysis is briefly explained next. First of all, the actual nonlinear ferromagnetic material is represented by another fictitious material based on energy equivalence. This amounts to replacing the DC B–H curve with a fictitious or effective B–H curve based on the following equation for a time period cycle T (Demerdash and Gillott(231)):

$$\frac{1}{2}\int_{0}^{B_{eff}} H_m \, dB_{eff} = \frac{4}{T}\int_{0}^{\frac{T}{4}} \left[\int_{0}^{B} H_m \sin(\omega t) \, dB \right] dt \qquad (5.1\text{–}75)$$

where:

H_m = peak value of magnetic field

B = magnetic flux density

B_{eff} = effective magnetic flux density

T = time period

ω = angular velocity

t = time

With the effective B–H curve, the time transient is suppressed, and the nonlinear transient problem is reduced to a nonlinear time–harmonic one. In this nonlinear analysis, all field quantities are all sinusoidal at a given frequency, similar to the linear harmonic analysis, except that a nonlinear solution has to be pursued.

It should be emphasized that in a nonlinear transient analysis, given a sinusoidal power source, the magnetic flux density B has a non–sinusoidal waveform. While in the nonlinear harmonic analysis, B is assumed sinusoidal. Therefore, it is not the true waveform, but rather represents an approximation of the fundamental time harmonic of the true flux density waveform. The time–averaged global force, torque and loss, which are determined by the approximate fundamental harmonics of fields, are then subsequently approximation to the true values, Numerical benchmarks show that the approximation is of satisfactory engineering accuracy.

5.2 Derivation of Electromagnetic Matrices

The finite element matrix equations can be derived by variational principles. These equations exist for linear and nonlinear material behavior as well as static and transient response. Based on the presence of linear or nonlinear materials (as well as other factors), the program chooses the appropriate Newton–Raphson method. The user may select another method with the (**NROPT** command (see Section 15.9)). When transient affects are to be considered a first order time integration scheme must be involved (**TIMINT** command (see Section 17.2)).

5.2.1 Magnetic Scalar Potential

The scalar potential formulations are restricted to static field analysis with partial orthotropic nonlinear permeability. The degrees of freedom (DOFs), element matrices, and load vectors are presented here in the following form (Zienkiewicz(75), Chari(73), and Gyimesi(141):

Degrees of freedom:

$\{\phi_e\}$ = magnetic scalar potentials (input/output quantity MAG)

Coefficient matrix:

$$\left[K^m\right] = \left[K^L\right] + \left[K^N\right] \tag{5.2–1}$$

$$\left[K^L\right] = \int_{vol} \left(\nabla\{N\}^T\right)^T [\mu] \left(\nabla\{N\}^T\right) d(vol) \tag{5.2–2}$$

$$\left[K^N\right] = \int_{vol} \frac{d\mu_h}{d|H|} \left(\{H\}^T\nabla\{N\}^T\right)^T \left(\{H\}^T\nabla\{N\}^T\right) \frac{d(vol)}{|H|} \tag{5.2–3}$$

Applied loads:

$$\{J_i\} = \int_{vol} \left(\nabla\{N\}^T\right)^T [\mu] \left(\{H_g\} + \{H_c\}\right) d(vol) \tag{5.2–4}$$

where: $\{N\}$ = shape functions $\left(\phi = \{N\}^T\{\phi_e\}\right)$

∇^T = gradient operator = $\left[\frac{\partial}{\partial x}\ \frac{\partial}{\partial y}\ \frac{\partial}{\partial z}\right]$

vol = volume of the element

$\{H_g\}$ = preliminary or "guess" magnetic field (see Section 5.1)

$\{H_c\}$ = coercive force vector (input quantities MGXX, MGYY, MGZZ on **MP** command))

$[\mu]$ = permeability matrix (derived from input material property MURX, MURY, and MURZ (**MP** command) and/or material curve B versus H (**TBPT** commands with **TB**,BH)) (see equations (5.1–6), (5.1–7) and (5.1–8))

$\frac{d\mu_h}{d|H|}$ = derivative of permeability with respect to magnitude of the magnetic field intensity (derived from the input material property curve B versus H (**TBPT** commands with **TB**,BH))

The material property curve is input in the form of B values versus H values and is then converted to a spline fit curve of μ versus H from which the permeability terms μ_h and $\frac{d\mu_h}{d|H|}$ are evaluated.

The coercive force vector is related to the remanent intrinsic magnetization vector as:

$$[\mu]\ \{H_c\} = \mu_o\ \{M_o\} \qquad (5.2–5)$$

where: μ_o = permeability of free space (input quantity MUZRO on **EMUNIT** command)

The Newton–Raphson (**NROPT** command) solution technique is necessary for nonlinear analyses. Adaptive descent is also recommended (**NROPT** command). When adaptive descent is used equation (5.2–1) becomes:

$$\left[K^m\right] = \left[K^L\right] + (1 - \xi)\left[K^N\right] \qquad (5.2–6)$$

where: ξ = descent parameter (see Section 15.9)

5.2.2 Magnetic Vector Potential

The vector potential formulation is applicable to both static and dynamic fields with partial orthotropy nonlinear permeability. The basic equation to be solved is of the form:

$$\left[\bar{C}\right]\ \{\dot{u}\} + [\bar{K}]\ \{u\} = \{\bar{J}_i\} \qquad (5.2–7)$$

The terms of this equation are defined below (Biro(120)); the edge–flux formulation matrices are obtained from these terms in section 14.117 following Gyimesi and Ostergaard(201).

Degrees of freedom:

$$\{u\} = \begin{Bmatrix} \{A_e\} \\ \{\nu_e\} \end{Bmatrix} \qquad (5.2–8)$$

where:

$\{A_e\}$ = magnetic vector potentials (input/output quantity AX, AY, AZ)

$\{\nu_e\}$ = time integrated electric scalar potential ($\nu = \int Vdt$) (input/output quantity VOLT)

The VOLT DOF is a time integrated electric potential to allow for symmetric matrices.

Coefficient matrices:

$$[\overline{K}] = \begin{bmatrix} [K^{AA}] & [0] \\ [K^{vA}] & [0] \end{bmatrix} \qquad (5.2–9)$$

$$\left[K^{AA}\right] = \left[K^{L}\right] + \left[K^{N}\right] + [K^{G}] \qquad (5.2–10)$$

$$\left[K^{L}\right] = \int_{vol} (\nabla \text{ x } [N_A]^T)^T[\nu] \; (\nabla \text{ x } [N_A]^T - [N_A][\sigma] \; (\{v\} \text{ x } \nabla \text{ x } [N_A]^T)) \; d(vol) \qquad (5.2–11)$$

$$\left[K^{G}\right] = \int_{vol} (\nabla \cdot [N_A]^T)^T \; [\nu] \; (\nabla \cdot [N_A]^T) \; d(vol) \qquad (5.2–12)$$

$$\left[K^{N}\right] = 2 \int_{vol} \frac{d\nu_h}{d(|B|^2)} \left(\{B\}^T(\nabla \text{ x } [N_A]^T)\right)^T \left(\{B\}^T(\nabla \text{ x } [N_A]^T)\right) \; d(vol) \qquad (5.2–13)$$

$$\left[K^{VA}\right] = -\int (\nabla \; [N]^T)^T \; [\sigma] \; \{v\} \text{ x } \nabla \text{ x } [N_A]^T \; d(vol) \qquad (5.2–14)$$

$$[\bar{C}] = \begin{bmatrix} [C^{AA}] & [C^{Av}] \\ [C^{Av}]^T & [C^{vv}] \end{bmatrix} \qquad (5.2\text{–}15)$$

$$\left[C^{AA}\right] = \int_{vol} [N_A]\ [\sigma]\ [N_A]^T\ d(vol) \qquad (5.2\text{–}16)$$

$$\left[C^{Av}\right] = \int_{vol} [N_A]\ [\sigma]\ \nabla\ \{N\}^T\ d(vol) \qquad (5.2\text{–}17)$$

$$\left[C^{vv}\right] = \int_{vol} (\nabla\{N\}^T)^T\ [\sigma]\ \nabla\{N\}^T\ d(vol) \qquad (5.2\text{–}18)$$

Applied loads:

$$\{\bar{J}_i\} = \begin{Bmatrix} \{J^A\} \\ \{I^t\} \end{Bmatrix} \qquad (5.2\text{–}19)$$

$$\{J^A\} = \{J^S\} + \{J^{pm}\} \qquad (5.2\text{–}20)$$

$$\{J^S\} = \int_{vol} \{J_s\}\ [N_A]^T\ d(vol) \qquad (5.2\text{–}21)$$

$$\{J^{pm}\} = \int_{vol} (\nabla\ x\ [N_A]^T)^T\ \{H_c\}\ d(vol) \qquad (5.2\text{–}22)$$

$$\{I^t\} = \int_{vol} \{J_t\}\ [N_A]^T\ d(vol) \qquad (5.2\text{–}23)$$

where: $\left[N_A\right]$ = matrix of element shape functions for {A}
$\left(\{A\} = [N_A]^T\{A_e\} \; ; \; \{A_e\}^T = \left\lfloor \{A_{xe}\}^T \; \{A_{ye}\}^T \; \{A_{ze}\}^T \right\rfloor\right)$

$\{N\}$ = vector of element shape functions for {V} $\left(V = \{N\}^T\{V_e\}\right)$

$\{J_s\}$ = source current density vector (input as JS on **BFE** command)

$\{J_t\}$ = total current density vector (input as JS on **BFE** command) (valid for 2–D analysis only)

vol = volume of the element

$\{H_c\}$ = coercive force vector (input as MGXX, MGYY, MGZZ on **MP** command)

ν_o = reluctivity of free space (derived from value using MUZRO on **EMUNIT** command)

$[\nu]$ = partially orthotropic reluctivity matrix (inverse of $[\mu]$, derived from input material property curve B versus H (**TBPT** commands with **TB**,BH))

$\dfrac{d\nu_h}{d(|B|)^2}$ = derivative of reluctivity with respect to the magnitude of magnetic flux squared (derived from input material property curve B versus H (**TBPT** commands with **TB**,BH))

$[\sigma]$ = orthotropic conductivity (input as RSVX, RSVY, RSVZ on **MP** command (inverse)) (see equation (5.1–11)).

$\{v\}$ = velocity vector

The coercive force vector is related to the remanent intrinsic magnetization vector as:

$$\{H_c\} = \frac{1}{\nu_o}\,[\nu]\,\{M_o\} \qquad (5.2\text{–}24)$$

The material property curve is input in the form of B values versus H values and is then converted to a spline fit curve of ν versus $|B|^2$ from which the isotropic reluctivity terms ν_h and $\dfrac{d\nu_h}{d\left(|B|^2\right)}$ are evaluated.

The above element matrices and load vectors are presented for the most general case of a vector potential analysis. Many simplifications can be made depending on the conditions of the specific problem. In 2–D there is only one component of the vector potential as opposed to three for 3–D problems (AX, AY, AZ).

Combining some of the above equations, the variational equilibrium equations may be written as:

$$\{A_e\}^T\Big([K^{AA}]\{A_e\} + [K^{AV}]\{\nu_e\} + [C^{AA}]d/dt\{A_e\} + [C^{AV}]d/dt\{\nu_e\} - \{J^A\}\Big) = 0 \qquad (5.2\text{–}25)$$

$$\{\nu_e\}^T\Big([K^{VA}]\{A_e\} + [K^{VV}]\{\nu_e\} + [C^{VA}]d/dt\{A_e\} + [C^{VV}]d/dt\{\nu_e\} - \{I^t\}\Big) = 0 \qquad (5.2\text{–}26)$$

Here T denotes transposition.

Static analyses require only the magnetic vector potential DOFs (KEYOPT controlled) and the K coefficient matrices. If the material behavior is nonlinear then the Newton–Raphson solution procedure is required (**NROPT** command (see Section 15.9)).

For 2–D dynamic analyses a current density load of either source ($\{J_s\}$) or total $\{J_t\}$ current density is valid. J_t input represents the impressed current expressed in terms of a uniformly applied current density. This loading is only valid in a skin–effect analysis with proper coupling of the VOLT DOFs. In 3–D only source current density is allowed. The electric scalar potential must be constrained properly in order to satisfy the fundamentals of electromagnetic field theory. This can be achieved by direct specification of the potential value (**D** command) as well as with coupling (**CP** command) and constraining (**CE** command).

The general transient analysis (**ANTYPE**,TRANS (see Section 15.4)) accepts nonlinear material behavior (field dependent [ν] and permanent magnets (MGXX, MGYY, MGZZ). Harmonic transient analyses (**ANTYPE**,HARMIC (see Section 17.4)) is a linear analyses with sinusoidal loads; therefore, it is restricted to linear material behavior without permanent magnets.

5.2.3 Electric Scalar Potential

The electric scalar potential formulation is applicable to electrostatic field analysis (with orthotropic permittivity). The DOFs, elemental matrices, and load vectors are presented as follows:

Degrees of Freedom:

$\{V_e\}$ = electric scalar potential (input/output quantity VOLT)

Coefficient Matrix:

$$[K_e^m] = \int_{vol} (\nabla\ \{N\}^T)^T\ [\epsilon]\ (\nabla\ \{N\}^T)\ d(vol) \qquad (5.2\text{–}27)$$

Applied Loads:

$$\{Q_e\} = \{Q_e^c\} + \{Q_e^{sc}\} \qquad (5.2\text{–}28)$$

$$\{Q_e^c\} = \int_{vol} \{\rho\} \; \{N\}^T \; d(vol) \tag{5.2–29}$$

$$\{Q_e^{sc}\} = \int_{s} \{\rho_s\} \; \{N\}^T \; ds \tag{5.2–30}$$

where:
{N} = vector of element shape functions $\left(V = \{N\}^T\{V_e\}\right)$
{ϱ} = charge density vector (input with CHRGD on **BF** command)
$\{\varrho_s\}$ = surface charge density vector (input with CHRGS on **SF** command)

5.3 Electromagnetic Field Evaluations

The basic electromagnetic field results include magnetic field intensity, magnetic flux density, magnetic forces and current densities. These types of evaluations are somewhat different for magnetic scalar and vector potential formulations.

5.3.1 Magnetic Scalar Potential Results

The first derived result is the magnetic field intensity which is divided into two parts (see Section 5.1); a generalized field $\{H_g\}$ and the gradient of the generalized potential $-\nabla\phi_g$. This gradient (referred to here as $\{H_\phi\}$) is evaluated at the integration points using the element shape function as:

$$\{H_\phi\} = -\nabla\{N\}^T\{\phi_g\} \tag{5.3–1}$$

where: ∇^T = gradient operator = $\left[\frac{\partial}{\partial x}\ \frac{\partial}{\partial y}\ \frac{\partial}{\partial z}\right]$

$\{N\}$ = shape functions

$\{\phi_g\}$ = nodal generalized potential vector

The magnetic field intensity is then:

$$\{H\} = \{H_g\} + \{H_\phi\} \tag{5.3–2}$$

where: $\{H\}$ = output quantity H

Then the magnetic flux density is computed from the field intensity:

$$\{B\} = [\mu]\,\{H\} \tag{5.3–3}$$

where: $\{B\}$ = magnetic flux density (output quantity B)

$[\mu]$ = permeability matrix (defined in equations (5.1–6), (5.1–7), and (5.1–8))

Nodal values of field intensity and flux density are computed from the integration points values as described in Section 13.6.

Magnetic forces are also available and are discussed below.

5.3.2 Magnetic Vector Potential Results

The magnetic flux density is the first derived result. It is defined as the curl of the magnetic vector potential. This evaluation is performed at the integration points using the element shape functions:

$$\{B\} = \nabla \times [N_A]^T\{A_e\} \tag{5.3–4}$$

where:
- $\{B\}$ = magnetic flux density (output quantity B
- ∇ x = curl operator
- $[N_A]$ = shape functions
- $\{A_e\}$ = nodal magnetic vector potential

Then the magnetic field intensity is computed from the flux density:

$$\{H\} = [\nu]\ \{B\} \tag{5.3–5}$$

where:
- $\{H\}$ = magnetic field intensity (output quantity H)
- $[\nu]$ = reluctivity matrix

Nodal values of field intensity and flux density are computed from the integration point value as described in Section 13.6.

Magnetic forces are also available and are discussed below.

For a vector potential transient analysis current densities are also calculated.

$$\{J_t\} = \{J_e\} + \{J_s\} + \{J_v\} \tag{5.3–6}$$

where:
- $\{J_t\}$ = total current density

$$\{J_e\} = -[\sigma]\left\{\frac{\partial A}{\partial t}\right\} = -[\sigma]\frac{1}{n}\sum_{i=1}^{n}[N_A]^T\{\dot{A}_e\} \tag{5.3–7}$$

where:
- $\{J_e\}$ = current density component due to {A}
- $[\sigma]$ = conductivity matrix
- n = number of integration points
- $[N_A]$ = element shape functions for {A} evaluated at the integration points
- $\{\dot{A}_e\}$ = time derivative of magnetic vector potential

and

$$\{J_s\} = -[\sigma]\nabla V = -[\sigma]\frac{1}{n}\sum_{i=1}^{n}\nabla\{N\}^T\{V_e\} \tag{5.3–8}$$

where: $\{J_s\}$ = current density component due to V

∇ = divergence operator

$\{V_e\}$ = electric scalar potential

$\{N\}$ = element shape functions for V evaluated at the integration points

and

$$\{J_v\} = \{v\} \text{ x } \{B\} \tag{5.3–9}$$

where $\{J_v\}$ = velocity current density vector

$\{v\}$ = applied velocity vector

$\{B\}$ = magnetic flux density (see equation (5.3–4))

5.3.3 Magnetic Forces

Magnetic forces are computed by elements using the vector potential method (PLANE13), PLANE53, and SOLID97) and the scalar potential method (SOLID5, SOLID96, and SOLID98). None are affected by the **ERESX** command. Three different techniques are used to calculate magnetic forces at the element level.

1. Lorentz forces.

Magnetic forces in current carrying conductors (element output quantity FJB) are numerically integrated using:

$$\left\{F^{jb}\right\} = \int_{vol} \{N\}^T(\{J\} \text{ x } \{B\}) \, d(vol) \tag{5.3–10}$$

where: $\{N\}$ = vector of shape functions

For a 2–D analysis, the corresponding electromagnetic torque about +Z is given by:

$$T^{jb} = \{Z\} \cdot \int_{vol} \{r\} \times (\{J\} \text{ x } \{B\}) \, d(vol) \tag{5.3–11}$$

where: $\{Z\}$ = unit vector along +Z axis

$\{r\}$ = position vector in the global Cartesian coordinate system

In a time–harmonic analysis, the time–averaged Lorentz force and torque are computed by:

$$\left\{F^{jb}_{av}\right\} = \frac{1}{2} \int_{vol} \{N\}^T\left(\{J\}^* x \{B\}\right) d(vol) \qquad (5.3–12)$$

and

$$T^{jb}_{av} = \{Z\} \cdot \int_{vol} \{r\} \times (\{J\} x \{B\}) d(vol) \qquad (5.3–13)$$

respectively.

where: $\{J\}^*$ = complex conjugate of $\{J\}$

2. Maxwell forces.

The Maxwell stress tensor is used to determine forces on ferromagnetic regions (element output quantity FMX). This force calculation is performed on surfaces of air material elements which have a nonzero face loading specified (MXWF on **SF** commands) (Moon(77)). For the 2–D application, this method uses extrapolated field values and results in the following numerically integrated surface integral:

$$\{F^{mx}\} = \frac{1}{\mu_o} \int_s \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \end{Bmatrix} ds \qquad (5.3–14)$$

where:
μ_o = permeability of free space (input on **EMUNIT** command)
T_{11} = $B_x^2 - \frac{1}{2} |B|^2$
T_{12} = $B_x B_y$
T_{21} = $B_x B_y$
T_{22} = $B_y^2 - \frac{1}{2} |B|^2$

3–D applications are an extension of the 2–D case.

For a 2–D analysis, the corresponding electromagnetic torque about +Z axis is given by:

$$T^{mx} = \{Z\} \cdot \frac{1}{\mu_o} \int_s \{r\} \times \left[(\hat{n} \cdot \{B\})\{B\} - \frac{1}{2} (\{B\} \cdot \{B\}) \hat{n}\right] ds \qquad (5.3–15)$$

where: $\hat{n}$ = unit surface normal in the global Cartesian coordinate system

In a time–harmonic analysis, the time–averaged Maxwell stress tensor force and torque are computed by:

$$\{F_{av}^{mx}\} = \frac{1}{2\mu_o}\int_s \left[\mathrm{Re}\left\langle(\hat{n}\cdot\{B\}^*)\{B\}\right\rangle - \frac{1}{2}(\{B\}\cdot\{B\}^*)\,\hat{n}\right] ds \qquad (5.3\text{–}16)$$

and

$$T_{av}^{mx} = \{Z\}\cdot\frac{1}{2\mu_o}\int_s \{r\}\times\left[\mathrm{Re}\left\langle(\hat{n}\cdot\{B\}^*)\{B\}\right\rangle - \frac{1}{2}(\{B\}\cdot\{B\}^*)\,\hat{n}\right] ds \qquad (5.3\text{–}17)$$

respectively.

where: $\{B\}^*$ = complex conjugate of {B}

Re{ } = denotes real part of a complex quantity

3. Virtual work forces.

Magnetic forces calculated using the virtual work method (element output quantity FVW) are obtained as the derivative of the energy versus the displacement (MVDI on **BF** commands) of the movable part. This calculation is valid for a layer of air elements surrounding a movable part (Coulomb(76)). To determine the total force acting on the body, the forces in the air layer surrounding it can be summed. The basic equation for force of an air material element in the s direction is:

$$F_s = \int_{vol}\{B\}^T\left\{\frac{\partial H}{\partial s}\right\} d(vol) + \int_{vol}\left[\int\{B\}^T\{dH\}\right]\frac{\partial}{\partial s}\,d(vol) \qquad (5.3\text{–}18)$$

where: F_s = force in element in the s direction

$\left\{\frac{\partial H}{\partial s}\right\}$ = derivative of field intensity with respect to displacement s

s = virtual displacement of the nodal coordinates taken alternately to be in the X, Y, Z global directions

vol = volume of the element

For a 2–D analysis, the corresponding electromagnetic torque about +Z axis is given by:

$$T^{vw} = \{Z\}\cdot\frac{1}{\mu_o}\int_{vol}\{r\}\times\left[\frac{1}{2}(\{B\}\cdot\{B\})\nabla\{s\} - (\{B\}\cdot\nabla\{s\})\{B\}\right] d(vol) \qquad (5.3\text{–}19)$$

In a time–harmonic analysis, the time–averaged virtual work force and torque are computed by:

$$\{F_{av}^{vw}\} = \frac{1}{2\mu_o} \int_{vol} \left[\frac{1}{2} (\{B\}^* \cdot \{B\}) \nabla\{s\} - \text{Re} \left\langle (\{B\}^* \cdot \nabla\{s\})\{B\} \right\rangle \right] d(vol) \qquad (5.3\text{–}20)$$

and

$$T_{av}^{vw} = \{Z\} \cdot \frac{1}{2\mu_o} \int_{vol} \{r\} \times \left[\frac{1}{2} (\{B\}^* \cdot \{B\}) \nabla\{s\} - \text{Re} \left\langle (\{B\}^* \cdot \nabla\{s\})\{B\} \right\rangle \right] d(vol) \qquad (5.3\text{–}21)$$

respectively.

5.3.4 Joule Heat

Joule heat is computed by elements using the vector potential method (PLANE13, PLANE53, and SOLID97) if the element has a nonzero resistivity (material property RSVX) and a nonzero current density (either applied J_s or resultant J_t). It is available as the output power loss (output quantity JHEAT) or as the coupled field heat generation load (**LDREAD**,HGEN).

Joule heat per element is computed as:

1. Static or Transient Magnetic Analysis

$$Q^j = \frac{1}{n} \sum_{i=1}^{n} [\rho]\ \{J_{ti}\} \cdot \{J_{ti}\} \qquad (5.3\text{–}22)$$

where:
Q^j = Joule heat per unit volume
n = number of integration points
$[\varrho]$ = resistivity matrix (input as RSVX, RSVY, RSVZ on **MP** command)
$\{J_{ti}\}$ = total current density in the element at integration point i

2. Harmonic Magnetic Analysis

$$Q^j = \text{Re} \left[\frac{1}{2n} \sum_{i=1}^{n} [\rho]\ \{J_{ti}\} \cdot \{J_{ti}\}^* \right] \qquad (5.3\text{–}23)$$

where:
Re = real component
$\{J_{ti}\}$ = complex total current density in the element at integration point i
$\{J_{ti}\}^*$ = complex conjugate of $\{J_{ti}\}$

Joule heating for the thermal–electric elements (PLANE67, LINK68, SOLID69, and SHELL157) is discussed in Section 11.2.

5.3.5 Electric Scalar Potential Results

The first derived result is the electric field. It is defined as the negative gradient of the electric scalar potential. This evaluation is performed at the integration points using the element shape functions:

$$\{E\} = -\nabla \{N\}^T\{V_e\} \qquad (5.3\text{–}24)$$

where: $\{E\}$ = electric field (output quantity EF)
∇ = gradient operator
$\{N\}$ = shape functions
$\{V_e\}$ = nodal electric scalar potential

The electric flux density is computed from the electric field (see equation (5.1–12))

$$\{D\} = [\epsilon]\ \{E\} \qquad (5.3\text{–}25)$$

where: $\{D\}$ = electric flux density (output quantity D)
$[\varepsilon]$ = permittivity matrix

Nodal values of field intensity and flux density are computed from the integration point values as described in Section 13.6.

Electrostatic forces are also available and are discussed below.

5.3.6 Electrostatic Forces

Electrostatic forces are determined using the Maxwell stress tensor. This force calculation is performed on surfaces of elements which have a nonzero face loading specified (MXWF on **SF** commands). For the 2–D application, this method uses extrapolated field values and results in the following numerically integrated surface integral:

$$\{F^{mx}\} = \epsilon_o \int_s \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{Bmatrix} n_1 \\ n_2 \end{Bmatrix} ds \qquad (5.3\text{–}26)$$

where: ε_o = free space permittivity (input as PERX on **MP** command)
T_{11} = $E_x^2 - \frac{1}{2} \mid E \mid^2$
T_{12} = $E_x\,E_y$
T_{21} = $E_y\,E_x$

$T_{22} = E_y^2 - \frac{1}{2} | E |^2$

n_1 = component of unit normal in x–direction

n_2 = component of unit normal in y–direction

s = surface area of the element face

$| E |^2 = E_x^2 + E_y^2$

3–D applications are an extension of the 2–D case.

5.3.7 Electric Constitutive Error

The dual constitutive error estimation procedure as implemented for the electrostatic p–elements SOLID127 and SOLID128 is activated with the **PEMOPTS** command and is briefly discussed in this section. Suppose a field pair $(\{\hat{E}\}, \{\hat{D}\})$, which verifies the Maxwell's equations (5.1–47) and (5.1–48), can be found for a given problem. This couple is the true solution if the pair also verifies the constitutive relation (5.1–49). Or, the couple is just an approsimate solution to the problem, and the quantity

$$\{\hat{e}\} = \{\hat{D}\} - [\varepsilon] \cdot \{\hat{E}\} \tag{5.3–27}$$

is called error in constitutive relation, as originally suggested by Ladeveze(274) for linear elasticity. To measure the error $\{\hat{e}\}$, the energy norm over he whole domain Ω is used:

$$\|\{\hat{e}\}\|_\Omega = \|\{\hat{D}\} - [\varepsilon] \cdot \{\hat{E}\}\|_\Omega \tag{5.3–28}$$

with

$$\|\{\sigma\}\|_\Omega = \left[\int_\Omega \{\sigma\}^T [\varepsilon]^{-1} \{\sigma\} \, d\Omega \right]^{\frac{1}{2}} \tag{5.3–29}$$

By virtue of Synge's hypercircle theorem(275), it is possible to define a relative error for the problem:

$$\varepsilon_\Omega = \frac{\|\{\hat{D}\} - [\varepsilon] \cdot \{\hat{E}\}\|_\Omega}{\|\{\hat{D}\} + [\varepsilon] \cdot \{\hat{E}\}\|_\Omega} \tag{5.3–30}$$

The global relative error (5.3–30) is seen as sum of element contributions:

$$\varepsilon_{\Omega}^2 = \sum_{E} \varepsilon_E^2 \tag{5.3–31}$$

where the relative error for an element E is given by

$$\varepsilon_E = \frac{\|\{\hat{D}\} - [\varepsilon] \cdot \{\hat{E}\}\|_E}{\|\{\hat{D}\} + [\varepsilon] \cdot \{\hat{E}\}\|_{\Omega}} \tag{5.3–32}$$

The global error ε_{Ω} allows to quantify the quality of the approximate solution pair $(\{\hat{E}\}, \{\hat{D}\})$, and local error ε_E allows to localize the error distribution in the solution domain as required in a p–adaptive analysis.

5.4 Voltage Forced and Circuit–Coupled Magnetic Field

The magnetic vector potential formulation discussed in Chapter 5 requires electric current density as input. In many industrial applications, a magnetic device is often energized by an applied voltage or by a controlling electric circuit. In this section, a brief outline of the theoretical foundation for modeling such voltage forced and circuit–coupled magnetic field problems is provided. The formulations apply to static, transient and harmonic analysis types.

To make the discussion simpler, a few definitions are introduced first. A stranded coil refers to a coil consisting of many turns of conducting wires. A massive conductor refers to an electric conductor where eddy currents must be accounted for. When a stranded coil is connected directly to an applied voltage source, we have a voltage forced problem. If a stranded coil or a massive conductor is connected to an electric circuit, we have a circuit–coupled problem. A common feature in both voltage forced and circuit–coupled problems is that the electric current in the coil or conductor must be treated as an additional unknown. The electric current is represented by the degree–of–freedom (DOF) label CURR.

5.4.1 Voltage Forced Magnetic Field

Assume that a stranded coil has an isotropic and constant magnetic permeability and electric conductivity. Then, by using the magnetic vector potential approach from Chapter 5, the following element matrix equation is derived.

$$\begin{bmatrix} [0] & [0] \\ [C^{iA}] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] \\ [0] & [K^{ii}] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{V_o\} \end{Bmatrix} \qquad (5.4\text{–}1)$$

where:

$\{A\}$ = nodal magnetic vector potential vector (AX, AY, AZ)

$\cdot$ = time derivative

$\{i\}$ = nodal electric current vector (CURR)

$[K^{AA}]$ = potential stiffness matrix

$[K^{ii}]$ = resistive stiffness matrix

$[K^{Ai}]$ = potential–current coupling stiffness matrix

$[C^{iA}]$ = inductive damping matrix

$\{V_o\}$ = applied voltage drop vector

The magnetic flux density {B}, the magnetic field intensity {H}, magnetic forces, and Joule heat can be calculated from the nodal magnetic vector potential {A} using equations (5.3–3) and (5.3–4).

The nodal electric current represents the current in a wire of the stranded coil. Therefore, there is only one independent electric current unknown in each stranded coil. In addition, there is no gradient or flux calculation associated with the DOF label CURR.

5.4.2 Circuit–Coupled Magnetic Field

When a stranded coil or a massive conductor is connected to an electric circuit, both the electric current and voltage (not the time–integrated voltage) should be treated as unknowns. To achieve a solution for this problem, the finite element equation and electric circuit equations must be solved simultaneously.

The modified nodal analysis method (McCalla(188)) is used to build circuit equations for the following linear electric circuit element options:

1. resistor
2. inductor
3. capacitor
4. voltage source
5. current source
6. stranded coil current source
7. 2–D massive conductor voltage source
8. 3–D massive conductor voltage source
9. mutual inductor
10. voltage–controlled current source
11. voltage–controlled voltage source
12. current–controlled voltage source
13. current–controlled current source

These circuit elements are implemented in element CIRCU124.

Assuming an isotropic and constant magnetic permeability and electric conductivity, the following element matrix equation is derived for a circuit–coupled stranded coil:

$$\begin{bmatrix} [0] & [0] & [0] \\ [C^{iA}] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] & [0] \\ [0] & [K^{ii}] & [K^{ie}] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \\ \{e\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix} \qquad (5.4\text{–}2)$$

where: {e} = nodal electromotive force drop (EMF)

$[K^{ie}]$ = current–emf coupling stiffness

For a circuit–coupled massive conductor, the matrix equation is:

$$\begin{bmatrix} [C^{AA}] & [0] & [0] \\ [0] & [0] & [0] \\ [C^{VA}] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [0] & [K^{AV}] \\ [0] & [0] & [0] \\ [0] & [K^{iV}] & [K^{VV}] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix} \qquad (5.4–3)$$

where: {V} = nodal electric voltage vector (VOLT)

$[K^{VV}]$ = voltage stiffness matrix

$[K^{iV}]$ = current–voltage coupling stiffness matrix

$[C^{AA}]$ = potential damping matrix

$[C^{VA}]$ = voltage–potential damping matrix

The magnetic flux density {B}, the magnetic field intensity {H}, magnetic forces and Joule heat can be calculated from the nodal magnetic vector potential {A} using equations (5.3–3) and (5.3–4).

5.5 High–Frequency Electromagnetic Simulation

In previous sections, it has been assumed that the electromagnetic field problem under consideration is either static or quasi–static. For quasi–static or eddy current problems, the displacement current is ignored, and Maxwell's equations (5.1–1) through (5.1–4) reduce to (5.1–39) through (5.1–41). This approach is valid when the excitation wavelength is much larger than physical dimensions of a given problem. For many other applications using higher frequencies, the wavelength can become comparable or smaller than the physical dimensions. Under such circumstances, the quasi–static approach becomes invalid. Instead, the full set of Maxwell's equations (5.1–1) through (5.1–4) have to be solved. The underlying problem is defined here as a high–frequency or full–wave problem, in contrast to low frequency or quasi–static ones in previous sections. The purpose of this section is to introduce basic full–wave formulations, and define useful output quantities.

5.5.1 Full–Wave Electromagnetic Formulations

Consider first a scenario depicted in Figure 5.5–1. In this figure, a closed surface S divides the space domain into an exterior and an interior part. An electromagnetic source located somewhere in the exterior domain generates an electromagnetic plane wave (E^{inc}, H^{inc}). The interior domain, denoted by (Ω), is the primary concern here. It may encompass waveguide, antenna, dielectric, conductor, ground plane, etc. One may notice that the situation is somewhat similar to that shown in Figure 5.1–1, except that the wavelength here is comparable to the dimension of (Ω). The objective next is to find the electromagnetic field within the solution domain (Ω).

Based on equations (5.1–1) and (5.1–2), the electric field E satisfies the following equation:

$$[\epsilon]\,\frac{\partial^2 \vec{E}}{\partial t^2} + [\sigma]\,\frac{\partial \vec{E}}{\partial t} + \nabla \times [\upsilon]\ \nabla \times \vec{E} = -\,\frac{\partial \vec{J}_s}{\partial t} \qquad (5.5\text{–}1)$$

where: $\vec{E}$ = electric field

$\vec{J}_s$ = source current density

See section 5.1 for other terms.

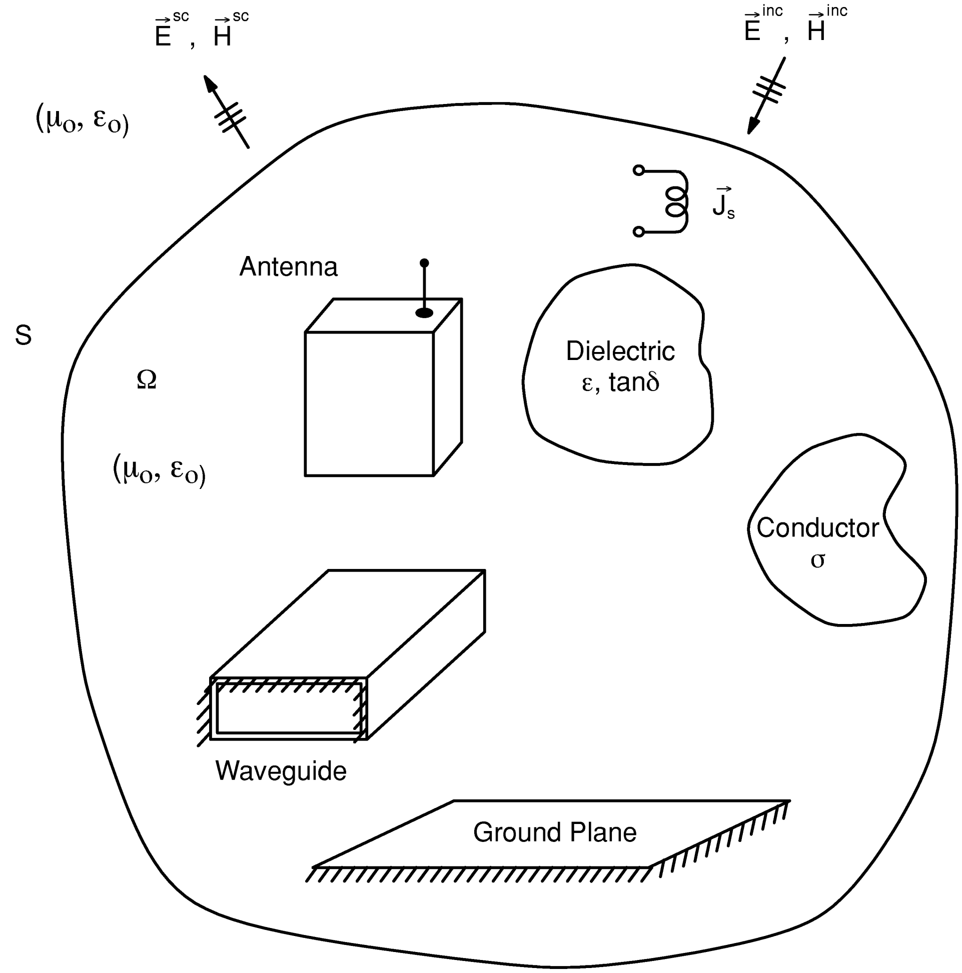

$\vec{E}^{sc}$, $\vec{H}^{sc}$
$\vec{E}^{inc}$, $\vec{H}^{inc}$
(μ_o, ε_o)
$\vec{J}_s$
Antenna
S
Ω
Dielectric
ε, tanδ
(μ_o, ε_o)
Conductor
σ
Waveguide
Ground Plane

Figure 5.5–1 A Typical High–Frequency Solution Domain

Equation (5.5–1) can be cast into an integral form:

$$\int_{\Omega} \vec{W}_i \cdot [\epsilon] \frac{\partial^2 \vec{E}}{\partial t^2} \, d(vol) + \int_{\Omega} \vec{W}_i \cdot [\sigma] \frac{\partial \vec{E}}{\partial t} \, d(vol) + \int_{\Omega} \nabla \times \vec{W}_i \cdot [\upsilon] \, \nabla \times \vec{E} \, d(vol)$$

$$+ \int_{S} \vec{W}_i \cdot \hat{n} \times [\upsilon] \, \nabla \times \vec{E} \, d(area) = - \int_{\Omega} \vec{W}_i \cdot \frac{\partial \vec{J}_s}{\partial t} \, d(vol) \tag{5.5–2}$$

where: $\vec{W}_i$ = ith vector weighting function

Temporarily let the surface integral vanish (we will deal with it later on). Then, a proper finite element implementation yields the following dynamic matrix equation:

$$[M] \frac{\partial^2}{\partial t^2} \{A_x\} + [C] \frac{\partial}{\partial t} \{A_x\} + [K] \{A_x\} = \{F\} \qquad (5.5\text{–}3)$$

where: $\{A_x\}$ = covariant components of electric field (AX DOFs)

$[W]$ = vector shape functions

$$[K] = \int_{vol} \left(\nabla \times [W]^T\right)^T [\upsilon] \left(\nabla \times [W]^T\right) d(vol)$$

$$[C] = \int_{vol} [W]\ [\sigma]\ [W]^T\ d(vol)$$

$$[M] = \int_{vol} [W]\ [\epsilon]\ [W]^T\ d(vol)$$

$$\{F\} = - \int_{vol} [W] \frac{\partial}{\partial t} \{J_s\}\ d(vol)$$

Refer to sections 14.119.2 and 14.120.2 for definitions of vector shape functions and physical meanings of $\{A_x\}$.

In deriving the dynamic matrix equation (5.5–3), no boundary condition of any kind has been incorporated. These are dealt with next. Before discussing these boundary conditions, the time–harmonic electromagnetic field will be focused upon. All field and source quantities are now assumed to have a time variation exp(jωt). For instance, the instantaneous electric field E(r,t) is related to its complex field through:

$$\vec{E}\ (\{r\}\ t) = \text{Re} \left\{\{E\}\ e^{j\omega t}\right\} = \frac{1}{2} \left[\{E\}\ e^{j\omega t} + (\{E\}\ e^{j\omega t})^*\right] \qquad (5.5\text{–}4)$$

where: superscript * = complex conjugate

It is important to notice that the magnitude of any instantaneous field represents the peak value that is related to its root–mean–square (RMS) value by peak = $\sqrt{2}$ RMS value.

Refer to Figure 5.5–1. Typical boundary conditions are of these types: (1) PEC (2) PMC (3) Impedance boundary (4) Waveguide matching boundary (5) Infinite boundary. Formal definitions of these are given below.

PEC Condition

A PEC condition stands for a Perfect Electric Conductor condition. On this boundary, the electric field E has no component tangential to the boundary:

$$\{\hat{n}\} \times \{E\} = 0 \qquad (5.5\text{–}5)$$

where: $\{\hat{n}\}$ = surface normal.

A PEC condition exists typically in two cases. One is on the surface of a highly electrically conducting conductor particularly when dealing with high frequencies (i.e., skin depth negligible). Another is on an anti–symmetry plane for the electric field E. It should be stated that the condition (5.5–5) does not automatically come into the integral form (5.5–2), but rather should be treated as a constraint.

PMC Condition

A PMC condition stands for a Perfect Magnetic Conductor condition. On this boundary, the magnetic field H has no component tangential to the boundary:

$$\{\hat{n}\} \times \{H\} = 0 \qquad (5.5\text{–}6)$$

A PMC condition exists typically in two cases. One is on the surface of a highly permeable material. Another is on an anti–symmetry plane for the magnetic field H. In contrast to the PEC condition, the condition (5.5–6) is automatically accounted for by equation (5.5–2).

Impedance Boundary Condition

An impedance boundary condition exists on a surface where the tangential electric field is related the tangential magnetic field by

$$\{\hat{n}\} \times \{E\} = Z_s \; \{\hat{n}\} \times \{H\} \times \{\hat{n}\} \qquad (5.5\text{–}7)$$

where: Z_S = surface impedance.

Z_S has both real and imaginary parts,

$$Z_s = R_s + jX_s \qquad (5.5\text{–}8)$$

where: R_S = surface resistivity and will be used in Section 5.5.2 for surface loss computation.

Typically, an impedance boundary is used to represent a conductor with a small skin depth (Mitzner(210)). To explain what this means, consider the air–conductor interface as shown in Figure 5.5–2. Suppose the conductor has a relative permeability μ_r and a conductivity σ. If σ is infinite, a PEC condition as discussed previously can be used. If σ is large, an electromagnetic field can only penetrate a very short distance into the conductor. Instead of physically modeling this short distance, one can replace the whole conductor by a surface impedance

$$Z_{sc} = (1 + j)\sqrt{\frac{\omega\mu_r\mu_o}{2\sigma}} \qquad (5.5–9)$$

Notice that the surface normal $\{\hat{n}\}$ is pointing into the conductor.

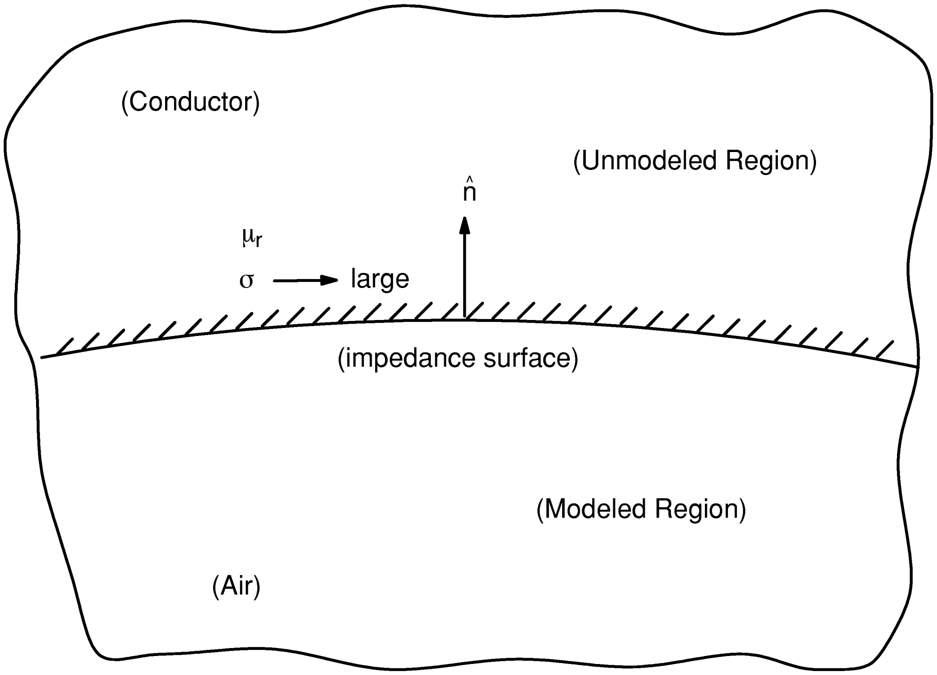


Figure 5.5–2 A Conductor with a Small Skin Depth is Approximated by an Impedance Condition

Waveguide Matching Condition

When it comes to numerical simulation of wave phenomena, it is necessary to truncate the space domain into a manageable solution domain as depicted in Figure 5.5–1 . Typically, domain truncation has to be made 1) at a waveguide port, and 2) at a nearby free–space surface. A waveguide matching condition, along with the infinite boundary condition to be discussed next, is an artificial condition introduced to account for any infinity effect. Since the basic function of the artificial condition is to absorb any out–going electromagnetic wave, it is oftentimes called an absorbing boundary condition (Mittra and Ramahi(211)). Before defining the condition, let the electric field {E} at the port surface be decomposed into an incident part and a scattered part:

$$\{E\} = \{E^{inc}\} + \{E^{sc}\} \qquad (5.5–10)$$

Consequently the surface integral in equations (5.5–2) is split into two parts:

$$\int_S \vec{W}_i \cdot \hat{n} \times \nabla \times \vec{E}\ d(area) = \int_S \vec{W}_i \cdot \hat{n} \times \nabla \times \vec{E}^{sc}\ d(area) + \vec{W}_i \cdot \hat{n} \times \nabla \times \vec{E}^{inc}\ d(area) \qquad (5.5–11)$$

A waveguide matching condition acts only on the scattered part, and has the form,

$$\{\hat{n}\} \times \nabla \times \{E\} \approx P_{port}\ (\{E^{sc}\},\ \beta) \qquad (5.5–12)$$

where: P_{port} = a surface differential operator

β = the corresponding propagation constant

Equation (5.5–12) is then used to approximate the first surface integral on the right–hand side of equation (5.5–11) . Here $\{E^{in}\}$ is viewed as the incident waveguide wave.

Infinite Boundary Condition

Very much similar to the waveguide matching condition, an infinite boundary condition is defined as

$$\{\hat{n}\} \times \nabla \times \{E^{sc}\} \approx P_{inf}\ (\{E^{sc}\},\ \omega) \qquad (5.5–13)$$

where: P_{inf} = surface differential operator

ω = frequency angular velocity

Equation (5.5–13) is employed to approximate the first surface integral on the right–hand side of equation (5.5–11). Here $\{E^{in}\}$ is interpreted as the incident plane wave.

5.5.2 Time–Harmonic High–Frequency Evaluations

A time–harmonic solution of the full–wave formulations in section 5.5.1 yields a nodal degrees–of–freedom $\{A_X\}$ for all nodes in the solution domain. The complex values $\{A_X\}$, however, are not immediately transparent to the need of analysts. Therefore, in this subsection, the DOF solutions are transformed into familiar engineering terms.

Electric Field

The electric field E is computed at the element level by using the vector shape functions [W]:

$$\{E\} = [W]\ (\text{Re}\ \{A_x\} + j\ \text{Im}\ \{A_x\}) \qquad (5.5–14)$$

Magnetic Field

The magnetic field H is derived at the element level by using the curl of the vector shape functions:

$$\{H\} = \frac{1}{\omega} \nabla \times [W]^T \left(\text{Im}\ \{A_x\} - j\ \text{Re}\ \{A_x\}\right) \qquad (5.5\text{–}15)$$

Poynting Vector

The time–average Poynting vector (i.e., average power density) over one period is defined by

$$\{P\}_{av} = \frac{1}{2}\ \text{Re}\left[\{E\} \times \{H\}^*\right] \qquad (5.5\text{–}16)$$

Stored Energy

The time–average stored electric energy and magnetic energy are given by:

$$\overline{W}_e = \int_{vol} \frac{1}{4}\ \{H\} \cdot [\mu]\ \{H\}^*\ d(vol) \qquad (5.5\text{–}17)$$

$$\overline{W}_m = \int_{vol} \frac{1}{4}\ \{E\} \cdot [\epsilon]\ \{E\}^*\ d(vol) \qquad (5.5\text{–}18)$$

respectively.

Dielectric Loss:

For a lossy dielectric with a loss tangent, the incurred time–average volumetric power loss is:

$$P_d = \int_{vol} \frac{1}{2}\ \{E\} \cdot \sigma_d\ \{E\}^*\ d(vol) \qquad (5.5\text{–}19)$$

where the dielectric conductivity is given by

$$\sigma_d = \omega\ \epsilon_o\ \epsilon_r\ \tan\ \delta \qquad (5.5\text{–}20)$$

Surface Power Loss:

On an impedance surface, the incurred time–average surface loss is calculated using surface resistivity as:

$$P_s = \int_{\text{area}} \frac{1}{2} R_s \{H\} \cdot \{H\}^* \, d(\text{area}) \qquad (5.5\text{–}21)$$

Surface Power

The complex surface power flowing along $\{\hat{n}\}$ direction is given by:

$$P_e = \int_{\text{area}} \frac{1}{2} \{E\} \times \{H\}^* \cdot \{\hat{n}\} \, d(\text{area}) \qquad (5.5\text{–}22)$$

5.6 Inductance Computation

5.6.1 Differential Inductance Definition

Consider a magnetic excitation system consisting of n coils each fed by a current, I_i. The flux linkage ψ_i of the coils is defined as the surface integral of the flux density over the area multiplied by the number of turns, N_i, of the of the pertinent coil. The relationship between the flux linkage and currents can be described by the secant inductance matrix, $[L_s]$:

$$\{\psi\} = [L_s\,(t, \{I\})]\,\{I\} + \{\psi_o\} \qquad (5.6\text{–}1)$$

where: $\{\psi\}$ = vector of coil flux linkages

t = time

$\{I\}$ = vector of coil currents.

$\{\psi_o\}$ = vector of flux linkages for zero coil currents (effect of permanent magnets)

Main diagonal element terms of $[L_s]$ are called self inductance, whereas off diagonal terms are the mutual inductance coefficients. $[L_s]$ is symmetric which can be proved by the principle of energy conservation.

In general, the inductance coefficients depend on time, t, and on the currents. The time dependent case is called time variant which is characteristic when the coils move. The inductance computation used by the program is restricted to time invariant cases. Note that time variant problems may be reduced to a series of invariant analyses with fixed coil positions. The inductance coefficents depend on the currents when nonlinear magnetic material is present in the domain.

The voltage vector, {U}, of the coils can be expressed as:

$$\{U\} = \frac{\partial}{\partial t}\,\{\psi\} \qquad (5.6\text{–}2)$$

In the time invariant non–linear case

$$\{U\} = \left(\frac{d[L_s]}{d\{I\}}\,\{I\} + [L_s]\right)\frac{\partial}{\partial t}\,\{I\} = [L_d\,\{I\}]\,\frac{\partial}{\partial t}\,\{I\} \qquad (5.6\text{–}3)$$

The expression in the bracket is called the differential inductance matrix, $[L_d]$. The circuit behavior of a coil system is governed by $[L_d]$: the induced voltage is directly proportional to the differential inductance matrix and the time derivative of the coil currents. In general, $[L_d]$ depends on the currents, therefore it should be evaluated for each operating point.

5.6.2 Review of Inductance Computation Methods

After a magnetic field analysis, the secant inductance matrix coefficients, L_{sij}, of a coupled coil system could be calculated at postprocessing by computing flux linkage as the surface integral of the flux density, {B}. The differential inductance coefficients could be obtained by perturbing the operating currents with some current increments and calculating numerical derivatives. However, this method is cumbersome, neither accurate nor efficient. A much more convenient and efficient method is offered by the energy perturbation method developed by Demerdash and Arkadan(225), Demerdash and Nehl(226) and Nehl et al(227). The energy perturbation method is based on the following formula:

$$L_{dij} = \frac{d^2W}{dI_i,\ dI_j} \qquad (5.6–4)$$

where W is the magnetic energy, I_i and I_j are the currents of coils i and j. The first step of this procedure is to obtain an operating point solution for nominal current loads by a nonlinear analysis. In the second step linear analyses are carried out with properly perturbed current loads and a tangent reluctivity tensor, ν_t, evaluated at the operating point. For a self coefficient, two, for a mutual coefficient, four, incremental analyses are required. In the third step the magnetic energies are obtained from the incremental solutions and the coefficients are calculated according to (5.6–4).

5.6.3 Inductance Computation Method Used

The inductance computation method used by the program is based on Gyimesi and Ostergaard(229) who revived Smythe's procedure(150).

The incremental energy W_{ij} is defined by

$$W_{ij} = \frac{1}{2} \int \{\Delta H\}\ \{\Delta B\}\ dV \qquad (5.6–5)$$

where {ΔH} and {ΔB} denote the increase of magnetic field and flux density due to current increments, ΔI_i and ΔI_j. The coefficients can be obtained from

$$W_{ij} = \frac{1}{2} L_{dij}\ \Delta I_i\ \Delta I_j \qquad (5.6–6)$$

This allows an efficient method that has the following advantages:

1. For any coefficient, self or mutual, only one incremental analysis is required.
2. There is no need to evaluate the absolute magnetic energy. Instead, an "incremental energy" is calculated according to a simple expression.
3. The calculation of incremental analysis is more efficient: The factorized stiffness matrix can be applied. (No inversion is needed.) Only incremental load vectors should be evaluated.

5.7 Electromagnetic Particle Tracing

Once the electromagnetic field is computed, particle trajectories can be evaluated by solving the equations of motion:

$$m \{a\} = \{F\} = q (\{E\} + \{v\} \times \{B\}) \tag{5.7–1}$$

where:

m	=	mass
q	=	charge
{E}	=	electric field vector
{B}	=	magnetic field vector
{F}	=	Lorentz force vector
{a}	=	acceleration vector
{v}	=	velocity vector

The tracing follows from element to element: the exit point of an old element becomes the entry point of a new element. Given the entry location and velocity for an element, the exit location and velocity can be obtained by integrating the equations of motion.

ANSYS particle tracing algorithm is based on Gyimesi and Lavers(230) and Gyimesi et al(228) exploiting the following assumptions:

1. No relativistic effects (velocity is much smaller than speed of light).
2. Either electric tracing, {B} = 0, or magnetic tracing {E} = 0.
3. Electrostatic or magnetostatic analysis
4. Constant {E} or {B} within an element.
5. Quadrangle, triangle, hexahedron, tetrahedron, wedge or pyramid element shapes bounded by planar surfaces.

These simplifications significantly reduce the computation time of the tracing algorithm because the trajectory can be given in an analytic form:

1. parabola in the case of electric tracing
2. helix in the case of magnetic tracing.

The exit point from an element is the point where the particle trajectory meets the plane of bounding surface of the element. It can be easily computed when the trajectory is a parabola. However, to compute the exit point when the trajectory is a helix, a transcendental equation must be solved. A Newton Raphson algorithm is implemented to obtain the solution. The starting point is carefully selected to ensure convergence to the correct solution. This is far from obvious: about 70 sub–cases are differentiated by the algorithm Gyimesi and Lavers(230). This tool allows particle tracing within an element accurate up to machine precision. This does not mean that the tracing is exact since the element field solution may be inexact. However, with mesh refinement, this error can be controlled.

Once a trajectory is computed, any available physical items can be printed or plotted along the path. For example, elapsed time, traveled distance, particle velocity components, temperature, field components, potential values, fluid velocity, acoustic pressure, mechanical strain, etc. Animation is also available.

The plotted particle traces consist of two branches: the first is a trajectory for a given starting point at a given velocity (forward ballastics); the second is a trajectory for a particle to hit a given target location at a given velocity (backward ballistics).

5.8 Maxwell Stress Tensor

The Maxwell stress tensor provides a convenient way of computing forces acting on bodies by evaluating a surface integral. This section describes the theoretical foundation of ANSYS macro commands, **FMAGBC** and **FMAGSUM**.

Following Vago and Gyimesi(239), this section derives the Maxwell stress tensor from Maxwell's equations (equations (5.1–1) thru (5.1–4). The derivation requires involved mathematical operations. Section 5.8.1 summarizes the vector and tensor algebraic notations. The fundamental identities of vector and tensor analysis are given in Section 5.8.2. Using these identities, equation (5.8–15) is derived in Section 5.8.3. Section 5.8.4 derives the Maxwell stress tensor from Maxwell's equations using equation (5.8–15). The fundamental vector and tensor algebraic equations can be found in Flugge(240) and Legally(241).

5.8.1 Notation

This section summarizes the notations of vector and tensor algebraic notations used to derive the Maxwell stress tensor.

where:	a	=	scalar
	{A}	=	vector
	[A]	=	2nd order tensor
	$\{1_x\}, \{1_y\}, \{1_z\}$	=	unit vectors
	[1]	=	2nd order unit tensor
	{A} · {B}	=	dot product vectors resulting in a scalar
	{A} * {B}	=	cross product of vectors providing vector
	{A} @ {B}	=	dyadic product of vectors providing 2nd order tensor
	grad	=	gradient of a scalar providing vector
	curl	=	rotation of a vector providing scalar
	div	=	divergence of a vector providing scalar
	Grad	=	gradient of a vector providing tensor

$$\text{Grad}\left\{u\{1_x\} + v\{1_y\} + w\{1_z\}\right\} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix}$$

Div = divergence of a tensor providing vector

$$\text{Div} \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} \text{div} \left(a \, \{1_x\} + b \, \{1_y\} + c \, \{1_z\} \right) \\ \text{div} \left(d \, \{1_x\} + e \, \{1_y\} + f \, \{1_z\} \right) \\ \text{div} \left(g \, \{1_x\} + h \, \{1_y\} + i \, \{1_z\} \right) \end{bmatrix}$$

5.8.2 Fundamental Relations

This section provides the fundamental identities of vector and tensor analysis. See Vago & Gyimesi(239), Flugge(240), and Lagally(241).

$$\{A\} * \left\{\{B\} * \{C\}\right\} = \{B\} \left\{\{A\} \cdot \{C\}\right\} - \left\{\{A\} \cdot \{B\}\right\} \quad (5.8\text{–}1)$$

$$\text{div} \left(\left\{a \; \{B\}\right\}\right) = a \; \text{div} \; \{B\} + \{B\} \; \text{grad} \; a \quad (5.8\text{–}2)$$

$$\text{div} \left\{\{A\} * \{B\}\right\} = \{B\} \; \text{curl} \; \{A\} - \{A\} \; \text{curl} \; \{B\} \quad (5.8\text{–}3)$$

$$\text{curl} \left\{a \; \{B\}\right\} = a \; \text{curl} \; \{B\} + \text{grad} \; a * \{B\} \quad (5.8\text{–}4)$$

$$\begin{aligned} \text{grad} \; (\{A\} \cdot \{B\}) = {} & \left[\text{Grad} \; \{A\}\right] B + \left[\text{Grad} \; \{B\}\right] A + \\ & \{A\} * \text{curl} \; \{B\} + \{B\} * \text{curl} \; \{A\} \end{aligned} \quad (5.8\text{–}5)$$

$$\begin{aligned} \text{curl} \; \left\{\{A\} \cdot \{B\}\right\} = {} & \left[\text{Grad} \; \{A\}\right] \{B\} + \left[\text{Grad} \; \{B\}\right] \{A\} + \\ & \{A\} \; \text{div} \; \{B\} + \{B\} \; \text{div} \; \{A\} \end{aligned} \quad (5.8\text{–}6)$$

$$\text{Grad} \left\{a \; \{B\}\right\} = a \; \text{Grad} \; \{B\} + \{B\} \; @ \; \text{grad} \; a \quad (5.8\text{–}7)$$

$$\text{Div} \left[\{A\} \; @ \; \{B\}\right] = \{A\} \; \text{div} \; \{B\} + [\text{Grad A}] \; \{B\} \quad (5.8\text{–}8)$$

from equation (5.8–8):

$$\text{Div} \left[\{B\} \; @ \; \{A\}\right] = \{B\} \; \text{div} \; \{A\} + [\text{Grad B}] \; \{A\} \quad (5.8\text{–}9)$$

$$\text{Div} \left[a \; [B]\right] = a \; \text{Div} \; [B] + [B] \; \text{grad} \; a \quad (5.8\text{–}10$$

5.8.3 Derived Relations

This section proves equation (5.8–15) using the fundamental identities of vector and tensor analysis.

From equation (5.8–10):

$$\mathrm{Div}\left[(\{A\} \cdot \{B\})\,[1]\right] = [1]\ \mathrm{grad}\,(\{A\} \cdot \{B\}) = \mathrm{grad}\,(\{A\} \cdot \{B\}) \tag{5.8–11}$$

From equations (5.8–5), (5.8–8), and (5.8–9):

$$\begin{aligned}\mathrm{grad}\ (\{A\} \cdot \{B\}) = {} & \mathrm{Div}\left[\{A\} @ \{B\}\right] - \{A\}\ \mathrm{div}\ \{B\} + \\ & \mathrm{Div}\left[\{B\} @ \{A\}\right] - \{B\}\ \mathrm{div}\ \{A\} \\ & \{A\} * \mathrm{curl}\ \{B\} + \{B\} * \mathrm{curl}\ \{A\}\end{aligned} \tag{5.8–12}$$

From equations (5.8–11) and (5.8–12):

$$\begin{aligned}&\mathrm{Div}\left[\{A\} @ \{B\} + \{B\} @ \{A\} - \{A\} \cdot \{B\}\ [1]\right] = \\ &\quad \{A\}\ \mathrm{div}\ \{B\} + \{B\}\ \mathrm{div}\ \{A\} - \{A\} * \mathrm{curl}\ \{B\} * \mathrm{curl}\ \{A\}\end{aligned} \tag{5.8–13}$$

Substitute {B} = a {A} in equation (5.8–13) and apply equations (5.8–1), (5.8–2), and (5.8–4)

$$\begin{aligned}&\mathrm{Div}\left[2\{A\} @ \{B\} - \{A\} \cdot \{B\}\ [1]\right] = \\ &\{A\}\ \mathrm{div}\ \{B\} + \{B\}\ \mathrm{div}\left\{\frac{1}{a}\ \{B\}\right\} - \{A\} * \mathrm{curl}\ \left\{a\ \{A\}\right\} \cdot \{B\} * \mathrm{curl}\ \{A\} = \\ &\{A\}\ \mathrm{div}\ \{B\} + \{B\}\left(\frac{1}{a}\ \mathrm{div}\ \{B\} - \frac{1}{a^2}\ \{B\} \cdot \mathrm{grad}\ a\right) - \\ &\{A\} * \left\{a\ \mathrm{curl}\ \{A\} + \mathrm{grad}\ a * \{A\}\right\} - \{B\} * \mathrm{curl}\ \{A\} = \\ &2\ \{A\}\ \mathrm{div}\ \{B\} - \{A\}\ (\{A\} \cdot \mathrm{grad}\ a) - \\ &2\ \{B\} * \mathrm{curl}\ \{A\} - \{A\} * \left\{\mathrm{grad}\ a * \{A\}\right\} = \\ &2\ \{A\}\ \mathrm{div}\ \{B\} - \{A\}\ (\{A\} \cdot \mathrm{grad}\ a) - 2\ \{B\} * \mathrm{curl}\ \{A\} - \\ &\quad (\{A\} \cdot \{A\})\ \mathrm{grad}\ a + \{A\}\ (\{A\} \cdot \mathrm{grad}\ a) = \\ &2\ \{A\}\ \mathrm{div}\ \{B\} - 2\ \{B\} * \mathrm{curl}\ \{A\} - (\{A\} \cdot \{A\})\ \mathrm{grad}\ a\end{aligned} \tag{5.8–14}$$

After arrangement,

$$
\begin{aligned}
& a\ \{A\} * \text{curl}\ \{A\} \ = \ \{A\}\ \text{div}\ \left\{a\ \{A\}\right\} - \frac{(\{A\}\ \cdot\ \{A\})}{2}\ \text{grad}\ a\ - \\
& -\text{Div}\ \left[\{A\}\ @\ a\ \{A\}\right] - \frac{(\{A\}\ \cdot a\ \{A\})}{2}\ [1]
\end{aligned}
\tag{5.8–15}
$$

5.8.4 Maxwell Tensor From Maxwell's Equations

This section derives the Maxwell stress tensor from Maxwell's equations using equation (5.1–15). Equation (5.1–20) constitutes the fundamental relation of Maxwell stress tensor, and equation (5.1–22) shows its application to evaluate forces by a surface integral.

Maxwell's equations are described in Section 5.1. The definitions used there are the same as used here. Take the vector product of equation (5.1–1) by {B} and equation (5.1–2) by {D} and add up these equations, providing:

$$
\begin{aligned}
& \{D\} * \text{curl}\ \{E\}\ +\ \{B\} * \text{curl}\ \{H\}\ = \\
& -\ \{D\} * \left\{\frac{dB}{dt}\right\}\ +\ \{B\} * \left\{\{J\}\ +\ \left\{\frac{dD}{dt}\right\}\right\}\ = \\
& -\ \{J\} * \{B\} - \mu\ \upsilon\ d/dt\ \left\{\{E\} * \{H\}\right\}
\end{aligned}
\tag{5.8–16}
$$

In a homogeneous linear constitutive case, equation (5.1–5) and equation (5.1–12) simplify:

$$
\{B\}\ =\ \mu\ \{H\}
\tag{5.8–17}
$$

$$
\{D\}\ =\ \upsilon\ \{E\}
\tag{5.8–18}
$$

Combining (5.8–15) and equation (5.8–18) provides:

$$
\begin{aligned}
& +\ \{E\} * \text{div}\ \{D\} - \frac{(\{E\}\ \cdot\ \{E\})}{2}\ \text{grad}\ \upsilon - \text{Div}\ [T_e]\ + \\
& +\ \{H\} * \text{div}\ \{B\} - \frac{(\{H\}\ \cdot\ \{H\})}{2}\ \text{grad}\ \mu - \text{Div}\ [T_m]\ = \\
& -\{J\} * \{B\} - \mu\ \upsilon\ \left\{\frac{d}{dt}\ E * H\right\}
\end{aligned}
\tag{5.8–19}
$$

After arrangement

$$\text{Div}\ [[T_e]\ +\ [T_m]]\ =\ \{f\}\ =$$

$$\rho\ \{E\}\ +\ \{J\}\ *\ \{B\} - \frac{(\{E\}\ \cdot\ \{E\})}{2}\ \text{grad}\ \upsilon \qquad (5.8\text{–}20)$$

$$\frac{\{H\}\ \cdot\ \{H\}}{2}\ \text{grad}\ \mu\ +\ \mu\ \upsilon \left\{\frac{dS}{dt}\right\}$$

where: $[T_e]$ = $\{E\}\ @\ \{D\} - \frac{(\{E\}\ \cdot\ \{D\})}{2}\ [1]$ = electric Maxwell stress tensor

$[T_m]$ = $\{H\}\ @\ \{B\} - \frac{(\{H\}\ \cdot\ \{B\})}{2}\ [1]$ = magnetic Maxwell stress tensor

$\{f\}$ = force density

ρ = charge density

$\{S\}$ = Poynting vector

Equation (5.8–20) is the same as equations (1.255) and (1.257) in Vago and Gyimesi(239). The terms in equation (5.8–20) can be interpreted as:

$\rho\ \{E\}\ +\ \{J\}\ *\ \{B\}$ = Lorentz foce density

$\mu\ \upsilon \left\{\frac{dS}{dt}\right\}$ = light radiation pressure

grad terms = force due to material inhomogenities

The force over a volume V:

$$\{F\}\ =\ \int \{f\}\ dV\ =\ \int \text{Div}\ [[T_e]\ +\ [T_m]]\ dV \qquad (5.8\text{–}21)$$

Applying Gauss theorem:

$$\{F\}\ =\ \int [[T_e]\ +\ [T_m]]\ \cdot\ \{dS\} \qquad (5.8\text{–}22)$$

i.e., the force is the surface integral of Maxwell stress tensor.

5.9 Electromechanical Transducer for MEMS

See Section 14.126 of the Theory Reference Manual.

5.10 Capacitance Computation by CMATRIX Macro

Capacitance computation is one of the primary goals of an electrostatic analysis. For the definition of ground (partial) and lumped capacitance matrices see Vago and Gyimesi(239). The knowledge of capacitance is essential in the design of electrostatic devices, Micro Electro Mechanical Systems (MEMS), transmission lines, printed circuit boards (PCB), electomagnetic interference and compability (EMI/EMC) etc. The computed capacitance can be the input of a subsequent MEMS analysis by an electrostructural tranducer element TRANS126; for theory see section 14.126.

The capacitance matrix of an electrostatic system can be computed in ANSYS by the **CMATRIX** macro. The capacitance calculation is based on the energy principle. For details see Gyimesi and Ostergaard(249, 250) and its successful application Hieke(251). The energy principle constitutes the basis for inductance matrix computation, as shown in section 5.6.

The electrostatic energy of a linear three electrode (the third is ground) system is:

$$W = \frac{1}{2} C_{11} V_1^2 + \frac{1}{2} C_{22} V_2^2 + C_{12} V_1 V_2 \tag{5.10–1}$$

where: W = electrostatic energy

V_1 = potential of first electrode with respect to ground

V_2 = potential of second electrode with respect to ground

C_{11} = self ground capacitance of first electrode

C_{22} = self ground capacitance of second electrode

C_{12} = mutual ground capacitance between electrodes

By applying appropriate voltages on electrodes, the coefficients of the ground capacitance matrix can be calculated from the stored static energy.

The charges on the conductors are:

$$Q_1 = C_{11} V_1 + C_{12} V_2 \tag{5.10–2}$$

$$Q_2 = C_{11} V_1 + C_{22} V_2 \tag{5.10–3}$$

where: Q_1 = charge of first electrode

Q_2 = charge of second electrode

The charge can be expressed by potential differences, too:

$$Q_1 = G_{11} V_1 + G_{12} (V_1 - V_2) \qquad (5.10\text{–}4)$$

$$Q_2 = G_{22} V_2 + G_{12} (V_2 - V_1) \qquad (5.10\text{–}5)$$

where: G_{11} = self lumped capacitance of first electrode

G_{22} = self lumped capacitance of second electrode

G_{12} = mutual lumped capacitance between electrode

The lumped capacitances can be realized by lumped capacitors as shown in Figure 5.10–1. Lumped capacitances are suitable for use in circuit simulators.

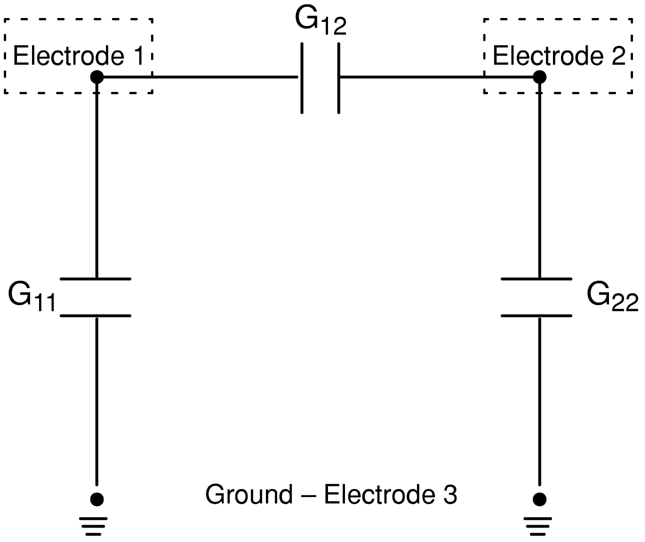


Figure 5.10–1 Lumped Capacitor Model of Two Conductors and Ground

In some cases, one of the electrodes may be located very far from the other electrodes. This can be modeled as an open electrode problem with one electrode at infinity. The open boundary region can be modeled by infinite elements, Trefftz method (see Section 5.11) or simply closing the FEM region far enough by an artificial Dirichlet boundary condition. In this case the ground key parameter should be activated in the **CMATRIX** command macro. This key assumes that there is a ground electrode at infinity.

The previous case should be distinguished from an open boundary problem without an electrode at infinity. In this case the ground electrode is one of the modeled electrodes. The FEM model size can be minimized in this case, too, by infinite elements or the Trefftz method. When calling the **CMATRIX** macro, however, the ground key should not be activated since there is no electrode at infinity.

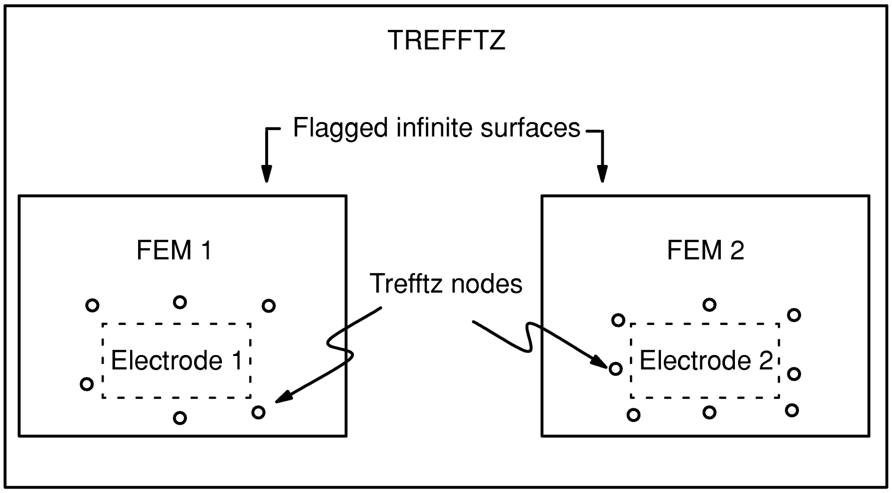


Figure 5.10–2 Trefftz and Multiple Finite Element Domains

The FEM region can be multiply connected. See for example Figure 5.10–2. The electrodes are far from each other: Meshing of the space between the electrodes would be computationally expensive and highly ineffective. Instead, a small region is meshed around each electrode and the rest of the region is modeled by the Trefftz method (see Section 5.11).

5.11 Open Boundary Analysis with a Trefftz Domain

The Trefftz method was introduced in 1926 by the founder of boundary element techniques, E. Trefftz(259, 260). The generation of Trefftz complete function systems was analyzed by Herrera(261). Zienkiewicz et al(262), Zielinski and Zienkiewicz(263), Zienkiewicz et al(264, 265, 266) exploited the energy property of the Trefftz method by introducing the Generalized Finite Element Method with the marriage a la mode: best of both worlds (finite and boundary elements) and successfully applied it to mechanical problems. Mayergoyz et al(267), Chari(268), and Chari and Bedrosian(269) successfully applied the Trefftz method with analytic Trefftz functions to electromagnetic problems. Gyimesi et al(255), Gyimesi and Lavers(256), and Gyimesi and Lavers(257) introduced the Trefftz method with multiple multipole Trefftz functions to electromagnetic and acoustic problems. This last approach successfully preserves the FEM–like positive definite matrix structure of the Trefftz stiffness matrix while making no restriction to the geometry (as opposed to analytic functions) and inheriting the excellent accuracy of multipole expansion.

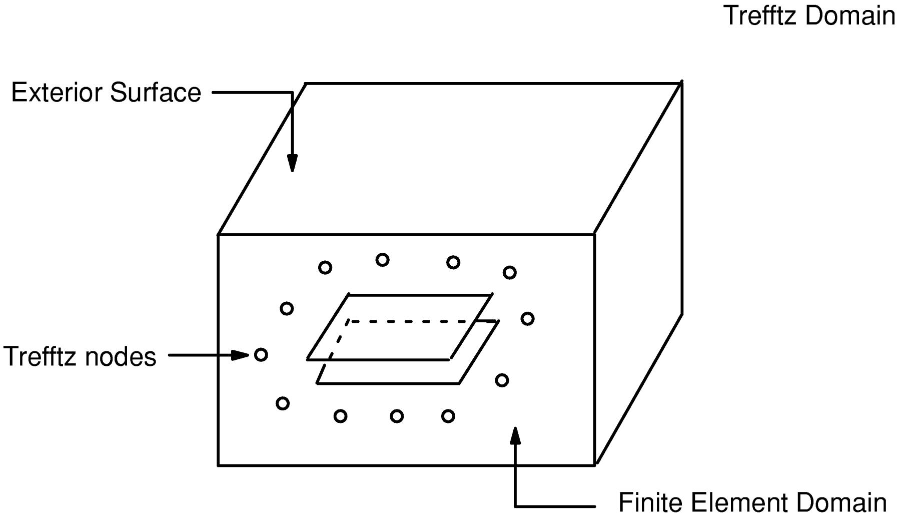


Figure 5.11–1 Typical Hybrid FEM–Trefftz Domain

Figure 5.11–1 shows a typical hybrid FEM–Trefftz domain. The FEM domain lies between the electrode and exterior surface. The Trefftz region lies outside the exterior surface. Within the finite element domain, Trefftz multiple multipole sources are placed to describe the electrostatic field in the Trefftz region according to Green's representation theorem. The FEM domain can be multiply connected as shown in Figure 5.11–2. There is minimal restriction regarding the geometry of the exterior surface. The FEM domain should be convex (ignoring void region interior to the model from conductors) and it should be far enough away so that a sufficiently thick cushion distributes the singularities at the electrodes and the Trefftz sources.

The energy of the total system is

$$W = \frac{1}{2}\{u\}^T[K]\{u\} + \frac{1}{2}\{w\}^T[L]\{w\} \tag{5.11–1}$$

where:
W = energy
{u} = vector of FEM DOFs
{w} = vector of Trefftz DOFs
[K] = FEM stiffness matric
[L] = Trefftz stiffness matrix

At the exterior surface, the potential continuity can be described by the following constraint equations:

$$[Q]\{u\} + [P]\{w\} = 0 \tag{5.11–2}$$

where:
[Q] = FEM side of constraint equations
[P] = Trefftz side of constraint equations

The continuity conditions are obtained by a Galerkin procedure. The conditional energy minimum can be found by the Lagrangian multiplier's method. This minimization process provides the (weak) satisfaction of the governing differential equations and continuity of the normal derivative (natural Neumann boundary condition.)

To treat the Trefftz region, the **TZEGEN** command macro creates a superelement and coupling constraint equations. The user needs to define only the Trefftz nodes (**TZAMESH** command macro).

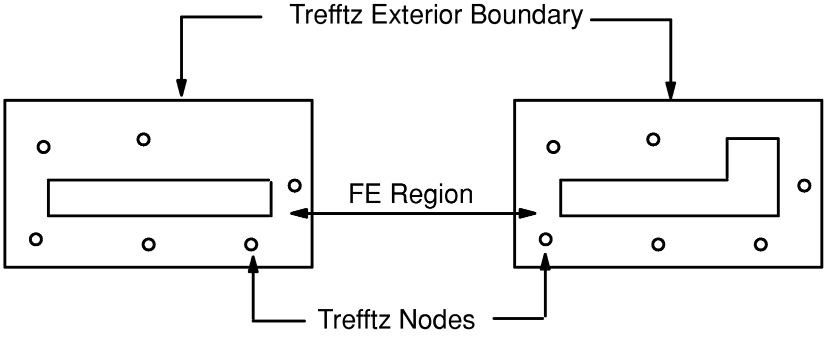


Figure 5.11–2 Multiple FE Domains Connected by One Trefftz Domain

Chapter 6
Heat Flow

ANSYS Theory Reference
Chapter 6 – Table of Contents

6.1 Heat Flow Fundamentals

6.1.1 Conduction and Convection

The first law of thermodynamics states that thermal energy is conserved. Specializing this to a differential control volume:

$$\rho c \left(\frac{\partial T}{\partial t} + \{v\}^T\{L\}T\right) + \{L\}^T\{q\} = \dddot{q} \qquad (6.1\text{–}1)$$

where:

ϱ = density (input as DENS on **MP** command)

c = specific heat (input as C on **MP** command)

T = temperature (=T(x,y,z,t))

t = time

$\{L\} = \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix}$ = vector operator

$\{v\} = \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix}$ = velocity vector for mass transport of heat (input as VX, VY, VZ on **R** command, PLANE55 and SOLID70 only).

{q} = heat flux vector (output quantities TFX, TFY, and TFZ)

$\dddot{q}$ = heat generation rate per unit volume (input on **BF** or **BFE** commands)

It should be realized that the terms {L}T and $\{L\}^T\{q\}$ may also be interpreted as ∇ T and ∇ · {q}, respectively, where ∇ represents the grad operator and ∇ · represents the divergence operator.

Next, Fourier's law is used to relate the heat flux vector to the thermal gradients:

$$\{q\} = -[D]\{L\}T \qquad (6.1\text{–}2)$$

where:

$[D] = \begin{bmatrix} K_{xx} & 0 & 0 \\ 0 & K_{yy} & 0 \\ 0 & 0 & K_{zz} \end{bmatrix}$ = conductivity matrix

K_{xx}, K_{yy}, K_{zz} = conductivity in the element x, y, and z directions, respectively (input as KXX, KYY, KZZ on **MP** command)

Combining equations (6.1–1) and (6.1–2),

$$\rho c \left(\frac{\partial T}{\partial t} + \{v\}^T\{L\}T\right) = \{L\}^T([D]\{L\}T) + \dddot{q} \tag{6.1–3}$$

Expanding equation (6.1–3) to its more familiar form:

$$\begin{aligned} &\rho c \left(\frac{\partial T}{\partial t} + v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z}\right) = \\ &\dddot{q} + \frac{\partial}{\partial x}\left(K_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial T}{\partial z}\right) \end{aligned} \tag{6.1–4}$$

It will be assumed that all effects are in the global Cartesian system.

Three types of boundary conditions are considered. It is presumed that these cover the entire element.

1. Specified temperatures acting over surface S_1:

$$T = T^* \tag{6.1–5}$$

where T^* is the specified temperature (input on **D** command).

2. Specified heat flows acting over surface S_2:

$$\{q\}^T\{\eta\} = -q^* \tag{6.1–6}$$

where: $\{\eta\}$ = unit outward normal vector
q^* = specified heat flow (input on **SF** or **SFE** commands)

3. Specified convection surfaces acting over surface S_3 (Newton's law of cooling):

$$\{q\}^T\{\eta\} = h_f\left(T_S - T_B\right) \tag{6.1–7}$$

where: h_f = film coefficient (input on **SF** or **SFE** commands) Evaluated at $(T_B + T_S)/2$ unless otherwise specified for the element
T_B = bulk temperature of the adjacent fluid (input on **SF** or **SFE** commands)
T_S = temperature at the surface of the model

Note that positive specified heat flow is into the boundary (i.e., in the direction opposite of {η}), which accounts for the negative signs in equations (6.1–6) and (6.1–7).

Combining equation (6.1–2) with equations (6.1–6) and (6.1–7)

$$\{\eta\}^T\ [D]\{L\}T = q^* \tag{6.1–8}$$

$$\{\eta\}^T\ [D]\{L\}T = h_f\left(T_B - T\right) \tag{6.1–9}$$

Premultiplying equation (6.1–3) by a virtual change in temperature, integrating over the volume of the element, and combining with equations (6.1–8) and (6.1–9) with some manipulation yields:

$$\int_{vol}\left(\rho c\ \delta T\left(\frac{\partial T}{\partial t} + \{v\}^T\{L\}T\right) + \{L\}^T\left(\delta T\right)\left([D]\{L\}T\right)\right) d(vol) = \tag{6.1–10}$$

$$\int_{S_2}\delta T\ q^*\ d(S_2) + \int_{S_3}\delta T\ h_f\ \left(T_B - T\right) d(S_3) + \int_{vol}\delta T\ \ddot{q}\ d(vol)$$

where:

vol = volume of the element

δT = an allowable virtual temperature (=δT(x,y,z,t))

6.1.2 Radiation

Radiant energy exchange between neighboring surfaces of a region or between a region and its surroundings can produce large effects in the overall heat transfer problem. Though the radiation effects generally enter the heat transfer problem only through the boundary conditions, the coupling is especially strong due to nonlinear dependence of radiation on surface temperature.

Extending the Stefan–Boltzmann Law for a system of N enclosures, the energy balance for each surface in the enclosure for a gray diffuse body is given by Siegal and Howell(88) (equation 8–19), which relates the energy losses to the surface temperatures:

$$\sum_{i=1}^{N}\left(\frac{\delta_{ji}}{\epsilon_i} - F_{ji}\ \frac{1-\epsilon_i}{\epsilon_i}\right)\frac{1}{A_i}\ Q_i = \sum_{i=1}^{N}\left(\delta_{ji} - F_{ji}\right)\ \sigma\ T_i^4 \tag{6.1–11}$$

where:

N = number of radiating surfaces

δ_{ij} = Kronecker delta

ε_i = effective emissivity (**EMIS** or **MP** command) of surface i

F_{ij} = radiation view factors (see below)

A_i = area of surface i

Q_i = energy loss of surface i

σ = Stefan–Boltzmann constant (**STEF** or **R** command)

T_i = absolute temperature of surface i

For a system of two surfaces radiating to each other, equation (6.1–11) can be simplified to give the heat transfer rate between surfaces i and j as:

$$Q_i = \sigma \; \epsilon_i \; F_{ij} \; A_i \; \left(T_i^4 - T_j^4\right) \tag{6.1–12}$$

where: T_i, T_j = absolute temperature at surface i and j, respectively

View Factors

The view factor, F_{ij}, is defined as the fraction of total radiant energy that leaves surface i which arrives directly on surface j, as shown in Figure 6.1–1. It can be expressed by the following equation:

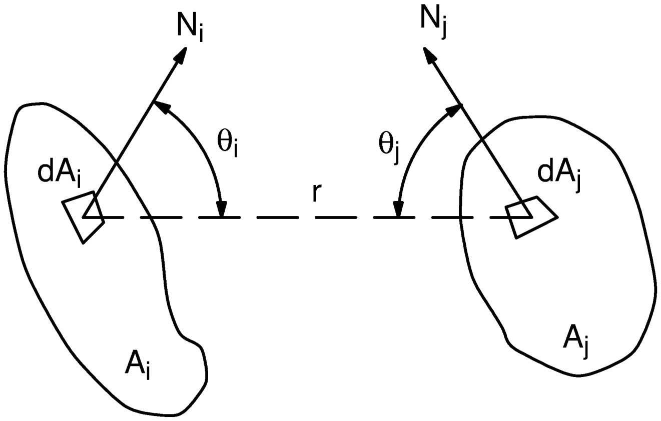


Figure 6.1–1 View Factor Calculation Terms

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\;\theta_i\;\cos\theta_j}{\pi r^2} \; d\!\left(A_j\right) \; d\!\left(A_i\right) \tag{6.1–13}$$

where: A_i,A_j = area of surface i and surface j

r = distance between differential surfaces i and j

θ_i = angle between N_i and the radius line to surface $d(A_j)$

θ_j = angle between N_j and the radius line to surface $d(A_i)$

N_i,N_j = surface normal of $d(A_i)$ and $d(A_j)$

Radiation Usage

Four methods for analysis of radiation problems are included:

1. Radiation link element LINK31 (Section 14.31). For simple problems involving radiation between two points or several pairs of points. The effective radiating surface area, the form factor and emissivity can be specified as real constants for each radiating point.
2. Surface effect elements – SURF151 in 2–D and SURF152 in 3–D for radiating between a surface and a point (Sections 14.151 and 14.152). The form factor between a surface and the point can be specified as a real constant or can be calculated from the basic element orientation and the extra node location.
3. Radiation matrix method (Section 6.4). For more generalized radiation problems involving two or more surfaces. The method involves generating a matrix of view factors between radiating surfaces and using the matrix as a superelement in the thermal analysis.
4. Radiosity solver method (Section 6.5). For generalized problems in 3–D involving two or more surfaces. The method involves calculating the view factor for the flagged radiating surfaces using the hemicube method and then solving the radiosity matrix coupled with the conduction problem.

6.2 Derivation of Heat Flow Matrices

As stated before, the variable T was allowed to vary in both space and time. This dependency is separated as:

$$T = \{N\}^T\{T_e\} \tag{6.2–1}$$

where:
T = $T(x,y,z,t)$ = temperature
$\{N\}$ = $\{N(x,y,z)\}$ = element shape functions
$\{T_e\}$ = $\{T_e(t)\}$ = nodal temperature vector

Thus, the time derivatives of equation (6.2–1) may be written as:

$$\dot{T} = \frac{\partial T}{\partial t} = \{N\}^T\{\dot{T}_e\} \tag{6.2–2}$$

δT has the same form as T:

$$\delta T = \{\delta T_e\}^T\{N\} \tag{6.2–3}$$

The combination {L}T is written as

$$\{L\}T = [B]\{T_e\} \tag{6.2–4}$$

where: $[B] = \{L\}\{N\}^T$

Now, the variational statement of equation (6.1–10) can be combined with equations (6.2–1) thru (6.2–4) to yield:

$$\begin{aligned}&\int_{vol} \rho c\ \{\delta T_e\}^T\{N\}\{N\}^T\ \{\dot{T}_e\}\ d(vol) + \int_{vol} \rho c\ \{\delta T_e\}^T\{N\}\{v\}^T[B]\{T_e\}\ d(vol)\\ &+ \int_{vol} \{\delta T_e\}^T[B]^T[D][B]\{T_e\}\ d(vol) = \int_{S_2} \{\delta T_e\}^T\{N\}\ q^*\, d(S_2)\\ &+ \int_{S_3} \{\delta T_e\}^T\{N\}\ h_f\left(T_B - \{N\}^T\{T_e\}\right)\ d(S_3) + \int_{vol} \{\delta T_e\}^T\{N\}\ \ddot{q}\ d(vol)\end{aligned} \tag{6.2–5}$$

Terms are defined in Section 6.1. ϱ is assumed to remain constant over the volume of the element. On the other hand, c and $\ddot{q}$ may vary over the element. Finally, $\{T_e\}$,

$\{\dot{T}_e\}$, and $\{\delta T_e\}$ are nodal quantities and do not vary over the element, so that they also may be removed from the integral. Now, since all quantities are seen to be premultiplied by the arbitrary vector $\{\delta T_e\}$, this term may be dropped from the resulting equation. Thus, equation (6.2–5) may be reduced to:

$$\rho\int_{vol} c\ \{N\}\ \{N\}^T\ d(vol)\ \left\{\dot{T}_e\right\} + \rho\int_{vol} c\ \{N\}\ \{v\}^T\ [B]\ d(vol)\ \{T_e\}$$

$$+\int_{vol} [B]^T[D][B]\ d(vol)\ \{T_e\} \ = \int_{S_2} \{N\}\ q^*\, d(S_2)\ + \tag{6.2–6}$$

$$\int_{S_3} T_B h_f\ \{N\}\ d(S_3) - \int_{S_3} h_f\{N\}\{N\}^T\{T_e\}\ d\left(S_3\right) + \int_{vol} \dddot{q}\{N\}\ d(vol)$$

Equation (6.2–6) may be rewritten as:

$$\left[C_e^t\right]\ \left\{\dot{T}_e\right\} + \left(\left[K_e^{tm}\right] + \left[K_e^{tb}\right] + \left[K_e^{tc}\right]\right)\{T_e\} \ = \{Q_e\} \ + \{Q_e^c\} \ + \left\{Q_e^g\right\} \tag{6.2–7}$$

where:

$$\left[C_e^t\right] = \rho\int_{vol} c\{N\}\{N\}^T\, d(vol) = \text{element specific heat (thermal damping) matrix}$$

$$\left[K_e^{tm}\right] = \rho\int_{vol} c\{N\}\{v\}^T\, [B]\ d(vol) = \text{element mass transport conductivity matrix}$$

$$\left[K_e^{tb}\right] = \int_{vol} [B]^T\ [D][B]\ d(vol) = \text{element diffusion conductivity matrix}$$

$$\left[K_e^{tc}\right] = \int_{S_3} h_f\ \{N\}\{N\}^T\, d(S_3) = \text{element convection surface conductivity matrix}$$

$$\left\{Q_e^f\right\} = \int_{S_2} \{N\}q^*\, d(S_2) = \text{element mass flux vector}$$

$$\{Q_e^c\} = \int_{S_3} T_B h_f\{N\}\ d(S_3) = \text{element convection surface heat flow vector}$$

$$\left\{Q_e^g\right\} = \int_{vol} \dddot{q}\ \{N\}\ d(vol) = \text{element heat generation load}$$

Comments on and modifications of the above definitions:

1. $\left[K_e^{tm}\right]$ is not symmetric.

2. $\left[K_e^{tc}\right]$ is calculated as defined above, for SOLID90 only. All other elements use a diagonal matrix, with the diagonal terms defined by the vector $\int_{S_3} h_f$ {N} d(S3).

3. $\left[C_e^t\right]$ is frequently diagonalized, as described in Section 13.2.

4. If $\left[C_e^t\right]$ exists and has been diagonalized as well as *Key* = ON on the **TIMINT** command, $\left\{Q_e^g\right\}$ has its terms adjusted so that they are proportioned to the main diagonal terms of $\left[C_e^t\right]$. $\left\{Q_e^j\right\}$, the heat generation rate vector for Joule heating is treated similarly, if present. This adjustment ensures that elements subjected to uniform heating will have a uniform temperature rise. However, this adjustment also changes nonuniform input of heat generation to an average value over the element.

5. For phase change problems, $\left[C_e^t\right]$ is evaluated from the enthalpy curve(42) if enthalpy is input (*Lab* = ENTH, **MP** command). This option should be used for phase change problems.

6.3 Heat Flow Evaluations

6.3.1 Integration Point Output

The element thermal gradients at the integration points are:

$$\{a\} = \{L\}T = \left\lfloor \frac{\partial T}{\partial x} \; \frac{\partial T}{\partial y} \; \frac{\partial T}{\partial z} \right\rfloor^T \qquad (6.3\text{–}1)$$

where:
{a} = thermal gradient vector (output quantity TG)
{L} = vector operator
T = temperature

Using shape functions, equation (6.3–1) may be written as:

$$\{a\} = [B]\ \{T\} \qquad (6.3\text{–}2)$$

where:
[B] = shape function derivative matrix evaluated at the integration points
{T} = nodal temperature vector

Then, the heat flux vector at the integration points may be computed from the thermal gradients:

$$\{q\} = -\ [D]\{a\} = -\ [D][B]\{T\} \qquad (6.3\text{–}3)$$

where:
{q} = heat flux vector (output quantity TF)
[D] = conductivity matrix (see equation (6.1–2))

Nodal gradient and flux vectors may be computed from the integration point values as described in Section 13.6.

6.3.2 Surface Output

The convection surface output is:

$$q^c = h_f \left(T_S - T_B\right) \qquad (6.3\text{–}4)$$

where:
q^c = heat flow per unit area due to convection
h_f = film coefficient (input on **SF** or **SFE** commands)

T_S = temperature at surface of model

T_B = bulk temperature of the adjacent fluid (input on **SF** or **SFE** commands)

6.4 Radiation Matrix Method

In the radiation matrix method, for a system of two radiating surfaces, equation (6.1–12) can be expanded as:

$$Q_i = \sigma\ \epsilon_i\ F_{ij}\ A_i\ \left(T_i^2 + T_j^2\right)\left(T_i + T_j\right)\left(T_i - T_j\right) \tag{6.4–1}$$

or

$$Q_i = K'\left(T_i - T_j\right) \tag{6.4–2}$$

where: $K' = \sigma\,\varepsilon_i\,F_{ij}\,A_i\left(T_i^2 + T_j^2\right)\left(T_i + T_j\right)$

K' cannot be calculated directly since it is a function of the unknowns T_i and T_j. The temperatures from previous iterations are used to calculate K' and the solution is computed iteratively.

For a more general case, equation (6.1–11) can be used to construct a single row in the following matrix equation:

$$[C]\{Q\} = [D]\left\{T^4\right\} \tag{6.4–3}$$

such that:

$$\text{each row j in } [C] = \left(\frac{\delta_{ji}}{\epsilon_i} - F_{ji}\,\frac{1-\epsilon_i}{\epsilon_i}\right)\frac{1}{A_i},\quad i = 1, 2\ ...N \tag{6.4–4}$$

$$\text{each row j in } [D] = \left(\delta_{ji} - F_{ji}\right)\sigma\qquad ,\quad i = 1, 2\ ...N \tag{6.4–5}$$

Solving for {Q}:

$$\{Q\} = \left[K^{ts}\right]\left\{T^4\right\} \tag{6.4–6}$$

and therefore:

$$\left[K^{ts}\right] = [C]^{-1}\ [D] \tag{6.4–7}$$

Equation (6.4–6) is analogous to equation (6.1–12) and can be set up for standard matrix equation solution by the process similar to the steps shown in equations (6.4–1) and (6.4–2).

$$\{Q\} = \left[K'\right]\{T\} \tag{6.4–8}$$

[K′] now includes T^3 terms and is calculated in the same manner as in equation (6.4–2). To be able to include radiation effects in elements other than LINK31, MATRIX50 (the substructure element) is used to bring in the radiation matrix. MATRIX50 has an option that instructs the solution phase to calculate [K′]. The AUX12 utility is used to create the substructure radiation matrix. AUX12 calculates the effective conductivity matrix, $[K^{ts}]$, in equation (6.4–6), as well as the view factors required for finding $[K^{ts}]$. The user defines flat surfaces to be used in AUX12 by overlaying nodes and elements on the radiating edge of a 2–D model or the radiating face of a 3–D model.

Two methods are available in the radiation matrix method to calculate the view factors (**VTYPE** command), the non–hidden method and the hidden method.

6.4.1 Non–hidden Method

The non–hidden procedure calculates a view factor for every surface to every other surface whether the view is blocked by an element or not. In this procedure, the following equation is used and the integration is performed adaptively.

For a finite element discretized model, equation (6.1–13) for the view factor F_{ij} between two surfaces i and j can be written as:

$$F_{ij} = \frac{1}{A_i} \sum_{p=1}^{m} \sum_{q=1}^{n} \left(\frac{\cos\theta_{ip} \, \cos\theta_{jq}}{\pi r^2} \right) A_{ip} \, A_{jq} \tag{6.4–9}$$

where: m = number of integration points on surface i

n = number of integration points on surface j

When the dimensionless distance between two viewing surfaces D, defined in equation (6.4–10), is less than 0.1, the accuracy of computed view factors is known to be poor (Siegal and Howell(88)).

$$D = \frac{d_{min}}{\sqrt{A_{max}}} \tag{6.4–10}$$

where: d_{min} = minimum distance between the viewing surfaces A1 and A2

A_{max} = max (A1, A2)

So, the order of surface integration is adaptively increased from order one to higher orders as the value of D falls below 8. The area integration is changed to contour integration when D becomes less than 0.5 to maintain the accuracy. The contour integration order is adaptively increased as D approaches zero.

6.4.2 Hidden Method

The hidden procedure is a simplified method which uses equation (6.1–13) and assumes that all the variables are constant, so that the equation becomes:

$$F_{ij} = \frac{A_j}{\pi r^2} \cos\theta_i \cos\theta_j \qquad (6.4\text{–}11)$$

The hidden procedure numerically calculates the view factor in the following conceptual manner. The hidden–line algorithm is first used to determine which surfaces are visible to every other surface. Then, each radiating, or "viewing," surface (i) is enclosed with a hemisphere of unit radius. This hemisphere is oriented in a local coordinate system (x′y′z′), whose center is at the centroid of the surface with the z axis normal to the surface, the x axis is from node I to node J, and the y axis orthogonal to the other axes. The receiving, or "viewed," surface (j) is projected onto the hemisphere exactly as it would appear to an observer on surface i.

As shown in Figure 6.4–1, the projected area is defined by first extending a line from the center of the hemisphere to each node defining the surface or element. That node is then projected to the point where the line intersects the hemisphere and transformed into the local system x′ y′ z′, as described in Kreyszig(23)

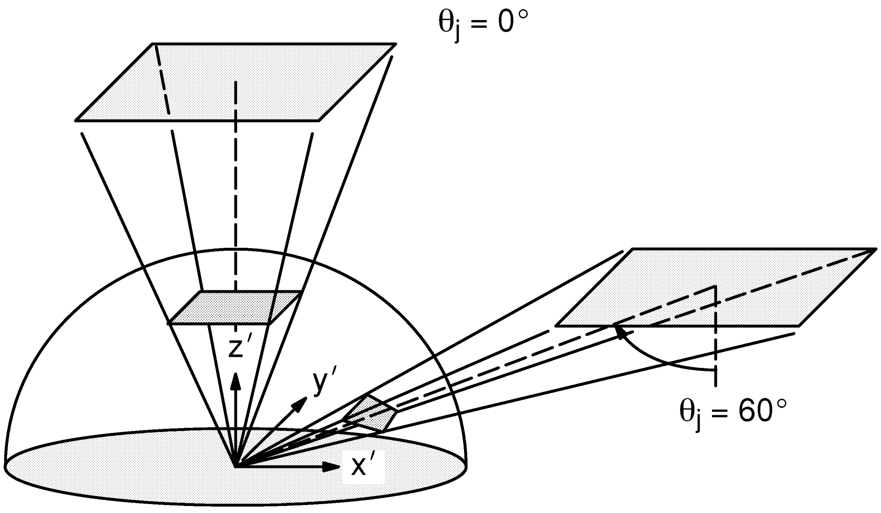


Figure 6.4–1 Receiving Surface Projection

The view factor, F_{ij}, is determined by counting the number of rays striking the projected surface j and dividing by the total number of rays (N_r) emitted by surface i. This method may violate the radiation reciprocity rule, that is, $A_i F_{i-j} \neq A_j F_{j-i}$.

6.4.3 View Factors of Axisymmetric Bodies:

When the radiation view factors between the surfaces of axisymmetric bodies are computed (GEOM,1,n), special logic is used. In this logic, the axisymmetric nature of the body is exploited to reduce the amount of computations. The user, therefore, needs only to build a model in plane 2–D representing the axisymmetric bodies as line “elements”.

Consider two axisymmetric bodies A and B as shown in Figure 6.4–2.

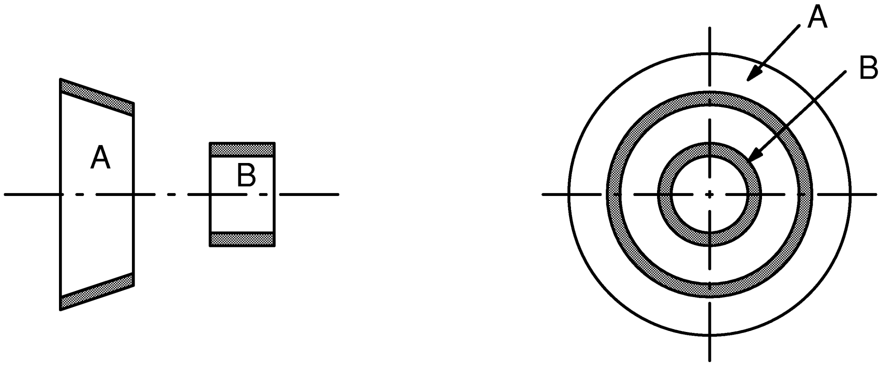


Figure 6.4–2 Axisymmetric Geometry

The view factor of body A to body B is computed by expanding the line “element” model into a full 3–D model of n circumferential segments (**GEOM**,1,n) as shown in Figure 6.4–3.

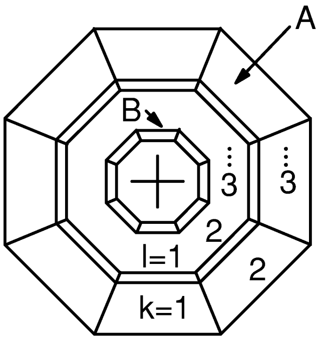


Figure 6.4–3 End View of Figure 6.4–2 Showing n = 8 Segments

View factor of body A to B is given by

$$F = \sum_{k=1}^{n} \sum_{\ell=1}^{n} F_{k-\ell} \tag{6.4–12}$$

where: $F_{k-\ell}$ = view factor of segment k on body A to segment ℓ on body B

The form factors between the segments of the axisymmetric bodies are computed using the method described in the previous section. Since the coefficients are symmetric, the summation equation (6.4–12) may be simplified as:

$$F = n \sum_{\ell = 1}^{n} F_{1 - \ell} \qquad (6.4–13)$$

Both hidden and non–hidden methods are applicable in the computation of axisymmetric view factors. However, the non–hidden method should be used if and only if there are no blocking surfaces. For example, if radiation between concentric cylinders are considered, the outer cylinder can not see part of itself without obstruction from the inner cylinder. For this case, the hidden method must be used, as the non–hidden method would definitely give rise to inaccurate view factor calculations.

6.4.4 Space Node

A space node may be defined (**SPACE** command) to absorb all energy not radiated to other elements. Any radiant energy not incident on any other part of the model will be directed to the space node. If the model is not a closed system, then the user must define a space node with its appropriate boundary conditions.

6.5 Radiosity Solution Method

In the radiosity solution method for the analysis of gray diffuse radiation between N surfaces, equation (6.1–11) is solved in conjunction with the basic conduction problem.

For the purpose of computation it is convenient to rearrange equation (6.1–11) into the following series of equations

$$\sum_{j=1}^{N} \left[\delta_{ij} - (1 - \epsilon_i)\, F_{ij}\right] q_j^o = \epsilon_i\, \sigma\, T_i^4 \qquad (6.5–1)$$

and

$$q_i = q_i^o - \sum_{j=1}^{N} F_{ij}\, q_j^o \qquad (6.5–2)$$

Equations (6.5–1) and (6.5–2) are expressed in terms of the outgoing radiative fluxes (radiosity) for each surface, q_j^o, and the net flux from each surface q_i. For known surface temperatures, T_i, in the enclosure, equation (6.5–1) forms a set of linear algebraic equations for the unknown, outgoing radiative flux (radiosity) at each surface. Equation (6.5–1) can be written as

$$[A]\ \{q^o\} = \{D\} \qquad (6.5–3)$$

where: $A_{ij} = \delta_{ij} - (1 - \varepsilon_i)\, F_{ij}$

q_j^o = radiosity flux for surface j

$D_i = \varepsilon_i\, \sigma\, T_i^4$

[A] is a full matrix due to the surface to surface coupling represented by the view factors and is a function of temperature due to the possible dependence of surface emissivities on temperature. Equation (6.5–3) is solved using a Newton–Raphson procedure for the radiosity flux $\{q^o\}$.

When the q^o values are available, equation (6.5–2) then allows the net flux at each surface to be evaluated. The net flux calculated during each iteration cycle is under–relaxed, before being updated using

$$q_i^{net} = \phi\ q_i^{k+1} + (1 - \phi)\ q_i^k \qquad (6.5\text{–}4)$$

where: ϕ = radiosity flux relaxation factor
k = iteration number

The net surface fluxes provide boundary conditions to the finite element model for the conduction process. The radiosity equation (6.5–3) is solved coupled with the conduction equation (6.2–6) using a segregated solution procedure until convergence of the radiosity flux and temperature for each time step or load step.

The surface temperatures used in the above computation must be uniform over each surface in order to satisfy conditions of the radiation model. In the finite element model, each surface in the radiation problem corresponds to a face or edge of a finite element. The uniform surface temperatures needed for use in equation (6.5–3) are obtained by averaging the nodal point temperatures on the appropriate element face.

For open enclosure problems using the radiosity method, an ambient temperature needs to be specified using a space temperature (**SPCTEMP** command) or a space node (**SPCNODE** command), to account for energy balance between the radiating surfaces and the ambient.

6.5.1 View Factor Calculation – Hemicube Method

For solution of radiation problems in 3D, the radiosity method calculates the view factors using the hemicube method as compared to the traditional double area integration method for three–dimensional geometry. Details using the Hemicube method for view factor calculation are given in Glass(272) and Cohen and Greenberg(276).

The hemicube method is based upon Nusselt's hemisphere analogy. Nusselt's analogy shows that any surface, which covers the same area on the hemisphere, has the same view factor. From this it is evident that any intermediate surface geometry can be used without changing the value of the view factors. In the hemicube method, instead of projecting onto a sphere, an imaginary cube is constructed around the center of the receiving patch. A patch in a finite element model corresponds to an element face of a radiating surface in an enclosure. The environment is transformed to set the center of the patch at the origin with the normal to the patch coinciding with the positive Z axis. In this orientation, the imaginary cube is the upper half of the surface of a cube, the lower half being below the 'horizon' of the patch. One full face is facing in the Z direction and four half faces are facing in the +X, –X, +Y, and –Y directions. These faces are divided into square 'pixels' at a given resolution, and the environment is then projected onto the five planar surfaces. Figure 6.5–1 shows the hemicube discretized over a receiving patch from the environment.

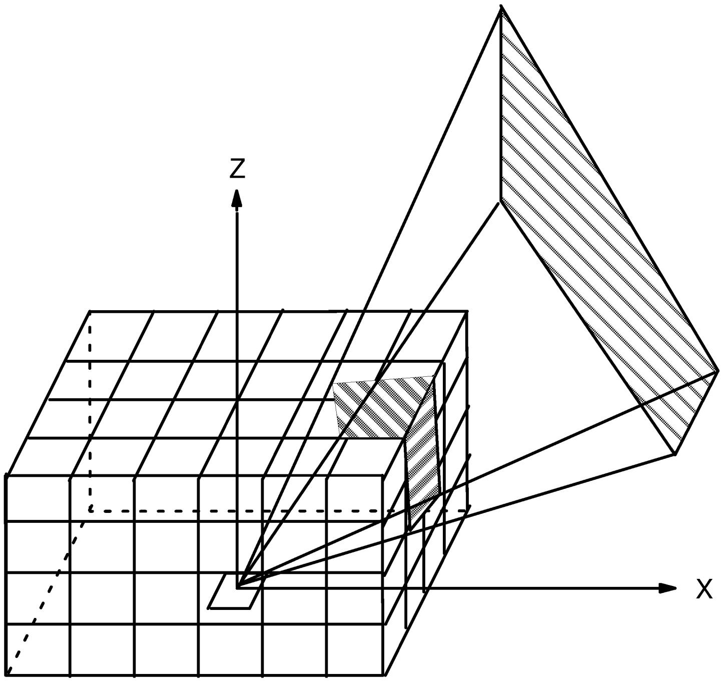


Figure 6.5–1   The Hemicube

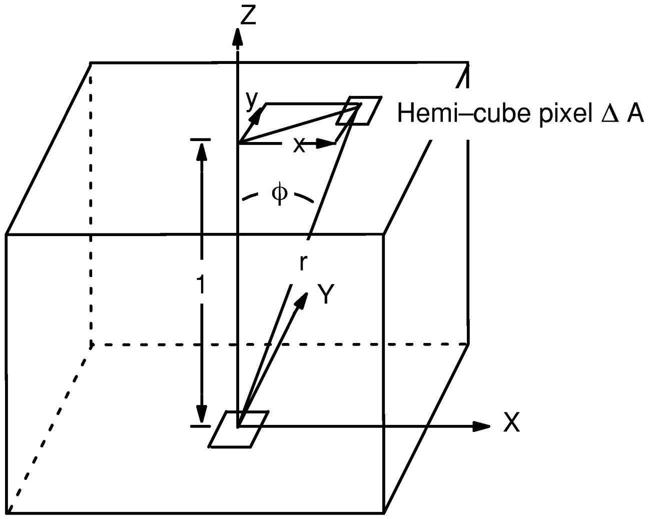


Figure 6.5–2   Derivation of Delta–View Factors for Hemicube Method

The contribution of each pixel on the cube's surface to the form–factor value varies and is dependent on the pixel location and orientation as shown in Figure 6.5–2. A specific delta form–factor value for each pixel on the cube is found from modified form of equation (6.1–13) for the differential area to differential area form–factor. If two patches project on the same pixel on the cube, a depth determination is made as to which patch is seen in that particular direction by comparing distances to each patch and selecting the nearer one. After determining which patch (j) is visible at each pixel on the hemicube, a summation of the delta form–factors for each pixel occupied by patch (j) determines the form–factor from patch (i) at the center of the cube to patch (j). This summation is performed for each patch (j) and a complete row of N form–factors is found.

At this point the hemicube is positioned around the center of another patch and the process is repeated for each patch in the environment. The result is a complete set of form–factors for complex environments containing occluded surfaces. The overall view factor for each surface on the hemicube is given by:

$$F_{ij} = \sum_{n=1}^{N} \Delta F_n = \frac{\cos \phi_i \;\; \cos \phi_j}{\pi r^2} \Delta A_j \tag{6.5–5}$$

where: N = number of pixels

ΔF = delta–view factor for each pixel

The resolution factor for the hemicube calculation can be issued using the **HEMIOPT** command. The hemicube resolution determines the accuracy of the view factor calculation and the speed at which they are calculated using the hemicube method. Default is set to 10. Higher values increase accuracy of the view factor calculation.

Chapter 7
Fluid Flow

ANSYS Theory Reference
Chapter 7 – Table of Contents

7.1 Fluid Flow Fundamentals

This chapter discusses the FLOTRAN solution method used with elements FLUID141 and FLUID142. These elements are used for the calculation of 2–D and 3–D velocity and pressure distributions in a single phase, Newtonian fluid. Thermal effects, if present, can be modeled as well.

The fluid flow problem is defined by the laws of conservation of mass, momentum, and energy. These laws are expressed in terms of partial differential equations which are discretized with a finite element based technique.

Assumptions about the fluid and the analysis are as follows:

1. The fluid is Newtonian.
2. There is only one phase.
3. The problem domain does not change.
4. The user must determine: (a) if the problem is laminar (default) or turbulent (**FLDATA**,SOLU,COMP,TRUE); (b) if the incompressible (default) or the compressible algorithm must be invoked (**FLDATA**,SOLU,COMP,TRUE).

7.1.1 Continuity Equation

From the law of conservation of mass law comes the continuity equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho V_x)}{\partial x} + \frac{\partial(\rho V_y)}{\partial y} + \frac{\partial(\rho V_z)}{\partial z} = 0 \qquad (7.1\text{–}1)$$

where: V_x, V_y and V_z = components of the velocity vector in the x, y and z directions, respectively

ϱ = density (see section 7.6.1)

x, y, z = global Cartesian coordinates

t = time

The rate of change of density can be replaced by the rate of change of pressure and the rate at which density changes with pressure:

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial P}\frac{\partial P}{\partial t} \qquad (7.1\text{–}2)$$

where: P = pressure

The evaluation of the derivative of the density with respect to pressure comes from the equation of state. If the compressible algorithm is used (**FLDATA,**SOLU,COMP,TRUE), an ideal gas is assumed:

$$\rho = \frac{P}{RT} \Rightarrow \frac{\partial \rho}{\partial P} = \frac{1}{RT} \tag{7.1–3}$$

where:
- R = gas constant
- T = temperature

If the incompressible solution algorithm is used (the default), the user can control the specification of the value with:

$$\frac{d\rho}{dP} = \frac{1}{\beta} \tag{7.1–4}$$

where:
- β = input as beta on the **FLDATA**,BULK command (default = 10^{15})

The default value of 10^{15} implies that for a perfectly incompressible fluid, pressure waves will travel infinitely fast throughout the entire problem domain, e.g. a change in mass flow will be seen downstream immediately .

7.1.2 Momentum Equations

In a Newtonian fluid, the relationship between the stress and rate of deformation of the fluid (in indicial notation) is:

$$\tau_{ij} = -P\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij}\, \lambda \frac{\partial u_i}{\partial x_i} \tag{7.1–5}$$

where:
- τ_{ij} = stress tensor
- u_i = orthogonal velocities ($u_1 = V_x$, $u_2 = V_y$, $u_3 = V_z$)
- μ = dynamic viscosity
- λ = second coefficient of viscosity

The final term, the product of the second coefficient of viscosity and the divergence of the velocity, is zero for a constant density fluid and is considered small enough to neglect in a compressible fluid.

Equation (7.1–5) transforms the momentum equations to the Navier–Stokes equations; however, these will still be referred to as the momentum equations elsewhere in this chapter.

The momentum equations, without further assumptions regarding the properties, are as follows:

$$\frac{\partial \rho V_x}{\partial t} + \frac{\partial(\rho V_x V_x)}{\partial x} + \frac{\partial(\rho V_y V_x)}{\partial y} + \frac{\partial(\rho V_z V_x)}{\partial z} = \rho g_x - \frac{\partial P}{\partial x}$$
$$+ R_x + \frac{\partial}{\partial x}\left(\mu_e \frac{\partial V_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_e \frac{\partial V_x}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_e \frac{\partial V_x}{\partial z}\right) + T_x \qquad (7.1\text{–}6)$$

$$\frac{\partial \rho V_y}{\partial t} + \frac{\partial(\rho V_x V_y)}{\partial x} + \frac{\partial(\rho V_y V_y)}{\partial y} + \frac{\partial(\rho V_z V_y)}{\partial z} = \rho g_y - \frac{\partial P}{\partial y}$$
$$+ R_y + \frac{\partial}{\partial x}\left(\mu_e \frac{\partial V_y}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_e \frac{\partial V_y}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_e \frac{\partial V_y}{\partial z}\right) + T_y \qquad (7.1\text{–}7)$$

$$\frac{\partial \rho V_z}{\partial t} + \frac{\partial(\rho V_x V_z)}{\partial x} + \frac{\partial(\rho V)_y V_z)}{\partial y} + \frac{\partial(\rho V_z V_z)}{\partial z} = \rho g_z - \frac{\partial P}{\partial z}$$
$$+ R_x + \frac{\partial}{\partial x}\left(\mu_e \frac{\partial V_z}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu_e \frac{\partial V_z}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu_e \frac{\partial V_z}{\partial z}\right) + T_z \qquad (7.1\text{–}8)$$

where: g_x, g_y, g_z = components of acceleration due to gravity (input on **ACEL** command)

ϱ = density (input as described in Section 7.6)

μ_e = effective viscosity (discussed below)

R_x, R_y, R_z = distributed resistances (discussed below

T_x, T_y, T_z = viscous loss terms (discussed below)

For a laminar case, the effective viscosity is merely the dynamic viscosity, a fluid property (input as described in Section 7.6. The effective viscosity for the turbulence model is described later in this section.

The terms R_x, R_y R_z represent any source terms the user may wish to add. An example is distributed resistance, used to model the effect of some geometric feature without modeling its geometry. Examples of this include flow through screens and porous media.

The terms T_x, T_y T_z are viscous loss terms which are eliminated in the incompressible, constant property case. The order of the differentiation is reversed in each term, reducing the term to a derivative of the continuity equation, which is zero.

$$T_x = \frac{\partial}{\partial x}\left(\mu \frac{\partial V_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial V_y}{\partial x}\right) + \frac{\partial}{\partial z}\left(\mu \frac{\partial V_z}{\partial x}\right) \qquad (7.1\text{–}9)$$

$$T_y = \frac{\partial}{\partial x}\left(\mu \frac{\partial V_x}{\partial y}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial V_y}{\partial y}\right) + \frac{\partial}{\partial z}\left(\mu \frac{\partial V_z}{\partial y}\right) \qquad (7.1\text{–}10)$$

$$T_z = \frac{\partial}{\partial x}\left(\mu\frac{\partial V_x}{\partial z}\right) + \frac{\partial}{\partial y}\left(\mu\frac{\partial V_y}{\partial z}\right) + \frac{\partial}{\partial z}\left(\mu\frac{\partial V_z}{\partial z}\right) \qquad (7.1\text{–}11)$$

The conservation of energy can be expressed in terms of the stagnation (total) temperature, often useful in highly compressible flows, or the static temperature, appropriate for low speed incompressible analyses.

7.1.3 Compressible Energy Equation

The complete energy equation is solved in the compressible case (**FLDATA**,SOLU,COMP,TRUE) with heat transfer (**FLDATA**,SOLU,TEMP,TRUE).

In terms of the total (or stagnation) temperature, the energy equation is:

$$\frac{\partial}{\partial t}(\rho C_p T_o) + \frac{\partial}{\partial x}(\rho V_x C_p T_o) + \frac{\partial}{\partial y}(\rho V_y C_p T_o) + \frac{\partial}{\partial z}(\rho V_z C_p T_o) =$$

$$\frac{\partial}{\partial x}\left(K\frac{\partial T_o}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial T_o}{\partial y}\right) + \frac{\partial}{\partial z}\left(K\frac{\partial T_o}{\partial z}\right) + W^v + E^k + Q_v + \Phi + \frac{\partial P}{\partial t} \qquad (7.1\text{–}12)$$

where: C_p = specific heat (input with **FLDATA**,NOMI,SPHT,value for fluid, **MP**,C command for non–fluid element)

T_o = total (or stagnation) temperature (input and output as TTOT)

K = thermal conductivity (input with **FLDATA**,NOMI,COND, value for fluid, **MP**,KXX command for non–fluid element)

W^v = viscous work term

Q_v = volumetric heat source (input with **BFE**,,HGEN)

Φ = viscous heat generation term

E^k = kinetic energy (defined later)

The static (TEMP) temperature is calculated from the total (TTOT) temperature from the kinetic energy:

$$T = T_0 - \frac{V^2}{2C_p} \qquad (7.1\text{–}13)$$

where: V = magnitude of the fluid velocity vector

The static and total temperatures for the non–fluid nodes will be the same.

The W^v, E^k and Φ terms are described next.

The viscous work term is:

$$
\begin{aligned}
W^{v} \; = \; & V_x\mu \left[\frac{\partial^2 V_x}{\partial x^2} + \frac{\partial^2 V_x}{\partial y^2} + \frac{\partial^2 V_x}{\partial z^2} + \frac{\partial}{\partial x}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z} \right) \right] \\
& + V_y\mu \left[\frac{\partial^2 V_y}{\partial x^2} + \frac{\partial^2 V_y}{\partial y^2} + \frac{\partial^2 V_y}{\partial z^2} + \frac{\partial}{\partial y}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z} \right) \right] \qquad (7.1\text{–}14) \\
& + V_z\mu \left[\frac{\partial^2 V_z}{\partial x^2} + \frac{\partial^2 V_z}{\partial y^2} + \frac{\partial^2 V_z}{\partial z^2} + \frac{\partial}{\partial z}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z} \right) \right]
\end{aligned}
$$

In the literature, this term is often expressed in tensor notation more compactly expressed as:

$$
W^{v} = u_j\mu \left[\frac{\partial}{\partial x_i} \frac{\partial u_j}{\partial x_i} + \frac{\partial}{\partial x_k} \frac{\partial u_k}{\partial x_j} \right] \qquad (7.1\text{–}15)
$$

where the repetition of a subscript implies a summation over the three orthogonal directions.

The kinetic energy term is

$$
E^{k} = -\frac{\partial}{\partial x}\left[\frac{K}{C_p} \frac{\partial}{\partial x}\left(\frac{1}{2} | V^2 | \right) \right] - \frac{\partial}{\partial y}\left[\frac{K}{C_p} \frac{\partial}{\partial y}\left(\frac{1}{2} | V^2 | \right) \right] - \frac{\partial}{\partial z}\left[\frac{K}{C_p} \frac{\partial}{\partial z}\left(\frac{1}{2} | V^2 | \right) \right] \qquad (7.1\text{–}16)
$$

Finally, the viscous dissipation term is

$$
\begin{aligned}
\Phi = & 2\mu \left[\left(\frac{\partial V_x}{\partial x} \right)^2 + \left(\frac{\partial V_y}{\partial y} \right)^2 + \left(\frac{\partial V_z}{\partial z} \right)^2 \right] \\
& + \mu \left[\left(\frac{\partial V_y}{\partial x} + \frac{\partial V_x}{\partial y} \right)^2 + \left(\frac{\partial V_z}{\partial y} + \frac{\partial V_y}{\partial z} \right)^2 + \left(\frac{\partial V_x}{\partial z} + \frac{\partial V_z}{\partial x} \right)^2 \right]
\end{aligned} \qquad (7.1\text{–}17)
$$

Again, in the literature this is usually expressed in tensor notation:

$$
\Phi \; = \; \mu \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} \right) \frac{\partial u_i}{\partial x_k} \qquad (7.1\text{–}18)
$$

7.1.4 Incompressible Energy Equation

The energy equation for the incompressible case may be derived from the one for the compressible case by neglecting the viscous work (W^v), the pressure work, viscous dissipation (ϕ), and the kinetic energy (E^k). As the kinetic energy is neglected, the static temperature (T) and the total temperature (T_o) are the same. The energy equation now takes the form of a thermal transport equation for the static temperature:

$$\frac{\partial}{\partial t}(\rho C_p T) + \frac{\partial}{\partial x}(\rho V_x C_p T) + \frac{\partial}{\partial y}(\rho V_y C_p T) + \frac{\partial}{\partial z}(\rho V_z C_p T)$$

$$= \frac{\partial}{\partial x}\left(K\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(K\frac{\partial T}{\partial z}\right) + Q_v \qquad (7.1\text{–}19)$$

7.1.5 Turbulence

If inertial effects are great enough with respect to viscous effects, the flow may be turbulent. The user is responsible for deciding whether or not the flow is turbulent. Turbulence modeling is activated with the command (**FLDATA,**SOLU,TURB,TRUE). Turbulence means that the instantaneous velocity is fluctuating at every point in the flowfield. The velocity is thus expressed in terms of a mean value and a fluctuating component:

$$V_x = \overline{V}_x + V_x' \qquad (7.1\text{–}20)$$

where:

$\overline{V}_x$ = mean component of velocity in x–direction

V_x' = fluctuating component of velocity in x–direction

If an expression such as this is used for the instantaneous velocity in the Navier–Stokes equations, the equations may then be time averaged, noting that the time average of the fluctuating component is zero, and the time average of the instantaneous value is the average value. The time interval for the integration is arbitrarily chosen as long enough for this to be true and short enough so that “real time” transient effects do not affect this integration.

$$\frac{1}{\delta_t}\int_0^{\delta_t} V_x' dt = 0 \quad ; \qquad \frac{1}{\delta_t}\int_0^{\delta_t} V_x dt = \overline{V}_x \qquad (7.1\text{–}21)$$

After the substitution of equation (7.1–20) into the momentum equations, the time averaging leads to additional terms. The velocities in the momentum equations are the averaged ones, and we drop the bar in the subsequent expression of the momentum equations, so that the absence of a bar now means the mean value. The extra terms are:

$$\sigma_x^R = -\frac{\partial}{\partial x}(\overline{\rho V_x' V_x'}) - \frac{\partial}{\partial y}(\overline{\rho V_x' V_y'}) - \frac{\partial}{\partial z}(\overline{\rho V_x' V_z'}) \qquad (7.1\text{–}22)$$

$$\sigma_y^R = -\frac{\partial}{\partial x}(\overline{\rho V_y' V_x'}) - \frac{\partial}{\partial y}(\overline{\rho V_y' V_y'}) - \frac{\partial}{\partial z}(\overline{\rho V_y' V_z'}) \qquad (7.1\text{–}23)$$

$$\sigma_z^R = -\frac{\partial}{\partial x}(\overline{\rho V_z' V_x'}) - \frac{\partial}{\partial y}(\overline{\rho V_z' V_y'}) - \frac{\partial}{\partial z}(\overline{\rho V_z' V_z'}) \qquad (7.1\text{–}24)$$

where: σ^R = Reynolds stress terms

In the eddy viscosity approach to turbulence modeling one puts these terms into the form of a viscous stress term with an unknown coefficient, the turbulent viscosity. For example:

$$-\overline{\rho V_x V_y} = \mu_t \frac{\partial V_x}{\partial y} \qquad (7.1\text{–}25)$$

The main advantage of this strategy comes from the observation that the representation of σ^R is of exactly the same form as that of the diffusion terms in the original equations. The two terms can be combined if an effective viscosity is defined as the sum of the laminar viscosity and the turbulent viscosity:

$$\mu_e = \mu + \mu_t \qquad (7.1\text{–}26)$$

The solution to the turbulence problem then revolves around the solution of the turbulent viscosity.

At this point, the reader may wonder why neither the Reynolds stress nor turbulent heat flux terms contain a fluctuating density. This is a result of the application of Favre averaging to equations (7.1–22) to (7.1–24). Bilger(187) gives an excellent description of Favre averaging. Basically this technique weights each term by the mean density to create a Favre averaged value for variable ϕ which does not contain a fluctuating density:

$$\tilde{\phi} \equiv \frac{\overline{\rho\phi}}{\overline{\rho}} \qquad (7.1\text{–}27)$$

The tilde incidates the Favre averaged variable. For brevity, reference is made to Bilger(187) for further details.

There are six turbulence models available in FLOTRAN. The model choice is made with the **FLDATA**,TURB,MODL,value command. The model acronym and the values used to invoke the different models are as follows:

Value	Model
0,1	Default, Standard k–ε
2	Zero Equation Model
3	RNG – Re–normalized Group Model
4	NKE – New k–ε model due to Shih
5	GIR – Model due to Girimaji
6	SZL – Shi, Zhu, Lumley model

The simplest model is the Zero Equation Model, and the other five models are the two equation standard k–ε model and four extensions of it. In the Zero Equation Model, the turbulent viscosity is calculated as:

$$\mu_t = \rho \, L_s^2 \, \sqrt{\Phi} \qquad (7.1–28)$$

where: μ_t = turbulent viscosity

Φ = viscous dissipation (equation (7.1–17))

$$L_s = \begin{cases} \text{value} & \text{if value on the \textbf{FLDATA},TURB,ZELS command is supplied} \\ \text{minimum} \begin{Bmatrix} .4 \;\; L_n \\ .09 \; L_c \end{Bmatrix} & \text{if value on the \textbf{FLDATA},TURB,ZELS command is not supplied} \end{cases}$$

L_n = shortest distance from the node to the closest wall

L_c = characteristic length scale (largest value of L_n encountered)

In the remaining models, the turbulent viscosity is calculated as a function of the turbulence parameters kinetic energy k (ENKE) and its dissipation rate (ENDS) through the expression below. In the RNG and standard models, C_{mu} is constant, while it varies in the other models.

$$\mu_t = C_\mu \, \rho \, \frac{k^2}{\epsilon} \qquad (7.1–29)$$

where: k = turbulent kinetic energy (input/output quantity ENKE)

ε = turbulent kinetic energy dissipation rate (input/output quantity ENDS)

Two equation models entail solving partial differential equations for turbulent kinetic energy (ENKE) and its dissipation rate (ENDS). The equations below are for the standard model (**FLDATA**,TURB,MODL,1 – default). The extensions and different calculations for the other k–ε models wil be discussed in turn. The basic equations are as follows:

Standard Model:

The reader is referred to Spalding and Launder(178) for details.

The Turbulent Kinetic Energy equation is:

$$\frac{\partial \rho k}{\partial t} + \frac{\partial(\rho V_x k)}{\partial x} + \frac{\partial(\rho V_y k)}{\partial y} + \frac{\partial(\rho V_z k)}{\partial z}$$

$$= \frac{\partial}{\partial x}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{\mu_t}{\sigma_k}\frac{\partial k}{\partial z}\right) \qquad (7.1\text{–}30)$$

$$+ \mu_t \Phi - \rho\epsilon + \frac{C_4 \beta \mu_t}{\sigma_t}\left(g_x \frac{\partial T}{\partial x} + g_y \frac{\partial T}{\partial y} + g_z \frac{\partial T}{\partial z}\right)$$

The Dissipation Rate equation is:

$$\frac{\partial \rho \epsilon}{\partial t} + \frac{\partial(\rho V_x \epsilon)}{\partial x} + \frac{\partial(\rho V_y \epsilon)}{\partial y} + \frac{\partial(\rho V_z \epsilon)}{\partial z}$$

$$= \frac{\partial}{\partial x}\left(\frac{\mu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\mu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{\mu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial z}\right) \qquad (7.1\text{–}31)$$

$$+ C_{1\epsilon}\, \mu_t \frac{\epsilon}{k}\Phi - C_2 \rho \frac{\epsilon^2}{k} + \frac{C_\mu \left(1 - C_3\right) \beta \rho k}{\sigma_t}\left(g_x \frac{\partial T}{\partial x} + g_y \frac{\partial T}{\partial y} + g_z \frac{\partial T}{\partial z}\right)$$

The final term in each equation are terms used to model the effect of buoyancy and are described by Viollet(177). Default values for the various constants in the standard model are provided by Lauder and Spalding(178). See Table 7.1–1.

Table 7.1–1 Standard Model Coefficients

Value	Default	Command
$C_1, C_{1\varepsilon}$	1.44	(**FLDATA**,TURB,C1,value)
C_2	1.92	(**FLDATA**,TURB,C2,value)
C_μ	0.09	(**FLDATA**,TURB,CMU,value)
σ_k	1.0	(**FLDATA**,TURB,SCTK,value)
σ_ε	1.3	(**FLDATA**,TURB,SCTD,value)
σ_t	1.0	(**FLDATA**,TURB,SCTM,value)
C_3	1.0	(**FLDATA**,TURB,BUC3,value)
C_4	0.0	(**FLDATA**,TURB,BUC4,value)
β	0.0	(**FLDATA**,TURB,BETA,value)

The solution to the turbulence equations is used to calculate the effective viscosity and the effective thermal conductivity:

$$\mu_e = \mu + C_\mu \rho \frac{k^2}{\epsilon} \tag{7.1–32}$$

$$K_e = K + \frac{\mu_t C_p}{\sigma_t} \tag{7.1–33}$$

where:
μ_e = effective viscosity
K_e = effective conductivity
σ_t = turbulent Prandtl number

The four extensions to the k–ε model have changes in either the C_μ term or in the source term of the dissipation equation. The new functions utilize two invariants constructed from the symmetric deformation tension S_{ij}, and the antisymmetric rotation tension W_{ij}. These are based on the velocity components V_k in the flowfield.

$$S_{ij} = \frac{1}{2} (V_{i,j} + V_{j,i}) \tag{7.1–34}$$

$$W_{ij} = \frac{1}{2} (V_{i,j} - V_{j,i}) + C_r \, \Omega_m \, \epsilon_{mij} \tag{7.1–35}$$

where:
C_r = constant depending on turbulence model used
Ω_m = angular velocity of the coordinate system
ε_{mij} = alternating tensor operator

The invariants are:

$$\eta = \frac{k}{\epsilon} \sqrt{2 S_{ij} \, S_{ij}} \tag{7.1–36}$$

and

$$\zeta = \frac{k}{\epsilon} \sqrt{2 W_{ij} \, W_{ij}} \tag{7.1–37}$$

RNG Turbulence Model:

In the RNG model (**FLDATA**,TURB,MODL,3) the constant $C_{1\varepsilon}$ in the dissipation equation (7.1–31), is replaced by a function of one of the invariants.

$$C_{1\epsilon} = 1.42 - \frac{\eta \left(1 - \frac{\eta}{\eta_\infty}\right)}{1 + \beta \eta^3} \tag{7.1–38}$$

Table 7.1–2 RNG Model Coefficients

Parameter	Default	Command
β_∞	0.12	**FLDATA**,RNGT,BETA,value
C_2	1.68	**FLDATA**,RNGT,C2,value
C_μ	0.085	**FLDATA**,RNGT,CMU,value
σ_k	0.72	**FLDATA**,RNGT.SCTK,value
σ_ε	0.72	**FLDATA**,RNGT,SCTD,value
η_∞	4.38	**FLDATA**,RNGT,ETAI,value

In the RNG model a constant C_μ is used. The value is specified with a separate command than the one used to specify the C_μ in the standard model. The same is true of the constant C_2. As shown in the above table. the diffusion multipliers have different values than the default model, and these parameters also have their own commands for the RNG model. The value of the rotational constant C_r in the RNG model is 0.0. Quantities in equation (7.1–31) not specified in Table 7.1–2 are covered by Table 7.1–1.

NKE Turbulence Model:

The NKE Turbulence model (**FLDATA**,TURB,MODL,4) uses both a variable C_μ term and a new dissipation source term.

The C_μ function used by the NKE model is a function of the invariants.

$$C_\mu = \frac{1}{4 + 1.5\sqrt{\eta^2 + \zeta^2}} \tag{7.1–39}$$

The production term for dissipation takes on a different form. From equation (7.1–31), the production term for the standard model is:

$$C_1\ \mu_t \frac{\epsilon}{k}\ \Phi \tag{7.1–40}$$

The NKE model replaces this with:

$$\rho C_{\epsilon,1}\sqrt{2S_{ij}\ S_{ij}}\,\epsilon \tag{7.1–41}$$

The constant in the dissipation rate equation (7.1–31) is modified in the NKE model to be:

$$C_{\epsilon,1} = \max\left(C_{1M}, \frac{\eta}{\eta + 5}\right) \tag{7.1–42}$$

The constant C_2 in the dissipation equation (7.1–31) of the NKE model has a different value than that for the corresponding term in the standard model. Also, the values for the diffusion multipliers are different. Commands are provided for these variables to distinguish them from the standard model parameters. So for the NKE model, the input parameters are as follows:

Table 7.1–3 NKE Turbulence Model Coefficients

Parameter	Default	Command
C_{1M}	0.43	**FLDATA**,NKET,C1MX,value
C_2	1.90	**FLDATA**,NKET,C2,value
σ_k	1.0	**FLDATA**,NKET,SCTK,value
σ_ε	1.2	**FLDATA**,NKET,SCTD,value

The value of the rotational constant C_r in the NKE model is 3.0. All parameters in equations (7.1–30) and (7.1–31) not covered by this table are covered in Table 7.1–1

GIR Turbulence Model:

The Girimaji model (**FLDATA**,TURB,MODL,5) relies on a complex function for the calculation of the C_μ coefficient. The coefficients in Table 7.1–4 (below) are used.

Table 7.1–4 GIR Turbulence Model Coefficients

Parameter	Default	Command
C_1^0	3.6	**FLDATA**,GIRT,G0,value
C_1^1	0.0	**FLDATA**,GIRT,G1,value
C_2	0.8	**FLDATA**,GIRT,G2,value
C_3	1.94	**FLDATA**,GIRT,G3,value
C_4	1.16	**FLDATA**,GIRT,G4,value

These input values are used in a series of calculations as follows

First of all the coefficients L_1^0 to L_4 have to be determined from the input coeficients. Note, these coefficients are also needed for the coefficients of the nonlinear terms of this model, which wiil be discussed later.

$$L_1^0 = \frac{C_1^0}{2} - 1 \quad L_1^1 = C_1^1 + 1 \quad L_2 = \frac{C_2}{2} - \frac{2}{3} \quad L_3 = \frac{C_3}{2} - 1 \quad L_4 = \frac{C_4}{2} - 1 \qquad (7.1–43)$$

Secondly, the following coefficients have to calculated:

$$p = -\frac{2L_1^0}{\frac{1}{2}\,\eta^2 L_1^1} \qquad r = \frac{L_1^0 L_2}{\left(\frac{1}{2}\,\eta^2 L_1^1\right)^2} \qquad \Theta = \arccos \frac{-b/2}{\sqrt{-a^3/27}}$$

$$q = \frac{1}{\left(\frac{1}{2}\,\eta^2 L_1^1\right)^2}\left[\left(L_1^0\right)^2 + \frac{1}{2}\,\eta^2 L_1^1 L_2 - \frac{1}{3}\,\eta^2\,(L_3)^2 + \zeta^2\,(L_4)^2\right] \tag{7.1–44}$$

$$a = q - \frac{p^2}{3} \qquad b = \frac{1}{27}\,(2p^3 - 9pq + 27r) \qquad D = \frac{b^2}{4} + \frac{a^3}{27}$$

With these coefficients we can now determine the coefficient C_μ from the following set of equations:

$$C_\mu = -\begin{cases} L_1^0 L_2 \Big/ \left[\left(L_1^0\right)^2 + \zeta^2\,(L_4)^2\right] & \text{if } \eta = 0 \\ L_1^0 L_2 \Big/ \left[\left(L_1^0\right)^2 - \frac{1}{3}\eta\,(L_3)^2 + \zeta_2\,(L_4)^2\right] & \text{if } L_1^1 = 0 \\ -\frac{p}{3} + \left(-\frac{b}{2} + \sqrt{D}\right)^{1/3} + \left(-\frac{b}{2} + \sqrt{D}\right)^{1/3} & \text{if } D > 0 \\ -\frac{p}{3} + 2\sqrt{\frac{-a}{3}}\,\cos\left(\frac{\Theta}{3}\right) & \text{if } D < 0,\ b < 0 \\ -\frac{p}{3} + 2\sqrt{\frac{-a}{3}}\,\cos\left(\frac{\Theta}{3} + \frac{2}{3}\,\pi\right) & \text{if } D < 0,\ b > 0 \end{cases} \tag{7.1–45}$$

and for the GIR model, the rotational term constant C_r is

$$C_r = \frac{C_4 - 4}{C_4 - 2} \tag{7.1–46}$$

SZL Turbulence Model:

The Shi–Zhu–Lemley turbulence model (**FLDATA**,TURB,MODL,6) uses a simple expression for the C_μ coefficient and uses the standard dissipation source terms.

The user controls three constants in the calculation of the coefficients:

$$C_\mu = \frac{A_{s1}}{A_{s2} + \eta + A_{s3}\,\zeta} \qquad (7.1–47)$$

The constants and their defaults are as follows:

Table 7.1–5 SZL Turbulence Model Coefficients

Constant	Default	Command
A_{s1}	0.66666	**FLDATA**,SZLT,SZL1,value
A_{s2}	1.25	**FLDATA**,SZLT,SZL2,value
A_{s3}	0.90	**FLDATA**,SZLT,SZL3,value

The value of the rotational constant C_r for the SZL model is 4.0.

The k–ε models are not valid immediately adjacent to the walls. A wall turbulence model is used for the wall elements. Given the current value of the velocity parallel to the wall at a certain distance from the wall, an approximate iterative solution is obtained for the wall shear stress. The equation is known as the “Log–Law of the Wall” and is discussed in White(181) and Launder and Spalding(178).

$$\frac{V_{tan}}{\sqrt{\frac{\tau}{\rho}}} = \frac{1}{\kappa}\left(\ln\frac{E\ \delta}{\nu}\sqrt{\frac{\tau}{\rho}}\right) \qquad (7.1–48)$$

where:

V_{tan}	=	velocity parallel to the wall
τ	=	shear stress
ν	=	kinematic viscosity (μ/ϱ)
κ	=	law of the wall constant (**FLDATA**,TURB,KAPP,value)
E	=	law of the wall constant (**FLDATA**,TURB,EWLL,value)
δ	=	distance from the wall

The default values of κ and E are 0.4 and 9.0 respectively, the latter corresponding to a smooth wall condition.

From the shear stress comes the calculation of the wall viscosity:

$$\mu_w = \delta\frac{\tau}{u_{tan}} \qquad (7.1–49)$$

The wall element viscosity value is the larger of the laminar viscosity and that calculated from equation (7.1–49).

Near wall values of the turbulent kinetic energy are obtained from the k–ε model. The near wall value of the dissipation rate is dominated by the length scale and is given by equation (7.1–50).

$$\epsilon_{nw} = \frac{C_\mu^{(.75)} k_{nw}^{(1.5)}}{\kappa\delta} \tag{7.1–50}$$

where: ε_{nw} = near wall dissipation rate

k_{nw} = near wall kinetic energy

The user may elect to use an alternative wall formulation directly based on the equality of turbulence production and dissipation. The alternative wall treatment is invoked through the command **FLDATA**,TURB,WALL,EQLB. This condition leads to the following expression for the wall parameter y$^+$ (see White(181) for more background):

$$y^+ = \frac{C_\mu^{1/4}\ \rho\ k_{nw}^{1/2}\ \delta}{\mu} \tag{7.1–51}$$

The wall element effective viscosity and thermal conductivity are then based directly on the value of y$^+$. The laminar sub–layer extends to y_t^+ , which is input on the **FLDATA**,TURB,TRAN command, with the default being 11.5.

For y$^+$ $< y_t^+$:

$$\mu_{eff} = \mu$$
$$K_{eff} = K \tag{7.1–52}$$

For y$^+$ $\geq$ y_t^+ :

$$\mu_{eff} = \frac{\mu\ \ y^+}{\frac{1}{\kappa}\ \ell n\ (Ey^+)} \tag{7.1–53}$$

$$K_{eff} = \frac{C_p}{\sigma_t}\ \frac{\mu\ y^+}{\left(\frac{1}{\kappa}\ \ell n\ Ey^+\ + P_{fn}\right)} \tag{7.1–54}$$

where: ℓn = natural logarithm

$$P_{fn} = \frac{(\pi/4)}{\sin\ (\pi/4)} \left(\frac{A}{\kappa}\right)^{1/2} \left(\frac{Pr}{\sigma_t} - 1\right) / \left(\frac{Pr}{\sigma_t}\right)^{1/4}$$

Pr = Prandtl number

Although the wall treatment should not affect the laminar solution, the shear stress calculation is part of the wall algorithm. Hence, shear stresses for laminar flow will be slightly different for the equilibrium model.

7.1.6 Pressure

For numerical accuracy reasons, the algorithm solves for a relative pressure rather than an absolute pressure.

Considering the possibility that the equations are solved in a rotating coordinate system, the defining expression for the relative pressure is:

$$P_{abs} = P_{ref} + P_{rel} - \rho_o \{g\} \cdot \{r\} + \frac{1}{2} \rho_o (\{\omega\} \times \{\omega\} \times \{r\}) \cdot \{r\} \qquad (7.1\text{–}55)$$

where: ϱ_o = reference density

P_{ref} = reference pressure

$\{g\}$ = acceleration vector due to gravity (input using **ACEL** command)

P_{abs} = absolute pressure

P_{rel} = relative pressure

$\{r\}$ = position vector of the fluid particle with respect to the rotating coordinate system

$\{\omega\}$ = constant angular velocity vector of the coordinate system (input using **CGOMGA** command)

The reference density is calculated from the equation of state defined by the property type using the nominal temperature (**FLDATA**,TEMP,NOMI,value).

The reference pressure is set by the user via the command **FLDATA**,PRES,REFE,value.

The momentum equation, again considering a rotating coordinate system, is expressed in vector notation as follows:

$$\rho \frac{D\{V\}}{Dt} + 2\rho \{\omega\} \times \{V\} + \rho \{\omega\} \times \{\omega\} \times \{r\} = \rho \{g\} - \nabla P_{abs} + \mu \nabla^2 \{V\} \qquad (7.1\text{–}56)$$

where: $\{V\}$ = vector velocity in the rotating coordinate system

μ = fluid viscosity (assumed constant for simplicity)

ϱ = fluid density

In the absence of rotation, {V} is simply the velocity vector in the global coordinate system.

The negative of the gradient of the absolute pressure is:

$$- \nabla P_{abs} = -\nabla P_{rel} - \rho_o \{g\} + \rho_o \{\omega\} \times \{\omega\} \times \{r\} \tag{7.1–57}$$

Inserting this expression into the vector form of the momentum equation puts it in terms of the relative pressure and the density differences.

$$\begin{aligned} \rho \frac{D\{V\}}{Dt} &+ 2 \rho \{\omega\} \times \{V\} + (\rho - \rho_o) \{\omega\} \times \{\omega\} \times \{r\} \\ &= (\rho - \rho_o) \{g\} - \nabla P_{rel} + \mu \nabla^2 \{V\} \end{aligned} \tag{7.1–58}$$

This form has the desirable feature (from a numerical precision standpoint) of expressing the forcing function due to gravity and the centrifugal acceleration in terms of density differences.

For convenience, the relative pressure output is that measured in the stationary global coordinate system. That is, the rotational terms are subtracted from the pressure calculated by the algorithm.

Conversely, the total pressure is output in terms of the rotating coordinate system frame. This is done for the convenience of those working in turbomachinery applications.

7.1.7 Multiple Species Transport

Several different fluids, each with different properties, are tracked if the multiple species option is invoked (**FLDATA**,SOLU,SPEC,TRUE).

A single momentum equation is solved for the flow field. The properties for this equation are calculated from those of the species fluids and their respective mass fractions if the user specifies the composite gas option (**FLDATA**,PROT,DENS,CGAS) for density or the composite mixture option (**FLDATA**,PROT,DENS,CMIX). CGAS only applies for density, but CMIX applies to density, viscosity or conductivity. If these options are not invoked, the species fluids are carried by a bulk fluid, with the momentum equation solved with properties of a single fluid.

The governing equations for species transport are the mass balance equations for each of the species.

For i = 1, . . . , n–1 (where n is the number of species)

$$\frac{\partial(\rho Y_i)}{\partial t} + \nabla \cdot (\rho Y_i V) + \nabla \cdot (\rho D_{mi} \nabla Y_i) = 0 \tag{7.1–59}$$

where: Y_i = mass fraction for the ith species

ϱ = bulk density (mass/length^3)

V = velocity vector (length/time)

D_{mi} = mass diffusion coefficient (length^2/time) (label MDIF on **MSPROP** command)

The equation for the nth species, selected by the user as the “algebraic species,” is not solved directly. The mass fraction for the nth species is calculated at each node from the identity:

$$Y_N = 1 - \sum_{i=1}^{n-1} Y_i \tag{7.1–60}$$

The diffusion information available for the species fluid is sometimes cast in terms of a Schmidt number for a species (not to be confused with the turbulent Schmidt number). The relationship between the Schmidt number and the mass diffusion coefficient is as follows:

$$Sc_i = \frac{\mu}{\rho D_{mi}} \tag{7.1–61}$$

In the above expression, the density and the viscosity are those of the bulk carrier fluid, or the “average” properties of the flow.

As with the general “bulk” momentum equation, the effect of turbulence is to increase the diffusion and is modeled with an eddy viscosity approach. First note that the laminar diffusion term can be cast in terms of the “laminar” Schmidt number associated with the species diffusion:

$$\nabla \cdot (\rho D_i \nabla Y_i) = \nabla \cdot \left(\frac{\mu}{Sc_i} \nabla Y_i \right) \tag{7.1–62}$$

In the presence of turbulence, an additional term is added:

$$\nabla \cdot \left(\frac{\mu}{Sc_i} \nabla Y_i \right) \rightarrow \nabla \cdot \left(\left(\frac{\mu}{Sc_i} + \frac{\mu_t}{Sc_{Ti}} \right) \nabla Y_i \right) \tag{7.1–63}$$

where: μ_t = turbulent viscosity (from the turbulence model)

Sc_{Ti} = turbulent Schmidt number

The turbulent Schmidt number defaults to 1.0 (typical for most problems) and can be modified with the **MSSPEC** command.

7.2 Derivation of Fluid Flow Matrices

A segregated, sequential solution algorithm is used. This means that element matrices are formed, assembled and the resulting system solved for each degree of freedom separately. Development of the matrices proceeds in two parts. In the first, the form of the equations is achieved and an approach taken towards evaluating all the terms. Next, the segregated solution algorithm is outlined and the element matrices are developed from the equations.

7.2.1 Discretization of the Equations

The momentum, energy, species transport, and turbulence equations all have the form of a scalar transport equation. There are four types of terms: transient, advection, diffusion, and source. For the purposes of describing the discretization methods, let us refer to the variable considered as ϕ. The form of the scalar transport equation is:

$$\frac{\partial}{\partial t}(\rho C_\phi \phi) + \frac{\partial}{\partial x}(\rho V_x C_\phi \phi) + \frac{\partial}{\partial y}(\rho V_y C_\phi \phi) + \frac{\partial}{\partial z}(\rho V_z C_\phi \phi) = \frac{\partial}{\partial x}\left(\Gamma_\phi \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma_\phi \frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma_\phi \frac{\partial \phi}{\partial y}\right) + S_\phi \tag{7.2–1}$$

where:
C_ϕ = transient and advection coefficient
Γ_ϕ = diffusion coefficient
S_ϕ = source terms

Table 7.2–1 below shows what the variables, coefficients, and source terms are for the transport equations. The pressure equation is derived using the continuity equation. Its form will unfold during the discussion of the segregated solver. The terms are defined in the previous section. SP01 through SP06 are the mass fraction degrees of freedom associated with the six species.

Since the approach is the same for each equation, only the generic transport equation need be treated. Each of the four types of terms will be outlined in turn. Since the complete derivation of the discretization method would require too much space, the methods will be outlined and the reader referred to more detailed expositions on the subjects.

Table 7.2–1 Transport Equation Representation

ϕ (DOF)	C_ϕ	Γ_ϕ	S_ϕ
VX	1	μ_e	$\varrho g_x - \partial p/\partial x + R_x$
VY	1	μ_e	$\varrho g_y - \partial p/\partial y + R_y$
VZ	1	μ_e	$\varrho g_z - \partial p/\partial z + R_z$
TEMP	C_p	K	$Q_v + E^k + W^v + \mu\Phi + \partial p/\partial t$
ENKE	1	μ_t / σ_k	$\mu_t\Phi / \mu - \varrho\varepsilon + C_4\beta\mu_t g_i (\partial T/\partial xi) / \sigma_t$
ENDS	1	$\mu_t / \sigma_\varepsilon$	$C_1\mu_t\varepsilon\Phi / k - C_2\varrho\varepsilon^2 / k + C_1C_\mu C_3\beta k g_i (\partial T/\partial x_i) / \sigma_t$
SP01–06	1	ϱ Dm$_i$	0

The discretization process, therefore, consists of deriving the element matrices to put together the matrix equation:

$$\left(\left[A_e^{\text{transient}}\right] + \left[A_e^{\text{advection}}\right] + \left[A_e^{\text{diffusion}}\right]\right)\{\phi_e\} = \left\{S_e^{\phi}\right\} \qquad (7.2\text{–}2)$$

Galerkin’s method of weighted residuals is used to form the element integrals. Denote by W^e the weighting function for the element, which is also the shape function.

Transient Term

The first of the element matrix contributions is from the transient term. The general form is simply:

$$\left[A_e^{\text{transient}}\right] = \int W^e \frac{\partial(\rho C_\phi \phi)^e}{\partial t}\, d(\text{vol}) \qquad (7.2\text{–}3)$$

A lumped mass approximation is used so that

$$\int W^e \frac{\partial(\rho C_\phi \phi)}{\partial t}\, d(\text{vol}) = \frac{\partial(\rho C_\phi \phi)}{\partial t} \int W^e d(\text{vol}) \qquad (7.2\text{–}4)$$

A backward difference scheme is used to evaluate the transient derivative. On a nodal basis, the following implicit formulation is used. The current time step is the nth time step and the expression involves the previous two time step results.

$$\frac{\partial(\rho\phi)}{\partial t} = \frac{(\rho\phi)_{n-2}}{2\Delta t} - \frac{4(\rho\phi)_{n-1}}{2\Delta t} + \frac{3(\rho\phi)_n}{2\Delta t} \qquad (7.2\text{–}5)$$

For a Volume of Fluid (VOF) analysis, the above equation is modified as only the results at one previous time step are needed:

$$\frac{\partial(\rho\phi)}{\partial t} = \frac{(\rho\phi)_n}{\Delta t} - \frac{(\rho\phi)_{n-1}}{\Delta t} \qquad (7.2\text{–}6)$$

The above first–order time difference scheme is chosen to be consistent with the current VOF advection algorithm.

The nth time step produces a contribution to the diagonal of the element matrix, while the derivatives from the previous time step form contributions to the source term.

Advection Term

Currently Flotran has two approaches to discretize the advection term. The monotone streamline upwind (MSU) approach is first order accurate and tends to produce smooth and monotone solutions. The streamline upwind/Petro–Galerkin (SUPG) approach is second order accurate and tends to produce oscillatory solutions.

Monotone Streamline Upwind Approach (MSU)

The advection term is handled through a monotone streamline approach based on the idea that pure advection transport is along characteristic lines. It is useful to think of the advection transport formulation in terms of a quantity being transported in a known velocity field. See Figure 7.2–1.

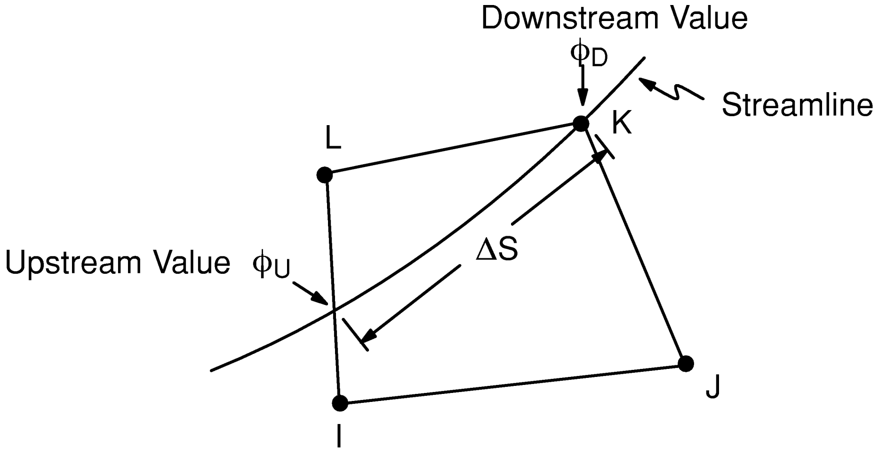


Figure 7.2–1 Streamline Upwind Approach

The velocity field itself can be envisioned as a set of streamlines everywhere tangent to the velocity vectors. The advection terms can therefore be expressed in terms of the streamline velocities.

In pure advection transport, one assumes that no transfer occurs across characteristic lines,i.e. all transfer occurs along streamlines. Therefore one may assume that the advection term,

$$\frac{\partial(\rho C_\phi V_x\ \phi)}{\partial x}+\frac{\partial(\rho C_\phi V_y\ \phi)}{\partial y}+\frac{\partial(\rho C_\phi V_z\ \phi)}{\partial z}=\frac{\partial(\rho C_\phi V_s\,\phi)}{\partial s} \qquad (7.2\text{–}7)$$

when expressed along a streamline, is constant through out an element:

$$\left[A_e^{advection}\right]=\frac{d(\rho C_\phi V_s \phi)}{ds}\int W^e\ d(vol) \qquad (7.2\text{–}8)$$

This formulation is made for every element, each of which will have only one node which gets contributions from inside the element. The derivative is calculated using a simple difference:

$$\frac{d(\rho C_\phi V_s)}{ds}=\frac{(\rho C_\phi V_s \phi)_U-(\rho C_\phi V_s \phi)_D}{\Delta s} \qquad (7.2\text{–}9)$$

where: D = subscript for value at the downstream node

U = subscript for value taken at the location at which the streamline through the downwind node enters the element

Δs = distance from the upstream point to the downstream node

The value at the upstream location is unknown but can be expressed in terms of the unknown nodal values it is between. See Figure 7.2–1 again.

The process consists of cycling through all the elements and identifying the downwind nodes. A calculation is made based on the velocities to see where the streamline through the downwind node came from. Weighting factors are calculated based on the proximity of the upwind location to the neighboring nodes.

Consult Rice and Schnipke(179) for more details .

The MSU approach is the default for all equations.

Streamline Upwind/Petro–Galerkin Approach (SUPG)

The SUPG approach consists of a Galerkin discretization of the advection term and an additional diffusion–like perturbation term which acts only in the advection direction.

$$\left[A_e^{\text{advection}}\right] = \int W^e \left\{ \frac{\partial(\rho V_x C_\phi \, \phi)}{\partial x} + \frac{\partial(\rho V_y C_\phi \, \phi)}{\partial y} + \frac{\partial(\rho V_z C_\phi \phi)}{\partial z} \right\} d(vol) \; +$$

(Galerkin)

$$C_{2\tau} \int \frac{zh}{2\, U_{mag}} \left\{ \frac{V_x \partial W^e}{\partial x} + \frac{V_y \partial W^e}{\partial y} + \frac{V_z \partial W^e}{\partial z} \right\} \qquad (7.2\text{–}10)$$

$$\left\{ \frac{V_x \partial(\rho C_\phi \, \phi)}{\partial x} + \frac{V_y \partial(\rho C_\phi \, \phi)}{\partial y} + \frac{V_z \partial(\rho C_\phi \, \phi)}{\partial z} \right\} d(vol)$$

(Perturbation)

where: $C_{2\tau}$ = global coefficient set to 1.0

h = element length along advection direction

U_{mag} = $\sqrt{V_x^2 + V_y^2 + V_z^2}$

$$z = \begin{cases} 1 & \text{if } 0 \leq Pe \leq 3 \\ Pe/3 & \text{if } Pe \geq 3 \end{cases}$$

$$Pe = \frac{\rho C_\phi \; U_{mag} \; h}{2 \; \Gamma_\phi} = \text{Peclet number}$$

It is clear from the SUPG approach that as the mesh is refined, the perturbation terms goes to zero and the Galkerin formulation approaches second order accuracy. The perturbation term provides the necessary stability which is missing in the pure Galkerin discretization. Consult Brooks and Hughes(224) for more details.

Diffusion Terms

The expression for the diffusion terms comes from an integration over the problem domain after the multiplication by the weighting function.

$$\text{Diffusion contribution} = \int W^e \frac{\partial}{\partial x} \left(\Gamma_\phi \frac{\partial \phi}{\partial x} \right) d(vol) + \int W^e \frac{\partial}{\partial y} \left(\Gamma_\phi \frac{\partial \phi}{\partial y} \right) d(vol) \qquad (7.2\text{–}11)$$

$$+ \int W^e \frac{\partial}{\partial z} \left(\Gamma_\phi \frac{\partial \phi}{\partial z} \right) d(vol)$$

The x, y and z terms are all treated in similar fashion. Therefore, the illustration is with the term in the x direction. An integration by parts is applied:

$$\int W^e \frac{\partial}{\partial x}\left(\Gamma_\phi \frac{\partial \phi}{\partial x}\right) d(vol) \;=\; \int \frac{\partial W^e}{\partial x}\;\Gamma_\phi \frac{\partial \phi}{\partial x}\;\; d(vol) \qquad (7.2\text{–}12)$$

Once the derivative of ϕ is replaced by the nodal values and the derivatives of the weighting function, the nodal values will be removed from the integrals

$$\frac{\partial \phi}{\partial x} = W^e_x \phi \qquad (7.2\text{–}13)$$

$$W^e_x = \frac{\partial W^e}{\partial x} \qquad (7.2\text{–}14)$$

The diffusion matrix may now be expressed as:

$$\left[A^{diffusion}_e\right] \;=\; \int W^e_x\;\Gamma_\phi\;W^e_x + W^e_y\;\Gamma_\phi W^e_y + W^e_z\;\Gamma_\phi\;W^e_z\;\; d(vol) \qquad (7.2\text{–}15)$$

Source Terms

The evaluation of the source terms consists of merely multiplying the source terms as depicted in Figure 7.2–1 by the weighting function and integrating over the volume.

$$S^e{}_\phi \;=\; \int W^e S_\phi\;\; d(vol) \qquad (7.2\text{–}16)$$

7.2.2 Segregated Solution Algorithm

Each degree of freedom is solved in sequential fashion. The equations are coupled, so that each equation is solved with intermediate values of the other degrees of freedom. The process of solving all the equations in turn and then updating the properties is called a global iteration. Before showing the entire global iteration structure, it is necessary to see how each equation is formed.

The preceding section outlined the approach for every equation except the pressure equation, which comes from the segregated velocity–pressure solution algorithm. In this approach, the momentum equation is used to generate an expression for the velocity in terms of the pressure gradient. This is used in the continuity equation after it has been integrated by parts.

The incompressible algorithm is a special case of the compressible algorithm. The change in the product of density and velocity from iteration to the next is approximating by considering the changes separately through a linearization process. Denoting by the superscript * values from the previous iteration, in the x direction, for example, results:

$$\varrho V_x = \varrho V_x^* + \varrho^* V_x - \varrho^* V_x^* \tag{7.2–17}$$

The continuity equation becomes:

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \frac{\partial(\rho^* V_x)}{\partial x} + \frac{\partial(\rho V_x^*)}{\partial x} + \frac{\partial(\rho^* V_y)}{\partial y} + \frac{\partial(\rho V_y^*)}{\partial y} \\ &\frac{\partial(\rho^* V_z)}{\partial z} + \frac{\partial(\rho V_z^*)}{\partial z} - \frac{\partial(\rho^* V_x^*)}{\partial x} - \frac{\partial(\rho^* V_y^*)}{\partial y} - \frac{\partial(\rho^* V_z^*)}{\partial z} = 0 \end{aligned} \tag{7.2–18}$$

The transient term in the continuity equation can be expressed in terms of pressure immediately by employing the ideal gas relationship:

$$\int W^e \frac{\partial \rho}{\partial t} \, d(vol) = \frac{\partial}{\partial t} \int W^e \frac{P}{RT} \, d(vol) \tag{7.2–19}$$

The backward differencing process is then applied directly to this term.

Application of Galerkin's method to the remaining terms yields:

$$\begin{aligned} &\int W \left[\frac{\partial(\rho^* V_x)}{\partial x} + \frac{\partial(\rho^* V_y)}{\partial y} + \frac{\partial(\rho^* V_z)}{\partial z} \right] d(vol) \\ &\quad + \int W \left[\frac{\partial(\rho V_x^*)}{\partial x} + \frac{\partial(\rho V_y^*)}{\partial y} + \frac{\partial(\rho V_z^*)}{\partial z} \right] d(vol) \\ &\quad - \int W \left[\frac{\partial(\rho^* V_x^*)}{\partial x} + \frac{\partial(\rho^* V_y^*)}{\partial y} + \frac{\partial(\rho^* V_z^*)}{\partial z} \right] d(vol) = 0 \end{aligned} \tag{7.2–20}$$

There are thus three groups of terms. In the first group, terms with the derivatives of the unknown new velocities must be integrated by parts to remove the derivative. The integration by parts of just these terms becomes:

$$\int W\left[\frac{\partial(\rho^{*}V_x)}{\partial x}+\frac{\partial(\rho^{*}V_y)}{\partial y}+\frac{\partial(\rho^{*}V_z)}{\partial z}\right]d(\text{vol})$$

$$=\int W\left[\rho^{*}V_x+\ \rho^{*}V_y+\rho^{*}V_z\right]d(\text{area}) \qquad (7.2\text{–}21)$$

$$-\int\left[\left(\rho^{*}V_x\right)\frac{\partial W}{\partial x}+\left(\rho^{*}V_y\right)\frac{\partial W}{\partial y}+\left(\rho^{*}V_z\right)\frac{\partial W}{\partial z}\right]d(\text{vol})$$

Illustrating with the x direction, the unknown densities in the second group expressed in terms of the pressures are:

$$\int W\frac{\partial}{\partial x}\left(\rho V_x^{*}\right)d(\text{vol})=\int\frac{W}{R}\frac{\partial}{\partial x}\left(V_x^{*}\frac{P}{T}\right)d(\text{vol}) \qquad (7.2\text{–}22)$$

In the third group, the values from the previous iteration are used to evaluate the integrals.

The next step is the derivation of an expression for the velocities in terms of the pressure gradient. When the momentum equations are solved, it is with a previous value of pressure. Write the algebraic expressions of the momentum equations assuming that the coefficient matrices consist of the transient, advection and diffusion contributions as before, and all the source terms are evaluated except the pressure gradient term.

$$AV_x=s_{\phi}-\sum_{e=1}^{E}W\left(\frac{\partial P}{\partial x}\right)^{e}d(\text{vol}) \qquad (7.2\text{–}23)$$

$$AV_y=s_{\phi}-\sum_{e=1}^{E}W\left(\frac{\partial P}{\partial y}\right)^{e}d(\text{vol}) \qquad (7.2\text{–}24)$$

$$AV_z=s_{\phi}-\sum_{e=1}^{E}W\left(\frac{\partial P}{\partial z}\right)^{e}d(\text{vol}) \qquad (7.2\text{–}25)$$

Each of these sets represents a system of N algebraic equations for N unknown velocities. It is possible, after the summation of all the element quantities, to show an expression for each velocity component at each node in terms of the velocities of its neighbors, the source terms which have been evaluated, and the pressure drop. Using the subscript “i” to denote the nodal equation, for i = 1 to N, where N is the number of fluid nodes:

$$V_{x_i} = \hat{V}_{x_i} - \frac{1}{a^x_{ii}} \int W\left(\frac{\partial p}{\partial x}\right) d(vol) \qquad (7.2\text{–}26)$$

$$V_{y_i} = \hat{V}_{y_i} - \frac{1}{a^y_{ii}} \int W\left(\frac{\partial p}{\partial y}\right) d(vol) \qquad (7.2\text{–}27)$$

$$V_{z_i} = \hat{V}_{z_i} - \frac{1}{a^z_{ii}} \int W\left(\frac{\partial p}{\partial z}\right) d(vol) \qquad (7.2\text{–}28)$$

where

$$\hat{V}_{x_i} = \frac{-\sum\limits_{j}^{j\neq i} a^x_{ij} V_{x_j} + S_x}{a^x_{ii}} \qquad (7.2\text{–}29)$$

$$\hat{V}_{y_i} = \frac{-\sum\limits_{j}^{j\neq i} a^y_{ij} V_{y_j} + S_y}{a^y_{ii}} \qquad (7.2\text{–}30)$$

$$\hat{V}_{z_i} = \frac{-\sum\limits_{j}^{j\neq i} a^z_{ij} V_{z_j} + S_z}{a^z_{ii}} \qquad (7.2\text{–}31)$$

and the a_{ij} represent the values in the x,y and z coefficient matrices for the three momentum equations.

For the purposes of this expression, the neighboring velocities for each node are considered as being known from the momentum equation solution. At this point, the assumption is made that the pressure gradient is constant over the element, allowing it to be removed from the integral. This means that only the weighting function is left in the integral, allowing a pressure coefficient to be defined in terms of the main diagonal of the momentum equations and the integral of the weighting function:

$$M_x = \frac{1}{a^x_{ii}} \sum_{e=1}^{N} W\ d(vol) \qquad (7.2\text{–}32)$$

$$M_y = \frac{1}{a^y_{ii}} \sum_{e=1}^{N} W \; d(vol) \qquad (7.2\text{–}33)$$

$$M_z = \frac{1}{a^z_{ii}} \sum_{e=1}^{N} W \; d(vol) \qquad (7.2\text{–}34)$$

Therefore, expressions for unknown nodal velocities have been obtained in terms of the pressure drop and a pressure coefficient.

$$V_x = \hat{V}_x - M_x \frac{\partial P}{\partial x} \qquad (7.2\text{–}35)$$

$$V_y = \hat{V}_y - M_y \frac{\partial P}{\partial y} \qquad (7.2\text{–}36)$$

$$V_z = \hat{V}_z - M_z \frac{\partial P}{\partial z} \qquad (7.2\text{–}37)$$

These expressions are used to replace the unknown velocities in the continuity equation to convert it into a pressure equation. The terms coming from the unknown velocities (replaced with the pressure gradient term) and with the unknown density (expressed in terms of the pressure) contribute to the coefficient matrix of the pressure equation while all the remaining terms will contribute to the forcing function.

The entire pressure equation can be written on an element basis, replacing the pressure gradient by the nodal pressures and the derivatives of the weighting function, putting all the pressure terms on the left hand side and the remaining terms on the right hand side (equation (7.2–38)).

$$
\begin{aligned}
&[P]^e \int \left[\frac{\partial W}{\partial x}\, \rho^* M_x \,\frac{\partial W}{\partial x} + \frac{\partial W}{\partial y}\, \rho^* M_y \,\frac{\partial W}{\partial y} + \frac{\partial W}{\partial z}\, \rho^* M_z \,\frac{\partial W}{\partial z} \right] d(\mathrm{vol})^e \\
&+ \int \frac{W}{R} \left[\frac{\partial}{\partial x} \left(V_x^* \frac{P}{T} \right) + \frac{\partial}{\partial y} \left(V_y^* \frac{P}{T} \right) + \frac{\partial}{\partial z} \left(V_z^* \frac{P}{T} \right) \right] d(\mathrm{vol})^e \\
&= \int \left[\frac{\partial W}{\partial x}\, \rho^* \hat{V}_x + \frac{\partial W}{\partial y}\, \rho^* \hat{V}_y + \frac{\partial W}{\partial z}\, \rho^* \hat{V}_z \right] d(\mathrm{vol})^e \qquad (7.2\text{–}38) \\
&+ \int W \left[\frac{\partial}{\partial x} \left(\rho^* V_x^* \right) + \frac{\partial}{\partial y} \left(\rho^* V_y^* \right) + \frac{\partial}{\partial z} \left(\rho^* V_z^* \right) \right] d(\mathrm{vol})^e \\
&- \int W \left[\rho^* V_x \right]^s d(\mathrm{area})^s - \int W \left[\rho^* V_y \right]^s d(\mathrm{area})^s - \int W \left[\rho^* V_z \right]^s d(\mathrm{area})^s
\end{aligned}
$$

It is in the development of the forcing function that the solution to the momentum equation comes into play: the “hat” velocities contribute to the source term of the pressure equation.

In the incompressible case, the second and fourth lines of the above equation disappear because the linearization defined in equation (7.2–17) is unnecessary. The second line is treated with the same advection routines that are used for the momentum equation.

The final step is the velocity update. After the solution for pressure equation, the known pressures are used to evaluate the pressure gradients. In order to ensure that a velocity field exists which conserves mass, the pressure term is added back into the “hat” velocities:

$$
V_x = \hat{V}_x - \frac{1}{a_{ii}^x} \int \left(W \frac{\partial W}{\partial x} \right) d(\mathrm{vol})^e \; [P]^e \qquad (7.2\text{–}39)
$$

$$
V_y = \hat{V}_y - \frac{1}{a_{ii}^y} \int \left(W \frac{\partial W}{\partial y} \right) d(\mathrm{vol})^e \; [P]^e \qquad (7.2\text{–}40)
$$

$$
V_z = \hat{V}_z - \frac{1}{a_{ii}^z} \int \left(W \frac{\partial W}{\partial z} \right) d(\mathrm{vol})^e \; [P]^e \qquad (7.2\text{–}41)
$$

The global iterative procedure is summarized by Table 7.2–2.

Table 7.2–2 Global Iteration Structure

Formulate – Solve VX Equation Approximately – Calculate $\hat{V}_x$
Formulate – Solve VY Equation Approximately – Calculate $\hat{V}_y$
Formulate – Solve VZ Equation Approximately – Calculate $\hat{V}_z$
Formulate Pressure Equation Using $\hat{V}_x,\ \hat{V}_y,\ \text{and}\ \hat{V}_z$
Solve Pressure Equation for PRES
Update Velocities based on $\hat{V}_x,\ \hat{V}_y,\ \hat{V}_z,$ and PRES
Formulate – Solve Energy Equation for TEMP
Solve Species Transport Equations
Update Temperature Dependent Properties
Solve Turbulence Equations for ENKE and ENDS
Update Effective Properties based on Turbulence Solution
Check Rate of Change of the Solution (Convergence Monitors)
End of Global Iteration

7.3 Volume of Fluid Method for Free Surface Flows

7.3.1 Overview

A free surface refers to an interface between a gas and a liquid where the difference in the densities between the two is quite large. Due to a low density, the inertia of the gas is usually negligible, so the only influence of the gas is the pressure acted on the interface. Hence, the region of gas need not be modeled, and the free surface is simply modeled as a boundary with constant pressure.

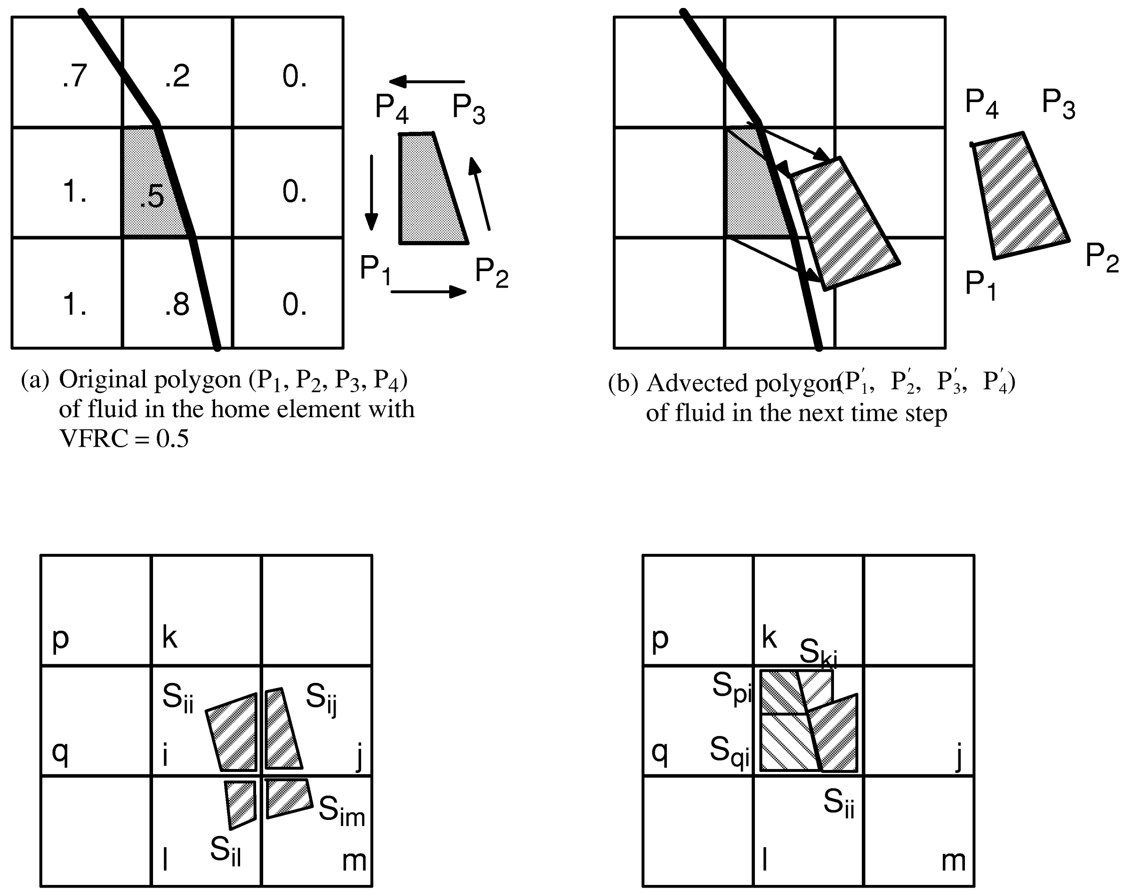
The volume of fluid (VOF) method (**FLDATA**,SOLU,VOF,TRUE) determines the shape and location of free surface based on the concept of a fractional volume of fluid. A unity value of the volume fraction (VFRC) corresponds to a full element occupied by the fluid (or liquid), and a zero value indicates an empty element containing no fluid (or gas). The VFRC value between zero and one indicates that the corresponding element is the partial (or surface) element. Further, the evolution of the free surface is computed through the following equation:

$$\frac{\partial F}{\partial t} + \vec{u} \cdot \nabla F = 0 \qquad (7.3\text{–}1)$$

Here, F denotes the volume fraction (or VFRC). In order to study complex flow problems, it is necessary to develop a VOF algorithm that is applicable to the unstructured mesh. Currently, the VOF capability is available only for quadrilateral elements in two dimensional planar and axisymmetric analyses.

7.3.2 CLEAR–VOF Advection

Here, CLEAR stands for Computational Lagrangian–Eulerian Advection Remap. This algorithm takes a new approach to compute the fluxes of fluid originating from a home element towards each of its immediate neighboring elements. Here, these fluxes are referred to as the VFRC fluxes. The idea behind the computation of the VFRC fluxes is to move the fluid portion of an element in a Lagrangian sense, and compute how much of the fluid remains in the home element, and how much of it passes into each of its neighboring elements. This process is illustrated in Figure 7.3–1 (a–d).



(a) Original polygon (P_1, P_2, P_3, P_4) of fluid in the home element with VFRC = 0.5

(b) Advected polygon(P'_1, P'_2, P'_3, P'_4) of fluid in the next time step

(c) Intersection of the advected polygon with the neighboring elements.

(d) Update the new area and the VFRC value for the home element.

Figure 7.3–1 A Typical Advection Step in the CLEAR–VOF Algorithm

First, the fluid portion inside each non–empty element is used to define a polygon in that element as shown in Figure 7.3–1 (a). If the element is full, the polygon of fluid coincides with the element. The vertices of this polygon are material points in the fluid flow. Each material point undergoes a Lagrangian displacement (ξ, η) which define the velocity components (V_x, V_y):

$$V_x = \frac{d\zeta}{dt} \tag{7.3–2}$$

$$V_y = \frac{d\eta}{dt} \tag{7.3–3}$$

After the velocity field is obtained through the normal Flotran solution procedure, the equations (7.3–2) and (7.3–3) can be used to compute the Lagrangian displacements:

$$\zeta = \int_{t}^{t+\delta t} V_x dt \tag{7.3–4}$$

$$\eta = \int_{t}^{t+\delta t} V_y dt \tag{7.3–5}$$

After the computation of the displacements for each vertex of the polygon, the new locations of these vertices can be obtained, as shown in Figure 7.3–1 (b). A portion of the new polygon of fluid will remains inside of the home element (S_{ii}), and several other parts will cross into the neighboring elements (S_{ij}, S_{il} and S_{im}) as illustrated in Figure 7.3–1 (c). The exact amount of fluid volume portions belonging to each element is determined by an algorithm for intersection of the advected polygon and the home element (or its immediate neighboring elements) with theoretical basis in computational geometry. For efficiency, algorithms are developed to compute the intersection of two convex polygons. The assumption of convexity holds by the grid generation characteristics for quadrilateral 2–D elements in ANSYS, and the advected polygons of fluid are maintained to convex shape through an automatic procedure for selecting the time step. In summary, this algorithm uses the following geometric calculations:

- Computation of the polygon area
- Relative location of a point with respect to a line segment
- Intersection of two line segments
- Relative location of a point with respect to a polygon
- Intersection of the two polygons

With the above geometric tools available, we can proceed to compute exactly how much of the advected fluid is still in the home element, and how much of it is located in the immediate neighboring elements. At this moment, a local conservation of the volume (or area) is checked, by comparing the volume of fluid in the initial polygon and the sum of all VFRC fluxes originating from the home element. A systematic error will occur if the time step is too large, where either the immediate neighbors of the home element fail to cover all the elements touched by the advected polygon, or the advected polygon lose the convexity. In either case, the time increment for VOF advection will be automatically reduced by half. This automatic reduction will continue until the local balance of volume is preserved.

After the advected polygons of fluid from all non–empty elements have been redistributed locally in the Eulerian fixed mesh, a sweep through all elements is necessary to update the volume fraction field. The new volume of fluid in each home element can be obtained by the sum of all VFRC fluxes originating from itself (S_{ii})and its immediate neighboring elements (S_{pi}, S_{qi} and S_{ki}), and the new volume fraction can simply obtained by dividing this sum by the volume of this home element as illustrated in Figure 7.3–1 (d).

7.3.3 CLEAR–VOF Reconstruction

In order to continue the VOF advection in the next time step, the new volume fraction is needed to reconstruct the new polygon of fluid in each non–empty element. In the present implementation, a piecewise linear reconstruction method is used where the interface is reconstructed as a line segment inside each partial element. Since the polygon of fluid coincides with the home element for every full element, there is no need for interface reconstruction for full elements. This process is illustrated in Figure 7.3–1.

In order to combine the unstructured mesh capability of the CLEAR–VOF with a piecewise linear method, the following procedure has been adopted for the interface reconstruction:

- Store the local distribution of updated volume fraction field and mesh geometry. Here, local means the home element and its immediate neighbors.
- Compute the unit normal vector $\hat{n}$ to the interface line inside the home element as the unit gradient vector of the volume fraction field in its neighborhood
- The equation of line in the home element is $g(\vec{x}) = \hat{n} \cdot \vec{x} + c = 0$. Once the unit vector $\hat{n}$ is found, the constant c is computed by requiring the volume fraction of the polygon of fluid delimited by the corresponding line interface to be equal to the given volume fraction for the home element.
- When a given value for c is computed, the volume fraction inside the home element is determined by constructing the polygon of fluid delimited by the line of equation inside the home element. It is thus necessary to retain the vertices of the home element inside the fluid, i.e., the vertices that verify $g(\vec{x}) > 0$, and the intersection points lie between the interface line and the edges of the home element.

In the present algorithm, the least squares gradient method has been chosen to compute the unit normal vector $\hat{n} = \nabla f / |\nabla f|$. This method is essentially independent of any mesh topology or dimensionality, and is thus able to handle any unstructured meshes. Further, the line constant c is obtained by solving an additional equation that imposes the conservation of fluid volume in the home element. The idea is that volume of the polygon of fluid, delimited inside the home element by the interface line, must correspond to the known VFRC value. The solution of this equation can be obtained iteratively by halving iteration of the interval [c_{min}, c_{max}]. The limits are found by allowing the interface line to pass through each of the home element vertices, computing the volume fraction and isolating the extreme cases F = 0 and F = 1.

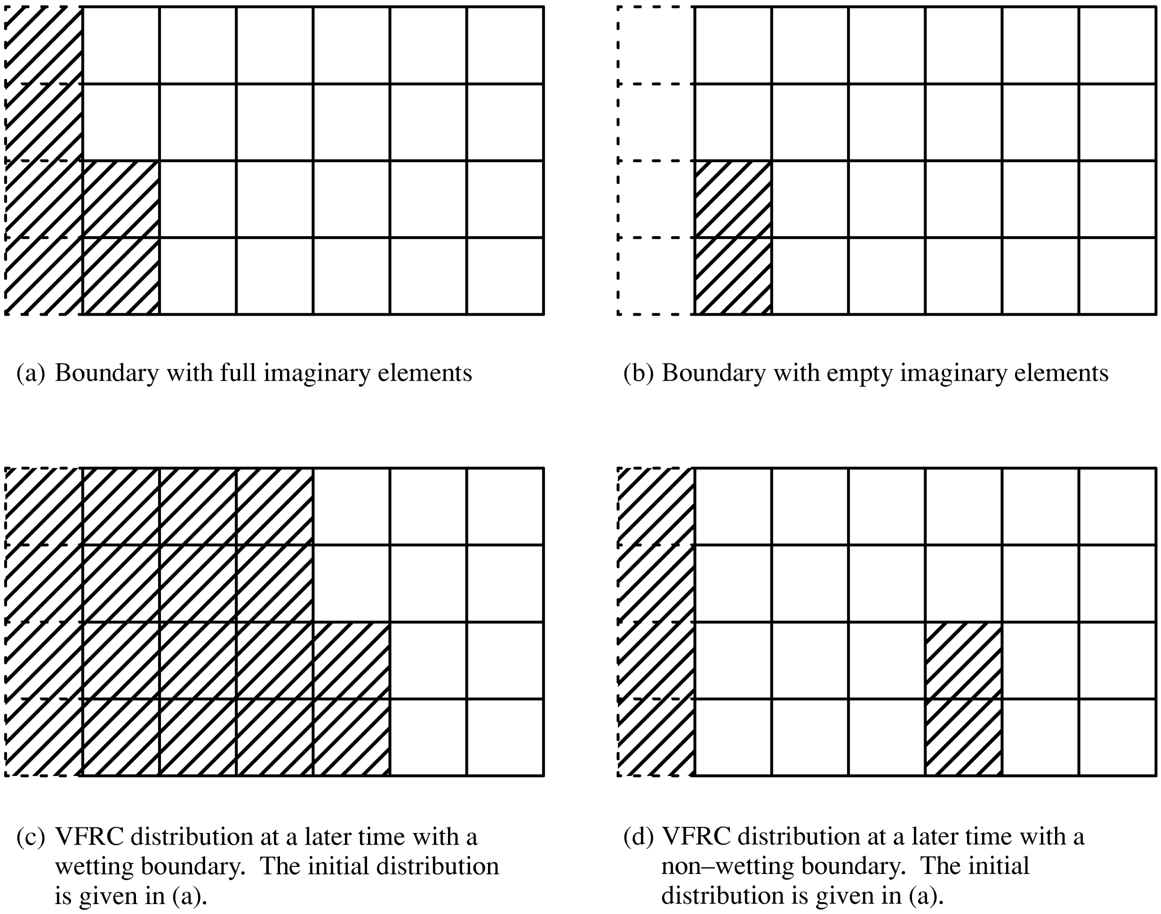
7.3.4 Treatment of Finite Element Equations

In a VOF (Volume of Fluid) analysis, each element can be identified as full, partially full, or empty. Full elements represent the fluid, and empty elements represent the void. Partial elements are regions of transition between the fluid and the void. In the present solution algorithm, the finite element equations are assembled only for partial and full elements, because empty elements have no effect on the motion of the fluid. The contributions of the full elements are treated in the usual manner as in other flow analyses, whereas those of the partial elements are modified to reflect the absence of fluid in parts of the elements.

In the solution algorithm, partial elements are reconstructed differently from the CLEAR–VOF reconstruction scheme. The nodes are moved towards the center of the element so that the reduced element preserves the same shape as the original element, and the ratio between the two is kept to be equal to the volume fraction of this partial element. The modified nodal coordinates are then used to evaluate the integration of the finite element equations over a reduced integration limit. It shall be noted that this modification is only intended for the evaluation of the finite element equations, and the actual spatial coordinates of the nodes are not changed.

For a VOF analysis, boundary conditions are required for boundary nodes that belong to at least one non–empty (partial or full) element. For boundary nodes belonging to only empty elements, on the other hand, the prescribed boundary conditions will remain inactive until those nodes are touched by fluid. Finally, boundary conditions are also applied to nodes that belong to at least one empty element and at least one non–empty element. These nodes represent the transition region between the fluid and the void. This free surface is treated as natural boundary conditions for all degrees of freedom except pressure. For the pressure, a constant value, set with the **FLDATA**,AMBV,PRES,value command, is imposed on the free surface.

In order to impose proper boundary conditions on the element–based volume fraction (VFRC), imaginary elements are created along the exterior boundary to act as neighbors to the elements forming the boundary. Two types of boundary conditions are applied on these imaginary elements. The imaginary elements can be specified as either full or empty depending on the imposed volume fraction value as shown in Figure 7.3–2 (a and b).



(a) Boundary with full imaginary elements

(b) Boundary with empty imaginary elements

(c) VFRC distribution at a later time with a wetting boundary. The initial distribution is given in (a).

(d) VFRC distribution at a later time with a non–wetting boundary. The initial distribution is given in (a).

Figure 7.3–2   Types of VFRC Boundary Conditions

Partial imaginary elements are not allowed on boundaries. These boundary volume fraction will serve as a neighbor value when determine the interface normal vector. For the full imaginary elements, a second boundary condition is specified to determine whether the fluid is advected into the computational domain. The boundary is then further identified as either wetting or non–wetting as shown in Figure 7.3–2 (c and d).

For the wetting boundary, the imaginary elements have to be full, and the fluid is advected into the domain. For the non–wetting boundary, the fluid or void can not be advected into the domain.

The two boundary VFRC loads can be set by either the **SFE** or **SFL** command. Use **SFE**,ELEM,Face No., VFRC,1,value to set the VFRC value and **SFE**,ELEM,Face No., VFRC,2,value to set the wetting status, or use **SFL**,Line,VFRC,VAL1I,,VAL2I to set the two loads at the same time (VAL1I is the VFRC value and VAL2I is the wetting status). Further, the initial VFRC loads can be set by the **ICE**,ELEM,VFRC,value command.

7.3.5 Treatment of Volume Fraction Field

In summary, the advection of the reconstructed polygon of fluid consists of the following steps:

1. Compute the new locations of the polygon vertices in the Lagrangian displacement step.
2. Determine the distribution of the advected fluid volume into the neighborhood using an algorithm for intersection of polygons.
3. Update the volume fraction at the new time step.

In the last step, the VFRC fluxes are regrouped to evaluate the total volume flowing into each home element. Since the volume fraction is just this volume divided by the volume of the home element, this evaluation of volume fraction is exact, and there exists no error in this step.

In the second step, the polygon of fluid at the new time level is only redistributed into its neighborhood, and no fluid shall be created or destroyed in this process. Therefore, the volume of fluid in the advected polygon shall be equal to the sum of all VFRC fluxes originating from this polygon. This conservation of the fluid volume will be violated only in two cases. The first one involves the failure of the polygon intersection algorithm. This will occur when the deformation of the advected polygon is too large during the Lagrangian step such that the convexity of the polygon is lost. The second one involves an incomplete coverage of the advected polygon by the immediate neighbors of the home element. In this case, some VFRC fluxes will flow into its far neighbors and will not be taken into account by the present algorithm. In either case, the time increment in the Lagrangian step will be reduced by half in order to reduce the Lagrangian deformation and the traveling distance of the advected polygon. This automatic reduction in time increment will continue until the local balance of fluid volume is preserved. You can also specify the number of VOF advection steps per solution step using the **FLDATA**,TIME,NTVF,value command.

In the Lagrangian step, the polygon of fluid undergoes a Lagrangian movement. The Lagrangian velocity is taken to be the same with the Eulerian velocity at a particular instance in time. The Lagrangian velocity is then used to calculate the displacements and the new locations of the polygon vertices. This new polygon is then used to intersect with the immediate neighbors of the home element in the next step. There do exist some potential problems in the numerical approximation of this algorithm. Consider a bulk of fluid flows along a no–slip wall emptying the elements behind it as time advances. In reality, however, there exist certain cases where the polygon may have two vertices lie on the no–slip wall during the reconstruction stage. In such cases, there will always a certain amount of volume left in the home element, which make it practically impossible to empty these wall elements. As time advances, the bulk of fluid may leave behind a row of partial elements rather empty elements. This phenomenon is usually referred to as the artificial formation and accumulation of droplets. In other words, a droplet is never reattached to the main fluid once it is formed. To eliminate those isolated droplets, the status of partial element's neighbors are always checked, and if necessary, a local adjustment will be performed. A partial element is reset to be empty if it is not adjacent to at least one full element. Similarly, a partial element is reset to be full if its immediate neighbors are all full elements to avoid an isolated partial element inside a bulk of fluid.

Another type of error introduced in the Lagrangian advection step is due to the imperfection of Eulerian velocity field. In the solution algorithm, the continuity equation is expressed in a Galerkin weak form. As a result, divergence–free condition is not satisfied exactly, and the error is usually in the same order with the discretization error. This error will further result in artificial compressibility of the polygon of fluid during the Lagrangian advection step, and thus introduce local and global imbalance in the fluid volume. Fortunately, both this type of error and that in the local adjustment of volume fraction field are very small compared to the total fluid volume. Unfortunately, the error due to the velocity divergence can accumulate exponentially as time advances. Hence a global adjustment is necessary to retain the global balance of the fluid volume. Currently, the volume fraction of partial elements are increased or decreased proportionally according to the global imbalance.

$$F_p = F_p + \frac{V_{imb}}{\sum_{p=1}^{N_p} F_p} F_p \tag{7.3–6}$$

where:

F_p = volume fraction of partial elements

N_p = total number of partial elements

V_{imb} = amount of the total volume imbalance = difference between the volume flowing across the external boundary (in – out) and the change of total volume inside the domain.

In the above practice, the volume fraction of a nearly full element may be artificially adjusted to an unphysical value greater than one, and will thus be reset to one. Although this global adjustment for partial elements introduces a numerical diffusion effect, it is believed that the benefit of global conservation of the fluid volume will certainly outweigh this effect. Hence, the global balance of the fluid volume is always checked, and if an imbalance occurs, it will adjust the volume fraction to enforce the global balance.

7.4 Fluid Solvers

The algorithm requires repeated solutions to the matrix equations during every global iteration. In some cases, exact solutions to the equations must be obtained, while in others approximate solutions are adequate. In certain situations, the equation need not be solved at all. It has been found that for the momentum equations, the time saved by calculating fast approximate solutions offsets the slightly slower convergence rates one obtains with an exact solution. In the case of the pressure equation, exact solutions are required to ensure conservation of mass. In a thermal problem with constant properties, there is no need to solve the energy equation at all until the flow problem has been converged.

To accomodate the varying accuracy requirements, two types of solvers are provided. Both types of solvers are iterative. The first is a sweeping method known as the Tri–Diagonal Matrix Algorithm (TDMA) and the second are semi–direct solvers known as conjugate direction methods. TDMA is used to obtain the approximate solution and the conjugate direction methods are used when exact solutions are needed.

The user has control over which method is applied to which degree of freedom through the **FLDATA** command: **FLDATA**,METH,dof,value

where: dof = degrees of freedom (VX,VY,VZ,PRES,TEMP,ENKE, or ENDS)

value = 1 – TDMA

= 2 – conjugate residual

= 3 – preconditioned conjugate residual method (for the case of the incompressible pressure equation, choice 3 means the preconditioned conjugate gradient is used)

The TDMA method is described in detail in Patankar(182). The method consists of breaking the problem into a series of tri–diagonal problems where any entries outside the tri–diagonal portion are treated as source terms using the previous values. For a completely unstructured mesh, or an arbitrarily numbered system, the method reduces to the Gauss–Seidel iterative method.

Since it is considered an approximate method, TDMA is not executed to convergence. Rather, the user specifies how many TDMA sweeps should be executed: **FLDATA**,TDMA,dof,value

where: dof = degree of freedom

value = number of sweeps (iterations)

The conjugate direction methods are the conjugate gradient (for symmetric systems) method and the conjugate residual method (for non–symmetric systems). These are iterative methods used to attempt an exact solution to the equation of interest. The conjugate gradient method is pre–conditioned with an incomplete Choleski decomposition and is used only for the pressure equation in incompressible flows. The sequential solution algorithm must allow space for a non–symmetric coefficient matrix for the momentum and energy equations. Only half this storage is required for the symmetric matrix and the other half is used to store the decomposition. The conjugate residual method can be used with or without preconditioning, the latter approach requiring significantly less computer memory. A convergence criterion and a maximum number of iterations is specified by the user: **FLDATA**,CONV,dof,value and **FLDATA**,MAXI,dof,value

where: dof = degree of freedom
CONV = implies value is the convergence criterion
MAXI = implies value is the maximum number of semi–direct iterations permitted.

The conjugate direction method develop a solution as a linear combination of orthogonal vectors. These vectors are generated one at a time during an iteration. In the case of the conjugate gradient method, the symmetry of the coefficient matrix and the process generating the vectors ensures that each one is automatically orthogonal to all of the previous vectors. In the non–symmetric case, the new vector at each iteration is made orthogonal to some user specified number of previous vectors (search directions). The user has control of the number: **FLDATA**,SRCH,dof,value

where: dof = degree of freedom
value = number of search directions requested.

More information on the conjugate directions is available from Hestenes and Stiefel(183) , Reid(184), and Elman(185).

7.5 Overall Convergence and Stability

7.5.1 Convergence

The fluid problem is non–linear in nature and convergence is not guaranteed. Some problems are transient in nature, and a steady state algorithm may not yield satisfactory results. Instabilities can result from a number of factors: the matrices may have poor condition numbers because of the finite element mesh or very large gradients in the actual solution. The fluid phenomena being observed could be unstable in nature.

Overall convergence of the segregated solver is measured through the convergence monitoring parameters. A convergence monitor is calculated for each degree of freedom at each global iteration. It is loosely normalized rate of change of the solution from one global iteration to the next and is calculated for each DOF as follows:

$$\text{ConvMon} = \frac{\sum_{i=1}^{N} \left| \phi_i^k - \phi_i^{k-1} \right|}{\sum_{i=1}^{N} \left| \phi_i^k \right|} \qquad (7.5\text{–}1)$$

where: N = total number of finite element nodes

ϕ = degree of freedom

k = current global iteration number

It is thus the sum of the absolute value of the changes over the sum of the absolute values of the degree of freedom.

The user may elect to terminate the calculations when the convergence monitors for pressure and temperature reach very small values. The convergence monitors are adjusted with: **FLDATA**,TERM,FLOW,value or **FLDATA**,TERM,TEMP,value The default values are 1.0x10^{-8}. Reduction of the rate of change to these values is not guaranteed. In some cases the problem is too unstable and in others the finite element mesh chosen leads to solution oscillation.

7.5.2 Stability

Three techniques are available to slow down and stabilize a solution. These are relaxation, inertial relaxation, and artificial viscosity.

Relaxation

Relaxation is simply taking as the answer some fraction of the difference between the previous global iteration result and the newly calculated values. In addition to the degrees of freedom, relaxation can be applied to the laminar properties (which may be a function of temperature and, in the case of the density of a gas, pressure) and the effective viscosity and effective conductivity calculated through the turbulence equations. Denoting by ϕ_i the nodal value of interest, the expression for relaxation is as follows:

$$\phi_i^{new} = (1 - r^{\phi})\ \phi_i^{old} + r^{\phi}\phi_i^{calc} \tag{7.5–2}$$

where: r^{ϕ} = relaxation factor for the variable.

The user has control over the relaxation factors using **FLDATA**,RELX,variable,value where the variable is VX,VY,VZ,PRES,TEMP,ENKE,ENDS,DENS,COND,EVIS, or ECON and the value is between 0.0 and 1.0 inclusive.

Experience has shown that the default values of 0.5 (0.8 for temperature) are optimum for most problems. Values greater than 0.5 for VX,VY, and VZ, with rare exceptions, will cause instability and divergence.

Inertial Relaxation

Inertial relaxation is used to make a system of equations more diagonally dominant. It is similar to a transient solution. It is most commonly used in the solution of the compressible pressure equation and in the turbulence equations. It is only applied to the DOF.

The algebraic system of equations to be solved may be represented as, for i = 1 to the number of nodes:

$$a_{ii}\phi_i + \sum_{j \neq i} a_{ij}\phi_j = f_i \tag{7.5–3}$$

With inertial relaxation, the system of equations becomes:

$$(a_{ii} + A_{ii}^d)\ \phi_i + \sum_{j \neq i} a_{ij}\ \phi_j = f_i + A_{ii}^d\ \phi_i^{old} \tag{7.5–4}$$

where: $A_{ii}^d = \dfrac{\int \rho W d(vol)}{B^{rf}}$

B^{rf} = inertial relaxation factor (input as VALUE on the **FLDATA26** command)

At convergence, ϕ_i^{old} (i.e. the value of the ϕ_i from the previous global iteration) and ϕ_i will be identical, so the same value will have been added to both sides of the equation.

This form of relaxation is always applied to the equations, but the default value of B^{rf} = 1.0x10^{15} effectively defeats it. The user has control over the inertial relaxation factor using **FLDATA26**,STAB,label,value, where label = MOME (momemtum),PRES (pressure), or TURB (turbulence) and value = B^{rf}. In practice, useful values of the inertial relaxation factor vary between 1.0 and 1.0x10^{-7}.

Artificial Viscosity

Artificial viscosity is a stabilization technique that has been found useful in compressible problems and incompressible problems involving distributed resistance. The technique serves to increase the diagonal dominance of the equations where the gradients in the momentum solution are the highest. Artificial viscosity enters the equations in the same fashion as the fluid viscosity. The additional terms are:

$$R_x = \mu_a \frac{\partial}{\partial x}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right) \tag{7.5–5}$$

$$R_y = \mu_a \frac{\partial}{\partial y}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right) \tag{7.5–6}$$

$$R_z = \mu_a \frac{\partial}{\partial z}\left(\frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z}\right) \tag{7.5–7}$$

where: μ_a = artificial viscosity

This formulation is slightly different from that of Harlow and Amsden(180) in that here μ_a is adjustable using **FLDATA**,STAB,VISC,value.

In each of the momentum equations, the terms resulting from the discretization of the derivative of the velocity in the direction of interest are additions to the main diagonal, while the terms resulting from the other gradients are added as source terms.

Note that since the artificial viscosity is multiplied by the divergence of the velocity, (zero for an incompressible fluid), it should not impact the final solution. For compressible flows, the divergence of the velocity is not zero and artificial viscosity must be regarded as a temporary convergence tool, to be removed for the final solution.

Useful values of artificial viscosity range from 1.0 down to 1.0x10^{-6}. In practice, its value should be less than 1000 times the effective viscosity for turbulent incompressible problems.

7.5.3 Residual File

One measure of how well the solution is converged is the magnitude of the nodal residuals throughout the solution domain. The residuals are calculated based on the "old" solution and the "new" coefficient matrices and forcing functions. Residuals are calculated for each degree of freedom (VX, VY, VZ, PRES, TEMP, ENKE, ENDS).

Denoting the DOF by ϕ, the matrix equation for the residual vector r may be written as follows:

$$[A^n_\phi]\ \{\phi^{n-1}\} - \{b^n_\phi\} = \{r_\phi\} \tag{7.5–8}$$

where the superscript refers to the global iteration number and the subscript associates the matrix and the forcing function with the degree of freedom ϕ.

The residuals provide information about where a solution may be oscillating.

The residual file (jobname.rdf) is only calculated if requested (**FLDATA**,OUTP,RESI,TRUE). It is read using the **FLREAD** command. After the **FLREAD** is executed, the residuals are accessed with the respective DOF labels, e.g. PRES then refers to the pressure residual.

The values at each node are normalized by the main diagonal value for that node in the coefficient matrix. This enables direct comparison between the value of the residual and value of the degree of freedom at the node.

7.5.4 Modified Inertial Relaxation

Similar to inertial relaxation, modified inertial relaxation (MIR) is used to make the system of equations more diagonally dominant. It is most commonly used to make the solution procedure by SUPG scheme more stable. The algebraic system of equations with modified inertial relaxation has the same form with equation (7.5–4), but the definition of the added diagonal term is different:

$$A^d_{ii} = B^{MIR} \int \left(\frac{\rho u}{h} + \frac{\Gamma}{h^2}\right) d(vol) \tag{7.5–9}$$

where:

ϱ = density

Γ = generalized diffusion coefficient

u = local velocity scale

h = local length scale

B^{MIR} = modified inertial relaxation factor (input as VALUE on the **FLDATA34**,MIR,label or **MSMIR**,Spec No. command).

In practice, useful values of B^{MIR} vary between 0.0 and 1.0.

7.6 Fluid Properties

Specific relationships are implemented for the temperature variation of the viscosity and thermal conductivity for both gases and liquids. These relationships were proposed by Sutherland and are discussed in White(181). The equation of state for a gas is assumed to be the ideal gas law. Density in a liquid may vary as a function of temperature through a polynomial. Fluid properties are isotropic. In addition to gas and liquid–type variations, non–Newtonian variations of viscosity are also included (Gartling(197) and Crochet et al(198)).

The relationships are:

7.6.1 Density

Constant: For the constant type, the density is:

$$\rho = \rho_N \tag{7.6–1}$$

where: ρ = density

ρ_N = nominal density (input as value on **FLDATA**,NOMI,DENS command)

Liquid: For the liquid type, the density is:

$$\rho = \rho_N + C_2^\rho \left(T - C_1^\rho\right) + C_3^\rho \left(T - C_1^\rho\right)^2 \tag{7.6–2}$$

where: P = absolute pressure

T = absolute temperature

C_1^ρ = first density coefficient (input as value on **FLDATA**,COF1,DENS command)

= absolute temperature at which $\rho = \rho_N \left(\text{if } C_2^\rho = P\right)$

C_2^ρ = second density coefficient (input as value on **FLDATA**,COF2,DENS command)

C_3^ρ = third density coefficient (input as value on **FLDATA**,COF3,DENS command)

Gas: For the gas type, the density is:

$$\rho = \rho_N \frac{P}{C_2^\rho} \frac{C_1^\rho}{T} \tag{7.6–3}$$

Table: For the table type, you enter density data as a function of temperature, via the **MPTEMP** and **MPDATA** commands.

User–Defined Density: In recognition of the fact that the density models described above can not satisfy the requests of all users, a user–programmable subroutine (UserDens) is also provided with access to the following variables: position, time, pressure, temperature, etc. See the *ANSYS Programmer's Manual* and Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide* for information about user written subroutines.

7.6.2 Viscosity

Constant: For the constant type, the viscosity is:

$$\mu = \mu_N \tag{7.6–4}$$

where: μ = viscosity

μ_N = nominal viscosity (input as value on **FLDATA**,NOMI,VISC command)

Liquid: For the liquid type, the viscosity is:

$$\mu = \mu_N \, e^A \tag{7.6–5}$$

where: $A = C_2^\mu \left(\frac{1}{T} - \frac{1}{C_1^\mu} \right) + C_3^\mu \left(\frac{1}{T} - \frac{1}{C_1^\mu} \right)^2$

C_1^μ = first viscosity coefficient (input as value on **FLDATA**,COF1,VISC command)

= absolute temperature at which $\mu = \mu_N$

C_2^μ = second viscosity coefficient (input as value on **FLDATA**,COF2,VISC command)

C_3^μ = third viscosity coefficient (input as value on **FLDATA**,COF3,VISC command)

Gas: For the gas type, the viscosity is:

$$\mu = \mu_N \left(\frac{T}{C_1^\mu} \right)^{1.5} \left(\frac{C_1^\mu + C_2^\mu}{T + C_2^\mu} \right) \tag{7.6–6}$$

In addition for non–Newtonian flows, additional viscosity types are available. A viscosity type is considered non–Newtonian if it displays dependence on the velocity gradient.

Power Law: For the power law model, the viscosity is:

$$\mu = \begin{cases} \mu_o K\ D^{n-1} & \text{for } D > D_o \\ \mu_o K\ D_o^{n-1} & \text{for } D \le D_o \end{cases} \qquad (7.6\text{–}7)$$

where: μ_o = nominal viscosity (input as value on **FLDATA**,NOM1,VISC command)

K = consistency index (input as value on **FLDATA**,COF2,VISC command)

D = $\sqrt{I_2}$

D_o = cutoff value for D (input as value on **FLDATA**,COF1,VISC command)

n = power (input as value on **FLDATA**,COF3,VISC command)

I_2 = second invariant of strain rate tensor

$= \frac{1}{2} \sum_i \sum_j L_{ij}\ L_{ij}$

L_{ij} = $\frac{1}{2}\left(U_{i,j} + U_{j,i}\right)$

$U_{i,j}$ = ith velocity component gradient in jth direction

This relationship is used for modeling polymers, blood, rubber solution, etc. For blood the following values are encountered in literature: μ_o is 0.35 poise, D_o is 226.5 sec^{-1}, n is 0.61, and K is 1 $\text{sec}^{0.39}$. The units of K depend on the value of n.

Carreau Model: For the Carreau Model, the viscosity is:

$$\mu = \mu_\infty + (\mu_o - \mu_\infty)\left(1 + (\lambda\ D)^2\right)^{\frac{n-1}{2}} \qquad (7.6\text{–}8)$$

μ_∞ = viscosity at infinite shear rate (input as value on **FLDATA**,COF1,VISC command)

μ_o = viscosity at zero shear rate (input as value on **FLDATA**,NOM1,VISC command)

λ = time constant (input as value on **FLDATA**,COF2,VISC command)

n = power (input as value on **FLDATA**,COF3,VISC command)

Typically the fluid viscosity behaves like a Power Law model for intermediate values of shear rate while remaining bounded for zero/infinite shear rates. This model removes some of the deficiencies associated with the Power Law model. The fluid is assumed to have lower and upper bounds on the viscosity. For blood the following values are encountered in literature: μ_o is 0.56 poise, μ_∞ is 0.0345 poise, λ is 3.313 sec., and n is 0.3568.

Bingham Model: For the “ideal” Bingham model, the viscosity is:

$$\mu \begin{cases} \mu_o + G/D & \text{if } \tau \geq G \qquad (7.6\text{–}9) \\ \infty & \text{if } \tau < G \qquad (7.6\text{–}10) \end{cases}$$

where:
μ_o = plastic viscosity
G = yield stress (plastic stress)
τ = stress level = $\sqrt{\frac{1}{2} \sum_i \sum_j \tau_{ij} \tau_{ij}}$
τ_{ij} = extra stress on ith face in the jth direction

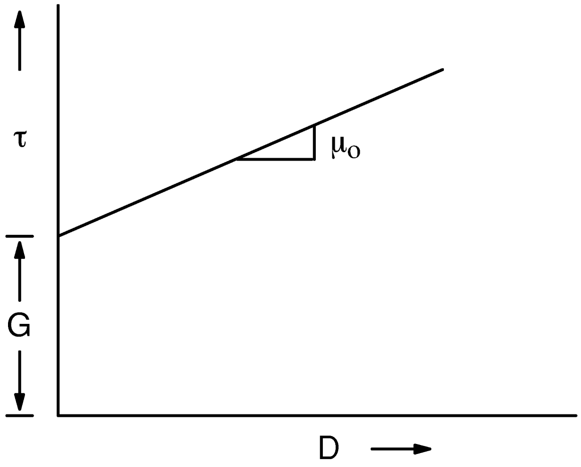


Figure 7.6–1 Stress vs. strain rate relationship for an “ideal” Bingham model

Figure 7.6–1 shows the stress–strain rate relationship.

So as long as the stress is below the plastic level, the fluid behaves as a rigid body. When the stress exceeds the plastic level the additional stress is proportional to the strain rate, i.e., the behavior is Newtonian. Numerically, it is difficult to model. In practice it is modelled as a “bi–viscosity” model:

$$\mu = \begin{cases} \mu_o + G/D & \text{if } D > \dfrac{G}{\mu_r - \mu_o} \\ \mu_r & \text{if } D \leq \dfrac{G}{\mu_r - \mu_o} \end{cases} \qquad (7.6\text{–}11)$$

where: μ_o = plastic viscosity (input as value on **FLDATA**,NOMI,VISC command)

G = yield stress (input as value on **FLDATA**,COF1,VISC command)

μ_r = Newtonian viscosity (input as value on **FLDATA**,COF2,VISC command)

Figure 7.6–2 shows the stress–strain rate relationship for the “bi–viscosity” Bingham model.

μ_r is chosen to at least an order of magnitude larger than μ_o. Typically μ_r is approximately 100 μ_o in order to replicate true Bingham fluid behavior. Bingham model can be used to model slurries and pastes.

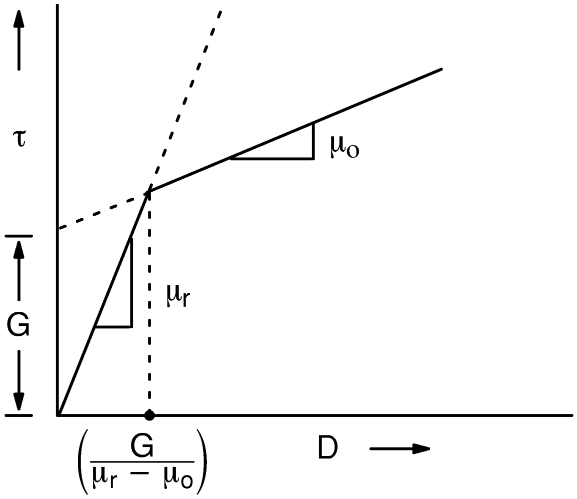


Figure 7.6–2 Stress vs. strain rate relationship for a “bi–viscosity” Bingham model

Table: For the table type, you enter viscosity data as a function of temperature, via the **MPTEMP** and **MPDATA** commands.

User–Defined Viscosity: In recognition of the fact that the viscosity models described above can not satisfy the requests of all users, a user–programmable subroutine (UserVisLaw) is also provided with access to the following variables: position, time, pressure, temperature, velocity component, velocity gradient component. See the *ANSYS Programmer’s Manual* and Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide* for information about user written subroutines.

7.6.3 Thermal Conductivity

Constant: For the constant type, the conductivity is:

$$K = K_N \tag{7.6–12}$$

K = conductivity

K_N = nominal conductivity (input as value on **FLDATA**,NOMI,COND command)

Liquid: For a liquid type, the conductivity is:

$$K = K_N \, e^B \tag{7.6–13}$$

where: $B = C_2^K \left(\frac{1}{T} - \frac{1}{C_1^K} \right) + C_3^K \left(\frac{1}{T} - \frac{1}{C_1^K} \right)^2$

C_1^K = first conductivity coefficient (input as value on **FLDATA**,COF1,COND command)

= absolute temperature at which K = K_N

C_2^K = second conductivity coefficient (input as value on **FLDATA**,COF2,COND command)

C_3^K = third conductivity coefficient (input as value on **FLDATA**,COF3,COND command)

Gas: For a gas type, the conductivity is:

$$K = K_N \left(\frac{T}{C_1^K} \right)^{1.5} \left(\frac{C_1^K + C_2^K}{T + C_2^K} \right) \tag{7.6–14}$$

Table: For the table type, you enter conductivity data as a function of temperature, via the **MPTEMP** and **MPDATA** commands.

User–Defined Conductivity: In recognition of the fact that the conductivity models described above can not satisfy the requests of all users, a user–programmable subroutine (UserCond) is also provided with access to the following variables: position, time, pressure, temperature, etc. See the *ANSYS Programmer's Manual* and Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide* for information about user written subroutines.)

7.6.4 Specific Heat

Constant: For the constant type, the specific heat is:

$$C_p = C_{pN} \tag{7.6–15}$$

where: C_{pN} = nominal specific heat

Table: For the table type, you specify specific heat data as a function of temperature via the **MPTEMP** and **MPDATA** commands.

User–Defined Specific Heat: In recognition of the fact that the specific heat models described above can not satisfy the requests of all users, a user–programmable subroutine (UserSpht) is also provided with access to the following variables: position, time, pressure, temperature, etc. See the *ANSYS Programmer's Manual* and Chapter 6 of the *ANSYS Advanced Analysis Techniques Guide* for information about user written subroutines.)

7.6.5 Multiple Species Property Options

For multiple species problems, the bulk properties can be calculated as a combination of the species properties by appropriate specification of the bulk property type. Choices are composite mixture, available for the density, viscosity, thermal conductivity, specific heat and composite gas, available only for the density.

Composite Mixture: For the composite mixture, (**FLDATA**,PROT,property,CMIX), each of the properties is a combination of the species properties:

$$\alpha_{bulk} = \sum_{i=1}^{N} Y_i \alpha_i \qquad (7.6\text{–}16)$$

where: α_{bulk} = bulk density, viscosity, conductivity or specific heat

α_i = values of density, viscosity, conductivity or specific heat for each of the species

Composite Gas: For a composite gas (**FLDATA**,PROT,DENS,CGAS), the bulk density is calculated as a function of the ideal gas law and the molecular weights and mass fractions.

$$\rho = \frac{P}{R\ T \sum_{i=1}^{N} \frac{Y_i}{M_i}} \qquad (7.6\text{–}17)$$

where: R = universal gas constant (input as UGAS on **MSDATA** command)

M_i = molecular weights of each species (input as MOLWT on **MSSPEC** command)

The most important properties in simulating species transport are the mass diffusion coefficient and the bulk properties. Typically, in problems with dilute species transport, the global properties will not be affected by the dilute species and can be assumed to be dependent only on the temperature (and pressure for gas density).

7.7 Derived Quantities

The derived quantities are total pressure, pressure coefficient, mach number, stream function, the wall parameter y–plus, and the wall shear stress. These quantities are calculated from the nodal unknowns and stored on a nodal basis. In parentheses are the labels assigned to the derived quantities.

7.7.1 Mach Number (MACH)

The Mach number is ratio of the speed of the fluid to the speed of sound in that fluid. Since the speed of sound is a function of the equation of state of the fluid, it can be calculated for a gas regardless of whether or not the compressible algorithm is used.

$$M = \frac{|V|}{(\gamma\ R\ T)^{1/2}} \qquad (7.7\text{–}1)$$

where:
M = Mach number (output as MACH)
γ = ratio of specific heats
$|V|$ = magnitude of velocity
R = ideal gas constant
T = absolute temperature

7.7.2 Total Pressure (PTOT)

The calculation differs, depending on whether the compressible option has been activated via **FLDATA**,SOLU,COMP,(true or false).

Compressible:

$$P_{tot} = (P + P_{ref}) \left(1 + \frac{\gamma - 1}{2}\ M^2\right)^{\frac{\gamma}{\gamma - 1}} - P_{ref} \qquad (7.7\text{–}2)$$

Incompressible:

$$P_{tot} = P + \frac{1}{2}\ \rho\ |V|^2 \qquad (7.7\text{–}3)$$

where:
P_{tot} = total pressure (output as PTOT)
P = relative pressure

P_{ref} = reference pressure
ϱ = density

7.7.3 Pressure Coefficient (PCOE)

The calculation is the same for compressible and incompressible cases.

$$P_{coef} = \frac{2\left(P - P_f\right)}{\rho_f V_f^2} \tag{7.7–4}$$

where: P_{coef} = pressure coefficient (output as PCOEF)
subscript f = freestream conditions

7.7.4 Y–Plus and Wall Shear Stress (YPLU and TAUW)

These quantities are part of the turbulence modeling of the wall conditions. First,

$$\frac{V_{tan}}{\sqrt{\frac{\tau_w}{\rho}}} = \frac{1}{\kappa}\ \ln\left(\frac{E\delta\rho}{\mu}\ \sqrt{\frac{\tau_w}{\rho}}\right) \tag{7.7–5}$$

where: μ = viscosity
δ = distance of the near wall node from the wal
V_{tan} = velocity at the near wall node parallel to the wall
E = constant in the turbulence model (defaults to 9.0)
κ = constant in the turbulence model (defaults to 0.4)
τ_w = wall shear stress

Then

$$y^+ = \delta\,\frac{\rho}{\mu}\ \sqrt{\frac{\tau_w}{\rho}} \tag{7.7–6}$$

where: y^+ = nondimensional distance from the wall (output as YPLUS)

7.7.5 Stream Function (STRM)

The stream function is computed for 2–D structures and is defined by way of its derivatives:

Cartesian Geometry

$$\frac{\partial\psi}{\partial x} = -\ \rho V_y \tag{7.7–7}$$

$$\frac{\partial\psi}{\partial y} = \rho V_x \tag{7.7–8}$$

Axisymmetric Geometry (about x)

$$\frac{\partial\psi}{\partial x} = y\rho V_y \tag{7.7–9}$$

$$\frac{\partial\psi}{\partial y} = -\ y\rho V_x \tag{7.7–10}$$

Axisymmetric Geometry (about y)

$$\frac{\partial\psi}{\partial x} = -\ x\rho V_y \tag{7.7–11}$$

$$\frac{\partial\psi}{\partial y} = x\rho V_x \tag{7.7–12}$$

Polar Coordinates

$$\frac{\partial\psi}{\partial r} = -\ \rho V_\theta \tag{7.7–13}$$

$$\frac{\partial\psi}{\partial \theta} = r\rho V_r \tag{7.7–14}$$

where:

ψ = stream function (output quantity STRM)
x, y = global Cartesian coordinates
r = radial coordinate (= $x^2 + y^2$)
θ = circumferential coordinate
V_x, V_y = global Cartesian velocity components
V_r, V_θ = polar velocity components

The stream function is zero at points where both V_x and V_y are zero. Thus, a zero value of the stream function would bound a recirculation region.

Chapter 8
Acoustics

ANSYS Theory Reference
Chapter 8 – Table of Contents

8.1 Acoustic Fluid Fundamentals

8.1.1 Governing Equations

In acoustical fluid–structure interaction problems, the structural dynamics equation needs to be considered along with the Navier–Stokes equations of fluid momentum and the flow continuity equation. The discretized structural dynamics equation can be formulated using the structural elements as shown in equation (17.2–1). The fluid momentum (Navier–Stokes) and continuity equations (equations (7.1–1) and (7.1–6) through (7.1–8)) are simplified to get the acoustic wave equation using the following assumptions (Kinsler(84)):

1. The fluid is compressible (density changes due to pressure variations).
2. The fluid is inviscid (no viscous dissipation).
3. There is no mean flow of the fluid.
4. The mean density and pressure are uniform throughout the fluid.

The acoustic wave equation is given by:

$$\frac{1}{c^2}\frac{\partial^2 P}{\delta t^2} - \nabla^2 P = 0 \tag{8.1–1}$$

where:
c = speed of sound $\left(\sqrt{k/\rho_o}\right)$ in fluid medium (input as SONC on **MP** command)
ϱ_o = mean fluid density (input as DENS on **MP** command)
k = bulk modulus of fluid
P = acoustic pressure (=P(x,y,z,t))
t = time

Since the viscous dissipation has been neglected, equation (8.1–1) is referred to as the lossless wave equation for propagation of sound in fluids. The discretized structural equation (17.2–1) and the lossless wave equation (8.1–1) have to be considered simultaneously in fluid–structure interaction problems. The lossless wave equation will be discretized in the next subsection followed by the derivation of the damping matrix to account for the dissipation at the fluid–structure interface. The fluid pressure acting on the structure at the fluid–structure interface will be considered in the final subsection to form the coupling stiffness matrix.

For harmonically varying pressure, i.e.

$$P = \bar{P}e^{j\omega t} \qquad (8.1\text{–}2)$$

where:

$\bar{P}$ = amplitude of the pressure

j = $\sqrt{-1}$

ω = 2πf

f = frequency of oscillations of the pressure

equation (8.1–1) reduces to the Helmholtz equation:

$$\frac{\omega^2}{c^2}\bar{P} + \nabla^2\bar{P} = 0 \qquad (8.1\text{–}3)$$

8.1.2 Discretization of the Lossless Wave Equation

The following matrix operators (gradient and divergence) are introduced for use in equation (8.1–1):

$$\nabla \cdot (\ ) = \{L\}^T = \lfloor \frac{\partial}{\partial x}\ \frac{\partial}{\partial y}\ \frac{\partial}{\partial z} \rfloor \qquad (8.1\text{–}4)$$

$$\nabla(\ ) = \{L\} \qquad (8.1\text{–}5)$$

Equation (8.1–1) is rewritten as follows:

$$\frac{1}{c^2}\frac{\partial^2 P}{\partial t^2} - \nabla \cdot \nabla P = 0 \qquad (8.1\text{–}6)$$

Using the notations given in equations (8.1–4) and (8.1–5), equation (8.1–6) becomes in matrix notation:

$$\frac{1}{c^2}\frac{\partial^2 P}{\partial t^2} - \{L\}^T(\{L\}P) = 0 \qquad (8.1\text{–}7)$$

The element matrices are obtained by discretizing the wave equation (8.1–7) using the Galerkin procedure (Bathe(2)). Multiplying equation (8.1–7) by a virtual change in pressure and integrating over the volume of the domain (Zienkiewicz(86)) with some manipulation yields:

$$\int_{vol}\frac{1}{c^2}\delta P\frac{\partial^2 P}{\partial t^2}\,d(vol) + \int_{vol}\left(\{L\}^T\delta P\right)(\{L\}P)\,d(vol) = \int_S \{n\}^T\delta P(\{L\}P)\,d(S) \qquad (8.1\text{–}8)$$

where:

vol = volume of domain

δP = a virtual change in pressure (=δP(x,y,z,t))

S = surface where the derivative of pressure normal to the surface is applied (a natural boundary condition)

{n} = unit normal to the interface S

In the fluid–structure interaction problem, the surface S is treated as the interface. For the simplifying assumptions made, the fluid momentum equations yield the following relationships between the normal pressure gradient of the fluid and the normal acceleration of the structure at the fluid–structure interface S (Zienkiewicz(86)):

$$\{n\} \cdot \{\nabla P\} = -\rho_o \{n\} \cdot \frac{\partial^2 \{U\}}{\partial t^2} \tag{8.1–9}$$

where: {U} = displacement vector of the structure at the interface

In matrix notation, equation (8.1–9) is given by:

$$\{n\}^T (\{L\}P) = -\rho_o \{n\}^T \left(\frac{\partial^2}{\partial t^2} \{U\} \right) \tag{8.1–10}$$

After substituting equation (8.1–10) into equation (8.1–8), the integral is given by:

$$\int_{vol} \frac{1}{c^2} \delta P \frac{\partial^2 P}{\partial t^2} \, d(vol) + \int_{vol} \left(\{L\}^T \delta P \right) (\{L\}P) \, d(vol) = -\int_S \rho_o \delta P \{n\}^T \left(\frac{\partial^2}{\partial t^2} \{U\} \right) d(S) \tag{8.1–11}$$

8.2 Derivation of Acoustics Fluid Matrices

Equation (8.1–11) contains the fluid pressure P and the structural displacement components u, v, and w as the dependent variables to solve. The finite element approximating shape functions for the spatial variation of the pressure and displacement components are given by:

$$P = \{N\}^T\{P_e\} \qquad (8.2\text{–}1)$$

$$U = \{N'\}^T\{U_e\} \qquad (8.2\text{–}2)$$

where: {N} = element shape function for pressure
{N′} = element shape function for displacements
{P_e} = nodal pressure vector
{U_e}={u_e},{v_e},{w_e} = nodal displacement component vectors

From equations (8.2–1) and (8.2–2), the second time derivative of the variables and the virtual change in the pressure can be written as follows:

$$\frac{\partial^2 P}{\partial t^2} = \{N\}^T\{\ddot{P}_e\} \qquad (8.2\text{–}3)$$

$$\frac{\partial^2}{\partial t^2}\{U\} = \{N'\}^T\{\ddot{U}_e\} \qquad (8.2\text{–}4)$$

$$\delta P = \{N\}^T\{\delta P_e\} \qquad (8.2\text{–}5)$$

Let the matrix operator {L} applied to the element shape functions {N} be denoted by:

$$[B] = \{L\}\{N\}^T \qquad (8.2\text{–}6)$$

Substituting equations (8.2–1) through (8.2–6) into equation (8.1–11), the finite element statement of the wave equation (8.1–1) is given by:

$$\int_{vol} \frac{1}{c^2}\{\delta P_e\}^T\{N\}\{N\}^T d(vol)\{\ddot{P}_e\} + \int_{vol} \{\delta P_e\}^T[B]^T[B]\ d(vol)\ \{P_e\}$$
$$+ \int_S \rho_o\ \{\delta P_e\}^T\{N\}\{n\}^T\{N'\}^T\ d(S)\ \{\ddot{U}_e\} = \{0\} \qquad (8.2\text{–}7)$$

where: $\{n\}$ = normal at the fluid boundary

Other terms are defined in Section 8.1. Terms which do not vary over the element are taken out of the integration sign. {δP$_e$} is an arbitrarily introduced virtual change in nodal pressure, and it can be factored out in equation (8.2–7). Since {δP$_e$} is not equal to zero, equation (8.2–7) becomes:

$$\frac{1}{c^2}\int_{vol} \{N\}\{N\}^T d(vol)\{\ddot{P}_e\} + \int_{vol} [B]^T[B]\ d(vol)\ \{P_e\}$$
$$+ \rho_o \int_S \{N\}\{n\}^T\ \{N'\}^T\ d(S)\ \{\ddot{U}_e\} = \{0\} \qquad (8.2\text{–}8)$$

Equation (8.2–8) can be written in matrix notation to get the discretized wave equation:

$$\left[M_e^P\right]\{\ddot{P}_e\} + \left[K_e^P\right]\{P_e\} + \rho_o\ [R_e]^T\ \{\ddot{U}_e\} = \{0\} \qquad (8.2\text{–}9)$$

where: $\left[M_e^P\right] = \frac{1}{c^2}\int_{vol} \{N\}\{N\}^T\ d(vol)$ = fluid mass matrix (fluid)

$\left[K_e^P\right] = \int_{vol} [B]^T[B]\ d(vol)$ = fluid stiffness matrix (fluid)

$\rho_o[R_e] = \rho_o \int_S \{N\}\{n\}^T\ \{N'\}^T\ d(S)$ = coupling mass matrix (fluid–structure interface)

8.3 Absorption of Acoustical Pressure Wave

8.3.1 Addition of Dissipation due to Damping at the Boundary

In order to account for the dissipation of energy due to damping, if any, present at the fluid boundary, a dissipation term is added to the lossless equation (8.1–1) to get (Craggs(85)):

$$\int_{vol} \delta P \frac{1}{c^2} \frac{\partial^2 P}{\partial t^2} d(vol) - \int_{vol} \delta P \ \{L\}^T(\{L\}P) \ d(vol)$$
$$+ \int_S \delta P \left(\frac{r}{\rho_o c}\right)\frac{1}{c} \frac{\partial P}{\partial t} d(S) = 0 \qquad (8.3\text{–}1)$$

where: r = characteristic impedance of the material at the boundary

Other terms are defined in Section 8.1.

Since it is assumed that the dissipation occurs only at the boundary surface S, the dissipation term in equation (8.3–1) is integrated over the surface S:

$$D = \int_S \delta P \ \left(\frac{r}{\rho_o c}\right)\frac{1}{c} \frac{\partial P}{\partial t} \ d(S) \qquad (8.3\text{–}2)$$

where: D = dissipation term

Using the finite element approximation for P given by equation (8.2–1):

$$D = \int_S \{\delta P_e\}^T\{N\} \left(\frac{r}{\rho_o c}\right)\frac{1}{c} \ \{N\}^T d(S) \left\{\frac{\partial P_e}{\partial t}\right\} \qquad (8.3\text{–}3)$$

Using the following notations:

$\beta = \frac{r}{\rho_o c}$ = boundary absorption coefficient (input as MU on **MP** command)

$\{\dot{P}_e\} = \left\{ \frac{\partial P_e}{\partial t} \right\}$

$\frac{\beta}{c}$ and $\{\delta P_e\}$ are constant over the surface of the element and can be taken out of the integration. Equation (8.3–3) is rewritten as:

$$D = \{\delta P_e\}^T \frac{\beta}{c} \int_S \{N\}\{N\}^T d(S) \; \{\dot{P}_e\} \qquad (8.3\text{–}4)$$

The dissipation term given by equation (8.3–4) is added to equation (8.2–7) to account for the energy loss at the absorbing boundary surface.

$$\left[C_e^P\right] \{\dot{P}_e\} = \frac{\beta}{c} \int_S \{N\}\{N\}^T d(S) \; \{\dot{P}_e\} \qquad (8.3\text{–}5)$$

where: $\left[C_e^P\right] = \frac{\beta}{c} \int_S \{N\}\{N\}^T d(S)$ = (fluid damping matrix)

Finally, combining equations (8.2–9) and (8.3–5), the discretized wave equation accounting for losses at the interface is given by:

$$\left[M_e^P\right] \{\ddot{P}_e\} + \left[C_e^P\right] \{\dot{P}_e\} + \left[K_e^P\right] \{P_e\} + \rho_o \, [R_e]^T \{\ddot{U}_e\} = 0 \qquad (8.3\text{–}6)$$

8.4 Acoustics Fluid – Structure Coupling

In order to completely describe the fluid–structure interaction problem, the fluid pressure load acting at the interface is now added to equation (17.2–1). This effect is included only if KEYOPT(2) = 0. So, the structure equation is rewritten here:

$$[M_e]\{\ddot{U}_e\} + [C_e]\{\dot{U}_e\} + [K_e]\{U_e\} = \{F_e\} + \{F_e^{pr}\} \qquad (8.4–1)$$

The fluid pressure load vector $\{F_e^{pr}\}$ at the interface S is obtained by integrating the pressure over the area of the surface:

$$\{F_e^{pr}\} = \int_S \{N'\}\ P\ \{n\}\ d(S) \qquad (8.4–2)$$

where:
$\{N'\}$ = shape functions employed to discretize the displacement components u, v, and w (obtained from the structural element)
$\{n\}$ = normal at the fluid boundary

Substituting the finite element approximating function for pressure given by equation (8.2–1) into (8.4–2):

$$\{F_e^{pr}\} = \int_S \{N'\}\ \{N\}^T\{n\}\ d(S)\ \{P_e\} \qquad (8.4–3)$$

By comparing the integral in equation (8.4–3) with the matrix definition of $\rho_o\ [R_e]^T$ in equation (8.2–9), it becomes clear that:

$$\{F_e^{pr}\} = [R_e]\ \{P_e\} \qquad (8.4–4)$$

where: $$[R_e]^T = \int_S \{N'\}\ \{N\}^T\{n\}\ d(S)$$

The substitution of equation (8.4–4) into equation (8.4–1) results in the dynamic elemental equation of the structure:

$$[M_e]\{\ddot{U}_e\} + [C_e]\{\dot{U}_e\} + [K_e]\{U_e\} - [R_e]\{P_e\} = \{F_e\} \qquad (8.4–5)$$

Equations (8.3–6) and (8.4–5) describe the complete finite element discretized equations for the fluid–structure interaction problem and are written in assembled form as:

$$\begin{bmatrix} [M_e] & [0] \\ [M^{fs}] & [M_e^p] \end{bmatrix} \begin{Bmatrix} \{\ddot{U}_e\} \\ \{\ddot{P}_e\} \end{Bmatrix} + \begin{bmatrix} [C_e] & [0] \\ [0] & [C_e^p] \end{bmatrix} \begin{Bmatrix} \{\dot{U}_e\} \\ \{\dot{P}_e\} \end{Bmatrix} + \begin{bmatrix} [K_e] & [K^{fs}] \\ [0] & [K_e^p] \end{bmatrix} \begin{Bmatrix} \{U_e\} \\ \{P_e\} \end{Bmatrix} = \begin{Bmatrix} \{F_e\} \\ \{0\} \end{Bmatrix} \tag{8.4–6}$$

where: $[M^{fs}] = \rho_o \, [R_e]^T$
$[K^{fs}] = - \, [R_e]$

For a problem involving fluid–structure interaction, therefore, the acoustic fluid element will generate all the submatrices with superscript p in addition to the coupling submatrices $\rho_o \, [R_e]^T$ and $[R_e]$. Submatrices without a superscript will be generated by the compatible structural element used in the model.

8.5 Acoustics Output Quantities

The pressure gradient is evaluated at the element centroid using the computed nodal pressure values.

$$\frac{\partial P}{\partial x} = \left\{\frac{\partial N}{\partial x}\right\}^T \{P_e\} \qquad (8.5\text{–}1)$$

$$\frac{\partial P}{\partial y} = \left\{\frac{\partial N}{\partial y}\right\}^T \{P_e\} \qquad (8.5\text{–}2)$$

$$\frac{\partial P}{\partial z} = \left\{\frac{\partial N}{\partial z}\right\}^T \{P_e\} \qquad (8.5\text{–}3)$$

where:

$\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y},$ and $\frac{\partial P}{\partial z}$ = gradients in x, y, and z directions, respectively, (output quantities PGX, PGY, and PGZ)

Other terms are defined in Sections 8.1 and 8.2.

The element fluid velocity is computed at the element centroid for the full harmonic analysis (**ANTYPE**,HARM with **HROPT**,FULL) by:

$$V_x = \frac{j}{\rho_o \omega} \frac{\partial P}{\partial x} \qquad (8.5\text{–}4)$$

$$V_y = \frac{j}{\rho_o \omega} \frac{\partial P}{\partial y} \qquad (8.5\text{–}5)$$

$$V_z = \frac{j}{\rho_o \omega} \frac{\partial P}{\partial z} \qquad (8.5\text{–}6)$$

where:

V_x, V_y, and V_z = components of the fluid velocity in the x, y, and z directions, respectively (output quantities VLX, VLY and VLZ)

ω = $2\pi f$

f = frequency of oscillations of the pressure wave (input on **HARFRQ** command)

j = $\sqrt{-1}$

The sound pressure level is computed by:

$$L_{sp} = 20 \log \left(\frac{P}{|P_{ref}|} \right) \tag{8.5–7}$$

where:

L_{sp} = sound pressure level (output as SOUND PR. LEVEL)

log = logarithm to the base 10

P_{ref} = reference pressure (input as PREF on **R** command, defaults to 20 x 10^{-6}

Chapter 9

ANSYS Theory Reference
Chapter 9 – Table of Contents

Chapter 10

ANSYS Theory Reference
Chapter 10 – Table of Contents

Chapter 11
Coupling

ANSYS Theory Reference
Chapter 11 – Table of Contents

11.0 Coupled Effects

11.0.1 Introduction

Coupled–field analyses are useful for solving problems where the coupled interaction of phenomena from various disciplines of physical science is significant. Several examples of this include: an electric field interacting with a magnetic field, a magnetic field producing structural forces, a temperature field influencing fluid flow, a temperature field giving rise to thermal strains and the usual influence of temperature dependent material properties. The latter two examples can be modeled with most non–coupled–field elements, as well as with coupled–field elements. The following elements have coupled–field capability:

SOLID5	3–D Coupled–Field Solid (Sections 5.2, 11.1, 14.5)
PLANE13	2–D Coupled–Field Solid (Sections 5.2, 5.3, 14.13)
FLUID29	2–D Acoustic Fluid (Sections 8.2, 14.29)
FLUID30	3–D Acoustic Fluid (Sections 8.2, 14.30)
PLANE53	2–D 8–Node Magnetic–Electric Solid (Sections 5.2, 5.3, 14.53)
SOLID62	3–D Coupled Magnetic–Structural Solid (Section 14.62)
FLUID66	Coupled Thermal–Flow Pipe (Section 14.66)
PLANE67	2–D Coupled Thermal–Electric Solid (Sections 11.2, 14.67)
LINK68	Coupled Thermal–Electric Line (Sections 11.2, 14.68)
SOLID69	3–D Coupled Thermal–Electric Solid (Sections 11.2, 14.69)
SOLID97	3–D Magnetic Solid (Section 14.97)
SOLID98	Tetrahedral Coupled–Field Solid (Sections 5.2, 11.1, 14.98)
CIRCU124	General Electric Circuit Element (Sections 5.4, 14.124)
FLUID141	2–D Fluid (Sections 7.2, 14.141)
FLUID142	3–D Fluid (Sections 7.2, 14.142)
SHELL157	Coupled Thermal–Electric Shell (Sections 11.2, 14.157)

There are certain advantages and disadvantages inherent with coupled–field formulations:

Advantages:

1. Allows for solutions to problems otherwise not possible with usual finite elements.

2. Simplifies modeling of coupled–field problems by permitting one element type to be used in a single analysis pass.

Disadvantages:

1. Increases wavefront (unless a segregated solver is used).
2. Inefficient matrix reformulation (if a section of a matrix associated with one phenomena is reformed, the entire matrix will be reformed).
3. Larger storage requirements.

11.0.2 Coupling

There are basically two methods of coupling distinguished by the finite element formulation techniques used to develop the matrix equations. These are illustrated here with two types of degrees of freedom ($\{X_1\}$, $\{X_2\}$):

1. Strong (simultaneous, full) coupling – where the matrix equation is of the form:

$$\begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{21}] & [K_{22}] \end{bmatrix} \begin{Bmatrix} \{X_1\} \\ \{X_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\} \\ \{F_2\} \end{Bmatrix} \tag{11.0–1}$$

   and the coupled effect is accounted for by the presence of the off–diagonal submatrices $[K_{12}]$ and $[K_{21}]$. This method provides for a coupled response in the solution after one iteration.

2. Weak (sequential) coupling – where the coupling in the matrix equation is shown in the most general form:

$$\begin{bmatrix} [K_{11}\ (\{X_1\},\{X_2\})] & [0] \\ [0] & [K_{22}\ (\{X_1\},\{X_2\})] \end{bmatrix} \begin{Bmatrix} \{X_1\} \\ \{X_2\} \end{Bmatrix} = \begin{Bmatrix} \{F_1\ (\{X_1\},\{X_2\})\} \\ \{F_2\ (\{X_1\},\{X_2\})\} \end{Bmatrix} \qquad (11.0–2$$

   and the coupled effect is accounted for in the dependency of $[K_{11}]$ and $\{F_1\}$ on $\{X_2\}$ as well as $[K_{22}]$ and $\{F_2\}$ on $\{X_1\}$. At least two iterations are required to achieve a coupled response.

The following is a list of the types of coupled–field analyses including types of coupling present in each:

Analysis Category		Coupling Method	Example Application
A.	Thermal Structural Analysis	W	High temperature turbine
B.,C.	Magneto–Structural Analysis	W	Solenoid, high energy magnets (MRI)
D.	Electro–Magnetic Analysis	S	Current fed massive conductors
E.,F.	Electro–Magnetic–Thermal–Structural Analysis	W	Direct current electro–mechanical devices in general
G.	Electro–Structural (Piezoelectric) Analysis	S	Transducers
H.	Thermo–Pressure Analysis	S, W	Piping networks
I.	Velocity–Thermo–Pressure Analysis	W	Fluid structure interaction
J.	Pressure–Structural (Acoustic) Analysis	S	Acoustics
K.	Thermo–Electric Analysis	W	High temperature electronics
L.	Magnetic–Thermal Analysis	W	Direct current transients: power interupts, surge protection
M.	Circuit–Magnetic	S	Circuit–fed solenoids, transformers, and motors
where: S = strong coupling, W = weak coupling			

The solution sequence follows the standard finite element methodology. Convergence is achieved when changes in all unknowns (i.e. DOF) and knowns, regardless of units, are less than the values specified on the **CNVTOL** command (except for FLUID141 and FLUID142). Some of the coupling described above is always or usually one–way. For example, in Category A, the temperatures affect the displacements of the structure by way of the thermal strains, but the displacements usually do not affect the temperatures.

The following descriptions of coupled phenomena will include:

1. Applicable element types
2. Basic matrix equation indicating coupling terms in bold print (FLUID141 and FLUID142 have coupling indicated with a different method)
3. Applicable analysis types, including the matrix and/or vector terms possible in each analysis type.

The nomenclature used on the following pages is given in Table 11.0–1 at the end of the section. In some cases, element KEYOPTs are used to select the DOF of the element. DOF will not be fully active unless the appropriate material properties are specified.

Some of the elements listed may not be applicable for a particular use as it may be only 1–D, whereas a 3–D element is needed (e.g. FLUID66).

A. Thermal–Structural Analysis (see Sections 2.2 and 6.2)

1. Element type: SOLID5, PLANE13, SOLID98
2. Matrix equation:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [C^t] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [K] & [0] \\ [0] & [K^t] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} \end{Bmatrix} \tag{11.0–3}$$

where:

$[K^t] = [K^{tb}] + [K^{tc}]$

$\{F\} = \{F^{nd}\} + \mathbf{\{F^{th}\}} + \{F^{pr}\} + \{F^{ac}\}$

$\{Q\} = \{Q^{nd}\} + \{Q^{g}\} + \{Q^{c}\}$

3. Analysis types: Static or Transient

B. Magneto–Structural Analysis (Vector Potential) (see Sections 5.2 and 11.1)

1. Element type: PLANE13, SOLID62
2. Matrix equation:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{A}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [C^m] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{A}\} \end{Bmatrix} + \begin{bmatrix} [K] & [0] \\ [0] & [K^m] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{A\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{J_i\} \end{Bmatrix} \tag{11.0–4}$$

where:

$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\} + \{F^{th}\} + \mathbf{\{F^{jb}\}} + \mathbf{\{F^{mx}\}}$

$\{J_i\} = \{J_i^{nd}\} + \{J^s\} + \{J^{pm}\}$

3. Analysis types: Static or Transient

C. Magneto–Structural Analysis (Scalar Potential)

1. Element type: SOLID5, SOLID98
2. Matrix equation:

$$\begin{bmatrix} [K] & [0] \\ [0] & [K^m] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{\phi\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{J_f\} \end{Bmatrix} \tag{11.0–5}$$

where:

$\{F\} = \{F^{nd}\} + \{F^{pr}\} + \{F^{ac}\} + \{F^{th}\} + \mathbf{\{F^{mx}\}}$

$\{J_f\} = \{J_f^{nd}\} + \{J^b\} + \{J^{pm}\}$

3. Analysis types: Static

D. Electro–Magnetic Analysis (see Sections 5.2 and 5.4)

1. Element type: PLANE13, PLANE53, SOLID97
2. Matrix equation:

$$\begin{bmatrix} [C^{AA}] & \mathbf{[C^{Av}]} \\ \mathbf{[C^{Av}]^T} & [C^{vv}] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{\dot{v}\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{\nu\} \end{Bmatrix} = \begin{Bmatrix} \{J_i\} \\ \{I\} \end{Bmatrix} \qquad (11.0\text{–}6)$$

where:

$$\{J_i\} = \{J_i^{nd}\} + \{J^s\} + \{J^{pm}\}$$

$$\{I\} = \{I^{nd}\} + \{I^t\}$$

3. Analysis types: Harmonic or Transient

E. Electro–Magneto–Thermo–Structural Analysis (see Sections 5.2, 11.1, and 11.2)

1. Element types: SOLID5, SOLID98
2. Matrix equation:

$$\begin{bmatrix} [M] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{T}\} \\ \{\ddot{V}\} \\ \{\ddot{\phi}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] & [0] & [0] \\ [0] & [C^t] & [0] & [0] \\ [0] & [0] & [0] & [0] \\ [0] & [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{T}\} \\ \{\dot{V}\} \\ \{\dot{\phi}\} \end{Bmatrix}$$

$$+ \begin{bmatrix} [K] & [0] & [0] & [0] \\ [0] & [K^t] & [0] & [0] \\ [0] & [0] & [K^v] & [0] \\ [0] & [0] & [0] & [K^m] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{T\} \\ \{V\} \\ \{\phi\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{Q\} \\ \{I\} \\ \{J_f\} \end{Bmatrix} \qquad (11.0\text{–}7)$$

where:

$$[K^t] = [K^{tb}] + [K^{tc}]$$

$$\{F\} = \{F^{nd}\} + \mathbf{\{F^{th}\}} + \{F^{ac}\} + \mathbf{\{F^{jb}\}} + \{F^{pr}\} + \mathbf{\{F^{mx}\}}$$

$$\{Q\} = \{Q^{nd}\} + \{Q^g\} + \mathbf{\{Q^j\}} + \{Q^c\}$$

$$\{I\} = \{I^{nd}\}$$

$$\{J_f\} = \{J_f^{nd}\} + \mathbf{\{J^g\}} + \{J^{pm}\}$$

3. Analysis types: Static or Transient

F. Electro–Magneto–Thermal Analysis (see Sections 5.2 and 11.2)

1. Element types: SOLID5, SOLID98
2. Matrix equation:

$$\begin{bmatrix} [C^t] & [0] & [0] \\ [0] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{V}\} \\ \{\dot{\phi}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] & [0] \\ [0] & [K^v] & [0] \\ [0] & [0] & [K^m] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{V\} \\ \{\phi\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{I\} \\ \{J_f\} \end{Bmatrix} \tag{11.0–8}$$

where:

$[K^t] = [K^{tb}] + [K^{tc}]$

$\{Q\} = \{Q^{nd}\} + \{Q^g\} + \mathbf{\{Q^j\}} + \{Q^c\}$

$\{I\} = \{I^{nd}\}$

$\{J_f\} = \{J_f^{nd}\} + \mathbf{\{J^g\}} + \{J^{pm}\}$

3. Analysis types: Static or Transient

G. **Piezoelectric Analysis** (see Section 11.1)

1. Element types: SOLID5, PLANE13, SOLID98
2. Matrix equation:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K] & [\mathbf{K^z}] \\ [\mathbf{K^z}]^T & [K^d] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{L\} \end{Bmatrix} \tag{11.0–9}$$

where:

$\{F\} = \{F^{nd}\} + \{F^{th}\} + \{F^{ac}\} + \{F^{pr}\}$

$\{L\} = \{L^{nd}\}$

3. Analysis types: Static, modal, harmonic, or transient

H. **Thermo–Pressure Analysis** (see Section 14.66)

1. Element type: FLUID66
2. Matrix equation:

$$\begin{bmatrix} [C^t] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{P}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] \\ [0] & [K^p] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{P\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{W\} \end{Bmatrix} \tag{11.0–10}$$

where:

$[K^t] = [K^{tb}] + [K^{tc}] + \mathbf{[K^{tm}]}$

$\{Q\} = \{Q^{nd}\} + \{Q^c\} + \{Q^g\}$

$\{W\} = \{W^{nd}\} + \mathbf{\{W^h\}}$

3. Analysis types: Static or Transient

I. **Velocity–Thermo–Pressure Analysis** (see Section 7.2)

1. Element type: FLUID141 and FLUID142
2. Matrix equation ([A] matrices combine effects of [C] and [K] matrices):

$$[A^{VX}]\ \{V_x\} = \{F^{NX}\} \tag{11.0–11}$$

$$[A^{VY}]\ \{V_y\} = \{F^{NY}\} \tag{11.0–12}$$

$$[A^{VZ}]\ \{V_z\} = \{F^{NZ}\} \tag{11.0–13}$$

$$[A^{P}]\ \{P\} = \{F^{P}\} \tag{11.0–14}$$

$$[A^{T}]\ \{T\} = \{F^{T}\} \tag{11.0–15}$$

$$[A^{K}]\ \{k\} = \{F^{K}\} \tag{11.0–16}$$

$$[A^{\epsilon}]\ \{\epsilon\} = \{F^{\epsilon}\} \tag{11.0–17}$$

where: $[A^{VX}]$ = advection–diffusion matrix for V_x velocities
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{T\}$, $\{k\}$, and $\{\varepsilon\}$
$[A^{VY}]$ = advection–diffusion matrix for V_y velocities
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{T\}$, $\{k\}$, and $\{\varepsilon\}$
$[A^{VZ}]$ = advection–diffusion matrix for V_z velocities
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{T\}$, $\{k\}$, and $\{\varepsilon\}$
$[A^{P}]$ = pressure coefficient matrix
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{T\}$, $\{k\}$, and $\{\varepsilon\}$
$[A^{T}]$ = advection–difusion matrix for temperature
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, and $\{T\}$
$[A^{k}]$ = advection–diffusion matrix for turbulent kinetic energy
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{k\}$, and $\{\varepsilon\}$
$[A^{\varepsilon}]$ = advection–diffusion matrix for dissipation energy
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{k\}$, and $\{\varepsilon\}$
$\{F^{VX}\}$ = load vector for V_x velocities
= function of previous $\{P\}$ and $\{T\}$
$\{F^{VY}\}$ = load vector for V_y velocities
= function of previous $\{P\}$ and $\{T\}$
$\{F^{VZ}\}$ = load vector for V_z velocities
= function of previous $\{P\}$ and $\{T\}$
$\{F^{P}\}$ = pressure load vector
= function of previous $\{V_x\}$, $\{V_y\}$ and $\{V_z\}$

$\{F^T\}$ = heat flow vector
= function of previous $\{T\}$
$\{F^k\}$ = turbulent kinetic energy load vector
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{T\}$, $\{k\}$, and $\{\varepsilon\}$
$\{F^\varepsilon\}$ = dissipation rate load vector
= function of previous $\{V_x\}$, $\{V_y\}$, $\{V_z\}$, $\{k\}$, and $\{\varepsilon\}$

3. Analysis types: Static or Transient

J. **Pressure–Structural (Acoustic) Analysis** (see Section 8.2)

1. Element type: FLUID29 and FLUID30 (with other structural elements)
2. Matrix equation:

$$\begin{bmatrix} [M] & [0] \\ [\mathbf{M^{fs}}] & [M^p] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{P}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [C^p] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{P}\} \end{Bmatrix} + \begin{bmatrix} [K] & [\mathbf{K^{fs}}] \\ [0] & [K^p] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{P\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{W\} \end{Bmatrix} \qquad (11.0\text{–}18)$$

where: $\{F\}$ = $\{F^{nd}\}$
$\{W\}$ = $\{W^{nd}\}$

Note that [M], [C], and [K] are provided by other elements.

3. Analysis types: Transient, harmonic and modal analyses can be performed. Applicable matrices are shown in the following table:

	MODAL			TRANSIENT	HARMONIC	
	DAMPED	UNSYMM.	SYMM.		UNSYMM.	SYMM.
$[M]$	*	*		*	*	
$[M^{fs}]$	*	*		*	*	
$[M^p]$	*	*	*	*	*	*
$[C]$	*			*	*	
$[C^p]$	*			*	*	*
$[K]$	*	*		*	*	
$[K^{fs}]$	*	*		*	*	
$[K^p]$	*	*	*	*	*	*
$\{F^{nd}\}$				*	*	

K. **Thermo–Electric Analysis** (see Section 11.2)

1. Element types: SOLID5, PLANE67, LINK68, SOLID69, SOLID98, SHELL157
2. Matrix equation:

$$\begin{bmatrix} [C^t] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] \\ [0] & [K^v] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{I\} \end{Bmatrix} \qquad (11.0\text{–}19)$$

where: $[K^t] = [K^{tb}] + [K^{tc}]$

$\{Q\} = \{Q^{nd}\} + \{Q^c\} + \{Q^g\} + \mathbf{\{Q^j\}}$

$\{I\} = \{I^{nd}\}$

3. Analysis types: Static or Transient

L. Magnetic–Thermal Analysis (see Section 5.2)

1. Element type: PLANE13
2. Matrix equation:

$$\begin{bmatrix} [C^{AA}] & [0] \\ [0] & [C^t] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{\dot{T}\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [0] \\ [0] & [K^t] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{T\} \end{Bmatrix} = \begin{Bmatrix} \{J_i\} \\ \{Q\} \end{Bmatrix} \qquad (11.0\text{–}20)$$

where: $[K^t] = [K^{tb}] + [K^{tc}]$

$\{J_i\} = \{J_i^{nd}\} + \{J^s\} + \{J^{pm}\}$

$\{Q\} = \{Q^{nd}\} + \{Q^g\} + \mathbf{\{Q^j\}} + \{Q^c\}$

3. Analysis types: Static or Transient

M. Circuit–Magnetic Analysis (see Section 5.4)

1. Element type: PLANE53, SOLID97, CIRCU124
2. Matrix equation:

$$\begin{bmatrix} [0] & [0] & [0] \\ [C^{iA}] & [0] & [0] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{A}\} \\ \{0\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^{AA}] & [K^{Ai}] & [0] \\ [0] & [K^{ii}] & [K^{ie}] \\ [0] & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{A\} \\ \{i\} \\ \{e\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix} \qquad (11.0\text{–}21)$$

3. Analysis types: Static, Transient, or Harmonic

Table 11.0–1 Nomenclature

Coefficient Matrices of Second Time Derivative of Unknowns

Symbol	Meaning	Discussed in Section
[M]	structural mass matrix	2.2
[M^{fs}]	fluid–structure coupling mass matrix	8.2
[M^{p}]	acoustic mass matrix	8.2

Coefficient Matrices of First Time Derivative of Unknowns

Symbol	Meaning	Discussed in Section
[C]	structural damping matrix	2.2
[C^{t}]	thermal specific heat matrix	6.2
[C^{AA}]	magnetic damping matrix	5.3
[C^{p}]	acoustic damping matrix	8.2
[C^{Av}]	magnetic–electric damping matrix	5.2
[C^{vv}]	electric damping matrix	5.2
[C^{iA}]	inductive damping matrix	5.4

Coefficient Matrices of Unknowns

Symbol	Meaning	Discussed in Section
[K]	structural stiffness matrix	2.2
[K^{t}]	thermal conductivity matrix (may consist of 1, 2, or 3 of the following 3 matrices)	6.2
[K^{tb}]	thermal conductivity matrix of material	6.2
[K^{tc}]	thermal conductivity matrix of convection surface	6.2
[K^{tm}]	thermal conductivity matrix associated w/mass transport	6.2
[K^{m}]	scalar magnetic potential coefficient matrix	5.2
[K^{AA}]	vector magnetic potential coefficient matrix	5.2
[K^{Ai}]	potential–current coupling stiffness matrix	5.4
[K^{ii}]	resistive stiffness matrix	5.4
[K^{ie}]	current–emf coupling stiffness	5.4
[K^{v}]	electric coefficient matrix	5.2

Symbol	Meaning	Discussed in Section
$[K^z]$	piezoelectric stiffness matrix	11.1
$[K^d]$	dielectric coefficient matrix	11.1

Coefficient Matrices of Unknowns continued

Symbol	Meaning	Discussed in Section
$[K^f]$	momentum matrix due to diffusion	7.2
$[K^g]$	buoyancy matrix	7.2
$[K^c]$	pressure gradient matrix	7.2
$[K^p]$	pressure coefficient or fluid stiffness matrix	7.2
$[K^{fs}]$	fluid–structure coupling stiffness matrix	7.2

Vectors of Knowns

Symbol	Meaning	Associated Input/Output Label	Discussed in Section
$\{F^{nd}\}$	applied nodal force vector	FX...MZ	2.2
$\{F^{nr}\}$	Newton–Raphson restoring load vector	FX...MZ	15.9
$\{F^{th}\}$	thermal strain force vector	FX...MZ	2.2
$\{F^{pr}\}$	pressure load vector	FX...MZ	2.2
$\{F^{ac}\}$	force vector due to acceleration effects (i.e., gravity)	FX...MZ	2.2
$\{F^{jb}\}$	Lorentz force vector	FX...FZ	5.2
$\{F^{mx}\}$	Maxwell force vector	FX...FZ	5.2
$\{F^{b}\}$	body force load vector due to non–gravity effects	FX...MZ	6.2
$\{Q^{nd}\}$	applied nodal heat flow rate vector	HEAT	6.2
$\{Q^{f}\}$	heat flux vector	HEAT	6.2
$\{Q^{c}\}$	convection surface vector	HEAT	6.2
$\{Q^{g}\}$	heat generation rate vector for causes other than Joule heating	HEAT	6.2
$\{Q^{j}\}$	heat generation rate vector for Joule heating	HEAT	11.2
$\{J_i^{nd}\}$	applied nodal source current vector (associated with {A})	CSGX, CSGY, CSGZ	5.2
$\{J_f^{nd}\}$	applied nodal flux vector (associated with $\{\phi\}$)	FLUX	5.2
$\{J^{g}\}$	Source (Biot–Savart) vector		5.2
$\{J^{pm}\}$	coercive force (permanent magnet) vector	FLUX	5.2
$\{J^{s}\}$	source current density vector	FLUX	5.2
$\{I^{nd}\}$	applied total current vector	AMPS	5.2
$\{I^{t}\}$	total current density vector	AMPS	5.2
$\{L^{nd}\}$	applied nodal charge vector	AMPS	11.1
$\{W^{nd}\}$	applied nodal fluid flow vector	FLOW	14.66
$\{W^{h}\}$	static head vector	FLOW	14.66

Vectors of Unknowns

Symbol	Meaning	Associated Input/Output Label	Discussed in Section
$\{u\}$	displacement vector	UX...ROTZ	2.2
$\{T\}$	thermal potential (temperature) vector	TEMP	6.2, 7.2
$\{V\}$	electric potential vector	VOLT	5.2
$\{\nu\}$	time integrated electric potential vector	VOLT	5.2
$\{\phi\}$	magnetic scalar potential vector	MAG	5.2
$\{A\}$	magnetic vector potential vector	AX, AY, AZ	5.2
$\{i\}$	electric current vector	CURR	5.4
$\{e\}$	electromagnetic force drop vector	EMF	5.4
$\{P\}$	pressure vector	PRES	7.2, 8.2
$\{V_x\}$	X component of velocity	VX	7.2
$\{V_y\}$	Y component of velocity	VY	7.2
$\{V_z\}$	Z component of velocity	VZ	7.2
$\{k\}$	turbulent kinetic energy	ENKE	7.2
$\{\varepsilon\}$	turbulent dissipation energy	ENDS	7.2
•	time derivative		
••	second time derivative		

11.1 Piezoelectrics

The PLANE13 capability of modeling piezoelectrics response exists in the following elements:

SOLID5 – Coupled–Field Solid Element
PLANE13 – 2–D Coupled–Field Solid Element
SOLID98 – Tetrahedral Coupled–Field Solid Element

Variational principles are used to develop the finite element equations which incorporate the piezoelectric effect (Allik(81)).

The electromechanical constitutive equations for linear material behavior are:

$$\{T\} = [c]\ \{S\} - [e]\ \{E\} \tag{11.1–1}$$

$$\{D\} = [e]^T\{S\} + [\epsilon]\ \{E\} \tag{11.1–2}$$

or equivalently

$$\begin{Bmatrix} \{T\} \\ \{D\} \end{Bmatrix} = \begin{bmatrix} [c] & [e] \\ [e]^T & -[\epsilon] \end{bmatrix} \begin{Bmatrix} \{S\} \\ -\{E\} \end{Bmatrix} \tag{11.1–3}$$

where:
$\{T\}$ = stress vector (referred to as $\{\sigma\}$ elsewhere in this manual)
$\{D\}$ = electric flux density vector
$\{S\}$ = strain vector (referred to as $\{\epsilon\}$ elsewhere in this manual)
$\{E\}$ = electric field vector
[c] = elasticity matrix (evaluated at constant electric field (referred to as [D] elsewhere in this manual))
[e] = piezoelectric matrix
$[\epsilon]$ = dielectric matrix (evaluated at constant mechanical strain)

Equations (11.1–1) and (11.1–2) are the usual constitutive equations for structural and electrical fields, respectively, except for the coupling terms involving the piezoelectric matrix [e].

The elasticity matrix [c] is the usual [D] matrix described in Section 2.1 and is input using the **MP** commands. It can be input directly in uninverted form [c] or in inverted form $[c]^{-1}$ as a general anisotropic symmetric matrix (input on **TBDATA** commands using **TB**,ANEL):

$$[c] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ & & c_{33} & c_{34} & c_{35} & c_{36} \\ & \text{Symmetric} & & c_{44} & c_{45} & c_{46} \\ & & & & c_{55} & c_{56} \\ & & & & & c_{66} \end{bmatrix} \quad (11.1\text{–}4)$$

The piezoelectric matrix [e] (input on **TBDATA** commands using **TB**,PIEZ) relate the electric field vector {E} in the order X, Y, Z to the stress vector {T} in the order X, Y, Z, XY, YZ, XZ and is of the form:

$$[e] = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \\ e_{41} & e_{42} & e_{43} \\ e_{51} & e_{52} & e_{53} \\ e_{61} & e_{62} & e_{63} \end{bmatrix} \quad (11.1\text{–}5)$$

The dielectic matrix [ε] uses input quantities PERX, PERY and PERZ on the **MP** commands and is of the form:

$$[\epsilon] = \begin{bmatrix} \epsilon_{11} & 0 & 0 \\ 0 & \epsilon_{22} & 0 \\ 0 & 0 & \epsilon_{33} \end{bmatrix} \quad (11.1\text{–}6)$$

The finite element discretization is performed by establishing nodal solution variables and element shape functions over an element domain which approximate the solution.

$$\{u_c\} = [N^u]^T\{u\} \tag{11.1–7}$$

$$V_c = \{N^V\}^T\{V\} \tag{11.1–8}$$

where:

$\{u_c\}$ = displacements within element domain in the x, y, z directions

V_c = electrical potential within element domain

$[N^u]$ = matrix of displacement shape functions

$\{N^V\}$ = vector of electrical potential shape function

$\{u\}$ = vector of nodal displacements

$\{V\}$ = vector of nodal electrical potential

Expanding these definitions:

$$[N^u]^T = \begin{bmatrix} N_1 & 0 & 0 & & N_n & 0 & 0 \\ 0 & N_1 & 0 & \dots & 0 & N_n & 0 \\ 0 & 0 & N_1 & \dots & 0 & 0 & N_n \end{bmatrix} \tag{11.1–9}$$

$$\{N^V\}^T = \lfloor\ N_1 \quad N_2 \dots N_n \rfloor \qquad (11.1\text{–}10$$

where: N_i = shape function for node i

$$\{u\} = \lfloor\ UX_1 \quad UY_2 \quad UZ_3 \ \dots \ UX_n \quad UY_n \quad UZ_n \rfloor^T \qquad (11.1\text{–}11$$

$$\{V\} = \begin{Bmatrix} V_1 \\ V_2 \\ \vdots \\ V_n \end{Bmatrix} \tag{11.1–12}$$

where: n = number of nodes of the element

Then the strain {S} and electric field {E} are related to the displacements and potentials, respectively, as:

$$\{S\} = [B_u]\ \{u\} \tag{11.1–13}$$

$$\{E\} = -[B_V]\ \{V\} \tag{11.1–14}$$

$$[B_u] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} [N^u]^T \qquad (11.1–15)$$

$$[B_V] = \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{Bmatrix} \{N^V\}^T \qquad (11.1–16)$$

After the application of the variational principle and finite element discretization (Allik(81)), the coupled finite element matrix equation derived for a one element model is:

$$\begin{bmatrix} [M] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\ddot{u}\} \\ \{\ddot{V}\} \end{Bmatrix} + \begin{bmatrix} [C] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{u}\} \\ \{\dot{V}\} \end{Bmatrix} + \begin{bmatrix} [K] & [K^z] \\ [K^z]^T & [K^d] \end{bmatrix} \begin{Bmatrix} \{u\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{F\} \\ \{L\} \end{Bmatrix} \qquad (11.1–17)$$

where a dot above a variable denotes a time derivative. The following equations provide an explanation of the submatrices in equation (11.1–17):

Structural mass:

$$[M] = \int_{vol} \rho \, [N^u][N^u]^T \, d(vol) \qquad (11.1–18)$$

where: ρ = mass density (input quantity DENS on **MP** command)

Structural damping:

Explanation of [C] found in Section 15.3 (valid for displacement DOF only).

Structural stiffness:

$$[K] = \int_{vol} [B_u]^T [c] [B_u] \, d(vol) \qquad (11.1\text{–}19)$$

Dielectric Conductivity:

$$\left[K^d\right] = -\int_{vol} \left[B_V\right]^T [\epsilon] \left[B_V\right] d(vol) \qquad (11.1\text{–}20)$$

Piezoelectric coupling matrix:

$$\left[K^z\right] = \int_{vol} [B_u]^T [e] \left[B_V\right] d(vol) \qquad (11.1\text{–}21)$$

Structural load vector:

{F} = vector of nodal forces, surface forces, and body forces (see Section 17.0).

Electrical load vector:

$\{L\} = \{L^{nd}\}$ = applied nodal charge vector

In the reduced mode–frequency analysis (**ANTYPE**,MODAL) the potential DOF is not usable as a master DOF in the reduction process since it has no mass and is, therefore, condensed into the master DOF.

In a harmonic response analysis (**ANTYPE**,HARMIC) the potential DOF is allowed as a master DOF.

Energy coefficients are calculated for each piezoelectric element as follows (Sato(83)):

Elastic energy:

$$U_E = \frac{1}{2} \{u\}^T [K]\{u\} \qquad (11.1\text{–}22)$$

Dielectric energy:

$$U_D = \frac{1}{2} \{V\}^T \left[K^d\right] \{V\} \qquad (11.1\text{–}23)$$

Electromechanical coupling energy:

$$U_M = \frac{1}{4}\left(\{u\}^T\left[K^z\right]\{V\} + \{V\}^T\left[K^z\right]^T\{u\}\right) \qquad (11.1\text{–}24)$$

The energy coefficients shown above are stored on the post data file as UE, UD, and UM, respectively. The above calculations are independent of the calculations performed if the **OUTPR**,VENG command is input (see Section 15.16).

11.2 Thermal–Electric Elements

11.2.1 Other Applicable Sections

Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. The development in Chapter 5 is also valid for electrical effects except that:

1. Temperature (T) is replaced by volts (V).
2. Heat flow (Q) is replaced by current (I).
3. The conductivities (KXX, KYY, KZZ) are replaced by the inverse of the resistivities (1.0/RSVX, 1.0/RSVY, 1.0/RSVZ).
4. No “convection surfaces” or “specific heat” (damping) equivalences are used. Electrical “damping” (inductance) is not included because it is normally on a completely different time scale than thermal damping.

11.2.2 Implementation

There are two forms of interrelationships between the electrical and thermal parts of the solution.

1. The Joule heat vector $\{Q^j\}$, used by the thermal part of the solution, is derived from the electrical part of the solution.
2. The electrical conductivity matrix $[K^v]$, used by the electrical part of the solution, has temperature–dependent material properties using temperatures computed in the thermal part of the solution.

The thermal and electrical aspects of the problem are combined into one element having two different types of working variables: temperature and voltages. The equilibrium equations for one element have the form of:

$$\begin{bmatrix} [C^t] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{0\} \end{Bmatrix} + \begin{bmatrix} [K^t] & [0] \\ [0] & [K^v] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{V\} \end{Bmatrix} = \begin{Bmatrix} \{Q^{nd}\} \\ \{I^{nd}\} \end{Bmatrix} + \begin{Bmatrix} \{Q^c\} \\ \{0\} \end{Bmatrix} + \begin{Bmatrix} \{Q^g\} \\ \{0\} \end{Bmatrix} + \begin{Bmatrix} \{Q^j\} \\ \{0\} \end{Bmatrix} \qquad (11.2\text{–}1)$$

where: $[C^t]$ = specific heat matrix
$[K^t]$ = thermal conductivity matrix (material and convection surfaces)
$[K^v]$ = electrical conductivity matrix
$\{T\}$ = nodal temperature vector

$\{\dot{T}\}$ = vector of variation of nodal temperatures with respect to time

$\{V\}$ = nodal voltage vector

$\{Q^{nd}\}$ = nodal heat flow vector

$\{I^{nd}\}$ = nodal current flow vector

$\{Q^c\}$ = convection surface vector

$\{Q^g\}$ = internal heat generation rate vector (input on **BF** or **BFE** commands)

$\{Q^j\}$ = vector of heat flows into a node caused by Joule heating (see below)

The Joule heating load vector is:

$$\left\{Q^j\right\} = \int_{vol} \{N\} \lfloor V_g \rfloor [\sigma] \left\{V_g\right\} d(vol) \tag{11.2–2}$$

where: $\{N\}$ = shape functions of the element

$\{V_g\}$ = vector of voltage gradients (see below)

$[\sigma]$ = conductivity matrix = $\begin{bmatrix} \sigma_x & 0 & 0 \\ 0 & \sigma_y & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho_x} & 0 & 0 \\ 0 & \frac{1}{\rho_y} & 0 \\ 0 & 0 & \frac{1}{\rho_z} \end{bmatrix}$

σ_x = electrical conductivity in the x–direction

ϱ_x = electrical resistivity in the x–direction (input as RSVX on **MP** commands)

The voltage gradients are:

$$\left\{V_g\right\} = [B] \left\{V_{n-1}\right\} \tag{11.2–3}$$

where: $[B]$ = matrix of derivatives of the shape functions in the x, y, and z directions

$\{V_{n-1}\}$ = nodal voltages from the previous iteration

11.2.3 Calculation of Output Gradients, Currents, and Fluxes

The output voltage gradients and current densities are computed by:

$$\left\{V_g\right\} = [B] \{V_n\} \tag{11.2–4}$$

where: $\{V_g\}$ = voltage gradient components (output quantities VG (X, Y, Z))

$\{V_n\}$ = nodal voltages from the present iteration

and

$$\left\{I_n^c\right\} = [\sigma] \left\{V_g\right\} \tag{11.2–5}$$

where: $\left\{I_n^c\right\}$ = current density components (output quantities CD (X, Y, Z))

The output thermal gradients and fluxes are computed by:

$$\left\{T_g\right\} = [B] \{T_n\} \tag{11.2–6}$$

where: $\{T_g\}$ = thermal gradient components (output quantities TG (X, Y, Z))

$\{T_n\}$ = nodal temperatures from the present iteration

and

$$\left\{Q_n^f\right\} = [D] \left\{T_g\right\} \tag{11.2–7}$$

where: $\left\{Q_n^f\right\}$ = thermal flux components (output quantities TF (X, Y, Z))

$[D]$ = conductivity matrix

Output with the subscript “SUM” is simply the vector sum of the components.

Chapter 12
Shape Functions

ANSYS Theory Reference
Chapter 12 – Table of Contents

12.0 Shape Functions

The shape functions for the elements are given in this chapter. They are referred to by the individual element descriptions in Chapter 14. All subheadings for this chapter are included in the table of contents to aid in finding a certain type of shape function.

The given functions are related to the nodal quantities by:

Table 12.0–1 Shape Function Labels

Variable	Input/Output Label	Meaning	Last Part of Equation Identifier
u	UX	Translation in the x (or s) direction	1
v	UY	Translation in the y (or t) direction	2
w	UZ	Translation in the z (or r) direction	3
θ_x	ROTX	Rotation about the x direction	4
θ_y	ROTY	Rotation about the y direction	5
θ_z	ROTZ	Rotation about the z direction	6
A_x	AX	X–component of vector magnetic potential	7
A_y	AY	Y–component of vector magnetic potential	8
A_z	AZ	Z–component of vector magnetic potential	9
V_x	VX	Velocity in the x–direction	10
V_y	VY	Velocity in the y–direction	11
V_z	VZ	Velocity in the z–direction	12
		Unused	13–18
P	PRES	Pressure	19
T	TEMP	Temperature	20
V	VOLT	Electric potential or source current	21
ϕ	MAG	Scalar magnetic potential	22
E^K	ENKE	Turbulent kinetic energy	23
E^D	ENDS	Energy dissipation	24

The vector correspondences are not exact, since, for example, u, v, and w are in the element coordinate system, whereas UX, UY, UZ represent motions in the nodal coordinate system. Generally, the element coordinate system is the same as the global Cartesian system, except for:

1. Line elements (Sections 12.1 to 12.4), where u motions are axial motions, and v and w are transverse motions.
2. Shell elements (Section 12.5), where u and v are in–plane motions and w is the out–of–plane motion.

Subscripted variables such as u_J refer to the u motion at node J. When these same variables have numbers for subscripts (e.g. u_1), nodeless variables for extra shape functions are being referred to. Coordinates s, t, and r are normalized, going from –1.0 on one side of the element to +1.0 on the other, and are not necessarily orthogonal to one another. L_1, L_2, L_3, and L_4 are also normalized coordinates, going from 0.0 at a vertex to 1.0 at the opposite side or face.

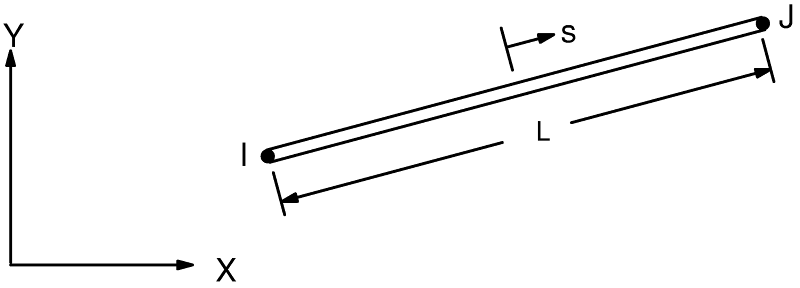
Elements with midside nodes allow those midside nodes to be dropped in most cases. A dropped midside node implies that the edge is and remains straight, and that any other effects vary linearly along that edge.

Gaps are left in the equation numbering to allow for additions. Labels given in subsection titles within parentheses are used to relate the given shape functions to their popular names, where applicable.

Some elements in Chapter 14 (notably the 8 node solids) imply that reduced element geometries (e.g., wedge) are not available. However, the tables in Chapter 14 refer only to the available shape functions. In other words, the shape functions used for the 8–node brick is the same as the 6–node wedge.

12.1 2–D Lines

This section contains shape functions for line elements without and with rotational degrees of freedom (RDOFs).



12.1.1 2–D Lines without RDOF

These shape functions are for 2–D line elements without RDOFs, such as LINK1 or LINK32.

$$u = \frac{1}{2}\left(u_I\ (1 - s) + u_J\ (1 + s)\right) \tag{12.1.1–1}$$

$$v = \frac{1}{2}\left(v_I\ (1 - s) + v_J\ (1 + s)\right) \tag{12.1.1–2}$$

$$T = \frac{1}{2}\left(T_I\ (1 - s) + T_J\ (1 + s)\right) \tag{12.1.1–20}$$

12.1.2 2–D Lines with RDOF

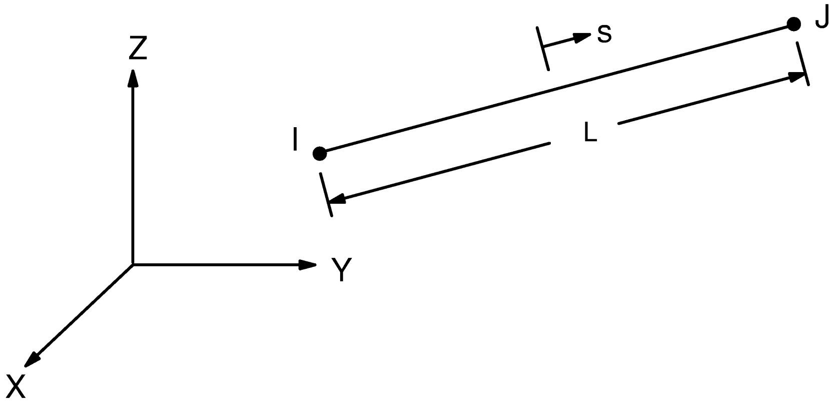
These shape functions are for 2–D line elements with RDOFs, such as BEAM3.

$$u = \frac{1}{2}\left(u_I\ (1–s) + u_J\ (1 + s)\right) \tag{12.1.2–1}$$

$$v = \frac{1}{2}\left(v_I\left(1 - \frac{s}{2}\left(3 - s^2\right)\right) + v_J\left(1 + \frac{s}{2}\left(3 - s^2\right)\right)\right)$$
$$+ \frac{L}{8}\left(\theta_{z,I}\left(1 - s^2\right)(1 - s) + \theta_{z,J}\left(1 - s^2\right)(1 + s)\right) \tag{12.1.2–2}$$

12.2 3–D Lines

This section contains shape functions for line elements without and with rotational degrees of freedom (RDOFs).



12.2.1 3–D 2 Node Lines without RDOF

These shape functions are for 3–D 2 node line elements without RDOFs, such as LINK8, LINK33, LINK68 or BEAM188.

$$u = \frac{1}{2}\left(u_I\ (1 - s) + u_J\ (1 + s)\right) \tag{12.2.1–1}$$

$$v = \frac{1}{2}\left(v_I\ (1 - s) + v_J\ (1 + s)\right) \tag{12.2.1–2}$$

$$w = \frac{1}{2}\left(w_I\ (1 - s) + w_J\ (1 + s)\right) \tag{12.2.1–3}$$

$$\theta_x = \frac{1}{2}\left(\theta_{xI}\ (1–s) + \theta_{xJ}\ (1 + s)\right) \tag{12.2.1–4}$$

$$\theta_y = \frac{1}{2}\left(\theta_{yI}\ (1–s) + \theta_{yJ}\ (1 + s)\right) \tag{12.2.1–5}$$

$$\theta_z = \frac{1}{2}\left(\theta_{zI}\ (1–s) + \theta_{zJ}\ (1 + s)\right) \tag{12.2.1–6}$$

$$P = \frac{1}{2}\left(P_I\ (1 - s) + P_J\ (1 + s)\right) \tag{12.2.1–19}$$

$$T = \frac{1}{2}\left(T_I\ (1 - s) + T_J\ (1 + s)\right) \tag{12.2.1–20}$$

$$V = \frac{1}{2}\left(V_I\ (1 - s) + V_J\ (1 + s)\right) \tag{12.2.1–21}$$

12.2.2 3–D 2 Node Lines with RDOF

These shape functions are for 3–D 2–node line elements with RDOFs, such as BEAM4.

$$u = \frac{1}{2}\left(u_I\ (1 - s) + u_J\ (1 + s)\right) \tag{12.2.2–1}$$

$$\begin{aligned} v = {} & \frac{1}{2}\left(v_I\left(1 - \frac{s}{2}\left(3 - s^2\right)\right) + v_J\left(1 + \frac{s}{2}\left(3 - s^2\right)\right)\right) \\ & + \frac{L}{8}\left(\theta_{z,I}\left(1 - s^2\right)(1 - s) - \theta_{z,J}\left(1 - s^2\right)(1 + s)\right) \end{aligned} \tag{12.2.2–2}$$

$$\begin{aligned} w = {} & \frac{1}{2}\left(w_I\left(1 - \frac{s}{2}\left(3 - s^2\right)\right) + w_J\left(1 + \frac{s}{2}\left(3 - s^2\right)\right)\right) \\ & - \frac{L}{8}\left(\theta_{y,I}\left(1 - s^2\right)(1 - s) - \theta_{y,J}\left(1 - s^2\right)(1 + s)\right) \end{aligned} \tag{12.2.2–3}$$

$$\theta_x = \frac{1}{2}\left(\theta_{x,I}\ (1 - s) + \theta_{x,J}\ (1 + s)\right) \tag{12.2.2–4}$$

12.2.3 3–D 3 Node Lines

These shape functions are for 3–D 3 node line elements such as BEAM189.

$$u = \frac{1}{2}\left(u_H\left(-s + s^2\right) + u_I\left(s + s^2\right)\right) + u_J\ (1 - s^2) \tag{12.2.3–1}$$

$$v = \frac{1}{2}\left(v_H\left(-s + s^2\right) + v_I\left(s + s^2\right)\right) + v_J\ (1 - s^2) \tag{12.2.3–2}$$

$$w = \frac{1}{2}\left(w_H\left(-s + s^2\right) + w_I\left(s + s^2\right)\right) + w_J\,(1 - s^2) \qquad (12.2.3\text{–}3)$$

$$\theta_x = \frac{1}{2}\left(\theta_{xH}\left(-s + s^2\right) + \theta_{xI}\left(s + s^2\right)\right) + \theta_{xJ}\,(1 - s^2) \qquad (12.2.3\text{–}4)$$

$$\theta_y = \frac{1}{2}\left(\theta_{yH}\left(-s + s^2\right) + \theta_{yI}\left(s + s^2\right)\right) + \theta_{yJ}\,(1 - s^2) \qquad (12.2.3\text{–}5)$$

$$\theta_z = \frac{1}{2}\left(\theta_{zH}\left(-s + s^2\right) + \theta_{zI}\left(s + s^2\right)\right) + \theta_{zJ}\,(1 - s^2) \qquad (12.2.3\text{–}6)$$

12.3 Axisymmetric Shells

This section contains shape functions for 2–node axisymmetric shell elements under axisymmetric load. These elements may have extra shape functions (ESF).

12.3.1 Axisymmetric Shell without ESF

These shape functions are for 2–node axisymmetric shell elements without extra shape functions, such as SHELL51 with KEYOPT(3) = 1.

$$u = \frac{1}{2}\left(u_I\ (1 - s) + u_J\ (1 + s)\right) \qquad (12.3.1\text{–}1)$$

$$v = \frac{1}{2}\left(v_I\ (1 - s) + v_J\ (1 + s)\right) \qquad (12.3.1\text{–}2)$$

$$\begin{aligned} w = {} & \frac{1}{2}\left(w_I\left(1 - \frac{s}{2}\left(3 - s^2\right)\right) + w_J\left(1 + \frac{s}{2}\left(3 - s^2\right)\right)\right) \\ & + \frac{L}{8}\left(\theta_I\left(1 - s^2\right)(1 - s) - \theta_J\left(1 - s^2\right)(1 + s)\right) \end{aligned} \qquad (12.3.1\text{–}3)$$

12.3.2 Axisymmetric Shell with ESF

These shape functions are for 2–node axisymmetric shell elements with extra displacement shape functions, such as SHELL51 with KEYOPT(3) = 0.

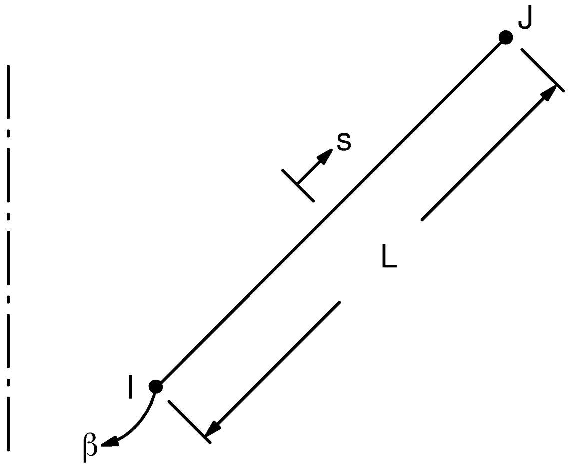
$$\begin{aligned} u = {} & \frac{1}{2}\left(u_I\left(1 - \frac{s}{2}\left(3 - s^2\right)\right) + u_J\left(1 + \frac{s}{2}\left(3 - s^2\right)\right)\right) \\ & + \frac{L}{8}\left(u_1\left(1 - s^2\right)(1 - s) - u_2\left(1 - s^2\right)(1 + s)\right) \end{aligned} \qquad (12.3.2\text{–}1)$$

$$\begin{aligned} v = {} & \frac{1}{2}\left(v_I\left(1 - \frac{s}{2}\left(3 - s^2\right)\right) + v_J\left(1 + \frac{s}{2}\left(3 - s^2\right)\right)\right) \\ & + \frac{L}{8}\left(v_1\left(1 - s^2\right)(1 - s) - v_2\left(1 - s^2\right)(1 + s)\right) \end{aligned} \qquad (12.3.2\text{–}2)$$

$$
\begin{aligned}
w = {} & \frac{1}{2}\left(w_I\left(1-\frac{s}{2}\left(3-s^2\right)\right) + w_J\left(1+\frac{s}{2}\left(3-s^2\right)\right)\right) \\
& + \frac{L}{8}\left(\theta_I\left(1-s^2\right)(1-s) - \theta_J\left(1-s^2\right)(1+s)\right)
\end{aligned}
\qquad (12.3.2\text{–}3)
$$

12.4 Axisymmetric Harmonic Shells

This section contains shape functions for 2–node axisymmetric shell elements under nonaxisymmetric (harmonic) load. These elements may have extra shape functions (ESF).



The shape functions of this section use the quantities sin$\ell\beta$ and cos$\ell\beta$, where ℓ = input quantity MODE on the **MODE** command. The sin$\ell\beta$ and cos$\ell\beta$ are interchanged if I_S = –1, where I_S = input quantity ISYM on the **MODE** command. If ℓ = 0, both sin$\ell\beta$ and cos$\ell\beta$ are set equal to 1.0.

12.4.1 Axisymmetric Harmonic Shells without ESF

These shape functions are for 2–node axisymmetric harmonic shell elements without extra shape functions, such as SHELL61 with KEYOPT(3) = 1.

$$u = \frac{1}{2}\left(u_I\ (1-s) + u_J\ (1+s)\right)\cos\ell\beta \qquad (12.4.1\text{–}1)$$

$$v = \frac{1}{2}\left(v_I\ (1-s) + v_J\ (1+s)\right)\sin\ell\beta \qquad (12.4.1\text{–}2)$$

$$w = \left(\frac{1}{2}\left(w_I\left(1-\frac{s}{2}\left(3-s^2\right)\right) + w_J\left(1+\frac{s}{2}\left(3-s^2\right)\right)\right)\right.$$
$$(12.4.1\text{–}3)$$
$$\left.+\frac{L}{8}\left(\theta_I\left(1-s^2\right)(1-s) - \theta_J\left(1-s^2\right)(1+s)\right)\right)\cos\ell\beta$$

12.4.2 Axisymmetric Harmonic Shells with ESF

These shape functions are for 2–node axisymmetric harmonic shell elements with extra shape functions, such as SHELL61 with KEYOPT(3) = 0.

$$\begin{aligned} u = & \Bigg(\frac{1}{2}\Bigg(u_I\left(1-\frac{s}{2}\left(3-s^2\right)\right) + u_J\left(1+\frac{s}{2}\left(3-s^2\right)\right)\Bigg)\Bigg) \\ & + \frac{L}{8}\Bigg(u_1\left(1-s^2\right)(1-s) - u_2\left(1-s^2\right)(1+s)\Bigg)\Bigg) \cos\ell\beta \end{aligned} \tag{12.4.2–1}$$

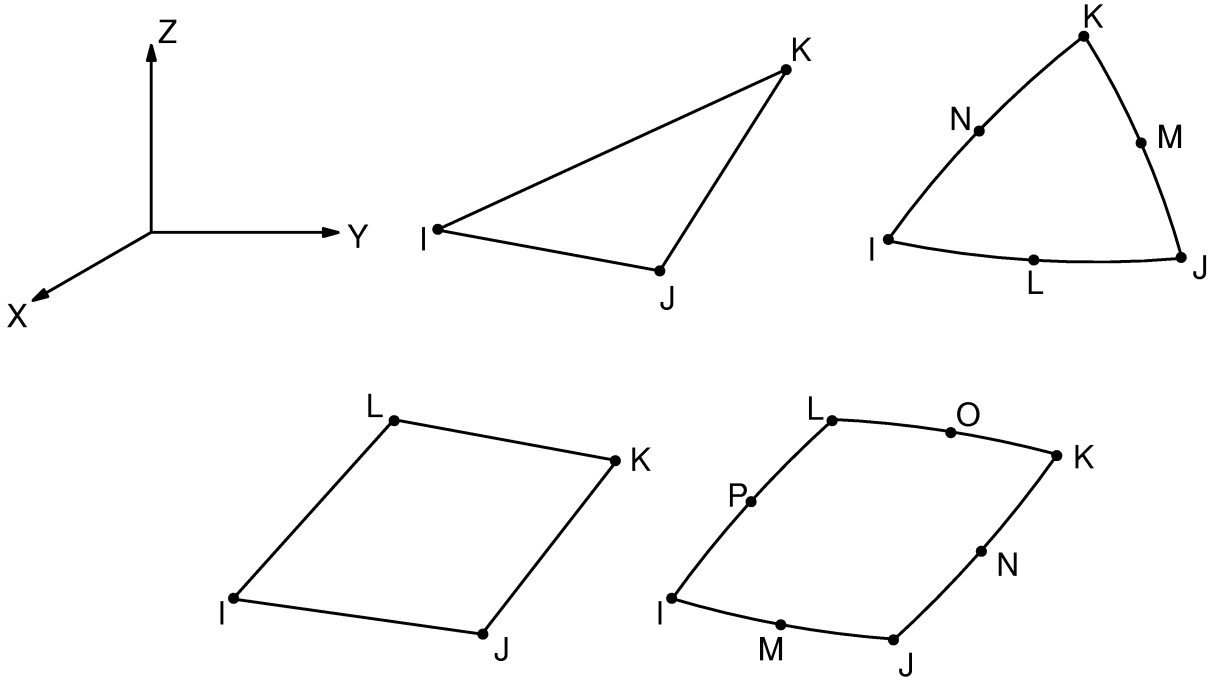
$$\begin{aligned} v = & \Bigg(\frac{1}{2}\Bigg(v_I\left(1-\frac{s}{2}\left(3-s^2\right)\right) + v_J\left(1+\frac{s}{2}\left(3-s^2\right)\right)\Bigg)\Bigg) \\ & + \frac{L}{8}\Bigg(v_1\left(1-s^2\right)(1-s) - v_2\left(1-s^2\right)(1+s)\Bigg)\Bigg) \sin\ell\beta \end{aligned} \tag{12.4.2–2}$$

$$\begin{aligned} w = & \Bigg(\frac{1}{2}\Bigg(w_I\left(1-\frac{s}{2}\left(3-s^2\right)\right) + w_J\left(1+\frac{s}{2}\left(3-s^2\right)\right)\Bigg)\Bigg) \\ & + \frac{L}{8}\Bigg(\theta_I\left(1-s^2\right)(1-s) - \theta_J\left(1-s^2\right)(1+s)\Bigg)\Bigg) \cos\ell\beta \end{aligned} \tag{12.4.2–3}$$

12.5 3–D Shells

This section contains shape functions for three–dimensional shell elements. These elements are available in a number of configurations, including certain combinations of the following features:

- triangular or quadrilateral.
  - ○ if quadrilateral, with or without extra shape functions (ESF).
- with or without rotational degrees of freedom (RDOF).
  - ○ if with RDOF, with or without shear deflections (SD).
- with or without midside nodes.



12.5.1 3–D 3–Node Triangular Shells without RDOF (CST)

These shape functions are for 3–D 3–node triangular shell elements without RDOFs, such as SHELL41 or SHELL57:

$$u = u_I L_1 + u_J L_2 + u_K L_3 \qquad (12.5.1\text{–}1)$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 \qquad (12.5.1–2)$$

$$w = w_I L_1 + w_J L_2 + w_K L_3 \qquad (12.5.1–3)$$

$$A_x = A_{xI} L_1 + A_{xJ} L_2 + A_{xK} L_3 \qquad (12.5.1–7)$$

$$A_y = A_{y_I} L_1 + A_{y_J} L_2 + A_{yK} L_3 \qquad (12.5.1–8)$$

$$A_z = A_{zI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \qquad (12.5.1–9)$$

$$T = T_I L_1 + T_J L_2 + T_K L_3 \qquad (12.5.1–20)$$

$$\phi = \phi_I L_1 + \phi_J L_2 + \phi_K L_3 \qquad (12.5.1–22)$$

12.5.2 3–D 6–Node Triangular Shells without RDOF (LST)

These shape functions are for 3–D 6–node triangular shell elements without RDOFs, such as the mass or stress stiffening matrix of SHELL93:

$$\begin{aligned} u = {} & u_I \left(2L_1 - 1\right) L_1 + u_J \left(2L_2 - 1\right) L_2 + u_K \left(2L_3 - 1\right) L_3 \\ & + u_L \left(4L_1L_2\right) + u_M \left(4L_2L_3\right) + u_N \left(4L_3L_1\right) \end{aligned} \qquad (12.5.2–1)$$

$$v = v_I \left(2L_1–1\right) \ldots \text{ (analogous to u)} \qquad (12.5.2–2)$$

$$w = w_I \left(2L_1–1\right) \ldots \text{ (analogous to u)} \qquad (12.5.2–3)$$

$$T = T_I \left(2L_1–1\right) \ldots \text{ (analogous to u)} \qquad (12.5.2–20)$$

$$V = V_I \left(2L_1 - 1\right) \ldots \text{ (analogous to u)} \qquad (12.5.2–21)$$

12.5.3 3–D 3–Node Triangular Shells with RDOF but without SD

These shape functions are for the 3–D 3–node triangular shell elements with RDOFs, but without shear deflection, such as SHELL63 when used as a triangle.

$$u = u_I L_1 + u_J L_2 + u_K L_3 \qquad (12.5.3\text{–}1)$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 \qquad (12.5.3\text{–}2)$$

w = not explicitly defined. A DKT element is used (Batoz(56), Razzaque(57)). (12.5.3–3)

12.5.4 3–D 3–Node Triangular Shells with RDOF and with SD

These shape functions are for 3–D 3–node triangular shell elements with RDOFs and with shear deflection, such as SHELL43 when used as a triangle.

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{3} N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{3} N_i \frac{rt_i}{2} \begin{bmatrix} a_{1,i} & b_{1,i} \\ a_{2,i} & b_{2,i} \\ a_{3,i} & b_{3,i} \end{bmatrix} \begin{Bmatrix} \theta_{x,i} \\ \theta_{y,i} \end{Bmatrix} \qquad \begin{matrix} (12.5.4\text{–}1) \\ (12.5.4\text{–}2) \\ (12.5.4\text{–}3) \end{matrix}$$

where:

N_i = shape functions given with u for PLANE42 (equation (12.6.1–1))

u_i = motion of node i

r = thickness coordinate

t_i = thickness at node i

{a} = unit vector in s direction

{b} = unit vector in plane of element and normal to {a}

$\theta_{x,i}$ = rotation of node i about vector {a}

$\theta_{y,i}$ = rotation of node i about vector {b}

Note that the nodal translations are in global Cartesian space, and the nodal rotations are based on element (s–t) space. Transverse shear strains in natural space (see Figure 12.5–1) are assumed as:

$$\tilde{\epsilon}_{13} = \frac{1}{2}(1+t)\,\tilde{\epsilon}_{13}^{A} + \frac{1}{2}(1-t)\,\tilde{\epsilon}_{13}^{C}$$

$$\tilde{\epsilon}_{23} = \frac{1}{2}(1+s)\,\tilde{\epsilon}_{23}^{D} + \frac{1}{2}(1-s)\,\tilde{\epsilon}_{23}^{B}$$

where:

$\tilde{\epsilon}_{13}^{A}$ = $\tilde{\epsilon}_{13}^{DI}$ | at A

$\tilde{\epsilon}_{13}^{C}$ = $\tilde{\epsilon}_{13}^{DI}$ | at C

$$\tilde{\epsilon}_{23}^{D} = \tilde{\epsilon}_{23}^{DI} \mid \text{at D}$$
$$\tilde{\epsilon}_{23}^{B} = \tilde{\epsilon}_{23}^{DI} \mid \text{at B}$$

DI stands for values computed from assumed displacement fields directly. These assumptions can be seen to cause geometric anisotropy. See Section 14.43 for more details. The in–plane RDOFs (KEYOPT(3) = 2) logic is based on Allman(113).

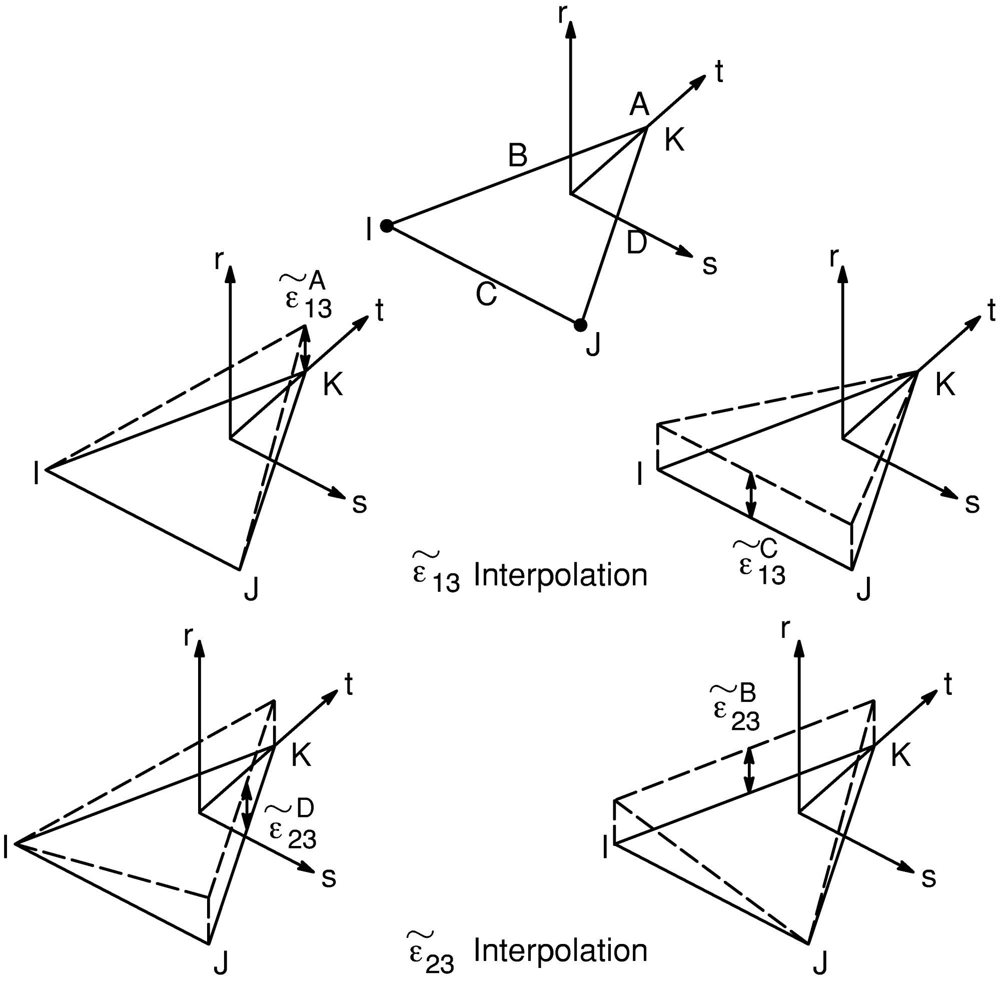

r
t
A
K
B
I
D
s
C
J
$\tilde{\varepsilon}_{13}^{A}$
$\tilde{\varepsilon}_{13}$ Interpolation
$\tilde{\varepsilon}_{13}^{C}$
$\tilde{\varepsilon}_{23}^{D}$
$\tilde{\varepsilon}_{23}^{B}$
$\tilde{\varepsilon}_{23}$ Interpolation

Figure 12.5–1 Interpolation Functions For The Transverse Strains

12.5.5 3–D 6–Node Triangular Shells with RDOF and with SD

These shape functions are for 3–D 6–node triangular shell elements with RDOFs and with shear deflection, such as for the stiffness matrix of SHELL93:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{6} N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{6} N_i \frac{rt_i}{2} \begin{bmatrix} a_{1,i} & b_{1,i} \\ a_{2,i} & b_{2,i} \\ a_{3,i} & b_{3,i} \end{bmatrix} \begin{Bmatrix} \theta_{x,i} \\ \\ \theta_{y,i} \end{Bmatrix} \qquad \begin{matrix} (12.5.5–1) \\ (12.5.5–2) \\ (12.5.5–3) \end{matrix}$$

where: N_i = shape functions given with u for PLANE2 (equation (12.6.2–1))

u_i, v_i, w_i = motion of node i

r = thickness coordinate

t_i = thickness at node i

{a} = unit vector in s direction

{b} = unit vector in plane of element and normal to {a}

$\theta_{x,i}$ = rotation of node i about vector {a}

$\theta_{y,i}$ = rotation of node i about vector {b}

Note that the nodal translations are in global Cartesian space, and the nodal rotations are based on element (s–t) space.

12.5.8 3–D 4–Node Quadrilateral Shells without RDOF and without ESF (Q4)

These shape functions are for 3–D 4–node triangular shell elements without RDOFs and without extra displacement shapes, such as SHELL41 with KEYOPT(2) = 1 and the magnetic interface element INTER115.

$$u = \frac{1}{4}\left(u_I (1 - s)(1 - t) + u_J (1 + s)(1 - t) + u_K (1 + s)(1 + t) + u_L (1 - s)(1 + t)\right) \qquad (12.5.8–1)$$

$$v = \frac{1}{4}\left(v_I (1–s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.5.8–2)$$

$$w = \frac{1}{4}\left(w_I (1–s) \ . \ . \ . \text{(analogous to u)}\right. \qquad (12.5.8–3)$$

$$A_x = \frac{1}{4}\left(A_{xI} (1 - s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.5.8–7)$$

$$A_y = \frac{1}{4}\left(A_{yI} (1 - s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.5.8–8)$$

$$A_z = \frac{1}{4}\left(A_{zI}(1-s)\text{ . . . (analogous to u)}\right. \tag{12.5.8–9}$$

$$T = \frac{1}{4}\left(T_I(1-s)\text{ . . . (analogous to u)}\right. \tag{12.5.8–20}$$

$$V = \frac{1}{4}\left(V_I(1-s)\text{ . . . (analogous to u)}\right. \tag{12.5.8–21}$$

$$\phi = \frac{1}{4}\left(\phi_I(1-s)\text{ . . . (analogous to u)}\right. \tag{12.5.8–22}$$

12.5.9 3–D 4–Node Quadrilateral Shells without RDOF but with ESF (QM6)

These shape functions are for 3–D 4–node quadrilateral shell elements without RDOFs but with extra shape functions, such as SHELL41 with KEYOPT(2) = 0:

$$\begin{aligned} u = \frac{1}{4}\Big(&u_I(1-s)(1-t) + u_J(1+s)(1-t) \\ &+ u_K(1+s)(1+t) + u_L(1-s)(1+t)\Big) \\ &+ u_1(1-s^2) + u_2(1-t^2) \end{aligned} \tag{12.5.9–1}$$

$$v = \frac{1}{4}\left(v_I(1\text{–}s)\text{ . . . (analogous to u)}\right. \tag{12.5.9–2}$$

12.5.10 3–D 8–Node Quadrilateral Shells without RDOF

These shape functions are for 3–D 8–node quadrilateral shell elements without RDOFs such as the mass or stress stiffening matrix of SHELL93:

$$\begin{aligned} u = \frac{1}{4}\Big(&u_I(1-s)(1-t)(-s-t-1) + u_J(1+s)(1-t)(s-t-1) \\ &+ u_K(1+s)(1+t)(s+t-1) + u_L(1-s)(1+t)(-s+t-1)\Big) \\ + \frac{1}{2}\Big(&u_M(1-s^2)(1-t) + u_N(1+s)(1-t^2) \\ &+ u_O(1-s^2)(1+t) + u_P(1-s)(1-t^2)\Big) \end{aligned} \tag{12.5.10–1}$$

$$v = \frac{1}{4}\left(v_I\ (1–s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.5.10–2)$$

$$w = \frac{1}{4}\left(w_I\ (1–s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.5.10–3)$$

$$T = \frac{1}{4}\left(T_I\ (1–s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.5.10–20)$$

$$V = \frac{1}{4}\left(V_I\ (1–s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.5.10–21)$$

12.5.11 3–D 4–Node Quadrilateral Shells with RDOF but without SD and without ESF

These shape functions are for 3–D 4–node quadrilateral shell elements with RDOFs but without shear deflection and without extra shape functions, such as SHELL63 with KEYOPT(3) = 1 when used as a quadrilateral:

$$\begin{aligned} u = \frac{1}{4}\Big(&u_I\ (1 - s)(1 - t) + u_J\ (1 + s)(1 - t) \\ &+ u_K (1 + s)(1 + t) + u_L\ (1 - s)(1 + t)\Big) \end{aligned} \qquad (12.5.11–1)$$

$$v = \frac{1}{4}\left(v_I\ (1–s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.5.11–2)$$

w = not explicitly defined. Four overlaid triangles (IJK, JKL, KLI, and LIJ) are defined as DKT elements (56), Razzaque(57). (12.5.11–3)

12.5.12 3–D 4–Node Quadrilateral Shells with RDOF but without SD and with ESF

These shape functions are for 3–D 4–node quadrilateral shell elements with RDOFs but without shear deflection and with extra shape functions, such as SHELL63 with KEYOPT(3) = 0 when used as a quadrilateral:

$$u = \frac{1}{4}\Big(u_I\,(1-s)(1-t) + u_J\,(1+s)(1-t)$$

$$+ u_K(1+s)(1+t) + u_L\,(1-s)(1+t)\Big) \qquad (12.5.12–1)$$

$$+ u_1(1-s^2) + u_2(1-t^2)\Big)$$

$$v = \frac{1}{4}\big(v_I\,(1–s)\;.\;.\;.\;\text{(analogous to u)} \qquad (12.5.12–2)$$

w = not explicitly defined. Four overlaid triangles (IJK, JKL, KLI, and LIJ) are defined as DKT elements, (Batoz(56), Razzaque(57)). (12.5.12–3)

12.5.13 3–D 4–Node Quadrilateral Shells with RDOF and with SD

These shape functions are for 3–D 4–node quadrilateral shell with RDOFs and with shear deflection such as SHELL43. Both use and nonuse of extra shape functions are considered.

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{4} N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{4} N_i \frac{rt_i}{2} \begin{bmatrix} a_{1,i} & b_{1,i} \\ a_{2,i} & b_{2,i} \\ a_{3,i} & b_{3,i} \end{bmatrix} \begin{Bmatrix} \theta_{x,i} \\ \\ \theta_{y,i} \end{Bmatrix} \qquad \begin{matrix} (12.5.13–1) \\ (12.5.13–2) \\ (12.5.13–3) \end{matrix}$$

where:

N_i = shape functions given with u for in equation (12.6.5–1). Extra shapes, if requested, use the shape functions of equation (12.6.6–1)

u_i, v_i, w_i = motion of node i

r = thickness coordinate

t_i = thickness at node i

{a} = unit vector in s direction

{b} = unit vector in plane of element and normal to {a}

$\theta_{x,i}$ = rotation of node i about vector {a}

$\theta_{y,i}$ = rotation of node i about vector {b}

Note that the nodal translations are in global Cartesian space, and the nodal rotations are based on element (s–t) space. Transverse shear strains in natural space (Figure 12.5–2) are assumed as:

$$\tilde{\epsilon}_{13} = \frac{1}{2}(1 + t)\,\tilde{\epsilon}_{13}^{A} + \frac{1}{2}(1 - t)\,\tilde{\epsilon}_{13}^{C}$$

$$\tilde{\epsilon}_{23} = \frac{1}{2}(1 + s)\,\tilde{\epsilon}_{23}^{D} + \frac{1}{2}(1 - s)\,\tilde{\epsilon}_{23}^{B}$$

where:

$\tilde{\epsilon}_{13}^{A} = \tilde{\epsilon}_{13}^{DI}$ | at A

$\tilde{\epsilon}_{13}^{C} = \tilde{\epsilon}_{13}^{DI}$ | at C

$\tilde{\epsilon}_{23}^{D} = \tilde{\epsilon}_{23}^{DI}$ | at D

$\tilde{\epsilon}_{23}^{B} = \tilde{\epsilon}_{23}^{DI}$ | at B

DI stands for values computed from assumed displacement fields directly (see Figure 12.5–2). See Section 14.43 for more details. The in–plane RDOFs logic is based on Yunus(117).

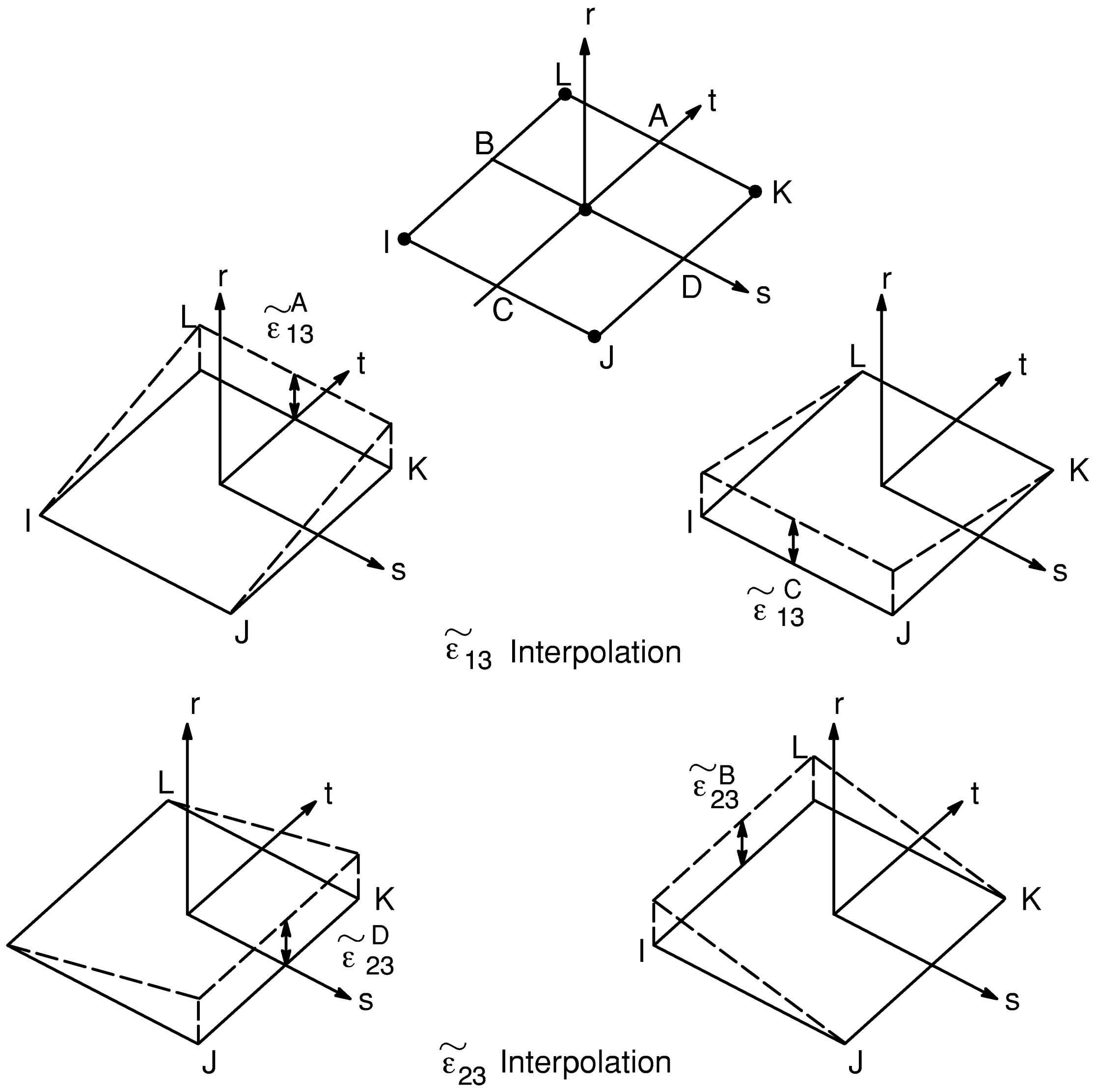

r
L
A
t
B
K
I
D
s
C
J
$\tilde{\varepsilon}^{A}_{13}$
$\tilde{\varepsilon}^{C}_{13}$
$\tilde{\varepsilon}_{13}$ Interpolation
$\tilde{\varepsilon}^{B}_{23}$
$\tilde{\varepsilon}^{D}_{23}$
$\tilde{\varepsilon}_{23}$ Interpolation

Figure 12.5–2 Interpolation Functions for the Transverse Strains

12.5.14 3–D 8–Node Quadrilateral Shells with RDOF and with SD

These shape functions are for 3–D 8–node quadrilateral shell elements with RDOFs and with shear deflection, such as for the stiffness matrix of SHELL93:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \sum_{i=1}^{8} N_i \begin{Bmatrix} u_i \\ v_i \\ w_i \end{Bmatrix} + \sum_{i=1}^{8} N_i \frac{rt_i}{2} \begin{bmatrix} a_{1,i} & b_{1,i} \\ a_{2,i} & b_{2,i} \\ a_{3,i} & b_{3,i} \end{bmatrix} \begin{Bmatrix} \theta_{x,i} \\ \theta_{y,i} \end{Bmatrix}$$

(12.5.14–1)

(12.5.14–2)

(12.5.14–3)

where: N_i = shape functions given with u for PLANE82 equation (12.6.7–1)

u_i, v_i, w_i = motion of node i

r = thickness coordinate

t_i = thickness at node i

{a} = unit vector in s direction

{b} = unit vector in plane of element and normal to {a}

$\theta_{x,i}$ = rotation of node i about vector {a}

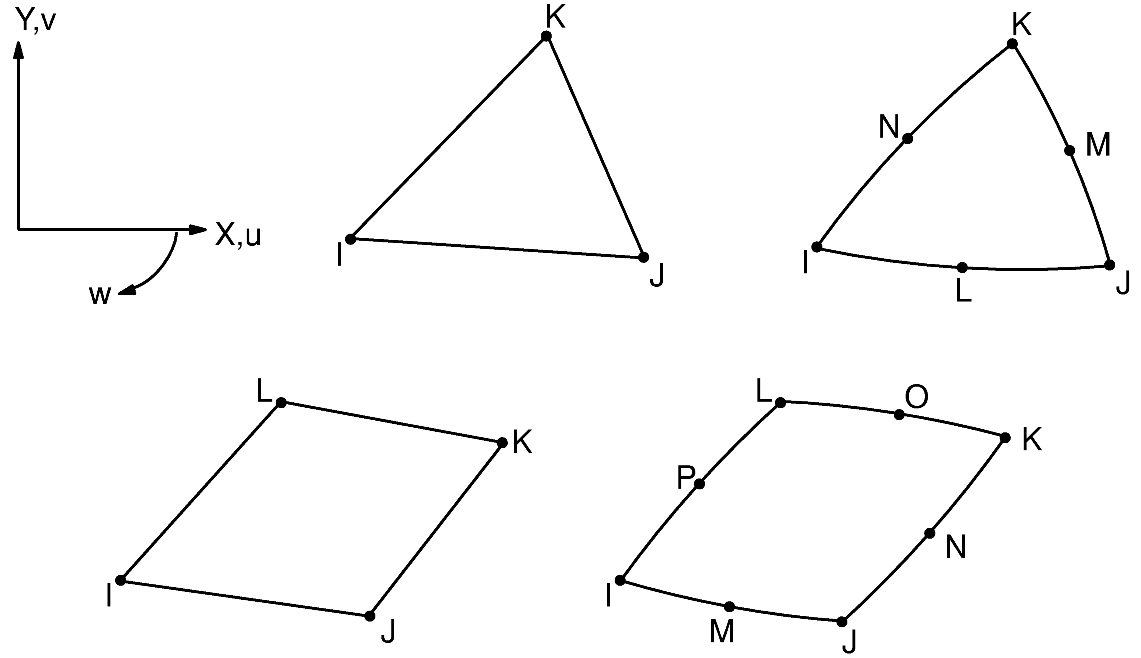
$\theta_{y,i}$ = rotation of node i about vector {b}

Note that the nodal translations are in global Cartesian space, and the nodal rotations are based on element (s–t) space.

12.6 2–D and Axisymmetric Solids

This section contains shape functions for two–dimensional and axisymmetric solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

- triangular or quadrilateral.
  - ○ if quadrilateral, with or without extra shape functions (ESF).
- with or without midside nodes.



12.6.1 2–D and Axisymmetric 3 Node Triangular Solids (CST)

These shape functions are for 2–D 3 node and axisymmetric triangular solid elements, such as PLANE13, PLANE42, PLANE67, or FLUID141 with only 3 nodes input:

$$u = u_I L_1 + u_J L_2 + u_K L_3 \qquad (12.6.1–1)$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 \qquad (12.6.1–2)$$

$$w = w_I L_1 + w_J L_2 + w_K L_3 \qquad (12.6.1–3)$$

$$A_z = A_{zI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \qquad (12.6.1–9)$$

$$V_x = V_{xI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \qquad (12.6.1–10)$$

$$V_y = V_{yI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \qquad (12.6.1–11)$$

$$V_z = V_{zI} L_1 + A_{zJ} L_2 + A_{zK} L_3 \qquad (12.6.1–12)$$

$$P = P_I L_1 + A_{zJ} L_2 + A_{zK} L_3 \qquad (12.6.1–19)$$

$$T = T_I L_1 + T_J L_2 + T_K L_3 \qquad (12.6.1–20)$$

$$V = V_I L_1 + V_J L_2 + V_K L_3 \qquad (12.6.1–21)$$

$$E^K = E_I^K L_1 + V_J L_2 + V_K L_3 \qquad (12.6.1–23)$$

$$E^D = E_I^D L_1 + V_J L_2 + V_K L_3 \qquad (12.6.1–24)$$

12.6.2 2–D and Axisymmetric 6 Node Triangular Solids (LST)

These shape functions are for 2–D 6 node and axisymmetric triangular solids, such as PLANE2 or PLANE35 (or PLANE77 or PLANE82 reduced to a triangle):

$$\begin{aligned} u = {} & u_I(2L_1 - 1)\,L_1 + u_J(2L_2 - 1)\,L_2 + u_K(2L_3 - 1) \\ & + u_L(4L_1L_2) + u_M(4L_2L_3) + u_N(4L_3L_1) \end{aligned} \qquad (12.6.2–1)$$

$$v = v_I(2L_1–1)\ \ L_1 + \ldots \text{ analogous to u)} \qquad (12.6.2–2)$$

$$w = w_I(2L_1–1)\ \ L_1 + \ldots \text{ analogous to u)} \qquad (12.6.2–3)$$

$$A_z = A_{zI}(2L_1–1)\,L_1 + \ldots \text{ analogous to u)} \qquad (12.6.2–9)$$

$$T = T_I(2L_1{-}1)\,L_1 + \ldots \text{ analogous to u)} \qquad (12.6.2–20)$$

$$V = V_I(2L_1{-}1)\,L_1 + \ldots \text{ analogous to u)} \qquad (12.6.2–21)$$

12.6.5 2–D and Axisymmetric 4 Node Quadrilateral Solid without ESF (Q4)

These shape functions are for the 2–D 4 node and axisymmetric quadrilateral solid elements without extra shape functions, such as PLANE13 with KEYOPT(2) = 1, PLANE42 with KEYOPT(2) = 1, LINK68, or FLUID141.

$$\begin{aligned} u = \frac{1}{4}\big(&u_I(1-s)(1-t) + u_J(1+s)(1-t) \\ &+ u_K(1+s)(1+t) + u_L(1-s)(1+t)\big) \end{aligned} \qquad (12.6.5–1)$$

$$v = \frac{1}{4}\left(v_I(1{-}s)\ldots \text{(analogous to u)}\right. \qquad (12.6.5–2)$$

$$w = \frac{1}{4}\left(w_I(1{-}s)\ldots \text{(analogous to u)}\right. \qquad (12.6.5–3)$$

$$A_z = \frac{1}{4}\left(A_{zI}(1{-}s)\ldots \text{(analogous to u)}\right. \qquad (12.6.5–9)$$

$$V_x = \frac{1}{4}\, V_{xI}(1{-}s)\ldots \text{(analogous to u)} \qquad (12.6.5–10)$$

$$V_y = \frac{1}{4}\, V_{yI}(1{-}s)\ldots \text{(analogous to u)} \qquad (12.6.5–11)$$

$$V_z = \frac{1}{4}\, V_{zI}(1{-}s)\ldots \text{(analogous to u)} \qquad (12.6.5–12)$$

$$P = \frac{1}{4}\, P_I(1{-}s)\ldots \text{(analogous to u)} \qquad (12.6.5–19)$$

$$T = \frac{1}{4}\left(T_I(1{-}s)\ldots \text{(analogous to u)}\right. \qquad (12.6.5–20)$$

$$V = \frac{1}{4}\left(V_I(1{-}s)\ldots \text{(analogous to u)}\right. \qquad (12.6.5–21)$$

$$E^K = \frac{1}{4}\left(E_I^K(1{-}s)\ .\ .\ .\ \text{(analogous to u)}\right. \tag{12.6.5–23}$$

$$E^D = \frac{1}{4}\left(E_I^D(1{-}s)\ .\ .\ .\ \text{(analogous to u)}\right. \tag{12.6.5–24}$$

12.6.6 2–D and Axisymmetric 4 Node Quadrilateral Solids with ESF (QM6)

These shape functions are for the 2–D 4 node and axisymmetric solid elements with extra shape functions, such as PLANE13 with KEYOPT(2) = 0 or PLANE42 with KEYOPT(2) = 0. (Taylor et al(49))

$$\begin{aligned} u = \frac{1}{4}\big(&u_I(1-s)(1-t) + u_J(1+s)(1-t) + u_K(1+s)(1+t) \\ &+ u_L(1-s)(1+t)\big) + u_1(1-s^2) + u_2(1-t^2) \end{aligned} \tag{12.6.6–1}$$

$$v = \frac{1}{4}\left(v_I(1{-}s)\ .\ .\ .\ \text{(analogous to u)}\right. \tag{12.6.6–2}$$

Equation (12.6.6–1) is adjusted for axisymmetric situations by removing the u_1 or u_2 term for elements near the centerline, in order to avoid holes or “doubled” material at the centerline.

12.6.7 2–D and Axisymmetric 8 Node Quadrilateral Solids (Q8)

These shape functions are for the 2–D 8 node and axisymmetric quadrilateral elements such as PLANE77 and PLANE82:

$$\begin{aligned} u = \frac{1}{4}\big(&u_I(1-s)(1-t)(-s-t-1) + u_J(1+s)(1-t)(s-t-1) \\ &+ u_K(1+s)(1+t)(s+t-1) + u_L(1-s)(1+t)(-s+t-1)\big) \\ &+ \frac{1}{2}\big(u_M(1-s^2)(1-t) + u_N(1+s)(1-t^2) \\ &+ u_O(1-s^2)(1+t) + u_P(1-s)(1-t^2)\big) \end{aligned} \tag{12.6.7–1}$$

$$v = \frac{1}{4}\left(v_I(1{-}s)\ .\ .\ .\ \text{(analogous to u)}\right. \tag{12.6.7–2}$$

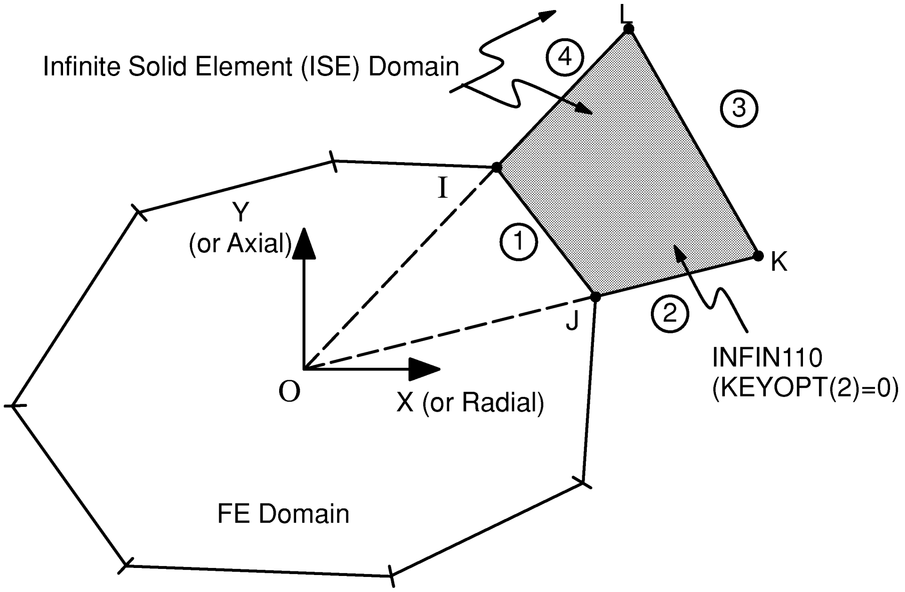
$$w = \frac{1}{4}\left(w_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \tag{12.6.7–3}$$

$$A_z = \frac{1}{4}\left(A_{zI}(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \tag{12.6.7–9}$$

$$T = \frac{1}{4}\left(T_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \tag{12.6.7–20}$$

$$V = \frac{1}{4}\left(V_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \tag{12.6.7–21}$$

12.6.8 2–D and Axisymmetric 4 Node Quadrilateral Infinite Solids



These Lagrangian isoparametric shape functions and “mapping” functions are for the 2–D and axisymmetric 4 node quadrilateral solid infinite elements such as INFIN110:

Lagrangian Isoparametric Shape Functions:

$$\begin{aligned} A_z &= \frac{1}{4}\left(A_{zI}\ (1-s)\ (t^2-t) + A_{zJ}\ (1+s)\ (t^2-t)\right) \\ &+ \frac{1}{2}\left(A_{zK}\ (1+s)\ (1-t^2) + A_{zL}\ (1-s)\ (1-t^2)\right) \end{aligned} \tag{12.6.8–9}$$

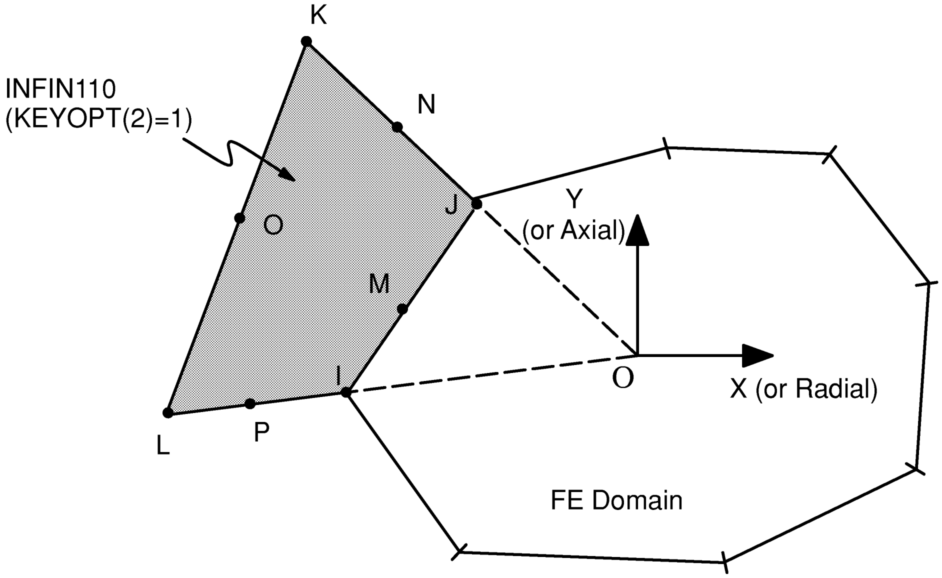
$$T = \frac{1}{4}\left(T_I(1-s) \ \ . \ . \ . \ \text{(analogous to } A_z)\right. \tag{12.6.8–20}$$

$$V = \frac{1}{4}\left(V_I(1 - s)\right. \ldots \text{ (analogous to } A_z\text{)} \qquad (12.6.8\text{–}21)$$

Mapping Functions:

$$x = x_I (1 - s)(- t) / (1 - t) + x_J (1 + s)(- t) / (1 - t) + \frac{1}{2} x_K (1 + s)(1 + t) / (1 - t) + \frac{1}{2} x_L (1 - s)(1 + t) / (1 - t) \qquad (12.6.8\text{–}33)$$

$$y = y_I (1 - s) \ \ldots \text{ (analogous to x)} \qquad (12.6.8\text{–}34)$$



12.6.9 2–D and Axisymmetric 8 Node Quadrilateral Infinite Solids

These Lagrangian isoparametric shape functions and “mapping” functions are for the 2–D and axisymmetric 8 node quadrilateral infinite solid elements such as INFIN110:

Lagrangian Isoparametric Shape Functions:

$$\begin{aligned} A_z = {} & \frac{1}{4}\left(A_{zI}\ (1 - s)(1 - t)(-1 - s - t)\right) + \frac{1}{2}\left(A_{zJ}\ (1 - s^2)(1 - t)\right) \\ & + \frac{1}{4}\left(A_{zK}\ (1 + s)(1 - t)(-1 + s - t)\right) + \frac{1}{2}\left(A_{zL}\ (1 + s)(1 - t^2)\right) \\ & + \frac{1}{2}\left(A_{zM}\ (1 - s)(1 - t^2)\right) \end{aligned} \qquad (12.6.9\text{–}9)$$

$$T = \left(T_I(1 - s)\right. \ \ldots \text{ (analogous to } A_z\text{)} \qquad (12.6.9\text{–}20)$$

$$V = (V_I(1 - s) \ldots \text{ (analogous to } A_z) \tag{12.6.9–21}$$

Mapping Functions:

$$\begin{aligned} x = {} & x_I\ (1 - s)(-1 - s - t) / (1 - t) + 2x_J\ (1 - s^2) / (1 - t) \\ & + x_K\ (1 + s)(-1 + s - t) / (1 - t) + \frac{1}{2}\, x_L\ (1 + s)(1 + t) / (1 - t) \\ & + \frac{1}{2}\, x_M\ (1 - s)(1 + t) / (1 - t) \end{aligned} \tag{12.6.9–33}$$

$$y = y_I\ (1 - s) \ldots \text{ (analogous to x)} \tag{12.6.9–34}$$

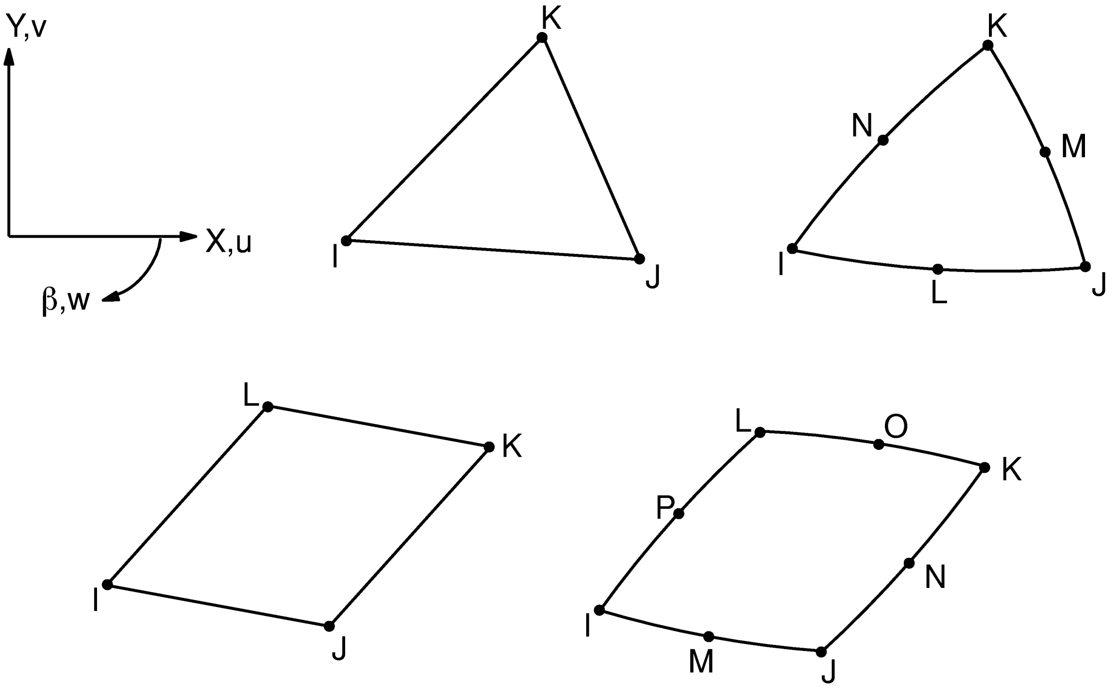
The shape and mapping functions for the nodes N, O and P are deliberately set to zero.

12.7 Axisymmetric Harmonic Solids

This section contains shape functions for axisymmetric harmonic solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

- triangular or quadrilateral.
    - ○ if quadrilateral, with or without extra shape functions (ESF).
- with or without midside nodes.

The shape functions of this section use the quantities $\sin\ell\beta$ and $\cos\ell\beta$, where ℓ = input quantity MODE on the **MODE** command. $\sin\ell\beta$ and $\cos\ell\beta$ are interchanged if $I_s = -1$, where I_s = input quantity ISYM on the **MODE** command. If $\ell = 0$, $\sin\ell\beta = \cos\ell\beta = 1.0$.



12.7.1 Axisymmetric Harmonic 3 Node Triangular Solids

These shape functions are for the 3 node axisymmetric triangular solid elements, such as PLANE25 with only 3 nodes input:

$$u = \left(u_I L_1 + u_J L_2 + u_K L_3\right) \cos\ell\beta \qquad (12.7.1\text{–}1)$$

$$v = \left(v_I L_1 + v_J L_2 + v_K L_3\right) \cos\ell\beta \tag{12.7.1–2}$$

$$w = \left(w_I L_1 + w_J L_2 + w_K L_3\right) \sin\ell\beta \tag{12.7.1–3}$$

$$T = \left(T_I L_1 + T_J L_2 + T_K L_3\right) \cos\ell\beta \tag{12.7.1–20}$$

12.7.2 Axisymmetric Harmonic 6 Node Triangular Solids

These shape functions are for the 6 node axisymmetric triangular solids elements, such as PLANE83 input as a triangle:

$$\begin{aligned} u = \Big(& u_I (2L_1 - 1)\, L_1 + u_J (2L_2 - 1)\, L_2 + u_K (2L_3 - 1) \\ & + u_L (4L_1 L_2) + u_M (4L_2 L_3) + u_N (4L_3 L_1) \Big) \cos\ell\beta \end{aligned} \tag{12.7.2–1}$$

$$v = \left(v_I (2L_1 – 1) \;\ldots\; \text{(analogous to u)} \;\ldots\right) \cos\ell\beta \tag{12.7.2–2}$$

$$w = \left(w_I (2L_1 – 1) \;\ldots\; \text{(analogous to u)} \;\ldots\right) \sin\ell\beta \tag{12.7.2–3}$$

$$T = \left(T_I (2L_1 – 1) \;\ldots\; \text{(analogous to u)} \;\ldots\right) \cos\ell\beta \tag{12.7.2–20}$$

12.7.5 Axisymmetric Harmonic 4 Node Quadrilateral Solids without ESF

These shape functions are for the 4 node axisymmetric harmonic quadrilateral solid elements without extra shape functions, such as PLANE25 with KEYOPT(2) = 1, or PLANE75:

$$\begin{aligned} u = \frac{1}{4} \big(& u_I (1 - s)\,(1 - t) + u_J\,(1 + s)\,(1 - t) \\ & + u_K (1 + s)(1 + t) + u_L (1 - s)(1 + t)\big) \cos\ell\beta \end{aligned} \tag{12.7.5–1}$$

$$v = \frac{1}{4}\left(v_I (1–s) \;\ldots\; \text{(analogous to u)} \;\ldots\right) \cos\ell\beta \tag{12.7.5–2}$$

$$w = \frac{1}{4}\left(w_I (1–s) \;\ldots\; \text{(analogous to u)}. \;.\;.\right) \sin\ell\beta \tag{12.7.5–3}$$

$$T = \frac{1}{4}\left(T_I(1-s)\ \ldots\ \text{(analogous to u)}\ \ldots\right)\cos\ell\beta \tag{12.7.5–20}$$

12.7.6 Axisymmetric Harmonic 4 Node Quadrilateral Solids with ESF

These shape functions are for the 4 node axisymmetric harmonic quadrilateral elements with extra shape functions, such as PLANE25 with KEYOPT(2) = 0.

$$\begin{aligned} u = \Big(\frac{1}{4}\big(&u_I(1-s)(1-t) + u_J(1+s)(1-t) + u_K(1+s)(1+t) \\ &+ u_L(1-s)(1+t)\big) + u_1(1-s^2) + u_2(1-t^2)\Big)\cos\ell\beta \end{aligned} \tag{12.7.6–1}$$

$$v = \left(\frac{1}{4}\left(v_I(1-s)\ \ldots\ \text{(analogous to u)}\ \ldots\right)\right)\cos\ell\beta \tag{12.7.6–2}$$

$$w = \left(\frac{1}{4}\left(w_I(1-s)\ \ldots\ \text{(analogous to u)}\ \ldots\right)\right)\sin\ell\beta \tag{12.7.6–3}$$

Unless ℓ (MODE) = 1, u_1 or u_2 and w_1 or w_2 motions are suppressed for elements near the centerline.

12.7.7 Axisymmetric Harmonic 8 Node Quadrilateral Solids

These shape functions are for the 8 node axisymmetric harmonic quadrilateral solid elements such as PLANE78 or PLANE83.

$$\begin{aligned} u = \Big(\frac{1}{4}\big(&u_I(1-s)(1-t)(-s-t-1) + u_J(1+s)(1-t)(s-t-1) \\ &+ u_K(1+s)(1+t)(s+t-1) + u_L(1-s)(1+t)(-s+t-1)\big) \\ + \frac{1}{2}\big(&u_M(1-s^2)(1-t) + u_N(1+s)(1-t^2) \\ &+ u_O(1-s^2)(1+t) + u_P(1-s)(1-t^2)\big)\Big)\cos\ell\beta \end{aligned} \tag{12.7.7–1}$$

$$v = \left(\frac{1}{4}\left(v_I(1-s)\ \ldots\ \text{(analogous to u)}\ \ldots\right)\right)\cos\ell\beta \tag{12.7.7–2}$$

$$w = \left(\frac{1}{4}\ \left(w_I(1-s)\ .\ .\ .\ \ \text{(analogous to u)}\ .\ .\ .\right) \sin\ell\beta \qquad (12.7.7\text{–}3)$$

$$T = \left(\frac{1}{4}\ \left(T_I(1-s)\ .\ .\ .\ \ \text{(analogous to u)}\ .\ .\ .\right) \cos\ell\beta \qquad (12.7.7\text{–}20)$$

12.8 3–D Solids

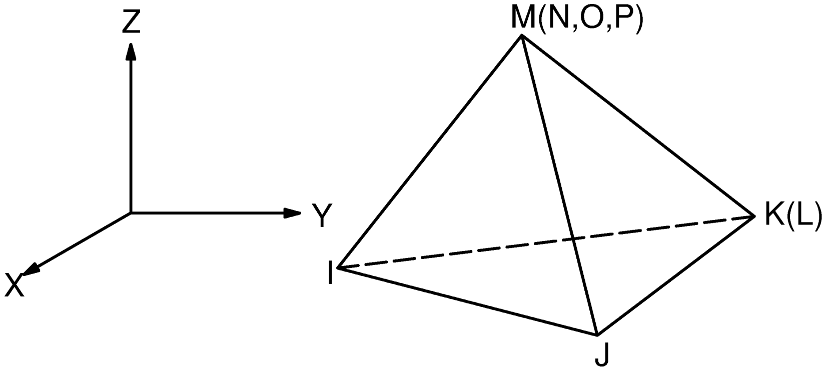
This section contains shape functions for three–dimensional solid elements. These elements are available in a number of configurations, including certain combinations of the following features:

- element shapes may be tetrahedra, pyramids, wedges, or bricks (hexahedra).
  - ○ if wedges or bricks, with or without extra shape functions (ESF).
- with or without rotational degrees of freedom (RDOF)
- with or without midside nodes

The wedge elements with midside nodes (15 node wedges) are either a condensation of the 20 node brick element or are based on wedge shape functions.

12.8.1 4 Node Tetrahedra without RDOF

This element is a condensation of an 8 node brick element such as SOLID5, FLUID30, SOLID45 or FLUID142



The resulting effective shape functions are:

$$u = u_I L_1 + u_J L_2 + u_K L_3 + u_M L_4 \tag{12.8.1–1}$$

$$v = v_I L_1 + v_J L_2 + v_K L_3 + v_M L_4 \tag{12.8.1–2}$$

$$w = w_I L_1 + w_J L_2 + w_K L_3 + w_M L_4 \tag{12.8.1–3}$$

$$V_x = V_{xI}L_1 + w_JL_2 + w_KL_3 + w_ML_4 \quad (12.8.1–10)$$

$$V_y = V_{yI}L_1 + w_JL_2 + w_KL_3 + w_ML_4 \quad (12.8.1–11)$$

$$V_z = V_{zI}L_1 + w_JL_2 + w_KL_3 + w_ML_4 \quad (12.8.1–12)$$

$$P = P_IL_1 + P_JL_2 + P_KL_3 + P_ML_4 \quad (12.8.1–19)$$

$$T = T_IL_1 + T_JL_2 + T_KL_3 + T_ML_4 \quad (12.8.1–20)$$

$$V = V_IL_1 + V_JL_2 + V_KL_3 + V_ML_4 \quad (12.8.1–21)$$

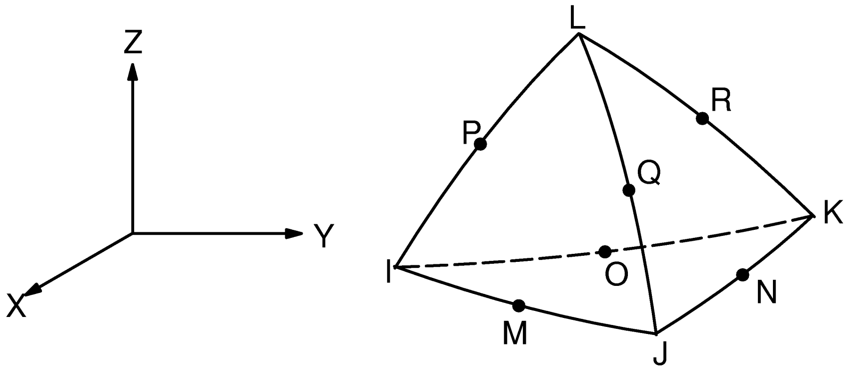
$$\phi = \phi_IL_1 + \phi_JL_2 + \phi_KL_3 + \phi_ML_4 \quad (12.8.1–22)$$

$$E^K = E_I^KL_1 + E_J^KL_2 + E_K^KL_3 + E_M^KL_4 \quad (12.8.1–23)$$

$$E^D = E_I^DL_1 + E_J^DL_2 + E_K^DL_3 + E_M^DL_4 \quad (12.8.1–24)$$

12.8.2 10 Node Tetrahedra

These shape functions are for 10 node tetrahedron elements such as SOLID98 and SOLID92:



$$u = u_I(2L_1 - 1)L_1 + u_J(2L_2 - 1)L_2 + u_K(2L_3 - 1)L_3$$
$$+ u_L(2L_4 - 1)L_4 + 4\,(u_M L_1 L_2 + u_N L_2 L_3 + u_O L_1 L_3 \qquad (12.8.2\text{–}1)$$
$$+ u_P L_1 L_4 + u_Q L_2 L_4 + u_R L_3 L_4)$$

$$v = v_I(2L_1{-}1)L_1 \ldots \text{(analogous to u)} \qquad (12.8.2\text{–}2)$$

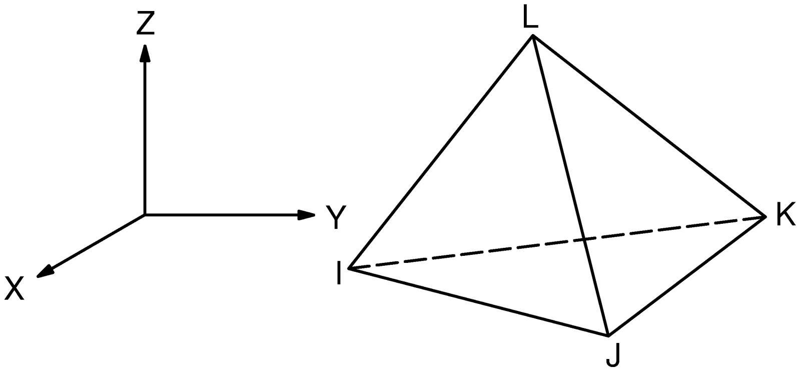
$$w = w_I(2L_1{-}1)L_1 \ldots \text{(analogous to u)} \qquad (12.8.2\text{–}3)$$

$$T = T_I(2L_1{-}1)L_1 \ldots \text{(analogous to u)} \qquad (12.8.2\text{–}20)$$

$$V = V_I(2L_1{-}1)L_1 \ldots \text{(analogous to u)} \qquad (12.8.2\text{–}21)$$

$$\phi = \phi_I(2L_1{-}1)L_1 \ldots \text{(analogous to u)} \qquad (12.8.2\text{–}22)$$

12.8.3 4 Node Tetrahedra with RDOF



There are no explicit shape functions for the 4 node tetrahedron element with RDOFs. Rather, a 10 node tetrahedron without RDOFs (for a total of 30 DOFs) is generated first. Then the motion of the midside node parallel to the edge of the midside node is condensed out and the remaining perpendicular motions converted to corner rotations (see Yunus et al(117)).

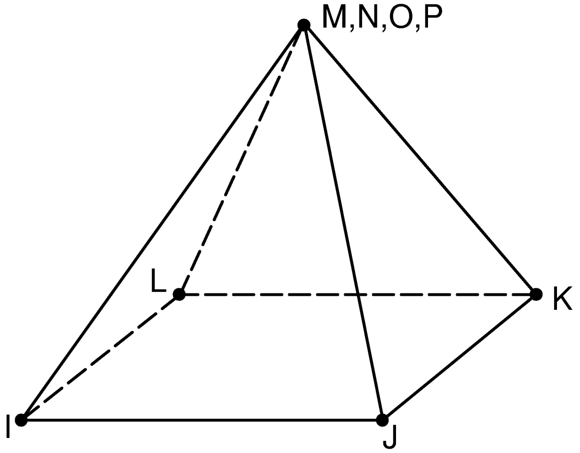
$$u = \text{condensed shape function as described above} \qquad (12.8.3\text{–}1)$$

$$v = \text{condensed shape function as described above} \qquad (12.8.3\text{–}2)$$

$$w = \text{condensed shape function as described above} \qquad (12.8.3\text{–}3)$$

12.8.6 5 Node Pyramids without RDOF

This element is a condensation of an 8 node brick element.

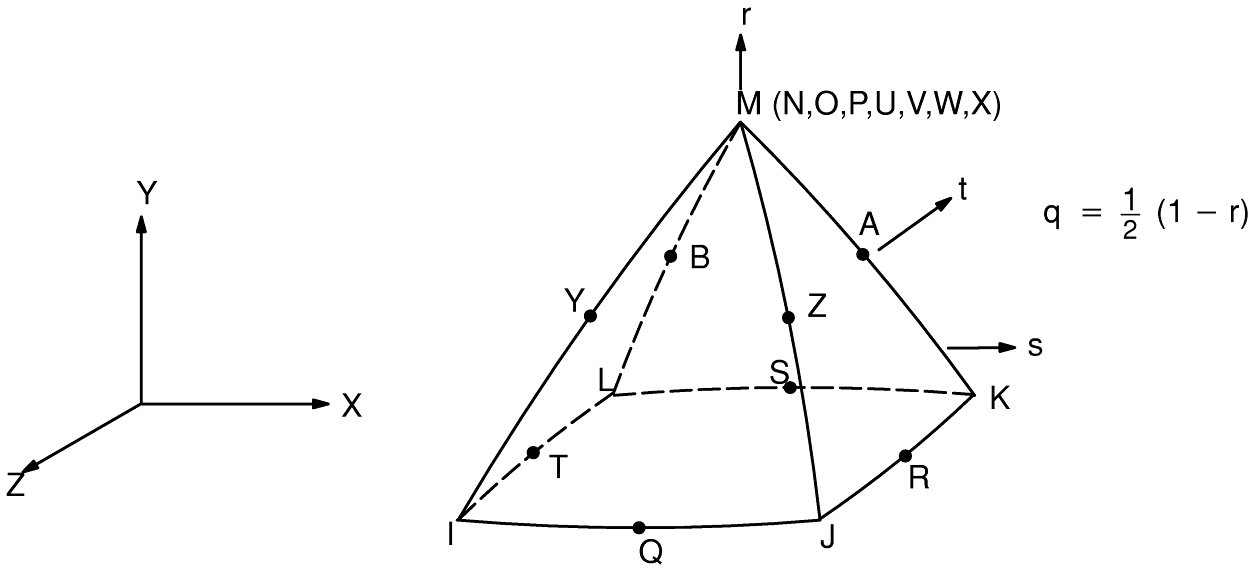


The resulting effective shape functions are:

$$\begin{aligned} T = \frac{1}{8} \big(& T_I(1-s)(1-t)(1-r) + T_J(1+s)(1-t)(1-r) \\ & + T_K(1+s)(1+t)(1-r) + T_L(1-s)(1+t)(1-r) \big) \qquad (12.8.6\text{–}20) \\ & + \frac{1}{2} T_M(1+r) \end{aligned}$$

12.8.7 13 Node Pyramids without RDOF

These shape functions are for 13 node pyramid elements which are based on a condensation of a 20 node brick element such as SOLID95:



$$
\begin{aligned}
u = \frac{q}{4}\Big(& u_I(1-s)(1-t)(-1-qs-qt) + u_J(1+s)(1-t)(-1+qs-qt) \\
&+ u_K(1+s)(1+t)(-1+sq+qt) + u_L(1-s)(1+t)(-1-qs+qt)\Big) \\
&+ u_M(1-q)(1-2q) \\
+ \frac{q^2}{2}\Big(& u_Q(1-t)(1-s^2) + u_R(1+s)(1-t^2) + u_S(1+t)(1-s^2) \\
&+ u_T(1-s)(1-t^2)\Big) \\
+ q(1-q)\Big(& u_Y(1-s-t+st) + u_Z(1+s-t-st) + u_A(1+s+t+st) \\
&+ u_B(1-s+t-st)\Big)
\end{aligned}
\qquad (12.8.7–1)
$$

$$v = \frac{q}{4}\Big(v_I(1-s)\ \ldots\ \text{(analogous to u)} \qquad (12.8.7–2)$$

$$w = \frac{q}{4}\Big(w_I(1-s)\ \ldots\ \text{(analogous to u)} \qquad (12.8.7–3)$$

$$T = \frac{q}{4}\Big(T_I(1-s)\ \ldots\ \text{(analogous to u)} \qquad (12.8.7–20)$$

$$V = \frac{q}{4}\Big(V_I(1-s)\ \ldots\ \text{(analogous to u)} \qquad (12.8.7–21)$$

12.8.8 5 Node Pyramids with RDOF

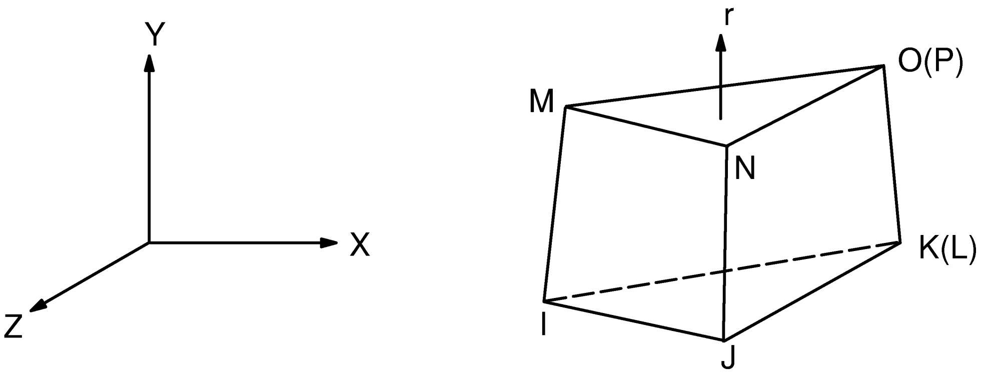
Explicit shape functions are not developed for the primary motions of the 5 node pyramid elements with RDOFs. Rather, a 13 node pyramid without RDOFs (for a total of 39 DOFs) is generated first. Then the motion parallel to the edge is condensed out and the remaining perpendicular motions converted to corner rotations (see Yunus et al(117)).

$$u = \text{condensed shape function as described above} \qquad (12.8.8\text{–}1)$$

$$v = \text{condensed shape function as described above} \qquad (12.8.8\text{–}2)$$

$$w = \text{condensed shape function as described above} \qquad (12.8.8\text{–}3)$$

12.8.11 6 Node Wedges without ESF and without RDOF



The 6 node wedge elements are a condensation of an 8 node brick such as SOLID5, FLUID30, or SOLID45. These shape functions are for 6 node wedge elements without extra shape functions:

$$\begin{aligned} u = \frac{1}{2}\Big(&u_I L_1(1-r) + u_J L_2(1-r) + u_K L_3(1-r) \\ &+ u_M L_1(1+r) + u_N L_2(1+r) + u_O L_3(1+r)\Big) \end{aligned} \qquad (12.8.11\text{–}1)$$

$$v = \frac{1}{2}\big(v_I L_1(1\text{–}r) \ldots \text{(analogous to u)} \qquad (12.8.11\text{–}2)$$

$$w = \frac{1}{2}\big(w_I L_1(1\text{–}r) \ldots \text{(analogous to u)} \qquad (12.8.11\text{–}3)$$

$$P = \frac{1}{2} \left(P_I L_1 (1{-}r) \ . \ . \ . \ \text{(analogous to u)} \right. \tag{12.8.11–19}$$

$$T = \frac{1}{2} \left(T_I L_1 (1{-}r) \ . \ . \ . \ \text{(analogous to u)} \right. \tag{12.8.11–20}$$

$$V = \frac{1}{2} \left(V_I L_1 (1{-}r) \ . \ . \ . \ \text{(analogous to u)} \right. \tag{12.8.11–21}$$

$$\phi = \frac{1}{2} \left(\phi_I L_1 (1{-}r) \ . \ . \ . \ \text{(analogous to u)} \right. \tag{12.8.11–22}$$

12.8.12 6 Node Wedges with ESF but without RDOF

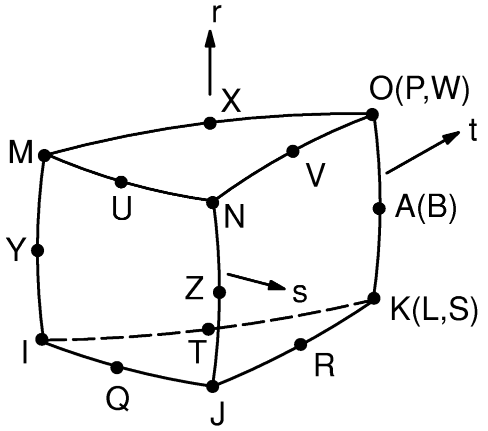
The 6 node wedge elements are a condensation of an 8 node brick such as SOLID5, FLUID30, or SOLID45. (Please see the illustration in Section 12.8.11.) These shape functions are for 6 node wedge elements with extra shape functions:

$$\begin{aligned} u = \frac{1}{2} \Big(& u_I L_1 (1 - r) + u_J L_2 (1 - r) + u_K L_3 (1 - r) \\ & + u_M L_1 (1 + r) + u_N L_2 (1 + r) + u_O L_3 (1 + r)\Big) + u_1 (1 - r^2) \end{aligned} \tag{12.8.12–1}$$

$$v = \frac{1}{2} \left(v_I L_1 (1{-}r) \ \ . \ . \ . \ \text{(analogous to u)} \right. \tag{12.8.12–2}$$

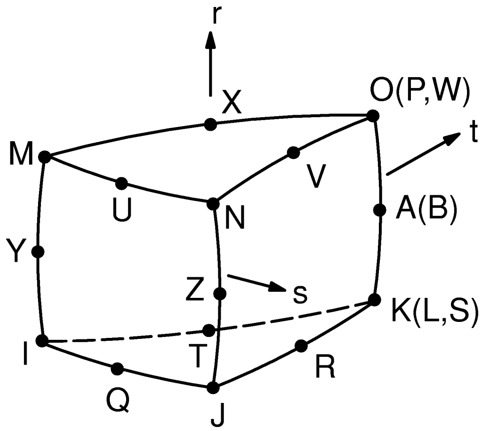
$$w = \frac{1}{2} \left(w_I L_1 (1{-}r) \ \ . \ . \ . \ \text{(analogous to u)} \right. \tag{12.8.12–3}$$

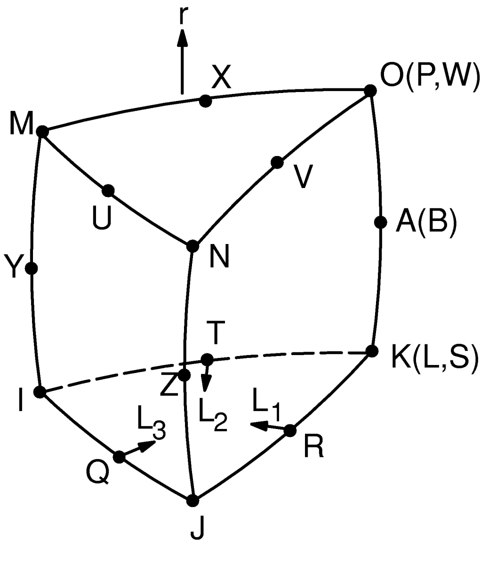
12.8.13 15 Node Wedges as a Condensation of the 20 Node Brick



These shape functions are for 15 node wedge elements such as SOLID90 that are based on a condensation of a 20 node brick element equation (12.8.20–1).

12.8.14 15 Node Wedges Based on Wedge Shape Functions





Elements such as SOLID95 in a wedge configuration use shape functions based on triangular coordinates and the r coordinate going from –1.0 to +1.0.

$$\begin{aligned} u = \frac{1}{2} \; & \Big(u_I\Big(L_1(2L_1 - 1)(1 - r) \; - L_1\big(1 - r^2\big)\Big) + u_J(L_2 - 1)(1 - r) - L_2\big(1 - r^2\big)\Big) \\ & + u_K\Big(L_3(2L_3 - 1)(1 - r) \; - L_3\big(1 - r^2\big)\Big) + u_M(L_1(2L_1 - 1)(1 + r) \\ & - L_1(1 - r^2)\big) + u_N\big(L_2(2L_2 - 1)(1 + r) - L_2(1 - r^2)\big) \\ & + u_O\Big(L_3(2L_3 - 1)(1 + r) - L_3(1 - r^2)\big)\Big) + 2\big(u_Q L_1 L_2(1 - r) \\ & + u_R L_2 L_3(1 - r) + u_T L_3 L_1(1 - r) + u_U L_1 L_2(1 + r) \\ & + u_V L_2 L_3(1 + r) + u_X L_3 L_1(1 + r)\big) + u_Y L_1\big(1 - r^2\big) \\ & + u_Z L_2\big(1 - r^2\big) + u_A L_3\big(1 - r^2\big) \end{aligned} \tag{12.8.14–1}$$

$$v = \frac{1}{2} \; \big(v_I L_1(2L_1{-}1\big) \; . \; . \; . \; \text{(analogous to u)} \tag{12.8.14–2}$$

$$w = \frac{1}{2} \; \big(w_I L_1(2L_1{-}1\big) \; . \; . \; . \; \text{(analogous to u)} \tag{12.8.14–3}$$

$$T = \frac{1}{2} \; \big(T_I L_1(2L_1{-}1\big) \; . \; . \; . \; \text{(analogous to u)} \tag{12.8.14–20}$$

12.8.15 6 Node Wedges with RDOF

Explicit shape functions are not developed for the primary motions of the 6 node pyramid elements with RDOFs. Rather, a 15 node wedge based on wedge shape functions without RDOFs (for a total of 45 DOFs) is generated first. Then the motion of the midside nodes parallel to the edge is condensed out and the remaining perpendicular motions converted to corner rotations. Finally one extra shape is added to give additional flexibility (see Yunus et al(117)).

$$u \; = \; \text{condensed shape function as described above} \tag{12.8.15–1}$$

$$v \; = \; \text{condensed shape function as described above} \tag{12.8.15–2}$$

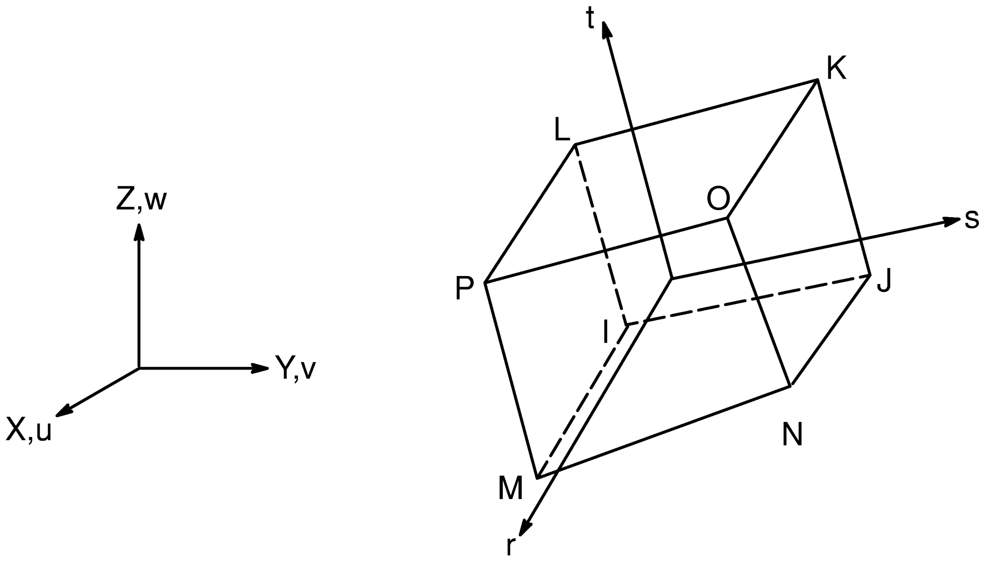
$$w \; = \; \text{condensed shape function as described above} \tag{12.8.15–3}$$

These motions are augmented by the extra shape:

$$w * \big(1 - r^2\big)$$

where: w* = motion parallel to the r axis

12.8.18 8 Node Bricks without ESF and without RDOF



These shape functions are for 8 node brick elements without extra shape functions such as SOLID5 with KEYOPT(3) = 1, FLUID30, SOLID45 with KEYOPT(1) = 1, or FLUID142:

$$\begin{aligned} u = \frac{1}{8} \big(& u_I(1-s)(1-t)(1-r) + u_J(1+s)(1-t)(1-r) \\ & + u_K(1+s)(1+t)(1-r) + u_L(1-s)(1+t)(1-r) \\ & + u_M(1-s)(1-t)(1+r) + u_N(1+s)(1-t)(1+r) \\ & + u_O(1+s)(1+t)(1+r) + u_P(1-s)(1+t)(1+r)\big) \end{aligned} \qquad (12.8.18\text{–}1)$$

$$v = \frac{1}{8} \left(v_I(1\text{–}s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.8.18\text{–}2)$$

$$w = \frac{1}{8} \left(w_I(1\text{–}s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.8.18\text{–}3)$$

$$A_x = \frac{1}{8} \left(A_{xI}(1\text{–}s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.8.18\text{–}7)$$

$$A_y = \frac{1}{8} \left(A_{yI}(1\text{–}s)\ .\ .\ .\ \text{(analogous to u)}\right. \qquad (12.8.18\text{–}8)$$

$$A_z = \frac{1}{8} \left(A_{zI}(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–9)$$

$$V_x = \frac{1}{8} \left(V_{xI}(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–10)$$

$$V_y = \frac{1}{8} \left(V_{yI}(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–11)$$

$$V_z = \frac{1}{8} \left(V_{zI}(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–12)$$

$$P = \frac{1}{8} \left(P_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–19)$$

$$T = \frac{1}{8} \left(T_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–20)$$

$$V = \frac{1}{8} \left(V_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–21)$$

$$\phi = \frac{1}{8} \left(\phi_I(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–22)$$

$$E^K = \frac{1}{8} \left(E_I^K(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–23)$$

$$E^D = \frac{1}{8} \left(E_I^D(1-s) \ . \ . \ . \ \text{(analogous to u)}\right. \qquad (12.8.18–24)$$

12.8.19 8 Node Bricks with ESF but without RDOF

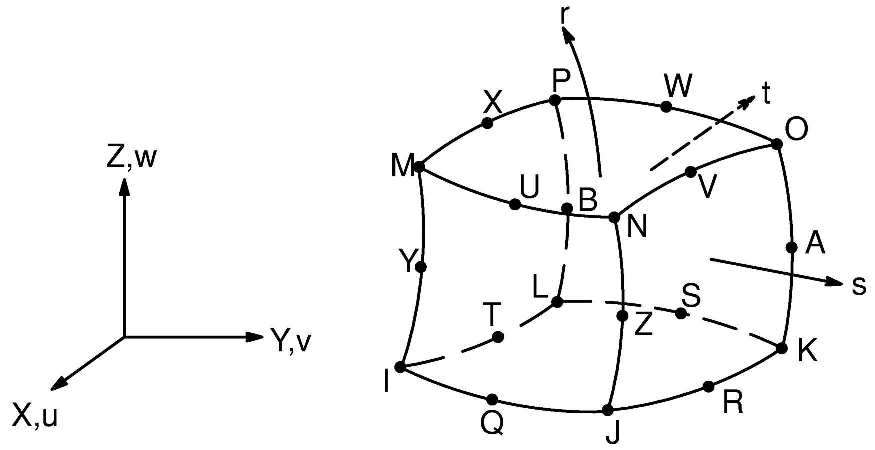
(Please see the illustration in Section 12.8.18.) These shape functions are for 8 node brick elements with extra shape functions such as SOLID5 with KEYOPT(3) = 0 or SOLID45 with KEYOPT(1) = 0:

$$\begin{aligned} u = \frac{1}{8} \big(& u_I(1 - s)(1 - t)(1 - r) + u_J(1 + s)(1 - t)(1 - r) \\ & + u_K(1 + s)(1 + t)(1 - r) + u_L(1 - s)(1 + t)(1 - r) \\ & + u_M(1 - s)(1 - t)(1 + r) + u_N(1 + s)(1 - t)(1 + r) \\ & + u_O(1 + s)(1 + t)(1 + r) + u_P(1 - s)(1 + t)(1 + r)\big) \\ & + u_1\left(1 - s^2\right) + u_2\left(1 - t^2\right) + u_3\left(1 - r^2\right) \end{aligned} \qquad (12.8.19\text{–}1)$$

$$v = \frac{1}{8} \left(v_I(1\text{–}s)\text{. . . (analogous to u)}\right) \qquad (12.8.19\text{–}2)$$

$$w = \frac{1}{8} \left(w_I(1\text{–}s)\text{. . . (analogous to u)}\right) \qquad (12.8.19\text{–}3)$$

12.8.20 20 Node Bricks without RDOF



These shape functions are used for 20 node solid elements such as SOLID90 or SOLID95.

$$
\begin{aligned}
u = \frac{1}{8}\Big(&u_I(1-s)(1-t)(1-r)(-s-t-r-2) + u_J(1+s)(1-t)(1-r)(s-t-r-2) \\
&+ u_K(1+s)(1+t)(1-r)(s+t-r-2) + u_L(1-s)(1+t)(1-r)(-s+t-r-2) \\
&+ u_M(1-s)(1-t)(1+r)(-s-t+r-2) + u_N(1+s)(1-t)(1+r)(s-t+r-2) \\
&+ u_O(1+s)(1+t)(1+r)(s+t+r-2) + u_P(1-s)(1+t)(1+r)(-s+t+r-2)\Big) \\
+ \frac{1}{4}\Big(&u_Q(1-s^2)(1-t)(1-r) + u_R(1+s)(1-t^2)(1-r) \\
&+ u_S(1-s^2)(1+t)(1-r) + u_T(1-s)(1-t^2)(1-r) \\
&+ u_U(1-s^2)(1-t)(1+r) + u_V(1+s)(1-t^2)(1+r) \\
&+ u_W(1-s^2)(1+t)(1+r) + u_X(1-s)(1-t^2)(1+r) \\
&+ u_Y(1-s)(1-t)(1-r^2) + u_Z(1+s)(1-t)(1-r^2) \\
&+ u_A(1+s)(1+t)(1-r^2) + u_B(1-s)(1+t)(1-r^2)\Big)
\end{aligned}
\tag{12.8.20–1}
$$

$$v = \frac{1}{8}\left(v_I(1-s)\ \ldots\ \text{(analogous to u)}\right. \tag{12.8.20–2}$$

$$w = \frac{1}{8}\left(w_I(1-s)\ \ldots\ \text{(analogous to u)}\right. \tag{12.8.20–3}$$

$$T = \frac{1}{8}\left(T_I(1-s)\ \ldots\ \text{(analogous to u)}\right. \tag{12.8.20–20}$$

$$V = \frac{1}{8}\left(V_I(1-s)\ \ldots\ \text{(analogous to u)}\right. \tag{12.8.20–21}$$

$$\phi = \frac{1}{8}\left(\phi_I(1-s)\ \ldots\ \text{(analogous to u)}\right. \tag{12.8.20–22}$$

12.8.21 8 Node Bricks with RDOF

Explicit shape functions are not developed for the primary motions of the 8 node brick element with RDOFs. Rather, a 20 node brick element without RDOFs (for a total of 60 DOFs) is generated first. Then the motion of the midside node parallel to the edge is condensed out and the remaining perpendicular motions converted to corner rotations. Finally, three extra shapes are added to give additional flexibility (see Yunus et al(117)).

$$u = \text{condensed shape function as described above} \tag{12.8.21–1}$$

$$v = \text{condensed shape function as described above} \tag{12.8.21–2}$$

$$w = \text{condensed shape function as described above} \tag{12.8.21–3}$$

These motions are augmented by the extra shapes:

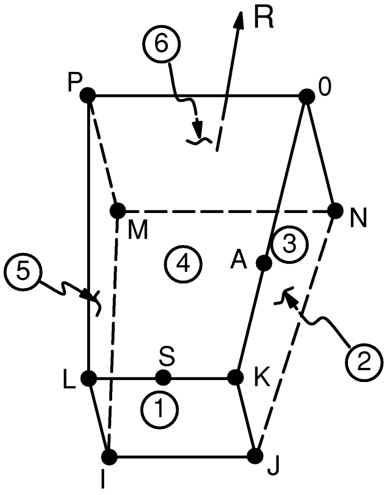
$$u^* \left(1 - s^2\right)$$

$$v^* \left(1 - t^2\right)$$

$$w^* \left(1 - r^2\right)$$

where: u* = motion parallel to the s axis
v* = motion parallel to the t axis
w* = motion parallel to the r axis

12.8.22 8 Node Infinite Bricks



These Lagrangian isoparametric shape functions and “mapping” functions are for the 3–D 8 node solid brick infinite elements such as INFIN111:

Lagrangian Isoparametric Shape Functions:

$$\begin{aligned} A_x = \frac{1}{8}\Big(& A_{xI}\,(1-s)\,(1-t)\,(r^2-r) \\ & + A_{xJ}\,(1+s)\,(1-t)\,(r^2-r) \\ & + A_{xK}\,(1+s)\,(1+t)\,(r^2-r) \\ & + A_{xL}\,(1-s)\,(1+t)\,(r^2-r)\Big) \\ + \frac{1}{4}\Big(& A_{xM}\,(1-s)\,(1-t)\,(1-r^2) \\ & + A_{xN}\,(1+s)\,(1-t)\,(1-r^2) \\ & + A_{xO}\,(1+s)\,(1+t)\,(1-r^2) \\ & + A_{xP}\,(1-s)\,(1+t)\,(1-r^2)\Big) \end{aligned} \qquad (12.8.22\text{–}7)$$

$$A_y = \frac{1}{8}\,(A_{yI}\,(1-s)\;.\;.\;.\;\text{(analogous to } A_x\text{)} \qquad (12.8.22\text{–}8)$$

$$A_z = \frac{1}{8}\,(A_{zI}\,(1-s)\;.\;.\;.\;\text{(analogous to } A_x\text{)} \qquad (12.8.22\text{–}9)$$

$$T = \frac{1}{8}\,(T_I\,(1-s)\;.\;.\;.\;\text{(analogous to } A_x\text{)} \qquad (12.8.22\text{–}20)$$

$$V = \frac{1}{8}\,(V_I\,(1-s)\;.\;.\;.\;\text{(analogous to } A_x\text{)} \qquad (12.8.22\text{–}21)$$

$$\phi = \frac{1}{8}\,(\phi_I\,(1-s)\;.\;.\;.\;\text{(analogous to } A_x\text{)} \qquad (12.8.22\text{–}22)$$

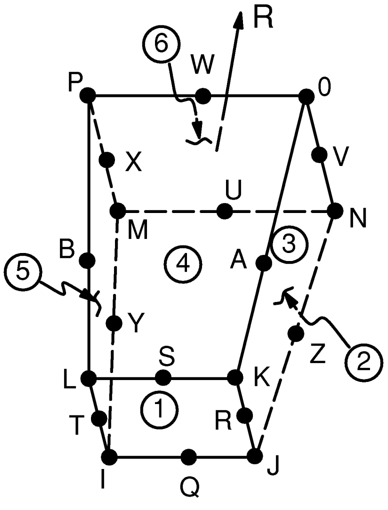
Mapping Functions:

$$
\begin{aligned}
x = \frac{1}{2} \Big(& x_I (1-s)(1-t)(-r) / (1-r) \\
& + x_J (1+s)(1-t)(-r) / (1-r) \\
& + x_K (1+s)(1+t)(-r) / (1-r) \\
& + x_L (1-s)(1+t)(-r) / (1-r) \Big) \\
+ \frac{1}{4} \Big(& x_M (1-s)(1-t)(1+r) / (1-r) \\
& + x_N (1+s)(1-t)(1+r) / (1-r) \\
& + x_O (1+s)(1+t)(1+r) / (1-r) \\
& + x_P (1-s)(1+t)(1+r) / (1-r) \Big)
\end{aligned}
\qquad (12.8.22\text{–}33)
$$

$$y = \frac{1}{2} (y_I (1-s) \; . \; . \; . \; \text{(analogous to x)} \qquad (12.8.22\text{–}34)$$

$$z = \frac{1}{2} (z_I (1-s) \; . \; . \; . \; \text{(analogous to x)} \qquad (12.8.22\text{–}35)$$

12.8.23 3–D 20 Node Infinite Bricks



These Lagrangian isoparametric shape functions and “mapping” functions are for the 3–D 20 node solid brick infinite elements such as INFIN111:

Lagrangian Isoparametric Shape Functions:

$$\begin{aligned}
A_x = {} & \frac{1}{8}\left(A_{xI}(1-s)(1-t)(1-r)(-s-t-r-2)\right) \\
& + \frac{1}{4}\left(A_{xJ}(1-s^2)(1-t)(1-r)\right) \\
& + \frac{1}{8}\left(A_{xK}(1+s)(1-t)(1-r)(s-t-r-2)\right) \\
& + \frac{1}{4}\left(A_{xL}(1+s)(1-t^2)(1-r)\right) \\
& + \frac{1}{8}\left(A_{xM}(1+s)(1+t)(1-r)(s+t-r-2)\right) \\
& + \frac{1}{4}\left(A_{xN}(1-s^2)(1+t)(1-r)\right) \\
& + \frac{1}{8}\left(A_{xO}(1-s)(1+t)(1-r)(-s+t-r-2)\right) \\
& + \frac{1}{4}\left(A_{xP}(1-s)(1-t^2)(1-r)\right) \\
& + \frac{1}{4}\left(A_{xQ}(1-s)(1-t)(1-r^2)\right) \\
& + \frac{1}{4}\left(A_{xR}(1+s)(1-t)(1-r^2)\right) \\
& + \frac{1}{4}\left(A_{xS}(1+s)(1+t)(1-r^2)\right) \\
& + \frac{1}{4}\left(A_{xT}(1-s)(1+t)(1-r^2)\right)
\end{aligned} \tag{12.8.23–7}$$

$$A_y = \frac{1}{8}\left(A_{xI}(1-s)\ .\ .\ .\ (\text{analogous to } A_x)\right) \tag{12.8.23–8}$$

$$A_z = \frac{1}{8}\left(A_{zI}(1-s)\ .\ .\ .\ (\text{analogous to } A_x)\right) \tag{12.8.23–9}$$

$$T = \frac{1}{8}\left(A_{TI}(1-s)\ .\ .\ .\ (\text{analogous to } A_x)\right) \tag{12.8.23–20}$$

$$V = \frac{1}{8}\left(A_{VI}(1-s)\ .\ .\ .\ (\text{analogous to } A_x)\right) \tag{12.8.23–21}$$

$$\phi = \frac{1}{8}\left(\phi_I(1-s)\ .\ .\ .\ (\text{analogous to } A_x)\right) \tag{12.8.23–22}$$

Mapping Functions:

$$
\begin{aligned}
x = {} & x_I (1 - s)(1 - t)(-s - t - r - 2) / (2 (1 - r)) \\
& + x_J (1 - s^2)(1 - t) / (1 - r) \\
& + x_K (1 + s)(1 - t)(s - t - r - 2) / (2 (1 - r)) \\
& + x_L (1 + s)(1 - t^2) / (1 - r) \\
& + x_M (1 + s)(1 + t)(s + t - r - 2) / (2 (1 - r)) \\
& + x_N (1 - s^2)(1 + t) / (1 - r) \\
& + x_O (1 - s)(1 + t)(- s + t - r - 2) / (2 (1 - r)) \\
& + x_P (1 - s)(1 - t^2) / (1 - r) \\
& + x_Q (1 - s)(1 - t)(1 + r) / (4 (1 - r)) \\
& + x_R (1 + s)(1 - t)(1 + r) / (4 (1 - r)) \\
& + x_S (1 + s)(1 + t)(1 + r) / (4 (1 - r)) \\
& + x_T (1 - s)(1 + t)(1 + r) / (4 (1 - r))
\end{aligned}
\qquad (12.8.23\text{–}33)
$$

$$y = y_I (1 - s) \ . \ . \ . \ \text{(analogous to x)} \qquad (12.8.23\text{–}34)$$

$$z = z_I (1 - s) \ . \ . \ . \ \text{(analogous to x)} \qquad (12.8.23\text{–}35)$$

The shape and mapping functions for the nodes U, V, W, X, Y, Z, A, and B are deliberately set to zero.

12.9 Electromagnetic Edge Elements

The shortcomings of electromagnetic analysis by nodal based continuous vector potential is discussed in section 5.1.5. These can be eliminated by edge shape functions described in this section. The edge element formulation constitutes the theoretical fundation of low–frequency electromagnetic element, 117.

Edge elements on tetrahedra and rectangular blocks have been introduced by Nedelec(204); first order and quadratic isoparametric hexahedra by van Welij(205) and Kameari(206), respectively. Difficulty with distorted hexahedral edge elements is reported by Jin(207) without appropriate resolution. Gyimesi and Ostergaard(201), (221), Ostergaard and Gyimesi(222, 223) explained the theoretical shortage of isoparametric hexahedra. Their non–conforming edge shape functions are implemented, eliminating the negative effect of element distortion. The extension of brick shapes to tetrahedra, wedge and pyramid geometries is given in Gyimesi and Ostergaard(221).

Sections 12.9.1 and 12.9.2 describe the 2D and 3D electromagnetic edge elements, respectively.

12.9.1 2D 8–Node Quad Geometry and Degrees of Freedoms (DOFs)

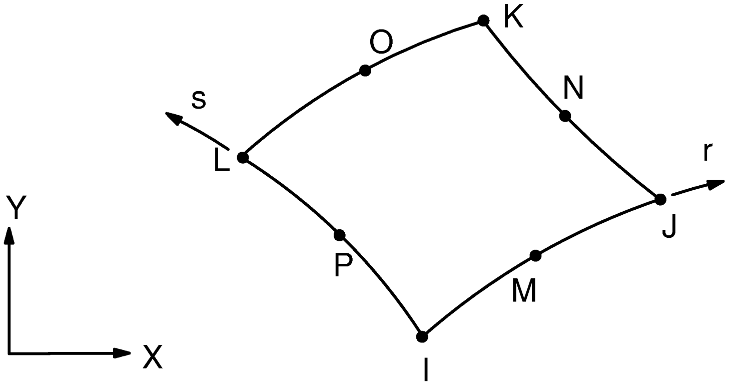


Figure 12.9–1 2D 8–Node Quad Edge Element

Figure 12.9–1 shows the geometry of 2D 8–node electromagnetic edge elements. The corner nodes, I,J,K and L are used to:

- describe the geometry

- orient the edges
- support time integrated electric potential DOFs, labelled VOLT

The side nodes, M,N,O and P are used to:

- support the edge–flux DOFs, labelled as AZ. The positive orientation of an edge is defined to point from the adjacent (to the edge) corner node with lower node number to the other adjacent node with higher node number. For example, edge, M, is oriented from node I to J if I has a smaller node number than J; otherwise it is oriented from J to I.

The edge–flux DOFs are used in both magnetostatic and dynamic analyses; the VOLT DOFs are used only for dynamic analysis.

The vector potential, A, and time integrated electric scalar potential, V, can be described as

$$A = A_M E_M + A_N E_N + A_O E_O + A_P E_P \tag{12.9–1}$$

$$V = V_I N_I + V_J N_J + V_K N_K + V_L N_L \tag{12.9–2}$$

where: A_M,A_N,A_O,A_P = edge–flux

AZ = nodal DOFs supported by the side nodes

V_I,V_J,V_K,V_L = time integrated electric scalar potential

VOLT = nodal DOFs supported by corner nodes

E_M,E_N,E_O,E_P = vector edge shape functions

N_I,N_J,N_K,N_L = scalar nodal shape functions

The following subsections describe these shape functions.

The global Cartesian coordinates, X and Y, can be expressed by the isoparametric coordinates, r and s.

$$X = N_I(r,s)\ X_I + N_J(r,s)\ X_J + N_K(r,s)\ X_K + N_L(r,s)\ X_L \tag{12.9–3}$$

$$Y = N_I(r,s)\ Y_I + N_J(r,s)\ Y_J + N_K(r,s)\ Y_K + N_L(r,s)\ Y_L \tag{12.9–4}$$

where:

X_I thru Y_L = global Cartesian coordinates of the corner nodes

N_I, N_J, N_K and N_L = first order scalar nodal shape functions

$$N_I = (1 - r)(1 - s) \tag{12.9–5}$$

$$N_J = r\,(1 - s) \tag{12.9–6}$$

$$N_K = r\,s \tag{12.9–7}$$

$$N_L = (1 - r)\,s \tag{12.9–8}$$

The isoparametric vector edge shape functions are defined as

$$E_M = +\,(1 - s)\text{ grad } r \tag{12.9–9}$$

$$E_N = r\text{ grad } s \tag{12.9–10}$$

$$E_O = -\,s\text{ grad } r \tag{12.9–11}$$

$$E_P = -\,(1 - r)\text{ grad } s \tag{12.9–12}$$

Note that the tangential component (the dot product with a unit vector pointing in the edge direction) of the vector edge shape functions disappears on all edges but one. The one on which the tangential component of an edge shape function is not zero is called a supporting edge which is associated with the pertinent side node.

Note also that the line integral of an edge shape function along the supporting edge is unity. The flux crossing a face is the closed line integral of the vector potential, A. Thus, the sum of the DOFs supported by side nodes around a face is the flux crossing the face. Therefore, these DOFs are called edge–flux DOFs.

12.9.2 3D 20–Node Brick Geometry and Degrees of Freedoms (DOFs)

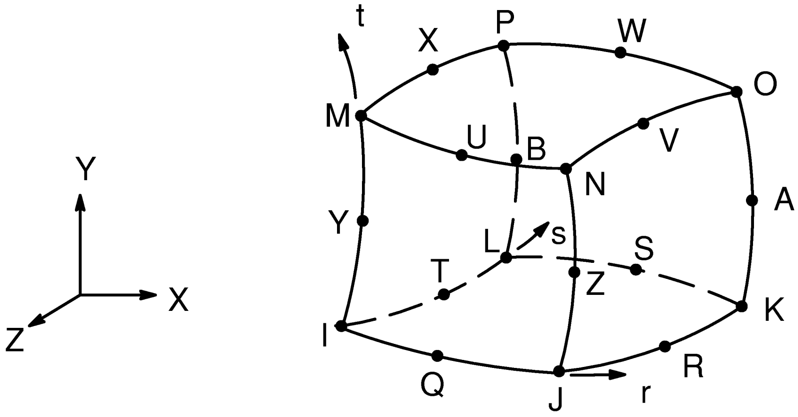


Figure 12.9–2 3D 20–Node Brick Edge Element

Figure 12.9–2 shows the geometry of 3D 20–node electromagnetic edge elements. The corner nodes, I ... P are used to:

- describe the geometry
- orient the edges
- support time integrated electric potential DOFs (labelled VOLT)

The side nodes, Q. . .A are used to:

- support the edge–flux DOFs, labelled as AZ
- define the positive orientation of an edge to point from the adjacent (to the edge) corner node with lower node number to the other adjacent node with higher node number. For example, edge, M, is oriented from node I to J if I has a smaller node number than J; otherwise it is oriented from J to I.

The edge–flux DOFs are used in both magnetostatic and dynamic analyses; the VOLT DOFs are used only for dynamic analysis.

The vector potential, A, and time integrated electric scalar potential, V, can be described as

$$A = A_Q E_Q + \cdots + A_B E_B \tag{12.9–13}$$

$$V = V_I N_I + \cdots + V_P N_P \tag{12.9–14}$$

where: $A_Q \ldots A_B$ = edge–flux

AZ = nodal DOFs supported by the side nodes
$V_I \ldots V_P$ = time integrated electric scalar potential
VOLT = nodal DOFs supported by corner nodes
$E_Q \ldots E_B$ = vector edge shape functions
$N_I \ldots N_P$ = scalar nodal shape functions

Do not confuse edge–flux DOF label, AZ, with the actual value of the DOF at node Z, A_Z.

The following subsections describe these shape functions.

The global Cartesian coordinates, X,Y and Z, can be expressed by the master coordinates, r,s and t.

$$X = N_I\ (r,s,t)\ X_I + \cdots + N_P\ (r,s,t)\ X_P \quad (12.9\text{–}15)$$

$$Y = N_I\ (r,s,t)\ Y_I + \cdots + N_P\ (r,s,t)\ Y_P \quad (12.9\text{–}16)$$

$$Z = N_I\ (r,s,t)\ Z_I + \cdots + N_P\ (r,s,t)\ Z_P \quad (12.9\text{–}17)$$

where; X_I, Y_I, $Z_{I\ \ldots}$ = global Cartesian coordinates of the corner nodes
$N_I \ldots N_p$ = first order scalar nodal shape functions

$$N_I = (1 - r)(1 - s)(1 - t) \quad (12.9\text{–}18)$$

$$N_J = r\ (1 - s)(1 - t) \quad (12.9\text{–}19)$$

$$N_K = r\ s\ (1 - t) \quad (12.9\text{–}20)$$

$$N_L = (1 - r)\ s\ (1 - t) \quad (12.9\text{–}21)$$

$$N_M = (1 - r)(1 - s)\ t \quad (12.9\text{–}22)$$

$$N_N = r\ (1 - s)\ t \quad (12.9\text{–}23)$$

$$N_O = r\ s\ t \quad (12.9\text{–}24)$$

$$N_P = (1 - r)\ s\ t \quad (12.9\text{–}25)$$

The isoparametric vector edge shape functions are defined as

$$E_Q = +(1-s)(1-t)\ \text{grad } r \tag{12.9–26}$$

$$E_R = +r\ (1-t)\ \text{grad } s \tag{12.9–27}$$

$$E_S = -s\ (1-t)\ \text{grad } r \tag{12.9–28}$$

$$E_T = -(1-r)(1-t)\ \text{grad } s \tag{12.9–29}$$

$$E_U = +(1-s)\ t\ \text{grad } r \tag{12.9–30}$$

$$E_V = +r\ t\ \text{grad } s \tag{12.9–31}$$

$$E_W = -s\ t\ \text{grad } r \tag{12.9–32}$$

$$E_X = -(1-r)\ t\ \text{grad } s \tag{12.9–33}$$

$$E_Y = +(1-s)(1-r)\ \ \text{grad } t \tag{12.9–34}$$

$$E_Z = +s\ (1-r)\ \ \text{grad } t \tag{12.9–35}$$

$$E_A = +s\ t\ \text{grad } t \tag{12.9–36}$$

$$E_B = +(1-s)\ r\ \text{grad } t \tag{12.9–37}$$

Note that the tangential component (the dot product with a unit vector pointing in the edge direction) of the vector edge shape functions disappears on all edges but one. The one on which the tangential component of an edge shape function is not zero is called a supporting edge which is accociated with the pertinent side node.

Note also that the line integral of an edge shape function along the supporting edge is unity. The flux crossing a face is the closed line integral of the vector potential, A. Thus, the sum of the DOFs supported by side nodes around a face is the flux crossing the face. Therefore, these DOFs are called edge–flux DOFs.

The 20 node brick geometry is allowed to degenerate to 10–node tetrahedron, 13–node pyramid or 15–node wedge shapes as described in Gyimesi and Ostergaard(221). The numerical bench–working shows that tetrahedra shapes are advantageous in air (no current) domains, whereas hexahedra are recommended for current carrying regions. Pyramids are applied to maintain efficient meshing between hexahedra and tetrahedra regions. Wedges are generally applied for 2–D like geometries, when longitudinal dimensions are longer than transverse sizes. In this case the cross–section can be meshed by area meshing and wedges are generated by extrusion.

Chapter 13
Element Tools

ANSYS Theory Reference
Chapter 13 – Table of Contents

13.1 Integration Point Locations

The ANSYS program makes use of both standard and nonstandard numerical integration formulas. The particular integration scheme used for each matrix or load vector is given with each element description in Chapter 14. Both standard and nonstandard integration formulas are described in this section. The numbers after the subsection titles are labels used to identify the integration point rule. For example, line (1, 2, or 3 points) represents the 1, 2, and 3 point integration schemes along line elements. Midside nodes, if applicable, are not shown in the figures in this section.

13.1.1 Lines (1, 2 or 3 Points)

The standard 1–D numerical integration formulas which are used in the element library are of the form:

$$\int_{-1}^{1} f(x)\ dx = \sum_{i=1}^{\ell} H_i\ f(x_i) \qquad (13.1\text{–}1)$$

where:

$f(x)$ = function to be integrated

H_i = weighting factor (see Table 13.1–1)

x_i = locations to evaluate function (see Table 13.1–1; these locations are usually the s, t, or r coordinates)

ℓ = number of integration (Gauss) points

Table 13.1–1 Gauss Numerical Integration Constants

Number of Integration Points	Integration Point Locations (x_i)	Weighting Factors (H_i)
1	0.00000 00000 00000	2.00000 00000 00000
2	± 0.57735 02691 89626	1.00000 00000 00000
3	± 0.77459 66692 41483 0.00000 00000 00000	0.55555 55555 55556 0.88888 88888 88889

For some integrations of multi–dimensional regions, the method of equation (13.1–1) is simply expanded, as shown below.

13.1.2 Quadrilaterals (2 x 2 or 3 x 3 Points)

The numerical integration of 2–D quadrilaterals gives:

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\ dxdy \ = \sum_{j=1}^{m}\sum_{i=1}^{\ell} H_j H_i\ f(x_i,\ y_j) \tag{13.1–2}$$

and the integration point locations are shown in Figure 13.1–1.

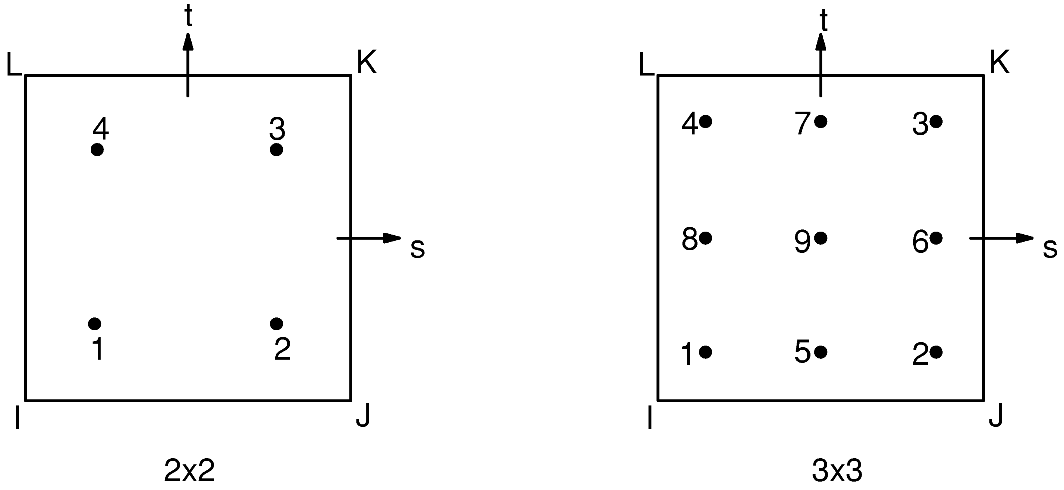


Figure 13.1–1 Integration Point Locations for Quadrilaterals

One element models with midside nodes (e.g. SOLID82) using a 2 x 2 mesh of integration points have been seen to generate spurious zero energy (hourglassing) modes.

13.1.3 Bricks and Pyramids (2 x 2 x 2 Points)

The 3–D integration of bricks and pyramids gives:

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(x,y,z)\ dxdydz \ = \sum_{k=1}^{n}\sum_{j=1}^{m}\sum_{i=1}^{\ell} H_k H_j H_i\ f(x_i,\ y_j,\ z_k) \tag{13.1–3}$$

and the integration point locations are shown in Figure 13.1–2.

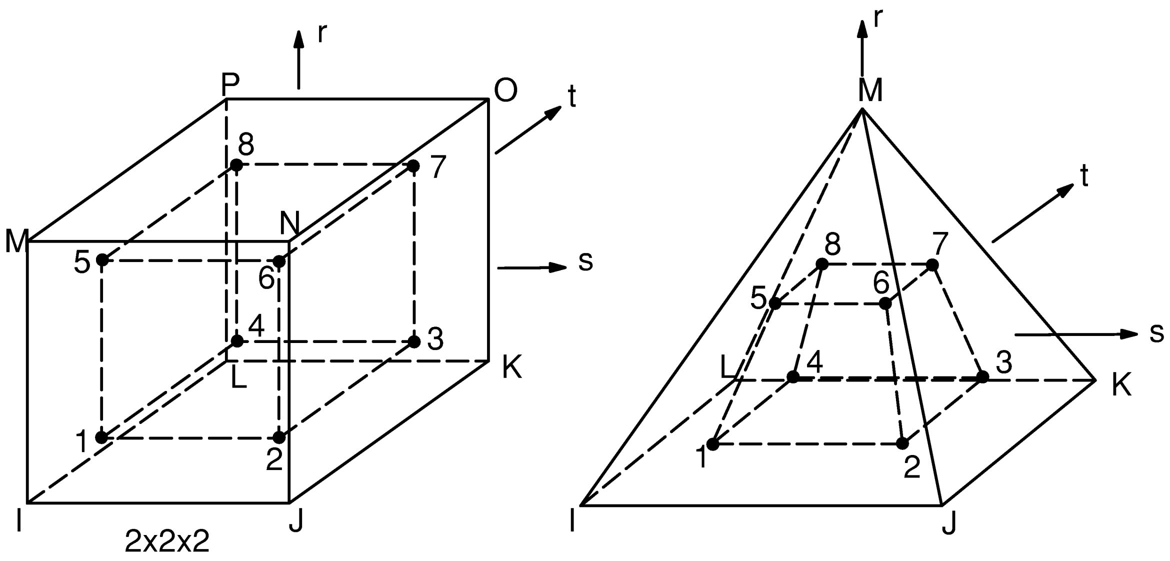


Figure 13.1–2 Integration Point Locations for Bricks and Pyramids

One element models with midside nodes using a 2 x 2 x 2 mesh of integration points have been seen to generate spurious zero energy (hourglassing) modes.

13.1.4 Triangles (1, 3, or 6 Points)

The integration points used for these triangles are given in Table 13.1–2 and appear as shown in Figure 13.1–3. L varies from 0.0 at an edge to 1.0 at the opposite vertex.

Table 13.1–2 Numerical Integration for Triangles

Type		Integration Point Location	Weighting Factor
1 Point Rule		$L_1 = L_2 = L_3 = .3333333$	1.000000
3 Point Rule		$L_1 = .66666\ 66666\ 66666$ $L_2 = L_3 = .16666\ 66666\ 66666$ (Permute L_1, L_2, and L_3 for other locations)	0.33333 33333 33333
6 Point Rule	Corner Points	$L_1 = 0.81684\ 75729\ 80459$ $L_2 = L_3 = 0.09157\ 62135\ 09771$ (Permute L_1, L_2, and L_3 for other locations)	0.10995 17436 55322
	Edge Center Points	$L_1 = 0.10810\ 30181\ 6807$ $L_2 = L_3 = 0.44594\ 84909\ 15965$ (Permute L_1, L_2, and L_3 for other locations)	0.22338 15896 78011

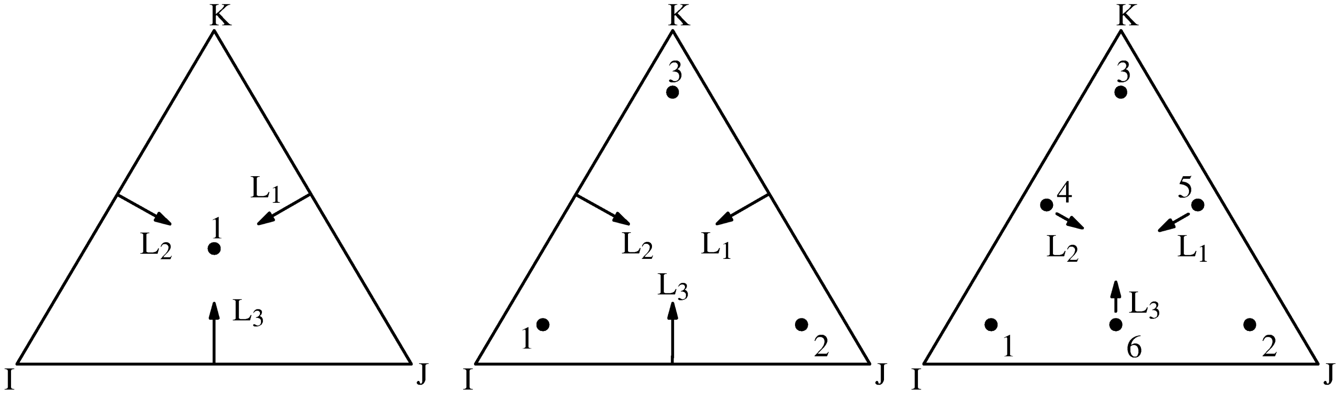


Figure 13.1–3 Integration Point Locations for Triangles

13.1.5 Tetrahedra (1, 4, or 5 Points)

The integration points used for tetrahedra are given in Table 13.1–3.

Table 13.1–3 Numerical Integration for Tetrahedra

Type	Integration Point Location		Weighting Factor
1 Point Rule	Center Point	$L_1 = L_2 = L_3 = L_4 =$ .25000 00000 00000	1.00000 00000 00000
4 Point Rule	Corner Points	$L_1 =$ .58541 01966 24968 $L_2 = L_3 = L_4 =$ .13819 66011 25010 (Permute L_1, L_2, L_3, and L_4 for other locations)	0.25000 00000 00000
5 Point Rule	Center Point	$L_1 = L_2 = L_3 = L_4 =$ .25000 00000 00000	–0.80000 00000 00000
	Corner Points	$L_1 =$ .50000 00000 00000 $L_2 = L_3 = L_4 =$ .16666 66666 66666 (Permute L_1, L_2, L_3, and L_4 for other three locations)	0.45000 00000 00000
11 Point Rule	Center Point	$L_1 = L_2 = L_3 = L_4 =$ .25000 00000 00000	0.01315 55555 55555
	Corner Points	$L_1 = L_2 = L_3 =$ .0714285714285714 $L_4 =$ .785714285714286 (Permute L_1, L_2, L_3, and L_4 for other three locations)	0.00762 22222 22222
	Edge Center Points	$L_1 = L_2 =$ 0.39940 35761 66799 $L_3 = L_4 =$ 0.10059 64238 33201 Permute L_1, L_2, L_3 and L_4 such that two of L_1, L_2, L_3 and L_4 equal 0.39940 35761 66799 and the other two equal 0.10059 64238 33201 for other five locations	0.02488 88888 88888

These appear as shown in Figure 13.1–4. L varies from 0.0 at a face to 1.0 at the opposite vertex.

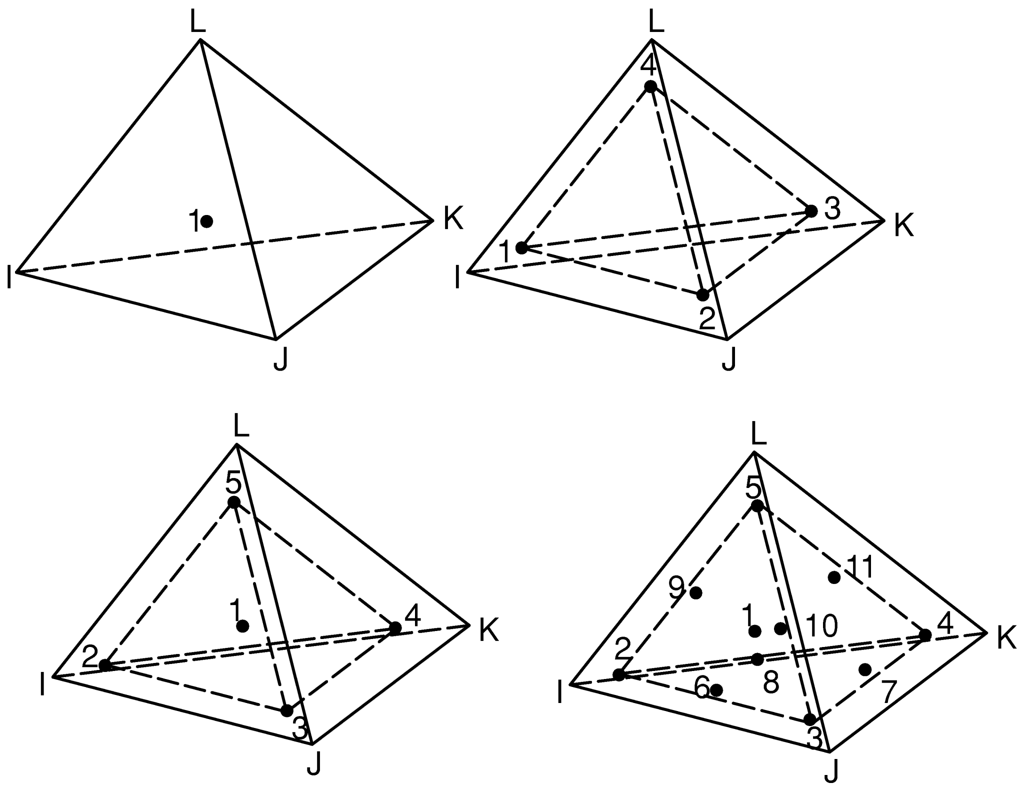


Figure 13.1–4   Integration Point Locations for Tetrahedra

13.1.6   Triangles and Tetrahedra (2 x 2 or 2 x 2 x 2 Points)

These elements use the same integration point scheme as for 4 node quadrilaterals and 8 node solids, as shown in Figure 13.1–5:

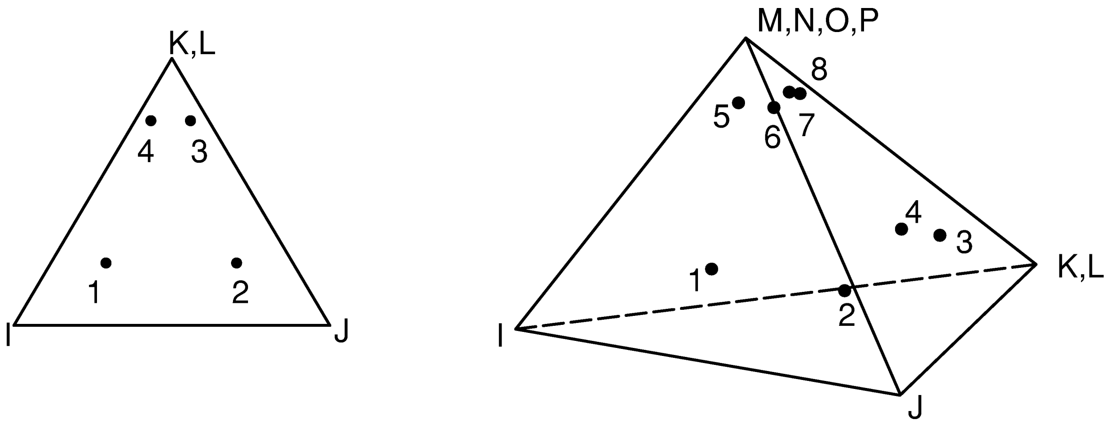
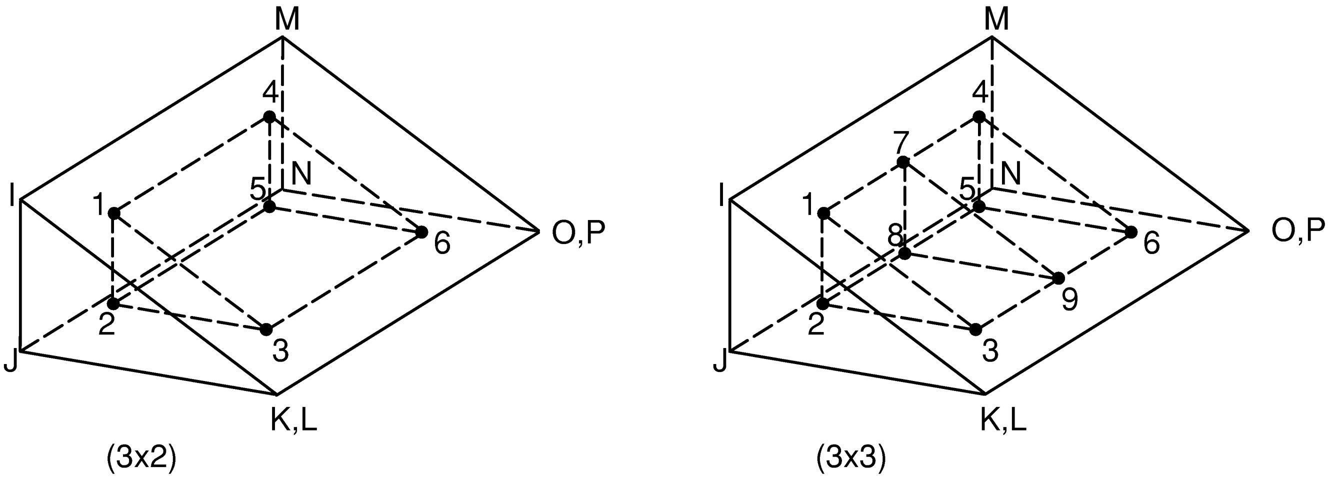


Figure 13.1–5   Integration Point Locations for Triangles and Tetrahedra

3x3 and 3x3x3 cases are handled similarly.

13.1.7 Wedges (3 x 2 or 3 x 3 Points)

These wedge elements use an integration scheme that combines linear and triangular integrations, as shown in Figure 13.1–6:



(3x2)

(3x3)

Figure 13.1–6 Integration Point Locations for Wedges

13.1.8 Wedges (2 x 2 x 2 Points)

These wedge elements use the same integration point scheme as for 8 node solid elements as shown by two orthogonal views in Figure 13.1–7:

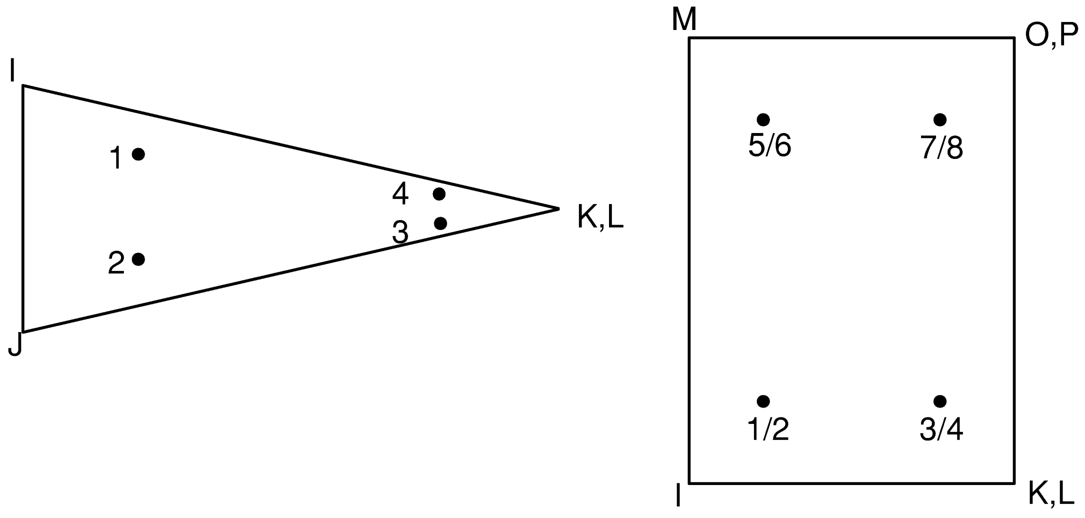


Figure 13.1–7 Integration Point Locations for Wedges

13.1.9 Bricks (14 Points)

The 20 node solid uses a different type of integration point scheme. This scheme places points close to each of the 8 corner nodes and close to the centers of the 6 faces for a total of 14 points. These locations are given in Table 13.1–4:

Table 13.1–4 Numerical Integration for 20 Node Brick

Type	Integration Point Location		Weighting Factor

14 Point Rule	Corner Points	s = ± .75868 69106 39328
		t = ± .75878 69106 39328
		r = ± .75878 69106 39328
			.33518 00554 01662

	Center Points	s = ± .79582 24257 54222, t = r = 0.0
		t = ± .79582 24257 54222, s = r = 0.0
		r = ± .79582 24257 54222, s = t = 0.0
			.88642 65927 97784

and are shown in Figure 13.1–8.

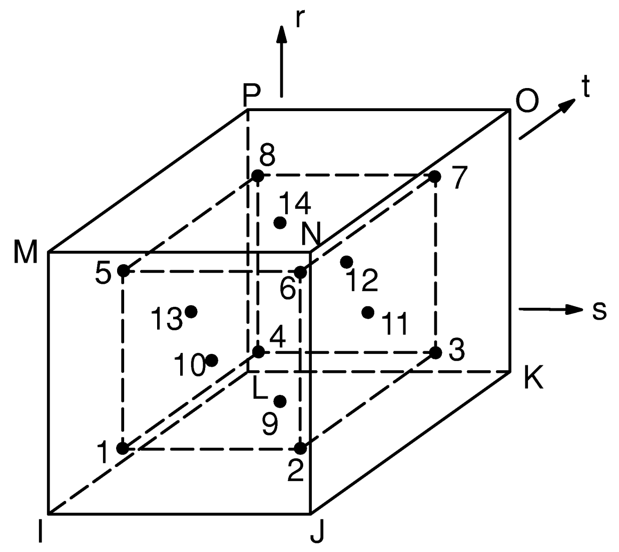


Figure 13.1–8 Integration Point Locations for 14 Point Rule

13.1.10 Nonlinear Bending (5 or 9 Points)

Both beam and shell elements that have nonlinear materials must have their effects accumulated thru the thickness. Both a 5 point and a 9 point scheme are used. These are nonstandard integration point locations, as both the top and bottom surfaces have an integration point in order to immediately detect the onset of the nonlinear effects.

Table 13.1–5 Thru–thickness Numerical Integration

Type	Integration Point Location*	Weighting Factor
5	± 0.500	0.1250000
	± 0.300	0.5787036
	0.000	0.5925926
9	± 0.500	0.0177507
	± 0.450	0.1906872
	± 0.375	0.0959028
	± 0.250	0.4287226
	0.000	0.5338734

*Thickness coordinate going from –0.5 to 0.5.

These locations are shown in Figure 13.1–9.

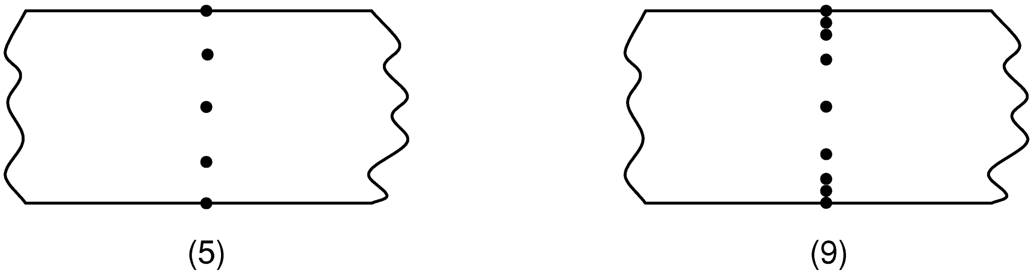


Figure 13.1–9 Nonlinear Bending Integration Point Locations

13.2 Lumped Matrices

Some of the elements have their consistent mass or specific heat matrices reduced to diagonal matrices. This is referred to as “lumping” and is accessed with the **LUMPM** command.

13.2.1 Diagonalization Procedure

One of two procedures is used for the diagonalization, depending on the order of the element shape functions. The mass matrix is used as an example.

For lower order elements (linear or bilinear) the diagonalized matrix is computed by summing rows (or columns). The steps are:

1. Compute the consistent mass matrix $\left(\left[M_e'\right]\right)$ in the usual manner.
2. Compute:

$$S(i) = \sum_{j=1}^{n} M_e'(i,j) \qquad \text{for } i = 1, n \tag{13.2–1}$$

where: n = number of degrees of freedom (DOFs) in the element

3. Set

$$M_e(i,j) = 0.0 \qquad \text{for } i \neq j \tag{13.2–2}$$

$$M_e(i,i) = S(i) \qquad \text{for } i = 1, n \tag{13.2–3}$$

For higher order elements the procedure suggested by Hinton, et al.(45), is used. The steps are:

1. Compute the consistent mass matrix $\left(\left[M_e'\right]\right)$ in the usual manner.
2. Compute:

$$S = \sum_{i=1}^{n} \sum_{j=1}^{n} M_e'(i,j) \tag{13.2–4}$$

$$D = \sum_{i=1}^{n} M_e'(i,i) \qquad (13.2–5)$$

3. Set:

$$M_e(i,j) = 0.0 \qquad \text{if } i \neq j \qquad (13.2–6)$$

$$M_e(i,i) = \frac{S}{D} M_e'(i,i) \qquad (13.2–7)$$

Note that this method ensures that:

1. The element mass is preserved
2. The element mass matrix is positive definite

It may be observed that if the diagonalization is performed by simply summing rows or columns in higher order elements, the resulting element mass matrix is not always positive definite.

13.2.2 Limitations of Lumped Mass Matrices

Lumped mass matrices, accessed with the **LUMP,ON** command, have the following limitations:

1. Elements containing both translational and rotational degrees of freedom will have mass contributions only for the translational degrees of freedom.
2. Lumping, by its very nature, eliminates the concept of mass coupling between degrees of freedom. Therefore, the following restrictions exist:
   a. Lumping is not allowed for FLUID29, FLUID30, or FLUID38 elements.
   b. Lumping is not allowed for BEAM44 elements when using member releases in the element UY or UZ directions.
   c. Lumping is not allowed for PIPE59 elements when using ‘added mass’ on the outside of the pipe. In this case, the implied coupling exists when the element x–axis is not parallel to one of the three nodal axes.
   d. A warning message will be output if BEAM23, BEAM24, BEAM44, or BEAM54 elements are used with explicit or implied offsets.
   e. The effect of the implied offsets is ignored by the lumping logic when used with warped SHELL63 elements.
   f. Lumping is not allowed for the mass matrix option of MATRIX27 elements if it is defined with non–zero off–diagonal terms.

g. The use of lumping with constraint equations may effectively cause the loss of some mass for dynamic analyses, resulting in higher frequencies. This loss comes about because of the generation of off–diagonal terms by the constraint equations, which then are ignored. (The exception to this loss is PowerDynamics, which uses lumped mass matrices without using the **LUMPM,ON** command, and loses no mass.)

13.3 Reuse of Matrices

Matrices are reused automatically as often as possible in order to decrease running time. The information below is made available for use in running time estimates.

13.3.1 Element Matrices

For static (**ANTYPE**,STATIC) or full transient dynamic (**ANTYPE**,TRANS with **TRNOPT**,FULL) analyses, element stiffness/conductivity, mass, and damping/specific heat, matrices ([K_e], [M_e], [C_e]) are always reused from iteration to iteration, except when:

1. The full Newton–Raphson option (**NROPT**,FULL) is used, or for the first equilibrium iteration of a time step when the modified Newton–Raphson option (**NROPT**,MODI) is used and the element has either nonlinear materials or large deformation (**NLGEOM**,ON) is active.
2. The element is nonlinear (e.g. gap, radiation, or control element) and its status changes.
3. MODE or ISYM (**MODE** command) have changed from the previous load step for elements PLANE25, SHELL61, PLANE75, PLANE78, FLUID81, or PLANE83.
4. $\left[K_e^t\right]$ will be reformulated if a convective film coefficient (input on the **SF** or **SFE** commands) on an element face changes. Such a change could occur as a ramp (**KBC**,0) within a load step.
5. The materials or real constants are changed by new input, or if the material properties have changed due to temperature changes for temperature–dependent input.

Element stress stiffness matrices [S_e] are never reused, as the stress normally varies from iteration to iteration.

13.3.2 Structure Matrices

The overall structure matrices are reused from iteration to iteration except when:

1. An included element matrix is reformed (see above).
2. The set of specified degrees of freedom (DOFs) is changed.
3. The integration time step size changes from that used in the previous substep for the transient (**ANTYPE**,TRANS) analysis.

4. The stress stiffening option (**SSTIF**,ON) has been activated.
5. Spin softening (input quantity *KSPIN*, **OMEGA** Command) is active.

and/or

6. The first iteration of a restart is being performed.

13.3.3 Override Option

The above tests are all performed automatically by the program. If the user wants to override the program's decision with respect to whether the matrices should be reformed or not, the **KUSE** command may be used. For example, if the user has temperature dependent input as the only cause which is forcing the reformulation of the matrices, and there is a load step where the temperature dependency is not significant, the matrices will not be reformed if the command **KUSE**,1 is input at that load step (followed by **KUSE**,0 in the following load step). On the other hand, if the user chooses **KUSE**,–1, all element matrices are reformed each iteration.

13.4 Temperature–Dependent Material Properties

Temperature–dependent material properties are evaluated once per element. Thus, the material properties are assumed to be constant over the volume of the element. The temperature used is:

$$T_c = \{N_o\}^T\{T\} \tag{13.4–1}$$

where:

T_c = the temperature at which the material property evaluation is done

$\{N_o\}$ = shape functions evaluated at the origin (centroid) of the element

$\{T\}$ = nodal temperatures

Exceptions:

1. Composite elements. Each material property is evaluated once per layer.
2. Heat transfer analysis using C_p (specific heat). As there may be step changes in C_p due to changes of state, it is evaluated at every integration point based on the applicable nodal temperatures and the shape functions evaluated at that integration point.

Whether shape functions are used or not, materials are evaluated at the temperature given, i.e. no account is made of the **TOFFST** command.

For a stress analysis, the temperatures used are based directly on the input. As temperature is the unknown in a heat transfer analysis, the material property evaluation cannot be handled in the same direct manner. For the first iteration of a heat transfer analysis, the material properties are evaluated at the uniform temperature (input on **BFUNIF** command). The properties of the second iteration are based on the temperatures of the first iteration. The properties of the third iteration are based on the temperatures of the second iteration, etc.

See Section 2.1.3 for a special discussion about the coefficient of thermal expansion.

13.5 Positive Definite Matrices

By definition, a matrix [D] (as well as its inverse $[D]^{-1}$) is positive definite if the determinants of all submatrices of the series:

$$\begin{bmatrix} D_{1,1} \end{bmatrix}, \quad \begin{bmatrix} D_{1,1} & D_{1,2} \\ D_{2,1} & D_{2,2} \end{bmatrix}, \quad \begin{bmatrix} D_{1,1} & D_{1,2} & D_{1,3} \\ D_{2,1} & D_{2,2} & D_{2,3} \\ D_{3,1} & D_{3,2} & D_{3,3} \end{bmatrix}, \quad \text{etc.}$$

including the determinant of the full matrix [D], are positive. The series could have started out at any other diagonal term and then had row and column sets added in any order. Thus, two necessary (but not sufficient) conditions for a symmetric matrix to be positive definite are given here for convenience:

$$D_{i,i} > 0.0 \tag{13.5–1}$$

$$D_{i,j} < \sqrt{D_{i,i}\ D_{j,j}} \tag{13.5–2}$$

If any of the above determinants are zero (and the rest positive), the matrix is said to be positive semidefinite. If all of the above determinants are negative, the matrix is said to be negative definite.

13.5.1 Matrices Representing the Complete Structure

In virtually all circumstances, matrices representing the complete structure with the appropriate boundary conditions must be positive definite. If they are not, the message “NEGATIVE PIVOT . . .” appears. This usually means that insufficient boundary conditions were specified. An exception is a piezoelectric analysis, which works with negative definite matrices, but does not generate any error messages.

13.5.2 Element Matrices

Element matrices are often positive semidefinite, but sometimes they are either negative or positive definite. For most cases where a negative definite matrix could inappropriately be created, the program will abort with a descriptive message.

13.6 Nodal and Centroidal Data Evaluation

Area and volume elements normally compute results most accurately at the integration points. The location of these data, which includes structural stresses, elastic and thermal strains, field gradients, and fluxes, can then be moved to nodal or centroidal locations for further study. This is done with extrapolation or interpolation, based on the element shape functions or simplified shape functions given in Table 13.6–1.

Table 13.6–1 Assumed Data Variation of Stresses

Geometry	Number of Integration Points	Assumed Data Variation
Triangles	3	a + bs + ct
Quadrilaterals	4	a + bs + ct + dst
Tetrahedra	4	a + bs + ct + dr
Hexahedra	8	a + bs + ct + dr + est + ftr + gsr + hstr

where:a,b,c,d,e,f,g,h = coefficients

s,t,r = element natural coordinates

The extrapolation is done or the integration point results are simply moved to the nodes, based on the usage of the **ERESX** command. If material nonlinearities exist in an element, the least squares fit can cause inaccuracies in the extrapolated nodal data or interpolated centroidal data. These inaccuracies are normally minor for plasticity, creep, or swelling, but are more pronounced in elements where an integration point may change status, such as SHELL41, SOLID65, etc.

There are a number of adjustments and special cases:

1. SOLID90 and SOLID95 use only the eight corner integration points.
2. SHELL63 uses a least square fitting procedure for the bending stresses. Data from all three integration points of each of the four triangles is used.
3. SHELL43, SOLID46, SHELL91, SHELL93, and SHELL99 use the procedure for quadrilaterals repeatedly at various levels through the element thickness.
4. Uniform stress cases, like a constant stress triangle, do not require the above processing.

13.7 Element Shape Testing

13.7.1 Overview

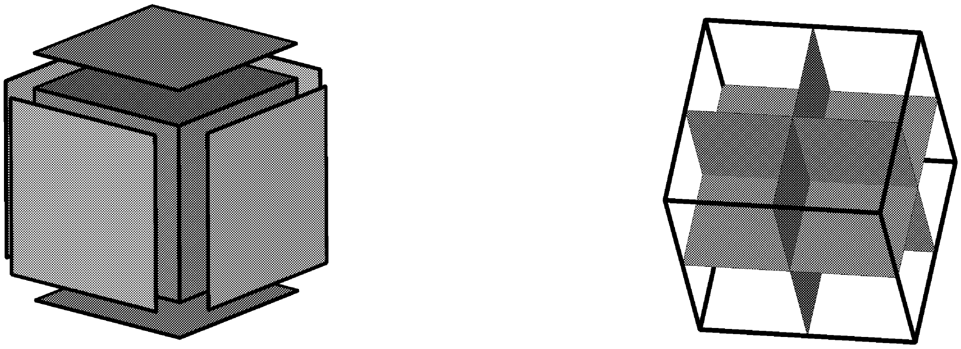
All continuum elements (2–D and 3–D solids, 3–D shells) are tested for acceptable shape as they are defined by the **E, EGEN**, **AMESH, VMESH,** or similar commands. This testing, described in the following sections, is performed by computing shape parameters (such as Jacobian ratio) which are functions of geometry, then comparing them to element shape limits whose default values are functions of element type and settings (but can be modified by the user on the **SHPP** command with *Lab*=MODIFY as described below). Nothing may be said about an element, one or more warnings may be issued, or it may be rejected with an error.

13.7.2 3–D Solid Element Faces and Cross–Sections

Some shape testing of 3–D solid elements (bricks [hexahedra], wedges, pyramids, and tetrahedra) is performed indirectly. Aspect ratio, parallel deviation, and maximum corner angle are computed for 3–D solid elements using the following steps:

1. Each of these 3 quantities is computed, as applicable, for each face of the element as though it were a quadrilateral or triangle in 3–D space, by the methods described in sections 13.7.3, 13.7.5, and 13.7.6.
2. Because some types of 3–D solid element distortion are not revealed by examination of the faces, cross–sections through the solid are constructed. Then, each of the 3 quantities is computed, as applicable, for each cross–section as though it were a quadrilateral or triangle in 3–D space.
3. The metric for the element is assigned as the worst value computed for any face or cross–section.

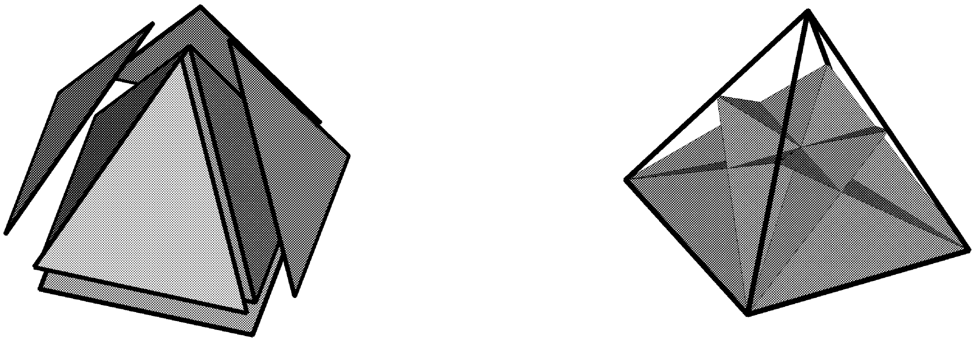
A brick element has 6 quadrilateral faces and 3 quadrilateral cross–sections (Figure 13.7–1). The cross–sections are connected to midside nodes, or to edge midpoints where midside nodes are not defined.



Element Faces Element Cross–Sections

Figure 13.7–1 Brick Element

A pyramid element has 1 quadrilateral face and 4 triangle faces, and 8 triangle cross–sections (Figure 13.7–2).



Element Faces Element Cross–Sections

Figure 13.7–2 Pyramid Element

As shown in Figure 13.7–3, each pyramid cross–section is constructed by passing a plane through one of the base edges perpendicular to one of the opposite edges. (Midside nodes, if any, are ignored.)

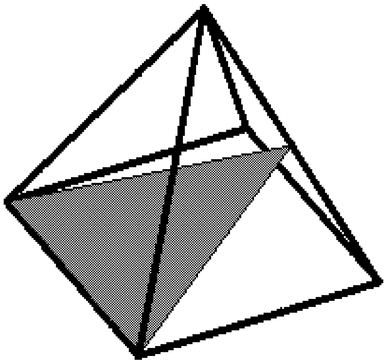


Figure 13.7–3 Pyramid Element Cross–Section Construction

A wedge element has 3 quadrilateral and 2 triangle faces, and has 3 quadrilateral and 1 triangle cross–sections. As shown in Figure 13.7–4, the cross–sections are connected to midside nodes, or to edge midpoints where midside nodes are not defined.

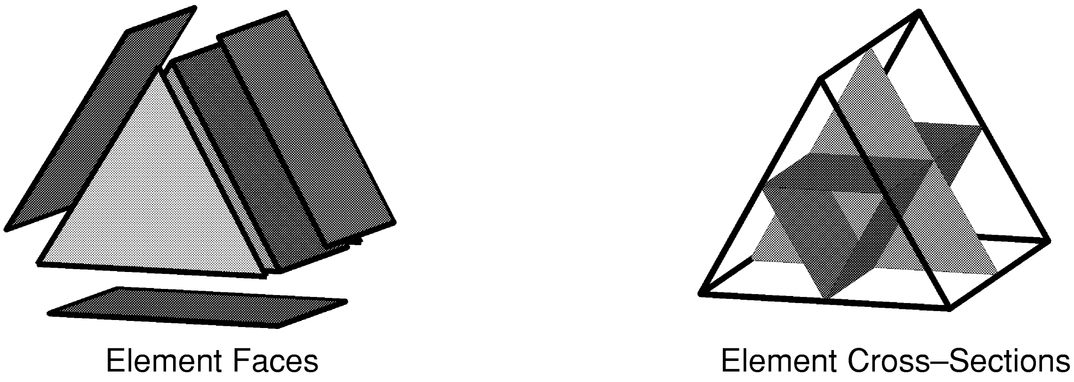
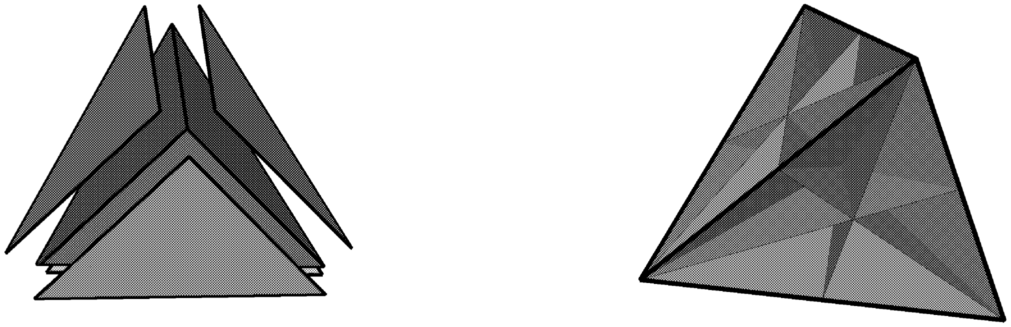


Figure 13.7–4 Wedge Element

A tetrahedron element has 4 triangle faces and 6 triangle cross–sections (Figure 13.7–5).



Element Faces

Element Cross–Sections

Figure 13.7–5 Tetrahedron Element

As shown in Figure 13.7–6, each tetrahedron cross–section is constructed by passing a plane through one of the edges perpendicular to the opposite edge. (Midside nodes, if any, are ignored.)

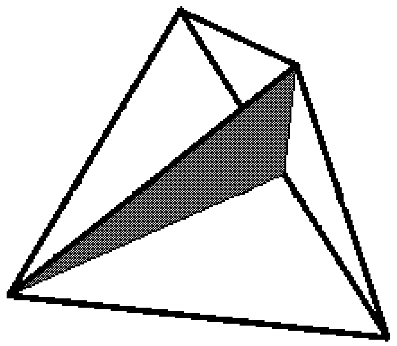


Figure 13.7–6 Tetrahedron Element Cross–Section Construction

13.7.3 Aspect Ratio

Aspect ratio is computed and tested for all except Emag or FLOTRAN elements (see Table 13.7–1). This shape measure has been reported in finite element literature for decades (Robinson(121)), and is one of the easiest ones to understand. Some analysts want to be warned about high aspect ratio so they can verify that the creation of any stretched elements was intentional. Many other analysts routinely ignore it.

Unless elements are so stretched that numeric roundoff could become a factor (aspect ratio > 1000), aspect ratio alone has little correlation with analysis accuracy. Finite element meshes should be tailored to the physics of the given problem; i.e., fine in the direction of rapidly changing field gradients, relatively coarse in directions with less rapidly changing fields. Sometimes this calls for elements having aspect ratios of 10, 100, or in extreme cases 1000. (Examples include shell or thin coating analyses using solid elements, thermal shock “skin” stress analyses, and fluid boundary layer analyses.) Attempts to artificially restrict aspect ratio could compromise analysis quality in some cases.

Aspect Ratio Calculation for Triangles

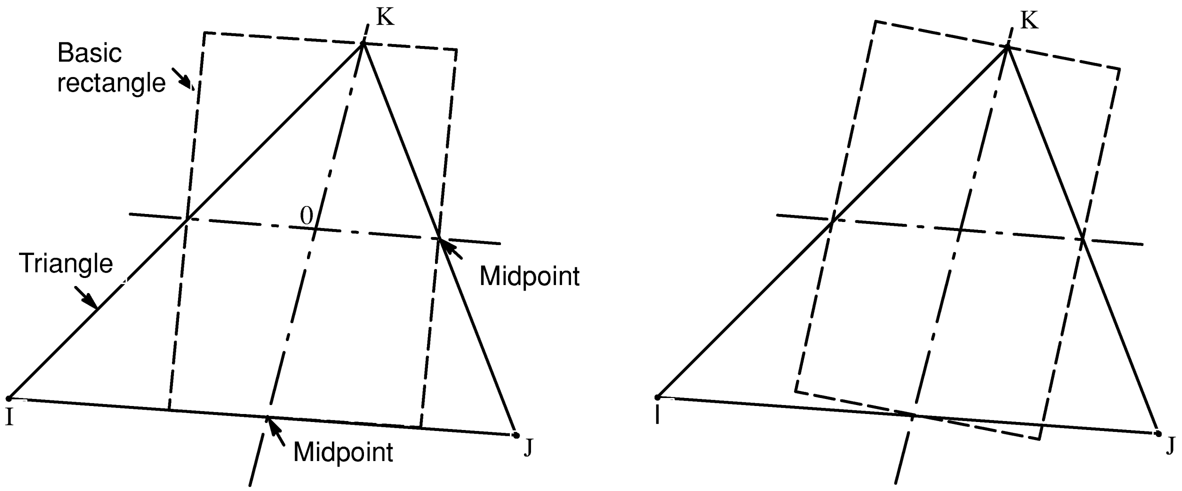


Figure 13.7–7 Triangle Aspect Ratio Calculation

The aspect ratio for a triangle is computed in the following manner, using only the corner nodes of the element (Figure 13.7–7):

1. A line is constructed from one node of the element to the midpoint of the opposite edge, and another through the midpoints of the other 2 edges. In general, these lines are not perpendicular to each other or to any of the element edges.
2. Rectangles are constructed centered about each of these 2 lines, with edges passing through the element edge midpoints and the triangle apex.
3. These constructions are repeated using each of the other 2 corners as the apex.
4. The aspect ratio of the triangle is the ratio of the longer side to the shorter side of whichever of the 6 rectangles is most stretched, divided by the square root of 3.

The best possible triangle aspect ratio, for an equilateral triangle, is 1. A triangle having an aspect ratio of 20 is shown in Figure 13.7–8.

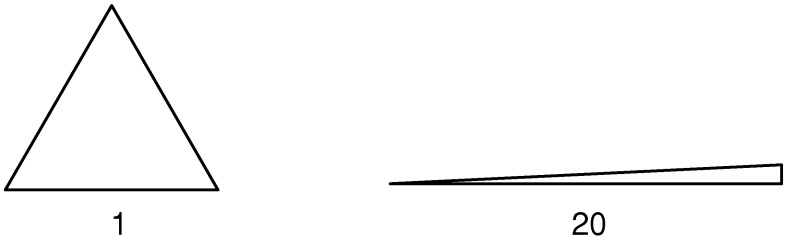


Figure 13.7–8 Aspect Ratios for Triangles

Aspect Ratio Calculation for Quadrilaterals

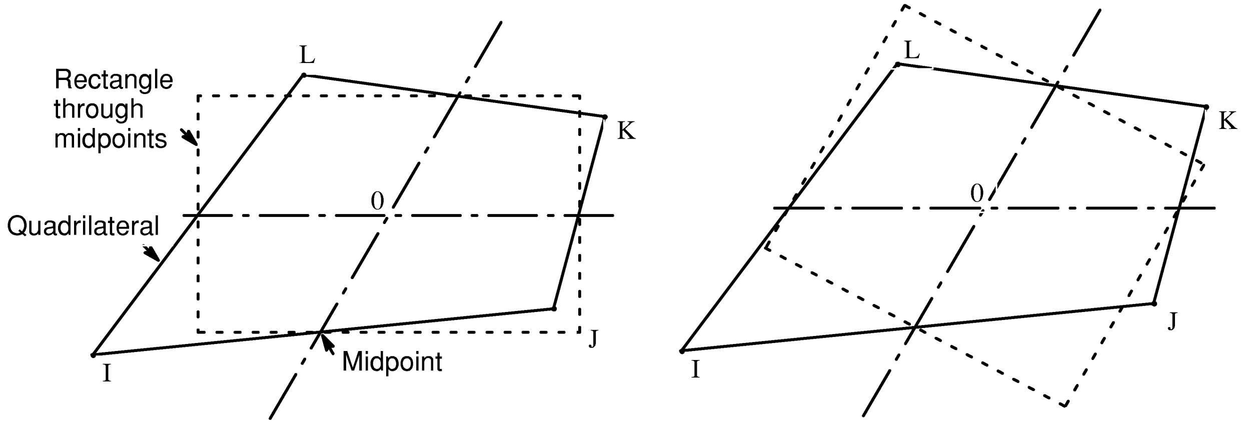


Figure 13.7–9 Quadrilateral Aspect Ratio Calculation

The aspect ratio for a quadrilateral is computed by the following steps, using only the corner nodes of the element (Figure 13.7–9):

1. If the element is not flat, the nodes are projected onto a plane passing through the average of the corner locations and perpendicular to the average of the corner normals. The remaining steps are performed on these projected locations.
2. Two lines are constructed that bisect the opposing pairs of element edges and which meet at the element center. In general, these lines are not perpendicular to each other or to any of the element edges.
3. Rectangles are constructed centered about each of the 2 lines, with edges passing through the element edge midpoints. The aspect ratio of the quadrilateral is the ratio of a longer side to a shorter side of whichever rectangle is most stretched.
4. The best possible quadrilateral aspect ratio, for a square, is one. A quadrilateral having an aspect ratio of 20 is shown in Figure 13.7–10.

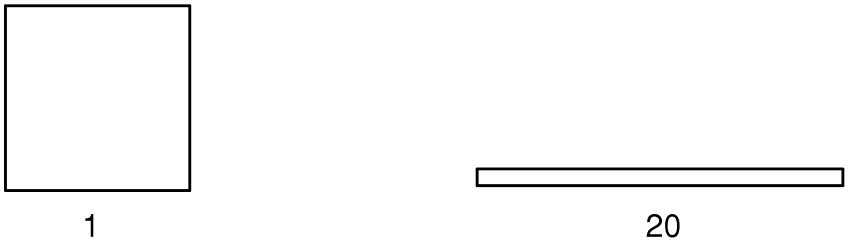


Figure 13.7–10   Aspect Ratios for Quadrilaterals

Table 13.7–1   Aspect Ratio Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,1	warning	20	Elements this stretched look to many users like they deserve warnings.	Disturbance of analysis results has not been proven. It’s difficult to avoid warnings even with a limit of 20.
SHPP,MODIFY,2	error	10^6	Informal testing has demonstrated solution error attributable to computer roundoff at aspect ratios of 1,000 to 100,000.	Threshold of roundoff problems depends on what computer is being used. Valid analyses should not be blocked.

13.7.4 Angle Deviation

Angle deviation from 90° corner angle is computed and tested only for the SHELL28 shear/twist panel quadrilateral (see Table 13.7–2). It is an important measure because the element derivation assumes a rectangle.

Angle Deviation Calculation

The angle deviation is based on the angle between each pair of adjacent edges, computed using corner node positions in 3–D space. It is simply the largest deviation from 90° of any of the 4 corner angles of the element.

The best possible deviation is 0° (Figure 13.7–11). Figure 13.7–11 also shows angle deviations of 5° and 30°, respectively.

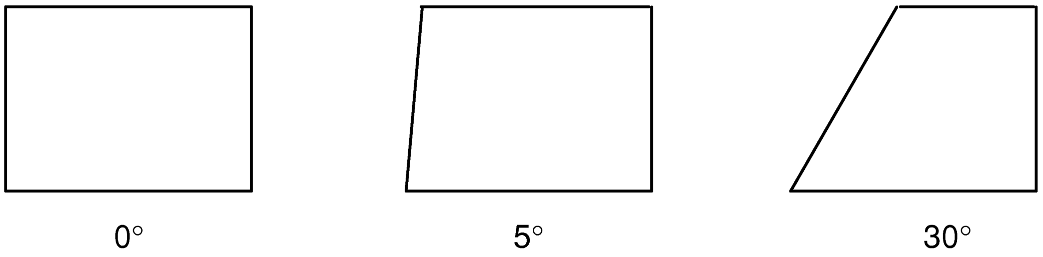


Figure 13.7–11 Angle Deviations for SHELL28

Table 13.7–2 Angle Deviation Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,7	warning	5°	Results degrade as the element deviates from a rectangular shape.	It's difficult to avoid warnings even with a limit of 5°.
SHPP,MODIFY,8	error	30°	Pushing the limit further does not seem prudent.	Valid analyses should not be blocked.

13.7.5 Parallel Deviation

Parallel deviation is computed and tested for all quadrilaterals or 3–D solid elements having quadrilateral faces or cross–sections, except Emag or FLOTRAN elements (see Table 13.7–3). Formal testing has demonstrated degradation of stress convergence in linear displacement quadrilaterals as opposite edges become less parallel to each other.

Parallel Deviation Calculation

Parallel deviation is computed using the following steps:

1. Ignoring midside nodes, unit vectors are constructed in 3–D space along each element edge, adjusted for consistent direction, as demonstrated in Figure 13.7–12.

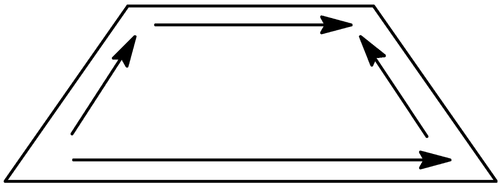


Figure 13.7–12 Parallel Deviation Calculation

2. For each pair of opposite edges, the dot product of the unit vectors is computed, then the angle (in degrees) whose cosine is that dot product. The parallel deviation is the larger of these 2 angles. (In the illustration above, the dot product of the 2 horizontal unit vectors is 1, and acos (1) = 0°. The dot product of the 2 vertical vectors is 0.342, and acos (0.342) = 70°. Therefore, this element's parallel deviation is 70°.)
3. The best possible deviation, for a flat rectangle, is 0°. Figure 13.7–13 shows quadrilaterals having deviations of 0°, 70°, 100°, 150°, and 170°.

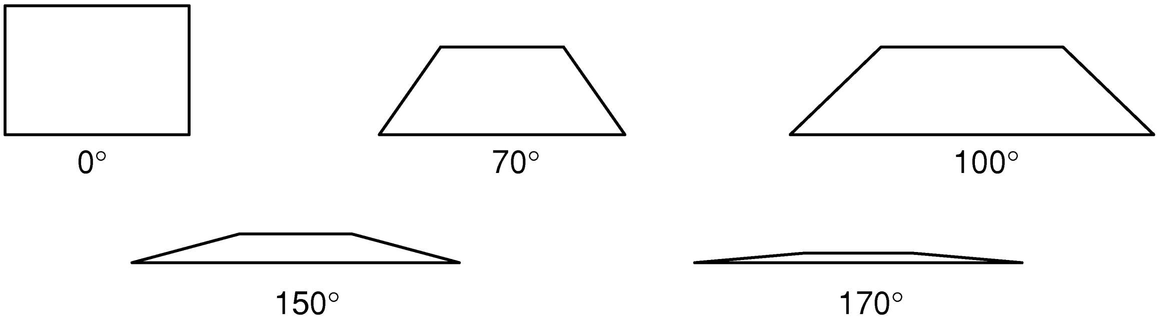


Figure 13.7–13 Parallel Deviations for Quadrilaterals

Table 13.7–3 Parallel Deviation Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,11	warning for elements without midside nodes	70°	Testing has shown results are degraded by this much distortion.	It's difficult to avoid warnings even with a limit of 70°.
SHPP,MODIFY,12	error for elements without midside nodes	150°	Pushing the limit further does not seem prudent.	Valid analyses should not be blocked.
SHPP,MODIFY,13	warning for elements with midside nodes	100°	Elements having deviations > 100° look like they deserve warnings.	Disturbance of analysis results for quadratic elements has not been proven.
SHPP,MODIFY,14	error for elements with midside nodes	170°	Pushing the limit further does not seem prudent.	Valid analyses should not be blocked.

13.7.6 Maximum Corner Angle

Maximum corner angle is computed and tested for all except Emag or FLOTRAN elements (see Table 13.7–4). Some in the finite element community have reported that large angles (approaching 180°) degrade element performance, while small angles don't.

Maximum Corner Angle Calculation

The maximum angle between adjacent edges is computed using corner node positions in 3–D space. (Midside nodes, if any, are ignored.) The best possible triangle maximum angle, for an equilateral triangle, is 60°. Figure 13.7–14 shows a triangle having a maximum corner angle of 165°. The best possible quadrilateral maximum angle, for a flat rectangle, is 90°. Figure 13.7–15 shows quadrilaterals having maximum corner angles of 90°, 140° and 180°.

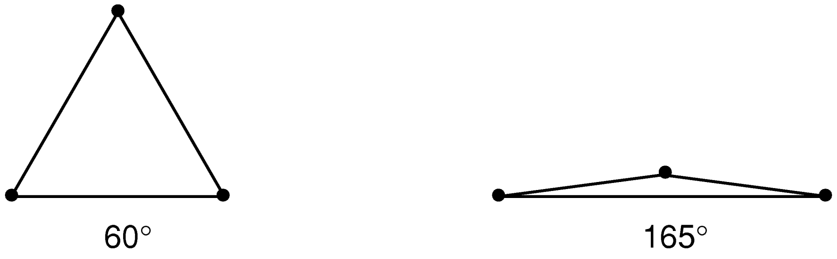


Figure 13.7–14 Maximum Corner Angles for Triangles

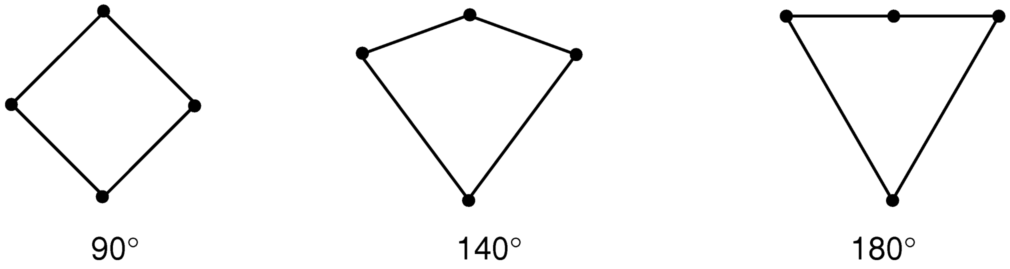


Figure 13.7–15 Maximum Corner Angles for Quadrilaterals

Table 13.7–4 Maximum Corner Angle Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,15	warning for triangles	165°	Any element this distorted looks like it deserves a warning.	Disturbance of analysis results has not been proven. It's difficult to avoid warnings even with a limit of 165°.
SHPP,MODIFY,16	error for triangles	179.9°	We can't allow 180°.	Valid analyses should not be blocked.
SHPP,MODIFY,17	warning for quadrilaterals without midside nodes	155°	Any element this distorted looks like it deserves a warning.	Disturbance of analysis results has not been proven. It's difficult to avoid warnings even with a limit of 155°.
SHPP,MODIFY,18	error for quadrilaterals without midside nodes	179.9°	We can't allow 180°.	Valid analyses should not be blocked.
SHPP,MODIFY,19	warning for quadrilaterals with midside nodes	165°	Any element this distorted looks like it deserves a warning.	Disturbance of analysis results has not been proven. It's difficult to avoid warnings even with a limit of 165°.
SHPP,MODIFY,20	error for quadrilaterals with midside nodes	179.9°	We can't allow 180°.	Valid analyses should not be blocked.

13.7.7 Jacobian Ratio

Jacobian ratio is computed and tested for all elements except triangles and tetrahedra that (a) are linear (have no midside nodes) or (b) have perfectly centered midside nodes (see Table 13.7–5). A high ratio indicates that the mapping between element space and real space is becoming computationally unreliable.

Jacobian Ratio Calculation

An element's Jacobian ratio is computed by the following steps, using the full set of nodes for the element:

1. At each sampling location listed in the table below, the determinant of the Jacobian matrix is computed and called R_J. R_J at a given point repesents the magnitude of the mapping function between element natural coordinates and real space. In an ideally–shaped element, R_J is relatively constant over the element, and does not change sign.

Element Shape	R_J Sampling Locations
10–noded tetrahedra – **SHPP**,LSTET,OFF	corner nodes
10–noded tetrahedra – **SHPP**,LSTET,ON	integration points
5–noded or 13–noded pyramids	base corner nodes and near apex node (apex R_J factored so that a pyramid having all edges the same length will produce a Jacobian ratio of 1)
8–noded quadrilaterals and 20–noded bricks	corner nodes and centroid
all other elements	corner nodes

2. The Jacobian ratio of the element is the ratio of the maximum to the minimum sampled value of R_J. If the maximum and minimum have opposite signs, the Jacobian ratio is arbitrarily assigned to be –100 (and the element is clearly unacceptable).

3. If the element is a midside–noded tetrahedron, an additional R_J is computed for a fictitious straight–sided tetrahedron connected to the 4 corner nodes. If that R_J differs in sign from any nodal R_J (an extremely rare occurrence), the Jacobian ratio is arbitrarily assigned to be –100.

4. The sampling locations for midside–noded tetrahedra depend upon the setting of the linear stress tetrahedra key on the **SHPP** command. The default behavior (**SHPP**,LSTET,OFF) is to sample at the corner nodes, while the optional behavior (**SHPP**,LSTET.ON) is to sample at the integration points (similar to as was done for the DesignSpace product). Sampling at the integration points will result in a lower Jacobian ratio than sampling at the nodes, but that ratio is compared to more restrictive default limits (see Table 13.7–5 below). Nevertheless, some elements which pass the LSTET,ON test fail the LSTET,OFF test – especially those having zero R_J at a corner node. Testing has shown that such elements have no negative effect on linear elastic stress accuracy. Their effect on other types of solutions has not been studied, which is why the more conservative test is recommended for general ANSYS usage.

A triangle or tetrahedron has a Jacobian ratio of 1 if each midside node, if any, is positioned at the average of the corresponding corner node locations. This is true no matter how otherwise distorted the element may be. Hence, this calculation is skipped entirely for such elements. Moving a midside node away from the edge midpoint position will increase the Jacobian ratio. Eventually, even very slight further movement will break the element (Figure 13.7–16). We describe this as “breaking” the element because it suddenly changes from acceptable to unacceptable– “broken”.

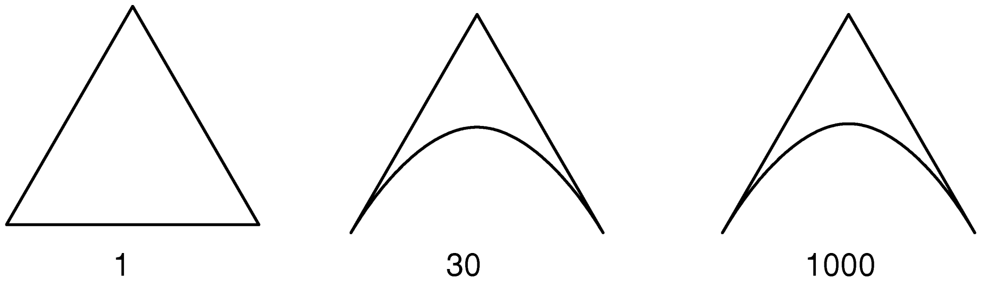


Figure 13.7–16   Jacobian Ratios for Triangles

Any rectangle or rectangular parallelepiped having no midside nodes, or having midside nodes at the midpoints of its edges, has a Jacobian ratio of 1. Moving midside nodes toward or away from each other can increase the Jacobian ratio. Eventually, even very slight further movement will break the element (Figure 13.7–17).

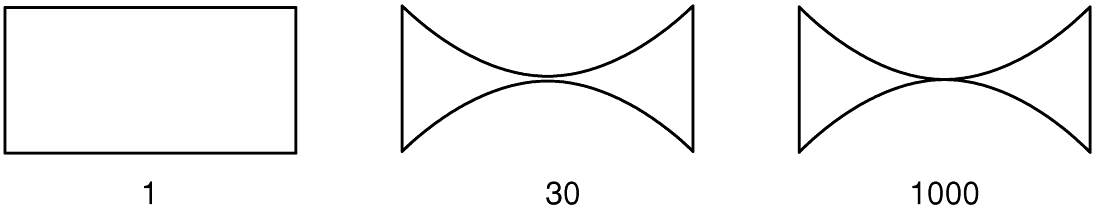


Figure 13.7–17   Jacobian Ratios for Quadrilaterals

A quadrilateral or brick has a Jacobian ratio of 1 if (a) its opposing faces are all parallel to each other, and (b) each midside node, if any, is positioned at the average of the corresponding corner node locations. As a corner node moves near the center, the Jacobian ratio climbs. Eventually, any further movement will break the element (Figure 13.7–18).

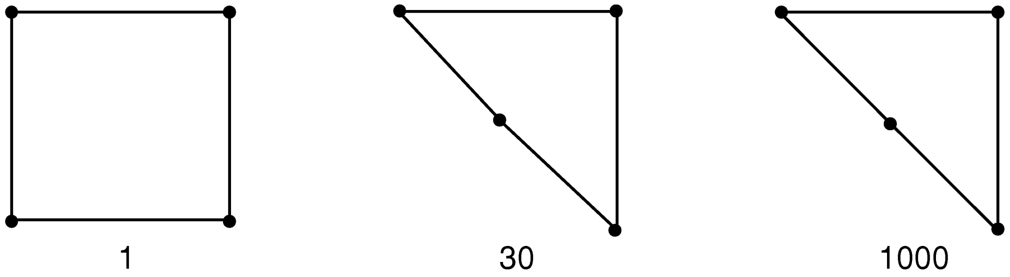


Figure 13.7–18   Jacobian Ratios for Quadrilaterials

Table 13.7–5 Jacobian Ratio Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,31	warning for h–elements	30 if **SHPP**,LSTET,OFF	A ratio this high indicates that the mapping between element and real space is becoming computationally unreliable.	Disturbance of analysis results has not been proven. It's difficult to avoid warnings even with a limit of 30.
		10 if **SHPP**,LSTET,ON		
SHPP,MODIFY,32	error for h–elements (except quarter point)	1,000 if **SHPP**,LSTET,OFF	Pushing the limit further does not seem prudent.	Valid analyses should not be blocked.
		40 if **SHPP**,LSTET,ON		
SHPP,MODIFY,33	warning for p–elements	30	A ratio this high indicates that the mapping between element & real space is becoming computationally unreliable.	Disturbance of analysisresults has not been proven. It's difficult to avoid warnings even with a limit of 30.
SHPP,MODIFY,34	error for p–elements	40	The mapping is more critical for p– than h–elements.	Valid analyses should not be blocked.

13.7.8 Warping Factor

Warping factor is computed and tested for some quadrilateral shell elements, and the quadrilateral faces of bricks, wedges, and pyramids (see Table 13.7–6 and Table 13.7–7). A high factor may indicate a condition the underlying element formulation cannot handle well, or may simply hint at a mesh generation flaw.

Warping Factor Calculation for Quadrilateral Shell Elements

A quadrilateral element's warping factor is computed from its corner node positions and other available data by the following steps:

1. An average element normal is computed as the vector (cross) product of the 2 diagonals (Figure 13.7–19).

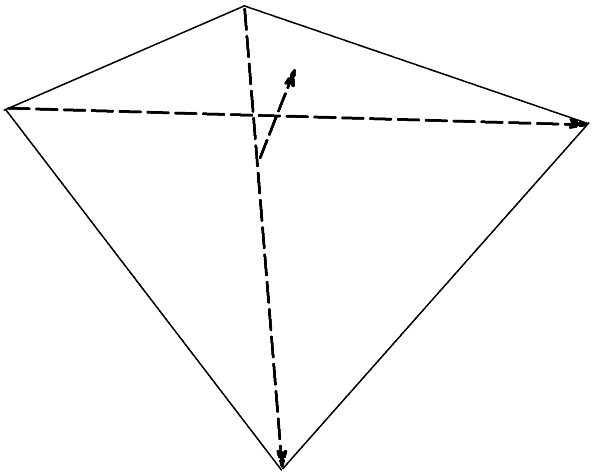


Figure 13.7–19 Shell Average Normal Calculation

2. The projected area of the element is computed on a plane through the average normal (the dotted outline on Figure 13.7–20).

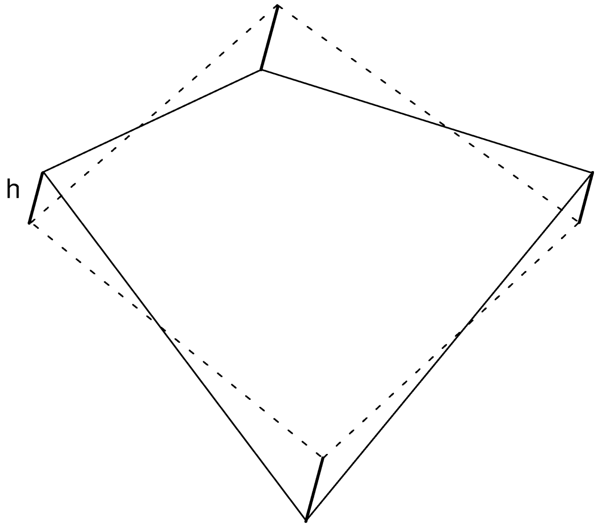


Figure 13.7–20 Shell Element Projected onto a Plane

3. The difference in height of the ends of an element edge is computed, parallel to the average normal. In Figure 13.7–20, this distance is 2h. Because of the way the average normal is constructed, h is the same at all four corners. For a flat quadrilateral, the distance is zero.

4. The “area warping factor” (F_a^w) for the element is computed as the edge height difference divided by the square root of the projected area.

5. For all shells except those in the “membrane stiffness only” group, if the real constant thickness is available, the “thickness warping factor” is computed as the edge height difference divided by the average element thickness. This could be substantially higher than the area warping factor computed in 4 (above).
6. The warping factor tested against warning and error limits (and reported in warning and error messages) is the larger of the area factor and, if available, the thickness factor.
7. The best possible quadrilateral warping factor, for a flat quadrilateral, is zero.
8. The warning and error limits for SHELL63 quadrilaterals in a large deflection analysis are much tighter than if these same elements are used with small deflection theory, so existing SHELL63 elements are retested any time the nonlinear geometry key is changed. However, in a large deflection analysis it is possible for warping to develop after deformation, causing impairment of nonlinear convergence and/or degradation of results. Element shapes are not retested during an analysis.

Figure 13.7–21 shows a “warped” element plotted on top of a flat one. Only the right–hand node of the upper element is moved. The element is a unit square, with a real constant thickness of 0.1.

When the upper element is warped by a factor of 0.01, it cannot be visibly distinguished from the underlying flat one.

When the upper element is warped by a factor of 0.04, it just begins to visibly separate from the flat one.

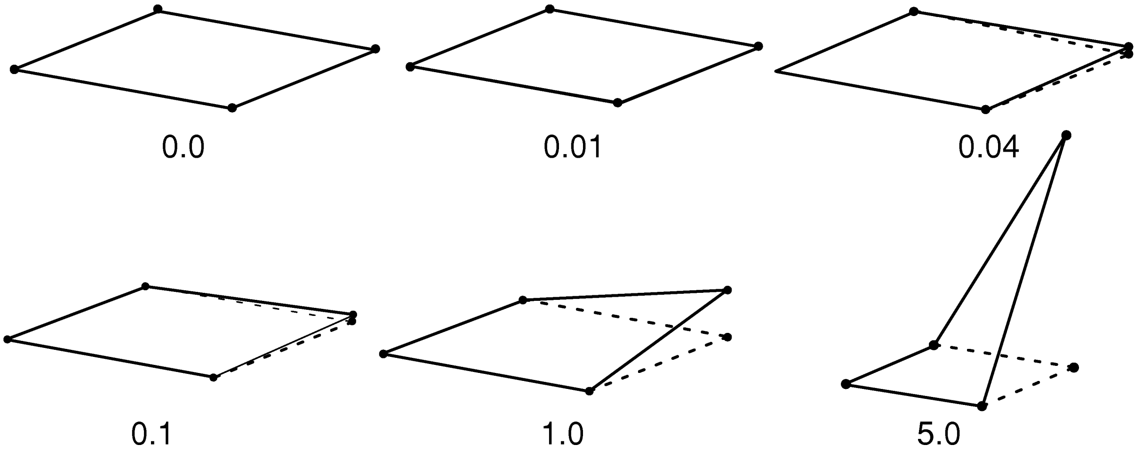


Figure 13.7–21 Quadrilateral Shell Having Warping Factor

Warping of 0.1 is visible given the flat reference, but seems trivial. However, it is well beyond the error limit for a membrane shell or a SHELL63 in a large deflection environment. Warping of 1.0 is visually unappealing. This is the error limit for most shells.

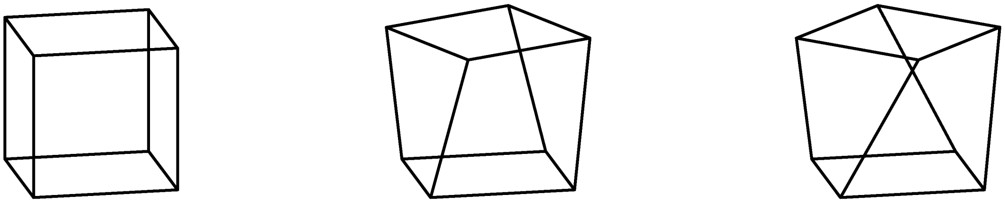
Warping beyond 1.0 would appear to be obviously unacceptable. However, SHELL43 and SHELL181 permit even this much distortion. Furthermore, the warping factor calculation seems to peak at about 7.0. Moving the node further off the original plane, even by much larger distances than shown here, does not further increase the warping factor for this geometry. Users are cautioned that manually increasing the error limit beyond its default of 5.0 for these elements could mean no real limit on element distortion.

Warping Factor Calculation for 3–D Solid Elements

The warping factor for a 3–D solid element face is computed as though the 4 nodes make up a quadrilateral shell element with no real constant thickness available, using the square root of the projected area of the face as described in 4 (above).

The warping factor for the element is the largest of the warping factors computed for the 6 quadrilateral faces of a brick, 3 quadrilateral faces of a wedge, or 1 quadrilateral face of a pyramid.

Any brick element having all flat faces has a warping factor of zero (Figure 13.7–22).



0.0   approximately 0.2   approximately 0.4

Figure 13.7–22   Warping Factor for Bricks

Twisting the top face of a unit cube by 22.5° and 45° relative to the base produces warping factors of about 0.2 and 0.4, respectively.

Table 13.7–6 Applicability of Warping Tests to Quadrilateral Shell Elements

Element Name	Description	Limits Group from Table 13.7–7	ANSYS internal key ielc(JSHELL)
SHELL28	Shear / Twist Panel	“shear / twist”	7
SHELL41	Membrane Shell	“membrane stiffness only”	4
SHELL43	Plastic Large Strain Shell	“bending with high warping limit”	2
INFIN47	3–D Infinite Boundary	“non–stress”	11
SHELL57	Thermal Shell	“non–stress”	11
SHELL63	Elastic Shell	“bending stiffness included” if KEYOPT(1) = 0 or 2	3
		“membrane stiffness only” if KEYOPT(1) = 1	4
SHELL91	Nonlinear Layered Structural Shell	none…element can curve out of plane	1
SHELL93	8–node Structural Shell	none…element can curve out of plane	1
SHELL99	Linear Layered Structural Shell	none…element can curve out of plane	1
INTER115	3–D Magnetic Interface	“non–stress”	11
SHELL143	Plastic Shell	“bending with high warping limit”	2
SHELL150	8–node Structural Shell p–Element	none…element can curve out of plane	1
SHELL157	Coupled Thermal–Electric Shell	“non–stress”	11
SHELL163	Explicit Thin Structural Shell	“bending with high warping limit”	2
SHELL181	Finite Strain Shell	“bending with high warping limit”	2

Table 13.7–7 Warping Factor Limits

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,51	warning for "bending with high warping limit" shells {ielc(JSHELL) =2}	1	Elements having warping factors > 1 look like they deserve warnings.	Element formulation derived from 8–node solid isn't disturbed by warping. Disturbance of analysis results has not been proven.
SHPP,MODIFY,52	same as above, error limit	5	Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.
SHPP,MODIFY,53	warning for "non–stress" shells or "bending stiffness included" shells without geometric nonlinearities {3,11}	0.1	The element formulation is based on flat shell theory, with rigid beam offsets for moment compatibility. Informal testing has shown that result error became significant for warping factor > 0.1.	It's difficult to avoid these warnings even with a limit of 0.1.
SHPP,MODIFY,54	same as above, error limit	1	Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.
SHPP,MODIFY,55	warning for "membrane stiffness only" shells {4}	0.00004	The element formulation is based on flat shell theory, without any correction for moment compatibility. The element cannot handle forces not in the plane of the element.	Informal testing has shown that the effect of warping < 0.00004 is negligible.
SHPP,MODIFY,56	same as above, error limit	0.04	Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.
SHPP,MODIFY,57	warning for "shear / twist" shells {7}	0.1	The element formulation is based on flat shell theory, with rigid beam offsets for moment compatibility. Informal testing has shown that result error became significant for warping factor > 0.1.	It's difficult to avoid these warnings even with a limit of 0.1.
SHPP,MODIFY,58	same as above, error limit	1	Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.

Command to modify	Type of Limit	Default	Why default is this tight	Why default is this loose
SHPP,MODIFY,59	warning for “bending stiffness included” shells with geometric nonlinearities {3}	0.00001	The element formulation is based on flat shell theory. The rigid beam offsets added to warped elements for moment compatibility do not work well with geometric nonlinearities. Informal testing has shown that nonlinear convergence was impaired and/or result error became significant for warping factor > 0.00001.	Even when the geometric construction of elements should make them perfectly flat, computational roundoff may cause some slight warping. This makes it impossible to enforce a limit much tighter than 0.00001.
SHPP,MODIFY,60	same as above, error limit	0.01	Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.
SHPP,MODIFY,67	warning for 3–D solid element quadrilateral face	0.2	A warping factor of 0.2 corresponds to about a 22.5° rotation of the top face of a unit cube. Brick elements distorted this much look like they deserve warnings.	Disturbance of analysis results has not been proven.
SHPP,MODIFY,68	same as above, error limit	0.4	A warping factor of 0.4 corresponds to about a 45° rotation of the top face of a unit cube. Pushing this limit further does not seem prudent.	Valid analyses should not be blocked.

Chapter 14
Element Library

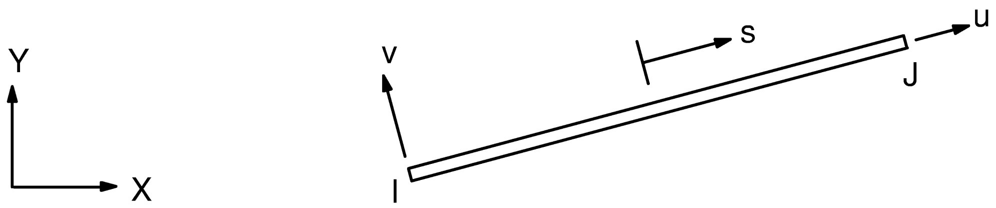
ANSYS Theory Reference
Chapter 14 – Table of Contents

14.0 Introduction

This chapter describes each element. The explanations are augmented by other sections referred to in this manual as well as the external references.

The table below the introductory figure of each element is intended to be complete, except that the Newton–Raphson load vector is omitted. This load vector always uses the same shape functions and integration points as the applicable stiffness, conductivity and/or coefficient matrix. Exceptions associated mostly with some nonlinear line elements are noted with the element description.

14.1 LINK1 — 2–D Spar (or Truss)



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.1.1–1)	None
Mass Matrix	Equations (12.1.1–1) and (12.1.1–2)	None
Stress Stiffness Matrix	Equation (12.1.1–2)	None
Thermal Load Vector	Equation (12.1.1–1)	None

Load Type	Distribution
Element Temperature	Linear along length
Nodal Temperature	Linear along length

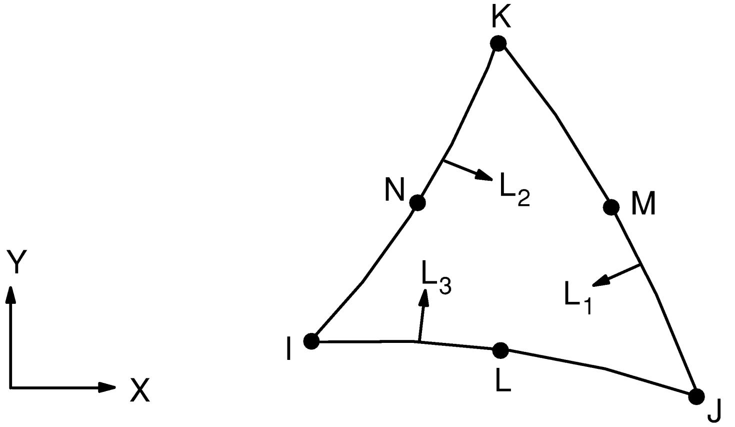
14.1.1 Assumptions and Restrictions

The element is not capable of carrying bending loads. The stress is assumed to be uniform over the entire element.

14.1.2 Other Applicable Sections

LINK8, the 3–D Spar, has analogous element matrices and load vectors described, as well as the stress printout.

14.2 PLANE2 — 2–D 6–Node Triangular Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.6.2–1) and (12.6.2–2)	3
Mass Matrix	Same as stiffness matrix	3
Stress Stiffness Matrix	Same as stiffness matrix	3
Thermal Load Vector	Same as stiffness matrix	3
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2

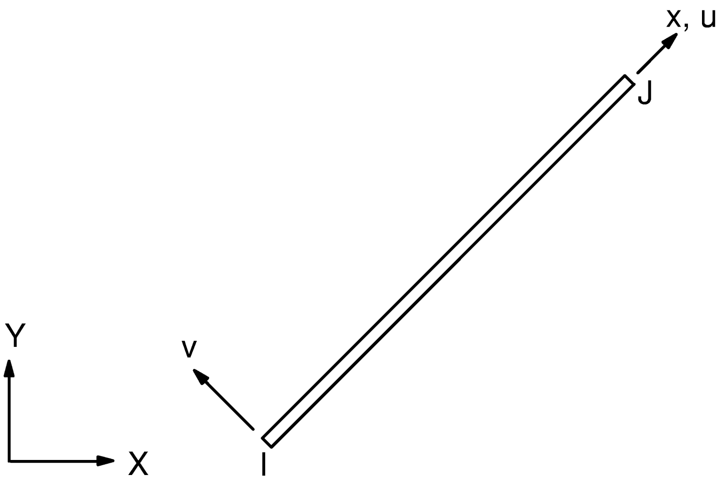
Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Pressure	Linear along each face

Reference: Zienkiewicz(39)

14.2.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.3 BEAM3 — 2–D Elastic Beam



Matrix Or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.1.2–1) and (12.1.2–2)	None
Mass Matrix	Same as stiffness matrix	None
Stress Stiffness Matrix	Equation (12.1.2–2)	None
Thermal and Pressure Load Vector	Same as stiffness matrix	None

Load Type	Distribution
Element Temperature	Linear thru thickness and along length
Nodal Temperature	Constant thru thickness, linear along length
Pressure	Linear along length

14.3.1 Element Matrices and Load Vectors

The element stiffness matrix in element coordinates is (Przemieniecki(28)):

$$[K_\ell] = \begin{bmatrix} \frac{AE}{L} & 0 & 0 & -\frac{AE}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3(1+\phi)} & \frac{6EI}{L^2(1+\phi)} & 0 & -\frac{12EI}{L^3(1+\phi)} & \frac{6EI}{L^2(1+\phi)} \\ 0 & \frac{6EI}{L^2(1+\phi)} & \frac{EI(4+\phi)}{L(1+\phi)} & 0 & -\frac{6EI}{L^2(1+\phi)} & \frac{EI(2-\phi)}{L(1+\phi)} \\ -\frac{AE}{L} & 0 & 0 & \frac{AE}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3(1+\phi)} & -\frac{6EI}{L^2(1+\phi)} & 0 & \frac{12EI}{L^3(1+\phi)} & -\frac{6EI}{L^2(1+\phi)} \\ 0 & \frac{6EI}{L^2(1+\phi)} & \frac{EI(2-\phi)}{L(1+\phi)} & 0 & -\frac{6EI}{L^2(1+\phi)} & \frac{EI(4+\phi)}{L(1+\phi)} \end{bmatrix} \tag{14.3–1}$$

where:

A = cross–section area (input as AREA on **R** command)

E = Young’s modulus (input as EX on **MP** command)

L = element length

I = moment of inertia (input as IZZ on **R** command)

$\phi = \frac{12EI}{GA^sL^2}$

G = shear modulus (input as GXY on **MP** command)

$A^s = \frac{A}{F^s}$ = shear area

F^s = shear deflection constant (input as SHEARZ on **R** command)

The element mass matrix in element coordinates for **LUMPM**,OFF is (Yokoyama(167)):

$$
\left[M_\ell\right] = (\rho A + m)\,L\,\left(1 - \epsilon^{in}\right)
\begin{bmatrix}
1/3 & 0 & 0 & 1/6 & 0 & 0 \\
0 & A(r,\phi) & C(r,\phi) & 0 & B(r,\phi) & -D(r,\phi) \\
0 & C(r,\phi) & E(r,\phi) & 0 & D(r,\phi) & -F(r,\phi) \\
1/6 & 0 & 0 & 1/3 & 0 & 0 \\
0 & B(r,\phi) & D(r,\phi) & 0 & A(r,\phi) & -C(r,\phi) \\
0 & -D(r,\phi) & -F(r,\phi) & 0 & -C(r,\phi) & E(r,\phi)
\end{bmatrix}
\tag{14.3–2}
$$

where: ϱ = density (input as DENS on **MP** command)

m = added mass per unit length (input as ADDMAS on **R** command)

ε^{in} = prestrain (input as ISTRN on **R** command)

$$A(r,\phi) = \frac{\frac{13}{35} + \frac{7}{10}\phi + \frac{1}{3}\phi^2 + \frac{6}{5}(r/L)^2}{(1+\phi)^2}$$

$$B(r,\phi) = \frac{\frac{9}{70} + \frac{3}{10}\phi + \frac{1}{6}\phi^2 - \frac{6}{5}(r/L)^2}{(1+\phi)^2}$$

$$C(r,\phi) = \frac{\left(\frac{11}{210} + \frac{11}{120}\phi + \frac{1}{24}\phi^2 + \left(\frac{1}{10} - \frac{1}{2}\phi\right)(r/L)^2\right)L}{(1+\phi)^2}$$

$$D(r,\phi) = \frac{\left(\frac{13}{420} + \frac{3}{40}\phi + \frac{1}{24}\phi^2 - \left(\frac{1}{10} - \frac{1}{2}\phi\right)(r/L)^2\right)L}{(1+\phi)^2}$$

$$E(r,\phi) = \frac{\left(\frac{1}{105} + \frac{1}{60}\phi + \frac{1}{120}\phi^2 + \left(\frac{2}{15} + \frac{1}{6}\phi + \frac{1}{3}\phi^2\right)(r/L)^2\right)L^2}{(1+\phi)^2}$$

$$F(r,\phi) = \frac{\left(\frac{1}{140} + \frac{1}{60}\phi + \frac{1}{120}\phi^2 + \left(\frac{1}{30} + \frac{1}{6}\phi + \frac{1}{6}\phi^2\right)(r/L)^2\right)L^2}{(1+\phi)^2}$$

r = $\sqrt{\frac{I}{A}}$ = radius of gyration

The element mass matrix in element coordinates for **LUMPM**,ON is:

$$\left[M_\ell\right] = \frac{(\rho A + m)\, L \left(1-\epsilon^{in}\right)}{2} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (14.3–3)$$

The element pressure load vector in element coordinates is:

$$\left\{F_\ell^{pr}\right\} = \lfloor P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5 \quad P_6 \rfloor^T \quad (14.3–4)$$

For uniform lateral pressure,

$$P_1 = P_4 = 0 \quad (14.3–5)$$

$$P_2 = P_5 = -\frac{PL}{2} \quad (14.3–6)$$

$$P_3 = -P_6 = -\frac{PL^2}{12} \quad (14.3–7)$$

where: P = uniform applied pressure (units = force/length) (input on **SFE** command)

Other standard formulas (Roark(48)) for P_1 through P_6 are used for linearly varying loads, partially loaded elements, and point loads.

14.3.2 Stress Calculations

The centroidal stress at end i is:

$$\sigma_i^{dir} = \frac{F_{x,i}}{A} \quad (14.3–8)$$

where: σ_i^{dir} = centroidal stress (output quantity SDIR)

$F_{x,i}$ = axial force (output quantity FORCE)

The bending stress is

$$\sigma_i^{bnd} = \frac{M_i\, t}{2I} \quad (14.3–9)$$

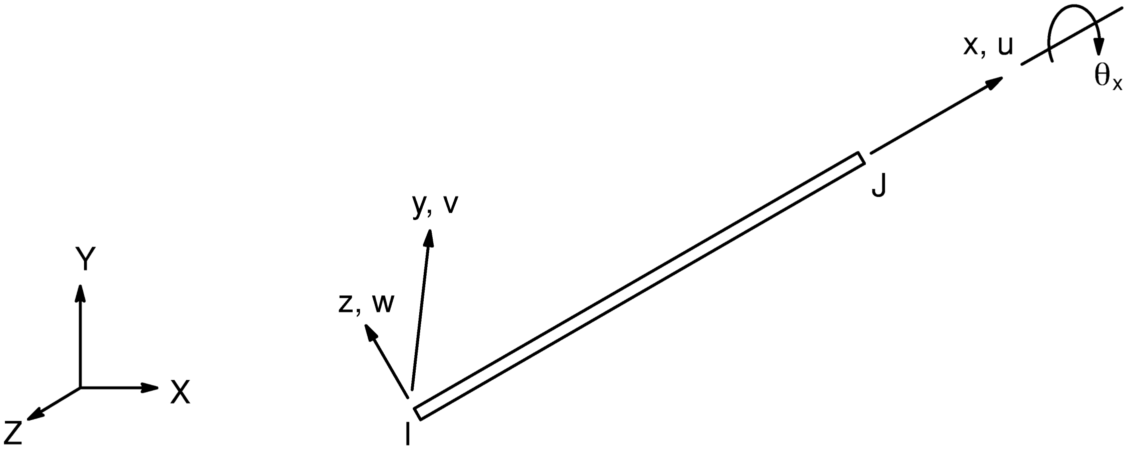
where: σ_i^{bnd} = bending stress at end i (output quantity SBEND)

M_i = moment at end i

t = thickness of beam in element y direction (input as HEIGHT on **R** command)

The presumption has been made that the cross–section is symmetric.

14.4 BEAM4 — 3–D Elastic Beam



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.2–1), (12.2.2–2), (12.2.2–3), and (12.2.2–4)	None
Mass Matrix	Same as stiffness matrix	None
Stress Stiffness Matrix	Equation (12.2.2–2) and (12.2.2–3)	None
Pressure Load Vector and Temperatures	Equation (12.2.2–1), (12.2.2–2) and (12.2.2–3)	None

Load Type	Distribution
Element Temperature	Bilinear across cross–section, linear along length
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Linear along length

14.4.1 Stiffness and Mass Matrices

The order of degrees of freedom (DOFs) is shown in Figure 14.4–1.

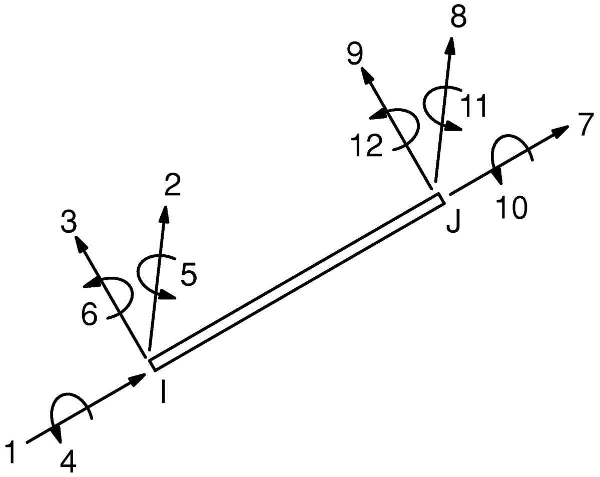


Figure 14.4–1 Order of Degrees of Freedom

The stiffness matrix in element coordinates is (Przemieniecki(28)):

$$[K_\ell] = \begin{bmatrix}
AE/L & & & & & & & & & & & \\
0 & a_z & & & & & & & & & & \\
0 & 0 & a_y & & & & & & & & & \\
0 & 0 & 0 & GJ/L & & & & & & \text{Symmetric} & & \\
0 & 0 & d_y & 0 & e_y & & & & & & & \\
0 & c_z & 0 & 0 & 0 & e_z & & & & & & \\
-AE/L & 0 & 0 & 0 & 0 & 0 & AE/L & & & & & \\
0 & b_z & 0 & 0 & 0 & d_z & 0 & a_z & & & & \\
0 & 0 & b_y & 0 & c_y & 0 & 0 & 0 & a_y & & & \\
0 & 0 & 0 & -GJ/L & 0 & 0 & 0 & 0 & 0 & GJ/L & & \\
0 & 0 & d_y & 0 & f_y & 0 & 0 & 0 & c_z & 0 & e_y & \\
0 & c_z & 0 & 0 & 0 & f_z & 0 & d_y & 0 & 0 & 0 & e_z
\end{bmatrix} \tag{14.4–1}$$

where:

A = cross–section area (input as AREA on **R** command)

E = Young’s modulus (input as EX on **MP** command)

L = element length

G = shear modulus (input as GXY on **MP** command)

J = torsional moment of inertia = $\begin{Bmatrix} J_x \text{ if } I_x = 0 \\ I_x \text{ if } I_x \neq 0 \end{Bmatrix}$

I_x = input as IXX on **RMORE** command

J_x = polar moment of inertia = $I_y + I_z$

z = $a(I_z,\phi_y)$

a_y = $a(I_y,\phi_z)$

b_z = $b(I_z,\phi_y)$

$\vdots$

f_z = $f(I_z,\phi_y)$

f_y = $f(I_y,\phi_z)$

and where: $a(I,\phi) = \dfrac{12EI}{L^3(1+\phi)}$

$b(I,\phi) = \dfrac{-12EI}{L^3(1+\phi)}$

$c(I,\phi) = \dfrac{6EI}{L^2(1+\phi)}$

$d(I,\phi) = \dfrac{-6EI}{L^2(1+\phi)}$

$e(I,\phi) = \dfrac{(4+\phi)EI}{L(1+\phi)}$

$f(I,\phi) = \dfrac{(2-\phi)EI}{L(1+\phi)}$

and where: $\phi_y = \dfrac{12EI_z}{GA^s_zL^2}$

$\phi_z = \dfrac{12EI_y}{GA^s_yL^2}$

I_i = moment of inertia normal to direction i (input as Iii on **R** command)

A^s_i = shear area normal to direction i = A/F^s_i

F^s_i = shear coefficient (input as SHEARi on **RMORE** command)

The mass matrix in element coordinates with **LUMPM**,OFF is (Yokoyama(167)):

$$[M_\ell] = M_t \begin{bmatrix}
1/3 & & & & & & & & & & & \\
0 & A_z & & & & & & & & & & \\
0 & 0 & A_y & & & & & & & & & \\
0 & 0 & 0 & J_x/3A & & & & & & \text{Symmetric} & & \\
0 & 0 & -C_y & 0 & E_y & & & & & & & \\
0 & C_z & 0 & 0 & 0 & E_z & & & & & & \\
1/6 & 0 & 0 & 0 & 0 & 0 & 1/3 & & & & & \\
0 & B_z & 0 & 0 & 0 & D_z & 0 & A_z & & & & \\
0 & 0 & B_y & 0 & -D_y & 0 & 0 & 0 & A_y & & & \\
0 & 0 & 0 & J_x/6A & 0 & 0 & 0 & 0 & 0 & J_x/3A & & \\
0 & 0 & D_y & 0 & F_y & 0 & 0 & 0 & C_y & 0 & E_y & \\
0 & -D_z & 0 & 0 & 0 & F_z & 0 & -C_z & 0 & 0 & 0 & E_z
\end{bmatrix} \qquad (14.4\text{–}2)$$

where: M_t = $(\varrho A+m)L(1-\varepsilon^{in})$

ϱ = density (input as DENS on **MP** command)

m = added mass per unit length (input as ADDMAS on **RMORE** command)

ε^{in} = prestrain (input as ISTRN on **RMORE** command)

A_z = $A(r_z,\phi_y)$

A_y = $A(r_y,\phi_z)$

B_z = $B(r_z,\phi_y)$

⋮

F_z = $F(r_z,\phi_y)$

F_y = $F(r_y,\phi_z)$

and where:

$$A(r,\phi) = \frac{\frac{13}{35} + \frac{7}{10}\phi + \frac{1}{3}\phi^2 + \frac{6}{5}(r/L)^2}{(1+\phi)^2}$$

$$B(r,\phi) = \frac{\frac{9}{70} + \frac{3}{10}\phi + \frac{1}{6}\phi^2 - \frac{6}{5}(r/L)^2}{(1+\phi)^2}$$

$$C(r,\phi) = \frac{\left(\frac{11}{210} + \frac{11}{120}\phi + \frac{1}{24}\phi^2 + \left(\frac{1}{10} - \frac{1}{2}\phi\right)(r/L)^2\right)L}{(1+\phi)^2}$$

$$D(r,\phi) = \frac{\left(\frac{13}{420} + \frac{3}{40}\phi + \frac{1}{24}\phi^2 - \left(\frac{1}{10} - \frac{1}{2}\phi\right)(r/L)^2\right)L}{(1+\phi)^2}$$

$$E(r,\phi) = \frac{\left(\frac{1}{105} + \frac{1}{60}\phi + \frac{1}{120}\phi^2 + \left(\frac{2}{15} + \frac{1}{6}\phi + \frac{1}{3}\phi^2\right)(r/L)^2\right)L^2}{(1+\phi)^2}$$

$$F(r,\phi) = \frac{-\left(\frac{1}{140} + \frac{1}{60}\phi + \frac{1}{120}\phi^2 + \left(\frac{1}{30} + \frac{1}{6}\phi - \frac{1}{6}\phi^2\right)(r/L)^2\right)L^2}{(1+\phi)^2}$$

$$r_y = \sqrt{\frac{I_{yy}}{A}} = \text{radius of gyration}$$

$$r_z = \sqrt{\frac{I_{zz}}{A}} = \text{radius of gyration}$$

The mass matrix in element coordinates with **LUMP**,ON is:

$$[M_\ell] = \frac{M_t}{2}\begin{bmatrix}
1 & & & & & & & & & & & \\
0 & 1 & & & & & & & & & & \\
0 & 0 & 1 & & & & & & & & & \\
0 & 0 & 0 & 0 & & & & & \text{Symmetric} & & & \\
0 & 0 & 0 & 0 & 0 & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \qquad (14.4\text{–}3)$$

14.4.2 Gyroscopic Damping Matrix

The element gyroscopic damping matrix is the same as for PIPE16, discussed in Section 14.16.

14.4.3 Pressure and Temperature Load Vector

The pressure and temperature load vector are computed in a manner similar to that of BEAM3 (Section 14.3).

14.4.4 Local to Global Conversion

The element coordinates are related to the global coordinates by:

$$\{u_\ell\} = [T_R]\{u\} \tag{14.4–4}$$

where: $\{u_\ell\}$ = vector of displacements in element Cartesian coordinates

$\{u\}$ = vector of displacements in global Cartesian coordinates

$$[T_R] = \begin{bmatrix} T & 0 & 0 & 0 \\ 0 & T & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{bmatrix}$$

[T] is defined by:

$$[T] = \begin{bmatrix} C_1C_2 & S_1C_2 & S_2 \\ (-C_1S_2S_3 - S_1C_3) & (-S_1S_2S_3 + C_1C_3) & S_3C_2 \\ (-C_1S_2C_3 + S_1S_3) & (-S_1S_2C_3 - C_1S_3) & C_3C_2 \end{bmatrix} \tag{14.4–5}$$

where: $S_1 = \begin{cases} \dfrac{Y_2 - Y_1}{L_{xy}} & \text{if } L_{xy} > d \\ 0.0 & \text{if } L_{xy} < d \end{cases}$

$S_2 = \dfrac{Z_2 - Z_1}{L}$

$S_3 = \sin(\theta)$

$$C_1 = \begin{cases} \dfrac{X_2 - X_1}{L_{xy}} & \text{if } L_{xy} > d \\ 1.0 & \text{if } L_{xy} < d \end{cases}$$

$$C_2 = \frac{L_{xy}}{L}$$

C_3 = cos (θ)

X_1, etc. = x coordinate of node 1, etc.

L_{xy} = projection of length onto X–Y plane

d = .0001 L

θ = input as THETA on **R** command

If a third node is given, θ is not used. Rather C_3 and S_3 are defined using:

$\{V_1\}$ = vector from origin to node 1

$\{V_2\}$ = vector from origin to node 2

$\{V_3\}$ = vector from origin to node 3

$\{V_4\}$ = unit vector parallel to global Z axis, unless element is almost parallel to Z axis, in which case it is parallel to the X axis.

Then,

$$\{V_5\} = \{V_3\} - \{V_1\} = \text{vector between nodes I and K} \tag{14.4–6}$$

$$\{V_6\} = \{V_2\} - \{V_1\} = \text{vector along element X axis} \tag{14.4–7}$$

$$\{V_7\} = \{V_6\} \times \{V_4\} \tag{14.4–8}$$

$$\{V_8\} = \{V_6\} \times \{V_5\} \tag{14.4–9}$$

and

$$C_3 = \frac{\{V_7\} \cdot \{V_8\}}{|\{V_7\}|\ |\{V_8\}|} \tag{14.4–10}$$

$$S_3 = \frac{\{V_6\} \cdot \left(\{V_9\} \times \{V_8\}\right)}{|\{V_6\}|\ |\{V_9\}|\ |\{V_8\}|} \tag{14.4–11}$$

The × and · refer to vector cross and dot products, respectively. Thus, the element stiffness matrix in global coordinates becomes:

$$[K_e] = [T_R]^T [K_\ell] [T_R] \tag{14.4–12}$$

$$[M_e] = [T_R]^T [M_\ell] [T_R] \tag{14.4–13}$$

$$[S_e] = [T_R]^T [S_\ell] [T_R] \tag{14.4–14}$$

$$\{F_e\} = [T_R]^T \{F_\ell\} \tag{14.4–15}$$

($[S_\ell]$ is defined in Section 3.1).

14.4.5 Stress Calculations

The centroidal stress at end i is:

$$\sigma_i^{dir} = \frac{F_{x,i}}{A} \tag{14.4–16}$$

where:

σ_i^{dir} = centroidal stress (output quantity SDIR)

$F_{x,i}$ = axial force (output quantity FX)

The bending stresses are

$$\sigma_{z,i}^{bnd} = \frac{M_{y,i}\ t_z}{2I_y} \tag{14.4–17}$$

$$\sigma_{y,i}^{bnd} = \frac{M_{z,i}\ t_y}{2I_z} \tag{14.4–18}$$

where:

$\sigma_{z,i}^{bnd}$ = bending stress in element x direction on the element + z side of the beam at end i (output quantity SBZ)

$\sigma_{y,i}^{bnd}$ = bending stress on the element in element x direction – y side of the beam at end i (output quantity SBY)

$M_{y,i}$ = moment about the element y axis at end i

$M_{z,i}$ = moment about the element z axis at end i

t_z = thickness of beam in element z direction (input as TKZ on **R** command)

t_y = thickness of beam in element y direction (input as TKY on **R** command)

The maximum and minimum stresses are:

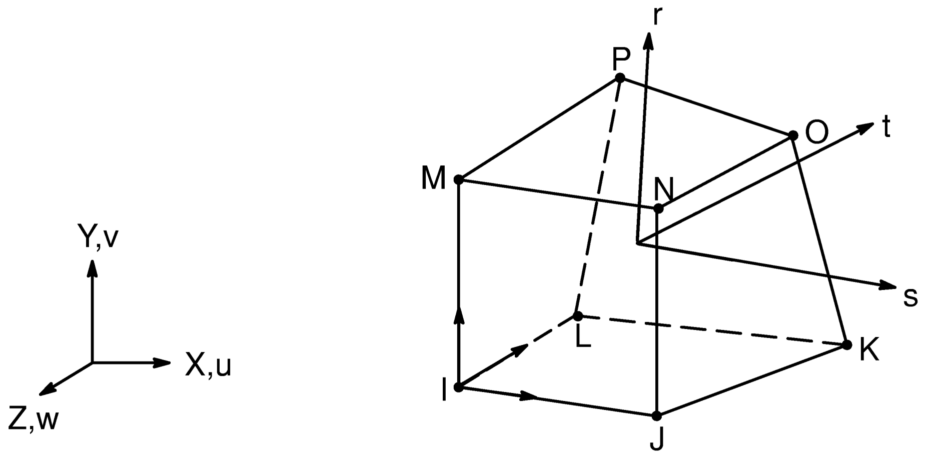
$$\sigma_i^{max} = \sigma_i^{dir} + \left|\sigma_{z,i}^{bnd}\right| + \left|\sigma_{y,i}^{bnd}\right| \tag{14.4–19}$$

$$\sigma_i^{min} = \sigma_i^{dir} - \left|\sigma_{z,i}^{bnd}\right| - \left|\sigma_{y,i}^{bnd}\right| \tag{14.4–20}$$

The presumption has been made that the cross–section is a rectangle, so that the maximum and minimum stresses of the cross–section occur at the corners. If the cross–section is of some other form, such as an ellipse, the user must replace equations (14.4–19) and (14.4–20) with other more appropriate expressions.

For long members, subjected to distributed loading (such as acceleration or pressure), it is possible that the peak stresses occur not at one end or the other, but somewhere in between. If this is of concern, the user should either use more elements or compute the interior stresses outside of the program.

14.5 SOLID5 — 3–D Coupled Field Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Equation (12.8.18–22)	2 x 2 x 2
Electrical Conductivity Matrix	Equation (12.8.18–21)	2 x 2 x 2
Thermal Conductivity Matrix	Equation (12.8.18–20)	2 x 2 x 2
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3) or, if modified extra shapes are included (KEYOPT(3) = 0), equations (12.8.19–1), (12.8.19–2), and (12.8.19–3)	2 x 2 x 2
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix without extra displacement shapes and conductivity matrix.	2 x 2 x 2
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Section 12.2	2 x 2 x 2

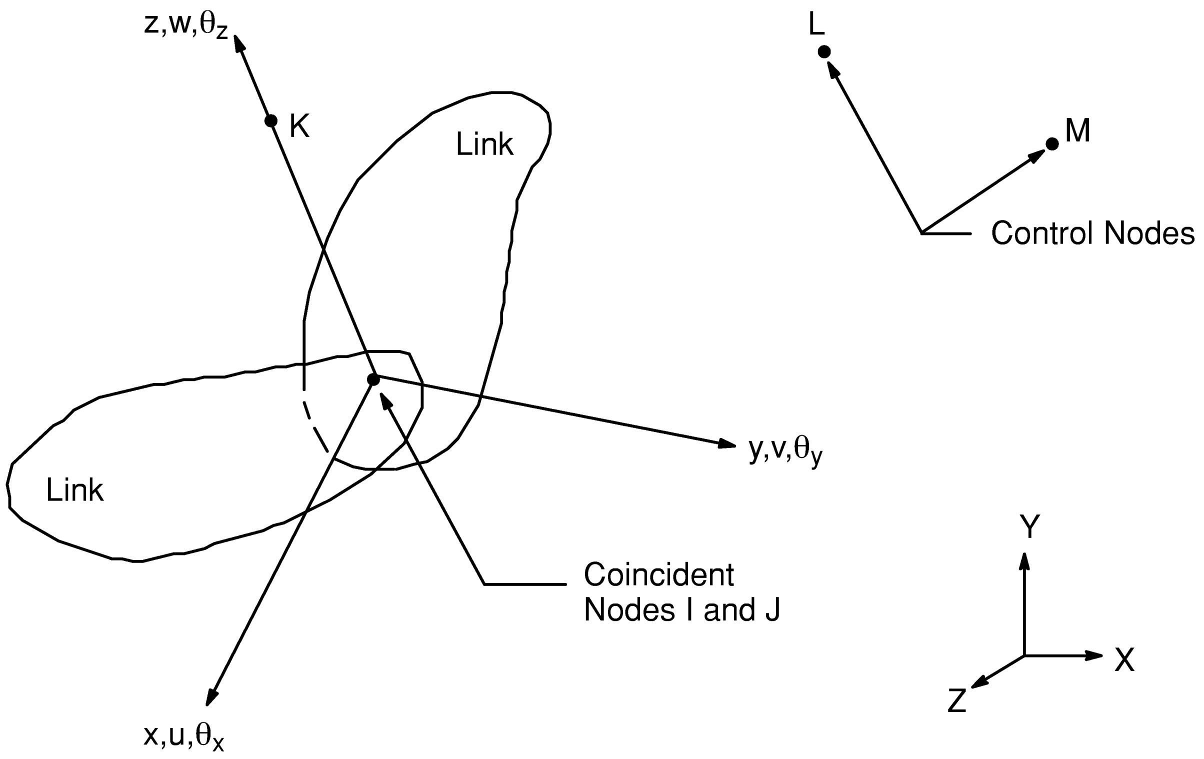
Matrix or Vector	Shape Functions	Integration Points
Mass Matrix (M)	Equation (12.8.18–1), (12.8.18–2), and (12.8.18–3)	2 x 2 x 2
Stress Stiffening Matrix	Same as mass matrix	2 x 2 x 2
Load Vector for Thermal Expansion	Same as stiffness matrix	2 x 2 x 2
Load Vector due to Imposed Thermal and Electric Gradients, Heat Generation, Joule Heating, Magnetic Forces, Magnetism due to Source Currents and Permanent Magnets	Same as coefficient or conductivity matrix	2 x 2 x 2
Load Vector due to Convection Surfaces and Pressures	Same as stiffness or conductivity matrix specialized to the surface.	2 x 2 x 2

References: Wilson (38), Taylor (49), Coulomb(76), Mayergoyz(119), Gyimesi(141, 149)

14.5.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Section 5.2 discusses the scalar potential method, which is used by this element. Section 11.1 discusses the piezoelectric capability used by the element. Section 13.1 describes integration point locations. Also, Section 14.69 discusses the thermo–electric capability.

14.7 COMBIN7 — Revolute Joint



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (nodes should be coincident)	None
Mass Matrix	None (lumped mass formulation)	None
Damping Matrix	None	None
Load Vector	None	None

14.7.1 Element Description

COMBIN7 is a 5–node, 3–D structural element that is intended to represent a pin (or revolute) joint. The pin element connects two links of a kinematic assemblage. Nodes I and J are active and physically represent the pin joint. Node K defines the initial (first iteration) orientation of the moving joint coordinate system (x, y, z), while nodes L and M are control nodes that introduce a certain level of feedback to the behavior of the element.

In kinematic terms, a pin joint has only one primary DOF, which is a rotation (θ_z) about the pin axis (z). The joint element has six DOFs per node (I and J) : three translations (u, v, w) and three rotations (θ_x, θ_y, θ_z) referenced to element coordinates (x, y, z). Two of the DOFs (θ_z for nodes I and J) represent the pin rotation. The remaining 10 DOFs have a relatively high stiffness (see below). Among other options available are rotational limits, feedback control, friction, and viscous damping.

Flexible behavior for the constrained DOF is defined by the following input quantities:

K_1 = spring stiffness for translation in the element x–y plane (input as K1 on **R** command)

K_2 = spring stiffness for translation in the element z direction (input as K2 on **R** command)

K_3 = spring stiffness for rotation about the element x and y axes (input as K3 on **R** command)

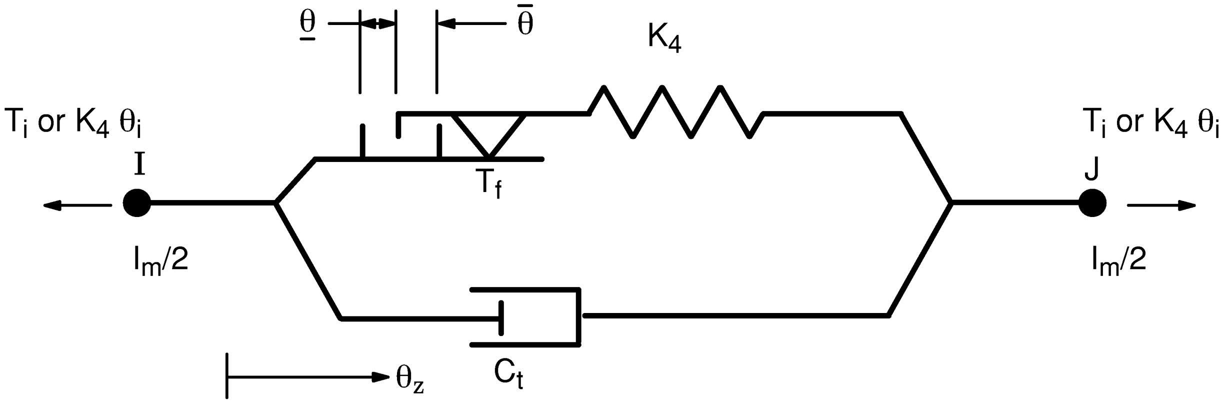


Figure 14.7–23 Joint Element Dynamic Behavior About the Revolute Axis

The dynamics of the primary DOF (θ_z) of the pin is shown in Figure 14.7–23. Input quantities are:

K_4 = rotational spring stiffness about the pin axis when the element is “locked” (input as K4 on **R** command)

T_f = friction limit torque (input as TF on **R** command)

C_t = rotational viscous friction (input as CT on **R** command)

T_i = imposed element torque (input as TLOAD on **RMORE** command)

$\underline{\theta}$ = reverse rotation limit (input as STOPL on **RMORE** command)

$\bar{\theta}$ = forward rotation limit (input as STOPU on **RMORE** command)

θ_i = imposed (or interference) rotation (input as ROT on **RMORE** command)

I_m = joint mass (input as MASS on **RMORE** command)

A simple pin can be modeled by merely setting $K_4 = 0$, along with $K_i > 0$ (i = 1 to 3). Alternately, when $K_4 > 0$, a simple pin is formed with zero friction ($T_f = 0$). The total differential rotation of the pin is given by:

$$\theta_t = \theta_{zJ} - \theta_{zI} \tag{14.7–1}$$

When friction is present ($T_f > 0$), this may be divided into two parts, namely:

$$\theta_t = \theta_f + \theta_K \tag{14.7–2}$$

where: θ_f = the amount of rotation associated with friction

θ_K = the rotation associated with the spring (i.e., spring torque /K_4)

One extreme condition occurs when $T_f = 0$, and it follows that $\theta_K = 0$ and $\theta_t = \theta_f$. On the other hand, when a high level of friction is specified to the extent that the spring torque never exceeds T_f, then it follows that $\theta_f = 0$ and $\theta_t = \theta_K$. When a negative friction torque is specified ($T_f < 0$), the pin axis is “locked” (or stuck) with revolute stiffness K_4. The pin also becomes locked when a stop is engaged, that is when:

$$\theta_f \geq \overline{\theta} \text{ (forward stop engaged)} \tag{14.7–3}$$

$$\theta_f \leq -\underline{\theta} \text{ (reverse stop engaged)} \tag{14.7–4}$$

Stopping action is removed when $\underline{\theta} = \overline{\theta} = 0$.

Internal self–equilibrating element torques are imposed about the pin axis if either T_i or θ_i are specified. If T_i is specified, the internal torques applied to the active nodes are:

$$T_J = -T_I = T_i \tag{14.7–5}$$

If a local rotation θ_i is input, it is recommended that one should set $T_f < 0$, $K_4 > 0$, and $T_i = 0$. Internal loads then become

$$T_J = -T_I = K_4\theta_i \tag{14.7–6}$$

Element Matrices

For this element, nonlinear behavior arises when sliding friction is present, stops are specified, control features are active, or large rotations are represented.

As mentioned above, there are two active nodes and six DOFs per node. Thus, the size of the element mass, damping, and stiffness matrices is 12 x 12, with a 12 x 1 load vector.

The stiffness matrix is given by:

$$[K] = \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 & 0 & -K_1 & 0 & 0 & 0 & 0 & 0 \\ & K_1 & 0 & 0 & 0 & 0 & 0 & -K_1 & 0 & 0 & 0 & 0 \\ & & K_2 & 0 & 0 & 0 & 0 & 0 & -K_2 & 0 & 0 & 0 \\ & & & K_3 & 0 & 0 & 0 & 0 & 0 & -K_3 & 0 & 0 \\ & & & & K_3 & 0 & 0 & 0 & 0 & 0 & -K_3 & 0 \\ & & & & & K_p & 0 & 0 & 0 & 0 & 0 & -K_p \\ & & & & & & K_1 & 0 & 0 & 0 & 0 & 0 \\ & & \text{Symmetry} & & & & & K_1 & 0 & 0 & 0 & 0 \\ & & & & & & & & K_2 & 0 & 0 & 0 \\ & & & & & & & & & K_3 & 0 & 0 \\ & & & & & & & & & & K_3 & 0 \\ & & & & & & & & & & & K_p \end{bmatrix} \qquad (14.7\text{–}7)$$

where:

$$K_p = \begin{cases} K_4, & \begin{cases} \text{if } \theta_f \geq \overline{\theta} \text{ or } \theta_f \leq -\underline{\theta} \text{ and both } \overline{\theta} \text{ and } \underline{\theta} \neq 0 \text{ (stop engaged);} \\ \text{or } T_f < 0 \text{ (locked);} \\ \text{or } K_4\theta_K < T_f \text{ (not sliding)} \end{cases} \\ 0, & \text{if } -\underline{\theta} < \theta_f < \overline{\theta} \text{ and } K_4\theta_K \geq T_f \geq 0 \text{ (sliding)} \end{cases}$$

The mass matrix is lumped and given by:

$$[M] = \frac{1}{2}\begin{bmatrix} M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & M & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & I_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & I_m & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & I_m & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & & M & 0 & 0 & 0 & 0 & 0 \\ & & \text{Symmetry} & & & & & M & 0 & 0 & 0 & 0 \\ & & & & & & & & M & 0 & 0 & 0 \\ & & & & & & & & & I_m & 0 & 0 \\ & & & & & & & & & & I_m & 0 \\ & & & & & & & & & & & I_m \end{bmatrix} \qquad (14.7\text{–}8)$$

where: M = total mass (input as MASS on **RMORE** command)

I_m = total mass moment of inertia (input as IMASS on **RMORE** command)

The damping matrix, derived from rotational viscous damping about the pin axis, is given as:

$$[C] = C_t \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ & & & & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & \text{Symmetry} & & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & & 0 & 0 & 0 & 0 \\ & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & 0 & 0 \\ & & & & & & & & & & & 1 \end{bmatrix} \qquad (14.7\text{–}9)$$

The applied load vector for COMBIN7 is given by:

$$\{F\} = \lfloor 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; -\left(T_i + K_4\theta_i\right) \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; \left(T_i + K_4\theta_i\right) \rfloor^T \qquad (14.7\text{–}10)$$

Modification of Real Constants

Four real constants (C_1, C_2, C_3, C_4) are used to modify other real constants for a dynamic analysis (**ANTYPE**,TRAN with **TRNOPT**,FULL). The modification is performed only if either $C_1 \neq 0$ or $C_3 \neq 0$ and takes the form:

$$R' = R + M \qquad (14.7\text{–}11)$$

where:

R' = modified real constant value

R = original real constant value

$$M = \begin{cases} C_1 \; |C_v|^{C2} + C_3|C_v|^{C4} \text{ if KEYOPT(9)} = 0 \\ f_1 \left(C_1, \; C_2, \; C_3, \; C_4, \; C_v\right) \text{ if KEYOPT(9)} = 1 \end{cases}$$

C_1, C_2, C_3, C_4 = input as C1, C2, C3 and C4 on **RMORE** command

C_v = control value (defined below)

f_1 = function defined by subroutine USERRC

By means of KEYOPT(7), the quantity R is as follows:

$$R = \begin{cases} K_1 & \text{if KEYOPT(7) = 0 to 1} \\ K_2 & \text{if KEYOPT(7) = 2} \\ K_3 & \text{if KEYOPT(7) = 3} \\ \vdots & \\ \text{ROT} & \text{if KEYOPT(7) = 13} \end{cases} \tag{14.7–12}$$

Negative values for R′ are set equal to zero for quantities T_f (KEYOPT(7)=6), $\underline{\theta}$ (KEYOPT(7)=11), and $\overline{\theta}$ (KEYOPT(7)=12).

The calculation for C_v depends on control nodes L and M, as well as KEYOPT(1), KEYOPT(3), and KEYOPT(4). The general formulation is given by:

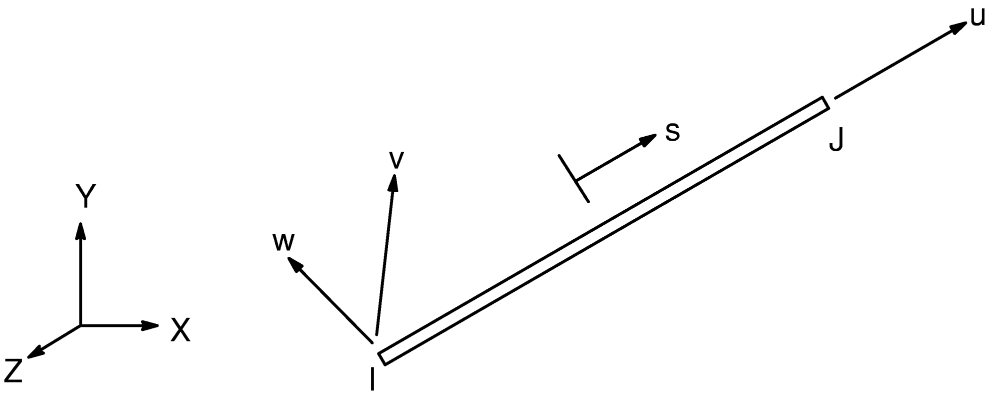
$$C_v = \begin{cases} \Delta u, & \text{if KEYOPT(1) = 1 or 0} \\ \dfrac{d(\Delta u)}{dt}, & \text{if KEYOPT(1) = 2} \\ \dfrac{d^2(\Delta u)}{dt^2}, & \text{if KEYOPT(1) = 3} \\ \displaystyle\int_o^t \Delta u \; dt, & \text{if KEYOPT(1) = 1 or 0} \\ t, & \text{if KEYOPT(1) = 1 or 0} \end{cases} \tag{14.7–13}$$

in which t is time and Δu is determined from

$$\Delta u = \begin{cases} u_L - u_M, & \text{if KEYOPT(3) = 0, 1} \\ v_L - v_M, & \text{if KEYOPT(3) = 2} \\ w_L - w_M, & \text{if KEYOPT(3) = 3} \\ \theta_{xL} - \theta_{xM}, & \text{if KEYOPT(3) = 4} \\ \theta_{yL} - \theta_{yM}, & \text{if KEYOPT(3) = 5} \\ \theta_{zL} - \theta_{zM}, & \text{if KEYOPT(3) = 6} \end{cases} \tag{14.7–14}$$

If KEYOPT(4) = 0, then the DOFs above are in nodal coordinates. The DOFs are in the moving element coordinates if KEYOPT(4) = 1.

14.8 LINK8 — 3–D Spar (or Truss)



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.1–1)	None
Mass Matrix	Equations (12.2.1–1), (12.2.1–2), and (12.2.1–3)	None
Stress Stiffening Matrix	Equations (12.2.1–2) and (12.2.1–3)	None
Thermal Load Vector	Equation (12.2.1–1)	None

Load Type	Distribution
Element Temperature	Linear along length
Nodal Temperature	Linear along length

Reference: Cook et al(117)

14.8.1 Assumptions and Restrictions

The element is not capable of carrying bending loads. The stress is assumed to be uniform over the entire element.

14.8.2 Element Matrices and Load Vector

All element matrices and load vectors described below are generated in the element coordinate system and are then converted to the global coordinate system. The element stiffness matrix is:

$$\left[K_\ell\right] = \frac{A\hat{E}}{L}\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \qquad (14.8\text{–}1)$$

where: A = element cross–sectional area (input as AREA on **R** command)

$\hat{E}$ = E, Young’s modulus (input as EX on **MP** command) if linear.
E_T, tangent modulus (see Section 4.1) if plasticity is present and the tangent matrix is to be computed (see Section 4.1 and 4.4).

L = element length

The element mass matrix with **LUMPM**,OFF is:

$$\left[M_\ell\right] = \frac{\rho AL(1 - \epsilon^{in})}{6}\left[\begin{array}{ccc|ccc} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{array}\right] \qquad (14.8\text{–}2)$$

where: ϱ = density (input as DENS on **MP** command)

ε^{in} = initial strain (input as ISTRN on **R** command)

The element mass matrix with **LUMP**,ON is:

$$[M_\ell] = \frac{\rho AL(1 - \epsilon^{in})}{2} \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \qquad (14.8–3)$$

The element stress stiffness matrix is:

$$[S_\ell] = \frac{F}{L} \left[\begin{array}{ccc:ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{array}\right] \qquad (14.8–4)$$

where: $F = \begin{cases} \text{for the first iteration: } A\,E\,\varepsilon^{in} \\ \text{for all subsequent iterations: the axial force in the element as computed in the previous stress pass of the element} \end{cases}$

The element load vector is:

$$\{F_\ell\} = \{F_\ell^a\} - \{F_\ell^{nr}\} \qquad (14.8–5)$$

where: $\{F_\ell^a\}$ = the applied load vector

$\{F_\ell^{nr}\}$ = the Newton–Raphson restoring force, if applicable.

The applied load vector is:

$$\{F_\ell^a\} = AE\epsilon_n^T \; \lfloor -1\ 0\ 0\ 1\ 0\ 0 \rfloor^T \qquad (14.8–6)$$

For a linear analysis or the first iteration of a nonlinear (Newton–Raphson) analysis ϵ^{T}_{n} is:

$$\epsilon^{T}_{n} = \epsilon^{th}_{n} - \epsilon^{in} \qquad (14.8\text{–}7)$$

with

$$\epsilon^{th}_{n} = \alpha_{n} \left(T_{n} - T_{ref}\right)$$

where:

α_n = coefficient of thermal expansion (input as ALPX on **MP** command) evaluated at T_n

T_n = average temperature of the element in this iteration

T_{ref} = reference temperature (input on **TREF** command)

For the subsequent iterations of a Newton–Raphson analysis:

$$\epsilon^{T}_{n} = \Delta\epsilon^{th}_{n} \qquad (14.8\text{–}8)$$

with the thermal strain increment computed through:

$$\Delta\epsilon^{th}_{n} = \alpha_{n} \left(T_{n} - T_{ref}\right) - \alpha_{n-1} \left(T_{n-1} - T_{ref}\right)$$

where:

α_n, α_{n-1} = coefficients of thermal expansion evaluated at T_n and T_{n-1}, respectively

T_n, T_{n-1} = average temperature of the element for this iteration and the previous iteration

The Newton–Raphson restoring force vector is:

$$\left\{F^{nr}_{\ell}\right\} = AE\epsilon^{el}_{n-1} \; \lfloor -1 \; 0 \; 0 \; 1 \; 0 \; 0 \; \rfloor^{T} \qquad (14.8\text{–}9)$$

where:

ϵ^{el}_{n-1} = the elastic strain for the previous iteration.

14.8.3 Force and Stress

For a linear analysis or the first iteration of a nonlinear (Newton–Raphson) analysis:

$$\epsilon^{el}_{n} = \epsilon_{n} - \epsilon^{th}_{n} + \epsilon^{in} \qquad (14.8\text{–}10)$$

where:

ϵ^{el}_{n} = elastic strain (output quantity EPELAXL)

ϵ_n = total strain = $\frac{u}{L}$

u = difference of nodal displacements in axial direction

ϵ_n^{th} = thermal strain (output quantity EPTHAXL)

For the subsequent iterations of a nonlinear (Newton–Raphson) analysis:

$$\epsilon_n^{el} = \epsilon_{n-1}^{el} + \Delta\epsilon - \Delta\epsilon^{th} - \Delta\epsilon^{pl} - \Delta\epsilon^{cr} - \Delta\epsilon^{sw} \tag{14.8–11}$$

where: $\Delta\varepsilon$ = strain increment = $\frac{\Delta u}{L}$

Δu = difference of nodal displacements increment in axial direction

$\Delta\varepsilon^{th}$ = thermal strain increment

$\Delta\varepsilon^{pl}$ = plastic strain increment

$\Delta\varepsilon^{cr}$ = creep strain increment

$\Delta\varepsilon^{sw}$ = swelling strain increment

The stress is:

$$\sigma = E\,\epsilon^{a} \tag{14.8–12}$$

where: σ = stress (output quantity SAXL)

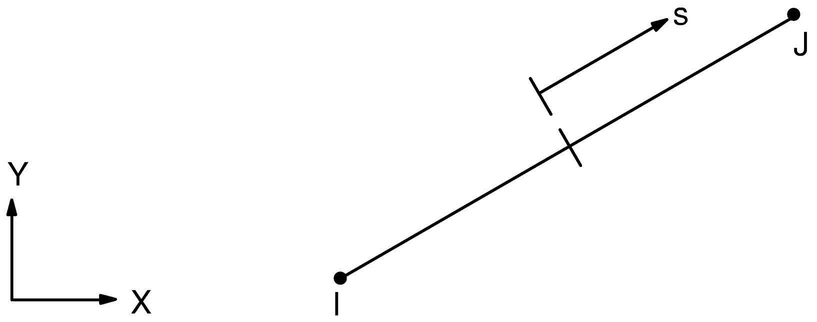
ε^{a} = adjusted strain = $\epsilon_n^{el} + \Delta\varepsilon^{cr} + \Delta\varepsilon^{sw}$

Thus, the strain used to compute the stress has the creep and swelling effects as of the beginning of the substep, not the end. Finally,

$$F = A\sigma \tag{14.8–13}$$

where: F = force (output quantity MFORX)

14.9 INFIN9 — 2–D Infinite Boundary



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix or Thermal Conductivity Matrix	$A = C_1 + C_2x$	None

References: Kagawa, Yamabuchi and Kitagami(122)

14.9.1 Introduction

This boundary element (BE) models the exterior infinite domain of the far–field magnetic and thermal problems. This element is to be used in combination with elements having a magnetic potential (AZ) or temperature (TEMP) as the DOF.

14.9.2 Theory

The formulation of this element is based on a first order infinite boundary element (IBE) that is compatible with first order quadrilateral or triangular shaped finite elements, or higher order elements with dropped midside nodes. For unbounded field problems., the model domain is set up to consist of an interior finite element domain, Ω_F, and a series of exterior BE subdomains, Ω_B, as shown in Figure 14.9–1. Each subdomain, Ω_B, is treated as an ordinary BE domain consisting of four segments: the boundary element I–J, infinite elements J–K and I–L, and element K–L; element K–L is assumed to be located at infinity.

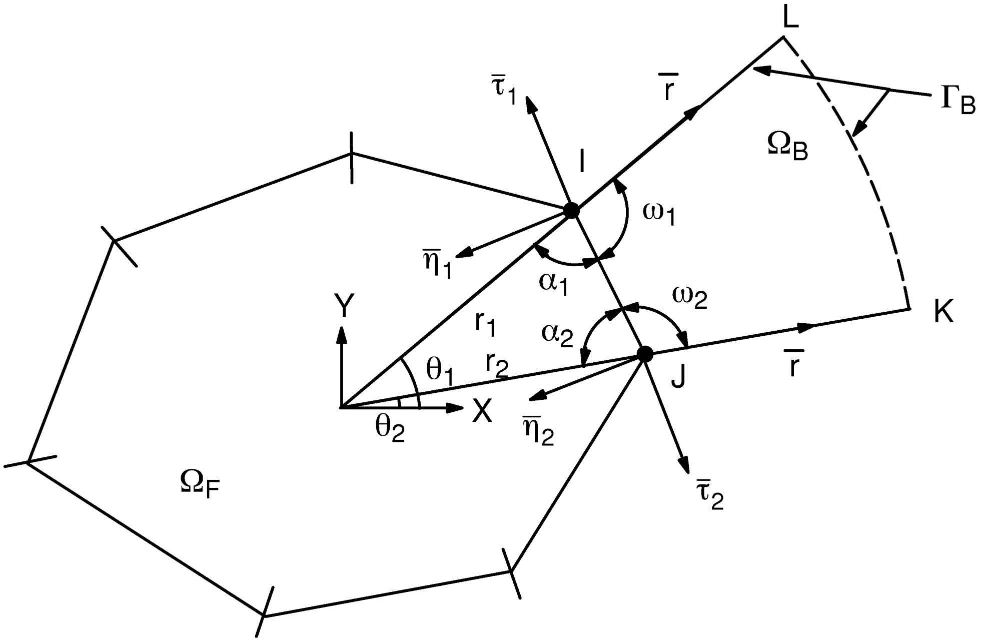


Figure 14.9–1 Definition of BE Subdomain and the Characteristics of the IBE

The approach used here is to write BE equations for Ω_B, and then convert them into equivalent load vectors for the nodes I and J. The procedure consists of four separate steps that are summarized below (see reference (122) for details).

First, a set of boundary integral equations is written for Ω_B. To achieve this, linear shape functions are used for the BE I–J:

$$N_1(s) = \frac{1}{2}\,(1 - s) \qquad (14.9\text{–}1)$$

$$N_2(s) = \frac{1}{2}\,(1 + s) \qquad (14.9\text{–}2)$$

Over the infinite elements J–K and I–L the potential (or temperature) ϕ and its derivative q (flux) are respectively assumed to be:

$$\phi(r) = \phi_i \left(\frac{r_i}{r}\right) , \; i = I, J \qquad (14.9\text{–}3)$$

$$q(r) = q_i \left(\frac{r_i}{r}\right)^2 , \; i = I, J \qquad (14.9\text{–}4)$$

The boundary integral equations are the same as presented in equation (14.47–5) except that the Green’s function in this case would be:

$$G(x,\xi) = \frac{1}{2\pi k}\ \ln\left(\frac{\sqrt{k}}{r}\right) \qquad (14.9\text{–}5)$$

where:

x = field point in boundary element

ξ = source point

k = magnetic reluctivity (inverse of free space permeability input on **EMUNIT** command) for AZ DOF (KEYOPT(1)=0)

or

thermal conductivity (input as KXX on **MP** command) for TEMP DOF (KEYOPT(1)=1)

Note that all the integrations in the present case are performed in closed form.

Second, in the absence of a source or sink in Ω_B, the flux q(r) is integrated over the boundary Γ_B of Ω_B and set to zero.

$$\int_{\Gamma_B} q\ d\Gamma = 0 \qquad (14.9\text{–}6)$$

Third, a geometric constraint condition that exists between the potential ϕ and its derivatives $\frac{\partial\phi}{\partial n} = q_n$ and $\frac{\partial\phi}{\partial\tau} = q_\tau$ at the nodes I and J is written as:

$$q_{n_i} = q_{\tau_i}\cos\alpha_i + \phi_i\,\frac{\sin\alpha_i}{r_i} \quad i = I, J \qquad (14.9\text{–}7)$$

Fourth, the energy flow quantity from Ω_B is written as:

$$w = \int_{\Gamma_B} q\ \phi\ d\Gamma \qquad (14.9\text{–}8)$$

This energy flow is equated to that due to an equivalent nodal {F} defined below.

The four steps mentioned above are combined together to yield, after eliminating q_n and q_τ,

$$[K]\{\phi\} = \{F\} \qquad (14.9\text{–}9)$$

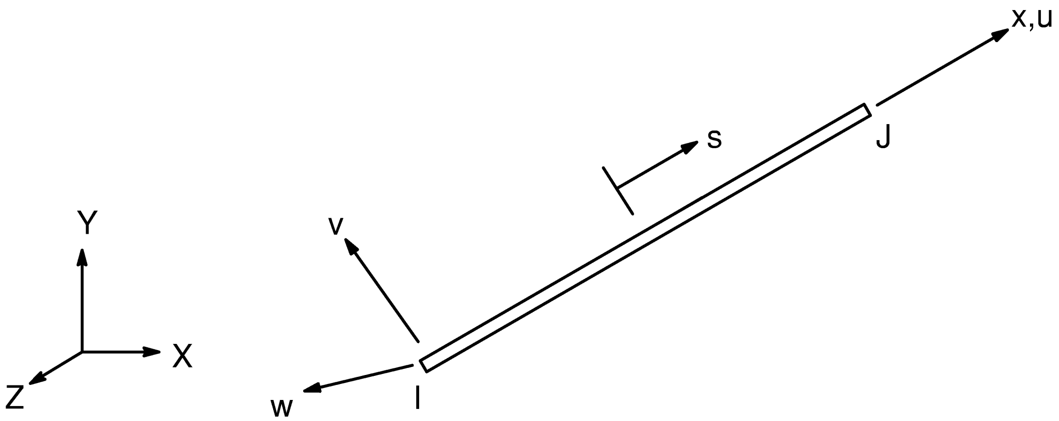
where: [K] = 2 x 2 equivalent unsymmetric element coefficient matrix

{ϕ} = 2 x 1 nodal DOFs, AZ or TEMP

{F} = 2 x 1 equivalent nodal force vector

For linear problems, the INFIN9 element forms the coefficient matrix [K] only. The load vector {F} is not formed. The coefficient matrix multiplied by the nodal DOF's represents the nodal load vector which brings the effects of the semi–infinite domain Ω_B onto nodes I and J.

14.10 LINK10 — Tension Only or Compression–only Spar



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.1–1)	None
Mass Matrix	Equations (12.2.1–1), (12.2.1–2), and (12.2.1–3)	None
Stress Stiffness Matrix	Equations (12.2.1–2) and (12.2.1–3)	None
Thermal Load	Equation (12.2.1–1)	None

Load Type	Distribution
Element Temperature	Linear along length
Nodal Temperature	Linear along length

14.10.1 Assumptions and Restrictions

The element is not capable of carrying bending loads. The stress is assumed to be uniform over the entire element.

14.10.2 Element Matrices and Load Vector

All element matrices and load vectors are generated in the element coordinate system and must subsequently then be converted to the global coordinate system. The element stiffness matrix is:

$$[K_\ell] = \frac{AE}{L}\begin{bmatrix} C_1 & 0 & 0 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -C_1 & 0 & 0 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (14.10\text{–}1)$$

where:
A = element cross–sectional area (input as AREA on **R** command)
E = Young's modulus (input as EX on **MP** command)
L = element length
C_1 = value given in Table 14.10–1

Table 14.10–1 Value of Stiffness Coefficient (C_1)

User Options	Strain is Currently Tensile	Strain is Currently Compressive
KEYOPT(2) = 0 KEYOPT(3) = 0	1.0	0.0
KEYOPT(2) > 0 KEYOPT(3) = 0	1.0	1.0×10^{-6}
KEYOPT(2) = 0 KEYOPT(3) = 1	0.0	1.0
KEYOPT(2) > 0 KEYOPT(3) = 1	1.0×10^{-6}	1.0

Meanings:

KEYOPT(2) = 0	No extra stiffness for non–load carrying case
KEYOPT(2) = 1,2	Has small stiffness for non–load carrying case
KEYOPT(3) = 0	Tension–only spar
KEYOPT(3) = 1	Compression–only spar

The element mass matrix is the same as for LINK8 (Section 14.8):

The element stress stiffness matrix is:

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_2 & 0 & 0 & -C_2 & 0 \\ 0 & 0 & C_2 & 0 & 0 & -C_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -C_2 & 0 & 0 & C_2 & 0 \\ 0 & 0 & -C_2 & 0 & 0 & C_2 \end{bmatrix} \qquad (14.10\text{–}2)$$

where: $F = \begin{cases} \text{for the first iteration: } A\,E\,\varepsilon^{in} \\ \text{for all subsequent iterations: the axial force} \\ \text{in the element (output quantity FORC)} \end{cases}$

C_2 = value given in Table 14.10–2.

Table 14.10–2 Value of Stress Stiffness Coefficient (C_2)

User Options	Strain is Currently Tensile	Strain is Currently Compressive
KEYOPT(2) < 2 KEYOPT(3) = 0	1.0	0.0
KEYOPT(2) = 2 KEYOPT(3) = 0	1.0	$\frac{AE}{F\ 10^6}$
KEYOPT(2) < 2 KEYOPT(3) = 1	0.0	1.0
KEYOPT(2) = 2 KEYOPT(3) = 1	$\frac{AE}{F\ 10^6}$	1.0

Meanings:

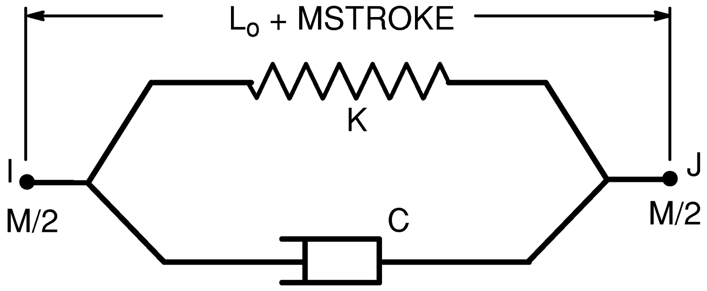
KEYOPT(2) = 0,1	No extra stress stiffness value
KEYOPT(2) = 2	Include extra stress stiffness value
KEYOPT(3) = 0	Tension–only spar
KEYOPT(3) = 1	Compression–only spar

The element applied load vector is:

$$\{F_\ell\} = AE\epsilon^T \lfloor -C_1 \quad 0 \quad 0 \quad C_1 \quad 0 \quad 0 \rfloor^T \tag{14.10–3}$$

where:
ε^T = $\alpha\Delta T - \varepsilon^{in}$
α = coefficient of thermal expansion (input as ALPX on **MP** command)
ΔT = $T_{ave} - T_{REF}$
T_{ave} = average temperature of element
T_{REF} = reference temperature (input on **TREF** command)
ε^{in} = prestrain (input as ISTRN on **R** command)

14.11 LINK11 — Linear Actuator



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.1–1)	None
Mass Matrix	None (lumped mass formulation)	None
Stress Stiffness Matrix	Equation (12.2.1–2) and (12.2.1–3)	None
Damping Matrix	Same as stiffness matrix	None

14.11.1 Assumptions and Restrictions

The element is not capable of carrying bending or twist loads. The force is assumed to be constant over the entire element.

14.11.2 Element Matrices and Load Vector

All element matrices and load vectors are described below. They are generated in the element coordinate system and are then converted to the global coordinate system. The element stiffness matrix is:

$$[K_\ell] = K \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \tag{14.11–1}$$

where: K = element stiffness (input as K on **R** command)

The element mass matrix is:

$$[M_\ell] = \frac{M}{2} \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \tag{14.11–2}$$

where: M = total element mass (input as M on **R** command)

The element damping matrix is:

$$[C_\ell] = C \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hdashline -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right] \tag{14.11–3}$$

where: C = element damping (input as C on **R** command)

The element stress stiffness matrix is:

$$[S_\ell] = \frac{F}{L}\left[\begin{array}{ccc:ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{array}\right] \tag{14.11–4}$$

where: F = the axial force in the element (output quantity FORCE)

L = current element length (output quantity CLENG)

The element load vector is:

$$\{F_\ell\} = \{F_\ell^{ap}\} - \{F_\ell^{nr}\} \tag{14.11–5}$$

where: $\{F_\ell^{ap}\}$ = applied force vector

$\{F_\ell^{nr}\}$ = Newton–Raphson restoring force

The applied force vector is:

$$\{F_\ell^{ap}\} = F' \lfloor -1\ 0\ 0\ 1\ 0\ 0 \rfloor^T \tag{14.11–6}$$

where: F′ = applied force thru surface load input using the PRES label

The Newton–Raphson restoring force vector is:

$$\{F_\ell^{nr}\} = F \lfloor -1\ 0\ 0\ 1\ 0\ 0 \rfloor^T \tag{14.11–7}$$

14.11.3 Force, Stroke, and Length

The element spring force is determined from

$$F = K\left(S_M - S_A\right) \tag{14.11–8}$$

where: F = element spring force (output quantity FORCE)

S_A = applied stroke (output quantity STROKE) thru surface load input using the PRES label

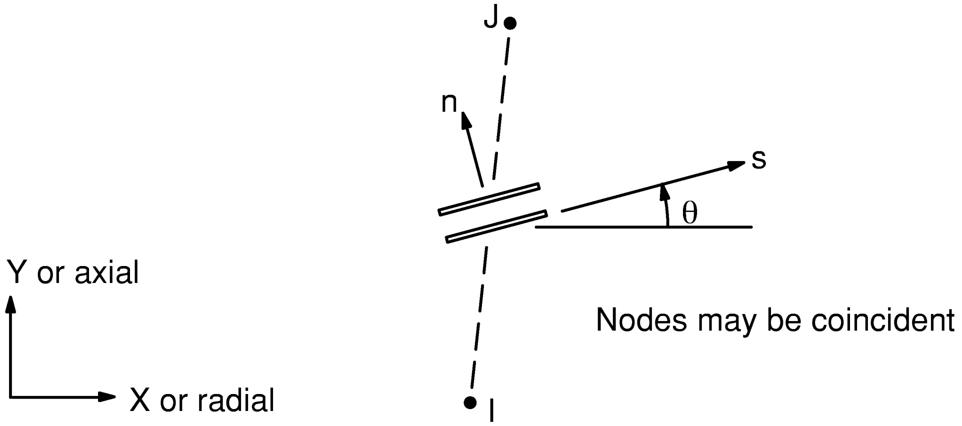
S_M = computed or measured stroke (output quantity MSTROKE)

The lengths, shown in the figure at the beginning of this section, are:

L_o = initial length (output quantity ILEN)

$L_o + S_M$ = current length (output quantity CLEN)

14.12 CONTAC12 — 2–D Point–to–Point Contact



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (nodes may be coincident)	None

Load Type	Distribution
Element Temperature	None – average used for material property evaluation
Nodal Temperature	None – average used for material property evaluation

14.12.1 Element Matrices

CONTAC12 may have one of three conditions if the elastic Coulomb friction option (KEYOPT(1) = 0) is used: closed and stuck, closed and sliding, or open. The following matrices are derived assuming that θ is input as 0.0.

1. Closed and stuck. This occurs if:

$$\mu \left|F_n\right| > \left|F_s\right| \tag{14.12–1}$$

where:
μ = coefficient of friction (input as MU on **MP** command)
F_n = normal force across gap
F_s = sliding force parallel to gap

The normal force is:

$$F_n = k_n \left(u_{n,J} - u_{n,I} - \Delta \right) \tag{14.12–2}$$

where:
k_n = normal stiffness (input as KN on **R** command
$u_{n,I}$ = displacement of node I in normal direction
$u_{n,J}$ = displacement of node J in normal direction
Δ = interference $\begin{cases} \text{input as INTF on R command} \\ \text{if KEYOPT(4) = 0} \\ = -\text{ d if KEYOPT(4) = 1} \end{cases}$
d = distance between nodes

The sliding force is:

$$F_s = k_s \left(u_{s,J} - u_{s,I} - u_o \right) \tag{14.12–3}$$

where:
k_s = sticking stiffness (input as KS on **R** command)
$u_{s,I}$ = displacement of node I in sliding direction
$u_{s,J}$ = displacement of node J in sliding direction
u_o = distance that nodes I and J have slid with respect to each other

The resulting element stiffness matrix (in element coordinates) is:

$$\left[K_\ell \right] = \begin{bmatrix} k_s & 0 & -k_s & 0 \\ 0 & k_n & 0 & -k_n \\ -k_s & 0 & k_s & 0 \\ 0 & -k_n & 0 & k_n \end{bmatrix} \tag{14.12–4}$$

and the Newton–Raphson load vector (in element coordinates) is:

$$\left\{ F_\ell^{nr} \right\} = \begin{Bmatrix} F_s \\ F_n \\ -F_s \\ -F_n \end{Bmatrix} \tag{14.12–5}$$

2. Closed and sliding. This occurs if:

$$\mu \, |F_n| = |F_s| \tag{14.12–6}$$

In this case, the element stiffness matrix (in element coordinates) is:

$$[K_\ell] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_n & 0 & -k_n \\ 0 & 0 & 0 & 0 \\ 0 & -k_n & 0 & k_n \end{bmatrix} \tag{14.12–7}$$

and the Newton–Raphson load vector is the same as in equation (14.12–5). If the unsymmetric option is chosen (KEYOPT(11) = 1), then the stiffness matrix includes the coupling between the normal and sliding directions; which for STAT = 2 is:

$$[K_\ell] = \begin{bmatrix} 0 & -\mu k_n & 0 & \mu k_n \\ 0 & k_n & 0 & -k_n \\ 0 & \mu k_n & 0 & -\mu k_n \\ 0 & -k_n & 0 & k_n \end{bmatrix} \tag{14.12–8}$$

3. Open — When there is no contact between nodes I and J. There is no stiffness matrix or load vector.

   Figure 14.12–1 shows the force–deflection relationships for this element. It may be seen in these figures that the element is nonlinear and therefore needs to be solved iteratively. Further, since energy lost in the slider cannot be recovered, the load needs to be applied gradually.

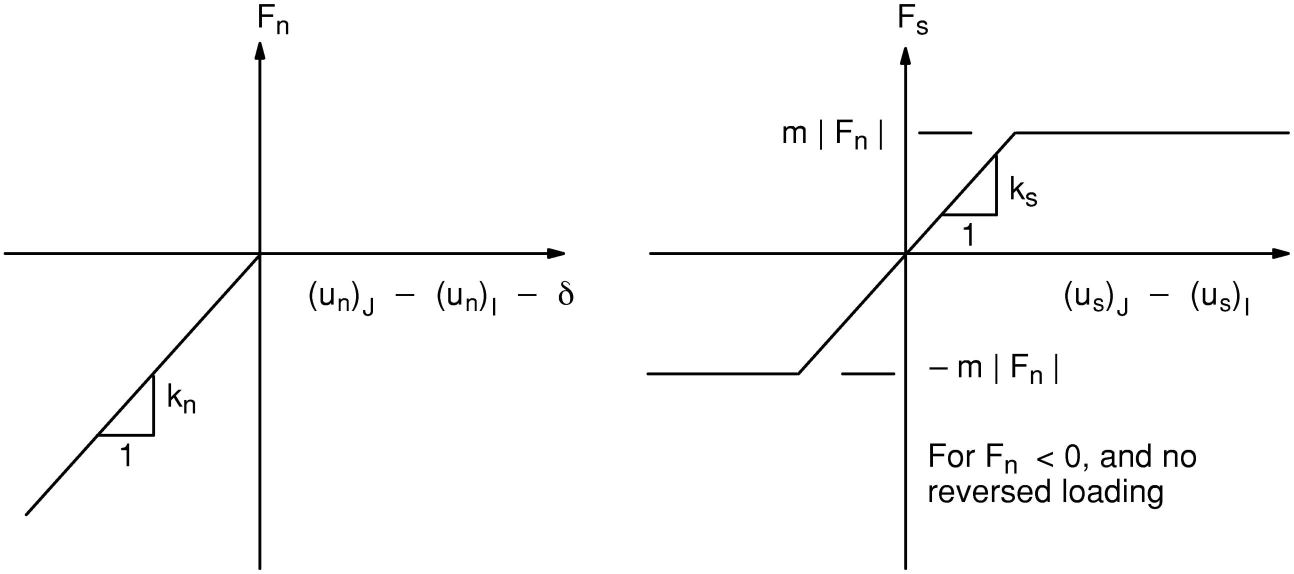


Figure 14.12–1 Force–Deflection Relations for Standard Case

14.12.2 Orientation of the Element

The element is normally oriented based on θ (input as THETA on **R** command). If KEYOPT(2) = 1, however, θ is not used. Rather, the first iteration has θ equal to zero, and all subsequent iterations have the orientation of the element based on the displacements of the previous iteration. In no case does the element use its nodal coordinates.

14.12.3 Rigid Coulomb Friction

If the user knows that a gap element will be in sliding status for the life of the problem, and that the relative displacement of the two nodes will be monotonically increasing, the rigid Coulomb friction option (KEYOPT(1) = 1) can be used to avoid convergence problems. This option removes the stiffness in the sliding direction, as shown in Figure 14.12–2. It should be noted that if the relative displacement does not increase monotonically, the convergence characteristics of KEYOPT(1) = 1 will be worse than for KEYOPT(1) = 0.

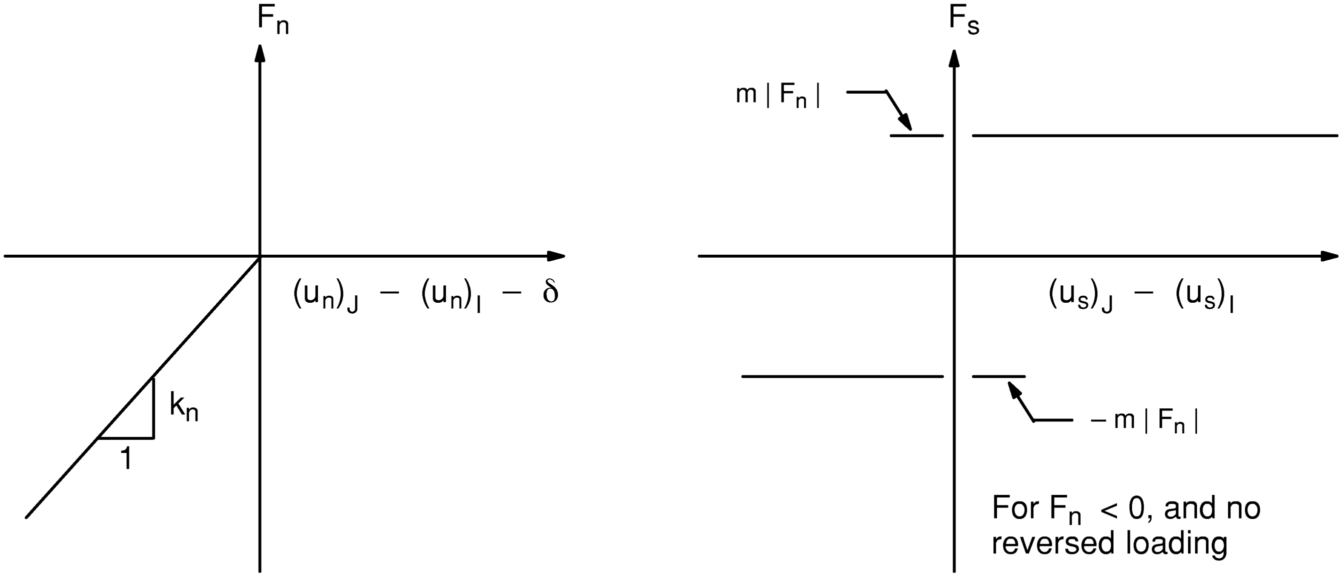
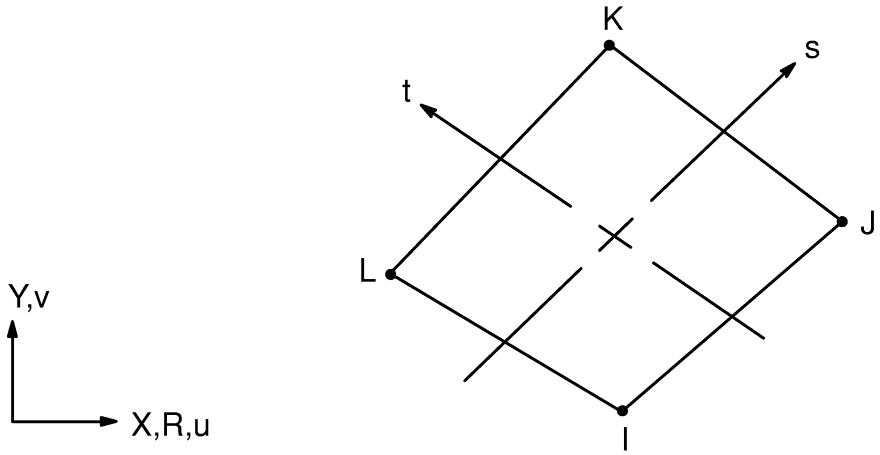


Figure 14.12–2 Force–Deflection Relations for Rigid Coulomb Option

14.13   PLANE13 — 2–D Coupled–Field Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Quad	Equation (12.6.5–9)	2 x 2
	Triangle	Equation (12.6.1–9)	1 if planar, and 3 if axisymmetric
Thermal Conductivity Matrix	Quad	Equation (12.6.5–20)	Same as coefficient matrix
	Triangle	Equation (12.6.1–20)	
Stiffness Matrix	Quad	Equations (12.6.5–1) and (12.6.5–2) and, if modified extra shapes are included (KEYOPT(2)=0) and element has 4 unique nodes) equations (12.6.6–1) and (12.6.6–2).	Same as coefficient matrix
	Triangle	Equations (12.6.1–1) and (12.6.1–2)	

Matrix or Vector	Geometry	Shape Functions	Integration Points
Mass Matrix	Quad	Equations (12.6.5–1) and (12.6.5–2)	Same as coefficient matrix
	Triangle	Equations (12.6.1–1) and (12.6.1–2)	Same as coefficient matrix
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Section 13.2		Same as coefficient matrix
Damping (Eddy Current) Matrix	Quad	Equations (12.6.5–9) and (12.6.5–21)	Same as coefficient matrix
	Triangle	Equations (12.6.1–9) and (12.6.1–21)	Same as coefficient matrix
Stress Stiffness Matrix	Same as mass matrix.		Same as coefficient matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix		2
Permanent Magnet and Applied Current Load Vector	Same as coefficient matrix		Same as coefficient matrix
Thermal and Magnetic Force Load Vector	Same as stiffness matrix		Same as coefficient matrix
Pressure Load Vector	Same as mass matrix specialized to the face		2

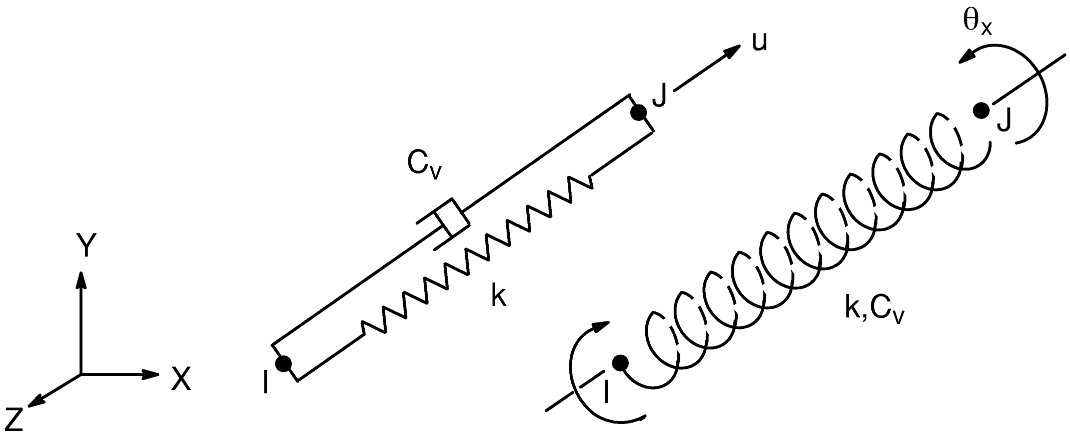
Load Type	Distribution
Current Density	Bilinear across element
Current Phase Angle	Bilinear across element
Heat Generation	Bilinear across element
Pressure	Linear along each face

References: Wilson(38), Taylor, et al (49), Silvester, et al (72),Weiss, et al (94), Garg, et al (95)

14.13.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Section 5.2 and 5.3 discuss the magnetic vector potential method, which is used by this element. Section 13.1 describes integration point locations. The diagonalization of the specific heat matrix is described in Section 13.2. Section 14.42 provides additional information on the element coordinate system, extra displacement shapes, and stress calculations.

14.14 COMBIN14 — Spring–Damper



Matrix Or Vector	Option	Shape* Functions	Integration Points
Stiffness Matrix	Longitudinal	Equation (12.2.1–1)	None
	Torsional	Equation (12.2.2–4)	None
Damping Matrix	Longitudinal	Equation (12.2.1–1)	None
	Torsional	Equation (12.2.2–4)	None
Stress Stiffening Matrix	Longitudinal	Equations (12.2.1–2) and (12.2.1–3)	None

* There are no shape functions used if the element is input on a one DOF per node basis (KEYOPT(2) > 0) as the nodes may be coincident.

14.14.1 Types of Input

COMBIN14 essentially offers two types of elements, selected with KEYOPT(2).

1. Single DOF per node (KEYOPT(2) > 0). The orientation is defined with the **KEYOPT** command and the two nodes are usually coincident.
2. Multiple DOFs per node (KEYOPT(2) = 0). The orientation is defined by the location of the two nodes; therefore, the two nodes must not be coincident.

14.14.2 Stiffness Pass

Consider the case of a single DOF per node first. The orientation is selected with KEYOPT(2). If KEYOPT(2) = 7 (pressure) or = 8 (temperature), the concept of orientation does not apply. The form of the element stiffness and damping matrices are:

$$[K_e] = k \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (14.14\text{–}1)$$

$$[C_e] = C_v \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (14.14\text{–}2)$$

where: k = input as K on **R** command

C_v = $C_{v1} + C_{v2}\,|v|$

C_{v1} = constant damping coefficient (input as CV1 on **R** command)

C_{v2} = linear damping coefficient (input as CV2 on **R** command)

v = relative velocity between nodes computed from the nodal Newmark velocities

Next, consider the case of multiple DOFs per node. Only the case with three DOFs per node will be discussed, as the case with two DOFs per node is simply a subset. The stiffness, damping, and stress stiffness matrices in element coordinates are developed as:

$$[K_\ell] = k \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (14.14\text{–}3)$$

$$[C_\ell] = C_v \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (14.14\text{–}4)$$

$$[S_\ell] = \frac{F}{L}\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{14.14–5}$$

where subscript ℓ refers to element coordinates.

and where: F = force in element from previous iteration

L = distance between the two nodes

There are some special notes that apply to the torsion case (KEYOPT(3) = 1):

1. Rotations are simply treated as a vector quantity. No other effects (including displacements) are implied.
2. In a large rotation problem (**NLGEOM**,ON), the coordinates do not get updated, as the nodes only rotate. (They may translate on other elements, but this does not affect COMBIN14 with KEYOPT(3) = 1). Therefore, there are no large rotation effects.
3. Similarly, as there is no axial force computed, no stress stiffness matrix is computed.

14.14.3 Output Quantities

The stretch is computed as:

$$\epsilon_o = \begin{cases} \frac{A}{L} & \text{if KEYOPT(2) = 0} \\ u'_J - u'_I & \text{if KEYOPT(2) = 1} \\ v'_J - v'_I & \text{if KEYOPT(2) = 2} \\ w'_J - w'_I & \text{if KEYOPT(2) = 3} \\ \theta'_{xJ} - \theta'_{xI} & \text{if KEYOPT(2) = 4} \\ \theta'_{yJ} - \theta'_{yI} & \text{if KEYOPT(2) = 5} \\ \theta'_{zJ} - \theta'_{zI} & \text{if KEYOPT(2) = 6} \\ P_J - P_I & \text{if KEYOPT(2) = 7} \\ T_J - T_I & \text{if KEYOPT(2) = 8} \end{cases} \tag{14.14–6}$$

where: ε_o = output quantity STRETCH

A = $(X_J - X_I)(u_J - u_I) + (Y_J - Y_I)(v_J - v_I) + (Z_J - Z_I)(w_J - w_I)$

X, Y, Z = coordinates in global Cartesian coordinates

u, v, w = displacements in global Cartesian coordinates

u', v', w' = displacements in nodal Cartesian coordinates (UX, UY, UZ)

$\theta_x', \ \theta_y', \ \theta_z'$ = rotations in nodal Cartesian coordinates (ROTX, ROTY, ROTZ)

P = pressure (PRES)

T = temperatures (TEMP)

If KEYOPT(3) = 1 (torsion), the expression for A has rotation instead of translations, and ε_o = output quantity TWIST. Next, the static force (or torque) is computed:

$$F_s = K \epsilon_o \tag{14.14–7}$$

where: F_s = output quantity FORC (TORQ if KEYOPT(3) = 1)

K = input quantity K

Finally, if a nonlinear transient dynamic (**ANTYPE**,TRANS, with **TIMINT**,ON) analysis is performed, a damping force is computed:

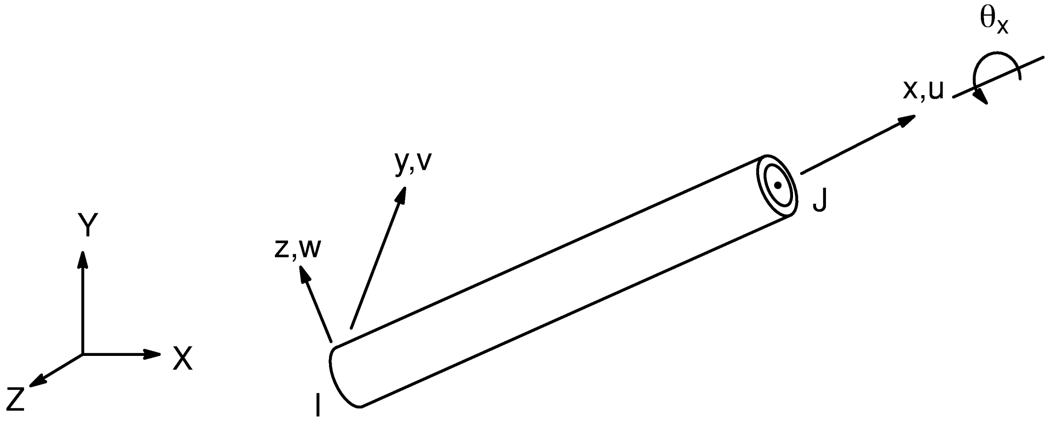
$$F_D = C_v V \tag{14.14–8}$$

where: F_D = output quantity DAMPING FORCE (DAMPING TORQUE if KEYOPT(3) = 1)

V = relative velocity

The relative velocity is computed using equation (14.14–6), where the nodal displacements u, v, w, etc. are replaced with the nodal Newmark velocities $\dot{u}, \ \dot{v}, \ \dot{w}$, etc.

14.16 PIPE16 — Elastic Straight Pipe



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.2–1), (12.2.2–2), (12.2.2–3), and (12.2.2–4)	None
Mass Matrix	Same as stiffness matrix	None
Stress Stiffness Matrix	Equation (12.2.2–2) and (12.2.2–3)	None
Load Vector (Pressure and Thermal)	Equation (12.2.2–1), (12.2.2–2), and (12.2.2–3)	None

Load Type	Distribution
Element Temperature	Linear thru thickness or across diameter, and along length
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Internal and External: constant along length and around circumference. Lateral: constant along length

14.16.1 Other Applicable Sections

The basic form of the element matrices is given with the 3–D beam element, BEAM4 (Section 14.4).

14.16.2 Assumptions and Restrictions

The element is assumed to be a thin–walled pipe except as noted. The corrosion allowance is used only in the stress evaluation, not in the matrix formulation.

14.16.3 Stiffness Matrix

The element stiffness matrix of PIPE16 is the same as for BEAM4, except that

$$A = A^w = \frac{\pi}{4}\left(D_o^2 - D_i^2\right) = \text{pipe wall cross–sectional area} \qquad (14.16–1)$$

$$I_y = I_z = I = \frac{\pi}{64}\left(D_o^4 - D_i^4\right)\frac{1}{C_f} = \text{bending moment of inertia} \qquad (14.16–2)$$

$$J = \frac{\pi}{32}\left(D_o^4 - D_i^4\right) = \text{torsional moment of inertia} \qquad (14.16–3)$$

and,

$$A_{si} = \frac{A}{2.0} \quad = \text{shear area} \qquad (14.16–4)$$

where: π = 3.141592653

D_o = outside diameter (input quantity OD on **R** command)

D_i = inside diameter = $D_o - 2t_w$

t_w = wall thickness (input quantity TKWALL on **R** command)

$$C_f = \begin{cases} 1.0 & \text{if } f = 0.0 \\ f & \text{if } f > 0.0 \end{cases}$$

f = flexibility factor (input quantity FLEX on **R** command)

Further, the axial stiffness of the element is defined as

$$K_\ell(1,1) = \begin{cases} \dfrac{A^wE}{L} & \text{if } k = 0.0 \\ k & \text{if } k > 0.0 \end{cases} \qquad (14.16–5)$$

where: $K_\ell(1,1)$ = axial stiffness of the element

E = Young's modulus (input as EX on **MP** command)

L = element length

k = alternate axial pipe stiffness (input quantity STIFF on **RMORE** command)

14.16.4 Mass Matrix

The element mass matrix of PIPE16 is the same as for BEAM4, except total mass of the element is assumed to be:

$$m_e = m_e^w + \left(\rho_{fl} A^{fl} + \rho_{in} A^{in}\right) L \qquad (14.16\text{–}6)$$

where: m_e = total mass of element

m_e^w = pipe wall mass

$$= \begin{cases} \rho\ A^w L & \text{if } m_w = 0.0 \\ m_w & \text{if } m_w > 0.0 \end{cases}$$

m_w = alternate pipe wall mass (input quantity MWALL on **RMORE** command)

ϱ = pipe wall density (input as DENS on **MP** command)

ϱ_{fl} = internal fluid density (input quantity DENSFL on **R** command)

A^{fl} = $\frac{\pi}{4} D_i^{\,2}$

ϱ_{in} = insulation density (input quantity DENSIN on **RMORE** command)

A^{in} = insulation cross–sectional area

$$= \begin{cases} \frac{\pi}{4}\left(D_{o+}^{\,2} - D_o^{\,2}\right) & \text{if } A_s^{in} = 0.0 \\ \frac{A^{in}\, t^{in}}{L} & \text{if } A_s^{in} > 0.0 \end{cases}$$

D_{o+} = $D_o + 2t^{in}$

t^{in} = insulation thickness (input quantity TKIN on **RMORE** command)

A_s^{in} = alternate representation of the surface area of the outside of the pipe element (input quantity AREAIN on **RMORE** command)

Also, the bending moments of inertia (equation (14.16–2)) are used without the C_f term.

14.16.5 Gyroscopic Damping Matrix

The element gyroscopic damping matrix is:

$$[C_e] = 2\Omega\rho AL \begin{bmatrix} 0 &&&&&&&&&&& \\ 0 & 0 &&&&&&&&&& \\ 0 & -g & 0 &&&&&&&&& \\ 0 & 0 & 0 & 0 &&&&&& \text{Antisymmetric} && \\ 0 & -h & 0 & 0 & 0 &&&&&&& \\ 0 & 0 & -h & 0 & -i & 0 &&&&&& \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 &&&&& \\ 0 & 0 & -g & 0 & -h & 0 & 0 & 0 &&&& \\ 0 & g & 0 & 0 & 0 & -h & 0 & -g & 0 &&& \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 && \\ 0 & -h & 0 & 0 & 0 & j & 0 & h & 0 & 0 & 0 & \\ 0 & 0 & -h & 0 & -j & 0 & 0 & 0 & h & 0 & -i & 0 \end{bmatrix} \qquad (14.16\text{–}7)$$

where:

Ω = rotation frequency about the positive x axis (input quantity SPIN on **RMORE** command)

$g = \dfrac{6/5\ r^2}{L^2\,(1+\phi)^2}$

$h = \dfrac{-\,(1/10 - 1/2\,\phi)\,r^2}{L\,(1+\phi)^2}$

$i = \dfrac{(2/15 + 1/6\,\phi + 1/3\,\phi^2)\,r^2}{(1+\phi)^2}$

$j = \dfrac{-\,(1/30 + 1/6\,\phi - 1/6\,\phi^2)\,r^2}{(1+\phi)^2}$

$r = \sqrt{I/A}$

$\phi = \dfrac{12\ EI}{G\ A^s L^2}$

G = shear modulus (input as GXY on **MP** command)

A^s = shear area (=A^w/2.0)

14.16.6 Stress Stiffness Matrix

The element stress stiffness matrix of PIPE16 is identical to that for BEAM4.

14.16.7 Load Vector

The element pressure load vector is

$$\{F_\ell\} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_{12} \end{Bmatrix} \tag{14.16–8}$$

where:

$F_1 = F_A + F_P$

$F_7 = -F_A + F_P$

$F_A = A^w \, E \, \epsilon_x^{pr}$

ϵ_x^{pr} = axial strain due to pressure load, defined below

$$F_P = \begin{cases} 0.0 & \text{if KEYOPT(5) = 0} \\ \dfrac{P_1 \, L \, C^A}{2} & \text{if KEYOPT(5) = 1} \end{cases}$$

$F_2 = F_8 = \dfrac{P_2 \, L \, C^A}{2}$

$F_3 = F_9 = \dfrac{P_3 \, L \, C^A}{2}$

$F_4 = F_{10} = 0.0$

$F_5 = -F_{11} = \dfrac{P_3 \, L^2 \, C^A}{12}$

$F_6 = -F_{12} = \dfrac{P_2 \, L^2 \, C^A}{12}$

P_1 = parallel pressure component in element coordinate system (force/unit length)

P_2, P_3 = transverse pressure components in element coordinate system (force/unit length)

$$C^A = \begin{cases} 1.0 & \text{if KEYOPT(5) = 0} \\ \text{positive sine of the angle between the axis of the element and the direction of the pressures, as defined by } P_1, P_2, \text{ and } P_3 & \text{if KEYOPT(5) = 1} \end{cases}$$

The transverse pressures are assumed to act on the centerline, and not on the inner or outer surfaces. The transverse pressures in the element coordinate system are computed by

$$\begin{Bmatrix} P_1 \\ P_2 \\ P_3 \end{Bmatrix} = [T] \begin{Bmatrix} P_X \\ P_Y \\ P_Z \end{Bmatrix} \qquad (14.16–9)$$

where:
[T] = conversion matrix defined in equation (14.4–6)
P_X = transverse pressure acting in global Cartesian X direction) (input quantity PX)
P_Y = transverse pressure acting in global Cartesian Y direction) (input quantity PY)
P_Z = transverse pressure acting in global Cartesian Z direction) (input quantity PZ)

ϵ_x^{pr}, the unrestrained axial strain caused by internal and external pressure effects, is needed to compute the pressure part of the element load vector (see Figure 14.16–1).

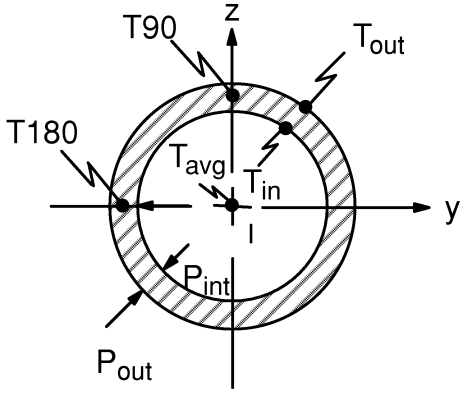


Figure 14.16–1 Thermal and Pressure Effects

ϵ_x^{pr} is computed using thick wall (Lame') effects:

$$\epsilon_x^{pr} = \frac{1}{E} (1 - 2\nu) \left(\frac{P_i\, D_i^{\,2} - P_o\, D_o^{\,2}}{D_o^{\,2} - D_i^{\,2}} \right) \qquad (14.16–10)$$

where:
ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)
P_i = internal pressure (input on **SFE** command)
P_o = external pressure (input on **SFE** command)

An element thermal load vector is computed also, based on thick wall effects.

14.16.8 Stress Calculation

The output stresses, computed at the outside surface and illustrated in Figure 14.16–2 and Figure 14.16–3, are calculated from the following definitions:

$$\sigma_{dir} = \frac{F_x + \frac{\pi}{4}\left(P_i\ D_i{}^2 - P_o\ D_o{}^2\right)}{a^w} \tag{14.16–11}$$

$$\sigma_{bend} = C_\sigma \frac{M_b\ r_o}{I_r} \tag{14.16–12}$$

$$\sigma_{tor} = \frac{M_x\ r_o}{J} \tag{14.16–13}$$

$$\sigma_h = \frac{P_i D_i - P_o D_o}{2t_e} \tag{14.16–14}$$

$$\sigma_{\ell f} = \frac{2\ F_s}{A^w} \tag{14.16–15}$$

where:

σ_{dir} = direct stress (output quantity SDIR)

F_x = axial force

a_w = $\frac{\pi}{4}\left(d_o{}^2 - D_i{}^2\right)$

d_o = $2\ r_o$

r_o = $\frac{D_o}{2} - t_c$

t_c = corrosion allowance (input quantity TKCORR on **RMORE** command)

σ_{bend} = bending stress (output quantity SBEND)

C_σ = stress intensification factor, defined in Table 14.16–1

M_b = bending moment = $\sqrt{M_y^2 + M_z^2}$

I_r = $\frac{\pi}{64}\left(d_o{}^4 - D_i{}^4\right)$

σ_{tor} = torsional shear stress (output quantity ST)

M_x = torsional moment

J = $2I_r$

σ_h = hoop pressure stress (output quantity SH)

R_i = $\frac{D_i}{2}$

t_e = $t_w - t_c$

$\sigma_{\ell f}$ = lateral force shear stress (output quantity SSF)

F_s = shear force = $\sqrt{F_y^2 + F_z^2}$

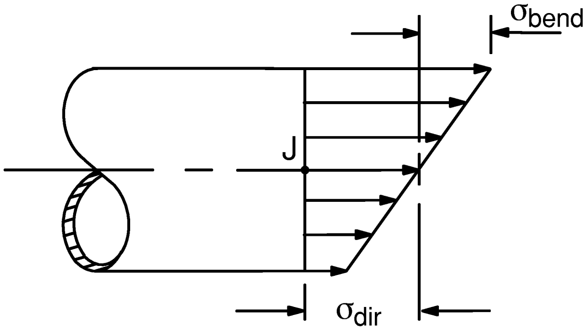


Figure 14.16–2 Elastic Pipe Output

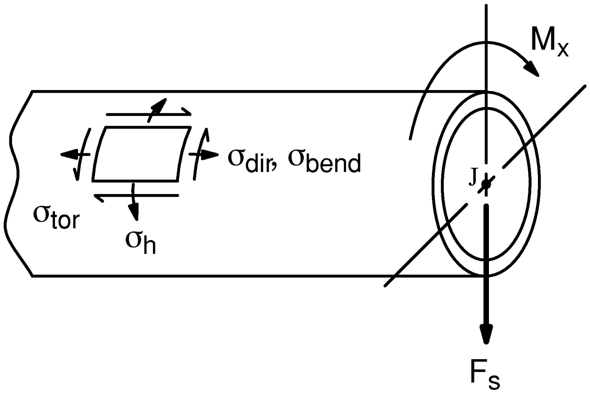


Figure 14.16–3 Elastic Pipe Output

Stress intensification factors are given in Table 14.16–1.

Table 14.16–1 Stress Intensification Factors

KEYOPT(2)	C_σ	
	At node I	At node J
0	$C_{\sigma,I}$	$C_{\sigma,J}$
1	$C_{\sigma,T}$	1.0
2	1.0	$C_{\sigma,T}$
3	$C_{\sigma,T}$	$C_{\sigma,T}$

Any entry in Table 14.16–1 either input as or computed to be less than 1.0 is set to 1.0. The entries are:

$C_{\sigma,I}$ = input quantity SIFI on **R** command

$C_{\sigma,J}$ = input quantity SIFJ on **R** command

$C_{\sigma,T}$ = “T” stress intensification factor (ASME Code(40))

$$= \frac{0.9}{\left(\frac{4\,t_w}{(D_i + d_o)}\right)^{2/3}}$$

σ_{th} (output quantity STH), which is in the postprocessing file, represents the stress due to the thermal gradient thru the thickness. If the temperatures are given as nodal temperatures, σ_{th} = 0.0. But, if the temperatures are input as element temperatures,

$$\sigma_{th} = -\frac{E\alpha\,(T_o - T_a)}{1 - \upsilon} \tag{14.16–16}$$

where: T_o = temperature at outside surface

T_a = temperature midway thru wall

Equation (14.16–16) is derived as a special case of equations (2.1–12), (2.1–13) and (2.1–15) with y as the hoop coordinate (h) and z as the radial coordinate (r). Specifically, these equations

1. are specialized to an isotropic material
2. are premultiplied by [D] and –1
3. have all motions set to zero, hence $\varepsilon_x = \varepsilon_h = \varepsilon_r = \gamma_{xh} = \gamma_{hr} = \gamma_{xr} = 0.0$
4. have $\sigma_r = \tau_{hr} = \tau_{xr} = 0.0$ since $r = R_o$ is a free surface.

This results in:

$$\begin{Bmatrix} \sigma^t_x \\ \sigma^t_h \\ \sigma^t_{xh} \end{Bmatrix} = \begin{bmatrix} -\frac{E}{1-\nu^2} & -\frac{\upsilon E}{1-\nu^2} & 0 \\ -\frac{\nu E}{1-\nu^2} & -\frac{E}{1-\nu^2} & 0 \\ 0 & 0 & -G \end{bmatrix} \begin{Bmatrix} \alpha\Delta T \\ \alpha\Delta T \\ 0 \end{Bmatrix} \tag{14.16–17}$$

or

$$\sigma^t_x = \sigma^t_h = -\frac{E\alpha\Delta T}{1 - \nu} = \sigma_{th} \tag{14.16–18}$$

and

$$\sigma^{t}_{xh} = 0 \tag{14.16–19}$$

Finally, the axial and shear stresses are combined with:

$$\sigma_x = \sigma_{dir} + A\ \sigma_{bend} + \sigma_{th} \tag{14.16–20}$$

$$\sigma_{xh} = \sigma_{tor} + B\ \sigma_{\ell f} \tag{14.16–21}$$

where: A, B = sine and cosine functions at the appropriate angle

σ_x = output quantity SAXL

σ_{xh} = output quantity SXH

The maximum and minimum principal stresses, as well as the stress intensity and the equivalent stress, are based on the stresses at two extreme points on opposite sides of the bending axis, as shown in Figure 14.16–4. If shear stresses due to lateral forces $(\sigma_{\ell f})$ are greater than the bending stresses, the two points of maximum shearing stresses due to those forces are reported instead. The stresses are calculated from the typical Mohr's circle approach in Figure 14.16–5.

The equivalent stress for Point 1 is based on the three principal stresses which are designated by small circles in Figure 14.16–5. Note that one of the small circles is at the origin. This represents the radial stress on the outside of the pipe, which is equal to zero (unless $P_o \neq 0.0$). Similarly, the points marked with an X represent the principal stresses associated with Point 2, and a second equivalent stress is derived from them.

Next, the program selects the largest of the four maximum principal stresses (σ_1, output quantities S1MX), the smallest of the four minimum principal stresses (σ_3, output quantity S3MN), the largest of the four stress intensities (σ_I, output quantity SINTMX), and the largest of the four equivalent stresses (σ_e, output quantity SEQVMX). Finally, these are also compared (and replaced as necessary) to the values at the right positions around the circumference at each end. These four values are then printed out and put on the postprocessing file.

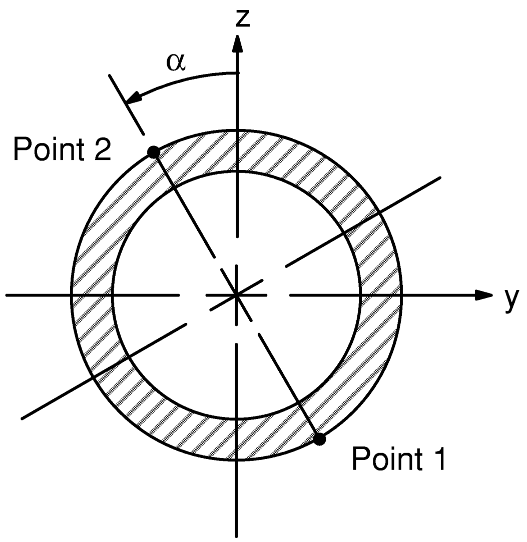


Figure 14.16–4 Stress Point Locations

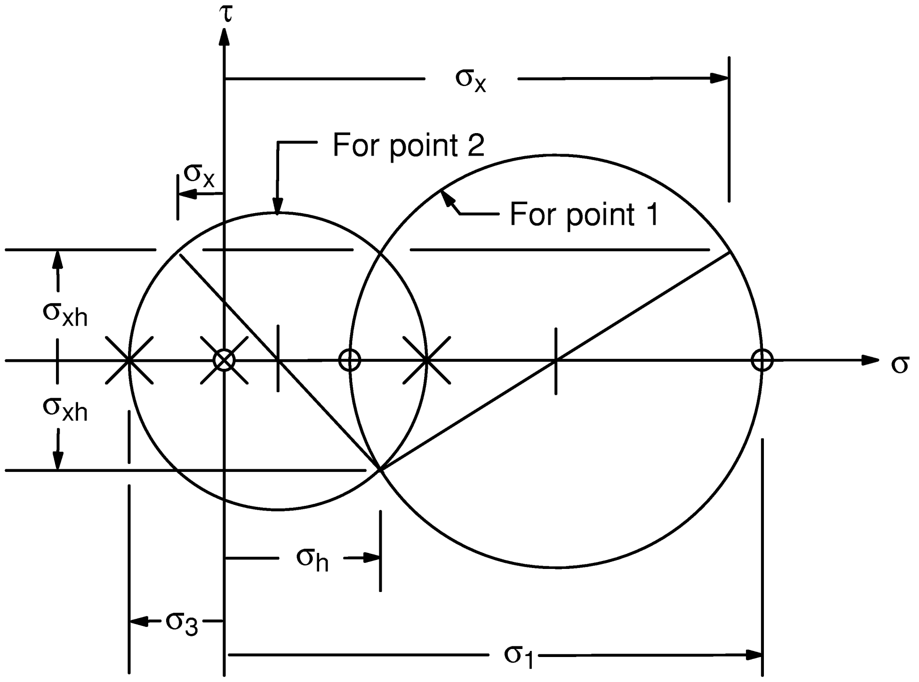


Figure 14.16–5 Mohr Circles

Three additional items are put on the postdata file for use with certain code checking. These are:

$$\sigma^{c}_{pr} = \frac{P_i D_o}{4t_w} \tag{14.16–22}$$

$$\sigma_{MI} = \sqrt{M_{XI}^2 + M_{YI}^2 + M_{ZI}^2}\ \frac{D_o}{2I} \qquad (14.16–23)$$

$$\sigma_{MJ} = \sqrt{M_{XJ}^2 + M_{YJ}^2 + M_{ZJ}^2}\ \frac{D_o}{2I} \qquad (14.16–24)$$

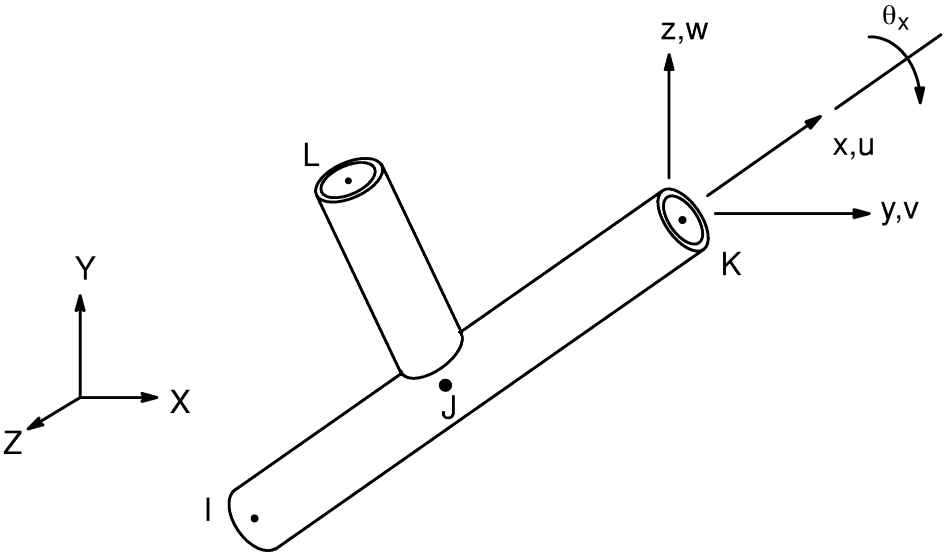
where: σ_{pr}^c = output quantity SPR2

σ_{MI} = output quantity SMI

σ_{MJ} = output quantity SMJ

M_{XI} = moment about the x axis at node I, etc.

14.17 PIPE17 — Elastic Pipe Tee



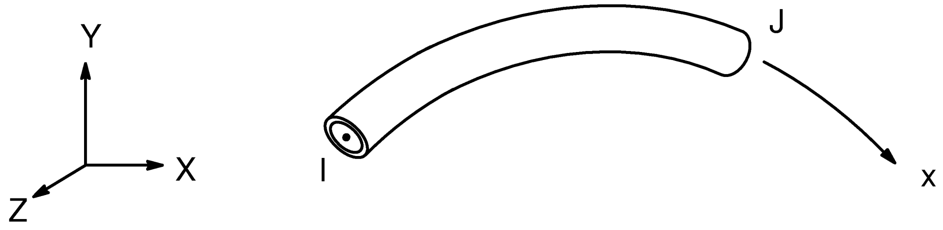
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.2–1), (12.2.2–2), (12.2.2–3), and (12.2.2–4)	None
Mass Matrix	Same as stiffness matrix	None
Stress Stiffness Matrix	Equation (12.2.2–2) and (12.2.2–3)	None
Load Vector (Pressure and Thermal)	Equation (12.2.2–1), (12.2.2–2), and (12.2.2–3)	None

Load Type	Distribution
Element Temperature	In each branch: linear thru thickness, constant along the length
Nodal Temperature	In each branch: constant thru thickness, linear along the length
Pressure	Internal and External: constant on all branches along the length and around the circumference Lateral: constant on each branch along the length

14.17.1 Other Applicable Sections

PIPE17 is essentially the same as three PIPE16 (elastic straight pipe) elements, as discussed in Section 14.16.

14.18 PIPE18 — Elastic Curved Pipe (Elbow)



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	No shape functions are explicitly used. Rather a flexibility matrix similar to that developed by Chen(4) is inverted and used.	None
Mass Matrix	No shape functions are used. Rather a lumped mass matrix using only translational degrees of freedom is used.	None
Thermal and Pressure Load Vector	Equation (12.2.2–1), (12.2.2–2), and (12.2.2–3)	None

Load Type	Distribution
Element Temperature	Linear thru thickness or across diameter, and along length
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Internal and External: constant along length and around the circumference Lateral: varies trigonometrically along length (see below)

14.18.1 Other Applicable Sections

Section 14.16 covers some of the applicable stress calculations.

14.18.2 Stiffness Matrix

The geometry in the plane of the element is given in Figure 14.18–1.

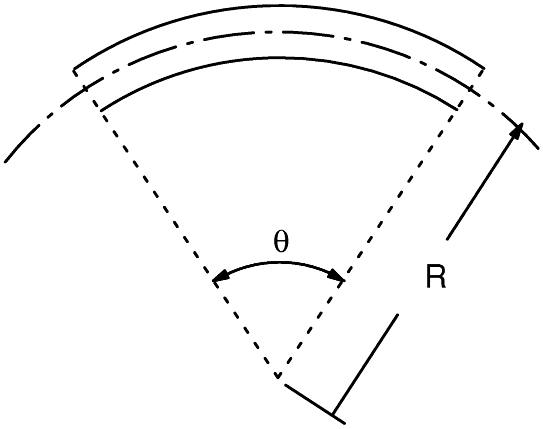


Figure 14.18–1 Plane Element

The stiffness matrix is developed based on an approach similar to that of Chen(4). The flexibility of one end with respect to the other is:

$$[f] = \begin{bmatrix} f_{11} & 0 & f_{13} & 0 & f_{15} & 0 \\ 0 & f_{22} & 0 & f_{24} & 0 & f_{26} \\ f_{31} & 0 & f_{33} & 0 & f_{35} & 0 \\ 0 & f_{42} & 0 & f_{44} & 0 & f_{46} \\ f_{51} & 0 & f_{53} & 0 & f_{55} & 0 \\ 0 & f_{62} & 0 & f_{64} & 0 & f_{66} \end{bmatrix} \qquad (14.18\text{–}1)$$

where:

$$f_{11} = \frac{R^3C_{fi}}{EI}\left(\frac{\theta}{2}\cos\theta - \frac{3}{2}\sin\theta + \theta\right) + \frac{R}{2EA^w}\left(\theta\cos\theta + \sin\theta\right) + \frac{2R\,(1 + \nu)}{EA^w}\left(\theta\cos\theta - \sin\theta\right)$$

$$f_{13} = -f_{31} = \frac{R^3C_{fi}}{EI}\left(\cos\theta - 1 + \frac{\theta}{2}\sin\theta\right) + \frac{R\theta\sin\theta}{EA^w}\left(\frac{5}{2} + 2\nu\right)$$

$$f_{15} = f_{51} = \frac{R^2 C_{fi}}{EI} (\sin \theta - \theta)$$

$$f_{22} = \frac{R^3 (1 + \nu)}{EI} (\theta - \sin \theta) + \frac{R^3}{2EI} \left(1 + \nu + C_{fo}\right) (\theta \cos \theta - \sin \theta) + \frac{R\theta \left(4(1 + \nu)\right)}{EA^w}$$

$$f_{24} = f_{42} = \frac{R^2}{2EI} \left(1 + \nu + C_{fo}\right) \left(\theta \cos \theta - \sin \theta\right)$$

$$f_{26} = -f_{62} = \frac{R^2}{EI} \left((1 + \nu) (\cos \theta - 1) + \frac{\theta}{2} \sin \theta \, (1 + \nu + C_{fo})\right)$$

$$f_{33} = \left(\frac{\theta}{2} \cos \theta - \frac{1}{2} \sin \theta\right) \left(\frac{R^3 C_{fi}}{EI} + \frac{R}{EA^w}\right) + \left(\frac{\theta}{2} \cos \theta + \frac{1}{2} \sin \theta\right)\left(\frac{4R(1 + \nu)}{EA^w}\right)$$

$$f_{35} = -f_{53} = \frac{R^2 C_{fi}}{EI} (\cos \theta - 1)$$

$$f_{44} = \frac{R}{2EI} \left(1 + \nu + C_{fo}\right) \theta \cos \theta + \frac{R}{2EI} \left(1 + \nu - C_{fo}\right) \sin \theta$$

$$f_{46} = -f_{64} = \frac{R}{2EI} \left(1 + \nu + C_{fo}\right) \theta \sin \theta$$

$$f_{55} = \frac{R C_{fi}}{EI} \theta$$

$$f_{66} = \frac{R}{2EI}\left(\left(1 + \nu + C_{fo}\right) \theta \cos \theta - \left(1 + \nu - C_{fo}\right) \sin \theta\right)$$

and where:

R = radius of curvature (input as RADCUR on **R** command) (see Figure 14.18–1)

θ = included angle of element (see Figure 14.18–1)

E = Young's modulus (input as EX on **MP** command)

ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

I = moment of inertia of cross–section = $\frac{\pi}{64} \left(D_o^4 - D_i^4\right)$

A^w = area of cross–section = $\frac{\pi}{4} \left(D_o^2 - D_i^2\right)$

D_o = outside diameter (input as OD on **R** command)

D_i = $D_o - 2t$ = inside diameter

t = wall thickness (input as TKWALL on **R** command)

$$C_{fi} = \begin{cases} C'_{fi} & \text{if } C'_{fi} > 0.0 \\ \dfrac{1.65}{h} \text{ or } 1.0, & \begin{array}{l}\text{whichever is greater if} \\ C'_{fi} = 0.0 \text{ and KEYOPT(3)=0} \\ \text{(ASME flexibility factor,} \\ \text{ASME Code(40))}\end{array} \\ \dfrac{1.65}{h\left(1 + \dfrac{PrX_K}{tE}\right)} \text{ or } 1.0 & \begin{array}{l}\text{whichever is greater if } C'_{fi} = 0.0 \\ \text{and KEYOPT(3)=1 (ASME} \\ \text{flexibility factor, ASME Code(40))}\end{array} \\ \dfrac{10 + 12h^2}{1 + 12h^2} & \begin{array}{l}\text{if } C'_{fi} = 0.0 \text{ and KEYOPT(3) = 2} \\ \text{(Karman flexiblity factor)}\end{array} \end{cases}$$

C'_{fi} = input as FLXI on **R** command

$h = \dfrac{t\,R}{r^2}$

r = average radius $\dfrac{(D_o - t)}{2}$

$$P = \begin{cases} P_i - P_o & \text{if } P_i - P_o > 0.0 \\ 0.0 & \text{if } P_i - P_o \leq 0.0 \end{cases}$$

P_i = internal pressure (input on **SFE** command)

P_o = external pressure (input on **SFE** command)

$$X_K = \begin{cases} 6\left(\dfrac{r}{t}\right)^{\frac{4}{3}}\left(\dfrac{R}{r}\right)^{\frac{1}{3}} & \text{if } \dfrac{R}{r} \geq 1.7 \\ 0.0 & \text{if } \dfrac{R}{r} < 1.7 \end{cases}$$

$$C'_{fo} = \begin{cases} C'_{fo} \text{ if } C'_{fo} > 0.0 \\ C_{fi} \text{ if } C'_{fo} = 0.0 \end{cases}$$

C'_{fo} = input as FLXO on **RMORE** command

The user should not use the KEYOPT(3) = 1 option if:

$$\theta_c R < 2r \qquad (14.18\text{–}2)$$

where: θ_c = included angle of the complete elbow, not just the included angle for this element (θ)

Next, the 6 x 6 stiffness matrix is derived from the flexibility matrix by inversion:

$$[K_o] = [f]^{-1} \qquad (14.18\text{–}3)$$

The full 12 x 12 stiffness matrix (in element coordinates) is derived by expanding the 6 x 6 matrix derived above and transforming to the global coordinate system.

14.18.3 Mass Matrix

The element mass matrix is a diagonal (lumped) matrix with each translation term being defined as:

$$m_t = \frac{m_e}{2} \tag{14.18–4}$$

where:

m_t = mass at each node in each translation direction

m_e = total mass of element

= $(\varrho A^w + \varrho_{fl} A^{fl} + \varrho_{in} A^{in}) R\theta$

ϱ = pipe wall density (input as DENS on **MP** command)

ϱ_{fl} = internal fluid density (input as DENSFL on **RMORE** command)

A^{fl} = $\frac{\pi}{4} D_i^{\ 2}$

ϱ_{in} = insulation density (input as DENSIN on **RMORE** command)

A^{in} = insulation cross–section area

= $\frac{\pi}{4} \left(D_{o+}^2 - D_o^{\,2} \right)$

D_{o+} = $D_o + 2\ t^{in}$

t^{in} = insulation thickness (input as TKIN on **RMORE** command)

14.18.4 Load Vector

The load vector in element coordinates due to thermal and pressure effects is:

$$\left\{ F_\ell^{th} \right\} + \left\{ F_\ell^{pr,i} \right\} = R\ \epsilon_x\ [K_e]\ \{A\} + \{ F_\ell^{pr,t} \} \tag{14.18–5}$$

where:

ε_x = strain caused by thermal as well as internal and external pressure effects (see equation (14.16–10))

$[K_e]$ = element stiffness matrix in global coordinates

$\{A\}$ = $\lfloor 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0 \ \vdots\ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ 0 \rfloor^T$

$\left\{ F_\ell^{pr,t} \right\}$ = element load vector due to transverse pressure

$\left\{ F_\ell^{pr,t} \right\}$ is computed based on input quantities PX, PY, PZ and curved beam formulas from Roark(48). Table 18, reference no. (loading) 3, 4, and 5 and 5c was used for in–plane effects and Table 19, reference no. (end restraint) 4e was used for out–of–plane effects. As a radial load varying trigonometrically along the length of the element was not one of the available cases given in Roark(48), an integration of a point radial load was done, using Loading 5c.

14.18.5 Stress Calculations

In the stress pass, the stress evaluation is similar to that for PIPE16 (Section 14.16). It is not the same as for PIPE60. The wall thickness is diminished by t_c, the corrosion allowance (input as TKCORR on **R** command). See also Section 14.16. The bending stress components are multiplied by stress intensification factors (C_σ). The “intensified” stresses are used in the principal and combined stress calculations. The factors are:

$$C_{\sigma,I} = \begin{cases} C_o & \text{, if SIFI} < 1.0 \\ \text{input as SIFI on } \mathbf{R} \text{ command} & \text{, if SIFI} > 1.0 \end{cases} \quad (14.18\text{–}6)$$

$$C_{\sigma,J} = \begin{cases} C_o & \text{, if SIFJ} < 1.0 \\ \text{input as SIFJ on } \mathbf{R} \text{ command} & \text{, if SIFJ} > 1.0 \end{cases} \quad (14.18\text{–}7)$$

$$C_o = \begin{cases} \dfrac{0.9}{h_e^{\,2/3}} & \\ & \text{whichever is greater (ASME Code (40))} \\ 1.0 & \end{cases} \quad (14.18\text{–}8)$$

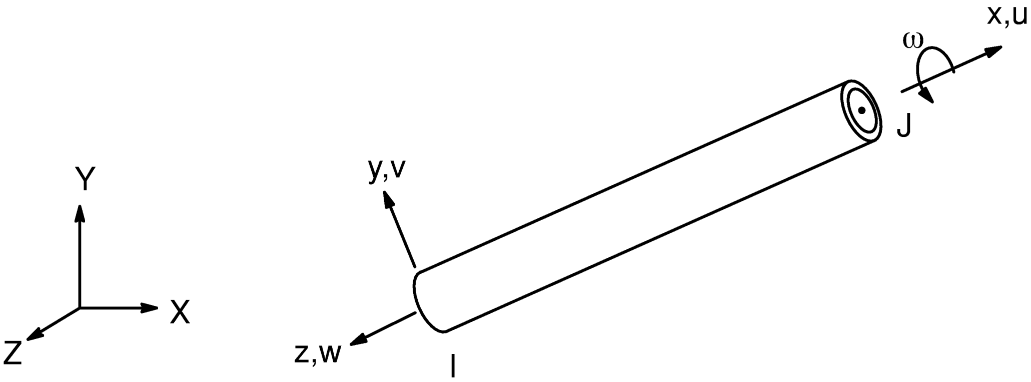
where:

$$h_e = \frac{16\ t_e\ R}{\left(D_i + d_o\right)^2}$$

$$t_e = t - t_c$$

$$d_o = D_o - 2\ t_c$$

14.20 PIPE20 — Plastic Straight Pipe



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.2–1), (12.2.2–2), (12.2.2–3), and (12.2.2–4)	None for elastic matrix. Same as Newton–Raphson load vector for tangent matrix with plasticity
Stress Stiffness Matrix	Equations (12.2.2–2) and (12.2.2–3)	None
Mass Matrix	Same as stiffness matrix	None
Pressure and Thermal Load Vector	Equations (12.2.2–1), (12.2.2–2), and (12.2.2–3)	None
Newton–Raphson Load Vector	Same as stiffness matrix	2 along the length and 8 points around circumference. The points are located midway between the inside and outside surfaces.

Load Type	Distribution
Element Temperature	Linear across diameter and along length
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Internal and External: constant along length and around circumference Lateral: constant along length

14.20.1 Assumptions and Restrictions

The radius/thickness ratio is assumed to be large.

14.20.2 Other Applicable Sections

Section 14.4 has an elastic beam element stiffness and mass matrix explicitly written out. Section 14.16 discusses the effect of element pressure and the elastic stress printout. Section 14.23 defines the tangent matrix with plasticity and the Newton–Raphson load vector.

14.20.3 Stress and Strain Calculation

PIPE20 uses four components of stress and strain in the stress calculation:

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_h \\ \sigma_r \\ \sigma_{xh} \end{Bmatrix} \tag{14.20–1}$$

where x, h, r are subscripts representing the axial, hoop and radial directions, respectively. Since only the axial and shear strains can be computed directly from the strain–displacement matrices, the strains are computed from the stresses as follows.

The stresses (before plasticity adjustment) are defined as:

$$\sigma_x = E\epsilon' + \frac{\pi}{4A^w}\left(D_i{}^2P_i - D_o{}^2P_o\right) \tag{14.20–2}$$

$$\sigma_h = \frac{1}{2t}\left(D_iP_i - D_oP_o\right) \tag{14.20–3}$$

$$\sigma_r = -\frac{1}{2}\left(P_i - P_o\right) \qquad (14.20–4)$$

$$\sigma_{xh} = \frac{2}{A^w}\left(F_y \sin\beta_j - F_z \cos\beta_j\right) + \frac{M_x D_m}{2\,J} \qquad (14.20–5)$$

where:
ϵ' = modified axial strain (see Section 14.23)
E = Young's modulus (input as EX on **MP** command)
P_i = internal pressure (input on **SFE** command)
P_o = external pressure (input on **SFE** command)
D_i = internal diameter = D_o–2t
D_o = input as OUTER DIA. on **R** command
t = wall thickness (input as WALL THICKNESS on **R** command)
A^w = $\frac{\pi}{4}\left(D_o^2 – D_i^2\right)$ = wall area
J = $\frac{\pi}{4}\ D_m^3 t$
D_m = $\left(D_i + D_o\right)/2$ = average diameter
β_j = angular position of integration point J (see Figure 14.20–1) (output quantity ANGLE)
F_y, F_z, M_x = forces on element node by integration point

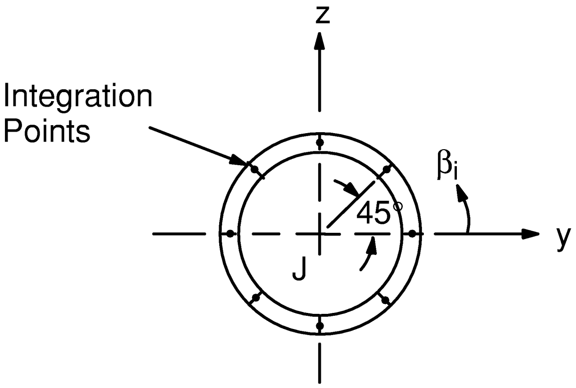


Figure 14.20–1 Integration Points for End J

The forces on the element (F_y, F_z, M_x) are computed from:

$$\{F_\ell\} = \left[T_R\right]\left(\left[K_e\right]\{\Delta u_e\} - \{F_e\}\right) \qquad (14.20–6)$$

where:
$\{F_\ell\}$ = output quantities FORCES ON MEMBER AT NODE
$[T_R]$ = global to local conversion matrix
$[K_e]$ = element stiffness matrix

$\{\Delta u_e\}$ = element incremental displacement vector

$\{F_e\}$ = element load vector from pressure, thermal and Newton–Raphson restoring force effects

The forces $\{F_\ell\}$ are in element coordinates while the other terms are given in global Cartesian coordinates. The forces used in equation (14.20–5) correspond to either those at node I or node J, depending at which end the stresses are being evaluated.

The modified total strains for the axial and shear components are readily calculated by:

$$\epsilon_x' = \frac{1}{E}\left(\sigma_x - \nu\left(\sigma_h + \sigma_r\right)\right) \tag{14.20–7}$$

$$\epsilon_{xh}' = \frac{\sigma_{xh}}{G} \tag{14.20–8}$$

where: ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

G = shear modulus (input as GXY on **MP** command)

The hoop and radial modified total strains are computed through:

$$\epsilon_h' = \epsilon_{h,n-1} + \Delta\epsilon_h \tag{14.20–9}$$

$$\epsilon_r' = \epsilon_{r,n-1} + \Delta\epsilon_r \tag{14.20–10}$$

where: $\varepsilon_{h,n-1}$ = hoop strain from the previous iteration

$\varepsilon_{r,n-1}$ = radial strain from the previous iteration

$\Delta\varepsilon_h$ = increment in hoop strain

$\Delta\varepsilon_r$ = increment in radial strain

The strains from the previous iterations are computed using:

$$\epsilon_{h,n-1} = \frac{1}{E}\left(\sigma_h - \nu\left(\sigma_{x,n-1} + \sigma_r\right)\right) \tag{14.20–11}$$

$$\epsilon_{r,n-1} = \frac{1}{E}\left(\sigma_r - \nu\left(\sigma_{x,n-1} + \sigma_h\right)\right) \tag{14.20–12}$$

where $\sigma_{x,n-1}$ is computed using equation (14.20–2) with the modified total strain from the previous iteration. The strain increments in equations (14.20–9) and (14.20–10) are computed from the strain increment in the axial direction:

$$\Delta\epsilon_h = \overline{D}_n^h \Delta\epsilon_x \tag{14.20–13}$$

$$\Delta\epsilon_r = \overline{D}_n^r \Delta\epsilon_x \tag{14.20–14}$$

where:

$\Delta\varepsilon_x$ = $\epsilon' - \epsilon'_{n-1}$ = axial strain increment

$\overline{D}_n^h,\ \overline{D}_n^r$ = factors relating axial strain increment to hoop and radial strain increments, respectively

These factors are obtained from the static condensation of the 3–D elasto–plastic stress–strain matrix to the 1–D component, which is done to form the tangent stiffness matrix for plasticity.

Equations (14.20–7) through (14.20–10) define the four components of the modified total strain from which the plastic strain increment vector can be computed (see Section 4.1). The elastic strains are:

$$\left\{\epsilon^{el}\right\} = \left\{\epsilon'\right\} - \left\{\Delta\epsilon^{pl}\right\} \tag{14.20–15}$$

where:

$\{\varepsilon^{el}\}$ = output quantities EPELAXL, EPELRAD, EPELH, EPELXH

$\{\Delta\varepsilon^{pl}\}$ = plastic strain increment

The stresses are then:

$$\{\sigma\} = [D]\left\{\epsilon^{el}\right\} \tag{14.20–16}$$

where:

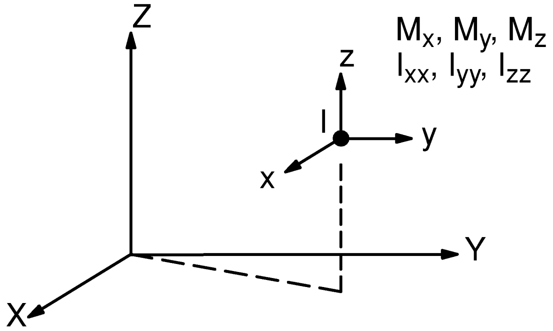
$\{\sigma\}$ = output quantities SAXL, SRAD, SH, SXH

$[D]$ = elastic stress–strain matrix

The definition of $\{\sigma\}$ given by equation (14.20–16) is modified in that σ_h and σ_r are redefined by equations (14.20–3) and (14.20–4) as the stress values and must be maintained, regardless of the amount of plastic strain.

As long as the element remains elastic, additional printout is given during the solution phase. The stress intensification factors (C_σ) of PIPE16 are used in this printout, but are not used in the printout associated with the plastic stresses and strains. The maximum principal stresses, the stress intensity, and equivalent stresses are compared (and replaced if necessary) to the values of the plastic printout at the eight positions around the circumference at each end. Also, the elastic printout is based on stresses at the outer fiber, but the plastic printout is based on mid–thickness stresses. Hence, some apparent inconsistency appears in the printout.

14.21 MASS21 — Structural Mass



Matrix or Vector	Shape Functions	Integration Points
Mass Matrix	None	None

The element mass matrix is:

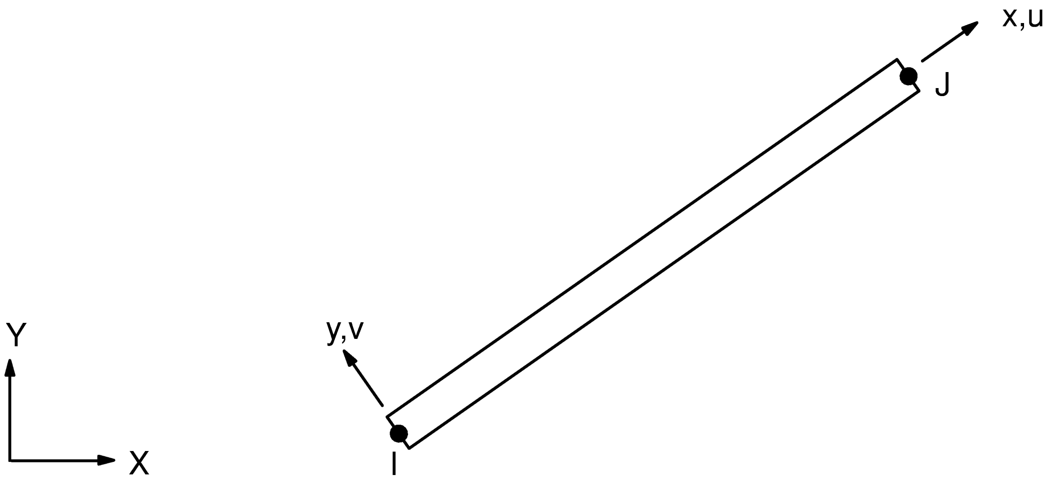
$$[M_e] = \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & 0 & 0 \\ 0 & 0 & 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 & e & 0 \\ 0 & 0 & 0 & 0 & 0 & f \end{bmatrix} \qquad (14.21\text{–}1)$$

where a, b, c, d, e, and f are input on the **R** command in the locations shown in the following table:

	KEYOPT(3)=0	KEYOPT(3)=2	KEYOPT(3)=3	KEYOPT(3)=4
a	1	1	1	1
b	2	1	1	1
c	3	1	–	–
d	4	–	–	–
e	5	–	–	–
f	6	–	2	–

For the mass summary, only the first real constant is used, regardless of which option of KEYOPT(3) is used. Analyses with inertial relief use the complete matrix.

14.23   BEAM23 — 2–D Plastic Beam



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.1.2–1) and (12.1.2–2)	None for elastic matrix. Same as Newton–Raphson load vector for tangent matrix with plasticity
Mass Matrix	Same as stress stiffness matrix	None
Stress Stiffness Matrix	Equation (12.1.2–2)	None
Thermal Load Vector	Same as stress stiffness matrix	None
Pressure Load Vector	Equation (12.1.2–2)	None
Newton–Raphson Load Vector	Same as stiffness matrix	3 along the length and 5 thru the thickness
Stress Evaluation	Same as stiffness matrix	Same as Newton–Raphson load vector

Load Type	Distribution
Element Temperature	Linear thru thickness and along length
Nodal Temperature	Constant thru thickness, linear along length
Pressure	Linear along length

14.23.1 Other Applicable Sections

The complete stiffness and mass matrices for an elastic 2–D beam element (BEAM3) are given in Section 14.3.

14.23.2 Integration Points

There are three sets of integration points along the length of the element, one at each end and one at the middle as shown in Figure 14.23–1.

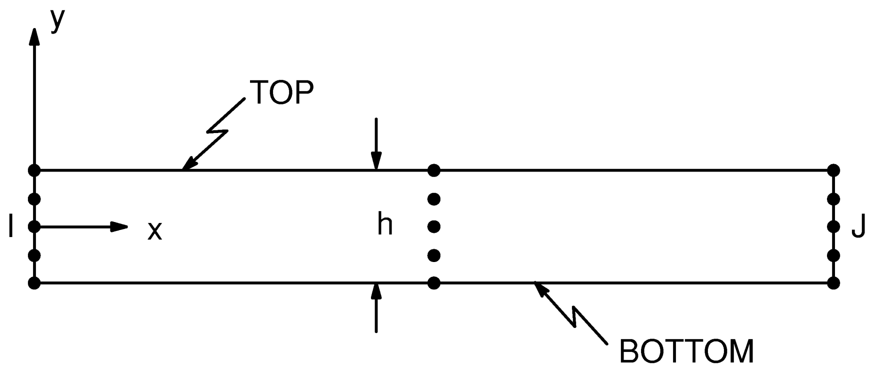


Figure 14.23–1 Integration Point Locations

h is defined as:

h = thickness or height of member (input quantity HEIGHT on **R** command)

The five integration points through the thickness are located at positions y = –0.5 h, –0.3 h, 0.0, 0.3 h, and 0.5 h. Each one of these points has a numerical integration factor associated with it, as well as an effective width, which are different for each type of cross–section. These are derived here in order to explain the procedure used in the element, as well as providing users with a good basis for selecting their own input values for the case of an arbitrary section (KEYOPT(6) = 4).

The criteria used for the element are:

1. The element, when under simple tension or compression, should respond exactly for elastic or plastic situations. That is, the area (A) of the element should be correct.
2. The first moment should be correct. This is nonzero only for unsymmetric cross–sections.
3. The element, when under pure bending, should respond correctly to elastic strains. That is, the (second) moment of inertia (I) of the element should be correct.
4. The third moment should be correct. This is nonzero only for unsymmetric cross–sections.
5. Finally, as is common for numerically integrated cross–sections, the fourth moment of the cross–section (I_4) should be correct.

For symmetrical sections an additional criterion is that symmetry about the centerline of the beam must be maintained. Thus,rather than five independent constants, there are only three. These three constants are sufficient to satisfy the previous three criteria exactly. Some other cases, such as plastic combinations of tension and bending, may not be satisfied exactly, but the discrepancy for actual problems is normally small. For the unsymmetric cross–section case, the user needs to solve five equations, not three. For this case, use of two additional equations representing the first and third moments are recommended. This case is not discussed further here.

The five criteria may be set up in equation form:

$$A = \int_{AREA} dA \tag{14.23–1}$$

$$I_1 = \int_{AREA} ydA \tag{14.23–2}$$

$$I_2 = \int_{AREA} y^2dA \tag{14.23–3}$$

$$I_3 = \int_{AREA} y^3dA \tag{14.23–4}$$

$$I_4 = \int_{AREA} y^4dA \tag{14.23–5}$$

where: dA = differential area

y = distance to centroid

These criteria can be rewritten in terms of the five integration points:

$$A = \sum_{i=1}^{5} H(i)\ L(i)\ h \tag{14.23–6}$$

$$I_1 = \sum_{i=1}^{5} H(i)\ L(i)\ h\ (hP(i)) \tag{14.23–7}$$

$$I_2 = \sum_{i=1}^{5} H(i)\ L(i)\ h\ \left(hP(i)\right)^2 \tag{14.23–8}$$

$$I_3 = \sum_{i=1}^{5} H(i)\ L(i)\ h\ \left(hP(i)\right)^3 \tag{14.23–9}$$

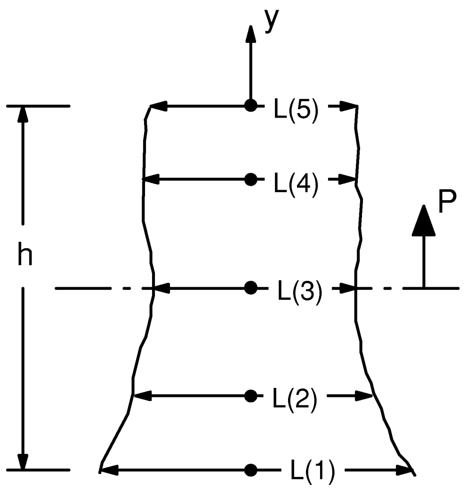
$$I_4 = \sum_{i=1}^{5} H(i)\ L(i)\ h\ \left(hP(i)\right)^4 \tag{14.23–10}$$

where: H(i) = weighting factor at point i

L(i) = effective width at point i

P(i) = integration point locations in y direction (P(1) = –0.5, P(2) = –0.3, etc.)

The L(i) follows physical reasoning whenever possible as in Figure 14.23–2.



(A(i) = L(i) x h)

Figure 14.23–2 Beam Widths

Starting with the case of a rectangular beam, all values of L(i) are equal to the width of the beam, which is computed from

$$L(i) = \frac{12\ I_{zz}}{h^3} \tag{14.23–11}$$

where: I_{zz} = moment of inertia (input quantity IZZ on **R** command)

Note that the area is not used in the computation of the width. As mentioned before, symmetry may be used to get H(1) = H(5) and H(2) = H(4). Thus, H(1), H(2), and H(3) may be derived by solving the simultaneous equations developed from the above three criteria. These weighting factors are used for all other cross–sections, with the appropriate adjustments made in L(i) based on the same criteria. The results are summarized in Table 14.23–1.

One interesting case to study is that of a rectangular cross–section that has gone completely plastic in bending. The appropriate parameter is the first moment of the area or

$$I_F = \int \left| y \right| \, dA \tag{14.23–12}$$

This results in

$$I_F = \sum_{i=1}^{5} H(i)\ L(i)\ h \left| h\ P(i) \right| \tag{14.23–13}$$

Table 14.23–1 Cross–Sectional Computation Factors

Numerical Integration Point (i)	Location Thru Thickness (P(i))	Numerical Weighting Factor (H(i))	Effective Width (L(i))			
			Rectangular	Pipe	Round Bar	Arbitrary Section
1	–.5	.06250000	$12I_{zz}/h^3$	$8.16445t_p$	$0.25341D_o$	A(–0.5)/h
2	–.3	.28935185	$12I_{zz}/h^3$	$2.64115t_p$	$0.79043D_o$	A(–0.3)/h
3	.0	.29629630	$12I_{zz}/h^3$	$2.00000t_p$	$1.00000D_o$	A(0.0)/h
4	.3	.28935185	$12I_{zz}/h^3$	$2.64115t_p$	$0.79043D_o$	A(0.3)/h
5	.5	.06250000	$12I_{zz}/h^3$	$8.16445t_p$	$0.25341D_o$	A(0.5)/h

where:

P(i) = location, defined as fraction of total thickness from centroid

I_{zz} = moment of inertia (input quantity IZZ on **R** command)

h = thickness (input quantity HEIGHT on **R** command)

t_p = pipe wall thickness (input quantity TKWALL on **R** command)

D_o = outside diameter (input quantity OD on **R** command)

A(i) = effective area based on width at location i (input quantities A(i) on **R** command)

Substituting in the values from Table 14.23–1, the ratio of the theoretical value to the computed value is 18/17, so that an error of about 6% is present for this case.

Note that the input quantities for the arbitrary cross–section (KEYOPT(6) = 4) are h, hL(1)(=A(–50)), hL(2)(=A(–30)), hL(3)(=A(0)), hL(4)(=A(30)), and hL(5)(=A(50)). It is recommended that the user try to satisfy equations (14.23–6) through (14.23–10) using this input option. These equations may be rewritten as:

$$A = 0.06250(A(-50) + A(50)) + 0.28935185(A(-30) + A(30)) + 0.29629630\ A(0) \tag{14.23–14}$$

$$I_1 = (0.0312500(-A(-50) + A(50)) + 0.008680556(-A(-30) + A(30)))h \tag{14.23–15}$$

$$I_2 = (0.01562500(A(-50) + A(50)) + 0.02604170(A(-30) + A(30)))h^2 \tag{14.23–16}$$

$$I_3 = (0.00781250(-A(-50) + A(50)) + 0.00781250(-A(-30) + A(30)))h^3 \tag{14.23–17}$$

$$I_4 = (.00390630(A(-50) + A(50)) + 0.00234375(A(-30) + A(30)))h^4 \tag{14.23–18}$$

Of course, $I_1 = I_3 = 0.0$ for symmetric sections.

Alternative to one of the above five equations, equation (14.23–13) can be used and rewritten as:

$$I_F = (0.031250(A(-50) + A(50)) + 0.08680554(A(-30) + A(30)))h \qquad (14.23\text{–}19)$$

Remember that I_2 is taken about the midpoint and that I_{zz} is taken about the centroid. The relationship between these two is:

$$I_{zz} = I_2 - A\,d^2 \qquad (14.23\text{–}20)$$

where:

$$d = \frac{I_1}{A} = h \sum_{i=1}^{5} H(i)\ L(i)\ P(I) \Big/ \sum_{i=1}^{5} H(i)\ L(i)$$

= 0.0 for symmetric cross–sections

14.23.3 Tangent Stiffness Matrix for Plasticity

The elastic stiffness, mass, and stress stiffness matrices are the same as those for a 2–D beam element (BEAM3). The tangent stiffness matrix for plasticity, however, is formed by numerical integration. This discussion of the tangent stiffness matrix as well as the Newton–Raphson restoring force of the next subsection has been generalized to include the effects of 3–D plastic beams. The general form of the tangent stiffness matrix for plasticity is:

$$[K_n] = \int_{vol} [B]^T\ [D_n]\ [B]\ d(vol) \qquad (14.23\text{–}21)$$

where: [B] = strain–displacement matrix

$[D_n]$ = elasto–plastic stress–strain matrix

This stiffness matrix for a general beam can also be written symbolically as:

$$[K] = \left[K^B\right] + \left[K^S\right] + \left[K^A\right] + \left[K^T\right] \qquad (14.23\text{–}22)$$

$[K^B]$ = bending contribution

$[K^S]$ = transverse shear contribution

$[K^A]$ = axial contribution

$[K^T]$ = torsional contribution

where the subscript n has been left off for convenience. As each of these four matrices use only one component of strain at a time, the integrand of equation (14.23–22) can be simplified from $[B]^T[D_n][B]$ to $\{B\}\ D_n \lfloor B \rfloor$. Each of these matrices will be subsequently described in detail.

1. Bending Contribution ([K^B]). The strain–displacement matrix for the bending stiffness matrix for bending about the z axis can be written as:

$$\lfloor B^B \rfloor = y \lfloor B^B_x \rfloor \tag{14.23–23}$$

where $\lfloor B^B_x \rfloor$ contains the terms of $\lfloor B^B \rfloor$ which are only a function of x (see Narayanaswami and Adelman(129)) :

$$\left\{B^B_x\right\} = \frac{1}{L^2 + 12\Phi} \left\{ \begin{array}{c} \frac{12x}{L} - 6 \\ 6x - 4L - \frac{12\Phi}{L} \\ -\left(\frac{12x}{L} - 6\right) \\ 6x - 2L + \frac{12\Phi}{L} \end{array} \right\} \tag{14.23–24}$$

where:

L = beam length

Φ = shear deflection constant (see Section 14.4)

The elasto–plastic stress–strain matrix has only one component relating the axial strain increment to the axial stress increment:

$$D_n = E_T \tag{14.23–25}$$

where E_T is the current tangent modulus from the stress–strain curve. Using these definitions equation (14.23–21) reduces to:

$$\left[K^B\right] = \int_{vol} \left\{B^B_x\right\} E_T y^2 \left\lfloor B^B_x \right\rfloor d(vol) \tag{14.23–26}$$

The numerical integration of equation (14.23–26) can be simplified by writing the integral as:

$$\left[K^B\right] = \int_L \left\{B^B_x\right\} \left[\int\!\!\int_{area} E_T y^2 \, d(area)\right] \left\lfloor B^B_x \right\rfloor dx \tag{14.23–27}$$

The integration along the length uses a two or three point Gauss rule while the integration through the cross–sectional area of the beam is dependent on the definition of the cross–section. For BEAM23, the integration through the thickness (area) is performed using the 5 point rule described in the previous section. Note that if the tangent modulus is the elastic modulus, $E_T = E$, the integration of equation (14.23–27) yields the exact linear bending stiffness matrix.

The Gaussian integration points along the length of the beam are interior, while the stress evaluation and, therefore, the tangent modulus evaluation is performed at the two ends and the middle of the beam for BEAM23. The value of the tangent modulus used at the integration point in evaluating equation (14.23–27) therefore assumes E_T is linearly distributed between the adjacent stress evaluation points.

2. Transverse Shear Contribution ($[K^S]$). The strain–displacement vector for the shear deflection matrix is (see Narayanaswami and Adelman(129)):

$$\{B^s\} = \frac{6\phi}{L^2 + 12\phi} \left\lfloor -\frac{2}{L} \quad -1 \quad \frac{2}{L} \quad -1 \right\rfloor^T \tag{14.23–28}$$

A plasticity tangent matrix for shear deflection is not required because either the shear strain component is ignored (BEAM23 and BEAM24) or where the shear strain component is computed (PIPE20), the plastic shear deflection is calculated with the initial–stiffness Newton–Raphson approach instead of the tangent stiffness approach. Therefore, since $D_n = G$ (the elastic shear modulus) equation (14.23–21) reduces to:

$$\left[K^S\right] = \int_{vol} \{B^s\}\ G\ \lfloor B^s \rfloor\ d(vol) \tag{14.23–29}$$

Integrating over the shear area explicitly yields:

$$\left[K^S\right] = GA_s \int_L \{B^s\}\ \lfloor B^s \rfloor\ dx \tag{14.23–30}$$

where A_s is the shear area (see Section 14.3). As $\lfloor B^s \rfloor$ is not a function of x in equation (14.23–28), the integral along the length of the beam in equation (14.23–30) could also be easily performed explicitly. However, it is numerically integrated with the two or three point Gauss rule along with the bending matrix $[K^B]$.

3. Axial Contribution ($[K^A]$). The strain–displacement vector for the axial contribution is:

$$\left\{B^A\right\} = \frac{1}{L} \left\lfloor 1 \quad -1 \right\rfloor^T \tag{14.23–31}$$

As with the bending matrix, $D_n = E_T$ and equation (14.23–21) becomes:

$$\left[K^A\right] = \int_{vol} \left\{B^A\right\} E_T \left\lfloor B^A \right\rfloor \, d(vol) \qquad (14.23–32)$$

which simplifies to:

$$\left[K^A\right] = \int_L \left\{B^A\right\} \left[\int\int_{area} E_T \, d(area) \right] \left\lfloor B^A \right\rfloor \, dx \qquad (14.23–33)$$

The numerical integration is performed using the same scheme as is used for the bending matrix.

4. Torsion Contribution ([K^T]). Torsional plasticity (PIPE20 only) is computed using the initial–stiffness Newton–Raphson approach. The elastic torsional matrix (needed only for the 3–D beams) is:

$$\left[K_T\right] = \frac{GJ}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (14.23–34)$$

14.23.4 Newton–Raphson Load Vector

The Newton–Raphson restoring force is:

$$\left\{F_n^{nr}\right\} = \int_{vol} [B]^T [D] \left\{\epsilon_n^{el}\right\} d(vol) \qquad (14.23–35)$$

where:

[D] = elastic stress–strain matrix

$\left\{\epsilon_n^{el}\right\}$ = elastic strain from the previous iteration

The load vector for a general beam can be written symbolically as:

$$\left\{F^{nr}\right\} = \left\{F_B^{nr}\right\} + \left\{F_S^{nr}\right\} + \left\{F_A^{nr}\right\} + \left\{F_T^{nr}\right\} \qquad (14.23–36)$$

where:

$\left\{F_B^{nr}\right\}$ = bending restoring force

$\left\{F_S^{nr}\right\}$ = shear deflection restoring force

$\left\{F_A^{nr}\right\}$ = axial restoring force

$\left\{F_T^{nr}\right\}$ = torsional restoring force

and where the subscript n has been left off for convenience. Again, as each of the four vectors use only one component of strain at a time, the integrand of equation

(14.23–35) can be simplified from $[B]^T[D]\left\{\epsilon_n^{el}\right\}$ to {B} D ϵ_n^{el}. The appropriate {B} vector for each contribution was given in the previous section. The following paragraphs describe D and ϵ_n^{el} for each of the contributing load vectors.

1. Bending Restoring Force $\left(\left\{F_B^{nr}\right\}\right)$. For this case, the elasticity matrix has only the axial component of stress and strain, therefore D = E, the elastic modulus. Equation (14.23–35) for the bending load vector is:

$$\left\{F_B^{nr}\right\} = E \int_L \left\{B_x^B\right\} \left[\iint_{area} y\ \epsilon^{el}\ d(area)\right] dx \tag{14.23–37}$$

The elastic axial strain is computed by:

$$\epsilon^{el} = \phi y + \epsilon^a - \epsilon^{th} - \epsilon^{pl} - \epsilon^{cr} - \epsilon^{sw} \tag{14.23–38}$$

where: ϕ = total curvature (defined below)

ϵ^a = total strain from the axial deformation (defined below)

ϵ^{th} = axial thermal strain

ϵ^{pl} = axial plastic strain

ϵ^{cr} = axial creep strain

ϵ^{sw} = axial swelling strain

The total curvature is:

$$\phi = \lfloor B_x^B \rfloor \left\{u^B\right\} \tag{14.23–39}$$

where $\{u^B\}$ is the bending components of the total nodal displacement vector {u}. The total strain from the axial deformation of the beam is:

$$\epsilon_a = \lfloor B^A \rfloor \left\{u^A\right\} = \frac{u_{XJ} - u_{XI}}{L} \tag{14.23–40}$$

where: $\{u^A\}$ = axial components for the total nodal displacement vector {u}

u_{XI}, u_{XJ} = axial displacement of nodes I and J

Equation (14.23–37) is integrated numerically using the same scheme outlined in the previous section. Again, since the nonlinear strain evaluation points for the plastic, creep and swelling strains are not at the same location as the integration points along the length of the beam, they are linearly interpolated.

2. Shear Deflection Restoring Force $\left(\left\{F_S^{nr}\right\}\right)$. The shear deflection contribution to the restoring force load vector uses D = G, the elastic shear modulus and the strain vector is simply:

$$\epsilon^{el} = \gamma_S \tag{14.23–41}$$

where γ_S is the average shear strain due to shear forces in the element:

$$\gamma_S = \lfloor B^S \rfloor \left\{u^B\right\} \tag{14.23–42}$$

The load vector is therefore:

$$\left\{F_S^{nr}\right\} = GA_S\gamma_S \int_L \left\{B^S\right\} dx \tag{14.23–43}$$

3. Axial Restoring Force $\left(\left\{F_A^{nr}\right\}\right)$. The axial load vector uses the axial elastic strain defined in equation (14.23–38) for which the load vector integral reduces to:

$$\left\{F_A^{nr}\right\} = E \int_L \left\{B^A\right\} \left[\int_{area} \epsilon^{el}\ d(area)\right] dx \tag{14.23–44}$$

4. Torsional Restoring Force $\left(\left\{F_T^{nr}\right\}\right)$. The torsional restoring force load vector (needed only for 3–D beams) uses D = G, the elastic shear modulus and the strain vector is:

$$\gamma_T^{el} = \gamma - \gamma^{pl} - \gamma^{cr} \tag{14.23–45}$$

where: γ_T^{el} = elastic torsional strain
γ = total torsional strain (defined below)
γ^{pl} = plastic shear strain
γ^{cr} = creep shear strain

The total torsional shear strain is defined by:

$$\gamma = \frac{(\theta_{XJ} - \theta_{XI})\ \rho}{L} \tag{14.23–46}$$

where: θ_{XI}, θ_{XJ} = total torsional rotations from {u} for nodes I, J, respectively.
ϱ = $\sqrt{\left(y^2 + z^2\right)}$ = distance from shear center

The load vector is:

$$\left\{F_T^{nr}\right\} = G \int_L \left\{B^T\right\} \left[\iint_{area} \rho^2 \gamma_T^{el} \, d(area) \right] dx \qquad (14.23–47)$$

where: $\{B^T\}$ = strain–displacement vector for torsion (same as axial equation (14.23–31)

14.23.5 Stress and Strain Calculation

The modified total axial strain at any point in the beam is given by:

$$\epsilon_n' = \phi^a y + \epsilon^a - \epsilon_n^{th} - \epsilon_{n-1}^{pl} - \epsilon_{n-1}^{sw} \qquad (14.23–48)$$

where: ϕ^a = adjusted total curvature

ϵ^a = adjusted total strain from the axial deformation

ϵ_n^{th} = axial thermal strain

ϵ_{n-1}^{pl} = axial plastic strain from the previous substep

ϵ_{n-1}^{cr} = axial creep strain from the previous substep

ϵ_{n-1}^{sw} = axial swelling strain from the previous substep

The total curvature and axial deformation strains are adjusted to account for the applied pressure and acceleration load vector terms. The adjusted curvature is:

$$\phi^a = \phi - \phi^{pa} \qquad (14.23–49)$$

where: ϕ = $[B^B]\{u^B\}$ = total curvature

ϕ^{pa} = pressure and acceleration contribution to the curvature

ϕ^{pa} is readily calculated through:

$$\phi^{pa} = \frac{M^{pa}}{EI} \qquad (14.23–50)$$

M^{pa} is extracted from the moment terms of the applied load vector (in element coordinates):

$$\left\{F^{pa}\right\} = \left\{F^{pr}\right\} + \left\{F^{ac}\right\} \qquad (14.23–51)$$

$\{F^{pr}\}$ is given in Section 14.3 and $\{F^{ac}\}$ is given in Section 17.1. The value used depends on the location of the evaluation point:

$$M^{pa} = \begin{cases} M_I^{pa} & \text{, if evaluation is at end I} \\ \frac{1}{4}\left(M_I^{pa} - M_J^{pa}\right) & \text{, if evaluation is at the middle} \\ M_J^{pa} & \text{, if evaluation is at end J} \end{cases} \qquad (14.23–52)$$

The adjusted axial deformation strain is:

$$\epsilon^a = \epsilon - \epsilon^{pa} \qquad (14.23–53)$$

where: ϵ = $[B^A]\{u^A\}$ = total axial deformation strain

ϵ^{pa} = pressure and acceleration contribution to the axial deformation strain

ε^{pa} is computed using:

$$\epsilon^{pa} = \frac{F_x^{pa}}{EA} \qquad (14.23–54)$$

where F_x^{pa} is calculated in a similar manner to M^{pa}.

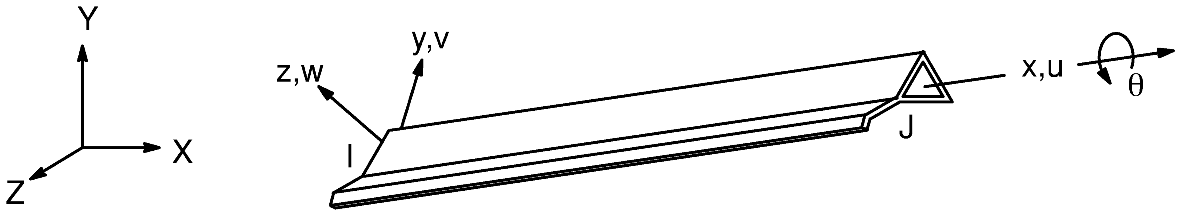
From the modified total strain (equation (14.23–48)) the plastic strain increment can be computed (see Section 4.1), leaving the elastic strain as:

$$\epsilon^{el} = \epsilon' - \Delta\epsilon^{pl} \qquad (14.23–55)$$

where $\Delta\varepsilon^{pl}$ is the plastic strain increment. The stress at this point in the beam is then:

$$\sigma = E\,\epsilon^{el} \qquad (14.23–56)$$

14.24 BEAM24 — 3–D Thin–Walled Beam



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.2.2–1), (12.2.2–2), (12.2.2–3), and (12.2.2–4)	Locations on the cross–section are user defined. No integration points are used along the length for elastic matrix. Same as Newton–Raphson load vector for tangent matrix with plasticity.
Mass Matrix	Same as stress stiffness matrix.	None
Stress Stiffness Matrix	Equations (12.2.2–2) and (12.2.2–3)	None
Thermal Load Vector	Equations (12.2.2–1), (12.2.2–2) and (12.2.2–3)	None
Pressure Load Vector	Equations (12.2.2–2) and (12.2.2–3)	None

Matrix or Vector	Shape Functions	Integration Points
Newton–Raphson Load Vector	Same as thermal load vector	2 along the length and 2 in each segment
Stress Evaluation	Same as thermal load vector	The user defined points on the cross–section are used at each end of the element

Load Type	Distribution
Element Temperature	Bilinear across cross–section and linear along length. See section entitled "Temperature Distribution Across Cross–Section" for more details.
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Linear along length. The pressure is assumed to act along the element x axis.

References: Oden(27), Galambos(13), Kollbrunner(21)

14.24.1 Assumptions and Restrictions

1. The wall thickness is small in comparison to the overall cross–section dimensions (thin–walled theory).
2. The cross–section does not change shape under deformation.
3. St. Venant's theory of torsion governs the torsional behavior. The cross–section is therefore assumed free to warp.
4. Only axial stresses and strains are used in determining the nonlinear material effects. Shear and torsional components are neglected.

14.24.2 Other Applicable Sections

Section 13.1 describes integration point locations. Section 14.4 has an elastic beam element stiffness and mass matrix explicitly written out. Section 14.23 defines the tangent matrix with plasticity, the Newton–Raphson load vector and the stress and strain computation.

14.24.3 Temperature Distribution Across Cross–Section

As stated above, the temperature is assumed to vary bilinearly across the cross–section (as well as along the length). Specifically,

$$\begin{aligned} T(x,y,z) &= \left(T_I + y\left(\frac{\partial T}{\partial y}\right)_I + z\left(\frac{\partial T}{\partial z}\right)_I\right)\left(1 - \frac{x}{L}\right) \\ &+ \left(T_J + y\left(\frac{\partial T}{\partial y}\right)_J + z\left(\frac{\partial T}{\partial z}\right)_J\right)\frac{x}{L} \end{aligned} \tag{14.24–1}$$

where: T(x,y,z) = temperature at integration point located at x, y, z

x, y, z = location of point in reference coordinate system (coordinate system defined by the nodes

T_i = temperature at node i (input quantities T1, T4 on **BFE** command)

$\left(\frac{\partial T}{\partial y}\right), \left(\frac{\partial T}{\partial z}\right)$ = temperature gradients defined below

L = length

The gradients are:

$$\left(\frac{\partial T}{\partial y}\right)_i = T_{yi} - T_i \tag{14.24–2}$$

$$\left(\frac{\partial T}{\partial z}\right)_i = T_{zi} - T_i \tag{14.24–3}$$

where: T_{yi} = temperature at one unit from the node i parallel to reference y axis (input quantities T2, T5 on **BFE** command)

T_{zi} = temperature at one unit from the node i parallel to reference z axis (input quantities T3, T6 on **BFE** command)

14.24.4 Calculation of Cross–Section Section Properties

The cross–section constants are determined by numerical integration, with the integration points (segment points) input by the user. The area of the kth segment (A_k) is:

$$A_k = \ell_k t_k \tag{14.24–4}$$

where: ℓ_k = length of segment k (input indirectly as Y and Z on **R** commands)

t_k = thickness of segment k (input as TK on **R** commands)

The total cross–section area is therefore

$$A = \sum A_k \qquad (14.24–5)$$

where: $\sum$ = implies summation over all the segments

The first moments of area with respect to the reference axes used to input the cross–section are

$$q_y = \frac{1}{2}\sum \left(z_i + z_j\right) A_k \qquad (14.24–6)$$

$$q_z = \frac{1}{2}\sum \left(y_i + y_j\right) A_k \qquad (14.24–7)$$

where: y_i, z_i = input coordinate locations at beginning of segment k

y_j, z_j = input coordinate locations at end of segment k

The centroidal location with respect to the origin of the reference axes is therefore

$$y_c = q_z/A \qquad (14.24–8)$$

$$z_c = q_y/A \qquad (14.24–9)$$

where: y_c, z_c = coordinates of the centroid

The moments of inertia about axes parallel to the reference axes but whose origin is at the centroid (y_c, z_c) can be computed by:

$$I_y = \frac{1}{3}\sum \left(\bar{z}_i^2 + \bar{z}_i\bar{z}_j + \bar{z}_j^2\right) A_k \qquad (14.24–10)$$

$$I_z = \frac{1}{3}\sum \left(\bar{y}_i^2 + \bar{y}_i\bar{y}_j + \bar{y}_j^2\right) A_k \qquad (14.24–11)$$

where: $\bar{y}$ = $y - y_c$

$\bar{z}$ = $z - z_c$

The product moment of inertia is

$$I_{yz} = \frac{1}{3}\sum \left(\bar{y}_i\bar{z}_i + \bar{y}_j\bar{z}_j\right) A_k + \frac{1}{6}\sum \left(\bar{y}_i\bar{z}_j + \bar{y}_j\bar{z}_i\right) A_k \qquad (14.24–12)$$

Note that these are simply Simpson’s integration rule applied to the standard formulas. The principal moments of inertia are at an angle θ_p with respect to the reference coordinate system Figure 14.24–1, where θ_p (output quantity THETAP) is calculated from:

$$\theta_p = \frac{1}{2} \tan^{-1}\left(\frac{2I_{yz}}{I_z - I_y}\right) \qquad (14.24–13)$$

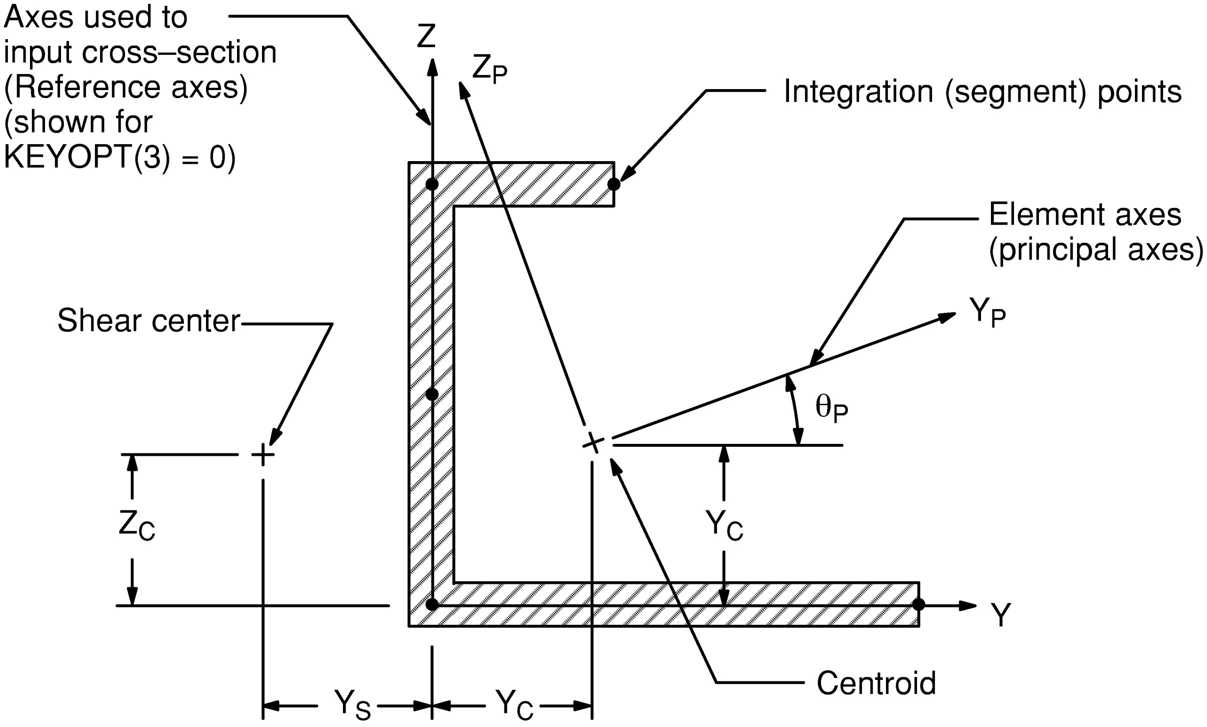

Axes used to input cross–section (Reference axes) (shown for KEYOPT(3) = 0)
Z
Z_P
Integration (segment) points
Element axes (principal axes)
Y_P
Shear center
θ_P
Z_C
Y_C
Y
Centroid
Y_S
Y_C

Figure 14.24–1 Cross–Section Input and Principal Axes

The principal moments of inertia with respect to the element coordinate system are therefore:

$$I_{yp} = \frac{1}{2}\left(I_y + I_z\right) + \frac{1}{2}\left(I_y - I_z\right)\cos\left(2\theta_p\right) - I_{yz}\sin\left(2\theta_p\right) \qquad (14.24–14)$$

and

$$I_{zp} = I_y + I_z - I_{yp} \qquad (14.24–15)$$

where: I_{yp} = output quantity IYP
I_{zp} = output quantity IZP

The torsional constant for a thin–walled beam of either open or closed (single cell only) cross–section is

$$J = \frac{4\ A_o^2}{\sum\limits^{c} \frac{\ell_k}{t_k}} + \frac{1}{3}\sum^{d} \ell_k t_k^3 \qquad (14.24–16)$$

where: J = torsional constant (output quantity J)

A_o = area enclosed by the centerline of the closed part of the cross–section = $\left| \frac{1}{2} \sum^{c} \left(z_i + z_j\right) \left(y_j - y_i\right) \right|$

$\sum^{c}$ = summation over the segments enclosing the area only

$\sum^{d}$ = summation over the remaining segments $\Big($not included in $\sum^{c}\Big)$

The shear center location with respect to the origin of the reference axes (Figure 14.24–1) is:

$$y_s = y_c + \frac{I_{yz} I_{\omega y} - I_z I_{\omega z}}{I_{yz}^2 - I_y I_z} \tag{14.24–17}$$

$$z_s = z_c + \frac{I_{yz} I_{\omega z} - I_y I_{\omega y}}{I_{yz}^2 - I_y I_z} \tag{14.24–18}$$

where: y_s, z_s = output quantity SHEAR CENTER

The sectorial products of inertia used to develop the above expressions are:

$$I_{\omega y} = \frac{1}{3} \sum \left(\omega_i \bar{y}_i + \omega_j \bar{y}_j\right) A_k + \frac{1}{6} \sum \left(\omega_i \bar{y}_j + \omega_j \bar{y}_i\right) A_k \tag{14.24–19}$$

$$I_{\omega z} = \frac{1}{3} \sum \left(\omega_i \bar{z}_i + \omega_j \bar{z}_j\right) A_k + \frac{1}{6} \sum \left(\omega_i \bar{z}_j + \omega_j \bar{z}_i\right) A_k \tag{14.24–20}$$

The sectorial products of inertia are analogous to the moments of inertia, except that one of the coordinates in the definition (such as equation (14.24–12)) is replaced with the sectorial coordinate ω. The sectorial coordinate of a point p on the cross–section is defined as

$$\omega_p = \int_o^s h \, ds \tag{14.24–21}$$

where h is the distance from some reference point (here the centroid) to the cross–section centerline and s is the distance along the centerline from an arbitrary starting point to the point p. h is considered positive when the cross–section is being transversed in the counterclockwise direction with respect to the centroid. Note that the absolute value of the sectorial coordinate is twice the area enclosed by the sector indicated in Figure 14.24–2.

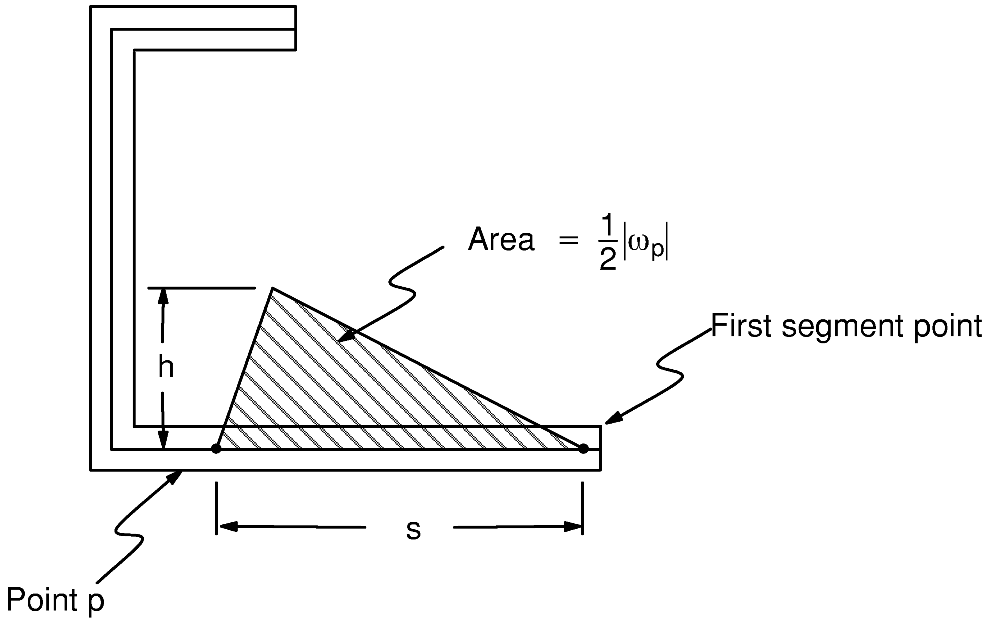


Figure 14.24–2 Definition of Sectorial Coordinate

Equation (14.24–21) can be rewritten using Simpson's integration rule as

$$\omega_p = \sum_1^s \bar{y}_i \left(\bar{z}_j - \bar{z}_i\right) - \bar{z}_i \left(\bar{y}_j - \bar{y}_i\right) \qquad (14.24\text{–}22)$$

where: $\sum^s$ = summation from the first segment input to the first segment containing point p.

If the segment is part of a closed section or cell, the sectorial coordinate is defined as

$$\omega_p = \sum_1^s \bar{y}_i \left(\bar{z}_j - \bar{z}_i\right) - \bar{z}_i \left(\bar{y}_j - \bar{y}_i\right) - \frac{2\,A_o}{\sum^c \frac{\ell_k}{t_k}} \frac{\ell_k}{t_k} \qquad (14.24\text{–}23)$$

The warping moment of inertia (output quantity IW) is computed as:

$$I_\omega = \frac{1}{2} \sum \left(\omega_{ni}^2 + \omega_{ni}\omega_{nj} + \omega_{nj}^2\right) A_k \qquad (14.24\text{–}24)$$

where the normalized sectorial coordinates ω_{ni} and ω_{nj} are defined in general as ω_{np} below. As BEAM24 ignores warping torsion, I_ω is not used in the stiffness formulation but it is calculated and printed for the user's convenience. A normalized sectorial coordinate is defined to be

$$\omega_{np} = \frac{1}{2A} \sum \left(\omega_{oi} + \omega_{oj}\right) A_k - \omega_{op} \tag{14.24–25}$$

where: ω_{op} = sectorial coordinate with respect to the shear center for integration point p

ω_{op} is defined as with the expressions for the sectorial coordinates equations (14.24–22) and (14.24–23) except that $\bar{y}$ and $\bar{z}$ are replaced by $\tilde{y}$ and $\tilde{z}$. These are defined by:

$$\tilde{y} = y - y_s \tag{14.24–26}$$

$$\tilde{z} = z - z_s \tag{14.24–27}$$

Thus, these two equations have been written in terms of the shear center instead of the centroid.

The location of the reference coordinate system affects the line of application of nodal and pressure loadings as well as the member force printout directions. By default, the reference coordinate system is coincident with the y–z coordinate system used to input the cross–section geometry (Figure 14.24–3a). If KEYOPT(3) = 1, the reference coordinate system x axis is coincident with the centroidal line while the reference y and z axes remain parallel to the input y–z axes (Figure 14.24–3b). The shear center and centroidal locations with respect to this coordinate system are

$$\begin{aligned} y_s &= y_{s,o} - y_{c,o} \\ z_s &= z_{s,o} - z_{c,o} \end{aligned} \tag{14.24–28}$$

and

$$\begin{aligned} y_c &= 0 \\ z_c &= 0 \end{aligned} \tag{14.24–29}$$

where the subscript o on the shear center and centroid on the right–hand side of equation (14.24–28) refers to definitions with respect to the input coordinate systems in equations (14.24–8), (14.24–9), (14.24–17), and (14.24–18). Likewise, if KEYOPT(3) = 2, the reference x axis is coincident with the shear centerline and the locations of the centroid and shear center are determined to be (Figure 14.24–3c).

$$\begin{aligned} y_c &= y_{c,o} - y_{s,o} \\ z_c &= z_{c,o} - z_{s,o} \end{aligned} \tag{14.24–30}$$

and

$$\begin{aligned} y_s &= 0 \\ z_s &= 0 \end{aligned} \qquad (14.24\text{–}31)$$

14.24.5 Offset Transformation

The stiffness matrix for a beam element (Section 14.4) is formulated with respect to the element coordinate (principal axis) system for the bending and axial behavior and the shear center for torsional behavior. The stiffness matrix and load vector in this system are $[K_\ell]$ and $\{F_\ell\}$. In general, the reference coordinate system in BEAM24 is non–coincident with the element system, hence a transformation between the coordinate systems is necessary. The transformation is composed of a rotational part that accounts for the angle between the reference y and z axes and the element y and z axes (principal axes) and a translational part that accounts for the offsets of the centroid and shear center. The rotational part has the form

$$[R] = \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix} \qquad (14.24\text{–}32)$$

where:

$$[\lambda] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_p & \sin\theta_p \\ 0 & -\sin\theta_p & \cos\theta_p \end{bmatrix} \qquad (14.24\text{–}33)$$

and θ_p is the angle defined in equation (14.24–13). The translational part is

$$[T] = \begin{bmatrix} I & T_1 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & T_2 \\ 0 & 0 & 0 & I \end{bmatrix} \qquad (14.24\text{–}34)$$

where [I] is the 3 x 3 identity matrix and [T$_i$] is

$$\left[T_i\right] = \begin{bmatrix} 0 & z_c & y_c \\ -z_s & 0 & x_i \\ y_s & -x_i & 0 \end{bmatrix} \tag{14.24–35}$$

in which y_c, z_c, y_s, and z_s are centroid and shear center locations with respect to the element coordinate system and x_i is the offset in the element x direction for end i. The material to element transformation matrix is then

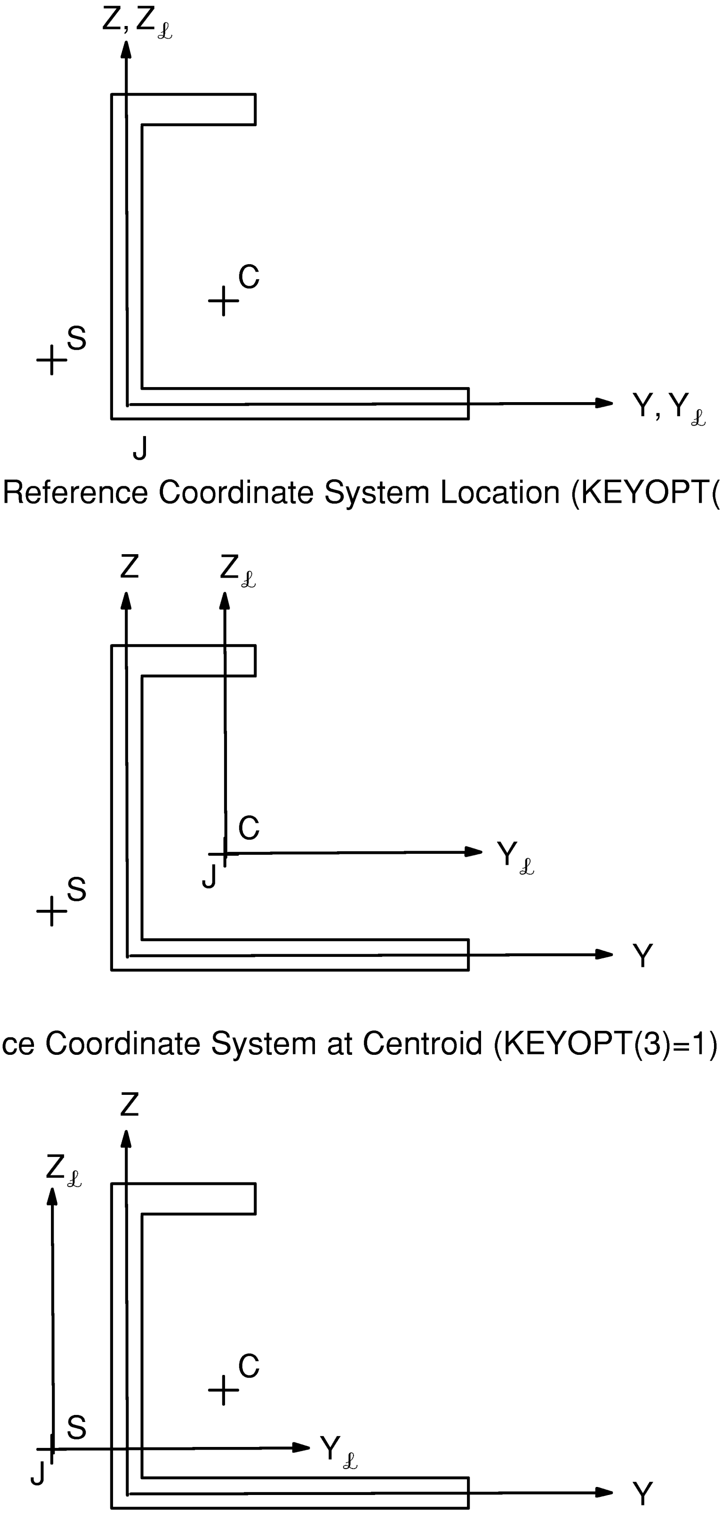
$$\left[O_f\right] = [R][T] \tag{14.24–36}$$

The transformation matrix [O$_f$] is used to transform the element matrices and load vector from the element to the reference coordinate system

$$\left[K_\ell'\right] = \left[O_f\right]^T\left[K_\ell\right]\left[O_f\right] \tag{14.24–37}$$

and

$$\left\{F_\ell'\right\} = \left[O_f\right]^T\left\{F_\ell\right\} \tag{14.24–38}$$



(a) Default Reference Coordinate System Location (KEYOPT(3)=0)

(b) Reference Coordinate System at Centroid (KEYOPT(3)=1)

(c) Reference Coordinate System at Shear Center (KEYOPT(3)=2)

Figure 14.24–3   Reference Coordinate System

The standard local to global transformation (Section 14.4) can then be used to calculate the element matrices and load vector in the global system:

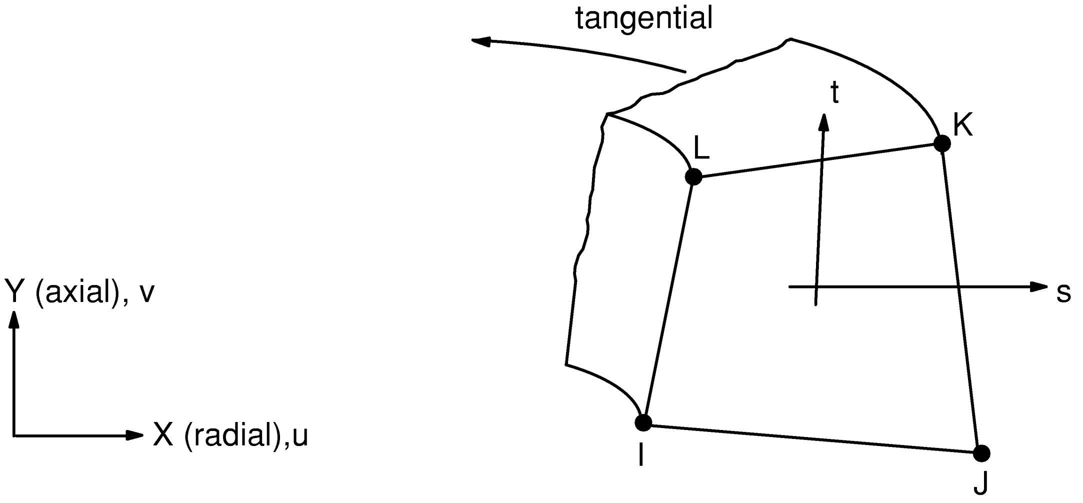
$$[K_e] = \left[T_R\right]^T \left[K_\ell'\right] \left[T_R\right] \tag{14.24–39}$$

and

$$\{F_e\} = \left[T_R\right]^T \left\{F_\ell'\right\} \tag{14.24–40}$$

The mass and stress stiffening matrices are similarly transformed. The material to element transformation (equation (14.24–37)) for the mass matrix, however, neglects the shear center terms y_s and z_s as the center of mass coincides with the centroid, not the shear center.

14.25 PLANE25 — 4–Node Axisymmetric–Harmonic Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.7.5–1), (12.7.5–2), and (12.7.5–3) or if modified extra shape functions are included (KEYOPT(2) = 0) and element has 4 unique nodes. Equations (12.7.6–1), (12.7.6–2), and (12.7.6–3)	2 x 2
Stiffness Matrix	Triangle	Equations (12.7.1–1), (12.7.1–2), and (12.7.1–3)	3

Matrix or Vector	Geometry	Shape Functions	Integration Points
Mass Matrix	Quad	Equation (12.6.5–1), (12.6.5–2), and (12.6.5–3)	2 x 2
	Triangle	Equation (12.6.1–1), (12.6.1–2), and (12.6.1–3)	3
Stress Stiffness Matrix	Quad	Equation (12.7.5–1), (12.7.5–2), and (12.7.5–3)	2 x 2
	Triangle	Equation (12.7.1–1), (12.7.1–2), and (12.7.1–3)	3
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stress stiffness matrix, specialized to the surface		2

Load Type	Distribution
Element Temperature	Bilinear across element, harmonic around circumference
Nodal Temperature	Bilinear across element, harmonic around circumference
Pressure	Linear along each face, harmonic around circumference

Reference: Wilson(38), Zienkiewicz(39), Taylor(49)

14.25.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.25.2 Assumptions and Restrictions

The material properties are assumed to be constant around the entire circumference, regardless of temperature dependent material properties or loading. For MODE > 0, the extreme values for combined stresses are obtained by computing these stresses at every 10/MODE degrees and selecting the extreme values.

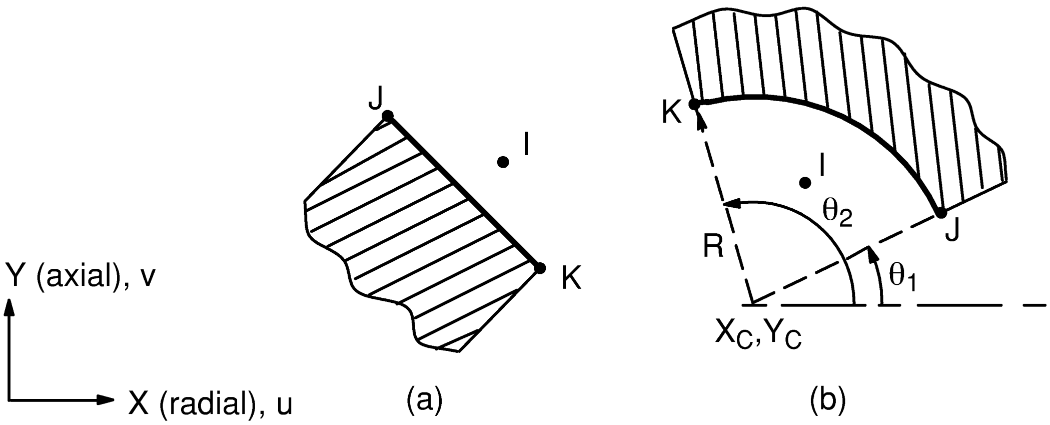
14.25.3 Use of Temperature

In general, temperatures have two effects on a stress analysis:

1. Temperature dependent material properties.
2. Thermal expansion

In the case of MODE = 0, there is no conflict between these two effects. However, if MODE > 0, questions arise. As stated in the assumptions, the material properties may not vary around the circumference, regardless of the temperature. That is, one side cannot be soft and the other side hard. The input temperature for MODE > 0 varies sinusoidally around the circumference. As no other temperatures are available to the element, the material properties are evaluated at T_{ref} (input on **TREF** command). The input temperature can therefore be used to model thermal bending. An approximate application of this would be a chimney subjected to solar heating on one side only. A variant on this basic procedure is provided by the temperature KEYOPT (KEYOPT(3) for PLANE25). This variant provides that the input temperatures be used only for material property evaluation rather than for thermal bending. This second case requires that α_x, α_y, and α_z (input on **MP** commands) all be input as zero. An application of the latter case is a chimney, which is very hot at the bottom and relatively cool at the top, subjected to a wind load.

14.26 CONTAC26 — 2–D Point–to–Ground Contact



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (nodes may be coincident)	None

14.26.1 Operation of Element

The figure above shows two different ways that CONTAC26 can be used. In each, the dark line connecting nodes J and K is presumed to be the surface where resistance begins (the contact surface). When node I is located in the shaded area, the normal force is negative and the element responds as a linear spring in the normal direction. If contact is maintained and friction is defined (KEYOPT(1) = 1 and $\mu > 0.0$, where μ = input as MU on **MP** command), both normal forces and sticking/sliding forces are active. The sticking/sliding forces act in a direction parallel to the contact surface. If contact is broken (node I is outside of the shaded area), no normal or parallel forces are transmitted.

If friction is not defined (frictionless contact), the functioning of the element is not path dependent. The element operates on the final position of the node I relative to the contact surface and not on the loading history. Thus, the element makes no distinction whether the final location I is reached via path (A–B) or by path (A–C–D–B) as shown in Figure 14.26–1.

Figure 14.26–2 shows the force–deflection relationships for this element in the normal direction. It may be seen from this figure that the element is nonlinear and, therefore, requires an iterative solution. For the straight line contact surface, the normal force acts perpendicular to the surface, and for the circular arc contact surface, the normal force is in the radial direction.

When friction is defined, the functioning of the element is irreversible. As shown in Figure 14.26–3, an elastic–Coulomb friction model acts in the tangential direction. For the straight line contact surface, the tangential direction is parallel to the surface; and for the circular arc contact surface, the tangential direction follows the arc. Corresponding element sticking/sliding forces act in the tangential direction. The usual Coulomb friction applies:

$$|F_s| \leq -\mu F_n \tag{14.26–1}$$

where: F_s = the tangential force (output quantity FS)

F_n = the normal force (output quantity FN)

μ = coefficient of friction (input as MU on **MP** command)

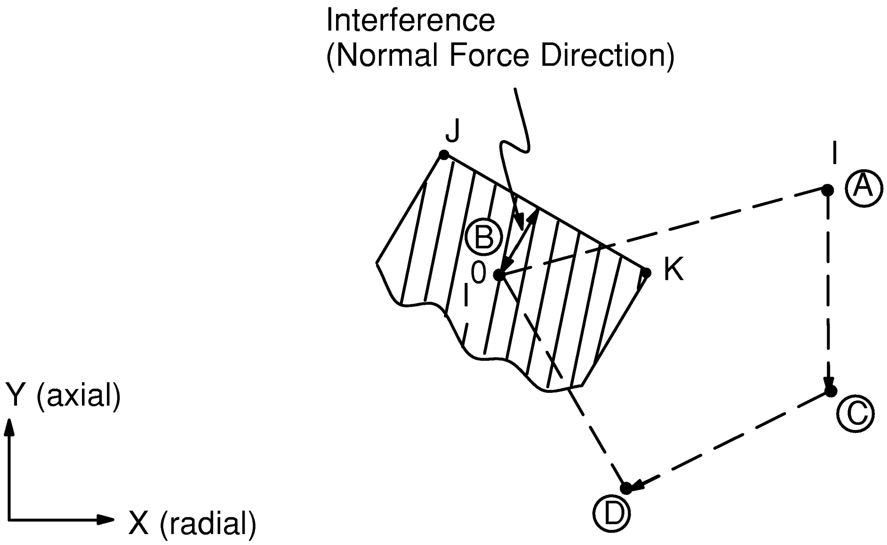


Figure 14.26–1 Element Behavior for Two Different Displacement Paths: (A–B) and A–C–D–B)

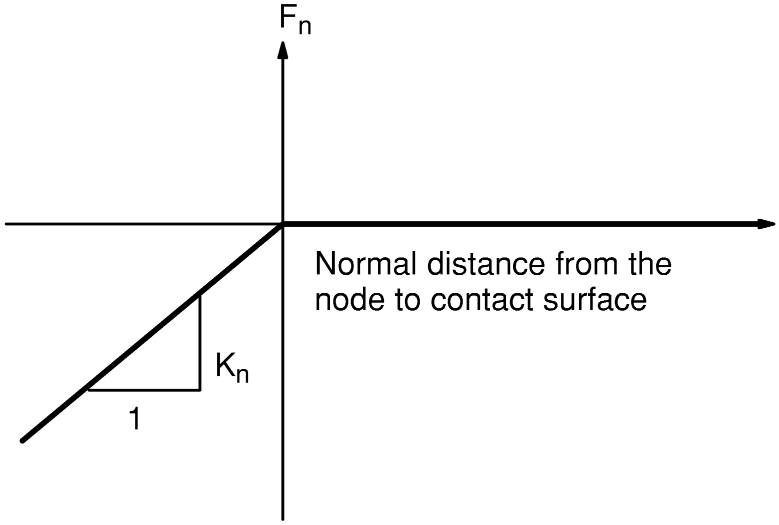


Figure 14.26–2 Force–Deflection Relationship in the Normal Direction

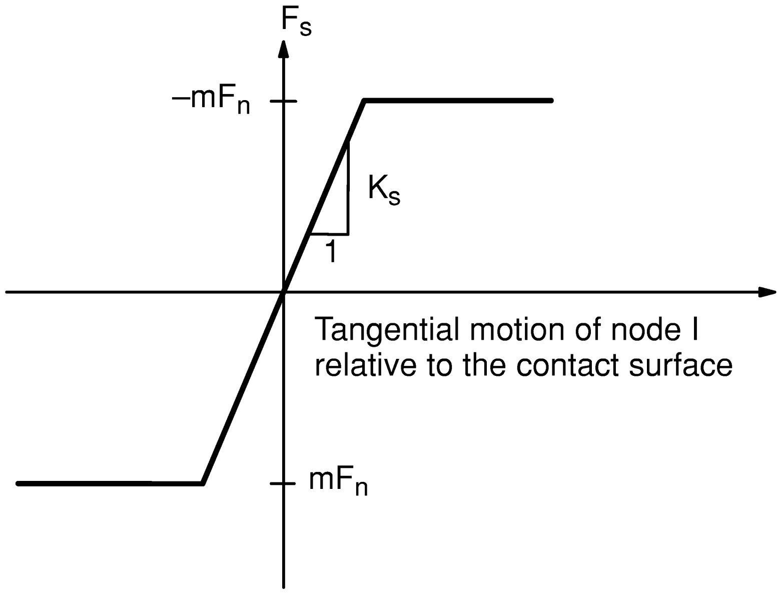


Figure 14.26–3 Force–Deflection Relationship in the Tangential Direction (No Unloading Is Indicated)

14.26.2 Element Matrices

CONTAC26 may have one of two conditions: node I is inside the shaded area or node I is outside the shaded area. The resulting element stiffness matrix in the contact surface coordinate system (x is tangential, y is normal) is:

$$[K_\ell] = \begin{bmatrix} K_s & 0 \\ 0 & K_n \end{bmatrix} \tag{14.26–2}$$

where: $K_n = \begin{cases} 0 \text{ if open } (I_s = 3) \\ k^n \text{ if closed } (I_s \le 2) \end{cases}$

$K_s = \begin{cases} 0 \text{ if not stuck } (I_s = +2, -2, \text{ or } 3) \\ k^s \text{ if stuck } (I_s = 1) \end{cases}$

k^n = normal stiffness (input as STIFN on **R** command)

k^s = sliding stiffness (input as STIFS on **R** command)

I_s = output quantity STAT

The Newton–Raphson load vector is:

$$\{F_\ell^{nr}\} = \begin{Bmatrix} F_s \\ F_n \end{Bmatrix} \tag{14.26–3}$$

where: F_s = force tangential (stick/sliding) to the contact surface

F_n = force normal to the contact surface (from the previous iteration)

14.26.3 Stress Pass

The stress pass output quantities are shown in Figure 4.26–2 of the *ANSYS Elements Reference* and reflect the latest possible information concerning the gap status. Therefore, for nonconverged iterations, it may not agree with reaction forces which are based on the previously calculated stiffness matrix.

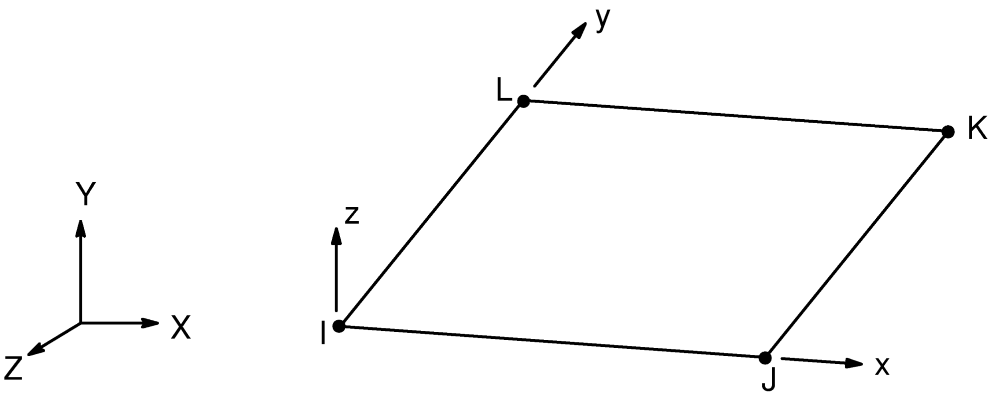
14.27 MATRIX27 — Stiffness, Damping, or Mass Matrix

Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix (KEYOPT(3) = 4)	None	None
Mass Matrix (KEYOPT(3) = 2)	None	None
Damping Matrix (KEYOPT(3) = 5)	None	None

14.27.1 Assumptions and Restrictions

All MATRIX27 matrices should normally be positive definite or positive semidefinite (see Section 13.5 for definition) in order to be valid structural matrices. The only exception to this occurs when other (positive definite) matrices dominate the involved DOFs and/or sufficient DOFs are removed by way of imposed constraints, so that the total (structure) matrix is positive definite.

14.28 SHELL28 — Shear/Twist Panel



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (see reference)	None
Mass Matrix	None (one–sixth of the mass of each of the IJK, JKL, KLI, and LIJ subtriangles is put at the nodes)	None
Stress Stiffness Matrix	No shape functions are used. Rather, the stress stiffness matrix is developed from the two diagonal forces used as spars	None

Reference: Garvey(116)

14.28.1 Assumptions and Restrictions

This element is based directly on the reference by Garvey(116). It uses the idea that shear effects can be represented by a uniform shear flow and nodal forces in the direction of the diagonals. The element only resists shear stress; direct stresses will not be resisted.

The shear panel assumes that only shearing stresses are present along the element edges. Similarly, the twist panel assumes only twisting moment, and no direct moment.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.28.2 Commentary

The element loses validity when used in shapes other than rectangular. For non–rectangular cases, the resulting shear stress is non–uniform, so that the patch test cannot be satisfied. Consider a rectangular element under uniform shear:

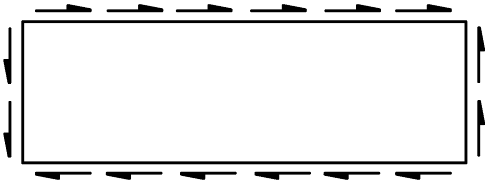


Figure 14.28–1 Uniform Shear on Rectangular Element

Then, add a fictional cut at 45° to break the rectangular element into two trapezoidal regions (elements):

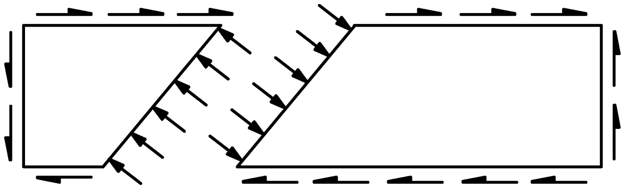


Figure 14.28–2 Uniform Shear on Separated Rectangular Element

As can be seen, shear forces as well as normal forces are required to hold each part of the rectangle in equilibrium for the case of "uniform shear." The above discussion for trapezoids can be extended to parallelograms. If the presumption of uniform shear stress is dropped, it is possible to hold the parts in equilibrium using only shear stresses along all edges of the quadrilateral (the presumption used by Garvey) but a truly uniform shear state will not exist.

14.28.3 Output Terms

The stresses are also computed using the approach of Garvey(116).

When all four nodes lie in a flat plane, the shear flows are related to the nodal forces by:

$$S^{fl}_{IJ} = \frac{F_{JI} - F_{IJ}}{\ell_{IJ}} \tag{14.28–1}$$

where: S^{fl}_{IJ} = shear flow along edge IJ (output quantity SFLIJ)

F_{JI} = force at node I from node J (output quantity FJI)

F_{IJ} = force at node J from node I (output quantity FIJ)

ℓ_{IJ} = length of edge I–J

The forces in the element z direction (output quantities FZI, FZJ, FZK, FZL) are zero for the flat case. When the flat element is also rectangular, all shear flows are the same. The stresses are:

$$\sigma_{xy} = \frac{S_{IJ}^{fl}}{t} \tag{14.28–2}$$

where: σ_{xy} = shear stress (output quantity SXY)

t = thickness (input as THCK on **R** command)

The logic to compute the results for the cases where all four nodes do not lie in a flat plane or the element is non–rectangular is more complicated and is not developed here.

The margin of safety calculation is:

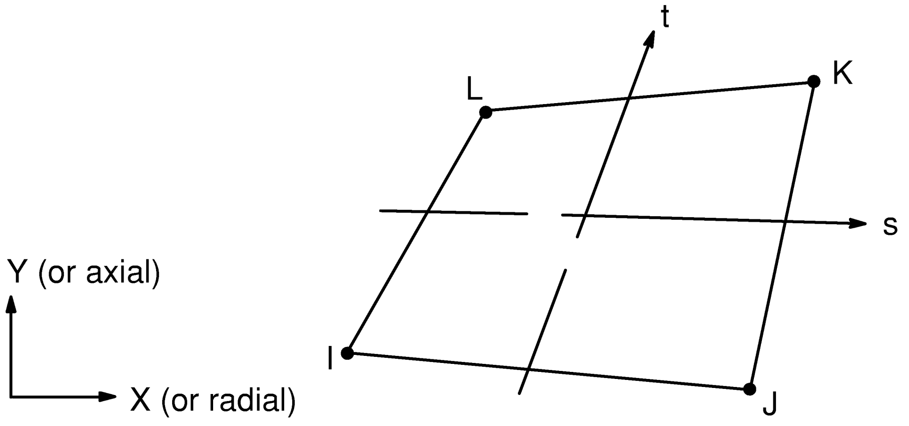
$$M_s = \begin{cases} \dfrac{\sigma_{xy}^{u}}{\sigma_{xy}^{m}} - 1.0 \text{ if both } \sigma_{xy}^{m} \text{ and } \sigma_{xy}^{u} \neq 0 \\ \\ 0.0 \text{ if either } \sigma_{xy}^{m} \text{ or } \sigma_{xy}^{u} = 0 \end{cases} \tag{14.28–3}$$

where: M_s = margin of safety (output quantity SMARGN)

σ_{xy}^{m} = maximum nodal shear stress (output quantity SXY(MAX))

σ_{xy}^{u} = maximum allowable shear stress (input as SULT on **R** command)

14.29 FLUID29 — 2–D Acoustic Fluid

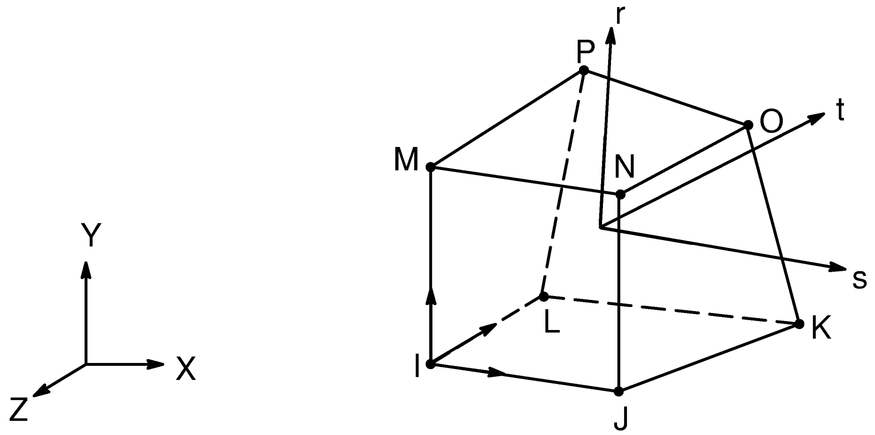


Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness Matrix	Equation (12.6.5–19)	2 x 2
Coupling Stiffness Matrix (fluid–structure interface)	Equation (12.6.5–1), (12.6.5–2), and (12.6.5–19) specialized to the interface	2
Fluid Mass Matrix	Equation (12.6.5–19)	2 x 2
Coupling Mass Matrix (fluid–structure interface)	Same as coupling stiffness matrix	2
Fluid Damping Matrix (fluid at fluid–structure interface)	Same as coupling stiffness matrix	2

14.29.1 Other Applicable Sections

Chapter 8 describes the derivation of acoustic element matrices and load vectors.

14.30 FLUID30 — 3–D Acoustic Fluid

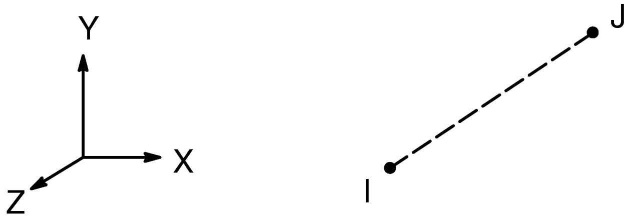


Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness Matrix	Equation (12.8.18–19)	2 x 2 x 2
Coupling Stiffness Matrix (fluid–structure interface)	Equation (12.8.18–1), (12.8.18–2), (12.8.18–3), and (12.8.18–19) specialized to the interface	2 x 2
Fluid Mass Matrix	Equation (12.8.18–19)	2 x 2 x 2
Coupling Mass Matrix (fluid–structure interface)	Same as coupling stiffness matrix	2 x 2
Fluid Damping Matrix (fluid at fluid–structure interface)	No shape functions are used. Instead, the area associated with each node at the interface is computed for the damping to act upon.	None

14.30.1 Other Applicable Sections

Chapter 8 describes the derivation of acoustic element matrices and load vectors.

14.31 LINK31 — Radiation Link



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	None used (nodes may be coincident)	None

14.31.1 Standard Radiation (KEYOPT(3) = 0)

The two–surface radiation equation (from equation (6.1–12)) that is solved (iteratively) is:

$$Q = \sigma \epsilon FA \left(T_I^4 - T_J^4\right) \tag{14.31–1}$$

where:
- Q = heat flow rate between nodes I and J (output quantity HEAT RATE)
- σ = Stefan–Boltzmann constant (input as SBC on **R** command)
- ε = emissivity (input as EMISSIVITY on **R** or EMIS on **MP** command)
- F = geometric form factor (input as FORM FACTOR on **R** command)
- A = area of element (input as AREA on **R** command)
- T_I, T_J = absolute temperatures at nodes I and J

The program uses a linear equation solver. Therefore, equation (14.31–1) is expanded as:

$$Q = \sigma \epsilon FA \left(T_I^2 + T_J^2\right)(T_I + T_J)(T_I - T_J) \tag{14.31–2}$$

and then rewritten as:

$$Q = \sigma \epsilon FA \left(T_{I,n-1}^2 + T_{J,n-1}^2\right)\left(T_{I,n-1} + T_{J,n-1}\right)\left(T_{I,n} - T_{J,n}\right) \tag{14.31–3}$$

where the subscripts n and n–1 refer to the current and previous iterations, respectively. It is then recast into finite element form:

$$\begin{Bmatrix} Q_I \\ Q_J \end{Bmatrix} = C_o \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} T_{I,n} \\ T_{J,n} \end{Bmatrix} \tag{14.31–4}$$

with

$$C_o = \sigma \epsilon FA \left(T^2_{I,n-1} + T^2_{J,n-1}\right)\left(T_{I,n-1} + T_{J,n-1}\right) \tag{14.31–5}$$

14.31.2 Empirical Radiation (KEYOPT(3) = 1)

The basic equation is:

$$Q = \sigma \epsilon \left(FT^4_I - AT^4_J\right) \tag{14.31–6}$$

instead of equation (14.31–1). This form leads to

$$C_o = \sigma \epsilon \left(F^{\frac{1}{2}} T^2_{I,n-1} + A^{\frac{1}{2}} T^2_{J,n-1}\right)\left(F^{\frac{1}{4}} T_{I,n-1} + A^{\frac{1}{4}} T_{J,n-1}\right) \tag{14.31–7}$$

instead of equation (14.31–5). And, hence the matrix equation (14.31–4) becomes:

$$\begin{Bmatrix} Q_I \\ Q_J \end{Bmatrix} = C_o \begin{bmatrix} F^{\frac{1}{4}} & -A^{\frac{1}{4}} \\ -F^{\frac{1}{4}} & A^{\frac{1}{4}} \end{bmatrix} \begin{Bmatrix} T_{I,n} \\ T_{J,n} \end{Bmatrix} \tag{14.31–8}$$

14.31.3 Solution

If the emissivity is input on a temperature dependent basis, equation (14.31–5) is rewritten to be:

$$C_o = \sigma FA \left(\beta^2_{I,n-1} + \beta^2_{J,n-1}\right)\left(\beta_{I,n-1} + \beta_{J,n-1}\right) \tag{14.31–9}$$

where: $\beta_i = T_i \left(\epsilon_i\right)^{\frac{1}{3}}$ (i = I or J)

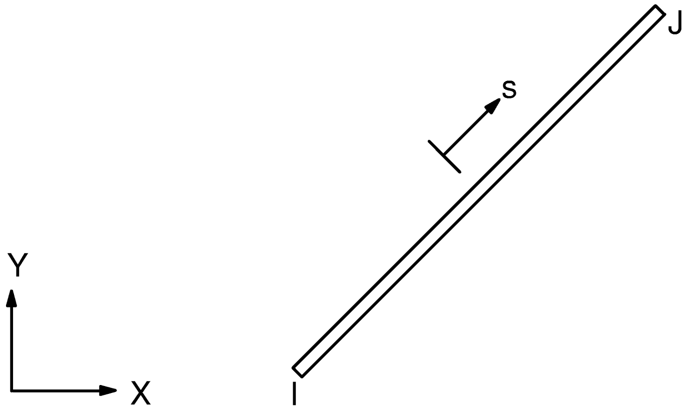
ε_i = emissivity at node i evaluated at temperature T^f_i

T_i^f = $T_i - T_{off}$

T_{off} = offset temperature (input on **TOFFST** command)

Equation (14.31–7) is handled analogously.

14.32 LINK32 — 2–D Conduction Bar



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equation (12.1.1–20)	None
Specific Heat Matrix	Equation (12.1.1–20)	None
Heat Generation Load Vector	Equation (12.1.1–20)	None

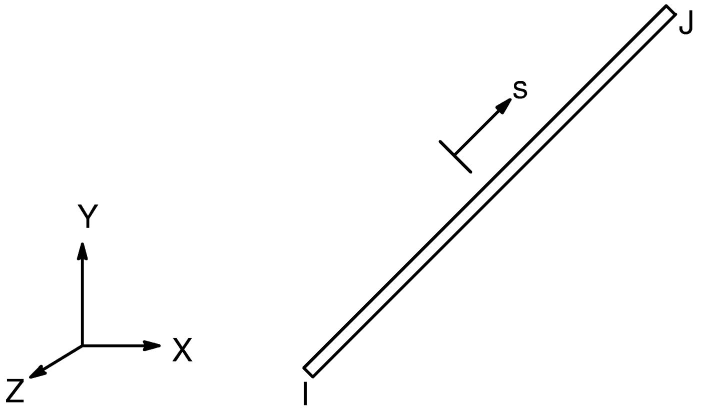
14.32.1 Other Applicable Sections

Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations.

14.32.2 Matrices and Load Vectors

The matrices and load vectors described in Section 14.33 apply here.

14.33 LINK33 — 3–D Conduction Bar



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equation (12.2.1–20)	None
Specific Heat Matrix	Equation (12.2.1–20)	None
Heat Generation Load Vector	Equation (12.2.1–20)	None

14.33.1 Other Applicable Sections

Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations.

14.33.2 Matrices and Load Vectors

The conductivity matrix is:

$$\left[K_e^t\right] = \frac{AK_x}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.33–1}$$

where: A = area (input as AREA on **R** command)

K_x = conductivity (input as KXX on **MP** command)

L = distance between nodes

The specific heat matrix is:

$$\left[C_e^t\right] = \frac{\rho C_p A L}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (14.33\text{–}2)$$

where: ϱ = density (input as DENS on **MP** command)
C_p = specific heat (input as C on **MP** command)

This specific heat matrix is a diagonal matrix with each diagonal being the sum of the corresponding row of a consistent specific heat matrix. The heat generation load vector is:

$$\{Q_e\} = \frac{\dddot{q}AL}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} \qquad (14.33\text{–}3)$$

where: $\dddot{q}$ = heat generation rate (input on **BF** or **BFE** command)

14.33.3 Output

The output is computed as:

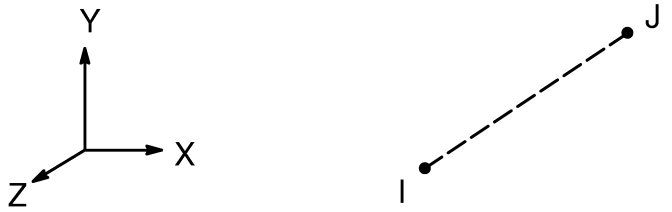
$$q = K_x \frac{(T_I - T_J)}{L} \qquad (14.33\text{–}4)$$

and

$$Q = qA \qquad (14.33\text{–}5)$$

where: q = output quantity THERMAL FLUX
T_I = temperature at node I
T_J = temperature at node J
Q = output quantity HEAT RATE

14.34 LINK34 — Convection Link



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	None used (nodes may be coincident)	None

14.34.1 Conductivity Matrix

The element conductivity (convection) matrix is

$$\left[K_e^t\right] = A\ h_f^{eff} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.34–1}$$

where:
A = area over which element acts (input as AREA on **R** command)
h_f^{eff} = effective film coefficient, defined by equation (14.34–2)

The effective film coefficient is:

$$h_f^{eff} = \begin{cases} \text{maximum of } \left(h_f', \ C_c\right) & \text{if KEYOPT(3) } = 3 \\ h_f' + C_c & \text{if KEYOPT(3) } \neq 3 \end{cases} \tag{14.34–2}$$

where:
h_f' = partial film coefficient term defined by equation (14.34–3)
C_c = input as CC on **R** command

The partial film coefficient term is:

$$h_f' = \begin{cases} F\ h_f & \text{if } n = 0.0 \\ F\ h_f \ |\Delta T_p|^n & \text{if } n \neq 0.0 \text{ and } \Delta T_p \neq 0 \\ 0.0 & \text{if } n \neq 0.0 \text{ and } \Delta T_p = 0 \end{cases} \tag{14.34–3}$$

where: $F = \begin{cases} T_B & \text{if } T_B > 0 \text{ and KEYOPT(3)} = 2 \\ 1.0 & \text{if } T_B \leq 0 \text{ or KEYOPT(3)} \neq 2 \end{cases}$

T_B = input quantity TBULK (input on **SFE** command)

$$h_f = \begin{cases} H(m_e) & \text{if KEYOPT(3)} \neq 2 \text{ or if KEYOPT(3)} = 2 \text{ and } h_f^{in} = 0.0 \\ H\left(-h_f^{in}\right) & \text{if KEYOPT(3)} = 2 \text{ and } h_f^{in} < 0.0 \\ h_f^{in} & \text{if KEYOPT(3)} = 2 \text{ and } h_f^{in} > 0.0 \end{cases}$$

$H(x)$ = input quantity CO on **MP**,HF command for material x

m_e = material number for this element (input on **MAT** command)

h_f^{in} = input quantity VAL1 (**SFE**,,,CONV,1 command)

ΔT_p = $T_{p,J} - T_{p,I}$

$T_{p,J}$ = temperature from previous iteration at node J

n = input quantity EN on **R** command

ΔT_p must be thought of as unitless, even though it is obviously derived from temperatures.

14.34.2 Output

The output is computed as:

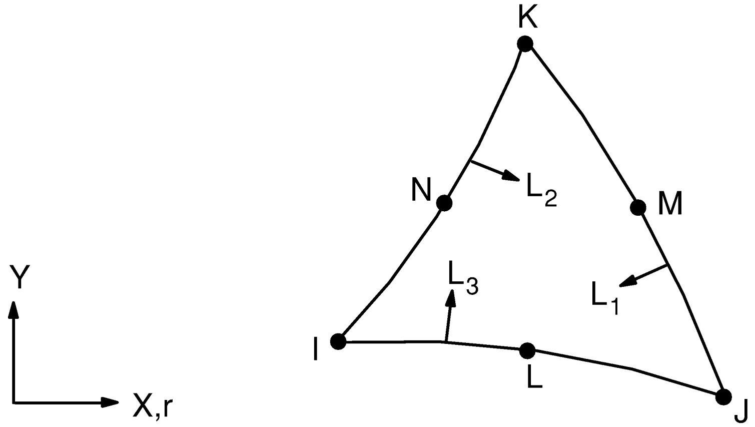
$$Q = Ah_f^{eff}\left(T_I - T_J\right) \tag{14.34–4}$$

where: Q = output quantity HEAT RATE

T_I = temperature at node I

T_J = temperature at node J

14.35 PLANE35 — 2–D 6–Node Triangular Thermal Solid

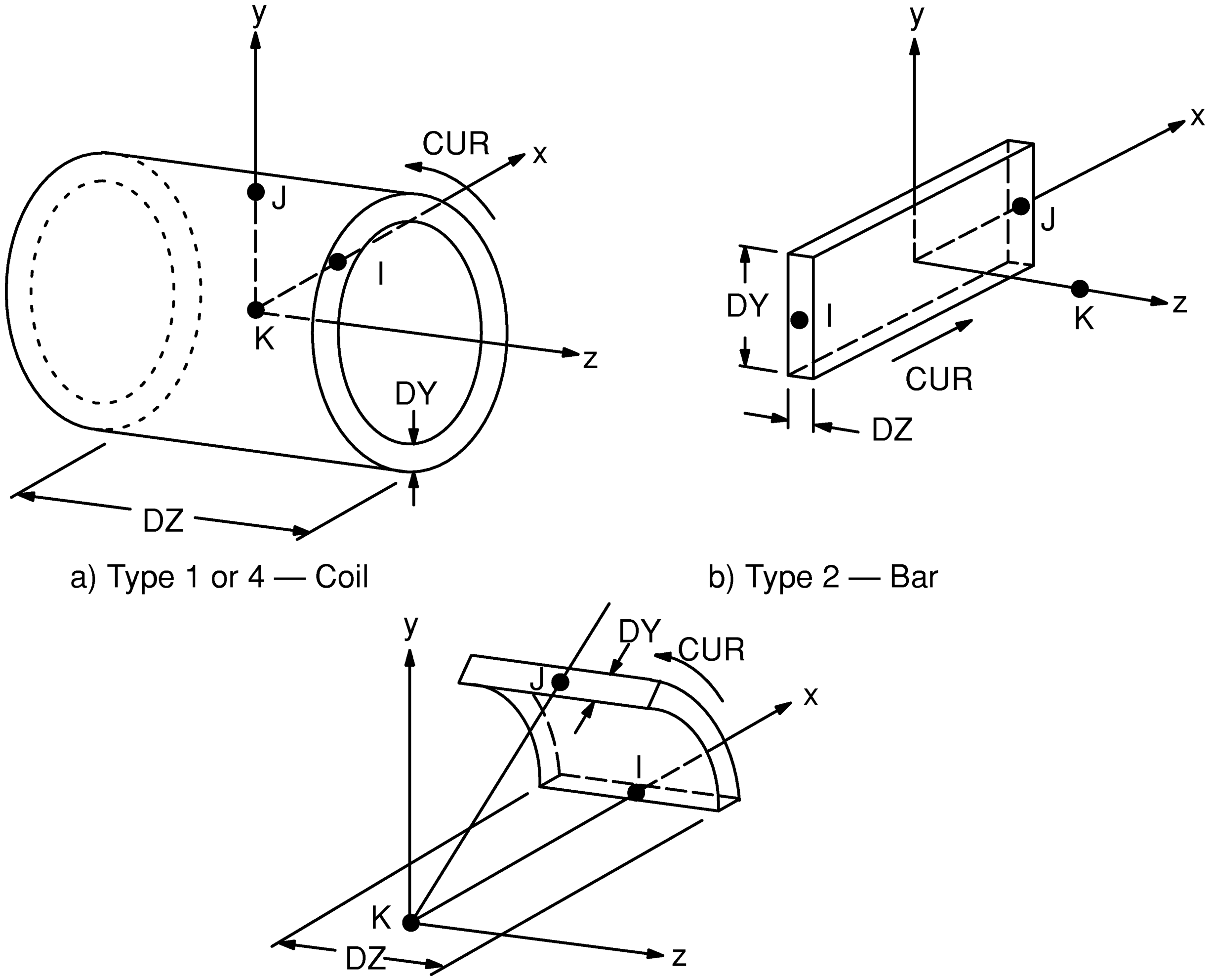


Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equation (12.6.2–20)	6
Specific Heat Matrix	Equation (12.6.2–20). If KEYOPT(1)=1, matrix is diagonalized as described in Section 13.2	6
Heat Generation Load Vector	Equation (12.6.2–20)	6
Convection Surface Matrix and Load Vector	Equation (12.6.2–20), specialized to the face	2

14.35.1 Other Applicable Sections

Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations.

14.36 SOURC36 — Current Source



a) Type 1 or 4 — Coil

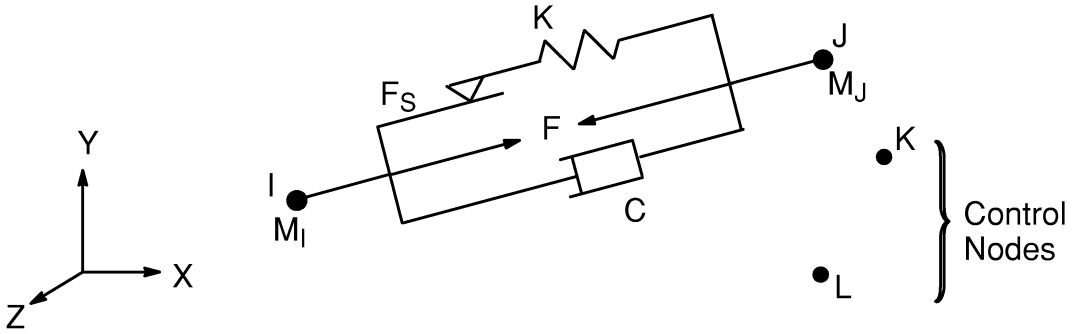
b) Type 2 — Bar

c) Type 3 or 5 — Arc

14.36.1 Description

The functionality of SOURC36 is basically one of user convenience. It provides a means of specifying the necessary data to evaluate the Biot–Savart integral (equation (5.1–17)) for the simple current source configurations, coil, bar and arc. The magnetic field {H_s} that results from this evaluation in turn becomes a load for the magnetic scalar potential elements (SOLID5, SOLID96 and SOLID98) as discussed in Chapter 5.

14.37 COMBIN37 — Control



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (nodes may be coincident)	None
Mass Matrix	None (lumped mass formulation)	None
Damping Matrix	None	None

14.37.1 Element Characteristics

COMBIN37 is a nonlinear, 1–D element with two active nodes and one or two control nodes. The element has spring–damper–sliding capability similar to COMBIN40. The degree of freedom (DOF) for the active nodes is selected using KEYOPT(3) and the DOF for the control nodes is selected using KEYOPT(2).

The action of the element in the structure is based upon the value of the control parameter (P_{cn}) (explained later), O_n and O_f (input quantities ONVAL and OFFVAL on **R** command), and the behavior switches KEYOPT(4) and (5). Figure 14.37–1 illustrates the behavior of one of the more common modes of operation of the element. It is analogous to the normal home thermostat during the winter.

The behavior of all possible combinations of KEYOPT(4) and (5) values is summarized in Figure 14.37–2 through Figure 14.37–7. Figure 14.37–5 is analogous to Figure 14.37–1. P_{cn} represents the control parameter (output quantity CONTROL PARAM). The element is active where the figure indicates on, and inactive where it indicates off. For some options, the element may be either on or off for P_{cn} between O_n and O_f, depending upon the last status change.

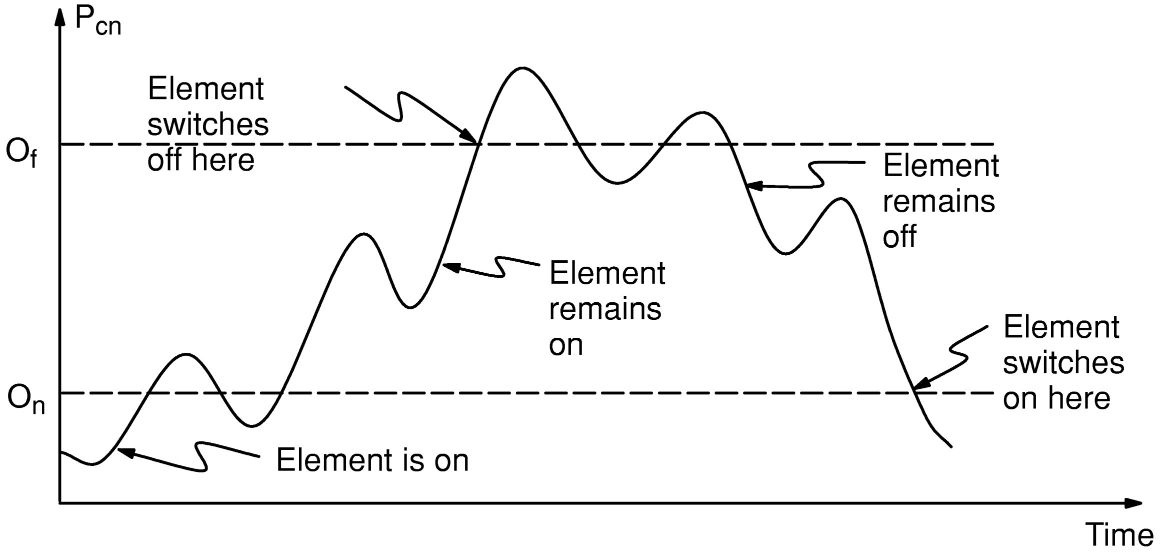


Figure 14.37–1 Element Behavior with KEYOPT(4) = 0, KEYOPT(5) = 1, and $O_f > O_n$

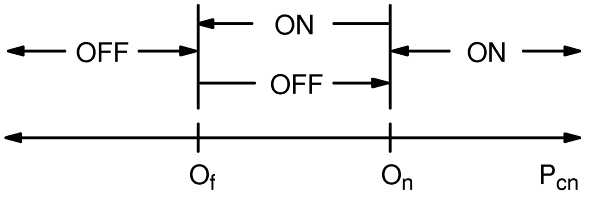


Figure 14.37–2 Element Behavior with KEYOPT(4) = 0, KEYOPT(5) = 0, and $O_f \leq O_n$

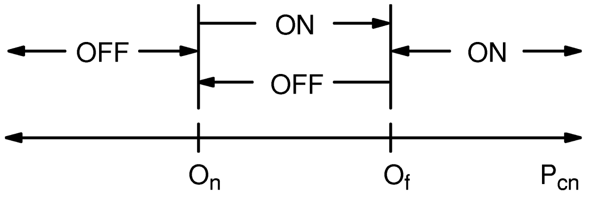


Figure 14.37–3 Element Behavior with KEYOPT(4) = 0, KEYOPT(5) = 0, and $O_f > O_n$

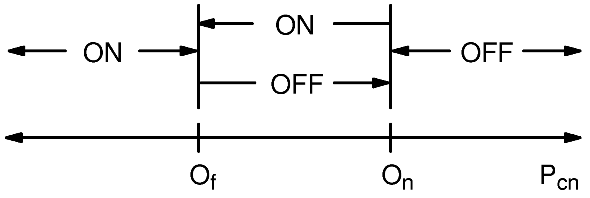


Figure 14.37–4 Element Behavior with KEYOPT(4) = 0, KEYOPT(5) = 1, and $O_f \leq O_n$

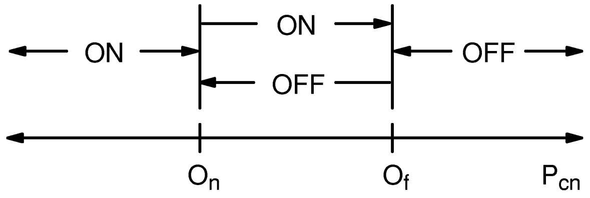


Figure 14.37–5 Element Behavior with KEYOPT(4) = 0, KEYOPT(5) = 1, and $O_f > O_n$

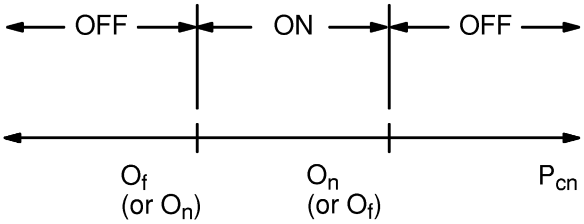


Figure 14.37–6 Element Behavior with KEYOPT(4) = 1 and and KEYOPT(5) = 0

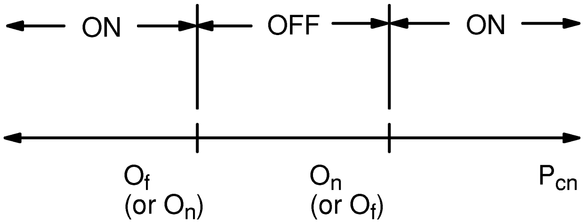


Figure 14.37–7 Element Behavior with KEYOPT(4) = 1 and KEYOPT(5) = 1

14.37.2 Element Matrices

When the element status is ON, the element matrices are:

$$[K_e] = k_o \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.37–1}$$

$$[M_e] = \begin{bmatrix} M_I & 0 \\ 0 & M_J \end{bmatrix} \tag{14.37–2}$$

$$[C_e] = C_o \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.37–3}$$

where: k_o = stiffness (input quantity STIF on **R** command)
M_I = mass at node I (input quantity MASI on **R** command)
M_J = mass at node J (input quantity MASJ on **R** command)
C_o = damping constant (input quantity DAMP on **R** command)

When the element status is OFF, all element matrices are set to zero.

14.37.3 Adjustment of Real Constants

If KEYOPT(6) > 0, a real constant is to be adjusted as a function of the control parameter as well as other real constants. Specifically,

$$\text{if KEYOPT(6)} = 0 \text{ or } 1,\ k_o' = k_o + D \tag{14.37–4}$$

$$\text{if KEYOPT(6)} = 2,\ C_o' = C_o + D \tag{14.37–5}$$

$$\text{if KEYOPT(6)} = 3,\ M_J' = M_J + D \tag{14.37–6}$$

$$\text{if KEYOPT(6)} = 4,\ O_n' = O_n + D \tag{14.37–7}$$

$$\text{if KEYOPT(6)} = 5,\ O_f' = O_f + D \tag{14.37–8}$$

$$\text{if KEYOPT(6)} = 6,\ F_A' = F_A + D \tag{14.37–9}$$

$$\text{if KEYOPT(6)} = 7,\ M_I' = M_I + D \tag{14.37–10}$$

$$\text{or, if KEYOPT(6)} = 8,\ F_S' = F_S + D \tag{14.37–11}$$

where:

$$D = \begin{cases} C_1\ |P_{cn}|^{C_2} + C_3\ |P_{cn}|^{C_4} & \text{if KEYOPT(9) = 0} \\ f_1\ (C_1,\ C_2,\ C_3,\ C_4,\ P_{cn}) & \text{if KEYOPT(9) = 1} \end{cases}$$

F_A = input quantity AFORCE ON **R** command

F_S = input quantity FSLIDE on **RMORE** command

C_1, C_2, C_3, C_4 = input quantities C1, C2, C3, and C4 on **RMORE** command

P_{cn} = control parameter (defined below)

f_1 = function defined by subroutine USERRC

If F_S' (or F_S, if KEYOPT(6) $\neq$ 8) is less than zero, it is reset to zero.

14.37.4 Evaluation of Control Parameter

The control parameter is defined as:

$$P_{cn} = \begin{cases} V & \text{if KEYOPT(1) = 0 or 1} \\ \dfrac{dV}{dt} & \text{if KEYOPT(1) = 2} \\ \dfrac{d^2V}{dt^2} & \text{if KEYOPT(1) = 3} \\ \int_o^t V dt & \text{if KEYOPT(1) = 4} \\ t & \text{if KEYOPT(1) = 5} \end{cases} \qquad (14.37\text{–}12)$$

where: $V = \begin{cases} u(K) - u(L) & \text{if node L is defined} \\ u(K) & \text{if node L is not defined} \end{cases}$

t = time (input on **TIME** command)

u = degree of freedom as selected by KEYOPT(2)

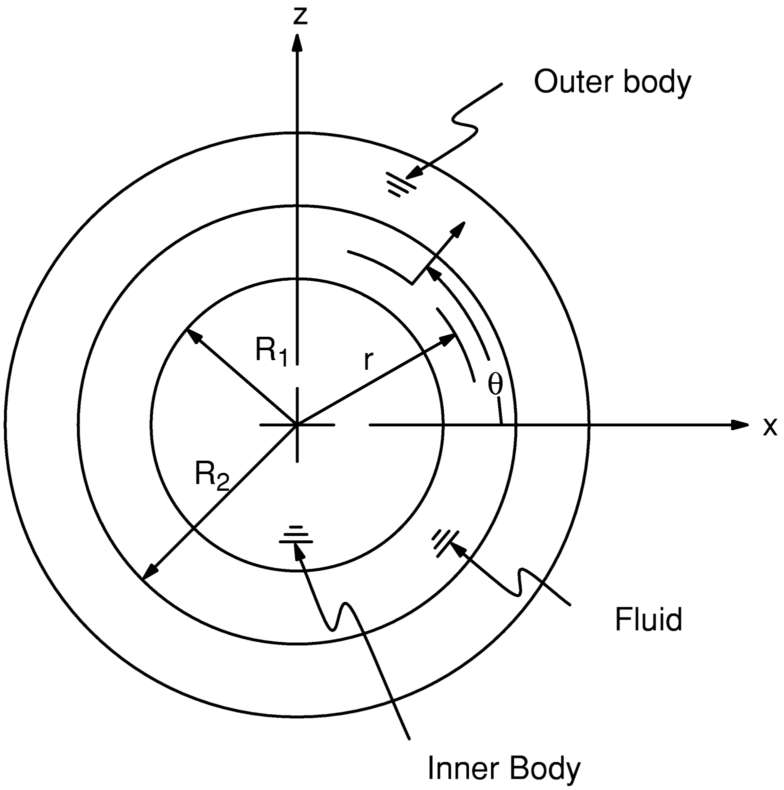
The assumed value of the control parameter for the first iteration (P^1_{cn}) is defined as:

$$P^1_{cn} = \begin{cases} \dfrac{O_n + O_f}{2} & \text{if } S^t = 1 \text{ or–1} \\ \text{or} & \\ T_{UNIF} & \text{if } S^t = 0 \text{ and KEYOPT(2) = 8} \\ \text{or} & \\ 0 & \text{all other cases} \end{cases} \qquad (14.37\text{–}13)$$

where: S^t = input as START on **R** command defining starting status where: 1 means ON, –1 means OFF

T_{UNIF} = uniform temperature (input on **BFUNIF** command)

14.38 FLUID38 — Dynamic Fluid Coupling



Matrix or Vector	Shape Functions	Integration Points
Mass Matrix	$u = \left(\frac{C_1}{r^2} - C_2\right)\cos\theta$ $w = \left(\frac{C_1}{r^2} + C_2\right)\sin\theta$ (KEYOPT(3) = 0 only)	None
Damping Matrix	Not defined	None

Reference: Fritz(12)

14.38.1 Description

This element is used to represent a dynamic coupling between two points of a structure. The coupling is based on the dynamic response of two points connected by a constrained mass of fluid. The points represent the centerlines of concentric cylinders. The fluid is contained in the annular space between the cylinders. The cylinders may be circular or have an arbitrary cross–section. The element has two DOFs per node: translations in the nodal x and z directions. The axes of the cylinders are assumed to be in the nodal y directions. These orientations may be changed with KEYOPT(6).

14.38.2 Assumptions and Restrictions

1. The motions are assumed to be small with respect to the fluid channel thickness.
2. The fluid is assumed to be incompressible.
3. Fluid velocities should be less than 10% of the speed of sound in the fluid.
4. The flow channel length should be small compared to the wave length for propagating vibratory disturbances (less than about 10%), in order to avoid the possibility of standing wave effects.

14.38.3 Mass Matrix Formulation

The mass matrix formulation used in the element is of the following form:

$$[M_e] = \begin{bmatrix} m_{11} & 0 & m_{13} & 0 \\ 0 & m_{22} & 0 & m_{24} \\ m_{31} & 0 & m_{33} & 0 \\ 0 & m_{42} & 0 & m_{44} \end{bmatrix} \qquad (14.38\text{–}1)$$

The m values are dependent upon the KEYOPT(3) value selected. For KEYOPT(3) = 0 (concentric cylinder case):

$$m_{11} = m_{22} = M \left(R_1^4 + R_1^2 \, R_2^2 \right) \qquad (14.38\text{–}2)$$

$$m_{13} = m_{31} = m_{24} = m_{42} = - M \left(2R_1^2 \, R_2^2 \right) \qquad (14.38\text{–}3)$$

$$m_{33} = m_{44} = M \left(R_1^2 \, R_2^2 + R_2^4 \right) \qquad (14.38\text{–}4)$$

where: $M = \dfrac{\pi \, L \, \rho}{R_2^2 - R_1^2}$ (Mass/Length^4)

ϱ = fluid mass density (input as DENS on **MP** command)

R_1 = radius of inner cylinder (input quantity R1 on **R** command)

R_2 = radius of outer cylinder (input quantity R2 on **R** command)

L = length of cylinders (input quantity L on **R** command)

Note that the shape functions are similar to that for PLANE25 or FLUID81 with MODE = 1. The element mass used in the evaluation of the total structure mass is $\pi \; L \; \rho \left(R_2^2 - R_1^2 \right)$.

For KEYOPT(3) = 2, which is a generalization of the above cylindrical values but for different geometries, the m values are as follows:

$$m_{11} = M_{hx} \tag{14.38–5}$$

$$m_{13} = m_{31} = -\left(M_1 + M_{hx} \right) \tag{14.38–6}$$

$$m_{33} = \left(M_1 + M_2 + M_{hx} \right) \tag{14.38–7}$$

$$m_{22} = M_{hz} \tag{14.38–8}$$

$$m_{24} = m_{42} = -\left(M_1 + M_{hz} \right) \tag{14.38–9}$$

$$m_{44} = M_1 + M_2 + M_{hz} \tag{14.38–10}$$

where: M_1 = mass of fluid displaced by the inner boundary (Boundary 1) (input quantity M1 on **R** command)

M_2 = mass of fluid that could be contained within the outer boundary (Boundary 2) in absence of the inner boundary (input quantity M2 on **R** command)

M_{hx}, M_{hz} = hydrodynamic mass for motion in the x and z directions, respectively (input quantities MHX and MHZ on **R** command)

The element mass used in the evaluation of the total structure mass is $M_2 - M_1$.

The lumped mass option (**LUMPM**,ON) is not available.

14.38.4 Damping Matrix Formulation

The damping matrix formulation used in the element is of the following form:

$$[C_e] = \begin{bmatrix} c_{11} & 0 & c_{13} & 0 \\ 0 & c_{22} & 0 & c_{24} \\ c_{31} & 0 & c_{33} & 0 \\ 0 & c_{42} & 0 & c_{44} \end{bmatrix} \tag{14.38–11}$$

The c values are dependent upon the KEYOPT(3) value selected. For KEYOPT(3) = 0:

$$c_{11} = c_{33} = C\Delta xW_x \tag{14.38–12}$$

$$c_{13} = c_{31} = -\ C\Delta xW_x \tag{14.38–13}$$

$$c_{22} = c_{44} = C\Delta zW_z \tag{14.38–14}$$

$$c_{24} = c_{42} = -\ C\Delta zW_z \tag{14.38–15}$$

where: C = $\dfrac{f\ \rho\ L\ R_1^2\left(R_1^2 + R_2^2\right)}{3\left(R_2 - R_1\right)^3}$ (Mass/Length)

W_x, W_z = estimate of resonant frequencies in the x and z response directions, respectively (input quantities WX, WZ on **RMORE** command)

f = Darcy friction factor for turbulent flow (input quantity F on **R** command)

Δx, Δz = estimate of peak relative amplitudes between inner and outer boundaries for the x and z motions, respectively (input quantities DX, DZ on **R** command)

For KEYOPT(3) = 2, the c values are as follows:

$$c_{11} = c_{33} = C_x\Delta xW_x \tag{14.38–16}$$

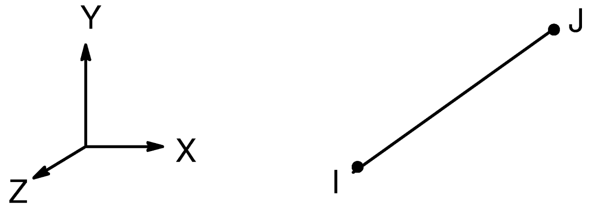
$$c_{13} = c_{31} = -\ C_x\Delta xW_x \tag{14.38–17}$$

$$c_{22} = c_{44} = C_z\Delta zW_z \tag{14.38–18}$$

$$c_{24} = c_{42} = -\ C_z\Delta zW_z \tag{14.38–19}$$

where: C_x, C_z = flow and geometry constants for the x and z motions, respectively (input quantities CX, CZ on **RMORE** command)

14.39 COMBIN39 — Nonlinear Spring



Matrix or Vector	Option	Shape* Functions	Integration Points
Stiffness Matrix	Longitudinal	Equation (12.2.2–1)	None
	Torsional	Equation (12.2.2–4)	None
Stress Stiffening Matrix	Longitudinal	Equations (12.2.1–2) and (12.2.1–3)	None

* There are no shape functions used if the element is input as a one DOF per node basis (KEYOPT(4) = 0) as the nodes are coincident.

14.39.1 Input

The user explicitly defines the force–deflection curve for COMBIN39 by the input of discrete points of force versus deflection. Up to 20 points on the curve may be defined, and are entered as real constants. This input is illustrated in Figure 14.39–1. The input curve must pass through the origin and must lie within the unshaded regions, if KEYOPT(1) = 1.

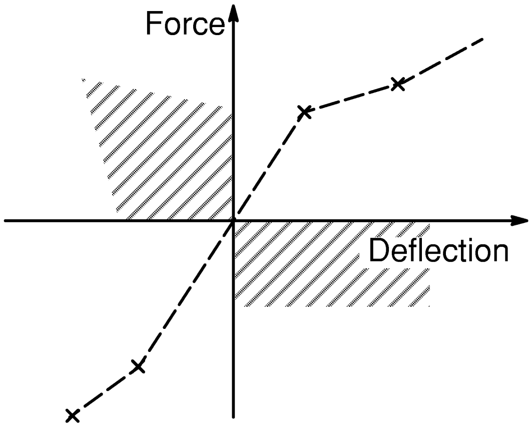


Figure 14.39–1 Input Force–Deflection Curve

The input deflections must be given in ascending order, with the minimum change of deflection of:

$$u_{i+1} - u_i > \Delta u_{min} \tag{14.39–1}$$

where: u_{i+1} = input quantity D2, D3, ... D20 on **R** or **RMORE** commands, whichever applies

u_i = input quantity D1, D2, ... D19 on **R** or **RMORE** commands, whichever applies

Δu_{min} = $\dfrac{u_{max} - u_{min}}{10^7}$

u_{max} = most positive input deflection

u_{min} = most negative input deflection

14.39.2 Element Stiffness Matrix and Load Vector

During the stiffness pass of a given iteration, COMBIN39 will use the results of the previous iteration to determine which segment of the input force–deflection curve is active. The stiffness matrix and load vector of the element are then:

$$[K_e] = K^{tg} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.39–2}$$

$$\{F_e^{nr}\} = F_1 \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} \tag{14.39–3}$$

where: K^{tg} = slope of active segment from previous iteration (output quantity SLOPE)

F_1 = force in element from previous iteration (output quantity FORCE)

If KEYOPT(4) > 0, equations (14.39–2) and (14.39–3) are expanded to 2 or 3 dimensions.

During the stress pass, the deflections of the current equilibrium iteration will be examined to see whether a different segment of the force–deflection curve should be used in the next equilibrium iteration.

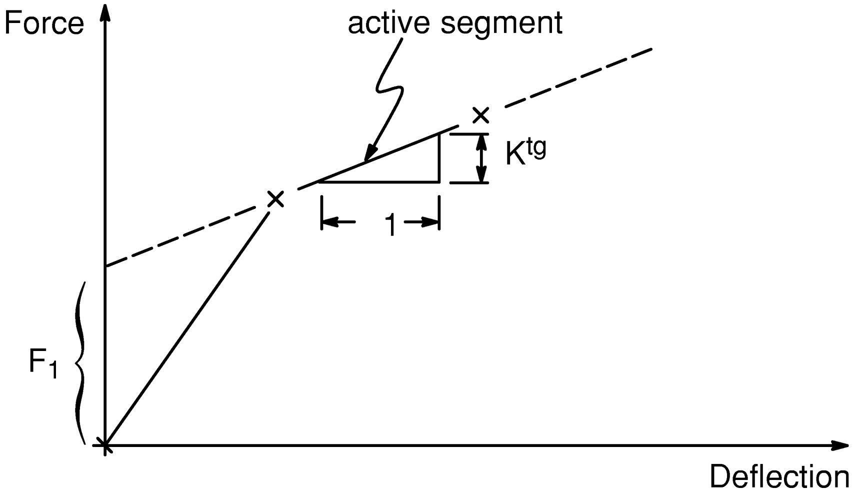


Figure 14.39–2 COMBIN39 – Stiffness Computation

14.39.3 Choices for Element Behavior

If KEYOPT(2) = 0 and if no force–deflection points are input for deflection less than zero, the points in the first quadrant are reflected through the origin (Figure 14.39–3).

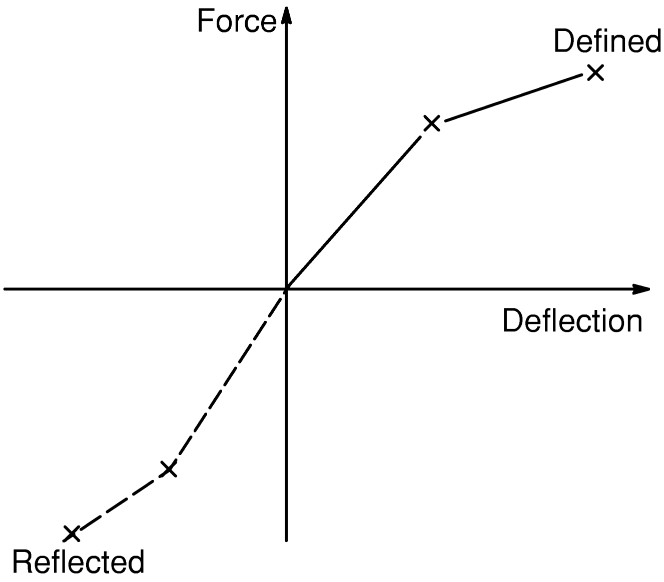


Figure 14.39–3 Input Force–Deflection Curve Reflected Through Origin

If KEYOPT(2) = 1, there will be no stiffness for the deflection less than zero (Figure 14.39–4).

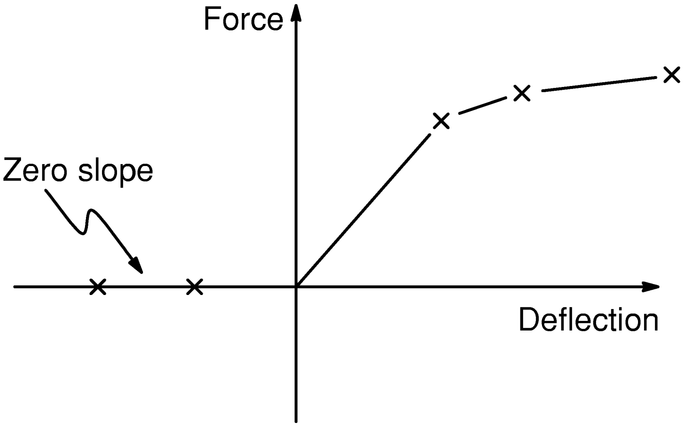


Figure 14.39–4 Force–Deflection Curve with KEYOPT(2) = 1

If KEYOPT(1) = 0, COMBIN39 is conservative. This means that regardless of the number of loading reversals, the element will remain on the originally defined force–deflection curve, and no energy loss will occur in the element. This also means that the solution is not path–dependent. If, however, KEYOPT(1) = 1, the element is nonconservative. With this option, energy losses can occur in the element, so that the solution is path–dependent. The resulting behavior is illustrated in Figure 14.39–5.

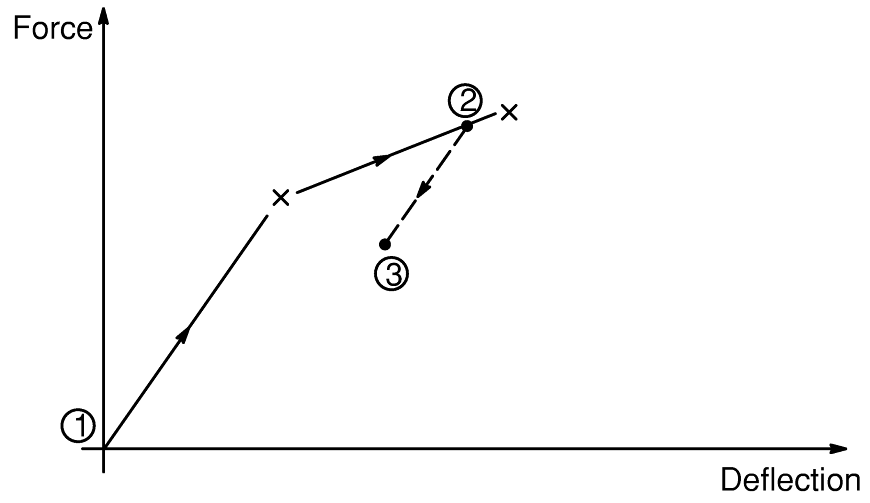


Figure 14.39–5 Nonconservative Unloading (KEYOPT(1) = 1)

When a load reversal occurs, the element will follow a new force–deflection line passing through the point of reversal and with slope equal to the slope of the original curve on that side of the origin (0+ or 0–). If the reversal does not continue past the force = 0 line, reloading will follow the straight line back to the original curve (Figure 14.39–6).

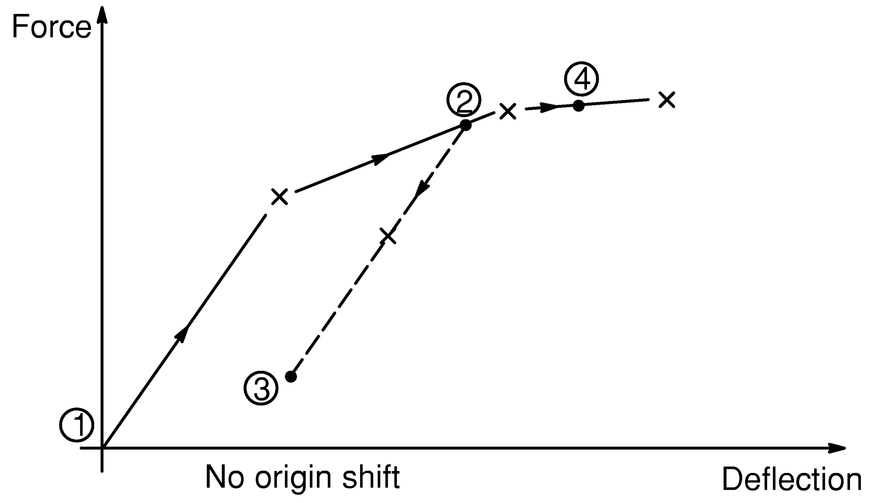


Figure 14.39–6 No Origin Shift on Reversed Loading (KEYOPT(1) = 1)

If the reversal continues past the force = 0 line, a type of origin shift occurs, and reloading will follow a curve that has been shifted a distance u_{orig} (output quantity UORIG) (Figure 14.39–7).

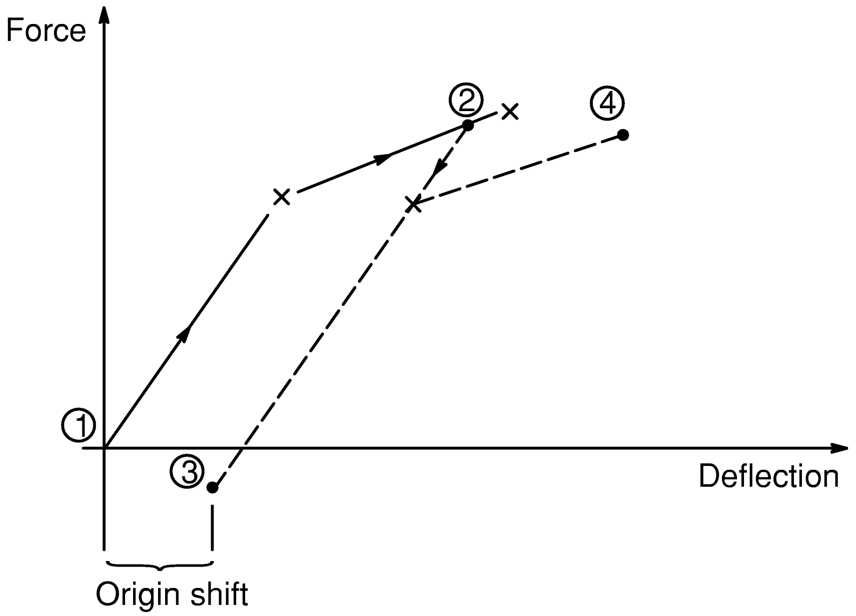


Figure 14.39–7 Origin Shift on Reversed Loading (KEYOPT(1) = 1)

A special option (KEYOPT(2) = 2) is included to model crushing behavior. With this option, the element will follow the defined tensile curve if it has never been loaded in compression. Otherwise, it will follow a reflection through the origin of the defined compressive curve (Figure 14.39–8).

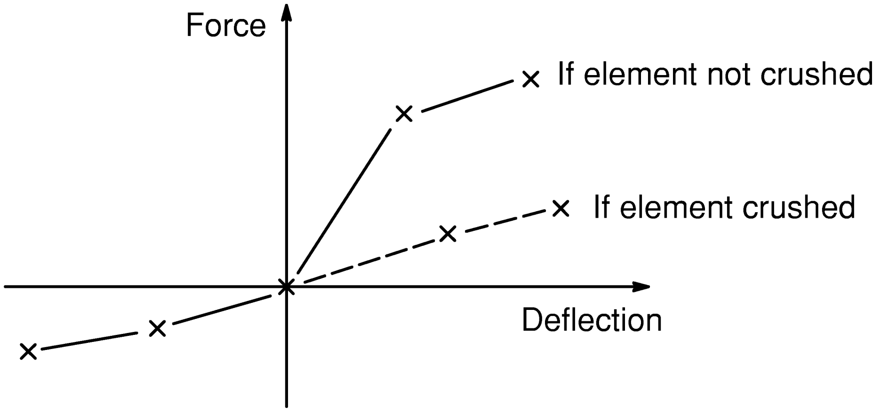
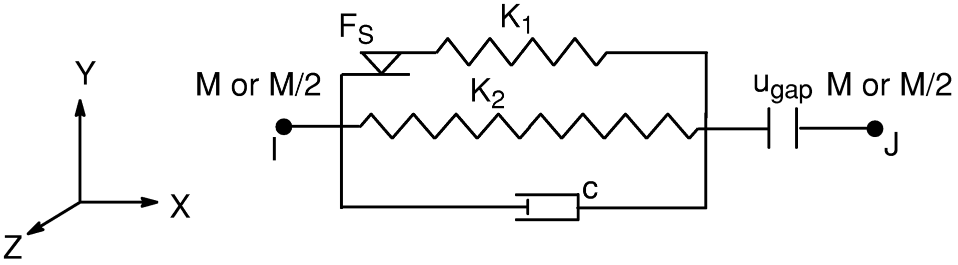


Figure 14.39–8 Crush Option (KEYOPT(2) = 2)

14.40 COMBIN40 — Combination



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (nodes may be coincident)	None
Mass Matrix	None (nodes may be coincident)	None
Damping Matrix	None (nodes may be coincident)	None

14.40.1 Characteristics of the Element

The force–deflection relationship for the combination element under initial loading is as shown in Figure 14.40–1 (for no damping).

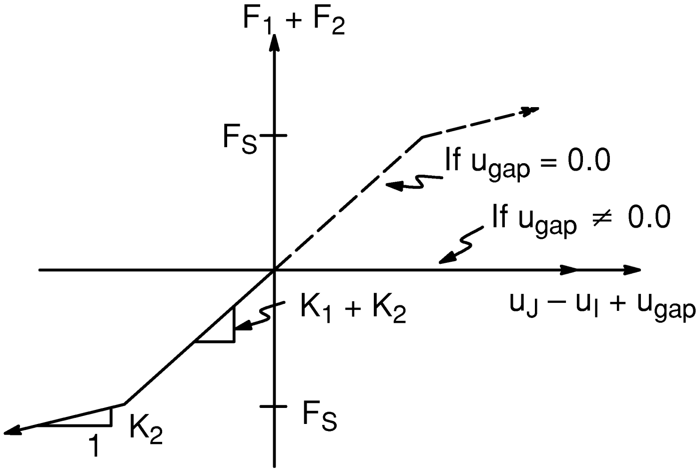


Figure 14.40–1 Force–Deflection Relationship

where: F_1 = force in spring 1 (output quantity F1)

F_2 = force in spring 2 (output quantity F2)

K_1 = stiffness of spring 1 (input quantity K1 on **R** command)

K_2 = stiffness of spring 2 (input quantity K2 on **R** command)

u_{gap} = initial gap size (input quantity GAP on **R** command) (if zero, gap capability removed)

u_I = displacement at node I

u_J = displacement at node J

F_S = force required in spring 1 to cause sliding (input quantity FSLIDE on **R** command)

14.40.2 Element Matrices for Structural Applications

The element mass matrix is:

$$[M_e] = M \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ if KEYOPT(6) } = 0 \tag{14.40–1}$$

$$[M_e] = \frac{M}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ if KEYOPT(6) } = 1 \tag{14.40–2}$$

$$[M_e] = M \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ if KEYOPT(6) } = 2 \tag{14.40–3}$$

where: M = element mass (input quantity M on **R** command)

If the gap is open during the previous iteration, all other matrices and load vectors are null vectors. Otherwise, the element damping matrix is:

$$[C_e] = c \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.40–4}$$

where: c = damping constant (input quantity C on **R** command)

The element stiffness matrix is:

$$[K_e] = k \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{14.40–5}$$

where: $k = \begin{cases} K_1 + K_2 & \text{if slider was not sliding in previous iteration} \\ K_2 & \text{if slider was sliding in previous iteration} \end{cases}$

and the element Newton–Raphson load vector is:

$$\{F_e^{nr}\} = (F_1 + F_2)\begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \qquad (14.40\text{–}6)$$

F_1 and F_2 are the current forces in the element.

14.40.3 Determination of F_1 and F_2 for Structural Applications

1. If the gap is open,

$$F_1 + F_2 = 0.0 \qquad (14.40\text{–}7)$$

If no sliding has taken place, $F_1 = F_2 = 0.0$. However, if sliding has taken place during unidirectional motion,

$$F_1 = \frac{u_s K_1 K_2}{K_1 + K_2} \qquad (14.40\text{–}8)$$

and thus

$$F_2 = -F_1 \qquad (14.40\text{–}9)$$

where: u_s = amount of sliding (output quantity SLIDE)

2. If the gap is closed and the slider is sliding,

$$F_1 = \pm F_S \qquad (14.40\text{–}10)$$

and

$$F_2 = K_2\, u_2 \qquad (14.40\text{–}11)$$

where: u_2 = $u_J - u_I + u_{gap}$ = output quantity STR2

3. If the gap is closed and the slider is not sliding, but had slid before,

$$F_1 = K_1\, u_1 \qquad (14.40\text{–}12)$$

where: u_1 = $u_2 - u_s$ = output quantity STR1

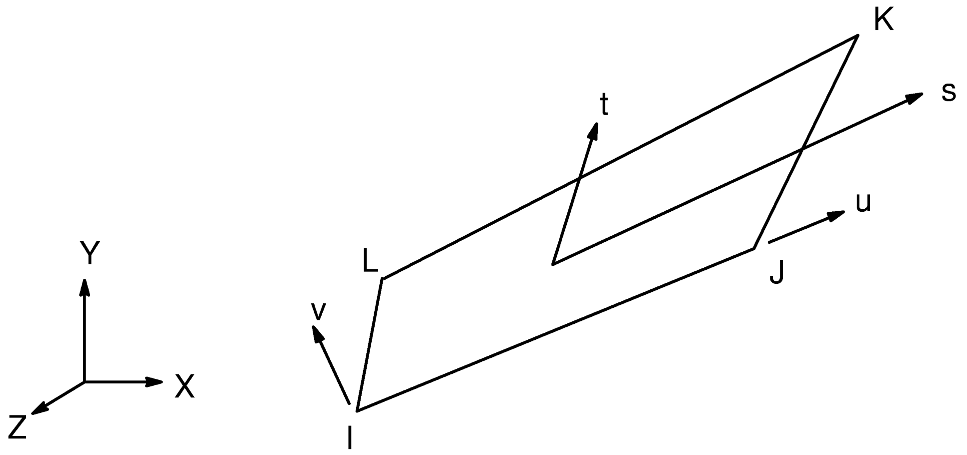
and

$$F_2 = K_2\, u_2 \qquad (14.40\text{–}13)$$

14.40.4 Thermal Analysis

The description above refers to structural analysis only. When this element is used in a thermal analysis, the conductivity matrix is $[K_e]$, the specific heat matrix is $[C_e]$ (based on input quantity M) and the Newton–Raphson load vector is $\{f_e^{nr}\}$, where F_1 and F_2 represent heat flow. Input quantity C is not used. The gap size u_{gap} is the temperature difference. Sliding, F_{slide}, is the element heat flow limit for conductor K_1.

14.41 SHELL41 — Membrane Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.8–1) and (12.5.8–2) and, if modified extra shape functions are included (KEYOPT(2) = 0) and element has 4 unique nodes Equations (12.5.9–1) and (12.5.9–2)	2 x 2
	Triangle	Equations (12.5.1–1) and (12.5.1–2)	1
Foundation Stiffness Matrix	Quad	Equation (12.5.8–3)	Same as stiffness matrix
	Triangle	Equation (12.5.1–3)	Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Mass Matrix	Quad	Equations (12.5.8–1), (12.5.8–2) and (12.5.8–3)	Same as stiffness matrix
	Triangle	Equations (12.5.1–1), (12.5.1–2), and (12.5.1–3)	Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix
Thermal and Normal Pressure Load Vector	Same as stiffness matrix		Same as stiffness matrix
Edge Pressure Load Vector	Same as mass matrix, specialized to the edge		2

Load Type	Distribution
Element Temperature	Bilinear in plane of element, constant thru thickness
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element and linear along each edge

References: Wilson(38), Taylor(49)

14.41.1 Assumptions and Restrictions

There is no out–of–plane bending stiffness.

When the 4–node option of this element is used, it is possible to input these four nodes so they do not lie in an exact flat plane. This is called a warped element, and such a nodal pattern should be avoided because equilibrium is lost. The element assumes that the resisting stiffness is at one location (in the plane defined by the cross product of the diagonals) and the structure assumes that the resisting stiffnesses are at other locations (the nodes). This causes an imbalance of the moments. The warping factor is computed as:

$$\phi = \frac{D}{\sqrt{A}} \tag{14.41–1}$$

where: D = component of the vector from the first node to the fourth node parallel to the element normal

A = element area

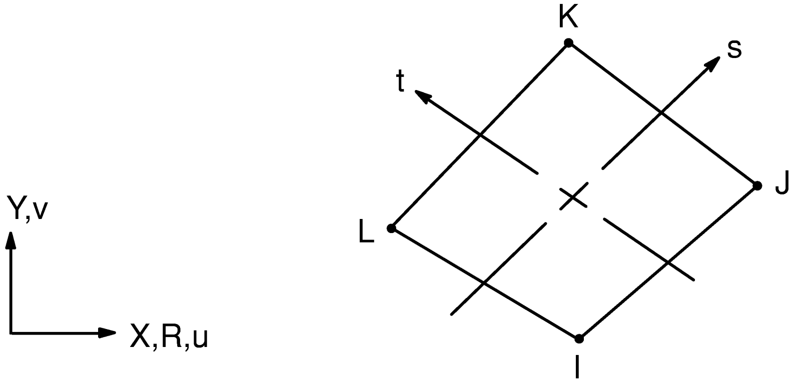
A warning message will print out if the warping factor exceeds 0.00004 and a fatal message occurs if it exceeds 0.04. Rigid offsets of the type used with SHELL63 are not used.

14.41.2 Wrinkle Option

When the wrinkle option is requested (KEYOPT(1) = 2), the stiffness is removed when the previous iteration is in compression, which is similar to the logic of the gap elements. This is referred to as the wrinkle option or cloth option. The following logic is used. First, the membrane stresses at each integration point are resolved into their principal directions so that shear is not directly considered. Then, three possibilities exist:

1. Both principal stresses are in tension. In this case, the program proceeds with the full stiffness at this integration point in the usual manner.
2. Both principal stresses are in compression. In this case, the contribution of this integration point to the stiffness is ignored.
3. One of the principal stresses is in tension and one is in compression. In this case, the integration point is treated as an orthotropic material with no stiffness in the compression direction and full stiffness in the tension direction. Then a tensor transformation is done to convert these material properties to the element coordinate system. The rest of the development of the element is done in the same manner is if the option were not used.

14.42 PLANE42 — 2–D Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.6.5–1) and (12.6.5–2) and, if modified extra shapes are included (KEYOPT(2) $\neq$ 1) and element has 4 unique nodes, equations (12.6.6–1) and (12.6.6–2)	2 x 2
	Triangle	Equations (12.6.1–1) and (12.6.1–2)	3 if axisymmetric and 1 if plane
Mass Matrix	Quad	Equations (12.6.5–1) and (12.6.5–2)	Same as stiffness matrix
	Triangle	Equations (12.6.1–1) and (12.6.1–2)	
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix
Pressure Load Vector	Same as mass matrix, specialized to face		2

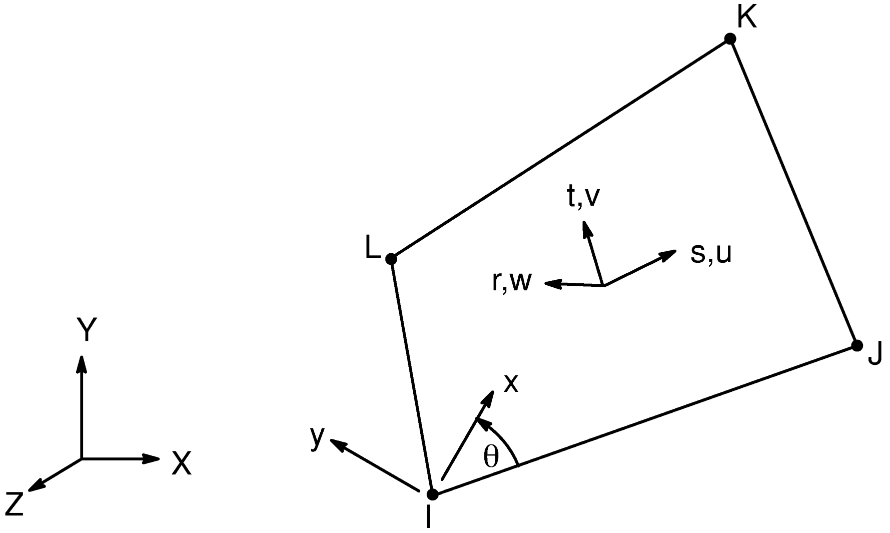
Load Type	Distribution
Element Temperature	Bilinear across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

References: Wilson(38), Taylor(49)

14.42.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.43 SHELL43 — Plastic Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.13–1), (12.5.13–2), and (12.5.13–3)	In–plane: 2 x 2 Thru–the–thickness: 2 (linear material) 5 (nonlinear material)
	Triangle	Equations (12.5.4–1), (12.5.4–2), and (12.5.4–3)	In–plane: 1 Thru–the–thickness: 2 (linear material) 5 (nonlinear material)
Mass Matrix	Quad	Equations (12.5.8–1), (12.5.8–2), and (12.5.8–3)	Same as stiffness matrix
	Triangle	Equations (12.5.1–1), (12.5.1–2), and (12.5.1–3)	Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Load Matrix	Same as stiffness matrix		Same as stiffness matrix
Transverse Pressure Load Vector	Quad	Equation (12.5.8–3)	2 x 2
	Triangle	Equation (12.5.1–3)	1
Edge Pressure Load Vector	Quad	Equations (12.5.8–1) and (12.5.8–2) specialized to the edge	2
	Triangle	Equations (12.5.1–1) and (12.5.1–2) specialized to the edge	2

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru thickness
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element and linear along each edge

References: Ahmad(1), Cook(5), Dvorkin(96), Dvorkin(97), Bathe and Dvorkin(98), Allman(113), Cook(114), MacNeal and Harder(115)

14.43.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.43.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same element (material) orientation.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.43.3 Assumed Displacement Shape Functions

The assumed displacement and transverse shear strain shape functions are given in Chapter 12. The basic shape functions are essentially a condensation of those used for SHELL93. The basic functions for the transverse shear strain have been changed to avoid shear locking (Dvorkin(96), Dvorkin(97), Bathe and Dvorkin(98)) and are pictured in Figure 14.43–1. One result of the use of these displacement and strain shapes is that elastic rectangular elements give constant curvature results for flat elements, and also, in the absence of membrane loads, for curved elements. Thus, for these cases, nodal stresses are the same as centroidal stresses. Both SHELL63 and SHELL93 can have linearly varying curvatures.

14.43.4 Stress–Strain Relationships

The material property matrix [D] for the element is:

$$[D] = \begin{bmatrix} AE_x & A\nu_{xy}E_x & 0 & 0 & 0 & 0 \\ A\nu_{xy}E_x & AE_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{yz}}{1.2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{xz}}{1.2} \end{bmatrix} \qquad (14.43\text{–}1)$$

where:

A = $\dfrac{E_y}{E_y - (\nu_{xy})^2 E_x}$

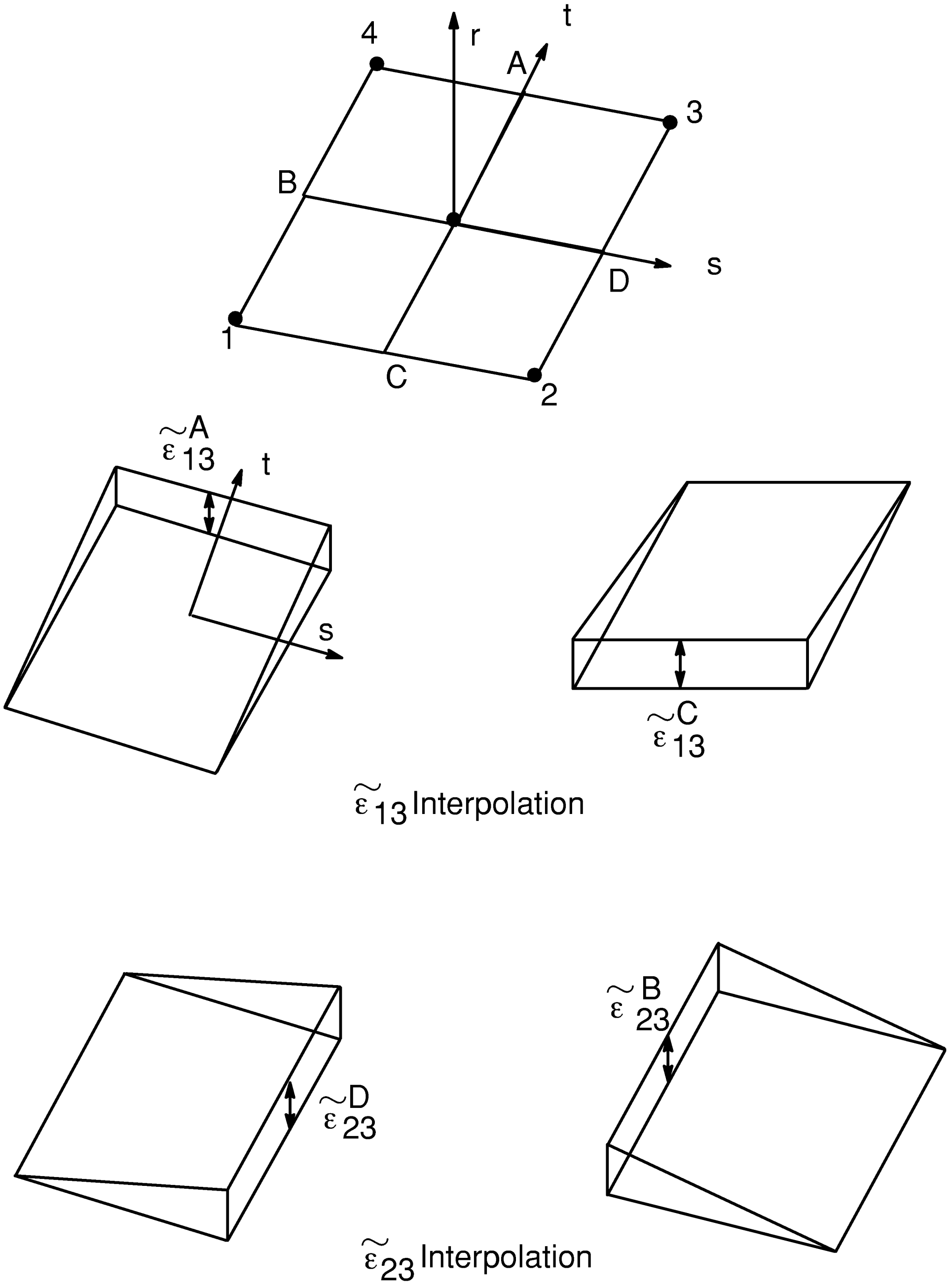
E_x = Young’s modulus in element x–direction (input as EX on **MP** command)

ν_{xy} = Poisson’s ratio in element x–y plane (input as NUXY on **MP** command)

G_{xy} = shear modulus in element x–y plane (input as GXY on **MP** command)

G_{yz} = shear modulus in element y–z plane (input as GYZ on **MP** command)

G_{xz} = shear modulus in element x–z plane (input as GXZ on **MP** command)


4
r
t
A
3
B
s
D
1
C
2
$\tilde{\varepsilon}_{13}^{A}$
t
s
$\tilde{\varepsilon}_{13}^{C}$
$\tilde{\varepsilon}_{23}^{B}$
$\tilde{\varepsilon}_{23}^{D}$

$\tilde{\varepsilon}_{13}$ Interpolation

$\tilde{\varepsilon}_{23}$ Interpolation

Figure 14.43–1 Shape Functions for the Transverse Strains

14.43.5 In–Plane Rotational DOF

If KEYOPT(3) is 0 or 1, there is no significant stiffness associated with the in–plane rotation DOF (rotation about the element r axis). A nominal value of stiffness is present (as described with SHELL63), however, to prevent free rotation at the node. KEYOPT(3) = 2 is used to include the Allman–type rotational DOFs (as described by

Allman(113) and Cook(114)). Such rotations improve the in–plane and general 3–D shell performance of the element. However, one of the outcomes of using the Allman rotation is that the element stiffness matrix contains up to two spurious zero energy modes (discussed below).

14.43.6 Spurious Mode Control with Allman Rotation

The first spurious mode is associated with constant rotations (Figure 14.43–2). The second spurious mode coincides with the well–known hourglass mode induced by reduced order integration (Figure 14.43–3). It is interesting to note that the hourglass spurious mode is elastically restrained for nonrectangular and multi–element configurations.

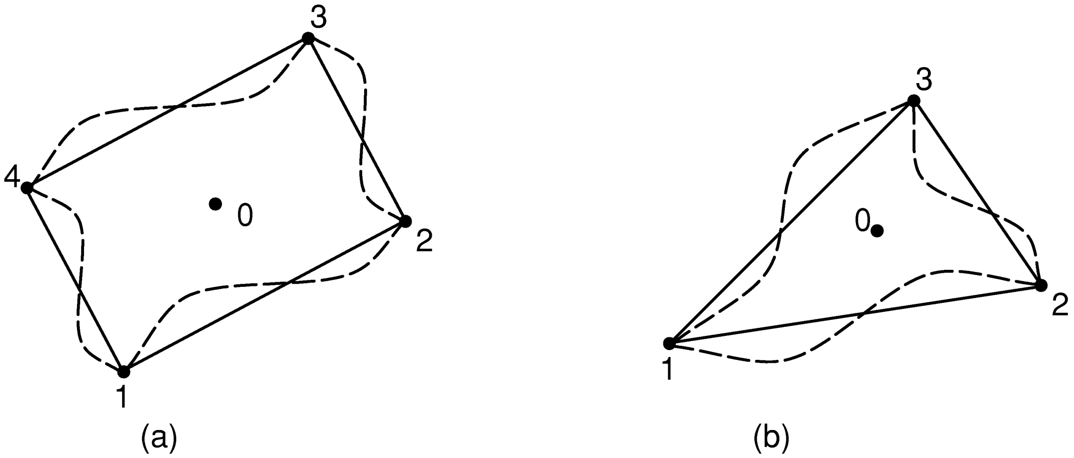


Figure 14.43–2 Constant in–plane rotation spurious mode ($\theta_{z1} = \theta_{z2} = \theta_{z3} = \theta_{z4}$)

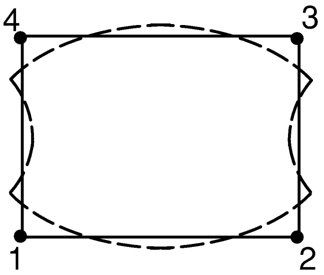


Figure 14.43–3 Hourglass mode ($\theta_{z1} = -\theta_{z2} = \theta_{z3} = -\theta_{z4}$)

The spurious modes are controlled on an elemental level using the concept suggested by MacNeal and Harder(115). For the constant rotation (Figure 14.43–2) spurious mode control, an energy penalty is defined as:

$$P_I = \delta_1 V \, \theta_I \, G_{xy} \, \theta_I \tag{14.43–2}$$

where: P_I = energy penalty I

δ_1 = penalty parameter (input quantity ZSTIF1 on **R** command)

V = element volume

G_{xy} = shear modulus (input on **MP** command)

θ_I = relative rotation, defined below

The relative rotation is computed at the element center as,

$$\theta_I = \theta_o - \frac{1}{n} \sum_{i=1}^{n} \theta_{zi} \tag{14.43–3}$$

where: $\theta_o = \frac{1}{2} \left(\left.\frac{\partial v}{\partial x}\right|_o - \left.\frac{\partial u}{\partial y}\right|_o \right)$

u, v = in–plane motions assuming edges remain straight

θ_{zi} = in–plane rotation at node i

n = number of nodes per element

$\Big|_o$ = evaluated at center of element

For the hourglass spurious modes which occur only for 4–noded elements, the energy penalty is taken as the inner product of the constraint force vector and the alternating rotational mode shapes as,

$$P_{II} = \delta_2 V \, \theta_{II} \, G_{xy} \, \theta_{II} \tag{14.43–4}$$

where: P_{II} = energy penalty II

δ_2 = penalty parameter (input quantity ZSTIF2 on **RMORE** command)

$\theta_{II} = \frac{1}{4} \left(\theta_{z1} - \theta_{z2} + \theta_{z3} - \theta_{z4} \right)$

Once the energy penalties (P_I and P_{II}) are defined, the associated stiffness augmentations can be calculated as,

$$\left[K^e_{ij} \right]_a = \frac{\partial^2 P_I}{\partial u_i \, \partial u_j} + \frac{\partial^2 P_{II}}{\partial u_i \, \partial u_j} \tag{14.43–5}$$

where: u_i = nodal displacement vector

This augmented stiffness matrix when added to the regular element stiffness matrix results in an effective stiffness matrix with no spurious modes.

14.43.7 Natural Space Extra Shape Functions with Allman Rotation

One of the outcomes of the Allman rotation is the dissimilar displacement variation along the normal and tangential directions of the element edges. The result of such variation is that the in–plane bending stiffness of the elements is too large by a factor $1/(1-\nu^2)$ and sometimes termed as Poisson's ratio locking. To overcome this difficulty, two natural space (s and t) nodeless in–plane displacement shape functions are added in the element stiffness matrix formulation and then condensed out at the element level. The element thus generated is free of Poisson's ratio locking. For details of a similar implementation, refer to Yunus et al (117).

14.43.8 Warping

A warping factor is computed as:

$$\phi = \frac{D}{t} \tag{14.43–6}$$

where: D = component of the vector from the first node to the fourth node parallel to the element normal

t = average thickness of the element

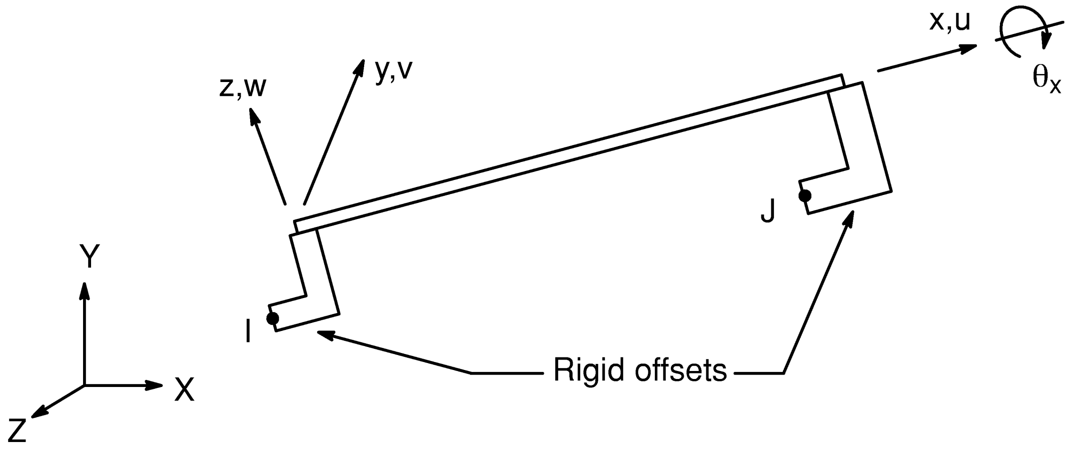
If $\phi > 1.0$, a warning message is printed.

14.43.9 Stress Output

The stresses at the center of the element are computed by taking the average of the four integration points on that plane.

The output forces and moments are computed as described in Section 2.3.

14.44 BEAM44 — 3–D Elastic Tapered Unsymmetrical Beam



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.2.2–1), (12.2.2–2), (12.2.2–3) and (12.2.2–4)	None
Foundation Stiffness Matrix	Same as stiffness matrix	None
Mass Matrix	If consistent mass matrix option is used (KEYOPT(2) = 0), same as stiffness matrix. If reduced mass matrix option is used (KEYOPT(2) = 1), equations (12.2.1–1), (12.2.1–2) and (12.2.1–3)	None
Stress Stiffness Matrix	Equations (12.2.2–2) and (12.2.2–3)	None
Load Vector (Pressures and Temperatures)	Equations (12.2.2–1), (12.2.2–2), and (12.2.2–3)	None

Load Type	Distribution
Element Temperature	Bilinear across cross–section, linear along length
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Linear along length

14.44.1 Other Applicable Sections

This element is an extension of BEAM4, so that the basic element formulation as well as the local to global matrix conversion logic is described in Section 14.4.

14.44.2 Assumptions and Restrictions

1. Normals before deformation remain straight and normal after deformation.
2. Offsets, if any, are assumed to be completely rigid.
3. If both offsets and also angular velocities or angular accelerations (input on **OMEGA**, **DOMEGA**, **CGOMGA**, or **DCGOMG** commands) are used, the radius used in the inertial force calculations does not account for the offsets.
4. Foundation stiffness effects are applied on the flexible length (i.e., before offsets are used).
5. Shear deflection effects are not included in the mass matrix, as they are for BEAM4.
6. Thermal bending assumes an (average) uniform thickness.

14.44.3 Tapered Geometry

When a tapered geometry is input, the program has no “correct” form to follow as the program does not know the shape of the cross–section. The supplied thicknesses are used only for thermal bending and stress evaluation. Consider the case of a beam with an area of 1.0 at one end and 4.0 at the other. Assuming all tapers are straight, the small end is a square, the large end is a 1.0 × 4.0 rectangular, and the midpoint of the beam would then have an area of 2.50. But if the large end is also square (2.0 × 2.0), the midpoint area would then be 2.25. Thus, there is no unique solution. All effects of approximations are reduced by ensuring that the ratios of the section properties are as close to 1.0 as possible. The discussion below indicates what is done for this element.

The stiffness matrix is the same as for BEAM4 (equation (14.4–1)), except that an averaged area is used:

$$A_{AV} = \left(A_1 + \sqrt{A_1 A_2} + A_2\right) / 3 \tag{14.44–1}$$

and all three moments of inertia use averages of the form:

$$I_{AV} = \left(I_1 + \sqrt[4]{I_1^{\,3} I_2} + \sqrt{I_1 I_2} + \sqrt[4]{I_1 I_2^{\,3}} + I_2\right) / 5 \tag{14.44–2}$$

The mass matrix is also the same as for BEAM4 (equation (14.4–2)), except the upper left quadrant uses section properties only from end I, the lower right quadrant uses section properties only from end J, and the other two quadrants use averaged values. For example, assuming no prestrain or added mass, the axial mass terms would be ϱA_1 L/3 for end I, ϱA_2 L/3 for end J, and $\varrho(A_1 + A_2)$ L/12 for both off–diagonal terms. Thus, the total mass of the element is: $\varrho(A_1 + A_2)$ L/2.

The stress stiffness matrix assumes a constant area as determined in equation (14.44–1).

Finally, the thermal load vector uses average thicknesses.

14.44.4 Shear Center Effects

The shear center effects affect only the torsional terms (M_x, θ_x). The rotation matrix $[R^s]$ (used below) is:

$$\left[R^s\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_1 & 0 & 0 \\ 0 & 0 & 0 & C_2 & 1 & 0 \\ 0 & 0 & 0 & C_3 & 0 & 1 \end{bmatrix} \tag{14.44–3}$$

where:

$$C_1 = \frac{L_{SC}}{L_G}$$

$$C_2 = -\frac{\Delta_y^s \; L_{SC}}{L_{SB} \; L_G}$$

$$C_3 = -\frac{\Delta_z^s}{L_{SB}}$$

$$L_{SC} = \sqrt{\left(L_G\right)^2 + \left(\Delta_y^s\right)^2 + \left(\Delta_z^s\right)^2}$$

$$L_{SB} = \sqrt{\left(L_G\right)^2 + \left(\Delta_y^s\right)^2}$$

Δ^s_y = $\Delta^s_{y2} - \Delta^s_{y1}$

Δ^s_z = $\Delta^s_{z2} - \Delta^s_{z1}$

Δ^s_{y2} = input quantity DYSC2 on **RMORE** command

L_G = actual flexible length, as shown in Figure 14.44–1

Note that only rotation about the shear centerline (θ_x) is affected. The shear center translations at node I are accounted for by:

$$\left[T^s_I\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\Delta_{z1}{}^s & 0 & 0 \\ 0 & 0 & 1 & \Delta_{y1}{}^s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (14.44\text{–}4)$$

A similar matrix $\left[T^s_J\right]$ is defined at node J based on Δ^s_{y2} and Δ^s_{z2}. These matrices are then combined to generate the [S_c] matrix:

$$\left[S_c\right] = \begin{bmatrix} \left[R^s T^s_I\right] & [0] \\ [0] & \left[R^s T^s_J\right] \end{bmatrix} \qquad (14.44\text{–}5)$$

This combination of [R] and [T] results because shear center offsets are measured in the element coordinate system (x^e y^e z^e in Figure 14.44–1). The element matrices are then transformed by

$$[K'_\ell] = [S_c]^T[K_\ell][S_c] \qquad (14.44\text{–}6)$$

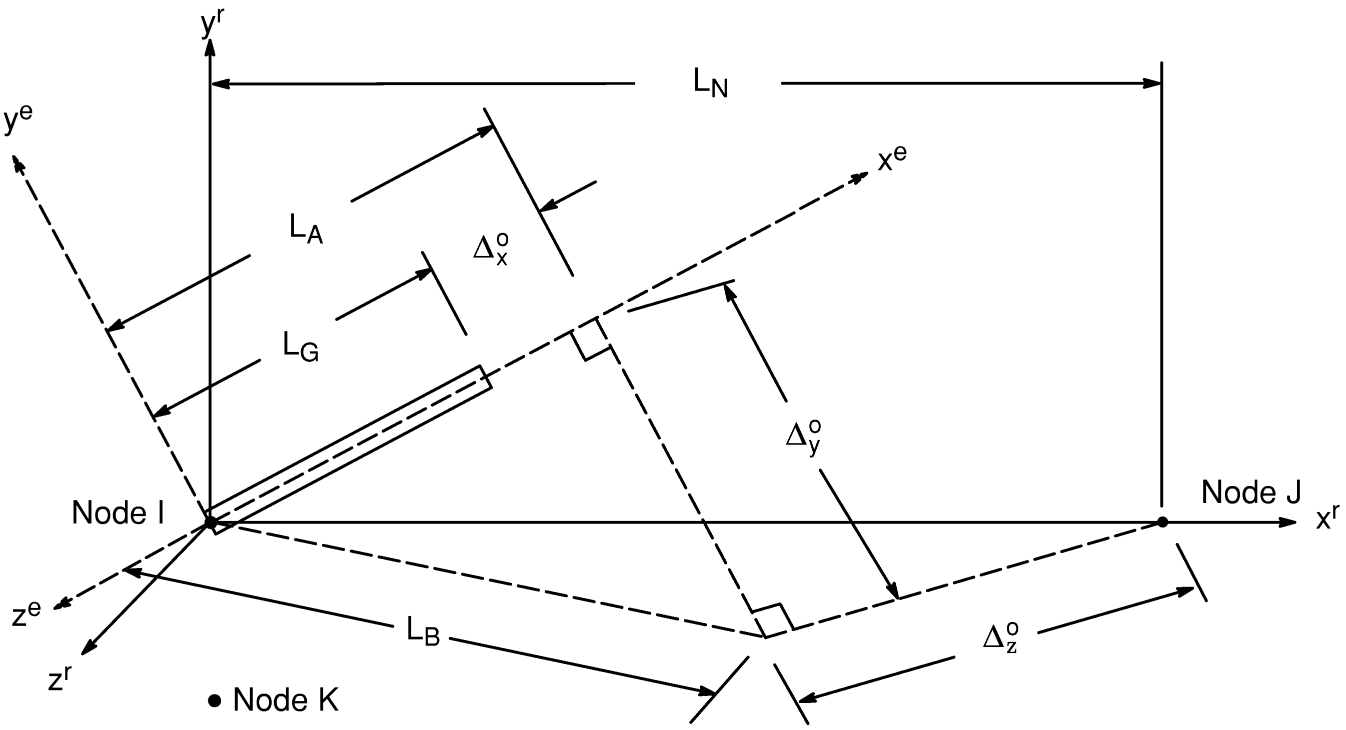
$$[S'_\ell] = [S_c]^T[S_\ell][S_c] \qquad (14.44\text{–}7)$$

$$\{F'_\ell\} = [S_c]^T\{F_\ell\} \qquad (14.44\text{–}8)$$

where: $\left[K_\ell\right]$ = element stiffness matrix in element (centroidal) coordinate system, similar to equation (14.4–1)

$\left[S_\ell\right]$ = element stress stiffness matrix in element (centroidal) coordinate system

$\left\{F_\ell\right\}$ = element load vector in element (centroidal) coordinate system, similar to equation (14.4–4).



1. Nodes I and J define the x^r axis
2. Nodes I, J, and K define the plane of the z^r axis
3. The z^e axis, as well as the Δ_z^o offset, lie parallel to the x^r–z^r plane
4. L_G is the flexible length

Figure 14.44–1 Offset Geometry

14.44.5 Offset at the Ends of the Member

It is convenient to define

$$\Delta_x^o = \Delta_{x2} - \Delta_{x1}$$

$$\Delta_y^o = \Delta_{y2} - \Delta_{y1}$$

$$\Delta_z^o = \Delta_{z2} - \Delta_{z1}$$

where: Δ_{x2} = input quantity DX2 on **RMORE** command

These definitions of Δ_i^o may be thought of as simply setting the offsets at node I to zero and setting the differential offset to the offset at node J, as shown in Figure 14.44–1. The rotation matrix $[R^o]$ implied by the offsets is defined by:

$$\lfloor u_x^e \; u_y^e \; u_z^e \; \theta_x^e \; \theta_y^e \; \theta_z^e \rfloor^T = [R^o] \lfloor u_x^r \; u_y^r \; u_z^r \; \theta_x^r \; \theta_y^r \; \theta_z^r \rfloor^T \qquad (14.44–9)$$

where: u_x^e, u_y^e, etc. = are in the element coordinate system

u_x^r, u_y^r, etc. = are in the reference coordinate system defined by the nodes

$$[R^o] = \begin{bmatrix} [r^o] & [0] \\ [0] & [r^o] \end{bmatrix}$$

$$[r^o] = \begin{bmatrix} \frac{L_A}{L_N} & \frac{\Delta_y^o}{L_B} & \frac{L_A \, \Delta_z^o}{L_N \, L_B} \\ -\frac{\Delta_y^o}{L_N} & \frac{L_A}{L_B} & -\frac{\Delta_y^o \; \Delta_z^o}{L_N \, L_B} \\ -\frac{\Delta_z^o}{L_N} & 0 & \frac{L_B}{L_N} \end{bmatrix}$$

To account for the translation of forces and moments due to offsets at node I, matrix $\left[T_i^o\right]$ is defined using Figure 14.44–2.

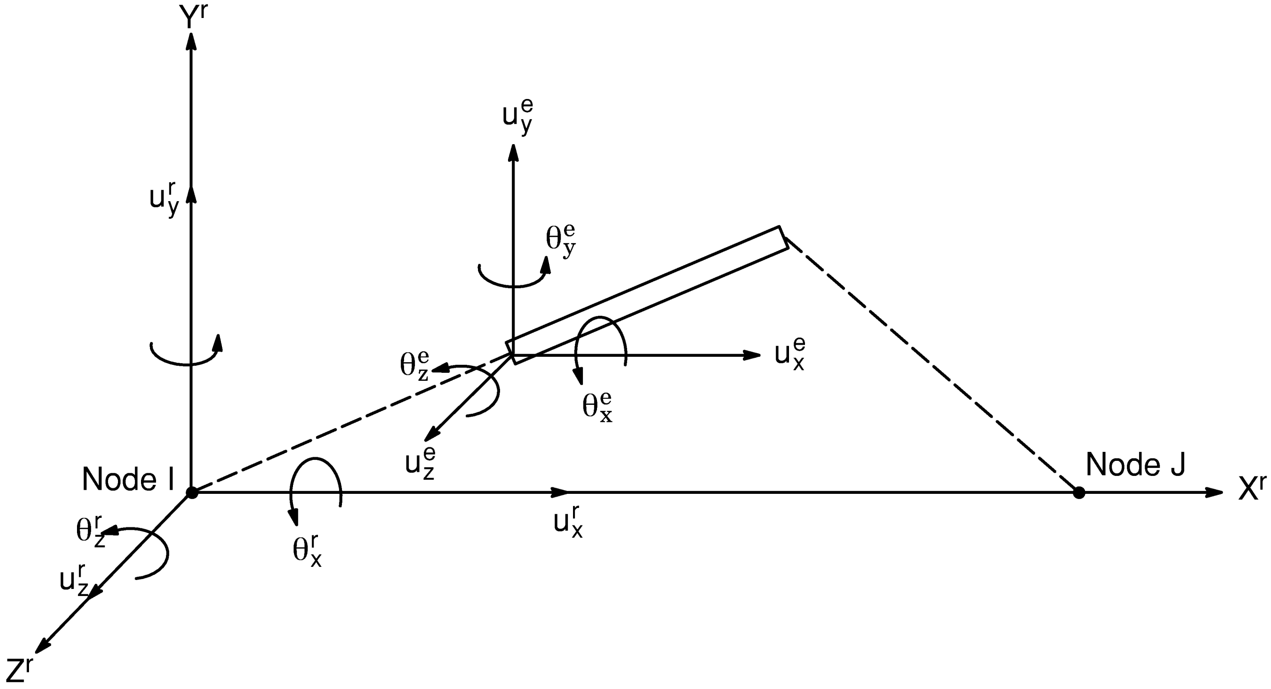


Figure 14.44–2 Translation of Axes

The two systems are related by:

$$\lfloor u^e_x \quad u^e_y \quad u^e_z \quad \theta^e_x \quad \theta^e_y \quad \theta^e_z \rfloor^T = \left[T^o_I\right] \lfloor u^r_x \quad u^r_y \quad u^r_z \quad \theta^r_x \quad \theta^r_y \quad \theta^r_z \rfloor^T \qquad (14.44\text{–}10)$$

where:

$$\left[T^o_I\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta_{z1} & -\Delta_{y1} \\ 0 & 1 & 0 & -\Delta_{z1} & 0 & \Delta_{x1} \\ 0 & 0 & 1 & \Delta_{y1} & -\Delta_{x1} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A similar matrix $\left[T^o_J\right]$ is defined at node J, based on Δ_{x2}, Δ_{y2}, and Δ_{z2}. These matrices are then combined to generate the $[O_F]$ matrix:

$$\left[O_F\right] = \begin{bmatrix} \left[T^o_I\right]\left[R^o\right] & [0] \\ [0] & \left[T^o_J\right]\left[R^o\right] \end{bmatrix} \qquad (14.44\text{–}11)$$

The basis for the above transformations is taken from Hall and Woodhead(15). The element matrices are then transformed again by:

$$[K''_\ell] = [O_F]^T[K'_\ell][O_F] \qquad (14.44\text{–}12)$$

$$[S''_\ell] = [O_F]^T[S'_\ell][O_F] \qquad (14.44\text{–}13)$$

$$[M'_\ell] = [O_F]^T[M_\ell][O_F] \qquad (14.44\text{–}14)$$

$$\{F''_\ell\} = [O_F]^T\{F'_\ell\} \qquad (14.44\text{–}15)$$

where: $[M_\ell]$ = element mass matrix in element (centroidal) coordinate system, similar to equation (14.4–2).

14.44.6 End Moment Release

End moment release (or end rotational stiffness release) logic is activated if either KEYOPT(7) or KEYOPT(8) > 0. The release logic is analogous to that discussed in

Section 17.6, with the dropped rotational DOF represented by the slave DOF. The processing of the matrices may be symbolized by:

$$[K_\ell''] => [K_\ell''] \quad \text{using static condensation (equation (17.6–8))} \tag{14.44–16}$$

$$[S_\ell''] => [S_\ell''] \begin{cases} \text{using Guyan reduction (equation (17.6–19)) if} \\ \textit{Type} = \text{BUCKLE on the } \textbf{ANTYPE} \text{ command} \\ \text{(case of linear buckling)} \\ \text{using static condensation (equation (17.6–8))} \\ \text{after being combined with } [K_\ell''] \text{ if } \textit{Type} \neq \text{BUCKLE} \\ \text{on the } \textbf{ANTYPE} \text{ command (cases other than linear} \\ \text{buckling)} \end{cases} \tag{14.44–17}$$

$$[M_\ell'] => [M_\ell'] \quad \text{using Guyan reduction (equation (17.6–19))} \tag{14.44–18}$$

$$\{F_\ell''\} => \{F_\ell''\} \quad \text{using static condensation (equation (17.6–9))} \tag{14.44–19}$$

14.44.7 Local to Global Conversion

The generation of the local to global transformation matrix $[T_R]$ is discussed in Section 14.4. Thus, the final matrix conversions are:

$$[K_e] = [T_R]^T[K_\ell''][T_R] \tag{14.44–20}$$

$$[S_e] = [T_R]^T[S_\ell''][T_R] \tag{14.44–21}$$

$$[M_e] = [T_R]^T[M_\ell'][T_R] \tag{14.44–22}$$

$$\{F_e\} = [T_R]^T\{F_\ell''\} \tag{14.44–23}$$

14.44.8 Stress Calculations

The axial stresses are computed analogously to BEAM4. The maximum stress at cross section i is computed by:

$$\sigma_i^{max} = \text{maximum of} \begin{cases} \sigma_i^{dir} + \sigma_{zt,i}^{bnd} + \sigma_{yt,i}^{bnd} \\ \sigma_i^{dir} + \sigma_{zt,i}^{bnd} + \sigma_{yb,i}^{bnd} \\ \sigma_i^{dir} + \sigma_{zb,i}^{bnd} + \sigma_{yb,i}^{bnd} \\ \sigma_i^{dir} + \sigma_{zb,i}^{bnd} + \sigma_{yt,i}^{bnd} \end{cases} \tag{14.44–24}$$

where:
σ^{dir} = output quantity SDIR
σ_{yt}^{bnd} = output quantity SBYT
σ_{yb}^{bnd} = output quantity SBYB
σ_{zt}^{bnd} = output quantity SBZT
σ_{zb}^{bnd} = output quantity SBZB

The minimum stress is analogously defined..

The shear stresses are computed as:

$$\tau_L^y = \frac{F^y}{A_s^y} \tag{14.44–25}$$

$$\tau_L^z = \frac{F^z}{A_s^z} \tag{14.44–26}$$

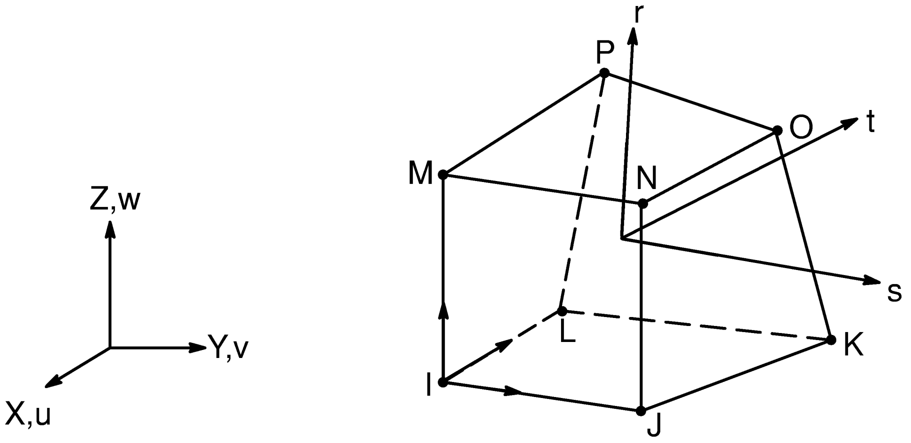
where:
$\tau_L^y,\ \tau_L^z$ = transverse shear stress (output quantities SXY, SXZ)
$F^y,\ F^z$ = transverse shear forces
$A_s^y,\ A_s^z$ = transverse shear areas (input quantities ARESZ1, etc. on **RMORE** command)

and

$$\tau_T = M_x\ C \tag{14.44–27}$$

where:
τ_T = torsional shear stress (output quantity SYZ)
M_x = torsion moment
C = user–supplied constant (input quantities TSF1 and TSF2 on **RMORE** command)

14.45 SOLID45 — 3–D Structural Solid



Matrix or Vector	Shape Functions		Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3) or, if modified extra shape functions are included (KEYOPT(1)=0) and element has 8 unique nodes, (12.8.19–1), (12.8.19–2), and (12.8.19–3)		2 x 2 x 2 if KEYOPT(2) = 0 1 if KEYOPT(2) = 1
Mass Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3)		Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Quad	Equation (12.5.8–1) and (12.5.8–2)	2 x 2
	Triangle	Equation (12.5.1–1) and (12.5.1–2)	3

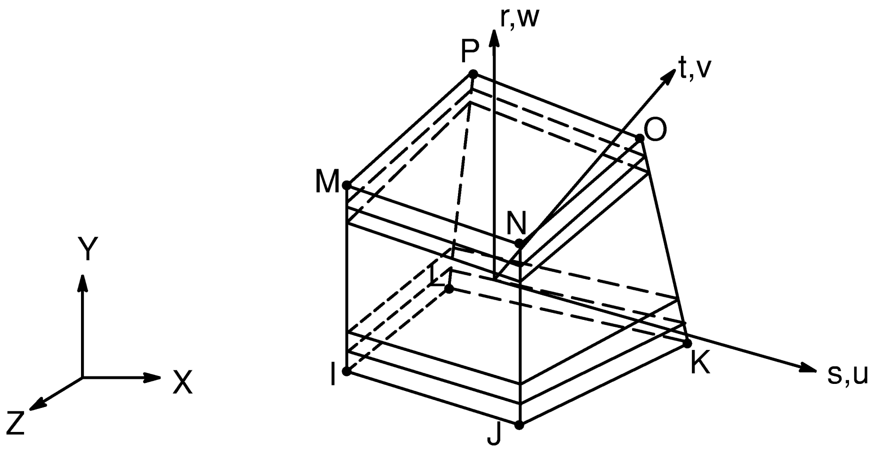
Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

Reference: Wilson(38), Taylor et al(49)

14.45.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. Uniform reduced integration technique (Flanagan and Belytschko(232)) can be chosen by using KEYOPT(2) = 1.

14.46 SOLID46 — 3–D Layered Structural Solid



Matrix or Vector	Shape Functions		Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3) and, if modified extra shape functions are included (KEYOPT(1)≠1) and element has 8 unique nodes (12.8.19–1), (12.8.19–2), and (12.8.19–3)		2 x 2 x 2
Mass Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3)		2 x 2 x 2
Stress Stiffness Matrix	Same as mass matrix		2 x 2 x 2
Thermal Load Vector	Same as stiffness matrix		2 x 2 x 2
Pressure Load Vector	Quad	Equation (12.5.8–1) and (12.5.8–2)	2 x 2
	Triangle	Equation (12.5.1–1) and (12.5.1–2)	3

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Wilson(38), Taylor et al(49)

14.46.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. The theory of SOLID46 is analogous to that given with SHELL99 (Section 14.99), except as given in this section.

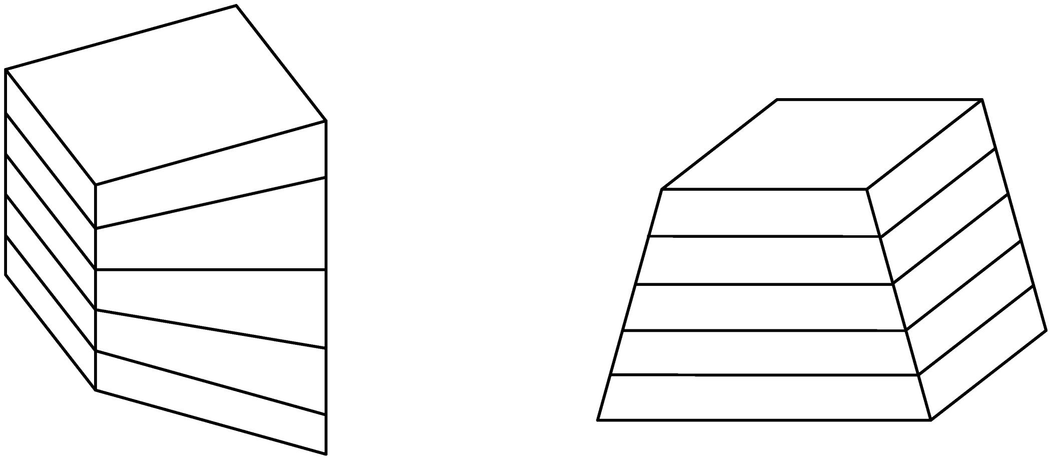
14.46.2 Assumptions and Restrictions

All material orientations are assumed to be parallel to the reference plane, even if the element has nodes inferring warped layers.

The numerical integration scheme for the thru–thickness effects are identical to that used in SHELL99. This may yield a slight numerical inaccuracy for elements having a significant change of size of layer area in the thru–thickness direction (see Figure 14.46–1). The main reason for such discrepancy stems from the approximation of the variation of the determinant of the Jacobian in the thru–thickness direction. The error is usually insignificant. However, users may want to try a patch–test problem to assess accuracy for their particular circumstances.

Unlike shell elements, SOLID46 cannot assume a zero transverse shear stiffness at the top and bottom surfaces of the element. Hence the interlaminar shear stress must be computed without using this assumption, which leads to relatively constant values thru the element.

The use of effective (“eff”) material properties developed below is based on heuristic arguments and numerical experiences rather than on a rigorous theoretical formulation. The fundamental difficulty is that multilinear displacement fields are attempted to be modeled by a linear (or perhaps quadratic) displacement shape function since the number of DOF per element must be kept to a minimum. A more rigorous solution can always be obtained by using more elements in the thru–the–layer direction. Numerical experimentation across a variety of problems indicates that the techniques used with SOLID46 give reasonable answers in most cases.



Parallel Faces at End
of Layers (Thru–thickness
integration close to exact)

Non–parallel Faces at End
of Layers (Thru–thickness
integration with slight
inaccuracy)

Figure 14.46–1 Offset Geometry

14.46.3 Stress–Strain Relationships

For layer j, the stress–strain relationships in the layer coordinate system are (from equations (2.1–12) thru (2.1–17):

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \epsilon_{xy} \\ \epsilon_{yz} \\ \epsilon_{xz} \end{Bmatrix} = \begin{Bmatrix} \alpha_{x,j}\Delta T \\ \alpha_{y,j}\Delta T \\ \alpha_{z,j}\Delta T \\ 0 \\ 0 \\ 0 \end{Bmatrix} + \begin{bmatrix} \frac{1}{E_{x,j}} & \frac{-\nu_{xy,j}}{E_{y,j}} & \frac{-\nu_{xz,j}}{E_{z,j}} & 0 & 0 & 0 \\ \frac{-\nu_{xy,j}}{E_{y,j}} & \frac{1}{E_{y,j}} & \frac{-\nu_{yz,j}}{E_{z,j}} & 0 & 0 & 0 \\ \frac{-\nu_{xz,j}}{E_{z,j}} & \frac{-\nu_{yz,j}}{E_{z,j}} & \frac{1}{E_{z,j}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy,j}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz,j}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xz,j}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} \tag{14.46–1}$$

where:

$\alpha_{x,j}$ = coefficient of thermal expansion of layer j in the layer x–direction (input as ALPX on **MP** command)

$E_{x,j}$ = Young’s modulus of layer j in the layer x–direction (input as EX on **MP** command)

$G_{xy,j}$ = shear modulus of layer j in layer x–y plane (input as GXY on **MP** command)

$\nu_{xy,j}$ = Poisson’s ratio of layer j in x–y plane (input as NUXY on **MP** command)

ΔT = $T - T_{ref}$

T = temperature at point being studied

T_{ref} = reference temperature (input on **TREF** command)

To help ensure continuity of stresses between the layers, equation (14.46–1) is modified to yield:

$$\begin{Bmatrix}\epsilon_x\\ \epsilon_y\\ \epsilon_z\\ \epsilon_{xy}\\ \epsilon_{yz}\\ \epsilon_{xz}\end{Bmatrix} = \begin{Bmatrix}\alpha_{x,j}\Delta T\\ \alpha_{y,j}\Delta T\\ \alpha_z^{eff}\Delta T\\ 0\\ 0\\ 0\end{Bmatrix} + \begin{bmatrix} \frac{1}{E_{x,j}} & \frac{-\nu_{xy,j}}{E_{y,j}} & \frac{-\nu_{xz,j}^{eff}}{E_z^{eff}} & 0 & 0 & 0\\ \frac{-\nu_{xy,j}}{E_{y,j}} & \frac{1}{E_{y,j}} & \frac{-\nu_{yz,j}^{eff}}{E_z^{eff}} & 0 & 0 & 0\\ \frac{-\nu_{xz,j}^{eff}}{E_z^{eff}} & \frac{-\nu_{yz,j}^{eff}}{E_z^{eff}} & \frac{1}{E_z^{eff}} & 0 & 0 & 0\\ 0 & 0 & 0 & \frac{1}{G_{xy,j}} & 0 & 0\\ 0 & 0 & 0 & 0 & D_{11,j}^G & D_{21,j}^G\\ 0 & 0 & 0 & 0 & D_{12,j}^G & D_{22,j}^G \end{bmatrix}\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \sigma_z\\ \sigma_{xy}\\ \sigma_{yz}\\ \sigma_{xz}\end{Bmatrix} \qquad (14.46\text{–}2)$$

where:

$$\alpha_z^{eff} = \frac{\sum_{j=1}^{N\ell} t_j \ \alpha_{z,j}}{t_{TOT}}$$

where:

$$\nu_{xz,j}^{eff} = \begin{cases} C & \text{if } C < .45\\ \text{or} & \\ \nu_{xz,j} & \text{if } C \geq .45 \end{cases}$$

$$C = \nu_{xz,j}\frac{E_z^{eff}}{E_{z,j}}$$

$$E_z^{eff} = \frac{t_{TOT}}{\sum_{j=1}^{N\ell}\frac{t_j}{E_{z,j}}}$$

$$[D^G]_j = \begin{bmatrix} D_{11,j}^G & D_{21,j}^G\\ D_{12,j}^G & D_{22,j}^G\end{bmatrix} = \left([T]_j^{-1}\right)^T\left[d^G\right]\ [T]_j^{-1} = \text{effective inverted shear moduli in layer system}$$

$[d^G]$ = $\frac{1}{t_{TOT}} \sum_{j=1}^{N\ell} t_j \left[A_\ell\right]_j^{-1}$ = effective inverted shear moduli in element system

$\left[A_\ell\right]_j$ = $[T]_j^T [D_z]_j [T]_j$

$[T]_j$ = transformation matrix to convert from layer to element systems

$[D_z]_j$ = $\begin{bmatrix} G_{yz,j} & 0 \\ 0 & G_{xz,j} \end{bmatrix}$

t_j = average thickness of layer j

t_{TOT} = average total thickness of element

N_ℓ = number of layers

14.46.4 General Strain and Stress Calculations

The following steps are used to compute strains and stresses at a typical point within layer j:

1. The strain vector $\left(\lfloor \epsilon_x,\ \epsilon_y,\ \epsilon_z,\ \epsilon_{xy},\ \epsilon_{yz},\ \epsilon_{xz} \rfloor\right)$ is determined from the displacements and the strain–displacement relationships, evaluated at the point of interest.
2. The strains are converted from element to layer coordinates.
3. The strains are adjusted for thermal effects, with the effective coefficient of thermal expansion in the z–direction being:

$$\alpha_z^{eff} = \frac{\sum_{j=1}^{N\ell} t_j\ \alpha_{z,j}}{t_{TOT}} \qquad (14.46–3)$$

4. The normal strain is then adjusted with

$$\epsilon_z^{'} = \epsilon_z \frac{E_z^{eff}}{E_{z,j}} \qquad (14.46–4)$$

5. The transverse shear strains are computed by way of the stresses all in the layer coordinate system.

$$\begin{Bmatrix} \sigma_{yz} \\ \sigma_{xz} \end{Bmatrix} = \left[D^G\right]^{-1} \begin{Bmatrix} \epsilon_{yz} \\ \epsilon_{xz} \end{Bmatrix} \qquad (14.46–5)$$

where: $\varepsilon_{yz}, \varepsilon_{xz}$ = shear strains based on strain–displacement relationships

Then,

$$\epsilon'_{yz,j} = \sigma_{yz} \;/\; G_{yz,j} \tag{14.46–6}$$

$$\epsilon'_{xz,j} = \sigma_{xz} \;/\; G_{xz,j} \tag{14.46–7}$$

where: $\epsilon'_{yz}, \epsilon'_{xz}$=shear strains based on continuity of shearing stresses

6. Finally, the strains are converted to stresses by the usual relationship:

$$\{\sigma\}_j = [D]_j \left(\{\epsilon\}_j - \left\{\epsilon^{th}\right\}_j\right) \tag{14.46–8}$$

where: $[D]_j$ = inverse of stress–strain matrix used in equation (14.46–1)

7. If the element has more than one layer and any layer has $\nu^{eff}_{xz,j}$ or $\nu^{eff}_{yz,j}$ exceeding 0.45, the normal stresses are computed based on nodal forces.

14.46.5 Interlaminar Shear Stress Calculation

It may be seen that the interlaminar shear stress will, in general, not be zero at a free surface. This is because the element has no knowledge as to whether or not the top or bottom face is a free surface or if there is another element attached to that face.

There are two methods of calculating interlaminar shear stress: by nodal forces and by evaluating stresses layer–by–layer.

Nodal Forces

The shear stresses over the entire volume are computed to be:

$$\sigma_{xz} = \frac{1}{4}\left[\frac{F^x_M - F^x_I}{A^{I-M}} + \frac{F^x_N - F^x_J}{A^{J-N}} + \frac{F^x_O - F^x_K}{A^{K-O}} + \frac{F^x_P - F^x_L}{A^{L-P}}\right] \tag{14.46–9}$$

$$\sigma_{yz} = \frac{1}{4}\left[\frac{F^y_M - F^y_I}{A^{I-M}} + \frac{F^y_N - F^y_J}{A^{J-N}} + \frac{F^y_O - F^y_K}{A^{K-O}} + \frac{F^y_P - F^y_L}{A^{L-P}}\right] \tag{14.46–10}$$

where: σ_{xz}, σ_{yz} = output quantities AVERAGE TRANSVERSE SHEAR STRESS COMPONENTS

F_I^x, F_I^y, etc. = forces at node I (etc.) parallel to the reference plane, with x being parallel to the element x direction

A^{I-M}, etc. = tributary area for node (evaluated by using the determinant of the Jacobian at the nearest integration point in base plane)

Evaluating Stresses Layer–by–Layer

This option is available only if KEYOPT(2) = 0 or 1 and simply uses the layer shear stresses for the interlaminar stresses. Thus, the interlaminar shear stresses in the element x direction are:

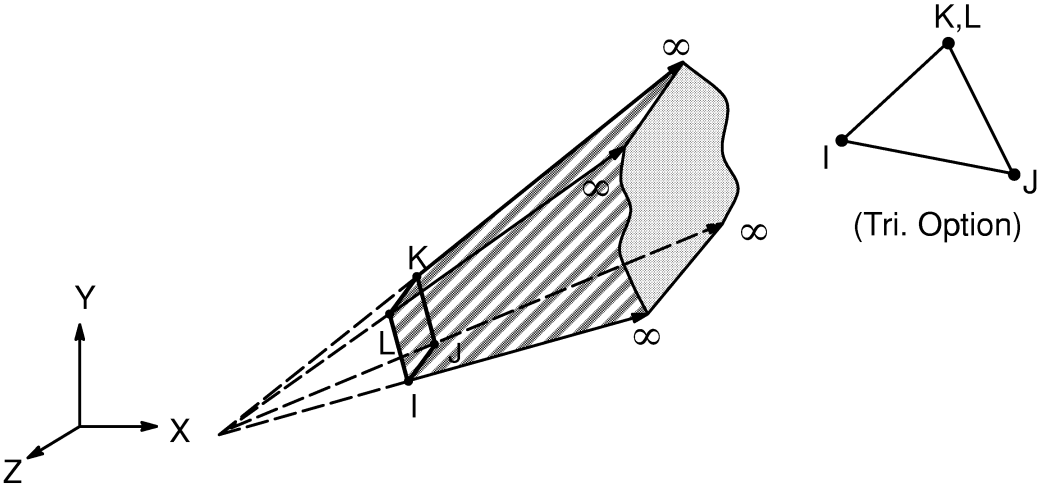
$$\sigma_{xz}^{1} = \sigma_{xz} \text{ at bottom of layer 1 (in plane I–J–K–L)} \quad (14.46\text{–}11)$$

$$\sigma_{xz}^{N_\ell+1} = \sigma_{xz} \text{ at top of layer NL (in plane M–N–O–P)} \quad (14.46\text{–}12)$$

$$\sigma_{xz}^{j} = \frac{1}{2}\,(\sigma_{xz} \text{ at top of layer j–1} + s_{xz} \text{ at bottom of layer j}) \quad \text{where } 1 < j < N_\ell \quad (14.46\text{–}13)$$

The σ_{xz} terms are the shear stresses computed from equation (14.46–8), except that the stresses have been converted to element coordinates. The interlaminar shear stresses in the element y direction are analogous. The maximum of the vector sum of σ_{xz} and σ_{yz} for all layers is printed with the label “MAXIMUM INTERLAMINAR SHEAR STRESS”.

14.47 INFIN47 — 3–D Infinite Boundary



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix or Thermal Conductivity Matrix	$\phi = N_I\phi_I + N_J\phi_J + N_K\phi_K,$ $N_I = \frac{1}{2A_o}\left[(x_J y_K - x_K y_J)\right.$ $\left.- (y_K - y_J)x + (x_K - x_J)y\right]$ $N_J = \frac{1}{2A_o}\left[(x_K y_I - x_I y_K)\right.$ $\left.- (y_I - y_K)x + (x_I - x_K)y\right]$ $N_K = \frac{1}{2A_o}\left[(x_I y_J - x_J y_I)\right.$ $\left.- (y_J - y_I)x + (x_J - x_I)y\right]$ A_o = area of triangle IJK	None on the boundary element IJK itself, however, 16–point 1–D Gaussian quadrature is applied for some of the integration on each of the edges IJ, JK, and KI of the infinite elements IJML, JKNM, and KILN (see Figure 14.47–1)

Reference: Kaljevic′, et al(130)

14.47.1 Introduction

This boundary element (BE) models the exterior infinite domain of the far–field magnetic and thermal problems. This element is to be used in combination with 3–D scalar potential solid elements, and can have magnetic scalar potential (MAG), or temperature (TEMP) as the DOF.

14.47.2 Theory

The formulation of this element is based on a first order triangular infinite boundary element (IBE), but the element can be used as a 4–noded quadrilateral as well. For unbounded field problems, the model domain is set up to consist of an interior volumetric finite element domain, Ω_F, and a series of exterior volumetric BE subdomains, Ω_B, as shown in Figure 14.47–1. Each subdomain, Ω_B, is treated as an ordinary BE domain consisting of five segments: the boundary element IJK, infinite elements IJML, JKNM and KILN, and element LMN; element LMN is assumed to be located at infinity (Figure 14.47–1).

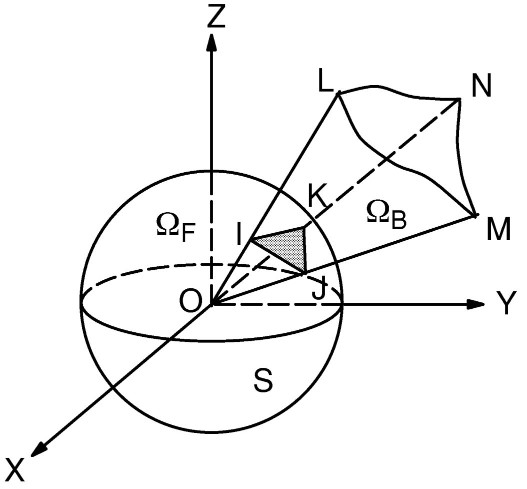


Figure 14.47–1 A Semi–infinite Boundary Element Zone and the Corresponding Boundary Element IJK

The approach used here is to write BE equations for Ω_B, and then convert them into equivalent load vectors for the nodes I, J and K. The procedure consists of four steps that are summarized below (see (130) for details).

First, a set of boundary integral equations is written for Ω_B. To achieve this, the potential (or temperature) and its normal derivatives (fluxes) are interpolated on the triangle IJK (Figure 14.47–1) by linear shape functions:

$$\phi(x, y) = N_I \ \phi_I + N_J \ \phi_J \ N_K \ \phi_K \qquad (14.47\text{–}1)$$

$$q_n(x, y) = N_I \, q_{nI} + N_J \, q_{nJ} + N_K \, q_{nK} \tag{14.47–2}$$

where:

ϕ = potential (or temperature)

q_n = $\frac{\partial \phi}{\partial n}$ = normal derivative (or flux)

N_I, N_J, N_K = linear shape functions defined earlier

ϕ_I, ϕ_J, ϕ_K = nodal potentials (or temperatures)

q_{nI}, q_{nJ}, q_{nK} = nodal normal derivatives (or fluxes)

n = normal to the surface IJK

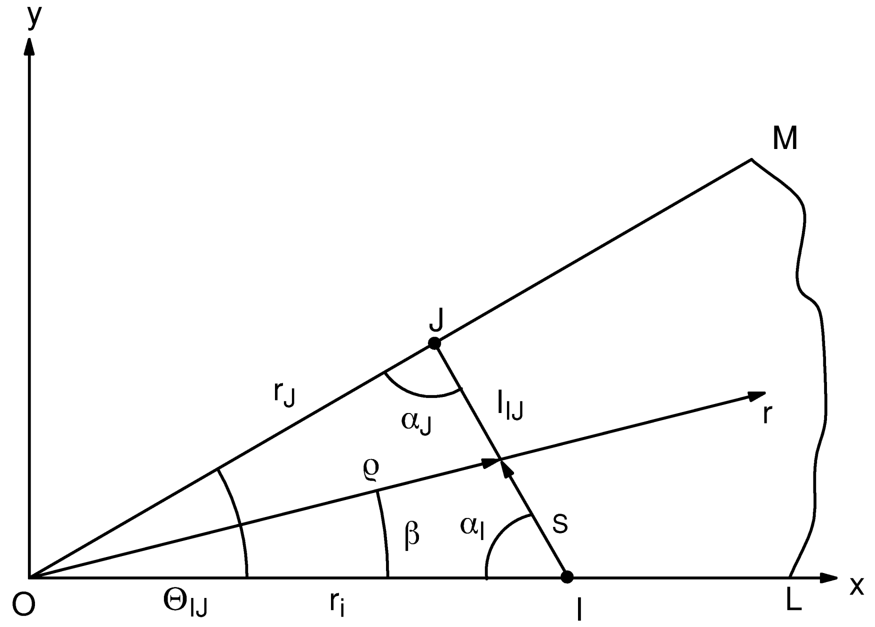


Figure 14.47–2 Infinite Element IJML (see Figure 14.47–1) and the Local Coordinate System

Over an infinite element, such as IJML (Figure 14.47–2), the dependent variables, i.e., potentials (or temperatures) and their normal derivatives (fluxes) are respectively assumed to be (Figure 14.47–2):

$$\phi(r, \beta) = \left\{ \left(1 - \frac{s}{L_{IJ}} \right) \phi_I + \left(\frac{s}{L_{IJ}} \right) \phi_J \right\} \left(\frac{\rho}{r} \right)^2 \tag{14.47–3}$$

$$q_\tau(r, \beta) = \left\{ \left(1 - \frac{s}{L_{IJ}} \right) q_{\tau J} + \left(\frac{s}{L_{IJ}} \right) q_{\tau J} \right\} \left(\frac{\rho}{r} \right)^3 \tag{14.47–4}$$

where: q_τ = $\frac{\partial \phi}{\partial \tau}$ = normal derivative (or flux) to infinite elements, e.g., IJML (Figure 14.47–2)

$q_{\tau I}, q_{\tau J}$ = nodal (nodes I and J) normal derivatives for infinite element IJML

s = a variable length from node I towards node J

L_{IJ} = length of edge IJ

ϱ = radial distance from the origin of the local coordinate system O to the edge IJ

r = radial distance from the edge IJ towards infinity

β = variable angle from x–axis for local polar coordinate system

τ = normal to infinite elements, viz., IJML

The boundary integral equations for Ω_B are now written as:

$$c(\xi)\phi(\xi) = \int_{\Gamma_B} \left[G(x,\xi)\, q(x) - F(x,\xi)\, \phi(x) \right] d\Gamma(x) \tag{14.47–5}$$

where: $c(\xi)$ = jump term in boundary element method

$G(x,\xi)$ = Green's function or fundamental solution for Laplace's equation

= $\frac{1}{4\pi k r}$

$F(x,\xi)$ = $\frac{\partial}{\partial n}[G(x,\xi)]$

(x,ξ) = field and source points, respectively

r = distance between field and source points

k = magnetic reluctivity (inverse of free space permeability input on **EMUNIT** command) for AZ DOF (KEYOPT(1)=0)

or

isotropic thermal conductivity (input as KXX on **MP** command) for TEMP DOF (KEYOPT(1)=1)

The integrations in equations (14.47–5) are performed in closed form on the boundary element IJK. The integrations on the infinite elements IJML, JKNM and KILN in the ‘r’ direction (Figure 14.47–2) are also performed in closed form. However, a 16–point Gaussian quadrature rule is used for the integrations on each of the edges IJ, JK and KI on the infinite elements.

Second, in the absence of a source or sink in Ω_B, the flux q(r) is integrated over the boundary Γ_B of Ω_B and set to zero:

$$\int_{\Gamma_B} q \, dr = 0 \qquad (14.47\text{–}6)$$

Third, geometric constraint conditions that exist between the potential ϕ (or temperature) and its derivatives $\frac{\partial \phi}{\partial n} = q_n$ and $\frac{\partial \phi}{\partial \tau} = q_\tau$ at the nodes I, J and K are written. These conditions would express the fact that the normal derivative q_n at the node I, say, can be decomposed into components along the normals to the two infinite elements IJML and KILN meeting at I and along OI.

Fourth, the energy flow quantity from Ω_B is written as:

$$w = \int_{\Gamma_B} q \, \phi \, dr \qquad (14.47\text{–}7)$$

This energy flow is equated to that due to an equivalent nodal force vector {F} defined below.

The four steps mentioned above are combined together to yield, after eliminating q_n and q_τ,

$$[K] \, \{\phi\} \equiv \{F\}_{eqv} \qquad (14.47\text{–}8)$$

where:
$[K]$ = 3 x 3 equivalent unsymmetric element coefficient matrix
$\{\phi\}$ = 3 x 1 nodal degrees of freedom, MAG or TEMP
$\{F\}_{eqv}$ = 3 x 1 equivalent nodal force vector

The coefficient matrix [K] multiplied by the nodal DOF's $\{\phi\}$ represents the equivalent nodal load vector which brings the effects of the semi–infinite domain Ω_B onto nodes I, J and K.

As mentioned in the beginning, the INFIN47 can be used with magnetic scalar potential elements to solve 3–D magnetic scalar potential problems (MAG degree of freedom). Magnetic scalar potential elements incorporate three different scalar potential formulations (see Section 5.1) selected with the **MAGOPT** command:

No.	Formulation	MAGOPT
1	Reduced Scalar Potential	0
2	Difference Scalar Potential	2 and 3
3	Generalized Scalar Potential	1, 2 and 3

Reduced Scalar Potential

If there is no “iron” in the problem domain, the reduced scalar potential formulation can be used both in the FE and the BE regimes. In this case, the potential is continuous across FE–BE interface. If there is “iron” in the FE domain, the reduced potential formulation is likely to produce “cancellation errors”.

Difference Scalar Potential

If there is “iron” and current in the FE region and the problem domain is singly–connected, we can use the difference potential formulation in order to avoid cancellation error. The formulation consists of two–step solution procedures:

1. **MAGOPT**, 2 Solution

   Here the first step consists of computing a magnetic field $\{H_o\}$ under the assumption that the magnetic permeability of iron is infinity, thereby neglecting any saturation. The reduced scalar potential ϕ is used in FE region and the total scalar potential ψ is used in BE region. In this case, the potential will be discontinuous across the FE–BE interface. The continuity condition of the magnetic field at the interface can be written as:

   $$-\nabla\psi\cdot\{\tau\} = -\nabla\phi\cdot\{\tau\} + \{H_s\}^T\{\tau\} \qquad (14.47–9)$$

   where: $\{\tau\}$ = tangent vector at the interface along element edge
   $\{H_s\}$ = magnetic field due to current sources

   Integrating the above equation along the interface, we obtain

   $$\psi_p = \phi_p - \int_{p_o}^{p} \{H_s\}^T\{\tau\}\, dt \qquad (14.47–10)$$

   If we take $\psi = \phi$ at a convenient point p_o on the interface, then the above equation defines the potential jump at any point p on the interface. Now, the total potential ψ can be eliminated from the problem using this equation, leading to the computation of the additional load vector,

   $$\{f_g\} = [K]\{g\} \qquad (14.47–11)$$

   where: $g_i = \int_{p_o}^{p_i} \{H_s\}^T\{\tau\}\, dt$
   $[K]$ = coefficient matrix defined with equation (14.47–8)

2. **MAGOPT**, 3 Solution

   In this step the total field, $\{H\} = \{H_o\} - \nabla\psi$, is computed where {H} is the actual field and $\{H_o\}$ is the field computed in step 1 above. Note that the same relation given in equation (5.1–38) uses ϕ_g in place of ψ. The total potential ψ is used in both FE and BE regimes. As a result, no potential discontinuity exists at the interface, but an additional load vector due to the field $\{H_o\}$ must be computed. Since the magnetic flux continuity condition at the interface of air and iron is:

$$\mu_I \frac{\partial\psi_I}{\partial n} - \mu_o \frac{\partial\psi_A}{\partial n} = - \mu_o \, \{H_o\}^T\{n\} \qquad (14.47\text{–}12)$$

   where: μ_o = magnetic permeability of free space (air)

   μ_I = magnetic permeability of iron

   The additional load vector may be computed as

$$\{f_f\} = - \int_s \mu_o \, \{N\}\{H_o\}^T\{n\} \, ds \qquad (14.47\text{–}13)$$

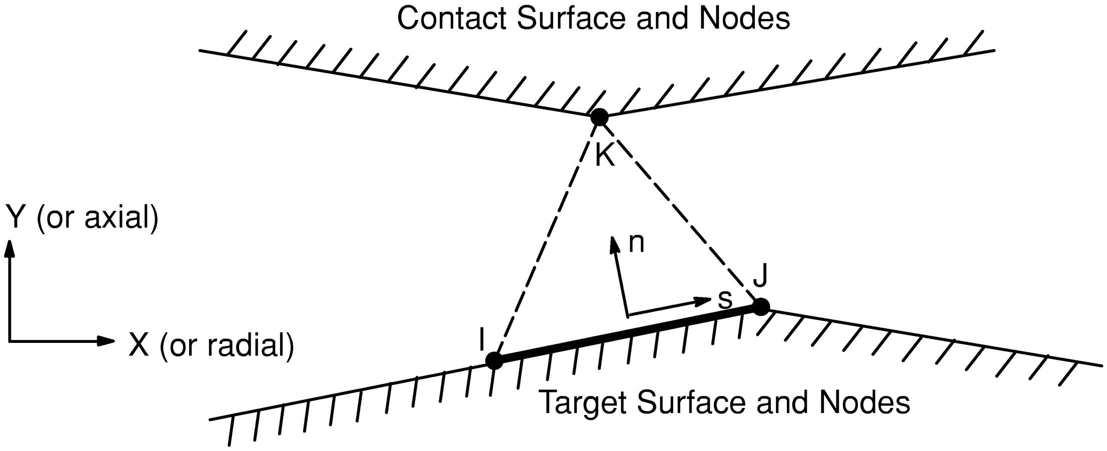
   where: {N} = weighting functions

Generalized Scalar Potential

If there is iron and current in the FE domain and the domain is multiply–connected, the generalized potential formulation can be used. It consists of three different steps.

1. **MAGOPT**, 1 Solution. This step computes magnetic field in the iron only. The boundary elements are not at all used for this step.
2. **MAGOPT**, 2 Solution. This step is exactly the same as the step 1 of the difference potential formulation.
3. **MAGOPT**, 3 Solution. This step is exactly the same as the step 2 of the difference potential formulation.

14.48 CONTAC48 — 2–D Point–to–Surface Contact



14.48.1 Introduction

CONTAC48 is a 3–node element that is intended for general contact analysis. In a general contact analysis, the area of contact between two (or more) bodies is generally not known in advance. In addition, the finite element models of the contacting bodies are generated is such a way that precise node–to–node contact is neither achievable nor desirable when contact is established. This type of contact situation precludes the use of node–to–node contact elements such as CONTAC12. CONTAC48 has the capability to represent general contact of models that are generated with arbitrary meshes. In other words, its use is not limited to known contact or node–to–node configurations.

CONTAC48 is applicable to 2–D geometries: plane strain, plane stress, or axisymmetry. It may be applied to contact of solid bodies or beams, to static or dynamic analyses, to problems with or without friction, and to flexible–to–flexible or rigid–to–flexible body contact. The combined mechanisms of structural contact and thermal contact conductance can also be represented by CONTAC48.

14.48.2 Contact Kinematics

Contact kinematics is concerned with the precise tracking of contact nodes and surfaces in order to define clear and unambiguous contact conditions. The primary aim is to delineate between open (i.e., not in contact) and closed (in contact) contact situation. This task is accomplished by various algorithms embedded in CONTAC48.

Contact and target definition – With reference to the introductory figure, two potential contact surfaces are referred to as either the “target surface” or the “contact surfaces”. The “target surface” is represented by “target nodes” I and J, and the “contact surface” is represented by the “contact node” K. It is usually the case that many CONTAC48 elements will be needed to fully represent a realistic contact problem. (To that end the **GCGEN** command of the PREP7 routine can be used to generate CONTAC48 elements.)

Pinball algorithm – In simple terms, contact occurs whenever the contact node penetrates the target surface. The first step in the determination of contact penetration is to make a distinction between near–field and far–field contact. Figure 14.48–1 shows several positions of a contact node with respect to a circle centered on the target surface (nodes I and J). This circle is referred to as a “pinball”. When a contact node is outside the pinball an “open” contact condition is assumed, irrespective of whether or not the contact node K is above or below the target. Penetration can only occur if the contact node is inside the pinball. The radius of the pinball defaults to be 50% greater than the distance between the target nodes and can be overridden by real constant PINB.

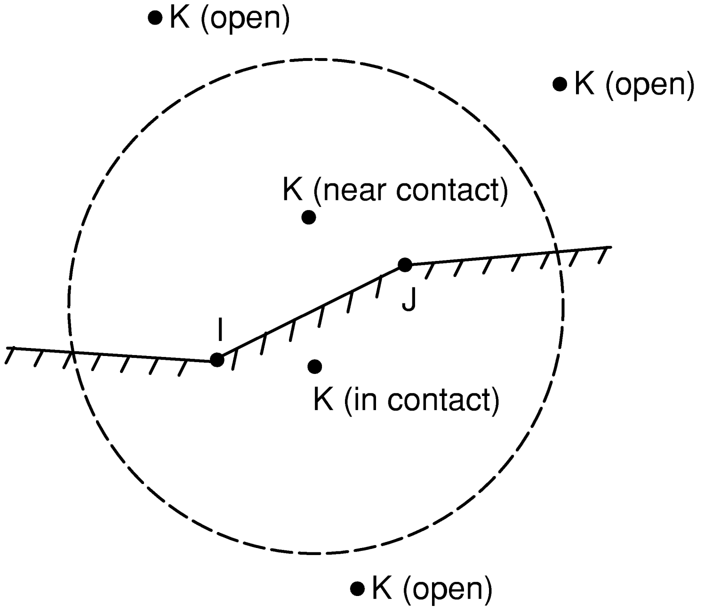


Figure 14.48–1 Definition of Near–Field and Far–Field Contact

Pseudo element algorithm – The next step in the determination of contact is to associate a single target to each contact node. Figure 14.48–2 shows the situation that occurs when one contact node is near (i.e., inside the pinball) several element targets. In particular it is seen that contact penetration has occurred and involves either the element defined by nodes I’, J’, and K or the element defined by nodes I”, J”, and K. If a clear distinction is not made it is possible that contact “voids” or “overlaps” can appear (see Figure 14.48–3). These voids and overlaps are unavoidable and are due, in the main, to piecewise discretization of surfaces that are actually curved. To remove this potential difficulty, solid “pseudo elements” are formed that surround each target. These solid elements are temporarily formed each equilibrium iteration and provide a continuous mapping for each contact node that is in or nearly in contact with a target. The kinematic information that is needed to build these pseudo elements is stored in a global contact data base that is updated each equilibrium iteration.

As can be seen in Figure 14.48–4, the pseudo element mapping indicates that node K is uniquely assigned to the target of the element defined by nodes I’, J’, and K and not to the element defined by I”, J”, and K.

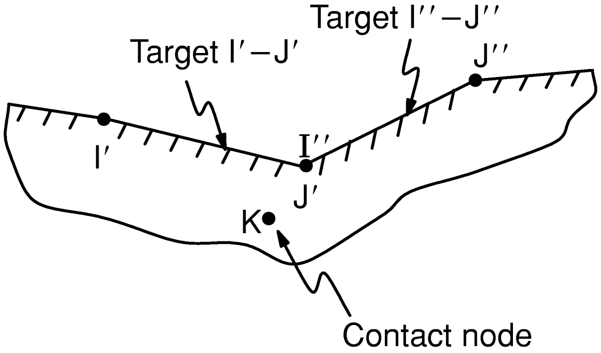


Figure 14.48–2 Contact Node with Two Potential Targets for Contact

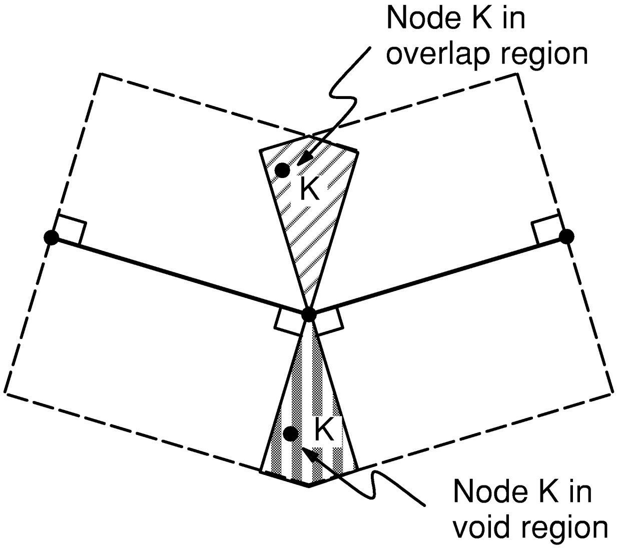


Figure 14.48–3 Potential Voids and Overlaps at Contact Intersection

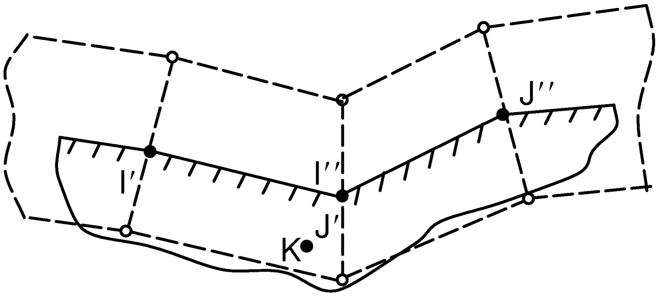


Figure 14.48–4 Pseudo Element

Contact gap and projection – The pinball and pseudo–element algorithms provide a one–to–one mapping between a contact node and a target. The final kinematic step is to calculate the amount of the open gap or the gap penetration of the contact node on the target plane, along with the point of projection of the contact node. In Figure 14.48–5 an element coordinate system is indicated. It is useful to first define the unit normal and unit tangent vector to the target plane.

$$\{n\} = \{\nu\} \text{ x } \{s\} \qquad (14.48\text{–}1)$$

$$\{s\} = \left(\{X_J\} - \{X_I\}\right) / L \qquad (14.48\text{–}2)$$

where: $\{\nu\}$ = unit normal vector to the global x–y plane

$\{X_I\}$ = position vector of node I

$\{X_J\}$ = position vector of node J

L = length of target segment = $||\{X_J\} - \{X_I\}||$

The gap (g) and projection point (s*), defined in local s–n coordinates, are

$$g = \left(\{X_K\} - \{X_I\}\right)^T \{n\} \qquad (14.48\text{–}3)$$

$$s^* = -1 + 2\left[\left(\{X_K\} - \{X_I\}\right)^T \{s\}\right] / L \qquad (14.48\text{–}4)$$

where: $\{X_K\}$ = position vector of node K

Contact penetration is assumed to occur if the value of g is found to be negative. It is important to mention that the target surface is internally expanded if input quantity TOLS on **R** command is specified, thereby increasing the chances that contact will occur. A positive gap value indicates an open contact condition. These various conditions of contact are referred to in CONTAC48's output as "status". These are

STAT (or OLDST) = 4 (open and outside the pinball)

STAT (or OLDST) = 3 (open and inside the pinball)

STAT (or OLDST) < 3 (contact has occurred) (STAT=1 or 2 are described below)

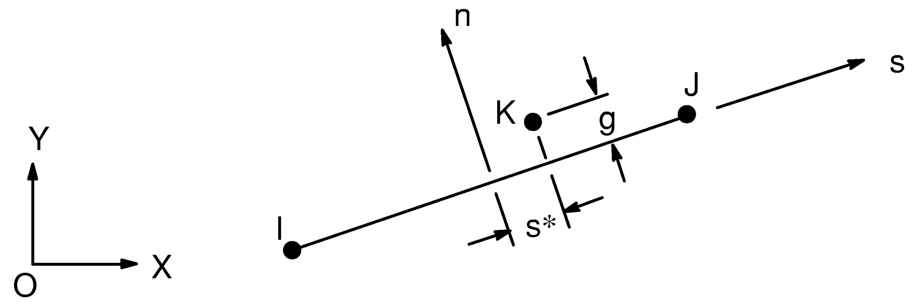


Figure 14.48–5 Location of Contact Node in Local Contact Coordinates

14.48.3 Contact Forces

As explained above contact is indicated when the contact node K penetrates the target surface defined by target nodes I and J. This penetration is represented by the magnitude of the gap (g) and is a violation of compatibility. In order to satisfy contact compatibility, forces are developed in a direction normal (n–direction) to the target that will tend to reduce the penetration to an acceptable numerical level. In addition to compatibility forces, friction forces are developed in a direction that is tangent (s–direction) to the target plane.

Normal forces – Two methods of satisfying contact compatibility are available for CONTAC48: a penalty method (KEYOPT(2)=0) or a combined penalty plus Lagrange multiplier method (KEYOPT(2)=1). The penalty method approximately enforces compatibility by means of a contact stiffness (i.e., the penalty parameter). The combined approach satisfies compatibility to a user–defined precision by the generation of additional contact forces that are referred to as Lagrange forces.

For the penalty method,

$$f_n = \begin{cases} K_n g & \text{if } g \le 0 \\ 0 & \text{if } g > 0 \end{cases} \tag{14.48–5}$$

where: K_n = contact stiffness (input quantity KN on **R** command)

For the combined method, the Lagrange multipler component of force is computed locally (for each element) and iteratively. It is expressed as

$$f_n = \min\,(0,\ K_n g + \lambda_{i+1}) \tag{14.48–6}$$

where: λ_{i+1} = Lagrange multiplier force at iteration i+1

$$= \begin{cases} \lambda_i + \alpha K_n g & \text{if } |\, g \,| \geq \varepsilon \\ \lambda_i & \text{if } |\, g \,| < \varepsilon \end{cases} \qquad (14.48\text{–}7)$$

ε = user–defined compatibility tolerance (input quantity TOLN on **R** command)

α = an internally computed factor ($\alpha < 1$)

Tangential forces – Tangential forces are due to friction that arises as the contact node meets and moves along the target. CONTAC48 allows three friction models: frictionless (KEYOPT(3)=0), elastic Coulomb friction (KEYOPT(3)=1), and rigid Coulomb friction (KEYOPT(3)=2). The Coulomb friction representation requires the specification of the coefficient of sliding friction (μ), which is supplied by means of material property MU on the **MP** command.

For the frictionless case, the tangential force is merely

$$f_s = 0 \qquad (14.48\text{–}8)$$

For elastic Coulomb friction it is necessary to calculate the tangential displacement of the contact node relative to the target. The total tangential motion is

$$u_s = \frac{1}{2}\,(s^* - s_o^*)\,L \qquad (14.48\text{–}9)$$

where s_o^* is the contact position at the previous converged solution or time point.

The deformation is decomposed into elastic (or sticking) and sliding (inelastic) components:

$$u_s = u_s^e + u_s^s \qquad (14.48\text{–}10)$$

where: u_s^e = elastic tangential deformation

u_s^s = sliding (inelastic) tangential deformation

The tangential force is

$$f_s = \begin{cases} K_t\, u_s^e < F\bar{f}_s & \text{if sticking (STAT = 1)} \\ \bar{f}_s & \text{if sliding (STAT = 2)} \end{cases} \qquad (14.48\text{–}11)$$

where: K_t = sticking stiffness (input quantity KT on **R** command)

$\bar{f}_s$ = sticking force limit of the Coulomb friction model

F = static/dynamic friction factor (input quantity FACT on **R** command)

The limiting sticking force is

$$\bar{f}_s = -\mu f_n \qquad (14.48–12)$$

where: μ = coefficient of sliding friction

= $\tan\phi$ (see Figure 14.48–6)

Equations (14.48–5) to (14.48–12) are merely a summary of contact forces and displacements. The actual computation that is performed uses a technique that is similar to that of non–associative theory of plasticity (see Section 4.1). In each substep that sliding friction occurs, an elastic predictor is computed in contact traction space (Figure 14.48–6). The predictor is modified with a radial return mapping function, providing both a small elastic deformation along with sliding response as developed by Giannakopoulos (135).

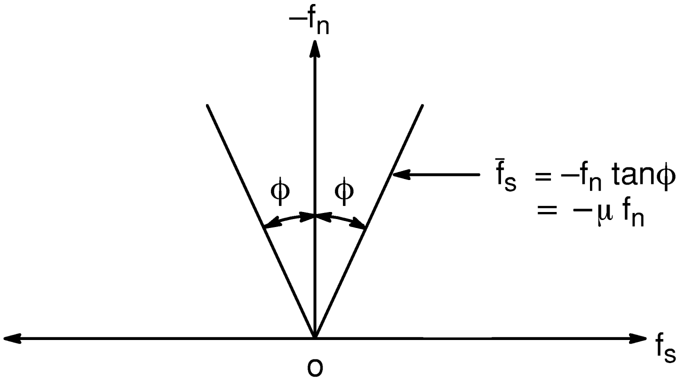


Figure 14.48–6 Contact Friction Space for Coulomb Friction

For the elastic Coulomb model, initial contact is always treated as elastic sticking (STAT=1), but with the tangential force set to zero ($f_s = 0$). In other words, the goal of initial contact is the determination of the penetration (g) and the contact point (s^*), irrespective of friction forces. All subsequent substeps will allow friction to develop according to equation (14.48–11).

Elastic contact tangential deformations are ignored in the rigid Coulomb friction model. The contact node K (if penetrated) is always assumed to be sliding on the target (STAT=2), where the tangential force is

$$f_s = \frac{u_s}{|u_s|}\bar{f}_s \qquad (14.48–13)$$

Contact force transition – A special situation arises when a contact node moves from one target to another. When this occurs, the contact history is passed from the target that was in contact to the target that is currently subjected to contact. In so doing, the path–dependence of friction is maintained and, for some problems, convergence behavior is seen to improve. The transition makes use of a contact data base that contain contact conditions and forces for all contact nodes in actual contact.

14.48.4 Stiffness Matrix and Load Vector

It is convenient to define two interpolation vectors in terms of the local s–direction coordinate. These interpolation vectors are evaluated at the point of projection (s = s*) of the contact node K to the target plane (see Figure 14.48–5), where s* is dimensionless and ranges from –1 to +1.

$$\{N_n\} = \lfloor 0 \quad -\frac{1}{2}(1 - s^*) \quad 0 \quad -\frac{1}{2}(1 + s^*) \quad 0 \quad 1 \rfloor^T \qquad (14.48\text{–}14$$

$$\{N_s\} = \lfloor -\frac{1}{2}(1 - s^*) \quad 0 \quad -\frac{1}{2}(1 + s^*) \quad 0 \quad 1 \quad 0 \rfloor^T \qquad (14.48\text{–}15)$$

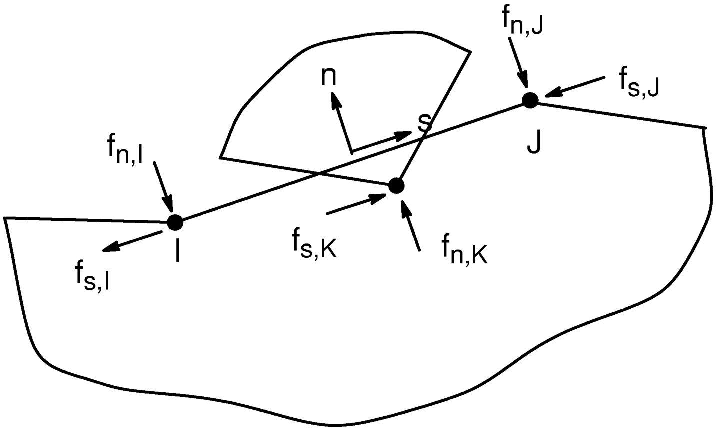


Figure 14.48–7 Nodal Contact Forces

Figure 14.48–7 shows all nodal contact forces. In the normal direction, the force applied to the contact node is defined in equation (14.48–6) and is balanced by opposite force applied to the target nodes; that is,

$$f_{n,K} = f_{n,I} + f_{n,J} = f_n \qquad (14.48\text{–}16)$$

Similarly, in the tangential direction,

$$f_{s,K} = f_{s,I} + f_{s,J} = f_s \qquad (14.48\text{–}17)$$

where f_s is defined by either equation (14.48–8), (14.48–11), and (14.48–13).

Using the interpolation vectors above, the element load vector (i.e., the Newton–Raphson restoring forces) is

$$\{f_\ell^{ne}\} = f_n\ \{N_n\} + f_s\ \{N_s\} \tag{14.48–18}$$

It has been shown (Wriggers et al.(137) and Parisch(132)) that a tangent stiffness matrix for contact is formed by the outer product of the interpolation vectors. In general form,

$$[K_\ell] = \begin{cases} K_n\{N_n\}\{N_n\}^T + K_s\{N_s\}\{N_s\}^T & \text{if sticking contact (STAT=1)} \\ K_n\{N_n\}\{N_n\}^T & \text{if sliding or frictionless contact (STAT=2)} \\ [0] & \text{if open contact (STAT=3 or 4)} \end{cases} \tag{14.48–19}$$

Certain terms are modified and added to equation (14.48–19) that are not given in full detail here. These additional terms are those related to adaptive decent as well as certain consistent tangent components.

14.48.5 Thermal/Structural Contact

Combined structural and thermal contact is specified if KEYOPT(1) = 1, which indicates that UX, UY, and TEMP degrees of freedom (DOFs) are active. When contact is established, heat is transferred across the interface in a direction normal to the target surface. The total heat flow from the target surface to the contact node is given as

$$q_n = \begin{cases} K_c\ (T^* - T_K) & \text{if in contact (STAT} \le 2) \\ 0 & \text{if open (STAT} > 2) \end{cases} \tag{14.48–20}$$

where:

K_c = contact conductance (input as COND on **R** command)

T^* = temperature of the target plane at the contact point

= $\frac{1}{2}\ (1 - s^*)\ T_I + \frac{1}{2}\ (1 + s^*)\ T_J$

T_I, T_J = current nodal temperatures

T_K = temperature of contact node K

The thermal conductivity matrix is:

$$\left[K_\ell^c\right] = \begin{cases} K_c\ \{N_n'\}\ \{N_n'\}^T & \text{if in contact (STAT} \le 2) \\ [0] & \text{if open (STAT} > 2) \end{cases} \tag{14.48–21}$$

The element thermal load vector is comprised of the Newton–Raphson restoring heat flows, and can be expressed as:

$$\{F_\ell^{nr}\} = q_n \{N_n'\} \tag{14.48–22}$$

where:

$$\{N_n'\} = \lfloor -\frac{1}{2}(1 - s^*) \quad -\frac{1}{2}(1 + s^*) \quad 1 \rfloor^T$$

The thermal load vector and conductivity matrix are assembled with the structural load vector and stiffness matrix, respectively, in a manner consistent with the defined DOFs.

14.48.6 Description of Element Output Quantities

Several of the variables discussed above appear as output quantities for the CONTAC48 element. These and all other output quantities are summarized below.

STAT = current status of element. If STAT = 1, sticking contact; if 2, sliding contact; if 3, open but inside pinball (i.e., close); and if 4, open and outside pinball (i.e., far away)

OLDST = status at the previous time step

NX, NY = global components of the target surface normal = n (equation (14.48–1))

FNTOT = total normal force = f_n (equation (14.48–5) or (14.48–6))

FNPF = Penalty function part of normal force = $K_n g$ ($g < 0$)

GAP = gap size = g (equation (14.48–3))

LEN = target length = L

LOC = normalized location of contact node on target plane = s^* (equation (14.48–4))

FS = tangential force = f_s (equations (14.48–8), (14.48–11), and (14.48–13))

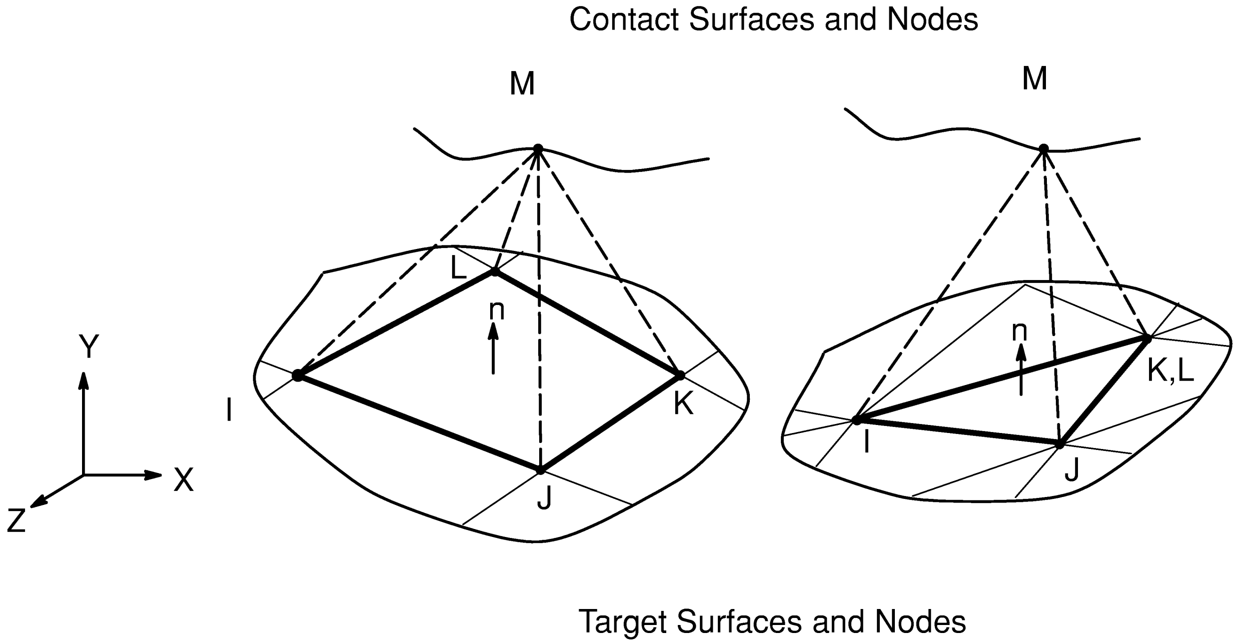
FSLIM = Coulomb limit force = $\bar{f}_s$ (equation (14.48–12))

MU = active friction coefficient = $F\mu$

ANGLE = principal angle of friction force. ANGLE = 0 if $f_s \geq 0$; ANGLE = 180° if $f_s < 0$

Q = heat flow at contact (equation (14.48–20))

14.49 CONTAC49 — 3–D Point–to–Surface Contact



14.49.1 Introduction

CONTAC49 is a 5–node element that is intended for general contact analysis. In a general contact analysis, the area of contact between two (or more) bodies is generally not known in advance. In addition the finite element models of the contacting bodies are generated is such a way that precise node–to–node contact is neither achievable nor desirable when contact is established. This type of contact situation precludes the use of node–to–node contact elements such as CONTAC52. The CONTAC49 element has the capability to represent general contact of models that are generated with arbitrary meshes. In other words, its use is not limited to known contact or node–to–node configurations.

CONTAC49 is applicable to 3–D geometries. It may be applied to contact of solid bodies or shells, to static or dynamic analyses, to problems with or without friction, and to flexible–to–flexible or rigid–to–flexible body contact. The combined mechanisms of structural contact and thermal contact conductance can also be modeled by CONTAC49.

14.49.2 Contact Kinematics

Contact kinematics is concerned with the precise tracking of contact nodes and surfaces in order to define clear and unambiguous contact conditions. The primary aim is to delineate between open (i.e., not in contact) and closed (in contact) contact situation. This task is accomplished by various algorithms embedded in the CONTAC49 element.

Contact and target definition – With reference to the introductory figure, two potential contact surfaces are referred to as either the “target surface” or the “contact surface”. The “target surface” is represented by “target nodes” I, J, K and L, and the “contact surface” is represented by the “contact node” M. It is usually the case that many CONTAC49 elements will be needed to fully represent a realistic contact problem. (To that end the **GCGEN** command of the PREP7 routine can be used to generate CONTAC49 elements.)

Pinball algorithm – In simple terms, contact occurs whenever the contact node (M) penetrates the target surface (I, J, K, L). The first step in the determination of contact penetration is to make a distinction between near–field and far–field contact. Referring to the 2–D case for simplicity (CONTAC48), Figure 14.48–1 shows several positions of a contact node with respect to the target surface. For CONTAC49 in 3–D the delimiting circle becomes a sphere which is referred to as the “pinball”. When a contact node is outside the pinball an “open” contact condition is assumed, irrespective of whether or not the contact node is above or below the target. Penetration can only occur once the contact node is inside the pinball. The radius of the pinball defaults to be 50% greater than the maximum of the two target surface diagonals and can be overridden by real constant PINB.

Pseudo element algorithm – The next step in the determination of contact is to associate a single target to each contact node depending upon the position of the contact node in space. This is accomplished by establishing solid “pseudo elements” for each target surface as shown in Figure 14.49–1. A unique association is formed whenever contact node M is found within a target’s pseudo element. If a clear distinction is not made it is possible that contact “voids” or “overlaps” can appear (see Figure 14.48–3 in the 2–D case). These voids and overlaps are unavoidable and are due, in the main, to piecewise discretization of surfaces that are actually curved. These solid elements are temporarily formed each equilibrium iteration and provide a continuous mapping for each contact node that is in or nearly in contact with a target. The kinematic information that is needed to build these pseudo elements is stored in a global contact data base that is updated each equilibrium iteration.

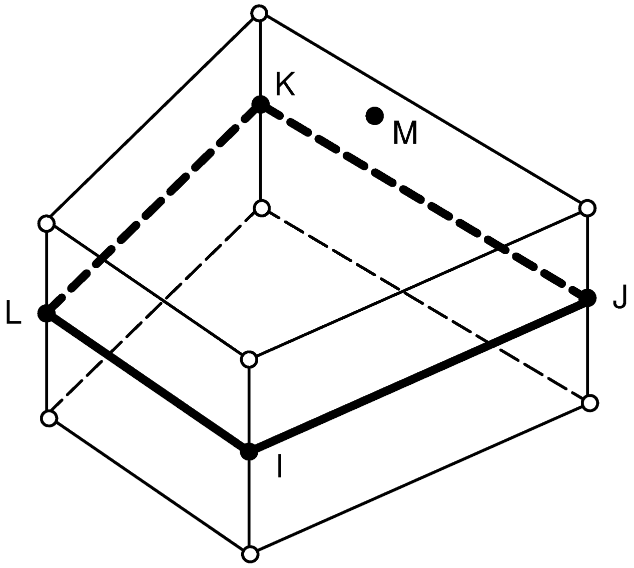


Figure 14.49–1 Pseudo Element

Contact gap and projection – The pinball and pseudo–element algorithms provide a one–to–one mapping between a contact node and a target. The final kinematic step is to determine the open gap or the gap penetration of the contact node on the target plane, along with the point of projection of the contact node. This is achieved by first modifying the target surface nodes to lie in a plane if they do not already, simplifying tangential surface calculations. In other words the warping of the target surface is ignored. In Figure 14.49–2 several coordinate systems are indicated. The global system is the usual X–Y–Z system. The next system is the natural s–t–n system of the planar target surface. A rectangular, invariant, x–y–z system is constructed from the natural s–t–n system in such a way that n– and z–directions are parallel. This enables straightforward tracking of the tangential contact motions. Finally a second rectangular x_e, y_e, z_e system is defined for element force output. Having defined the modified (unwarped) target surface and the various coordinate systems, the contact kinematics of gap and location are left to be defined.

With reference to Figure 14.49–2, the contact location (s*, t*) is computed by an iterative Newton's method based upon a normal projection of the contact node to the target plane. At the projected contact point a value of gap (g) is determined by the contact node's location with respect to the target plane. Contact penetration is assumed to occur if the value of g is found to be negative, and the s* and t* projections are found to be in the natural space bounds of the target. For the later condition, the target surface is internally expanded if input quantity TOLS on **R** command is specified, thereby increasing the chances that a contact node will come into contact with the target plane. A positive gap value indicates an open contact condition.

These various conditions of contact are referred to in CONTAC49's output as "status". They are:

STAT (or OLDST) = 4	open and outside the pinball
STAT (or OLDST) = 3	open and inside the pinball
STAT (or OLDST) < 3	contact has occurred (STAT=1 or 2 are described below)

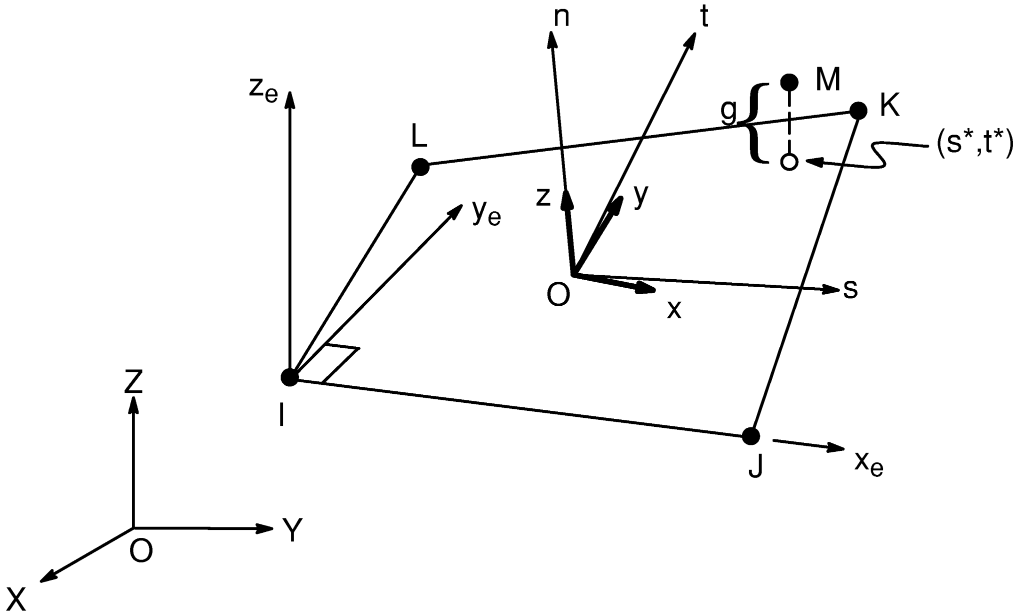


Figure 14.49–2 Target Coordinate Systems

14.49.3 Contact Forces

As explained above, contact is indicated when the contact node M penetrates the target surface defined by target nodes I, J, K, and L. This penetration is represented by the magnitude of the gap (g) and is a violation of compatibility. In order to satisfy contact compatibility, forces are developed in a direction normal (n–direction) to the target that will tend to reduce the penetration to an acceptable numerical level. In addition to compatibility forces, friction forces are developed in directions that are tangent to the target plane. The normal and tangential friction forces that are described here are referenced to the local x–y–z system shown in Figure 14.49–2.

Normal forces – Two methods of satisfying contact compatibility are available for CONTAC49: a penalty method (KEYOPT(2)=0) or a combined penalty plus Lagrange multiplier method (KEYOPT(2)=1). The penalty method approximately enforces compatibility by means of a contact stiffness (i.e., the penalty parameter). The combined approach satisfies compatibility to a user–defined precision by the generation of additional contact forces (i.e., Lagrange forces).

For the penalty method,

$$f_n = \begin{cases} K_n g & \text{if } g \le 0 \\ 0 & \text{if } g > 0 \end{cases} \tag{14.49–1}$$

where: K_n = contact stiffness (input quantity KN on **R** command)

For the combined method, the Lagrange multipler component of force is computed locally (for each element) and iteratively. It is expressed as

$$f_n = \min\ (0,\ K_n g + \lambda_{i+1}) \tag{14.49–2}$$

where: λ_{i+1} = Lagrange multiplier force at iteration i+1

$$= \begin{cases} \lambda_i + \alpha K_n g & \text{if } |\,g\,| \ \ge \varepsilon \\ \lambda_i & \text{if } |\,g\,| \ < \varepsilon \end{cases} \tag{14.49–3}$$

ε = user–defined compatibility tolerance (input quantity TOLN on **R** command)

α = an internally computed factor ($\alpha < 1$)

Tangential forces – Tangential forces are due to friction that arises as the contact node meets and moves along the target. The CONTAC49 element considers three friction models: frictionless (KEYOPT(3)=0), elastic Coulomb friction (KEYOPT(3)=1), and rigid Coulomb friction (KEYOPT(3)=2). The Coulomb friction representations requires the specification of the coefficient of sliding friction (μ), which is supplied by means of material property MU on the **MP** command.

For the frictionless case (KEYOPT(3)=0) the tangential force is merely:

$$f_x = f_y = 0 \tag{14.49–4}$$

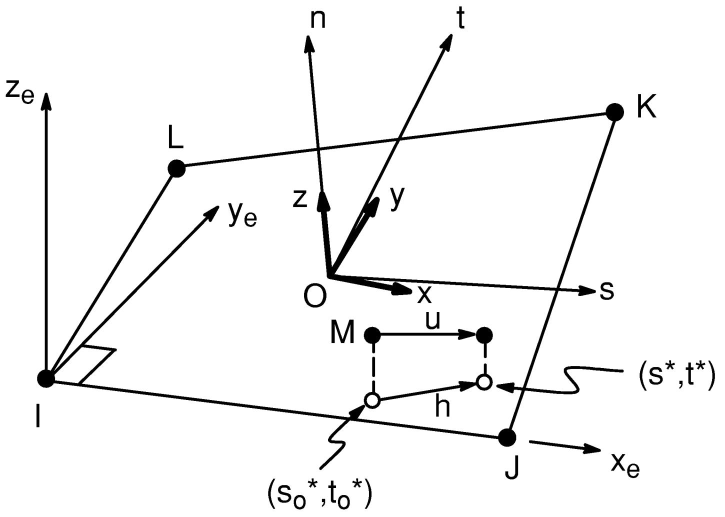


Figure 14.49–3 Location of Contact Node on the Target Plane

For elastic Coulomb friction (KEYOPT(3)=1) it is necessary to calculate the tangential deformations of the contact node relative to the target. Figure 14.49–3 shows the total motion (u) of contact node M along the target plane. It is seen that the total tangential displacement (η) is represented by the projection of the total contact node motion to the unwarped plane of the target. Two projection points are mapped in natural coordinates (s, t). The point (s^*, t^*) is the current projection position, and the tangential deformation is tracked from the point (s_o^*, t_o^*) that is associated with the previous converged solution (i.e., the previous time point). The deformation is first separated into x and y components, such that

$$\eta = \left(\eta_x^2 + \eta_y^2\right)^{1/2} \tag{14.49–5}$$

where:

η_x = component of η in the local x direction

η_y = component of η in the local y direction

Next, the deformation is decomposed into elastic (or sticking) and sliding (or inelastic) components.

$$\begin{aligned} \eta_x &= \eta_x^e + \eta_x^s \\ \eta_y &= \eta_y^e + \eta_y^s \end{aligned} \tag{14.49–6}$$

Related tangential forces are:

$$\begin{aligned} f_x &= K_t\ \eta_x^e \\ f_y &= K_t\ \eta_y^e \end{aligned} \tag{14.49–7}$$

where:

K_t = sticking stiffness (input quantity KT on **R** command)

It follows that the magnitude of the tangential forces is

$$f_s = \left(f_x^2 + f_y^2\right)^{1/2} \tag{14.49–8}$$

The delineation between sticking and sliding conditions is expressed by:

$$f_s = \bar{f}_s, \qquad \text{if sliding (STAT = 2)} \tag{14.49–9}$$

$$f_s < F\bar{f}_s, \qquad \text{if sticking (STAT = 1)} \tag{14.49–10}$$

where:

F = static/dynamic friction factor (input quantity FACT on **R** command)

The limiting sliding force in the Coulomb model is

$$\bar{f}_s = -\mu\ f_n \tag{14.49–11}$$

Equations (14.49–1) thru (14.49–11) are merely a summary of contact forces and displacements. The actual computation that is performed uses a technique that is similar to that of non–associative theory of plasticity (see Section 4.1). In each substep that sliding friction occurs, an elastic predictor is computed in contact traction space (Figure 14.49–4). The predictor is modified via a radial return mapping function, providing both a small elastic deformation along with sliding response as developed by Giannakopoulos(135).

For the elastic Coulomb model, initial contact is always treated as elastic sticking (STAT=1), but with the tangential force set to zero (f_s=0). In other words, the goal of initial contact is intentionally limited to the determination of the penetration (g) and the contact point (s^*, t^*), irrespective of friction forces. All subsequent substeps will allow friction to develop according to Equations (14.49–9) and (14.49–10).

Turning attention to the rigid Coulomb model of friction (KEYOPT(3)=2), elastic contact deformations is ignored. If the contact node M is penetrating the target, it is always assumed to be sliding (STAT=2). Tangential forces are

$$\begin{aligned} f_x &= \frac{\eta_x}{\eta}\ \bar{f}_s \\ f_y &= \frac{\eta_y}{\eta}\ \bar{f}_s \end{aligned} \tag{14.49–12}$$

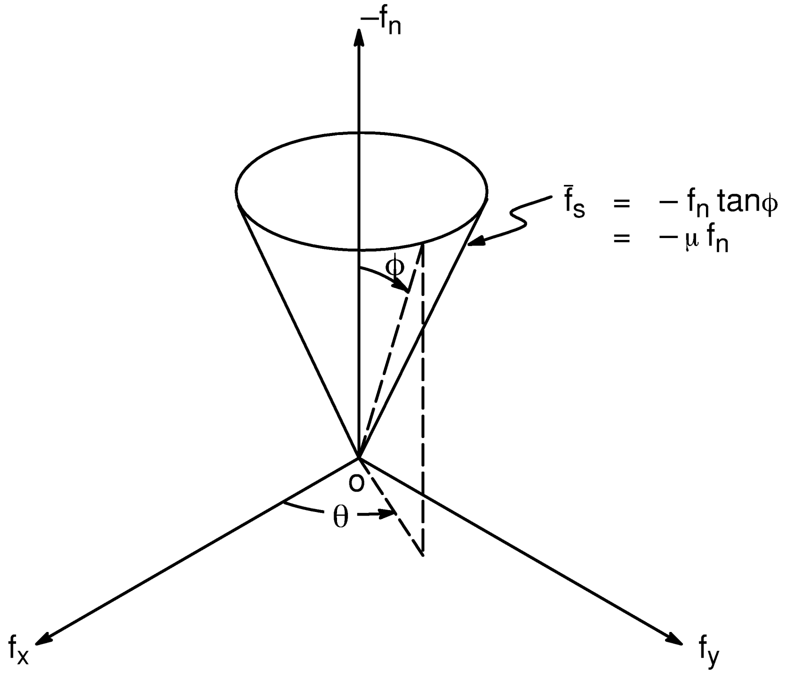


Figure 14.49–4 Contact Friction Space for Coulomb Friction

Contact force transition – A special situation arises when a contact node moves from one target to another. When this occurs, the contact history is passed from the target that was in contact to the target that is currently subjected to contact. In so doing, the path–dependence of friction is maintained and, for some problems, convergence behavior is seen to improve. The transition makes use of a contact database that contain contact conditions and forces for all contact nodes in actual contact.

14.49.4 Stiffness Matrix And Load Vector

It is convenient to define three interpolation vectors in terms of the local s–t coordinates. These interpolation vectors are evaluated at the point of projection (s=s*, and t=t*) of the contact node M to the target plane (see Figure 14.49–2). (Note that s and t are dimensionless coordinates, ranging from –1 to 1 for quadrilateral targets and 0 to 1 for triangular targets.)

$$\{N_n\}^T = \lfloor 0 \quad 0 \quad q_1 \quad 0 \quad 0 \quad q_2 \quad 0 \quad 0 \quad q_3 \quad 0 \quad 0 \quad q_4 \quad 0 \quad 0 \quad 1 \rfloor \tag{14.49–13}$$

$$\{N_x\}^T = \lfloor q_1 \quad 0 \quad 0 \quad q_2 \quad 0 \quad 0 \quad q_3 \quad 0 \quad 0 \quad q_4 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \rfloor \tag{14.49–14}$$

$$\{N_y\}^T = \lfloor 0 \quad q_1 \quad 0 \quad 0 \quad q_2 \quad 0 \quad 0 \quad q_3 \quad 0 \quad 0 \quad q_4 \quad 0 \quad 0 \quad 1 \quad 0 \rfloor \tag{14.49–15}$$

For the 4–node target, individual interpolates are

$$\begin{aligned}
q_1 &= -\frac{1}{4}(1 - s^*)(1 - t^*) \\
q_2 &= -\frac{1}{4}(1 + s^*)(1 - t^*) \\
q_3 &= -\frac{1}{4}(1 + s^*)(1 + t^*) \\
q_4 &= -\frac{1}{4}(1 - s^*)(1 + t^*)
\end{aligned} \tag{14.49–16}$$

And for the 3–node target

$$\begin{aligned}
q_1 &= -s^* \\
q_2 &= -t^* \\
q_3 &= -1 + s^* + t^* \\
q_4 &= 0
\end{aligned} \tag{14.49–17}$$

Figure 14.48–7 shows all nodal contact forces for the CONTAC48 element, the 3–node 2–D contact element. A similar representation exists for both the 5–node and 4–node CONTAC49 element. In the normal direction, the force applied to the contact node (M) is defined in equations (14.49–1) and (14.49–3) and is balanced by opposite forces applied to the target nodes; that is,

$$f_{n,M} = f_{n,I} + f_{n,J} + f_{n,K} + f_{n,L} = f_n \tag{14.49–18}$$

Similarly, in the tangential directions,

$$\begin{aligned} f_{x,M} &= f_{x,I} + f_{x,J} + f_{x,K} + f_{x,L} = f_x \\ f_{y,M} &= f_{y,I} + f_{y,J} + f_{y,K} + f_{y,L} = f_y \end{aligned} \tag{14.49–19}$$

where f_x and f_y are defined in the previous subsection.

Using the interpolation vector above, the element load vector (i.e., the Newton–Raphson restoring forces) is:

$$\left\{F_{\ell}^{ne}\right\} = f_n \{N_n\} + f_x \{N_x\} + f_y \{N_y\} \tag{14.49–20}$$

It has been shown (Wriggers et al.(137), Stein et al.(138)) that a tangent stiffness matrix for contact is formed by the outer product of the interpolation vectors. In general form,

$$[K_{\ell}] = \begin{cases} K_n \{N_n\}\{N_n\}^T + K_s (\{N_x\}\{N_x\}^T + \{N_y\}\{N_y\}^T), & \text{if sticking contact (STAT=1)} \\ K_n \{N_n\} \{N_n\}^T & \text{if sliding or frictionless contact (STAT=2)} \\ [0] & \text{if open contact (STAT=3 or 4)} \end{cases} \tag{14.49–21}$$

Certain terms are modified and added to Equation (14.49–21) that are not given in full detail here. These additional terms are those related to adaptive decent as well as certain internal element manipulations.

14.49.5 Thermal/Structural Contact

Combined structural and thermal contact is specified if KEYOPT(1) = 1, which indicates that UX, UY, and TEMP degrees of freedom (DOFs) are active. When contact is established, heat is transferred across the interface in a direction normal to the target surface. The total heat flow from the target surface to the contact node is given as

$$q_n = \begin{cases} K_c \, (T^* - T_M) & \text{if in contact (STAT} \le 2) \\ 0 & \text{if open (STAT} > 2) \end{cases} \tag{14.49–22}$$

where: K_c = contact conductance (input quantity COND on **R** command)

T^* = temperature of the target plane at the contact point

= $- q_1 T_I - q_2 T_J - q_3 T_K - q_4 T_L$

T_I, T_J, T_K, T_L = current temperatures of the target nodes

T_M = temperature of contact node M

The thermal conductivity matrix is:

$$\left[K_\ell^c\right] = \begin{cases} K_c \, \{N_n'\} \, \{N_n'\}^T & \text{if in contact (STAT} \le 2) \\ [0] & \text{if open (STAT} > 2) \end{cases} \tag{14.49–23}$$

The element thermal load vector is comprised of the Newton–Raphson restoring heat flows, and can be expressed as:

$$\{F_\ell^{nr}\} = q_n \, \{N_n'\} \tag{14.49–24}$$

where: $\left\{N_n'\right\} = \left\lfloor q_1 \quad q_2 \quad q_3 \quad q_4 \quad 1 \right\rfloor^T$

The thermal load vector and conductivity matrix are assembled with the structural load vector and stiffness matrix, respectively, in a manner consistent with the defined DOFs.

14.49.6 Description Of Element Output Quantities

Several of the variables discussed above appear as output quantities for the CONTAC49 element. These and all other output quantities are summarized below.

STAT = current status of element. If STAT = 1, sticking contact; if 2, sliding contact; if 3, open but inside pinball (i.e., close); and if 4, open and outside pinball (i.e., far away)

OLDST = status at the previous time step

NX,NY,NZ = global x–y–z components of the target surface normal

FNTOT = total normal force = f_n (Equation (14.49–1) or (14.49–2))

FNPF = Penalty function part of normal force = $K_n g$ ($g < 0$) (Equation (14.49–1))

GAP = gap size = g

AREA = target area

LOC1,LOC2 = normalized locations of contact node on target plane = s^* and t^*

FS1,FS2 = tangential forces that are rotated into element (x_e, y_e, z_e). FS1 is in the x_e direction, and FS2 in y_e (see Figure 14.49–2)

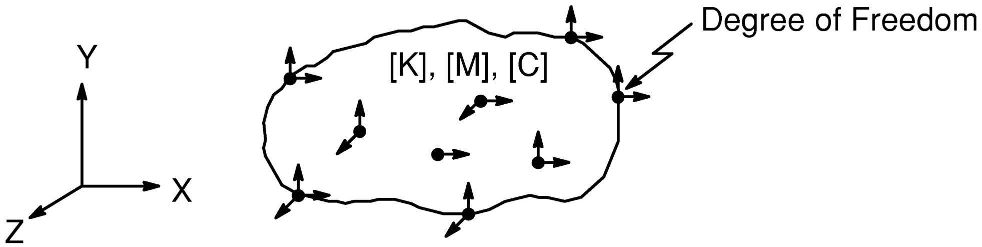
FSLIM = Coulomb limit force = $\bar{f}_s$ (Equation (14.49–11))

MU = active friction coefficient = $F\mu$

ANGLE = $\theta = \tan^{-1} (f_y / f_x)$ principal angle of friction force (see Figure 14.49–4)

Q = heat flow at contact (equation (14.49–22))

14.50 MATRIX50 — Superelement (or Substructure)



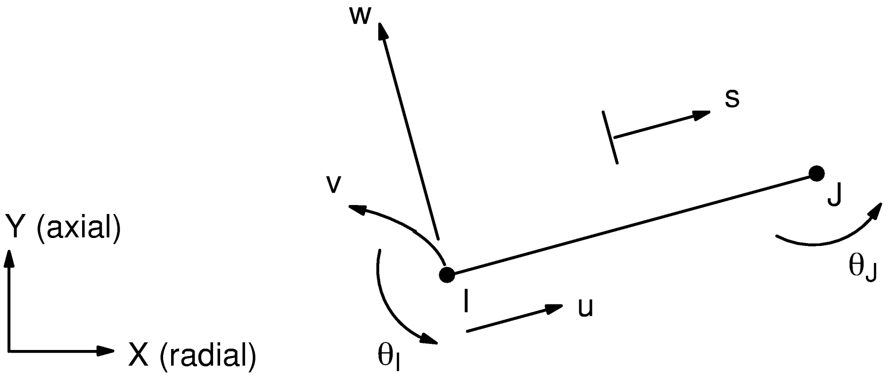
Matrix or Vector	Shape Functions	Integration Points
Stiffness/ Conductivity Matrix	Same as the constituent elements	Same as the constituent elements
Mass/Specific Heat Matrix	Same as the constituent elements reduced down to the master degrees of freedom	Same as the constituent elements
Damping Matrix	Same as mass/specific heat matrix	Same as the constituent elements
Stress Stiffness/ Convection Surface Matrix (used only when added to the Stiffness Matrix)	Same as constituent elements	Same as the constituent elements
Gravity, Thermal and Pressure/Heat Generation and Convection Surface Load Vectors	Same as constituent elements	Same as the constituent elements

Load Type	Distribution
Element Temperature and Heat Generation Rate	As input during generation run
Pressure/Convection Surface Distribution	As input during generation run

14.50.1 Other Applicable Sections

Superelements are discussed in Section 17.6.

14.51 SHELL51 — Axisymmetric Structural Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.3.2–1), (12.3.2–2), and (12.3.2–3). If extra shape functions are not included (KEYOPT(3) = 1): equations (12.3.1–1), (12.3.1–2), and (12.3.1–3).	3 along length
Mass Matrix	Same as stiffness matrix without extra shapes. Equations (12.3.1–1), (12.3.1–2), and (12.3.1–3).	Same as stiffness matrix
Stress Stiffness Matrix	Equations (12.3.1–1), (12.3.1–2), and (12.3.1–3).	Same as stiffness matrix
Thermal and Pressure Load Vector	Same as stiffness matrix	Same as stiffness matrix
Newton–Raphson Load Vector	Same as stiffness matrix	3 along length, 9 thru thickness

Load Type	Distribution
Element Temperature	Linear thru thickness and along length, constant around circumference

Load Type	Distribution
Nodal Temperature	Constant thru thickness, linear along length, constant around circumference
Pressure	Linear along length, constant around circumference

14.51.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. Section 14.61 contains information also applicable to SHELL51.

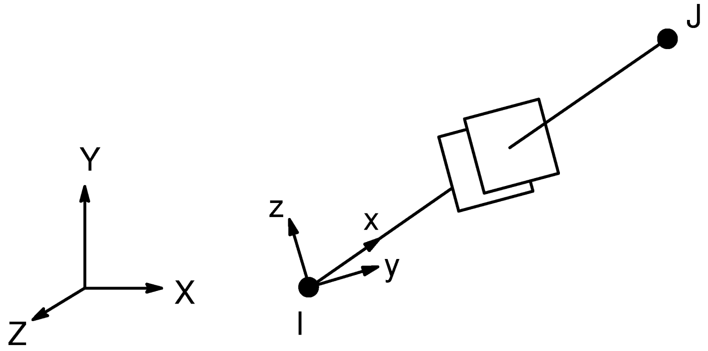
14.51.2 Integration Point Locations for Nonlinear Material Effects

The locations and weighting factors for the nine point integration rule through the element thickness for the nonlinear material effects is given in Table 13.1–5. Nonlinear material values are only computed at the midpoint between the nodes. When these values are needed for other integration points along the length, they are simply transferred from the midpoint. This is to avoid “sawtoothing effects”.

14.51.3 Large Deflections

Unlike other line elements, SHELL51 uses the rotational strain approach (Kohnke(20)).

14.52 CONTAC52 — 3–D Point–to–Point Contact



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Normal Direction	None	None
	Sliding Direction	None	None

Load Type	Distribution
Element Temperature	None – average used for material property evaluation
Nodal Temperature	None – average used for material property evaluation

14.52.1 Other Applicable Sections

Section 14.12 has many aspects also valid for CONTAC52, including normal and sliding force determinations, rigid Coulomb friction (KEYOPT(1) = 1), and the force–deflection relationship shown in Figure 14.12–1.

14.52.2 Element Matrices

CONTAC52 may have one of three conditions: closed and stuck, closed and sliding, or open.

If the element is closed and stuck, the element stiffness matrix (in element coordinates) is:

$$[K_\ell] = \begin{bmatrix} k_n & 0 & 0 & -k_n & 0 & 0 \\ 0 & k_s & 0 & 0 & -k_s & 0 \\ 0 & 0 & k_s & 0 & 0 & -k_s \\ -k_n & 0 & 0 & k_n & 0 & 0 \\ 0 & -k_s & 0 & 0 & k_s & 0 \\ 0 & 0 & -k_s & 0 & 0 & k_s \end{bmatrix} \qquad (14.52\text{–}1)$$

where: k_n = normal stiffness (input quantity KN on **R** command)

k_s = sticking stiffness (input quantity KS on **R** command)

The Newton–Raphson load vector is:

$$\{F_\ell^{nr}\} = \begin{Bmatrix} F_n \\ F_{sy} \\ F_{sz} \\ -F_n \\ -F_{sy} \\ -F_{sz} \end{Bmatrix} \qquad (14.52\text{–}2)$$

where: F_n = normal force across gap (from previous iteration)

F_s = sticking force across gap (from previous iteration)

If the element is closed and sliding in both directions, the element stiffness matrix (in element coordinates) is:

$$[K_\ell] = \begin{bmatrix} k_n & 0 & 0 & -k_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -k_n & 0 & 0 & k_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (14.52\text{–}3)$$

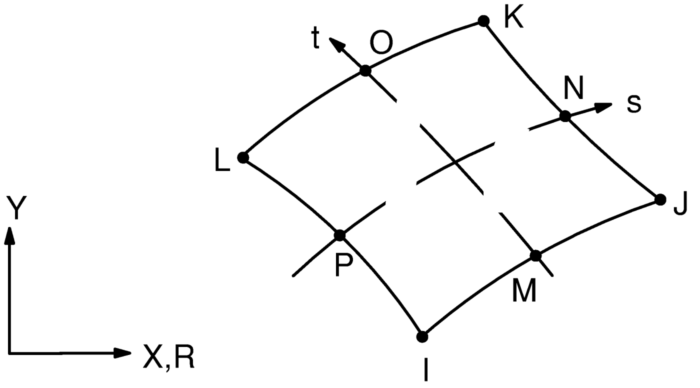
and the Newton–Raphson load vector is the same as in equation (14.52–2). For details on the unsymmetric option (KEYOPT(11) = 1), see Section 14.12.2

If the element is open, there is no stiffness matrix or load vector.

14.52.3 Orientation of Element

For both small and large deformation analysis, the orientation of the element is unchanged. The element is oriented so that the normal force is in line with the original position of the two nodes.

14.53 PLANE53 — 2–D 8–Node Magnetic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Quad	Equation (12.6.7–9)	2 x 2
	Triangle	Equation (12.6.2–9)	3
Damping (Eddy Current) Matrix	Quad	Equations (12.6.7–9) and (12.6.7–21)	Same as coefficient matrix
	Triangle	Equations (12.6.2–9) and (12.6.2–21)	Same as coefficient matrix
Permanent Magnet and Applied Current Load Vector	Same as coefficient matrix		Same as coefficient matrix

Load Type	Distribution
Current Density, Voltage Load and Phase Angle Distribution	Bilinear across element

References: Silvester et al.(72), Weiss et al.(94), Garg et al.(95)

14.53.1 Other Applicable Sections

Section 5.2 has a complete derivation of the matrices and load vectors of a general magnetic analysis element. Section 11.0 contains a discussion of coupled field analyses. Section 13.1 describes integration point locations.

14.53.2 Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the solution varies linearly along that edge.

14.53.3 The VOLT DOF in 2D and Axisymmetric Skin Effect Analysis

KEYOPT(1) = 1 can be used to model skin effect problems. The corresponding DOFs are AZ and VOLT. Here, AZ represents the z– or θ–component of the magnetic vector potential for 2D or axisymmetric geometry, respectively. VOLT has different meanings for 2D and axisymmetric geometry. The difference is explained below for a transient case.

A skin effect analysis is used to find the eddy current distribution in a massive conductor when a source current is applied to it. In a general 3D case, the (total) current density {J} is given by

$$\{J\} = -\sigma\frac{\partial\{A\}}{\partial t} - \sigma\frac{\partial\{\nabla\nu\}}{\partial t} \qquad (14.53\text{–}1)$$

where: ν = (time–integrated) electric scalar potential

Refer to Section 5.3.2 for definitions of other variables. For a 2D massive conductor, the z–component of {J} may be rewritten as:

$$J_z = -\sigma\frac{\partial A_z}{\partial t} + \sigma\frac{\partial\{\nabla\tilde{V}\}}{\partial t} \qquad (14.53\text{–}2)$$

where $\Delta\tilde{V}$ may be termed as the (time–integrated) source voltage drop per unit length and is defined by:

$$\Delta\tilde{V} = -\hat{z}\cdot\nabla\nu \qquad (14.53\text{–}3)$$

For an axisymmetric massive conductor, the θ–component of {J} may be rewritten as

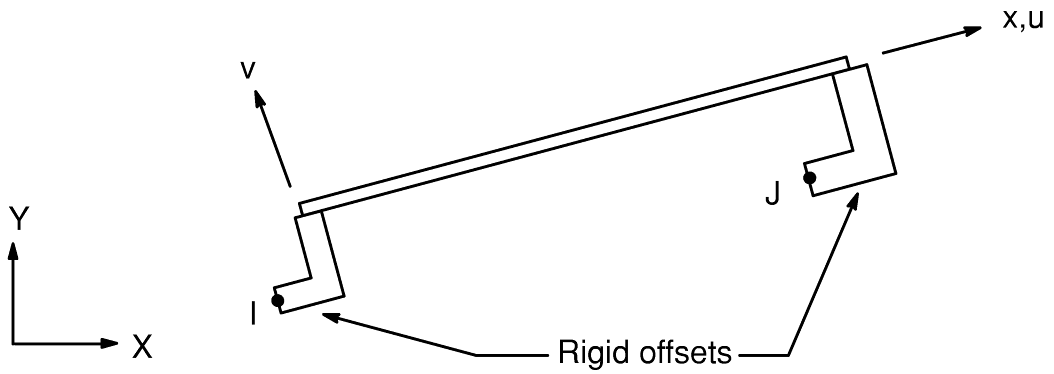
$$J_\theta = -\sigma\frac{\partial A_\theta}{\partial t} + \frac{\sigma}{2\pi r}\frac{\partial\{\nabla\tilde{V}\}}{\partial t} \qquad (14.53\text{–}4)$$

where the (time–integrated) source voltage drop in a full 2π radius is defined by

$$\Delta\tilde{V} = -2\pi r\hat{\theta} \cdot \nabla\nu \qquad (14.53–5)$$

When KEYOPT(1)=1, the VOLT DOF represents the definition given by equation (14.53–3) and (14.53–5) for a 2D and axisymmetric conductor, respectively. Also, all VOLT DOFs in a massive conductor region must be coupled together so that $\Delta\tilde{V}$ has a single value.

14.54 BEAM54 — 2–D Elastic Tapered Unsymmetric Beam



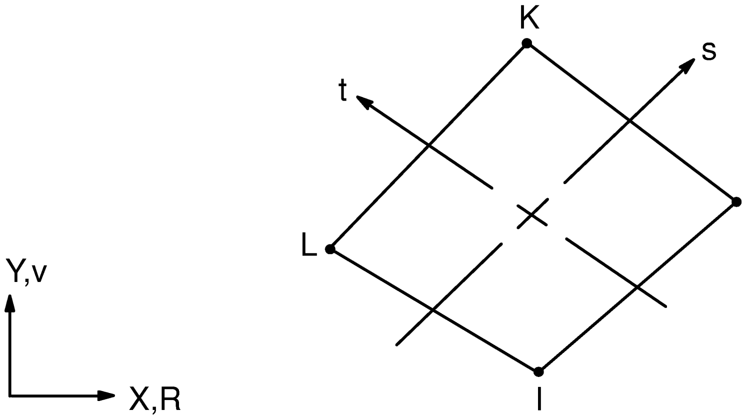
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.1.2–1) and (12.1.2–2)	None
Foundation Stiffness Matrix	Same as stiffness matrix	None
Mass Matrix	Same as stiffness matrix	None
Stress Stiffness Matrix	Equation (12.1.2–2)	None
Thermal Load Vector	Same as stiffness matrix	None
Pressure Load Vector	Equation (12.1.2–2)	None

Load Type	Distribution
Element Temperature	Linear thru thickness, linear along length
Nodal Temperature	Constant thru thickness, linear along length
Pressure	Linear along length

14.54.1 Derivation of Matrices

All matrices and load vectors are derived in the same way as for BEAM44, the 3–D Elastic Tapered Unsymmetrical Beam Element, except that they are reduced to 2–D. Further, the same assumptions and restrictions apply.

14.55 PLANE55 — 2–D Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Quad	Equation (12.6.5–20)	2 x 2
	Triangle	Equation (12.6.1–20)	1 if planar 3 if axisymmetric
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Section 13.2.		Same as conductivity matrix
Heat Generation Load Vector	Same as conductivity matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix evaluated at the face		2

14.55.1 Other Applicable Sections

Chapter 6 describes the derivation of the element matrices and load vectors as well as heat flux evaluations. Section 14.70 describes fluid flow in a porous medium, accessed in PLANE55 with KEYOPT(9) = 1. Section 13.1 describes integration point locations.

14.55.2 Mass Transport Option

If KEYOPT(8) > 0, the mass transport option is included as described in Section 6.1 with equation (6.1–1) and by K_e^{tm} of equation (6.2–7). The solution accuracy is dependent on the element size. The accuracy is measured in terms of the non–dimensional criteria called the element Peclet number (Gresho(58)):

$$P_e = \frac{VL\rho C_p}{2K} \qquad (14.55\text{–}1)$$

where: V = magnitude of the velocity vector

L = element length dimension along the velocity vector direction

ϱ = density of the fluid (input as DENS on **MP** command)

C_p = specific heat of the fluid (input as C on **MP** command)

K = equivalent thermal conductivity along the velocity vector direction

The terms V, L, and K are explained more thoroughly below:

$$V = \left(V_x{}^2 + V_y{}^2\right)^{1/2} \qquad (14.55\text{–}2)$$

where: V_x = fluid velocity (mass transport) in x direction (input quantity VX on **R** command)

V_y = fluid velocity (mass transport) in y direction (input quantity VY on **R** command)

Length L is calculated by finding the intersection points of the velocity vector which passes through the element origin and intersects at the element boundaries.

For orthotropic materials, the equivalent thermal conductivity K is given by:

$$K = K_x K_y \left[\frac{\left(1 + m^2\right)}{K^2{}_y + m^2 K_x{}^2}\right]^{1/2} \qquad (14.55\text{–}3)$$

where:: K_x, K_y = thermal conductivities in the x and y directions (input quantities KXX and KYY on **MP** command)

m = slope of the velocity vector in the element coordinate system

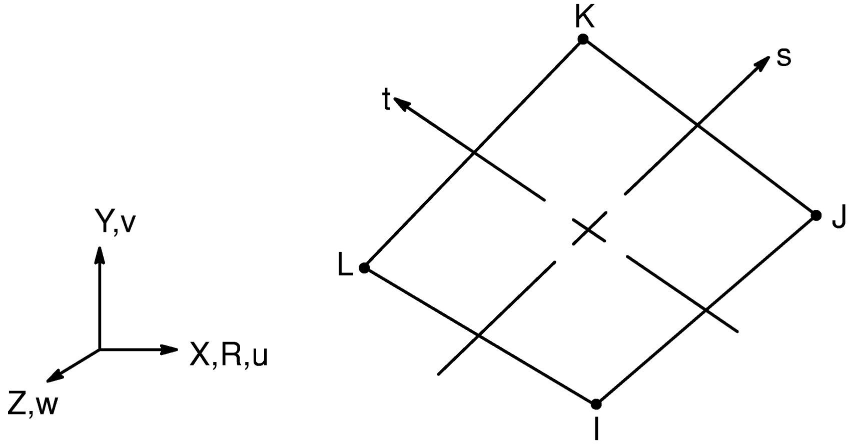
= $\frac{V_y}{V_x}$ (if KEYOPT(4) = 0)

For the solution to be physically valid, the following condition has to be satisfied (Gresho(58)):

$$P_e < 1 \qquad (14.55\text{–}4)$$

This check is carried out during the element formulation and an error message is printed out if equation (14.55–4) is not satisfied. When this error occurs, the problem should be re–run after reducing the element size in the direction of the velocity vector.

14.56 HYPER56 — 2–D 4–Node Mixed U–P Hyperelastic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.6.5–1), (12.6.5–2), and (12.6.5–3)	2 x 2
	Triangle	Equations (12.6.1–1), (12.6.1–2), and (12.6.1–3)	3 if axisymmetric and 1 if plane
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as mass matrix, specialized to face		2

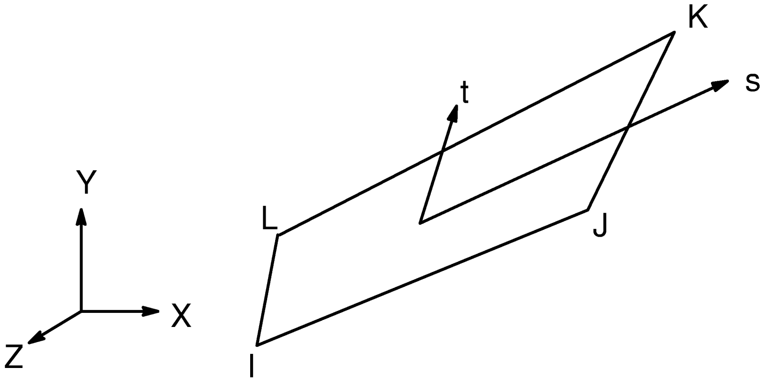
Load Type	Distribution
Element Temperature	Bilinear across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

References: Oden(123), Sussman(124)

14.56.1 Other Applicable Sections

For the basic formulation refer to Section 14.58. The hyperelastic material model (Mooney–Rivlin) is described in Section 4.5. Section 13.1 describes integration point locations.

14.57 SHELL57 — Thermal Shell

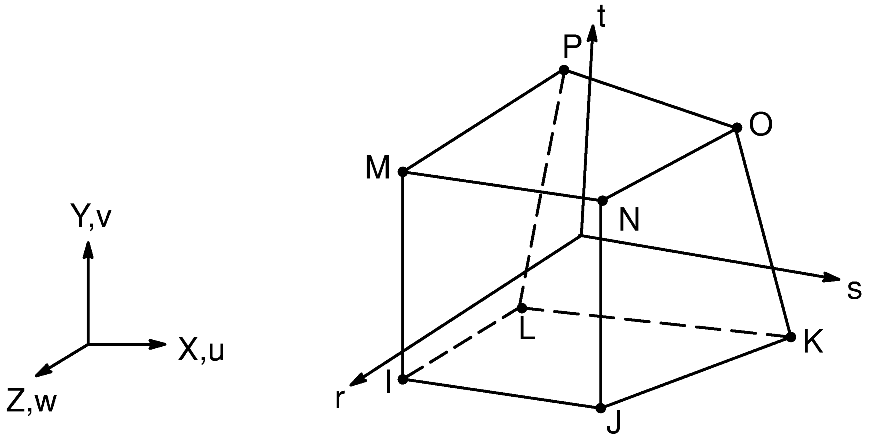


Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Quad	Equation (12.5.8–20)	2 x 2
	Triangle	Equation (12.5.1–20)	3
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Section 13.2		Same as conductivity matrix
Heat Generation Load Vector	Same as conductivity matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix		Same as conductivity matrix

14.57.1 Other Applicable Sections

Chapter 6 describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations.

14.58 HYPER58 — 3–D 8–Node Mixed U–P Hyperelastic Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2) and (12.8.18–3)	2 x 2 x 2
Mass Matrix	Same as stiffness matrix	2 x 2 x 2
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2 x 2

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Oden(123), Sussman(124)

14.58.1 Other Applicable Sections

The hyperelastic material model (Mooney–Rivlin) is described in Section 4.5. Section 13.1 describes integration point locations.

14.58.2 Mixed Hyperelastic Element Derivation

A mixed formulation is used that utilizes a modified strain energy density containing hydrostatic pressure as an explicit solution variable. Since it uses separate interpolations for the displacements and the hydrostatic pressure, it is referred to as the u–p (displacement–pressure) formulation. The essentials of the u–p formulation are summarized below. For details see references Oden and Kikuchi(123), Sussman and Bathe(124), and Zienkiewicz et al.(125).

14.58.3 Modified Strain Energy Density

The u–p formulation starts with a modified potential that explicitly includes the pressure variables:

$$W + Q = W - \frac{1}{2K}\,(p - \overline{p})^{\,2} \tag{14.58–1}$$

where: Q = energy augmentation due to volume constraint condition
K = bulk modulus
p = pressure obtainable from W alone
$\overline{p}$ = separately interpolated pressure (output stress item HPRES)

The original potential, W, for a Mooney–Rivlin material, which would be applicable for slightly incompressible rubber–like materials, is given by equation (4.5–8). Note that the last term of equation (4.5–8) provides the pressure p.

The displacements are discretized using standard isoparametric interpolations, whereas the pressure $\overline{p}$ is discretized by a polynomial expansion of the following form without any association with any nodes.

$$\overline{p} = p_1 + p_2 s + p_3 t + p_4 st + - - - \tag{14.58–2}$$

where: s, t = element coordinates in natural space

14.58.4 Finite Element Matrices

The finite element matrices in terms of the incremental displacements and pressures are given by:

$$\begin{bmatrix} K^{uu} & K^{up} \\ K^{pu} & K^{pp} \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{p} \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \end{Bmatrix} - \begin{Bmatrix} R^{u} \\ R^{p} \end{Bmatrix} \tag{14.58–3}$$

where: $\{F\}$ = external nodal forces

$\{\dot{u}\}, \ \{\dot{p}\}$ = displacement and pressure increments respectively

$\{R^u\}$ and $\{R^p\}$ are the Newton–Raphson restoring force vectors (elsewhere referred to as $\{F^{nr}\}$):

$$R^{u}_{i} = \frac{\partial}{\partial \dot{u}_i} \left[\int_{vol} \left(W - \frac{1}{2K} (p - \overline{p})^2 \right) d(vol) \right] = \int_{vol} \overline{S}_{kl} \frac{\partial E_{kl}}{\partial \dot{u}_i} \ d(vol) \tag{14.58–4}$$

$$R^{p}_{i} = \frac{\partial}{\partial \dot{p}_i} \left[\int_{vol} \left(W - \frac{1}{2K} (p - \overline{p})^2 \right) d(vol) \right] = \int_{vol} \frac{1}{K} (p - \overline{p}) \ \frac{\partial \overline{p}}{\partial \dot{p}_i} d(vol) \tag{14.58–5}$$

$$\begin{aligned} K^{uu}_{ij} &= \frac{\partial R^{u}_{i}}{\partial u_j} = \text{ displacement–only stiffness} \\ &= \int_{vol} C^{uu}_{klrs} \frac{\partial E_{kl}}{\partial \dot{u}_i} \ \frac{\partial E_{rs}}{\partial \dot{u}_j} d(vol) + \int_{vol} \overline{S}_{kl} \ \frac{\partial^2 E_{kl}}{\partial \dot{u}_i \partial \dot{u}_j} \ d(vol) \end{aligned} \tag{14.58–6}$$

$$\begin{aligned} K^{up}_{ij} &= \frac{\partial R^{u}_{i}}{\partial \dot{p}_j} = \frac{\partial R^{p}_{j}}{\partial u_i} = K^{pu}_{ji} = \text{ displacement–pressure coupled stiffness} \\ &= \int_{vol} \frac{1}{K} \ \frac{\partial p}{\partial E_{kl}} \ \frac{\partial E_{kl}}{\partial \dot{u}_i} \ \frac{\partial \overline{p}}{\partial \dot{p}_j} \ d(vol) \end{aligned} \tag{14.58–7}$$

$$\begin{aligned} K^{pp}_{ij} &= \frac{\partial R^{u}_{i}}{\partial \dot{p}_j} = \text{ pressure–only stiffness} \\ &= \int_{vol} \frac{\partial \overline{p}}{\partial \dot{p}_i} \left(- \frac{1}{K} \right) \frac{\partial \overline{p}}{\partial \dot{p}_j} \ d(vol) \end{aligned} \tag{14.58–8}$$

In the above,

$$\overline{S}_{kl} = S_{kl} - \frac{1}{K}\,(p - \overline{p})\,\frac{\partial p}{\partial E_{kl}} \qquad (14.58–9)$$

$$C^{uu}_{klrs} = \frac{\partial^2 W}{\partial E_{kl}\partial E_{rs}} - \frac{1}{K}\,\frac{\partial p}{\partial E_{kl}}\,\frac{\partial p}{\partial E_{rs}} - \frac{1}{K}\,(p - \overline{p})\,\frac{\partial^2 p}{\partial E_{kl}\partial E_{rs}} \qquad (14.58–10)$$

where: C^{uu}_{klrs} = augmented incremental moduli

The new augmented stress tensor $\overline{S}_{kl}$ has the property that the pressure corresponding to these new stresses $\overline{S}_{kl}$ when added with the pressure computed directly from the displacement configuration equals the separately interpolated pressure.

14.58.5 Incompressibility

The analysis of rubber–like materials poses computational difficulties in that these materials are almost incompressible. The fact that the volume changes very little while the material undergoes large strains often leads to displacement locking. In the u–p hyperelastic elements this difficulty is circumvented by enforcing the incompressibility constraint through a constraint equation. This constraint equation relates the separately interpolated pressure $(\overline{p})$ (output quantity HPRES) to the pressure (p) computed from the displacements and attempts to maintain the volume constraint in an average integrated sense over an element.

To be effective, there should be enough pressure DOFs $\overline{p}_i$, but the number of $\overline{p}_i$ DOFs in a model must be smaller than the number of unconstrained kinematic DOFs u_i (UX, UY, etc.) in order to allow deformation to occur at all. As a guideline, the number of unconstrained kinematic DOFs should be at least twice the number of pressure DOFs for 2–D problems, and at least three times the number of pressure DOFs for axisymmetric or 3–D problems.

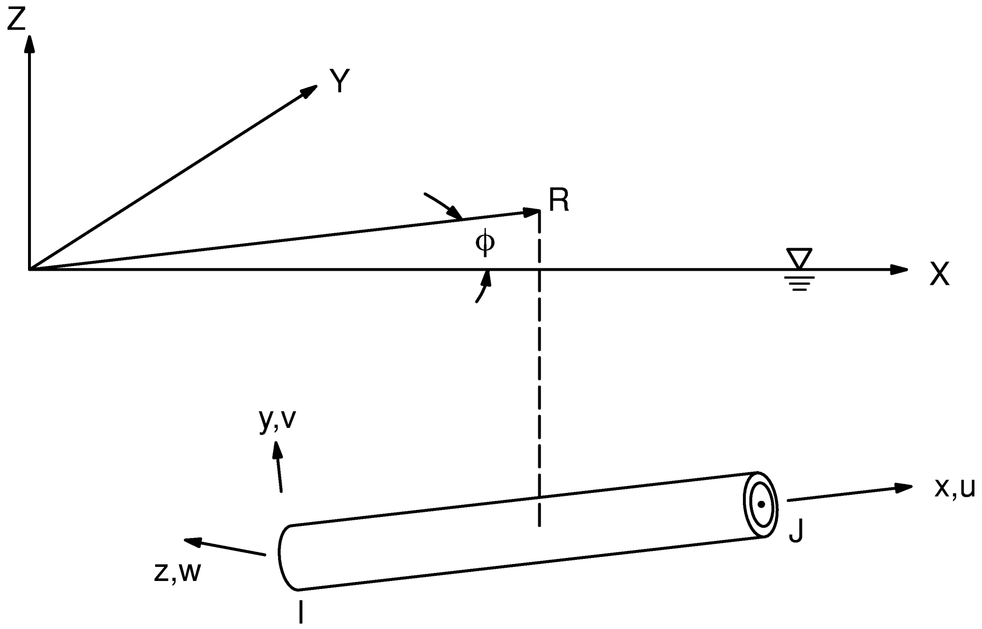
14.58.6 Instabilities in the Material Constitutive Law

Instability may sometimes occur due to real buckling, or it may occur due to the mathematical procedure used in the formulation. For example, the application of a load in a single step that leads to a very large strain, say 100% or more, may cause instability. Furthermore, if there is a complex variation of the hydrostatic pressure, the number of pressure DOFs may not be adequate to describe the behavior. This may lead to a local volume change, associated with a decrease in total energy. In those cases, local mesh refinement or the use of higher order elements is recommended.

14.58.7 Existence of Multiple Solutions

For nonlinear problems, more than one stable solution may exist for a given set of boundary conditions. The case of a hollow hemisphere with zero prescribed loads is an example of such multiple solutions. Here the two equilibrium solutions are: the undeformed stress–free state and the inverted self–equilibrating state. Stable equilibrium solutions do not pose any difficulty; however, if the equilibrium becomes unstable at some point (e.g. incipient buckling) during the analysis, the solution procedure might collapse.

14.59 PIPE59 — Immersed Pipe or Cable



Matrix or Vector	Options	Shape Functions	Integration Points
Stiffness Matrix	Pipe Option (KEYOPT(1) ≠ 1)	Equations (12.2.2–1), (12.2.2–2), (12.2.2–3), and (12.2.2–4)	None
	Cable Option (KEYOPT(1) = 1)	Equation (12.2.1–1), (12.2.1–2), and (12.2.1–3)	None
Stress Stiffness Matrix	Pipe Option (KEYOPT(1) ≠ 1)	Equation (12.2.2–2) and (12.2.2–3)	None
	Cable Option (KEYOPT(1) = 1)	Equation (12.2.1–2) and (12.2.1–3)	None

Matrix or Vector	Options	Shape Functions	Integration Points
Mass Matrix	Pipe Option (KEYOPT(1)≠1) with consistent mass matrix (KEYOPT(2)=0)	Equation (12.2.2–1), (12.2.2–2), and (12.2.2–3)	None
	Cable Option (KEYOPT(1)=1) or reduced mass matrix (KEYOPT(2)=1)	Equation (12.2.1–1), (12.2.1–2), and (12.2.1–3)	None
Thermal, Pressure, and Hydrostatic Load Vector	Same as stiffness matrix		None
Hydrodynamic Load Vector	Same as stiffness matrix		2

Load Type	Distribution
Element Temperature*	Linear thru thickness or across diameter, and along length
Nodal Temperature*	Constant across cross–section, linear along length
Pressure	Linearly varying (in Z direction) internal and external pressure caused by hydrostatic effects. Exponentially varying external overpressure (in Z direction) caused by hydrodynamic effects

* Immersed elements with no internal diameter assume the temperatures of the water.

14.59.1 Overview of the Element

PIPE59 is similar to PIPE16 (or LINK8 if the cable option (KEYOPT(1)=1) is selected). The principal differences are that the mass matrix includes the:

1. Outside mass of the fluid (“added mass”) (acts only normal to the axis of the element),
2. Internal structural components (pipe option only),

and the load vector includes:

a. Hydrostatic effects

b. Hydrodynamic effects

14.59.2 Location of the Element

The origin for any problem containing PIPE59 must be at the free surface (mean sea level). Further, the Z axis is always the vertical axis, pointing away from the center of the earth.

The element may be located in the fluid, above the fluid, or in both regimes simultaneously. There is a tolerance of only $\frac{D_e}{8}$ below the mud line, for which

$$D_e = D_o + 2\ t_i \tag{14.59–1}$$

where:

t_i = thickness of external insulation (input quantity TKIN on **RMORE** command)

D_o = outside diameter of pipe/cable (input quantity DO on **R** command)

The mud line is located at distance d below the origin, where d is the input quantity DEPTH on the **TBDATA** commands with **TB**,WATER (water motion table). This condition is checked with:

$$Z(N) > -\left(d + \frac{D_e}{8}\right) \qquad \leftarrow \text{no error message} \tag{14.59–2}$$

$$Z(N) \leq -\left(d + \frac{D_e}{8}\right) \qquad \leftarrow \text{fatal error message} \tag{14.59–3}$$

where Z(N) is the vertical location of node N. If it is desired to generate a structure below the mud line, the user can set up a second material property for those elements using a greater d and deleting hydrodynamic effects. Alternatively, the user can use a second element type such as PIPE16, the elastic straight pipe element.

If the problem is a large deflection problem, greater tolerances apply for second and subsequent iterations:

$$Z(N) > -\left(d + 10\ D_e\right) \qquad \leftarrow \text{no error message} \tag{14.59–4}$$

$$-\left(d + 10\ D_e\right) \geq Z(N) > -\ (2d) \qquad \leftarrow \text{warning message} \tag{14.59–5}$$

$$-(2d) \geq Z(N) \qquad \leftarrow \text{fatal error message} \qquad (14.59–6)$$

where Z(N) is the present vertical location of node N. In other words, the element is allowed to sink into the mud for 10 diameters before generating a warning message. If a node sinks into the mud a distance equal to the water depth, the run is terminated. If the element is supposed to lie on the ocean floor, gap elements must be provided.

14.59.3 Stiffness Matrix

The element stiffness matrix for the pipe option (KEYOPT(1)$\neq$ 1) is the same as for BEAM4 (equation (14.4–1)), except that $K_\ell(4,1) = K_\ell(1,4) = K_\ell(10,7) = K_\ell(7,10) = T_T$ and $K_\ell(7,4) = K_\ell(4,7) = K_\ell(10,1) = K_\ell(1,10) = -T_T$.

where:

$$T_T = \begin{cases} 0 & \text{if KEYOPT(1) = 0 (standard option for torque balanced cable or pipe)} \\ \dfrac{G_T\left(D_o^3 - D_i^3\right)}{L} & \text{if KEYOPT(1) =2 (twist–tension option for non–torque balanced cable or pipe)} \end{cases}$$

G_T = twist–tension stiffness constant, which is a function of the helical winding of the armoring (input quantity GXZ on **MP** command, may be negative)

D_i = inside diameter of pipe = $D_o - 2\ t_w$

t_w = wall thickness (input quantity TWALL on **R** command)

L = element length

$A = \frac{\pi}{4}\left(D_o^2 - D_i^2\right)$ = cross–sectional area

$I = \frac{\pi}{64}\left(D_o^4 - D_i^4\right)$ = moment of inertia

J = 2I

The element stiffness matrix for the cable option (KEYOPT(1) = 1) is the same as for LINK8.

14.59.4 Mass Matrix

The element mass matrix for the pipe option (KEYOPT(1) $\neq$ 1) and KEYOPT(2) = 0) is the same as for BEAM4 (equation (14.4–2)), except that $M_\ell(1,1)$, $M_\ell(7,7)$, $M_\ell(1,7)$, and $M_\ell(7,1)$, as well as M(4,4), M(10,10), M(4,10), and M(10,4), are multiplied by the factor (M_a/M_t).

where: $M_t = (m_w + m_{int} + m_{ins} + m_{add})\ L$

= mass/unit length for motion normal to the axis of the element

M_a = $(m_w + m_{int} + m_{ins})\ L$

= mass/unit length for motion parallel to the axis of the element

m_w = $(1-\varepsilon^{in})\varrho \frac{\pi}{4} \left(D_o^2 - D_i^2\right)$

ϱ = density of the pipe wall (input as DENS on **MP** command)

ε^{in} = initial strain (input quantity ISTR on **RMORE** command)

m_{int} = mass/unit length of the internal fluid and additional hardware (input quantity CENMPL on **RMORE** command)

m_{ins} = $(1-\varepsilon_{in})\ \varrho_i \frac{\pi}{4} \left(D_e^2 - D_o^2\right)$

ϱ_i = density of external insulation (input quantity DENSIN on **RMORE** command)

m_{add} = $(1-\varepsilon_{in})\ C_I \varrho_w \frac{\pi}{4}\ D_e^2$

C_I = coefficient of added mass of the external fluid (input quantity CI on **RMORE** command)

ϱ_w = fluid density (input quantity DENSW on **TBDATA** commands with **TB**,WATER)

The element mass matrix for the cable option (KEYOPT(1) = 1) or the reduced mass matrix option (KEYOPT(2) $\neq$ 0) is the same form as for LINK8 except that $M_\ell(1,1)$, $M_\ell(4,4)$, $M_\ell(1,4)$ and $M_\ell(4,1)$ are also multiplied by the factor (M_a/M_t).

14.59.5 Load Vector

The element load vector consists of two parts:

1. Distributed force per unit length to account for hydrostatic (buoyancy) effects ($\{F/L\}_b$) as well as axial nodal forces due to internal pressure and temperature effects $\{F_x\}$.
2. Distributed force per unit length to account for hydrodynamic effects (current and waves) ($\{F/L\}_d$).

The hydrostatic and hydrodynamic effects work with the original diameter and length, i.e., initial strain and large deflection effects are not considered.

Hydrostatic Effects

Hydrostatic effects may affect the outside and the inside of the pipe. Pressure on the outside crushes the pipe and buoyant forces on the outside tend to raise the pipe to the water surface. Pressure on the inside tends to stabilize the pipe cross–section.

The buoyant force for a totally submerged element acting in the positive z direction is:

$$\{F/L\}_b = C_b \rho_w \frac{\pi}{4} D_e^2\ \{g\} \qquad (14.59–7)$$

where: $\{F/L\}_b$ = vector of loads per unit length due to buoyancy
C_b = coefficient of buoyancy (input quantity CB on **RMORE** command)
$\{g\}$ = acceleration vector

Also, an adjustment for the added mass term is made.

The crushing pressure at a node is:

$$P_o^s = -\rho_w \, gz + P_o^a \qquad (14.59\text{–}8)$$

where: P_o^s = crushing pressure due to hydrostatic effects
g = acceleration due to gravity
z = vertical coordinate of the node
P_o^a = input external pressure (input on **SFE** command)

The internal (bursting) pressure is:

$$P_i = -\rho_o \, g(z - S_{fo}) + P_i^a \qquad (14.59\text{–}9)$$

where: P_i = internal pressure
ϱ_o = internal fluid density (input quantity DENSO on **R** command)
S_{fo} = z coordinate of free surface of fluid (input quantity FSO on **R** command)
P_i^a = input internal pressure (input quantity on **SFE** command)

To ensure that the problem is physically possible as input, a check is made to see if the cross–section collapses under the hydrostatic effects. The cross–section is assumed to be unstable if:

$$P_o^s - P_i > \frac{E}{4\left(1 - \nu^2\right)} \left(\frac{2t_w}{D_o}\right)^3 \qquad (14.59\text{–}10)$$

where: E = Young’s modulus (input as EX on **MP** command)
ν = Poisson’s ratio (input as PRXY or NUXY on **MP** command)

The axial force correction term (F_x) is computed as

$$F_x = AE\epsilon_x \qquad (14.59\text{–}11)$$

where ε_x, the axial strain (see equation 2.1–12) is:

$$\epsilon_x = \alpha \Delta T + \frac{1}{E}\left(\sigma_x - \nu\left(\sigma_h + \sigma_r\right)\right) \qquad (14.59\text{–}12)$$

where: α = coefficient of thermal expansion (input as ALPX on **MP** command)

ΔT = $T_a - T_{REF}$

T_a = average element temperature

T_{REF} = input on **TREF** command

σ_x = axial stress, computed below

σ_h = hoop stress, computed below

σ_r = radial stress, computed below

The axial stress, assuming the ends are closed, is:

$$\sigma_x = \frac{P_i D_i^2 - P_o D_o^2}{D_o^2 - D_i^2} \tag{14.59–13}$$

and using the Lamé stress distribution,

$$\sigma_h = \frac{P_i D_i^2 - P_o D_o^2 + \frac{D_i^2 D_o^2}{D^2}\left(P_i - P_o\right)}{D_o^2 - D_i^2} \tag{14.59–14}$$

$$\sigma_r = \frac{P_i D_i^2 - P_o D_o^2 - \frac{D_i^2 D_o^2}{D^2}\left(P_i - P_o\right)}{D_o^2 - D_i^2} \tag{14.59–15}$$

where: P_o = $P_o^s + P_o^d$

P_o^d = hydrodynamic pressure, described below

D = diameter being studied

P_i and P_o are taken as average values along each element. Combining equations (14.59–12) thru (14.59–15).

$$\epsilon_x = \alpha\Delta T + \frac{1 - 2\nu}{E}\,\frac{P_i D_i^2 - P_o D_o^2}{D_o^2 - D_i^2} \tag{14.59–16}$$

Note that if the cross–section is solid (D_i = 0.), equation (14.59–14) reduces to:

$$\epsilon_x = \alpha\Delta T - \frac{1 - 2\nu}{E} P_o \tag{14.59–17}$$

Hydrodynamic Effects

Hydrodynamic effects may occur because the structure moves in a motionless fluid, the structure is fixed but there is fluid motion, or both the structure and fluid are moving. The fluid motion consists of two parts: current and wave motions. The current is input by giving the current velocity and direction (input quantities W(i) and θ(i)) at up to eight different vertical stations (input quantity Z(i)). The velocity and direction are interpolated linearly between stations. The current is assumed to flow horizontally only. The wave may be input using one of four wave theories in Table 14.59–1 with the input coming from the **TBDATA** commands with **TB**,WATER.

Table 14.59–1 Wave Theory Table

KWAV	Description of Wave Theory
0	Small amplitude wave theory, modified with empirical depth decay function, (Wheeler(35))
1	Small amplitude wave theory, unmodified (Airy wave theory), (Wheeler(35))
2	Stokes fifth order wave theory, (Skjelbreia et al(31))
3	Stream function wave theory, (Dean(59))

The free surface of the wave is defined by

$$\eta_s = \sum_{i=1}^{N_w} \eta_i = \sum_{i=1}^{N_w} \frac{H_i}{2} \cos \beta_i \qquad (14.59–18)$$

where: η_s = total wave height

N_w = number of wave components = $\begin{cases} \text{number of waves} & \text{if } K_w \neq 2 \\ 5 & \text{if } K_w = 2 \end{cases}$

K_w = wave theory key (input quantity KWAV on the **TBDATA** commands with **TB**,WATER)

η_i = wave height of component i

H_i = surface coefficient = $\begin{cases} \text{input quantity A(i)} & \text{if } K_w = 0 \text{ or } 1 \\ \text{derived from other input} & \text{if } K_w = 2 \text{ or } 3 \end{cases}$

$$\beta_i = \begin{cases} 2\pi\left(\dfrac{R}{\lambda_i} + \dfrac{t}{\tau_i} + \dfrac{\psi_i}{360}\right) & \text{if KEYOPT(5) = 0 and } K_w = 0 \text{ or } 1 \\ 2\pi\left(\dfrac{R}{\lambda_i} + \dfrac{t}{\tau_i} + \dfrac{\psi_i}{360}\right)\text{(i)} & \text{if KEYOPT(5) = 0 and } K_w = 2 \text{ or } 3 \\ 0.0 & \text{if KEYOPT(5) = 1} \\ \dfrac{\pi}{2} & \text{if KEYOPT(5) = 2} \\ -\dfrac{\pi}{2} & \text{if KEYOPT(5) = 3} \\ \pi & \text{if KEYOPT(5) = 4} \end{cases}$$

R = radial distance to point on element from origin in the X–Y plane in the direction of the wave

λ_i = wave length = $\begin{cases} \text{input quantity WL(i) if WL(i) >} \\ \text{0.0 and if } K_W = 0 \text{ or } 1; \\ \text{otherwise derived from equation (14.59–19)} \end{cases}$

t = time elapsed (input quantity TIME on **TIME** command) (Note that the default value of TIME is usually not desired. If zero is desired, 10^{-12} can be used).

τ_i = wave period = $\begin{cases} \text{input quantity } \tau\text{(i)} & \text{if } K_W \neq 3 \\ \text{derived from other input if } K_W +3 \end{cases}$

ψ_i = phase shift = input quantity ψ(i)

If WL(i) is not input (set to zero) and $K_W < 2$, λ_i is computed iteratively from:

$$\lambda_i = \lambda_i^d \tanh\left(\frac{2\pi d}{\lambda_i}\right) \qquad (14.59–19)$$

where:

λ_i = output quantity small amplitude wave length

λ_i^d = $\dfrac{g(\tau_i)^2}{2\pi}$ = output quantity deep water wave length

g = acceleration due to gravity (Z direction) (input on **ACEL** command)

d = water depth (input quantity DEPTH on the **TBDATA** commands with **TB**,WATER)

Each component of wave height is checked that it satisfies the “Miche criterion” if $K_W \neq 3$. This is to ensure that the wave is not a breaking wave, which the included wave theories do not cover. A breaking wave is one that spills over its crest, normally in shallow water. A warning message is issued if:

$$H_i > H_b \qquad (14.59\text{–}20)$$

where: $H_b = 0.142\, \lambda_i \tanh\left(\frac{2\pi d}{\lambda_i}\right)$ = height of breaking wave (14.59–21)

When using wave loading, there is an error check to ensure that the input acceleration does not change after the first load step, as this would imply a change in the wave behavior between load steps.

For K_w = 0 or 1, the particle velocities at integration points are computed as a function of depth from:

$$\vec{v}_R = \sum_{i=1}^{N_w} \frac{\cosh\left(k_i \bar{Z} f\right)}{\sinh\left(k_i d\right)} \frac{2\pi}{\tau_i}\, \eta_i + \vec{v}_D \qquad (14.59\text{–}22)$$

$$\vec{v}_Z = \sum_{i=1}^{N_w} \frac{\sinh\left(k_i \bar{Z} f\right)}{\sinh\left(k_i d\right)}\, \dot{\eta}_i \qquad (14.59\text{–}23)$$

where: $\vec{v}_R$ = radial particle velocity

$\vec{v}_Z$ = vertical particle velocity

k_i = $2\pi/\lambda_i$

$\bar{Z}$ = height of integration point above the ocean floor = d+Z

$\dot{\eta}_i$ = time derivative of η_i

$\vec{v}_D$ = drift velocity (input quantity W on **TBDATA** command with **TB**,WATER)

$$f = \begin{cases} \dfrac{d}{d + \eta_s} & \text{if } K_w = 0 \text{ (Wheeler(35))} \\ 1.0 & \text{if } K_w = 1 \text{ (small amplitude wave theory)} \end{cases}$$

The particle accelerations are computed by differentiating $\vec{v}_R$ and $\vec{v}_Z$ with respect to time. Thus:

$$\dot{\vec{v}}_R = \sum_{i=1}^{N_w} \frac{\cosh\left(k_i \bar{Z} f\right)}{\sinh\left(k_i d\right)} \left(\frac{2\pi}{\tau_i}\right) \left(\dot{\eta}_i - C\eta_i\right) \qquad (14.59\text{–}24)$$

$$\dot{\vec{v}}_Z = \sum_{i=1}^{N_w} \frac{\sinh\left(k_i \bar{Z} f\right)}{\sinh\left(k_i d\right)} \left(\frac{2\pi}{\tau_i}\right) \left(-\frac{2\pi}{\tau_i}\ \dot{\eta}_i - C\eta_i \left(\frac{\tau}{2\pi}\right)\right) \qquad (14.59\text{–}25)$$

where:

$$C = \begin{cases} \dot{\eta}_s \dfrac{2\pi}{\lambda_i} \dfrac{\overline{Z}\, d}{(d + \eta_s)^2} & \text{if } K_w = 0 \text{ (Wheeler (35))} \\ 0.0 & \text{if } K_w = 1 \text{ (small amplitude wave theory)} \end{cases}$$

Expanding equation 2.29 of the Shore Protection Manual (43) for a multiple component wave, the wave hydrodynamic pressure is:

$$P_o^d = \rho_w \, g \sum_{i=1}^{N_w} \eta_i \frac{\cosh\left[2\pi \dfrac{\overline{Z}}{\lambda_i}\right]}{\cosh\left[2\pi \dfrac{d}{\lambda_i}\right]} \qquad (14.59\text{–}26)$$

However, use of this equation leads to non–zero total pressure at the surface at the crest or trough of the wave. Thus, equation (14.59–26) is modified to be:

$$P_o^d = \rho_w \, g \sum_{i=1}^{N_w} \eta_i \frac{\cosh\left[2\pi \dfrac{\overline{Z} d}{\lambda_i (d + \eta_s)}\right]}{\cosh\left[2\pi \dfrac{d}{\lambda_i}\right]} \qquad (14.59\text{–}27)$$

which does result in a total pressure of zero at all points of the free surface. This dynamic pressure, which is calculated at the integration points during the stiffness pass, is extrapolated to the nodes for the stress pass. The hydrodynamic pressure for Stokes fifth order wave theory is:

$$P_o^d = \rho_w \, g \sum_{i=1}^{5} \eta_i \frac{\cosh\left(2\pi \dfrac{\overline{Z}}{\lambda_i}\right)}{\cosh\left(2\pi \dfrac{d}{\lambda_i}\right)} \qquad (14.59\text{–}28)$$

Other aspects of the Stokes fifth order wave theory are discussed by Skjelbreia et al (31). The modification as suggested by Nishimura et al (143) has been included. The stream function wave theory is described by Dean(59).

If both waves and current are present, the question of wave–current interaction must be dealt with. Three options are made available thru K_{cr} (input quantity KCRC on the **TBDATA** commands with **TB**,WATER):

For $K_{cr} = 0$, the current velocity at all points above the mean sea level is simply set equal to W_o, where W_o is the input current velocity at Z = 0.0. All points below the mean sea level have velocities selected as though there were no wave.

For K_{cr} = 1, the current velocity profile is “stretched” or “compressed” to fit the wave. In equation form, the Z coordinate location of current measurement is adjusted by

$$Z'(j) = Z(j)\,\frac{d + \eta_s}{d} \qquad (14.59\text{–}29)$$

where: Z(j) = Z coordinate location of current measurement (input quantity Z(j))

Z(j) = adjusted value of Z(j)

For K_{cr} = 2, the same adjustment as for K_{cr} = 1 is used, as well as a second change that accounts for “continuity.” That is,

$$W'(j) = W(j)\,\frac{d}{d + \eta_s} \qquad (14.59\text{–}30)$$

where: W(j) = velocity of current at this location (input quantity W(j))

W(j) = adjusted value of W(j)

These three options are shown pictorially in Figure 14.59–1.

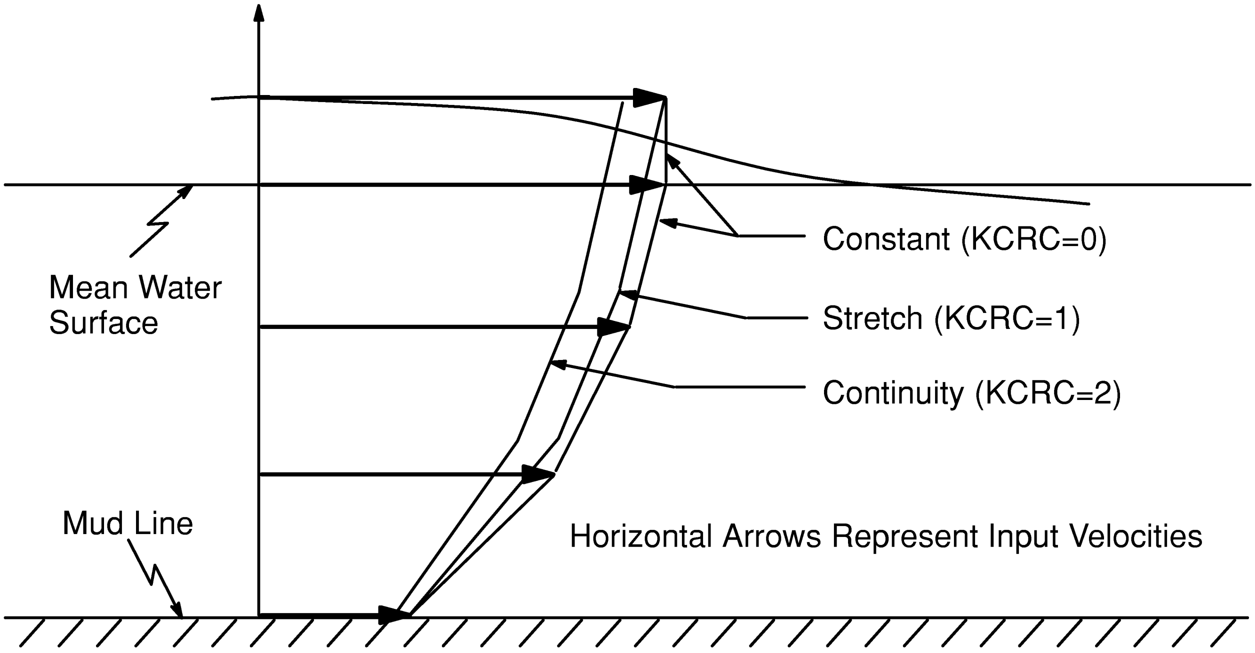


Figure 14.59–1 Velocity Profiles for Wave–Current Interactions

To compute the relative velocities ($\{\dot{u}_n\}$, $\{\dot{u}_t\}$), both the fluid particle velocity and the structure velocity must be available so that one can be subtracted from the other. The fluid particle velocity is computed using relationships such as equations (14.59–22) and (14.59–23) as well as current effects. The structure velocity is available through the Newmark time integration logic (see Section 17.2).

Finally, a generalized Morison's equation is used to compute a distributed load on the element to account for the hydrodynamic effects:

$$\begin{aligned}\{F/L\}_d \; &= \; C_D \rho_w \frac{D_e}{2} \left|\{\dot{u}_n\}\right| \{\dot{u}_n\} \; + \; C_M \rho_w \frac{\pi}{4} D_e^2 \; \{\dot{v}_n\} \\ &\quad + \; C_T \rho_w \frac{D_e}{2} \left|\{\dot{u}_t\}\right| \{\dot{u}_t\}\end{aligned} \qquad (14.59\text{–}31)$$

where: $\{F/L\}_d$ = vector of loads per unit length due to hydrodynamic effects

C_D = coefficient of normal drag (see below)

ϱ_w = water density $\left(\frac{\text{mass}}{\text{length}^3}\right)$ (input quantity DENSW on **TBDATA** command with **TB**,WATER)

D_e = outside diameter of the pipe with insulation (length)

$\{\dot{u}_n\}$ = normal relative particle velocity vector $\left(\frac{\text{length}}{\text{time}}\right)$

C_M = coefficient of inertia (input quantity CM on **R** command)

$\{\dot{v}_n\}$ = normal particle acceleration vector $\left(\frac{\text{length}}{\text{time}^2}\right)$

C_T = coefficient of tangential drag (see below)

$\{\dot{u}_t\}$ = tangential relative particle velocity vector $\left(\frac{\text{length}}{\text{time}}\right)$

Two integration points along the length of the element are used to generate the load vector. Integration points below the mud line are simply bypassed. For elements intersecting the free surface, the integration points are distributed along the wet length only. If the reduced load vector option is requested (KEYOPT(2) = 2), the moment terms are set equal to zero.

The coefficients of drag (C_D,C_T) may be defined in one of two ways:

1. They may be input as fixed numbers using the real constant table (input quantities CD and CT on **R** and **RMORE** commands), or
2. They may be input as functions of Reynold's number using the water motion table (input quantities RE, CD, and CT on the **TBDATA** commands with **TB**,WATER).

The dependency on Reynold's number (Re) may be expressed as:

$$C_D \; = \; f_D \; (\text{Re}) \qquad (14.59\text{–}32)$$

where: f_D = relationship defined by input quantities RE and CD of the water motion table

Re = $\{\dot{u}_n\}\dfrac{D_e\rho_w}{\mu}$

μ = viscosity (input as VISC on **MP** command)

and

$$C_T = f_T\ (\mathrm{Re}) \tag{14.59–33}$$

where: f_T = relationships defined by input quantities RE and CT on the **TBDATA** commands with **TB**,WATER

Re = $\{\dot{u}_t\}\dfrac{D_e\rho_w}{\mu}$

μ may be input as a temperature–dependent quantity, where the temperatures used are those given by input quantities T(i) of the water motion table.

14.59.6 Stress Output

The below two equations are specialized either to end I or to end J.

The stress output for the pipe format (KEYOPT(1) $\neq$ 1), is similar to PIPE16 (Section 14.16). The average axial stress is:

$$\sigma_x = \frac{F_n}{A} + \frac{D_i^2\ P_i\ -\ D_o^2\ P_o}{D_o^2\ -\ D_i^2} \tag{14.59–34}$$

where: σ_x = average axial stress (output quantity SAXL)

F_n = axial element reaction force (output quantity FX, adjusted for sign)

P_i = internal pressure

P_o = external pressure = $P_o^s + P_o^d$

and the hoop stress is:

$$\sigma_h = \frac{2\ P_i\ D_i^2 -\ P_o\left(D_o^2 + D_i^2\right)}{D_o^2\ -\ D_i^2} \tag{14.59–35}$$

where: σ_h = hoop stress at the outside surface of the pipe (output quantity SH)

Average values of P_i and P_o are reported as first and fifth items of the output quantities ELEMENT PRESSURES. Equation (14.59–35) is a specialization of equation (14.59–14). The outside surface is chosen as the bending stresses usually dominate over pressure induced stresses.

All stress results are given at the nodes of the element. However, the hydrodynamic pressure had been computed only at the two integration points. These two values are then used to compute hydrodynamic pressures at the two nodes of the element by extrapolation.

The stress output for the cable format (KEYOPT(1) = 1 with D_i = 0.0) is similar to that for LINK8 (Section 14.8), except that the stress is given with and without the external pressure applied:

$$\sigma_{xI} = \frac{F_\ell}{A} + P_o \tag{14.59–36}$$

$$\sigma_{eI} = \frac{F_\ell}{A} \tag{14.59–37}$$

$$F_a = A\sigma_{xI} \tag{14.59–38}$$

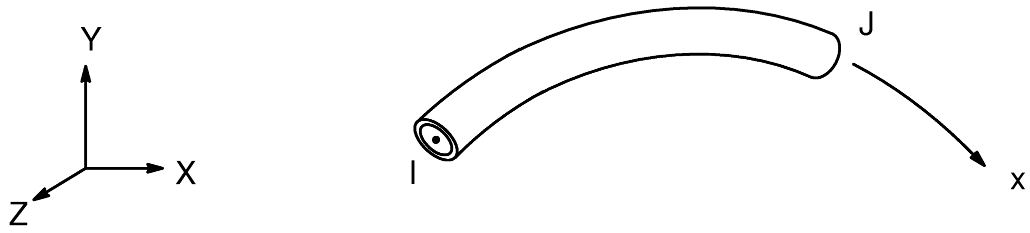
where: σ_{xI} = output quantity SAXL

σ_{eI} = output quantity SEQV

F_ℓ = axial force on node (output quantity FX)

F_a = axial force in the element (output quantity FAXL)

14.60 PIPE60 — Plastic Curved Pipe (Elbow)



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	No shape functions are explicitly used. Rather, a flexibility matrix similar to that developed by Chen (4) is inverted and used.	None
Mass Matrix	No shape functions are used. Rather a lumped mass matrix using only translational DOF is used.	None
Pressure, Thermal, and Newton–Raphson Load Vector	No shape functions are explicitly used. See development below.	8 around circumference at each end of the element. The points are located midway between the inside and outside surfaces

Load Type	Distribution
Element Temperature	Bilinear across cross–section, linear along length
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Internal and External: constant along length and around circumference. Lateral: varies trigonometrically along length

14.60.1 Assumptions and Restrictions

The radius/thickness ratio is assumed to be large.

14.60.2 Other Applicable Sections

The stiffness and mass matrices are identical to those derived for PIPE18 in Section 14.18. Section 14.16 discusses some aspects of the elastic stress printout.

14.60.3 Load Vector

The element load vector is computed in a linear analysis by:

$$\{F_\ell\} = [K_\ell]\left\{u^F\right\} \qquad (14.60\text{–}1)$$

and in a nonlinear (Newton–Raphson) analysis by:

$$\{F_\ell\} = [K_\ell]\left(\left\{u^F\right\} - \{u_{n-1}\}\right) \qquad (14.60\text{–}2)$$

where: $\{F_\ell\}$ = element load vector (in element coordinates) (applied loads minus Newton–Raphson restoring force) from previous iteration

$[K_\ell]$ = element stiffness matrix (in element coordinates)

$\{u^F\}$ = induced nodal displacements in the element (see equation (14.60–3))

$\{u_{n-1}\}$ = displacements of the previous iteration

The element coordinate system is a cylindrical system as shown in Figure 14.60–1.

The induced nodal displacement vector $\{u^F\}$ is defined by:

$$\{u^F\} = \left\{ \begin{array}{ll}
-\dfrac{R}{4}\sin\dfrac{\theta}{4}\cos\dfrac{\theta}{4}\displaystyle\sum_{j=1}^{8}\epsilon_j^{(1)} & \\
0 & \\
-\dfrac{R}{4}\sin^2\dfrac{\theta}{4}\displaystyle\sum_{j=1}^{8}\epsilon_j^{(1)} & \\
\dfrac{R\theta}{4D_m}\displaystyle\sum_{j=1}^{8}\gamma_j^{(1)} & \\
\dfrac{R\theta}{6D_m}\displaystyle\sum_{j=1}^{8}\dfrac{\epsilon_j^{(1)}}{\cos\beta_j} & j \neq 2, j \neq 6 \\
\dfrac{R\theta}{6D_m}\displaystyle\sum_{j=1}^{8}\dfrac{\epsilon_j^{(1)}}{\sin\beta_j} & j \neq 4, j \neq 8 \\
\dfrac{R}{4}\sin\dfrac{\theta}{4}\cos\dfrac{\theta}{4}\displaystyle\sum_{j=1}^{8}\epsilon_j^{(2)} & \\
0 & \\
-\dfrac{R}{4}\sin^2\dfrac{\theta}{4}\displaystyle\sum_{j=1}^{8}\epsilon_j^{(2)} & \\
-\dfrac{R\theta}{4D_m}\displaystyle\sum_{j=1}^{8}\gamma_j^{(2)} & \\
-\dfrac{R\theta}{6D_m}\displaystyle\sum_{j=1}^{8}\dfrac{\epsilon_j^{(2)}}{\cos\beta_j} & j \neq 2, j \neq 6 \\
-\dfrac{R\theta}{6D_m}\displaystyle\sum_{j=1}^{8}\dfrac{\epsilon_j^{(2)}}{\sin\beta_j} & j \neq 4, j \neq 8
\end{array} \right\} \qquad (14.60\text{–}3)$$

where:

$\epsilon_j^{(1)} = \epsilon^{th} + \epsilon_x^{pr} + \epsilon_x^{pl} + \epsilon_x^{cr} + \epsilon^{sw}$ at end I

$\epsilon_j^{(2)} = \epsilon^{th} + \epsilon_x^{pr} + \epsilon_x^{pl} + \epsilon_x^{cr} + \epsilon^{sw}$ at end J

$\gamma_j^{(1)} = \gamma_{xh}^{pl} + \gamma_{xh}^{cr}$ at end I

$\gamma_j^{(2)} = \gamma_{xh}^{pl} + \gamma_{xh}^{cr}$ at end J

ϵ^{th} = $\alpha \left(T_j - T_{REF}\right)$ (= thermal strain)

α = thermal coefficient of expansion (input as ALPX on **MP** command)

T_j = temperature at integration point j

ϵ_x^{pr} = axial strain due to pressure (see equation (14.16–10)

ϵ_x^{pl} = plastic axial strain (see Section 4.1)

ϵ_x^{cr} = axial creep strain (see Section 4.3)

ϵ^{sw} = swelling strain (see Section 4.4)

γ_{xh}^{pl} = plastic shear strain (see Section 4.1)

γ_{xh}^{cr} = creep shear strain (see Section 4.3)

R = radius of curvature (input as RADCUR on **R** command)

D_m = 1/2 (D_o + D_i) (= average diameter)

D_o = outside diameter (input as OD on **R** command)

D_i = D_o – 2t (= inside diameter)

t = thickness (input as TKWALL on **R** command)

θ = subtended angle of the elbow

β_j = angular position of integration point j on the circumference Figure 14.60–2 (output quantity ANGLE)

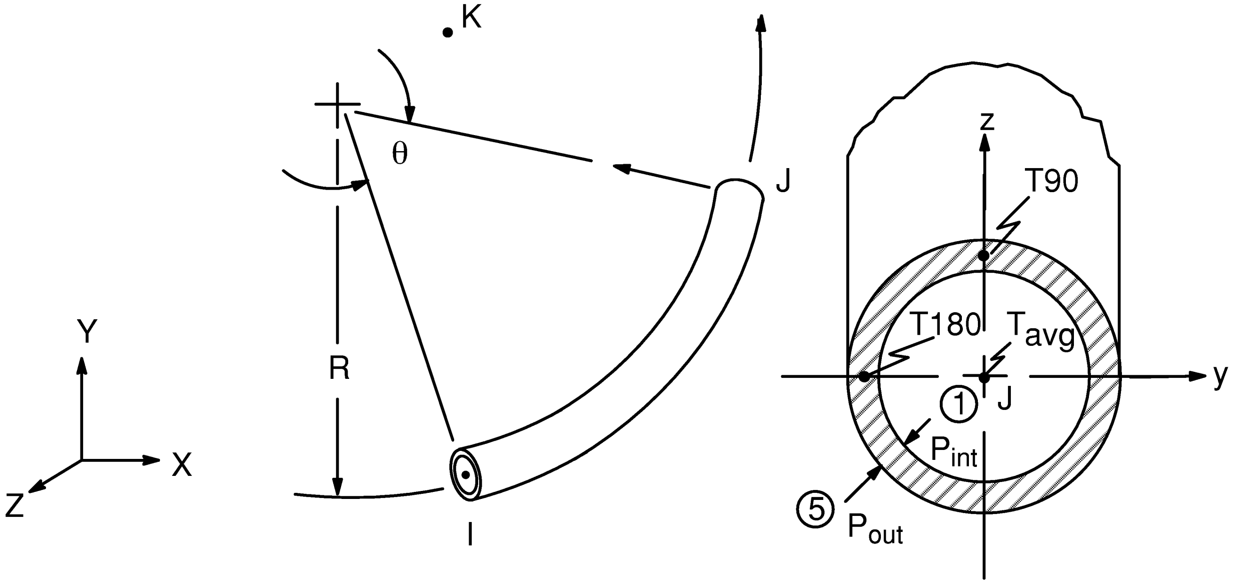


Figure 14.60–1 3–D Plastic Curved Pipe Element Geometry

There are eight integration points around the circumference at each end of the element, as shown in Figure 14.60–2. The assumption has been made that the elbow has a large radius–to–thickness ratio so that the integration points are located at the midsurface of the shell. Since there are integration points only at each end of the element, the subtended angle of the element should not be too large. For example, if there are effects other than internal pressure and in–plane bending, the elements should have a subtended angle no larger than 45°.

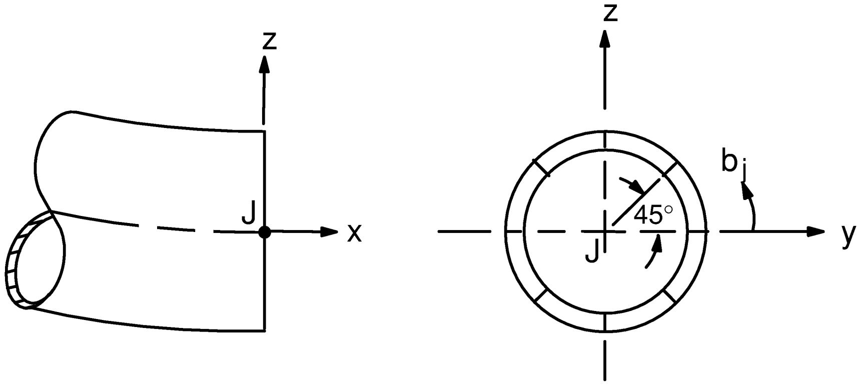


Figure 14.60–2 Integration Point Locations at End J

14.60.4 Stress Calculations

The stress calculations take place at each integration point, and have a different basis than for PIPE18, the elastic elbow element. The calculations have three phases:

1. Computing the modified total strains (ε′).
2. Using the modified total strains and the material properties, computing the change in plastic strains and then the current elastic strains.
3. Computing the current stresses based on the current elastic strains.

Phase 2 is discussed in Section 4.1. Phase 1 and 3 are discussed below. Phase 1: The modified total strains at an integration point are computed as:

$$\{\epsilon'\} = [D]^{-1}\{\sigma_b\} \tag{14.60–4}$$

where: $\epsilon' = \begin{Bmatrix} \epsilon_x^{d'} \\ \epsilon_h^{d'} \\ \epsilon_r^{'} \\ \gamma_{xh}^{'} \end{Bmatrix}$

$$[D]^{-1} = \begin{bmatrix} \frac{1}{E} & -\frac{\nu}{E} & -\frac{\nu}{E} & 0 \\ -\frac{\nu}{E} & \frac{1}{E} & -\frac{\nu}{E} & 0 \\ -\frac{\nu}{E} & -\frac{\nu}{E} & \frac{1}{E} & 0 \\ 0 & 0 & 0 & \frac{2(1+\nu)}{E} \end{bmatrix}$$

x, h, r = subscripts representing axial, hoop, and radial directions, respectively

E = Young's modulus (input as EX on **MP** command)

ν = Poisson's ratio (input as PRXY or NUXY on **MP** command)

$\{\sigma_b\}$, the integration point stress vector before plasticity computations, is defined as:

$$\{\sigma_b\} = \begin{Bmatrix} \sigma_x \\ \sigma_h \\ \sigma_r \\ \tau_{xh} \end{Bmatrix} \tag{14.60–5}$$

These terms are defined by:

$$\sigma_x = \frac{F_x}{A^w} + S_y M_y + S_z M_z + \frac{D_i P_i - D_o P_o}{4t} \tag{14.60–6}$$

$$\sigma_h = \nu S_y M_y + \nu S_z M_z + \left(\frac{D_o}{2t} - \frac{2}{5} \right) \left[\frac{R + \frac{1}{2} r \sin\phi_j}{R + r \sin\phi_j} \right] (P_i - P_o) \tag{14.60–7}$$

$$\sigma_r = -\frac{P_i + P_o}{2} \tag{14.60–8}$$

$$\tau_{xh} = -\frac{2}{A^w} \left(F_y \cos\beta_j + F_z \sin\beta_j \right) - \frac{S_x M_x}{2} \tag{14.60–9}$$

where: F_y, F_z, M_x = forces on element at node by integration point (see equation (14.60–10) below)

A^w = $\frac{\pi}{4} \left(D_o^2 - D_i^2 \right)$

S_x = $\dfrac{32 D_o}{\pi \left(D_o^4 - D_i^4 \right)}$

S_y = $-S_x \left(\sin\phi_j + C_2 \left((1.5 C_1 - 18.75) \sin 3\phi_j + 11.25 \sin 5\phi_j \right) \right)$

$$S_z = S_x\left(\cos\phi_j + C_2\left((1.5C_1 - 18.75)\cos 3\phi_j + 11.25\cos 5\phi_j\right)\right)$$

$$\phi_j = \beta_j - \frac{\pi}{2}$$

$$r = \frac{D_o + D_i}{4}$$

P_i = internal pressure (input on **SFE** command)

P_o = external pressure (input on **SFE** command)

$$C_1 = 17 + 600\ C_3^{\,2} + 480\ \frac{PR^2}{Ert}$$

$$C_2 = \frac{1}{\left(1 - \nu^2\right)\left(C_1C_4 - 6.25 - 4.5C_1\right)}$$

$$C_3 = \frac{Rt}{r^2\ \sqrt{1 - \nu^2}}$$

$$C_4 = 5 + 6\ C_3^{\,2} + 24\ \frac{PR^2}{Ert}$$

$$P = P_i - P_o$$

Note that S_y and S_z are expressed in three–term Fourier series around the circumference of the pipe cross–section. These terms have been developed from the ASME Code(60). Note also that ϕ_j is the same angle from the element y axis as β_j is for PIPE20. The forces on both ends of the element (F_y, M_x, etc.) are computed from:

$$\{F_e\} = [T_R]\ \left([K_e^p]\ \{\Delta u_e\} - \{F_\ell\}\right) \qquad (14.60\text{–}10)$$

where: $\{F_e\}$ = $\lfloor F_{xI}\ \ldots\ M_{zJ} \rfloor^T$ = forces on element in element coordinate system

$[T_R]$ = global to local conversion matrix (note that the local x axis is not straight but rather is curved along the centerline of the element)

$[K_e]$ = element stiffness matrix (global Cartesian coordinates)

$\{\Delta u_e\}$ = element incremental displacement vector

Phase 3: Performed after the plasticity calculations, Phase 3 is done simply by:

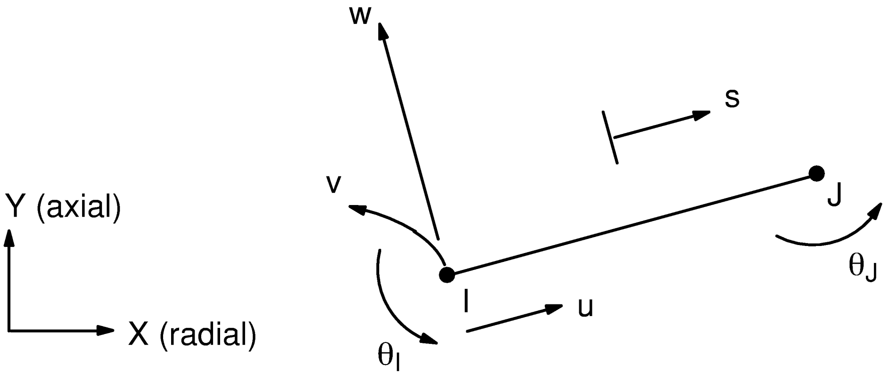
$$\{\sigma\} = [D]\left\{\epsilon^e\right\} \qquad (14.60\text{–}11)$$

where: $\{\varepsilon^e\}$ = elastic strain after the plasticity calculations

The {σ} vector, which is used for output, is defined with the same terms as in equation (14.60–5). But lastly, σ_r is redefined by equation (14.60–8) as this stress value must be maintained, regardless of the amount of plastic strain.

As long as the element remains elastic, additional printout is given during the solution phase. The stress intensification factors (C_σ) of PIPE18 are used in this printout, but are not used in the printout associated with the plastic stresses and strains. The maximum principal stresses, the stress intensity, and equivalent stresses are compared (and replaced if necessary) to the values of the plastic printout at the eight positions around the circumference at each end. Also, the elastic printout is based on outer–fiber stresses, but the plastic printout is based on mid–thickness stresses. Further, other thin–walled approximations in equations (14.60–6) and (14.60–7) are not used by the elastic printout. Hence some apparent inconsistency appears in the printout.

14.61 SHELL61 — Axisymmetric–Harmonic Structural Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.4.2–1), (12.4.2–2), and (12.4.2–3) . If extra shape functions are not included (KEYOPT(3) = 1): equations (12.4.1–1), (12.4.1–2), and (12.4.1–3)	3 along length
Mass Matrix	Equations (12.3.1–1), (12.3.1–2), and (12.3.1–3)	Same as stiffness matrix
Stress Stiffness Matrix	Equations (12.4.1–1), (12.4.1–2), and (12.4.1–3)	Same as stiffness matrix
Thermal and Pressure Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element Temperature	Linear through thickness and along length, harmonic around circumference
Nodal Temperature	Constant through thickness, linear along length, harmonic around circumference
Pressure	Linear along length, harmonic around circumference

Reference: Zienkiewicz(39)

14.61.1 Other Applicable Sections

Chapter 2 discusses fundamentals of linear elements. Section 13.1 describes integration point locations. Section 14.25 has a discussion on temperature, applicable to this element.

14.61.2 Assumptions and Restrictions

The material properties are assumed to be constant around the entire circumference, regardless of temperature dependent material properties or loading.

14.61.3 Stress, Force, and Moment Calculations

Element output comes in two forms:

1. Stresses as well as forces and moments per unit length: This printout is controlled by the KEYOPT(6). The thru–the–thickness stress locations are shown in Figure 14.61–1. The stresses are computed using standard procedures as given in Section 2.3. The stresses may then be integrated thru the thickness to give forces per unit length and moments per unit length at requested points along the length:

$$T_x = \sigma_x\big|_c t \qquad (14.61\text{–}1)$$

$$T_z = \sigma_z\big|_c t \qquad (14.61\text{–}2)$$

$$T_{xz} = \sigma_{xz}\big|_c t \qquad (14.61\text{–}3)$$

$$M_x = \left(\sigma_x\big|_t - \sigma_x\big|_b\right)\frac{t^2}{12} \tag{14.61–4}$$

$$M_z = \left(\sigma_z\big|_t - \sigma_z\big|_b\right)\frac{t^2}{12} \tag{14.61–5}$$

$$M_{xz} = \left(\sigma_{xz}\big|_t - \sigma_{xz}\big|_b\right)\frac{t^2}{12} \tag{14.61–6}$$

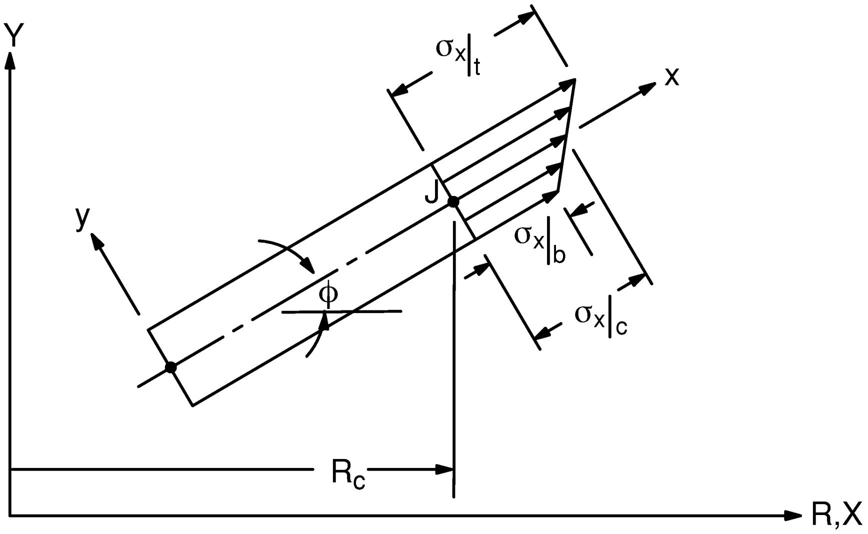


Figure 14.61–1 Stress Locations

where:

T_x, T_z, T_{xz}, M_x, M_z, M_{xz} = output as TX, TZ, TXZ, MX, MZ, MXZ, respectively

t = thickness (input quantities TK(I), TK(J) on **R** command)

σ_x, σ_y, σ_z, σ_{xz} = output as SX, SY, SZ, and SXZ, respectively

$\sigma_x\big|_c$ = x stress at centerplane (also nodal locations)

= $\left(\sigma_x\big|_t + \sigma_x\big|_b\right)/2$

$\sigma_x\big|_t$ = x stress at top

$\sigma_x\big|_b$ = x stress at bottom

2. Forces and moments on a circumference basis: This printout is controlled by KEYOPT(4). The values are computed using:

$$\{F_\ell\} = \left[T_R\right]^T\left(\left[K_e\right]\{u_e\} - \left\{F_e^{th}\right\} - \left\{F_e^{pr}\right\}\right) \tag{14.61–7}$$

where:

$$F_\ell = \lfloor F^r_{x,1} \;\; F^r_{y,1} \;\; F^r_{z,1} \;\; M^r_{z,1} \;\; F^r_{x,2} \;\; F^r_{y,2} \;\; F^r_{z,2} \;\; M^r_{z,2} \rfloor^T$$

= output as MFOR and MMOM

$[T_R]$ = local to global transformation matrix

$[K_e]$ = element stiffness matrix

$\{u_e\}$ = nodal displacements

$\left\{F_e^{th}\right\}$ = element thermal load vector

$\left\{F_e^{pr}\right\}$ = element pressure load vector

Another difference between the two types of output are the nomenclature conventions. Since the first group of output uses a shell nomenclature convention and the second group of output uses a nodal nomenclature convention, M_z and M^r_z represent moments in different directions.

The rest of this subsection will describe some of the expected relationships between these two methods of output at the ends of the element. This is done to give a better understanding of the terms, and possibly detect poor internal consistency, suggesting that a finer mesh is in order. It is advised to concentrate on the primary load carrying mechanisms. In order to relate these two types of output in the printout, they have to be requested with both KEYOPT(6) $\geq$ 1 and KEYOPT(4) = 1. Further, care must be taken to ensure that the same end of the element is being considered.

The axial reaction force based on the stress over an angle $\Delta\beta$ is:

$$F^r_x = \int_{-t/2}^{t/2} \left[\frac{\left(\sigma_x\big|_t + \sigma_x\big|_b\right)}{2} + \frac{\left(\sigma_x\big|_t - \sigma_x\big|_b\right) y}{t} \right] \Delta\beta \; (R_c - y \sin\phi) \, dy \qquad (14.61\text{–}8)$$

or

$$F^r_x = \Delta\beta \left[\frac{\left(\sigma_x\big|_t + \sigma_x\big|_b\right)}{2} R_c t - \left(\sigma_x\big|_t - \sigma_x\big|_b\right) \sin\phi \, \frac{t^2}{12} \right] \qquad (14.61\text{–}9)$$

where: R_c = radius at midplane

t = thickness

The reaction moment based on the stress over an angle $\Delta\beta$ is:

$$M_x^r = \int_{-t/2}^{t/2} \left[\frac{\left(\sigma_x\big|_t + \sigma_x\big|_b\right)}{2} + \frac{\left(\sigma_x\big|_t - \sigma_x\big|_b\right) y}{t} \right] y \Delta\beta \left(R_c - y \sin\phi\right) dy \qquad (14.61\text{–}10)$$

or

$$M_x^r = \Delta\beta \left[-\frac{\left(\sigma_x\big|_t + \sigma_x\big|_b\right)}{2} \frac{t^3 \sin\phi}{12} + \left(\sigma_x\big|_t - \sigma_x\big|_b\right) R_c \frac{t^2}{12} \right] \qquad (14.61\text{–}11)$$

Since SHELL61 computes stiffness matrices and load vectors using the entire circumference for axisymmetric structures, $\Delta\beta = 2\pi$. Using this fact, the definition of $\sigma_x\big|_c$, and equations (14.61–1) and (14.61–4), equations (14.61–9) and (14.61–11) become:

$$F_x^r = 2\pi \left(R_c T_x - \sin\phi \, M_x\right) \qquad (14.61\text{–}12)$$

$$M_z^r = 2\pi \left(-\frac{t^2 \sin\phi}{12} T_x + R_c M_x \right) \qquad (14.61\text{–}13)$$

As the definition of ϕ is critical for these equations, Figure 14.61–2 is provided to show ϕ in all four quadrants.

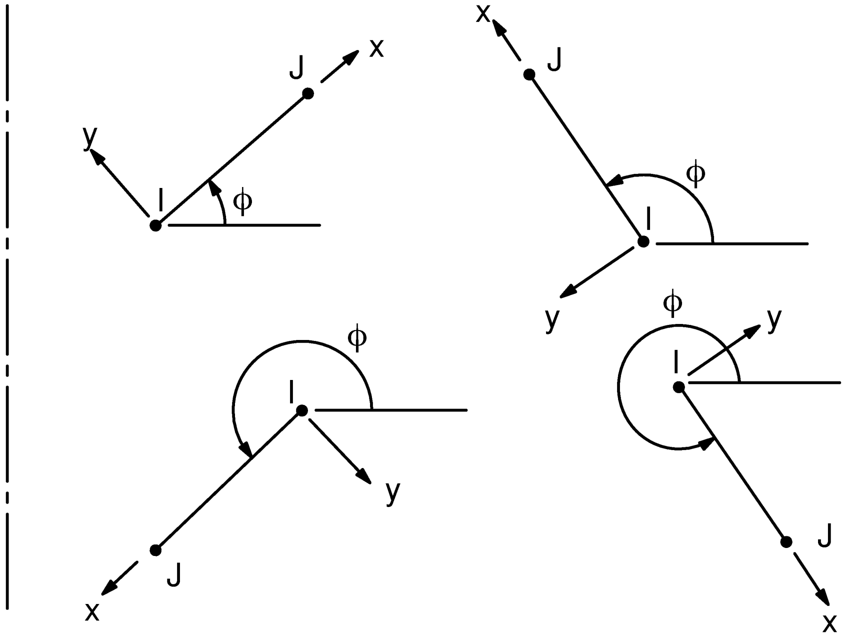


Figure 14.61–2 Element Orientations

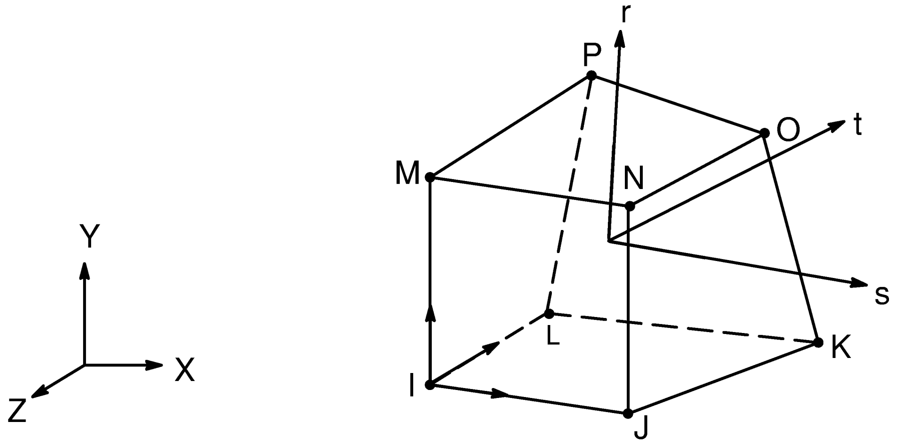
In a uniform stress (σ_x) environment, a reaction moment will be generated to account for the greater material on the outside side. This is equivalent to moving the reaction point outward a distance y_f. y_f is computed by:

$$y_f = \frac{M_z^r}{F_x^r} \qquad (14.61\text{–}14)$$

Using equations (14.61–12) and (14.61–13) and setting M_x to zero gives:

$$y_f = -\frac{t^2 \sin \phi}{12 R_c} \qquad (14.61\text{–}15)$$

14.62 SOLID62 — 3–D Coupled Magnetic–Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Vector Potential Coefficient Matrix	Equations (12.8.18–7), (12.8.18–8) and (12.8.18–9)	2 x 2 x 2
Damping (Eddy Current) Matrix	Equations (12.8.18–1), (12.8.18–2) and (12.8.18–3)	2 x 2 x 2
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3) or, if modified extra shape functions are included (KEYOPT(1)=0) and element has 8 unique nodes (12.8.19–1), (12.8.19–2), and (12.8.19–3)	2 x 2 x 2
Mass Matrix	Same as damping matrix	2 x 2 x 2
Stress Stiffness Matrix	Same as damping matrix	2 x 2 x 2
Thermal Load Vector	Same as stiffness matrix	2 x 2 x 2

Matrix or Vector	Shape Functions		Integration Points
Magnetic Force Load Vector	Same as damping matrix		2 x 2 x 2
Pressure Load Vector	Quad	Equation (12.5.8–1) and (12.5.8–2)	2 x 2
	Triangle	Equation (12.5.1–1) and (12.5.1–2)	3
Permanent Magnet and Applied Current Load Vector	Same as coefficient matrix		2 x 2 x 2

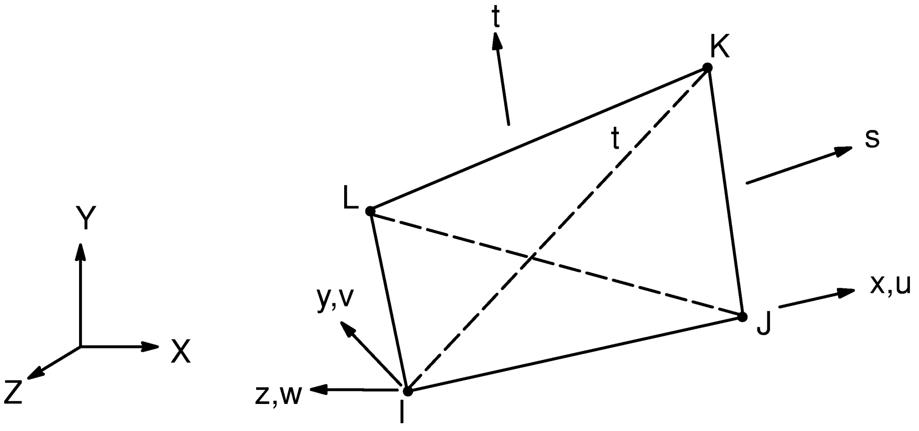
Load Type	Distribution
Current Density and Phase Angle	Trilinear thru element
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Wilson(38), Taylor et al.(49), Coulomb(76), Biro et al.(120)

14.62.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Sections 5.2 and 5.3 contain a discussion of the 2–D magnetic vector potential formulation which is similar to the 3–D formulation of this element. Section 13.1 describes integration point locations.

14.63 SHELL63 — Elastic Shell



Matrix or Vector		Geometry	Shape Functions	Integration Points
Stiffness Matrix	Membrane	Quad	Equations (12.5.11–1) and (12.5.11–2) (and, if modified extra shape functions are included (KEYOPT(3)=0) and element has 4 unique nodes, equations (12.5.12–1) thru (12.5.12–3))	2 x 2
		Triangle	Equations (12.5.3–1) thru (12.5.3–3)	1
	Bending	Four triangles that are overlaid are used. These subtriangles refer to equation (12.5.3–3)		3 (for each triangle)

Matrix or Vector		Geometry	Shape Functions	Integration Points
Mass Matrix	Membrane	Quad	Equations (12.5.8–1), (12.5.8–2) and (12.5.8–3)	2 x 2
		Triangle	Equations (12.5.1–1), (12.5.1–2) and (12.5.1–3)	1
	Bending	Four triangles that are overlaid are used. These triangles connect nodes IJK, IJL, KLI, and KLJ. w is defined as given in Zienkiewicz(39)		3 (for each triangle)
Transverse Pressure Load Vector	Reduced shell pressure loading (KEYOPT(6)=0) (Load vector excludes moments)		One–sixth (one–third for triangles) of the total pressure times the area is applied to each node normal of each subtriangle of the element	None
	Consistent shell pressure loading (KEYOPT(6)=2) (Load vector includes moments)		Same as mass matrix	Same as mass matrix
Edge Pressure Load Vector	Quad	Equations (12.5.8–1) and (12.5.8–2) specialized to the edge		2
	Triangle	Equations (12.5.1–1) and (12.5.1–2) specialized to the edge		2
Foundation Stiffness Matrix	Same as mass matrix			Same as mass matrix
Stress Stiffness Matrix	Same as mass matrix			Same as mass matrix
Thermal Load Vector	Same as stiffness matrix			Same as stiffness matrix

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru thickness
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element, linear along each edge

14.63.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.63.2 Foundation Stiffness

If K_f, the foundation stiffness, is input, the out–of–plane stiffness matrix is augmented by three or four springs to ground. The number of springs is equal to the number of distinct nodes, and their direction is normal to the plane of the element. The value of each spring is:

$$K_{f,i} = \frac{\Delta\ K_f}{N_d} \tag{14.63–1}$$

where:
$K_{f,i}$ = normal stiffness at node i
Δ = element area
K_f = foundation stiffness (input as EFS on **R** command)
N_d = number of distinct nodes

The output includes the foundation pressure, computed as:

$$\sigma_p = \frac{K_f}{4}\left(w_I + w_J + w_K + w_L\right) \tag{14.63–2}$$

where:
σ_p = foundation pressure (output as FOUND, PRESS)
w_I, etc. = lateral deflection at node I, etc.

14.63.3 In–Plane Rotational Stiffness

The in–plane rotational (drilling) DOF has no stiffness associated with it, based on the shape functions. A small stiffness is added to prevent a numerical instability following the approach presented by Kanok–Nukulchai(26) for non–warped elements if KEYOPT(1) = 0. KEYOPT(3) = 2 is used to include the Allman–type rotational DOFs (as described with SHELL43).

14.63.4 Warping

If all four nodes are not defined to be in the same flat plane (or if an initially flat element loses its flatness due to large displacements (**NLGEOM**,ON)), additional calculations are performed in SHELL63. The purpose of the additional calculations is to convert the matrices and load vectors of the element from the points on the flat plane in which the element is derived to the actual nodes. Physically, this may be thought of as adding short rigid offsets between the flat plane of the element and the actual nodes. (For the membrane stiffness only case (KEYOPT(1)=1), the limits given with SHELL41 are used). When these offsets are required, it implies that the element is not flat, but rather it is "warped." To account for the warping, the following procedure is used: First, the normal to element is computed by taking the vector cross–product (the common normal) between the vector from node I to node K and the vector from node J to node L. Then, the check can be made to see if extra calculations are needed to account for warped elements. This check consists of comparing the normal to each of the four element corners with the element normal as defined above. The corner normals are computed by taking the vector cross–product of vectors representing the two adjacent edges. All vectors are normalized to 1.0. If any of the three global Cartesian components of each corner normal differs from the equivalent component of the element normal by more than .00001, then the element is considered to be warped.

A warping factor is computed as:

$$\phi = \frac{D}{t} \tag{14.63–3}$$

where: D = component of the vector from the first node to the fourth node parallel to the element normal

t = average thickness of the element

If: $\phi \leq$ 0.1 no warning message is printed

$.10 \leq \phi \leq$ 1.0 a warning message is printed

$1.0 < \phi$ a message suggesting the use of triangles is printed and the run terminates

To account for the warping, the following matrix is developed to adjust the output matrices and load vector:

$$[W] = \begin{bmatrix} [w_1] & [0] & [0] & [0] \\ [0] & [w_2] & [0] & [0] \\ [0] & [0] & [w_3] & [0] \\ [0] & [0] & [0] & [w_4] \end{bmatrix} \tag{14.63–4}$$

$$\left[w_i\right] = \begin{bmatrix} 1 & 0 & 0 & 0 & Z_i^o & 0 \\ 0 & 1 & 0 & Z_i^o & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (14.63–5)$$

where: Z_i^o = offset from the average plane at node i

and the DOF are in the usual order of UX, UY, UZ, ROTX, ROTY, and ROTZ. To ensure the location of the average plane goes through the middle of the element, the following condition is met:

$$Z_1^o + Z_2^o + Z_3^o + Z_4^o = 0 \qquad (14.63–6)$$

14.63.5 Options for Non–Uniform Material

SHELL63 can be adjusted for non–uniform materials, using an approach similar to that of Takemoto and Cook(107). Considering effects in the element x direction only, the loads are related to the displacement by:

$$T_x = t\ E_x\ \epsilon_x \qquad (14.63–7)$$

$$M_x = -\frac{t^3\ E_x}{12\left(1 - \nu_{xy}\ \nu_{yx}\right)}\ \kappa_x \qquad (14.63–8)$$

where:
F_x = force per unit width
t = thickness (input quantities TK(I), TK(J), TK(K), TK(L) on **R** command)
E = Young’s modulus in x direction (input as EX on **MP** command)
ε_x = strain of middle fiber in x direction
M_x = moment per unit length
ν_{xy} = Poisson’s ratio (input as PRXY or NUXY on **MP** command)
ν_{yx} = Poisson’s ratio (see Section 2.1)
κ_x = curvature in x direction

A non–uniform material may be represented with equation (14.63–8) as:

$$M_x = -\ C_r\ \frac{t^3\ E_x}{12\left(1 - \nu_{xy}\ \nu_{yx}\right)}\ \kappa_x \qquad (14.63–9)$$

where: C_r = input quantity RMI on **RMORE** command

The above discussion relates only to the formulation of the stiffness matrix.

Similarly, stresses for uniform materials are determined by:

$$\sigma_x^{top} = E\left(\epsilon_x + \frac{t}{2}\,\kappa_x\right) \tag{14.63–10}$$

$$\sigma_x^{bot} = E\left(\epsilon_x - \frac{t}{2}\,\kappa_x\right) \tag{14.63–11}$$

where: σ_x^{top} = x direction stress at top fiber

σ_x^{bot} = x direction stress at bottom fiber

For non–uniform materials, the stresses are determined by:

$$\sigma_x^{top} = E\left(\epsilon_x + c_t\,\kappa_x\right) \tag{14.63–12}$$

$$\sigma_x^{bot} = E\left(\epsilon_x - c_b\,\kappa_x\right) \tag{14.63–13}$$

where: c_t = input quantity CTOP, **RMORE** command

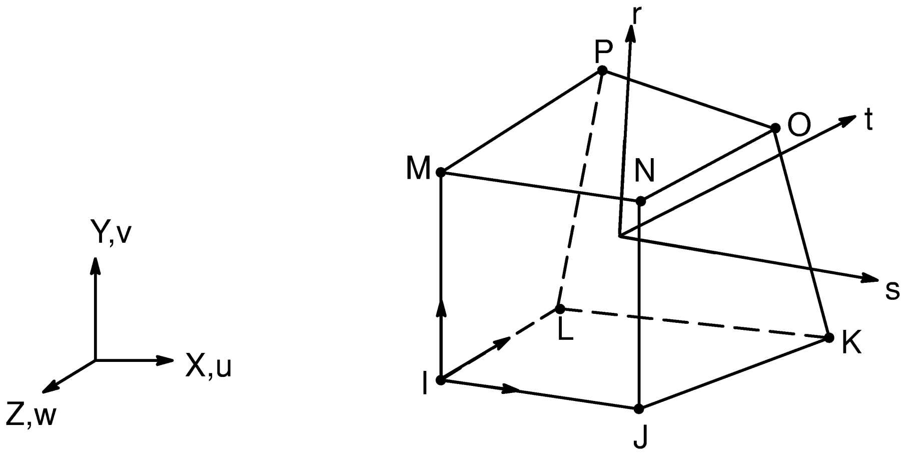
c_b = input quantity CBOT, **RMORE** command

The output quantities MX, MY, MXY are determined from the output stresses rather than from equation (14.63–9).

14.63.6 Usage of ERESX Command

The **ERESX**,NO command may be used to specify that integration point results are to be copied to the nodes. For the case of quadrilateral shaped elements, the bending results of each subtriangle are averaged and copied to the node of the quadrilateral which shares two edges with that subtriangle.

14.64 SOLID64 — 3–D Anistropic Structural Solid



<table>
<tr><th>Matrix or Vector</th><th colspan="2">Shape Functions</th><th>Integration Points</th></tr>
<tr><td>Stiffness Matrix</td><td colspan="2">Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3) or if modified extra shape functions are included (KEYOPT(1)=0)
and element has 8 unique nodes: equations (12.8.19–1), (12.8.19–2), and (12.8.19–3)</td><td>2 x 2 x 2</td></tr>
<tr><td>Mass Matrix</td><td colspan="2">Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3)</td><td>2 x 2 x 2</td></tr>
<tr><td>Stress Stiffness Matrix</td><td colspan="2">Same as mass matrix</td><td>2 x 2 x 2</td></tr>
<tr><td>Thermal Load Vector</td><td colspan="2">Same as stiffness matrix</td><td>2 x 2 x 2</td></tr>
<tr><td rowspan="2">Pressure Load Vector</td><td>Quad</td><td>Equation (12.5.8–1) and (12.5.8–2)</td><td>2 x 2</td></tr>
<tr><td>Triangle</td><td>Equation (12.5.1–1) and (12.5.1–2)</td><td>3</td></tr>
</table>

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Wilson(38), Taylor(49)

14.64.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. Section 13.5 defines positive definite matrices.

14.64.2 Stress–Strain Matrix

As referred to in Section 2.1, the stresses and strains are related by:

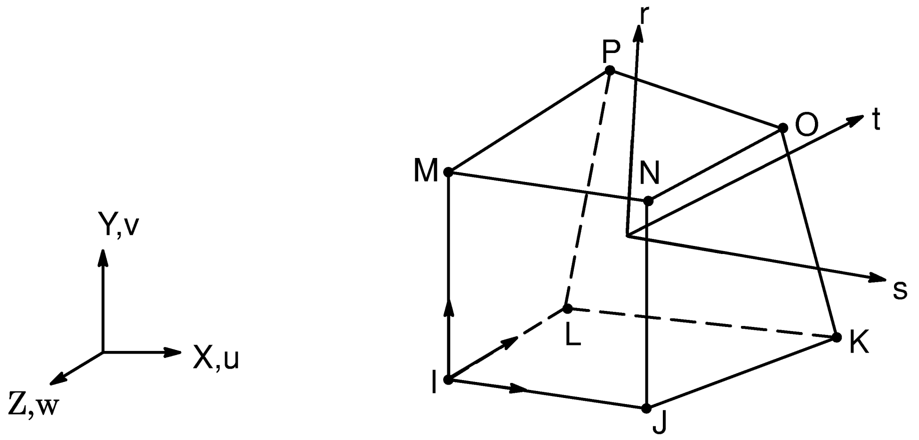
$$\{\sigma\} = [D]\left(\{\epsilon\} - \left\{\epsilon^{th}\right\}\right) \tag{14.64–1}$$

when using **TB**,ANEL,,,,0 or

$$\{\epsilon\} = \left\{\epsilon^{th}\right\} + [D]^{-1}\{\sigma\} \tag{14.64–2}$$

when using **TB**,ANEL,,,,1. The input must use the same order of components as given in Section 2.1, i.e. ε_x, ε_y, ε_z, ε_{xy}, ε_{yz}, ε_{xz}. While $\{\varepsilon^{th}\}$ is restricted to orthotropic input, [D] may be input as a full anisotropic matrix because 21 independent values are used in its makeup for this element. Symmetry of the [D] matrix is ensured, but it is up to the user to provide values so that the matrix is positive definite. If it is not, the program will terminate.

14.65 SOLID65 — 3–D Reinforced Concrete Solid



Matrix or Vector	Shape Functions		Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2) and (12.8.18–3), or if modified extra shape functions are included (KEYOPT(1)=0) and element has 8 unique nodes equations (12.8.19–1), (12.8.19–2) and (12.8.19–3)		2 x 2 x 2
Mass Matrix	Equations (12.8.18–1), (12.8.18–2) and (12.8.18–3)		2 x 2 x 2
Thermal Load Vector	Same as stiffness matrix		2 x 2 x 2
Pressure Load Vector	Quad	Equation (12.5.8–1) and (12.5.8–2)	2 x 2
	Triangle	Equation (12.5.1–1) and (12.5.1–2)	3

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: William and Warnke(37), Wilson(38), Taylor(49)

14.65.1 Assumptions and Restrictions

1. Cracking is permitted in three orthogonal directions at each integration point.
2. If cracking occurs at an integration point, the cracking is modeled through an adjustment of material properties which effectively treats the cracking as a "smeared band" of cracks, rather than discrete cracks.
3. The concrete material is assumed to be initially isotropic.
4. Whenever the reinforcement capability of the element is used, the reinforcement is assumed to be "smeared" throughout the element.
5. In addition to cracking and crushing, the concrete may also undergo plasticity, with the Drucker–Prager failure surface being most commonly used. In this case, the plasticity is done before the cracking and crushing checks.

14.65.2 Description

SOLID65 allows the presence of four different materials within each element; one matrix material (e.g. concrete) and a maximum of three independent reinforcing materials. The concrete material is capable of directional integration point cracking and crushing besides incorporating plastic and creep behavior. The reinforcement (which also incorporates creep and plasticity) has uniaxial stiffness only and is assumed to be smeared throughout the element. Directional orientation is accomplished through user specified angles.

14.65.3 Linear Behavior – General

The stress–strain matrix [D] used for this element is defined as:

$$[D] = \left(1 - \sum_{i=1}^{N_r} V_i^R\right) [D^c] + \sum_{i=1}^{N_r} V_i^R \, [D^r]_i \qquad (14.65\text{–}1)$$

where: N_r = number of reinforcing materials (maximum of three, all reinforcement is ignored if MAT1 (explained below) equals zero. Also, if MAT1, MAT2, or MAT3 equals the concrete material number, the reinforcement with that material number is ignored)

V_i^R = ratio of the volume of reinforcing material i to the total volume of the element (input as VRi on **R** command)

$[D^c]$ = stress–strain matrix for concrete, defined by equation (14.65–2)

$[D^r]_i$ = stress–strain matrix for reinforcement i, defined by equation (14.65–3)

MAT1, MAT2, MAT3=material numbers associated with material behavior of reinforcement (input as MAT1, MAT2, and MAT3 on **R** command)

14.65.4 Linear Behavior – Concrete

The matrix $[D^c]$ is derived by specializing the orthotropic stress–strain relations defined by equation (2.1–4) to the case of an isotropic material or

$$[D^c] = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \qquad (14.65–2)$$

where: E = Young's modulus for concrete (input as EX on **MP** command)

ν = Poisson's ratio for concrete (input as PRXY or NUXY on **MP** command)

14.65.5 Linear Behavior – Reinforcement

The orientation of the reinforcement i within an element is depicted in Figure 14.65–1. The element coordinate system is denoted by (X, Y, Z) and $\left(x_i^r,\ y_i^r,\ z_i^r\right)$ describes the coordinate system for reinforcement type i. The stress–strain matrix with respect to each coordinate system $\left(x_i^r,\ y_i^r,\ z_i^r\right)$ has the form

$$\begin{Bmatrix} \sigma_{xx}^r \\ \sigma_{yy}^r \\ \sigma_{zz}^r \\ \sigma_{xy}^r \\ \sigma_{yz}^r \\ \sigma_{xz}^r \end{Bmatrix} = \begin{bmatrix} E_i^r & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \epsilon_{xx}^r \\ \epsilon_{yy}^r \\ \epsilon_{zz}^r \\ \epsilon_{xy}^r \\ \epsilon_{yz}^r \\ \epsilon_{xz}^r \end{Bmatrix} = \left[D^r\right]_i \begin{Bmatrix} \epsilon_{xx}^r \\ \epsilon_{yy}^r \\ \epsilon_{zz}^r \\ \epsilon_{xy}^r \\ \epsilon_{yz}^r \\ \epsilon_{xz}^r \end{Bmatrix} \qquad (14.65\text{–}3)$$

where: E_i^r = Young's modulus of reinforcement type i (input as EX on **MP** command)

It may be seen that the only nonzero stress component is σ_{xx}^r, the axial stress in the x_i^r direction of reinforcement type i. The reinforcement direction x_i^r is related to element coordinates X, Y, Z through

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = \begin{Bmatrix} \cos\theta_i\ \cos\phi_i \\ \sin\theta_i\ \cos\phi_i \\ \sin\phi_i \end{Bmatrix} x_i^r = \begin{Bmatrix} \ell_1^r \\ \ell_2^r \\ \ell_3^r \end{Bmatrix} x_i^r \qquad (14.65\text{–}4)$$

where: θ_i = angle between the projection of the x_i^r axis on XY plane and the X axis (input quantities THETA1, THETA2, and THETA3 on **R** command)

ϕ_i = angle between the x_i^r axis and the XY plane (input quantities PHI1, PHI2, and PHI3 on **R** command)

ℓ_i^r = direction cosines between x_i^r axis and element X, Y, Z axes

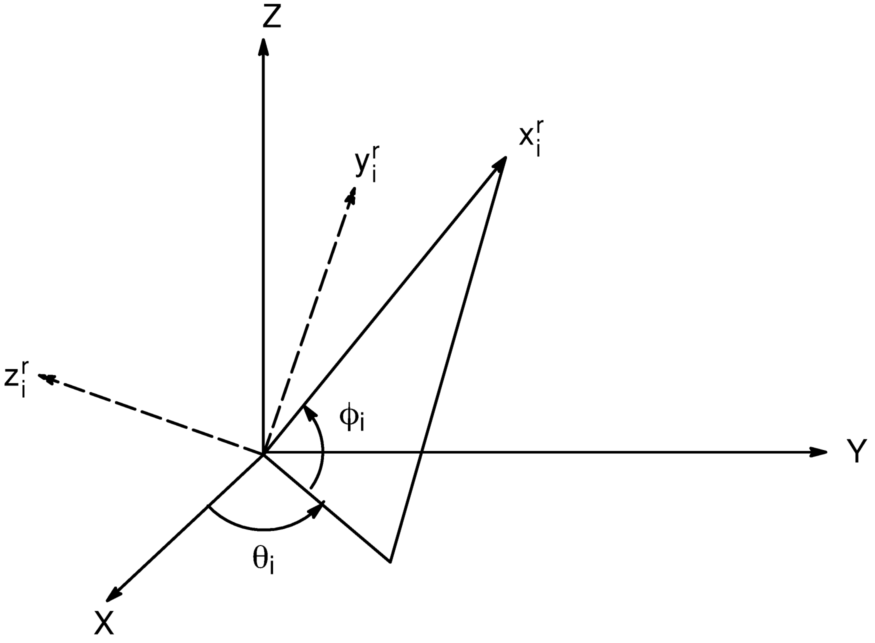


Figure 14.65–1 Reinforcement Orientation

Since the reinforcement material matrix is defined in coordinates aligned in the direction of reinforcement orientation, it is necessary to construct a transformation of the form

$$\left[D^R\right]_i = [T^r]^T[D^r]_i[T^r] \tag{14.65–5}$$

in order to express the material behavior of the reinforcement in global coordinates. The form of this transformation by Schnobrich(29) is

$$[T^r] = \begin{bmatrix} a_{11}^2 & a_{12}^2 & a_{13}^2 & a_{11}a_{12} & a_{12}a_{13} & a_{11}a_{13} \\ a_{21}^2 & a_{22}^2 & a_{23}^2 & a_{21}a_{22} & a_{22}a_{23} & a_{21}a_{23} \\ a_{31}^2 & a_{32}^2 & a_{33}^2 & a_{31}a_{32} & a_{32}a_{33} & a_{31}a_{33} \\ 2a_{11}a_{21} & 2a_{12}a_{22} & 2a_{13}a_{23} & \begin{array}{l} a_{11}a_{22}+ \\ a_{12}a_{21} \end{array} & \begin{array}{l} a_{12}a_{23}+ \\ a_{13}a_{32} \end{array} & \begin{array}{l} a_{11}a_{23}+ \\ a_{13}a_{21} \end{array} \\ 2a_{21}a_{31} & 2a_{22}a_{32} & 2a_{23}a_{33} & \begin{array}{l} a_{21}a_{32}+ \\ a_{22}a_{31} \end{array} & \begin{array}{l} a_{22}a_{33}+ \\ a_{23}a_{32} \end{array} & \begin{array}{l} a_{21}a_{33}+ \\ a_{13}a_{21} \end{array} \\ 2a_{11}a_{31} & 2a_{12}a_{32} & 2a_{13}a_{33} & \begin{array}{l} a_{11}a_{32}+ \\ a_{12}a_{31} \end{array} & \begin{array}{l} a_{12}a_{33}+ \\ a_{13}a_{32} \end{array} & \begin{array}{l} a_{11}a_{33}+ \\ a_{13}a_{31} \end{array} \end{bmatrix} \tag{14.65–6}$$

where the coefficients a_{ij} are defined as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \ell_1^r & \ell_2^r & \ell_3^r \\ m_1^r & m_2^r & m_3^r \\ n_1^r & n_2^r & n_3^r \end{bmatrix} \tag{14.65–7}$$

The vector $\lfloor \ell_1^r \quad \ell_2^r \quad \ell_3^r \rfloor^T$ is defined by equation (14.65–4) while $\lfloor m_1^r \quad m_2^r \quad m_3^r \rfloor^T$ and $\lfloor n_1^r \quad n_2^r \quad n_3^r \rfloor^T$ are unit vectors mutually orthogonal to $\lfloor \ell_1^r \quad \ell_2^r \quad \ell_3^r \rfloor^T$ thus defining a Cartesian coordinate referring to reinforcement directions. If the operations presented by equation (14.65–5) are performed substituting equation (14.65–3) and equation (14.65–6), the resulting reinforcement material matrix in element coordinates takes the form

$$[D^r]_i = E_i^r \ \{A_d\} \ \{A_d\}^T \tag{14.65–8}$$

where: $\{A_d\} = \lfloor a_{11}^2 \quad a_{21}^2 \ ... \ a_{11} \ a_{13} \rfloor^T$

Therefore, the only direction cosines used in $[D^R]_i$ involve the uniquely defined unit vector $\lfloor \ell_1^r \quad \ell_2^r \quad \ell_3^r \rfloor^T$.

14.65.6 Nonlinear Behavior – Concrete

As mentioned previously, the matrix material (e.g. concrete) is capable of plasticity, creep, cracking and crushing. The plasticity and creep formulations are the same as those implemented in SOLID45 (see Section 4.1). The concrete material model with its cracking and crushing capabilities is discussed in Section 4.7. This material model predicts either elastic behavior, cracking behavior or crushing behavior. If elastic behavior is predicted, the concrete is treated as a linear elastic material (discussed above). If cracking or crushing behavior is predicted, the elastic, stress–strain matrix is adjusted as discussed below for each failure mode.

14.65.7 Modeling of a Crack

The presence of a crack at an integration point is represented through modification of the stress–strain relations by introducing a plane of weakness in a direction normal to the crack face. Also, a shear transfer coefficient β_t (constant C_1 on the **TBDATA** command with **TB**,CONCR) is introduced which represents a shear strength reduction factor for those subsequent loads which induce sliding (shear) across the crack face. The stress–strain relations for a material that has cracked in one direction only become:

$$\left[D_c^{ck}\right] = \frac{E}{(1+\nu)} \begin{bmatrix} \frac{R^t(1+\nu)}{E} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ 0 & \frac{\nu}{1-\nu} & \frac{1}{1-\nu} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\beta_t}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\beta_t}{2} \end{bmatrix} \qquad (14.65\text{–}9)$$

where the superscript ck signifies that the stress strain relations refer to a coordinate system parallel to principal stress directions with the x^{ck} axis perpendicular to the crack face. If KEYOPT(7) = 0, R^t = 0.0. If KEYOPT(7) = 1, R^t is the slope (secant modulus) as defined in Figure 14.65–2. R^t works with adaptive descent and diminishes to 0.0 as the solution converges

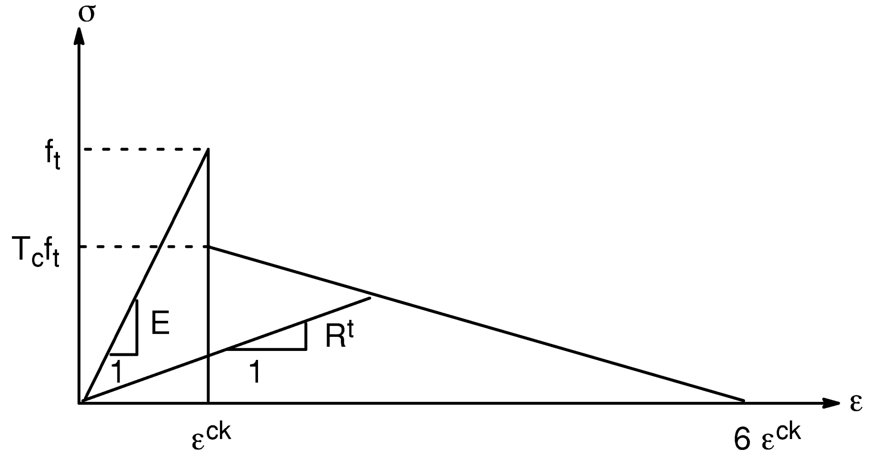


Figure 14.65–2 Strength of Cracked Condition

In Figure 14.65–2,

f_t = unaxial tensile cracking stress (input as C_3 on the **TBDATA** command)

T_c = multiplier for amount of tensile stress relaxation (input as C_9 on the **TBDATA** command, defaults to 0.6)

If the crack closes, then all compressive stresses normal to the crack plane are transmitted across the crack and only a shear transfer coefficient β_c (constant C_2 on the

TBDATA command with **TB**,CONCR) for a closed crack is introduced. Then $\left[D_c^{ck}\right]$ can be expressed as

$$\left[D_c^{ck}\right] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_c\frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_c\frac{(1-2\nu)}{2} \end{bmatrix} \qquad (14.65\text{–}10)$$

The stress–strain relations for concrete that has cracked in two directions are:

$$\left[D_c^{ck}\right] = E\begin{bmatrix} \frac{R^t}{E} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{R^t}{E} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} \end{bmatrix} \qquad (14.65\text{–}11)$$

If both directions reclose,

$$\left[D_c^{ck}\right] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_c\frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_c\frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_c\frac{(1-2\nu)}{2} \end{bmatrix} \qquad (14.65\text{–}12)$$

The stress–strain relations for concrete that has cracked in all three directions are:

$$\left[D_c^{ck}\right] = E\begin{bmatrix} \frac{R^t}{E} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{R^t}{E} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{R^t}{E} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\beta_t}{2(1+\nu)} \end{bmatrix} \qquad (14.65\text{–}13)$$

If all three cracks reclose, equation (14.65–12) is followed. In total there are 16 possible combinations of crack arrangement and appropriate changes in stress–strain relationships incorporated in SOLID65. A note is output if $1 > \beta_c > \beta_t > 0$ are not true.

The transformation of $\left[D_c^{ck}\right]$ to element coordinates has the form

$$[D_c] = \left[T^{ck}\right]^T\left[D_c^{ck}\right]\left[T^{ck}\right] \qquad (14.65\text{–}14)$$

where $[T^{ck}]$ has a form identical to equation (14.65–6) and the three columns of [A] in equation (14.65–7) are now the principal direction vectors.

The open or closed status of integration point cracking is based on a strain value ϵ^{ck}_{ck} called the crack strain. For the case of a possible crack in the x direction, this strain is evaluated as

$$\epsilon^{ck}_{ck} = \begin{cases} \epsilon^{ck}_{x} + \dfrac{\nu}{1-\nu}\,(\epsilon^{ck}_{y} + \epsilon^{ck}_{z}) & \text{if no cracking has occurred} \\ \epsilon^{ck}_{x} + \nu\,\epsilon^{ck}_{z} & \text{if y direction has cracked} \\ \epsilon^{ck}_{x} & \text{if y and z direction have cracked} \end{cases} \tag{14.65–15}$$

where:

ϵ^{ck}_{x}, ϵ^{ck}_{y}, and ϵ^{ck}_{z} = three normal component strains in crack orientation.

The vector $\{\varepsilon^{ck}\}$ is computed by:

$$\left\{\epsilon^{ck}\right\} = \left[T^{ck}\right]\left\{\epsilon'\right\} \tag{14.65–16}$$

where: $\{\varepsilon'\}$ = modified total strain (in element coordinates)

$\{\varepsilon'\}$, in turn, is defined as:

$$\left\{\epsilon'_{n}\right\} = \left\{\epsilon^{el}_{n-1}\right\} + \left\{\Delta\epsilon_{n}\right\} - \left\{\Delta\epsilon^{th}_{n}\right\} - \left\{\Delta\epsilon^{pl}_{n}\right\} \tag{14.65–17}$$

where:

n = substep number

$\left\{\epsilon^{el}_{n-1}\right\}$ = elastic strain from previous substep

$\{\Delta\varepsilon_n\}$ = total strain increment (based on $\{\Delta u_n\}$, the displacement increment over the substep)

$\left\{\Delta\epsilon^{th}_{n}\right\}$ = thermal strain increment

$\left\{\Delta\epsilon^{pl}_{n}\right\}$ = plastic strain increment

If ϵ^{ck}_{ck} is less than zero, the associated crack is assumed to be closed.

If ϵ^{ck}_{ck} is greater than or equal to zero, the associated crack is assumed to be open. When cracking first occurs at an integration point, the crack is assumed to be open for the next iteration.

14.65.8 Modeling of Crushing

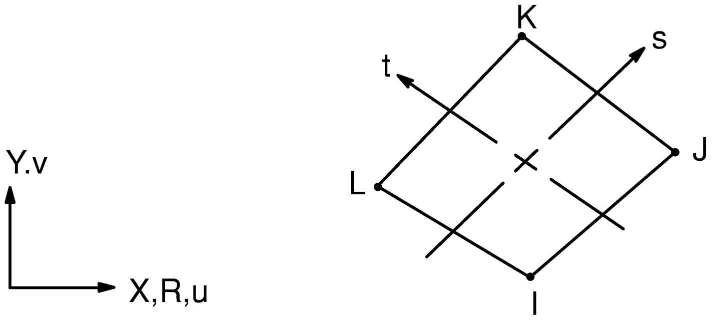
If the material at an integration point fails in uniaxial, biaxial, or triaxial compression, the material is assumed to crush at that point. In SOLID65, crushing is defined as the complete deterioration of the structural integrity of the material (e.g. material spalling).

Under conditions where crushing has occurred, material strength is assumed to have degraded to an extent such that the contribution to the stiffness of an element at the integration point in question can be ignored.

14.65.9 Nonlinear Behavior – Reinforcement

The one–dimensional creep and plasticity behavior for SOLID65 reinforcement is modeled in the same manner as for LINK8.

14.67 PLANE67 — 2–D Coupled Thermal–Electric Solid



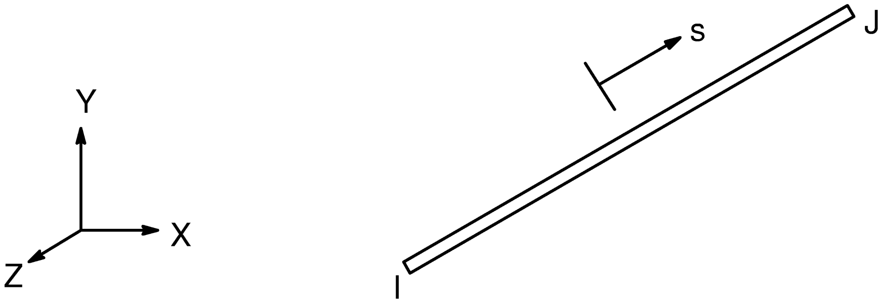
Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Quad	Equations (12.6.5–20) and (12.6.5–21)	2 x 2
	Triangle	Equations (12.6.1–20) and (12.6.1–21)	3
Specific Heat Matrix	Same as for thermal conductivity matrix. Matrix is diagonalized as described in Section 13.2.		Same as conductivity matrix
Heat Generation Load Vector	Same as specific heat matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as specific heat matrix evaluated at the face		2

Reference: Kohnke and Swanson(19)

14.67.1 Other Applicable Sections

Section 11.2 discusses thermal–electric elements. Section 13.1 describes integration point locations.

14.68 LINK68 — Coupled Thermal–Electric Line



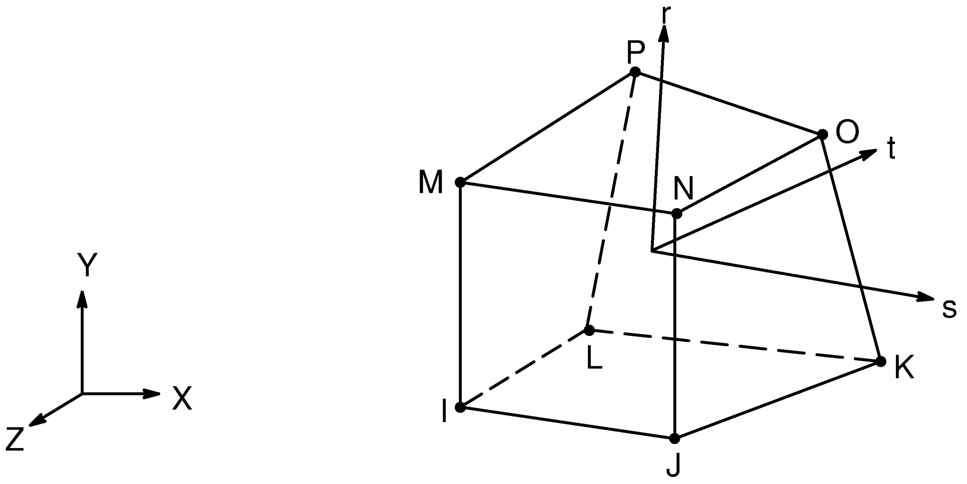
Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrices	Equations (12.2.1–20) and (12.2.1–21)	None
Specific Heat Matrix	Equation (12.2.1–20)	None
Heat Generation Load Vector	Equation (12.2.1–20)	None

Reference: Kohnke and Swanson(19)

14.68.1 Other Applicable Sections

Section 11.2 discusses thermal–electric elements.

14.69 SOLID69 — 3–D Coupled Thermal–Electric Solid



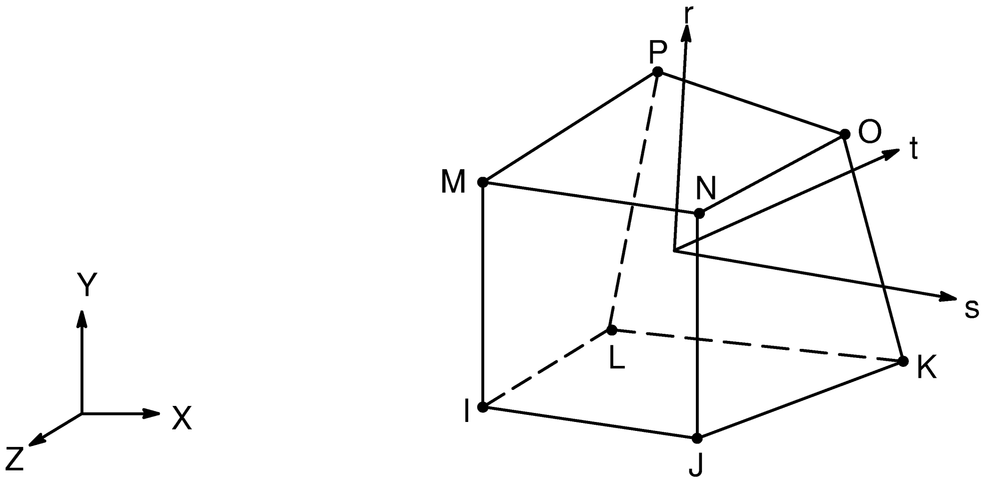
Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equations (12.8.18–20) and (12.8.18–21)	2 x 2 x 2
Specific Heat Matrix	Equation (12.8.18–20). Matrix is diagonalized as described in Section 13.2	2 x 2 x 2
Heat Generation Load Vector	Equation (12.8.18–20)	2 x 2 x 2
Convection Surface Matrix and Load Vector	Equation (12.8.18–20), specialized to the face	None

Reference: Kohnke and Swanson(19)

14.69.1 Other Applicable Sections

Section 11.2 discusses thermal–electric elements. Section 13.1 describes integration point locations.

14.70 SOLID70 — 3–D Thermal Solid



Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equation (12.8.18–20)	2 x 2 x 2
Specific Heat Matrix	Equation (12.8.18–20). Matrix is diagonalized as described in Section 13.2	Same as conductivity matrix
Heat Generation Load Vector	Equation (12.8.18–20)	2 x 2 x 2
Convection Surface Matrix and Load Vector	Equation (12.8.18–20) specialized to the face	2 x 2

14.70.1 Other Applicable Sections

Section 6.2 has a complete derivation of the matrices and load vectors of a general thermal analysis element. Section 13.1 describes integration point locations. Mass transport is discussed in Section 14.55.

14.70.2 Fluid Flow in a Porous Medium

An option (KEYOPT(7) = 1) is available to convert SOLID70 to a nonlinear steady–state fluid flow element. Pressure is the variable rather than temperature. From equation (6.2–7), the element conductivity matrix is:

$$\left[K_e^{tb}\right] = \int_{vol} [B]^T [D] \, [B] \; d(vol) \qquad (14.70\text{–}1)$$

[B] is defined by equation (6.2–7) and for this option, [D] is defined as:

$$[D] = \begin{bmatrix} \dfrac{K_x^{\infty} \; \rho}{\mu + K_x^{\infty} \; E} & 0 & 0 \\ 0 & \dfrac{K_y^{\infty} \; \rho}{\mu + K_y^{\infty} \; E} & 0 \\ 0 & 0 & \dfrac{K_z^{\infty} \; \rho}{\mu + K_z^{\infty} \; E} \end{bmatrix}$$

where: K_x^{∞} = absolute permeability of the porous medium in the x direction (input as KXX on **MP** command)

ϱ = mass density of the fluid (input as DENS on **MP** command)

μ = viscosity of the fluid (input quantity VISC on **MP** command)

E = $\rho \; \beta \; S^{\alpha}$

β = visco–inertial parameter of the fluid (input quantity C on **MP** command)

S = seepage velocity (at centroid from previous iteration, defined below)

α = input as MU on **MP** command

For this option, no “specific heat” matrix or “heat generation” load vector is computed.

The pressure gradient components are computed by:

$$\begin{Bmatrix} g_x^p \\ g_y^p \\ g_z^p \end{Bmatrix} = [B] \; \{T_e\} \qquad (14.70\text{–}2)$$

where: g_x^p = output quantity PRESSURE GRADIENT (X)

$\{T_e\}$ = vector of element temperatures (pressures)

The pressure gradient is computed from:

$$g^p = \sqrt{\left(g_x^p\right)^2 + \left(g_y^p\right)^2 + \left(g_z^p\right)^2} \qquad (14.70\text{–}3)$$

where: g^p = output quantity PRESSURE GRADIENT (TOTAL)

The mass flux components are:

$$\begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix} = -\,[D] \begin{Bmatrix} g_x^p \\ g_y^p \\ g_z^p \end{Bmatrix} \qquad (14.70\text{–}4)$$

The vector sum of the mass flux components is:

$$f = \sqrt{f_x^2 + f_y^2 + f_z^2} \qquad (14.70\text{–}5)$$

where: f = output quantity MASS FLUX

The fluid velocity components are:

$$\begin{Bmatrix} S_x \\ S_y \\ S_z \end{Bmatrix} = \frac{1}{\rho} \begin{Bmatrix} f_x \\ f_y \\ f_z \end{Bmatrix} \qquad (14.70\text{–}6)$$

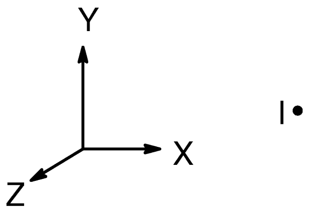
where: S_x = output quantity FLUID VELOCITY (X)

and the maximum fluid velocity is:

$$S = \frac{f}{\rho} \qquad (14.70\text{–}7)$$

where: S = output quantity FLUID VELOCITY (TOTAL)

14.71 MASS71 — Thermal Mass



Matrix Or Vector	Shape Functions	Integration Points
Specific Heat Matrix	None	None
Heat Generation Load Vector	None	None

14.71.1 Specific Heat Matrix

The specific heat matrix for this element is simply:

$$\left[C_e^t\right] = \left[C^o\right] \tag{14.71–1}$$

C^o is defined as:

$$C^o = \begin{cases} \varrho C_p\,(\text{vol}) & \text{if KEYOPT(3) = 0} \\ C_a & \text{if KEYOPT(3) = 1} \end{cases} \tag{14.71–2}$$

where:

ϱ = density (input as DENS on **MP** command)
C_p = specific heat (input as C on **MP** command)
vol = volume (input as CONI on **R** command)
C_a = capacitance (input as CONI on **R** command)

14.71.2 Heat Generation Load Vector

The heat generation load vector is:

$$\{Q_e^g\} = \left\{A_q\right\} \tag{14.71–3}$$

where: A_q =

$$\begin{cases} Q_R & \text{if } A_1 \text{ thru } A_6 \text{ are not provided} \\ \text{or} & \\ A_1 + A_2T + A_3T^{A_4} + A_5T^{A_6} & \text{if } A_1 \text{ thru } A_6 \text{ are provided} \end{cases}$$

Q_R = heat rate (input quantity QRATE on **MP** command)

A_1, A_2, etc. = input quantities A1, A2, etc. on **R** command

T = $T_\ell + T_o$ = absolute temperature

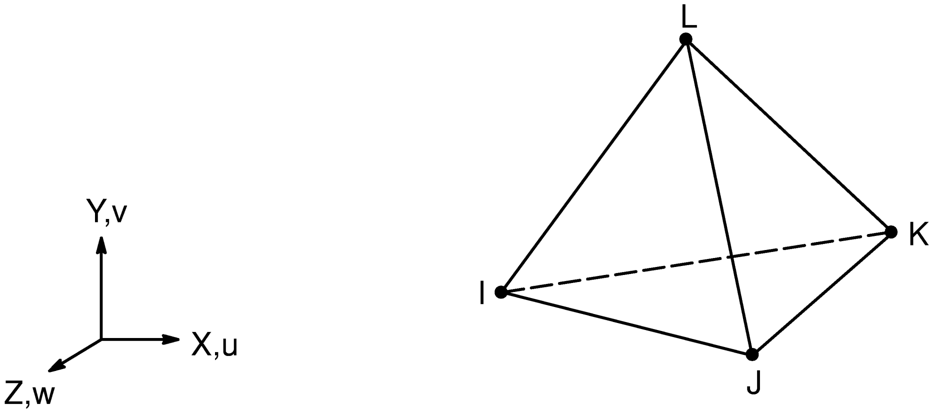
$$T_\ell = \begin{cases} T_{unif} \text{ for first iteration} \\ T_\ell' \text{ for second and subsequent iterations} \end{cases}$$

T_{unif} = input on **BFUNIF** command

T_ℓ' = temperature from previous iteration

T_o = offset temperature (input on **TOFFST** command)

14.72 SOLID72 — 4–Node Tetrahedral Structural Solid with Rotations



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.3–1) thru (12.8.3–3)	5 (direct stress) 1 (shear stress)
Mass Matrix	Same as stiffness matrix	5
Stress Stiffness Matrix	Same as stiffness matrix	5
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Equation (12.8.3–1) thru (12.8.3–3) specialized to the face	3

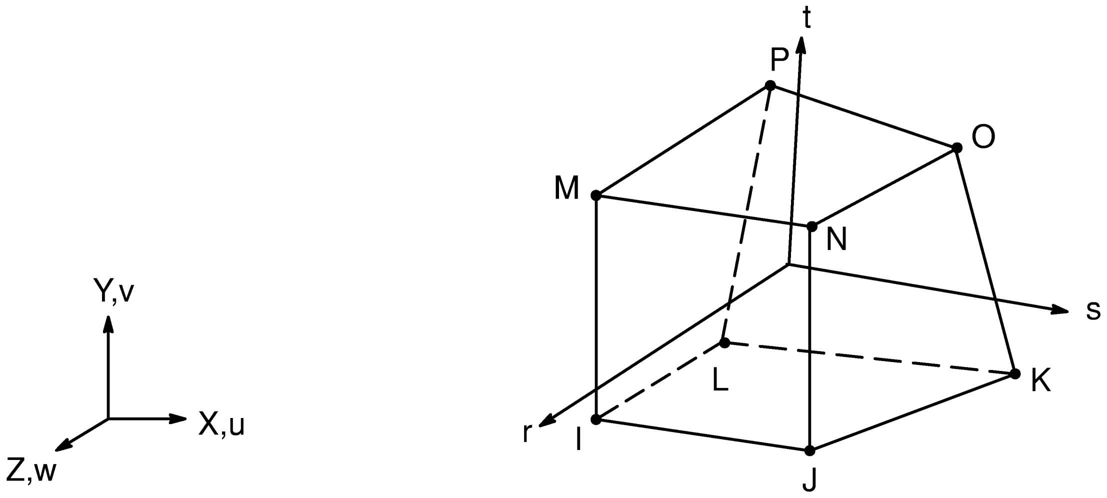
Load Type	Distribution
Element Temperature	Equation (12.8.1–20)
Nodal Temperature	Equation (12.8.1–20)
Pressure	Bilinear across each face

Reference: Yunus et al.(117)

14.72.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.73 SOLID73 — 3–D 8–Node Structural Solid with Rotations



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Brick	Equations (12.8.21–1) thru (12.8.21–3)	2 x 2 x 2
	Wedge	Equation (12.8.15–1) thru (12.8.15–3)	3 x 2
	Pyramid	Equations (12.8.8–1) thru (12.8.8–3)	2 x 2 x 2
	Tet	Equations (12.8.3–1) thru (12.8.3–3)	5 (direct stress) 1 (shear stress)
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Stress Stiffness Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Pressure Load Vector	Quad	Uses shape functions for stiffness matrices, specialized to the face	2 x 2
	Triangle		3

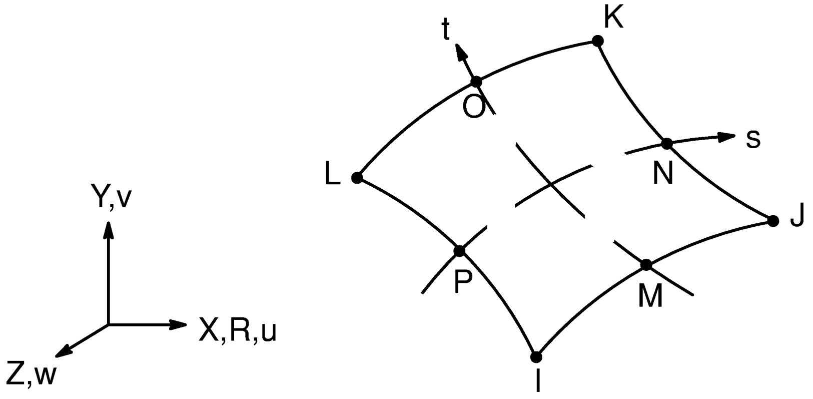
Load Type	Geometry	Distribution
Element and Nodal Temperatures	Brick	Equation (12.8.18–20)
	Wedge	Equation (12.8.11–20)
	Pyramid	Equation (12.8.6–20)
	Tet	Equation (12.8.1–20)
Pressure	Bilinear across each face	

Reference: Yunus et al.(117)

14.73.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.74 HYPER74 — 2–D 8–Node Mixed U–P Hyperelastic Solid



Matrix Or Vector	Geometry	Shape Functions	Integration Points
Stiffnesss Matrix	Quad	Equations (12.6.7–1), (12.6.7–2), and (12.6.7–3)	3 x 3
	Triangle	Equations (12.6.2–1), (12.6.2–2), and (12.6.2–3)	3
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal and Newton–Raphson Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

References: Oden(123), Sussman(124)

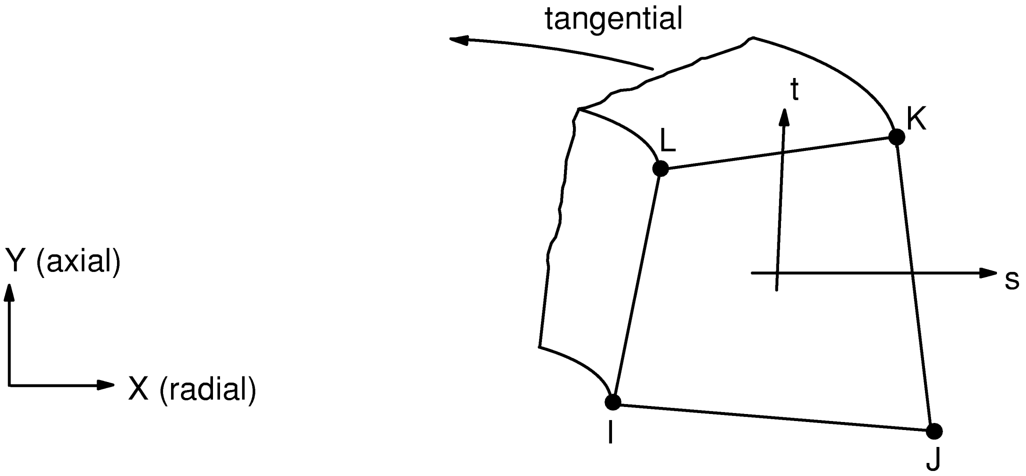
14.74.1 Other Applicable Sections

For the basic formulation refer to Section 14.58. The hyperelastic material model (Mooney–Rivlin) is described in Section 4.5. Section 13.1 describes integration point locations.

14.74.2 Assumptions and Restrictions

A dropped midside node implies that the edge is and remains straight.

14.75 PLANE75 — Axisymmetric–Harmonic Thermal Solid

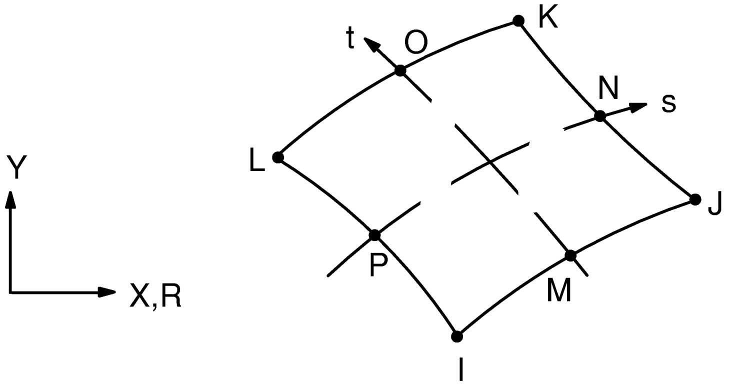


Matrix Or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Quad	Equation (12.7.5–20)	2 x 2
	Triangle	Equation (12.7.1–20)	3
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Section 13.2		Same as conductivity matrix
Heat Generation Load Vector	Same as conductivity matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix specialized to the face		2

14.75.1 Other Applicable Sections

Chapter 6 describes the derivation of the element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations.

14.77 PLANE77 — 2–D 8–Node Thermal Solid



Matrix Or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Quad	Equation (12.6.7–20)	3 x 3
	Triangle	Equation (12.6.2–20)	6
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, matrix is diagonalized as described in Section 13.2		Same as conductivity matrix
Heat Generation Load Vector	Same as conductivity matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as conductivity matrix, specialized to the face		2

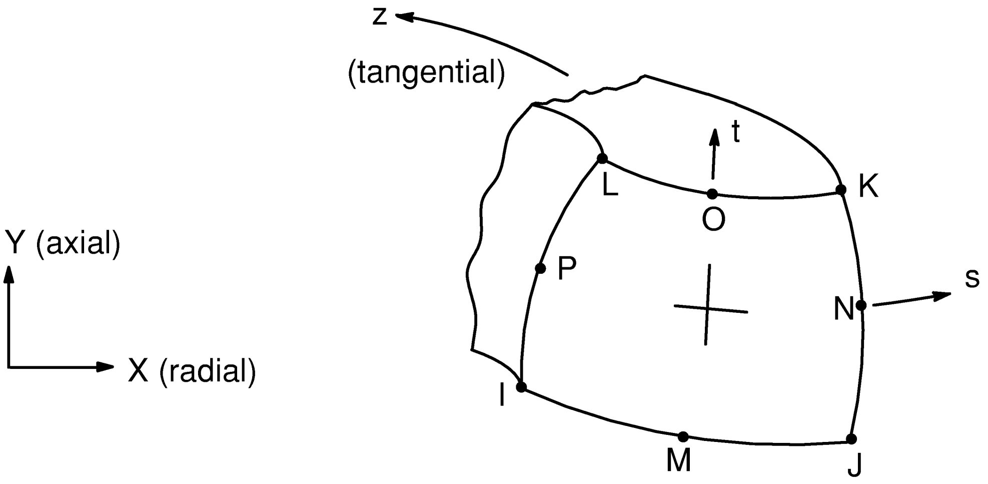
14.77.1 Other Applicable Sections

Chapter 6 describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations. If KEYOPT(1) = 1, the specific heat matrix is diagonalized as described in Section 13.2.

14.77.2 Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the temperature varies linearly along that edge.

14.78 PLANE78 — Axisymmetric–Harmonic 8–Node Thermal Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Quad	Equation (12.7.7–20)	3 x 3
	Triangle	Equation (12.7.2–20)	6
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, matrix is diagonalized as described in Section 13.2		Same as conductivity matrix
Heat Generation Load Vector	Same as conductivity matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Same as stiffness matrix, specialized to the face		2

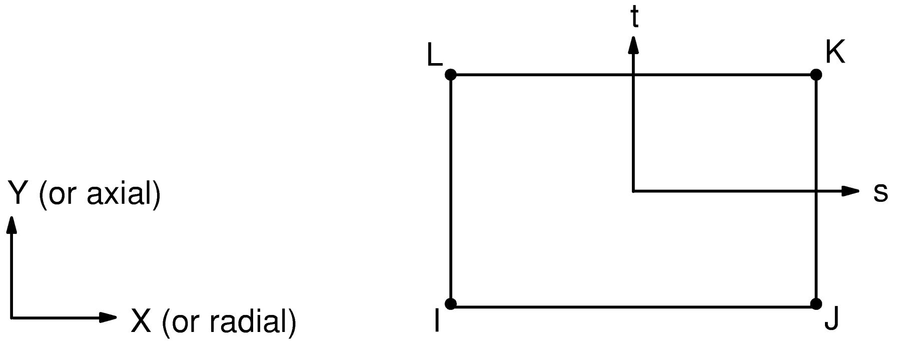
14.78.1 Other Applicable Sections

Chapter 6 describes the derivation of the thermal element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations.

14.78.2 Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the temperature varies linearly along that edge.

14.79 FLUID79 — 2–D Contained Fluid



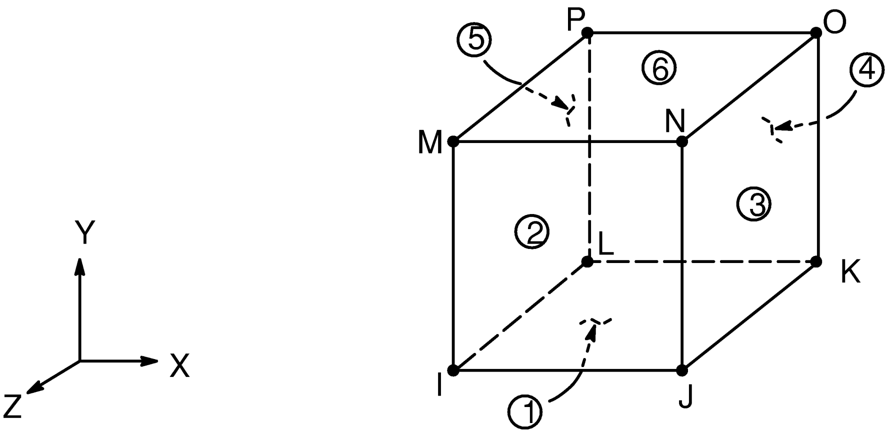
Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffnesss Matrix	Quad	Equations (12.6.5–1) and (12.6.5–2)	1 x 1 for bulk strain effects and 2 x 2 for shear and rotational resistance effects
	Triangle	Equations (12.6.1–1) and (12.6.1–2)	1 x 1 for bulk strain effects and 3 for shear and rotational resistance effects
Mass Matrix	Same as stiffness matrix. Matrix is diagonalized as in Section 13.2.		Same as for shear effects
Damping Matrix	Same as stiffness matrix.		Same as for shear effects
Temperature Load Vector	Same as stiffness matrix		1 x 1
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2

Load Type	Distribution
Element Temperature	Average of the four nodal temperatures is used throughout the element
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

14.79.1 Other Applicable Sections

Chapter 2 describes the derivation of element matrices and load vectors. The fluid aspects of this element are the same as described for FLUID80 (Section 14.80). Section 13.1 describes integration point locations.

14.80 FLUID80 — 3–D Contained Fluid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.8.18–1), (12.8.18–2), and (12.8.18–3)	1 x 1 x 1 for bulk strain effects. 2 x 2 x 2 for shear and rotational resistance effects
Mass Matrix	Same as stiffness matrix. Matrix is diagonalized as described in Section 13.2	2 x 2 x 2
Damping Matrix	Same as stiffness matrix	2 x 2 x 2
Temperature Load Vector	Same as stiffness matrix	1 x 1 x 1
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2 x 2

Load Type	Distribution
Element Temperature	Average of the 8 nodal temperatures is used throughout element
Nodal Temperature	Average of the 8 nodal temperatures is used throughout element
Pressure	Bilinear across each face

14.80.1 Other Applicable Sections

Chapter 2 describes the derivation of element matrices and load vectors. Section 13.1 describes integration point locations.

14.80.2 Assumptions and Restrictions

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.80.3 Material Properties

Rather than equation (2.1–3), the stress–strain relationships used to develop the stiffness matrix and thermal load vector are:

$$\begin{Bmatrix} \epsilon_{bulk} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \\ R_x \\ R_y \\ R_z \end{Bmatrix} = \begin{Bmatrix} 3\alpha\Delta T \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} + \begin{bmatrix} \frac{1}{K} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{S} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{S} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{S} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{B} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{B} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{B} \end{bmatrix} \begin{Bmatrix} P \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \\ M_x \\ M_y \\ M_z \end{Bmatrix} \quad (14.80\text{–}1)$$

where: ε_{bulk} = bulk strain = $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}$

α = thermal coefficient of expansion (input as ALPX on **MP** command)

ΔT = change of temperature from reference temperature

K = fluid elastic (bulk) modulus (input quantity EX on **MP** command)

P = pressure

γ = shear strain

S = K x 10^{-9} (arbitrarily small number to give element some shear stability)

τ = shear stress

R_i = rotation about axis i

B = K x 10^{-9} (arbitrarily small number to give element some rotational stability)

M_i = twisting force about axis i

A damping matrix is also developed based on:

$$\begin{Bmatrix} \dot{\epsilon}_{bulk} \\ \dot{\gamma}_{xy} \\ \dot{\gamma}_{yz} \\ \dot{\gamma}_{xz} \\ \dot{R}_x \\ \dot{R}_y \\ \dot{R}_z \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\eta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\eta} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\eta} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{c} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{c} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{c} \end{bmatrix} \begin{Bmatrix} P \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \\ M_x \\ M_y \\ M_z \end{Bmatrix} \qquad (14.80\text{–}2)$$

where: η = viscosity (input as VISC on **MP** command)

c = .00001*η

and the ($\cdot$) represents differentiation with respect to time.

A lumped mass matrix is developed, based on the density (input as DENS on **MP** command).

14.80.4 Free Surface Effects

The free surface is handled with an additional special spring effect. The necessity of these springs can be seen by studying a U–Tube, as shown in Figure 14.80–1.

Note that if the left side is pushed down a distance of Δh, the displaced fluid mass is:

$$M_D = \Delta h \; A \; \rho \tag{14.80–3}$$

where:

M_D = mass of displaced fluid

Δh = distance fluid surface has moved

A = cross–sectional area of U–Tube

ϱ = fluid density

Then, the force required to hold the fluid in place is

$$F_D = M_D \; g \tag{14.80–4}$$

where:

F_D = force required to hold the fluid in place

g = acceleration due to gravity (input on **ACEL** command)

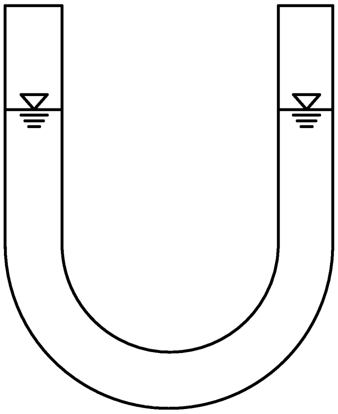


Figure 14.80–1 U–Tube with Fluid

Finally, the stiffness at the surface is the force divided by the distance, or

$$K_s = \frac{F_D}{\Delta h} = \rho A g \tag{14.80–5}$$

This expression is generalized to be:

$$K_s = \rho A_F \left(g_x \; C_x + g_y \; C_y + g_z \; C_z \right) \tag{14.80–6}$$

where:

A_F = area of the face of the element

g_i = acceleration in the i direction

C_i = ith component of the normal to the face of the element

This results in adding springs from each node to ground, with the spring constants being positive on the top of the element, and negative on the bottom. For an interior node, positive and negative effects cancel out and, at the bottom where the boundary must be fixed to keep the fluid from leaking out, the negative spring has no effect. If KEYOPT(2)=1, positive springs are added only to faces located at z = 0.0.

14.80.5 Other Assumptions and Limitations

The surface springs tend to retard the hydrostatic motions of the element from their correct values. The hydrodynamic motions are not changed. From the definition of bulk modulus,

$$u_s = \int_o^H \frac{P}{K}\, dz \qquad (14.80\text{–}7)$$

where:

u_s = vertical motion of a static column of fluid (unit cross–sectional area)

H = height of fluid column

P = pressure at any point

z = distance from free surface

The pressure is normally defined as:

$$P = \rho g z \qquad (14.80\text{–}8)$$

But this pressure effect is reduced by the presence of the surface springs, so that

$$P = \rho g z - K_s u_s = \rho g\,(z - u_s) \qquad (14.80\text{–}9)$$

Combining equations (14.80–7) and (14.80–9) and integrating,

$$u_s = \frac{\rho g}{K}\left(\frac{H^2}{2} - u_s H\right) \qquad (14.80\text{–}10)$$

or

$$u_s = \frac{1}{1 - \dfrac{H\rho g}{K}}\,\frac{\rho g}{K}\,\frac{H^2}{2} \qquad (14.80\text{–}11)$$

If there were no surface springs,

$$u_s = \frac{\rho g}{K}\,\frac{H^2}{2} \qquad (14.80\text{–}12)$$

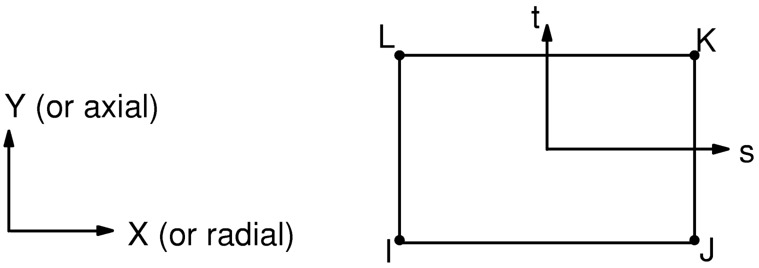
Thus the error for hydrostatic effects is the departure from 1.0 of the factor (1 / (1–Hρg/K)), which is normally quite small.

The 1 x 1 x 1 integration rule is used to permit the element to “bend” without the bulk modulus resistance being mobilized, i.e.

Figure 14.80–2   “Bending Without Resistance”

While this motion is permitted, other motions in a static problem often result, which can be thought of as energy–free eddy currents. For this reason, small shear and rotational resistances are built in, as indicated in equation (14.80–1).

14.81 FLUID81 — Axisymmetric–Harmonic Contained Fluid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffnesss Matrix	Quad	Equations (12.7.5–1), (12.7.5–2), and (12.7.5–3)	1 for bulk strain effects and 2 x 2 for shear and rotational resistance effects
	Triangle	Equations (12.7.1–1), (12.7.1–2), and (12.7.1–3)	1 for bulk strain effects and 3 for shear and rotational resistance effects
Mass Matrix	Quad	Equations (12.6.5–1), (12.6.5–2), and (12.6.5–3)	2 x 2
	Triangle	Equations (12.6.1–1), (12.6.1–2), and (12.6.1–3)	3
Damping Matrix	Same as stiffness matrix		Same as stiffness matrix
Temperature Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2

LOAD TYPE	DISTRIBUTION
Element Temperature	Average of the four nodal temperatures is used throughout the element
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

14.81.1 Other Applicable Sections

Chapter 2 describes the derivation of element matrices and load vectors. The fluid aspects of this element are the same as described for FLUID80 (Section 14.80) except that a consistent mass matrix is also available (**LUMPM**,OFF). Section 13.1 describes integration point locations.

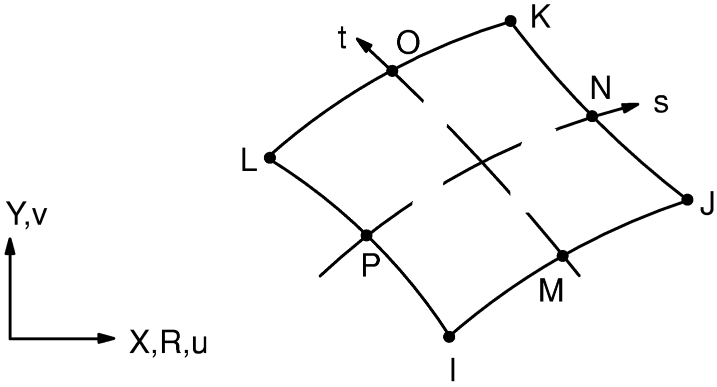
14.81.2 Assumptions and Restrictions

The material properties are assumed to be constant around the entire circumference, regardless of temperature dependent material properties or loading.

14.81.3 Load Vector Correction

When MODE equals 1 or greater (input on **MODE** command), the gravity that is required to be input for use as a gravity spring (input as ACELY on **ACEL** command) also is erroneously multiplied by the mass matrix for a gravity force effect. This erroneous effect is cancelled out by an element load vector that is automatically generated during the element stiffness pass.

14.82 PLANE82 — 2–D 8–Node Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffnesss Matrix	Quad	Equations (12.6.7–1) and (12.6.7–2)	2 x 2
	Triangle	Equations (12.6.2–1) and (12.6.2–2)	3
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Stress Stiffness Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2 along face

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

Reference: Zienkiewicz(39)

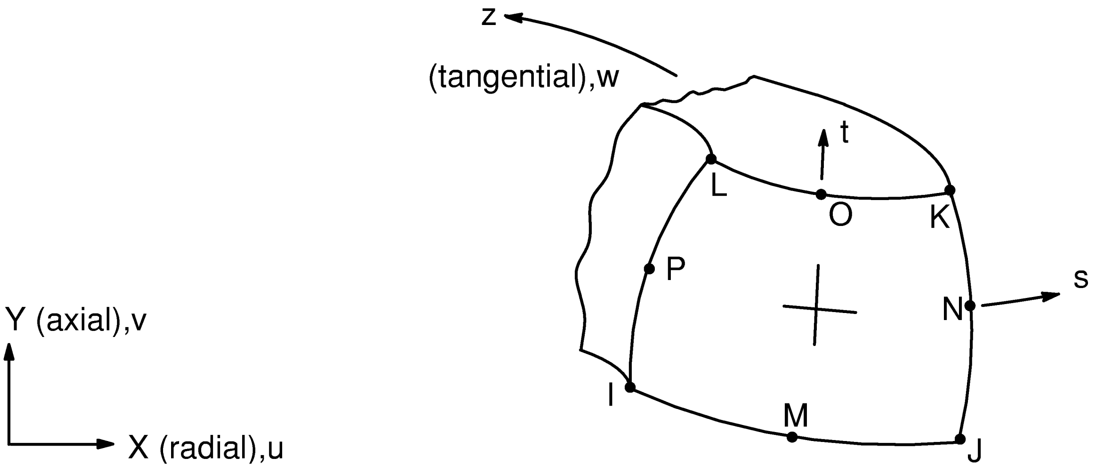
14.82.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.82.2 Assumptions and Restrictions

A dropped midside node implies that the face is and remains straight.

14.83 PLANE83 — 8–Node Axisymmetric–Harmonic Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.7.7–1), (12.7.7–2), and (12.7.7–3)	2 x 2
	Triangle	Equations (12.7.2–1), (12.7.2–2), and (12.7.2–3)	3
Mass Matrix	Quad	Equations (12.6.7–1), (12.6.7–2), and (12.6.7–3)	2 x 2
	Triangle	Equations (12.6.2–1), (12.6.2–2), and (12.6.2–3)	3
Stress Stiffness Matrix	Same as stiffness matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2

Load Type	Distribution
Element Temperature	Same as shape functions across element, harmonic around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face, harmonic around circumference

Reference: Zienkiewicz(39)

14.83.1 Other Applicable Sections

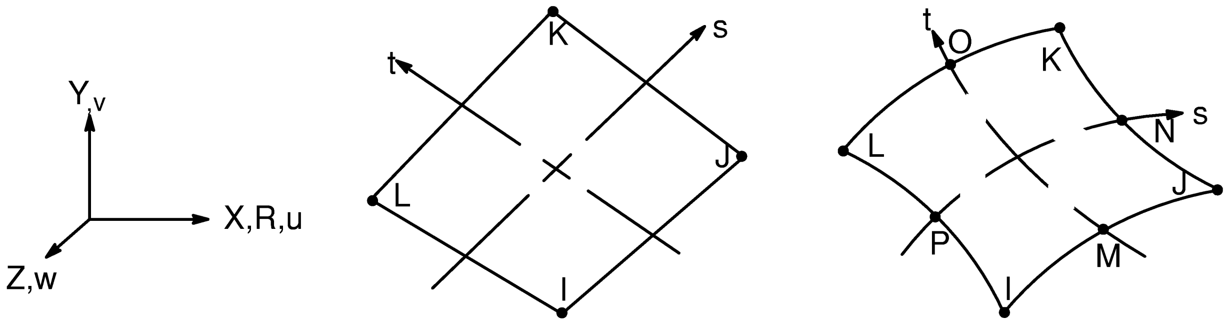
Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. Section 14.25 has a discussion of temperature applicable to this element.

14.83.2 Assumptions and Restrictions

A dropped midside node implies that the edge is and remains straight.

The material properties are assumed to be constant around the entire circumference, regardless of temperature–dependent material properties or loading. For MODE > 0, extreme values for combined stresses are obtained by computing these stresses at every 10/MODE degrees and selecting the extreme values.

14.84 HYPER84 — 2–D 8–Node Hyperelastic Solid



Matrix or Vector	Geometry		Shape Functions	Integration Points
Stiffness Matrix	4 node (KEYOPT(1)=0)	Quad	Equations (12.6.5–1), (12.6.5–2), and (12.6.5–3)	2 x 2 (if KEYOPT(6) = 1, use 1 x 1 for volumetric terms)
		Tri.	Equations (12.6.1–1), (12.6.1–2), and (12.6.1–3)	1 (plane strain) 3 (axisymmetric) (if KEYOPT(6)=1, use 1 x 1 for volumetric terms)
	8 node (KEYOPT(1)=1)	Quad	Equations (12.6.7–1) (12.6.7–2) and (12.6.7–3)	3 x 3 (if KEYOPT(6) = 1, use 2 x 2 for volumetric terms)
		Tri.	Equations (12.6.2–1), (12.6.2–2), and (12.6.2–3)	3 (if KEYOPT(6)=1, use 1 x 1 for volumetric terms)

Matrix or Vector	Shape Functions	Integration Points
Mass Matrix	Same as stiffness matrix	2 x 2 (KEYOPT(1)=0) 3 x 3 (KEYOPT(1)=1)
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2 (KEYOPT(1)=0) 3 (KEYOPT(1)=1)

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

Reference: Oden(27), Zienkiewicz(39), Rivlin(89), Kao(90), Mooney(91), and Blatz(92)

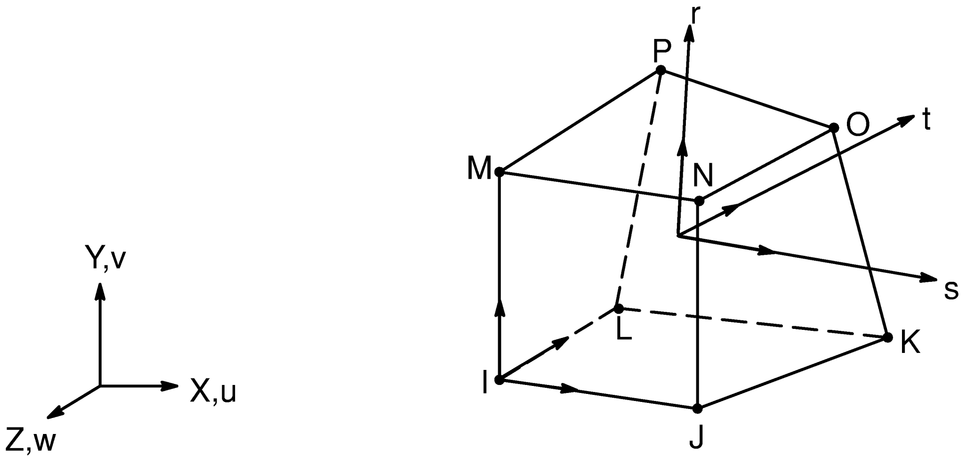
14.84.1 Assumptions and Restrictions

A dropped midside node implies that the edge is and remains straight.

14.84.2 Other Applicable Sections

For the basic element formulation refer to Section 14.86. The hyperelastic material models (Mooney–Rivlin and Blatz–Ko) are described in Section 4.5. Section 13.1 describes integration point locations.

14.86 HYPER86 — 3–D 8–Node Hyperelastic Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2) and (12.8.18–3)	2 x 2 x 2 (if KEYOPT(6) = 1, use 1 x 1 x 1 for volumetric terms)
Mass Matrix	Same as stiffness matrix	2 x 2 x 2
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2 x 2

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Oden(27), Zienkiewicz(39), Rivlin(89), Kao(90), Mooney(91), and Blatz(92)

14.86.1 Other Applicable Sections

The hyperelastic material models (Mooney–Rivlin and Blatz–Ko) are described in Section 4.5. Section 13.1 describes integration point locations.

14.86.2 Virtual Work Statement

The variational principle employed to derive the incremental stiffness matrix of the hyperelastic finite elements described in this section is the incremental principle of virtual work. Internal and external work as well as their increments are expressed in an equilibrium statement for an element.

$$\delta U + \delta \dot{U} = \delta V + \delta \dot{V} \qquad (14.86\text{–}1)$$

where:

δU = internal virtual work

$\delta \dot{U}$ = increment of internal virtual work

δV = external virtual work

$\delta \dot{V}$ = increment of external virtual work

The internal virtual work is expressed as the integral over the volume of the current strain energy density function, W. The external virtual work is the work of the surface pressures over the current surfaces, as well as the work of the nodal point loads. Equation (14.86–1) can be expressed as follows:

$$\int_{vol} \delta W \; d(vol) + \int_{vol} \delta \dot{W} \; d(vol) = \sum_{\substack{\text{element}\\ \text{surfaces}}} \int_{S} P \; \hat{n}_i \; \delta u_i \; dS + \sum_{\substack{\text{element}\\ \text{nodes}}} F_i^{(n)} \; \delta\Delta_i^{(n)} \qquad (14.86\text{–}2)$$

where:

vol = current element volume

W = strain energy density function per unit current volume

P = scalar pressure magnitude

$\hat{n}_i$ = components of the unit normal of the current deformed surface

$F_i^{(n)}$ = applied nodal forces in the i direction at node n

δu_i = displacement field variations of the i coordinate

S = current deformed surface area of elemen

$\delta\Delta_i^{(n)}$ = variation of nodal displacement in the i direction at node n

14.86.3 Element Matrix Derivation

Equation (14.86–2) is the basic equilibrium relationship used to derive the element stiffness matrix and load vectors. Details of the strain energy density function and its variation with respect to current strain are outlined here.

The strain energy density is a function of the current strain components,

$$W = W\left(C_{ij}\right) \tag{14.86–3}$$

where: C_{ij} = components of the right Cauchy–Green deformation tensor (defined below)

Many forms of this functional dependence are possible. The strain energy density functions available are given in Section 4.5.

Without selecting any particular form of W, expressions for the variation of W and an increment of the variation of W are given as follows:

$$\delta W = \frac{\partial W}{\partial C_{ij}} \ \delta C_{ij} \tag{14.86–4}$$

$$\delta \dot{W} = \frac{\partial^2 W}{\partial C_{ij} \partial C_{kl}} \ \delta C_{ij} \ \dot{C}_{kl} + \frac{\partial W}{\partial C_{ij}} \ \delta \dot{C}_{ij} \tag{14.86–5}$$

The deformation tensor [C] is comprised of the products of the deformation gradients [f]

$$C_{ij} = f_{ki} f_{kj} = \text{component of the Cauchy–Green strain tensor} \tag{14.86–6}$$

$$\delta C_{ij} = \delta f_{ki} f_{kj} + f_{ki} \ \delta f_{kj} \tag{14.86–7}$$

$$\delta \dot{C}_{ij} = \delta f_{ki} \dot{f}_{kj} + \delta f_{kj} \ \dot{f}_{ki} \tag{14.86–8}$$

$$\dot{C}_{ij} = \dot{f}_{ki} f_{kj} + f_{ki} \ \dot{f}_{kj} \tag{14.86–9}$$

where: $f_{ij} = \frac{\partial x_i}{\partial X_j}$

X_i = undeformed position of a particle in direction i

x_i = $X_i + u_i$ = deformed position of a particle in direction i

u_i = displacement of particle in direction i

Substitution of (14.86–4) through (14.86–9) into (14.86–1) yields an element equilibrium equation in terms of the external loads and internal strains, as shown by:

$$2\int_{vol}\left(\frac{\partial W}{\partial C_{ij}}\; f_{ki}\delta f_{kj} + \frac{\partial W}{\partial C_{ij}}\; \dot{f}_{ki}\delta f_{kj} + 2\frac{\partial^2 W}{\partial C_{ij}\partial C_{kl}}\; f_{mi}\delta f_{nj} f_{mk}\dot{f}_{nl}\right) d(vol)$$

$$= \sum_{\substack{element\\surfaces}} \int_{S_o} P \; det\left(\underset{\sim}{f}\right)\, f_{ij}^{-1}\; \hat{N}_j\; \delta u_i\; dS_o + \sum_{\substack{element\\nodes}} F_i^{(n)}\; \delta\Delta_i^{(n)} \qquad (14.86–10)$$

where:
$\frac{\partial^2 W}{\partial C_{ij}\partial C_{kl}}$ = incremental moduli (fourth order tensor)
$\hat{N}_j$ = components of the normal to the original undeformed surface
δu_i = variation of the i coordinate displacement field
S_o = undeformed surface area over which P acts

It can be shown that there is a common virtual factor to all terms in equation (14.86–10). Converting from tensor to matrix form, equation (14.86–10) becomes:

$$[K_e(u)]\; \{\dot{u}\} = \{F^{pr}\} + \{F^{nd}\} - \{R(u\,)\} \qquad (14.86–11)$$

where:
$[K_e(u)]$ = current element stiffness matrix
$\{F^{pr}\}$ = total current applied pressures (normal to current surface)
$\{F^{nd}\}$ = total current applied nodal point loads
$\{R(u)\}$ = current Newton–Raphson restoring force vector
$\{\dot{u}\}$ = unknown nodal displacement increments
$\{u\}$ = current total nodal displacements before this solution

Equation (14.86–11) gives the element stiffness equation. The unknown quantity is $\{\dot{u}\}$ while the stiffness matrix and restoring force vector are functions of the current value of displacements {u}, through the deformation gradient [f] and derivations of the strain energy density $\partial W/\partial C_{ij}$ and $\partial^2 W/\partial C_{ij}\partial C_{kl}$.

14.86.4 Reduced Integration on Volumetric Term in Stiffness Matrix

This formulation may produce numerical instability in the nearly incompressible range (Poisson's ratio (ν) 0.5). This can be viewed from the following energy relation:

$$W = \frac{1}{2} \int_{vol} \{\epsilon\}^T \{\sigma\}\ d(vol) \qquad (14.86\text{–}12)$$

where: $\{\varepsilon\}$ = strain vector = $\lfloor \epsilon_{xx} \ \ \epsilon_{yy} \ \ \epsilon_{zz} \ \ \epsilon_{xy} \ \ \epsilon_{yz} \ \ \epsilon_{xz} \rfloor^T$

$\{\sigma\}$ = stress vector = $\lfloor \sigma_{xx} \ \ \sigma_{yy} \ \ \sigma_{zz} \ \ \sigma_{xy} \ \ \sigma_{yz} \ \ \sigma_{xz} \rfloor^T$

In the case of isotropic materials, the stress–strain relation in terms of shear and bulk modulus for 3–D stress state is:

$$\{\sigma\} = (G[D_s] + K[D_v])\ \{\epsilon\} \qquad (14.86\text{–}13)$$

where: G = $\frac{E}{2(1+\nu)}$ = shear modulus

K = $\frac{E}{3(1-2\nu)}$ = bulk modulus

$$[D_s] = \left[\begin{array}{ccc:ccc} \frac{4}{3} & -\frac{2}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & \frac{4}{3} & -\frac{2}{3} & 0 & 0 & 0 \\ -\frac{2}{3} & -\frac{2}{3} & \frac{4}{3} & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$[D_v] = \left[\begin{array}{ccc:ccc} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

subscript s = deviatoric
subscript v = volumetric

The bulk modulus used in equation (14.86–13) becomes unbounded as Poisson's ratio, ν, approaches 0.5. Now using equation (14.86–13) in (14.86–12):

$$W = W_s + W_v \tag{14.86–14}$$

where:

$W_s = \frac{1}{2} \int_{vol} \lfloor \epsilon \rfloor \; G[D_s]\{\epsilon\} \; d(vol)$ = deviatoric (shear) strain energy

$W_v = \frac{1}{2} \int_{vol} \lfloor \epsilon \rfloor \; K[D_v]\{\epsilon\} \; d(vol)$ = volumetric strain energy

Now using the derivative given in Section 2.2, the discretized finite element relationship becomes:

$$(\alpha G[K_s] + [K_v]) \; \{u\} = \alpha\left\{F^{nd}\right\} \tag{14.86–15}$$

where:

$\alpha = \frac{1}{K}$

$[K_s]$ = stiffness associated with shear energy

$[K_v]$ = stiffness associated with volumetric energy

$\{u\}$ = nodal displacements

$\{F^{nd}\}$ = nodal load vector

As Poisson's ratio (ν) approaches 0.5, α approaches 0.0 so that equation (14.86–15) reduces to:

$$[K_v]\{u\} = \{0\} \tag{14.86–16}$$

In equation (14.86–16) if $[K_v]$ is non–singular, only the trivial solution is possible (i.e., {u} = {0}). To enforce a nontrivial solution, $[K_v]$ has to be singular and is achieved by a reduced order integration scheme.

14.86.5 Description of Additional Output Strain Measures

The geometric strain measures output for the hyperelastic element are (1) unit extension and (2) angle change with respect to the global Cartesian system.

The stretch ratio provides the basis for interpretation of the finite strain tensor. The change of length per unit of original length (unit extension) is defined as:

$$\Xi^p = \frac{ds - dS}{dS} = \frac{ds}{dS} - 1 = \Lambda - 1 \tag{14.86–17}$$

where: ds = current length
dS = original length
Λ = stretch ratio
Ξ^p = unit extension

Defining the physical components of Lagrangian strain as follows:

$$E^p_{ij} = \frac{E_{ij}}{\sqrt{G_{ii}}\ \sqrt{G_{jj}}} = \frac{\frac{1}{2}\left(C_{ij} - \delta_{ij}\right)}{\sqrt{G_{ii}}\ \sqrt{G_{jj}}} \qquad (14.86–18)$$

where: E^p_{ij} = Lagrangian strain tensor
C_{ij} = right Cauchy–Green tensor
G_{ij} = metric tensor of reference curvilinear system

The output strains in the directions of the global axes are the unit extensions and are defined as follows:

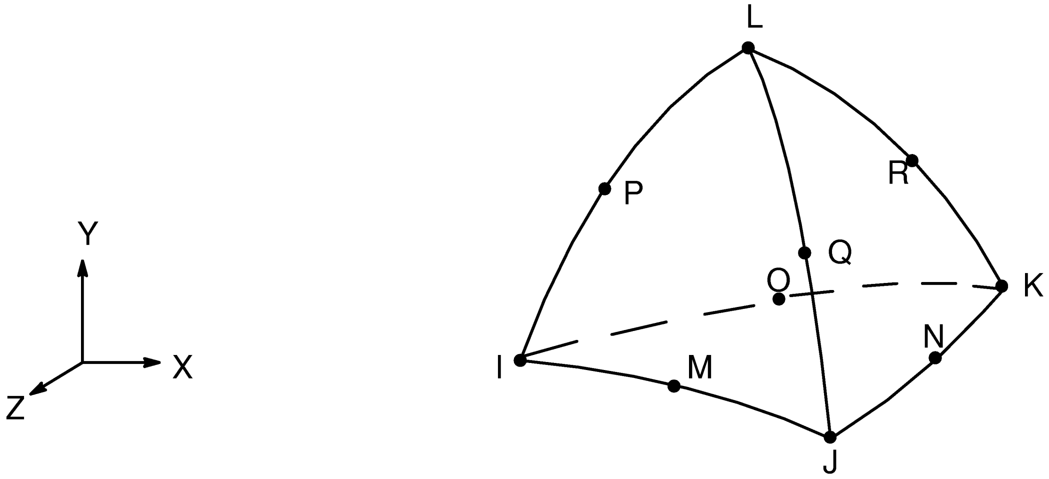
$$\Xi^p_{ii} = \sqrt{1 - 2E^p_{ii}} - 1 \qquad (14.86–19)$$

where: Ξ^p_{ii} = unit extension (output quantities UNEXTN (X, Y, Z))
$\sqrt{1 - 2E^p_{ii}}$ = stretch ratio measure

The shear rotations (output quantities ROTANG (XY, YZ, XZ)) are defined as the angle change from the reference configuration. The equation for the angle change $\Delta\phi_{ij}$ is:

$$\Delta\phi_{ij} = \frac{\pi}{2} - \theta_{ij} = \sin^{-1}\frac{2E^p_{ij}}{\left(\sqrt{1 + 2E^p_{ii}}\right)\ \left(\sqrt{1 + 2E^p_{jj}}\right)} \qquad (14.86–20)$$

14.87 SOLID87 — 3–D 10–Node Tetrahedral Thermal Solid

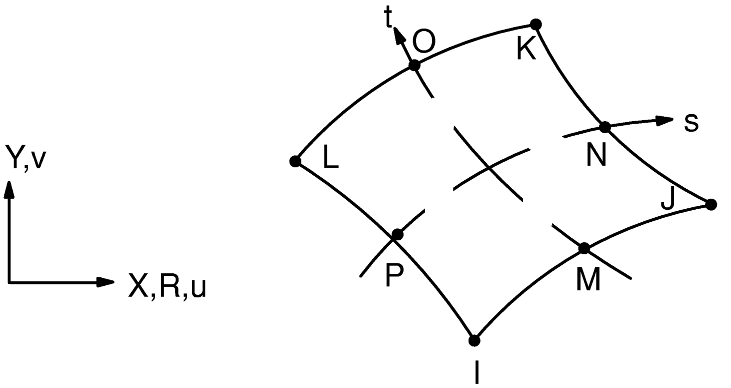


Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equation (12.8.2–20)	4
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, the matrix is diagonalized as described in Section 13.2	11
Heat Generation Load Vector	Equation (12.8.2–20)	4
Convection Surface Matrix and Load Vector	Equation (12.8.2–20) specialized to the face	6

14.87.1 Other Applicable Sections

Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations. If KEYOPT(1) = 1, the specific heat matrix is diagonalized as described in Section 13.2.

14.88 VISCO88 — 2–D 8–Node Viscoelastic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.6.7–1) and (12.6.7–2)	2 x 2
	Triangle	Equations (12.6.2–1) and (12.6.2–2)	3
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Stress Stiffness Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2 along face

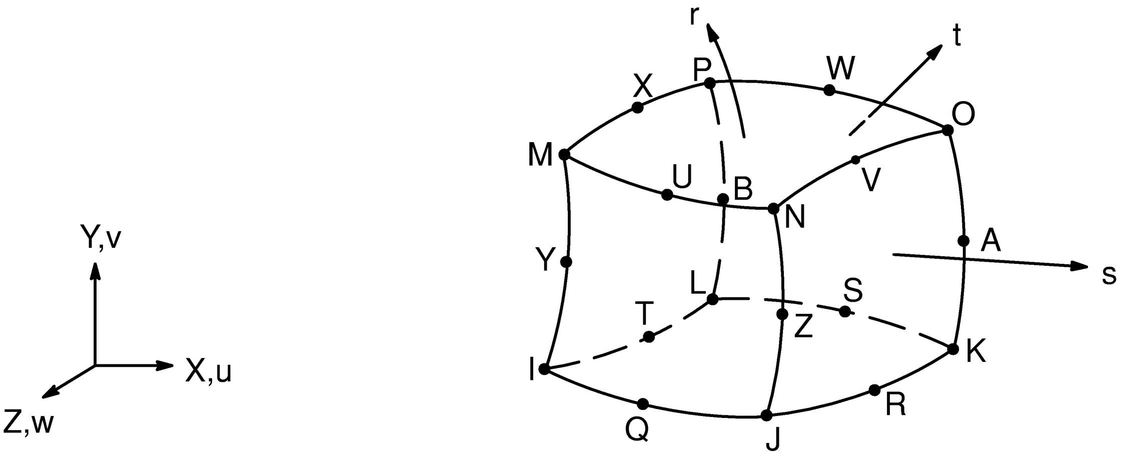
Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

References: Zienkiewicz(39), Markovsky et al.(108), Scherer and Rekhson(109), Narayanaswamy(110), Zienkiewicz et al.(111), Taylor et al.(112)

14.88.1 Other Applicable Sections

Section 4.6 describes the basic theory regarding viscoelasticity and Section 13.1 describes integration point locations.

14.89 VISCO89 —20–Node Viscoelastic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Brick	Equations (12.8.20–1), (12.8.20–2), and (12.8.20–3)	14
	Wedge	Equations (12.8.14–1), (12.8.14–2), and (12.8.14–3)	3 x 3
	Pyramid	Equations (12.8.7–1), (12.8.7–2), and (12.8.7–3)	2 x 2 x 2
	Tet	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Mass Matrix	Same as stiffness matrix.		Same as stiffness matrix
Stress Stiffness Matrix	Same as stiffness matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal and Newton–Raphson Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Quad	Equations (12.5.10–1) and (12.5.10–2)	3 x 3
	Triangle	Equations (12.5.2–1) and (12.5.2–2)	6

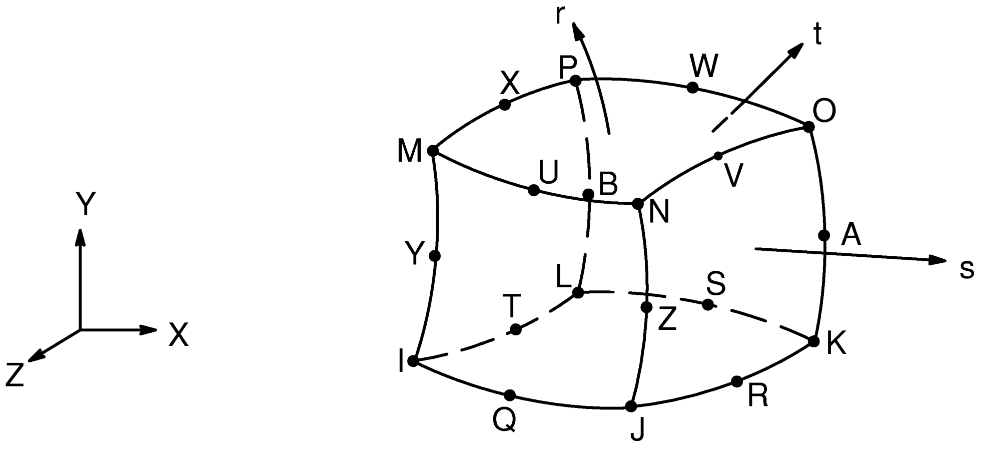
Load Type	Distribution
Element Temperature	Same as shape functions across element
Nodal Temperature	Same as shape functions across element
Pressure	Bilinear across each face

References: Zienkiewicz(39), Markovsky et al.(108), Scherer and Rekhson(109), Narayanaswamy(110), Zienkiewicz et al.(111), Taylor et al.(112)

14.89.1 Other Applicable Sections

Section 4.6 describes the basic theory regarding viscoelasticity and Section 13.1 describes integration point locations. If KEYOPT(3) = 1, the matrix is diagonalized as described in Section 13.2.

14.90 SOLID90 — 20–Node Thermal Solid

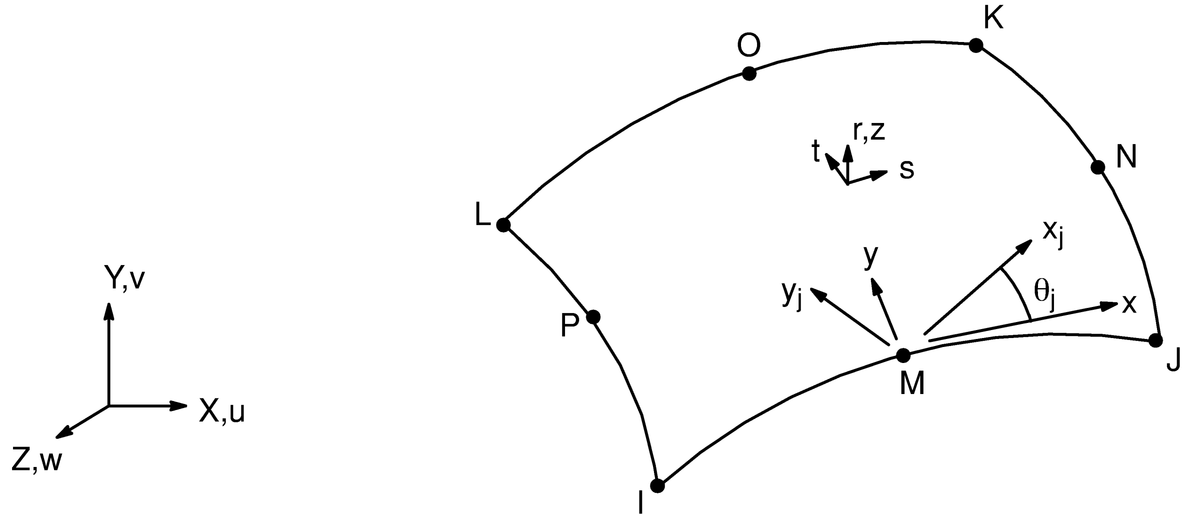


Matrix or Vector	Geometry	Shape Functions	Integration Points
Conductivity Matrix	Brick	Equation (12.8.20–20)	14
	Wedge	Equation (12.8.14–20)	3 x 3
	Pyramid	Equation (12.8.7–20)	2 x 2 x 2
	Tet	Equation (12.8.2–20)	4
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(1) = 1, the matrix is diagonalized as described in Section 13.2.		Same as conductivity matrix
Heat Generation Load Vector	Same as conductivity matrix		Same as conductivity matrix
Convection Surface Matrix and Load Vector	Quad	Equation (12.5.10–20)	3 x 3
	Triangle	Equation (12.5.2–20)	6

14.90.1 Other Applicable Sections

Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Section 13.1 describes integration point locations.

14.91 SHELL91 — Nonlinear Layered Structural Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.14–1), (12.5.14–2), and (12.5.14–3)	Thru–the–thickness: 3 for each layer In–plane: 2 x 2
	Triangle	Equations (12.5.5–1), (12.5.5–2), and (12.5.5–3)	Thru–the–thickness: 3 for each layer In–plane: 3
Mass Matrix	Quad	Equations (12.5.10–1), (12.5.10–2), and (12.5.10–3)	Same as stiffness matrix
	Triangle	Equations (12.5.2–1), (12.5.2–2), and (12.5.2–3)	Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Transverse Pressure Load Vector	Quad	Equation (12.5.10–3)	2 x 2
	Triangle	Equation (12.5.2–3)	3
Edge Pressure Load Vector	Same as in–plane mass matrix specialized to the edge.		2

Load Type	Distribution
Element Temperature	Linear thru each layer, bilinear in plane of element
Nodal Temperature	Constant thru thickness, bilinear in plane of element
Pressure	Bilinear in plane of element, linear along each edge

Reference: Ahmad(1), Cook(5)

14.91.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. The mass matrix is diagonalized as described in Section 13.2. Section 14.99 describes the failure criterion.

14.91.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same element (material) orientation.

There is no significant stiffness associated with rotation about the element r axis. A nominal value of stiffness is present using the approach of Zienkiewicz(39), however, to prevent free rotation at the node.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.91.3 Stress–Strain Relationship

The material property matrix $[D]_j$ for the layer j is:

$$[D]_j = \begin{bmatrix} BE_{x,j} & B\nu_{xy,j}E_{x,j} & 0 & 0 & 0 & 0 \\ B\nu_{xy,j}E_{x,j} & BE_{y,j} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{xy,j} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{yz,j}}{f} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{xz,j}}{f} \end{bmatrix} \qquad (14.91–1)$$

where:

$B = \dfrac{E_{y,j}}{E_{y,j} - \left(\nu_{xy,j}\right)^2 E_{x,j}}$

$E_{x,j}$ = Young's modulus in layer x direction of layer j (input as EX on **MP** command)

$\nu_{xy,j}$ = Poisson's ratio in layer x–y plane of layer j (input as NUXY on **MP** command)

$G_{xy,j}$ = shear modulus in layer x–y plane of layer j (input as GXY on **MP** command)

$f = \left\{ \begin{matrix} 1.2 \\ 1.0 + .2 \dfrac{A}{25t^2} \end{matrix} \right\}$, whichever is greater

A = element area (in s–t plane)

t = average total thickness

The above definition of f is designed to avoid shear locking. Unlike most other elements, the temperature–dependent material properties are evaluated at each of the in–plane integration points, rather than only at the centroid.

14.91.4 Stress, Force and Moment Calculations

The shape functions assume that the transverse shear strains are constant thru the thickness. However, these strains must be zero at the free surface. Therefore, unless nonlinear materials are used or the sandwich option is used (KEYOPT(9) = 1), they are adjusted by:

$$\sigma'_{xz,j} = \frac{3}{2}\left(1 - r^2\right)\sigma_{xz,j} \qquad (14.91\text{–}2)$$

$$\sigma'_{yz,j} = \frac{3}{2}\left(1 - r^2\right)\sigma_{yz,j} \qquad (14.91\text{–}3)$$

where typically:

$\sigma'_{xz,j}$ = adjusted value of transverse shear stress

$\sigma_{xz,j}$ = transverse shear stress as computed from strain–displacement relationships

r = normal coordinate, varying from –1.0 (bottom) to +1.0 (top)

Even with this adjustment, these strains will not be exact due to the variable nature of the material properties thru the thickness. However, for thin shell environments, these strains and their resulting stresses are small in comparison to the x, y, and xy components. The interlaminar shear stresses are equivalent to the transverse shear stresses at the layer boundaries and are computed using equilibrium considerations, and hence are more accurate for most applications.

Force and Moment Summations

The in–plane forces are computed as:

$$T_x = \sum_{j=1}^{N_\ell} t_j \left[\frac{\sigma^t_{x,j} + \sigma^b_{x,j}}{2}\right] \qquad (14.91\text{–}4)$$

$$T_y = \sum_{j=1}^{N_\ell} t_j \left[\frac{\sigma^t_{y,j} + \sigma^b_{y,j}}{2}\right] \qquad (14.91\text{–}5)$$

$$T_{xy} = \sum_{j=1}^{N_\ell} t_j \left[\frac{\sigma^t_{xy,j} + \sigma^b_{xy,j}}{2}\right] \qquad (14.91\text{–}6)$$

where typically: T_x = output quantity

N_ℓ = number of layers

$\sigma^t_{x,j}$ = stress at top of layer j in element x direction

$\sigma^b_{x,j}$ = stress at bottom of layer j in element x direction

t_j = thickness of layer j

The out–of–plane moments are computed as:

$$M_x = \frac{1}{6}\sum_{j=1}^{N_\ell} t_j \left(\sigma^b_{x,j}\left(2z^b_j + z^t_j\right) + \sigma^t_{x,j}\left(2z^t_j + z^b_j\right)\right) \qquad (14.91\text{–}7)$$

$$M_y = \frac{1}{6}\sum_{j=1}^{N_\ell} t_j \left(\sigma^b_{y,j}\left(2z^b_j + z^t_j\right) + \sigma^t_{y,j}\left(2z^t_j + z^b_j\right)\right) \qquad (14.91\text{–}8)$$

$$M_{xy} = \frac{1}{6}\sum_{j=1}^{N_\ell} t_j \left(\sigma^b_{xy,j}\left(2z^b_j + z^t_j\right) + \sigma^t_{xy,j}\left(2z^t_j + z^b_j\right)\right) \qquad (14.91\text{–}9)$$

where, typically: M_x = output quantity MX

z^b_j = z coordinate of bottom layer of j

z^t_j = z coordinate of top of layer j

z = coordinate normal to shell, with z=0 being at shell midsurface

The transverse shear forces are computed as:

$$N_x = \sum_{j=1}^{N_\ell} t_j\, \sigma_{xz,j} \qquad (14.91\text{–}10)$$

$$N_y = \sum_{j=1}^{N_\ell} t_j\, \sigma_{yz,j} \qquad (14.91\text{–}11)$$

where, typically:

N_x = output quantity NX

$\sigma_{xz,j}$ = average transverse shear stress in layer j in element x–z plane

For this computation of transverse shear forces, the shear stresses have not been adjusted as shown in equation (14.91–2) and (14.91–3).

Interlaminar Shear Stress Calculation

In the absence of body forces, the in–plane equilibrium equations of infinitesimally small volume are:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} = 0 \qquad (14.91\text{–}12)$$

$$\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} = 0 \qquad (14.91–13)$$

Rewriting these in incremental form,

$$\Delta\sigma_{xz} = -\Delta z\left(\frac{\Delta\sigma_x}{\Delta x} + \frac{\Delta\sigma_{xy}}{\Delta y}\right) \qquad (14.91–14)$$

$$\Delta\sigma_{yz} = -\Delta z\left(\frac{\Delta\sigma_{yx}}{\Delta x} + \frac{\Delta\sigma_y}{\Delta y}\right) \qquad (14.91–15)$$

Setting these equations in terms of layer j,

$$\Delta\sigma_{xz,j} = -t_j\left(\frac{\Delta\sigma_{x,j}}{\Delta x} + \frac{\Delta\sigma_{xy,j}}{\Delta y}\right) \qquad (14.91–16)$$

$$\Delta\sigma_{yz,j} = -t_j\left(\frac{\Delta\sigma_{yx,j}}{\Delta x} + \frac{\Delta\sigma_{y,j}}{\Delta y}\right) \qquad (14.91–17)$$

where:

$\Delta\sigma_{x,j} = \left(\sigma^2_{x,j} + \sigma^3_{x,j} - \sigma^1_{x,j} - \sigma^4_{x,j}\right)/2$

$\Delta\sigma_{xy,j} = \left(\sigma^4_{xy,j} + \sigma^3_{xy,j} - \sigma^1_{xy,j} - \sigma^2_{xy,j}\right)/2$

$\Delta\sigma_{yx,j} = \left(\sigma^2_{xy,j} + \sigma^3_{xy,j} - \sigma^1_{xy,j} - \sigma^4_{xy,j}\right)/2$

$\Delta\sigma_{y,j} = \left(\sigma^4_{y,j} + \sigma^3_{y,j} - \sigma^1_{y,j} - \sigma^2_{y,j}\right)/2$

$\sigma^3_{x,j}$ = stress in element x direction in layer j at integration point 3

Δx and Δy are shown in Figure 14.91–1.

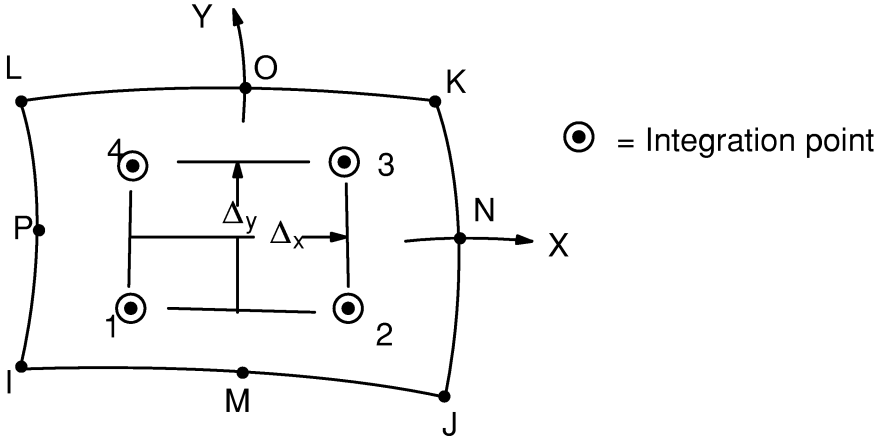


Figure 14.91–1 Integration Point Locations

Thus, the interlaminar shear stress is:

$$\tau_x^k = \sum_{j=1}^{k} \Delta\sigma_{xz,j} - S_x \sum_{j=1}^{k} t_j \qquad (14.91–18)$$

$$\tau_y^k = \sum_{j=1}^{k} \Delta\sigma_{yz,j} - S_y \sum_{j=1}^{k} t_j \qquad (14.91–19)$$

where, typically, τ_x^k = interlaminar shear stress between layers k and k+1 (output quantity ILSXZ)

S_x = $\dfrac{\sum_{j=1}^{N\ell} \Delta\sigma_{xz,j}}{t}$ (= correction term)

t = total thickness

14.91.5 Sandwich Option

If KEYOPT(9) = 1, SHELL91 uses “sandwich ” logic. This causes:

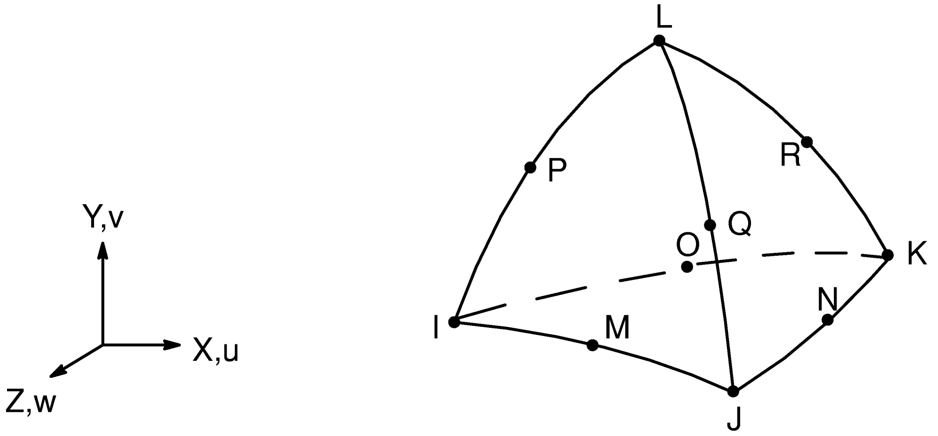
a. The term f in equation (14.91–1) to be set to 1.0 for the middle layer (core).

b. The transverse shear moduli (G_{yz} and G_{xz}) are set to zero for the top and bottom layers.

c. The transverse shear strains and stresses in the face plate (non–core) layers are set to 0.0.

d. As mentioned earlier, the adjustment to the transverse shear strains and stresses in the core as suggested by equations (14.91–2) and (14.91–3) is not done.

14.92 SOLID92 — 3–D 10–Node Tetrahedral Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Stress Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Mass Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Thermal Load Vector	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Pressure Load Vector	Equation (12.8.2–1), (12.8.2–2), and (12.8.2–3) specialized to the face	6

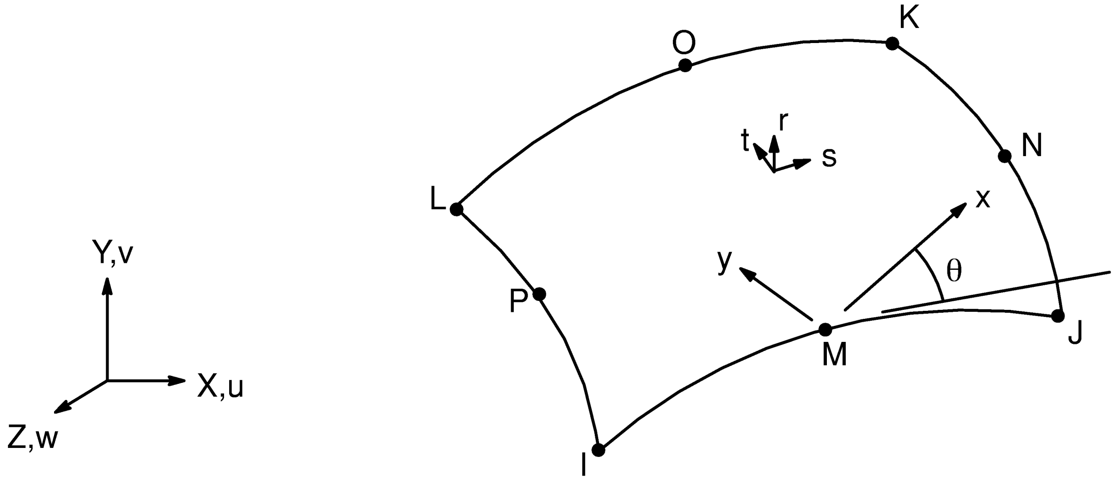
Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Pressure	Linear over each face

Reference: Zienkiewicz(39)

14.92.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.93 SHELL93 — 8–Node Structural Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.14–1), (12.5.14–2), and (12.5.14–3)	Thru–the–thickness: 2 (linear material) 5 (nonlinear material) In–plane: 2 x 2
	Triangle	Equations (12.5.5–1), (12.5.5–2), and (12.5.5–3)	Thru–the–thickness: 2 (linear material) 5 (nonlinear material) In–plane: 3
Mass Matrix	Quad	Equations (12.5.10–1), (12.5.10–2), and (12.5.10–3)	Same as stiffness matrix
	Triangle	Equations (12.5.2–1), (12.5.2–2), and (12.5.2–3)	Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Transverse Pressure Load Vector	Quad	Equation (12.5.10–3)	2 x 2
	Triangle	Equation (12.5.2–3)	3
Edge Pressure Load Vector	Same as in–plane mass matrix, specialized to the edge		2

Load Type	Distribution
Element Temperature	Linear thru thickness, bilinear in plane of element
Nodal Temperature	Constant thru thickness, bilinear in plane of element
Pressure	Bilinear in plane of element, linear along each edge

Reference: Ahmad (1) Cook(5)

14.93.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. The mass matrix is diagonalized as described in Section 13.2.

14.93.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same element (material) orientation.

There is no significant stiffness associated with rotation about the element r axis. A nominal value of stiffness is present (as described with SHELL63), however, to prevent free rotation at the node.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.93.3 Stress–Strain Relationships

The material property matrix [D] for the element is:

$$[D] = \begin{bmatrix} BE_x & B\nu_{xy}E_x & 0 & 0 & 0 & 0 \\ B\nu_{xy}E_x & BE_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{yz}}{f} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{xz}}{f} \end{bmatrix} \qquad (14.93\text{–}1)$$

where:

$B = \dfrac{E_y}{E_y - (\nu_{xy})^2 E_x}$

E_x = Young's modulus in element x direction (input as EX on **MP** command)

ν_{xy} = Poisson's ratio in element x–y plane (input as NUXY on **MP** command)

G_{xy} = shear modulus in element x–y plane (input as GXY on **MP** command)

$f = \begin{Bmatrix} 1.2 \\ 1.0 + .2 \dfrac{A}{25t^2} \end{Bmatrix}$, whichever is greater

A = element area (in s–t plane)

t = average thickness

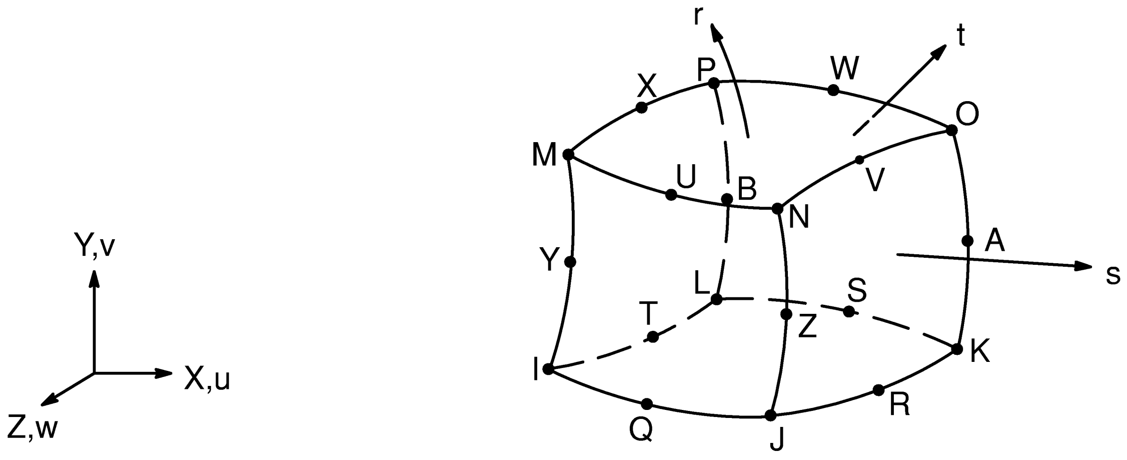
The above definition of f is designed to avoid shear locking.

14.93.4 Stress Output

The stresses at the center of the element are computed by taking the average of the four integration points on that plane. See Section 13.6 for more details.

The output forces and moments are computed as described in Section 2.3.

14.95 SOLID95 — 20–Node Structural Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Brick	Equations (12.8.20–1), (12.8.20–2), and (12.8.20–3)	14 if KEYOPT(11)=0 and 2 x 2 x 2 if KEYOPT(11)=1
	Wedge	Equations (12.8.14–1), (12.8.14–2), and (12.8.14–3)	3 x 3
	Pyramid	Equations (12.8.7–1), (12.8.7–2), and (12.8.7–3)	2 x 2 x 2
	Tet	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Mass Matrix	Same as stiffness matrix.		Same as stiffness matrix
Stress Stiffness Matrix	Same as stiffness matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Quad	Equations (12.5.10–1) and (12.5.10–2)	3 x 3
	Triangle	Equations (12.5.2–1) and (12.5.2–2)	6

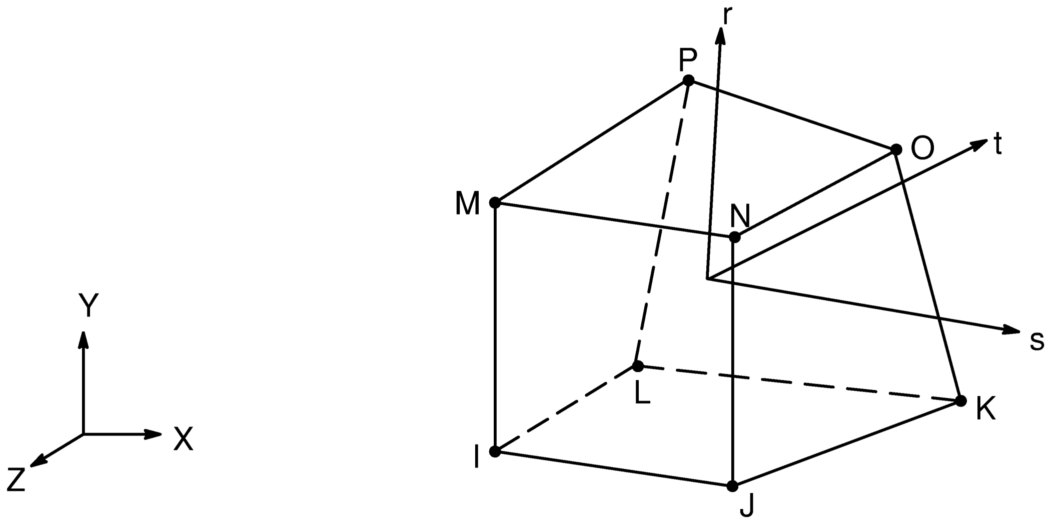
Load Type	Distribution
Element Temperature	Same as shape functions thru element
Nodal Temperature	Same as shape functions thru element
Pressure	Bilinear across each face

Reference: Zienkiewicz(39)

14.95.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. If KEYOPT(3)=1, the mass matrix is diagonalized as described in Section 13.2.

14.96 SOLID96 — 3–D Magnetic Scalar Solid



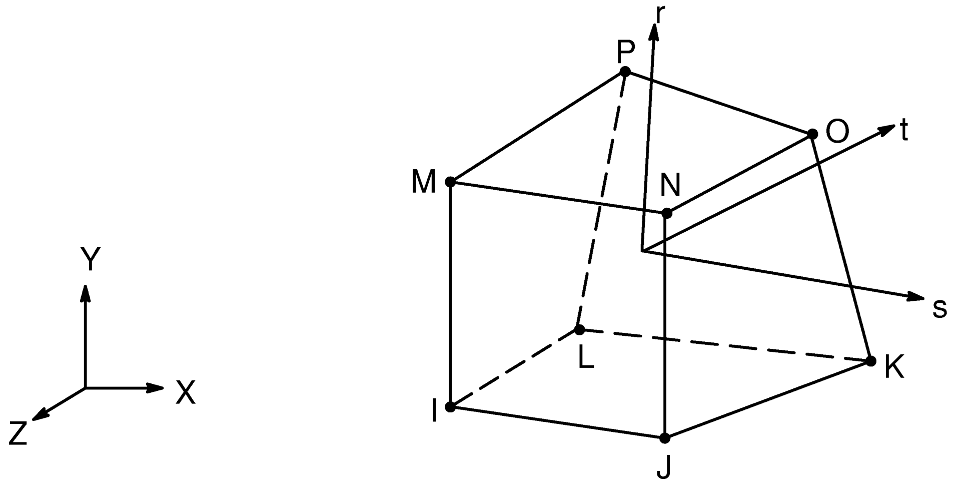
Matrix or Vector	Shape Functions	Integration Points
Magnetic Scalar Potential Coefficient Matrix	Equation (12.8.18–22)	2 x 2 x 2
Load Vector of Magnetism due to Permanent Magnets, and Source Currents	Same as coefficient matrix	2 x 2 x 2

References: Coulomb(76), Mayergoyz(119), Gyimesi(141, 149)

14.96.1 Other Applicable Sections

Section 5.2 discusses the magnetic scalar potential method used by this element. Section 13.1 describes integration point locations.

14.97 SOLID97 — 3–D Magnetic Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Vector Potential Coefficient Matrix	Equations (12.8.18–7), (12.8.18–8), and (12.8.18–9)	2 x 2 x 2
Electric Potential Coefficient Matrix	Equation (12.8.18–21)	2 x 2 x 2
Load Vector of Magnetism due to Source Currents, Permanent Magnets, and Applied Currents	Same as coefficient matrix	2 x 2 x 2

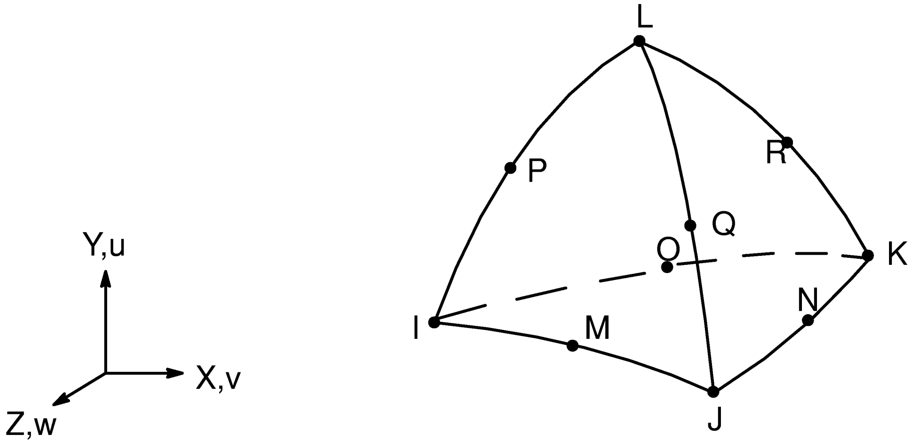
Load Type	Distribution
Current Density, Voltage Load and Phase Angle Distribution	Trilinearly thru element

References: Coulomb(76), Mohammed(118), Biro et al(120)

14.97.1 Other Applicable Sections

Sections 5.2 and 5.3 contain a discussion of the 2–D magnetic vector potential formulation which is similar to the 3–D formulation of this element. Section 13.1 describes integration point locations.

14.98 SOLID98 — Tetrahedral Coupled–Field Solid



Matrix or Vector	Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Equation (12.8.2–22)	4
Electric Conductivity Matrix	Equation (12.8.2–21)	4
Thermal Conductivity Matrix	Equation (12.8.2–20)	4
Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Piezoelectric Coupling Matrix	Same as combination of stiffness matrix and conductivity matrix	4
Specific Heat Matrix	Same as conductivity matrix. If KEYOPT(3) = 1, matrix is diagonalized as described in Section 13.2	11
Mass Matrix	Same as stiffness matrix	4

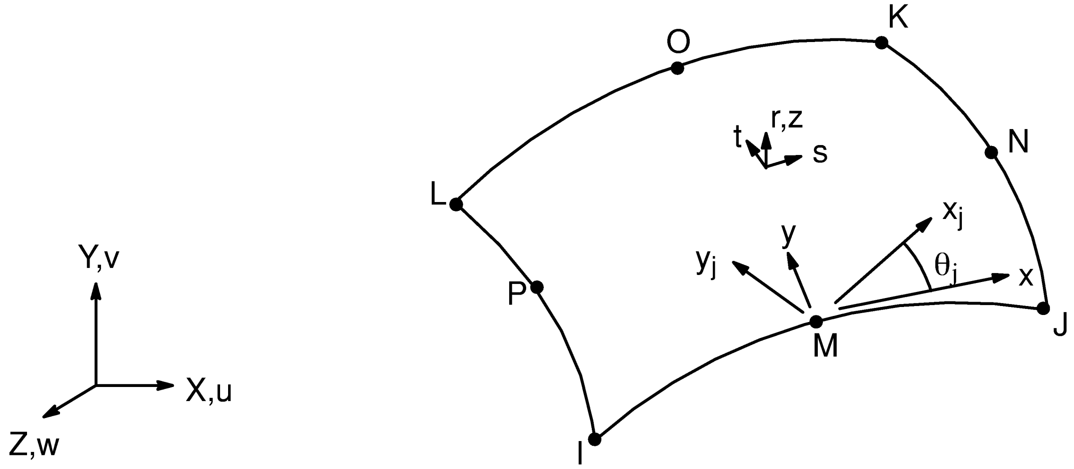
Matrix or Vector	Shape Functions	Integration Points
Load Vector due to Imposed Thermal and Electric Gradients, Heat Generation, Joule Heating, Magnetic Forces, Permanent Magnet and Magnetism due to Source Currents	Same as coefficient or conductivity matrix	4
Thermal Expansion Load Vector	Same as stiffness matrix	4
Load Vector due to Convection and Pressures	Same as stiffness or conductivity matrix, specialized to the face	6

References: Zienkiewicz(39), Coulomb(76), Mayergoyz(119), Gyimesi(141)

14.98.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Chapter 6 describes the derivation of thermal element matrices and load vectors as well as heat flux evaluations. Section 5.2 describes the scalar potential method, which is used by this element. Section 11.1 discusses the piezoelectric capability used by the element. Section 13.1 describes integration point locations. If KEYOPT(3)=1, the specific heat matrix is diagonalized as described in Section 13.2. Also, Section 14.69 discusses the thermoelectric capability.

14.99 SHELL99 — Linear Layered Structural Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.14–1), (12.5.14–2) and (12.5.14–3)	Thru the thickness: 2 In–plane: 2 x 2
	Triangle	Equations (12.5.5–1), (12.5.5–2) and (12.5.5–3)	Thru the thickness: 2 In–plane: 3
Mass Matrix	Quad	Equations (12.5.10–1), (12.5.10–2) and (12.5.10–3)	Same as stiffness matrix
	Triangle	Equations (12.5.2–1), (12.5.2–2) and (12.5.2–3)	Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Transverse Pressure Load Vector	Quad	Equation (12.5.10–3)	2 x 2
	Triangle	Equation (12.5.2–3)	3
Edge Pressure Load Vector	Same as in–plane mass matrix, specialized to the edge		2

Load Type	Distribution
Element Temperature	Linear thru thickness, bilinear in plane of element
Nodal Temperature	Constant thru thickness, bilinear in plane of element
Pressure	Bilinear in plane of element, linear along each edge

References: Ahmad(1), Cook(5), Yunus et al(139)

14.99.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations. The mass matrix is diagonalized as described in Section 13.2.

14.99.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same material orientation.

There is no significant stiffness associated with rotation about the element r axis. A nominal value of stiffness is present using the approach of Zienkiewicz(39), however, to prevent free rotation at the node.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.99.3 Direct Matrix Input

SHELL99 has two options for the direct input of the matrices that account for the stiffness and mass effects as well as one thermal load distribution. This permits the user to incorporate the results of their own composite material programs, as well as lifting any restriction as to the number of layers.

If KEYOPT(2) = 3, the matrices $[E_0]$, $[E_1]$, $[E_2]$, $[E_3]$, and $[E_4]$ are input directly as input quantities A, B, D, E, and F, respectively on the **R** and **RMORE** commands. For the thermal load, the vectors $\{S_0\}$, $\{S_1\}$, and $\{S_2\}$ are input directly as input quantities MT, BT, and QT on the **R** and **RMORE** commands. If KEYOPT(2) = 2, $[E_3]$, $[E_4]$, and {QT} are not used. Further, for both cases, the average density is input directly as input quantity AVDENS on the **RMORE** command.

Considering the KEYOPT(2) = 2 case for a flat shell, the thru thickness accumulated effects can be derived following the theoretical formulation given in reference (139) as:

$$\left[E_0\right] = \sum_{j=1}^{N_\ell} \int_{r_j^{bt}}^{r_j^{tp}} [T_m]_j^T [D]_j [T_m]_j \ dr \tag{14.99–1}$$

$$\left[E_1\right] = \sum_{j=1}^{N_\ell} \int_{r_j^{bt}}^{r_j^{tp}} r[T_m]_j^T [D]_j [T_m]_j \ dr \tag{14.99–2}$$

$$\left[E_2\right] = \sum_{j=1}^{N_\ell} \int_{r_j^{bt}}^{r_j^{tp}} r^2[T_m]_j^T [D]_j [T_m]_j \ dr \tag{14.99–3}$$

$$\left\{S_0\right\} = \sum_{j=1}^{N_\ell} \int_{r_j^{bt}}^{r_j^{tp}} [T_m]_j^T [D]_j \left\{\epsilon^{th}\right\}_j \ dr \tag{14.99–4}$$

$$\{S_1\} = \sum_{j=1}^{N_\ell} \int_{r_j^{bt}}^{r_j^{tp}} r[T_m]_j^T[D]_j \left\{\epsilon^{th}\right\}_j \, dr \qquad (14.99–5)$$

where:
N_ℓ = number of layers
$[D]_j$ = stress–strain relationships at point of interest within layer j
$[T_m]$ = layer to element transformation matrix

The previous five definitions can be used to define the forces and moments on a unit square out of the flat shell:

$$\begin{Bmatrix} \{N\} \\ \{M\} \end{Bmatrix} = \begin{bmatrix} [E_0] & [E_1] \\ [E_1] & [E_2] \end{bmatrix} \begin{Bmatrix} \{\epsilon\} \\ \{\kappa\} \end{Bmatrix} + \begin{Bmatrix} \{S_0\} \\ \{S_1\} \end{Bmatrix} \qquad (14.99–6)$$

where:
{N} = forces per unit length
{M} = moments per unit length
$\{\varepsilon\}$ = strains
$\{\varkappa\}$ = curvatures

Each of the above matrices and load vectors are of sizes 6 x 6 and 6 x 1, as opposed to the 3 x 3 and 3 x 1 sizes commonly used in thin shell analysis. Thus, if only 3 x 3 matrix information is available, it is recommended to input the 6 x 6 matrices in the following form (using $[E_0]$ as an example):

$$[E_o] = \begin{bmatrix} G_{11} & G_{12} & 0 & G_{13} & 0 & 0 \\ G_{12} & G_{22} & 0 & G_{23} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ G_{13} & G_{23} & 0 & G_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & H & 0 \\ 0 & 0 & 0 & 0 & 0 & H \end{bmatrix} \qquad (14.99–7)$$

where:
G = 3 x 3 matrix of terms available from outside of the ANSYS program
H = relatively large number to suppress shear deflections (perhaps 10^3 x G_{33})

As discussed earlier, the values in $[E_0]$ (as well as other matrices) used by the ANSYS program for either the layer or matrix input may be printed with KEYOPT(10) = 1 in order to verify the input.

For matrix input, the required stress vector $\{N_c\}$ needed for stress stiffening is computed as:

$$\{N_c\} = \left([E_0]\,[B_0] + [E_1]\,[B_1]\right)\{\delta\} + \{S_0\} \tag{14.99–8}$$

where: $\{\delta\}$ = $\{u_e\}$ from the previous iteration

14.99.4 Stress Calculations

Strains and stresses are computed at the top and bottom of each layer (KEYOPT(9) = 0) or at the midthickness (KEYOPT(9) = 1). The strains within layer j are:

$$\{\epsilon\}_j = [T_m]_j[B]\;\{u_e\} \tag{14.99–9}$$

where: $\{u_e\}$ = element displacement vector

The stresses within layer j are:

$$\{\sigma\}_j = [D]_j\left(\{\epsilon\}_j - \{\epsilon^{th}\}_j\right) \tag{14.99–10}$$

where: $\{\varepsilon^{th}\}_j$ = thermal strain in layer j

14.99.5 Force and Moment Summations

First, all stresses are converted from the layer orientation to the element orientation:

$$\{\sigma_e\}_j = [T_m]_j^T\{\sigma\}_j \tag{14.99–11}$$

where: $\{\sigma_e\}_j$ = stresses in element orientation

To simplify the below descriptions, the subscript e is dropped. The in–plane forces are computed as:

$$T_x = \sum_{j=1}^{N_\ell} t_j\left[\frac{\sigma^t_{x,j} + \sigma^b_{x,j}}{2}\right] \tag{14.99–12}$$

$$T_y = \sum_{j=1}^{N_\ell} t_j\left[\frac{\sigma^t_{y,j} + \sigma^b_{y,j}}{2}\right] \tag{14.99–13}$$

$$T_{xy} = \sum_{j=1}^{N_\ell} t_j \left[\frac{\sigma^t_{xy,j} + \sigma^b_{xy,j}}{2} \right] \tag{14.99–14}$$

where, typically, T_x = output quantity TX

$\sigma^t_{x,j}$ = stress at top of layer j in element x direction

$\sigma^b_{x,j}$ = stress at bottom of layer j in element x direction

t_j = thickness of layer j

The out–of–plane moments are computed as:

$$M_x = \frac{1}{6} \sum_{j=1}^{N_\ell} t_j \left(\sigma^b_{x,j} \left(2z^b_j + z^t_j \right) + \sigma^t_{x,j} \left(2z^t_j + z^b_j \right) \right) \tag{14.99–15}$$

$$M_y = \frac{1}{6} \sum_{j=1}^{N_\ell} t_j \left(\sigma^b_{y,j} \left(2z^b_j + z^t_j \right) + \sigma^t_{y,j} \left(2z^t_j + z^b_j \right) \right) \tag{14.99–16}$$

$$M_{xy} = \frac{1}{6} \sum_{j=1}^{N_\ell} t_j \left(\sigma^b_{xy,j} \left(2z^b_j + z^t_j \right) + \sigma^t_{xy,j} \left(2z^t_j + z^b_j \right) \right) \tag{14.99–17}$$

where, typically, M_x = output quantity MX

z^b_j = z coordinate of bottom layer j

z^t_j = z coordinate of top of layer j

z = coordinate normal to shell, with z=0 being at shell midsurface

The transverse shear forces are computed as:

$$N_x = \sum_{j=1}^{N_\ell} t_j \ \sigma_{xz,j} \tag{14.99–18}$$

$$N_y = \sum_{j=1}^{N_\ell} t_j \ \sigma_{yz,j} \tag{14.99–19}$$

where, typically, N_x = output quantity NX

$\sigma_{xz,j}$ = average transverse shear stress in layer j in element x–z plane

For this computation of transverse shear forces, the shear stresses have not been adjusted as shown in the next subsection.

14.99.6 Shear Strain Adjustment

The shape functions assume that the transverse shear strains are constant thru the thickness. However, these strains must be zero at the free surface. Therefore, they are adjusted by:

$$\epsilon'_{xz,j} = \frac{3}{2}\left(1 - r^2\right) \epsilon_{xz,j} \tag{14.99–20}$$

$$\epsilon'_{yz,j} = \frac{3}{2}\left(1 - r^2\right) \epsilon_{yz,j} \tag{14.99–21}$$

where typically, $\epsilon'_{xz,j}$ = adjusted value of transverse shear strain

$\epsilon_{xz,j}$ = transverse shear strain as computed from strain–displacement relationships

r = normal coordinate, varying from –1.0 (bottom) to +1.0 (top)

Even with this adjustment, these strains will not be exact due to the variable nature of the material properties thru the thickness. However, for thin shell environments, these strains and their resulting stresses are small in comparison to the x, y, and xy components. The interlaminar shear stresses are equivalent to the transverse shear stresses at the layer boundaries and are computed using equilibrium considerations, and hence are more accurate for most applications.

14.99.7 Interlaminar Shear Stress Calculations

In the absence of body forces, the in–plane equilibrium equations of infinitesimally small volume are

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} = 0 \tag{14.99–22}$$

$$\frac{\partial \sigma_{yx}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} = 0 \tag{14.99–23}$$

Rewriting these in incremental form,

$$\Delta\sigma_{xz} = -\,\Delta z\left(\frac{\Delta\sigma_x}{\Delta x} + \frac{\Delta\sigma_{xy}}{\Delta y}\right) \tag{14.99–24}$$

$$\Delta\sigma_{yz} = -\Delta z\left(\frac{\Delta\sigma_{yx}}{\Delta x} + \frac{\Delta\sigma_y}{\Delta y}\right) \qquad (14.99–25)$$

Setting these equations in terms of layer j,

$$\Delta\sigma_{xz,j} = -t_j\left(\frac{\Delta\sigma_{x,j}}{\Delta x} + \frac{\Delta\sigma_{xy,j}}{\Delta y}\right) \qquad (14.99–26)$$

$$\Delta\sigma_{yz,j} = -t_j\left(\frac{\Delta\sigma_{yx,j}}{\Delta x} + \frac{\Delta\sigma_{y,j}}{\Delta y}\right) \qquad (14.99–27)$$

where:

$\Delta\sigma_{x,j} = \left(\sigma^2_{x,j} + \sigma^3_{x,j} - \sigma^1_{x,j} - \sigma^4_{x,j}\right)/2.0$

$\Delta\sigma_{xy,j} = \left(\sigma^3_{xy,j} + \sigma^4_{xy,j} - \sigma^1_{xy,j} - \sigma^2_{xy,j}\right)/2.0$

$\Delta\sigma_{yx,j} = \left(\sigma^2_{xy,j} + \sigma^3_{xy,j} - \sigma^1_{xy,j} - \sigma^4_{xy,j}\right)/2.0$

$\Delta\sigma_{y,j} = \left(\sigma^3_{y,j} + \sigma^4_{y,j} - \sigma^1_{y,j} - \sigma^2_{y,j}\right)/2.0$

$\sigma^3_{x,j}$ = stress in element x direction in layer j at integration point 3

Δx and Δy are shown in Figure 14.99–1.

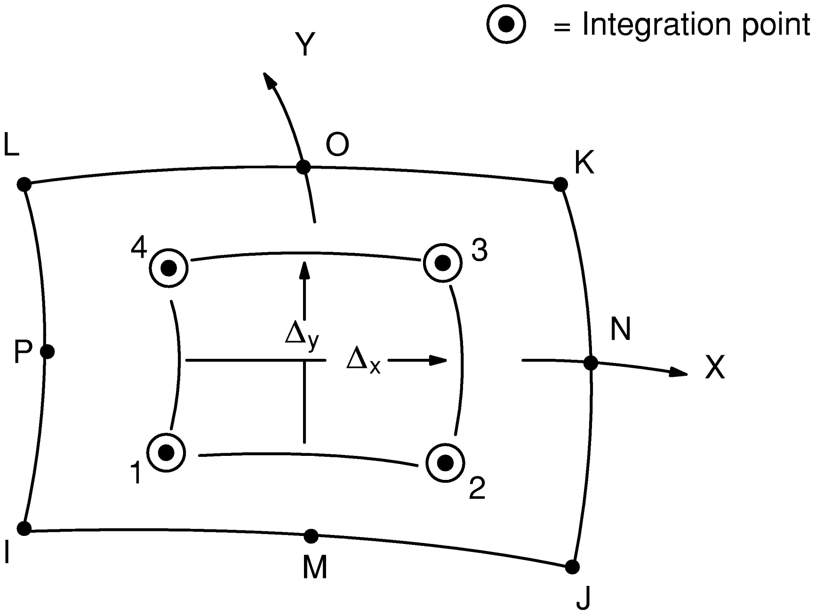


Figure 14.99–1 Integration Point Locations

The interlaminar shear stress components between layer k and layer k+1 may now be written as:

$$\tau_x^k = \sum_{j=1}^{k} \Delta\sigma_{xz,j} - A_x \sum_{j=1}^{k} t_j \qquad (14.99–28)$$

$$\tau_y^k = \sum_{j=1}^{k} \Delta\sigma_{yz,j} - A_y \sum_{j=1}^{k} t_j \qquad (14.99–29)$$

where, typically, τ_x^k = output quantity ILSXZ

A_x = $\dfrac{\sum_{j=1}^{N} \Delta\sigma_{xz,j}}{t}$ (= correction term)

t = total thickness

Finally,

$$\tau^k = \sqrt{\left(\tau_x^k\right)^2 + \left(\tau_y^k\right)^2} \qquad (14.99–30)$$

where: τ^k = output quantity ILSUM

The maximum of all values τ^k is τ_{max}^k (output quantity ILMAX). If τ_{max}^k is less than a small number β, the interlaminar shear stress printout is suppressed and the postdata values are set to zero. β is determined by:

$$\beta = 10^{-8} \sum \left(|\sigma_x| + |\sigma_y| + |\tau_{xy}|\right) \qquad (14.99–31)$$

where the summation is over all integration points in the top and bottom layers (or in layers LP1 and LP2, if requested).

Finally, a check is made on the validity of the interlaminar shear stresses. R is defined as:

$$R = \frac{t\sqrt{A_x^2 + A_y^2}}{\tau_{max}^k} \qquad (14.99–32)$$

where: R = output quantity Max. adjustment / Max. stress

R is output if it is greater than 0.1.

14.99.8 Failure Criteria

Possible failure of a material can be evaluated by up to six different criteria, of which three are predefined. They are evaluated at the top and bottom (or middle) of each layer at each of the in–plane integration points. The failure criteria are:

Maximum strain failure criteria

$$\xi_1 = \text{maximum of} \left\{ \begin{array}{l} \dfrac{\epsilon_{xt}}{\epsilon^f_{xt}} \text{ or } \dfrac{\epsilon_{xc}}{\epsilon^f_{xc}} \quad \text{whichever is applicable} \\[2ex] \dfrac{\epsilon_{yt}}{\epsilon^f_{yt}} \text{ or } \dfrac{\epsilon_{yc}}{\epsilon^f_{yc}} \quad \text{whichever is applicable} \\[2ex] \dfrac{\epsilon_{zt}}{\epsilon^f_{zt}} \text{ or } \dfrac{\epsilon_{zc}}{\epsilon^f_{zc}} \quad \text{whichever is applicable} \\[2ex] \dfrac{|\epsilon_{xy}|}{\epsilon^f_{xy}} \\[2ex] \dfrac{|\epsilon_{yz}|}{\epsilon^f_{yz}} \\[2ex] \dfrac{|\epsilon_{xz}|}{\epsilon^f_{xz}} \end{array} \right. \tag{14.99–33}$$

where:

ξ_1 = value of maximum strain failure criterion (output quantity FC1) (accessed with input on **TBDATA** command with **TB**,FAIL)

ε_{xt} = $\left\{ \begin{array}{l} 0 \\ \epsilon_x \end{array} \right.$ whichever is greater

ε_x = strain in layer x–direction

ε_{xc} = $\left\{ \begin{array}{l} \epsilon_x \\ 0 \end{array} \right.$ whichever is lesser

ϵ^f_{xt} = failure strain in layer x–direction in tension (input on **TBDATA** command with **TB**,FAIL)

Maximum stress failure criteria

$$\xi_2 = \text{maximum of} \begin{cases} \dfrac{\sigma_{xt}}{\sigma^f_{xt}} \text{ or } \dfrac{\sigma_{xc}}{\sigma^f_{xc}} \text{ whichever is applicable} \\[2ex] \dfrac{\sigma_{yt}}{\sigma^f_{yt}} \text{ or } \dfrac{\sigma_{yc}}{\sigma^f_{yc}} \text{ whichever is applicable} \\[2ex] \dfrac{\sigma_{zt}}{\sigma^f_{zt}} \text{ or } \dfrac{\sigma_{zc}}{\sigma^f_{zc}} \text{ whichever is applicable} \\[2ex] \dfrac{|\sigma_{xy}|}{\sigma^f_{xy}} \\[2ex] \dfrac{|\sigma_{yz}|}{\sigma^f_{yz}} \\[2ex] \dfrac{|\sigma_{xz}|}{\sigma^f_{xz}} \end{cases} \qquad (14.99\text{–}34)$$

where:

ξ_2 = value of maximum stress failure criterion (output quantity FC2)

$\sigma_{xt} = \begin{cases} 0 \\ \sigma_x \end{cases}$ whichever is greater

σ_x = stress in layer x–direction

$\sigma_{xc} = \begin{cases} \sigma_x \\ 0 \end{cases}$ whichever is lesser

σ^f_{xt} = failure stress in layer x–direction in tension (input on **TBDATA** command with **TB**,FAIL)

Tsai–Wu failure criteria

If TBDATA,3,1 is used after TBTEMP,,CRIT, the criterion used is the “strength index”:

$$\xi_3 = A + B \qquad (14.99\text{–}35)$$

and if TBDATA,3,2 is used after TBTEMP,,CRIT, the criterion used is the inverse of the “strength ratio”:

$$\xi_3 = 1.0 / \left(-\frac{B}{2A} + \sqrt{(B / 2A)^2 + 1.0 / A}\right) \qquad (14.99\text{–}36)$$

where: ξ_3 = value of Tsai–Wu failure criterion (output quantity FC3) (accessed with input **TBDATA** command with **TB**,FAIL)

$$A = -\frac{(\sigma_x)^2}{\sigma^f_{xt}\ \sigma^f_{xc}} - \frac{\left(\sigma_y\right)^2}{\sigma^f_{yt}\ \sigma^f_{yc}} - \frac{(\sigma_z)^2}{\sigma^f_{zt}\ \sigma^f_{zc}} + \frac{\left(\sigma_{xy}\right)^2}{\left(\sigma^f_{xy}\right)^2} + \frac{\left(\sigma_{yz}\right)^2}{\left(\sigma^f_{yz}\right)^2} + \frac{(\sigma_{xz})^2}{\left(\sigma^f_{xz}\right)^2}$$

$$+ \frac{C_{xy}\sigma_x\sigma_y}{\sqrt{\sigma^f_{xt}\ \sigma^f_{xc}\ \sigma^f_{yt}\ \sigma^f_{yc}}} + \frac{C_{yz}\sigma_y\sigma_z}{\sqrt{\sigma^f_{yt}\ \sigma^f_{yc}\ \sigma^f_{zt}\ \sigma^f_{zc}}} + \frac{C_{xz}\,\sigma_x\sigma_z}{\sqrt{\sigma^f_{xt}\ \sigma^f_{xc}\ \sigma^f_{zt}\ \sigma^f_{zc}}}$$

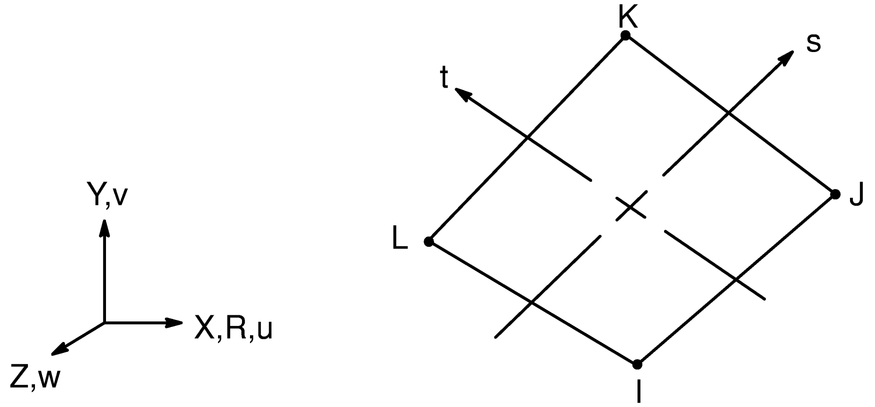
$$B = \left(\frac{1}{\sigma^f_{xt}} + \frac{1}{\sigma^f_{xc}}\right)\sigma_x + \left(\frac{1}{\sigma^f_{yt}} + \frac{1}{\sigma^f_{yc}}\right)\sigma_y + \left(\frac{1}{\sigma^f_{zt}} + \frac{1}{\sigma^f_{zc}}\right)\sigma_z$$

$C_{xy,}\ C_{yz},\ C_{xz}$ = x–y, y–z, x–z, respectively, coupling coefficient for Tsai–Wu theory (input on **TBDATA** command with **TB**,FAIL, defaults to –1.0)

The Tsai–Wu failure criteria used here are 3–D versions of the failure criterion reported in of Tsai and Hahn(190) for the ‘strength index’ and of Tsai(93) for the ‘strength ratio’. Apparent differences are:

1. The program input used negative values for compression limits, whereas Tsai uses positive values for all limits.
2. The program uses C_{xy} instead of the F^*_{xy} used by Tsai and Hahn with C_{xy} being twice the value of F^*_{xy}.

14.106 VISCO106 — 2–D Viscoplastic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.6.5–1), (12.6.5–2), and (12.6.5–3)	2 x 2
	Triangle	Equations (12.6.1–1), (12.6.1–2), and (12.6.1–3)	3 if axisymmetric and 1 if plane
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as mass matrix, specialized to face		2

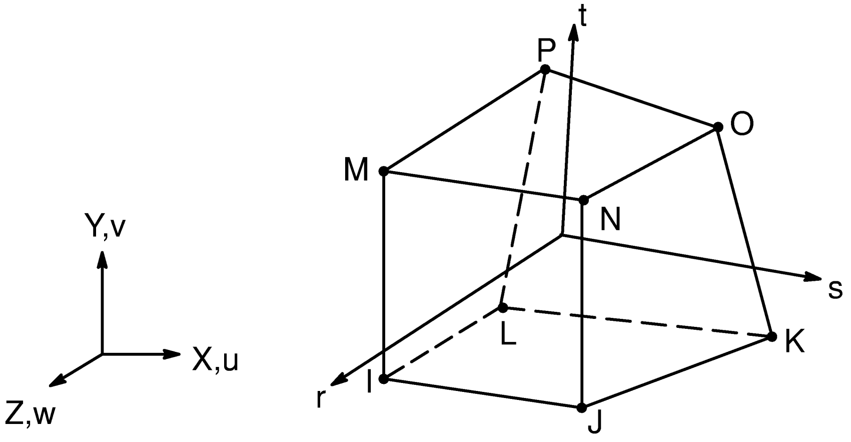
Load Type	Distribution
Element Temperature	Bilinear across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

References: Oden(123), Weber et al.(127), Anand(159) and Brown et al.(147)

14.106.1 Other Applicable Sections

For the basic element formulation refer to Section 14.107. Rate–dependent plasticity (Anand's model) is described in Section 4.2. Section 13.1 describes integration point locations.

14.107 VISCO107 — 3–D Viscoplastic Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2) and (12.8.18–3)	2 x 2 x 2
Mass Matrix	Same as stiffness matrix	2 x 2 x 2
Thermal Load Vector	Same as stiffness matrix	2 x 2 x 2
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2 x 2

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

References: Oden(123), Weber et al.(127)

14.107.1 Basic Assumptions

This section discusses the basic theory of the large strain viscoplastic elements. The elements developed use the updated Lagrangian concept along with logarithmic (Hencky) strain and Cauchy (True) stress measures. The material is limited to be isotropic in nature and elastic strains are assumed to be small relative to plastic strains. Further the plastic flow is assumed to be isochoric (i.e. volume preserving) and both the rate–independent and rate–dependent elastic–plastic constitutive relationship is considered.

The strain energy calculation is based on integration of loading rates and a large time increment may produce inaccurate energy values, even though stress–strain solutions are quite accurate.

14.107.2 Element Tangent Matrices and Newton–Raphson Restoring Force

The formulation considered is highly nonlinear in nature both from the point of view of kinematic or geometric consideration as well as constitutive behavior. The full Newton–Raphson solution option is utilized when the Newton–Raphson restoring force is given by (see equation (15.9–2):

$$\{F^{nr}\} = \int_{vol} [B]^T\{\sigma\}\ d(vol) \qquad (14.107\text{–}1)$$

where:

$[B]$ = strain–displacement matrix

$\{\sigma\}$ = Cauchy stress

Equation (14.107–1) is modified by assuming a decomposition of the Cauchy stress into the deviatoric part plus the pressure part:

$$\{\sigma\} = \{\sigma'\} - \{q\}P \qquad (14.107\text{–}2)$$

where:

$\{\sigma'\}$ = Cauchy stress deviator

$\lfloor q \rfloor$ = $\lfloor 1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \rfloor$

P = hydrostatic stress = – ($\sigma_x + \sigma_y + \sigma_z$) / 3

The negative value of P is output as HPRES.

The pressure is separately interpolated to conveniently allow for enforcement of the incompressibility constraint associated with large plastic strains (Oden and Kikuchi(123)). The restoring force can now be rewritten as:

$$\{F^{nr}\} = \int_{vol} [B]^T\{\sigma'\}\ d(vol) - \int [B]^T\{q\}\ P\ d(vol) \qquad (14.107\text{–}3)$$

The incompressibility constraint during plastic flow is enforced through the augmentation of the momentum equations with the additional equation:

$$\int_{vol} [N^P]^T\ (\Delta J - \Delta\hat{J}\ (\Delta P))\ d(vol) = 0 \qquad (14.107\text{–}4)$$

where: $[N^P]$ = shape function associated with the independently interpolated pressure DOF

ΔJ = determinant of the relative deformation gradient (the relative volume change)

$\Delta\hat{J}$ = constitutively prescribed function expressing the pressure–volume change relationship and expressed as:

$$\Delta\hat{J} = \exp\ \frac{-\Delta P}{K} \qquad (14.107\text{–}5)$$

where: K = elastic bulk modulus for the material (= E / (3(1–2ν)))

E = Young's modulus (input as EX on **MP** command)

ν = Poisson's ratio (input as NUXY on **MP** command)

The total Cauchy stress is calculated by finding the deviatoric part from the constitutive equations using the strains calculated from nodal displacements and subtracting the separately interpolated pressure, i.e.:

$$\{\bar{\sigma}\} = \{\sigma'\} - \{q\}\ P_o \qquad (14.107\text{–}6)$$

where: P_o = interpolated from the pressure field

The stiffness matrix for the static analysis is constructed by evaluating the exact Jacobian of the discretized system. This yields an equation of the form:

$$\begin{bmatrix} K^{uu} & K^{up} \\ K^{pu} & K^{pp} \end{bmatrix} \begin{Bmatrix} \Delta u \\ \Delta P \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \end{Bmatrix} - \begin{Bmatrix} F^u \\ F^p \end{Bmatrix} \qquad (14.107\text{–}7)$$

where: $\{F\}$ = external nodal forces

$\{\Delta u\}, \{\Delta P\}$ = increments of displacement and pressure, respectively

$\{F^u\}$ = $\int_{vol} [B]^T\{\bar{\sigma}\}\ d(vol)$

$$\{F^p\} = \int_{vol} [N^p]^T \left(\Delta J - \Delta\hat{J}(\Delta P)\right) d(vol)$$

$$[K^{uu}] = \frac{\partial}{\partial u}\left[\int_{vol} [B]^T\{\overline{\sigma}\}\ d(vol)\right]$$

$$[K^{up}] = [K^{pu}]^T = \frac{\partial}{\partial p}\left[\int_{vol} [B]^T\{\overline{\sigma}\}\ d(vol)\right]$$

$$[K^{pp}] = \frac{\partial}{\partial p}\left[\int_{vol} [N^p]\left(\Delta J - \Delta\hat{J}(\Delta P)\right) d(vol)\right]$$

The tangent matrix developed in equation (14.107–7) has two parts, namely the constitutive part and geometric part. From the requirement of full tangent matrices, both the constitutive and geometric parts are essential, but the numerical efficiency and stability considerations can prove to be different. Thus, it is left on the user to control the inclusion of stress (geometric) stiffness by the **SSTIF**,ON or **SSTIF**,OFF command. Symmetry of the stiffness matrix is achieved by assuming small strain increments for the constitutive part and negligible volume change during the step for the geometric part. The assumptions generally result in good convergence characteristic for these elements even when these assumptions of small strain increments and negligible volume change are violated. Additional detail of the stiffness matrix can be found in Weber et al.(127).

For the constitutive part of the rate–dependent plasticity (Anand's model), see Section 4.2. Section 13.1 describes the integration point locations.

14.107.3 Plastic Energy Output

$$E_e^{pl} = \sum_{i=1}^{NINT}\sum_{j=1}^{NCS} \{\sigma\}^T\left\{\Delta\epsilon^{pl}\right\} vol_i \qquad (14.107–8)$$

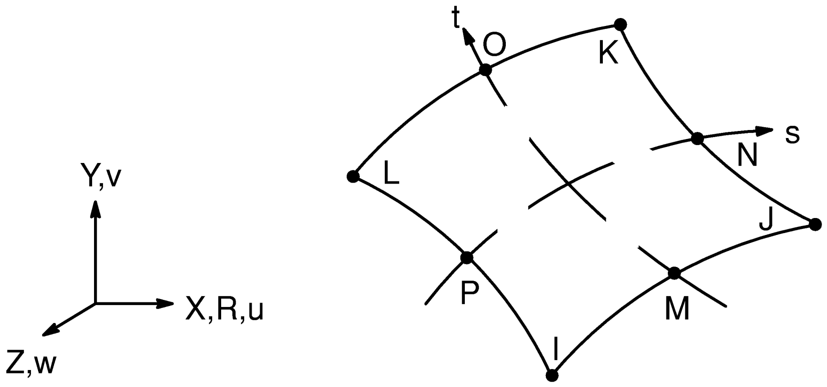
= plastic energy per unit volume (accessed with NL,PLWK on the **ETABLE** command (VISCO106, VISCO107 and VISCO108 only))

NINT = number of integration points

NCS = total number of converged substeps

$\{\Delta\varepsilon^{pl}\}$ = plastic strain increment

14.108 VISCO108 — 2–D 8–Node Viscoplastic Solid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffnesss Matrix	Quad	Equations (12.6.7–1), (12.6.7–2), and (12.6.7–3)	3 x 3
	Triangle	Equations (12.6.2–1), (12.6.2–2), and (12.6.2–3)	3
Mass Matrix	Same as stiffness matrix		Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix		Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face		2

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

References: Oden(123), Weber et al(127), Anand(159) and Brown et al(147)

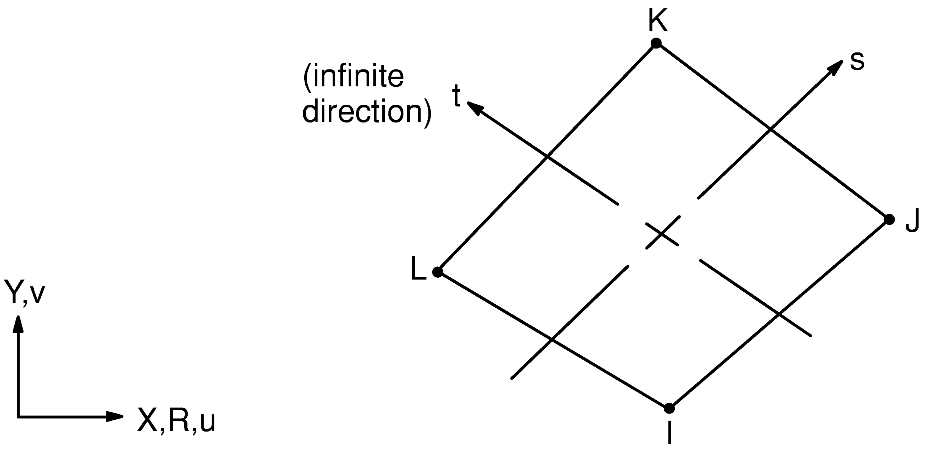
14.108.1 Other Applicable Sections

For the basic element formulation refer to Section 14.107. Rate–dependent plasticity (Anand's model) is described in Section 4.2. Section 13.1 describes integration point locations.

14.108.2 Assumptions and Restrictions

A dropped midside node implies that the edge is and remains straight.

14.110 INFIN110 — 2–D Infinite Solid



Matrix or Vector	Mapping and Shape Functions	Integration Points
Magnetic Potential Coefficient Matrix	Equations (12.6.8–9), (12.6.8–33), and (12.6.8–34)	2 x 2
Thermal Conductivity Matrix	Equations (12.6.8–20), (12.6.8–33), and (12.6.8–34)	2 x 2
Specific Heat Matrix	Equations (12.6.8–20), (12.6.8–33), and (12.6.8–34)	2 x 2
Electrical Potential Coefficient Matrix	Equations (12.6.8–21), (12.6.8–33), and (12.6.8–34)	2 x 2

References: Zienkiewicz et al.(169), Damjanic’ and Owen(170), Marques and Owen(171), Li et al.(172)

14.110.1 Mapping Functions

The theory for the infinite mapping functions is briefly summarized here. Consider the one–dimensional situation shown in Figure 14.110–1

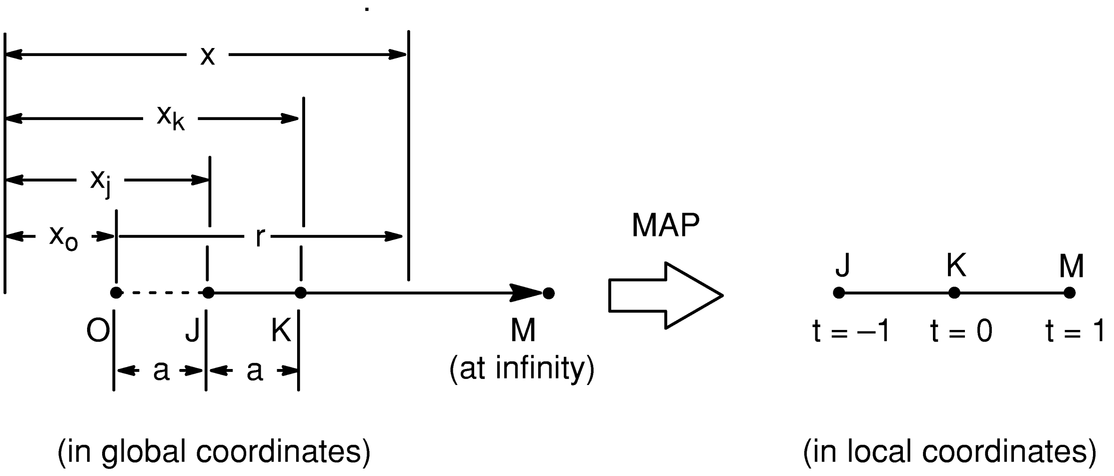


Figure 14.110–1 Global to Local Mapping of a One–Dimensional Infinite Element

The one–dimensional element may be thought of as one edge of the infinite element of Figure 14.110–2. It extends from node J, through node K to the point M at infinity and is mapped onto the parent element defined by the local coordinate system in the range –1 ≤ t ≤ 1.

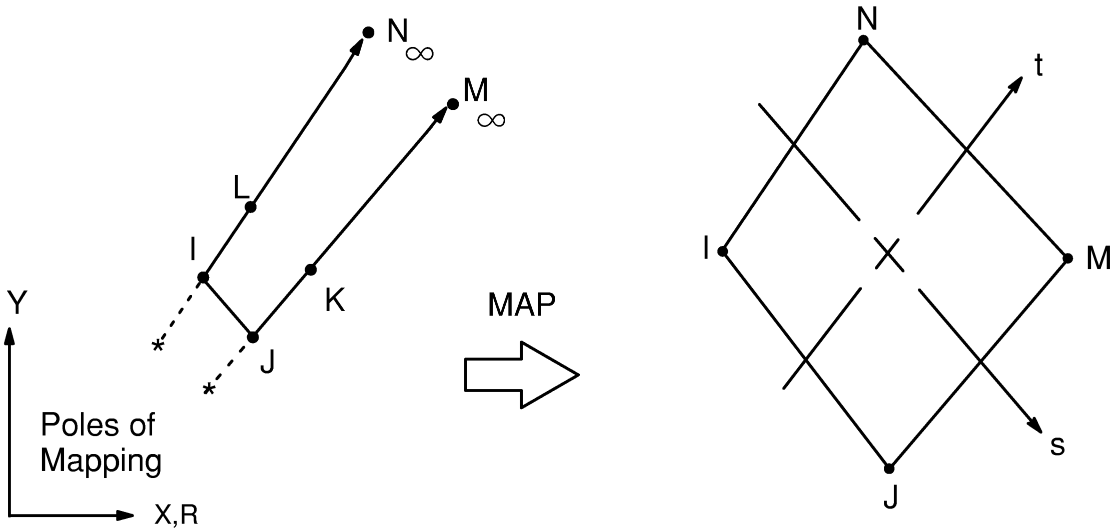


Figure 14.110–2 Mapping of 2–D Solid Infinite Element

The position of the “pole”, x_o, is arbitrary, and once chosen, the location of node K is defined by

$$x_K = 2x_J - x_o \tag{14.110–1}$$

The interpolation from local to global positions is performed as

$$x(t) = M_J(t)\ x_J + M_K(t)\ x_K \tag{14.110–2}$$

where:

$M_J(t) = -2t / (1 - t)$

$M_K(t) = 1 - M_J(t)$

Examining the above mapping, it can be seen that t = –1, 0, 1 correspond respectively to the global positions x = x_J, x_K, ∞, respectively.

The basic field variable, A (A_z for KEYOPT(1) = 0, VOLT for KEYOPT(1) = 1 or TEMP for KEYOPT(1) = 2) can be interpolated using standard shape functions, which when written in polynomial form becomes

$$A(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + - - - \tag{14.110–3}$$

Solving equation (14.110–2) for t yields

$$t = 1 - \frac{2a}{r} \tag{14.110–4}$$

where:

r = distance from the pole, O, to a general point within the element

a = $x_K - x_J$ as shown in Figure 14.110–2

Substituting (14.110–4) into (14.110–3) gives

$$A(t) = c_0 + \frac{c_1}{r} + \frac{c_2}{r^2} + \frac{c_3}{r^3} + - - - \tag{14.110–5}$$

Where c_0 = 0 is implied since the variable A is assumed to vanish at infinity.

The expression (14.110–5) is truncated at the quadratic (r^2) term in the present implementation. Expression (14.110–5) also shows the role of the pole position, O.

In 2–D (Figure 14.110–2) mapping is achieved by the shape function products. The mapping functions and the Lagrangian isoparametric shape functions for 2–D and axisymmetric 4 node quadrilaterals are given in Section 12.6.8. The shape functions for the nodes M and N are not needed as the field variable, A, is assumed to vanish at infinity.

14.110.2 Matrices

The matrices can be written as:

$$[K_e] = \int_{vol} [B]^T[D]\ [B]\ d(vol) \tag{14.110–6}$$

$$[C_e] = \int_{vol} C_c\ \{N\}\ \{N\}^T\ d(vol) \tag{14.110–7}$$

where: $\{N\}$ = shape functions given in Section 12.6.8
$[B]$ = defined later

1. AZ DOF (KEYOPT(1) = 0)

   $[K_e]$ = magnetic potential coefficient matrix

   $[D] = \frac{1}{\mu_o}\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$

   μ_o = magnetic permeability of free space (input on **EMUNIT** command)

   The infinite elements can be used in magnetodynamic analysis even though these elements do not compute mass matrices. This is because air has negligible conductivity.

2. VOLT DOF (KEYOPT(1) = 1)

   $[K_e]$ = electrical potential coefficient matrix

   $[D] = \begin{bmatrix}\epsilon_x & 0\\ 0 & \epsilon_y\end{bmatrix}$

   $\varepsilon_x, \varepsilon_y$ = electrical permitivity (input as PERX and PERY on **MP** command)

3. TEMP DOF (KEYOPT(1) = 2)

   $[K_e]$ = thermal conductivity matrix

   $[C_e]$ = specific heat matrix

   $[D] = \begin{bmatrix}k_x & 0\\ 0 & k_y\end{bmatrix}$

   k_x, k_y = thermal conductivities in the x and y direction (input as KXX and KYY on **MP** command)

   C_c = $\varrho\ C_p$

   ϱ = density of the fluid (input as DENS on **MP** command)

   C_p = specific heat of the fluid (input as C on **MP** command)

Although it is assumed that the nodal DOFs are zero at infinity, it is possible to solve thermal problems in which the nodal temperatures tend to some constant value, T_o, rather than zero. In that case, the temperature differential, θ (= $T - T_o$), may be thought to be posed as the nodal DOF. The actual temperature can then be easily found from $T = \theta + T_o$. For transient analysis, θ must be zero at infinity t > 0, where t is time. Neumann boundary condition is automatically satisfied at infinity. Note that with these infinite elements, meaningful steady–state thermal analysis can be performed only when the problem is driven by heat sources or sinks (**BF** or **BFE** command with *Lab* = HGEN).

The $\{B_i\}$ vectors of the [B] matrix in equation (14.110–6) contain the derivatives of N_i with respect to the global coordinates which are evaluated according to

$$\{B_i\} = \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial s} \\ \frac{\partial N_i}{\partial t} \end{Bmatrix} \qquad (14.110\text{–}8)$$

where: [J] = Jacobian matrix which defines the geometric mapping

[J] is given by

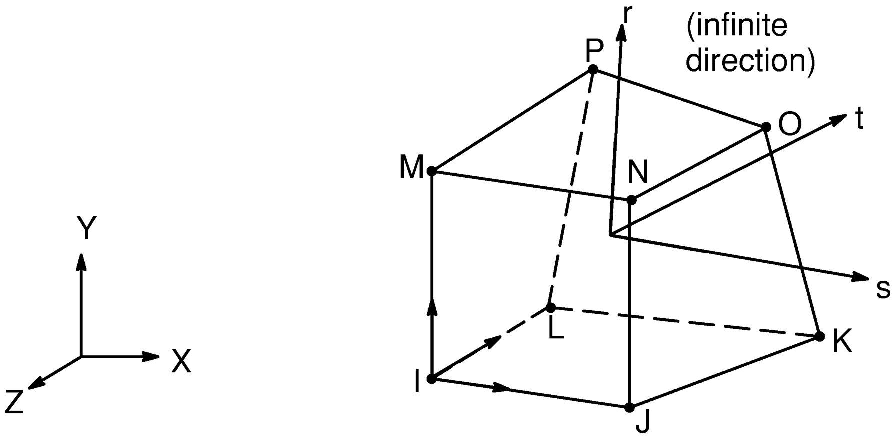
$$[J] = \sum_{i=1}^{4} \begin{bmatrix} \frac{\partial M_i}{\partial s} x_i & \frac{\partial M_i}{\partial s} y_i \\ \frac{\partial M_i}{\partial t} x_i & \frac{\partial M_i}{\partial t} y_i \end{bmatrix} \qquad (14.110\text{–}9)$$

The mapping functions [M] in terms of s and t are given in Section 12.6.8. The domain differential d(vol) must also be written in terms of the local coordinates, so that

$$d(vol) = dx\ dy = |\,J\,|\ ds\ dt \qquad (14.110\text{–}10)$$

Subject to the evaluation of $\{B_i\}$ and d(vol), which involves the mapping functions, the element matrices $[K_e]$ and $[C_e]$ may now be computed in the standard manner using Gaussian quadrature.

14.111   INFIN111 — 3–D Infinite Solid

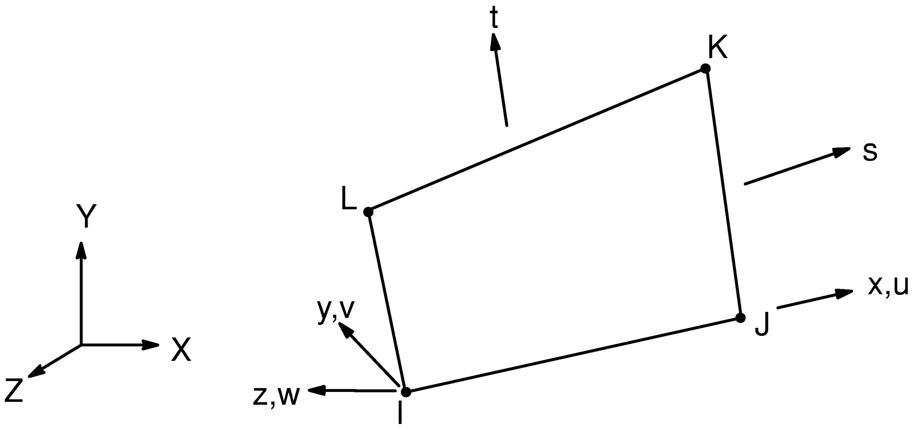


Matrix or Vector	Mapping and Shape Functions	Integration Points
Magnetic or Electrical Scalar Potential Coefficient Matrix or Thermal Conductivity Matrix	Equations (12.8.22–20), (12.8.22–21), (12.8.22–22), (12.8.22–33), (12.8.22–34) and (12.8.22–35)	2 x 2 x 2
Specific Heat Matrix	Equations (12.8.22–20), (12.8.22–33), (12.8.22–34) and (12.8.22–35)	2 x 2 x 2
Magnetic Vector Potential Coefficient Matrix	Equations (12.8.22–7), (12.8.22–8), (12.8.22–9), (12.8.22–33), (12.8.22–34) and (12.8.22–35)	2 x 2 x 2

14.111.1   Other Applicable Sections

See Section 14.110 for the theoretical development of infinite solid elements. The derivation presented in Section 14.110 for 2–D can be extended to 3–D in a straightforward manner.

14.115 INTER115 — 3–D Magnetic Interface



Matrix or Vector	Geometry	Shape Functions	Integration Points
Coefficient Matrix	Quad	Equation (12.5.8–7), (12.5.8–8), (12.5.8–9), and (12.5.8–22)	2 x 2
	Triangle	Equation (12.5.1–7), (12.5.1–8), (12.5.1–9), and (12.5.1–22)	1
Load Vector	Same as coefficient matrix		Same as coefficient matrix

14.115.1 Element Matrix Derivation

Problem Description

A general 3–D electromagnetics problem is schematically shown in Figure 14.115–1. The analysis region of the problem may be divided into three parts. Ω_1 is the region of conduction, in which the conductivity, σ, is not zero so that eddy currents may be induced. Ω_1 may also be a ferromagnetic region so that the permeability μ is much larger than that of the free space, μ_o. However, no source currents exist in Ω_1. Both Ω_2 and Ω_3 are regions free of eddy currents. There may be source currents present in these regions. A distinction is made between Ω_2 and Ω_3 to ensure that the scalar potential region, Ω_3, is single–connected and to provide an option to place the source currents in either the vector potential or the scalar potential region. Γ_B and Γ_H represent boundaries on which fluxes are parallel and normal respectively.

In Ω_1, due to the non–zero conductivity and / or high permeability, the magnetic vector potential together with the electric scalar potential are employed to model the influence of eddy currents. In Ω_2, only the magnetic vector potential is used. In Ω_3, the total magnetic field is composed of a reduced field which is derived from the magnetic reduced scalar potential, ϕ, and the field, H_s, which is computed using the Biot–Savart law.

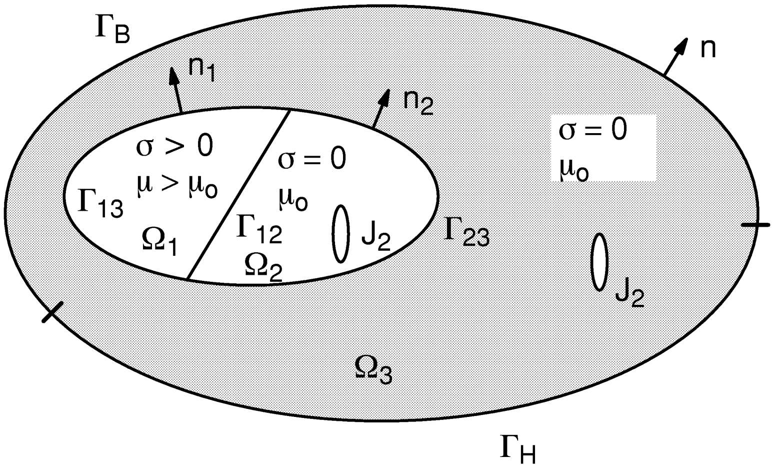


Figure 14.115–1 A General Electromagnetics Analysis Field and Its Component Regions

14.115.2 The A, V–A–ϕ Formulation

The equations relating the various field quantities are constituted by the following subset of Maxwell's equations with the displacement currents neglected.

$$\left.\begin{aligned} &\nabla \times \{H\} - \{J_s\} - \{J_e\} = \{0\} \\ &\nabla \times \{E\} + \left\{\frac{\partial B}{\partial t}\right\} = \{0\} \\ &\nabla \cdot \{B\} = 0 \end{aligned}\right\} \quad \text{in } \Omega_1 \qquad (14.115\text{–}1)$$

$$\left.\begin{aligned} &\nabla \times \{H\} = \{J_s\} \\ &\nabla \cdot \{B\} = 0 \end{aligned}\right\} \quad \text{in } \Omega_2 \cup \Omega_3 \qquad (14.115\text{–}2)$$

The constitutive relationships are:

$$\{B\} = [\mu]\ \{H\} \qquad (14.115\text{–}3)$$

The boundary and interface conditions, respectively, are:

$$\{B\}^T \cdot \{n\} = 0 \qquad \text{on } \Gamma_B \qquad (14.115\text{–}4)$$

$$\{H\}\ x\ \{n\} = \{0\} \qquad \text{on } \Gamma_H \qquad (14.115\text{–}5)$$

$$\left.\begin{array}{l} \{B_1\}^T \cdot \{n_1\} + \{B_2\} \cdot \{n_2\} = 0 \\ \{H_1\}\ x\ \{n_1\} + \{H_2\}\ x\ \{n_2\} = \{0\} \end{array}\right\} \text{on } \Gamma_{12}, \Gamma_{13}, \text{and } \Gamma_{23} \qquad (14.115\text{–}6)$$

Variables are defined in Section 5.1.

By introducing the magnetic vector potential, {A} (AX, AY, AZ), both in Ω_1 and Ω_2; the electric scalar potential V (VOLT) in Ω_1; and the generalized scalar potential ϕ_g (MAG) in Ω_3, the field quantities can be written in terms of various potentials as:

$$\{B\} = \nabla\ x\ \{A\} \qquad \text{in } \Omega_1 \text{ and } \Omega_2 \qquad (14.115\text{–}7)$$

$$\{E\} = -\left\{\frac{\partial A}{\partial t}\right\} - \nabla\ V \qquad \text{in } \Omega_1 \qquad (14.115\text{–}8)$$

$$\{H\} = \{H_s\} - \nabla\ \phi_g \qquad \text{in } \Omega_3 \qquad (14.115\text{–}9)$$

In order to make the solution of potential {A} unique, the Columb gauge condition is applied to define the divergence of {A} in addition to its curl.

Substituting equations (14.115–7) through (14.115–9) into the field equations and the boundary conditions equations (14.115–1) through (14.115–6) and using the Galerkin form of the method of weighted residual equations, the weak form of the differential equations in terms of the potentials {A}, V and ϕ_g can be obtained. Through some algebraic manipulations and by applying the boundary as well as interface conditions, respectively, the finite element equations may be written as:

$$\int_{\Omega_1+\Omega_2} \left(\left(\nabla \text{ x } [N_A]^T\right)^T [\nu] \left(\nabla \text{ x } \{A\}\right) + [\nu] \left(\nabla \cdot [N_A]^T\right)^T \left(\nabla \cdot \{A\}\right) + [\sigma] [N_A]^T \cdot \left\{\frac{\partial A}{\partial t}\right\} + [\sigma] [N_A]^T \cdot \nabla \frac{\partial v}{\partial t} \right) d\Omega - \int_{\Gamma_{13}+\Gamma_{23}} [N_A]^T \cdot \left(\nabla \phi_g \text{ x } \{n_3\}\right) d\Gamma \qquad (14.115\text{–}10)$$

$$= - \int_{\Gamma_{13}+\Gamma_{23}} [N_A]^T \cdot \left(\{H_s\} \text{ x } \{n_2\}\right) d\Gamma + \int_{\Omega_2} [N_A]^T \cdot \{J_2\} \, d\Omega$$

$$\int_{\Omega_1} \left([\sigma] \nabla \{N\} \cdot \left\{\frac{\partial A}{\partial t}\right\} + [\sigma] \nabla \{N\} \cdot \nabla \frac{\partial v}{\partial t} \right) d\Omega = 0 \qquad (14.115\text{–}11)$$

$$- \int_{\Omega_3} [\mu] \left(\nabla \{N\}\right)^T \cdot \nabla \phi_g \, d\Omega + \int_{\Gamma_{23}} \{N\} \{n_2\} \cdot (\nabla \text{ x } \{A\}) \, d\Gamma \qquad (14.115\text{–}12)$$

$$+ \int_{\Gamma_{13}} \{N\} \{n_1\} \cdot (\nabla \text{ x } \{A\}) \, d\Gamma = - \int_{\Omega_2} \left(\nabla \{N\}\right)^T \cdot [\mu] \{H_s\} \, d\Omega$$

where: $[N_A]$ = matrix of element shape functions for {A}

$\{N\}$ = vector of element shape function for both V and ϕ

v = related to the potential V as:

$$V = \frac{\partial v}{\partial t} \qquad (14.115\text{–}13)$$

A number of interface terms arise in the above equations because of the coupling of vector potential and scalar potential formulations across different regions. These are the terms that involve integration over the surface shared by two adjoining subregions and are given as:

$$I_1 = - \int_{\Gamma_{13}+\Gamma_{23}} [N_A] \cdot (\nabla \phi_g \text{ x } \{n_3\}) \, d\Gamma \qquad (14.115\text{–}14)$$

$$I_2 = - \int_{\Gamma_{13}+\Gamma_{23}} \{N\} \{n_3\} \cdot (\nabla \text{ x } \{A\}) \, d\Gamma \qquad (14.115\text{–}15)$$

$$I_3 = -\int_{\Gamma_{13}+\Gamma_{23}} [N_A] \cdot (\{H_s\} \times \{n_3\})\, d\Gamma \tag{14.115–16}$$

where: Γ_{ij} = surface at the interface of subregions Ω_i and Ω_j, respectively.

The term, I_3, contributes to the load vector while the terms, I_1 and I_2, contribute to the coefficient matrix. The asymmetric contributions of I_1 and I_2 to the coefficient matrix may be made symmetric following the procedure by Emson and Simkin(176). After some algebraic manipulations including applying the Stokes' theorem, we get

$$I_2 = I_{21} + I_{22} \tag{14.115–17}$$

$$I_{21} = -\int_{\Gamma_{13}+\Gamma_{23}} (\nabla \{N\} \times \{n_3\}) \cdot \{A\}\, d\Gamma \tag{14.115–18}$$

$$I_{22} = -\oint_{\Gamma_{13}+\Gamma_{23}} \{N\}\, \{A\} \cdot d\bar{\ell} \tag{14.115–19}$$

It is observed from equation (14.115–17) that the integrals represented by I_1 and I_2 are symmetric if the condition $I_{22} = 0$ is satisfied. The integral given by I_{22} is evaluated along a closed path lying on the interface. If the interface lies completely inside the region of the problem, the integrals over the internal edges will cancel each other; if the integral path is on a plane of symmetry, the tangential component of {A} will be zero, so the integral will be vanish; and if the integral path is on the part of the boundary where the scalar potential is prescribed, the terms containing N will be omitted and the symmetry of the matrix will be ensured. Therefore, the condition that ensures symmetry can usually be satisfied. Even if, as in some special cases, the condition can not be directly satisfied, the region may be remeshed to make the interface of the vector and scalar potential regions lie completely inside the problem domain. Thus, the symmetry condition can be assumed to hold without any loss of generality.

Replacing the vector and scalar potentials by the shape functions and nodal degrees of freedom as described by equations (14.115–20) through (14.115–23),

$$\{A\} = [N_A]^T \{A_e\} \tag{14.115–20}$$

$$\left\{\frac{\partial A}{\partial t}\right\} = [N_A]^T \{\dot{A}_e\} \tag{14.115–21}$$

$$\phi_g = \{N\}^T \{\phi_e\} \tag{14.115–22}$$

$$V = \frac{\partial v}{\partial t} = \{N\}^T\{V_e\} \tag{14.115–23}$$

the above manipulations finally result in the following set of finite element equations:

$$\begin{aligned}
&\int_{\Omega_1} \left[\left(\nabla \text{ x } [N_A]^T\right)^T [\nu]\ (\nabla \text{ x } [N_A]^T) + [\nu]\ \nabla \cdot [N_A]^T\ \nabla \cdot [N_A]^T\right] d\Omega\ \{A_e\} \\
&\quad + \int_{\Omega_1} [\sigma]\ [N_A]^T \cdot [N_A]\ d\Omega\ \{\dot{A}_e\} + \int_{\Omega_1} [\sigma]\ [N_A]^T \cdot \nabla\ \{N\}\ d\Omega\ \{V_e\} \\
&\quad - \int_{\Gamma_{13}} [N_A]^T \cdot \left(\nabla\ \{N\} \text{ x } \{n_3\}\right) d\Gamma\{\phi_e\} = -\int_{\Gamma_{13}} [N_A]^T\ \cdot [N_A] \text{ x } \{n_3\}\ d\Gamma\ \{H_s\}
\end{aligned} \tag{14.115–24}$$

$$\int_{\Omega_1} [\sigma]\ \nabla\ \{N\}^T\ \cdot\ [N_A]\ d\Omega\ \{\dot{A}_e\} + \int_{\Omega_1} [\sigma]\ \nabla\ \{N\}^T\ \cdot\ \nabla\ \{N\}\ d\Omega\ \{V_e\} = 0 \tag{14.115–25}$$

$$\begin{aligned}
&\int_{\Omega_2} \left[[\nu]\ (\nabla \text{ x } [N_A])^T \cdot (\nabla \text{ x } [N_A]) + [\nu]\ \nabla \cdot [N_A]\ \nabla \cdot [N_A]\right] d\Omega\ \{A_e\} \\
&\quad - \int_{\Gamma_{23}} [N_A]^T \cdot (\nabla\ \{N\} \text{ x } \{n_3\})\ d\Gamma\ \{\phi_e\} \\
&\quad = -\int_{\Gamma_{23}} [N_A]^T \cdot [N_A] \text{ x } \{n_3\}\ d\Gamma\{H_s\} + \int_{\Omega_2} [N_A]^T[N_A]\ d\Omega\ \{J_2\}
\end{aligned} \tag{14.115–26}$$

$$\begin{aligned}
&-\int_{\Omega_3} [\mu]\ \nabla\ \{N\}^T \cdot \nabla\ \{N\}\ d\Omega\ \{\phi_e\} - \int_{\Gamma_{13}+\Gamma_{23}} (\nabla\ \{N\} \text{ x } \{n_3\})^T\ \cdot\ [N_A]\ d\Gamma\ \{A_e\} \\
&\quad = -\int_{\Omega_2} [\mu]\ \nabla\ \{N\}^T[N_A]\ d\Omega\ \{H_s\}
\end{aligned} \tag{14.115–27}$$

Equations (14.115–24) through (14.115–27) represent a symmetric system of equations for the entire problem.

The interface elements couple the vector potential and scalar potential regions, and therefore have AX,AY, AZ and MAG degrees of freedom at each node. The coefficient matrix and the load vector terms in equations (14.115–24) through (14.115–27) are computed in the magnetic vector potential elements (SOLID97), the scalar potential elements SOLID96, SOLID98 with KEYOPT(1) = 10, or SOLID5 with KEYOPT(1) = 10) and the interface elements (INTER115). The only terms in these equations that are computed in the interface elements are given by:

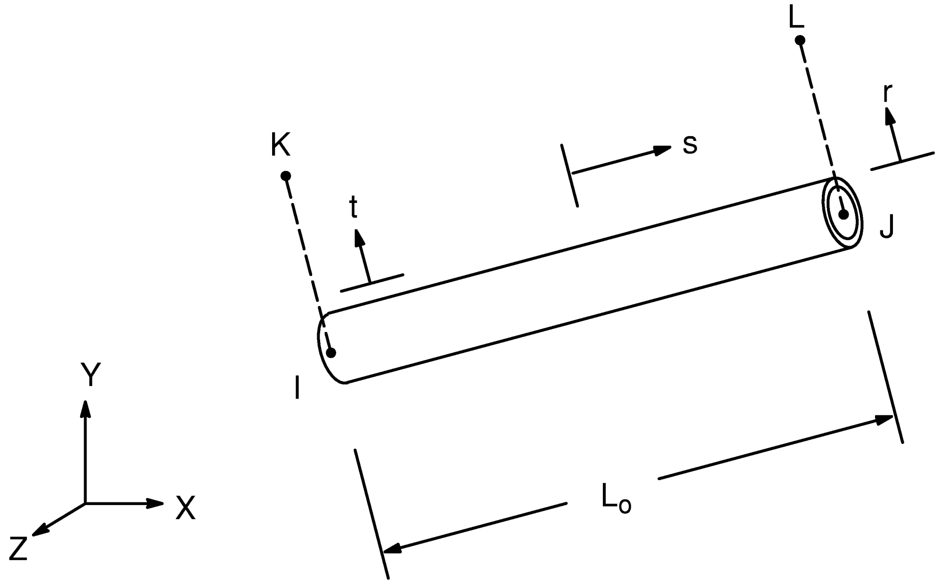
Coefficient Matrix:

$$[K] = -\int_{\Gamma_{13}+\Gamma_{23}} [N_A]^T \cdot (\nabla \{N\} \text{ x } \{n_3\})\, d\Gamma - \int_{\Gamma_{13}+\Gamma_{23}} (\nabla \{N\} \text{ x } \{n_3\})^T \cdot [N_A]\, d\Gamma \qquad (14.115\text{–}28)$$

Load Vector:

$$\{F\} = -\int_{\Gamma_{13}+\Gamma_{23}} [N_A]^T \cdot ([N_A] \text{ x } \{n_3\})\, d\Gamma \qquad (14.115\text{–}29)$$

14.116 FLUID116 — Coupled Thermal–Fluid Pipe



Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Conductivity Matrix	Between nodes I and J	Equation (12.2.1–20)	None
	Convection between nodes I and K and between nodes J and L (optional)	None	None
Pressure Conductivity Matrix	Between nodes I and J	Equation (12.2.1–19)	None

Matrix or Vector	Geometry	Shape Functions	Integration Points
Specific Heat Matrix	Equation (12.2.1–20)		None
Heat Generation Load Vector	Equation (12.2.1–20)		None

14.116.1 Assumptions and Restrictions

Transient pressure and compressibility effects are also not included.

14.116.2 Combined Equations

The thermal and pressure aspects of the problem have been combined into one element having two different types of working variables: temperatures and pressures. The equilibrium equations for one element have the form of:

$$N_c \begin{bmatrix} [C^t] & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\dot{T}\} \\ \{0\} \end{Bmatrix} + N_c \begin{bmatrix} [K^t] & [0] \\ [0] & [K^p] \end{bmatrix} \begin{Bmatrix} \{T\} \\ \{P\} \end{Bmatrix} = \begin{Bmatrix} \{Q\} \\ \{w\} \end{Bmatrix} + N_c \begin{Bmatrix} \{Q^g\} \\ \{H\} \end{Bmatrix} \qquad (14.116\text{–}1)$$

where:
$[C^t]$ = specific heat matrix for one channel
$\{T\}$ = nodal temperature vector
$\{\dot{T}\}$ = vector of variations of nodal temperature with respect to time
$\{P\}$ = nodal pressure vector
$[K^t]$ = thermal conductivity matrix for one channel (includes effects of convection and mass transport)
$[K^p]$ = pressure conductivity matrix for one channel
$\{Q\}$ = nodal heat flow vector (input quantity HEAT on **F** command)
$\{w\}$ = nodal fluid flow vector (input quantity FLOW on **F** command)
$\{Q^g\}$ = internal heat generation vector for one channel
$\{H\}$ = gravity and pumping effects vector for one channel
N_c = number of parallel flow channels (input quantity N_c on **R** command)

14.116.3 Thermal Matrix Definitions

Specific Heat Matrix

The specific heat matrix is a diagonal matrix with each term being the sum of the corresponding row of a consistent specific heat matrix:

$$\left[C^t\right] = A_c \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (14.116–2)$$

where: $A_c = \dfrac{\rho_u C_p A L_o}{2}$

ρ_u = effective density

$$= \begin{cases} \rho & \text{if } R_{gas} = 0.0 \\ \text{or} & \\ \dfrac{P}{R_{gas}\ T_{abs}} \quad \text{(ideal gas law)} & \text{if } R_{gas} \neq 0.0 \end{cases}$$

ρ = mass density (input as DENS on **MP** command

P = pressure (average of first two nodes)

T_{abs} = T + TOFFST = absolute temperature

T = temperature (average of first two nodes)

TOFFST = offset temperature (input on **TOFFST** command)

C_p = specific heat (input as C on **MP** command)

A = flow cross–sectional area (input quantity A on **R** command)

L_o = length of member (distance between nodes I and J)

R_{gas} = gas constant (input quantity R_{gas} on **R** command)

Thermal Conductivity Matrix

The thermal conductivity matrix is given by:

$$\left[K^t\right] = \begin{bmatrix} B_1 + B_2 - B_4 & -B_1 + B_4 & -B_2 & 0 \\ -B_1 - B_5 & B_1 + B_3 + B_5 & 0 & -B_3 \\ -B_2 & 0 & B_2 & 0 \\ 0 & -B_3 & 0 & B_3 \end{bmatrix} \qquad (14.116–3)$$

where: $B_1 = \dfrac{AK_s}{\ell}$

K_s = thermal conductivity (input as KXX on **MP** command)

B_2 = $h\ A_I$

h = film coefficient (defined below)

A_I = lateral area of pipe associated with end I (input quantity $(A_n)_I$ on **R** command) (defaults to $\frac{\pi D\ell}{2}$)

B_3 = $h\ A_J$

A_j = lateral area of pipe associated with end J (input quantity $(A_n)_J$ on **R** command) (defaults to $\frac{\pi D\ell}{2}$)

D = hydraulic diameter (input quantity D on **R** Command)

$$B_4 = \begin{cases} wC_p \text{ if flow is from node J to node I} \\ 0 \text{ if flow is from node I to node J} \end{cases}$$

$$B_5 = \begin{cases} wC_p \text{ if flow is from node I to node J} \\ 0 \text{ if flow is from node J to node I} \end{cases}$$

w = mass fluid flow rate in the element

w may be determined by the program or may be input by the user:

$$w = \begin{cases} \text{computed from previous iteration} & \text{if pressure is a degree of freedom} \\ \text{or} & \\ \text{input as VAL1 on } \textbf{SFE,,,}\text{HFLUX command} & \text{if pressure is not a degree of freedom} \end{cases} \tag{14.116–4}$$

The above definitions of B_4 and B_5, as used by equation NO TAG, cause the energy change due to mass transport to be lumped at the outlet node.

The film coefficient h is defined as:

$$h = \begin{cases} \text{input as HF on } \textbf{MP} \text{ command} & \text{if KEYOPT(4) = 0} \\ \text{or} & \\ \frac{Nu\ K_s}{D} & \text{if KEYOPT(4) = 1} \\ \text{or} & \\ \text{defined by TB, HFLM table} & \text{if KEYOPT(4) = 2, 3, or 4} \\ \text{or} & \\ \text{defined by user programmable feature User116Hf} & \text{if KEYOPT(4) = 5} \end{cases} \tag{14.116–5}$$

Nu, the Nusselt number, is defined for KEYOPT(4) = 1 as:

$$Nu = N_1 + N_2 \, Re^{N_3} \, Pr^{N_4} \tag{14.116–6}$$

where: N_1 to N_4 = input on **R** commands

Re = $\frac{w\ D}{\mu A}$

= Reynold's number

μ = viscosity (input as VISC on **MP** command)

Pr = $\frac{C_p\ \mu}{K_s}$

= Prandtl number

A common usage of equation (14.116–6) is the Dittus–Boelter correlation for fully developed turbulent flow in smooth tubes (Holman(55)):

$$Nu = 0.023 \, Re^{0.8} \, Pr^{a} \tag{14.116–7}$$

where: $a = \begin{cases} 0.4 & \text{for heating} \\ 0.3 & \text{for cooling} \end{cases}$

Heat Generation Load Vector

The internal heat generation load vector is due to both average heating effects and viscous damping:

$$\{Q^g\} = \begin{Bmatrix} Q_n \\ Q_n \\ 0 \\ 0 \end{Bmatrix} \tag{14.116–8}$$

where: $Q_n = \frac{L_o}{2}\left(A\dddot{q} + \pi \, V_{DF} \, C_{ver} \, F \, \mu \, v^2\right)$

$\dddot{q}$ = internal heat generation rate per unit volume

= input on **BF** or **BFE** commands

V_{DF} = viscous damping multiplier (input on **RMORE** command)

C_{ver} = units conversion factor (input on **RMORE** command)

F = flow type factor

$= \begin{cases} 8.0 & \text{if} & Re \leq 2500.0 \\ 0.21420 & \text{if} & Re > 2500.0 \end{cases}$

v = average velocity

The expression for the viscous damping part of Q_n is based on fully developed laminar flow.

14.116.4 Fluid Equations

Bernoulli's equation is:

$$Z_I + \frac{P_I}{\gamma} + \frac{v_I^2}{2g} + \frac{P_{PMP}}{\gamma} = Z_J + \frac{P_J}{\gamma} + \frac{v_J^2}{2g} + C_L \frac{v_a^2}{2g} \quad (14.116\text{–}9)$$

where:

Z = coordinate in the negative acceleration direction

P = pressure

γ = ρg

g = acceleration of gravity

v = velocity

P_{PMP} = pump pressure (input as P_p on **R** command)

C_L = loss coefficient

The loss coefficient is defined as:

$$C_L = \frac{f\ell}{D} + \beta\ell \quad (14.116\text{–}10)$$

where:

$$\beta = \begin{cases} \dfrac{f\ell_a}{D\ell} & \text{if KEYOPT(8) } = 0 \\ \text{or} & \\ \dfrac{k}{\ell} & \text{if KEYOPT(8) } = 1 \end{cases}$$

= extra flow loss factor

ℓ_a = additional length to account for extra flow losses (input as L_a on **R** command)

k = loss coefficient for typical fittings (input as K on **R** command)

f = Moody friction factor, defined below:

For the first iteration of the first load step,

$$f = \begin{cases} f_m & \text{if } f_m \neq 0.0 \\ 1.0 & \text{if } f_m = 0.0 \end{cases} \quad (14.116\text{–}11)$$

where: f_m = input as MU on **MP** command

For all subsequent iterations

$$f = \begin{cases} f_x & \text{if KEYOPT(7)} = 0 \\ f_m & \text{if KEYOPT(7)} = 1 \\ \text{defined by TB, FLOW} & \text{if KEYOPT(7)} = 2, 3 \end{cases} \qquad (14.116\text{–}12)$$

The smooth pipe empirical correlation is:

$$f_x = \begin{cases} \dfrac{64}{Re} & 0 < Re \le 2500 \\ \text{or} & \\ \dfrac{0.316}{(Re)^{1/4}} & 2500 < Re \end{cases} \qquad (14.116\text{–}13)$$

Bernoulli's equation (14.116–9) may be simplified for this element, since the cross–sectional area of the pipe does not change. Therefore, continuity requires all velocities not to vary along the length. Hence $v_1 = v_2 = v_a$, so that Bernoulli's equation (14.116–9) reduces to:

$$Z_I - Z_J + \frac{P_I - P_J}{\gamma} + \frac{P_{PMP}}{\gamma} = C_L \frac{v^2}{2g} \qquad (14.116\text{–}14)$$

Writing equation (14.116–14) in terms of mass flow rate (w = ρAv), and rearranging terms to match the second half of equation (14.116–1),

$$\frac{2\rho A^2}{C_L} (P_I - P_J) = w^2 + \frac{2g\rho^2 A^2}{C_L} \left(-Z_I + Z_J - \frac{P_{PMP}}{\gamma}\right) \qquad (14.116\text{–}15)$$

Since the pressure drop ($P_I - P_J$) is not linearly related to the flow (w), a nonlinear solution will be required. As the w term may not be squared in the solution, the square root of all terms is taken in a heuristic way:

$$A \sqrt{\frac{2\rho}{C_L}} \sqrt{P_I - P_J} = w + A \sqrt{\frac{2\rho}{C_L}} \left((-Z_I + Z_J) \rho g - P_{PMP}\right) \qquad (14.116\text{–}16)$$

Defining:

$$B_c = A\sqrt{\frac{2\rho}{C_L}} \tag{14.116–17}$$

and

$$P_L = (-Z_I + Z_J)\,\rho g - P_{PMP} \tag{14.116–18}$$

equation (14.116–16) reduces to:

$$B_c\sqrt{P_I - P_J} = w + B_c P_L \tag{14.116–19}$$

Hence, the pressure conductivity matrix is based on the term $\frac{B_c}{\sqrt{P_I - P_J}}$ and the pressure (gravity and pumping) load vector is based on the term $B_c P_L$.

Two further points:

1. B_c is generalized as:

$$B_c = \begin{cases} A\sqrt{\dfrac{2\rho}{C_L}} & \text{if KEYOPT(6) = 0} \\ \text{input as C on } \textbf{R} \text{ command,} & \text{if KEYOPT(6) = 1} \\ \text{defined by TB,FCON} & \text{if KEYOPT(6) = 2 or 3} \\ \text{defined by user programmable} & \\ \quad\text{feature, User116Cond} & \text{if KEYOPT(6) = 4} \end{cases} \tag{14.116–20}$$

2. $(-Z_I + Z_J)g$ is generalized as:

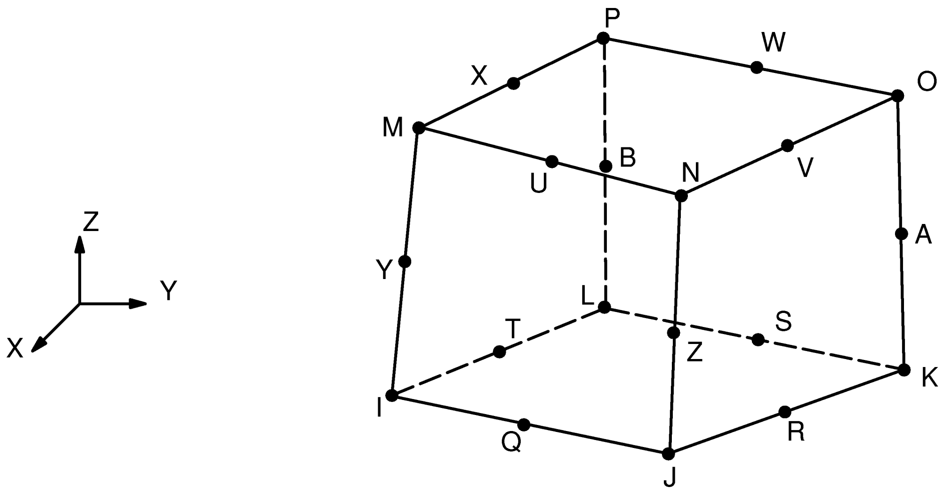
$$(-Z_I + Z_J)g = \{\Delta x\}^T\{a_t\} \tag{14.116–21}$$

where:

$\{\Delta x\}$ = vector from node I to node J

$\{a_t\}$ = translational acceleration vector which includes effects of angular velocities (see section 15.1)

14.117 SOLID117 — 3–D Magnetic Edge



Matrix or Vector	Shape Functions	Integration Points
Edge Formulation of Magnetic Vector Potential Coefficient Matrix	Equations (12.8.18–7), (12.8.18–8), and (12.8.18–9) for magnetic vector potential; Equations (12.9–26) thru (12.9–37) for edge–flux	2 x 2 x 2
Electric Potential Coefficient Matrix	Equations (12.9–18) thru (12.9–25)	2 x 2 x 2
Load Vector of Magnetism due to Source Currents, Permanent Magnets, and Applied Currents	Same as coefficient matrix	2 x 2 x 2

Load Type	Distribution
Current Density, Voltage Load and Phase Angle Distribution	Trilinearly varying over the thru element

References: Biro et al(120), Gyimesi and Ostergaard(201), Gyimesi and Ostergaard (221), Ostergaard and Gyimesi(222), Ostergaard and Gyimesi(223), Preis(203), Nedelec(204), Kameari(206), Jin(207)

14.117.1 Other Applicable Sections

The following sections describe the theorem of the magnetic edge element using edge flux DOF:

- 5.1.2 Magnetic Vector Potential
- 5.1.4 Edge Flux Degrees of Freedom
- 5.1.5 Harmonic Analysis Using Complex Formalism
- 5.2.2 Magnetic Vector Potential
- 12.9 Electromagnetic Edge Elements
- 13.1 Integration Point Locations

Section 4.117 of the *Elements Reference* serves as a reference user guide and is provided in the Electromagnetic Field Analysis Part of the Analysis Guides. Chaper 6, 7 and 8 describe respectively static, harmonic and transient analyses by magnetic element SOLID117.

14.117.2 Matrix formulation of low frequency edge element and tree gauging

This low frequency electromagnetic element eliminates the shortcomings of nodal vector potential formulation discussed in section 5.1.5. The pertinent shape functions are presented in section 12.9.

The column vector of nodal vector potential components in SOLID97 is denoted by $\{A_e\}$, that of time integrated scalar potentials by $\{\nu_e\}$. (See definitions in section 5.2.2.) The vector potential, {A}, can be expressed by linear combinations of both corner node vector potential DOFs, $\{A_e\}$, as in SOLID97, and side node edge–flux DOFs, {AZ}. For this reason there is a linear relationship between $\{A_e\}$ and {AZ}.

$$\{A_e\} = [T^R]\ \{A^Z\} \tag{14.117–1}$$

where: $[T^R]$ = transformation matrix. Relationship (14.117–1) allows to compute the stiffness and damping matrices as well as load vectors of SOLID117 in terms of SOLID97.

Substituting (14.117–1) into (5.2–25) and (5.2–26) provides

$$\{A^Z\}^T\Big([K^{ZZ}]\{A_z\} + [K^{ZV}]\{\nu_e\} + [C^{ZZ}]d/dt\{A_z\} + [C^{ZV}]d/dt\{\nu_e\} - \{J^Z\}\Big) = 0 \quad (14.117–2)$$

$$\{\nu_e\}^T\Big([K^{VZ}]\{A_z\} + [K^{VV}]\{\nu_e\} + [C^{VZ}]d/dt\{A_z\} + [C^{VV}]d/dt\{\nu_e\} - \{1^t\}\Big) = 0 \quad (14.117–3)$$

where:

$$[K^{ZZ}] = [T^R]^T\ [K^{AA}]\ [T^R] \quad (14.117–4)$$

$$[C^{ZZ}] = [T^R]^T\ [C^{AA}]\ [T^R] \quad (14.117–5)$$

$$[K^{ZV}] = [T^R]^T\ [K^{AA}]\ [T^R] \quad (14.117–6)$$

$$[C^{ZV}] = [T^R]^T\ [C^{AV}]\ [T^R] \quad (14.117–7)$$

$$[J^Z] = [T^R]^T\ [J^A] \quad (14.117–8)$$

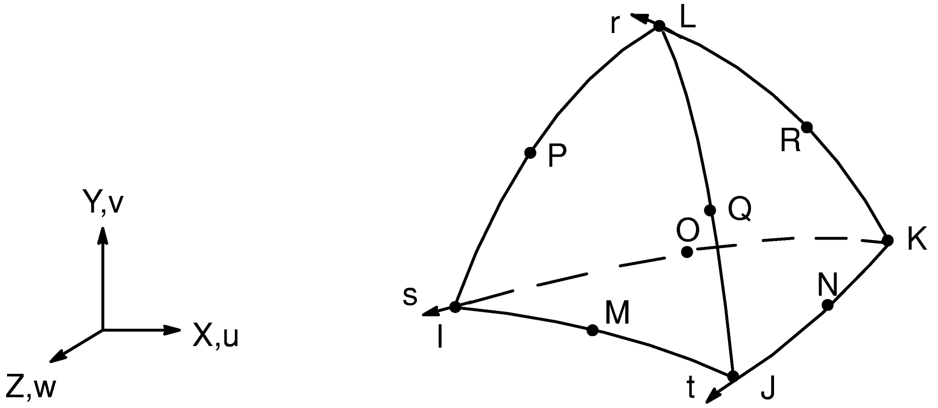
$$[K^{VZ}] = [K^{VA}]\ [T^R] \quad (14.117–9)$$

$$[C^{VZ}] = [C^{VA}]\ [T^R] \quad (14.117–10)$$

Equations (14.117–2) and (14.117–3) need to be properly gauged to obtain uniqueness. For more on this topic see for example Preiss et al(203). SOLID117 applies a tree gauging algorithm. It considers the relationship between nodes and edges by a topological graph. A fundamental tree of a graph is an assemly of edges constituting a path over which there is one and only one way between different nodes. It can be shown that the edge–flux DOFs over the fundamental tree can be set to zero providing uniqueness without violating generality.

The tree gauging applied is transparent to most users. At the solution phase the extra constraints are automatically supplied over the tree edges on top of the set of constraints provided by users. After equation solution, the extra constraints are removed. This method is good for most of the practical problems. However, expert users may apply their own gauging for specific problems by turning the tree gauging off by the command, GAUGE,OFF.

14.119 HF119 — 3–D High–Frequency Tetrahedral Solid



Matrix or Vector	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness, Mass and Damping Matrices	Equations (12.8.2–1), (12.8.2–2) and (12.8.2–3)	Polynomial variable in order of 1	Variable
Surface PORT, INF, IMPD, SHLD Load Vectors	Equations (12.5.2–1) and (12.5.2–2)	Polynomial variable in order of 1	Variable

Load Type	Distribution
Surface Loads	Linear across each face

14.119.1 Other Applicable Sections

Section 5.6 describes the derivation of element matrices and load vectors as well as results evaluations.

14.119.2 Solution Shape Functions – H (curl) Conforming Elements

HF119, along with HF120, uses a set of vector solution functions, which belong to the finite element functional space, H(curl), introduced by Nedelec(158). These vector functions have, among others, a very useful property, i.e., they possess tangential continuity on the boundary between two adjacent elements. This property fits naturally the need of HF119 to solve the electric field E based on the Maxwell's equations, since E is only tangentially continuous across material interfaces.

Similar to HF120 as discussed in section 14.120.2, the electric field E is approximated by:

$$\vec{E}(\vec{r}) = \sum_{i=1}^{N_v} E_i \, \vec{W}_i \, (\vec{r}) \qquad (14.119\text{–}1)$$

where:
W_i = vector shape functions defined in the tetrahedral element
E_i = covariant components of E at proper locations (AX DOFs)
N_v = number of vector functions

Refer to the tetrahedral element shown at the beginning of this subsection. The geometry of the element is represented by the following mapping:

$$\vec{r} = \sum_{j=1}^{10} N_j \, (L_1, \; L_2, \; L_3, \; L_4) \; \vec{r}_j \qquad (14.119\text{–}2)$$

where:
N_j = nodal shape functions
L_j = volume coordinates
r_j = nodal coordinates

Consider the local oblique coordinate system (s, t, r) based on node K. A set of unitary vectors can be defined as:

$$\vec{a}_1 = \frac{\partial \vec{r}}{\partial L_1} - \frac{\partial \vec{r}}{\partial L_3} \qquad \vec{a}_2 = \frac{\partial \vec{r}}{\partial L_2} - \frac{\partial \vec{r}}{\partial L_3} \qquad \vec{a}_3 = \frac{\partial \vec{r}}{\partial L_4} - \frac{\partial \vec{r}}{\partial L_3} \qquad (14.119\text{–}3)$$

These defines subsequently the gradients of the four volume coordinates:

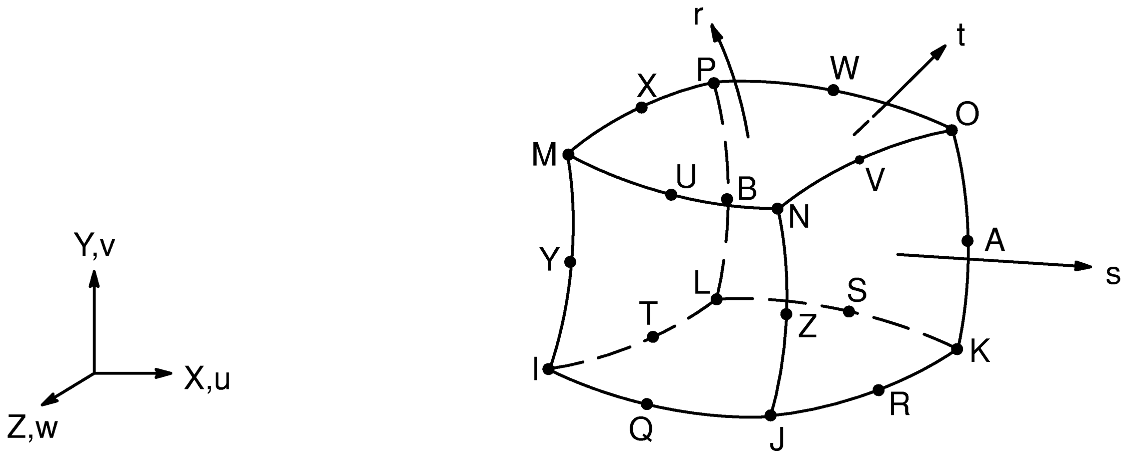
$$\begin{cases} \nabla L_1 = \dfrac{\vec{a}_2 \times \vec{a}_3}{J_t} \quad \nabla L_2 = \dfrac{\vec{a}_3 \times \vec{a}_1}{J_t} \\ \nabla L_4 = \dfrac{\vec{a}_1 \times \vec{a}_2}{J_t} \quad \nabla L_3 = - \nabla L_1 - \nabla L_2 - \nabla L_4 \\ J_t = \vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3 \end{cases} \tag{14.119–4}$$

The vector shape functions for the first order tetrahedral element can be conveniently defined as

$$\vec{W}_{ij} = L_i \nabla L_j - L_j \nabla L_i \qquad i, j = I, J, K, L \qquad i \neq j \tag{14.119–5}$$

The first order element is often referred to as the Whitney element(Whitney (208)).

14.120 HF120 — High–Frequency Brick Solid



Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness, Mass and Damping Matrices	Brick	Equations (12.8.20–1), (12.8.20–2), and (12.8.20–3)	Polynomial variable in order from 1 to 2	Variable
	Wedge	Equations (12.8.14–1), (12.8.14–2), and (12.8.14–3)	Polynomial variable in order from 1 to 2	Variable

Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Surface PORT, INF, IMPD, SHLD Load Vectors	Quad	Equations (12.5.10–1) and (12.5.10–2)	Polynomial variable in order from 1 to 2	Variable
	Triangle	Equations (12.5.2–1) and (12.5.2–2)	Polynomial variable in order from 1 to 2	Variable

Load Type	Distribution
Surface Loads	Bilinear across each face

14.120.1 Other Applicable Sections

Section 5.6 describes the derivation of element matrices and load vectors as well as result evaluations.

14.120.2 Solution Shape Functions – H(curl) Conforming Element

HF120 uses a set of vector solution functions, which belong to the finite element functional space, H(curl), introduced by Nedelec(158). These vector functions have, among others, a very useful property, i.e., they possess tangential continuity on the boundary between two adjacent elements. This property fits naturally the need of HF120 to solve the electric field E based on the Maxwell's equations, since E is only tangentially continuous across material interfaces.

Let W_i,, i=1,,,N_V be such vector shape functions defined in the brick element. The electric field E is approximated by:

$$\vec{E}(\vec{r}) = \sum_{i=1}^{N_v} E_i \; \vec{W}_i \; (\vec{r}) \tag{14.120–1}$$

where:

$\vec{r}$ = position vector within the element.

N_V = number of vector shape functions

E_i = covariant components of E

In the following, three aspects in equation (14.120–1) are explained, i.e., how to define the W_i functions, how to choose the number of functions N_v, and what are the physical meanings of the associated expansion coefficients E_i. Recall that coefficients E_i are represented by the AX DOF in HF120.

To proceed, a few geometric definitions associated with an oblique coordinate system are necessary. Refer to the brick element shown at the beginning of this subsection. The geometry of the element is determined by the following mapping:

$$\vec{r} = \sum_{j=1}^{20} N_i\ (s,\ t,\ r)\ \vec{r}_i \tag{14.120–2}$$

where: N_i = standard isoparametric shape functions

$\vec{r}_i$ = global coordinates for the 20 nodes.

Based on the mapping, a set of unitary basis vectors can be defined (Stratton(209)):

$$\vec{a}_1 = \frac{\partial \vec{r}}{\partial s} \qquad \vec{a}_2 = \frac{\partial \vec{r}}{\partial t} \qquad \vec{a}_3 = \frac{\partial \vec{r}}{\partial r} \tag{14.120–3}$$

These are simply tangent vectors in the local oblique coordinate system (s, t, r). Alternatively, a set of reciprocal unitary basis vectors can also be defined:

$$\left\{ \begin{array}{ll} \vec{a}^{\,1} = \dfrac{\vec{a}_2 \times \vec{a}_3}{J} & \vec{a}^{\,2} = \dfrac{\vec{a}_3 \times \vec{a}_1}{J} \\ \vec{a}^{\,3} = \dfrac{\vec{a}_1 \times \vec{a}_2}{J} & J = \vec{a}_1 \cdot \vec{a}_2 \times \vec{a}_3 \end{array} \right. \tag{14.120–4}$$

A vector F may be represented using either set of basis vectors:

$$\vec{F} = \sum_{i=1}^{3} f^i\ \vec{a}_i = \sum_{j=1}^{3} f_j\ \vec{a}^{\,j} \tag{14.120–5}$$

where f_j = covariant components

f^i = contravariant components.

Given the covariant components of a vector F, its curl is found to be

$$\nabla \times \vec{F} = \frac{1}{J} \begin{vmatrix} \vec{a}_1 & \vec{a}_2 & \vec{a}_3 \\ \dfrac{\partial}{\partial s} & \dfrac{\partial}{\partial t} & \dfrac{\partial}{\partial r} \\ f_1 & f_2 & f_3 \end{vmatrix} \tag{14.120–6}$$

Having introduced the above geometric concepts, appropriate vector shape functions for the brick element are defined next. For the first order element (KEYOPT(1) = 1), there is one function associated with each edge:

$$\vec{w}_i = \begin{cases} \phi_i \, \vec{a}^{\,1}, & i = Q,\ S,\ U,\ W \\ \phi_i \, \vec{a}^{\,2}, & i = R,\ T,\ V,\ X \\ \phi_i \, \vec{a}^{\,3}, & i = Y,\ Z,\ A,\ B \end{cases} \qquad (14.120\text{–}7)$$

where: ϕ_i = scalar functions.

Therefore, N_v=12.

Now consider the second order brick (KEYOPT(1)=2). There are two functions defined for each edge. For example for node Q:

$$\vec{w}_i^{(1)} = \phi_i^{(1)} \, \vec{a}^{\,1}, \qquad \vec{w}_i^{(2)} = \phi_i^{(2)} \, \vec{a}^{\,1}. \qquad (14.120\text{–}8)$$

In addition, there are two functions defined associated with each face of the brick. For example, for the face MNOP (r = 1):

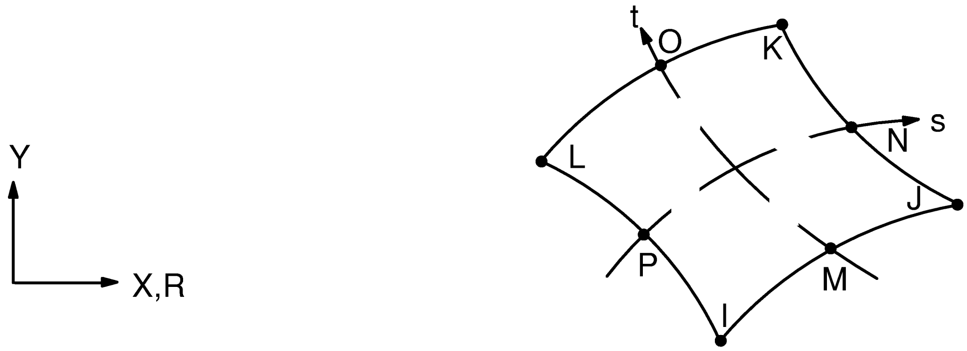
$$\vec{w}_f^{(1)} = \phi_f^{(1)} \, \vec{a}^{\,1}, \qquad \vec{w}_f^{(2)} = \phi_f^{(2)} \, \vec{a}^{\,2}. \qquad (14.120\text{–}9)$$

The total number of functions are N_v = 36.

Since each vector functions W_i has only one covariant component, it becomes clear that each expansion coefficients E_i in (1), i.e., the AX DOF, represents a covariant component of the electric field E at a proper location, aside from a scale factor that may apply. The curl of E can be readily computed by using equation (14.120–6).

Similarly, we can define vector shape functions for the wedge shape by combining functions from the brick and tetrahedral shapes. See Section 14.119.2 for tetrahedral functions.

14.121 PLANE121 — 2–D 8–Node Electrostatic Solid



Matrix Or Vector	Geometry	Shape Functions	Integration Points
Coefficient Matrix	Quad	Equation (12.6.7–21)	3 x 3
	Triangle	Equation (12.6.2–21)	3
Charge Density Load Vector	Same as coefficient matrix		3 x 3 for quadrilateral 3 for triangle
Surface Charge Density and Load Vector	Same as coefficient matrix, specialized to the face		2

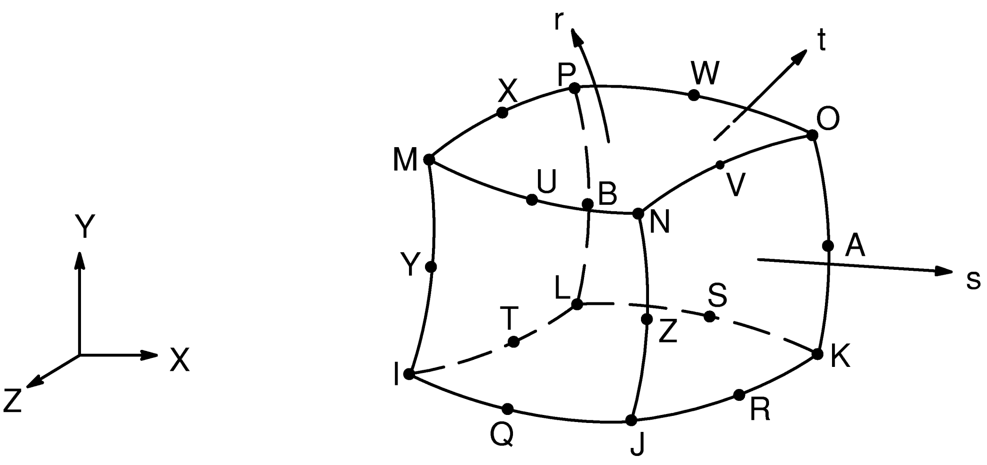
14.121.1 Other Applicable Sections

Chapter 5 describes the derivation of the electrostatic element matrices and load vectors as well as electric field evaluations. Section 13.1 describes integration point locations.

14.121.2 Assumptions and Restrictions

A dropped midside node implies that the edge is straight and that the potential varies linearly along that edge.

14.122 SOLID122 — 20–Node Electrostatic Solid

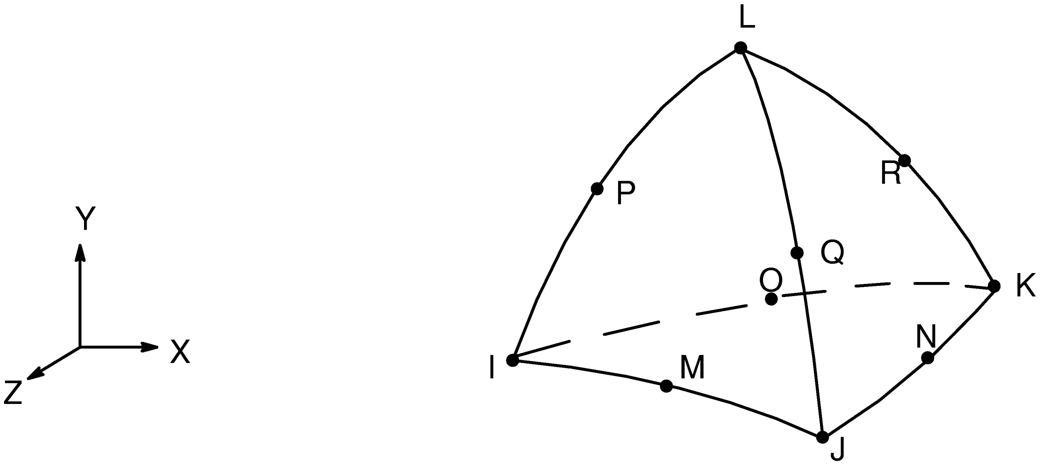


Matrix Or Vector	Geometry	Shape Functions	Integration Points
Coefficient Matrix	Brick	Equation (12.8.20–21)	14
	Wedge	Condensation of equation (12.8.20–21)	3 x 3
	Pyramid	Equation (12.8.7–21)	8
	Tet	Equation (12.8.2–21)	4
Charge Density Load Vector	Same as coefficient matrix		Same as coefficient matrix
Surface Charge Density Load Vector	Quad	Equation (12.5.10–21)	3 x 3
	Triangle	Equation (12.5.2–21)	6

14.122.1 Other Applicable Sections

Chapter 5 describes the derivation of electrostatic element matrices and load vectors as well as electric field evaluations. Section 13.1 describes integration point locations.

14.123 SOLID123 — 3–D 10–Node Tetrahedral Electrostatic Solid

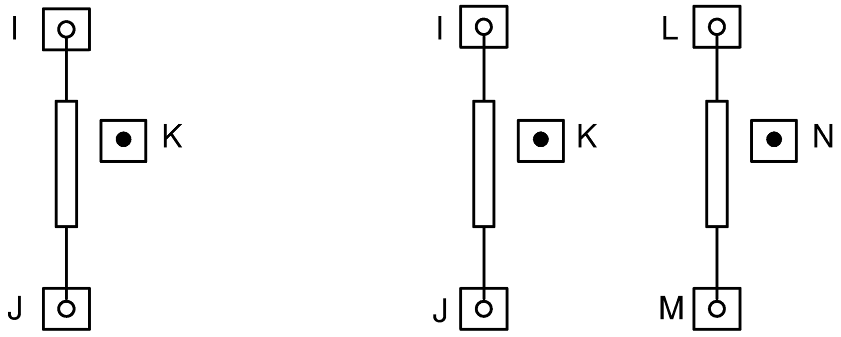


Matrix or Vector	Shape Functions	Integration Points
Coefficient Matrix	Equation (12.8.2–21)	4
Charge Density Load Vector	Equation (12.8.2–21)	4
Charge Density Surface Load Vector	Equation (12.8.2–21) specialized to the face	6

14.123.1 Other Applicable Sections

Chapter 5 describes the derivation of electrostatic element matrices and load vectors as well as electric field evaluations. Section 13.1 describes integration point locations.

14.124 CIRCU124 — General Electric Circuit Element



(a) Independent Circuit Elements (b) Dependent Circuit Elements

Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None (lumped)	None
Damping Matrix	None (lumped, harmonic analysis only)	None
Load Vector	None (lumped)	None

14.124.1 Electric Circuit Elements

CIRCU124 contains 13 linear electric circuit element options. These may be classified into two groups:

1. Independent Circuit Element options, defined by 2 or 3 nodes:

a. Resistor	(KEYOPT(1) = 0)
b. Inductor	(KEYOPT(1) = 1)
c. Capacitor	(KEYOPT(1) = 2)
d. Current Source	(KEYOPT(1) = 3)
e. Voltage Source	(KEYOPT(1) = 4)

2. Dependent Circuit Element options, defined by 3, 4, 5, or 6 nodes:

a. Stranded coil current source	(KEYOPT(1) = 5)
b. 2–D massive conductor voltage source	(KEYOPT(1) = 6)
c. 3–D massive conductor voltage source	(KEYOPT(1) = 7)
d. Mutual inductor	(KEYOPT(1) = 8)
e. Voltage–controlled current source	(KEYOPT(1) = 9)
f. Voltage–controlled voltage source	(KEYOPT(1) = 10)
g. Current–controlled voltage source	(KEYOPT(1) = 11)
h. Current–controlled current source	(KEYOPT(1) = 12)

14.124.2 Electric Circuit Element Matrices

All circuit options in CIRCU124 are based on Kirchhoff's Current Law. These options use stiffness matrices based on a simple lumped circuit model.

For transient analysis, an inductor with nodes I and J can be presented by:

$$\frac{\theta\Delta t}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} V_I^{n+1} \\ V_J^{n+1} \end{Bmatrix} = \begin{Bmatrix} -I_L^{n+1} \\ I_L^{n+1} \end{Bmatrix} \qquad (14.124\text{–}1)$$

where:

L = inductance

V_I = voltage at node I

V_J = voltage at node J

Δt = time increment

θ = time integration parameter

n = time step n

$I_L^{n+1} = \frac{(1-\theta)\ \Delta t}{L}\left(V_I^n - V_J^n\right) + i_L^n$

$i_L^{n+1} = \frac{\theta\Delta t}{L}\left(V_I^{n+1} - V_J^{n+1}\right) + I_L^{n+1}$

A capacitor with nodes I and J is represented by:

$$\frac{C}{\theta\Delta t}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} V_I^{n+1} \\ V_J^{n+1} \end{Bmatrix} = \begin{Bmatrix} -I_c^{n+1} \\ I_c^{n+1} \end{Bmatrix} \qquad (14.124\text{–}2)$$

where:

C = capacitance

$$I_c^{n+1} = -\frac{c}{\theta\Delta t}\left(V_I^n - V_J^n\right) - \frac{1-\theta}{\theta}\, i_c^n$$

$$i_c^{n+1} = \frac{c}{\theta\Delta t}\left(V_I^{n+1} - V_J^{n+1}\right) + I_c^{n+1}$$

Similarly, a mutual inductor with nodes I, J, K and L has the following matrix:

$$\frac{\theta\Delta t}{L_1L_2 - M^2}\begin{bmatrix} L_2 & -L_2 & -M & M \\ -L_2 & L_2 & M & -M \\ -M & M & L_1 & -L_1 \\ M & -M & -L_1 & L_1 \end{bmatrix}\begin{Bmatrix} V_I \\ V_J \\ V_K \\ V_L \end{Bmatrix} = \begin{Bmatrix} -I_1^{n+1} \\ I_1^{n+1} \\ -I_2^{n+1} \\ I_2^{n+1} \end{Bmatrix} \qquad (14.124\text{–}3)$$

where:

L1 = input side inductance

L2 = output side inductance

M = mutual inductance

$$I_1^{n+1} = \frac{(1-\theta)\ \Delta t}{L_1L_2 - M^2}\left[L_2\left(V_I^n - V_J^n\right) - M\left(V_K^n - V_L^n\right)\right] + i_1^n$$

$$I_2^{n+1} = \frac{(1-\theta)\ \Delta t}{L_1L_2 - M^2}\left[-M\left(V_I^n - V_J^n\right) + L_1\left(V_K^n - V_L^n\right)\right] + i_2^n$$

$$i_1^{n+1} = \frac{\theta\Delta t}{L_1L_2 - M^2}\left[L_2\left(V_I^{n+1} - V_J^{n+1}\right) - M\left(V_K^{n+1} - V_L^{n+1}\right)\right] + I_1^{n+1}$$

$$i_2^{n+1} = \frac{\theta\Delta t}{L_1L_2 - M^2}\left[-M\left(V_I^{n+1} - V_J^{n+1}\right) + L_1\left(V_K^{n+1} - V_L^{n+1}\right)\right] + I_2^{n+1}$$

For harmonic analysis, the above three circuit element options have only a damping matrix. For an inductor:

$$\left(-\frac{1}{\omega^2 L}\right)\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (14.124\text{–}4)$$

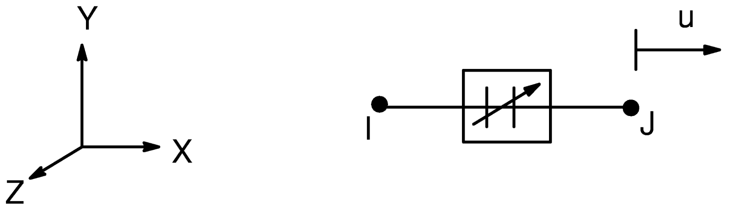
for a capacitor:

$$j\omega C\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad (14.124\text{–}5)$$

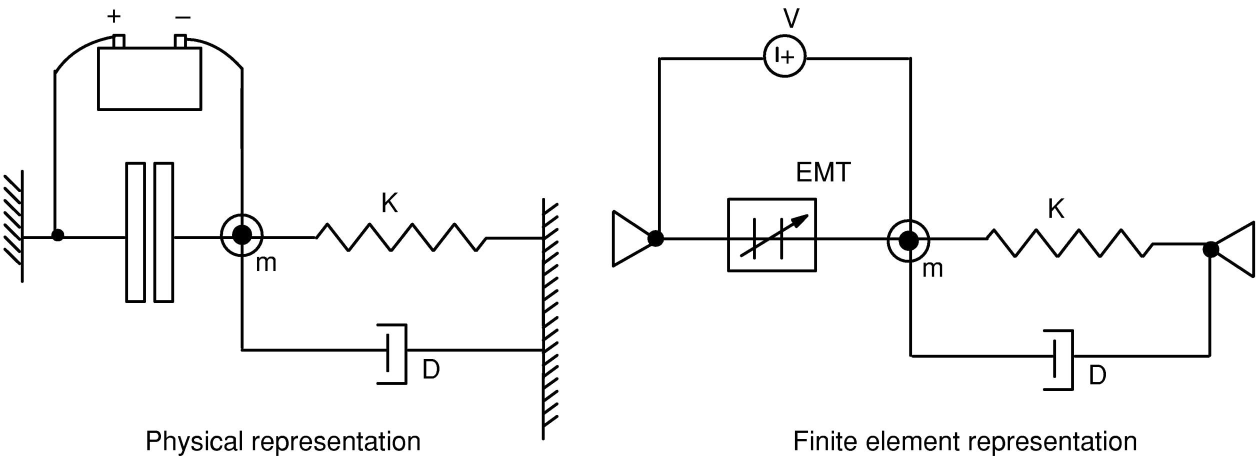
and for a mutual inductor:

$$\left(-\frac{1}{\omega^2\ (L_1L_2 - M^2)}\right)\begin{bmatrix} L_2 & -L_2 & -M & M \\ -L_2 & L_2 & M & -M \\ -M & M & L_1 & -L_1 \\ M & -M & -L_1 & L_1 \end{bmatrix} \qquad (14.124\text{–}6)$$

14.126 TRANS126 — Electromechanical Transducer for MEMS



The electromechanical transducer element, TRANS126, realizes strong coupling between distributed and lumped mechanical and electrostatic systems. For details abouts its theory see Gyimesi and Ostergaard(248). TRANS126 is especially suitable for the analysis of Micro Electromechanical Systems (MEMS): accelerometers, pressure sensors, micro actuators, gyroscopes, torsional actuators, filters, HF switches, etc.



Physical representation

Finite element representation

Figure 14.126–1 Electromechanical Transducer

See, for example, Figure 14.126–1 with a damped spring mass resonator driven by a parallel plate capacitor fed by a voltage generator constituting an electromechanical system. The left side shows the physical layout of the transducer connected to the mechanical system, the right side shows the equivalent electromechanical transducer element connected to the mechanical system.

TRANS126 is a 2 noded element each node having a structural (UX, UY or UZ) and an electrical (VOLT) DOFs. The force between the plates is attractive:

$$F = \frac{1}{2} \frac{dC}{dx} V^2 \tag{14.126–1}$$

where:
F = force
C = capacitance
x = gap size
V = voltage between capacitor electrodes

The capacitance can be obtained by ANSYS from a static FEM analysis using the **CMATRIX** macro. For the theory of computing the capacitance matrix see section 5.10.

The current is

$$I = C \frac{dV}{dt} + \frac{dC}{dx} v \; V \tag{14.126–2}$$

where:
I = current
t = time
v = velocity of gap opening $\left(= \frac{dx}{dt} \right)$

The first term is the usual capacitive current due to voltage change; the second term is the motion induced current.

For small signal analysis:

$$F = F_0 + D_{xv} \; v + D_{xv} \frac{dV}{dt} + K_{xx} \; \Delta x + K_{xv} \; \Delta V \tag{14.126–3}$$

$$I = I_0 + D_{vx} \; v + D_{vv} \frac{dV}{dt} + K_{vx} \; \Delta x + K_{vv} \; \Delta V \tag{14.126–4}$$

where:
F_0 = force at the operating point
I_0 = current at the operating point
[D] = linearized damping matrices
[K] = linearized stiffness matrices
Δx = gap change between the operating point and the actual solution
ΔV = voltage change between the operating point and the actual solution

The stiffness and damping matrices charaterize the transducer for small signal prestressed harmonic, modal and transient analyses.

For large signal static and transient analyses, the Newton–Raphson algorithm is applied with F_0 and I_0 constituting the Newton–Raphson restoring force and [K] and [D] the tangent stiffness and damping matrices.

$$K_{xx} = \frac{dF}{dx} = \frac{1}{2} C'' V^2 \qquad (14.126\text{–}5)$$

where: K_{xx} = electrostatic stiffness (output as ESTIF)
F = electrostatic force between capacitor plates
V = voltage between capacitor electrodes
C'' = second derivative of capacitance with respect to gap displacement

$$K_{vv} = \frac{dI}{dV} = C' v \qquad (14.126\text{–}6)$$

where: K_{vv} = motion conductivity (output as CONDUCT)
I = current
C' = first derivative of capacitance with respect to gap displacement
v = velocity of gap opening

Definitions of additional post items for the electromechanical transducer are as follows:

$$P_m = F v \qquad (14.126\text{–}7)$$

where: P_m = mechanical power (output as MECHPOWER)
F = force between capacitor plates
v = velocity of gap opening

$$P_e = V I \qquad (14.126\text{–}8)$$

where: P_e = electrical power (output as ELECPOWER)
V = voltage between capacitor electrodes
I = current

$$W_c = \frac{1}{2} C V^2 \qquad (14.126\text{–}9)$$

where: W_c = electrostatic energy of capacitor (output as CENERGY)
V = voltage between capacitor electrodes
C = capacitance

$$F = \frac{1}{2} \frac{dC}{dx} V^2 \qquad (14.126\text{–}10)$$

where:

F = electrostatic force between capacitor plates (output as EFORCE)

C = capacitance

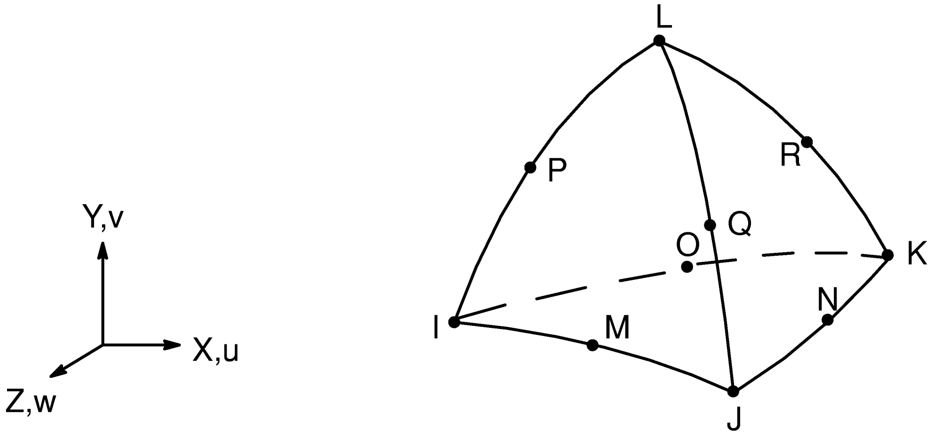
x = gap size

$\frac{dC}{dx}$ = first derivative of capacitance with regard to gap

V = voltage between capacitor electrodes

$\frac{dV}{dt}$ = voltage rate (output as DVDT)

14.127 SOLID127 — Tet Electrostatic p–Element

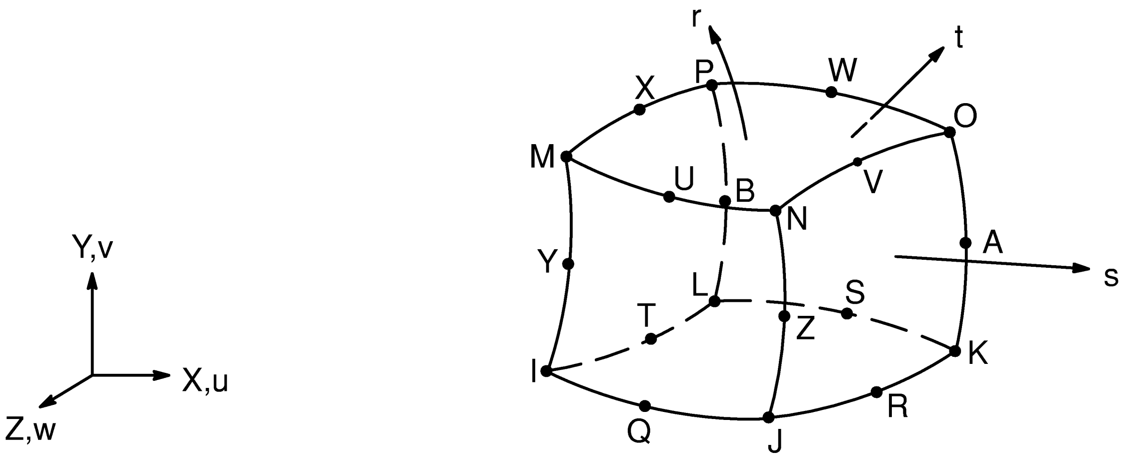


Matrix or Vector	Geometric Shape Functions	Solution Shape Functions	Integration Points
Coefficient Matrix	Equations (12.8.2–1), (12.8.2–2) and (12.8.2–3)	Polynomial variable in order from 2 to 8	Variable
Charge Density Load Vector	Equations (12.8.2–1), (12.8.2–2) and (12.8.2–3)	Polynomial variable in order from 2 to 8	Variable
Surface Charge Density Load Vector	Same as coefficient matrix specialized to face	Polynomial variable in order from 2 to 8	Variable

14.127.1 Other Applicable Sections

Chapter 5 describes the derivation of electrostatic element matrices and load vectors as well as electric field evaluations. Section 13.1 describes integration point locations.

14.128 SOLID128 — Brick Electrostatic p–Element



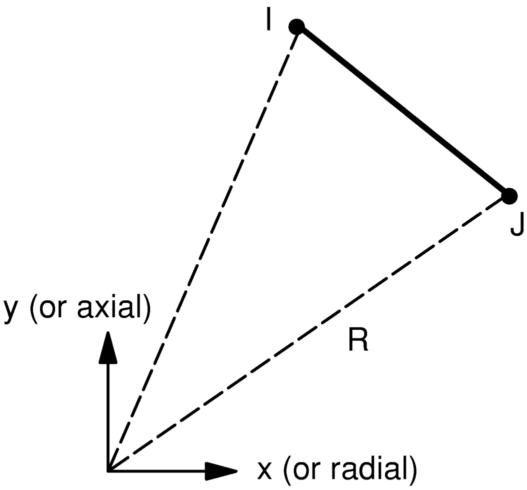
Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Coefficient Matrix	Brick	Equations (12.8.20–1), (12.8.20–2), and (12.8.20–3)	Polynomial variable in order from 2 to 8	Variable
	Wedge	Equations (12.8.14–1), (12.8.14–2), and (12.8.14–3)	Polynomial variable in order from 2 to 8	Variable
Charge Density Load Vector	Same as coefficient matrix		Polynomial variable in order from 2 to 8	Variable

Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Surface Charge Density Load Vector	Quad	Equations (12.5.10–1) and (12.5.10–2)	Polynomial variable in order from 2 to 8	Variable
	Triangle	Equations (12.5.2–1) and (12.5.2–2)	Polynomial variable in order from 2 to 8	Variable

14.128.1 Other Applicable Sections

Chapter 5 describes the derivation of electrostatic element matrices and load vectors as well as electric field evaluations. Section 13.1 describes integration point locations.

14.129 FLUID129 — 2–D Infinite Acoustic

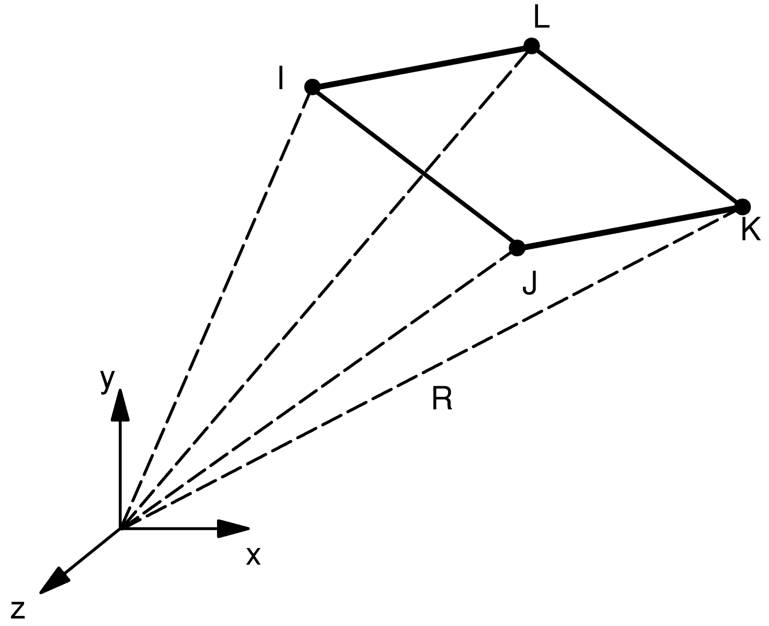


Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness Matrix	Equation (12.2.1–19)	2
Fluid Damping Matrix	Same as fluid stiffness matrix	2

14.129.1 Other Applicable Sections

The mathematical formulation and finite element discretization are presented in Section 14.130.

14.130 FLUID130 — 3–D Infinite Acoustic



Matrix or Vector	Shape Functions	Integration Points
Fluid Stiffness Matrix	Equation (12.6.5–19)	2 x 2
Fluid Damping Matrix	Same as Fluid Stiffness Matrix	2 x 2

14.130–1 Mathematical Formulation and F.E. Discretization

The exterior structural acoustics problem typically involves a structure submerged in an infinite, so–called acoustic fluid. The latter characterization implies that the fluid is homogeneous, linear, compressible and inviscid. When considering small barotropic disturbances of the pressure and density about an equilibrium state, it can be shown that the pressure field P [PRES] within the fluid is described by the scalar wave equation as:

$$\nabla^2 p = \frac{1}{c^2}\ddot{P} \quad \text{in } \Omega^+ \tag{14.130–1}$$

where: c = speed of sound in the fluid (input as SONC on **MP** command)

$\ddot{P}$ = second derivative of pressure with respect to time

Ω^{+} = unbounded region occupied by the fluid

In addition to equation (14.130–1), the following Sommerfeld radiation condition (which simply states that the waves generated within the fluid are outgoing) needs to be satisfied at infinity:

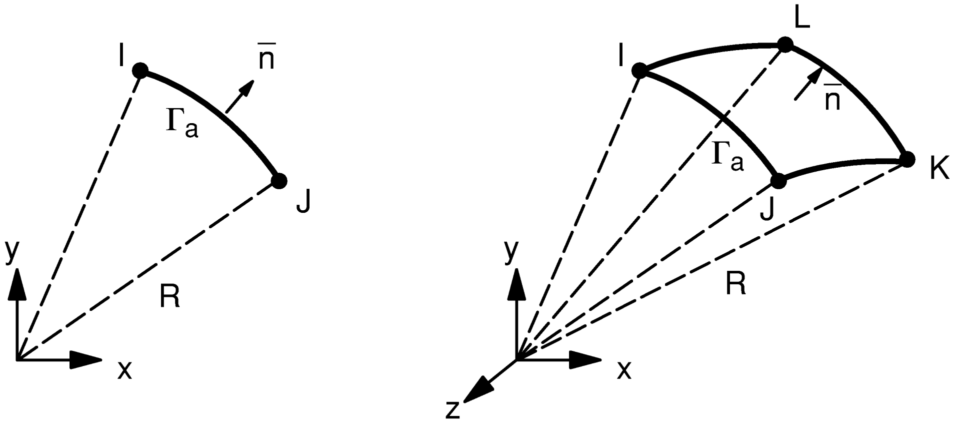
$$\lim_{r \to \infty} r^{\frac{d-1}{2}} \left(P_r + \frac{1}{c}\dot{P} \right) = 0. \qquad (14.130\text{–}2)$$

where: r = distance from the origin

P_r = pressure derivative along the radial direction

d = dimensionality of the problem (i.e., $d = 3$ or $d = 2$ if Ω^{+} is 3D or 2D, respectively

A primary difficulty associated with the use of finite elements for the modeling of the infinite medium stems precisely from the need to satisfy the Sommerfeld radiation condition, equation (14.130–2). A typical approach for tackling the difficulty consists of truncating the unbounded domain Ω^{+} by the introduction of an absorbing (artificial) boundary Γ_a at some distance from the structure.



The equation of motion (14.130–1) is then solved in the annular region Ω^{f} which is bounded by the fluid–structure interface Γ and the absorbing boundary Γ_a. In order, however, for the resulting problem in Ω^{f} to be well–posed, an appropriate condition needs to be specified on Γ_a. Towards this end, the following second–order conditions are used (Kallivokas etal.(218)) on Γ_a:

In two dimensions:

$$\dot{P}_n + \gamma\, P_n = -\frac{1}{c}\ddot{P} + \left(\frac{1}{2}\kappa - \frac{\gamma}{c}\right)\dot{P} + \frac{1}{2}c\, P_{\lambda\lambda} + \left(\frac{1}{8}\kappa^2 c + \frac{1}{2}\kappa\gamma\right) P \qquad (14.130\text{–}3)$$

where: n = outward normal to Γ_a

P_n = pressure derivative in the normal direction
$P_{\lambda\lambda}$ = pressure derivative along Γ_α
κ = curvature of Γ_a
γ = stability parameter

In three dimensions:

$$\dot{P}_n + \gamma\ P_n = -\tfrac{1}{c}\ddot{P} + \left(H - \tfrac{\gamma}{c}\right)\dot{P} + H\gamma\ P + \frac{c}{2\ \sqrt{EG}}\left[\left(\sqrt{\frac{G}{E}}\ P_u\right)_u + \left(\sqrt{\frac{G}{E}}\ P_v\right)_v\right] + \tfrac{c}{2}(H^2 - K)\ P \qquad (14.130–4)$$

where: n = outward normal
u and v = orthogonal curvilinear surface coordinates (e.g., the meridonional and polar angles in sperical coordinates)
P_u, P_v = pressure derivatives in the Γ_a surface directions
H and K = mean and Gaussian curvature, respectively
E and G = usual coefficients of the first fundamental form

14.130–2 Finite Element Discretization

Following a Galerkin based procedure, equation (14.130–1) is multiplied by a virtual quantity δP and integrated over the annular domain Ω^{f}· By using the divergence theorem on the resulting equation it can be shown that:

$$\frac{1}{c^2}\int_{\Omega^f} \delta P\ddot{P}d\Omega^f + \int_{\Omega^f} \nabla\delta P \cdot \nabla P d\Omega^f - \int_{\Gamma_a} \delta P\ P_n d\Gamma_a = -\int_{\Gamma} \delta P\ P_n d\Gamma \qquad (14.130–5)$$

Upon discretization of equation (14.130–5), the first term on the left hand side will yield the mass matrix of the fluid while the second term will yield the stiffness matrix.

Next, the following finite element approximations for quantities on the absorbing boundary Γ_a placed at a radius R and their virtual counterparts are introduced:

$$P(x,t) = \mathbf{N}_1^T\ (x)P(t)\ \ ,\ \ q^{(1)}(x,t) = \mathbf{N_2}\ (x)q^{(1)}(t)\ \ ,\ \ q^{(2)}(x,t) = \mathbf{N}_3^T(x)q^{(2)}(t) \qquad (14.130–6)$$

$$\delta P(x) = \delta P^T\mathbf{N_1}(x)\ \ ,\ \ \delta q^{(1)}(x) = \delta q^{(1)T}\mathbf{N_2}(x)\ \ ,\ \ \delta q^{(2)}(x) = \delta q^{(2)T}\mathbf{N_3}(x) \qquad (14.130–7)$$

where: $\mathbf{N}_1, \mathbf{N}_2, \mathbf{N}_3$ = vectors of shape functions (= {N_1}, {N_2}, {N_3})

P, $q^{(1)}$, $q^{(2)}$ = unknown nodal values (P is output as degree of freedom PRES. $q^{(1)}$ and $q^{(2)}$ are solved for but not output).

Furthermore, the shape functions in equations (14.130–6) and (14.130–7) are set to:

$$\mathbf{N}_1 = \mathbf{N}_2 = \mathbf{N}_3 = \mathbf{N} \qquad (14.130\text{–}8)$$

The element stiffness and damping matrices reduce to:

For two dimensional case:

$$\left[K_a^{2D}\right] = \frac{1}{8R}\begin{bmatrix} 4\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & 4R^2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & -\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e \\ 4R^2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & -4R^2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & 0 \\ -\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & 0 & \int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e \end{bmatrix} \qquad (14.130\text{–}9)$$

$$\left[C_a^{2D}\right] = \frac{1}{8c}\begin{bmatrix} 8\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & 0 & 0 \\ 0 & -4R^2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e & 0 \\ 0 & 0 & \int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\,d\lambda_e \end{bmatrix} \qquad (14.130\text{–}10)$$

where: $d\lambda_e$ = arc–length differential

These matrices are 6 x 6 in size, having 2 nodes per element with 3 degrees of freedom per node (P, $q^{(1)}$, $q^{(2)}$).

For three dimensional case:

$$\left[K_a^{3D}\right] = \frac{1}{2R}\begin{bmatrix} 2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, dA_e & R^2 \int\limits_{\Gamma_a^e} \overline{\nabla}^s\, \mathbf{N} \cdot \overline{\nabla}^s\, \mathbf{N}^T\, dA_e \\ R^2 \int\limits_{\Gamma_a^e} \overline{\nabla}^s\, \mathbf{N}^T \cdot \overline{\nabla}^s\, \mathbf{N}\, dA_e & -R^2 \int\limits_{\Gamma_a^e} \overline{\nabla}^s\, \mathbf{N} \cdot \overline{\nabla}^s\, \mathbf{N}^T\, dA_e \end{bmatrix} \quad (14.130\text{–}11)$$

$$\left[C_a^{3D}\right] = \frac{1}{2c}\begin{bmatrix} 2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, dA_e & 0 \\ 0 & -R^2 \int\limits_{\Gamma_a^e} \overline{\nabla}^s\, \mathbf{N} \cdot \overline{\nabla}^s\, \mathbf{N}^T\, dA_e \end{bmatrix} \quad (14.130\text{–}12)$$

where: dA_e = area differential

These matrices are 8 x 8 in size, having 4 nodes per element with 2 degrees of freedom per node (P, q) (Barry et al. (217)).

For axisymmetric case:

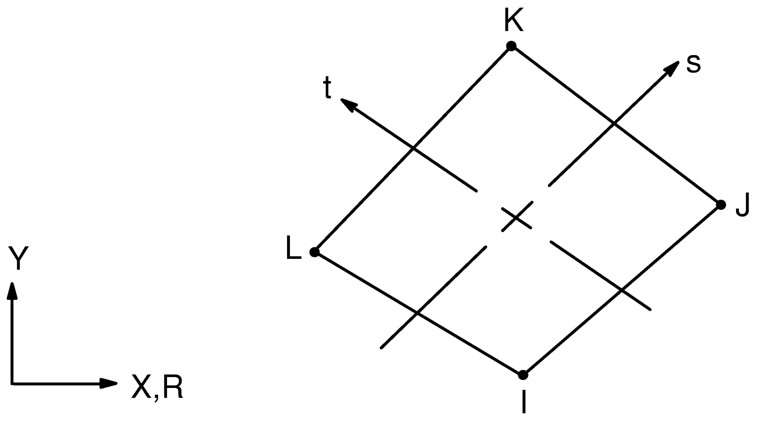
$$\left[K_a^{2Da}\right] = \frac{\pi}{R}\begin{bmatrix} 2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, xd\lambda_e & R^2 \int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, xd\lambda_e \\ R^2 \int\limits_{\Gamma_a^e} \mathbf{N}^T\,\mathbf{N}\, xd\lambda_e & -R^2 \int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, xd\lambda_e \end{bmatrix} \quad (14.130\text{–}13)$$

$$\left[C_a^{2Da}\right] = \frac{\pi}{c}\begin{bmatrix} 2\int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, xd\lambda_e & 0 \\ 0 & -R^2 \int\limits_{\Gamma_a^e} \mathbf{N}\,\mathbf{N}^T\, xd\lambda_e \end{bmatrix} \quad (14.130\text{–}14)$$

where: x = radius

These matrices are 4 x 4 in size having 2 nodes per element with 2 degrees of freedom per node (P, q).

14.141 FLUID141 — 2–D Fluid



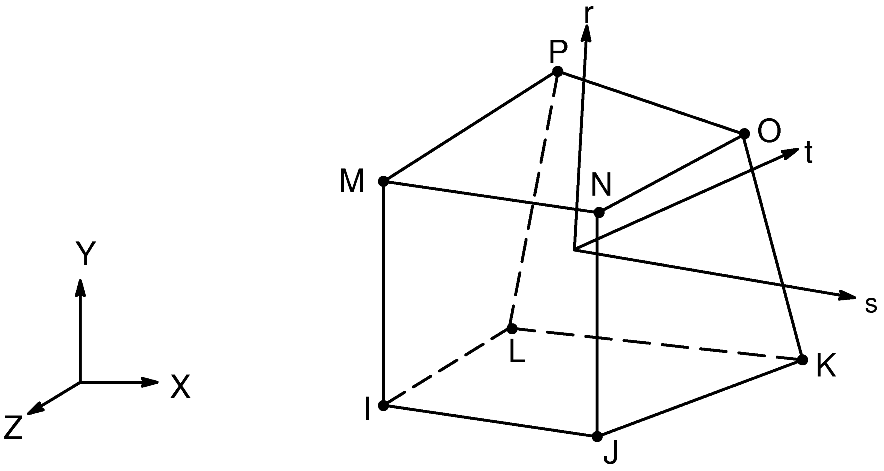
Matrix or Vector	Geometry	Shape Functions	Integration Points
Advection–Diffusion Matrices for Momentum Equations (X, Y and Z)	Quad	Equations (12.6.5–10), (12.6.5–11), and (12.6.5–12)	if 2–D 1 (default) or 2 x 2; if axisymmetric 1 or 2 x 2 (default) (adjustable with the **FLDA**,QUAD,MOMD command)
	Triangle	Equations (12.6.5–10), (12.6.5–11), and (12.6.5–12)	1
Advection–Diffusion Matrix for Pressure	Quad	Equation (12.6.5–19)	Same as for momentum equation, but adjustable with the **FLDA**,QUAD,PRSD command
	Triangle	Equation (12.6.5–19)	
Advection–Diffusion Matrix for Energy (Temperature)	Quad	Equation (12.6.5–20)	Same as for momentum, equations but adjustable with the **FLDA**,QUAD,THRD command
	Triangle	Equation (12.6.5–20)	

Matrix or Vector	Geometry	Shape Functions	Integration Points
Advection–Diffusion Matrices for Turbulent Kinetic Energy and Dissipation Rate	Quad	Equations (12.6.5–23) and (12.6.5–24)	Same as for momentum, equations but adjustable with the **FLDA**,QUAD,TRBD command
	Triangle	Equations (12.6.5–23) and (12.6.5–24)	
Momentum Equation Source Vector	Same as momentum equation matrix		Same as momentum equations, but adjustable with the **FLDA**,QUAD,MOMS command
Pressure Equation Source Vector	Same as pressure matrix		Same as pressure equations, adjustable with the **FLDA**,QUAD,PRSS command
Heat Generation Vector	Same as temperature matrix		Same as temperature equations, adjustable with the **FLDA**,QUAD,THRS command
Turbulent Kinetic Energy and Dissipation Rate Source Term Vectors	Same as kinetic energy and dissipation rate matrices		Same as kinetic energy and dissipation rate equations, adjustable with the **FLDA**,QUAD,TRBS command
Distributed Resistance Source Term Vector	Same as momentum equation matrix		1
Convection Surface Matrix and Load Vector and Heat Flux Load Vector	One–half of the element face length times the heat flow rate is applied at each edge node		None

14.141.1 Other Applicable Sections

Chapter 7 describes the derivation of the applicable matrices, vectors, and output quantities. Chapter 6 describes the derivation of the heat transfer logic, including the film coefficient treatment.

14.142 FLUID142 — 3–D Fluid



Matrix or Vector	Geometry	Shape Functions	Integration Points
Advection–Diffusion Matrix for Momentum Equations (X, Y and Z)	Brick, Pyramid, and Wedge	Equations (12.8.18–10), (12.8.18–11) and (12.8.18–12)	1 (default) or 2 x 2x 2 (adjustable with the **FLDA**,QUAD,MOMD command)
	Tet	Equations (12.8.18–10), (12.8.18–11) and (12.8.18–12)	1
Advection–Diffusion Matrix for Pressure	Brick, Pyramid, and Wedge	Equation (12.8.18–19)	Same as for equation momentum, but adjustable with the **FLDA**,QUAD,PRSD command
	Tet	Equation (12.8.18–19)	1

Matrix or Vector	Geometry	Shape Functions	Integration Points
Advection–Diffusion Matrix for Energy (Temperature)	Brick, Pyramid, and Wedge	Equation (12.8.18–20)	Same as for momentum, equations but adjustable with the **FLDA**,QUAD,THRD command
	Tet	Equation (12.8.18–20)	1
Advection–Diffusion Matrices for Turbulent Kinetic Energy and Dissipation Rate	Brick, Pyramid, and Wedge	Equations (12.8.18–23) and (12.8.18–24)	Same as for momentum, equations but adjustable with the **FLDA**,QUAD,TRBD command
	Tet	Equations (12.8.18–23) and (12.8.18–24)	1
Momentum Equation Source Vector	Brick, Pyramid, and Wedge	Equations (12.8.18–10), (12.8.18–11) and (12.8.18–12)	1 (default) or 2 x 2x 2 (adjustable with the **FLDA**,QUAD,MOMS command)
	Tet	Equations (12.8.18–10), (12.8.18–11) and (12.8.18–12)	1
Pressure Equation Source Vector	Brick, Pyramid, and Wedge	Equation (12.8.18–19)	Same as for equation momentum, but adjustable with the **FLDA**,QUAD,PRSS command
	Tet	Equation (12.8.18–19)	1
Heat Generation Vector	Brick, Pyramid, and Wedge	Equation (12.8.18–20)	Same as for momentum, equations but adjustable with the **FLDA**,QUAD,THRS command
	Tet	Equation (12.8.18–20)	1

Matrix or Vector	Geometry	Shape Functions	Integration Points
Turbulent Kinetic Energy and Dissipation Rate Source Term Vectors	Brick, Pyramid, and Wedge	Equations (12.8.18–23) and (12.8.18–24)	Same as for momentum, equations but adjustable with the **FLDA**,QUAD,TRBS command
	Tet	Equations (12.8.18–23) and (12.8.18–24)	1
Distributed Resistance Source Term Vector	Same as momentum equation source vector		Same as momentum equation source vector
Convection Surface Matrix and Load Vector and Heat Flux Load Vector	Brick, Pyramid, and Wedge	One–fourth of the element surface area times the heat flow rate is applied at each face node	None
	Tet	One–third of the element surface area times the heat flow rate is applied at each face node	

14.142.1 Other Applicable Sections

Chapter 7 describes the derivation of the applicable matrices, vectors, and output quantities. Chapter 6 describes the derivation of the heat transfer logic, including the film coefficient treatment.

14.142.2 Distributed Resistance Main Diagonal Modification

Suppose the matrix equation representation for the momentum equation in the X direction written without distributed resistance may be represented by the expression:

$$A_x^m \, V_x = b_x^m \qquad (14.142\text{–}1)$$

The source terms for the distributed resistances are summed:

$$D^{Rx} = \left[\rho\ K_x\ |V| + \frac{f_x\ \rho\ |V|}{D_{hx}} + C_x\ \mu \right] \qquad (14.142\text{–}2)$$

where:

D^{Rx} = distributed resistance in the x direction

K_x = loss coefficient in the X direction

ϱ = density

f_x = friction factor for the X direction

μ = viscosity

C_x = permeability in the X direction

$| V |$ = velocity magnitude

D_{hx} = hydraulic diameter in the X direction

Consider the ith node algebraic equation. The main diagonal of the A matrix and the source terms are modified as follows:

$$A_{ii}^{mx} = A_{ii}^{mx} + D_i^{Rx} \qquad (14.142\text{–}3)$$

$$b_i^{mx} = b_i^{mx} + 2\ D_i^{Rx}\ V_x \qquad (14.142\text{–}4)$$

14.142.3 Turbulent Kinetic Energy Source Term Linearization

The source terms are modified for the turbulent kinetic energy k and the turbulent kinetic energy dissipation rate ε to prevent negative values of kinetic energy.

The source terms for the kinetic energy combine as follows:

$$S_k = \mu_t \frac{\partial V_i}{\partial X_j} \left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i} \right) - \rho\epsilon \qquad (14.142\text{–}5)$$

where the velocity spatial derivatives are written in indicial notation and μ_t is the turbulent viscosity:

$$\mu_t = C_\mu \rho\ \frac{k^2}{\epsilon} \qquad (14.142\text{–}6)$$

where:

ϱ = density

C_μ = constant

The source term may thus be rewritten:

$$S_k = \mu_t \frac{\partial V_i}{\partial X_j} \left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i} \right) - C_\mu \rho^2 \frac{k^2}{\mu_t} \qquad (14.142–7)$$

A truncated Taylor series expansion of the kinetic energy term around the previous (old) value is expressed:

$$S_k = S_{k\,\text{old}} + \left. \frac{\partial S_k}{\partial k} \right|_{k_{\text{old}}} (k - k_{\text{old}}) \qquad (14.142–8)$$

The partial derivative of the source term with respect to the kinetic energy is:

$$\frac{\partial S_k}{\partial k} = -2\, C_\mu \rho^2 \frac{k}{\mu_t} \qquad (14.142–9)$$

The source term is thus expresed

$$S_k = \mu_t \frac{\partial V_i}{\partial X_j} \left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i} \right) + C_\mu \rho^2 \frac{k^2_{\text{old}}}{\mu_t} - 2\, C_\mu \rho^2 \frac{k_{\text{old}}}{\mu_t}\, k \qquad (14.142–10)$$

The first two terms are the source term, and the final term is moved to the coefficient matrix. Denote by A^k the coefficient matrix of the turbulent kinetic energy equation before the linearization. The main diagonal of the ith row of the equation becomes:

$$A^k_{ii} = A^k_{ii} + 2\, C_\mu \rho^2 \frac{k_{\text{old}}}{\mu_t} \qquad (14.142–11)$$

and the source term is:

$$S_k = \mu_t \frac{\partial V_i}{\partial X_j} \left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i} \right) + C_\mu \rho^2 \frac{k^2_{\text{old}}}{\mu_t} \qquad (14.142–12)$$

14.142.4 Turbulent Kinetic Energy Dissipation Rate Source Term Linearization

The source term for the dissipation rate is handled in a similar fashion.

$$S_\epsilon = C_1 \mu_t \frac{\epsilon}{k} \frac{\partial V_i}{\partial X_j} \left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i} \right) - C_2 \rho \frac{\epsilon^2}{k} \qquad (14.142–13)$$

Replace ε using the expression for the turbulent viscosity to yield

$$S_\epsilon = C_1 C_\mu \rho k \frac{\partial V_i}{\partial X_j}\left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i}\right) - C_2 \rho \frac{\epsilon^2}{k} \qquad (14.142\text{–}14)$$

A truncated Taylor series expansion of the dissipation source term around the previous (old) value is expressed

$$S_\epsilon = S_{\epsilon\,\mathrm{old}} + \left.\frac{\partial S_\epsilon}{\partial \epsilon}\right|_{\epsilon_{\mathrm{old}}} (\epsilon - \epsilon_{\mathrm{old}}) \qquad (14.142\text{–}15)$$

The partial derivative of the dissipation rate source term with respect to ε is:

$$\frac{\partial S_\epsilon}{\partial \epsilon} = -2\ C_2 \rho \frac{\epsilon}{k} \qquad (14.142\text{–}16)$$

The dissipation source term is thus expressed

$$S_\epsilon = C_1 C_\mu \rho k \frac{\partial V_i}{\partial X_j}\left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i}\right) + C_2 \rho \frac{\epsilon^2_{\mathrm{old}}}{k} - 2\ C_2 \rho \frac{\epsilon_{\mathrm{old}}}{k}\,\epsilon \qquad (14.142\text{–}17)$$

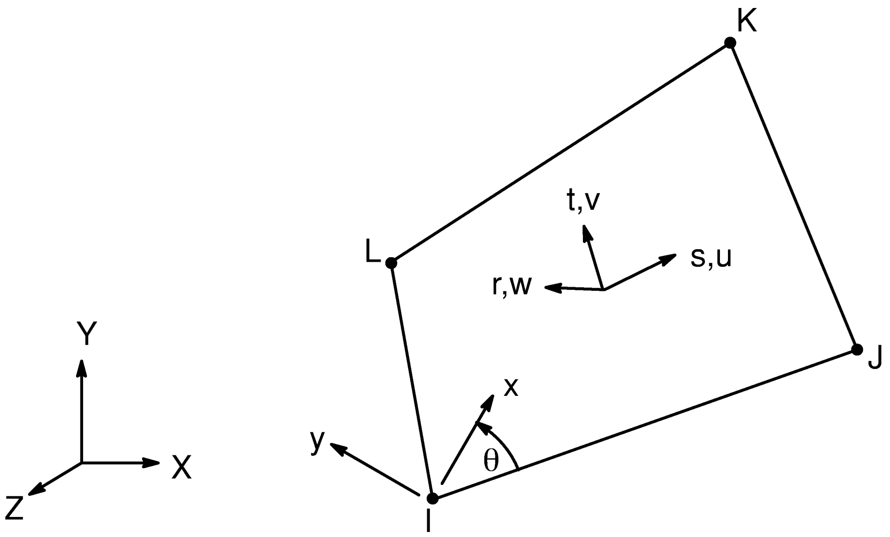
The first two terms are the source term, and the final term is moved to the coefficient matrix. Denote by A^ε the coefficient matrix of the turbulent kinetic energy dissipation rate equation before the linearization. The main diagonal of the ith row of the equation becomes:

$$A^\epsilon_{ii} = A^\epsilon_{ii} + 2\ C_2 \rho \frac{\epsilon_{\mathrm{old}}}{k} \qquad (14.142\text{–}18)$$

and the source term is:

$$S_\epsilon = C_1\ C_\mu\ \rho\ k \frac{\partial V_i}{\partial X_j}\left(\frac{\partial V_i}{\partial X_j} + \frac{\partial V_j}{\partial X_i}\right) + C_\mu\ \rho \frac{\epsilon^2_{\mathrm{old}}}{k} \qquad (14.142\text{–}19)$$

14.143 SHELL143 — Plastic Shell



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.13–1), (12.5.13–2), and (12.5.13–3)	In–plane: 2 x 2 Thru–the–thickness: 2 (linear material) 5 (nonlinear material)
	Triangle	Equations (12.5.4–1), (12.5.4–2), and (12.5.4–3)	In–plane: 1 Thru–the–thickness: 2 (linear material) 5 (nonlinear material)
Mass Matrix	Quad	Equations (12.5.8–1), (12.5.8–2), and (12.5.8–3)	Same as stiffness matrix
	Triangle	Equations (12.5.1–1), (12.5.1–2), and (12.5.1–3)	Same as stiffness matrix
Stress Stiffness Matrix	Same as mass matrix		Same as stiffness matrix

Matrix or Vector	Geometry	Shape Functions	Integration Points
Thermal Load Matrix	Same as stiffness matrix		Same as stiffness matrix
Transverse Pressure Load Vector	Quad	Equation (12.5.8–3)	2 x 2
	Triangle	Equation (12.5.1–3)	1
Edge Pressure Load Vector	Quad	Equations (12.5.8–1) and (12.5.8–2) specialized to the edge	2
	Triangle	Equations (12.5.1–1) and (12.5.1–2) specialized to the edge	2

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru thickness
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element and linear along each edge

References: Ahmad(1), Cook(5), Dvorkin(96), Dvorkin(97), Bathe and Dvorkin(98), Allman(113), Cook(114), MacNeal and Harder(115)

14.143.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.143.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same element (material) orientation.

This element does not generate a consistent mass matrix; only the lumped mass matrix is available.

14.143.3 Assumed Displacement Shape Functions

The assumed displacement and transverse shear strain shape functions are given in Chapter 12. The basic shape functions are essentially a condensation of those used for SHELL93. The basic functions for the transverse shear strain have been changed to avoid shear locking (Dvorkin(96), Dvorkin(97), Bathe and Dvorkin(98)) and are pictured in Figure 14.43–1. One result of the use of these displacement and strain shapes is that elastic rectangular elements give constant curvature results for flat elements, and also, in the absence of membrane loads, for curved elements. Thus, for these cases, nodal stresses are the same as centroidal stresses. Both SHELL63 and SHELL93 can have linearly varying curvatures.

14.143.4 Stress–Strain Relationships

The material property matrix [D] for the element is described in Section 14.43.4. It is the same as SHELL43.

14.143.5 In–Plane Rotational DOF

If KEYOPT(3) is 0 or 1, there is no significant stiffness associated with the in–plane rotation DOF (rotation about the element r axis). A nominal value of stiffness is present (as described with SHELL63), however, to prevent free rotation at the node. KEYOPT(3) = 2 is used to include the Allman–type rotational DOFs (as described by Allman(113) and Cook(114)). Such rotations improve the in–plane and general 3–D shell performance of the element. However, one of the outcomes of using the Allman rotation is that the element stiffness matrix contains up to two spurious zero energy modes.

14.143.6 Spurious Mode Control with Allman Rotation

This procedure is described in Section 14.43.6. The same procedure as implemented in SHELL43 is used here.

14.143.7 Natural Space Extra Shape Functions with Allman Rotation

One of the outcomes of the Allman rotation is the dissimilar displacement variation along the normal and tangential directions of the element edges. The result of such variation is that the in–plane bending stiffness of the elements is too large by a factor $1/(1–\nu^2)$ and sometimes termed as Poisson's ratio locking. To overcome this difficulty, two natural space (s and t) nodeless in–plane displacement shape functions are added in the element stiffness matrix formulation and then condensed out at the element level.

The element thus generated is free of Poisson's ratio locking. For details of a similar implementation, refer to Yunus et al(117).

14.143.8 Warping

A warping factor is computed as:

$$\phi = \frac{D}{t} \qquad (14.143\text{–}1)$$

where: D = component of the vector from the first node to the fourth node parallel to the element normal

t = average thickness of the element

If $\phi > 1.0$, a warning message is printed.

14.143.9 Consistent Tangent

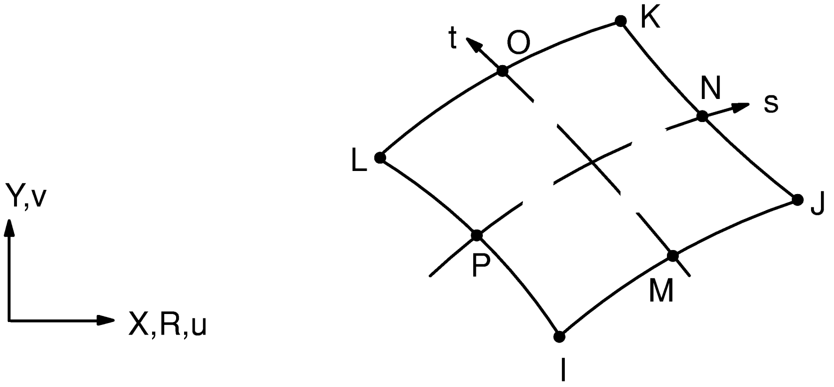
A consistent tangent matrix implemented with the finite rotation capability is available by using KEYOPT(2) = 1. The theory is described in Section 3.2.9.

14.143.10 Stress Output

The stresses at the center of the element are computed by taking the average of the four integration points on that plane.

The output forces and moments are computed as described in Section 2.3.

14.145 PLANE145 — 2–D Quadrilateral Structural Solid p–Element



Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.6.7–1) and (12.6.7–2)	Polynomial variable in order from 2 to 8	Variable
	Triangle	Equations (12.6.2–1) and (12.6.2–2)	Polynomial variable in order from 2 to 8	Variable
Thermal and Inertial Load Vector	Same as stiffness matrix.		Polynomial variable in order from 2 to 8	Variable
Pressure Load Vector	Same as stiffness matrix, specialized to the edge		Polynomial variable in order from 2 to 8	Variable

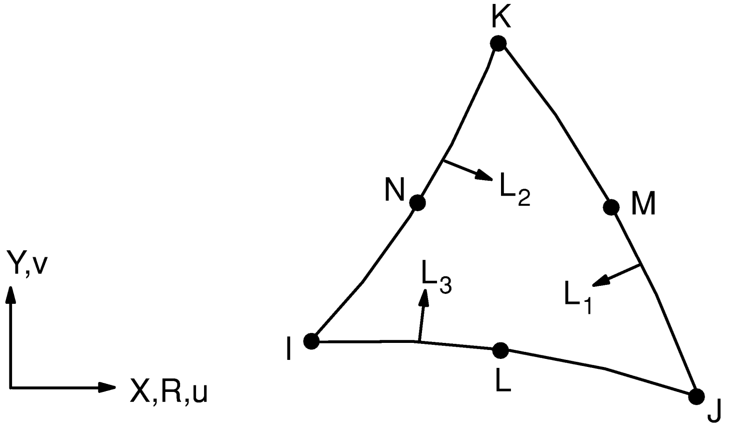
Load Type	Distribution
Element Temperature	Same as geometric shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear across each face

Reference: Szabo and Babuska(192)

14.145.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations.

14.146 PLANE146 — 2–D Triangular Structural Solid p–Element



Matrix or Vector	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness Matrix	Equations (12.6.2–1) and (12.6.2–2)	Polynomial variable in order from 2 to 8	Variable
Thermal and Inertial Load Vector	Same as stiffness matrix	Polynomial variable in order from 2 to 8	Variable
Pressure Load Vector	Same as stiffness matrix, specialized to the edge	Polynomial variable in order from 2 to 8	Variable

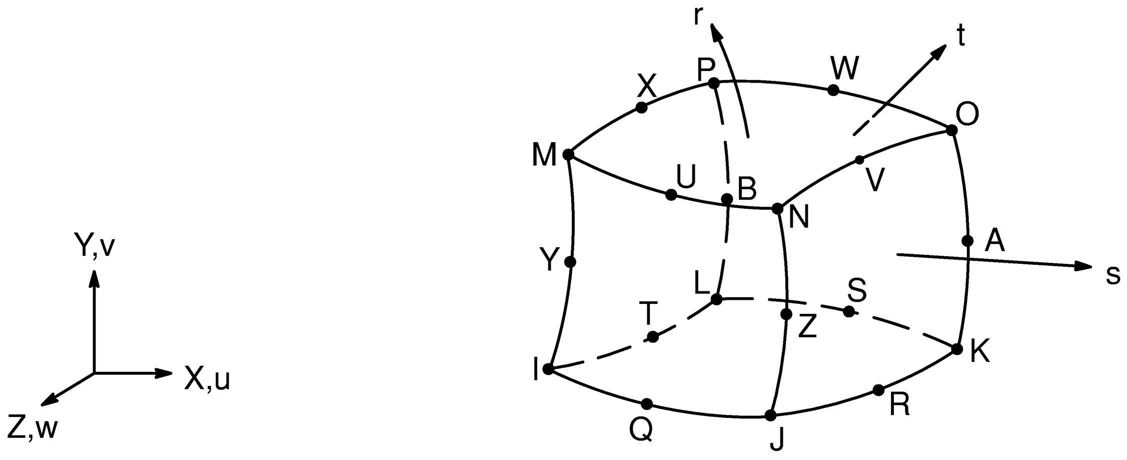
Load Type	Distribution
Element Temperature	Same as geometric shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear across each face

Reference: Szabo and Babuska(192)

14.146.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations.

14.147 SOLID147 — 3–D Brick Structural Solid p–Element



Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness Matrix	Brick	Equations (12.8.20–1), (12.8.20–2), and (12.8.20–3)	Polynomial variable in order from 2 to 8	Variable
	Wedge	Equations (12.8.14–1), (12.8.14–2), and (12.8.14–3)	Polynomial variable in order from 2 to 8	Variable
Thermal and Inertial Load Vector	Same as stiffness matrix.		Polynomial variable in order from 2 to 8	Variable

Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Pressure Load Vector	Quad	Equations (12.5.10–1) and (12.5.10–2)	Polynomial variable in order from 2 to 8	Variable
	Triangle	Equations (12.5.2–1) and (12.5.2–2)	Polynomial variable in order from 2 to 8	Variable

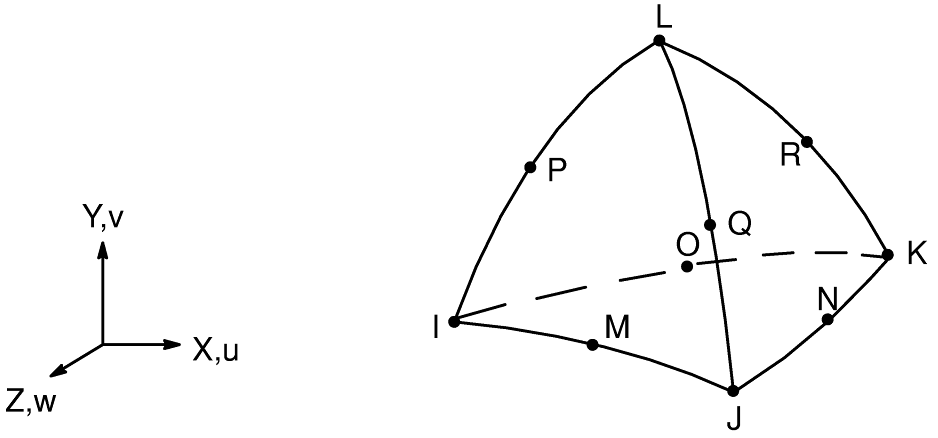
Load Type	Distribution
Element Temperature	Same as geometric shape functions thru element
Nodal Temperature	Same as geometric shape functions thru element
Pressure	Bilinear across each face

Reference: Szabo and Babuska(192)

14.147.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations.

14.148 SOLID148 — 3–D Tetrahedral Structural Solid p–Element



Matrix or Vector	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2) and (12.8.2–3)	Polynomial variable in order from 2 to 8	Variable
Thermal and Inertial Load Vector	Equations (12.8.2–1), (12.8.2–2) and (12.8.2–3)	Polynomial variable in order from 2 to 8	Variable
Pressure Load Vector	Same as stiffness matrix specialized to face	Polynomial variable in order from 2 to 8	Variable

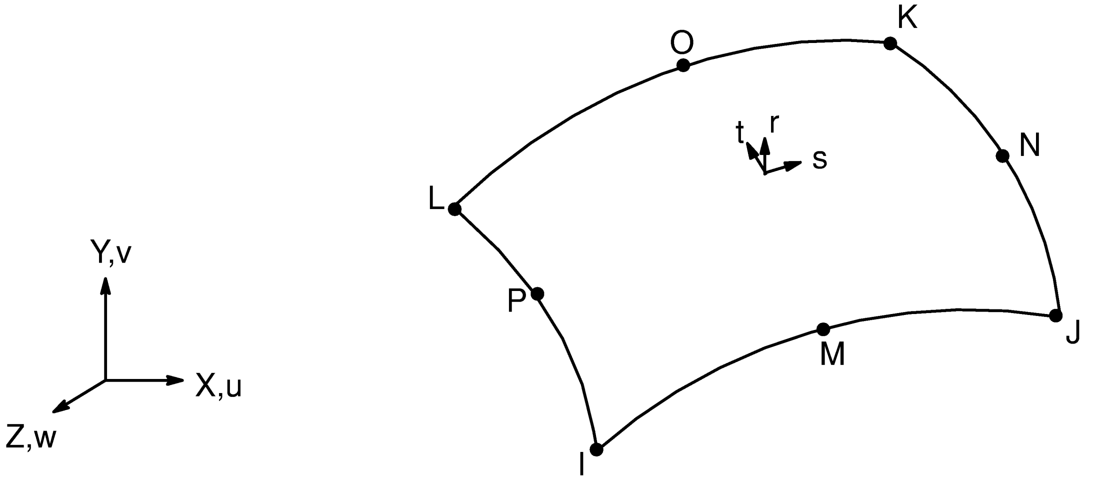
Load Type	Distribution
Element Temperature	Same as geometric shape functions
Nodal Temperature	Same as geometric shape functions
Pressure	Linear across each face

Reference: Szabo and Babuska(192)

14.148.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations.

14.150 SHELL150 — 8–Node Structural Shell p–Element



Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Stiffness Matrix	Quad	Equations (12.5.14–1), (12.5.14–2), and (12.5.14–3)	Polynomial variable in order from 2 to 8	Thru–the–thickness: 2 In–plane: Variable
	Triangle	Equations (12.5.5–1), (12.5.5–2), and (12.5.5–3)	Polynomial variable in order from 2 to 8	Thru–the–thickness:2 In–plane: Variable
Thermal and Inertial Load Vector	Same as stiffness matrix		Polynomial variable in order from 2 to 8	Same as stiffness matrix

Matrix or Vector	Geometry	Geometric Shape Functions	Solution Shape Functions	Integration Points
Transverse Pressure Load Vector	Quad	Equation (12.5.10–3)	Polynomial variable in order from 2 to 8	Variable
	Triangle	Equation (12.5.2–3)	Polynomial variable in order from 2 to 8	Variable
Edge Pressure Load Vector	Same as in–plane stiffness matrix, specialized to the edge		Polynomial variable in order from 2 to 8	Variable

Load Type	Distribution
Element Temperature	Linear thru thickness, bilinear in plane of element
Nodal Temperature	Constant thru thickness, bilinear in plane of element
Pressure	Bilinear across plane of element, linear along each edge

Reference: Ahmad (1), Cook(5), Szabo and Babuska(192)

14.150.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations.

14.150.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same element (material) orientation.

There is no significant stiffness associated with rotation about the element r axis.

This element uses a lumped (translation only) inertial load vector.

14.150.3 Stress–Strain Relationships

The material property matrix [D] for the element is:

$$[D] = \begin{bmatrix} BE_x & B\nu_{xy}E_x & 0 & 0 & 0 & 0 \\ B\nu_{xy}E_x & BE_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{G_{yz}}{f} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{G_{xz}}{f} \end{bmatrix} \qquad (14.150\text{–}1)$$

where: $B = \dfrac{E_y}{E_y - (\nu_{xy})^2\ E_x}$

E_x = Young’s modulus in element x direction (input as EX on **MP** command)

ν_{xy} = Poisson’s ratio in element x–y plane (input as NUXY on **MP** command)

G_{xy} = shear modulus in element x–y plane (input as GXY on **MP** command)

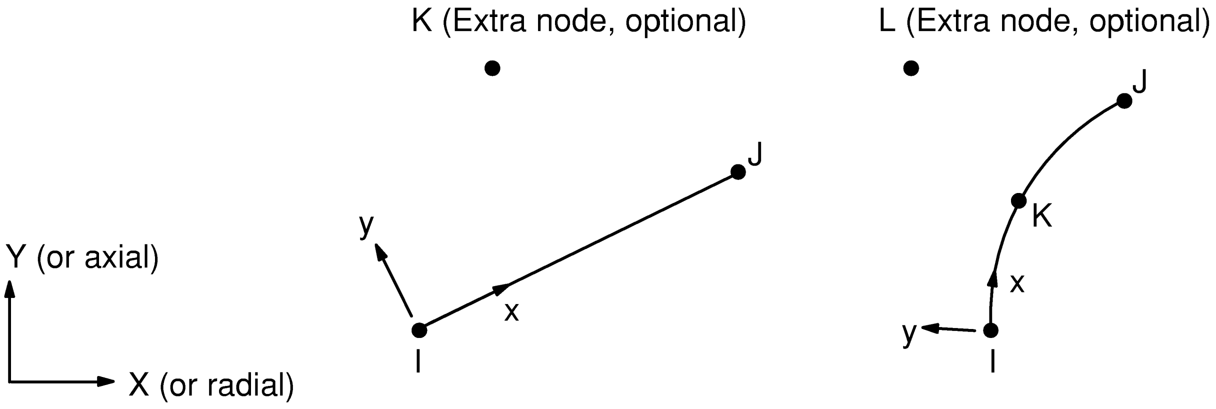
$f = \left\{ \begin{array}{l} 1.2 \\ 1.0 + .2\ \dfrac{A}{25t^2} \end{array} \right\}$, whichever is greater

A = element area (in s–t plane)

t = average thickness

The above definition of f is designed to avoid shear locking.

14.151 SURF151 — 2–D Thermal Surface Effect

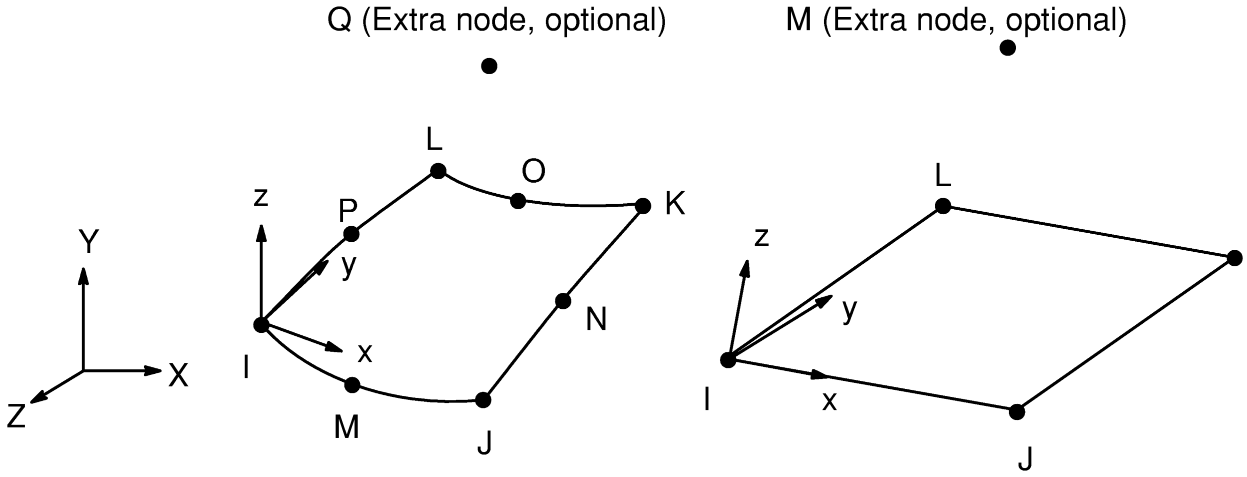


Matrix or Vector	Shape Functions	Integration Points
All	w = $C_1 + C_2x$ with no midside node w = $C_1 + C_2x + C_3x^2$ with midside node	2 2

Load Type	Distribution
All Loads	Same as shape functions

The logic is very similar to that given for SURF152 in Section 14.152.

14.152 SURF152 — 3–D Thermal Surface Effect



Matrix or Vector	Geometry	Midside Nodes	Shape Functions	Integration Points
Convection Surface Matrix and Load Vector	Quad	If KEYOPT(4)=0 (has midside nodes)	Equation (12.5.10–20)	3 x 3
		If KEYOPT(4)=1 (has no midside nodes)	Equation (12.5.8–20)	2 x 2
	Triangle	If KEYOPT(4)=0 (has midside nodes)	Equation (12.5.2–20)	6
		If KEYOPT(4)=0 (has no midside nodes)	Equation (12.5.1–20)	3
Heat Generation Load Vector	Same as convection surface matrix.			

Load Type	Distribution
All Loads	Same as shape functions

14.152.1 Matrices and Load Vectors

When the extra node is not present, the logic is the same as given and as described in Section 6.2. The discussion below relates to theory that uses the extra node.

The conductivity matrix is based on one–dimensional flow to and away from the surface. The form is conceptually the same as for LINK33 (equation (14.33–1)) except that the surface has four or eight nodes instead of only one node. Using the example of convection and no midside nodes are requested (KEYOPT(4) = 1) (resulting in a 5 x 5 matrix), the first four terms of the main diagonal are:

$$\int_{\text{area}} h_f \; \{N\} \; d(\text{area}) \qquad (14.152\text{–}1)$$

where: h_f = film coefficient (input on **SFE** command with *KVAL*=1)
h_u (If KEYOPT(5) = 1 and user programmable feature USRSURF116 output argument KEY(1) = 1, this definition supercedes the other.)

h_u = output argument for film coefficient of USRSURF116

{N} = vector of shape functions

which represents the main diagonal of the upper–left corner of the conductivity matrix. The remaining terms of this corner are all zero. The last main diagonal term is simply the sum of all four terms of expression (14.152–1) and the off–diagonal terms in the fifth column and row are the negative of the main diagonal of each row and column, respectively.

If midside nodes are present (KEYOPT(4) = 0) (resulting in a 9 x 9 matrix) expression (14.152–1) is replaced by:

$$\int_{\text{area}} h_f \; \{N\}\{N\}^T \; d(\text{area}) \qquad (14.152\text{–}2)$$

which represents the upper–left corner of the conductivity matrix. The last main diagonal is simply the sum of all 64 terms of expression (14.152–2) and the off–diagonal terms in the ninth column and row are the negative of the sum of each row and column respectively.

Radiation is handled similarly, except that the approach discussed for LINK31 in Section 14.31 is used. A load vector is also generated. The area used is the area of the element. The form factor is discussed in a subsequent section.

An additional load vector is formed when using the extra node by:

$$\{Q^c\} = [K^{tc}]\,\{T^{ve}\} \tag{14.152–3}$$

where: $\{Q^c\}$ = load vector to be formed

$[K^{tc}]$ = element conductivity matrix due to convection

$\{T^{ve}\}$ = $\lfloor 0 \quad 0 \quad \ldots \quad 0 \quad T_v^G \rfloor^T$

T_v^G = $\begin{cases} \text{output argument TEMVEL if the user programmable feature USRSURF116 is used.} \\ T_v \text{ if KEYOPT(6) = 1 (see next section)} \\ 0.0 \text{ for all other cases} \end{cases}$

TEMVEL from USRSURF116 is the difference between the bulk temperature and the temperatue of the extra node.

14.152.2 Adiabatic Wall Temperature as Bulk Temperature

There is special logic that accesses FLUID116 information where FLUID116 has had KEYOPT(2) set equal to 1. This logic uses SURF151 or SURF152 with the extra node present (KEYOPT(5) = 1) and computes an adiabatic wall temperature (KEYOPT(6) = 1). For this case, T_v, as used above, is defined as:

$$T_v = \frac{V^2\,F_r}{2C_p^f} \tag{14.152–4}$$

where: V = relative velocity (see equation (14.152–5))

F_r = recovery factor

= $\begin{cases} F_r^i & \text{if } F_r^i > 0 \\ (Pr)^n & \text{if } F_r^i = 0 \end{cases}$

F_r^i = input as NRF on **R** command

Pr = Prandtl number

= $\dfrac{C_p^f \cdot \nu^f}{K_x^f}$

C_p^f = specific heat of fluid (from FLUID116)

ν^f = viscosity of fluid (from FLUID116)

K_x^f = conductivity of the fluid (from FLUID116)

$$n = \begin{cases} 0.5000 & \text{if Re} < 2500.0 \\ 0.3333 & \text{if Re} > 2500.0 \end{cases}$$

Re = Reynold's number

$$= \frac{\rho^f \; V \; D}{\nu^f}$$

ρ^f = density of fluid (from FLUID116)

D = diameter of fluid pipe (from FLUID116)

The relative velocity is computed as:

$$V = \begin{cases} V_{116} & \text{if } \Omega - F_s \; \Omega_F = 0 \\ R \; (\Omega - F_s \; \Omega_F) & \text{if } \Omega - F_s \; \Omega_F \neq 0 \end{cases} \qquad (14.152\text{–}5)$$

where: V_{116} = velocity of fluid at extra node (from FLUID116)

Ω = angular velocity (input as OMEG on **R** command)

R = average radius of this element

F_s = slip factor (from FLUID116)

Ω_F = angular velocity of fluid (from FLUID116)

If V is computed to be negative, it is converted to be positive. The adiabatic wall temperature is reported as:

$$T_{aw} = T_{ex} + \frac{V^2}{2C^f_p} \; (F_r - 1) \qquad (14.152\text{–}6)$$

where: T_{aw} = adiabatic wall temperature

T_{ex} = temperature of extra node

14.152.3 Radiation Form Factor Calculation

The form factor is computed as:

$$F = \begin{cases} \text{input as FORMF on } \mathbf{R} \text{ command if KEYOPT(9) = 1} \\ \text{B} \qquad \text{if KEYOPT(9) = 2 or 3} \end{cases} \qquad (14.152\text{–}7)$$

also,

F = output quantity FORM FACTOR

Developing B further

$$B = \begin{cases} \cos\alpha \text{ if } \alpha \leq 90^\circ \\ -\cos\alpha \text{ if } \alpha > 90^\circ \text{ and KEYOPT(9) = 2} \\ 0 \quad \text{if } \alpha > 90^\circ \text{ and KEYOPT(9) = 3} \end{cases}$$

α = angle between element z axis at integration point being processed and the line connecting the integration point and the extra node (see Figure 14.152–1)

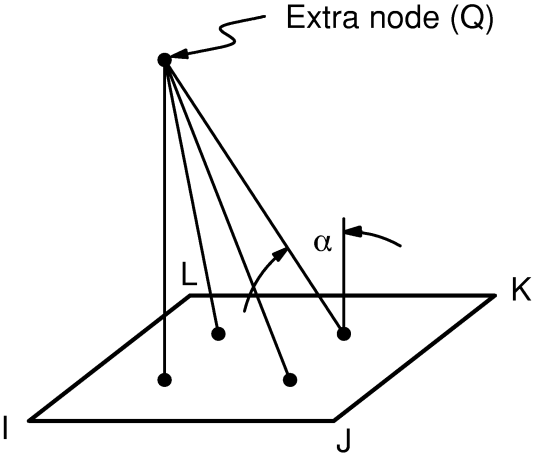


Figure 14.152–1 Form Factor Calculation

F is then used in the two–surface radiation equation:

$$Q_e^r = \sigma\, \epsilon\, A\, F \left(T^4 - T_Q^4\right) \tag{14.152–8}$$

where:

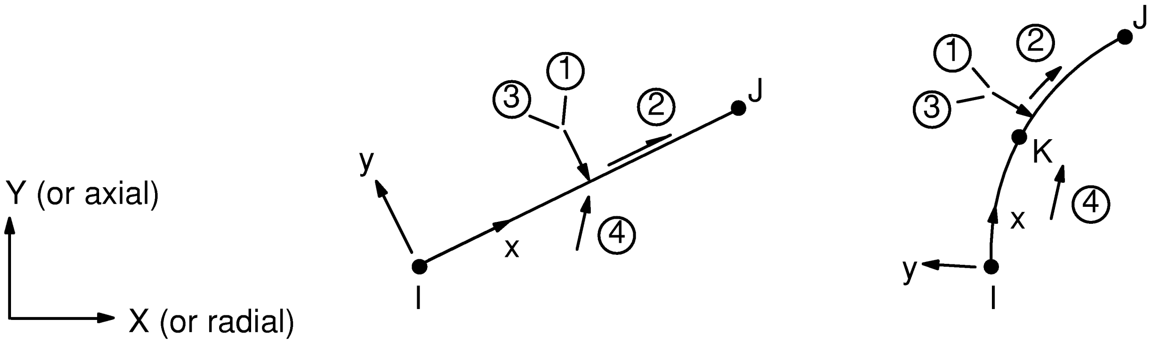
σ = Stephan Boltzman constant (input as SBCONST on **R** command)

ε = emissivity (input as EMIS on **MP** command)

A = element area

Note that this “form factor” does not have any distance affects. Thus, if distances are to be included, they must all be similar in size, as in an object on or near the earth being warmed by the sun. For this case, distance affects can be included by an adjusted value of σ.

14.153 SURF153 — 2–D Structural Surface Effect



Matrix or Vector	Shape Functions	Integration Points
All	$w = C_1 + C_2x$ with no midside node $w = C_1 + C_2x + C_3x^2$ with midside node	2 2

Load Type	Distribution
All Loads	Same as shape functions

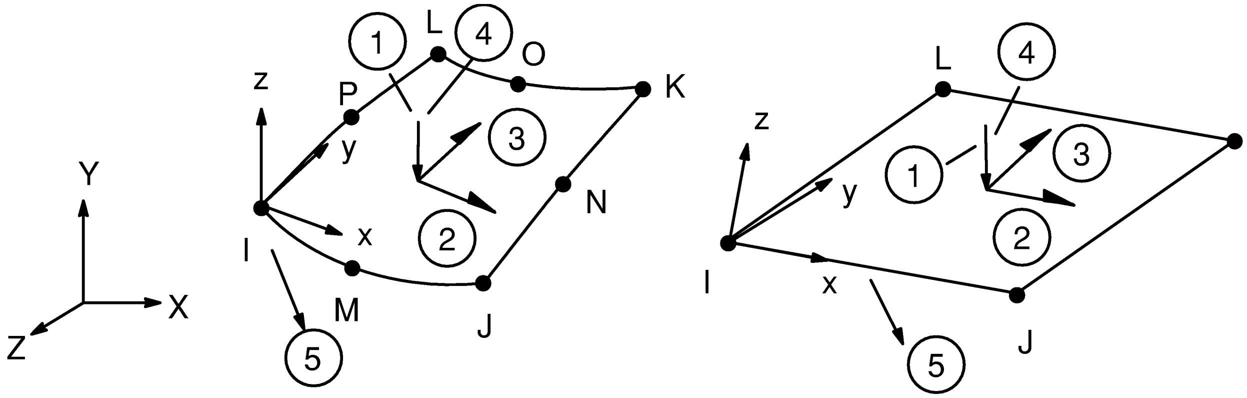
The logic is very similar to that given for SURF154 in Section 14.154 with the differences noted below:

1. For surface tension (SURT $\neq$ 0 where SURT is input on **R** command) in axisymmetric models (KEYOPT(3) = 1), an average force is used on both end nodes.

2. For surface tension with midside nodes, no load is applied at the middle node, and only the component directed towards the other end node is used.

3. For applied pressure on face 1, there is no pressure load stiffness matrix.

14.154 SURF154 — 3–D Structural Surface Effect



Matrix or Vector	Geometry	Midside Nodes	Shape Functions	Integration Points
Convection Surface Matrix and Load Vector	Quad	If KEYOPT(4)=0 (has midside nodes)	Equations (12.5.10–1), (12.5.10–2), and (12.5.10–3)	3 x 3
		If KEYOPT(4)=1 (has no midside nodes)	Equations (12.5.8–1), (12.5.8–2), and (12.5.8–3)	2 x 2
	Triangle	If KEYOPT(4)=0 (has midside nodes)	Equations (12.5.5–1), (12.5.5–2), and (12.5.5–3)	6
		If KEYOPT(4)=0 (has no midside nodes)	Equations (12.5.3–1), (12.5.3–2), and (12.5.3–3)	3

Matrix or Vector	Geometry	Midside Nodes	Shape Functions	Integration Points
Damping Matrix	Same as stiffness matrix			
Mass Matrix	Same as stiffness matrix			
Stress Stiffness Matrix	Same as stiffness matrix			
Pressure Load Vector	Same as stiffness matrix			
Surface Tension Load Vector	Same as stiffness matrix			

Load Type	Distribution
All Loads	Same as shape functions

The stiffness matrix is:

$$[K_e^f] = \text{element foundation stiffness matrix}$$

$$= k^f \int_A \{N_z\} \{N_z\}^T \, dA \tag{14.154–1}$$

k^f = foundation stiffness (input as EFS on **R** command)

A = area of element

$\{N_z\}$ = vector of shape functions representing motions normal to the surface

The mass matrix is:

$$[M_e] = \text{element mass matrix}$$

$$= \rho \int_A t_h \{N\}\{N\}^T \, dA + A_d \int_A \{N\}\{N\}^T dA \tag{14.154–2}$$

where: t_h = thickness (input as TKI, TKJ, TKK, TKL on **RMORE** command)

ϱ = density (input as DENS on **MP** command)

$\{N\}$ = vector of shape functions

A_d = added mass per unit area (input as ADMSUA on **R** command)

If the command **LUMPM**,ON is used, $[M_e]$ is diagonalized as described in Section 13.2.

The element damping matrix is:

$$[C_e] = \text{element damping matrix} = \mu \int_A \{N\}\{N\}^T \, dA \tag{14.154–3}$$

where: μ = dissipation (input as VISC on **MP** command)

The element stress stiffness matrix is:

$$[S_e] = \text{element stress stiffness matrix} = \int_A [S_g]^T [S_m][S_g] \, dA \tag{14.154–4}$$

where: $[S_g]$ = derivatives of shape functions of normal motions

$$[S_m] = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

s = in–plane force per unit length (input as SURT on **R** command)

If pressure is applied to face 1, the pressure load stiffness matrix is computed as described in Section 3.3.4.

The element load vector is:

$$\{F_e\} = \left\{F_e^{st}\right\} + \left\{F_e^{pr}\right\} \tag{14.154–5}$$

where: $\left\{F_e^{st}\right\}$ = surface tension force vector

$$= s \int_E \left\{N_p\right\} dE$$

$\{N_p\}$ = vector of shape functions representing in–plane motions normal to the edge

E = edge of element

$\left\{F_e^{pr}\right\}$ = pressure load vector

$$= \int_A \Big(\{N_x\} \, P_x + \{N_y\} \, P_y - \{N_z\} \, P_z$$

$$+ P_v \, Z_f \left(p_x \, \{N_X\} + p_y \, \{N_Y\} + p_z \, \{N_Z\} \right) \Big) \, dA$$

$\{N_x\}$ = vector of shape functions representing motion in element x direction

$\{N_y\}$ = vector of shape functions representing motion in element y direction

$P_x, \ P_y, \ P_z$ = distributed pressures over element in element x, y, and z directions (input quantities *VAL1* thru *VAL4* with *LKEY* = 2, 3, 1, respectively, on **SFE** command)

P_v = uniform pressure magnitude

$$P_v = \begin{cases} P_1 \cos\theta & \text{if KEYOPT(11) = 0 or 1} \\ P_1 & \text{if KEYOPT(11) = 2} \end{cases}$$

P_1 = input quantity VAL1 with LKEY = 5 on **SFE** command

θ = angle between element normal and applied load direction

$$Z_f = \begin{cases} 1.0 & \text{if KEYOPT(12) = 0 or } \cos\theta \le 0.0 \\ 0.0 & \text{if KEYOPT(12) = 1 and } \cos\theta > 0.0 \end{cases}$$

$$p_x = \begin{cases} D_x / \sqrt{D_x^2 + D_y^2 + D_z^2} & \text{if KEYOPT(11)} \ne 1 \\ 0.0 & \text{if KEYOPT(11) = 1} \end{cases}$$

$$p_y = \begin{cases} D_y / \sqrt{D_x^2 + D_y^2 + D_z^2} & \text{if KEYOPT(11)} \ne 1 \\ 0.0 & \text{if KEYOPT(11) = 1} \end{cases}$$

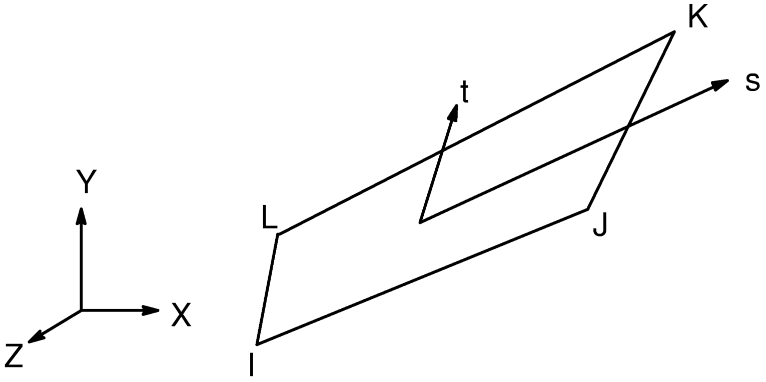
$$p_z = D_z / \sqrt{D_x^2 + D_y^2 + D_z^2}$$

$D_x, \ D_y, \ D_z$ = vector directions (input quantities VAL2 thru VAL4 with LKEY = 5 on **SFE** command)

$\{N_X\}, \ \{N_Y\}, \ \{N_Z\}$ = vectors of shape functions in global Cartesian coordinates

The integration used to arrive at $\{F_e^{pr}\}$ is the usual numerical integration, even if KEYOPT(6) $\ne$ 0. The output quantities “average face pressures” are the average of the pressure values at the integration points.

14.157 SHELL157 — Coupled Thermal–Electric Shell

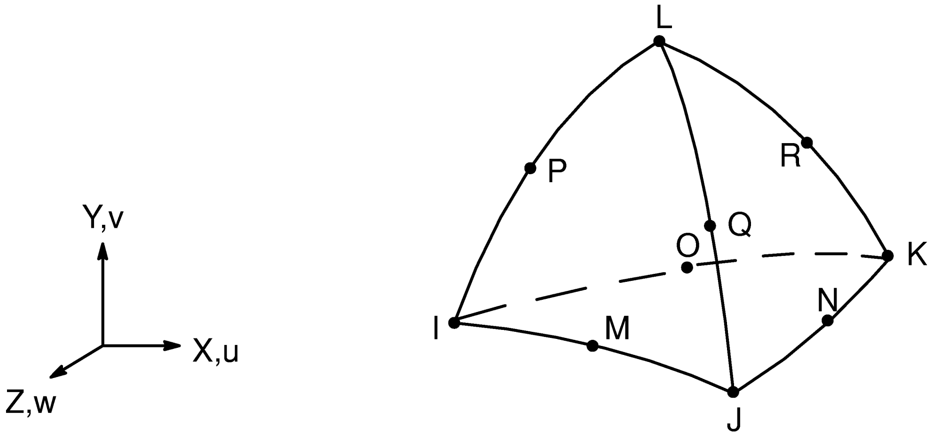


Matrix or Vector	Shape Functions	Integration Points
Conductivity Matrix	Equations (12.5.8–20) and (12.5.8–21)	2 x 2
Specific Heat Matrix	Same as conductivity matrix. Matrix is diagonalized as described in Section 13.2	2 x 2
Heat Generation Load Vector	Same as conductivity matrix	2 x 2
Convection Surface Matrix and Load Vector	Same as conductivity matrix	2 x 2

14.157.1 Other Applicable Sections

Section 11.2 discusses thermal–electric elements. Section 13.1 describes integration point locations.

14.158 HYPER158 — 3–D 10–Node Tetrahedral Mixed U–P Hyperelastic Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Mass Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Thermal Load Vector	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Pressure Load Vector	Equation (12.8.2–1), (12.8.2–2), and (12.8.2–3) specialized to the face	6

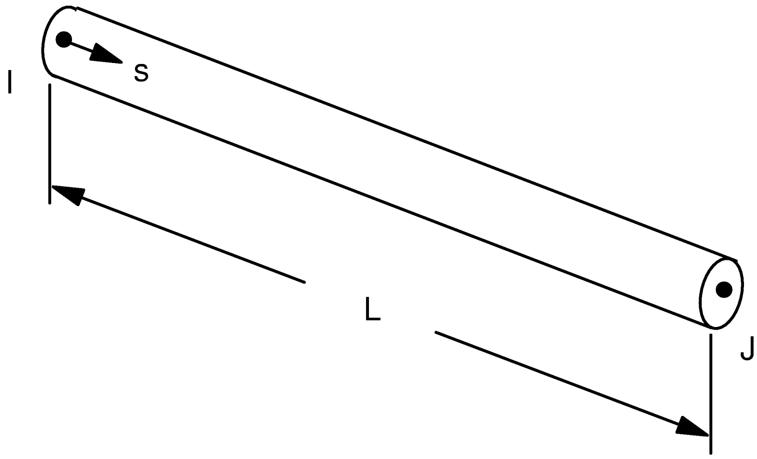
Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Pressure	Linear over each face

Reference: Oden and Kikuchi(123), Sussman and Bathe(124)

14.158.1   Other Applicable Sections

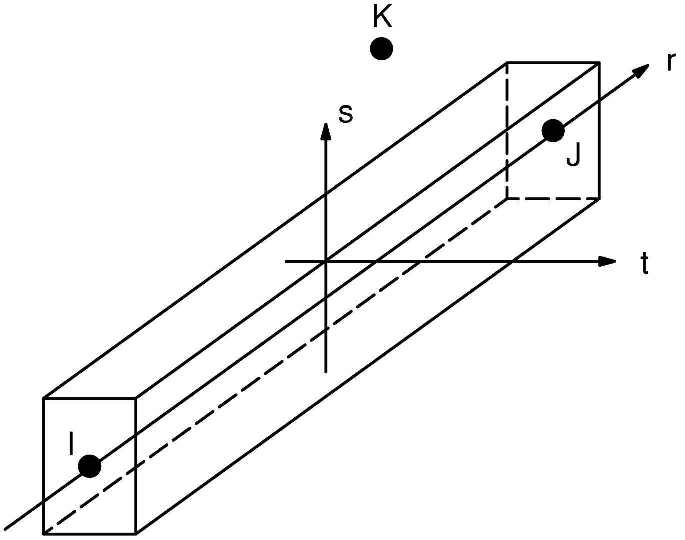
For the basic element formulation, refer to Section 14.58. The hyperelastic material model (Mooney–Rivlin) is described in 4.5. Section 13.1 describes integration point locations.

14.160 LINK160 — Explicit 3–D Spar



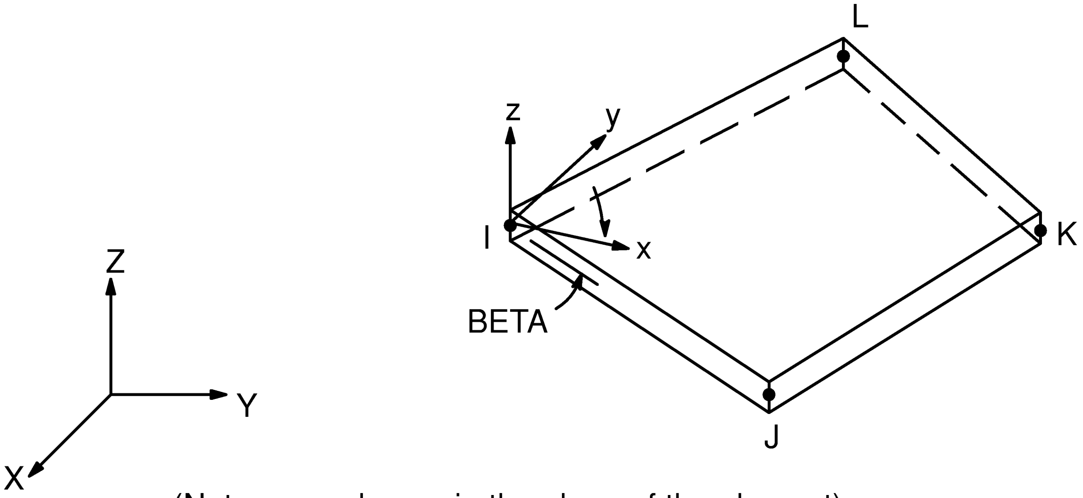
For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

14.161 BEAM161 — Explicit 3–D Beam



For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

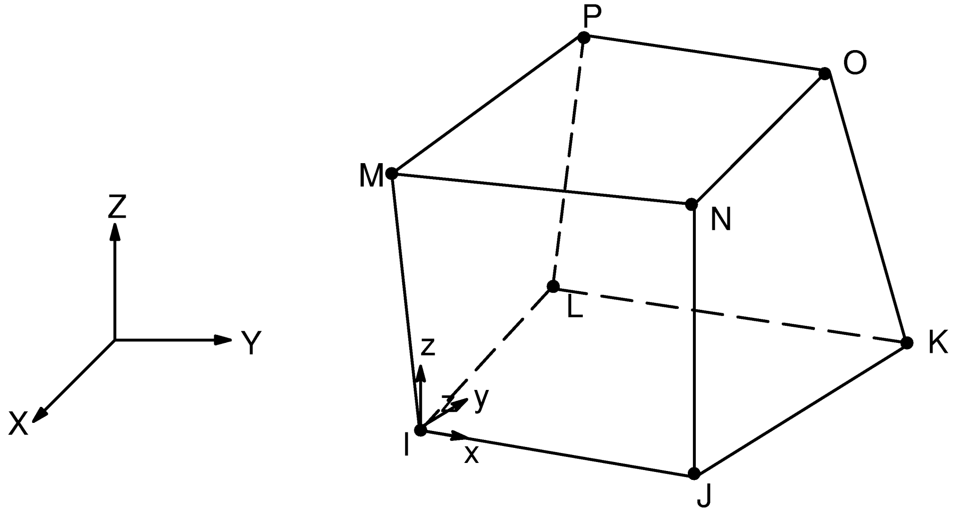
14.163 SHELL163 — Explicit Thin Structural Shell



(Note – x and y are in the plane of the element)

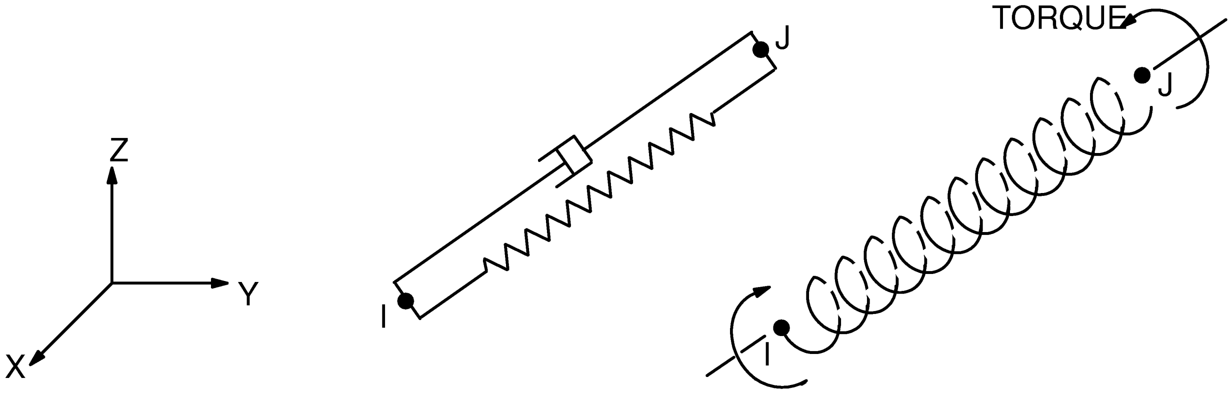
For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

14.164 SOLID164 — Explicit 3–D Structural Solid



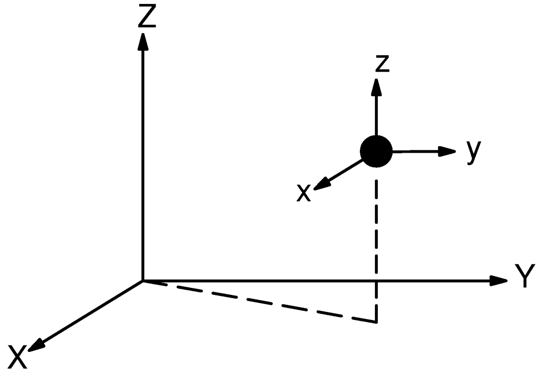
For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

14.165 COMBI165 — Explicit Spring–Damper



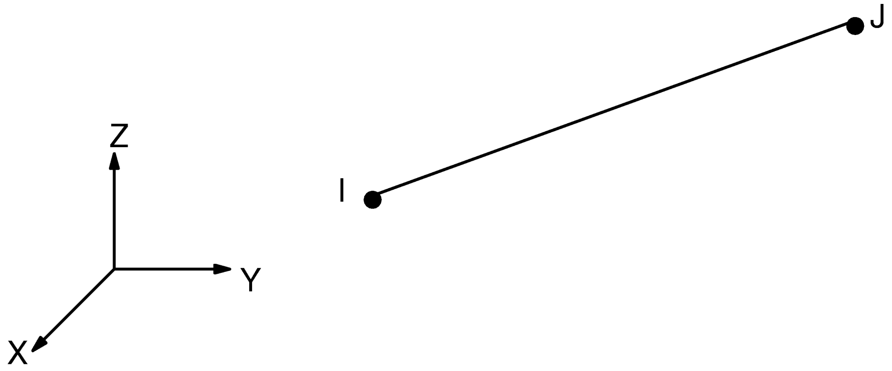
For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

14.166 MASS166 — Explicit 3–D Structural Mass



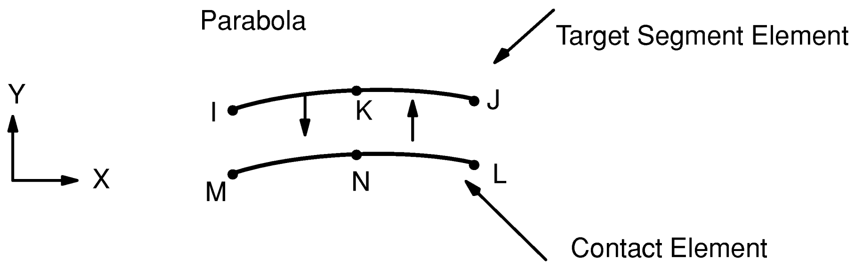
For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

14.167 LINK167 — Explicit Tension–Only Spar



For all theoretical information about this element, see the LS–DYNA3D Theoretical Manual(199).

14.169 TARGE169 — 2–D Target Segment



14.169.1 Other Applicable Sections

Section 14.170 discusses Target Elements

14.169.2 Segment Types

TARGE169 supports six 2–D segment types; see Figure 14.169–1

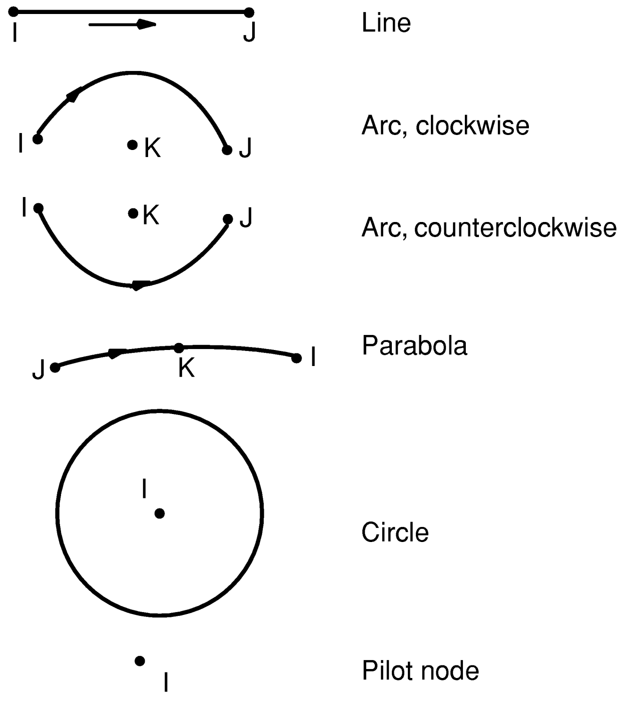
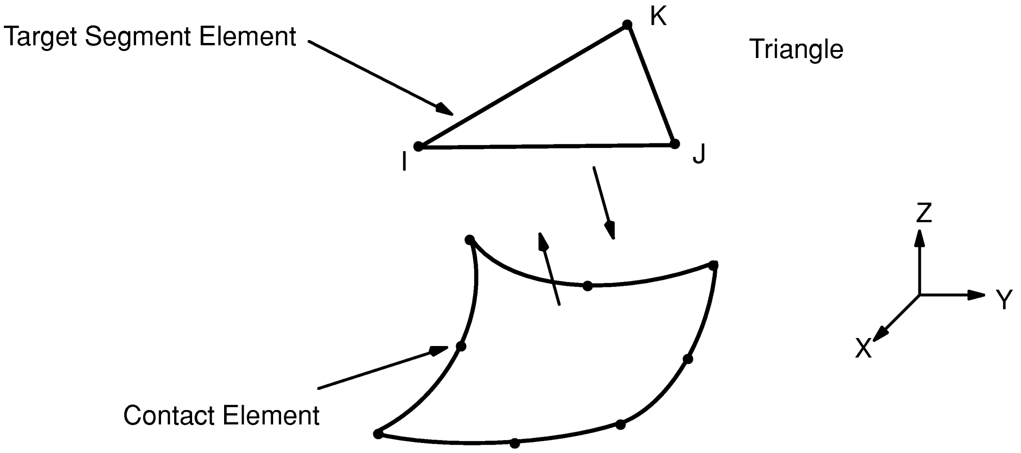


Figure 14.169–1 2–D Segment Types

14.170 TARGE170 — 3–D Target Segment



14.170.1 Introduction

In studying the contact between two bodies, the surface of one body is conventionally taken as a contact surface and the surface of the other body as a target surface. The “contact–target” pair concept has been widely used in finite element simulations. For rigid–flexible contact, the contact surface is associated with the deformable body; and the target surface must be the rigid surface. For flexible–flexible contact, both contact and target surfaces are associated with deformable bodies. The contact and target surfaces constitute a “Contact Pair”.

TARGE170 is used to represent various 3–D target surfaces for the associated contact elements (CONTA173 and CONTA174). The contact elements themselves overlay the solid elements describing the boundary of a deformable body that is potentially in contact with the rigid target surface, defined by TARGE170. Hence, a “target” is simply a geometric entity in space that senses and responds when one or more contact elements move into a target segment element.

14.170.2 Segment Types

The target surface is modelled through a set of target segments; typically several target segments comprise one target surface. Each target segment is a single element with a specific shape or segment type. TARGE170 supports eight 3–D segment types; see Figure 14.170–1

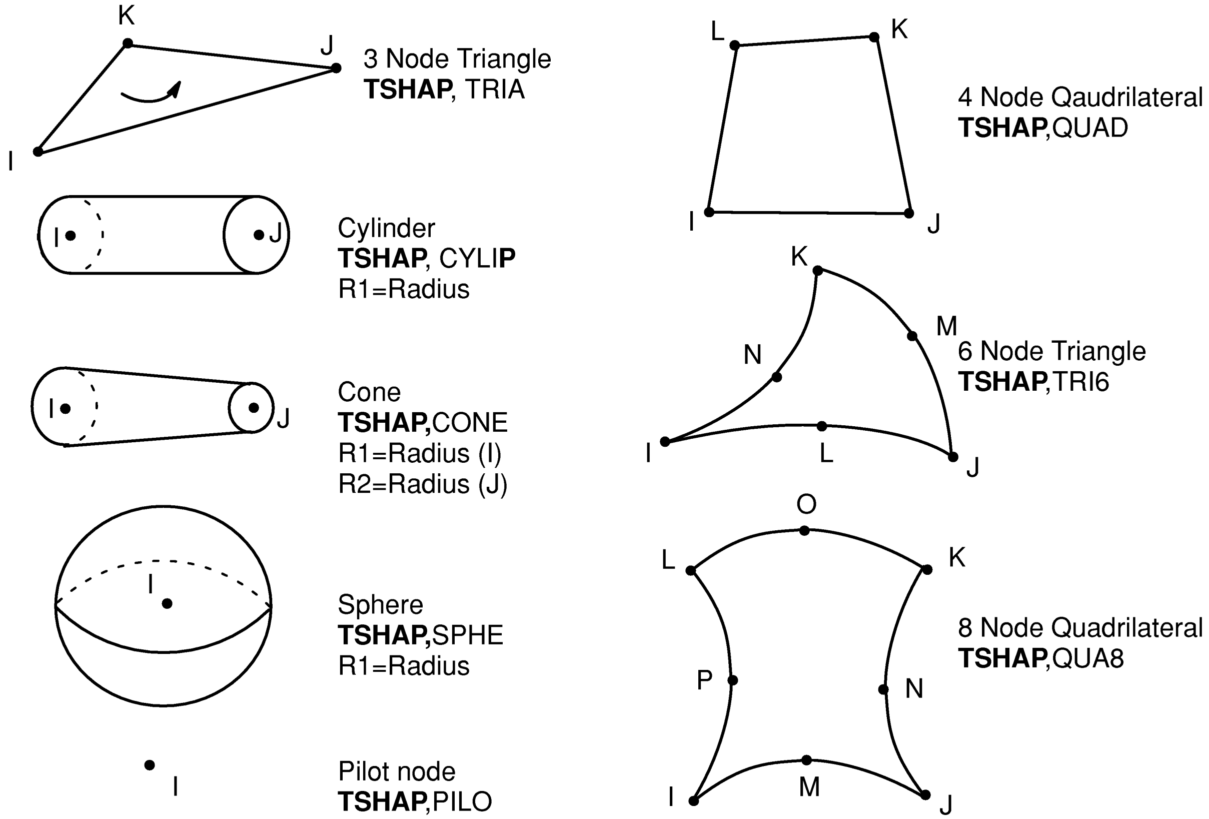
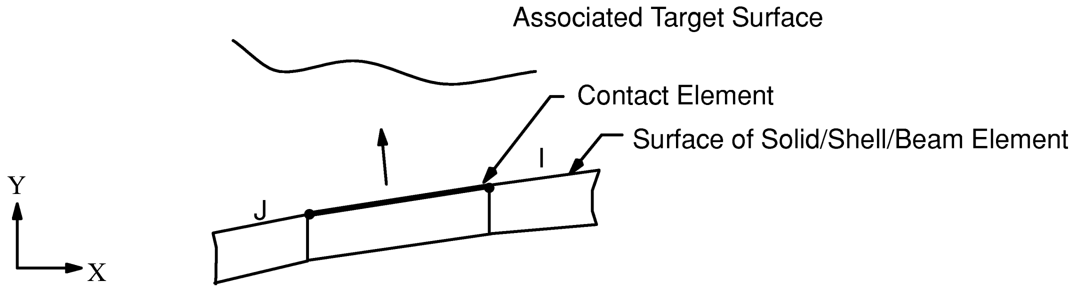


Figure 14.170–1 3–D Segment Types

14.170.3 Reaction Forces

The reaction forces on the entire rigid target surface are obtained by summing all the nodal forces of the associated contact elements. The reaction forces are accumulated on the pilot node. If the pilot node has not been explicitly defined by the user, one of the target nodes (generally the one with the smallest number) will be used to accumulate the reaction forces.

14.171 CONTA171 — 2–D Surface–to–Surface Contact

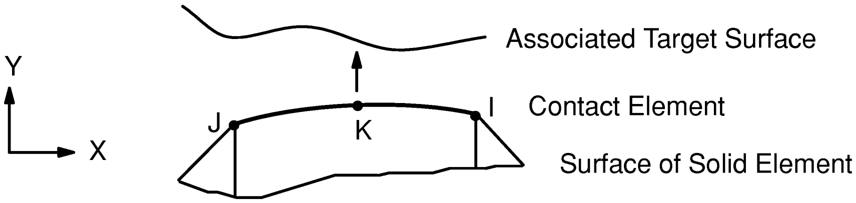

Associated Target Surface
Contact Element
Surface of Solid/Shell/Beam Element
I
J
Y
X

Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	$W = C_1 + C_2 x$	2

14.171.1 Other Applicable Sections

The CONTA171 description is the same as for CONTA174 (Section 14.174) except that it is 2–D and there are no midside nodes.

14.172 CONTA172 — 2–D 3–Node Surface–to–Surface Contact

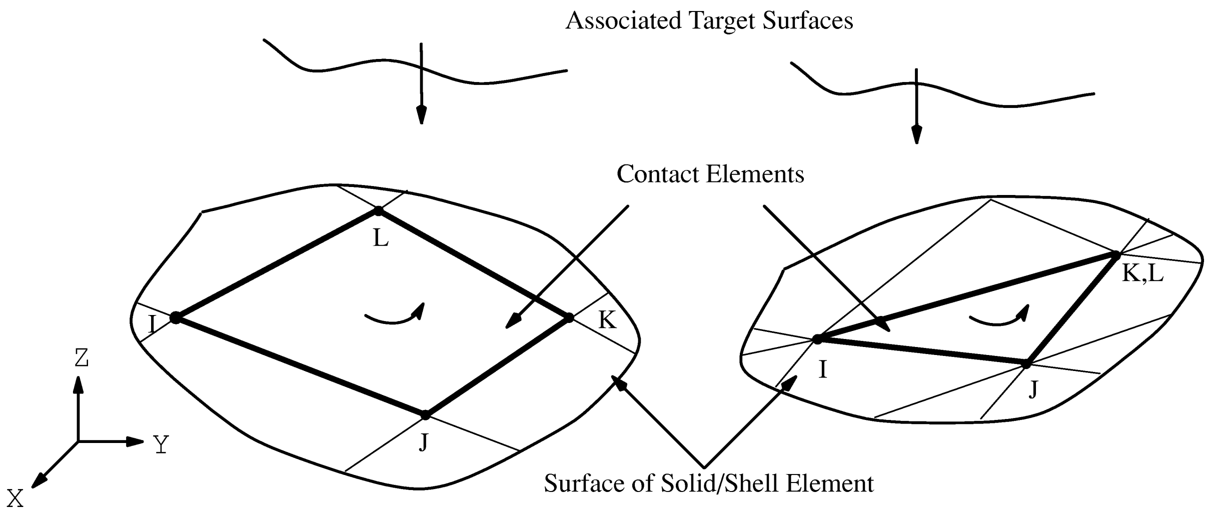


Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	$W = C_1 + C_2 x + C_3 x^2$	2

14.172.1 Other Applicable Sections

The CONTA172 description is the same as for CONTA174 (Section 14.174) except that it is 2–D.

14.173 CONTA173 — 3–D Surface–to–Surface Contact

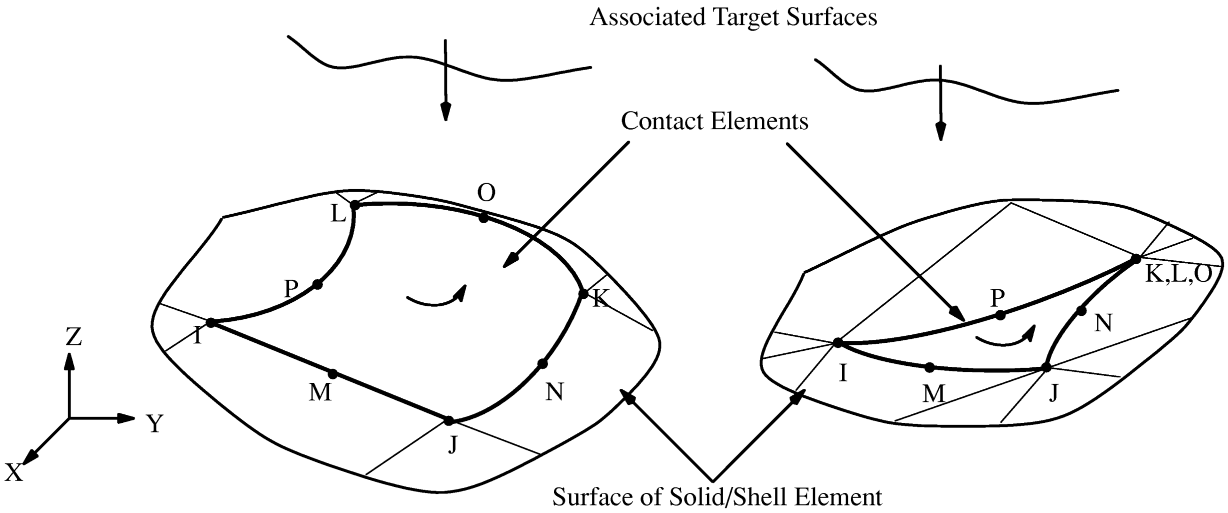


Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix and Stress Stiffness Matrix	Quad	Equations (12.5.8–1), (12.5.8–2) and (12.5.8–3)	2 x 2
	Triangle	Equations (12.5.3–1), (12.5.3–2), and (12.5.3–3)	3

14.173.1 Other Applicable Sections

The CONTA173 description is the same as for CONTA174 (Section 14.174) except there are no midside nodes.

14.174 CONTA174 — 3–D 8–Node Surface–to–Surface Contact



Matrix or Vector	Geometry	Shape Functions	Integration Points
Stiffness Matrix and Stress Stiffness Matrix	Quad	If KEYOPT(4) = 0 (has midside nodes) equations (12.5.10–1), (12.5.10–2) and (12.5.10–3)	2 x 2
	Triangle	If KEYOPT(4) = 0 (has midside nodes) equations (12.5.5–1), (12.5.5–2) and (12.5.5–3)	3

14.174.1 Introduction

CONTA174 is an 8–node element that is intended for general rigid–flexible and flexible–flexible contact analysis. In a general contact analysis, the area of contact between two (or more) bodies is generally not known in advance. CONTA174 is applicable to 3–D geometries. It may be applied to contact of solid bodies, or shells, to static or dynamic analyses, to problems with or without friction.

14.174.2 Contact Kinematics

Contact Pair

In studying the contact between two bodies, the surface of one body is conventionally taken as a contact surface and the surface of the other body as a target surface. The “contact–target” pair concept has been widely used in finite element simulations. For rigid–flexible contact, the contact surface is associated with the deformable body; and the target surface must be the rigid surface. For flexible–flexible contact, both contact and target surfaces are associated with deformable bodies. The contact and target surfaces constitute a “Contact Pair”.

CONTA174 contact element is associated with the 3–D target segment elements (TARGE170) via a shared real contact set number. This element is located on the surface of 3–D solid, shell element~~s~~ (called underlying element) without midside node. It has the same geometric characteristics as the underlying elements. The contact surface can be either/both side of the shell or beam elements.

Location of Contact Detection

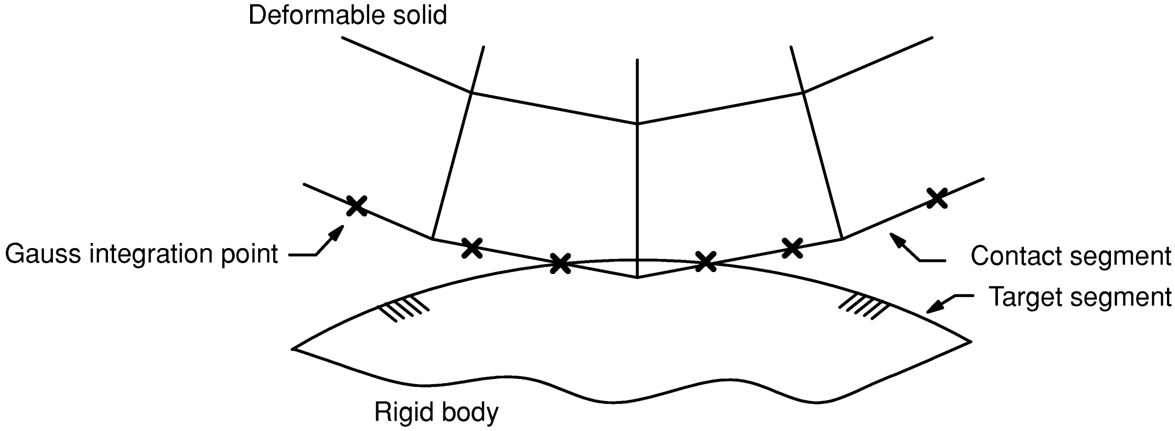


Figure 14.174–1 Contact Detection Point Location at Gauss Point

CONTA174 is surface–to–surface contact element. The contact detection points (i.e. the integration points) are located either at nodal points or Gauss points. The contact elements is constrained against penetration into target surface at its integration points. However, the target surface can, in principle, penetrate through into the contact surface. See Figure 14.174–1. CONTA174 uses Gauss integration points as a default (Cescotto and Charlier(213), Cescotto and Zhu(214)), which generally provides more accurate results than those using nodes themselves as the integration points. Also the contact node may slip off the edge of the rigid surface when contact detection points are located at nodal level. If this happens, the contact state will be lost and the convergence difficulties will follow. Other disadvantage with the use of nodal contact points is that: when for a uniform pressure, the kinemtically equivalent forces at the nodes are unrepresentative and indicate release at corners.

Penetration Distance

The penetration distance is measured along the normal direction of contact surface located at integration points to the target surface (Cescotto and Charlier(214)). See Figure 14.174–2. It is uniquely defined even the geometry of the target surface is not smooth. Such discontinuities may be due to physical corners on the target surface, or may be introduced by a numerical discretization process (e.g. finite elements). Based on the present way of calculating penetration distance there are no restriction on the shape of the rigid target surface. Smoothing is not always necessary typically for the concave corner. For the convex corner, we still recommend the user to smooth out the region of abrupt curvature changes (see Figure 14.174–3).

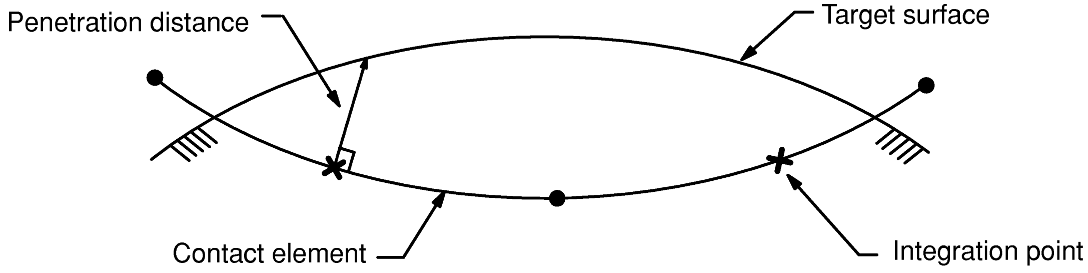


Figure 14.174–2 Penetration Distance

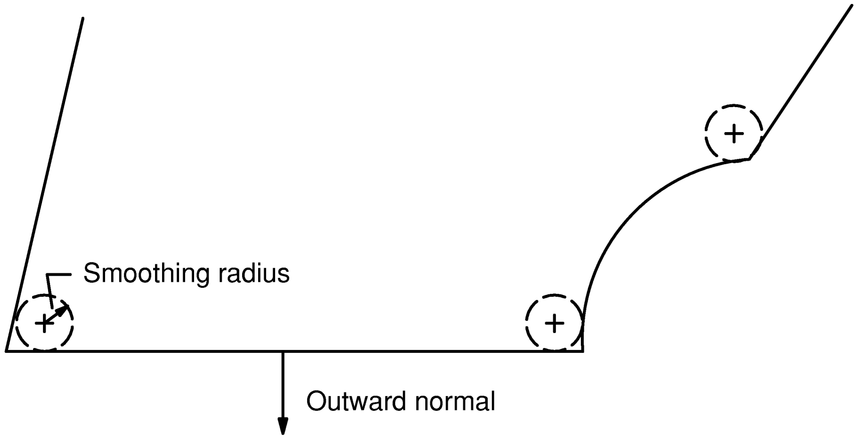


Figure 14.174–3 Smoothing Convex Corner

Pinball Algorithm

The position and the motion of a contact element relative to its associated target surface determines the contact element status. The program monitors each contact element and assigns a status:

STAT=0 Open far–field contact
STAT=1 Open near–field contact
STAT=2 Sliding contact
STAT=3 Sticking contact

A contact element is considered to be in near–field contact element enters a pinball region, which is centered on the integration point of the contact element. The computational cost of searching for contact depends on the size of the pinball region. Far–field contact element calculations are simple and add little computational demands. The near–field calculations (for contact elements that are nearly or actually in contact) are slower and more complex. The most complex calculations occur the elements are in actual contact.

Setting a proper pinball region is useful to overcome spurious contact definitions if the target surface has several convex regions. The current default setting should be appropriate for most contact problems.

14.174.3 Frictional Model

Coulomb's Law

In the basic Coulomb friction model, two contacting surfaces can carry shear stresses up to a certain magnitude across their interface before they start sliding relative to each other. The state is known as sticking. The Coulomb friction model is defined as:

6/30/99 yyz dv−5563 contact cohesion

$$\tau_{\text{lim}} = \mu P + b \tag{14.174–1}$$

$$|\tau| \le \tau_{\text{lim}} \tag{14.174–2}$$

where:

τ_{lim} = limit shear stress

τ = equivalent shear stress

μ = frictional coefficient (input as MU on **MP** command)

P = contact normal pressure

b = contact cohesion (input as COHE on **R** command)

Once the equivalent shear stress exceeds τ_{lim}, the contact and target surfaces will slide relative to each other. This state is known as sliding. The sticking/sliding calculations determine when a point transitions from sticking to sliding or vice versa. The contact cohesion provides sliding resistance even with zero normal pressure,

CONTA174 provides an option for defining a maximum equivalent shear stress τ_{max} (input as TAUMAX on **RMORE** command) so that, regardless of the magnitude of the contact pressure, sliding will occur if the magnitude of the equivalent shear stress reaches this value. See Figure 14.174–4.

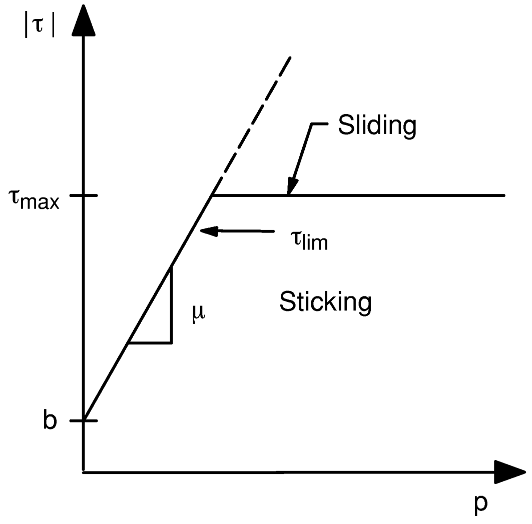


Figure 14.174–4 Friction Model

Integration of Frictional Law

The integration of the frictional mode is similar to that of non–associated theory of plasticity (see Section 4.1). In each substep that sliding friction occurs, an elastic predictor is computed in contact traction space. The predictor is modified with a radial return mapping function, providing both a small elastic deformation along sliding response as developed by Giannakopoulos(135).

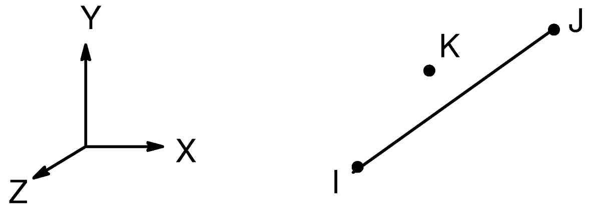
Algorithmic Symmetrization

Contact problems involving friction produce non–symmetric stiffness. Using an unsymmetric solver is more computationally expensive than a symmetric solver for each iteration. For this reason, a symmetrization algorithm developed by Laursen and Simo(216) is used by which most frictional contact problems can be solved using solvers for symmetric systems. If frictional stresses have a substantial influence on the overall displacement field and the magnitude of the frictional stresses is highly solution dependent, any symmetric approximation to the stiffness matrix may provide a low rate of convergence. In such cases, we had to use unsymmetric stiffness to improve convergence.

14.174.4 Contact Algorithm

For this surface–to–surface contact elements, either the augmented Lagrangian method (Simo and Laursen(215)) or the penalty method is used. The augmented Lagrangian method is an iterative series of penalty updates to find the exact Lagrange multiplies (.i.e. contact tractions). Compared to the penalty method, the augmented Lagrangian method usually leads to better conditioning and is less sensitive to the magnitude of contact penalty coefficient.

14.179 PRETS179 — Pre–tension



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	None	None

Load Type	Distribution
Pre–tension Force	Applied on pre–tension node K across entire pre–tension section

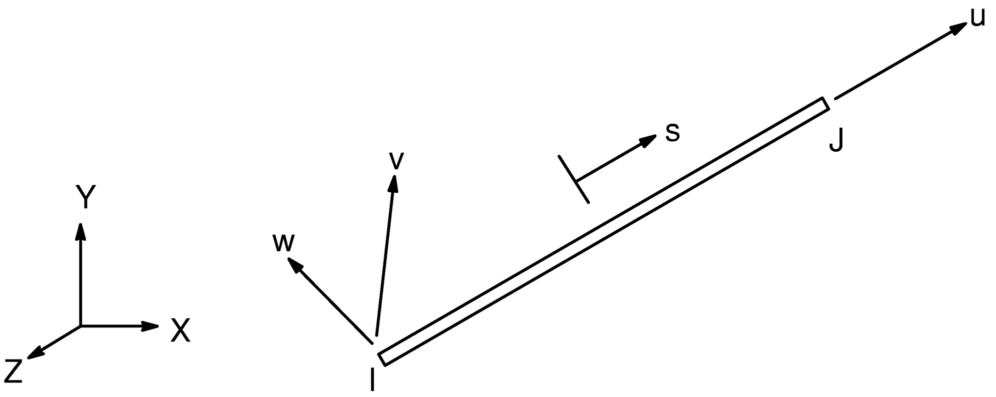
14.179.1 Introduction

The element is used to represent a two or three dimensional section for a bolted structure. The pre–tension section can carry a pre–tension load. The pre–tension node (K) on each section is used to control and monitor the total tension load.

14.179.2 Assumptions and Restrictions

The pre–tension element is not capable of carrying bending or torsion loads.

14.180 LINK180 — 3–D Finite Strain Spar (or Truss)



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equation (12.2.1–1)	1
Mass Matrix	Equations (12.2.1–1), (12.2.1–2), and (12.2.1–3)	1
Stress Stiffening Matrix	Same as mass matrix	1
Thermal Load Vector	Same as stiffness matrix	1
Newton Raphson Load Vector	Same as stiffness matrix	1

Load Type	Distribution
Element Temperature	Linear along length
Nodal Temperature	Linear along length

Reference: Cook et al(117)

14.180.1 Assumptions and Restrictions

The theory for this element is a reduction of the theory for BEAM189. The reductions include only 2 nodes, no bending or shear effects, no pressures, and the entire element as only one integration point.

The element is not capable of carrying bending loads. The stress is assumed to be uniform over the entire element.

14.180.2 Element Mass Matrix

All element matrices and load vectors described below are generated in the element coordinate system and are then converted to the global coordinate system. The element stiffness matrix is:

The element mass matrix is:

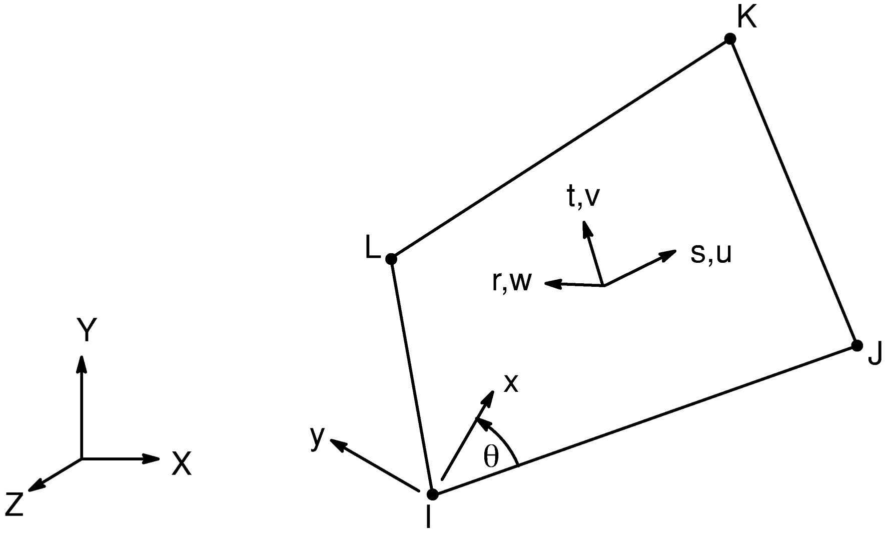
$$\left[M_\ell\right] = \frac{\rho A L}{2}\left[\begin{array}{ccc:ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \qquad (14.180\text{–}1)$$

where: A = element cross–sectional area (input as AREA on **R** command)

L = element length

ϱ = density (input as DENS on **MP** command)

14.181 SHELL181 — Large Strain Shell



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.5.13–1), (12.5.13–2), and (12.5.13–3)	In–plane: 2 x 2 (KEYOPT(3) = 1, 2) 1 x 1 (KEYOPT(3) = 0) Thru–the–thickness: 5
Mass Matrix	Equations (12.5.8–1), (12.5.8–2), and (12.5.8–3)	Closed form integration
Stress Stiffness Matrix	Same as mass matrix	Same as stiffness matrix
Thermal Load Matrix	Same as stiffness matrix	Same as stiffness matrix
Transverse Pressure Load Vector	Equation (12.5.8–3)	2 x 2
Edge Pressure Load Vector	Equations (12.5.8–1) and (12.5.8–2) specialized to the edge	2

Load Type	Distribution
Element Temperature	Bilinear in plane of element, linear thru thickness
Nodal Temperature	Bilinear in plane of element, constant thru thickness
Pressure	Bilinear in plane of element and linear along each edge

References: Ahmad(1), Cook(5), Dvorkin(96), Dvorkin(97), Bathe and Dvorkin(98), Allman(113), Cook(114), MacNeal and Harder(115)

14.181.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.181.2 Assumptions and Restrictions

Normals to the centerplane are assumed to remain straight after deformation, but not necessarily normal to the centerplane.

Each pair of integration points (in the r direction) is assumed to have the same element (material) orientation.

14.181.3 Assumed Displacement Shape Functions

The assumed displacement and transverse shear strain shape functions are given in Chapter 12. The basic functions for the transverse shear strain have been changed to avoid shear locking (Dvorkin(96), Dvorkin(97), Bathe and Dvorkin(98)) and are pictured in Figure 14.181–1.

14.181.4 Warping

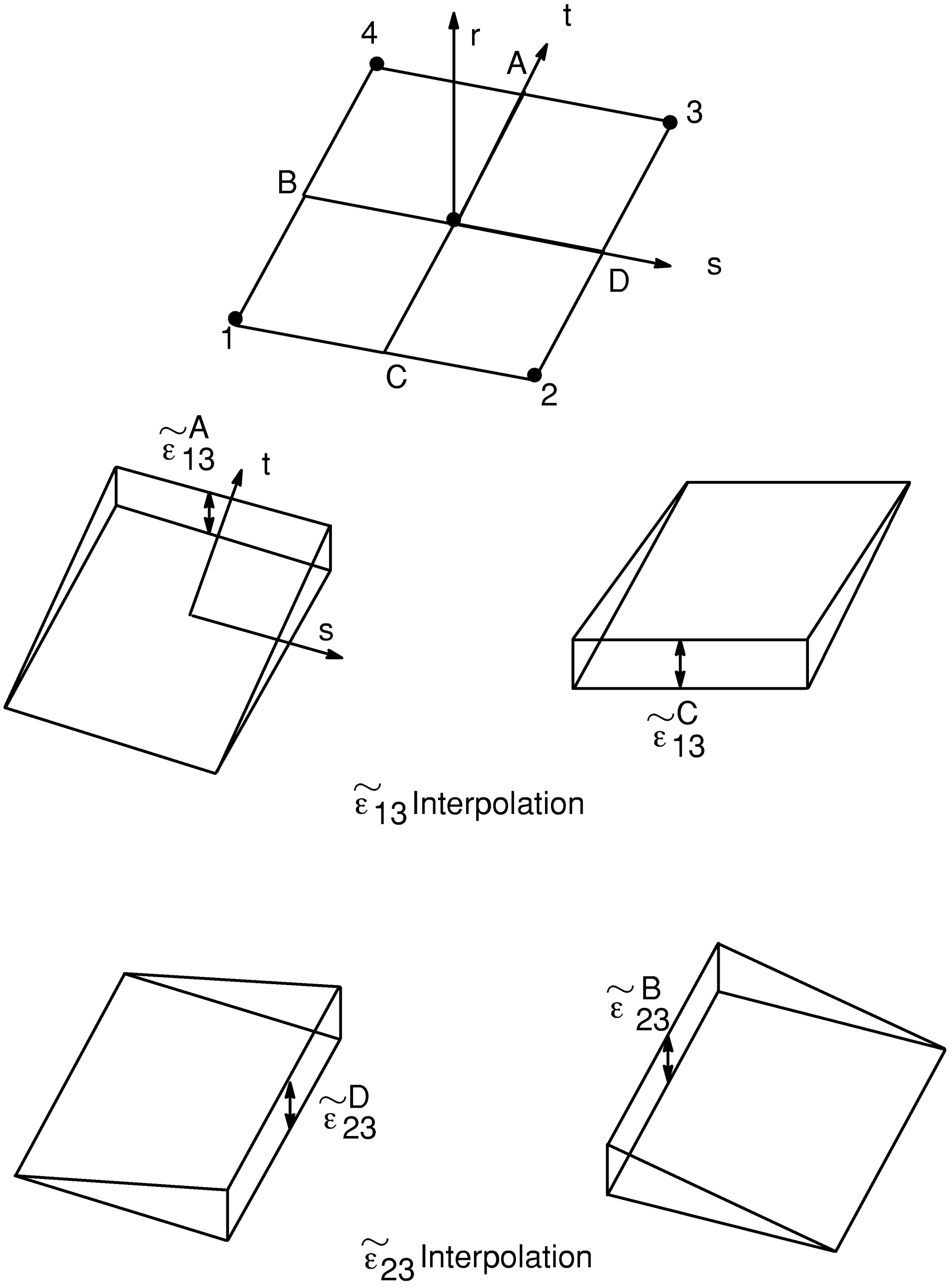
A warping factor is computed as:

$$\phi = \frac{D}{t} \tag{14.43–7}$$

where: D = component of the vector from the first node to the fourth node parallel to the element normal

t = average thickness of the element

If $\phi > 1.0$, a warning message is printed.

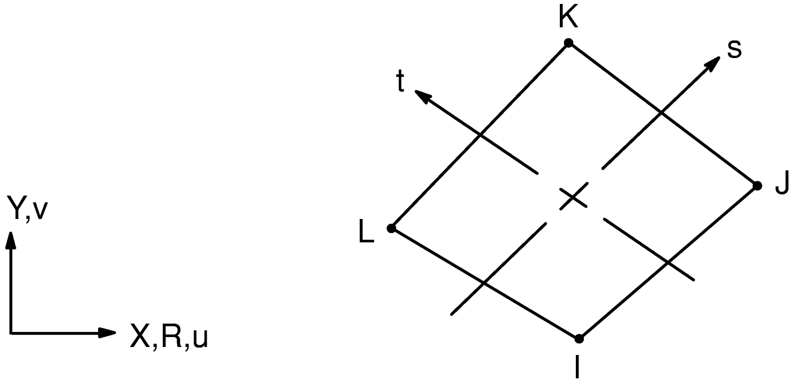

4
r
t
A
3
B
s
D
1
C
2
$\tilde{\varepsilon}_{13}^{A}$
t
s
$\tilde{\varepsilon}_{13}^{C}$
$\tilde{\varepsilon}_{23}^{D}$
$\tilde{\varepsilon}_{23}^{B}$

$\tilde{\varepsilon}_{13}$ Interpolation

$\tilde{\varepsilon}_{23}$ Interpolation

Figure 14.181–1 Shape Functions for the Transverse Strains

14.182 PLANE182 — 2–D Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.6.5–1) and (12.6.5–2)	See Section 4.182 of the Element Reference Manual
Mass Matrix	Same as stiffness matrix	Same as stiffness matrix
Stress Stiffness Matrix	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to face	2

Load Type	Distribution
Element Temperature	Bilinear across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

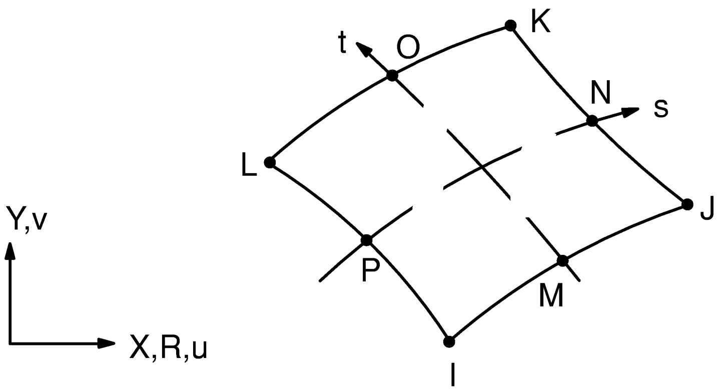
14.182.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.182.2 Theory

This element uses $\overline{\mathrm{B}}$ method (selective reduced integration technique for volumetric terms) (Hughes(219), Nagtegaal et al(220)) if KEYOPT(1) = 0. If KEYOPT(1) = 1, the uniform reduced integration technique (Flanagan and Belytschko(232)) is used.

14.183 PLANE183 — 2–D 8–Node Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.6.7–1) and (12.6.7–2)	2 x 2
Mass Matrix	Same as stiffness matrix	3 x 3
Stress Stiffness Matrix	Same as stiffness matrix	Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix, specialized to the face	2 along face

Load Type	Distribution
Element Temperature	Same as shape functions across element, constant thru thickness or around circumference
Nodal Temperature	Same as element temperature distribution
Pressure	Linear along each face

Reference: Zienkiewicz(39)

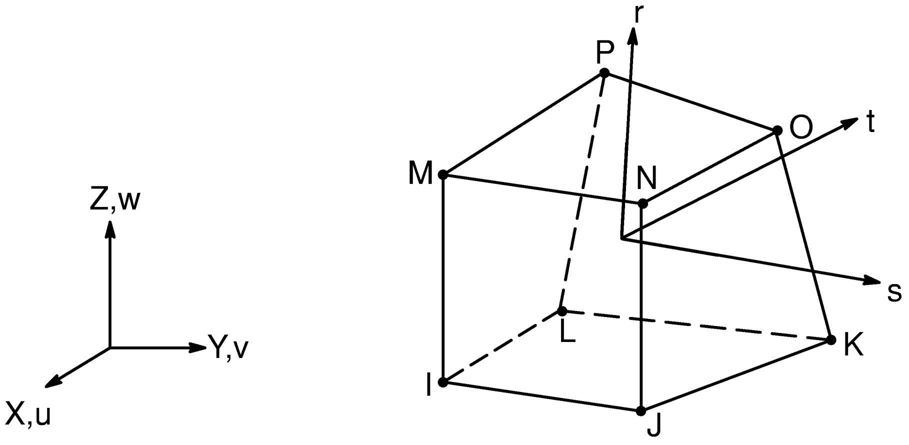
14.183.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.183.2 Assumptions and Restrictions

A dropped midside node implies that the face is and remains straight.

14.185 SOLID185 — 3–D Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.18–1), (12.8.18–2), and (12.8.18–3)	See Section 4.185 of the Element Reference Manual
Mass Matrix	Same as stiffness matrix	2 x 2 x 2
Stress Stiffness Matrix	Same as stiffness matrix	Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Equation (12.5.8–1) and (12.5.8–2)	2 x 2

Load Type	Distribution
Element Temperature	Trilinear thru element
Nodal Temperature	Trilinear thru element
Pressure	Bilinear across each face

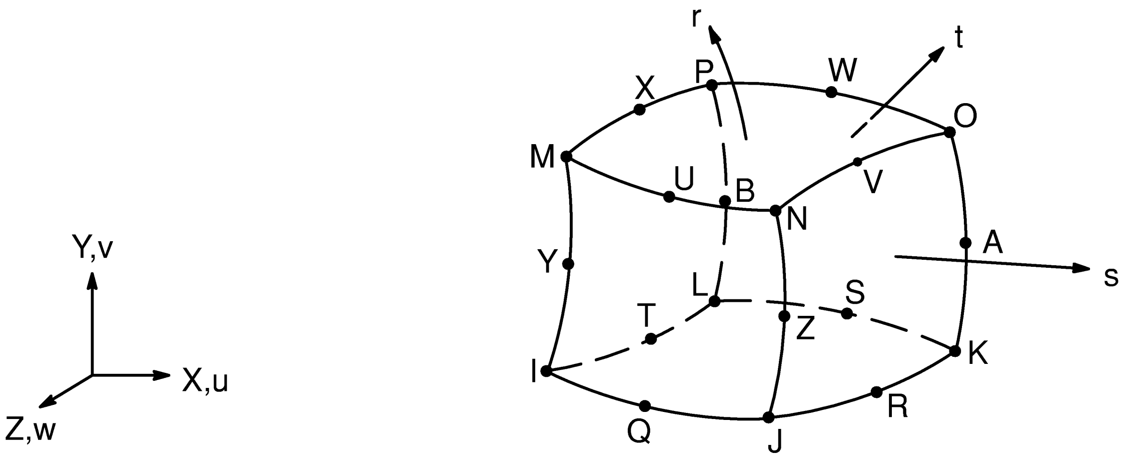
14.185.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.185.2 Theory

This element uses $\overline{B}$ method (selective reduced integration technique for volumetric terms) (Hughes(219), Nagtegaal et al(220)) if KEYOPT(2) = 0. If KEYOPT(2) = 1, the uniform reduced integration technique (Flanagan and Belytschko(232)) is used.

14.186 SOLID186 — 20–Node Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.20–1), (12.8.20–2), and (12.8.20–3)	2 x 2 x 2
Mass Matrix	Same as stiffness matrix.	3 x 3 x 3
Stress Stiffness Matrix	Same as stiffness matrix	Same as stiffness matrix
Thermal Load Vector	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Equations (12.5.10–1) and (12.5.10–2)	3 x 3

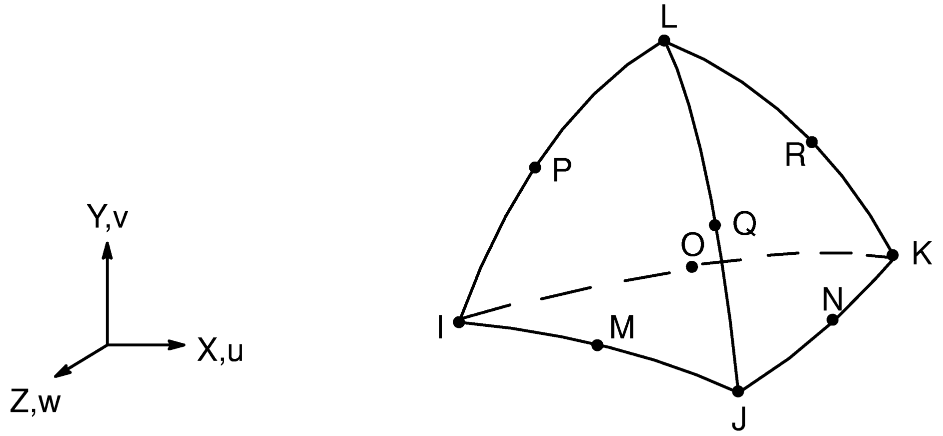
Load Type	Distribution
Element Temperature	Same as shape functions thru element
Nodal Temperature	Same as shape functions thru element
Pressure	Bilinear across each face

Reference: Zienkiewicz(39)

14.186.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.187 SOLID187 — 3–D 10–Node Tetrahedral Structural Solid



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Stress Stiffness Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Mass Matrix	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Thermal Load Vector	Equations (12.8.2–1), (12.8.2–2), and (12.8.2–3)	4
Pressure Load Vector	Equation (12.8.2–1), (12.8.2–2), and (12.8.2–3) specialized to the face	6

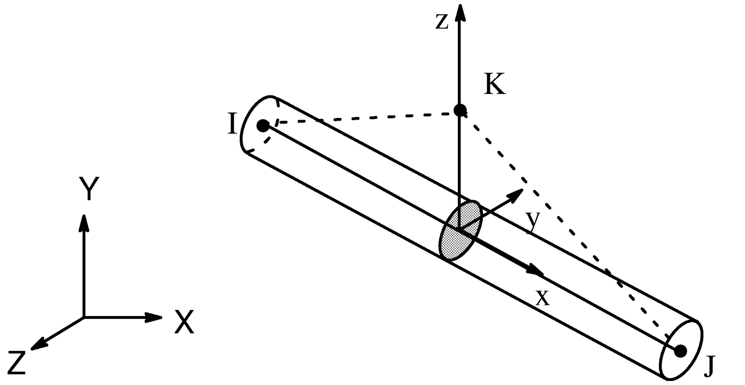
Load Type	Distribution
Element Temperature	Same as shape functions
Nodal Temperature	Same as shape functions
Pressure	Linear over each face

Reference: Zienkiewicz(39)

14.187.1 Other Applicable Sections

Chapter 2 describes the derivation of structural element matrices and load vectors as well as stress evaluations. Section 13.1 describes integration point locations.

14.188 BEAM188 — 3–D Finite Strain Linear Beam



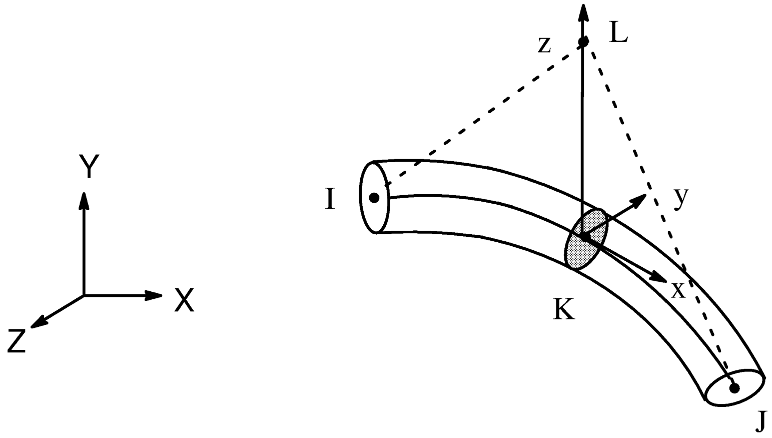
Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.2.1–1), (12.2.1–2), (12.2.1–3), (12.2.1–4), (12.2.1–5) and (12.2.1–6)	Along the length: 1 Across the section: see Section 14.189
Mass Matrix	Same as stiffness matrix	Along the length: 2 Across the section: 1
Stress Stiffness Matrix	Same as stiffness matrix	Same as stiffness matrix
Thermal Load Matrix	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix	Same as mass matrix
Newton–Raphson Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element Temperature	Bilinear across cross–section and linear along length (see Section 14.24.3 for details)
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Linear along length. The pressure is assumed to act along the element x–axis.

References: Simo and Vu–Quoc(237), Ibrahimbegovic(238).

The theory for this element is the same as BEAM189 described in Section 14.189, except that it is a linear, 2–noded beam element.

14.189 BEAM189 — 3–D Finite Strain Quadratic Beam



Matrix or Vector	Shape Functions	Integration Points
Stiffness Matrix	Equations (12.2.3–1), (12.2.3–2), (12.2.3–3), (12.2.3–4), (12.2.3–5) and (12.2.3–6)	Along the length: 2 Across the section: see text below
Mass Matrix	Same as stiffness matrix	Along the length: 3 Across the section: 1
Stress Stiffness Matrix	Same as stiffness matrix	Same as stiffness matrix
Thermal Load Matrix	Same as stiffness matrix	Same as stiffness matrix
Pressure Load Vector	Same as stiffness matrix	Same as mass matrix
Newton–Raphson Load Vector	Same as stiffness matrix	Same as stiffness matrix

Load Type	Distribution
Element Temperature	Bilinear across cross–section and linear along length (see Section 14.24.3 for details)
Nodal Temperature	Constant across cross–section, linear along length
Pressure	Linear along length. The pressure is assumed to act along the element x–axis.

References: Simo and Vu–Quoc(237), Ibrahimbegovic(238).

14.189.1 Assumptions and Restrictions

The elements are based on Timoshenko beam theory, and hence shear deformation effects are included. The element is a quadratic (3–noded) beam element in 3–D with six degrees of freedom at each node. The DOF at each node includes translations in x, y, and z directions, and rotations about the x, y, and z directions. Warping of cross sections is considered optionally (KEYOPT(1)).

This element is well–suited for linear, large rotation, and/or large strain nonlinear applications. If KEYOPT(2) = 0, the cross sectional dimensions are scaled uniformly as a function of axial strain in nonlinear analysis such that the volume of the element is preserved.

The element includes stress stiffness terms, by default, in any analysis with NLGEOM,ON. The stress stiffness terms provided enable the elements to analyze flexural, lateral and torsional stability problems (using eigenvalue buckling or collapse studies with arc length methods). Pressure load stiffness (Section 3.3.4) is included.

Transverse shear strain is constant through cross–section, i.e., cross sections remain plane and undistorted after deformation. The element can be used for slender or stout beams. Due to the limitations of first order shear deformation theory, only moderately “thick” beams may be analyzed. Slenderness ratio of a beam structure may be used in judging the applicability of the element. It is important to note that this ratio should be calculated using some global distance measures, and not based on individual element dimensions. A slenderness ratio greater than 30 is recommended.

Currently these elements support only elastic relation between transverse shear forces and transverse shear strains. Orthotropic elastic material properties with isotropic hardening plasticity options BISO, MISO may be used. User may specify transverse shear stiffnesses using real constants.

The St. Venant warping functions for torsional behavior is determined in the underformed state, and is used to define shear strain even after yielding. The element does not provide options to recalculate the torsional shear distribution on cross sections during the analysis and possible partial plastic yielding of cross section. As such, large inelastic deformation due to torsional loading should be treated with caution and carefully verified.

The elements are provided with section relevant quantities (area of integration, position, Poisson function, function derivatives, etc.) automatically at a number of section points by the use of section commands. Each section is assumed to be an assembly of predetermined number of 9 node cells (see Figure 14.189–1) which illustrates section model of a rectangular section). Each cell has 4 integration points.

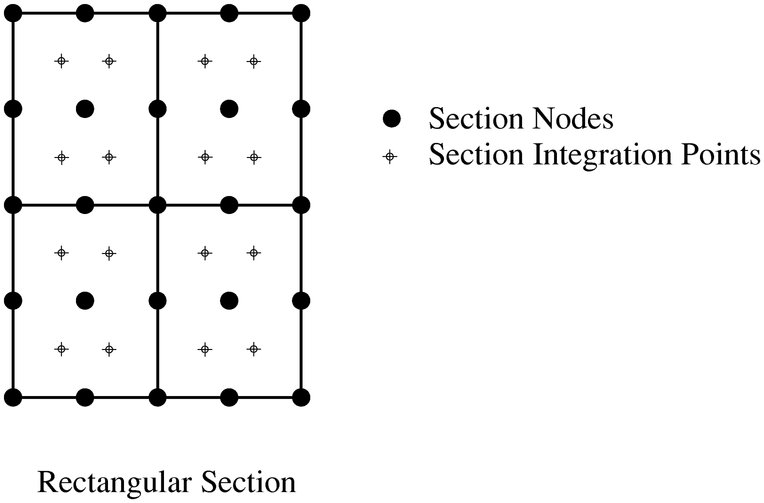


Figure 14.189–1 Section Model

When the material has inelastic behavior or the temperature varies across the section, constitutive calculations are performed at each of the section integration points. For all other cases, the element uses the pre–calculated properties of the section at each element integration point along the length. The restrained warping formulation used may be found in Timoshenko and Gere(246) and Schulz and Fillippou(247).

14.189.2 Stress Evaluation

Several stress evaluation options exist. The section strains and generalized stresses are evaluated at element integration points and then extrapolated to the nodes of the element.

If the material is elastic, stresses and strains are available after extrapolation in cross–section at the nodes of section mesh. If the material is plastic, stresses and strains are moved to the section nodes (from section integration points).

Chapter 15
Analysis Tools

ANSYS Theory Reference
Chapter 15 – Table of Contents

15.1 Acceleration Effect

Accelerations are applicable only to elements with displacement degrees of freedom (DOFs).

The acceleration vector $\{a_c\}$ which causes applied loads consists of a vector with a term for every degree of freedom in the model. In the below description, a typical node will be considered, having a specific location and accelerations associated with the three translations and three rotations:

$$\{a_c\} = \begin{Bmatrix} \{a_t\} \\ \{a_r\} \end{Bmatrix} \tag{15.1–1}$$

where: $\{a_t\} = \{a_t^d\} + \{a_t^r\} + \{a_t^I\}$ = translational acceleration vector

$\{a_r\} = \{a_r^r\} + \{a_r^I\}$ = rotational acceleration vector

$\{a_t^d\}$ = accelerations in global Cartesian coordinates (input on **ACEL** command) (15.1–2)

$\{a_t^r\}$ = translational acceleration vector due to rotations (discussed below)

$\{a_t^I\}$ = translational acceleration vector due to inertia relief (see Section 15.2)

$$\{a_r^r\} = \{\dot{\omega}\} + \{\dot{\Omega}\} \tag{15.1–3}$$

$\{a_r^r\}$ = angular acceleration vector due to input rotational accelerations

$\{\dot{\omega}\}$ = angular acceleration vector defined about the global Cartesian origin (input on **DOMEGA** command)

$\{\dot{\Omega}\}$ = angular acceleration vector of the overall structure about the point CG (input on **DCGOMG** command)

$\{a_r^I\}$ = rotational acceleration vector due to inertia relief (see section 15.2)

The translational acceleration vector due to rotations is:

$$\{a_t^r\} = \{\dot{\Omega}\} \times (\{R\} + \{r\}) + \{\Omega\} \times (\{\Omega\} \times (\{R\} + \{r\})) + 2\{\Omega\} \times (\{\omega\} \times \{r\}) + \{\dot{\omega}\} \times \{r\} + \{\omega\} \times (\{\omega\} \times \{r\}) \quad (15.1\text{–}4)$$

where: $\times$ = vector cross product

$\{R\}$ = vector from CG to the global Cartesian origin (computed from input on **CGLOC** command, with direction opposite as shown in Figure 15.1–1)

$\{r\}$ = position vector (see Figure 15.1–1)

$\{\omega\}$ = angular velocity vector defined about the global Cartesian origin (input on **OMEGA** command)

$\{\Omega\}$ = angular velocity vector defined about the point CG (input on the **CGOMGA** command)

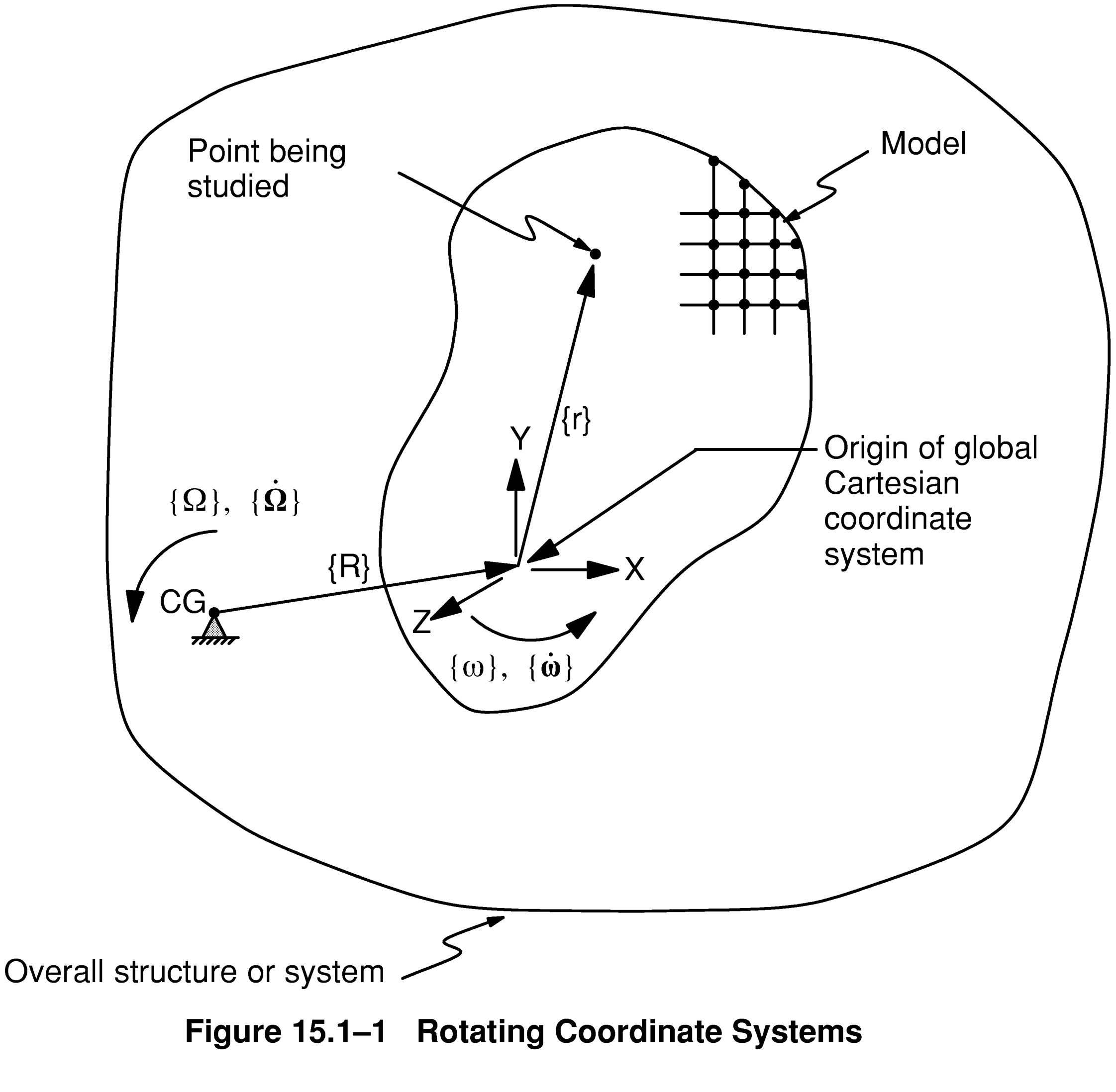

Point being studied
Model
{r}
Y
Origin of global Cartesian coordinate system
$\{\Omega\}$, $\{\dot{\Omega}\}$
{R}
CG
X
Z
$\{\omega\}$, $\{\dot{\omega}\}$
Overall structure or system

Figure 15.1–1 Rotating Coordinate Systems

15.2 Inertia Relief

Inertia relief is applicable only to linear analyses with elements containing only displacement degrees of freedom (DOFs).

An equivalent free–body analysis is performed if the **ANTYPE**,STATIC (static analysis) and if *KEY* = 1 on the **IRLF**,1 commands are used. This is a technique in which the applied forces and torques are balanced by inertial forces induced by an acceleration field. Consider the application of an acceleration field (to be determined) that precisely cancels or balances the applied loads:

$$\{F_t^a\} + \int_{vol} \{a_t^I\} \rho \, d(vol) = \{0\} \tag{15.2–1}$$

$$\{F_r^a\} + \int_{vol} \{r\} \times \left(\{a_r^I\} \times \{r\}\right) \rho \, d(vol) = \{0\} \tag{15.2–2}$$

where: $\{F_t^a\}$ = force components of the applied load vector

$\{a_t^I\}$ = translational acceleration vector due to inertia relief (to be determined)

ρ = density

vol = volume of model

$\{F_r^a\}$ = moment components of the applied load vector

$\{r\}$ = position vector = $\lfloor X \; Y \; Z \rfloor^T$

$\{a_r^I\}$ = rotational acceleration vector due to inertia relief (to be determined)

$\times$ = vector cross product

In the finite element implementation, the position vector $\{r\}$ and the moment in the applied load vector $\{F_r^a\}$ are taken with respect to the origin. Considering further specialization for finite elements, equations (15.2–1) and (15.2–2) are rewritten in equivalent form as:

$$\{F_t^a\} + [M_t] \{a_t^I\} = \{0\} \tag{15.2–3}$$

$$\{F_r^a\} + [M_r] \{a_r^I\} = \{0\} \tag{15.2–4}$$

where: $[M_t]$ = mass tensor for the entire finite element model (developed below)

$[M_r]$ = mass moments and mass products of the inertia tensor for the entire finite element model (developed below)

Once $[M_t]$ and $[M_r]$ are developed, then $\{a_t^I\}$ and $\{a_r^I\}$ in equations (15.2–3) and (15.2–4) can be solved. The corresponding output quantities “TRANSLATIONAL ACCELERATIONS” for $\{a_t^I\}$ and “ROTATIONAL ACCELERATIONS” for $\{a_r^I\}$ are under the label “INERTIA RELIEF SUMMARY”.

The computation for $[M_t]$ and $[M_r]$ proceeds on an element–by–element basis:

$$[M_t] = \sum [m_e] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \int_{vol} \rho \; d(vol) \qquad (15.2–5)$$

$$[M_r] = \sum [I_e] = \int_{vol} \begin{bmatrix} y^2+z^2 & -xy & -xz \\ -xy & x^2+z^2 & -yz \\ -xz & -yz & x^2+y^2 \end{bmatrix} \rho \; d(vol) \qquad (15.2–6)$$

in which $[m_e]$ and $[I_e]$ relate to individual elements, and the summations are for all elements in the model. The components in $[M_t]$ and $[M_r]$ are output under the heading “PRECISE MASS SUMMARY” with $[M_t]$ labeled as “TOTAL MASS” and $[M_r]$ as “MOMENTS AND PRODUCTS OF INERTIA TENSOR ABOUT ORIGIN”.

The evaluation for components of $[m_e]$ are simply obtained from a row–by–row summation applied to the elemental mass matrix over translational (x, y, z) degrees of freedom. It should be noted that $[m_e]$ is a diagonal matrix ($m_{xy} = 0$, $m_{xz} = 0$, etc.). The computation for $[I_e]$ is somewhat more involved, but can be summarized in the following form:

$$[I_e] = [b]^T [M_e] [b] \qquad (15.2–7)$$

where: $[M_e]$ = elemental mass matrix (which may be either lumped or consistent)

$[b]$ = matrix which consists of nodal positions and unity components

The forms of $[b]$ and, of course, $[M_e]$ are dependent on the type of element under consideration. The description of element mass matrices $[M_e]$ is given in Section 2.2. The derivation for $[b]$ comes about by comparing equations (15.2–2) and (15.2–4) on a per element basis, and eliminating $\{F_r^a\}$ to yield

$$[\mathrm{I_e}\} \;=\; \left\{a_r^I\right\} \;=\; \int_{vol} \{r\} \; x \; \left\{a_r^I\right\} \; x \; \{r\} \;\; \rho \;\; d(vol) \qquad (15.2–8)$$

where: vol = element volume

After a little manipulation, the acceleration field in (15.2–8) can be dropped, leaving the definition of $[\mathrm{I_e}]$ in (15.2–7).

It can be shown that if the mass matrix in (15.2–7) is derived in a consistent manner, then the components in $[\mathrm{I_e}]$ are quite precise. This is demonstrated as follows. Consider the inertia tensor in standard form:

$$[\mathrm{I_e}] \;=\; \int_{vol} \begin{bmatrix} y^2 + z^2 & -xy & -xz \\ -xy & x^2 + z^2 & -yz \\ -xz & -yz & x^2 + y^2 \end{bmatrix} \rho \;\; d(vol) \qquad (15.2–9)$$

which can be rewritten in product form:

$$[\mathrm{I_e}] \;=\; \int_{vol} [Q]^T[Q] \;\rho\;\; d(vol) \qquad (15.2–10)$$

The matrix [Q] is a skew–symmetric matrix.

$$[Q] \;=\; \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \qquad (15.2–11)$$

Next, shape functions are introduced by way of their basic form,

$$\{r\} \;=\; \lfloor X \; Y \; Z \rfloor^T \;=\; [N] \lfloor x_1 \; y_1 \; z_1 \; x_2 \; y_2 \; z_2 \; ... \; \rfloor^T \qquad (15.2–12)$$

where: [N] = usual matrix containing individual shape functions

Omitting the tedious algebra, equations (15.2–11) and (15.2–12) are combined to obtain

$$[Q] \;=\; [N] \;\; [b] \qquad (15.2–13)$$

where:

$$[b]^T \;=\; \begin{bmatrix} 0 & -z_2 & y_1 & 0 & -z_2 & y_2 ... \\ z_1 & 0 & -x_1 & z_2 & 0 & -x_2 ... \\ -y_1 & x_1 & 0 & -y_2 & x_2 & 0 ... \end{bmatrix} \qquad (15.2–14)$$

Inserting (15.2–14) into (15.2–10) leads to

$$[I_e] = [b]^T \int_{vol} [N]^T \rho \, [N] \, d(vol) \, [b] \qquad (15.2\text{–}15)$$

Noting that the integral in (15.2–15) is the consistent mass matrix for a solid element,

$$[M_e] = \int_{vol} [N]^T \rho \, [N] \, d(vol) \qquad (15.2\text{–}16)$$

So it follows that (15.2–7) is recovered from the combination of (15.2–15) and (15.2–16).

As stated above, the exact form of $[b]$ and $[M_e]$ used in equation (15.2–7) varies depending on the type of element under consideration. Equations (15.2–14) and (15.2–16) apply to all solid elements (in 2–D, z = 0). For discrete elements, such as beams and shells, certain adjustments are made to $[b]$ in order to account for moments of inertia corresponding to individual rotational degrees of freedom. For 3–D beams, for example, $[b]$ takes the form:

$$[b]^T = \begin{bmatrix} 0 & -z_2 & y_1 & 1 & 0 & 0 & 0 & -z_2 & y_2 & 1 & 0 & 0 \ldots \\ z_1 & 0 & -x_1 & 0 & 1 & 0 & z_2 & 0 & -x_2 & 0 & 1 & 0 \ldots \\ -y_1 & x_1 & 0 & 0 & 0 & 1 & -y_2 & x_2 & 0 & 0 & 0 & 1 \ldots \end{bmatrix} \qquad (15.2\text{–}17)$$

In any case, it is worth repeating that precise $[I_e]$ and $[M_r]$ matrices result when consistent mass matrices are used in equation (15.2–7).

If *KEY* = 1 on the **IRLF** command, then the m_x, m_y, and m_z diagonal components in $[M_t]$ as well as all tensor components in $[M_r]$ are calculated. Then the acceleration fields $\{a_t^I\}$ and $\{a_r^I\}$ are computed by the inversion of equations (15.2–3) and (15.2–4). The body forces that correspond to these accelerations are added to the user–imposed load vector, thereby making the net or resultant reaction forces null. If *KEY* = –1 on the **IRLF** command, then only a mass summary for $[M_t]$ and $[M_r]$ is made.

The calculations for $[M_t]$, $[M_r]$, $\{a_t^I\}$ and $\{a_r^I\}$ are made at every substep of every load step where they are requested, reflecting changes in material density and applied loads.

Several limitations apply, and it is useful to list them below.

1. Element mass and/or density must be defined in the model.
2. Substructures are not allowed.

3. In a model containing both 2–D and 3–D elements, only $M_t(1,1)$ and $M_t(2,2)$ in $[M_t]$ and $M_r(3,3)$ in $[M_r]$ are correct in the “PRECISE MASS SUMMARY”. All other terms in $[M_t]$ and $[M_r]$ should be ignored. The acceleration balance is, however, correct.
4. Axisymmetric elements are not allowed.
5. If grounded gap elements are in the model, their status should not change from their original status. Otherwise the exact kinematic constraints stated above might be violated.
6. The “CENTER OF MASS” does not include the effects of offsets or tapering on beam elements (BEAM23, BEAM24, BEAM44, and BEAM54) as well as the layered elements (SHELL91, SHELL99, and SOLID46). Breaking up each tapered element into several elements will give a more accurate solution.

15.3 Damping Matrices

The damping matrix ([C]) may be used in harmonic, damped modal and transient analyses as well as substructure generation. In its most general form, it is:

$$[C] = \alpha\,[M] + (\beta + \beta_c)\,[K] + \sum_{j=1}^{NMAT} \beta_j \left[K_j\right] + \sum_{k=1}^{NEL} [C_k] + \left[C_\xi\right] \qquad (15.3\text{–}1)$$

where: [C] = structure damping matrix

α = constant mass matrix multiplier (input on **ALPHAD** command)

[M] = structure mass matrix

β = constant stiffness matrix multiplier (input on **BETAD** command)

β_c = variable stiffness matrix multiplier (see equation (15.3–4))

[K] = structure stiffness matrix

NMAT = number of materials with DAMP input

β_j = constant stiffness matrix multiplier for material j (input as DAMP on **MP** command) (see note below)

$\left[K_j\right]$ = portion of structure stiffness matrix based on material j

NEL = number of elements with specified damping

$[C_k]$ = element damping matrix

$\left[C_\xi\right]$ = frequency–dependent damping matrix (see below)

Element damping matrices are available for:

BEAM4	3–D Elastic Beam	COMBIN37	Control
COMBIN7	Revolute Joint	FLUID38	Dynamic Fluid Coupling
LINK11	Linear Actuator	COMBIN40	Combination
COMBIN14	Spring–Damper	FLUID79	2–D Contained Fluid
PIPE16	Elastic Straight Pipe	FLUID80	3–D Contained Fluid
SURF19	2–D Surface Effect	FLUID81	Axisymmetric–Harmonic Contained Fluid
SURF22	3–D Surface Effect		
MATRIX27	Stiffness, Damping, or Mass Matrix		

Note that [K], the structure stiffness matrix, may include plasticity and/or large deflection effects (i.e., may be the tangent matrix).

The frequency–dependent damping matrix $[C_\xi]$ is specified indirectly by defining a damping ratio, ξ^d. This effect is available only in the Spectrum (**ANTYPE**,SPECTR), the Harmonic Response with mode superposition (**ANTYPE**,HARM with **HROPT**,MSUP) Analyses, as well as the Transient Analysis with mode superposition (**ANTYPE**,TRANS with **TRNOPT**,MSUP).

$[C_\xi]$ may be calculated from the specified ξ^d as follows:

$$\{u_i\}^T\,[C_\xi]\,\{u_i\} = 4\pi f_i \xi_i^d \tag{15.3–2}$$

where:
ξ_i^d = damping ratio for mode shape i (defined below)
$\{u_i\}$ = shape of mode i
f_i = frequency associated with mode shape i

The damping ratio ξ_i^d is the combination of:

$$\xi_i^d = \xi + \xi_{mi} \tag{15.3–3}$$

where:
ξ = constant damping ratio (input on **DMPRAT** command)
ξ_{mi} = modal damping ratio for mode shape i (input on **MDAMP** command)

Material–dependent damping may be used for ξ_{mi} if it is defined in the modal analysis, see equation (17.7–1).

Actually ξ_i^d is used directly. $[C_\xi]$ is never explicitly computed.

β_c, available for the Harmonic Response Analyses (**ANTYPE**,HARM with **HROPT**,FULL or **HROPT**,REDUC), is used to give a constant damping ratio, regardless of frequency. The damping ratio is the ratio between actual damping and critical damping. The stiffness matrix multiplier is related to the damping ratio by:

$$\beta_c = \frac{\xi}{\pi f} \tag{15.3–4}$$

where:
ξ = constant damping ratio (input on **DMPRAT** command)
f = frequency in the range between f_B and f_E
f_B = beginning frequency (input as FREQB, **HARFRQ** command)
f_E = end frequency (input as FREQE, **HARFRQ** command)

15.4 Element Reordering

The ANSYS program provides a capability for reordering the elements to reduce the wavefront. Since the wavefront solver processes the elements sequentially, the order of the elements greatly affects the size of the wavefront. To minimize the wavefront is to minimize the number of DOFs that are active at the same time.

Each element has a location, or order, number which represents its sequence in the solution process. Initially, this order number is equal to the identification number of the element. Reordering changes the order number for each element. (The element identification numbers are not changed during reordering and are used in preprocessing and postprocessing.) The new order is used only during the solution phase and is transparent to the user (but can be displayed with the /**PNUM** command). Reordering can be accomplished in one of three ways:

15.4.1 Reordering Based on Topology with a Program Defined Starting Surface

This sorting algorithm is used by default, requiring no explicit action by the user. It may also be accessed with the **WAVES** command, but without a **WSTART** command.. The starting surface is defined by the program using a graph theory algorithm (Hoit and Wilson(99), Cuthill and McKee(100), Georges and McIntyre(101)). The automatic algorithm defines a set of accumulated nodal and element weights as suggested by Hoit and Wilson(99). These accumulated nodal and element weights are then used to develop the element ordering scheme.

15.4.2 Reordering Based on Topology with a User Defined Starting Surface

This sorting algorithm uses the **WAVES** command and is based on a starting surface input on the **WSTART** command, and then possibly guided by other surfaces. These surfaces, as required by the algorithm, consist of lists of nodes (wavelists) which are used to start and stop the ordering process. The steps taken by the program are:

1. Define each coupled node set and constraint equation as an element.
2. Bring in wavelist (defined on **WSTART** command).
3. Define candidate elements (elements having nodes in present wavelist, but not in any other wavelist).

4. If no candidate elements were found, go to step 2 and start again for next wavelist. If no more wavelists, then stop.
5. Find the best candidate based on:
   a. element that brings in the least number of new nodes (nodes not in present wavelist) – Subset A of candidate elements.
   b. if Subset A has more than one element, then element from Subset A on the surface of the model – Subset B of candidate elements.
   c. if Subset B has more than one element, then element from Subset B with the lowest element number.
6. Remove processed nodes from wavelist and include new nodes from best candidate.
7. If best candidate element is not a coupled node set or constraint equation, then save element.
8. Repeat steps 3 to 7 until all elements have been processed.

Restrictions on the use of reordering based on topology are:

- Master DOFs and imposed displacement conditions are not considered.
- Any discontinuous models must have at least one node from each part included in a list.

15.4.3 Reordering Based on Geometry

This sorting algorithm is accessed with the **WSORT** command. It is performed by a sweep through the element centroids along one of the three global or local axes, either in the positive or negative direction.

15.4.4 Automatic Reordering

If no reordering was explicitly requested and the **NOORDER** command has not been issued, models are automatically reordered before solution. Both methods outlined in Section 15.4.1 and 15.4.3 (in three positive directions) are used and the ordering resulting in the smallest wavefront is used.

15.5 Automatic Master DOF Selection

The program permits the user to select the master degrees of freedom (MDOF) (input on **M** command), the program to select them (input on **TOTAL** command), or any combination of these two options. Any user selected MDOF are always retained DOFs during the Guyan reduction. Consider the case where the program selects all of the MDOF. (This method is described by Henshell and Ong(9)). Define:

N_S = Number of MDOFs to be selected

N_A = Number of total active DOFs in the structure

The procedure then goes through the following steps:

1. The first N_S completed DOFs that are encountered by the wavefront solver are initially presumed to be MDOF. (An option is available to exclude the rotational DOFs (*NRMDF* = 1, **TOTAL** Command).
2. The next DOF is brought into the solver. All of the N_S + 1 DOFs then have the quantity (Q_i) computed:

$$Q_i = \frac{K_{ii}}{M_{ii}} \tag{15.5–1}$$

where:

K_{ii} = ith main diagonal term of the current stiffness matrix

M_{ii} = ith main diagonal term of the current mass matrix (or stress stiffness matrix for buckling)

If K_{ii} or M_{ii} is zero or negative, row i is eliminated. This removes tension DOFs in buckling.

3. The largest of the Q_i terms is identified and then eliminated.
4. All remaining DOFs are thus processed in the same manner. Therefore, N_A – N_S DOFs are eliminated.

It may be seen that there sometimes is a path dependency on the resulting selection of MDOF. Specifically, one selection would result if the elements are read in from left to right, and a different one might result if the elements are read in from right to left. However, this difference usually yields insignificant differences in the results.

The use of this algorithm presumes a reasonably regular structure. If the structure has an irregular mass distribution, the automatically selected MDOF may be concentrated totally in the high mass regions, in which case the manual selection of some MDOF should be used.

15.6 Automatic Time Stepping

The method of automatic time stepping (or automatic loading) is one in which the time step size and/or the applied loads are automatically determined in response to the current state of the analysis under consideration. This method may be applied to structural, thermal, electric, and magnetic analyses that are performed in the time domain using the **TIME** command, and includes static (or steady state) and dynamic (or transient) situations. In general, it works with **ANTYPE**,STATIC or **ANTYPE**,TRANS. The key automatic load command, however, is **AUTOTS** which turns this feature "on" or "off".

An important point to be made here is that automatic loading always works through the adjustment of the time step size; and that the loads that are applied are automatically adjusted if ramped boundary conditions are activated (**KBC**,0). In other words the time step size is always subjected to possible adjustment when automatic loading is engaged. Applied loads and boundary conditions, however, will vary according to how they are applied and how the **KBC** command is used. That is why this method may also be thought of as automatic loading.

There are two important features of the automatic time stepping algorithm. The first feature concerns the ability to estimate the next time step size, based on current and past analysis conditions, and make proper load adjustments. In other words, given conditions at the current time, t_n, and the previous time increment, Δt_n, the primary aim is to determine the next time increment, Δt_{n+1}. Since the determination of Δt_{n+1} is largely predictive, this part of the automatic time stepping algorithm is referred to as the *time step prediction* component.

The second feature of automatic time stepping is referred to as the *time step bisection* component. Its purpose is to decide whether or not to reduce the present time step size, Δt_n, and redo the substep with a smaller step size. For example, working from the last converged solution at time point t_{n-1}, the present solution begins with a predicted time step, Δt_n. Equilibrium iterations are performed; and if proper convergence is either not achieved or not anticipated, this time step is reduce to $\Delta t_n/2$ (i.e., it is bisected), and the analysis begins again from time t_{n-1}. Multiple bisections can occur per substep for various reasons (discussed later).

15.6.1 Time Step Prediction

At a given converged solution at time, t_n, and with the previous time increment, Δt_n, the goal is to predict the appropriate time step size to use as the next substep. This step size is derived from the results of several unrelated computations and is most easily expressed as the minimization statement:

$$\Delta t_{n+1} = \min\ (\Delta t_{eq},\ \Delta t_1,\ \Delta t_2,\ \Delta t_g,\ \Delta t_c,\ \Delta t_p) \qquad (15.6–1)$$

where: Δt_{eq} = time increment which is limited by the number of equilibrium iterations needed for convergence at the last converged time point. The more iterations required for convergence, the smaller the predicted time step. This is a general measure of all active nonlinearities. Increasing the value of *NEQIT* on the **NEQIT** command will tend to promote larger time step sizes.

Δt_1 = time increment which is limited by the response eigenvalue computation for 1st order systems (e.g., thermal transients). Limiting values are found on the **TINTP** command (described below).

Δt_2 = time increment which is limited by the response frequency computation for 2nd order systems (e.g., structural dynamics). The aim is to maintain 20 points per cycle (described below).

Δt_g = time increment that represents the time point at which a gap or a nonlinear (multi–status) element will change abruptly from one condition to another (status change). KEYOPT(7) allows further control for the CONTAC elements.

Δt_c = time increment based on the the allowable creep strain increment (described below).

Δt_p = time increment based on the allowable plastic strain increment. The limit is set at 5% per time step (described below).

Several trial step sizes are calculated, and the minimum one is selected for the next time step. This predicted value is further restricted to a range of values expressed by

$$\Delta t_{n+1} \leq \min\ (F\Delta t_n,\ \Delta t_{max}) \qquad (15.6–2)$$

and

$$\Delta t_{n+1} \geq \max\ (\Delta t_n/F,\ \Delta t_{min}) \qquad (15.6–3)$$

where: F = increase/decrease factor. F = 2, if static analysis; F = 3, if dynamic (see the **ANTYPE** and **TIMINT** commands)

Δt_{max} = *DTMAX* from the **DELTIM** command (or the equivalent quantity calculated from the **NSUBST** command)

Δt_{min} = *DTMIN* from the **DELTIM** command (or the equivalent quantity calculated from the **NSUBST** command)

In other words, the current time step is increased or decreased by at most a factor of 2 (or 3 if dynamic), and it may not be less than Δt_{min} or greater than Δt_{max}.

15.6.2 Time Step Bisection

When bisection occurs, the current substep solution (Δt_n) is removed, and the time step size is reduced by 50%. If applied loads are ramped (**KBC**,0), then the current load increment is also reduced by the same amount. One or more bisections can take place for several reasons, namely:

1. The number of equilibrium iterations used for this substep exceeds the number allowed (**NEQIT** command).
2. It appears likely that all equilibrium iterations will be used.
3. A *negative main diagonal* was encountered in the solution, suggesting instability.
4. The largest calculated displacement DOF exceeds the limit (*DLIM* on the **NCNV** command).
5. An illegal element distortion is detected (e.g., negative radius in an axisymmetric analysis)

More than one bisection may be performed per substep. However, bisection of the time–step size is limited by the minimum size that is defined by *DTMIN* input on the **DELTIM** command (or the equivalent **NSUBST** input).

15.6.3 The Response Eigenvalue for 1st Order Transients

The response eigenvalue is used in the computation of Δt_1 and is defined as:

$$\lambda_r = \frac{\{\Delta u\}^T [K^T] \{\Delta u\}}{\{\Delta u\}^T [C] \{\Delta u\}} \qquad (15.6\text{–}4)$$

where: λ_r = response eigenvalue (item RESEIG for POST26 **SOLU** command and ***GET** command)

$\{\Delta u\}$ = substep solution vector (t_{n-1} to t_n)

$[K^T]$ = the Dirichlet matrix. In a heat transfer or an electrical conduction analysis this matrix is referred to as the conductivity matrix; in magnetics this is called the magnetic “stiffness”. The superscript T denotes the use of a tangent matrix in nonlinear situations

$[C]$ = the damping matrix. In heat transfer this is called the specific heat matrix.

The product of the response eigenvalue and the previous time step (Δt_n) has been employed by Hughes(145) for the evaluation of 1st order explicit/implicit systems. In Hughes(145) the quantity $\Delta t_n \lambda$ is referred to as the “oscillation limit”, where λ is the maximum eigenvalue. For unconditionally stable systems, the primary restriction on time–step size is that the inequality $\Delta t_n \lambda \gg 1$ should be avoided. Hence it is very conservative to propose that $\Delta t_n \lambda \simeq 1$.

Since the time integration used employs the trapezoidal rule (equation (17.2–22)), all analyses of 1st order systems are unconditionally stable. The response eigenvalue supplied by means of equation (15.6–4) represents the dominate eigenvalue and not the maximum; and the time–step restriction above is restated as:

$$\Delta t_n \lambda_r \cong f \qquad (f < 1) \tag{15.6–5}$$

This equation expresses the primary aim of automatic time stepping for 1st order transient analyses. The quantity $\Delta t_n \lambda_r$ appears as the oscillation limit output during automatic loading. The default is f = 1/2, and can be changed by means of the *OSLIM* and *TOL* input quantities on the **TINTP** command. The quantity Δt_1 is approximated as:

$$\frac{\Delta t_1}{\Delta t_n} = \frac{f}{\lambda_r \Delta t_n} \tag{15.6–6}$$

15.6.4 The Response Frequency for Structural Dynamics

The response frequency is used in the computation of Δt_2 and is defined as (Bergan(105)):

$$f_r{}^2 = \frac{\{\Delta u\}^T[K^T]\{\Delta u\}}{(2\pi)^2\{\Delta u\}^T[M]\{\Delta u\}} \tag{15.6–7}$$

where:

f_r = response frequency (item RESFRQ for POST26 **SOLU** command and ***GET** command)

$\{\Delta u\}$ = substep solution vector (t_{n-1} to t_n)

$[K^T]$ = tangent stiffness matrix

$[M]$ = mass matrix

This equation is a nonlinear form of Rayleigh's quotient. The related response period is:

$$T_r = 1/f_r \tag{15.6–8}$$

Using T_r, the time increment limited by the response frequency is:

$$\Delta t_2 = T_r/20 \tag{15.6–9}$$

15.6.5 Creep Time Increment

The time step size may be increased or decreased by comparing the value of the creep ratio C_{max} (Section 4.3) to the creep criterion C_{cr}. C_{cr} is equal to .10 unless it is redefined by the **CRPLIM** command. The time step estimate is computed as:

$$\Delta t_c = \Delta t_n \frac{C_{cr}}{C_{max}} \qquad (15.6\text{–}10)$$

Δt_c is used in equation (15.6–1) only if it differs from Δt_n by more than 10%.

15.6.6 Plasticity Time Increment

The time step size is increased or decreased by comparing the value of the effective plastic strain increment $\Delta\hat{\epsilon}_n^{pl}$ (equation (4.1–23)) to 0.05 (5%). The time step estimate is computed as:

$$\Delta t_p = \Delta t_n \frac{.05}{\Delta\hat{\epsilon}_n^{pl}} \qquad (15.6\text{–}11)$$

Δt_p is used in equation (15.6–1) only if it differs from Δt_n by more than 10%.

15.7 Wavefront Solver

The ANSYS program uses the wavefront or frontal solution procedure discussed by Irons(17) and Melosh and Bamford(25) for the system of the simultaneous linear equations developed by the finite element method. The number of equations which are active after any element has been processed during the solution procedure is called the wavefront at that point. The method used places a wavefront restriction on the problem definition, which depends upon the amount of memory available for a given problem. Several thousand DOFs on the wavefront can be handled in memory on some currently available computers. Wavefront limits tend to be restrictive only for the analysis of arbitrary 3–D solids.

In the wavefront procedure, the sequence in which the elements are processed in the solver (the element “order”) is crucial to minimize the size of the wavefront. Alternatively, for a banded solver, the band width is minimized by paying close attention to the ordering of the nodes. In both cases, this minimization is important for reasons of efficiency and problem size.

The computer time required for the solution procedure is proportional to the square of the mean wavefront size. Therefore, it is advantageous to be able to estimate and minimize the wavefront size. The wavefront size is determined by the sequence in which the elements are arranged. The node numbers of all elements are scanned to determine which element is the last to use each node. As the total system of equations is assembled from the element matrices, the equations for a node which occurs for the last time are algebraically solved in terms of the remaining unknowns and eliminated from the assembled matrix by Gauss elimination. The active equations are represented by:

$$\sum_{j=1}^{L} K_{kj}\, u_j = F_k \tag{15.7–1}$$

where:
K_{kj} = stiffness term relating the force at DOF k to the displacement at DOF j
u_j = nodal displacement of DOF j
F_k = nodal force of DOF k
k = equation (row) number
j = column number
L = number of equations

To eliminate a typical equation i = k, the equation is first normalized to

$$\sum_{j=1}^{L} \frac{K_{ij}}{K_{ii}} u_j = \frac{F_i}{K_{ii}} \tag{15.7–2}$$

This is rewritten as:

$$\sum_{j=1}^{L} K_{ij}^{*} u_j = F_i^{*} \tag{15.7–3}$$

where:

$$K_{ij}^{*} = \frac{K_{ij}}{K_{ii}}$$

$$F_i^{*} = \frac{F_i}{K_{ii}}$$

Here K_{ii} is known as the “pivot.” If the absolute value of K_{ii} is sufficiently small, it is numerically indistinguishable from zero. This usually means the structure is insufficiently constrained (or needs more master DOFs for reduced analyses).

Pivots are categorized as shown in Figure 15.7–1:

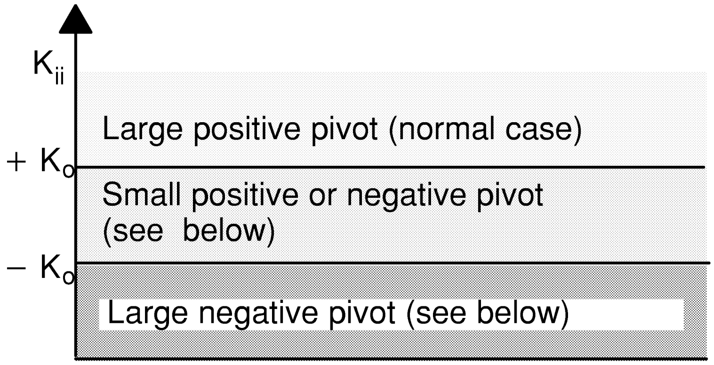

K_{ii}
Large positive pivot (normal case)
$+ K_o$
Small positive or negative pivot (see below)
$- K_o$
Large negative pivot (see below)

Figure 15.7–1 Ranges of Pivot Values

In Figure 15.7–1,

$$K_o = \left\{ \begin{array}{c} 10^{-13}\ (K_{max}) \\ \text{or} \\ 7.888 \text{ x } 10^{-31} \end{array} \right\} \text{whichever is greater}$$

K_{max} = max (K_{ii}) encountered up to this point in the wavefront

The number of small positive and negative pivots (N) are reported with the message:

> There are (N) small equation solver pivot terms. This may occur during a Newton–Raphson iteration procedure, is so noted, and usually can be ignored. Otherwise, it usually represents an unconstrained structure or a reduced analysis with insufficient master DOFs and generates an error message.

Large negative pivots will cause the error message:

> Large negative pivot value (value) at node (node, dof).

Variations of this message exist for problems with buckling, stress stiffening or spin softening. Large or small negative pivots for piezoelectric, acoustic, coupled fluid–thermal, circuits, interface elements, as well as for arc length usages are not counted or flagged as they are commonly expected.

Equation (15.7–3) is written to a file for later back–substitution. The remaining equations are modified as:

$$K^*_{kj} = K_{kj} - K_{ki} K^*_{ij} \tag{15.7–4}$$

$$F_k = F_k - K_{ki} F^*_i \tag{15.7–5}$$

where: $k \neq i$

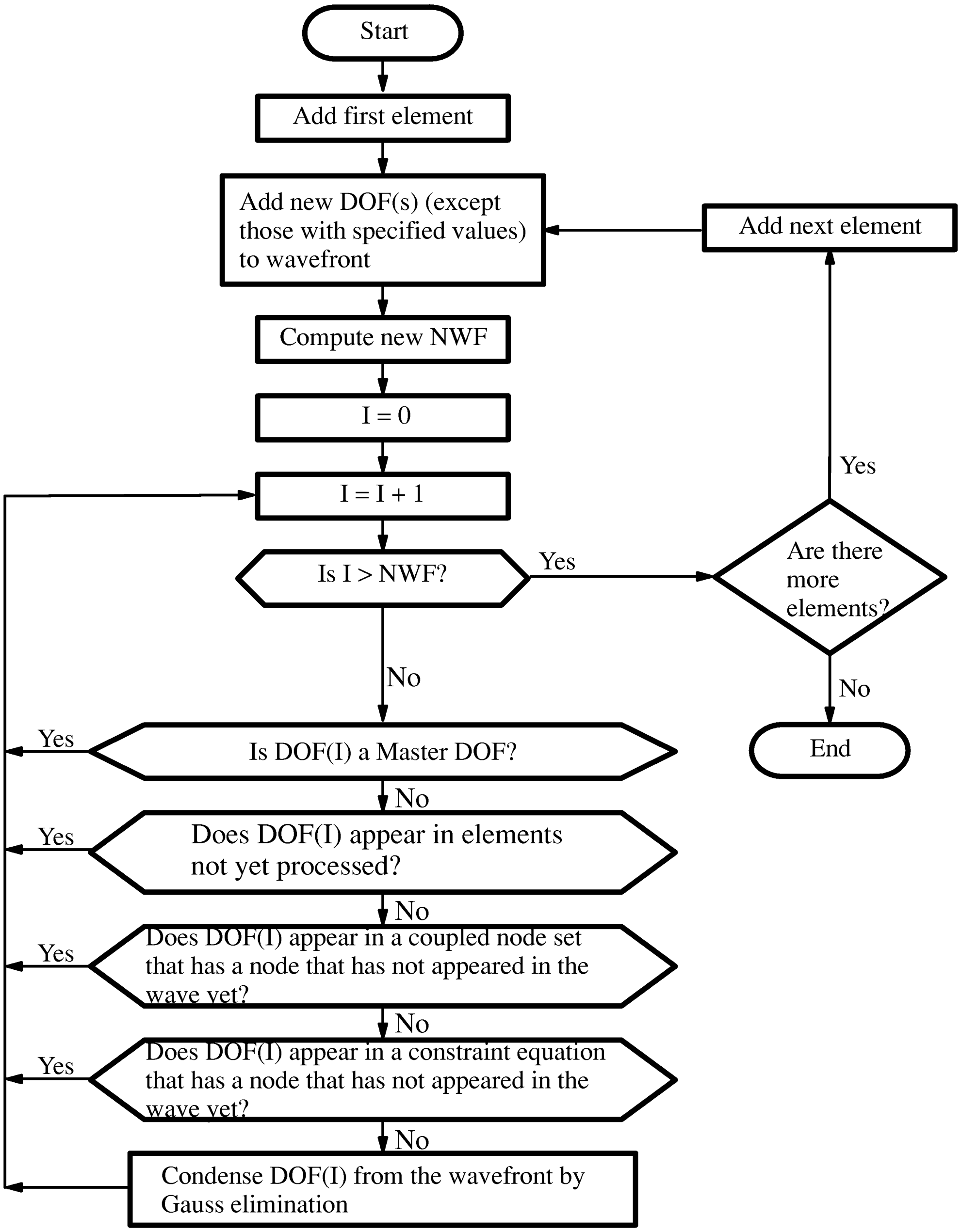
so that

$$\sum_{j=1}^{L-1} K^*_{kj} u_j = F^*_k \tag{15.7–6}$$

where k varies from 1 to L–1. Having eliminated row i from equation (15.7–6), the process is repeated for all other rows eligible to be eliminated.

The equations for a node which occurs for the first time are added to the assembled matrix as the solution progresses. Thus, the assembled matrix expands and contracts as node make their first and last appearance in the element definitions. The varying size of the active matrix is the instantaneous wavefront size.

When several elements are connected to the same node point, the DOFs associated with these elements remain active in memory until the wavefront “passes” all elements connected to the node. DOFs related by constraint equations or coupled nodes remain active until the wavefront “passes” all elements connected to the related DOFs. Master DOFs remain active in memory and are not deleted from the wavefront. This procedure is shown by the flow chart shown in Figure 15.7–2.


Start
Add first element
Add new DOF(s) (except those with specified values) to wavefront
Add next element
Compute new NWF
I = 0
I = I + 1
Yes
Is I > NWF?
Yes
Are there more elements?
No
End
No
Yes
Is DOF(I) a Master DOF?
No
Yes
Does DOF(I) appear in elements not yet processed?
No
Yes
Does DOF(I) appear in a coupled node set that has a node that has not appeared in the wave yet?
No
Yes
Does DOF(I) appear in a constraint equation that has a node that has not appeared in the wave yet?
No
Condense DOF(I) from the wavefront by Gauss elimination

DOF = Degree(s) of freedom
NWF = Number of DOF in wavefront after matrices have been assembled
I = Index for DOF, where $0 < I \leq NWF$

Figure 15.7–2 Wavefront Flow Chart

To reduce the maximum wavefront size, the elements must be ordered so that the element for which each node is first mentioned is as close as possible in sequence to the element for which it is mentioned last. In geometric terms, the elements should be ordered so that the wavefront sweeps through the model continuously from one end to the other in the direction which has the largest number of nodes. For example, consider a rectangular model having six nodes in one direction and 20 nodes in the other direction, as shown in Figure 15.7–3.

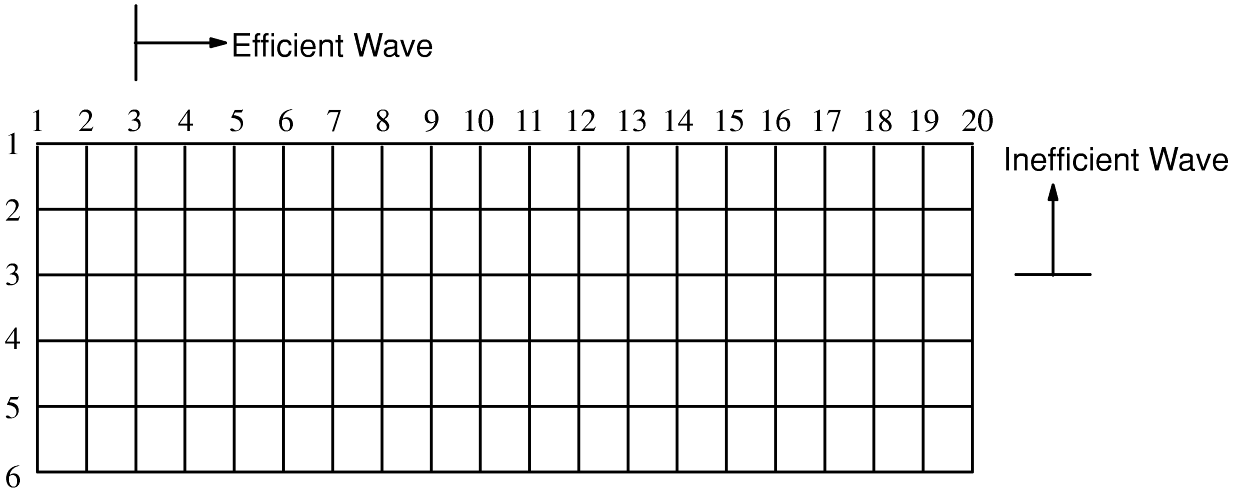


Figure 15.7–3 Sample Mesh

The elements should be ordered along one 6–node edge and progress toward the other 6–node edge. In this way, equations will be deleted from the assembled matrix as soon as possible after they are added, thus minimizing the wavefront size. All elements, including those of different types, should be included in the “one sweep” definition. See Section 15.4 for element reordering to reduce wavefront size.

See Section 13.3 for when matrices can be reused.

15.8 Constraint Equations

15.8.1 Derivation of Matrix and Load Vector Operations

Given the set of L linear simultaneous equations in unknowns u_j (same as equation (15.7–1)):

$$\sum_{j=1}^{L} K_{kj}\, u_j = F_k \qquad (1 \le k \le L) \tag{15.8–1}$$

subject to the linear constraint equation (input on **CE** command)

$$\sum_{j=1}^{L} C_j\, u_j = C_o \tag{15.8–2}$$

normalize equation (15.8–2) with respect to the prime DOF u_i by dividing by C_i to get:

$$\sum_{j=1}^{L} C_j^*\, u_j = C_o^* \tag{15.8–3}$$

where: $C_j^* = C_j/C_i$

$C_o^* = C_o/C_i$

which is written to a file for back–substitution instead of equation (15.7–3). Equation (15.8–3) is expanded (recall $C_i^* = 1$) as:

$$u_i + \sum_{j=1}^{L} C_j^*\, u_j = C_o^* \qquad (j \ne i) \tag{15.8–4}$$

Equation (15.8–1) may be similarly expanded as:

$$K_{ki}\, u_i + \sum_{j=1}^{L} K_{kj}\, u_j = F_k \qquad (j \ne i) \tag{15.8–5}$$

Multiply equation (15.8–4) by K_{ki} and subtract from equation (15.8–5) to get:

$$\sum_{j=1}^{L} \left(K_{kj} - C_j^* K_{ki}\right) u_j = F_k - C_o^* K_{ki} \quad (j \neq i) \tag{15.8–6}$$

Specializing equation (15.8–6) for k=i allows it to be written as:

$$\sum_{j=1}^{L} \left(K_{ij} - C_j^* K_{ii}\right) u_j = F_i - C_o^* K_{ii} \qquad (j \neq i) \tag{15.8–7}$$

This may be considered to be a revised form of the constraint equation. Introducing a Lagrange multiplier λ_k, equations (15.8–6) and (15.8–7) may be combined as:

$$\begin{aligned} &\sum_{j=1}^{L} \left(K_{kj} - C_j^* K_{ki}\right) u_j - F_k + C_o^* K_{ki} \\ &+ \lambda_k \left[\sum_{j=1}^{L} \left(K_{ij} - C_j^* K_{ii}\right) u_j - F_i + C_o^* K_{ii}\right] = 0 \quad (j \neq i) \end{aligned} \tag{15.8–8}$$

By the standard Lagrange multiplier procedure (see Denn(8)):

$$\lambda_k = \frac{\partial u_i}{\partial u_k} \tag{15.8–9}$$

Solving equation (15.8–4) for u_i,

$$u_i = C_o^* - \sum_{j=1}^{L} C_j^* u_j \quad (j \neq i) \tag{15.8–10}$$

so that

$$\lambda_k = -C_k^* \tag{15.8–11}$$

Substituting equation (15.8–11) into (15.8–8) and rearranging terms,

$$\begin{aligned} &\sum_{j=1}^{L} \left(K_{kj} - C_j^* K_{ki} - C_k^* K_{ij} + C_k^* C_j^* K_{ii}\right) u_j \\ &= F_k - C_o^* K_{ki} - C_k^* F_i + C_k^* C_o^* K_{ii} \quad (j \neq i) \end{aligned} \tag{15.8–12}$$

or

$$\sum_{j=1}^{L-1} K^*_{kj}\, u_j = F^*_k \quad (1 \le k \le L-1) \tag{15.8–13}$$

where: K^*_{kj} = $K_{kj} - C^*_j K_{ki} - C^*_k K_{ij} + C^*_k C^*_j K_{ii}$ (replaces (15.7–4) in Gauss elimination)

F^*_k = $F_k - C^*_o K_{ki} - C^*_k F_i + C^*_k C^*_o K_{ii}$ (replaces (15.7–5) in Gauss elimination)

15.8.2 Constraint Equation Reaction Forces

The net force at each DOF of the constraint equations is printed in the Constraint Equation Reaction Forces Table which reports the net forces at the DOFs with specified displacements. Constraint equation reactions are the forces required to hold the structure in the relative position required by the constraint equations. The computation is analogous to equation (15.12–7):

$$\{F^r_h\} = [K_{fh}]^T \{u_f\} + [K_{hh}]\{u_h\} - \{F^a_h\} \tag{15.8–14}$$

where: h = DOFs in constraint equations

f = DOFs not in constraint equations

15.8.3 Interface Constraint Equations

Parts of models whose nodes do not match at their interface may be tied together by constraint equations generated by the **CEINTF** command. Figure 15.8–1 illustrates such a case.

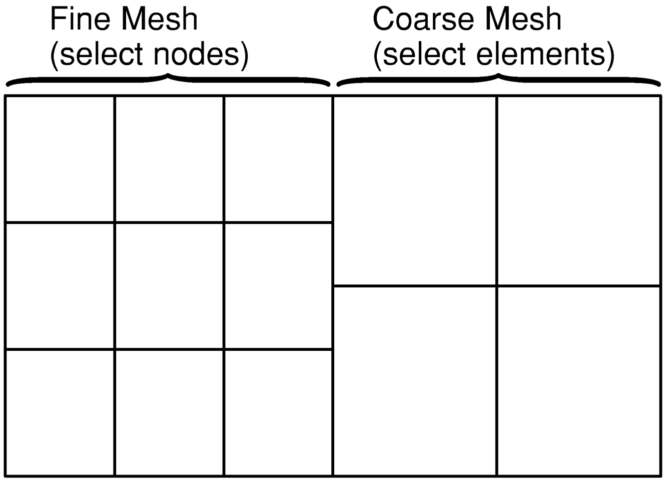


Figure 15.8–1 Interface Constraint Equation Situation

The procedure requires the selection of nodes on one side of the interface (the finer mesh is recommended) and the selection of elements on the other side of the interface (the coarser mesh is recommended). The reason for selecting nodes on the finer mesh side of the interface is that more constraint equations will be generated, ensuring a better connection between the sections.

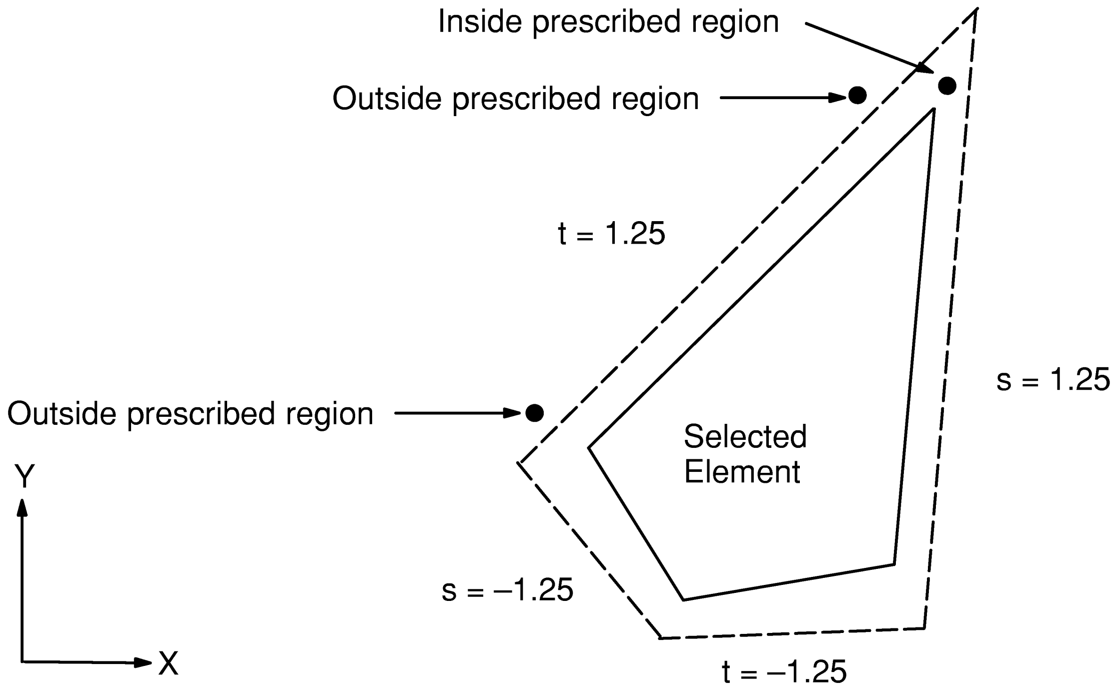


Figure 15.8–2 Tolerances for Nearness of Selected Nodes to Selected Elements

The first step is for each node on the interface to select an element on the opposite side. The selected element is the first element for which the node falls within the element, or, if the node does not fall within any element, it is the first element for which the node falls within a prescribed region around the element.

The element is expanded in each of its natural coordinate directions from –1.0 to 1.0 to –1.0–A to 1.0+A, where A = TOLER input on **CEINTF** command (default = 0.25). This prescribed region is shown in Figure 15.8–2 for a quadtilateral element of general shape using the default value of A. For triangular elements, the prescribed region is described by: each of the 3 triangular natural coordinates > –A, and the sum of the 3 triangular natural coordinates < 1+A.

A similar tolerance computation applies to the MOVETOL argument of **CEINTF**. If a node is within t_x or t_y of a surface in the x or y direction respectively, it is moved to that surface. If a node is within both the t_x and the t_y MOVETOL distances of the element corner node, it is moved to the element corner node location.

These same concepts are generalized to three dimensions for 3–D models.

Once the element is selected, the shape functions of the element are used to prepare constraint equations such that the node follows the motion of the selected element. This generally violates compatibility as shown in Figure 15.8–3, but the violation becomes less severe as the mesh is refined.

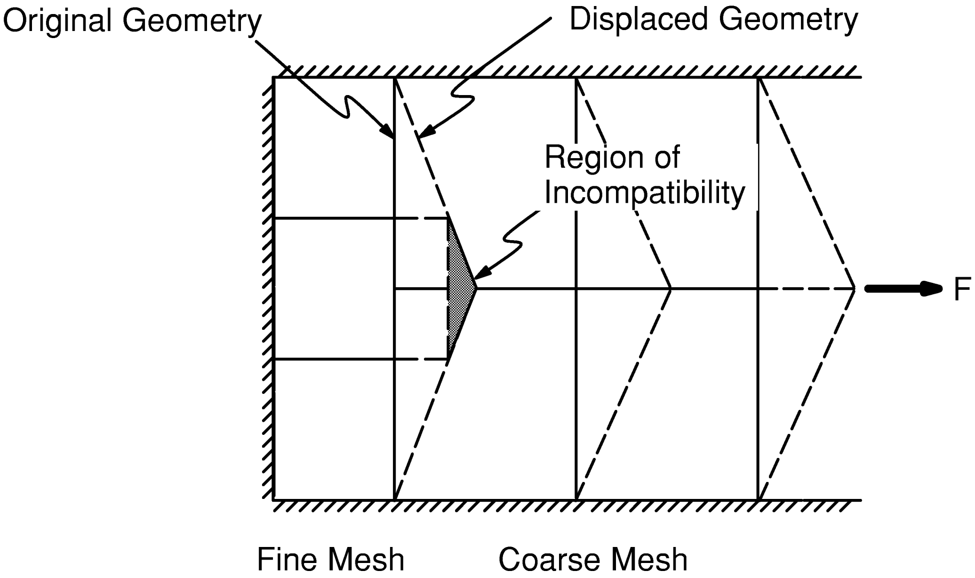


Figure 15.8–3 Incompatibility of Adjacent Elements

15.9 Newton–Raphson Procedure

15.9.1 Overview

The finite element discretization process yields a set of simultaneous equations:

$$[K]\{u\} = \{F^a\} \tag{15.9–1}$$

where:

$[K]$ = coefficient matrix

$\{u\}$ = vector of unknown DOF (degree of freedom) values

$\{F^a\}$ = vector of applied loads

If the coefficient matrix [K] is itself a function of the unknown DOF values (or their derivatives) then equation (15.9–1) is a nonlinear equation. The Newton–Raphson method is an iterative process of solving the nonlinear equations and can be written as (Bathe(2)):

$$\left[K_i^T\right]\{\Delta u_i\} = \{F^a\} - \{F_i^{nr}\} \tag{15.9–2}$$

$$\{u_{i+1}\} = \{u_i\} + \{\Delta u_i\} \tag{15.9–3}$$

where:

$\left[K_i^T\right]$ = Jacobian matrix (tangent matrix)

i = subscript representing the current equilibrium iteration

$\{F_i^{nr}\}$ = vector of restoring loads corresponding to the element internal loads

Both $\left[K_i^T\right]$ and $\{F_i^{nr}\}$ are evaluated based on the values given by $\{u_i\}$. The right–hand side of equation (15.9–2) is the residual or out–of–balance load vector; i.e., the amount the system is out of equilibrium. A single solution iteration is depicted graphically in Figure 15.9–1 for a one DOF model. In a structural analysis, $\left[K_i^T\right]$ is the tangent stiffness matrix, {u$_i$} is the displacement vector and $\{F_i^{nr}\}$ is the restoring force vector calculated from the element stresses. In a thermal analysis, $\left[K_i^T\right]$ is the conductivity matrix, {u$_i$} is the temperature vector and $\{F_i^{nr}\}$ is the resisting load vector calculated from the element heat flows. In an electromagnetic analysis, $\left[K_i^T\right]$ is the Dirichlet matrix, {u$_i$} is the magnetic potential vector, and $\{F_i^{nr}\}$ is the resisting load vector calculated from element magnetic fluxes. In a transient analysis, $\left[K_i^T\right]$ is the effective coefficient matrix and $\left\{F_i^{nr}\right\}$ is the effective applied load vector which includes the inertia and damping effects.

As seen in the following figures, more than one Newton–Raphson iteration is needed to obtain a converged solution. The general algorithm proceeds as follows:

1. Assume $\{u_0\}$. $\{u_0\}$ is usually the converged solution from the previous time step. On the first time step, $\{u_0\}$ = {0}.
2. Compute the updated tangent matrix $\left[K_i^T\right]$ and the restoring load $\left\{F_i^{nr}\right\}$ from configuration $\{u_i\}$.
3. Calculate $\{\Delta u_i\}$ from equation (15.9–2).
4. Add $\{\Delta u_i\}$ to $\{u_i\}$ in order to obtain the next approximation $\{u_{i+1}\}$ (equation (15.9–3)).
5. Repeat steps 2 to 4 until convergence is obtained.

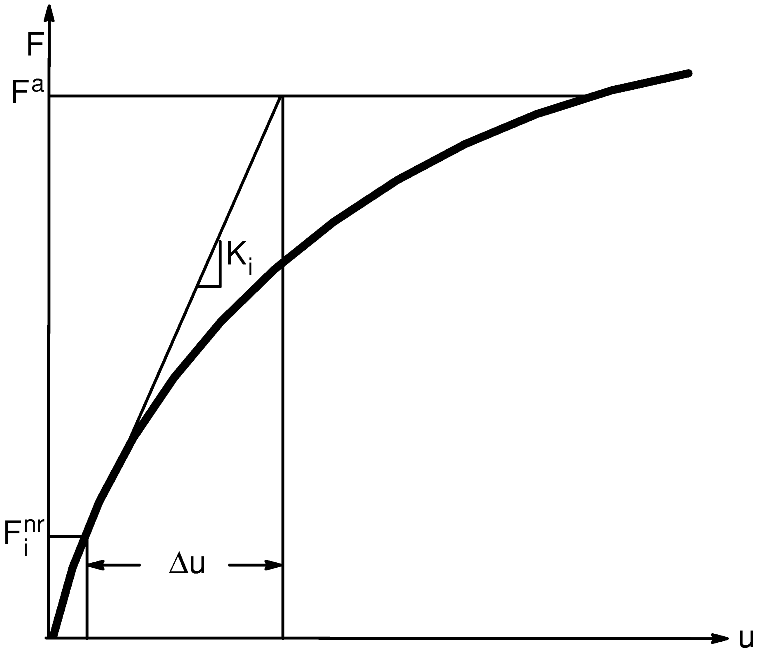


Figure 15.9–1 Newton–Raphson Solution – One Iteration

Figure 15.9–2 shows the solution of the next iteration (i+1) of the example from Figure 15.9–1. The subsequent iterations would proceed in a similar manner.

The solution obtained at the end of the iteration process would correspond to load level $\{F^a\}$. The final converged solution would be in equilibrium, such that the restoring load vector $\{F_i^{nr}\}$ (computed from the current stress state, heat flows, etc.) would equal the applied load vector $\{F^a\}$ (or at least to within some tolerance). None of the intermediate solutions would be in equilibrium.

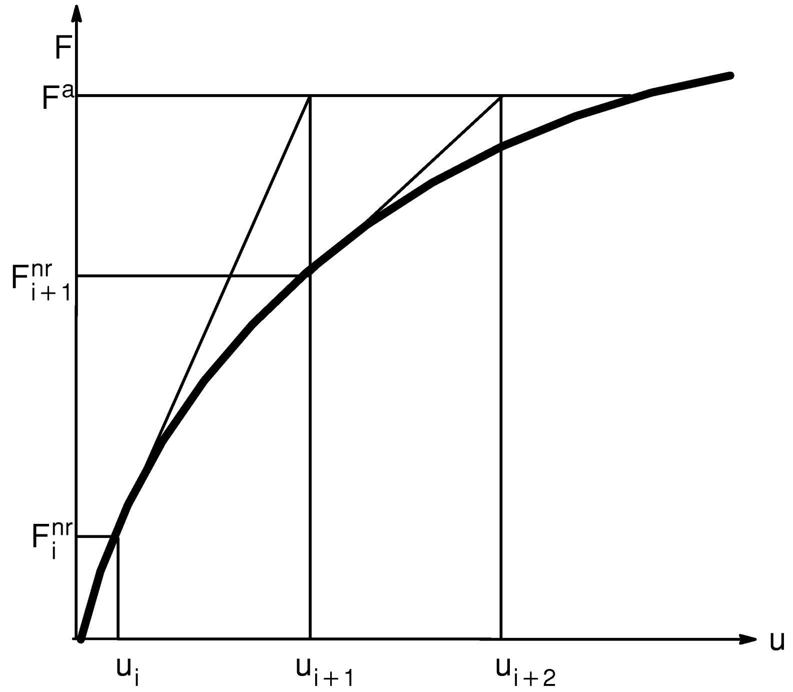


Figure 15.9–2 Newton–Raphson Solution – Next Iteration

If the analysis included path–dependent nonlinearities (such as plasticity), then the solution process requires that some intermediate steps be in equilibrium in order to correctly follow the load path. This is accomplished effectively by specifying a step–by–step incremental analysis; i.e., the final load vector $\{F^a\}$ is reached by applying the load in increments and performing the Newton–Raphson iterations at each step:

$$\left[K^T_{n,i}\right]\{\Delta u_i\} = \{F^a_n\} - \left\{F^{nr}_{n,i}\right\} \qquad (15.9\text{–}4)$$

where: $\left[K_{n,i}\right]$ = tangent matrix for time step n, iteration i

$\{F^a_n\}$ = total applied force vector at time step n

$\left\{F^{nr}_{n,i}\right\}$ = restoring force vector for time step n, iteration i

This process is the incremental Newton–Raphson procedure and is shown in Figure 15.9–3. The Newton–Raphson procedure guarantees convergence if and only if the solution at any iteration $\{u_i\}$ is “near” the exact solution. Therefore, even without a path–dependent nonlinearity, the incremental approach (i.e., applying the loads in increments) is sometimes required in order to obtain a solution corresponding to the final load level.

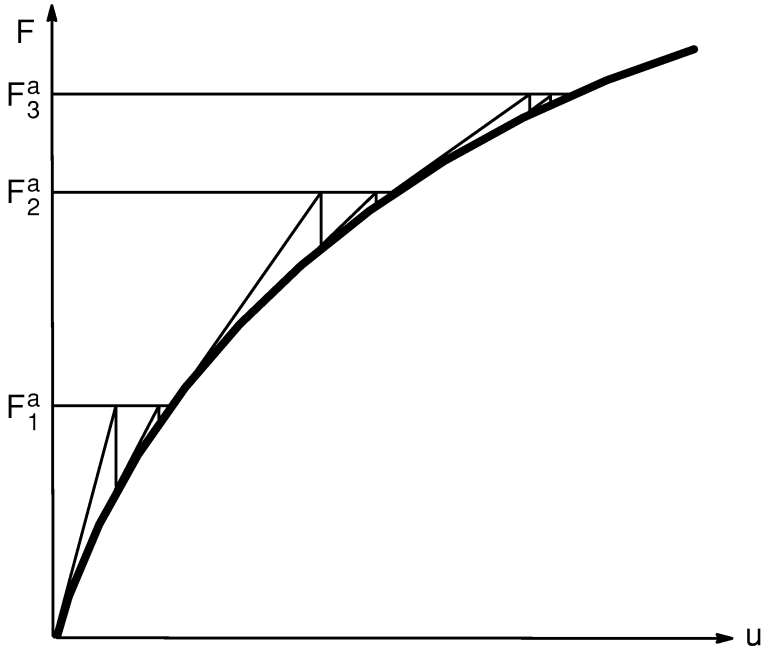


Figure 15.9–3 Incremental Newton–Raphson Procedure

When the stiffness matrix is updated every iteration (as indicated in equations (15.9–2) and (15.9–4)) the process is termed a full Newton–Raphson solution procedure (**NROPT**,FULL). Alternatively, the stiffness matrix could be updated less frequently using the modified Newton–Raphson procedure (**NROPT**,MODI). Specifically, for static or transient analyses, it would be updated only during the first or second iteration of each substep, respectively. Use of the initial–stiffness procedure (**NROPT**,INIT) prevents any updating of the stiffness matrix, as shown in Figure 15.9–4. If a multistatus element is in the model, however, it would be updated at iteration in which it changes status, irrespective of the Newton–Raphson option. The modified and initial–stiffness Newton–Raphson procedures converge more slowly than the full Newton–Raphson procedure, but they require fewer matrix reformulations and inversions. A few elements form an approximate tangent matrix so that the convergence characteristics are somewhat different.

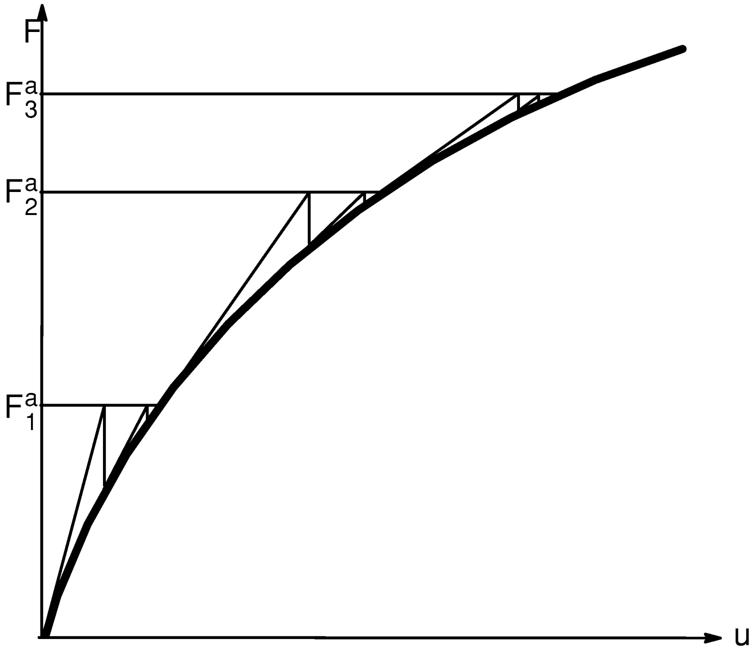


Figure 15.9–4 Initial–Stiffness Newton–Raphson

15.9.2 Convergence

The iteration process described in the previous section continues until convergence is achieved. Up to *NEQIT* (**NEQIT** command) equilibrium iterations are performed in order to obtain convergence.

Convergence is assumed when

$$||\{R\}|| < \epsilon_R \, R_{ref} \qquad \text{(out–of–balance convergence)} \qquad (15.9–5)$$

and/or

$$||\{\Delta u_i\}|| < \epsilon_u \, u_{ref} \qquad \text{(DOF increment convergence)} \qquad (15.9–6)$$

where {R} is the residual vector:

$$\{R\} = \{F^a\} - \{F^{nr}\} \qquad (15.9–7)$$

which is the right–hand side of the Newton–Raphson equation (15.9–2). $\{\Delta u_i\}$ is the DOF increment vector, ϵ_R and ϵ_u are tolerances (*TOLER* on the **CNVTOL** command) and R_{ref} and u_{ref} are reference values (*VALUE* on the **CNVTOL** command). ||•|| is a vector norm; that is, a scalar measure of the magnitude of the vector (defined below).

Convergence, therefore, is obtained when size of the residual (disequilibrium) is less than a tolerance times a reference value and/or when the size of the DOF increment is less than a tolerance times a reference value. The default is to use out–of–balance convergence checking only. The default tolerance are .001 (for both ε_u and ε_R).

There are three available norms (*NORM* on the **CNVTOL** command) to choose from:

1. infinite norm $\|\{R\}\|_\infty = \max |R_i|$
2. L1 norm $\|\{R\}\|_1 = \sum |R_i|$
3. L2 norm $\|\{R\}\|_2 = \left(\sum R_i^2\right)^{\frac{1}{2}}$

For DOF increment convergence, substitute Δu for R in the above equations. The infinite norm is simply the maximum value in the vector (maximum residual or maximum DOF increment), the L1 norm is the sum of the absolute value of the terms, and the L2 norm is the square root of the sum of the squares (SRSS) value of the terms, also called the Euclidean norm. The default is to use the L2 norm.

The default out–of–balance reference value R_{ref} is $||\{F^a\}||$. For DOFs with imposed displacement constraints, $\{F^{nr}\}$ at those DOFs are used in the computation of R_{ref}. For structural DOFs, if $||\{F^a\}||$ falls below 1.0, then R_{ref} uses 1.0 as its value. This occurs most often in rigid body motion (e.g., stress–free rotation) analyses. For thermal DOFs, if $||\{F^a\}||$ falls below 1.0E–6, then R_{ref} uses 1.0E–6 as its value. For all other DOFs, R_{ref} uses 0.0. The default reference value u_{ref} is $||\{u\}||$.

15.9.3 Predictor

The solution used for the start of each time step n $\{u_{n,0}\}$ is usually equal to the current DOF solution $\{u_{n-1}\}$. The tangent matrix $[K_{n,0}]$ and restoring load $\{F^{n,0}\}$ are based on this configuration. The predictor option (**PRED** command) extrapolates the DOF solution using the previous history in order to take a better guess at the next solution.

In static analyses, the prediction is based on the displacement increments accumulated over the previous time step, factored by the time–step size:

$$\{u_{n,0}\} = \{u_{n-1}\} + \beta\{\Delta u_n\} \tag{15.9–8}$$

where: $\{\Delta u_n\}$ = displacement increment accumulated over the previous time step

n = current time step

$$\{\Delta u_n\} = \sum_{i=1}^{NEQIT} \{\Delta u_i\} \tag{15.9–9}$$

and β is defined as:

$$\beta = \frac{\Delta t_n}{\Delta t_{n-1}} \qquad (15.9\text{–}10)$$

where:
Δt_n = current time–step size
Δt_{n-1} = previous time–step size

β is not allowed to be greater than 5.

In transient analyses, the prediction is based on the current velocities and accelerations using the Newmark formulas for structural DOFs:

$$\{u_{n,0}\} = \{u_{n-1}\} + \{\dot{u}_{n-1}\}\Delta t_n + (\tfrac{1}{2} - \alpha)\ \{\ddot{u}_{n-1}\}\Delta t_n^2 \qquad (15.9\text{–}11)$$

where:
$\{u_{n-1}\}, \{\dot{u}_{n-1}\}, \{\ddot{u}_{n-1}\}$ = current displacements, velocities and accelerations
Δt_n = current time–step size
α = Newmark parameter (input on **TINTP** command)

For thermal, magnetic and other first order systems, the prediction is based on the trapezoidal formula:

$$\{u_{n,0}\} = \{u_{n-1}\} + (1 - \theta)\ \{\dot{u}_{n-1}\}\Delta t_n \qquad (15.9\text{–}12)$$

where:
$\{u_{n-1}\}$ = current temperatures (or magnetic potentials)
$\{\dot{u}_{n-1}\}$ = current rates of these quantities
θ = trapezoidal time integration parameter (input on **TINTP** command)

See Section 17.2 for more details on the transient procedures.

The subsequent equilibrium iterations provide DOF increments $\{\Delta u\}$ with respect to the predicted DOF value $\{u_{n,0}\}$, hence this is a predictor–corrector algorithm.

15.9.4 Adaptive Descent

Adaptive descent (*Adptky* on the **NROPT** command) is a technique which switches to a “stiffer” matrix if convergence difficulties are encountered, and switches back to the full tangent as the solution convergences, resulting in the desired rapid convergence rate (Eggert(152)).

The matrix used in the Newton–Raphson equation (equation (15.9–2)) is defined as the sum of two matrices:

$$[K_i^T] = \xi\ [K^S] + (1 - \xi)\ [K^T] \qquad (15.9\text{–}13)$$

where: $[K^S]$ = secant (or most stable) matrix
$[K^T]$ = tangent matrix
ξ = descent parameter

The program adaptively adjusts the descent parameter (ξ) during the equilibrium iterations as follows:

1. Start each substep using the tangent matrix (ξ=0).
2. Monitor the change in the residual $||\{R\}||_2$ over the equilibrium iterations:
   a. If it increases (indicating possible divergence):
      - remove the current solution if $\xi < 1$, reset ξ το 1 and redo the iteration using the secant matrix
      - if already at $\xi = 1$, continue iterating

   b. If it decreases (indicating converging solution):
      - If $\xi = 1$ (secant matrix) and the residual has decreased for three iterations in a row (or 2 if ξ was increased to 1 during the equilibrium iteration process by (a.) above), then reduce ξ by a factor of 1/4 (set it to 0.25) and continue iterating.
      - If the $\xi < 1$, decrease it again by a factor of 1/4 and continue iterating. Once ξ is below 0.0156, set it to 0.0 (use the tangent matrix).
3. If a negative main–diagonal (indicating an ill–conditioned matrix) is encountered:
   - If $\xi < 1$, remove the current solution, reset $\xi = 1$ and redo the iteration using the secant matrix.
   - If $\xi = 1$, bisect the time step if automatic time stepping is active, otherwise terminate the execution.

The nonlinearities which make use of adaptive descent (that is, they form a secant matrix if $\xi > 0$) include: plasticity, contact, stress stiffness with large strain, nonlinear magnetics using the scalar potential formulation, the concrete element SOLID65 with KEYOPT(7) = 1, and the membrane shell element SHELL41 with KEYOPT(1) = 2. Adaptive descent is used by default in these cases unless the line search or arc–length options are on. It is only available with full Newton–Raphson, where the matrix is updated every iteration. Full Newton–Raphson is also the default for plasticity, contact and large strain nonlinearities.

Elements where use of adaptive descent is possible are: PLANE2, SOLID5, COMBIN7, CONTAC12, PLANE13, CONTAC26, COMBIN37, COMBIN40, SHELL41, PLANE42, SHELL43, SOLID45, CONTAC48, CONTAC49, CONTAC52, SOLID62, SOLID64, SOLID65, PLANE82, VISCO88, VISCO89, SHELL91, SOLID92, SHELL93, SOLID95, SOLID96, SOLID98, VISCO106, VISCO107, and VISCO108.

15.9.5 Line Search

The line search option (**LNSRCH** command) attempts to improve a Newton–Raphson solution $\{\Delta u_i\}$ by scaling the solution vector by a scalar value termed the line search parameter.

Consider equation (15.9–3) again:

$$\{u_{i+1}\} = \{u_i\} + \{\Delta u_i\} \tag{15.9–14}$$

In some solution situations, the use of the full $\{\Delta u_i\}$ leads to solution instabilities. Hence, if the line search option is used, equation (15.9–14) is modified to be:

$$\{u_{i+1}\} = \{u_i\} + s\{\Delta u_i\} \tag{15.9–15}$$

where: s = line search parameter, $0.05 < s < 1.0$

s is automatically determined by minimizing the energy of the system, which reduces to finding the zero of the nonlinear equation:

$$g_s = \{\Delta u_i\}^T \left(\{F^a\} - \{F^{nr}(s\{\Delta u_i\})\}\right) \tag{15.9–16}$$

where: g_s = gradient of the potential energy with respect to s

An iterative solution scheme based on regula falsi is used to solve equation (15.9–16) (Schweizerhof and Wriggers (153)). Iterations are continued until either:

1. g_s is less than 0.5 g_o, where g_o is the value of equation (15.9–16) at s=0.0 (that is, using $\{F^{nr}_{n-1}\}$ for $\{F^{nr}$ $(s\{\Delta u\})\}$).
2. g_s is not changing significantly between iterations.
3. Six iterations have been performed.

If $g_o > 0.0$, no iterations are performed and s is set to 1.0. s is not allowed below 0.05.

The scaled solution $\{\Delta u_i\}$ is used to update the current DOF values $\{u_{i+1}\}$ in equation (15.9–3) and the next equilibrium iteration is performed.

15.9.6 Arc–Length Method

The arc–length method (**ARCLEN**,ON) is suitable for nonlinear static equilibrium solutions of unstable problems. Applications of the arc–length method involves the tracing of a complex path in the load–displacement response into the buckling/post buckling regimes. The arc–length method uses the explicit spherical iterations to maintain the orthogonality between the arc–length radius and orthogonal directions as described by Forde and Stiemer(174). It is assumed that all load magnitudes are controlled by a single scalar parameter (i.e., the total load factor). Unsmooth or discontinuous load–displacement response in the cases often seen in contact analyses and elastic–perfectly plastic analyses cannot be traced effectively by the arc–length solution method. Mathematically, the arc–length method can be viewed as the trace of a single equilibrium curve in a space spanned by the nodal displacement variables and the total load factor. Therefore, all options of the Newton–Raphson method are still the basic method for the arc–length solution. As the displacement vectors and the scalar load factor are treated as unknowns, the arc–length method itself is an automatic load step method (**AUTOTS**,ON is not needed). For problems with sharp turns in the load–displacement curve or path dependent materials, it is necessary to limit the arc–length radius (arc–length load step size). The initial arc–length radius is defined by the **NSUBST** command. During the solution, the arc–length method will vary the arc–length radius at each arc–length substep according to the degree of nonlinearities that is involved.

Parameters MAXARC and MINARC defined by the **ARCLEN** command can be used to limit the range of variation of the arc–length radius.

In the arc–length procedure, nonlinear equation (15.9–2) is recast associated with the total load factor λ:

$$\left[K_i^T\right]\left\{\Delta u_i\right\} = \lambda \left\{F^a\right\} - \left\{F_i^{nr}\right\} \qquad (15.9–17)$$

where λ is normally within the range –1.0 ≥ λ ≥ 1.0. Writing the proportional loading factor λ in an incremental form yields at substep n and iteration i (see Figure 15.9–5):

$$\left[K_i^T\right]\left\{\Delta u_i\right\} - \Delta\lambda \left\{F^a\right\} = (\lambda_n + \lambda_i) \{F^a\} - \left\{F_i^{nr}\right\} = - \{R_i\} \qquad (15.9–18)$$

where: Δλ = incremental load factor (as shown in Figure 15.9–5)

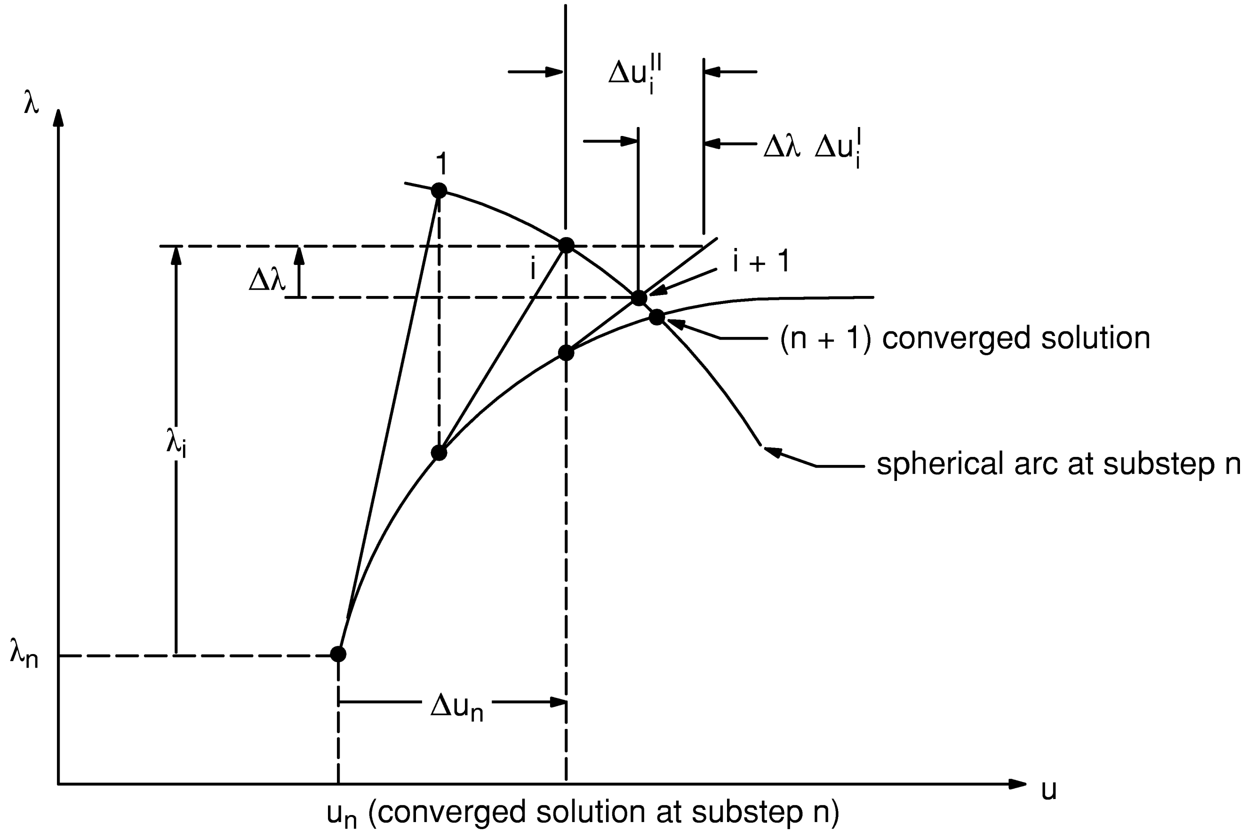

λ
Δu_i^{II}
$\Delta\lambda\ \Delta u_i^{I}$
1
i
i + 1
Δλ
(n + 1) converged solution
λ_i
spherical arc at substep n
λ_n
Δu_n
u
u_n (converged solution at substep n)

Figure 15.9–5 Arc–Length Approach with Full Newton–Raphson Method

The incremental displacement $\{\Delta u_i\}$ can be written into two parts following equation (15.9–18):

$$\{\Delta u_i\} = \Delta\lambda \{\Delta u_i^{I}\} + \{\Delta u_i^{II}\} \tag{15.9–19}$$

where:

$\{\Delta u_i^{I}\}$ = displacement due to a unit load factor

$\{\Delta u_i^{II}\}$ = displacement increment from the conventional Newton–Raphson method

These are defined by:

$$\{\Delta u_i^{I}\} = [K_i^T]^{-1} \{F^a\} \tag{15.9–20}$$

$$\{\Delta u_i^{II}\} = -[K_i^T]^{-1} \{R_i\} \tag{15.9–21}$$

In each arc–length iteration, it is necessary to use equations (15.9–20) and (15.9–21) to solve for $\left\{\Delta u_i^I\right\}$ and $\left\{\Delta u_i^{II}\right\}$. The incremental load factor $\Delta\lambda$ in equation (15.9–19) is determined by the arc–length equation which can be written as, for instance, at iteration i (see Figure 15.9–5):

$$\ell_i^2 = \lambda_i^2 + \beta^2 \; \{\Delta u_n\}^T\{\Delta u_n\} \qquad (15.9\text{–}22)$$

where: β = scaling factor (with units of displacement) used to ensure the correct scale in the equations

Δu_n = sum of all the displacement increments Δu_i of this iteration

The arc–length radius ℓ_i is forced, during the iterations, to be identical to the radius iteration ℓ_1 at the first iteration, i.e.

$$\ell_i = \ell_{i-1} = \ . \ . \ . = \ell_1 \qquad (15.9\text{–}23)$$

While the arc–length radius ℓ_1 at iteration 1 of a substep is determined by using the initial arc–length radius (defined by the **NSUBST** command), the limit range (defined by the **ARCLEN** command) and some logic of the automatic time (load) step method (Section 15.6).

Equations (15.9–19) together with (15.9–22) uniquely determines the solution vector $\{\Delta u_i, \; \Delta\lambda\}^T$. However, there are many ways to solve for $\Delta\lambda$ approximately. The explicit spherical iteration method is used to ensure orthogonality (Forde and Stiemer(174)). In this method, the required residual r_i (a scalar) for explicit iteration on a sphere is first calculated. Then the arc–length load increment factor is determined by formula:

$$\Delta\lambda = \frac{r_i - \{\Delta u_n\}^T \left\{\Delta u_i^{II}\right\}}{\beta^2\lambda_i + \{\Delta u_n\}^T \left\{\Delta u_i^{I}\right\}} \qquad (15.9\text{–}24)$$

The method works well even in the situation where the vicinity of the critical point has sharp solution changes. Finally, the solution vectors are updated according to (see Figure 15.9–5):

$$\{u_{i+1}\} \;\; = \;\; \{u_n\} + \{\Delta u_n\} + \{\Delta u_i\} \qquad (15.9\text{–}25)$$

and

$$\lambda_{i+1} \;\; = \;\; \lambda_n + \lambda_i + \Delta\lambda \qquad (15.9\text{–}26)$$

where: n = current substep number

Values of λ_n and $\Delta\lambda$ are available in POST26 (**SOLU** command) corresponding to labels ALLF and ALDLF, respectively. The normalized arc–length radius label ARL (POST26) corresponds to value $\ell_i \;/\; \ell_i^0$, where ℓ_i^0 is the initial arc–length radius defined by the **NSUBST** command through equation (15.9–22) (an arc–length radius at the first iteration of the first substep).

In the case where the applied loads are greater or smaller than the maximum or minimum critical loads, arc–length will continue the iterations in cycles because $|\lambda|$ does not approach unity. It is recommended to use the **ARCTRM** or **NCNV** commands to terminate the arc–length iterations.

15.10 Eigenvalue and Eigenvector Extraction

The eigenvalue and eigenvector problem needs to be solved for mode–frequency and buckling analyses. It has the form of:

$$[K]\{\phi_i\} = \lambda_i [M]\{\phi_i\} \qquad (15.10\text{–}1)$$

where:
$[K]$ = structure stiffness matrix
$\{\phi_i\}$ = eigenvector
λ_i = eigenvalue
$[M]$ = structure mass matrix

For prestressed modal analyses, the [K] matrix includes the stress stiffness matrix [S]. For eigenvalue buckling analyses, the [M] matrix is replaced with the stress stiffness matrix [S]. The discussions given in the rest of this section assume a modal analysis (**ANTYPE**,MODAL) except as noted, but are equally applicable to eigenvalue buckling analyses.

The eigenvalue and eigenvector extraction procedures available include the reduced, subspace, block Lanczos, unsymmetric, damped, and QR damped methods (**MODOPT** command) outlined in Table 15.10–1. The Power Dynamics method uses subspace iterations, but employs the PCG solver. Each method is discussed subsequently. Shifting, applicable to all methods, is discussed at the end of this section.

Table 15.10–1 Eigenvalue and Eigenvector Extraction Procedures

Type of Procedure	**Full Extraction from Reduced Matrices**	**Partial Extraction from Full Matrices**				
Method	Reduced	Subspace	Block Lanczos	Unsymmetric eigensolver	Damped eigensolver	QR damped eigensolver
Usages	Any (but not recommended for buckling)	Symmetric	Symmetric (not available for buckling)	Unsymmetric matrices	Symmetric or unsymmetric damped systems	Symmetric damped systems
Input	**MODOPT**, REDUC	**MODOPT**, SUBSP	**MODOPT**, LANB	**MODOPT**, UNSYM	**MODOPT**, DAMP	**MODOPT**, QRDAMP
Guyan Reduction	Yes	No	No	No	No	No
Extraction Technique	HBI	Subspace, which internally uses Jacobi	Lanczos which internally uses QL algorithm	Lanczos, which internally uses QR iterations		QR algorithm for reduced modal damping matrix

The PowerDynamics method is the same as the subspace method, except it uses the iterative solver instead of the frontal direct equation solver to solve equation (15.10–13).

15.10.1 Reduced Method (**MODOPT**,REDUC)

For the reduced procedure, the system of equations is first condensed down to those DOFs associated with the master DOFs by Guyan reduction. This condensation procedure is discussed in Section 17.6 (equations (17.6–8) and (17.6–19)). The set of n master DOFs characterize the natural frequencies of interest in the system. The selection of the master DOFs is discussed in more detail in Section 15.5 of this manual and in Chapter 3 of the *ANSYS Structural Analysis Guide*. This technique preserves the potential energy of the system but modifies, to some extent, the kinetic energy. The kinetic energy of the low frequency modes is less sensitive to the condensation than the kinetic energy of the high frequency modes. The number of master DOFs selected should usually be at least equal to twice the number of frequencies of interest. This reduced form may be expressed as:

$$\left[\hat{K}\right]\left\{\hat{\phi}_i\right\} = \lambda_i \left[\hat{M}\right]\left\{\hat{\phi}_i\right\} \tag{15.10–2}$$

where: $\left[\hat{K}\right]$ = reduced stiffness matrix (known)

$\left\{\hat{\phi}_i\right\}$ = eigenvector (unknown)

λ_i = eigenvalue (unknown)

$\left[\hat{M}\right]$ = reduced mass matrix (known)

Next, the actual eigenvalue extraction is performed. The extraction technique employed is the HBI (Householder–Bisection–Inverse iteration) extraction technique and consists of the following five steps:

A. Transformation of the Generalized Eigenproblem to a Standard Eigenproblem

Equation (15.10–2) must be transformed to the desired form which is the standard eigenproblem (with [A] being symmetric):

$$[A]\{\psi\} = \lambda\{\psi\} \tag{15.10–3}$$

This is accomplished by the following steps:

Premultiply both sides of equation (15.10–2) by $\left[\hat{M}\right]^{-1}$:

$$\left[\hat{M}\right]^{-1}\left[\hat{K}\right]\left\{\hat{\phi}\right\} = \lambda\left\{\hat{\phi}\right\} \tag{15.10–4}$$

Decompose $\left[\hat{M}\right]$ into $[L][L]^T$ by Cholesky decomposition, where [L] is a lower triangular matrix. Combining with equation (15.10–4),

$$[L]^{-T}\,[L]^{-1}\left[\hat{K}\right]\left\{\hat{\phi}\right\} = \lambda\left\{\hat{\phi}\right\} \tag{15.10–5}$$

It is convenient to define:

$$\left\{\hat{\phi}\right\} = [L\}^{-T}\{\psi\} \tag{15.10–6}$$

Combining equations (15.10–5) and (15.10–6), and reducing yields:

$$[L]^{-1}\left[\hat{K}\right][L]^{-T}\{\psi\} = \lambda\{\psi\} \tag{15.10–7}$$

or

$$[A]\{\psi\} = \lambda\,\{\psi\} \tag{15.10–8}$$

where: $[A] = [L]^{-1}\left[\hat{K}\right][L]^{-T}$

Note that the symmetry of [A] has been preserved by this procedure.

B. Reduce [A] to tridiagonal form

This step is performed by Householder's method through a series of similarity transformations yielding

$$[B] = [T]^T[A][T] \tag{15.10–9}$$

where: $[B]$ = tridiagonalized form of $[A]$,

$[T]$ = matrix constructed to tridiagonalize $[A]$, solved for iteratively (Bathe(2))

The eigenproblem is reduced to:

$$[B]\{\psi\} = \lambda\{\psi\} \tag{15.10–10}$$

Note that the eigenvalues (λ) have not changed through these transformations, but the eigenvectors are related by:

$$\left\{\hat{\phi}_i\right\} = [L]^{-T}\,[T]\{\psi_i\} \tag{15.10–11}$$

C. Eigenvalue calculation

Use Sturm sequence checks with the bisection method to determine the eigenvalues.

D. Eigenvector calculation

The eigenvectors are evaluated using inverse iteration with shifting. The eigenvectors associated with multiple eigenvalues are evaluated using initial vector deflation by Gram–Schmidt orthogonalization in the inverse iteration procedure.

E. Eigenvector transformation

After the eigenvectors $\{\psi_i\}$ are evaluated, $\left\{\hat{\phi}_i\right\}$ mode shapes are recovered through equation (15.10–11).

In the expansion pass, the eigenvectors are expanded from the master DOFs to the total DOFs.

15.10.2 Subspace Method (**MODOPT**,SUBSP)

The subspace iteration method is described in detail by Bathe(2). Enhancements as suggested by Wilson and Itoh(166) are also included as outlined subsequently. The basic algorithm consists of the following steps:

1. Define the initial shift s:

   a. In a modal analysis (**ANTYPE**,MODAL), s = *FREQB* on the **MODOPT** command (defaults to $-4\pi^2$).

   b. In a buckling analysis (**ANTYPE**,BUCKLE), s = *SHIFT* on the **BUCOPT** command (defaults to 0.0).

2. Initialize the starting vectors $[X_0]$ (described below).

3. Triangularize the shifted matrix

$$[K^*] = [K] + s[M] \tag{15.10–12}$$

where: [K] = assembled stiffness matrix
[M] = assembled mass (or stress stiffness) matrix

A Sturm sequence check (described below) is performed if this is a shift point other than the initial shift and it is requested (*Strmck*=ALL (default) or PART on the **SUBOPT** command).

4. For each subspace iteration n (1 to *NUMSSI* (**SUBOPT** command)), do steps 5 to 14:

5. Form $[F] = [M][X_{n-1}]$ and scale [F] by $\{\lambda_{n-1}\}$

where: $\{\lambda_{n-1}\}$ = previously estimated eigenvalues

6. Solve for $[\overline{X}_n]$:

$$[K^*][\overline{X}_n] = [F] \tag{15.10–13}$$

These equations are solved using the frontal direct equation solver (**EQSLV**,FRONT) or the iterative PCG solver (**EQSLV**,PCG) (which is known as the PowerSolver.).

7. Scale the vectors $[\overline{X}_n]$ by $\{(\lambda_{n-1} - s) / \lambda_{n-1}\}$

8. M–orthogonalize the vectors to the previously converged vectors (Gram–Schmidt orthogonalization).

9. Define the subspace matrices $[\overline{K}]$ and $[\overline{M}]$:

$$[\overline{K}] = [\overline{X}_n]^T[K][\overline{X}_n] \tag{15.10–14}$$

$$[\overline{M}] = [\overline{X}_n]^T[M][\overline{X}_n] \tag{15.10–15}$$

10. Adjust for the shift, $[\overline{K}^*] = [\overline{K}] + s[\overline{M}]$

11. Compute the eigenvalues and vectors of the subspace using a generalized Jacobi iteration:

$$[\bar{K}^*][Q] = [\bar{M}][Q]\{\lambda_n\} \qquad (15.10\text{–}16)$$

where: $[Q]$ = subspace eigenvectors
$\{\lambda_n\}$ = updated eigenvalues

12. Update the approximation to the eigenvectors:

$$[X_n] = [\bar{X}_n][Q] \qquad (15.10\text{–}17)$$

13. If any negative or redundant modes are found, remove them and create a new random vector.
14. Check for convergence (described below):

    a. All requested modes converged? If yes, go to step 15.

    b. If a new shift is required (described below), go to step 3

    c. Go to the next iteration, step 4
15. Perform a final Sturm sequence check if requested (*Strmck*=ALL (default) on the **SUBOPT** command).

Steps 5 thru 12 are only done on the unconverged vectors: once an eigenvalue has converged, the associated eigenvector is no longer iterated on. The Gram–Schmidt procedure (step 8) ensures that the unconverged eigenvectors remain orthogonal to the converged vectors not being iterated on. The remainder of this section details some of the steps involved.

Convergence

The convergence check (step 14a) requires that all of the requested eigenvalues satisfy the convergence ratio:

$$e_i = \left| \frac{(\lambda_i)_n - (\lambda_i)_{n-1}}{B} \right| < \text{tol} \qquad (15.10\text{–}18)$$

where: $(\lambda_i)_n$ = value of ith eigenvalue as computed in iteration n
$(\lambda_i)_{n-1}$ = value of ith eigenvalue as computed in iteration n–1
$B = \begin{cases} 1.0 \\ (\lambda_i)_n \end{cases}$ whichever is greater
tol = tolerance value, set to 1.0E–5

Starting Vectors

The number of starting (iteration) vectors used is determined from:

$$q = p + d \qquad (15.10\text{–}19)$$

where:
p = requested number of modes to extract (*NMODE* on the **MODOPT** or **BUCOPT** commands)
d = number of extra iteration vectors to use (*NPAD* on the **SUBOPT** command, defaults to 4)

The q starting vectors $[X_0]$ (step 2) are initialized as follows. For each predefined rigid–body motion (*Dof* on the **RIGID** command), define a rigid–body vector:

1. if a translational rigid–body motion, set the DOF slot in $\{X_0\}$ to 1.0 ($\{X_0\}$ is a column of $[X_0]$).
2. if a rotational rigid–body motion, set the DOF slot in $\{X_0\}$ corresponding to a unit rotation about the global origin corresponding to the *Dof* label.

The rigid–body vectors are M–orthogonalized (Gram–Schmidt orthogonalization). The remainder of the vectors are initialized to random vectors.

Sturm Sequence Check

The Sturm sequence check computes the number of negative pivots encountered during the triangularization of the shifted matrix $[K^*]$. This number will match the number of converged eigenvalues unless some eigenvalues have been missed. In that case, more iteration vectors must be used (*NPAD* on the **SUBOPT** command) or the initial shift (see step 1) was past the first mode. For the final Sturm sequence check, the shift used is defined as:

$$s = \lambda_p + 0.1\,(\lambda_{p+1} - \lambda_p) \qquad (15.10\text{–}20)$$

where:
λ_p = eigenvalue of the last requested mode
λ_{p+1} = eigenvalue of the next computed mode

Shifting Strategy

In order to improve the rate of convergence during the iteration process, a shifting strategy is adopted as follows (step 14b):

1. If the current converged mode(s) is zero(s) and the next mode i+1 is non–zero, shift to just below the non–zero mode:

$$s = \lambda_{i+1} - \begin{cases} .05\lambda_{i+1} & \text{if } \lambda_{i+1} \text{ is close to being converged} \\ .5\lambda_{i+1} & \text{if not} \end{cases} \qquad (15.10\text{–}21)$$

2. If the number of iterations since the last shift exceeds *NSHIFT* (on the **SUBOPT** command, defaults to 5), then shift to just below the next unconverged mode i+1:

$$s = \lambda_{i+1} - \begin{cases} .05(\lambda_{i+1} - \lambda_i) & \text{if } \lambda_{i+1} \text{ is close to being converged} \\ .5(\lambda_{i+1} - \lambda_i) & \text{if not} \end{cases} \qquad (15.10\text{–}22)$$

If the mode is part of a cluster, then the next lowest unique mode is used to define the shift. If this is the first shift, then use:

$$s = .9\lambda_1 \qquad (15.10\text{–}23)$$

The more shifts that are allowed (smaller *NSHIFT* value), the faster the convergence, but the more matrix triangularizations that must be performed.

Sliding Window

To improve the efficiency of the iterations, a subset q_w of the q iteration vectors may be used in the iteration process, where q_w is defined with *SUBSIZ* on the **SUBOPT** command (q_w defaults to q). Steps 2 through 14 are performed using these working vectors. When a vector converges, it is removed from the iteration process and it is replaced by a new random vector until all p requested vectors have been found.

Work Space Requirements

In order to reduce the amount of work space (memory) required for very large models, only a few of the q_w vectors may be kept in memory and the rest kept on a scratch file on disk. *NPERBK* on the **SUBOPT** command specifies how many vectors (q_m) are kept in memory (default is to process all q vectors in memory).

The work space required (in double precision words) is the maximum of:

$$M = 3q + N + N_m + 2Nq_m + 3q^2 \qquad (15.10\text{–}24)$$

or

$$M = 3q + 2N + w(w + 1)/2 + 3(w + 2) + 2(w + 2) \qquad (15.10\text{–}25)$$

where:

q = total number of iteration vectors, *NMODE + NPAD* (*NMODE* is from the **MODOPT** or **BUCOPT** command and *NPAD* is from the **SUBOPT** command (defaults to 4))

N = total number of DOFs

N_m = 1 if the mass matrix is not lumped (**LUMPM**,OFF) N if the mass matrix is lumped (**LUMPM**,ON)

q_m = number of in–memory working vectors (*NPERBK* on the **SUBOPT** command, defaults to q) w = maximum wavefront

This is output (in megabytes) under the heading “WORK SPACE”, which also includes other (non–changeable) work space requirements. To increase the available work space, use the ANSYS command line option “–m”.

15.10.3 Block Lanczos (MODOPT,LANB)

The block Lanczos eigenvalue extraction method is available for large symmetric eigenvalue problems. Typically, this solver is applicable to the type of problems solved using the subspace eigenvalue method, however, at a faster convergence rate.

A block shifted Lanczos algorithm, as found in Grimes et al(195) is the theoretical basis of the eigensolver. The method employs an automated shift strategy, combined with Sturm sequence checks, to extract the number of eigenvalues requested. The Sturm sequence check also ensures that the requested number of eigenfrequencies beyond the user supplied shift frequency (FREQE on the **MODOPT** command) is found without missing any modes.

The block Lanczos algorithm is a variation of the classical Lanczos algorithm, where the Lanczos recursions are performed using a block of vectors, as opposed to a single vector. Additional theoretical details on the classical Lanczos method can be found in Rajakumar and Rogers(196).

Use of the block Lanczos method for solving large models (100,000 DOF, for example) with many constraint equations (CE) can require a significant amount of computer memory. This occurs when certain constraint equations lead to a huge wavefront size. For this reason, the Lagrange Multiplier approach is implemented to treat constraint equations in the block Lanczos eigensolver, rather than explicitly eliminating them prior to writing matrices to file.full. For details about the Lagrange Multiplier formulation theory refer to Cook(5).

15.10.4 Unsymmetric Method (MODOPT,UNSYM)

The unsymmetric eigensolver is applicable whenever the system matrices are unsymmetric. For example, an acoustic fluid–structure interaction problem using FLUID30 elements results in unsymmetric matrices. Also, certain problems involving the input matrix element MATRIX27, such as in rotor dynamics can give rise to unsymmetric system matrices. A generalized eigenvalue problem given by the following equation

$$[K]\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{15.10–26}$$

can be setup and solved using the mode–frequency analysis option **ANTYPE**,MODAL. The matrices [K] and [M] are the system stiffness and mass matrices, respectively. Either or both [K] and [M] can be unsymmetric. $\{\phi_i\}$ is the eigenvector.

The method employed to solve the unsymmetric eigenvalue problem is the Lanczos algorithm. Starting from two random vectors $\{v_1\}$ and $\{w_1\}$, the system matrices [K] and [M] (size n) are transformed into a tridiagonal matrix [B] (subspace size q, where $q \le n$), through the Lanczos biorthogonal transformation as discussed in Rajakumar and Rogers(16). Eigenvalues of the [B] matrix, μ_i, are computed as approximations of the original system eigenvalues λ_i. The QR algorithm (Wilkinson(18)) is used to extract the eigenvalues of the [B] matrix. As the subspace size q is increased, μ's will converge to closely approximate the eigenvalues of the original problem.

The transformed problem is a standard eigenvalue problem given by:

$$[B]\{y_i\} = \mu_i\{y_i\} \tag{15.10–27}$$

The eigenvalues and eigenvectors of equations (15.10–26) and (15.10–27) are related by:

$$\lambda_i = \frac{1}{\mu_i} \tag{15.10–28}$$

$$\{\phi_i\} = [V]\{y_i\} \tag{15.10–29}$$

where: $[V]$ = matrix of Lanczos vectors (size n x q).

For the unsymmetric modal analysis, the real part (ω_i) of the complex frequency is used to compute the element kinetic energy.

This method does not perform a Sturm Sequence check for possible missing modes. At the lower end of the spectrum close to the shift (**FREQB**), the frequencies usually converge without missing modes.

15.10.5 Damped Method (MODOPT,DAMP)

The damped eigensolver is applicable only when the system damping matrix needs to be included in equation (15.10–1), where the eigenproblem becomes a quadratic eigenvalue problem given by:

$$[K]\{\phi_i\} + \bar{\lambda}_i[C]\{\phi_i\} = -\bar{\lambda}_i^2[M]\{\phi_i\} \tag{15.10–30}$$

where: $\bar{\lambda}_i = \sqrt{-\lambda_i}$ (defined below)

[C] = damping matrix

Matrices may be symmetric or unsymmetric. For problems involving rotordynamic stability, spinning structures with gyroscopic effects, and/or damped structural eigenfrequencies, the above equation needs to be solved to get the complex eigenvalues $\bar{\lambda}_i$ given by:

$$\bar{\lambda}_i = \sigma_i \pm j\,\omega_i \qquad (15.10\text{–}31)$$

where:

$\bar{\lambda}_i$ = complex eigenvalue

σ_i = real part of the eigenvalue

ω_i = imaginary part of the eigenvalue

j = $\sqrt{-1}$

The dynamic response of the system is given by:

$$\{u_i\} = \{\phi_i\}\, e^{(\sigma_i \pm j\,\omega_i)\,t} \qquad (15.10\text{–}32)$$

where:

t = time

For the ith eigenvalue, the system is stable if σ_i is negative and unstable if σ_i is positive.

The method employed to solve the damped eigenvalue problem is the Lanczos algorithm (Rajakumar and Ali(142)). Starting from four random vectors $\{v_1\}$, $\{w_1\}$, $\{p_1\}$, and $\{q_1\}$, the system matrices [K], [M], and [C] are transformed into a subspace tridiagonal matrix [B] of size q ($q \le n$), through the Lanczos generalized biorthogonal transformation. Eigenvalues of the [B] matrix, μ_i, are computed as an approximation of the original system eigenvalues $\bar{\lambda}_i$. The QR algorithm (Wilkinson(18)) is used to extract the eigenvalues of the [B] matrix. As the subspace size q is increased, the eigenvalues μ_i will converge to closely approximate the eigenvalues of the original problem. The transformed problem is given by equation (15.10–27) and from there on, the eigenvalues and eigenvectors computation follow along the same lines as for the unsymmetric eigensolver.

This method does not perform a Sturm Sequence check for possible missing modes. At the lower end of the spectrum close to the shift (**FREQB**), the frequencies usually converge without missing modes.

For the damped modal analysis, the imaginary part (ω_i) of the complex frequency is used to compute the element kinetic energy.

15.10.6 QR Damped Method (MODOPT, QRDAMP)

The QR damped method is a procedure for determining the complex eigenvalues of damped linear systems. When a symmetric damping matrix is present, this method is more computationally efficient than the damped method. This method employs the modal coordinate transformation of system matrices (orthogonal to mass and stiffness matrices) to reduce the eigenproblem into the modal subspace. QR algorithm is then used to calculate eigenvalues of the resulting upper Hessenberg matrix in modal subspace.

The equations of elastic structural systems without external excitation can be written in the following form:

$$[M]\ \{\ddot{u}\} + [C]\ \{\dot{u}\} + [K]\ \{u\} = \{0\} \qquad (15.10\text{–}33)$$

(See equation (17.2–1) for definitions).

For both classically (α and β damping) and non–classically damped structures, it has been recognized that operating in the modal subspace is more convenient than in the nodal space. In order to reduce the number of variables, the following coordinate transformation (see equation (15.11–3)) is used:

$$\{u\} = [\Phi]\ \{y\} \qquad (15.10\text{–}34)$$

where:

$[\Phi]$ = eigenvector matrix normalized with respect to the mass matrix [M]

$\{\phi_i\}$ = ith column of $[\Phi]$ = eigenvector of mode i

$\{y\}$ = vector of modal coordinates

n = number of modes to extract

By using equation (15.10–34) in equation (15.10–33), we can write the differential equations of motion in the modal subspace as follows:

$$[I]\ \{\ddot{y}\} + [\Phi]^T[C]\ [\Phi]\ \{\dot{y}\} + [\Lambda^2]\ \{y\} = \{0\} \qquad (15.10\text{–}35)$$

where: $[\Lambda]$ = a diagonal matrix containing the first n eigen frequencies ω_i

The modal damping matrix $[\Phi]^T[C][\Phi]$ is either a diagonal matrix with each diagonal term being $2\xi_i\omega_i$ (where ξ_i is the damping ratio of the ith mode) for classically damped systems or a symmetric and non–negative definite matrix for non–classically damped systems.

Introducing the 2n–dimensional state variable vector approach, equation (15.10–35) can be written in reduced form as follows:

$$\{\dot{Z}\} = [D]\ \{Z\} \qquad (15.10\text{–}36)$$

where:

$$\{Z\} = \begin{Bmatrix} \{y\} \\ \{\dot{y}\} \end{Bmatrix}$$

$$[D] = \begin{bmatrix} [O] & [I] \\ -[\Lambda^2] & -[\Phi]^T[C][\Phi] \end{bmatrix}$$

The 2n eigenvalues of equation (15.10–36) are calculated by using the QR algorithm (Press et al(254).

15.10.7 Shifting

In some cases it is desirable to shift the values of eigenvalues either up or down. These fall in two categories:

1. Shifting down, so that the solution of problems with rigid body modes does not require working with a singular matrix. or

or

2. Shifting up, so that the bottom range of eigenvalues will not be computed, because they had effectively been converted to negative eigenvalues. This will, in general, result in better accuracy for the higher modes. The shift introduced is:

$$\lambda = \lambda_o + \lambda_i \tag{15.10–37}$$

where:

λ = desired eigenvalue

λ_o = eigenvalue shift

λ_i = eigenvalue that is extracted

λ_o, the eigenvalue shift is computed as:

$$\lambda_o = \begin{cases} \text{Input quantity } \textit{SHIFT}\text{, } \textbf{BUCOPT} \text{ command if } \textbf{ANTYPE}\text{,BUCKL analysis} \\ \text{or} \\ (2\pi f)^2 \text{ where f = input quantity } \textit{FREQB}\text{, } \textbf{MODOPT} \text{ command if } \textbf{ANTYPE}\text{,MODAL analysis} \end{cases} \tag{15.10–38}$$

Equation (15.10–37) is combined with equation (15.10–1) to give:

$$[K]\{\phi_i\} = (\lambda_o + \lambda_i)[M]\{\phi_i\} \tag{15.10–39}$$

Rearranging,

$$([K] - \lambda_o[M])\{\phi_i\} = \lambda_i[M]\{\phi_i\} \tag{15.10–40}$$

or

$$[K]' \{\phi_i\} = \lambda_i [M] \{\phi_i\} \qquad (15.10\text{–}41)$$

where: $[K]' = [K] - \lambda_o [M]$

It may be seen that if $[K]$ is singular, as in the case of rigid body motion, $[K]'$ will not be singular if $[M]$ is positive definite (which it normally is) and if λ_o is input as a negative number. A default shift of f_o = –1.0 is used for a modal analysis.

Once λ_i is computed, λ is computed from equation (15.10–37) and reported. Large shifts with the subspace iteration method are not recommended as they introduce some degradation of the convergence and this may affect accuracy of the final results.

15.10.8 Repeated Eigenvalues

Repeated roots or eigenvalues are possible to compute. This occurs, for example, for a thin, axisymmetric pole. Two independent sets of orthogonal motions are possible.

In these cases, the eigenvectors are not unique, as there are an infinite number of correct solutions. However, in the special case of two or more identical but disconnected structures run as one analysis, eigenvectors may include components from more than one structure. To reduce confusion in such cases, it is recommended to run a separate analysis for each structure.

15.11 Mode Superposition Method

Mode superposition method is a method of using the natural frequencies and mode shapes from the modal analysis (**ANTYPE**,MODAL) to characterize the dynamic response of a structure to transient (**ANTYPE**,TRANS with **TRNOPT**,MSUP, Section 17.2), or steady harmonic (**ANTYPE**,HARM with **HROPT**,MSUP, Section 17.4) excitations.

The equations of motion may be expressed as in equation (17.2–1):

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \qquad (15.11–1)$$

{F} is the time–varying load vector, given by

$$\{F\} = \left\{F^{nd}\right\} + s\left\{F^{s}\right\} \qquad (15.11–2)$$

where: $\left\{F^{nd}\right\}$ = time varying nodal forces

s = load vector scale factor (input on **LVSCALE** command)

$\left\{F^{s}\right\}$ = load vector from the modal analysis (see below)

The load vector $\left\{F^{s}\right\}$ is computed when doing a modal analysis and its generation is the same as for a substructure load vector, described in Section 17.6.

The following development is similar to that given by Bathe(2):

Define a set of modal coordinates y_i such that

$$\{u\} = \sum_{i=1}^{n} \{\phi_i\}\, y_i \qquad (15.11–3)$$

where: $\{\phi_i\}$ = the mode shape of mode i

n = the number of modes to be used (input on **TRNOPT** or **HROPT** commands)

Note that equation (15.11–3) hinders the use of nonzero displacement input, since defining y_i in terms of {u} is not straight forward. The inverse relationship does exist (equation (15.11–3)) for the case where all the displacements are known, but not when only some are known. Substituting equation (15.11–3) into (15.11–1),

$$[M]\sum_{i=1}^{n} \{\phi_i\}\,\ddot{y}_i + [C]\sum_{i=1}^{n} \{\phi_i\}\,\dot{y}_i + [K]\sum_{i=1}^{n} \{\phi_i\}\,y_i = \{F\} \qquad (15.11–4)$$

Premultiply by a typical mode shape $\left\{\phi_j\right\}^T$:

$$\left\{\phi_j\right\}^T [M] \sum_{i=1}^{n} \{\phi_i\} \ddot{y}_i + \left\{\phi_j\right\}^T [C] \sum_{i=1}^{n} \{\phi_i\} \dot{y}_i + \left\{\phi_j\right\}^T [K] \sum_{i=1}^{n} \{\phi_i\} y_i = \left\{\phi_j\right\}^T \{F\} \tag{15.11–5}$$

The orthogonality condition of the natural modes states that

$$\left\{\phi_j\right\}^T [M] \{\phi_i\} = 0 \quad i \neq j \tag{15.11–6}$$

$$\left\{\phi_j\right\}^T [K] \{\phi_i\} = 0 \quad i \neq j \tag{15.11–7}$$

In the mode superposition method, only Rayleigh or constant damping is allowed so that

$$\left\{\phi_j\right\}^T [C] \{\phi_i\} = 0 \quad i \neq j \tag{15.11–8}$$

Applying these conditions to equation (15.11–5), only the i=j terms remain:

$$\left\{\phi_j\right\}^T [M] \left\{\phi_j\right\} \ddot{y}_j + \left\{\phi_j\right\}^T [C] \left\{\phi_j\right\} \dot{y}_j + \left\{\phi_j\right\}^T [K] \left\{\phi_j\right\} y_j = \left\{\phi_j\right\}^T \{F\} \tag{15.11–9}$$

The coefficients of $\ddot{y}_j$, $\dot{y}_j$, and y_j, are derived as follows:

A. Coefficient of $\ddot{y}_j$:

By the normality condition (equation (17.3–6)),

$$\left\{\phi_j\right\}^T [M] \left\{\phi_j\right\} = 1 \tag{15.11–10}$$

B. Coefficient of $\dot{y}_j$:

The damping term is based on treating the modal coordinate as a single DOF system (shown in (15.11–1)) for which:

$$\left\{\phi_j\right\}^T [C] \left\{\phi_j\right\} = C_j \, \phi_j^2 \tag{15.11–11}$$

and

$$\left\{\phi_j\right\}^T [M] \left\{\phi_j\right\} = M_j \, \phi_j^2 = 1 \qquad (15.11–12)$$

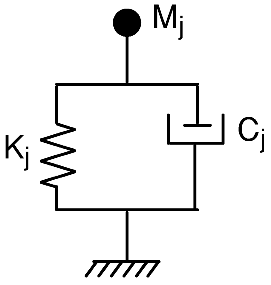


Figure 15.11–1 Single Degree of Freedom Oscillator

Equation (15.11–12) can give a definition of ϕ_j:

$$\phi_j = \frac{1}{\sqrt{M_j}} \qquad (15.11–13)$$

From (Tse(68)),

$$C_j = 2 \, \xi_j \sqrt{K_j \, M_j} \qquad (15.11–14)$$

where: ξ_j = fraction of critical damping for mode j

and,

$$\omega_j = \sqrt{\left(K_j / M_j\right)} \qquad (15.11–15)$$

where: ω_j = natural circular frequency of mode j

Combining equations (15.11–13) thru (15.11–15) with (15.11–11),

$$\begin{aligned}\left\{\phi_j\right\}^T [C] \left\{\phi_j\right\} &= 2 \, \xi_j \sqrt{K_j \, M_j} \left[\frac{1}{\sqrt{M_j}}\right]^2 \\ &= 2 \, \xi_j \, \omega_j\end{aligned} \qquad (15.11–16)$$

C. Coefficient of y_j:

From equation (17.3–3),

$$[K]\left\{\phi_j\right\} = \omega_j^2\,[M]\left\{\phi_j\right\} \qquad (15.11–17)$$

Premultiply by $\left\{\phi_j\right\}^T$,

$$\left\{\phi_j\right\}^T [K]\left\{\phi_j\right\} = \omega_j^2 \left\{\phi_j\right\}^T [M]\left\{\phi_j\right\} \qquad (15.11–18)$$

Substituting equation (15.11–10) for the mass term,

$$\left\{\phi_j\right\}^T [K]\left\{\phi_j\right\} = \omega_j^2 \qquad (15.11–19)$$

For convenient notation, let

$$f_j = \left\{\phi_j\right\}^T \{F\} \qquad (15.11–20)$$

represent the right–hand side of equation (15.11–9). Substituting equations (15.11–10), (15.11–16), (15.11–19) and (15.11–20) into (15.11–9), the equation of motion of the modal coordinates is obtained:

$$\ddot{y}_j + 2\omega_j\,\xi_j\,\dot{y}_j + \omega_j^2\,y_j = f_j \qquad (15.11–21)$$

Since j represents any mode, equation (15.11–21) represents n uncoupled equations in the n unknowns y_j. The advantage of the uncoupled system is that all the computationally expensive matrix algebra has been done in the eigensolver, and long transients may be analyzed inexpensively (**ANTYPE**,TRAN with **TRNOPT**,MSUP) in modal coordinates with equation (15.11–21). In harmonic analysis (**ANTYPE**,HARM with **HROPT**,MSUP), frequencies may be scanned faster than by the reduced harmonic response (**HROPT**,REDUC) method.

The y_j are converted back into geometric displacements {u} (the system response to the loading) by using equation (15.11–3). That is, the individual modal responses y_j are superimposed to obtain the actual response, and hence the name “mode superposition”.

If the modal analysis was performed using the reduced method (**MODOPT**,REDUC), then the matrices and load vectors in the above equations would be in terms of the master DOFs (i.e. $\{\hat{u}\}$).

15.11.1 Modal Damping

The modal damping, ξ_j, is the combination of several ANSYS damping inputs:

$$\xi_j = \left(\alpha/2\ \omega_j\right) + \left(\beta\omega_j/2\right) + \xi + \xi_{mj} \tag{15.11–22}$$

where:
α = uniform mass damping multiplier (input on **ALPHAD** command)
β = uniform stiffness damping multiplier (input on **BETAD** command)
ξ = constant damping ratio (input on **DMPRAT** command)
ξ_{mj} = modal damping ratio (input on **MDAMP** command)

Material–dependent damping may be used for ξ_{mj} if it is defined in the modal analysis; see equation (17.7–1). Because of the assumption in equation (15.11–8), explicit damping in such elements as COMBIN14 is not allowed by the mode superposition procedure.

15.12 Solving for Unknowns and Reactions

In general, the equations that are solved for static linear analyses are:

$$[K]\{u\} = \{F\} \tag{15.12–1}$$

or

$$[K]\{u\} = \{F^a\} + \{F^r\} \tag{15.12–2}$$

where: $[K]$ = total stiffness or conductivity matrix = $\sum_{m=1}^{N} [K_e]$

$\{u\}$ = nodal degree of freedom (DOF) vector

N = number of elements

$[K_e]$ = element stiffness or conductivity matrix

$\{F^r\}$ = nodal reaction load vector

$\{F^a\}$, the total applied load vector, is defined by:

$$\{F^a\} = \{F^{nd}\} + \{F^e\} \tag{15.12–3}$$

where: $\{F^{nd}\}$ = applied nodal load vector

$\{F^e\}$ = total of all element load vector effects (pressure, acceleration, thermal, gravity)

Equations (15.12–1) thru (15.12–3) are similar to equations (17.1–1) thru (17.1–3).

If sufficient boundary conditions are specified on {u} to guarantee a unique solution, equation (15.12–1) can be solved to obtain the node DOF values at each node in the model.

Rewriting equation (15.12–2) for linear analyses by separating out the matrix and vectors into those DOFs with and without imposed values,

$$\begin{bmatrix} [K_{cc}] & [K_{cs}] \\ [K_{cs}]^T & [K_{ss}] \end{bmatrix} \begin{Bmatrix} \{u_c\} \\ \{u_s\} \end{Bmatrix} = \begin{Bmatrix} \{F_c^a\} \\ \{F_s^a\} \end{Bmatrix} + \begin{Bmatrix} \{F_c^r\} \\ \{F_s^r\} \end{Bmatrix} \tag{15.12–4}$$

where: s = subscript representing DOFs with imposed values (<u>s</u>pecified DOFs) c = subscript representing DOFs without imposed values (computed DOFs)

Note that $\{u_s\}$ is known, but not necessarily equal to {0}. Since the reactions at DOFs without imposed values must be zero, equation (15.12–4) can be written as:

$$\begin{bmatrix} [K_{cc}] & [K_{cs}] \\ [K_{cs}]^T & [K_{ss}] \end{bmatrix} \begin{Bmatrix} \{u_c\} \\ \{u_s\} \end{Bmatrix} = \begin{Bmatrix} \{F_c^a\} \\ \{F_s^a\} \end{Bmatrix} + \begin{Bmatrix} \{0\} \\ \{F_s^r\} \end{Bmatrix} \tag{15.12–5}$$

The top part of equation (15.12–5) may be solved for $\{u_c\}$:

$$\{u_c\} = [K_{cc}]^{-1}\left(-[K_{cs}]\{u_s\} + \{F_c^a\}\right) \tag{15.12–6}$$

The actual numerical solution process is not as indicated here but is done more efficiently using the wavefront equation solver, discussed in Section 15.7.

15.12.1 Reaction Forces

The reaction vector $\{F_s^r\}$, may be developed for linear models from the bottom part of equation (15.12–5):

$$\{F_s^r\} = [K_{cs}]^T\{u_c\} + [K_{ss}]\{u_s\} - \{F_s^a\} \tag{15.12–7}$$

where: $\{F_s^r\}$ = output using either **OUTPR**,RSOL or **PRRSOL** command

Alternatively, the nodal reaction load vector may be considered over all DOFs by combining equation (15.12–2) and (15.12–3) to get:

$$\{F^r\} = [K]\{u\} - \{F^{nd}\} - \{F^e\} \tag{15.12–8}$$

where only the loads at imposed DOF are output. Where applicable, the transient/dynamic effects are added:

$$\{F^r\} = [M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} - \{F^{nd}\} - \{F^e\} \tag{15.12–9}$$

where:
[M] = total mass matrix
[C] = total damping or conductivity matrix
$\{\ddot{u}\}$, $\{\dot{u}\}$ = defined below

The element static nodal loads are:

$$\{F_e^k\} = -[K_e]\{u_e\} + \{F_e^e\} \tag{15.12–10}$$

where: $\{F_e^k\}$ = element nodal loads (output using **OUTPR**, **NLOAD**, or **PRESOL** commands)

e = subscript for element matrices and load vectors

The element damping and inertial loads are:

$$\{F_e^D\} = -[C_e]\{\dot{u}\} \tag{15.12–11}$$

$$\{F_e^I\} = [M_e]\{\ddot{u}\} \tag{15.12–12}$$

where: $\{F_e^D\}$ = element damping nodal load (output using **OUTPR**, **NLOAD**, or **PRESOL** commands)

$\{F_e^I\}$ = element inertial nodal load (output using **OUTPR**, **NLOAD**, or **PRESOL** commands)

Thus,

$$\{F^r\} = -\sum_{m=1}^{N}\left(\{F_e^K\} + \{F_e^D\} + \{F_e^I\}\right) - \{F^{nd}\} \tag{15.12–13}$$

The derivatives of the nodal DOF with respect to time are:

$\{\dot{u}\}$ = first derivative of the nodal DOF with respect to time, e.g. velocity

$\{\ddot{u}\}$ = second derivative of the nodal DOF with respect to time, e.g. acceleration

Sections 17.2 and 17.4 discuss the transient and harmonic damping and inertia loads.

15.12.2 Disequilibrium

The following circumstances could cause a disequilibrium, usually a moment disequilibrium:

Program Option	Explanation of Possible Difficulty
non–planar, 4–noded membrane shell elements SHELL41 SHELL63 with KEYOPT(1)=1	If the four nodes do not lie in a flat plane moment equilibrium may not be preserved, as no internal corrections are done. However, the program requires such elements to be input very close to flat.
nodal coupling constraint equations (**CP**, **CE** commands)	The user can write any form of relationship between the displacements, and these may include fictitious forces or moments. Thus, the reaction forces printout can be used to detect input errors.

Program Option	Explanation of Possible Difficulty
MATRIX27 User generated super–element matrix	The user has the option to input almost any type of erroneous input, so that such input should be checked carefully. For example, all terms representing UX degrees of freedom of one UX row of the matrix should sum to zero to preserve equilibrium.
COMBIN7, CONTAC12 COMBIN37, FLUID38 COMBIN39 and COMBIN40	Noncoincident nodes can cause a moment disequilibrium. (This is usually not a problem if one of the nodes is attached to a non–rotating ground).
COMBIN14 (with KEYOPT(2) > 0) COMBIN27, COMBIN37, FLUID38, COMBIN39 and COMBIN40	Elements with one node having a different nodal coordinate system from the other are inconsistent.

The following circumstances could cause an apparent disequilibrium:

1. All nodal coordinate systems are not parallel to the global Cartesian coordinate system. However, if all nodal forces are rotated to the global Cartesian coordinate system, equilibrium should be seen to be satisfied.
2. The solution is not converged. This applies to the potential discrepancy between applied and internal element forces in a nonlinear analysis.
3. The mesh is too coarse. This may manifest itself for elements where there is an element force printout at the nodes, such as SHELL61 (axisymmetric–harmonic structural shell).
4. Stress stiffening only (**SSTIF**,ON), (discussed in Section 3.3) is used. Note that moment equilibrium seems not to be preserved in equation (3.3–12). However, if the implicit updating of the coordinates is also considered (**NLGEOM**,ON), equilibrium will be seen to be preserved.
5. DOFs that are restrained by constraint equations have specified displacements. In this case, the reported reaction force will be repeated in both the imposed displacement reaction table and the constraint equation reaction table. For equilibrium checks, only one of the numbers should be used. See Section 15.8 for the constraint equation description.
6. The “TOTAL” of the moments (MX, MY, MZ) given with the reaction forces does not necessarily represent equilibrium. It only represents the sum of all applicable moments. Moment equilibrium would also need the effects of forces taken about an arbitrary point.
7. Axisymmetric models are used with forces or pressures with a radial component. These loads will often be partially equilibrated by hoop stresses, which do not show up in the reaction forces.

8. Shell elements have an elastic foundation described. The load carried by the elastic foundation is not seen in the reaction forces.

9. CONTAC26 elements are used. The loads transmitted to the rigid surfaces are not seen in the reaction forces.

15.13 Conjugate Gradient Solvers

The ANSYS program offers three conjugate gradient (CG) iterative solvers as alternatives to the frontal solver (default). The first solver (**EQSLV**,JCG) is the Jacobi Conjugate Gradient (JCG) solver (Mahinthakumar and Hoole(144)) which is suitable for well–conditioned problems. Well–conditioned problems often arise from heat transfer, acoustics, magnetics and solid 2–D /3–D structural analyses. The JCG solver is available for real and complex matrices as well as symmetric and unsymmetric matrices. The second solver (**EQSLV**,PCG) is the Preconditioned Conjugate Gradient (PCG) solver which is efficient and reliable for all types of analyses including the ill–conditioned beam/shell structural analysis. The PCG solver is made available through a license from Computational Applications and System Integration, Inc. of Champaign, Illinois (USA). The PCG solver is only valid for real symmetric and positive definite stiffness matrices. The third solver (**EQSLV**,ICCG) is the Incomplete Cholesky Conjugate Gradient (ICCG) solver (internally developed, unpublished work). The ICCG solver is more robust than the JCG solver for handling ill–conditioned matrices. The ICCG solver is available for real and complex matrices as well as symmetric and unsymmetric matrices.

The typical system of equations to be solved iteratively is given as :

$$[K]\ \{u\} = \{F\} \tag{15.13–1}$$

where:

[K] = global coefficient matrix

{u} = unknown vector

{F} = global load vector

In the CG method, the solution is found as a series of vectors $\{p_i\}$:

$$\{u\} = \alpha_1\ \{p_1\} + \alpha_2\ \{p_2\} + \ \cdots\ + \alpha_m\ \{p_m\} \tag{15.13–2}$$

where m is no larger than the matrix size n. The scheme is guaranteed to converge in n or fewer iterations on an infinite precision machine. However, since the scheme is implemented on a machine with finite precision, it sometimes requires more than n iterations to converge. The solvers allow up to a maximum of 2n iterations. If it still does not converge after the 2n iterations, the solution will be abandoned with an error message. The unconverged situation is often due to an inadequate number of boundary constraints being used (rigid body motion). The rate of convergence of the CG algorithm is proportional to the square root of the conditioning number of [K] where the condition number of [K] is equal to the ratio of the maximum eigenvalue of [K] to the minimum eigenvalue of [K] . A preconditioning procedure is used to reduce the condition number of linear equations (15.13–1). In the JCG algorithm, the diagonal terms of [K] are used as the preconditioner [Q], while in the ICCG and PCG algorithms, a more sophisticated preconditioner [Q] is used. The CG algorithm with preconditioning is shown in Figure 15.13–1. Convergence is achieved when:

$$\frac{\{R_i\}^T \{R_i\}}{\{F\}^T\{F\}} \leq \epsilon^2 \tag{15.13–3}$$

where: ε = user supplied tolerance (*TOLER* on the **EQSLV** command; output quantity SPECIFIED TOLERANCE)

$\{R_i\}$ = $\{F\} - [K] \{u_i\}$

$\{u_i\}$ = solution vector at iteration i

also, for the JCG and ICCG solvers:

$$\{R_i\}^T \{R_i\} = \text{output quantity CALCULATED NORM} \tag{15.13–4}$$

$$\{F\}^T \{F\} \epsilon^2 = \text{output quantity TARGET NORM} \tag{15.13–5}$$

It is assumed that the initial starting vector $\{u_0\}$ is a zero vector. All three conjugate gradient solvers require less disk space than the frontal solver, as File.TRI used to store the factorized matrix used by the frontal solver is not needed. However, the in–core option of the CG solver requires larger memory as the global sparse stiffness matrix $[K]$ is stored in memory. The ICCG and PCG solvers require twice as much memory as the JCG solver because the preconditioner which is also stored in memory is similar in size to the $[K]$ matrix . An out–of–memory option of JCG solver is available; however, the I/O operations in the out–of–memory process require considerable elapse time.

The PCG solver is faster than the JCG solver. The ICCG solver may be faster than the JCG solver. All three solvers run faster than the frontal solvers for large wavefront problems. All solvers are valid with both constraint equations (**CE** commands) and coupling (**CP** commands).

```
$\{u_0\} = \{0\}$
$\{R_0\} = \{F\}$
$\{z_0\} = [Q]^{-1} \{F\}$
Do i = 1, n
    If (Norm(R) $\leq \varepsilon^2$) then
        set $\{u\} = \{u_{i-1}\}$
        quit loop
    Else
        If (i = 1) then
            $\beta_1 = 0$
            $\{p_1\} = \{R_0\}$
            $\alpha_1 = \frac{\{z_0\}^T \{R_0\}}{\{p_1\}^T [K] \{p_1\}}$
            $\{R_1\} = \{R_0\} - \alpha_1 [K] \{p_1\}$
            $\{u_1\} = \{u_0\} + \alpha_1 \{p_1\}$
        Else
            Applying preconditioning: $\{z_{i-1}\} = [Q]^{-1} \{R_{i-1}\}$
            $\beta_i = \frac{\{z_{i-1}\}^T \{R_{i-1}\}}{\{z_{i-2}\}^T \{R_{i-2}\}}$
            $\{p_i\} = \{z_{i-1}\} + \beta_i \{p_{i-1}\}$
            $\alpha_i = \frac{\{z_{i-1}\}^T \{R_{i-1}\}}{\{p_i\}^T [K] \{p_i\}}$
            $\{R_i\} = \{R_{i-1}\} - \alpha_i [K] \{p_i\}$
            $\{u_i\} = \{u_{i-1}\} + \alpha_i \{p_i\}$
        Endif
    Endif
End loop
```

Figure 15.13–1 The CG Algorithm with Preconditioning

15.14 Modal Analysis of Cyclic Symmetric Structures

Given a cyclic symmetric (periodic) structure such as a fan wheel, a modal analysis can be performed for the entire structure by modelling only one sector of it. A proper basic sector represents a pattern that, if repeated n times in cylindrical coordinate space (CSYS=1), would yield the complete structure as shown in Figure 15.14–1.

Also see Section 15.12 for other possible anomalies.

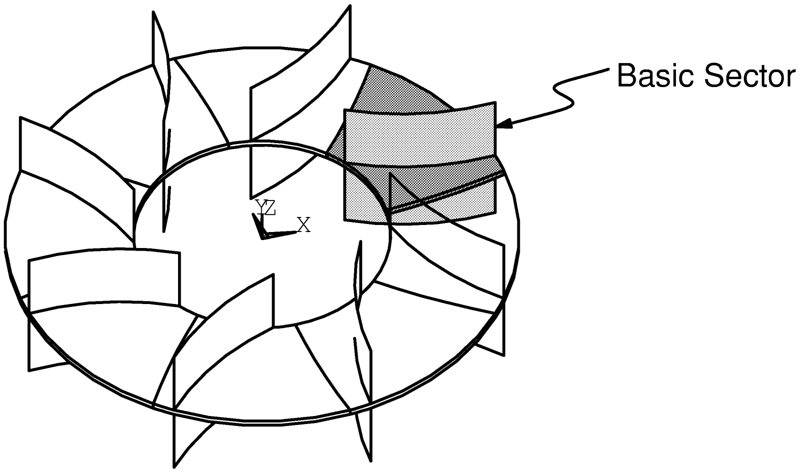


Figure 15.14–1 Typical Cyclic Symmetric Structure

In a flat circular membrane, mode shapes are identified by nodal diameters. In this context, a nodal diameter is defined as a location, θ, in cylindrical coordinate space, which has zero displacement. For more information, see Chapter 3 of the *ANSYS Structural Analysis Guide*.

Constraint relationships (equations) can be defined to relate the lower and higher angle edges of the basic sector to allow calculation of natural frequencies related to a given number of nodal diameters. The basic sector, as shown in Figure 15.14–2, is used twice in the modal analysis to satisfy the required constraint relationships and to obtain nodal displacements. This technique was adapted from Adapco(148).

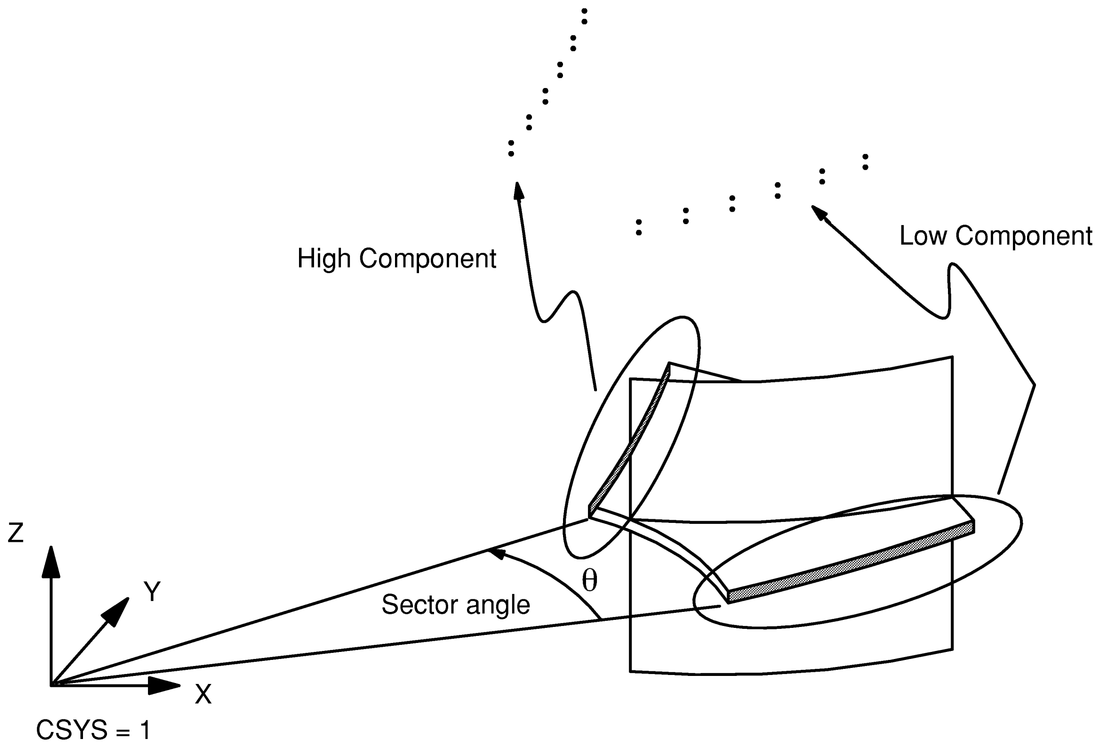


Figure 15.14–2 Basic Sector Definition

Constraint equations relating the lower and higher angle edges of the two sectors are written:

$$\begin{Bmatrix} u'_A \\ u'_B \end{Bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{Bmatrix} u_A \\ u_B \end{Bmatrix} \qquad (15.14\text{–}1)$$

where: u_A, u_B = calculated displacements on higher angle side of basic sectors

$u'_A,\ u'_B$ = displacements on lower angle side of basic sectors determined from constraint relationships

α = $N_D * \theta_N$

N_D = number of nodal diameters

θ_N = basic sector angle of starting model

Three steps of the procedure are briefly described:

1. The macro command **CYCGEN** generates the additional sector.

2. The macro command **CYCSOL** initiates the modal cyclic symmetry analysis, for which the user provides a nodal diameter range (NDMIN and NDMAX), the name of the component comprising the set of nodes at the lower angle (the name of the component comprising the set of nodes at the higher angle is optional), and the number of sectors composing the entire structure. This command creates the constraint relationships and solver for eigenvalues and eigenvectors for each nodal derivation.
3. In postprocessing, the user issues the **EXPAND** or /**EXPAND** command to expand the model for display and indicates the number of sectors that are to be expanded.

15.14.1 Complete Mode Shape Derivation

The mode shape in each sector is determined from the modal displacements. The displacement component (x, y, or z) at any point, i, in the full structure is:

$$u_i = A \cos (\alpha (I - 1) + \phi) \tag{15.14–2}$$

where:

I = sector number

$A = \dfrac{u_A}{\cos\phi}$

$\phi = \tan^{-1}\left(\dfrac{u_{B,i}}{u_{A,i}}\right)$

$u_{A,i}\ ,\ u_{B,i}$ = modal analysis component displacement values at point i in sectors A, B

The complete procedure addressing both standard and prestressed modal analyses of cyclic symmetric structures is contained in Chapter 3 of the *ANSYS Structural Analysis Guide*.

15.15 Mass Moments of Inertia

The computation of the mass moments and products of inertia, as well as the model centroids, is described in this section. The model centroids are computed as:

$$X_c = \frac{A_x}{M} \qquad (15.15\text{–}1)$$

$$Y_c = \frac{A_y}{M} \qquad (15.15\text{–}2)$$

$$Z_c = \frac{A_z}{M} \qquad (15.15\text{–}3)$$

where typical terms are:

X_c = X coordinate of model centroid (output quantity XC)

$A_x = \sum_{i=1}^{N} m_i X_i$

N = number of elements

m_i = mass of element i

$= \begin{cases} \text{function of real constants, if applicable} \\ \text{or} \\ \varrho V_i \end{cases}$

ϱ = element density, based on average element temperature

V_i = volume of element i

X_i = X coordinate of the centroid of element i

$= \sum_{j=1}^{N_i^n} X_{ij} / N_i^n$

N_i^n = number of nodes of element i

X_{ij} = x coordinate of node J of element i

M $= \sum_{i=1}^{N} m_i$ = mass of model (output quantity TOTAL MASS)

The moments and products of inertia with respect to the origin are:

$$I_{xx} = \sum_{i=1}^{N} m_i\left(\left(Y_i\right)^2 + \left(Z_i\right)^2\right) \tag{15.15–4}$$

$$I_{yy} = \sum_{i=1}^{N} m_i\left(\left(X_i\right)^2 + \left(Z_i\right)^2\right) \tag{15.15–5}$$

$$I_{zz} = \sum_{i=1}^{N} m_i\left(\left(X_i\right)^2 + \left(Y_i\right)^2\right) \tag{15.15–6}$$

$$I_{xy} = -\sum_{i=1}^{N} m_i\left(\left(X_i\right)\left(Y_i\right)\right) \tag{15.15–7}$$

$$I_{yz} = -\sum_{i=1}^{N} m_i\left(\left(Y_i\right)\left(Z_i\right)\right) \tag{15.15–8}$$

$$I_{xz} = -\sum_{i=1}^{N} m_i\left(\left(X_i\right)\left(Z_i\right)\right) \tag{15.15–9}$$

where typical terms are:

I_{xx} = mass moment of inertia about the X axis through the model centroid (output quantity IXX)

I_{xy} = mass product of inertia with respect to the X and Y axes through the model centroid (output quantity IXY)

The moments and products of inertia with respect to the model centroid (the components of the inertia tensor) are:

$$I'_{xx} = I_{xx} - M\left((Y_c)^2 + (Z_c)^2\right) \tag{15.15–10}$$

$$I'_{yy} = I_{yy} - M\left((X_c)^2 + (Z_c)^2\right) \tag{15.15–11}$$

$$I'_{zz} = I_{zz} - M\left((X_c)^2 + (Y_c)^2\right) \tag{15.15–12}$$

$$I'_{xy} = I_{xy} + M\ X_c\ Y_c \tag{15.15–13}$$

$$I'_{yz} = I_{yz} + M\ Y_c\ Z_c \tag{15.15–14}$$

$$I'_{xz} = I_{xz} + M\ X_c\ Z_c \tag{15.15–15}$$

where typical terms are:

I'_{xx} = mass moment of inertia about the X axis through the model centroid (output quantity IXX)

I'_{xy} = mass product of inertia with respect to the X and Y axes through the model centroid (output quantity IXY)

Accuracy of the Calculations

The above calculations are not intended to be precise for all situations, but rather have been programmed for speed. It may be seen from the above development that only the mass (m_i) and the centroid (X_i, Y_i, and Z_i) of each element are included. Effects that are not considered are:

1. The mass being different in different directions.
2. The presence of rotational inertia terms.

These can occur, for example, with MASS21. Another approximation is associated with the definition of X_i. It is simply the average of all attached nodes. It is not precise for trapezoidal shaped elements as well as beam or shell elements with offsets.

Thus, if these effects are important, a separate analysis can be performed using inertia relief (**IRLF,–1)** to find more precise centroids and moments of inertia. Inertia relief logic uses the element mass matrices directly; however, its centroid calculations also do not include the effects of offsets.

It should be emphasized that the computations for displacements, stresses, reactions, etc. are correct with none of the above approximations.

Effect of KSUM, LSUM, ASUM, and VSUM Commands

The centroid and mass moment of inertia calculations for keypoints, lines, areas, and volumes (PREP7 **KSUM**, **LSUM**, **ASUM**, **VSUM**, and ***GET** commands) use equations similar to (15.15–1) through (15.15–15) with the following changes:

1. Only selected solid model entities are included.
2. Lines (**LSUM**), areas (**ASUM**), and volumes (**VSUM**) are approximated by numerically integrating to account for rotary inertias.
3. Keypoints (**KSUM**) are assumed to be unit masses without rotary inertia.
4. Lines (**LSUM**) are assumed to have unit mass per unit length.

5. Each area (**ASUM**) uses the thickness as:

$$t = \begin{cases} \text{first real constant in the table assigned to the} \\ \text{area by the } \textbf{AATT} \text{ or } \textbf{AMESH} \text{ command} \\ \text{1.0 if there is no such assignment or real constant table} \end{cases} \tag{15.15–16}$$

where: t = thickness

6. Each area (**ASUM**) or volume (**VSUM**) is assumed to have density as:

$$\rho = \begin{cases} \text{input density (DENS) for the material assigned} \\ \text{to the area of volume by the } \textbf{AATT/VATT} \text{ or} \\ \textbf{AMESH/VMESH} \text{ command} \\ \text{1.0 if there is no such assignment or material property} \end{cases} \tag{15.15–17}$$

where: ϱ = density

Composite material elements presume the material number defined with the **MAT** command.

15.16 Energies

Energies are available by setting Item = VENG on the **OUTPR** command or items SENE, TENE, KENE, and AENE on the **ETABLE** command. For each element,

$$E_e^{po} = \begin{cases} \frac{1}{2} \sum_{i=1}^{NINT} \{\sigma\}^T \left\{\epsilon^{el}\right\} \text{vol}_i + E_e^{pl} & \text{if element allows only displacement and rotational degree of freedom (DOF), either is nonlinear or uses integration points, and is not a p–element.} \\ \frac{1}{2} \{u\}^T \left([K_e] + [S_e]\right) \{u\} & \text{all other cases} \end{cases} \quad (15.16\text{–}1)$$

= potential energy (strain energy) (accessed with SENE or TENE on the **ETABLE** command)

$$E_e^{ki} = \frac{1}{2} \{\dot{u}\}^T [M_e] \{\dot{u}\} \quad (15.16\text{–}2)$$

= kinetic energy (accessed with KENE on the **ETABLE** command) (computed only for transient and modal analyses)

$$E_e^{art} = \sum_{j=1}^{NCS} \frac{1}{2} \{\gamma\}^t [Q] \{\gamma\} \quad (15.16\text{–}3)$$

= artificial energy (accessed with AENE on the **ETABLE** command) (SOLID45, SOLID182, SOLID185, SHELL181 only)

where:

NINT	=	number of integration points
$\{\sigma\}$	=	stress vector
$\{\varepsilon^{el}\}$	=	elastic strain vector
vol_i	=	volume of integration point ~~is~~
E_e^{pl}	=	plastic strain energy
$[K_e]$	=	element stiffness/conductivity matrix
$[S_e]$	=	element stress stiffness matrix
$\{u\}$	=	element DOF vector
$\{\dot{u}\}$	=	time derivative of element DOF vector
$[M_e]$	=	element mass matrix
NCS	=	total number of converged substeps

$\{\gamma\}$ = hourglass strain energy defined in Flanagan and Belytschko(242) due to one point integrations.

$[Q]$ = hourglass control stiffness defined in Flanagan and Belytschko(242).

As may be seen from the bottom part of equation (15.16–1) as well as equation (15.16–2), all types of DOFs are combined, e.g., SOLID5 using both UX, UY, UZ, TEMP, VOLT, and MAG DOF. An exception to this is the piezoelectric elements, described in Section 11.1, which do report energies by separate types of DOFs in the NMISC record of element results. See Section 15.10 when complex frequencies are used. Also, if the bottom part of equation (15.16–1) is used, any nonlinearities are ignored. Elements with other incomplete aspects with respect to energy are reported in Table 15.16–1.

Artificial energy has no physical meaning. It is used to control the hourglass mode introduced by reduced integration. The rule–of–thumb to check if the element is stable or not due to the use of reduced integration is if $\frac{\text{AENE}}{\text{SENE}} < 5\%$ is true.When this inequalilty is true, the element using reduced integration is considered stable (i.e., functions the same way as fully integrated element).

Table 15.16–1 Exceptions for Element Energies

	Element	Exceptions
1.	BEAM4	Warping* thermal gradient not included
2.	PIPE16	Thru–wall thermal gradient not included
3.	PIPE17	Thru–wall thermal gradient not included
4.	PIPE18	Thru–wall thermal gradient not included
5.	FLUID29	No potential energy
6.	FLUID30	No potential energy
7.	LINK31	No potential energy
8.	LINK34	No potential energy
9.	COMBIN39	No potential energy
10.	SHELL41	Foundation stiffness effects are not included
11.	BEAM44	Warping* thermal gradient not included
12.	PIPE59	Thru–wall thermal gradient not included
13.	PIPE60	Nonlinear and thermal effects are not included
14.	SHELL61	Thermal effects not included
15.	SHELL63	Foundation stiffness effects not included
16.	SHELL99	Foundation stiffness effects not included
17.	FLUID141	No potential energy
18.	FLUID142	No potential energy
19.	PLANE145	Thermal effects not included
20.	PLANE146	Thermal effects not included
21.	PLANE147	Thermal effects not included
22.	PLANE148	Thermal effects not included
23.	PLANE150	Thermal effects not included

* Warping implies for example that temperatures T1 + T3 $\neq$ T2 + T4, i.e. some thermal strain is locked in.

A discussion of error energy is given in Section 19.7.

For VISCO106, VISCO107, and VISCO108, a plastic energy per unit volume is also available. See Section 14.107.

Chapter 16

ANSYS Theory Reference
Chapter 16 – Table of Contents

Chapter 17
Analysis Procedures

ANSYS Theory Reference
Chapter 17 – Table of Contents

17.0 Analysis Procedures

This chapter of the manual is designed to give users an understanding of the theoretical basis of the overall analysis procedures. The derivation of the individual element matrices and load vectors is discussed in Sections 2.2, 5.2, 6.2, 7.2, and 8.2.

In the matrix displacement method of analysis based upon finite element idealization, the structure being analyzed must be approximated as an assembly of discrete regions (called elements) connected at a finite number of points (called nodes). If the “force–displacement” relationship for each of these discrete structural elements is known (the element “stiffness” matrix) then the “force–displacement relationship” for the entire “structure” can be assembled using standard matrix methods. These methods are well documented (see, for example, Zienkiewicz(39)) and are also discussed in Chapter 15. Thermal, fluid flow, and electro–magnetic analyses are done on an analogous basis by replacing the above words in quotes with the appropriate terms. However, the terms displacement, force, and stiffness will be frequently used throughout this chapter, even though it is understood that the concepts apply to all valid effects also.

All analysis types for iterative or transient problems automatically reuse the element matrices or the overall structural matrix whenever it is applicable. See Section 13.3 for more details.

17.1 Static Analysis (ANTYPE,STATIC)

17.1.1 Assumptions and Restrictions

It is valid for all degrees of freedom (DOFs). Inertial and damping effects are ignored, except for static acceleration fields.

17.1.2 Description of Structural Systems

The overall equilibrium equations for linear structural static analysis are:

$$[K]\{u\} = \{F\} \tag{17.1–1}$$

or

$$[K]\{u\} = \{F^a\} + \{F^r\} \tag{17.1–2}$$

where:
[K] = total stiffness matrix = $\sum_{m=1}^{N} [K_e]$
{u} = nodal displacement vector
N = number of elements
$[K_e]$ = element stiffness matrix (described in Chapter 14) (may include the element stress stiffness matrix (described in Section 3.3))
$\{F^r\}$ = reaction load vector

$\{F^a\}$, the total applied load vector, is defined by:

$$\{F^a\} = \{F^{nd}\} + \{F^{ac}\} + \sum_{m=1}^{N} \left(\{F_e^{th}\} + \{F_e^{pr}\}\right) \tag{17.1–3}$$

where:
$\{F^{nd}\}$ = applied nodal load vector
$\{F^{ac}\}$ = –[M]$\{a_c\}$ = acceleration load vector

$[M]$ = total mass matrix = $\sum_{m=1}^{N} [M_e]$

$[M_e]$ = element mass matrix (described in Section 2.2)

$\{a_c\}$ = total acceleration vector (defined in Section 15.1)

$\left\{F_e^{th}\right\}$ = element thermal load vector (described in Section 2.2)

$\left\{F_e^{pr}\right\}$ = element pressure load vector (described in Section 2.2)

To illustrate the load vectors in equation (17.1–2), consider a one element column model, loaded only by its own weight, as shown in Figure 17.1–1. Note that the lower applied gravity load is applied directly to the imposed displacement, and therefore causes no strain; nevertheless, it contributes to the reaction load vector just as much as the upper applied gravity load.

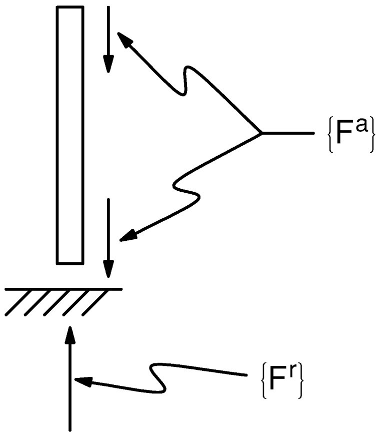


Figure 17.1–1 Applied and Reaction Load Vectors

Warning: If the stiffness for a certain DOF is zero, any applied loads on that DOF are ignored.

Section 15.12 discusses the solution of equation (17.1–2) and the computation of the reaction loads. Section 15.9 describes the global equation for a nonlinear analysis. Inertia relief is discussed in Section 15.2.

17.1.3 Description of Thermal, Magnetic and Other First Order Systems

The overall equations for linear 1st order systems are the same as for a linear structural static analysis, equations (17.1–1) and (17.1–2). [K], though, is the total coefficient matrix (e.g. the conductivity matrix in a thermal analysis) and {u} is the nodal DOF values. $\{F^a\}$, the total applied load vector, is defined by:

$$\{Q^a\} = \{Q^{nd}\} + \sum_{m=1}^{N} \{Q_e\} \qquad (17.1\text{–}4)$$

Table 17.1–1 relates the nomenclature used in Sections 6.2 and 5.2 for thermal, magnetic and electrical analyses to equation (17.1–2) and (17.1–4). See Table 11.0–1 for a more detailed nomenclature description.

Table 17.1–1 Nomenclature

	{u}	**$\{F^{nd}\}$**	**$\{F_e\}$**
Thermal	{T} temperatures	$\{Q^{nd}\}$ heat flow	$\{Q_e\}+\ \{Q_e^g\}\ +\ \{Q_e^c\}$ heat flux heat generation convection
Scalar Magnetic	{φ} scalar potential	$\{F^{nd}\}$ flux	$\{F_e\}$ coercive force
Vector Magnetic	{A} vector potential	$\{F^{nd}\}$ current segment	$\{F_e\}$ current density and coercive force
Electrical	{V} voltage	$\{I^{nd}\}$ current	–

Section 15.12 discusses the solution of equation (17.1–2) and Section 15.9 describes the global equation for a nonlinear analysis.

17.2 Transient Analysis (ANTYPE,TRANS)

The transient analysis solution method used depends on the DOFs involved. Structural, acoustic, and other second order systems (that is, the systems are second order in time) are solved using one method and the thermal, magnetic, electrical and other first order systems are solved using another. Each method is described subsequently. If the analysis contains both first and second order DOFs (e.g. structural and magnetic), then each DOF is solved using the appropriate method. For matrix coupling between first and second order effects such as for piezoelectric analysis, a combined procedure is used.

17.2.1 Assumptions and Restrictions

1. Initial conditions are known.
2. No gyroscopic or Coriolis effects are included in a structural analysis (except for the gyroscopic damping in BEAM4 and PIPE16).

17.2.2 Description of Structural and Other Second Order Systems

The transient dynamic equilibrium equation of interest is as follows for a linear structure:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F^a\} \qquad (17.2\text{–}1)$$

where:
$[M]$ = structural mass matrix
$[C]$ = structural damping matrix
$[K]$ = structural stiffness matrix
$\{\ddot{u}\}$ = nodal acceleration vector
$\{\dot{u}\}$ = nodal velocity vector
$\{u\}$ = nodal displacement vector
$\{F^a\}$ = applied load vector

There are two methods in the ANSYS program which can be employed for the solution of the linear equation (17.2–1): the forward difference time integration method and the Newmark time integration method. The forward difference method is used for explicit transient analyses and is described in the ANSYS/LS–DYNA3D Theoretical

Manual(199). The Newmark method is used for implicit transient analyses and is described below.

The Newmark method uses finite difference expansions in the time interval Δt, in which it is assumed that (Bathe(2)):

$$\{\dot{u}_{n+1}\} = \{\dot{u}_n\} + \left[(1-\delta)\{\ddot{u}_n\} + \delta\{\ddot{u}_{n+1}\}\right]\Delta t \tag{17.2–2}$$

$$\{u_{n+1}\} = \{u_n\} + \{\dot{u}_n\}\Delta t + \left[\left(\frac{1}{2}-\alpha\right)\{\ddot{u}_n\} + \alpha\{\ddot{u}_{n+1}\}\right]\Delta t^2 \tag{17.2–3}$$

where:

α, δ = Newmark integration parameters

Δt = $t_{n+1} - t_n$

$\{u_n\}$ = nodal displacement vector at time t_n

$\{\dot{u}_n\}$ = nodal velocity vector at time t_n

$\{\ddot{u}_n\}$ = nodal acceleration vector at time t_n

$\{u_{n+1}\}$ = nodal displacement vector at time t_{n+1}

$\{\dot{u}_{n+1}\}$ = nodal velocity vector at time t_{n+1}

$\{\ddot{u}_{n+1}\}$ = nodal acceleration vector at time t_{n+1}

Since the primary aim is the computation of displacements $\{u_{n+1}\}$, the governing equation (17.2–1) is evaluated at time t_{n+1} as:

$$[M]\{\ddot{u}_{n+1}\} + [C]\{\dot{u}_{n+1}\} + [K]\{u_{n+1}\} = \{F^a\} \tag{17.2–4}$$

The solution for the displacement at time t_{n+1} is obtained by first rearranging equations (17.2–2) and (17.2–3), such that:

$$\{\ddot{u}_{n+1}\} = a_0\left(\{u_{n+1}\} - \{u_n\}\right) - a_2\{\dot{u}_n\} - a_3\{\ddot{u}_n\} \tag{17.2–5}$$

$$\{\dot{u}_{n+1}\} = \{\dot{u}_n\} + a_6\{\ddot{u}_n\} + a_7\{\ddot{u}_{n+1}\} \tag{17.2–6}$$

where:

$$a_0 = \frac{1}{\alpha\Delta t^2} \qquad a_4 = \frac{\delta}{\alpha} - 1$$

$$a_1 = \frac{\delta}{\alpha\Delta t} \qquad a_5 = \frac{\Delta t}{2}\left(\frac{\delta}{\alpha} - 2\right)$$

$$a_2 = \frac{1}{\alpha\Delta t} \qquad a_6 = \Delta t(1-\delta)$$

$$a_3 = \frac{1}{2\alpha} - 1 \qquad a_7 = \delta\Delta t$$

Noting that $\{\ddot{u}_{n+1}\}$ in equation (17.2–5) can be substituted into equation (17.2–6), equations for $\{\ddot{u}_{n+1}\}$ and $\{\dot{u}_{n+1}\}$ can be expressed only in terms of the unknown $\{u_{n+1}\}$. The equations for $\{\ddot{u}_{n+1}\}$ and $\{\dot{u}_{n+1}\}$ are then combined with equation (17.2–4) to form:

$$\begin{aligned}&\left(a_0[M] + a_1[C] + [K]\right)\{u_{n+1}\} = \{F^a\} + \\ &[M]\left(a_0\{u_n\} + a_2\{\dot{u}_n\} + a_3\{\ddot{u}_n\}\right) + [C]\left(a_1\{u_n\} + a_4\{\dot{u}_n\} + a_5\{\ddot{u}_n\}\right)\end{aligned} \qquad (17.2\text{–}7)$$

Once a solution is obtained for $\{u_{n+1}\}$, velocities and accelerations are updated as described in equations (17.2–5) and (17.2–6).

As described by Zienkiewicz(39), the solution of equation (17.2–4) by means of Newmark equations (17.2–2) and (17.2–3) is unconditionally stable for:

$$\alpha \geq \frac{1}{4}\left(\frac{1}{2} + \delta\right)^2, \quad \delta \geq \frac{1}{2}, \quad \frac{1}{2} + \delta + \alpha > 0 \qquad (17.2\text{–}8)$$

The Newmark parameters are related to the input as follows:

$$\alpha = \frac{1}{4}(\cdot 1 + \gamma)^2, \quad \delta = \frac{1}{2} + \gamma \qquad (17.2\text{–}9)$$

where: γ = amplitude decay factor (input on **TINTP** command).

Alternatively, the α and δ parameters may be input directly on the **TINTP** command. By inspection of equations (17.2–8) and (17.2–9), unconditional stability is achieved when $\delta = \frac{1}{2} + \gamma$, $\alpha \geq \frac{1}{4}\,(1 + \gamma)^2$, and $\gamma \geq 0$. Thus all solutions of equation (17.2–4) are stable if $\gamma \geq 0$. For a piezoelectirc analysis, the Crank–Nicolson and constant average acceleration methods must both be requested (α = 0.25, δ = 0.5, and θ (*THETA*) = 0.5 on the **TINTP** command).

Typically the amplitude decay factor (γ) in equation (17.2–9) takes a small value (the default is 0.005). The Newmark method becomes the constant average acceleration method when $\gamma = 0$, which in turns means $\alpha = \frac{1}{4}$ and $\delta = \frac{1}{2}$ (Bathe(2)). Results from the constant average acceleration method do not show any numerical damping in terms of displacement amplitude errors. If other sources of damping are not present, the lack of numerical damping can be undesirable in that the higher frequencies of the structure can produce unacceptable levels of numerical noise (Zienkiewicz(39)). A certain level of numerical damping is usually desired and is achieved by degrading the Newmark approximation by setting $\gamma > 0$.

Solution

Three methods of solution to equation (17.2–7) are available: full, reduced and mode superposition (**TRNOPT** command) and each are described subsequently.

Full solution method (TRNOPT,FULL)

The full solution method solves equation (17.2–7) directly and makes no additional assumptions. In a nonlinear analysis, the Newton–Raphson method (Section 15.9) is employed along with the Newmark assumptions. The inversion of equation (17.2–7) (or its nonlinear equivalent) employs the same wavefront solver used for a static analysis in Section 15.7. Section 15.6 discusses the procedure for the program to automatically determine the time step size required for each time step.

Inherent to the Newmark method is that the values of $\{u_o\}, \ \{\dot{u}_o\}, \text{ and } \{\ddot{u}_o\}$ at the start of the transient must be known. Non–zero initial conditions are input either with the **IC** command or by performing a static analysis load step (or load steps) prior to the start of the transient itself. Static load steps are performed in a transient analysis by turning off the transient time integration effects with the **TIMINT** command. The transient itself is then started by **TIMINT**,ON. The default with **ANTYPE**,TRANS is **TIMINT**,ON; that is, to start the transient immediately. (This implies $\{u\} = \{\dot{u}\} = \{\ddot{u}\} = 0$). The initial conditions are outlined in the subsequent paragraphs. Cases referring to “no previous load step” mean that the first load step is transient.

Initial Displacement

The initial displacements are:

$$\{u_o\} = \begin{cases} \{0\} & \text{if no previous load step available and no \textbf{IC} command is used} \\ \{u_s'\} & \text{if no previous load step available but \textbf{IC} command(s) is used} \\ \{u_s\} & \text{if previous load step available (which had \textbf{TIMINT},OFF)} \end{cases} \qquad (17.2–10)$$

where: $\{u_o\}$ = vector of initial displacements

$\{u_s'\}$ = displacement vector specified by the **IC** command

$\{u_s\}$ = displacement vector resulting from a static (**TIMINT**,OFF) analysis of the previous load step

Initial Velocity

The initial velocities are:

$$\{\dot{u}_o\} = \begin{cases} \{0\} & \text{if no previous load step available and no } \textbf{IC} \text{ command is used} \\ \{\dot{u}_s'\} & \text{if no previous load step available but } \textbf{IC} \text{ command(s) is used} \\ \dfrac{\{u_s\} - \{u_{s-1}\}}{\Delta t} & \text{if previous load step available (which had } \textbf{TIMINT}\text{,OFF)} \end{cases} \qquad (17.2\text{–}11)$$

where: $\{\dot{u}_o\}$ = vector of initial velocities

$\{\dot{u}_s'\}$ = vector of velocities specified by **IC** command

$\{u_s\}$ = displacements from a static (**TIMINT**,OFF) analysis of the previous load step

$\{u_{s-1}\}$ = displacement corresponding to the time point before $\{u_s\}$ solution. $\{u_{s-1}\}$ is {0} if $\{u_s\}$ is the first solution of the analysis (i.e. load step 1 substep 1).

Δt = time increment between s and s–1

Initial Acceleration

The initial acceleration is simply:

$$\{\ddot{u}_o\} = \{0\} \qquad (17.2\text{–}12)$$

where: $\{\ddot{u}_o\}$ = vector of initial accelerations

If a nonzero initial acceleration is required as for a free fall problem, an extra load step at the beginning of the transient can be used. This load step would have a small time span, step boundary conditions, and a few time steps which would allow the acceleration to be well represented at the end of the load step.

Nodal and Reaction Load Computation

Inertia, damping and static loads on the nodes of each element are computed.

The inertial load part of the element output is computed by:

$$\{F_e^m\} = [M_e]\{\ddot{u}_e\} \qquad (17.2\text{–}13)$$

where: $\{F_e^m\}$ = vector of element inertial forces

$[M_e]$ = element mass matrix

$\{\ddot{u}_e\}$ = element acceleration vector

The acceleration of a typical DOF is given by equation (17.2–5) for time t_{n+1}. The acceleration vector $\{\ddot{u}_e\}$ is the average acceleration between time t_{n+1} and time t_n, since the Newmark assumptions (equations (17.2–2) and (17.2–3)) assume the average acceleration represents the true acceleration.

The damping load part of the element output is computed by:

$$\{F_e^c\} = [C_e]\{\dot{u}_e\} \qquad (17.2\text{–}14)$$

where: $\{F_e^c\}$ = vector of element damping forces

$[C_e]$ = element damping matrix

$\{\dot{u}_e\}$ = element velocity vector

The velocity of a typical DOF is given by equation (17.2–6).

The static load is part of the element output computed in the same way as in a static analysis (Section 15.12). The nodal reaction loads are computed as the negative of the sum of all three types of loads (inertia, damping, and static) over all elements connected to a given fixed displacement node.

Reduced solution method (TRNOPT,REDUC)

The reduced solution method uses reduced structure matrices to solve the time–dependent equation of motion (equation (17.2–1)) for linear structures. The solution method imposes the following additional assumptions and restrictions:

1. Constant [M], [C], and [K] matrices. (A gap condition is permitted as described below.). This implies no large deflections or change of stress stiffening, as well as no plasticity, creep, or swelling.
2. Constant time step size.
3. No element load vectors. This implies no pressures or thermal strains. Only nodal forces applied directly at master DOF or acceleration effects acting on the reduced mass matrix are permitted.
4. Non–zero displacements may be applied only at master DOF.

Description of Analysis

This method usually runs faster than the full transient dynamic analysis by several orders of magnitude, principally because the matrix on the left–hand side of equation (17.2–7) needs to be inverted only once and the transient analysis is then reduced to a series of matrix multiplications. Also, the technique of “matrix reduction” discussed in Section 17.6 is used in this method, so that the matrix representing the system will be reduced to the essential DOFs required to characterize the response of the system. These essential DOFs are referred to as the “master degrees of freedom”. Their automatic selection is discussed in Section 15.5 and guidelines for their manual selection are given in Chapter 3 of the *ANSYS Structural Analysis Guide*. The reduction of equation (17.2–7) for the reduced transient method results in:

$$\left(a_0[\hat{M}] + a_1[\hat{C}] + [\hat{K}]\right)\left\{\hat{u}_{n+1}\right\} = \{\hat{F}\} + [\hat{M}]\left(a_0\{\hat{u}_n\} + a_2\left\{\dot{\hat{u}}_n\right\} + a_3\left\{\ddot{\hat{u}}_n\right\}\right) + [\hat{C}]\left(a_1\{\hat{u}_n\} + a_4\left\{\dot{\hat{u}}_n\right\} + a_5\left\{\ddot{\hat{u}}_n\right\}\right) \qquad (17.2\text{–}15)$$

where the coefficients (a_i) are defined after equation (17.2–6). The ^ symbol is used to denote reduced matrices and vectors. $[\hat{K}]$ may contain prestressed effects (**PSTRES**,ON) corresponding to a non–varying stress state as described in Section 3.3. These equations, which have been reduced to the master DOFs, are then solved by inverting the left–hand side of equation (17.2–15) and performing a matrix multiplication at each time step.

For the initial conditions, a static solution is done at time = 0 using the given loads to define $\{\hat{u}_o\}$. $\{\dot{\hat{u}}_o\}$ and $\{\ddot{\hat{u}}_o\}$ are assumed to be zero.

A "quasi–linear" analysis variation is also available with the reduced method. This variation allows interfaces (gaps) between any of the master DOFs and ground, or between any pair of master DOFs. If the gap is initially closed, these interfaces are accounted for by including the stiffness of the interface in the stiffness matrix, but if the gap should later open, a force is applied in the load vector to nullify the effect to the stiffness. If the gap is initially open, it causes no effect on the initial solution, but if it should later close, a force is again applied in the load vector.

The force associated with the gap is:

$$F_{gp} = k_{gp}\, u_g \qquad (17.2\text{–}16)$$

where: k_{gp} = gap stiffness (input quantity *STIF*, **GP** command)

u_g = $u_A - u_B - u^{gp}$

u_A, u_B = displacement across gap (must be master degrees of freedom)

u^{gp} = initial size of gap (input quantity *GAP*, **GP** command)

This procedure adds an explicit term to the implicit integration procedure. An alternate procedure is to use the full method, modeling the linear portions of the structure as superelements and the gaps as gap elements. This latter procedure (implicit integration) normally allows larger time steps because it modifies both the stiffness matrix and load vector when the gaps change status.

Expansion Pass

The expansion pass of the reduced transient analysis involves computing the displacements at slave DOFs (see equation (17.6–16)) and computing element stresses.

Nodal load output consists of the static loads only as described for a static analysis (Section 15.12). The reaction load values represent the negative of the sum of the above static loads over all elements connected to a given fixed displacement node. Damping and inertia forces are not included in the reaction loads.

Mode Superposition Method (TRNOPT,MSUP)

The mode superposition method uses the natural frequencies and mode shapes of a linear structure to predict the response to transient forcing functions. This solution method imposes the following additional assumptions and restrictions:

1. Constant [K] and [M] matrices. (A gap condition is permitted as described under the reduced solution method.) This implies no large deflections or change of stress stiffening, as well as no plasticity, creep, or swelling.
2. Constant time step size.
3. There are no element damping matrices. However, various types of system damping are available.
4. Time varying imposed displacements are not allowed.

The development of the general mode superposition procedure is described in Section 15.11. Equation (15.11–21) is integrated through time for each mode by the Newmark method.

The initial values of the displacements {u} come from a static analysis (equation (17.1–1)) of

$$[K]\{u_o\} = \{F_o\} \tag{17.2–17}$$

where: $\{F_o\}$ = the forces applied at time = 0

so that

$$\{u_o\} = [K]^{-1}\{F_o\} \tag{17.2–18}$$

The initial values of the modal coordinates are computed from $\{u_o\}$ by

$$\{y_o\} = [\Phi]^{-1}\{u_o\} \tag{17.2–19}$$

where: $[\Phi]$ = is the matrix of mode shape vectors

Equation (17.2–19) is an alternate form of equation (15.11–3). $\{\dot{y}_o\}$ and $\{\ddot{y}_o\}$ are assumed to be zero.

The load vector, which must be converted to modal coordinates (equation (15.11–20)) at each time step, is given by

$$\{F\} = \left\{F^{nd}\right\} + s\{F^{s}\} + \left\{F_{gp}\right\} + \{F_{ma}\} \qquad (17.2–20)$$

where: $\left\{F^{nd}\right\}$ = nodal force vector

s = load vector scale factor, (input quantity *FACT*, **LVSCALE** command)

$\{F^{s}\}$ = load vector from the modal analysis (see Section 15.11).

$\left\{F_{gp}\right\}$ = gap force vector (equation (17.2–16)).

$\{F_{ma}\}$ = inertial force $\left(\{F_{ma}\} = [M]\,\{a\}\right)$

$\{a\}$ = acceleration vector (**ACEL** command) (see Section 15.1)

If the modal analysis was performed using the reduced method (**MODOPT**,REDUC), then the matrices and vectors in the above equations would be in terms of the master DOFs (e.g. $\{\hat{u}\}$).

Expansion Pass

The expansion pass of the mode superposition transient analysis involves computing the displacements at slave DOFs if the reduced modal analysis was used (**MODOPT**,REDUC) (see equation (17.6–16)) and computing element stresses.

Nodal load output consists of the static loads only as described for a static analysis (Section 15.12). The reaction load values represent the negative of the sum of the static loads over all elements connected to a given fixed displacement node. Damping and inertia forces are not included in the reaction loads.

17.2.3 Description of Thermal, Magnetic and Other First Order Systems

The governing equation of interest is as follows:

$$[C]\{\dot{u}\} + [K]\{u\} = \{F^{a}\} \qquad (17.2–21)$$

where: [C] = damping matrix

[K] = coefficient matrix

{u} = vector of DOF values

$\{\dot{u}\}$ = time rate of the DOF values

$\{F^{a}\}$ = applied load vector

In a thermal analysis, [C] is the specific heat matrix, [K] the conductivity matrix, {u} the vector of nodal temperatures and $\{F^{a}\}$ the applied heat flows. Table 17.2–1 relates the nomenclature used in Sections 6.2 and 5.2 for thermal, magnetic and electrical analyses to equation (17.2–21).

Table 17.2–1 Nomenclature

	{u}		{F^a}	
Thermal	{T}	temperatures	$\{Q^a\}$	heat flow
Scalar Magnetic	$\{\phi\}$	scalar potential	$\{F^a\}$	flux
Vector Magnetic	{A}	vector potential	$\{F^a\}$	current segment
Electrical	{V}	voltage	$\{I^a\}$	current

TRNOPT,REDUC and **TRNOPT**,MSUP do not apply to first order systems.

The procedure employed for the solution of equation (17.2–21) is the generalized trapezoidal rule (Hughes(165)):

$$\{u_{n+1}\} = \{u_n\} + (1 - \theta)\,\Delta t\,\{\dot{u}_n\} + \theta\Delta t\,\{\dot{u}_{n+1}\} \qquad (17.2\text{–}22)$$

where:
θ = transient integration parameter (input on **TINTP** command)
Δt = $t_{n+1} - t_n$
$\{u_n\}$ = nodal DOF values at time t_n
$\{\dot{u}_n\}$ = time rate of the nodal DOF values at time t_n (computed at previous time step)

Equation (17.2–21) can be written at time t_{n+1} as:

$$[C]\{\dot{u}_{n+1}\} + [K]\{u_{n+1}\} = \{F^a\} \qquad (17.2\text{–}23)$$

Substituting $\{\dot{u}_{n+1}\}$ from equation (17.2–22) into this equation yields:

$$\left(\frac{1}{\theta\Delta t}\,[C] + [K]\right)\{u_{n+1}\} = \{F^a\} + [C]\left(\frac{1}{\theta\Delta t}\,\{u_n\} + \frac{1-\theta}{\theta}\,\{\dot{u}_n\}\right) \qquad (17.2\text{–}24)$$

The solution of equation (17.2–24) employs the same wavefront (or conjugate gradient if **EQSLV**,JCG is used) solver used for static analysis in Section 17.1. Once $\{u_{n+1}\}$ is obtained, $\{\dot{u}_{n+1}\}$ is updated using equation (17.2–22). In a nonlinear analysis, the Newton–Raphson method (Section 15.9) is employed along with the generalized trapezoidal assumption, equation (17.2–22).

The transient integration parameter θ (**TINTP** command) defaults to 0.5 (Crank–Nicholson method) if **SOLC**,OFF and 1.0 (backward Euler method) if **SOLC**,ON. If $\theta = 1$, the method is referred to as the backward Euler method. For all $\theta > 0$, the system equations that follow are said to be implicit. In addition, for the more limiting case of $\theta \geq 1/2$, the solution of these equations is said to be unconditionally stable; i.e., stability is not a factor in time step (Δt) selection. The selection of θ using the **TINTP** command is therefore limited to

$$\frac{1}{2} \leq \theta \leq 1 \qquad (17.2\text{–}25)$$

which corresponds to an unconditionally stable, implicit method. For a piezoelectric analysis, the Crank–Nicolson and constant average acceleration methods must both be requested (α (ALPHA) = 0.25, δ (DELTA) = 0.5, and θ = 0.5 on the **TINTP** command). Since the $\{\dot{u}_n\}$ influences $\{u_{n+1}\}$, sudden changes in loading need to be handled carefuly for values of θ < 1.0. See the *ANSYS Basic Analysis Procedures Guide* for more details.

The generalized–trapezoidal method requires that the values of $\{u_o\}$ and $\{\dot{u}_o\}$ at the start of the transient must be known. Non–zero initial conditions are input either with the **IC** command (for $\{u_o\}$) or by performing a static analysis load step (or load steps) prior to the start of the transient itself. Static load steps are performed in a transient analysis by turning off the transient time integration effects with the **TIMINT** command. The transient itself is then started by **TIMINT**,ON. The default with **ANTYPE**,TRANS is **TIMINT**,ON; that is, to start the transient immediately. This implies $(\{u\} = \{\dot{u}\} = \{0\})$. The initial conditions are outlined in the subsequent paragraphs.

Initial DOF Values

The initial DOF values for first order systems are:

$$\{u_o\} = \begin{cases} \{a\} & \text{if no previous load step available and no \textbf{IC} command is used} \\ \{u_s'\} & \text{if no previous load step available but the \textbf{IC} command is used} \\ \{u_s\} & \text{if previous load step available (which had \textbf{TIMINT},OFF)} \end{cases} \qquad (17.2\text{–}26)$$

where:
$\{u_o\}$ = vector of initial DOF values
$\{a\}$ = vector of uniform DOF values
$\{u_s'\}$ = DOF vector specified by the **IC** command
$\{u_s\}$ = DOF vector resulting from a static (**TIMINT**,OFF) analysis of the previous load step available

{a} is set to TEMP (**BFUNIF** command) and/or to the temperature specified by the **IC** command for thermal DOFs (temperatures) and zero for other DOFs.

Nodal and Reaction Load Computation

Damping and static loads on the nodes of each element are computed.

The damping load part of the element output is computed by:

$$\{F_e^c\} = [C_e]\{\dot{u}_e\} \qquad (17.2\text{–}27)$$

where:
$\{F_e^c\}$ = vector of element damping loads
$[C_e]$ = element damping matrix

$\{\dot{u}_e\}$ = element velocity vector

The velocity of a typical DOF is given by equation (17.2–22). The velocity vector $\{\dot{u}_e\}$ is the average velocity between time t_n and time t_{n+1}, since the general trapezoidal rule (equation (17.2–22)) assumes the average velocity represents the true veocity.

The static load is part of the element output computed in the same way as in a static analysis (Section 15.12). The nodal reaction loads are computed as the negative of the sum of both types of loads (damping and static) over all elements connected to a given fixed DOF node.

17.3 Mode–Frequency Analysis (ANTYPE,MODAL)

17.3.1 Assumptions and Restrictions

1. Valid for structural and fluid degrees of freedom (DOFs).
2. The structure has constant stiffness and mass effects.
3. There is no damping, unless the damped eigensolver (**MODOPT**,DAMP) is selected.
4. The structure has no time varying forces, displacements, pressures, or temperatures applied (free vibration).

17.3.2 Description of Analysis

This analysis type is used for natural frequency and mode shape determination. The equation of motion for an undamped system, expressed in matrix notation using the above assumptions is:

$$[M]\{\ddot{u}\} + [K]\{u\} = \{0\} \qquad (17.3\text{–}1)$$

Note that [K], the structure stiffness matrix, may include prestress effects (**PSTRES**,ON). For a discussion of the damped eigensolver (**MODOPT**,DAMP) see Section 15.10.

For a linear system, free vibrations will be harmonic of the form:

$$\{u\} = \{\phi\}_i \cos \omega_i t \qquad (17.3\text{–}2)$$

where:

$\{\phi\}_i$ = eigenvector representing the mode shape of the ith natural frequency

ω_i = ith natural circular frequency (radians per unit time)

t = time

Thus, equation (17.3–1) becomes:

$$\left(-\omega_i^2[M] + [K]\right) \{\phi\}_i = \{0\} \qquad (17.3\text{–}3)$$

This equality is satisfied if either $\{\phi\}_i = \{0\}$ or if the determinant of $\left([K] - \omega^2 [M]\right)$ is zero. The first option is the trivial one and, therefore, is not of interest. Thus, the second one gives the solution:

$$\left|[K] - \omega^2[M]\right| = 0 \tag{17.3–4}$$

This is an eigenvalue problem which may be solved for up to n values of ω^2 and n eigenvectors $\{\phi\}_i$ which satisfy equation (17.3–3) where n is the number of DOFs. The eigenvalue and eigenvector extraction techniques are discussed in Section 15.10.

Rather than outputting the natural circular frequencies (ω), the natural frequencies (f) are output; where:

$$f_i = \frac{\omega_i}{2\pi} \tag{17.3–5}$$

where: f_i = i^{th} natural frequency (cycles per unit time)

If *Nrmkey* = OFF on the **MODOPT** command (default), then each eigenvector $\{\phi\}_i$ is normalized such that:

$$\{\phi\}_i^T [M] \{\phi\}_i = 1 \tag{17.3–6}$$

If *Nrmkey* = ON on the **MODOPT** command, $\{\phi\}_i$ is normalized such that its largest component is 1.0 (unity).

For **MODOPT**,REDUC, the n eigenvectors can then be expanded in the expansion pass using the **MXPAND** command to the full set of structure modal displacement DOFs using:

$$\{\phi_s\}_i = -[K_{ss}]^{-1}[K_{sm}] \{\hat{\phi}\}_i \tag{17.3–7}$$

where: $\{\phi_s\}_i$ = slave DOFs vector of mode i (slave degrees of freedom are those DOFs that had been condensed out)

$[K_{ss}]$, $[K_{sm}]$ = submatrix parts as shown in equation (17.6–2)

$\{\hat{\phi}\}_i$ = master DOF vector of mode i

If **MXPAND**,,,,YES, then the modal element forces and stresses are also computed. These expanded modal solutions are written to the results file as load step number 1. The field SIGNIF on the **MXPAND** command will be ignored if modal expansion is performed before or without the **SPORT**,SPRS or **SPORT**,DDAM options of the spectrum analysis (**ANTYPE**,SPECTRUM).

A discussion of effective mass is given in Section 17.7.

17.4 Harmonic Response Analyses (ANTYPE, HARMIC)

This analysis solves the time–dependent equations of motion (equation (17.2–1)) for linear structures undergoing steady–state vibration.

17.4.1 Assumptions and Restrictions

1. Valid for structural, fluid and magnetic degrees of freedom (DOFs).
2. The entire structure has constant stiffness, damping, and mass effects.
3. All loads and displacements vary sinusoidally at the same known frequency (although not necessarily in phase)
4. Element loads are assumed to be real (in–phase) only, except for current density.

17.4.2 Description of Analysis

Consider the general equation of motion for a structural system (equation (17.2–1)).

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F^a\} \qquad (17.4–1)$$

where:
$[M]$ = structural mass matrix
$[C]$ = structural damping matrix
$[K]$ = structural stiffness matrix
$\{\ddot{u}\}$ = nodal acceleration vector
$\{\dot{u}\}$ = nodal velocity vector
$\{u\}$ = nodal displacement vector
$\{F^a\}$ = applied load vector

As stated above, all points in the structure are moving at the same known frequency, however, not necessarily in phase. Also, it is known that the presence of damping causes phase shifts. Therefore, the displacements may be defined as:

$$\{u\} = \left\{u_{max}\ e^{i\phi}\right\} e^{i\Omega t} \qquad (17.4–2)$$

where:
u_{max} = maximum displacement
i = square root of –1
Ω = imposed circular frequency (radians/time) = $2\pi f$

f = imposed frequency (cycles/time) (input quantities *FREQB* and *FREQE* on the **HARFRQ** command)

t = time

ϕ = displacement phase shift (radians)

Note that u_{max} and ϕ may be different at each DOF. The use of complex notation allows a compact and efficient description and solution of the problem. Equation (17.4–2) can be rewritten as:

$$\{u\} = \{u_{max}\,(\cos\phi + i\,\sin\phi)\}\,e^{i\Omega t} \tag{17.4–3}$$

or as:

$$\{u\} = (\{u_1\} + i\{u_2\})\,e^{i\Omega t} \tag{17.4–4}$$

where:

$\{u_1\} = \{u_{max}\cos\phi\}$ = real displacement vector (input quantity *VALUE* on **D** command, when specified)

$\{u_2\} = \{u_{max}\sin\phi\}$ = imaginary displacement vector (input quantity *VALUE2* on **D** command, when specified)

The force vector can be specified analogously to the displacement:

$$\{F\} = \left\{F_{max}\,e^{i\psi}\right\}\,e^{i\Omega t} \tag{17.4–5}$$

$$\{F\} = \{F_{max}\,(\cos\psi + i\,\sin\psi)\}\;\;e^{i\Omega t} \tag{17.4–6}$$

$$\{F\} = (\{F_1\} + i\{F_2\})\,e^{i\Omega t} \tag{17.4–7}$$

where: F_{max} = force amplitude

ψ = force phase shift (radians)

$\{F_1\} = \{F_{max}\cos\psi\}$ = real force vector (input quantity *VALUE* on **F** command, when specified)

$\{F_1\} = \{F_{max}\sin\psi\}$ = imaginary force vector (input quantity on *VALUE2* on **F** command, when specified)

Substituting equations (17.4–4) and (17.4–7) into equation (17.4–1) gives:

$$\left(-\,\Omega^2[M] + i\Omega[C] + [K]\right)\;(\{u_1\} + i\{u_2\})\,e^{i\Omega t} = (\{F_1\} + i\{F_2\})\,e^{i\Omega t} \tag{17.4–8}$$

The dependence on time $\left(e^{i\Omega t}\right)$ is the same on both sides of the equation and may therefore be removed:

$$\left([K] - \Omega^2 [M] + i\Omega [C]\right) \left(\{u_1\} + i\{u_2\}\right) = \{F_1\} + i\{F_2\} \tag{17.4–9}$$

The solution of this equation is discussed later.

17.4.3 Complex Displacement Output

The output complex displacement at each DOF may be given in one of two forms:

1. If the command **HROUT**,ON is given, the output is of the same form as u_1 and u_2 as defined in equation (17.4–4).
2. If the command **HROUT**,OFF is given, the output is of the form u_{max} and ϕ (amplitude and phase angle (in degrees)), as defined in equation (17.4–3). These two terms are computed at each DOF as:

$$u_{max} = \sqrt{u_1^2 + u_2^2} \tag{17.4–10}$$

$$\phi = \tan^{-1} \frac{u_2}{u_1} \tag{17.4–11}$$

Note that the phase angle ϕ is relative to the input forcing phase angle ψ.

17.4.4 Nodal and Reaction Load Computation

Inertia, damping and static loads on the nodes of each element are computed.

The real and imaginary inertia load parts of the element output are computed by:

$$\{F_1^m\}_e = (2\pi\ \Omega)^2 [M_e] \{u_1\}_e \tag{17.4–12}$$

$$\{F_2^m\}_e = (2\pi\ \Omega)^2 [M_e] \{u_2\}_e \tag{17.4–13}$$

where:

$\{F_1^m\}_e$	=	vector of element inertia forces (real part)
$[M_e]$	=	element mass matrix
$\{u_1\}_e$	=	element real displacement vector
$\{F_2^m\}_e$	=	vector of element inertia forces (imaginary part)
$\{u_2\}_e$	=	element imaginary displacement vector

The real and imaginary damping loads part of the element output are computed by:

$$\{F_1^c\}_e = -2\pi\ \Omega\ [C_e]\ \{u_2\}_e \tag{17.4–14}$$

$$\{F_2^c\}_e = 2\pi\ \Omega\ [C_e]\ \{u_1\}_e \tag{17.4–15}$$

where: $\{F_1^c\}_e$ = vector of element damping forces (real part)

$[C_e]$ = element damping matrix

$\{F_2^c\}_e$ = vector of element damping forces (imaginary part)

The real static load is computed the same way as in a static analysis (Section 15.12) using the real part of the displacement solution $\{u_1\}_e$. The imaginary static load is computed also the same way, using the imaginary part $\{u_2\}_e$. Note that the imaginary part of the element loads (e.g. $\{F^{pr}\}$) are normally zero, except for current density loads.

The nodal reaction loads are computed as the sum of all three types of loads (inertia, damping, and static) over all elements connected to a given fixed displacement node.

17.4.5 Solution

Three methods of solution to equation (17.4–9) are available: full, reduced and mode superposition (**HROPT** command) and each are described subsequently.

Full solution method (HROPT,FULL)

The full solution method solves equation (17.4–9) directly. Equation (17.4–9) may be expressed as:

$$[K_c]\ \{u_c\} = \{F_c\} \tag{17.4–16}$$

where c denotes a complex matrix or vector. Equation (17.4–16) is solved using the same wavefront solver used for a static analysis in Section 15.7, except that it is done using complex arithmetic.

Reduced solution method (HROPT,REDUC)

The reduced solution method uses reduced structure matrices to solve the equation of motion (equation (17.4–1). This solution method imposes the following additional assumptions and restrictions:

1. No element load vectors (e.g. pressures or thermal strains). Only nodal forces appied directly at master DOF or acceleration effects acting on the reduced mass matrix are permitted.
2. Non–zero displacements may be applied only at master DOF.

This method usually runs faster than the full harmonic analysis by several orders of magnitude, principally because the technique of “matrix reduction” discussed in Section 17.6 is used so that the matrix representing the system will be reduced to the essential DOFs required to characterize the response of the system. These essential DOFs are referred to as the “master degrees of freedom”. Their automatic selection is discussed in Section 15.5 and guidelines for their manual selection are given in Chapter 3 of the *ANSYS Structural Analysis Guide*. The reduction of equation (17.4–9) for the reduced method results in:

$$\left([\hat{K}] - \Omega^2 [\hat{M}] + i \, \Omega \, [\hat{C}]\right) \left(\{\hat{u}_1\} + i \, \{\hat{u}_2\}\right) = \{\hat{F}_1\} + i \, \{\hat{F}_2\} \qquad (17.4\text{–}17)$$

where the ^ denotes reduced matrices and vectors. These equations, which have been reduced to the master DOFs, are then solved in the same way as the full method. $[\hat{K}]$ may contain prestressed effects (**PSTRES**,ON) corresponding to a non–varying stress state, described in Section 3.3.

Expansion pass

The reduced harmonic response method produces a solution of complex displacements at the master DOFs only. In order to complete the analysis, an expansion pass is performed (**EXPASS**,ON). As in the full method, both a real and imaginary solution corresponding to $\{\hat{u}_1\}$ and $\{\hat{u}_2\}$ can be expanded (see equation (17.6–16)) and element stresses obtained (**HREXP**,ALL).

Alternatively, a solution at a certain phase angle may be obtained (**HREXP**, *ANGLE*). The solution is computed at this phase angle for each master DOF by:

$$\hat{u} = \hat{u}_{max} \cos\left(\phi - \theta\right) \qquad (17.4\text{–}18)$$

where:

$\hat{u}_{max}$ = amplitude given by equation (17.4–10)

ϕ = computed phase angle given by equation (17.4–11)

θ = $\theta' \frac{2\pi}{360}$

θ' = input quantity *ANGLE* (in degrees), **HREXP** Command

This solution is then expanded and stresses obtained for these displacements. In this case, only the real part of the nodal loads is computed.

Mode superposition method (HROPT,MSUP)

The mode superposition method uses the natural frequencies and mode shapes to compute the response to a sinusoidally varying forcing function. This solution method imposes the following additional assumptions and restrictions:

1. Non–zero imposed harmonic displacements are not allowed.
2. There are no element damping matrices. However, various types of system damping are available.

The development of the general mode superposition procedure is given in Section 15.11. The equation of motion (equation (17.4–1)) is converted to modal form, as described in Section 15.11. Equation (15.11–21) is:

$$\ddot{y}_j + 2\omega_j \xi_j \dot{y}_j + \omega_j^2 y_j = f_j \qquad (17.4\text{–}19)$$

where: y_j = modal coordinate

ω_j = natural circular frequency of mode j

ξ_j = fraction of critical damping for mode j

f_j = force in modal coordinates

The load vector which is converted to modal coordinates (equation (15.11–20)) is given by

$$\{F\} = \{F^{nd}\} + s\{F^s\} \qquad (17.4\text{–}20)$$

where: $\{F^{nd}\}$ = nodal force vector

s = load vector scale factor, (input quantity *FACT*, **LVSCALE** command)

$\{F^s\}$ = load vector from the modal analysis (see Section 15.11).

For a steady sinusoidal vibration, f_j has the form

$$f_j = f_{jc} e^{i\Omega t} \qquad (17.4\text{–}21)$$

where: f_{jc}= complex force amplitude

Ω= imposed circular frequency

For equation (17.4–19) to be true at all times, y_j must have a similar form as f_j, or

$$y_j = y_{jc} e^{i\Omega t} \qquad (17.4\text{–}22)$$

where: y_{jc} = complex amplitude of the modal coordinate for mode j.

Differentiating equation (17.4–22), and substituting equations (17.4–21) and (17.4–22) into equation (17.4–19),

$$-\Omega^2 y_{jc}\, e^{i\Omega t} + 2\omega_j\, \xi_j \left(i\Omega y_{jc}\, e^{i\Omega t}\right) + \omega_j^{\,2}\, y_{jc}\, e^{i\Omega t} = f_{jc}\, e^{i\Omega t} \qquad (17.4\text{–}23)$$

Collecting coefficients of y_{jc} and dividing by $e^{i\Omega t}$

$$\left(-\Omega^2 + i2\omega_j\Omega\, \xi_j + \omega_j^{\,2}\right) y_{jc} = f_{jc} \qquad (17.4\text{–}24)$$

solving for y_{jc},

$$y_{jc} = \frac{f_{jc}}{\left(\omega_j^{\,2} - \Omega^2\right) + i\left(2\omega_j\, \Omega\xi_j\right)} \qquad (17.4\text{–}25)$$

The contribution from each mode is:

$$\left\{C_j\right\} = \left\{\phi_j\right\} y_{jc} \qquad (17.4\text{–}26)$$

where: $\left\{C_j\right\}$ = contribution of mode j (output if *Mcont* = ON on the **HROUT** command)

$\{\phi_j\}$ = mode shape for mode j

Finally, the complex displacements are obtained from equation (15.11–3) as

$$\{u_c\} = \sum_{j=1}^{n} \left\{C_j\right\} \qquad (17.4\text{–}27)$$

where: $\{u_c\}$ = vector of complex displacements

If the modal analysis was performed using the reduced method (**MODOPT**,REDUC), then the vectors $\{\phi\}$ and $\{u_c\}$ in the above equations would be in terms of the master DOFs (i.e. $\{\hat{\phi}\}$ and $\{\hat{u}_c\}$).

17.4.6 Automatic Frequency Spacing

In harmonic response analysis, the imposed frequencies that involve the most kinetic energy are those near the natural frequencies of the structure. The automatic frequency spacing or “cluster” option (*Clust*=ON on the **HROUT** command) provides an approximate method of choosing suitable imposed frequencies. The nearness of the imposed frequencies to the natural frequencies depends on damping, because the resonance peaks narrow when the damping is reduced. Figure 17.4–1 shows two typical resonance peaks and the imposed frequencies chosen by this method, which are computed from:

$$\Omega^i_{-j} = \omega_i / a_{ij} \qquad (17.4\text{–}28)$$

$$\Omega^i_{+j} = \omega_i a_{ij} \qquad (17.4\text{–}29)$$

Equation (17.4–28) gives frequencies slightly less than the natural circular frequency ω_i. Equation (17.4–29) gives slightly higher frequencies. The spacing parameter a_{ij} is defined as:

$$a_{ij} = 1 + \left(\xi_i\right)^b \qquad (17.4\text{–}30)$$

where: ξ_i = modal damping as defined by equation (15.11–22). (If ξ_i is computed as 0.0, it is redefined to be 0.005 for this equation only).

$b = \dfrac{2(N - j)}{N - 1}$

N = input quantity *NSBSTP*, **NSUBST** command, which may be between 2 and 10. Anything outside this range defaults to 4.

j = 1, 2, 3, ...2N

Each natural frequency, as well as frequencies midway between, are also chosen as imposed frequencies.

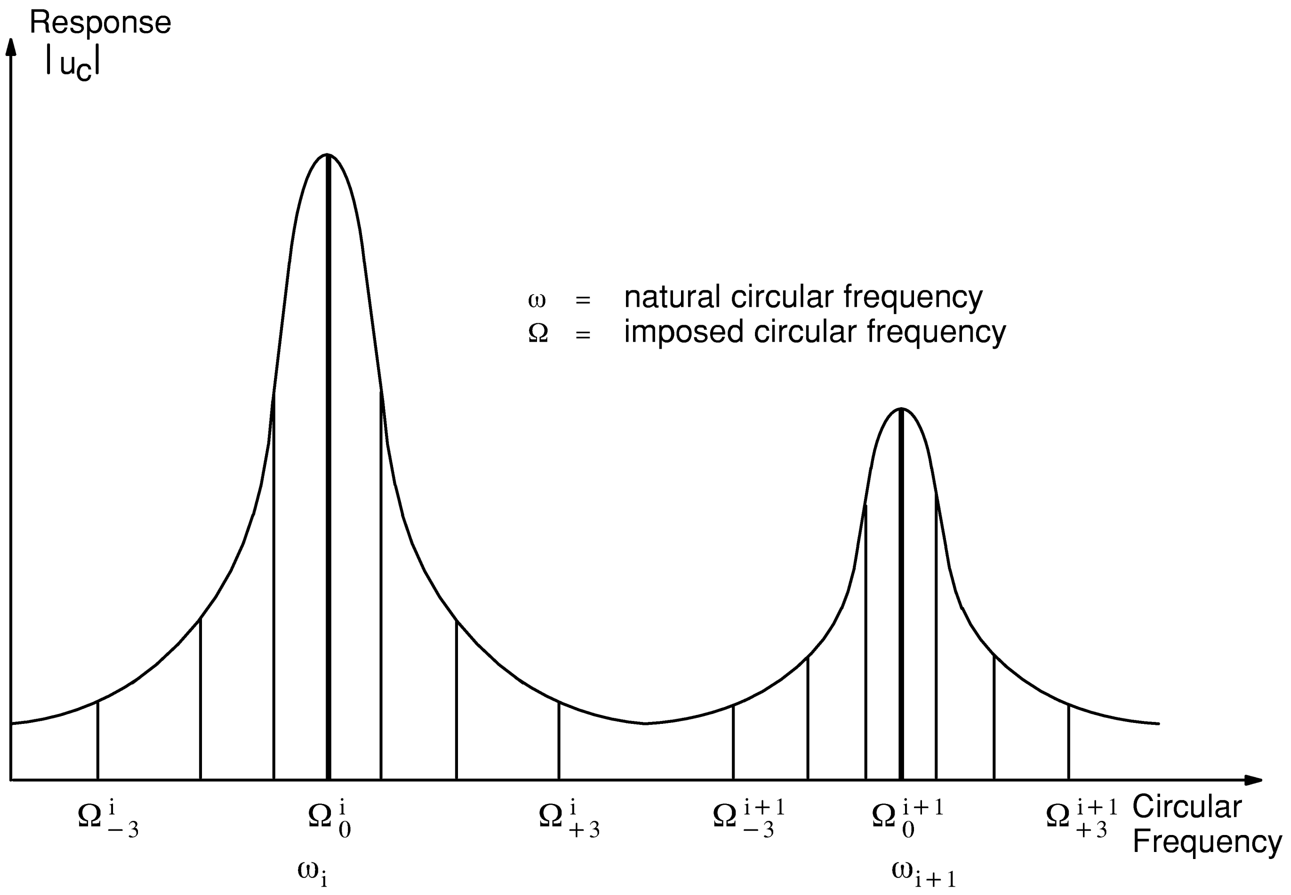


Figure 17.4–1 Automatic Frequency Spacing

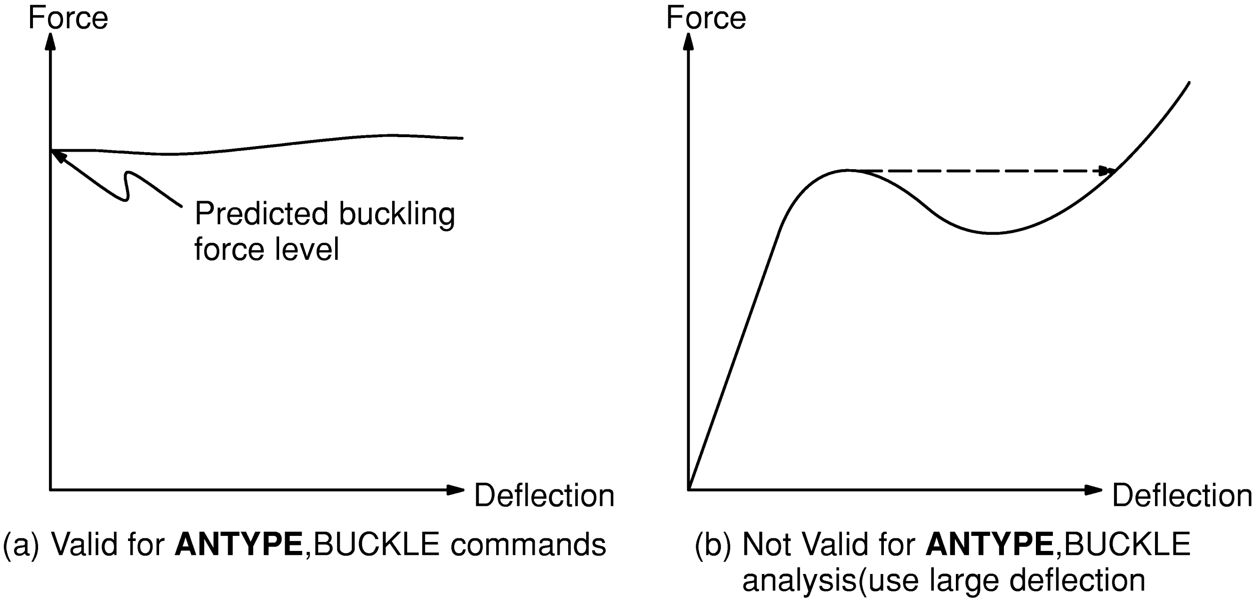
17.4.7 Expansion Pass

The expansion pass of the mode superposition method involves computing the complex displacements at slave DOFs if the reduced modal analysis was used **MODOPT**,REDUC (see equation (17.6–16)) and computing element stresses. The expansion pass is the same as the reduced method discussed in the previous section.

17.5 Buckling Analysis (ANTYPE, BUCKLE)

17.5.1 Assumptions and Restrictions

1. Valid for structural degrees of freedom (DOFs) only.
2. The structure fails suddenly, with a horizontal force–deflection curve (see Figure 17.5–1).
3. The structure has constant stiffness effects.
4. A static solution with **PSTRES**,ON was run.



(a) Valid for **ANTYPE**,BUCKLE commands

(b) Not Valid for **ANTYPE**,BUCKLE analysis(use large deflection analysis)

Figure 17.5–1 Types of Buckling Problems

17.5.2 Description of Analysis

This analysis type is for bifurcation buckling using a linearized model of elastic stability. Bifurcation buckling refers to the unbounded growth of a new deformation pattern. A linear structure with a force–deflection curve similar to Figure 17.5–1(a) is well modeled by a **ANTYPE**,BUCKLE analysis, whereas a structure with a curve like Figure 17.5–1(b) is not (a large deflection analysis (**NLGEOM**,ON) is appropriate, see Section 3.2). The buckling problem is formulated as an eigenvalue problem:

$$\left([K] + \lambda_i [S]\right) \{\psi\}_i = \{0\} \tag{17.5–1}$$

where:
$[K]$ = stiffness matrix
$[S]$ = stress stiffness matrix
λ_i = ith eigenvalue (used to multiply the loads which generated [S])
$\{\psi\}_i$ = ith eigenvector of displacements

The eigenproblem is solved as discussed in Section 15.10. The eigenvectors are normalized so that the largest component is 1.0. Thus, the stresses (when output) may only be interpreted as a relative distribution of stresses.

If the first eigenvalue closest to the shift point is negative (indicating that the loads applied in a reverse sense will cause buckling), the program will terminate. In the subspace method, the error message “Number of stress–stiffness DOF is less than requested modes” will be issued. In the reduced method, the error message “There are N tension MDOF in the reduced matrices ...” is output. The reduced method will attempt to reduce the number of MDOF until only compression (that is, positive stress–stiffness diagonals) MDOF are left. In order to circumvent this difficulty, apply an initial shift close to the expected eigenvalue with the **BUCOPT** command.

17.6 Substructuring Analysis (ANTYPE, SUBSTR)

The substructure analysis uses the technique of matrix reduction to reduce the system matrices to a smaller set of DOFs. Matrix reduction is also used by the reduced modal, reduced harmonic and reduced transient analyses.

17.6.1 Assumptions and Restrictions (within superelement)

1. Any degree of freedom (DOF) may be used.
2. The elements have constant stiffness, damping, and mass effects (e.g., material properties do not change with temperature.)

17.6.2 Description of Analysis

A superelement (substructure) may be used in any analysis type. It simply represents a collection of elements that are reduced to act as one element. This one (super) element may then be used in the actual analysis (use pass) or be used to generate more superelements (generation or use pass). To reconstruct the detailed solutions (e.g. displacements and stresses) within the superelement, an expansion pass may be done. See the *ANSYS Basic Analysis Procedures Guide* for loads which are applicable to a substructure analysis.

17.6.3 Statics

Consider the basic form of the static equations (equation (17.1–1)):

$$[K]\ \{u\} = \{F\} \tag{17.6–1}$$

{F} includes nodal, pressure, and temperature effects. It does not include $\{F^{nr}\}$ (see Section 15.9). The equations may be partitioned into two groups, the master (retained) DOFs, here denoted by the subscript “m,” and the slave (removed) DOFs, here denoted by the subscript “s.”

$$\begin{bmatrix} [K_{mm}] & [K_{ms}] \\ [K_{sm}] & [K_{ss}] \end{bmatrix} \begin{Bmatrix} \{u_m\} \\ \{u_s\} \end{Bmatrix} = \begin{Bmatrix} \{F_m\} \\ \{F_s\} \end{Bmatrix} \tag{17.6–2}$$

or expanding:

$$[K_{mm}]\{u_m\} + [K_{ms}]\{u_s\} = \{F_m\} \qquad (17.6–3)$$

$$[K_{sm}]\{u_m\} + [K_{ss}]\{u_s\} = \{F_s\} \qquad (17.6–4)$$

The master DOFs should include all DOFs of all nodes on surfaces that connect to other parts of the structure. If accelerations are to be used in the use pass or if the use pass will be a transient analysis, master DOFs throughout the rest of the structure should also be used to characterize the distributed mass. The automatic selection of master DOFs is discussed in more detail in Section 15.5, and guidelines for their selection are given in Chapter 3 of the *ANSYS Structural Analysis Guide*. Solving equation (17.6–4) for $\{u_s\}$,

$$\{u_s\} = [K_{ss}]^{-1}\{F_s\} - [K_{ss}]^{-1}[K_{sm}]\{u_m\} \qquad (17.6–5)$$

Substituting $\{u_s\}$ into equation (17.6–3)

$$\Big[[K_{mm}] - [K_{ms}][K_{ss}]^{-1}[K_{sm}]\Big]\{u_m\} = \{F_m\} - [K_{ms}][K_{ss}]^{-1}\{F_s\} \qquad (17.6–6)$$

or,

$$[\hat{K}]\{\hat{u}\} = \{\hat{F}\} \qquad (17.6–7)$$

where:

$$[\hat{K}] = [K_{mm}] - [K_{ms}][K_{ss}]^{-1}[K_{sm}] \qquad (17.6–8)$$

$$\{\hat{F}\} = \{F_m\} - [K_{ms}][K_{ss}]^{-1}\{F_s\} \qquad (17.6–9)$$

$$\{\hat{u}\} = \{u_m\}$$

$[\hat{K}]$ and $\{\hat{F}\}$ are the superelement coefficient (e.g. stiffness) matrix and load vector, respectively.

In the preceding development, the load vector for the superelement has been treated as a total load vector. The same derivation may be applied to any number of independent load vectors, which in turn may be individually scaled in the superelement use pass. For example, the analyst may wish to apply thermal, pressure, gravity, and other loading conditions in varying proportions. Expanding the right–hand sides of equations (17.6–3) and (17.6–4) one gets, respectively:

$$\{F_m\} = \sum_{i=1}^{N} \{F_{mi}\} \tag{17.6–10}$$

$$\{F_s\} = \sum_{i=1}^{N} \{F_{si}\} \tag{17.6–11}$$

where: N = number of independent load vectors.

Substituting into equation (17.6–9):

$$\{\hat{F}\} = \sum_{i=1}^{N} \{F_{mi}\} - [K_{ms}]\ [K_{ss}]^{-1} \sum_{i=1}^{N} \{F_{si}\} \tag{17.6–12}$$

To have independently scaled load vectors in the use pass, expand the left–hand side of equation (17.6–12)

$$\{\hat{F}\} = \sum_{i=1}^{N} \left\{\hat{F}_i\right\} \tag{17.6–13}$$

Substituting equation (17.6–13) into equation (17.6–12):

$$\left\{\hat{F}_i\right\} = \{F_{mi}\} - [K_{ms}]\ [K_{ss}]^{-1}\ \{F_{si}\} \tag{17.6–14}$$

If the load vectors are scaled in the use pass such that:

$$\{\hat{F}\} = \sum_{i=1}^{N} b_i \left\{\hat{F}_i\right\} \tag{17.6–15}$$

where b_i is the scaling factor (*FACT* on the **LVSCALE** command), then equation (17.6–5) becomes:

$$\{u_s\} = [K_{ss}]^{-1} \sum_{i=1}^{N} b_i \{F_{si}\} - [K_{ss}]^{-1}[K_{sm}]\ \{u_m\} \tag{17.6–16}$$

Equation (17.6–16) is used in the expansion pass to obtain the DOF values at the slave DOFs.

The above section entitled “Statics” is equally applicable at an element level for elements with extra displacement shapes. The master DOFs become the nodal DOFs and the slave DOFs become the nodeless or extra DOFs.

17.6.4 Transients

The general form of the equations for transients is equations (17.2–1) and (17.2–21)):

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = \{F\} \tag{17.6–17}$$

For substructuring, an equation of the form:

$$[\hat{M}]\left\{\ddot{\hat{u}}\right\} + [\hat{C}]\left\{\dot{\hat{u}}\right\} + [\hat{K}]\{\hat{u}\} = \{\hat{F}\} \tag{17.6–18}$$

is needed. $[\hat{K}]$ and $\{\hat{F}\}$ are computed as they are for the static case (equation (17.6–8) and (17.6–9)). The computation of the reduced mass matrix is done by:

$$\begin{aligned}[\hat{M}] = {} & [M_{mm}] - [K_{ms}][K_{ss}]^{-1}[M_{sm}] - [M_{ms}][K_{ss}]^{-1}[K_{sm}] \\ & + [K_{ms}][K_{ss}]^{-1}[M_{ss}][K_{ss}]^{-1}[K_{sm}]\end{aligned} \tag{17.6–19}$$

This simplification was suggested by Guyan(14) because direct partitioning and condensation are not practical (the condensed matrices would be functions of the time derivatives of displacement and very awkward to implement). The damping matrix is handled similarly:

$$\begin{aligned}[\hat{C}] = {} & [C_{mm}] - [K_{ms}][K_{ss}]^{-1}[C_{sm}] - [C_{ms}][K_{ss}]^{-1}[K_{sm}] \\ & + [K_{ms}][K_{ss}]^{-1}[C_{ss}][K_{ss}]^{-1}[K_{sm}]\end{aligned} \tag{17.6–20}$$

Equation (17.6–16) is also used to expand the DOF values to the slave DOFs in the transient case.

17.7 Spectrum Analysis (ANTYPE,SPECTR)

Two types of spectrum analyses are supported: the deterministic response spectrum method and the non–deterministic random vibration method. Both excitation at the support and excitation away from the support are allowed. The three response spectrum methods are the single–point, multiple–point and dynamic design analysis method. The random vibration method uses the power spectral density (PSD) approach.

17.7.1 Assumptions and Restrictions

1. The structure is linear.
2. For single–point response spectrum analysis (**SPOPT**,SPRS) and dynamic design analysis method (**SPOPT**,DDAM), the structure is excited by a spectrum of known direction and frequency components, acting uniformly on all support points or on specified unsupported master degrees of freedom (DOFs).
3. For multi–point response spectrum (**SPOPT**,MPRS) and power spectral density (**SPOPT**,PSD) analyses, the structure may be excited by different input spectra at different support points or unsupported nodes. Up to ten different simultaneous input spectra are allowed.

17.7.2 Description of Analysis

The spectrum analysis capability is a separate analysis type (**ANTYPE**,SPECTR) and it must be preceeded by a mode–frequency analysis. If mode combinations are needed, the required modes must also be expanded, as described in Section 17.3.

The four options available are the single–point response spectrum method (**SPOPT**,SPRS), the dynamic design analysis method (**SPOPT**,DDAM), the random vibration method (**SPOPT**,PSD) and the multiple–point response spectrum method (**SPOPT**,MPRS). Each option is discussed in detail subsequently.

17.7.3 Single–Point Response Spectrum (**SPOPT**,SPRS)

Both excitation at the support (base excitation) and excitation away from the support (force excitation) are allowed. Table 17.7.1 summarizes these options as well as the input associated with each.

Table 17.7.1 Types of Spectrum Loading

	Excitation Option	
	Excitation at Support	**Excitation Away From Support**
Spectrum Input	Response spectrum table (**FREQ** and **SV** commands)	Amplitude multiplier table (**FREQ** and **SV** commands)
Orientation of Load	Direction vector, input on **SED** command	X, Y, or Z direction at each node, selected by FX, FY, or FZ on **F** command
Distribution of Loads	Constant on all support points	Amplitude in X, Y, or Z directions selected by *VALUE* on **F** command
Type of Input	Velocity Acceleration Displacement	Force
KSV on **SVTYP** command	0 2 3,4	1

Damping

Damping is evaluated for each mode and is defined as:

$$\xi_i' = \frac{\beta\omega_i}{2} + \xi_c + \frac{\sum\limits_{m=1}^{NMAT} \xi_m E_m^s}{\sum\limits_{m=1}^{NMAT} E_m^s} + \xi_i \qquad (17.7\text{–}1)$$

where:

ξ_i' = effective damping ratio for mode i

β = input quantity *VALUE*, **BETAD** command

ω_i = undamped natural circular frequency of the i^{th} mode

ξ_c = input quantity *RATIO*, **DMPRAT** command

ξ_m = damping ratio for material m (input quantity DAMP on **MP** command)

E_m^s = $\frac{1}{2}\{\phi_i\}^T[K_m]\{\phi_i\}$ = strain energy

$\{\phi_i\}$ = displacement vector for mode i

$[K_m]$ = stiffness matrix of part of structure of material m

ξ_i = modal damping ratio of mode i (**MDAMP** command)

Note that the material dependent damping contribution is computed in the modal expansion phase, so that this damping contribution must be included there.

Participation Factors and Mode Coefficients

The participation factors for the given excitation direction are defined as:

$$\gamma_i = \{\phi\}_i^T[M]\{D\} \quad \text{for the base excitation option} \tag{17.7–2}$$

$$\gamma_i = \{\phi\}_i^T\{F\} \quad \text{for the force excitation option} \tag{17.7–3}$$

where:

γ_i = participation factor for the ith mode

$\{\phi\}_i$ = eigenvector normalized using equation (17.3–6) (*Nrmkey* on the **MODOPT** command has no effect)

{D} = vector describing the excitation direction (see equation (17.7–4))

{F} = input force vector

The vector describing the excitation direction has the form:

$$\{D\} = [T]\{e\} \tag{17.7–4}$$

where:

$\{D\} = \left[D_1^a \; D_2^a \; D_3^a \; . \; . \; .\right]^T$

D_j^a = excitation at DOF j in direction a; a may be either X, Y, Z, or rotations about one of these axes.

$$[T] = \begin{bmatrix} 1 & 0 & 0 & 0 & (Z - Z_o) & -(Y - Y_o) \\ 0 & 1 & 0 & -(Z - Z_o) & 0 & (X - X_o) \\ 0 & 0 & 1 & (Y - Y_o) & -(X - X_o) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

X, Y, Z = global Cartesian coordinates of a point on the geometry

X_o, Y_o, Z_o = global Cartesian coordinates of point about which rotations are done (reference point)

and {*e*} = six possible unit vectors

We can calculate the statically equivalent actions at j due to rigid–body displacements of the reference point using the concept of translation of axes [T].

For spectrum analysis, the D^a values may be determined in one of two ways:

1. For D values based on the **SED** command:

$$D_X = \frac{S_X}{B} \tag{17.7–5}$$

$$D_Y = \frac{S_Y}{B} \tag{17.7–6}$$

$$D_Z = \frac{S_Z}{B} \tag{17.7–7}$$

where: S_X, S_Y, S_Z = input quantities *SEDX*, *SEDY*, and *SEDZ*, respectively, on **SED** command

B = $\sqrt{(S_X)^2 + (S_Y)^2 + (S_Z)^2}$

2. or, for D values based on the **SED** and **ROCK** command:

$$D_X = S_X + R_X \tag{17.7–8}$$

$$D_Y = S_Y + R_Y \tag{17.7–9}$$

$$D_Z = S_Z + R_Z \tag{17.7–10}$$

R is defined by:

$$\begin{Bmatrix} R_X \\ R_Y \\ R_Z \end{Bmatrix} = \begin{Bmatrix} C_X \\ C_Y \\ C_Z \end{Bmatrix} \times \begin{Bmatrix} r_X \\ r_Y \\ r_Z \end{Bmatrix} \tag{17.7–11}$$

where: C_X, C_Y, C_Z = input quantities *OMX*, *OMY*, and *OMZ*, respectively, on **ROCK** command

× = vector cross product operator

r_X = $X^n - L_X$

r_Y = $Y^n - L_Y$

r_Z = $Z^n - L_Z$

X^n, Y^n, Z^n = coordinate of node n

L_X, L_Y, L_Z = location of center of rotation (input quantities *CGX*, *CGY*, and *CGZ* on **ROCK** command)

The displacement, velocity or acceleration vector for each mode is computed from the eigenvector by using a “mode coefficient”:

$$\{r\}_i = \omega_i^m A_i\{\phi\}_i \qquad (17.7\text{–}12)$$

where: $m = \begin{cases} 0 & \text{if label = DISP} \\ 1 & \text{if label = VELO} \\ 2 & \text{if label = ACEL} \end{cases}$

label = third field on the mode combination commands (**SRSS**, **CQC**, **GRP**, **DSUM**, **NRLSUM**)

A_i = mode coefficient (see below)

The mode coefficient is computed in five different ways, depending on the type of excitation (**SVTYP** command).

1. For **SVTYP**, 0 (velocity excitation of base)

$$A_i = \frac{S_{vi}\,\gamma_i}{\omega_i} \qquad (17.7\text{–}13)$$

where: S_{vi} = spectral velocity for the ith mode (obtained from the input velocity spectrum at frequency f_i and effective damping ratio ξ_i')

f_i = ith natural frequency (cycles per unit time = $\frac{\omega_i}{2\pi}$)

ω_i = ith natural circular frequency (radians per unit time)

2. For **SVTYP**, 1 (force excitation)

$$A_i = \frac{S_{fi}\,\gamma_i}{\omega_i^{\,2}} \qquad (17.7\text{–}14)$$

where: S_{fi} = spectral force for the ith mode (obtained from the input amplitude multiplier table at frequency f_i and effective damping ratio ξ_i').

3. For **SVTYP**, 2 (acceleration excitation of base)

$$A_i = \frac{S_{ai}\,\gamma_i}{\omega_i^{\,2}} \qquad (17.7\text{–}15)$$

where: S_{ai} = spectral acceleration for the ith mode (obtained from the input acceleration response spectrum at frequency f_i and effective damping ratio ξ_i').

4. For **SVTYP**, 3 (displacement excitation of base)

$$A_i = S_{ui}\,\gamma_i \qquad (17.7\text{–}16)$$

where: S_{ui} = spectral displacement for the i^{th} mode (obtained from the input displacement response spectrum at frequency f_i and effective damping ratio ξ_i').

5. For **SVTYP**, 4 (power spectral density (PSD), Vanmarcke(34))

$$A_i = \frac{\gamma_i}{\omega_i^2}\left[S_{pi}\,\omega_i\left(\frac{\pi}{4\xi} - 1\right) + \int_o^{\omega_i} S_p\,d\omega\right]^{\frac{1}{2}} \qquad (17.7\text{–}17)$$

where: S_{pi} = power spectral density for the i^{th} mode (obtained from the input PSD spectrum at frequency f_i and effective damping ratio ξ_i')

ξ = damping ratio (input quantity *RATIO*, **DMPRAT** command, defaults to .01)

The integral in equation (17.7–17) is approximated as:

$$\int_o^{\omega_i} S_p\,d\omega = \sum_{j=1}^{L_i} S_{pj}\;\Delta f \qquad (17.7\text{–}18)$$

where: L_i = f_i (in integer form)

S_{pj} = power spectral density evaluated at frequency (f) equal to j (in real form)

Δf = effective frequency band for f_i = 1.

When S_{vi}, S_{fi}, S_{ai}, S_{ui}, or S_{pi} are needed between input frequencies, log–log interpolation is done in the space as defined.

The spectral values and the mode coefficients output in the “RESPONSE SPECTRUM CALCULATION SUMMARY” table are evaluated at the input curve with the lowest damping ratio, not at the effective damping ratio ξ_i'.

Combination of Modes

The modal displacements, velocity and acceleration (equation (17.7–12)) may be combined in different ways to obtain the response of the structure. For all excitations but the PSD this would be the maximum response, and for the PSD excitation, this would be the 1–σ (standard deviation) relative response. The response includes dof response as well as element results and reaction forces if computed in the expansion operations (*Elcalc* = YES on the **MXPAND** command).

In the case of **SPOPT**,SPRS or **SPOPT**,DDAM options of the spectrum analysis (**ANTYPE**,SPEC), it is possible to expand only those modes whose significance factor exceeds the significant threshold value (SIGNIF value on **MXPAND** command). Note that the mode coefficients must be available at the time the modes are expanded.

Only those modes having a significant amplitude (mode coefficient) are chosen for mode combination. A mode having a coefficient of greater than the SIGNIF value (mode combination commands **SRSS**, **CQC**, **GRP**, **DSUM**, **NRLSUM** and **PSDCOM**) of the maximum mode coefficient (all modes are scanned) is considered significant.

The spectrum option provides five options for the combination of modes. They are:

a. Complete Quadratic Combination Method (**CQC**)

b. Grouping Method (**GRP**)

c. Double Sum Method (**DSUM**)

d. SRSS Method (**SRSS**)

e. NRL–SUM Method (**NRLSUM**)

These methods generate coefficients for the combination of mode shapes. This combination is done by a generalization of the method of the square root of the sum of the squares which has the form:

$$R_a = \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \epsilon_{ij} \; R_i \; R_j \right]^{\frac{1}{2}} \qquad (17.7\text{–}19)$$

where:

R_a = total modal response

N = total number of expanded modes

ϵ_{ij} = coupling coefficient. The value of ϵ_{ij} = 0.0 implies modes i and j are independent and approaches 1.0 as the dependency increases

R_i = $A_i\Psi_i$ = modal response in the ith mode (equation (17.7–12))

R_j = $A_j\Psi_j$ = modal response in the jth mode

A_i = mode coefficient for the ith mode

A_j = mode coefficient for the jth mode

Ψ_i = the ith mode shape

Ψ_j = the jth mode shape

Ψ_i and Ψ_j may be the dof response, reactions, or stresses. The dof response, reactions, or stresses may be displacement, velocity or acceleration depending on the selected “label” on the mode combination command **SRSS**, **CQC**, **DSUM**, **GRP** or **NRLSUM**.

The mode combination instructions are written to File.MCOM by the mode combination command. Inputting this file in POST1 automatically performs the mode combination.

a. Complete Quadratic Combination Method

This method, accessed with **CQC** command, is based on Wilson, et al(65).

$$R_a = \left[\left| \sum_{i=1}^{N} \sum_{j=i}^{N} k \, \epsilon_{ij} \, R_i \, R_j \right| \right]^{\frac{1}{2}} \qquad (17.7\text{–}20)$$

where:

$$k = \begin{cases} 1 \text{ if } i = j \\ 2 \text{ if } i \neq j \end{cases}$$

$$\epsilon_{ij} = \frac{8\left(\xi'_i \xi'_j\right)^{\frac{1}{2}} \left(\xi'_i + r\xi'_j\right) r^{3/2}}{\left(1 - r^2\right)^2 + 4\xi'_i \, \xi'_j \, r\left(1 + r^2\right) + 4\left(\xi'^{\,2}_i + \xi'^{\,2}_j\right) r^2}$$

$$r = \omega_j / \omega_i$$

b. Grouping Method

This method, accessed with **GRP** command, is from the NRC Regulatory Guide(41). For this case, equation (17.7–19) specializes to:

$$R_a = \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \epsilon_{ij} \left| R_i \, R_j \right| \right]^{\frac{1}{2}} \qquad (17.7\text{–}21)$$

where:

$$\epsilon_{ij} = \begin{cases} 1.0 \text{ if } \left| \dfrac{\omega_j - \omega_i}{\omega_i} \right| \leq 0.1 \\ 0.0 \text{ if } \left| \dfrac{\omega_j - \omega_i}{\omega_i} \right| > 0.1 \end{cases}$$

Closely spaced modes are divided into groups that include all modes having frequencies lying between the lowest frequency in the group and a frequency 10% higher. No one frequency is to be in more than one group.

c. Double Sum Method

The Double Sum Method (NRC Regulatory Guide(41)) is accessed with **DSUM** command. For this case, equation (17.7–19) specializes to:

$$R_a = \left[\sum_{i=1}^{N} \sum_{j=1}^{N} \epsilon_{ij} \left| R_i \; R_j \right| \right]^{\frac{1}{2}} \qquad (17.7–22)$$

where:

$$\epsilon_{ij} = \frac{1}{1 + \left(\dfrac{\omega_i' - \omega_j'}{\xi_i''\omega_i + \xi_j''\,\omega_j} \right)^2}$$

ω_i' = damped natural circular frequency of the ith mode

ω_i = undamped natural circular frequency of the ith mode

ξ_i'' = modified damping ratio of the ith mode

The damped natural frequency is computed as:

$$\omega_i' = \omega_i \left(1 - \left(\xi_i' \right)^2 \right)^{\frac{1}{2}} \qquad (17.7–23)$$

The modified damping ratio (ξ_i'') is defined to account for the earthquake duration time:

$$\xi_i'' = \xi_i' + \frac{2}{t_d \omega_i} \qquad (17.7–24)$$

where: t_d = earthquake duration time, fixed at 10 units of time

d. SRSS Method

The SRSS (Square Root of the Sum of the Squares) Method is accessed with **SRSS** command, is from the NRC Regulatory Guide(41). For this case, equation (17.7–19) reduces to:

$$R_a = \left[\sum_{i=1}^{N} \left(R_i \right)^2 \right]^{\frac{1}{2}} \qquad (17.7–25)$$

e. NRL–SUM Method

The NRL–SUM (Naval Research Laboratory Sum) method (O'Hara and Belsheim(107)) is accessed with **NRLSUM** command, and calculates the maximum modal response as:

$$R_a = \left| R_{a1} \right| + \left[\sum_{i=2}^{N} \left(R_{ai} \right)^2 \right]^{\frac{1}{2}} \qquad (17.7–26)$$

where: $|R_{a1}|$ = absolute value of the largest modal displacement, stress or reaction at the point

R_{ai} = displacement, stress or reaction contributions of the same point from other modes.

Reduced Mass Summary

For the reduced modal analysis, a study of the mass distribution is made. First, each row of the reduced mass matrix is summed and then output in a table entitled “Reduced Mass Distribution”. Then all UX terms of this table are summed and designated M_x^s. UY and UZ terms are handled similarly. Rotational master DOFs are not summed. M_x^s, M_y^s, and M_z^s are output as “MASS(X, Y, Z) . . .”. They are normally slightly less than the mass of the whole structure. If any of the three is more or significantly less, probably a large part of the mass is relatively close to the reaction points, rather than close to master DOFs. In other words, the master DOFs either are insufficient in number or are poorly located.

Effective Mass

The effective mass (output quantity EFFECTIVE MASS) for the ith mode (which is a function of excitation direction) is (Clough and Penzien(80)):

$$M_{ei} = \frac{\gamma_i{}^2}{\{\phi\}_i^T[M]_i\ \{\phi\}_i} \tag{17.7–27}$$

Note from equation (17.3–6) that

$$\{\phi\}_i^T\ [M]\ \{\phi\}_i = 1 \tag{17.7–28}$$

so that $M_{ei} = \gamma_i^2$. This does not apply to the force spectrum, for which the excitation is independent of the mass distribution.

17.7.4 Dynamic Design Analysis Method (SPOPT,DDAM)

For the DDAM (Dynamic Design Analysis Method) procedure (O’Hara and Belsheim(107)), modal weights in thousands of pounds (kips) are computed from the participation factor:

$$w_i = \frac{386\ \gamma_i^2}{1000} \tag{17.7–29}$$

where: w_i = modal weight in kips

386 = acceleration due to gravity (in/sec^2)

The mode coefficients are computed by:

$$A_i = \frac{S_{ai}\,\gamma_i}{\omega_i^2} \qquad (17.7\text{–}30)$$

where: S_{ai} = the greater of $\left\{\begin{array}{l} A_m \\ S_x \end{array}\right.$ (17.7–31)

A_m = input as Amin on the **ADDAM** command (defaults to 2316.0)

and where: S_x = the lesser of $\left\{\begin{array}{l} 386A \\ \omega_i V \end{array}\right.$ (17.7–32)

and where: A = spectral acceleration

$$= \left\{\begin{array}{ll} A_f A_a \dfrac{(A_b + w_i)\,(A_c + w_i)}{(A_d + w_i)^2} & \text{if } A_d \neq 0 \\ A_f A_a \dfrac{(A_b + w_i)}{(A_c + w_i)} & \text{if } A_d = 0 \end{array}\right. \qquad (17.7\text{–}33)$$

V = spectral velocity

$$= V_f V_a \frac{(V_b + w_i)}{(V_c + w_i)} \qquad (17.7\text{–}34)$$

$A_f,\ A_a,\ A_b,\ A_c,\ A_d$ = input as AF, AA, AB, AC, AD on the **ADDAM** command

$V_f,\ V_a,\ V_b,\ V_c$ = input as VF, VA, VB, VC on the **VDDAM** command

DDAM procedure is normally used with the NRL–SUM method of mode combination, which was described in the section on the single–point response spectrum. Note that unlike equation (17.3–6), O'Hara and Belsheim(107) normalize the mode shapes to the largest modal displacements. As a result, the NRL–1396 participation factors γ_i and mode coefficients A_i will be different.

17.7.5 Random Vibration Method (**SPOPT**,PSD)

The method allows multiple power spectral density (PSD) inputs (up to ten) in which these inputs can be:

1. full correlated,
2. uncorrelated, or
3. partially correlated.

The procedure is based on computing statistics of each modal response and then combining them. It is assumed that the excitations are stationary random processes.

Description of the Method

For partially correlated nodal and base excitations, the complete equations of motions are segregated into the free and the restrained (support) DOF as:

$$\begin{bmatrix}[M_{ff}] & [M_{fr}] \\ [M_{rf}] & [M_{rr}]\end{bmatrix}\begin{Bmatrix}\{\ddot{u}_f\} \\ \{\ddot{u}_r\}\end{Bmatrix}+\begin{bmatrix}[C_{ff}] & [C_{fr}] \\ [C_{rf}] & [C_{rr}]\end{bmatrix}\begin{Bmatrix}\{\dot{u}_f\} \\ \{\dot{u}_r\}\end{Bmatrix}+\begin{bmatrix}[K_{ff}] & [K_{fr}] \\ [K_{rf}] & [K_{rr}]\end{bmatrix}\begin{Bmatrix}\{u_f\} \\ \{u_r\}\end{Bmatrix}=\begin{Bmatrix}\{F\} \\ \{0\}\end{Bmatrix} \qquad (17.7–35)$$

where $\{u_f\}$ are the free DOF and $\{u_r\}$ are the restrained DOF (excited by random loading, i.e., those with unit value of displacement on **D** command). Note that the restrained DOF that are not excited are not included in equation (17.7–35) (zero displacement on **D** command). {F} is the nodal force excitation activated by a value of force on the **F** command. The value of force can be other than unity, allowing for scaling of the participation factors.

The free displacements can be decomposed into pseudo–static and dynamic parts as:

$$\{u_f\} = \{u_s\} + \{u_d\} \qquad (17.7–36)$$

The pseudo–static displacements may be obtained from equation (17.7–35) by excluding the first two terms on the left–hand side of the equation and by replacing $\{u_f\}$ by $\{u_s\}$:

$$\{u_s\} = -[K_{ff}]^{-1}[K_{fr}]\{u_r\} = [A]\{u_r\} \qquad (17.7–37)$$

in which $[A] = -[K_{ff}]^{-1}[K_{fr}]$. Physically, the elements along the i^{th} column of [A] are the pseudo–static displacements due to a unit displacement of the support DOFs excited by the i^{th} base PSD. These displacements are written as load step 2 on the .rst file. Substituting equations (17.7–37) and (17.7–36) into (17.7–35) and assuming light damping yields:

$$\begin{aligned}&[M_{ff}]\{\ddot{u}_d\} + [C_{ff}]\{\dot{u}_d\} + [K_{ff}]\{u_d\} \\ &\simeq \{F\} - \left([M_{ff}][A] + [M_{fr}]\right)\{\ddot{u}_r\}\end{aligned} \qquad (17.7–38)$$

The second term on the right–hand side of the above equation represents the equivalent forces due to support excitations.

Using the mode superposition analysis of Section 15.11 and rewriting equation (15.11–3) as:

$$\{u_d(t)\} = [\phi]\ \{y(t)\} \qquad (17.7–39)$$

the above equations are decoupled yielding:

$$\ddot{y}_j + 2\xi_j \omega_j \dot{y}_j + \omega_j^2 y_j = G_j, \quad (j = 1, 2, ..., n) \tag{17.7–40}$$

where: n = number of mode shapes chosen for evaluation (input quantity *NMODE* on **SPOPT** command)

y_j = generalized displacements

ω_j and ξ_j = natural circular frequencies and modal damping ratios

The modal loads G_j are defined by:

$$G_j = \{\Gamma_j\}^T \{\ddot{u}_r\} + \gamma_j \tag{17.7–41}$$

The modal participation factors corresponding to support excitation are given by:

$$\{\Gamma_j\} = -\left([M_{ff}]\,[A] + [M_{fr}]\right)^T \{\phi_j\} \tag{17.7–42}$$

and for nodal excitation:

$$\gamma_j = \{\phi_j\}^T \{F\} \tag{17.7–43}$$

Note that, for simplicity, equations for nodal excitation problems are developed for a single PSD table. Multiple nodal excitation PSD tables are, however, allowed in the program.

These factors are calculated as a result of the action command **PFACT** when defining base or nodal excitation cases and are written to the .psd file. Mode shapes $\{\phi_j\}$ should be normalized with respect to the mass matrix as in equation (17.3–6).

Input Power Spectral Densities

Input power spectral densities (PSD's) of the nodal and base excitations $\bar{S}_{\ell m}(\omega)$ and $\hat{S}_{\ell m}(\omega)$ are defined using several commands. **PSDUNIT** identifies the spectrum as a displacement, velocity, acceleration, pressure or force PSD, while the **PSDFRQ** and **PSDVAL** commands assign frequency– spectral value pairs to the input PSD curve(s). The relationship between multiple input spectra are described in the later subsection, "Cross Spectral Terms for Partially Correlated Input PSD's".

Response Power Spectral Densities and Mean Square Response

Using the theory of random vibrations, the response PSD's can be computed from the input PSD's with the help of transfer functions for single DOF systems H(ω) and by using mode superposition techniques (**RPSD** command in POST26). The response PSD's for i^{th} DOF are given by:

Dynamic Part:

$$S_{d_i}(\omega) = \sum_{j=1}^{n} \sum_{k=1}^{n} \phi_{ij}\phi_{ik} \left[\sum_{\ell=1}^{r_1} \sum_{m=1}^{r_1} \gamma_{\ell j}\, \gamma_{mk}\, H_j^*(\omega)\, H_k(\omega)\, \overline{S}_{\ell m}(\omega) + \sum_{\ell=1}^{r_2} \sum_{m=1}^{r_2} \Gamma_{\ell j}\, \Gamma_{mk}\, H_j^*(\omega)\, H_k(\omega)\, \hat{S}_{\ell m}(\omega) \right] \tag{17.7–44}$$

Pseudo–Static Part:

$$S_{s_i}(\omega) = \sum_{\ell=1}^{r_2} \sum_{m=1}^{r_2} A_{i\ell}\, A_{im} \left(\frac{1}{\omega^4}\, \hat{S}_{\ell m}(\omega) \right) \tag{17.7–45}$$

Covariance Part:

$$S_{sd_i}(\omega) = \sum_{j=1}^{n} \sum_{\ell=1}^{r_2} \sum_{m=1}^{r_2} \phi_{ij}\, A_{i\ell} \left(-\frac{1}{\omega^2}\, \Gamma_{mj}\, H_j(\omega)\, \hat{S}_{\ell m}(\omega) \right) \tag{17.7–46}$$

where: n = number of mode shapes chosen for evaluation (input quantity *NMODE* on **SPOPT** command)

r_1 and r_2 = number of nodal (away from support) and base PSD tables, respectively

The transfer functions for the single DOF system assume different forms depending on the type (options on the **PSDUNIT** command) of the input PSD and the type of response desired (options on the **PSDRES** command). The forms of the transfer functions for displacement as the output are listed below for different inputs.

1. Input = force or acceleration (FORC, ACEL, or ACCG):

$$H_j(\omega) = \frac{1}{\omega_j^2 - \omega^2 + i\,(2\xi_j\omega_j\omega)} \tag{17.7–47}$$

2. Input = displacement (DISP):

$$H_j(\omega) = \frac{\omega^2}{\omega_j^2 - \omega^2 + i\,(2\xi_j\omega_j\omega)} \tag{17.7–48}$$

3. Input = velocity (VELO):

$$H_j(\omega) = \frac{i\omega}{\omega_j^2 - \omega^2 + i\,(2\xi_j\omega_j\omega)} \tag{17.7–49}$$

where: ω = forcing frequency

ω_j = natural circular frequency for jth mode

i = $\sqrt{-1}$

Now, random vibration analysis can be used to show that the absolute value of the mean square response of the i^{th} free displacement is (ABS option on the **PSDRES** command):

$$\sigma_{f_i}^2 = \int_0^\infty S_{d_i}(\omega)\, d\omega + \int_0^\infty S_{s_i}(\omega)\, d\omega + 2\left|\int_0^\infty S_{sd_i}(\omega)\, d\omega\right|_{Re} \tag{17.7–50}$$

$$= \sigma_{d_i}^2 + \sigma_{s_i}^2 + 2\, C_v\,(u_{s_i}\,,\ u_{d_i})$$

where: $|\ |_{Re}$ = denotes the real part of the argument

$\sigma_{d_i}^2$ = variance of the i^{th} relative (dynamic) free displacements (REL option on the **PSDRES** command)

$\sigma_{s_i}^2$ = variance of the i^{th} pseudo–static displacements

$C_v\,(u_{s_i}\,,\ u_{d_i})$ = covariance between the static and dynamic displacements

The general formulation described above gives simplified equations for several situations commonly encountered in practice. For fully correlated nodal excitations and identical support motions, the subscripts ℓ and m would drop out from the equations (17.7–44) thru (17.7–46). When only nodal excitations exist, the last two terms in equation (17.7–50) do not apply, and only the first term within the large parentheses in equation (17.7–44) needs to be evaluated. For uncorrelated nodal force and base excitations, the cross PSD's (i.e. $\ell \neq$ m) are zero, and only the terms for which ℓ = m in equations (17.7–44) thru (17.7–46) need to be considered.

Equations (17.7–44) thru (17.7–46) can be rewritten as:

$$S_{d_i}(\omega) = \sum_{j=1}^{n} \sum_{k=1}^{n} \phi_{ij}\, \phi_{ik}\, R_{jk}(\omega) \tag{17.7–51}$$

$$S_{s_i}(\omega) = \sum_{\ell=1}^{r_2} \sum_{m=1}^{r_2} A_{i\ell}\, A_{im}\, \overline{R}_{\ell m}(\omega) \tag{17.7–52}$$

$$S_{sd_i}(\omega) = \sum_{j=1}^{n} \sum_{\ell=1}^{r_2} \phi_{ij}\, A_{i\ell}\, \hat{R}_{j\ell}(\omega) \tag{17.7–53}$$

where:

$R_{jk}(\omega), \overline{R}_{\ell m}(\omega), \hat{R}_{j\ell}(\omega)$ = modal PSD's, terms within large parentheses of equations (17.7–44) thru (17.7–46)

Closed–form solutions for piecewise linear PSD in log–log scale are employed to compute each integration in equation (17.7–50) (Chen and Ali(193) and Harichandran(194)) .

Subsequently, the variances become:

$$\sigma^2_{d_i} = \sum_{j=1}^{n} \sum_{k=1}^{n} \phi_{ij}\ \phi_{ik}\ Q_{jk} \tag{17.7–54}$$

$$\sigma^2_{s_i} = \sum_{\ell=1}^{r_2} \sum_{m=1}^{r_2} A_{i\ell}\ A_{im}\ \overline{Q}_{\ell m} \tag{17.7–55}$$

$$\sigma^2_{sd_i} = \sum_{j=1}^{n} \sum_{\ell=1}^{r_2} \phi_{ij}\ A_{i\ell}\ \hat{Q}_{j\ell} \tag{17.7–56}$$

The modal covariance matrices Q_{jk}, $\overline{Q}_{\ell m}$, and $\hat{Q}_{j\ell}$ are available in the .PSD file. Note that equations (17.7–54) thru (17.7–56) represent mode combination (**PSDCOM** command) for random vibration analysis.

The variance for stresses, nodal forces or reactions can be computed (*Elcalc* = YES on **SPOPT** (if *Elcalc* = YES on **MXPAND**)) from equations similar to (17.7–54) thru (17.7–56). If the stress variance is desired, replace the mode shapes (ϕ_{ij}) and static displacements ($A_{i\ell}$) with mode stresses ($\overline{\phi}_{ij}$) and static stresses ($\overline{A}_{i\ell}$). Similarly, if the node force variance is desired, replace the mode shapes and static displacements with mode nodal forces ($\hat{\phi}_{ij}$) and static nodal forces ($\hat{A}_{i\ell}$). Finally, if reaction variances are desired, replace the mode shapes and static displacements with mode reaction ($\tilde{\phi}_{ij}$) and static reactions ($\tilde{A}_{i\ell}$). Furthermore, the variances of the first and second time derivatives (VELO and ACEL options respectively on the **PSDRES** command) of all the quantities mentioned above can be computed using the following relations:

$$S_{\dot{u}}(\omega) = \omega^2\ S_u(\omega) \tag{17.7–57}$$

$$S_{\ddot{u}}(\omega) = \omega^4\ S_u(\omega) \tag{17.7–58}$$

Cross Spectral Terms for Partially Correlated Input PSDs

For excitation defined by more than a single input PSD, cross terms which determine the degree of correlation between the various PSDs are defined as:

$$[S(\omega)] = \begin{bmatrix} S_{11}(\omega) & C_{12}(\omega) + iQ_{12}(\omega) & C_{13}(\omega) + iQ_{13}(\omega) \\ C_{12}(\omega) - iQ_{12}(\omega) & S_{22}(\omega) & C_{23}(\omega) + iQ_{23}(\omega) \\ C_{13}(\omega) - iQ_{13}(\omega) & C_{23}(\omega) - iQ_{23}(\omega) & S_{33}(\omega) \end{bmatrix} \quad (17.7\text{–}59)$$

where: $S_{nn}(\omega)$ = input PSD spectra which are related. Defined by the **PSDVAL** command and located as table number (*TBLNO*) n

$C_{nm}(\omega)$ = cospectra which make up the real part of the cross terms. Defined by the **COVAL** command where n and m (*TBLNO1* and *TBLNO2*) identify the matrix location of the cross term

$Q_{nm}(\omega)$ = quadspectra which make up the imaginary part of the cross terms. Defined by the **QDVAL** command where n and m (*TBLNO1* and *TBLNO2*) identify the matrix location of the cross term

The normalized cross PSD function is called the coherence function and is defined as:

$$\gamma^2_{nm}(\omega) = \frac{| \ C_{nm}(\omega) - i \ Q_{nm}(\omega) \ |^2}{S_{nn}(\omega) \ S_{mm}(\omega)} \quad (17.7\text{–}60)$$

where: $0 \le \gamma^2_{nm}(\omega) \le 1$

Although the above example demonstrates the cross correlation for 3 input spectra, this matrix may range in size from 2 x 2 to 10 x 10 (i.e., maximum number of tables is 10).

For the special case in which all cross terms are zero, the input spectra are said to be uncorrelated. Note that correlation between nodal and base excitations is not allowed.

Spatial Correlation

The degree of correlation between excited nodes may also be controlled with the **PSDSPL** command. Depending upon the distance between excited nodes and the values of *RMIN* and *RMAX*, an overall excitation PSD can be constructed such that excitation at the nodes may be uncorrelated, partially correlated or fully correlated. If the distance between excited nodes is less than *RMIN*, then the two nodes are fully correlated; if the distance is greater than *RMAX*, then the two nodes are uncorrelated; if the distance lies between *RMIN* and *RMAX*, excitation is partially correlated based on the actual distance between nodes. The figure below indicates how *RMIN*, *RMAX* and the correlation are related. When the **PSDSPL** command is used, the **PFACT** command must be used with *Parcor* = SPAT. The *Excit* field on the **PFACT** command can be NODE or BASE.

For two excitation points 1 and 2, the PSD would be:

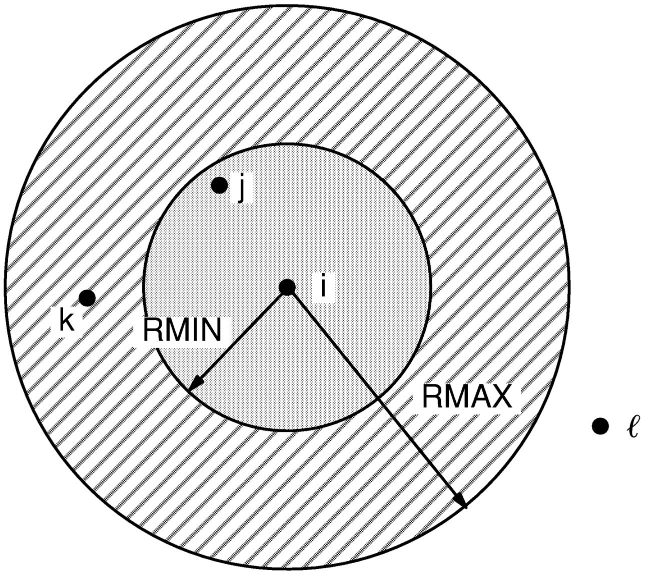
$$[S(\omega)] = S_o(\omega)\begin{bmatrix} 1 & \alpha_{12} \\ \alpha_{12} & 1 \end{bmatrix} \tag{17.7–61}$$

where:

$$\alpha_{12} = \begin{cases} \dfrac{RMAX - D_{12}}{RMAX - RMIN} & \text{if } RMIN < D_{12} < RMAX \\ 1 & \text{if } D_{12} \leq RMIN \\ 0 & \text{if } D_{12} \geq RMAX \end{cases}$$

D_{12} = distance between the two excitation points 1 and 2

$S_o(\omega)$ = basic input PSD (**PSDVAL** and **PSDFRQ** commands)



Node i excitation is fully correlated with node j excitation
Node i excitation is partially correlated with node k excitation
Node i excitation is uncorrelated to node ℓ excitation

Figure 17.7–1 Sphere of Influence Relating Spatially Correlated PSD Excitation

Wave Propagation

To include wave propagation effects (**PSDWAV** command) of a random loading, the excitation PSD is constructed as:

$$S_{\ell m}(\omega) = S_o(\omega)\left(e^{-i\omega d_{\ell m}}\right) \qquad (17.7\text{–}62)$$

where: $d_{\ell m} = \dfrac{\lfloor D_{\ell m} \rfloor \{V\}}{V^2}$ = delay

$\{D_{\ell m}\}$ = $\{x_m\} - \{x_\ell\}$ = separation vector between excitations points ℓ and m

$\{V\}$ = velocity of propagation of the wave (input quantities *VX*, *VY* and *VZ* on **PSDWAV** command)

$\{x_\ell\}$ = nodal coordinates of excitation point ℓ

More than one simultaneous wave or spatially correlated PSD inputs are permitted, in which case the input excitation [S(ω)] reflects the influence of two or more uncorrelated input spectra. In this case, partial correlation among the basic input PSD's is not currently permitted. When the **PSDWAV** command is used, the **PFACT** command must be used with *Parcor* = WAVE. The *Excit* field on the **PFACT** command can be NODE or BASE.

17.7.6 Multi–Point Response Spectrum Method (**SPOPT**,MPRS)

The response spectrum analysis due to multi–point support and nodal excitations allows up to ten different input spectra tables which are assumed to be unrelated (uncorrelated) to each other.

Most of the ingredients for performing multi–point response spectrum analysis are already developed in the previous subsection of the random vibration method. Assuming that the participation factors, $\Gamma_{j\ell}$, for the ℓ^{th} input spectrum table have already been computed (by equation (17.7–42), for example), the mode coefficients for the ℓ^{th} table are obtained as:

$$B_{j\ell} = \Gamma_{j\ell}\, S_{j\ell} \qquad (17.7\text{–}63)$$

where: $S_{j\ell}$ = interpolated input response spectrum for the ℓ^{th} table at the j^{th} natural frequency (defined by the **PSDFRQ**, **PSDVAL** and **PSDUNIT** commands)

The mode coefficients are combined using SRSS:

$$A_j = \left(B_{j1}^2 + B_{j2}^2 + B_{j3}^2 + \ldots\right)^{\frac{1}{2}} \qquad (17.7\text{–}64)$$

The mode shapes, mode stresses, etc. are multiplied by the mode coefficients to compute modal quantities, which can then be combined with the help of any of the available mode combination techniques (SRSS, CQC, Double Sum, Grouping or NRL–SUM Method), as described in the previous section on the single–point response spectrum method.

Chapter 18
Pre and Postprocessing Tools

ANSYS Theory Reference
Chapter 18 – Table of Contents

18.1 Integration and Differentiation Procedures

18.1.1 Single Integration Procedure

(accessed with *VOPER command, INT1 operation; similar capability is in POST26, INT1 command)

Given two vectors Y (parameter *Par1*) and X (parameter *Par2*), and an integration constant C_1 (input quantity *CON1*), parameter *ParR* is replaced by the accumulated integral of Y over X as follows:

$$\text{Set } Y_1^* = C_1 \quad \text{(for example, this would be the initial displacement of X represents time and Y represents velocity)} \tag{18.1–1}$$

Then for each remaining point in the vector, set:

$$Y_n^* = Y_{n-1}^* + \frac{1}{2}\left(Y_n + Y_{n-1}\right)\left(X_n - X_{n-1}\right) \quad n = 2, L \tag{18.1–2}$$

where: Y_n^* = integrated value of Y up to point n in the vector
L = length of the vectors

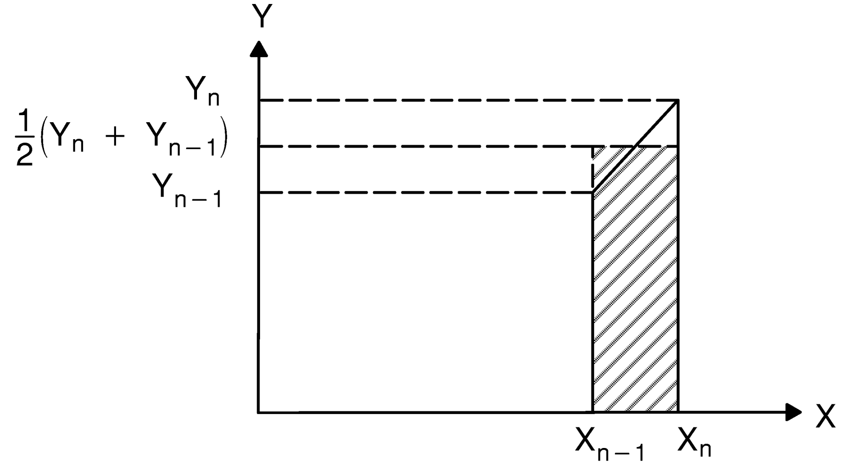


Figure 18.1–1 Integration Procedure

18.1.2 Double Integration Procedure

(accessed with *VOPER command, INT2 operation)

Given two vectors Y (parameter *Par1*) and X (parameter *Par2*), integration constants C_1 and C_2, (input quantities *CON1* and *CON2*) set:

$$Y_1^{*} = C_1 \quad \text{(for example, this would be the initial velocity if X represents time and Y represents acceleration)} \tag{18.1–3}$$

$$Y_1^{**} = C_2 \quad \text{(for example, this would be the initial displacement if X represents time and Y represents acceleration)} \tag{18.1–4}$$

Then, for each remaining point in the vector,

$$Y_n^{*} = Y_{n-1}^{*} + \frac{1}{2}\left(Y_n + Y_{n-1}\right)\left(X_n - X_{n-1}\right) \tag{18.1–5}$$

$$\begin{aligned} Y_n^{**} &= Y_{n-1}^{**} + \left(X_n - X_{n-1}\right) Y_{n-1}^{*} \\ &+ \frac{1}{6}\left(X_n - X_{n-1}\right)^2 \left(2Y_{n-1} + Y_n\right) \end{aligned} \tag{18.1–6}$$

18.1.3 Differentiation Procedure

(accessed with *VOPER Command, DER1 Operation; similar capability is in POST26 DERIV command)

Given two vectors Y (parameter *Par1*) and X (parameter *Par2*), the derivative is found by averaging the slopes of two adjacent intervals (central difference procedure):

$$\dot{Y}_{n+1} = \frac{\frac{Y_{n+2} - Y_{n+1}}{X_{n+2} - X_{n+1}}\left(X_{n+1} - X_n\right) + \frac{Y_{n+1} - Y_n}{X_{n+1} - X_n}\left(X_{n+2} - X_{n+1}\right)}{X_{n+2} - X_n} \tag{18.1–7}$$

A constant second derivative is assumed for the starting and ending intervals.

$$\dot{Y}_1 = \frac{Y_2 - Y_1}{X_2 - X_1} \tag{18.1–8}$$

$$\dot{Y}_L = \frac{Y_L - Y_{L-1}}{X_L - X_{L-1}} \tag{18.1–9}$$

18.1.4 Double Differentiation Procedure

(accessed by *VOPER Command, DER2 Operation)

This is performed by simply repeating the differentiation procedure reported above.

18.2 Fourier Coefficient Evaluation

Fourier coefficients may be evaluated using the ***MFOURI** command. Given two vectors defining data points to be fit (parameters *CURVE* and *THETA*) and two more vectors defining which terms of the trigonometric series are desired to be computed (parameters *MODE* and *ISYM*), parameter *COEFF* has those desired coefficients. The curve fitting cannot be perfect, as there are more data than unknowns. Thus, an error R_i will exist at each data point:

$$\begin{aligned}
R_1 &= A_1 + A_2 \cos\theta_1 + A_3 \sin\theta_1 + A_4 \cos 2\theta_1 + A_5 \sin 2\theta_1 \\
&\quad + A_6 \cos 3\theta_1 + A_7 \sin 3\theta_1 + \dots + A_L\, F(M\theta_1) - C_1 \\[2ex]
R_2 &= A_1 + A_2 \cos\theta_2 + A_3 \sin\theta_2 + A_4 \cos 2\theta_2 + A_5 \sin 2\theta_2 \\
&\quad + A_6 \cos 3\theta_2 + A_7 \sin 3\theta_2 + \dots + A_L\, F(M\theta_2) - C_2 \\
&\vdots \\
R_i &= A_1 + A_2 \cos\theta_i + A_3 \sin\theta_i + A_4 \cos 2\theta_i + A_5 \sin 2\theta_i \\
&\quad + A_6 \cos 3\theta_i + A_7 \sin 3\theta_i + \dots + A_L\, F(M\theta_i) - C_i \\
&\vdots \\
R_m &= A_1 + A_2 \cos\theta_m + A_3 \sin\theta_m + A_4 \cos 2\theta_m + A_5 \sin 2\theta_m \\
&\quad + A_6 \cos 3\theta_m + A_7 \sin 3\theta_m + \dots + A_L\, F(M\theta_m) - C_m
\end{aligned} \tag{18.2–1}$$

where:
R_i = error term (residual) associated with data point i
A = desired coefficients of Fourier series (parameter *COEFF*)
θ_i = angular location of data points i (parameter *THETA*)
L = number of terms in Fourier series
F = sine or cosine, depending on ISYM (parameter *ISYM*)
M = multiplier on θ_i (parameter *MODE*)
C_i = value of data point i (parameter *CURVE*)
m = number of data points (length of *CURVE* parameter array)

Equations (18.2–1) can be reduced to matrix form as:

$$\{R\}_{m,1} = [G]_{m,L}\ \{A\}_{L,1} - \{C\}_{L,1} \tag{18.2–2}$$

where:

$\{R\}$ = vector of error terms

$[G]$ = matrix of sines and cosines, evaluated at the different data points

$\{A\}$ = vector of desired coefficients

$\{C\}$ = vector of data points

Note that m > L. If m = L, the coefficients would be uniquely determined with {R} = O and equation (18.2–2) being solved for {A} by direct inversion.

The method of least squares is used to determine the coefficients {A}. This means that $\sum_{i=1}^{m} (R_i)^2$ is to be minimized. The minimization is represented by

$$\frac{\partial \sum_{i=1}^{m} \left(R_i\right)^2}{\partial\ A_j} = 0 \tag{18.2–3}$$

where A_j is the jth component of {A}. Note that

$$\{R\}^T \{R\} = \sum_{i=1}^{m} \left(R_i\right)^2 \tag{18.2–4}$$

The form on the left–hand side of equation (18.2–4) is the more convenient to use. Performing this operation on equation (18.2–2),

$$\{R\}^T \{R\} = \{A\}^T [G]^T [G] \{A\} - 2\{A\}^T [G]^T \{C\} + \{C\}^T \{C\} \tag{18.2–5}$$

Minimizing this with respect to $\{A\}^T$ (equation (18.2–3)), it may be shown that:

$$\{0\} = 2[G]^T [G] \{A\} - 2[G]^T \{C\} \tag{18.2–6}$$

or

$$[G]^T [G] \{A\} = [G]^T \{C\} \tag{18.2–7}$$

Equation (18.2–7) is known as the “normal equations” used in statistics. Finally,

$$\{A\} = \left([G]^T [G]\right)^{-1} [G]^T \{C\} \tag{18.2–8}$$

$[G^T]$ could not have been “cancelled out” of equation (18.2–7) because it is not a square matrix. However, $[G]^T[G]$ is square.

In spite of the orthogonal nature of a trigonometric series, the value of each computed coefficient is dependent on the number of terms requested because of the least squares fitting procedure which takes place at the input data points. Terms of a true Fourier series are evaluated not by a least squares fitting procedure, but rather by the integration of a continuous function (e.g. Euler formulas, p. 469 of Kreyszig(23)).

18.3 Statistical Procedures

18.3.1 Mean, Covariance, Correlation Coefficient

The ***MOPER** command is used to calculate the mean, variance, covariance, and correlation coefficients of a multiple subscripted parameter. Refer to Kreyszig(162) for the basis of the following formulas. All operations are performed on columns to conform to the ANSYS database structure. The covariance, variance and mean operations are initiated when COVAR is inserted in the *Oper* field of the ***MOPER** command. In the ANSYS program, the covariance is a measure of the association between columns.

The following notation is used:

$[x]$ = starting matrix
i = row index of first array parameter matrix
j = column index of first array parameter matrix
m = number of rows in first array parameter matrix
n = number of columns in first array parameter matrix
subscripts s, t = selected column indices
$[S]$ = covariance matrix n x n
$[c]$ = correlation matrix n x n
σ_s^2 = variance

The mean of a column is:

$$\bar{x}_j = \sum_{i=1}^{m} \frac{x_{ij}}{m} \tag{18.3–1}$$

The covariance of the columns s and t is:

$$S_{st} = \sum_{i=1}^{m} \frac{(x_{is} - \bar{x}_s)\ (x_{it} - \bar{x}_t)}{m - 1} \tag{18.3–2}$$

The variance, σ_s^2, of column s is the diagonal term S_{ss} of the covariance matrix [S]. The equivalent common definition of variance is:

$$\sigma_s^2 = \sum_{i=1}^{m} \frac{(x_{is} - \overline{x}_s)^2}{m - 1} \tag{18.3–3}$$

The correlation coefficient is a measure of the independence or dependence of one column to the next. The correlation and mean operations are initiated when CORR is inserted in the *Oper* field of the ***MOPER** command and is based on Hoel(163).

Correlation coefficient:

$$C_{st} = \frac{S_{st}}{\sqrt{S_{ss}}\ \sqrt{S_{tt}}} \tag{18.3–4}$$

The value S of the terms of the coefficient matrix range from –1.0 to 1.0 where:

–1.0	=	fully inversely related
0.0	=	fully independent
1.0	=	fully directly related

18.3.2 Random Samples of a Uniform Distribution

RAND in the *Func* field on the ***VFILL** command fills a vector with a random sample of real numbers based on a uniform distribution with given lower and upper bounds (see Figure 18.3–1):

$$f(x) = 1.0 \qquad a \le x \le b \tag{18.3–5}$$

where: a = lower bound (input as CON1 on ***VFILL** command)

b = upper bound (input as CON2 on ***VFILL** command)

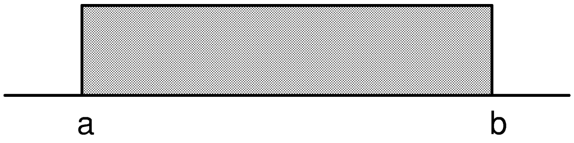


Figure 18.3–1 Uniform Density

The numbers are generated using the URN algorithm of Swain and Swain(161). The initial seed numbers are hard coded into the routine.

18.3.3 Random Samples of a Gaussian Distribution

GDIS in the *Func* field on the ***VFILL** command fills a vector with a random sample of real numbers based on a Gaussian distribution with a known mean and standard deviation.

First, random numbers P(x), with a uniform distribution from 0.0 to 1.0, are generated using a random number generator. These numbers are used as probabilities to enter a cumulative standard normal probability distribution table (Abramowitz and Stegun(160)), which can be represented by Figure 18.3–2 or the Gaussian distribution function:

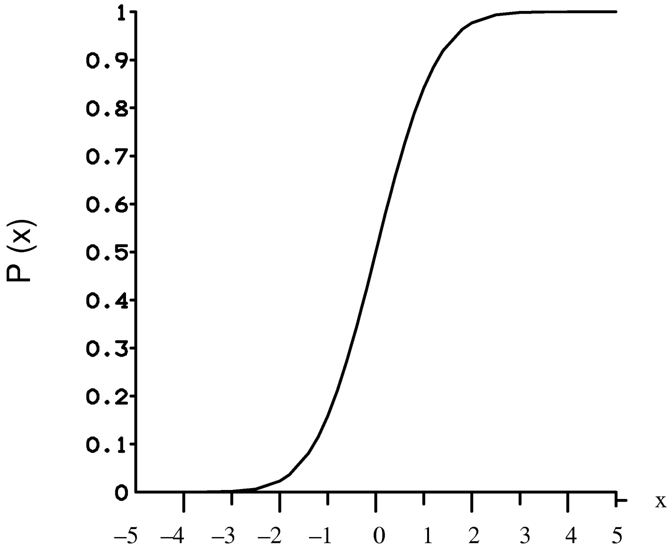


Figure 18.3–2 Cumulative Probability Function

$$P(x) = \int_{-\infty}^{x} f(t)\ dt \tag{18.3–6}$$

$$= \text{no closed form}$$

where f(t) = Gaussian density function

The table maps values of P(x) into values of x, which are standard Gaussian distributed random numbers from –5.0 to 5.0, and satisfy the Gaussian density function (Figure 18.3–3):

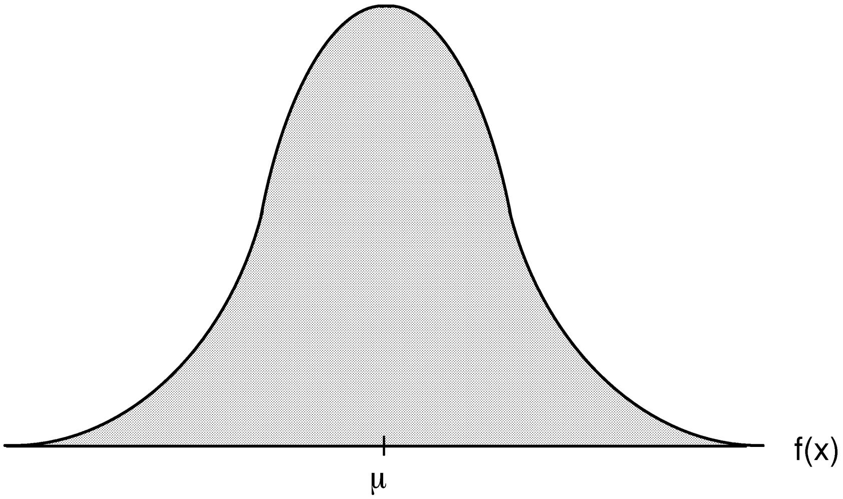


Figure 18.3–3 Gaussian Density

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2 / 2\sigma^2} \qquad -\infty < x < \infty \tag{18.3–7}$$

where:

μ = mean (input as CON1 on ***VFILL** command)

σ = standard deviation (input as CON2 on ***VFILL** command)

The x values are transformed into the final Gaussian distributed set of random numbers, with the given mean and standard deviation, by the transformation equation:

$$z = \sigma x + \mu \tag{18.3–8}$$

18.3.4 Random Samples of a Triangular Distribution

TRIA in the *Func* field on the ***VFILL** command fills a vector with a random sample of real numbers based on a triangular distribution with a known lower bound, peak value location, and upper bound.

First, random numbers P(x) are generated as in the Gaussian example. These P(x) values (probabilities) are substituted into the triangular cumulative probability distribution function:

$$P(x) = \begin{cases} 0 & \text{if } x < a \\ \dfrac{(x - a)^2}{(b - a)\ (c - a)} & \text{if } a \le x \le c \\ 1 - \dfrac{(b - x)^2}{(b - a)\ (b - c)} & \text{if } c < x \le b \\ 1 & \text{if } b < x \end{cases} \qquad (18.3\text{–}9)$$

where: a = lower bound (input as CON1 on ***VFILL** command)
c = peak location (input as CON2 on ***VFILL** command)
b = upper bound (input as CON3 on ***VFILL** command)

which is solved for values of x. These x values are random numbers with a triangular distribution, and satisfy the triangular density function (Figure 18.3–4):

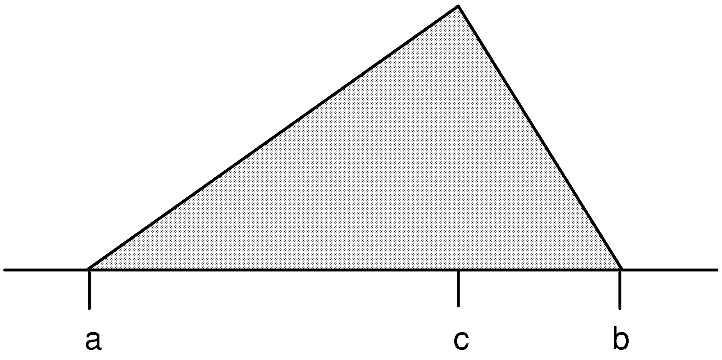


Figure 18.3–4 Triangular Density

$$f(x) = \begin{cases} \dfrac{2\ (x - a)}{(b - a)\ (c - a)} & \text{if } a \le x \le c \\ \dfrac{2\ (b - x)}{(b - a)\ (b - c)} & \text{if } c < x \le b \\ 0 & \text{otherwise} \end{cases} \qquad (18.3\text{–}10)$$

18.3.5 Random Samples of a Beta Distribution

BETA in the *Func* field on the ***VFILL** command fills a vector with a random sample of real numbers based on a beta distribution with known lower and upper bounds and α and β parameters.

First, random numbers P(x) are generated as in the Gaussian example. These random values are used as probabilities to enter a cumulative beta probability distribution table, generated by the program. This table can be represented by a curve similar to (Figure 18.3–2), or the beta cumulative probability distribution function:

$$P(x) = \int_{-\infty}^{x} f(t)\ dt \qquad (18.3\text{–}11)$$

$$= \text{no closed form}$$

The table maps values of P(x) into x values which are random numbers from 0.0 to 1.0. The values of x have a beta distribution with given α and β values, and satisfy the beta density function (Figure 18.3–5):

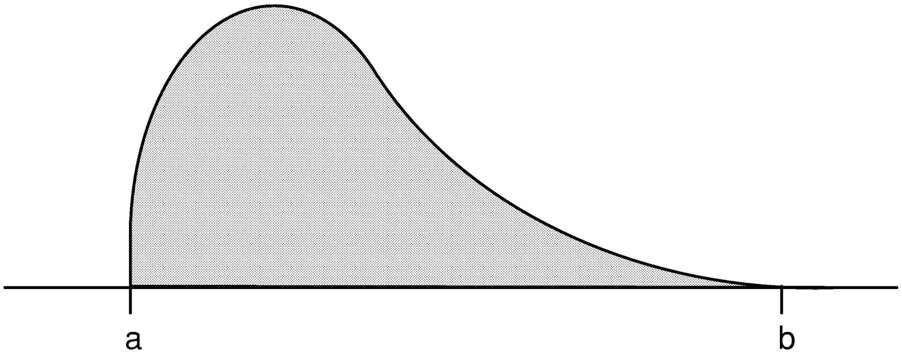


Figure 18.3–5 Beta Density

$$f(x) = \begin{cases} \dfrac{x^{\alpha-1}\ (1 - x)^{\beta-1}}{B\ (\alpha,\ \beta)} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \qquad (18.3\text{–}12)$$

where:

a	=	lower bound (input as CON1 on ***VFILL** command)
b	=	upper bound (input as CON2 on ***VFILL** command)
α	=	alpha parameter (input as CON3 on ***VFILL** command)
β	=	beta parameter (input as CON4 on ***VFILL** command)
B (α, β)	=	beta function
	=	$\int_{0}^{1} t^{\alpha-1}\ (1 - t)^{\beta-1}\ dt \qquad \text{for } \alpha > 0, \beta > 0$
f(t)	=	beta density function

The x values are transformed into the final beta distributed set of random numbers, with given lower and upper bounds, by the transformation equation:

$$z = a + (b - a)\ x \qquad (18.3\text{–}13)$$

18.3.6 Random Samples of a Gamma Distribution

GAMM in the *Func* field on the ***VFILL** command fills a vector with a random sample of real numbers based on a gamma distribution with a known lower bound for α and β parameters.

First, random numbers P(x) are generated as in the Gaussian example. These random values are used as probabilities to enter a cumulative gamma probability distribution table, generated by the program. This table can be represented by a curve similar to Figure 18.3–6, or the gamma cumulative probability distribution function:

$$P(x) = \int_{-\infty}^{x} f(t)\ dt \qquad (18.3\text{–}14)$$

$$= \text{no closed form}$$

where: f(t) = gamma density function.

The table maps values of P(x) into values of x, which are random numbers having a gamma distribution with given α and β values, and satisfy the gamma distribution density function (Figure 18.3–6):

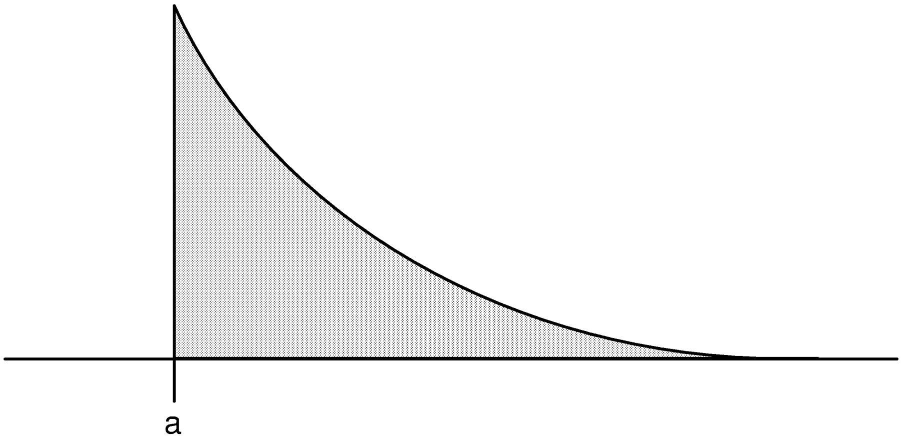


Figure 18.3–6 Gamma Density

$$f(x) = \begin{cases} \dfrac{\beta^{-\alpha}\ x^{\alpha-1}\ e^{-x/\beta}}{\Gamma(\alpha)} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (18.3\text{–}15)$$

where: $\Gamma(\alpha+1) = \int_{o}^{\infty} t^{\alpha}\ e^{-t}\ dt \qquad \alpha \geq 0$

α = alpha parameter of gamma function (input as CON2 on ***VFILL** command)

β = beta parameter of gamma density function (input as CON3 on ***VFILL** command)

a = lower bound (input as CON1 on ***VFILL** command)

The x values are relocated relative to the given lower bound by the transformation equation:

$$z = a + x \qquad (18.3\text{–}16)$$

Chapter 19
Postprocessing

ANSYS Theory Reference
Chapter 19 – Table of Contents

19.1 POST1 — Derived Nodal Data Processing

19.1.1 Derived Nodal Data Computation

The computation of derived data (data derived from nodal unknowns) is discussed in Chapters 3 through 8. Derived nodal data is available for solid and shell elements (except SHELL51 and SHELL61). Available data include stresses, strains, thermal gradients, thermal fluxes, pressure gradients, electric fields, electric flux densities, magnetic field intensities, magnetic flux densities, and magnetic forces. Structural nonlinear data is processed in a similar fashion and includes equivalent stress, stress state ratio, hydrostatic pressure, accumulated equivalent plastic strain, plastic state variable, and plastic work.

POST1 averages the component tensor or vector data at nodes used by more than one element.

$$\sigma_{ik} = \frac{\sum_{j=1}^{N_k} \sigma_{ijk}}{N_k} \qquad (19.1\text{–}1)$$

where:

σ_{ik} = average derived data component i at node k

σ_{ijk} = derived data component i of element j at node k

N_k = number of elements connecting to node k

Combining principal tensor data (principal stress, principal strain) or vector magnitudes at the nodes may either be computed using the averaged component data (*KEY* = 0, **AVPRIN** command):

$$\sigma_{ck} = f\left(\sigma_{ik}\right) \qquad (19.1\text{–}2)$$

where:

$f\left(\sigma_{ik}\right)$ = function to compute principal data from component data as given in Chapters 3 through 8.

or be directly averaged (*KEY* = 1, **AVPRIN** command):

$$\sigma_{ck} = \frac{\sum_{j=1}^{k} \sigma_{cjk}}{N_k} \qquad (19.1\text{–}3)$$

where: σ_{ck} = averaged combined principal data at node k

σ_{cjk} = combined principal data of element j at node k

19.2 POST1 — Vector and Surface Operations

19.2.1 Vector Operations

The dot product of two vectors $\{A\} \left(= A_x\hat{i} + A_y\hat{j} + A_z\hat{k}\right)$ and $\{B\} \left(= B_x\hat{i} + B_y\hat{j} + B_z\hat{k}\right)$ is provided with the **VDOT** command as:

$$\{A\} \cdot \{B\} = A_xB_x + A_yB_y + A_zB_z \tag{19.2–1}$$

The cross product of two vectors {A} and {B} is also provided with the **VCROSS** command as:

$$\{A\} \text{ X } \{B\} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} \tag{19.2–2}$$

In both operations, the components of vectors {A} and {B} are transformed to global Cartesian coordinates before the calculations. The results of the **VCROSS** command are also in global Cartesian coordinates.

19.2.2 Surface Operations (Integration of Values Across a Free Surface)

The **INTSRF** command can be used to integrate nodal values across a free surface. The free surface is determined by a selected set of nodes which must lie on an external surface of the selected set of elements.

Only pressure values can be integrated (for purposes of lift and drag calculations in fluid flow analyses). As a result of the integration, force and moment components in the global Cartesian coordinate system are:

$$\{F_t\} = \int_{area} \{p\} \; d(area) \tag{19.2–3}$$

$$\{F_r\} = \int_{area} \{r\} \ x \ \{p\} \ d(area) \tag{19.2–4}$$

where: $\{F_t\}$ = force components

$\{F_r\}$ = moment components

$\{r\}$ = position vector = $\lfloor \ X \ Y \ Z \ \rfloor^T$

$\{p\}$ = distributed pressure vector

area = surface area

In the finite element implementation, the position vector {r} is taken with respect to the origin.

19.3 POST1 — Path Operations

General vector calculus may be performed along any arbitrary 2–D or 3–D path through a solid element model. Nodal data, element data, and data stored with **ETABLE** command may be mapped onto the path and operated on as described below.

19.3.1 Defining the Path

A path is defined by first establishing path parameters (**PATH** command) then defining path points which create the path (**PPATH** command). The path points may be nodes, or arbitrary points defined by geometry coordinates. A segment is a line connecting two path points. The number of path points used to create a path is defined by the Npts parameter on the **PATH** command. For each path segment the number of divisions used to descritize the path is defined by the nDiv parameter on the **PATH** command. The descritized path divisions are interpolated between path points in the currently active coordinate system (**CSYS** command), or as defined on the **PPATH** command. A typical segment is shown in Figure 19.3–1 as going from points N_1 to N_2, where N_1 and N_2 are input quanities *NODE1* and *NODE2*, respectively for the first segment.

The geometry of each point along the path is stored. The geometry consists of the global Cartesian coordinates (output label XG, YG, ZG) and the length from the first path point along the path (output label S). The geometry is available for subsequent operations.

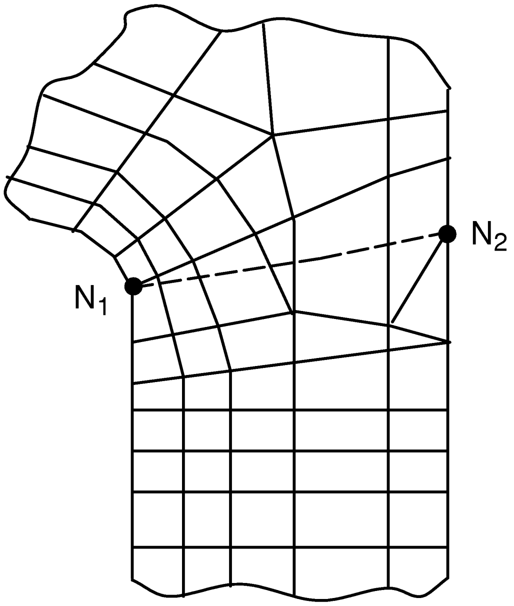


Figure 19.3–1 Typical Path Segment

19.3.2 Defining Orientation Vectors of the Path

In addition, position (R), unit tangent (T), and unit normal (N) vectors to a path point are available as shown in Figure 19.3–2. These three vectors are defined in the active Cartesian coordinate system.

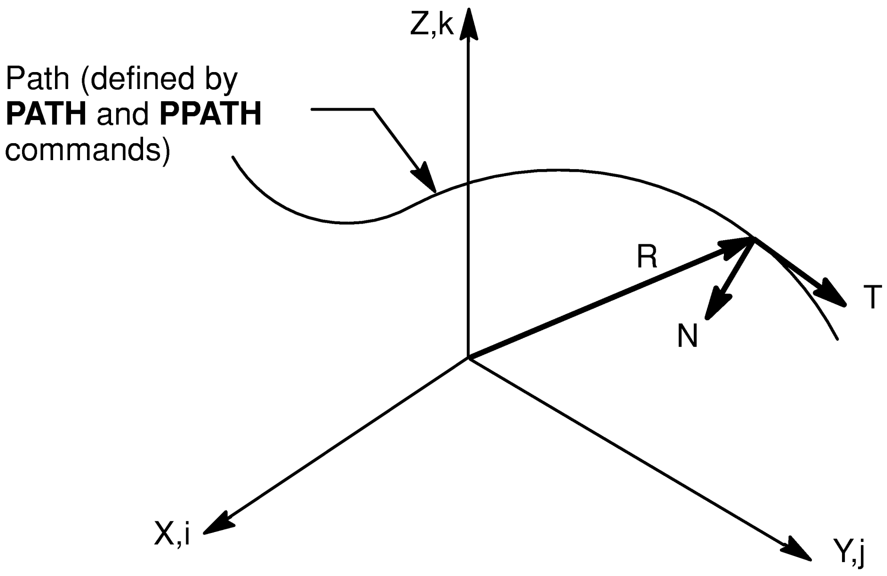


Figure 19.3–2 Position and Unit Vectors of a Path

The position vector R (stored with **PVECT**,RADI command) is defined as:

$$\{R\} = \begin{Bmatrix} x_n \\ y_n \\ z_n \end{Bmatrix} \tag{19.3–1}$$

where: x_n = x coordinate in the active Cartesian system of path point n, etc.

The unit tangent vector T (stored with **PVECT**,TANG command) is defined as:

$$\{T_1\} = C \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{Bmatrix} \quad \text{(for first path point)} \tag{19.3–2}$$

$$\{T_n\} = C \begin{Bmatrix} x_{n+1} - x_{n-1} \\ y_{n+1} - y_{n-1} \\ z_{n+1} - z_{n-1} \end{Bmatrix} \quad \text{(for intermediate path point)} \tag{19.3–3}$$

$$\{T_L\} = C \begin{Bmatrix} x_L - x_{L-1} \\ y_L - y_{L-1} \\ z_L - z_{L-1} \end{Bmatrix} \text{ (for last path point)} \qquad (19.3\text{–}4)$$

where: x,y,z = coordinate of a path point in the active Cartesian system n = 2 to (L–1)

L = number of points on the path

C = scaling factor so that {T} is a unit vector

The unit normal vector N (**PVECT**,NORM command) is defined as:

$$\{N\} = \{T\} \text{ x } \{k\} \ / \ |\{T\} \text{ x } \{k\}| \qquad (19.3\text{–}5)$$

where: x = cross product operator

$$\{k\} = \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

{N} is not defined if {T} is parallel to {k}.

19.3.3 Mapping Nodal and Element Data onto the Path

Having defined the path, the nodal or element data (as requested by *Item,Comp* on the **PDEF** command) may be mapped onto the path. For each path point, the selected elements are searched to find an element containing that geometric location. In the lower order finite element example of Figure 19.3–3, point N_o has been found to be contained by the element described by nodes N_a, N_b, N_c and N_d. Nodal degree of freedom data is directly available at nodes N_a, N_b, N_c and N_d. Element result data may be interpreted either as averaged data over all elements connected to a node (as described in the Nodal Data Computation topic, see Section 19.1). or as unaveraged data taken only from the element containing the path interpolation point through the *Avglab* option on the **PDEF** command. When using the material discontinuity option (MAT option on the **PMAP** command) unaveraged data is mapped automatically.

Caution should be used when defining a path for use with the unaveraged data option. Avoid defining a path **(PPATH** command) directly along element boundaries since the choice of element for data interpolation may be unpredictable. Path values at nodes use the element from the immediate preceeding path point for data interpolation.

The value at the point being studied (i.e. point N_o) is determined by using the element shape functions together with these nodal values. Principal results data (principal stresses, strains, flux density magnitude, etc.) are mapped onto a path by first interpolating item components to the path and then calculating the principal value from the interpolated components.

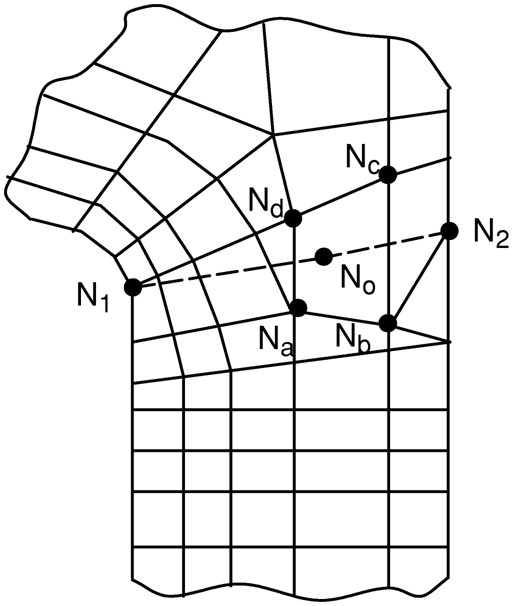


Figure 19.3–3 Mapping Data

Higher order elements include midside nodal (DOF) data for interpolation. Element data at the midside nodes are averaged from corner node values before interpolation.

19.3.4 Operating on Path Data

Once nodal or element data are defined as a path item, its associated path data may be operated on in several ways. Path items may be combined by addition, multiplication, division, or exponentiation (**PCALC** command). Path items may be differentiated or integrated with respect to any other path item (**PCALC** command). Differentiation is based on a central difference method without weighting:

$$\dot{A}_1 = \frac{A_2 - A_1}{B_2 - B_1} \text{ x S} \qquad \text{(for first path point)} \tag{19.3–6}$$

$$\dot{A}_n = \frac{A_{n+1} - A_{n-1}}{B_{n+1} - B_{n-1}} \text{ x S} \qquad \text{(for intermediate path points)} \tag{19.3–7}$$

$$\dot{A}_L = \frac{A_L - A_{L-1}}{B_L - B_{L-1}} \text{ x S} \qquad \text{(for last path point)} \tag{19.3–8}$$

where: A = values associated with LAB1, on the **PCALC**,DERI command

B = values associated with LAB2, on the **PCALC**,DERI command n = 2 to (L–1)

L = number of points on the path

S = scale factor (input quantity FACT1, on the **PCALC**,DERI command)

If the denominator is zero for equations (19.3–6) thru (19.3–8), then the derivative is set to zero.

Integration is based on the rectangular rule (see Figure 18.1–1 for an illustration):

$$A_1^* = 0.0 \qquad (19.3–9)$$

$$A_n^* = A_{n-1}^* + \frac{1}{2}\left(A_n + A_{n+1}\right)\left(B_n - B_{n-1}\right) \text{ x } S \qquad (19.3–10)$$

Path items may also be used in vector dot (**PDOT** command) or cross (**PCROSS** command) products. The calculation is the same as the one described in the Vector Dot and Cross Products Topic, above. The only difference is that the results are not transformed to be in the global Cartesian coordinate system.

19.4 POST1 — Stress Linearization

An option is available to allow a separation of stresses through a section into constant (membrane) and linear (bending) stresses. An approach similar to the one used here is reported by Gordon(63). The stress linearization option is accessed using the **PRSECT, PLSECT**, or **FSSECT** commands for a path defined by two nodes with the **LPATH** command. The section is defined by a path consisting of two end points (nodes N_1 and N_2) as shown in Figure 19.4–1 (nodes) and 47 intermediate points (automatically determined by linear interpolation in the active display coordinate system (**DSYS**). Nodes N_1 and N_2 are normally both presumed to be at free surfaces.

Initially, a path must be defined and the results mapped onto that path as defined above. The logic for most of the remainder of the stress linearization calculation depends on whether the structure is axisymmetric or not, as indicated by the value of ϱ (input as *RHO* on **PRSECT**, **PLSECT**, or **FSSECT** commands). For ϱ = 0.0, the structure is not axisymmetric (Cartesian case); and for nonzero values of ϱ, the structure is axisymmetric. The explicit definition of ϱ, as well as the discussion of the treatment of axisymmetric structures, is discussed later.

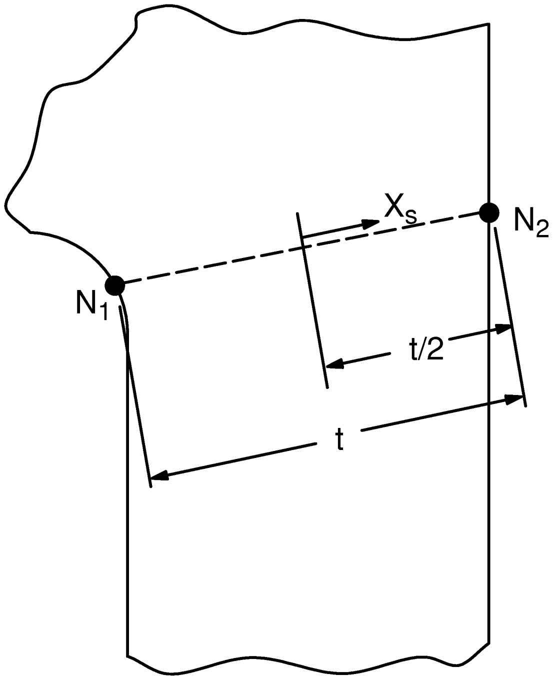


Figure 19.4–1 Coordinates of Cross Section

19.4.1 Cartesian Case

Refer to Figure 19.4–2 for a graphical representation of stresses. The membrane values of the stress components are computed from:

$$\sigma_i^m = \frac{1}{t}\int_{-t/2}^{t/2} \sigma_i \; dx_s \tag{19.4–1}$$

where:
σ_i^m = membrane value of stress component i
t = thickness of section, as shown in Figure 19.4–1
σ_i = stress component i along path from results file (‘total’ stress)
x_s = coordinate along path, as shown in Figure 19.4–1

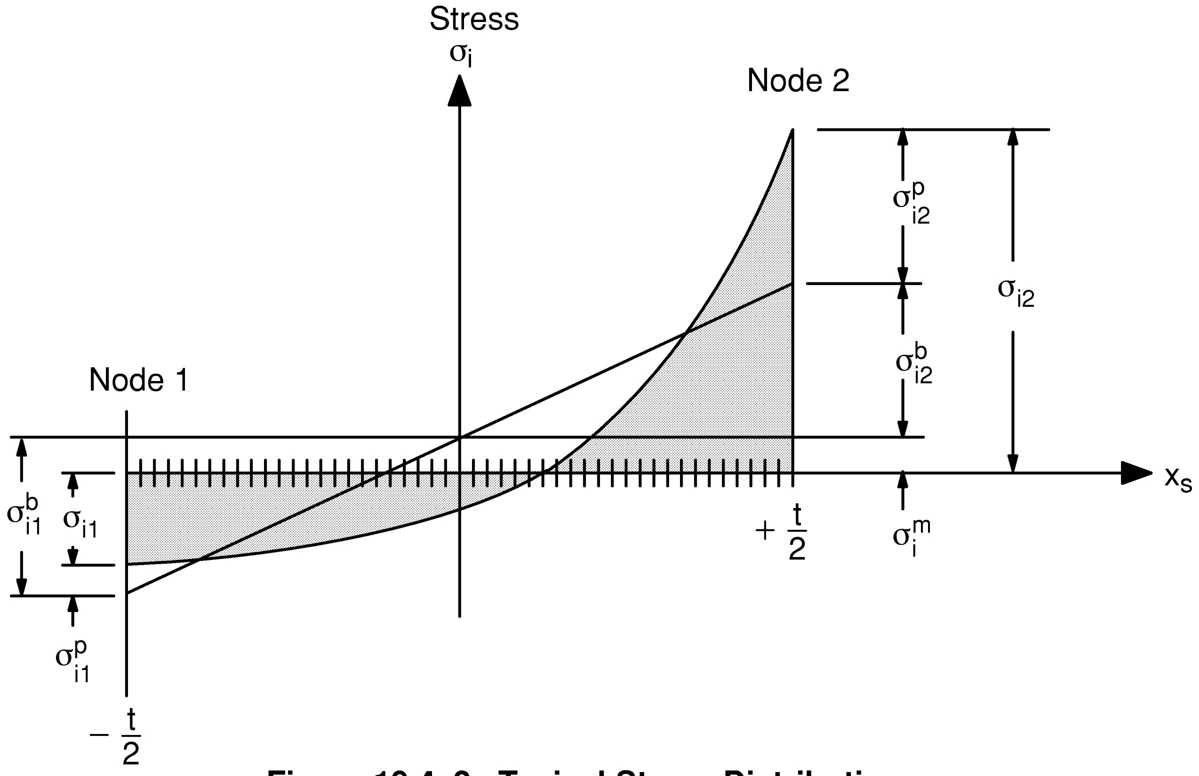


Figure 19.4–2 Typical Stress Distribution

The subscript i is allowed to vary from 1 to 6, representing σ_x, σ_y, σ_z, σ_{xy}, σ_{yz}, and σ_{xz}; respectively. These stresses are in global Cartesian coordinates. Strictly speaking, the integrals such as the one above are not literally performed; rather it is evaluated by numerical integration:

$$\sigma_i^m = \frac{1}{48}\left[\frac{\sigma_{i,1}}{2} + \frac{\sigma_{i,49}}{2} + \sum_{j=2}^{47} \sigma_{i,j}\right] \tag{19.4–2}$$

where: $\sigma_{i,j}$ = total stress component i at point j along path

The integral notation will continue to be used, for ease of reading.

The “bending” values of the stress components at node N$_1$ are computed from:

$$\sigma^b_{i1} = \frac{-6}{t^2} \int_{-t/2}^{t/2} \sigma_i \; x_s \; dx_s \qquad (19.4\text{–}3)$$

where: σ^b_{i1} = bending value of stress component i at node N$_1$

The bending values of the stress components at node N$_2$ are simply

$$\sigma^b_{i2} = -\,\sigma^b_{i1} \qquad (19.4\text{–}4)$$

where: σ^b_{i2} = bending value of the stress component i at node N$_2$

The “peak” value of stress at a point is the difference between the total stress and the sum of the membrane and bending stresses. Thus, the peak stress at node N$_1$ is:

$$\sigma^p_{i1} = \sigma_{i1} - \sigma^m_i - \sigma^b_{i1} \qquad (19.4\text{–}5)$$

where: σ^p_{i1} = peak value of stress component i at node N$_1$

σ_{i1} = value of total stress component i at node N$_1$

Similarly, for node N$_2$,

$$\sigma^p_{i2} = \sigma_{i2} - \sigma^m_i - \sigma^b_{i2} \qquad (19.4\text{–}6)$$

At the center point (x = 0.0)

$$\sigma^p_{ic} = \sigma_{ic} - \sigma^m_i \qquad (19.4\text{–}7)$$

where: σ^p_{ic} = peak value of stress component i at center

σ_{ic} = computed (total) value of stress component i at center

19.4.2 Axisymmetric Case (General)

The axisymmetric case is the same, in principle, as the Cartesian case, except for the fact that there is more material at a greater radius than at a smaller radius. Thus, the neutral axis is shifted radially outward a distance x$_f$, as shown in Figure 19.4–3. The axes shown in Figure 19.4–3 are Cartesian, i.e., the logic presented here is only valid for structures axisymmetric in the global cylindrical system. As stated above, the axisymmetric case is selected if $\varrho \neq 0.0$. ϱ is defined as the radius of curvature of the midsurface in the X–Y plane, as shown in Figure 19.4–4. A point on the centerplane of the torus has its curvatures defined by two radii: ϱ and the radial position R_c. Both of these radii will be used in the forthcoming development. In the case of an axisymmetric straight section such as a cylinder, cone, or disk, $\varrho = \infty$, so that input quantity *RHO* on the **PRSECT**, **PLSECT**, or **FSSECT** commands should be input as a large number (or –1.).

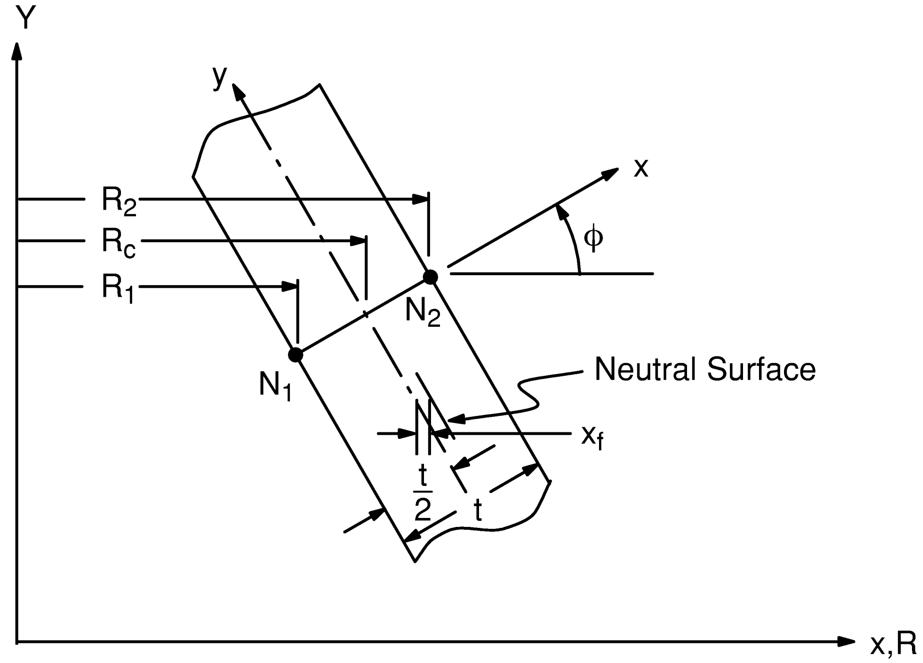


Figure 19.4–3 Axisymmetric Cross–Section

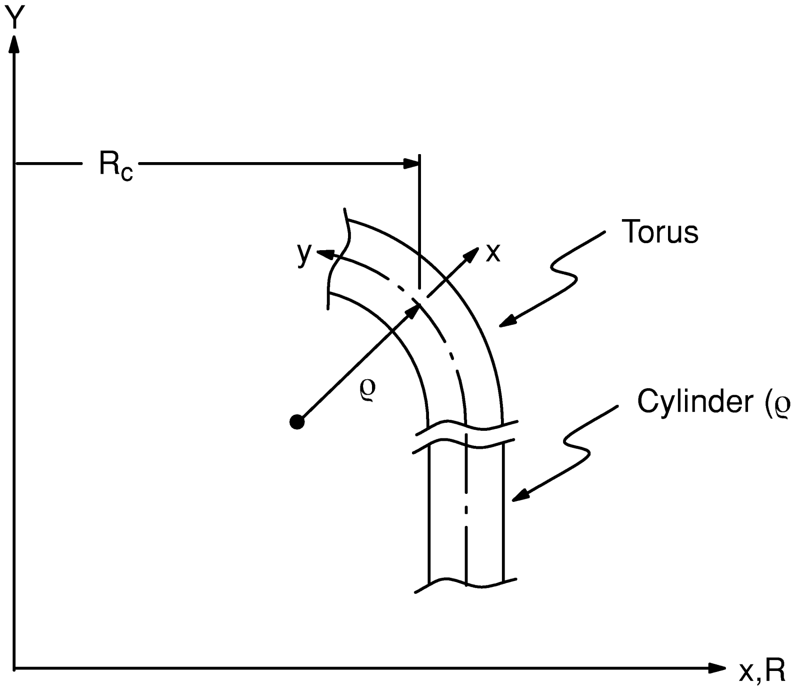


Figure 19.4–4 Geometry Used for Axisymmetric Evaluations

Each of the components for the axisymmetric case needs to be treated separately. For this case, the stress components are rotated into section coordinates, so that x stresses are parallel to the path and y stresses are normal to the path.

Starting with the y direction membrane stress, the force over a small sector is:

$$F_y = \int_{-t/2}^{t/2} \sigma_y \; R \; \Delta\theta \; dx \qquad (19.4\text{–}8)$$

where:

F_y = total force over small sector

σ_y = actual stress in y (meridional) direction

R = radius to point being integrated

$\Delta\theta$ = angle over a small sector in the hoop direction

t = thickness of section (distance between nodes N_1 and N_2)

The area over which the force acts is:

$$A_y = R_c \, \Delta\theta \;\, t \qquad (19.4\text{–}9)$$

where:

A_y = area of small sector

$R_c = \dfrac{R_1 + R_2}{2}$

R_1 = radius to node N_1

R_2 = radius to node N_2

Thus, the average membrane stress is:

$$\sigma_y^m = \frac{F_y}{A_y} = \frac{\int_{-t/2}^{t/2} \sigma_y \ R \ dx}{R_c t} \tag{19.4–10}$$

where: σ_y^m = y membrane stress

To process the bending stresses, the distance from the center surface to the neutral surface is needed. This distance is shown in Figure 19.4–3 and is:

$$x_f = \frac{t^2 \cos\phi}{12 \ R_c} \tag{19.4–11}$$

The derivation of equation (19.4–11) is the same as for y_f given at the end of Section 14.61. Thus, the bending moment may be given by:

$$M = \int_{-t/2}^{t/2} \left(x - x_f\right) \ dF \tag{19.4–12}$$

or

$$M = \int_{-t/2}^{t/2} \left(x - x_f\right) \ \sigma_y \ R \ \Delta\theta \ dx \tag{19.4–13}$$

The moment of inertia is:

$$I = \frac{1}{12} \ R_c \ \Delta\theta t^3 - R_c \ \Delta\theta \ t \ x_f^2 \tag{19.4–14}$$

The bending stresses are:

$$\sigma^b = \frac{Mc}{I} \tag{19.4–15}$$

where: c = distance from the neutral axis to the extreme fiber

Combining the above three equations,

$$\sigma_{y1}^b = \frac{M(x_1 - x_f)}{I} \tag{19.4–16}$$

or

$$\sigma^{b}_{y1} = \frac{x_1 - x_f}{R_c t\left(\frac{t^2}{12} - {x_f}^2\right)} \int_{-t/2}^{t/2} \left(x - x_f\right) \sigma_y \; R \; dx \qquad (19.4\text{–}17)$$

where: σ^{b}_{y1} = y bending stress at node N_1

Also,

$$\sigma^{b}_{y2} = \frac{M(x_2 - x_f)}{I} \qquad (19.4\text{–}18)$$

or

$$\sigma^{b}_{y2} = \frac{x_2 - x_f}{R_c t\left(\frac{t^2}{12} - {x_f}^2\right)} \int_{-t/2}^{t/2} \left(x - x_f\right) \sigma_y \; R \; dx \qquad (19.4\text{–}19)$$

where: σ^{b}_{y2} = y bending stress at node N_2

σ_x represents the stress in the direction of the thickness. Thus, σ_{x1} and σ_{x2} are the negative of the pressure (if any) at the free surface at nodes N_1 and N_2, respectively. A membrane stress is computed as:

$$\sigma^{m}_{x} = \frac{1}{t} \int_{-t/2}^{t/2} \sigma_x \; dx \qquad (19.4\text{–}20)$$

where: σ^{m}_{x} = the x membrane stress

When input quantity *KBR* on **PRSECT**, **PLSECT**, or **FSSECT** commands is set to 1, bending stresses are equated to zero:

$$\sigma^{b}_{x1} = 0 \qquad (19.4\text{–}21)$$

$$\sigma^{b}_{x2} = 0 \qquad (19.4\text{–}22)$$

Otherwise (KBR = 0), bending stresses are computed as:

$$\sigma^{b}_{x1} = \sigma_{x1} - \sigma^{m}_{x} \qquad (19.4\text{–}23)$$

$$\sigma^{b}_{x2} = \sigma_{x2} - \sigma^{m}_{x} \qquad (19.4\text{–}24)$$

where: σ^b_{x1} = x bending stress at node N_1

σ_{x1} = total x stress at node N_1

σ^b_{x2} = x bending stress at node N_2

σ_{x2} = total x stress at node N_2

The hoop stresses are processed next.

$$\sigma^m_h = \frac{F_h}{A_h} = \frac{\Delta\phi \int_{-t/2}^{t/2} \sigma_h \left(\rho + x\right) dx}{\Delta\phi \ \rho \ t} \qquad (19.4\text{–}25)$$

where: σ^m_h = hoop membrane stress

F_h = total force over small sector

$\Delta\phi$ = angle over small sector in the meridional (y) direction

σ_h = hoop stress

A_h = area of small sector in the x–y plane

ϱ = radius of curvature of the midsurface of the section (input quantity *RHO*)

x = coordinate thru cross–section

t = thickness of cross–section

Equation (19.4–25) can be reduced to:

$$\sigma^m_h = \frac{1}{t} \int_{-t/2}^{t/2} \sigma_h \left(1 + \frac{x}{\rho}\right) dx \qquad (19.4\text{–}26)$$

Using logic analogous to that needed to derive equations ((19.4–17)) and ((19.4–19)), the hoop bending stresses are computed by:

$$\sigma^b_{h1} = \frac{x_1 - x_h}{t\left(\frac{t^2}{12} - x_h{}^2\right)} \int_{-t/2}^{t/2} \left(x - x_h\right) \sigma_h \left(1 + \frac{x}{\rho}\right) dx \qquad (19.4\text{–}27)$$

and

$$\sigma^b_{h1} = \frac{x_2 - x_h}{t\left(\frac{t^2}{12} - x_h{}^2\right)} \int_{-t/2}^{t/2} \left(x - x_h\right) \sigma_h \left(1 + \frac{x}{\rho}\right) dx \qquad (19.4\text{–}28)$$

where:

$$x_h = \frac{t^2}{12\rho} \qquad (19.4\text{–}29)$$

for hoop–related calculations of equations (19.4–27) and (19.4–28).

An xy membrane shear stress is computed as:

$$\sigma^m_{xy} = \frac{1}{R_c t} \int_{-t/2}^{t/2} \sigma_{xy}\ R\ dx \qquad (19.4\text{–}30)$$

where:
σ^m_{xy} = xy membrane shear stress
σ_{xy} = xy shear stress

Since the shear stress distribution is assumed to be parabolic and equal to zero at the ends, the xy bending shear stress is set to 0.0. The other two shear stresses (σ_{xz}, σ_{yz}) are assumed to be zero.

All peak stresses are computed from

$$\sigma_i^{\,P} = \sigma_i - \sigma_i^{\,m} - \sigma_i^{\,b} \qquad (19.4\text{–}31)$$

where:
σ_i^P = peak value of stress component i
σ_i = total value of stress of component i

19.4.3 Axisymmetric Case (Specializations for Centerline)

At this point it is important to mention one exceptional configuration related to the y–direction membrane and bending stress calculations above. For paths defined on the centerline (X = 0), R_c = 0 and cosΦ = 0, and therefore equations (19.4–10), (19.4–11) (19.4–17), and (19.4–19) are undefined. Since centerline paths are also vertical (ϕ = 90°), it follows that R = R_c, and R_c is directly cancelled from stress equations (19.4–10), (19.4–17), and (19.4–18). However, x_f remains undefined. Figure 19.4–5 shows a centerline path from N_1 to N_2 in which the inside and outside wall surfaces form perpendicular intersections with the centerline.

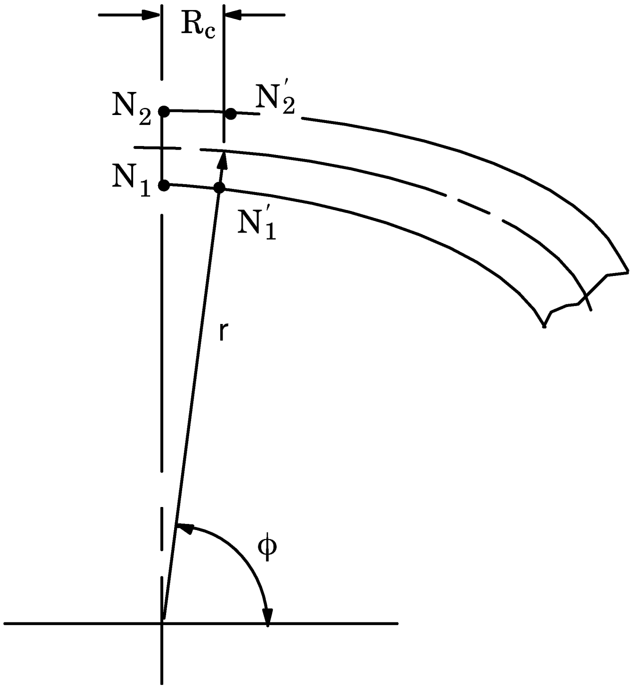


Figure 19.4–5 Centerline Sections

For this configuration it is evident that cos $\phi = R_c / \varrho$ as ϕ approaches 90° (or as $N_1' - N_2'$ approaches $N_1 - N_2$). Thus for any paths very near or exactly on the centerline, equation (19.4–11) is generalized to be:

$$x_f = \begin{cases} \dfrac{t^2 \cos\phi}{12 R_c} & \text{if } R_c \geq \dfrac{t}{1000} \\[2em] \dfrac{t^2}{12\rho} & \text{if } R_c < \dfrac{t}{1000} \end{cases} \qquad (19.4\text{–}32)$$

The second option of equation (19.4–32) applied to centerline paths is an accurate representation for spherical/elliptical heads and flat plates. It is incorrect for axisymmetric shapes that do not form perpendicular intersections with the centerline (e.g., conical heads). For such shapes (as shown in Figure 19.4–6) centerline paths must not be selected.

19.4.4 Stress Linearization Output

When the **PRSECT** command is given, the membrane, bending, peak, and total stresses are printed. When the **PLSECT** command is given, the desired *Item,Comp* on the command is plotted versus distance (x) showing the membrane, membrane plus bending, and total stress. When the **FSSECT** command is given, the membrane plus bending and total stresses are stored for later manipulation in the FATIGUE module as described below.

Principal stress items are computed from the six linearized stress components. For the output given with **PRSECT** and **PLSECT** commands, the results are displayed in the active coordinate system.

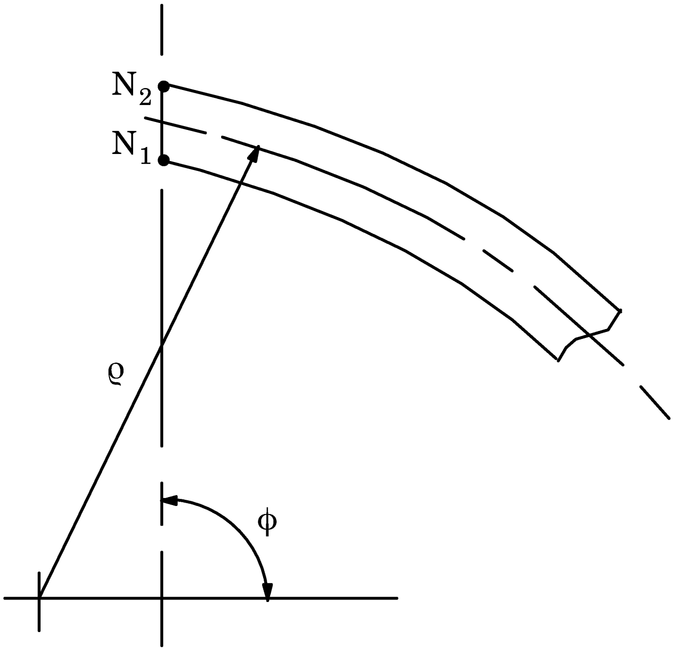


Figure 19.4–6 Non–Perpendicular Intersections

19.5 POST1 — Fatigue Module

The FATIGUE module of POST1 combines the effects of stress cycling over many cycles involving all stress components at a point in the structure. The procedure is explained in the *ANSYS Structural Analysis Guide*.

The module automatically calculates all possible stress ranges and keeps track of their number of occurrences, using a technique commonly known as the "rain flow" range–counting method. At a selected nodal location, a search is made throughout all of the events for the pair of loadings (stress vectors) that produces the most severe stress–intensity range. The number of repetitions possible for this range is recorded, and the remaining number of repetitions for the events containing these loadings is decreased accordingly. At least one of the source events will be "used up" at this point; remaining occurrences of stress conditions belonging to that event will subsequently be ignored. This process continues until all ranges and numbers of occurrences have been considered.

The fatigue calculations rely on the ASME Boiler and Pressure Vessel Code, Section III (and Section VIII, Division 2)(60) for guidelines on range counting, simplified elastic–plastic adaptations, and cumulative fatigue summations by Miner's rule.

The following steps are initiated by the **FTCALC** command.

1. Each loading is compared to each other loading to compute a maximum alternating shear stress:

   A. First, a vector of stress differences is computed:

$$\{\sigma\}_{i,j} = \{\sigma\}_i - \{\sigma\}_j \qquad (19.5\text{–}1)$$

where $\{\sigma\}_i$ = stress vector for loading ℓ_i

$\{\sigma\}_j$ = stress vector for loading ℓ_j

   B. Second, a stress intensity (σ_I (i,j)) is computed based on $\{\sigma\}_{i,j}$, using equation (2.3–7).

   C. Then, the interim maximum alternating shear stress is:

$$\sigma^d_{i,j} = \frac{\sigma_I(i,j)}{2} \qquad (19.5\text{–}2)$$

   D. The maximum alternating shear stress is calculated as:

$$\sigma^c_{i,j} = K_e\ \sigma^d_{i,j} \qquad (19.5\text{–}3)$$

where K_e is determined by:

Analysis Type	Range	K_e
ELASTIC (Based on peak stresses)	All	1.0
SIMPLIFIED ELASTIC PLASTIC (Based on linearized stress components)	$\sigma_n < 3\ S_m$	1.0
	$3\ S_m < \sigma_n < 3\ m\ S_m$	$1.0 + \frac{(1-n)}{n(m-1)}\left(\frac{\sigma_n}{3S_m} - 1\right)$
	$3\ m\ S_m < \sigma_n$	$\frac{1.0}{n}$

where: σ_n = a stress intensity equivalent of 2 $\sigma^d_{i,j}$ except that it is based on *linearized* stresses, not actual stresses. Linearized stresses are those generated by the **FSSECT** command. (Note that nomenclature is not the same in Section 19.4 as in this section.)

S_m = design stress–intensity obtained from the S_m versus temperature table. The table must be input using the **FP** commands inputting *SM1* to *SM10* and *T1* to *T10*.

m = input quantity *M* on **FP** command (m > 1.0)

n = input quantity *N* on **FP** command (0.0 < n < 1.0)

2. There are a total of (L/2) (L–1) loading case combinations, where L is the number of loadings. These loadings are then sorted (the rain flow method), with the highest value of $\sigma^c_{i,j}$ first.

3. Designate the highest value of $\sigma^c_{i,j}$ as occurring with loading ℓ_i, event k_i together with loading ℓ_j, event k_j. Let M_T be the minimum number of times that either event k_i or event k_j is expected to occur. Compute a usage factor following Miner's rule as:

$$f_u = \frac{M_T}{M_A} \tag{19.5–4}$$

where: f_u = usage factor (output quantity PARTIAL USAGE)

M_A = number of allowable cycles at this stress amplitude level. Obtained by entering the allowable alternating stress amplitude (S_a) versus cycles (N) table from the S_a axis and reading the allowable number of cycles M_A corresponding to $\sigma^c_{i,j}$. The table must be input using the **FP** commands inputting *S1* to *S20* for S_a and *N1* to *N20* for N.

Next, cumulatively add f_u to f^c_u, where f^c_u = output quantity CUMULATIVE FATIGUE USAGE. Then decrease the number of possible occurrences of both event k_i and event k_j by M_T (so that one of them becomes zero).

4. Repeat step 3, using the next highest value of $\sigma^{c}_{i,j}$ until all of the $\sigma^{c}_{i,j}$ values have been exhausted.. It may be seen that the number of times this cycle is performed is equal to the number of events (or less).

19.6 POST1 — Electromagnetic Macros

Electromagnetic macros are macro files created to perform specific postprocessing operations for electromagnetic field analysis. Macros performing computational analysis are detailed in this section.

19.6.1 FLUXV Macro

The **FLUXV** macro calculates the flux passing through a surface defined by a closed line contour (**LPATH** command). The macro is applicable to 2–D and 3–D magnetic field analysis employing the magnetic vector potential A. For 2–D planar analyses, the flux value is per unit depth.

The flux passing through a surface S can be calculated as:

$$\phi = \int_{\text{area}} \{B\} \cdot \{n\} \; d(\text{area}) \qquad (19.6\text{–}1)$$

where:
- ϕ = flux enclosed by the bounding surface S
- {B} = flux density vector
- {n} = unit normal vector
- area = area of the bounding surface S

Equation (19.6–1) can be rewritten in terms of the definition of the vector potential as:

$$\phi = \int_{\text{area}} (\nabla \times \{A\}) \cdot \{n\} \; d(\text{area}) \qquad (19.6\text{–}2)$$

where:
- {A} = magnetic vector potential

By applying Stokes theorem, the surface integral reduces to a line integral of A around a closed contour;

$$\phi = \int_{\ell} \{A\} \cdot d\ell \qquad (19.6\text{–}3)$$

where:
- ℓ = length of the bounding contour line

The macro interpolates values of the vector potential, A, to the closed contour path defined by the **LPATH** command and integrates to obtain the flux using equation (19.6–3). In the axisymmetric case, the vector potential is multiplied by 2πr to obtain the total flux for a full circumferential surface (where “r” is the x–coordinate location of the interpolation point).

19.6.2 FOR2D Macro

The **FOR2D** macro makes use of the Maxwell stress tensor(77) to evaluate the force on a body. The Maxwell stress approach computes local stress at all points of a bounding surface and then sums the local stresses by means of a surface integral to find the net force on a body. The force can be expressed as:

$$\{F^{mx}\} = \frac{1}{\mu} \int_{area} [T] \cdot \{n\} \; d(area) \qquad (19.6\text{–}4)$$

where: $\{F^{mx}\}$ = total force vector on the body

$[T]$ = Maxwell stress tensor (see equation (5.3–14))

μ = permeability of the bounding region

In 2–D planar analyses the surface integral reduces to a line integral and the resulting force is per unit depth. The macro requires a pre–specified path (**LPATH** command) to create the bounding surface. The bounding surface (or line path) should encompass the body for which the force is to be calculated. In principle, the bounding surface (line) is the surface of the body itself. However, in practice it is common to place the path within the air domain surrounding the body. This is perfectly satisfactory and does not violate the principle of the Maxwell stress tensor since the air carries no current and has no magnetic properties different from free space.

The macros interpolate values of flux density, B, to the path defined by the **LPATH** command and integrates to obtain the force on the body as in equation (19.6–4).

19.6.3 MMF Macro

The **MMF** macro calculates the magnetomotive force (current) along a contour defined by the **LPATH** command according to Amperes’ theorem:

$$I_{mmf} = \int_{\ell} \{H\} \cdot d\ell \qquad (19.6\text{–}5)$$

where: I_{mmf} = magnetomotive force

$\{H\}$ = magnetic field intensity vector

The macro interpolates values of magnetic field intensity, H, to the path defined by the **LPATH** command and integrates to obtain the I_{mmf} as in equation (19.6–5). In a static analysis or transverse electromagnetic (TEM) and transverse electric (TE) wave guide mode computation, I_{mmf} can be interpreted as a current passing the surface bounded by the closed contour.

19.6.4 POWERH Macro

The **POWERH** macro calculates the power dissipated in a conducting solid body under the influence of a time–harmonic electromagnetic field. The r.m.s. power loss is calculated from the equation (see Section 5.1.5 for further details):

$$P_{rms} = \frac{1}{2} \int_{vol} \rho \; | \tilde{J}_t |^2 \; d(vol) \qquad (19.6\text{–}6)$$

where:

P_{rms} = rms power loss

ϱ = material resistivity

J_t = total current density

~ = complex quantity

The macro evaluates equation (19.6–6) by integrating over the selected element set according to:

$$P_{rms} = \frac{1}{2} \operatorname{Re} \left\{ \sum_{i=1}^{n} \left(\left([\rho_i] \; \{\tilde{J}_{ti}\} \right)^{*} \cdot \{\tilde{J}_{ti}\} \right) vol_i \right\} \qquad (19.6\text{–}7)$$

where:

n = number of elements

Re{ } = real component of a complex quantity

$[\varrho_i]$ = resistivity tensor (matrix)

$\{\tilde{J}_{ti}\}$ = total eddy current density vector for element i

vol_i = element volume

* = complex conjugate operator

For 2–D planar analyses, the resulting power loss is per unit depth.

For high frequency analysis, dielectric losses from lossy materials are calculated as per equation (19.6–54). Surface losses on boundaries with specified impedance are calculated as per equation (19.6–53).

19.6.5 SRCS Macro

The **SRCS** macro calculates terminal parameter quantities for a stranded coil with a d.c. current. The macro is applicable to linear magnetostatic analysis. In addition, the far–field boundary of the model must be treated with either a flux–normal (Neumann condition), flux–parallel (Dirichlet condition), or modelled with infinite elements.

Energy Supplied

The energy supplied to the coil for a linear system is calculated as:

$$W = \frac{1}{2} \int_{vol} \{A\} \cdot \{J_s\} \, d(vol) \qquad (19.6–8)$$

where:

W	=	energy input to coil
$\{A\}$	=	nodal vector potential
$\{J_s\}$	=	d.c. source current density
vol	=	volume of the coil

Terminal Inductance

The inductance as seen by the terminal leads of the coil is calculated as:

$$L = \frac{2W}{i^2} \qquad (19.6–9)$$

where:

L	=	terminal inductance
i	=	coil current (per turn)

Flux Linkage

The total flux linkage of a coil can be calculated from the terminal inductance and coil current,

$$\lambda = Li \qquad (19.6–10)$$

where: λ = flux linkage

Terminal Voltage

For a coil operating with an a.c. current at frequency ω (Hz), a voltage will appear at the terminal leads. Neglecting skin effects and saturation, a static analysis gives the correct field distribution. For the assumed operating frequency, the terminal voltage can be found. From Faraday's law,

$$u = \frac{d\lambda}{dt} \qquad (19.6–11)$$

where: u = terminal voltage

Under a sinusoidal current at an operating frequency ω, the flux linkage will vary sinusoidally

$$\lambda = \lambda_m \sin \omega t \qquad (19.6\text{–}12)$$

where: λ_m = zero–to–peak magnitude of the flux linkage

The terminal voltge is therefore:

$$u = U \cos \omega t \qquad (19.6\text{–}13)$$

where: U = $\omega\lambda_m$ = zero–to–peak magnitude of the terminal voltage (parameter VLTG returned by the macro)

For 2–D planar analyses, the results are per unit depth.

19.6.6 TORQ2D and TORQC2D Macros

The **TORQ2D** and **TORQC2D** macros make use of the Maxwell stress tensor (Coulomb(168)) to evaluate the torque on a body for a 2–D planar analysis. The torque integrand is evaluated at all points of a bounding surface about the body, and then summed to find the net torque on the body. The torque can be expressed as:

$$T = \frac{1}{\mu} \int_{\text{area}} \left[(\{B\} \cdot \{n\}) (\{R\} \times \{B\}) - \frac{|B|^2}{2} (\{R\} \times \{n\}) \right] d(\text{area}) \qquad (19.6\text{–}14)$$

where:

T = total torque on a body

μ = permeability of the bounding region

$\{B\}$ = flux density vector

$\{n\}$ = unit normal vector to the path

$\{R\}$ = position vector

area = area of the bounding surface

In 2–D planar analyses, the surface integral reduces to a line integral and the torque results are per unit depth. The macro **TORQ2D** requires a pre–specified path (**LPATH** command) to create the bounding surface. The bounding surface (or line path) should encompass the body for which the torque, about the global origin, is to be calculated.

In principle the bounding surface (line) is the surface of the body itself. However, in practice, it is common to place the path within the air domain surrounding the body. This is perfectly satisfactory and does not violate the principle of the Maxwell stress tensor since the air carries no current and has no magnetic properties different from free space.

The macro **TORQC2D** is a specialized version of the **TORQ2D** macro, limited to a circular bounding surface (line) about the global origin. The macro creates its own path (**LPATH**) for evaluation. For the case of a circular path, equation (19.6–14) reduces to:

$$T = \frac{1}{\mu} \int_{\text{area}} \left[\mu \left(\{B\} \cdot \{n\}\right) \left(\{R\} \times \{B\}\right)\right] d(\text{area}) \tag{19.6–15}$$

The macro **TORQC2D** makes use of equation (19.6–15) to evaluate torque.

For both torque macros, flux density, B, is interpolated to the path and integrated according to equation (19.6–14) or (19.6–15) to obtain the torque on a body.

19.6.7 SENERGY Macro

The **SENERGY** macro calculates the stored energy and co–energy in a magnetic field. For the static or transient analysis, the stored magnetic energy is calculated as:

$$W_s = \int_0^B \{H\} \cdot \{dB\} \tag{19.6–16}$$

where: W_s = stored magnetic energy

The magnetic co–energy is calculated as:

$$W_c = \int_0^H \{B\} \cdot \{dH\} \tag{19.6–17}$$

where: W_c = stored magnetic co–energy

For permanent magnetic materials, the stored magnetic energy is calculated as:

$$W_s = \int_{B_r}^B \{H\} \cdot \{dB\} \tag{19.6–18}$$

where: B_r = residual induction

and the co–energy as:

$$W_c = \int_0^H \{B\} \cdot \{dH\} \qquad (19.6\text{–}19)$$

For time–harmonic analysis, the r.m.s. stored magnetic energy is calculated as:

$$W_{rms} = \mathrm{Re}\,\frac{1}{4} \int \{\tilde{B}\} \cdot \{\tilde{H}\}^* \, d(vol) \qquad (19.6\text{–}20)$$

where: W_{rms} = r.m.s. stored energy

For 2–D planar analyses, the results are per unit depth.

19.6.8 EMAGERR Macro

The **EMAGERR** macro calculates the relative error in an electrostatic or electromagnetic field analysis. The relative error measure is based on the difference in calculated fields between a nodal–averaged continuous field representation and a discontinuous field represented by each individual element's–nodal field values. An average error for each element is calculated. Within a material, the relative error is calculated as:

Electrostatics

Electric Field:

$$E_{ei} = \frac{1}{n} \sum_{j=1}^{n} \left| E_j - E_{ij} \right| \qquad (19.6\text{–}21)$$

where: E_{ei} = relative error for the electric field (magnitude) for element i

E_j = nodal averaged electric field (magnitude)

E_{ij} = electric field (magnitude) of element i at node j

n = number of vertex nodes in element i

Electric Flux Density:

$$D_{ei} = \frac{1}{n} \sum_{j=1}^{n} \left| D_j - D_{ij} \right| \qquad (19.6\text{–}22)$$

where: D_{ei} = relative error for the electric flux density (magnitude) for element i

D_j = nodal averaged electric flux density (magnitude)

D_{ij} = electric flux density (magnitude) of element i at node j

A normalized relative error norm measure is also calculated based on the maximum element nodal calculated field value in the currently selected element set.

$$E_{nei} = E_{ei} / E_{max} \tag{19.6–23}$$

where: E_{max} = maximum element nodal electric field (magnitude)

$$D_{nei} = D_{ei} / D_{max} \tag{19.6–24}$$

where: D_{max} = maximum element nodal electric flux density (magnitude)

Electromagnetics

Magnetic Field Intensity:

$$H_{ei} = \frac{1}{n} \sum_{j=1}^{n} | H_j - H_{ij} | \tag{19.6–25}$$

where: H_{ei} = relative error for the magnetic field intensity (magnitude) for element i

H_j = nodal averaged magnetic field intensity (magnitude)

H_{ij} = magnetic field intensity (magnitude) of element i at node j

Magnetic Flux Density:

$$B_{ei} = \frac{1}{n} \sum_{j=1}^{n} | B_j - B_{ij} | \tag{19.6–26}$$

where: B_{ei} = relative error for the magnetic flux density (magnitude) for element i

B_j = nodal averaged magnetic flux density (magnitude)

B_{ij} = magnetic flue density (magnitude) of element i at node j

A normalized relative error measure is also calculated based on the maximum element nodal calculated field value in the currently selected element set.

$$\mathrm{H_{nei} = H_{ei} \,/\, H_{max}} \tag{19.6–27}$$

where: $\mathrm{H_{max}}$ = maximum element nodal magnetic field intensity (magnitude)

$$\mathrm{B_{nei} = B_{ei} \,/\, B_{max}} \tag{19.6–28}$$

where: $\mathrm{B_{max}}$ = maximum nodal averaged magnetic flux density (magnitude)

19.6.9 SPARM Macro

The **SPARM** macro calculates S–parameters for two ports of a multi–port waveguide. The first port (port i) is the driven port, while the second port (port j) is matched. The S–parameters are calculated as:

$$\mathrm{S_{ii}} = \frac{\mathrm{b_n^{(i)}}}{\mathrm{a_n^{(i)}}} \tag{19.6–29}$$

$$\mathrm{S_{ji}} = \frac{\mathrm{b_n^{(j)}}}{\mathrm{a_n^{(i)}}} \tag{19.6–30}$$

where:

$$\mathrm{a_n^{(i)}} = \frac{\int\!\!\int_{\Omega_i} \mathrm{E_{t,inc}} \cdot \mathrm{e_{t,n}^{(i)}}\, d\Omega}{\int\!\!\int_{\Omega_i} \mathrm{e_{t,n}^{(i)}} \cdot \mathrm{e_{t,n}^{(i)}}\, d\Omega}$$

$$\mathrm{b_n^{(i)}} = \frac{\int\!\!\int_{\Omega_i} \left(\mathrm{E_{t,total}^{(i)}} - \mathrm{E_{t,inc}}\right) \cdot \mathrm{e_{t,n}^{(i)}}\, d\Omega}{\int\!\!\int_{\Omega_i} \mathrm{e_{t,n}^{(i)}} \cdot \mathrm{e_{t,n}^{(i)}}\, d\Omega}$$

$$\mathrm{b_n^{(j)}} = \frac{\int\!\!\int_{\Omega_j} \mathrm{E_{t,total}^{(j)}} \cdot \mathrm{e_{t,n}^{(j)}}\, d\Omega}{\int\!\!\int_{\Omega_j} \mathrm{e_{t,n}^{(j)}} \cdot \mathrm{e_{t,n}^{(j)}}\, d\Omega}$$

Ω_i = cross section of waveguide i

$E_{t,inc}$ = tangential electric field at port i

$e^{(i)}_{t,n}$ = tangential eigen electric field at port i

$E^{(i)}_{t,total}$ = total tangential electric field from EMAG solution at port i

19.6.10 EMF Macro

The **EMF** macro calculates the electromotive force (voltage drop) between two conductors defined along a path contour (**PATH** command):

$$V_{emf} = \int_{\ell} \{E\} \cdot d\ell \qquad (19.6\text{–}31)$$

where: V_{emf} = electromotive force (voltage drop)

$\{E\}$ = electric field vector

The macro interpolates values of the electric field, E, to the path defined by the **PATH** command and integrates to obtain the electromotive force (voltage drop). The path may span multiple materials of differing permittivity. At least one path point should reside in each material transversed by the path. In static analysis or transverse electromagnetic (TEM) and transverse magnetic (TM) wave guide mode computation, V_{emf} can be interpreted as a voltage drop.

19.6.11 IMPD Macro

The **IMPD** macro calculates the impedance of a device from the calculated V_{emf} and I_{mmf} values. The V_{emf} is calculated using the **EMF** macro and the I_{mmf} is calculated using the **MMF** macro. Impedance calculations are valid for transverse electromagnetic (TEM) waves in coaxial waveguide structures. The impedance is calculated as:

$$Z = \frac{V^{Re}_{emf} + j\ V^{im}_{emf}}{I^{Re}_{mmf} + j\ I^{im}_{mmf}} \qquad (19.6\text{–}32)$$

where: V and I = voltage drop and current, respectively

Re and Im = represent real and imaginary parts of complex terms

19.6.12 REFLCOEF Macro

The **REFLCOEF** macro calculates equivalent transmission–line parameters for a coax type guiding wave structure. For a lossless guiding structure, the total mode voltage, V(Z), and mode current, I(Z), associated with a +Z propagating field take on the form:

$$V(Z) = Ae^{-j\beta Z} + Be^{\,j\beta Z} \qquad (19.6\text{–}33)$$

$$I(Z) = \frac{A}{Z_o}\,e^{-j\beta Z} - \frac{B}{Z_o}\,e^{j\beta Z} \qquad (19.6\text{–}34)$$

where:

Z_o = characteristic impedance for any mode

A = amplitude of the incident voltage wave (see below)

B = amplitude of the backscattered voltage wave (see below)

We can consider the propogating waves in terms on an equivalent two–wire transmission line terminated at Z = ℓ by a load impedance Z_ℓ.

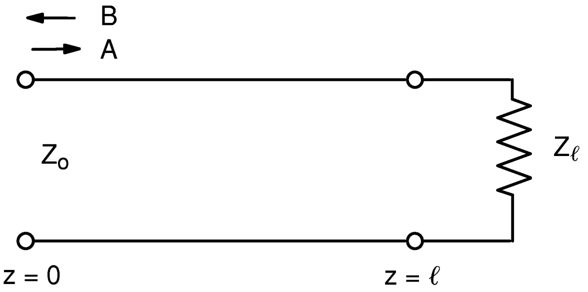


Figure 19.6–1 Equivalent Two–Wire Transmission Line

The voltage term “A” in equation (19.6–33) can be considered as the amplitude of the incident wave, and voltage term “B” as the amplitude of the mode voltage wave backscattered off the load impedance Z_ℓ.

Thus,

$$Z_\ell = Z_o\,\frac{Ae^{-j\beta\ell} + Be^{j\beta\ell}}{Ae^{-j\beta\ell} - Be^{j\beta\ell}} \qquad (19.6\text{–}35)$$

Rearranging we have,

$$\Gamma = \frac{B}{A} = \frac{(Z_\ell\ /\ Z_o) - 1}{(Z_\ell\ /\ Z_o) + 1} \qquad (19.6\text{–}36)$$

where:

Γ = voltage reflection coefficient (output as REFLC)

The voltage standing–wave ratio is calculated as:

$$S = \frac{1 + |\Gamma|}{1 - |\Gamma|} \tag{19.6–37}$$

where: S = voltage standing–wave ratio (output as VSWR)

For a matched load ($Z_\ell = Z_o$) there is no reflection (Γ = 0) and the S = 1. If Z_ℓ is a short circuit, B = –A, Γ = –1, and the S is infinite. If Z_ℓ is an open circuit, B = A, Γ = +1, and the S once again is infinite.

The reflection coefficient is frequently expressed in dB form by introducing the concept of return loss defined by:

$$L_R = -20 \log_{10} |\Gamma| \tag{19.6–38}$$

where: L_R = return loss in dB (output as RL)

The macro calculates the above transmission line parameters in terms of the incident, reference and total voltage.

Assuming a propagating distance ℓ from the input port, the total voltage ($V_{tot}|_\ell$) is calculated by a path integral using the emf macro from the calculated harmonic solution. The path for the line integral must start from the inner conductor and extend to the outer conductor at the propagation distance ℓ. The most common calculation will occur for a distance ℓ = 0. In this case the reflection coefficient is calculated at the excited port.

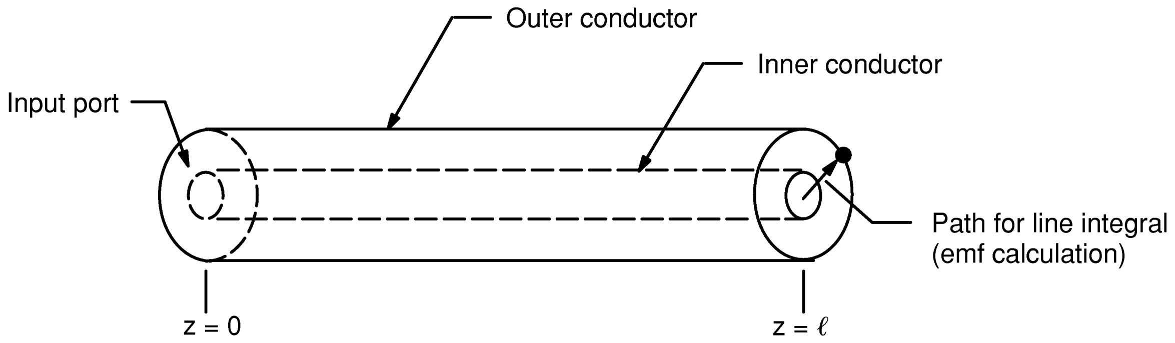


Figure 19.6–2 Coaxial Cable Diagram

The total voltage at Z = ℓ can be expressed in terms of the incident voltage (V_{in}) and the reflected voltage (V_{ref1}):

$$V_{tot}|_\ell = V_{in}|_\ell + v_{ref1}|_\ell \tag{19.6–39}$$

The incident voltage at the propagating distance ℓ from the input port is calculated as:

$$V_{in}|_{\ell} = A^{Re} \cos(\omega t - \beta\ell) - A^{Im} \sin(\omega t - \beta\ell) \qquad (19.6\text{–}40)$$

where:

A^{Re} = Real component of the incident wave

A^{Im} = Imaging component of the incident wave

β = wave number = $2\pi f/c$

f = frequency (Hz.)

c = propagation velocity number

= $3 \text{ x } 10^8 / \sqrt{\epsilon_r}$

ε_r = relative permittivity

A^{Re} and A^{Im} are calculated from the input voltage (VALUE3) and phase angle (VALUE4) of the **PORTOPT** command

$$A^{Re} = A \cos \phi \qquad (19.6\text{–}41)$$

$$A^{Im} = A \sin \phi \qquad (19.6\text{–}42)$$

where:

A = Incident voltage drop between coax conductors

ϕ = phase angle

The total voltage (calculated from the electric field solution) can be expressed as:

$$V_{tot}\Big|_{\ell} = V_{tot}^{Re} \cos \omega t - V_{tot}^{Im} \sin \omega t \qquad (19.6\text{–}43)$$

where:

V_{tot}^{Re} = real component of the voltage

V_{tot}^{Im} = imaging component of the voltage

Combining terms from equations (19.6–39) and (19.6–43), the reflected voltage may be calculated

$$\begin{aligned} V_{ref}|_{\ell} &= \left(V_{tot}^{Re} - A^{Re} \cos \beta\ell + A^{Im} \sin \beta\ell\right) \cos \omega t \\ &+ \left(- V_{tot}^{Im} - A^{Re} \sin \beta\ell + A^{Im} \cos \beta\ell\right) \sin \omega t \end{aligned} \qquad (19.6\text{–}44)$$

The reflected voltage can be expressed as

$$V_{ref}|_{\ell} = V_{ref}^{Re}|_{\ell} \cos \omega t - V_{ref}^{Im}|_{\ell} \sin \omega t \qquad (19.6\text{–}45)$$

Thus,

$$V_{ref}^{Re}|_{\ell} = V_{tot}^{Re} - A^{Re} \cos \beta\ell + A^{Im} \sin \beta\ell \qquad (19.6\text{–}46)$$

$$V^{\mathrm{Im}}_{\mathrm{ref}}|_{\ell} = V^{\mathrm{Im}}_{\mathrm{tot}} + A^{\mathrm{Re}} \sin \beta\ell - A^{\mathrm{Im}} \cos \beta\ell \qquad (19.6\text{–}47)$$

The magnitude of the voltage reflection coefficient at the propogating distance ℓ is calculated as:

$$|\Gamma| = \frac{|V_{\mathrm{ref}}|_{\ell}|}{|V_{\mathrm{in}}|_{\ell}|} \qquad (19.6\text{–}48)$$

The phase angle of the voltage reflection coefficient is calculated as:

$$\phi_{\Gamma} = \tan^{-1}\left[V^{\mathrm{Im}}_{\mathrm{ref}}|_{\ell} \ / \ V^{\mathrm{Re}}_{\mathrm{ref}}|_{\ell}\right] - \tan^{-1}\left[V^{\mathrm{Im}}_{\mathrm{in}}|_{\ell} \ / \ V^{\mathrm{Re}}_{\mathrm{in}}|_{\ell}\right] \qquad (19.6\text{–}49)$$

19.6.13 QFACT Macro

The **QFACT** macro calculates the quality factor, Q, used to measure the sharpness of a cavity resonance in a high frequency eigenvalue analysis. The Q factor can be expressed as:

$$Q = 2\pi f_o \frac{W}{P_m + P_d} \qquad (19.6\text{–}50)$$

where: f_o = resonant frequency (Hz.)
W = stored energy
$= \frac{1}{2} \int_v \{D\} \cdot \{E\}^* \, dV$

The surface impedance, Z_s, responsible for surface (metallic) losses can be expressed as:

$$Z_s = R_s + jX_s = R_s\,(1 + j) \qquad (19.6\text{–}51)$$

with the surface resistance, R_s, defined as:

$$R_s = \sqrt{\frac{2\pi f_o \mu}{2\sigma}} \qquad (19.6\text{–}52)$$

where: μ = permeability
σ = metal conductivity

The surface impedance terms R_s and X_s are input via the **SF** command (IMPD label).

The surface loss, P_c, over the conducting surface is thus calculated as:

$$P_c = \frac{1}{2} \int R_s \, \{H\} \cdot \{H\}^* \, dS \qquad (19.6–53)$$

The dielectric loss, P_d, due to a lossy material is calculated as:

$$P_d = \frac{1}{2} \int_v \sigma_d \{E\} \cdot \{E\}^* \, dV \qquad (19.6–54)$$

where: σ_d = dielectric conductivity

= $2\pi \; f_o \; \epsilon_o \; \epsilon_r \; (\tan\delta)$

ε_o = free space permittivity (8.85 x 10^{-12} F/m)

ε_r = relative permittivity

tan δ = loss tangent (material property LSST on **MP** command)

The quality factor can be separated into components

$$\frac{1}{Q} = \frac{1}{Q_c} + \frac{1}{Q_d} \qquad (19.6–55)$$

where: Q_c = conductivity quality factor

= $2\pi \; f_o \; W/P_m$

Q_d = dielectric quality factor

= $2\pi \; f_o \; W/P_d$

19.7 POST1 — Error Approximation Technique

19.7.1 Error Approximation Technique for Displacement–Based Problems

The error approximation technique used by POST1 (**PRERR** command) for displacement–based problems is similar to that given by Zienkiewicz and Zhu(102). The essentials of the method are summarized below.

The usual continuity assumption used in many displacement based finite element formulations results in a continuous displacement field from element to element, but a discontinuous stress field. To obtain more acceptable stresses, averaging of the element nodal stresses is done. Then, returning to the element level, the stresses at each node of the element are processed to yield:

$$\left\{\Delta\sigma_n^i\right\} = \left\{\sigma_n^a\right\} - \left\{\sigma_n^i\right\} \qquad (19.7\text{–}1)$$

where:
$\left\{\Delta\sigma_n^i\right\}$ = stress error vector at node n of element i

$\left\{\sigma_n^a\right\}$ = averaged stress vector at node n = $\dfrac{\sum_{i=1}^{N_e^n} \left\{\sigma_n^i\right\}}{N_e^n}$

N_e^n = number of elements connecting to node n

$\left\{\sigma_n^i\right\}$ = stress vector of node n of element i

Then, for each element

$$e_i = \frac{1}{2}\int_{vol} \{\Delta\sigma\}^T \, [D]^{-1} \, \{\Delta\sigma\} \; d(vol) \qquad (19.7\text{–}2)$$

where:
e_i = energy error for element i (accessed with **ETABLE** (SERR item) command)

vol = volume of the element (accessed with **ETABLE** (VOLU item) command)

[D] = stress–strain matrix evaluated at reference temperature

$\{\Delta\sigma\}$ = stress error vector at points as needed (evaluated from all $\{\Delta\sigma_n\}$ of this element)

The energy error over the model is:

$$e = \sum_{i=1}^{N_r} e_i \tag{19.7–3}$$

where: e = energy error over the entire (or part of the) model (accessed with ***GET** (SERSM item) command)

N_r = number of elements in model or part of model

The energy error can be normalized against the strain energy.

$$E = 100\left(\frac{e}{U + e}\right)^{\frac{1}{2}} \tag{19.7–4}$$

where: E = percentage error in energy norm (accessed with **PRERR**, **PLDISP**, **PLNSOL** (U item), ***GET** (SEPC item) commands)

U = strain energy over the entire (or part of the) model (accessed with ***GET** (SENSM item) command)

$= \sum_{i=1}^{N_r} E_{ei}^{po}$

E_{ei}^{po} = strain energy of element i (accessed with **ETABLE** (SENE item) command) (see Section 15.16)

The e_i values can be used for adaptive mesh refinement. It has been shown by Babuska and Rheinboldt(103) that if e_i is equal for all elements, then the model using the given number of elements is the most efficient one. This concept is also referred to as "error equilibration".

At the bottom of all nodal stresses printed out with the **PRNSOL** or **PRESOL** command, which may consist of the 6 component stresses, the 5 combined stresses, or both, a summary printout labeled: ESTIMATED BOUNDS CONSIDERING THE EFFECT OF DISCRETIZATION ERROR gives minimum nodal values and maximum nodal values. These are:

$$\sigma_j^{mnb} = \min\left(\sigma_{j,n}^a - \Delta\sigma_n\right) \tag{19.7–5}$$

$$\sigma_j^{mxb} = \max\left(\sigma_{j,n}^a + \Delta\sigma_n\right) \tag{19.7–6}$$

where min and max are over the selected nodes, and where: σ_j^{mnb} = output quantity VALUE (printout) or SMNB (plot) for nodal minimum of stress quantity

σ_j^{mxb} = output quantity VALUE (printout) or SMXB (plot) for nodal maximum of stress quantity

j = subscript to refer to either a particular stress component or a particular combined stress

$$\sigma_{j,n}^{a} = \begin{cases} \sigma_{j,n}^{avg} \text{ if the } \textbf{PLNSOL} \text{ or } \textbf{PRNSOL} \text{ command is used} \\ \sigma_{j,n}^{max} \text{ if the } \textbf{PLESOL} \text{ command is used} \end{cases}$$

$\sigma_{j,n}^{avg}$ = average of stress quantity j at node n of element attached to node n

$\sigma_{j,n}^{max}$ = maximum of stress quantity j at node n of element attached to node n

$\Delta\sigma_n$ = root mean square of all $\Delta\sigma_i$ from elements connecting to node n

$\Delta\sigma_i$ = maximum absolute value of any component of $\{\Delta\sigma_n^i\}$ for all nodes connecting to element (accessed with **ETABLE** (SDSG item) command)

19.7.2 Error Approximation Technique for Temperature–Based Problems

The error approximation technique used by POST1 (**PRERR** command) for temperature based problems in similar to that given by Huang and Lewis(126). The essentials of the method are summarized below.

The usual continuity assumption results in a continuous temperature field from element to element, but a discontinuous thermal flux field. To obtain more acceptable fluxes, averaging of the element nodal thermal fluxes is done. Then, returning to the element level, the thermal fluxes at each node of the element are processed to yield:

$$\{\Delta q_n^i\} = \{q_n^a\} - \{q_n^i\} \qquad (19.7\text{–}7)$$

where: $\{\Delta q_n^i\}$ = thermal flux error vector at node n of element i

$\{q_n^a\}$ = averaged thermal flux vector at node n = $\dfrac{\sum_{i=1}^{N_e^n} \{q_n^i\}}{N_e^n}$

N_e^n = number of elements connecting to node n

$\{q_n^i\}$ = thermal flux vector of node n of element

Then, for each element

$$e_i = \frac{1}{2}\int_{vol} \{\Delta q\}^T \, [D]^{-1} \, \{\Delta q\} \;\; d(vol) \qquad (19.7\text{–}8)$$

where:

e_i = energy error for element i (accessed with **ETABLE** (TERR item) command)

vol = volume of the element (accessed with **ETABLE** (VOLU item) command)

[D] = conductivity matrix evaluated at reference temperature

$\{\Delta q\}$ = thermal flux error vector at points as needed (evaluated from all $\{\Delta q_n\}$ of this element)

The energy error over the model is:

$$e = \sum_{i=1}^{N_r} e_i \qquad (19.7\text{–}9)$$

where:

e = energy error over the entire (or part of the) model (accessed with ***GET** (TERSM item) command)

N_r = number of elements in model or part of model

The energy error can be normalized against the thermal dissipation energy.

$$E = 100\left(\frac{e}{U + e}\right)^{\frac{1}{2}} \qquad (19.7\text{–}10)$$

where:

E = percentage error in energy norm (accessed with **PRERR**, **PLNSOL,** (TEMP item) or ***GET** (TEPC item) commands)

U = thermal dissipation energy over the entire (or part of the) model (accessed with ***GET** (TENSM item) command)

$$= \sum_{i=1}^{N_r} E_{ei}^{po}$$

E_{ei}^{po} = thermal dissipation energy of element i (accessed with **ETABLE** (TENE item) command) (see Section 15.16)

The e_i values can be used for adaptive mesh refinement. It has been shown by Babuska and Rheinboldt(103) that if e_i is equal for all elements, then the model using the given number of elements is the most efficient one. This concept is also referred to as “error equilibration”.

At the bottom of all nodal fluxes printed out with the **PRNSOL** command, which consists of the 3 thermal fluxes, a summary printout labeled: ESTIMATED BOUNDS CONSIDERING THE EFFECT OF DISCRETIZATION ERROR gives minimum nodal values and maximum nodal values. These are:

$$q_j^{mnb} = \min \left(q_{j,n}^a - \Delta q_n \right) \tag{19.7–11}$$

$$q_j^{mxb} = \max \left(q_{j,n}^a + \Delta q_n \right) \tag{19.7–12}$$

where min and max are over the selected nodes, and

where: q_j^{mnb} = output quantity VALUE (printout) or SMNB (plot) for nodal minimum of thermal flux quantity

q_j^{mxb} = output quantity VALUE (printout) or SMXB (plot) for nodal maximum of thermal flux quantity

j = subscript to refer to either a particular thermal flux component or a particular combined thermal flux

$$q_{j,n}^a = \begin{cases} q_{j,n}^{avg} \text{ if the \textbf{PLNSOL} or \textbf{PRNSOL} command is used} \\ q_{j,n}^{max} \text{ if the \textbf{PLESOL} command is used} \end{cases}$$

$q_{j,n}^{avg}$ = average of thermal flux quantity j at node n of element attached to node n

$q_{j,n}^{max}$ = maximum of thermal flux quantity j at node n of element attached to node n

Δq_n = maximum of all Δq_i from elements connecting to node n

Δq_i = maximum absolute value of any component of $\left\{\Delta q_n^i\right\}$ for all nodes connecting to element (accessed with **ETABLE** (TDSG item) command)

19.8 POST1 — Crack Analysis (KCALC Command)

The **KCALC** Command determines the stress intensity factors at a crack for a linear elastic fracture mechanics analysis. The analysis uses a fit of the nodal displacements in the vicinity of the crack. The actual displacements at and near a crack for linear elastic materials are (Paris and Sih(106)):

$$u = \frac{K_I}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa - 1)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}\right) - \frac{K_{II}}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa + 3)\sin\frac{\theta}{2} + \sin\frac{3\theta}{2}\right) + 0(r) \tag{19.8–1}$$

$$v = \frac{K_I}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa - 1)\sin\frac{\theta}{2} - \sin\frac{3\theta}{2}\right) - \frac{K_{II}}{4G}\sqrt{\frac{r}{2\pi}}\left((2\kappa + 3)\cos\frac{\theta}{2} + \cos\frac{3\theta}{2}\right) + 0(r) \tag{19.8–2}$$

$$w = \frac{2K_{III}}{G}\sqrt{\frac{r}{2\pi}}\sin\frac{\theta}{2} + 0(r) \tag{19.8–3}$$

where: u,v,w = displacements in a local Cartesian coordinate system as shown in Figure 19.8–1.

r, θ = coordinates in a local cylindrial coordinate system also shown in Figure 19.8–1.

G = shear modulus

K_I, K_{II}, K_{III} = stress intensity factors relating to deformation shapes shown in Figure 19.8–2

$$\varkappa = \begin{cases} 3 - 4\nu & \text{if plane strain or axisymmetric} \\ \dfrac{3-\nu}{1+\nu} & \text{if plane stress} \end{cases}$$

ν = Poisson's ratio

0(r) = terms of order r or higher

Evaluating equations (19.8–1) thru (19.8–3) at θ = ± 180.0° and dropping the higher order terms yields:

$$u = +\frac{K_{II}}{2G}\sqrt{\frac{r}{2\pi}}(1 + \kappa) \tag{19.8–4}$$

$$v = + \frac{K_I}{2G} \sqrt{\frac{r}{2\pi}} \; (1 + \kappa) \tag{19.8–5}$$

$$w = + \frac{2\ K_{III}}{G} \sqrt{\frac{r}{2\pi}} \tag{19.8–6}$$

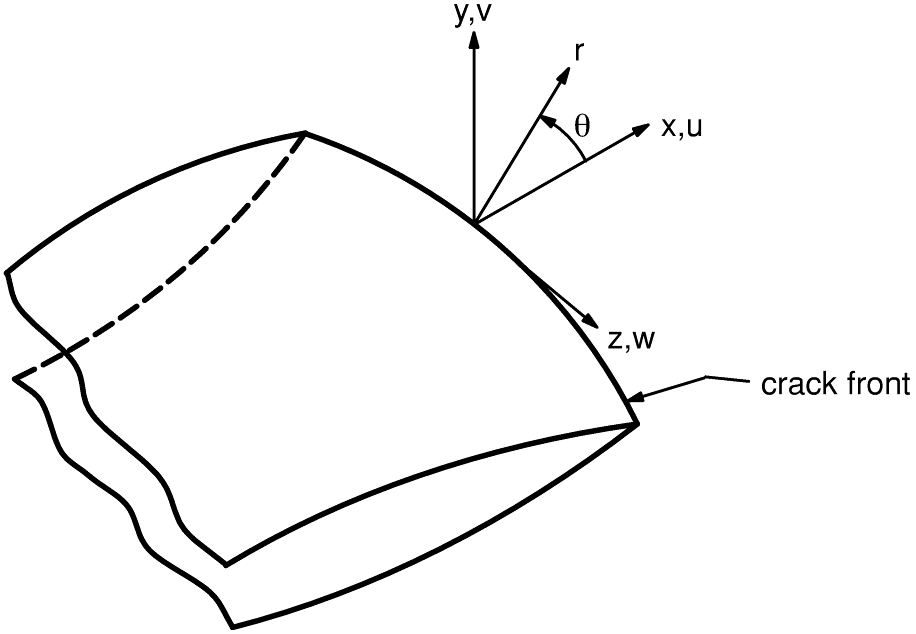


Figure 19.8–1 Local Coordinates Measured From a 3–D Crack Front

The crack width is shown greatly enlarged, for clarity.

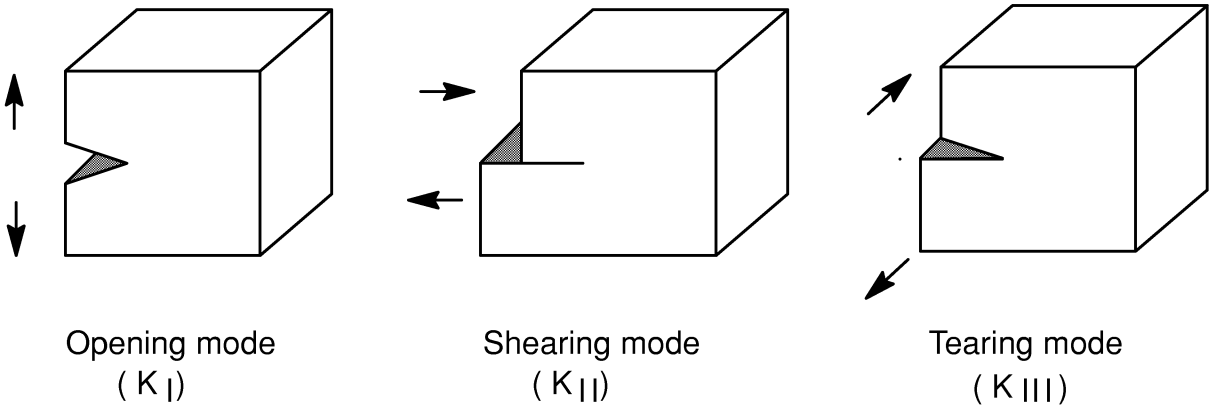


Figure 19.8–2 The Three Basic Modes of Fracture

For models symmetric about the crack plane (half–crack model, Figure 19.8–3(a)), equations (19.8–4) to (19.8–6) can be reorganized to give:

$$K_I = \sqrt{2\pi}\ \frac{2\ G}{1\ +\ \kappa}\ \frac{|v|}{\sqrt{r}} \tag{19.8–7}$$

$$K_{II} = \sqrt{2\pi}\ \frac{2\ G}{1\ +\ \kappa}\ \frac{|u|}{\sqrt{r}} \tag{19.8–8}$$

$$K_{III} = \sqrt{2\pi}\ 2G\ \frac{|w|}{\sqrt{r}} \tag{19.8–9}$$

and for the case of no symmetry (full–crack model, Figure 19.8–3(b)),

$$K_I = \sqrt{2\pi}\ \frac{G}{1\ +\ \kappa}\ \frac{|\Delta v|}{\sqrt{r}} \tag{19.8–10}$$

$$K_{II} = \sqrt{2\pi}\ \frac{G}{1\ +\ \kappa}\ \frac{|\Delta u|}{\sqrt{r}} \tag{19.8–11}$$

$$K_{III} = \sqrt{2\pi}\ \frac{G}{1\ +\kappa}\ \frac{|\Delta w|}{\sqrt{r}} \tag{19.8–12}$$

where Δv, Δu, and Δw are the motions of one crack face with respect to the other.

As the above six equations are similar, consider only the first one further. The final factor is $\frac{|v|}{\sqrt{r}}$, which needs to be evaluated based on the nodal displacements and locations. As shown in Figure 19.8–3(a), three points are available. v is normalized so that v at node I is zero. Then A and B are determined so that

$$\frac{|v|}{\sqrt{r}} = A\ +\ B\ r \tag{19.8–13}$$

at points J and K. Next, let r approach 0.0:

$$\lim_{r \to o} \frac{|v|}{\sqrt{r}} = A \tag{19.8–14}$$

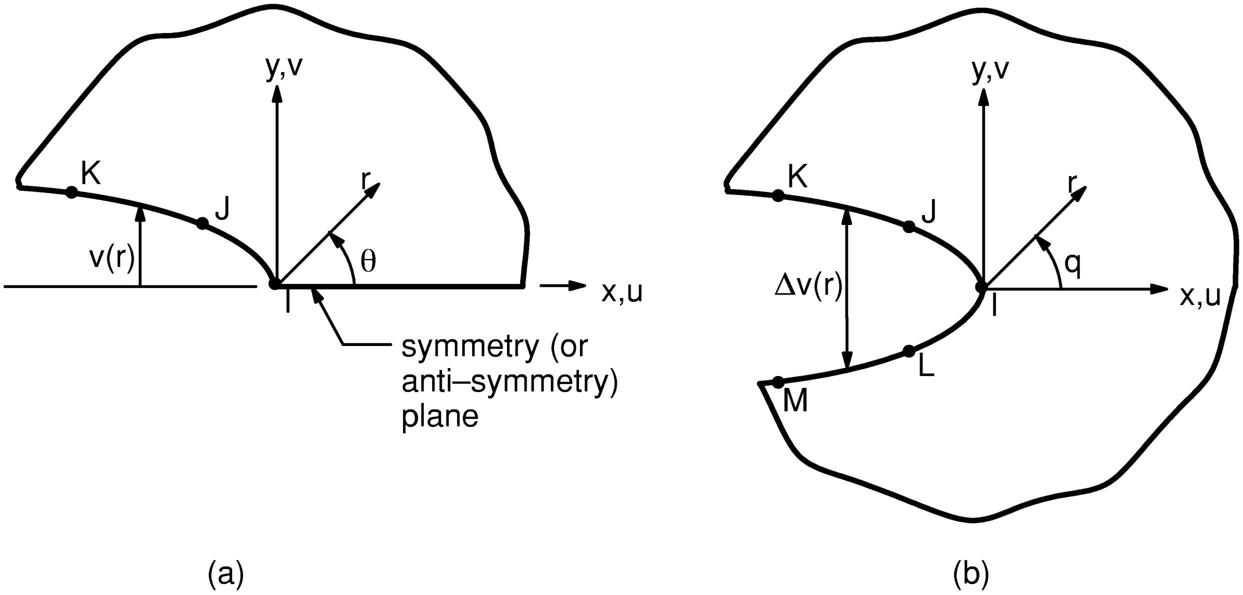


Figure 19.8–3 Nodes Used for the Approximate Crack–Tip Displacements: (a) Half Model, (b) Full Model

Thus, equation (19.8–7) becomes:

$$K_I = \sqrt{2\pi}\ \frac{2\ G\ A}{1 + \kappa} \qquad (19.8\text{–}15)$$

Equations (19.8–8) thru (19.8–12) are also fit in the same manner.

19.9 POST1 — Harmonic Solid and Shell Element Postprocessing

19.9.1 Thermal Solid Elements (PLANE75 and PLANE78)

Data processed in a harmonic fashion includes nodal temperatures, element data stored on a per node basis (thermal gradient and thermal flux) and nodal heat flow. Nodal temperature is calculated at harmonic angle θ for each node j.

$$T_{j\theta} = (\mathrm{FACT})(\mathrm{K})T_j \qquad (19.9\text{–}1)$$

where: FACT = scaling factor (input quantity *FACT*, **SET** command)

K = cos nθ if mode is symmetric (input quantity *ISYM*=1 on **MODE** command
sin nθ if mode is antisymmetric (input quantity *ISYM*=–1 on **MODE** command

n = mode number (input quantity *MODE* on **MODE** command)

θ = angle at which harmonic calculation is being made (input quantity *ANGLE*, **SET** command)

Thermal gradient (TG) is calculated at harmonic angle θ for each node k of element j:

$$\mathrm{TG}(x)_{jk\theta} = (\mathrm{FACT})(\mathrm{K})\mathrm{TG}(x)_{jk} \qquad (19.9\text{–}2)$$

$$\mathrm{TG}(y)_{jk\theta} = (\mathrm{FACT})(\mathrm{K})\mathrm{TG}(y)_{jk} \qquad (19.9\text{–}3)$$

$$\mathrm{TG}(z)_{jk\theta} = (\mathrm{FACT})(\mathrm{L})\mathrm{TG}(z)_{jk} \qquad (19.9\text{–}4)$$

where: L = sin nθ if mode is symmetric (input quantity *ISYM*=1 on **MODE** command
cos nθ if mode is antisymmetric (input quantity *ISYM*=–1 on **MODE** command

Nodal heat flow is processed in the same way as temperature. Thermal flux is processed in the same way as thermal gradient.

19.9.2 Structural Solid Elements (PLANE25 and PLANE83)

Data processed in a harmonic fashion include nodal displacements, nodal forces, and element data stored on a per node basis (stress and elastic strain).

Nodal displacement is calculated at harmonic angle θ for each node j:

$$U(x)_{j\theta} = (FACT)(K)U(x)_j \tag{19.9–5}$$

$$U(y)_{j\theta} = (FACT)(K)U(y)_j \tag{19.9–6}$$

$$U(z)_{j\theta} = (FACT)(L)U(z)_j \tag{19.9–7}$$

Stress is calculated at harmonic angle θ for each node k of element j:

$$S(x)_{jk\theta} = (FACT)(K)S(x)_j \tag{19.9–8}$$

$$S(y)_{jk\theta} = (FACT)(K)S(y)_j \tag{19.9–9}$$

$$S(z)_{jk\theta} = (FACT)(K)S(z)_j \tag{19.9–10}$$

$$S(xy)_{jk\theta} = (FACT)(K)S(xy)_j \tag{19.9–11}$$

$$S(yz)_{jk\theta} = (FACT)(L)S(yz)_j \tag{19.9–12}$$

$$S(xz)_{jk\theta} = (FACT)(L)S(xz)_j \tag{19.9–13}$$

Nodal forces are processed in the same way as nodal displacements. Strains are processed in the same way as stresses.

19.9.3 Structural Shell Element (SHELL61)

Data processed in a harmonic fashion include displacements, nodal forces, member forces, member moments, in–plane element forces, out–of–plane element moments, stress, and elastic strain.

Nodal displacement is calculated at harmonic angle θ for each node j:

$$U(x)_{j\theta} = (FACT)(K)U(x)_j \tag{19.9–14}$$

$$U(y)_{j\theta} = (FACT)(K)U(y)_j \tag{19.9–15}$$

$$U(z)_{j\theta} = (FACT)(L)U(z)_j \tag{19.9–16}$$

$$ROT(z)_{j\theta} = (FACT)(K)ROT(z)_{j\theta} \tag{19.9–17}$$

Stress is calculated at harmonic angle θ for each node/interior point k of element j:

$$S(M)_{jk\theta} = (FACT)(K)S(M)_{jk} \tag{19.9–18}$$

$$S(H)_{jk\theta} = (FACT)(K)S(H)_{jk} \tag{19.9–19}$$

$$S(THK)_{jk\theta} = (FACT)(K)S(THK)_{jk} \tag{19.9–20}$$

$$S(MH)_{jk\theta} = (FACT)(L)S(MH)_{jk} \tag{19.9–21}$$

In–plane element forces at harmonic angle θ for each node/interior point k of element j:

$$T(x)_{jk\theta} = (FACT)(K)T(x)_{jk} \tag{19.9–22}$$

$$T(z)_{jk\theta} = (FACT)(K)T(z)_{jk} \tag{19.9–23}$$

$$T(xz)_{jk\theta} = (FACT)(L)T(xz)_{jk} \tag{19.9–24}$$

Nodal forces, member forces, and member moments are processed in the same way as nodal displacements. Strains are processed in the same way as stresses. Finally, out–of–plane element moments are processed in the same way as in–plane element forces.

19.10 POST26 — Data Operations

Table 19.10–1 shows the operations that can be performed on the time–history data stored by POST26. Input quantities *FACTA*, *FACTB*, *FACTC*, and table *IC* are omitted from Table 19.10–1 for clarity of the fundamental operations. All operations are performed in complex variables. The operations create new tables which are also complex numbers.

The **INT1** and **DERIV** commands operate similarly to the respective operations in the ***VOPER** Command (Section 18.1).

Table 19.10–1 POST26 Operations

POST26 Command	Description	Real Operation and Result	Complex Operation	Complex Result
ADD	Addition	$a + c$	$(a + ib) + (c + id)$	$(a + c) + i(b + d)$
PROD	Multiplication	$a \text{ x } c$	$(a + ib) \text{ x } (c + id)$	$(ac - bd) + i(ad + bc)$
QUOT	Division	$\frac{a}{c}$	$\frac{(a + ib)}{(c + id)}$	$\frac{(ac + bd) + i(-ad + bc)}{(c^2 + d^2)}$
ABS	Absolute Value	$\lvert a \rvert$	$\lvert a + ib \rvert$	$\sqrt{a^2 + b^2}$
ATAN	Arc Tangent	0	atan (a + ib)	atan (b / a)
SQRT	Square Root	$\sqrt{a}$	$\sqrt{a + ib}$	$(a^2 + b^2)^{\frac{1}{4}}(\cos\frac{\theta}{2} + i\sin\frac{\theta}{2})$
LARGE	Largest Variable	Maximum of a and c		
SMALL	Smallest Variable	Minimum of a and c		
DERIV	Derivative	$\frac{da}{dc}$	$\frac{d(a + ib)}{dc}$	$\frac{da}{dc} + \frac{i\,db}{dc}$
INT1	Integration	$\int adc$	$\int (a + ib)\ dc$	$\int adc + i\int bdc$
CLOG	Common Logarithm	$\log_{10} a$	$\log_{10} (a + ib)$	$\log_{10} e(\ell n \sqrt{a^2 + b^2} + i\theta)$
NLOG	Natural Logarithm	$\ell n\ a$	$\ell n\ (a + ib)$	$\ell n \sqrt{a^2 + b^2} + i\theta$
EXP	Exponential	e^a	$e^{(a + ib)}$	$e^a (\cos b + i \sin b)$
CONJUG	Complex Conjugate	a	conj (a + ib)	a – ib
REALVAR	Real Part	a	real (a + ib)	a
IMAGIN	Imaginary Part	0	imag (a + ib)	b
DATA	Read Data into Table	–	–	–
FILL	Fill Table with Data	–	–	–

POST26 Command	Description	Real Operation and Result	Complex Operation	Complex Result
RESP	See Response Spectrum Generator Description (Section 19.11)			
NOTES: 1. a + ib (from Table *IA or IX*) and c + id (from Table *IB or IY*) are complex numbers, where *IA* and *IB* are input quantities on above commands. 2. $\theta = \tan^{-1}\left(\frac{b}{a}\right)$				

19.11 POST26 — Response Spectrum Generator (RESP)

Given a motion as output from a transient dynamic analysis, POST26 generates a response spectrum in terms of displacement, velocity, or acceleration.

A response spectrum is generated by imposing the motion of the point of interest on a series of single–mass oscillators over a period of time and calculating the maximum displacement, velocity, or acceleration. This is illustrated in Figure 19.11–1.

In Figure 19.11–1, the following definitions are used:

M_i	=	mass of oscillator i
C_i	=	damping of oscillator i
K_i	=	stiffness of oscillator i
u_i	=	motion of oscillator i
u_b	=	motion of point of interest

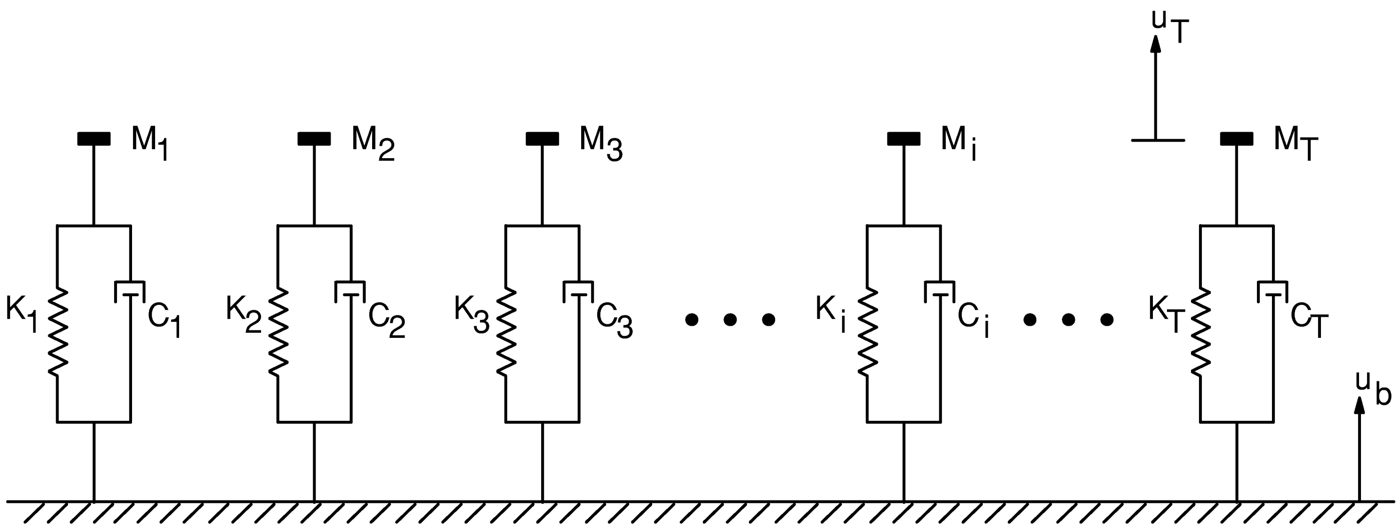


Figure 19.11–1 Single Mass Oscillators

In the absence of damping, the natural frequency of an oscillator i is:

$$\omega_i = \sqrt{\frac{K_i}{M_i}} \qquad (19.11\text{–}1)$$

The basic equation of motion of the oscillator can be given as a one degree of freedom (DOF) version of equation (17.2–1):

$$M_i \ \ddot{u}_i + C_i \dot{u}_i^r + K_i \, u_i^r = 0 \qquad (19.11–2)$$

where: a dot (·) over a variable = derivative with respect to time

u_i^r, the relative motion of oscillator i, is defined by:

$$u_i^r = u_i - u_b \qquad (19.11–3)$$

The damping is given by:

$$\xi_i = \frac{C_i}{C_{cr,i}} \qquad (19.11–4)$$

where: $C_{cr,i} = 2\sqrt{K_i \ M_i}$ = critical damping coefficient

Equation (19.11–1) thru (19.11–4) are combined to give:

$$\ddot{u}_i^r + 2\xi_i \ \omega_i \ \dot{u}_i^r + \omega_i^2 \ u_i^r = -\ddot{u}_b \qquad (19.11–5)$$

This equation is solved essentially as a linear transient dynamic analysis (**ANTYPE**,TRANS with **TRNOPT**,REDUC).

19.11.1 Time Step Size

The time step size (Δt) is selected in the following way. If data is from a full transient analysis (**ANTYPE**,TRANS with **TRNOPT**,FULL):

Δt = input quantity *DTIME* on **RESP** command

or if DTIME is not given:

$$\Delta t = \frac{1}{20 f_{max}} \qquad (19.11–6)$$

where: f_{max} = highest frequency value of table *LFTAB* defined with command **RESP**.

If the data is from a reduced transient analysis (**ANTYPE**,TRANS with **TRNOPT**,REDUC), Δt is the integration time step size used in the analysis (**DELTIM** command)

The transient data from an **ANTYPE**,TRANS with **TRNOPT**, FULL analysis is taken from the next available time step used in the analysis. This can cause a decrease in accuracy at higher frequencies if Δt is less than the time step size of the input transient.

19.12 POST1 and POST26 — Interpretation of Equivalent Strains

The equivalent strain for the strains EPEL, EPPL, EPCR, EPTH, and EPTOT is computed in postprocessing using the von Mises equation:

$$\epsilon_{eq} = \frac{1}{\sqrt{2}\ (1+\nu')}\left[(\epsilon_x - \epsilon_y)^2 + (\epsilon_y - \epsilon_z)^2 + (\epsilon_z - \epsilon_x)^2 + \frac{3}{2}\,(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{xz}^2)\right]^{\frac{1}{2}} \qquad (19.12\text{–}1)$$

where: ε_x, ε_y, etc. = appropriate component strain values
ν' = Effective Poisson's ratio input as EFFNU on the **AVPRIN** command (POST1 only)

The equivalent strain is output with the EQV or PRIN component label in POST1 using the **PRNSOL**, **PLNSOL**, **PDEF**, or **ETABLE** commands and in POST26 using the **ESOL** command.

19.12.1 Physical Interpretation of Equivalent Strain

The von Mises equation is a measure of the "shear" strain in the material and does not account for the hydrostatic straining component. For example, strain values of $\varepsilon_x = \varepsilon_y = \varepsilon_z = 0.001$ yield an equivalent strain $\varepsilon_{eq} = 0.0$.

19.12.2 Elastic Strain

The equivalent elastic strain is related to the equivalent stress when $\nu' = \nu$ (input as PRXY or NUXY on **MP** command) by:

$$\sigma_{eq} = E\,\epsilon_{eq}^{el} \qquad (19.12\text{–}2)$$

where: σ_{eq} = equivalent stress (output using SEQV)
ϵ_{eq}^{el} = equivalent elastic strain (output using EPEL, EQV)
E = Young's modulus

Note that when $\nu' = 0$ (the default value) then the equivalent elastic strain is related via

$$\sigma_{eq} = 2G\,\epsilon_{eq}^{el} \qquad (19.12\text{–}3)$$

where: G = shear modulus

19.12.3 Plastic Strain

For plasticity, the accumulated effective plastic strain is defined by (see equations (4.1–22) and (4.1–37)):

$$\epsilon^{pl}_{eqa} = \sum \Delta\, \epsilon^{pl}_{eq} \qquad (19.12\text{–}4)$$

where: ϵ^{pl}_{eqa} = accumulated effective plastic strain (output using NL, EPEQ (or NL, PSV for VISCO106, VISCO107, and VISCO108 when Anand's model is not used))

$$\Delta\epsilon^{pl}_{eqa} = \frac{\sqrt{2}}{3}\left[(\Delta\epsilon^{pl}_{x}–\Delta\epsilon^{pl}_{y})^2 + (\Delta\epsilon^{pl}_{y}–\Delta\epsilon^{pl}_{z})^2 + (\Delta\epsilon^{pl}_{z}–\Delta\epsilon^{pl}_{x})^2 + \frac{3}{2}\left({\Delta\gamma^{pl}_{xy}}^2 + {\Delta\gamma^{pl}_{yz}}^2 + {\Delta\gamma^{pl}_{xz}}^2\right)\right]^{\frac{1}{2}}$$

This can be related to ϵ^{pl}_{eq} (output using EPPL, EQV) only under proportional loading situations during the initial loading phase and only when ν' is set to 0.5.

19.12.4 Creep Strain

As with the plastic strains, to compute the equivalent creep strain (EPCR,EQV), use $\nu' = 0.5$.

19.12.5 Total Strain

The equivalent total strain (output using EPTOT, EQV) in an analysis with plasticity or creep should use an appropriate value of ν'. If $\epsilon^{pl}_{eq} >> \epsilon^{el}_{eq}$, use $\nu' = 0.5$. For other values, use an effective Poisson's ratio between ν and 0.5. One method of estimating this is through:

$$\nu' = \frac{1}{2} - \left(\frac{1}{2} - \nu\right)\frac{\epsilon^{el}_{eq}}{\epsilon^{tot}_{eq}} \qquad (19.12\text{–}5)$$

19.13 POST26 — Response Power Spectral Density (RPSD)

The cross response PSD between two items is computed using the equation:

$$\begin{aligned} S_{pq}(\omega) = & \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\left(\phi_{pj}\,\phi_{qk} + \phi_{qj}\,\phi_{pk}\right)}{2} R_{jk}(\omega) \\ & + \sum_{l=1}^{r_2} \sum_{m=1}^{r_2} \frac{\left(A_{pl}\,A_{qm} + A_{ql}\,A_{pm}\right)}{2} \overline{R}_{lm}(\omega) \\ & + \sum_{j=1}^{n} \sum_{l=1}^{r_2} \frac{\left(\phi_{pj}\,A_{ql} + \phi_{qj}\,A_{pl}\right)}{2} \hat{R}_{jl}(\omega) \end{aligned} \tag{19.13–1}$$

where: p = input quantity IA on **RPSD** command

q = input quantity IB on **RPSD** command

p and q can be displacements, stresses, or reaction forces.

All other variables in equation (19.13–1) are defined in Section 17.7. When p = q, the above cross response PSD becomes the auto response PSD.

19.14 POST26 — Computation of Covariance (CVAR)

The covariance between two items p and q is computed using the equation:

$$\sigma^2_{pq} = \sum_{j=1}^{n} \sum_{k=1}^{n} \frac{\left(\phi_{pj}\,\phi_{qk} + \phi_{qj}\,\phi_{pk}\right)}{2} Q_{jk}\,(\omega) + \sum_{l=1}^{r_2} \sum_{m=1}^{r_2} \frac{\left(A_{pl}\,A_{qm} + A_{ql}\,A_{pm}\right)}{2} \overline{Q}_{lm}\,(\omega) + \sum_{j=1}^{n} \sum_{l=1}^{r_2} \left(\phi_{pj}\,A_{ql} + \phi_{qj}\,A_{pl}\right) \hat{Q}_{jl}\,(\omega) \qquad (19.14\text{–}1)$$

where:
p = input quantity IA on **CVAR** command
q = input quantity IB on **CVAR** command
p and q can be displacements, stresses, or reaction forces.

All other variables in equation (19.14–1) are defined in Section 17.7. When p = q, the above covariance becomes the variance.

Chapter 20
Design Optimization

ANSYS Theory Reference
Chapter 20 – Table of Contents

20.0 Introduction to Optimization

In ANSYS, there are two fundamentally different types of optimization. This chapter is designed to give users a basic understanding of the overall theory for both types.

The first is referred to as design optimization; it works entirely with the ANSYS Parametric Design Language (APDL) and is contained within its own module (/**OPT**). Design optimization is largely concerned with controlling user–defined, APDL functions/parameters that are to be constrained or minimized using standard optimization methods (e.g., function minimization, gradients, design of experiments). Sections 20.1 to 20.8 describe the theoretical underpinnings for design optimization.

Section 20.9 describes a technique known as topological optimization. This is a form of shape optimization. It is sometimes referred to as layout optimization in the literature. The goal of topological optimization is to find the best use of material for a body such that an objective criteria (i.e. global stiffness, natural frequency, etc.) takes out a maximum/mimimum value subject to given constraints (i.e. volume reduction). Topological optimization is not part of the design optimization module (/**OPT**) but works within the bounds of the standard ANSYS preprocessing, solution, and post processing structures **(**/PREP**,** /SOLUTION, and /POST1**)**, and it does not require ADPL.

20.1 Introduction to Design Optimization

The optimization module (/**OPT**) is an integral part of the ANSYS program that can be employed to determine the optimum design. This optimum design is the best design in some predefined sense. Among many examples, the optimum design for a frame structure may be the one with minimum weight or maximum frequency; in heat transfer, the minimum temperature; or in magnetic motor design, the maximum peak torque. In many other situations minimization of a single function may not be the only goal, and attention must also be directed to the satisfaction of predefined constraints placed on the design (e.g., limits on stress, geometry, displacement, heat flow).

While working towards an optimum design, the ANSYS optimization routines employ three types of variables that characterize the design process: design variables, state variables, and the objective function. These variables are represented by scalar parameters in ANSYS Parametric Design Language (APDL). The use of ADPL is an essential step in the optimization process.

The independent variables in an optimization analysis are the design variables. The vector of design variables is indicated by:

$$\mathbf{x} = \lfloor x_1 \quad x_2 \quad x_3 \quad . \; . \; . \; x_n \rfloor \tag{20.1–1}$$

Design variables are subject to *n* constraints with upper and lower limits, that is,

$$\underline{x_i} \;\leq\; x_i \;\leq\; \overline{x_i} \qquad (i \;=\; 1, 2, 3, \ldots, n) \tag{20.1–2}$$

where: n = number of design variables.

The design variable constraints are often referred to as side constraints and define what is commonly called feasible design space.

Now, minimize

$$f \;=\; f(\mathbf{x}) \tag{20.1–3}$$

subject to

$$g_i(\mathbf{x}) \;\leq\; \overline{g_i} \qquad (i \;=\; 1, 2, 3, \ldots, m_1) \tag{20.1–4}$$

$$\underline{h}_i \leq h_i(\mathbf{x}) \qquad (i = 1,2,3,...,m_2) \tag{20.1–5}$$

$$\underline{w}_i \leq w_i(\mathbf{x}) \leq \overline{w}_i \qquad (i = 1,2,3,\ldots,m_3) \tag{20.1–6}$$

where: f = objective function

g_i, h_i, w_i = state variables containing the design, with underbar and overbars representing lower and upper bounds respectively (input as MIN, MAX on **OPVAR** command)

$m_1 + m_2 + m_3$ = number of state variables constraints with various upper and lower limit values

The state variables can also be referred to as dependent variables in that they vary with the vector **x** of design variables.

Equations (20.1–3) through (20.1–6) represent a constrained minimization problem whose aim is the minimization of the objective function f under the constraints imposed by equations (20.1–2), (20.1–4), (20.1–5), and (20.1–6)

20.1.1 Feasible versus Infeasible Design Sets

Design configurations that satisfy all constraints are referred to as feasible designs. Design configurations with one or more violations are termed infeasible. In defining feasible design space, a tolerance is added to each state variable limit. So if **x*** is a given design set defined as

$$\mathbf{x}^* = \lfloor x_1^* \;\; x_2^* \;\; x_3^* \;\; ... \;\; x_n^* \rfloor \tag{20.1–7}$$

The design is deemed feasible only if

$$g_i^* = g_i(\mathbf{x}^*) \leq \overline{g}_i + \alpha_i \qquad (i = 1,2,3,\ldots,m_1) \tag{20.1–8}$$

$$\underline{h}_i - \beta_i \leq h_i^* = h_i(\mathbf{x}^*) \qquad (i = 1,2,3,\ldots,m_2) \tag{20.1–9}$$

$$\underline{w}_i - \gamma_i \leq w_i^* = w_i(\mathbf{x}^*) \leq \overline{w}_{\bar{i}} + \gamma_i \qquad (i = 1,2,3,\ldots,m_3) \tag{20.1–10}$$

where: α_i, β_i, and γ_i = tolerances (input as TOLER on **OPVAR** command).

and: $$\underline{x}_i \leq x_i^* \leq \overline{x}_i \qquad (i = 1,2,3,\ldots,n) \tag{20.1–11}$$

(since no tolerances are added to design variable constraints)

Equations (20.1–8) to (20.1–11) are the defining statements of a feasible design set in the ANSYS optimization routines.

20.1.2 The Best Design Set

As design sets are generated by methods or tools (discussed below) and if an objective function is defined, the *best* design set is computed and its number is stored. The best set is determined under one of the following conditions.

1. If one or more feasible sets exist the best design set is the feasible one with the lowest objective function value. In other words, it is the set that most closely agrees with the mathematical goals expressed by equations (20.1–3) to (20.1–6).
2. If all design sets are infeasible, the best design set is the one closest to being feasible, irrespective of its objective function value.

20.1.3 Optimization Methods and Design Tools

The ANSYS optimization procedure offers several methods and tools that in various ways attempt to address the mathematical problem stated above. ANSYS optimization methods perform actual minimization of the objective function of equation (20.1–3). It will be shown that they transform the constrained problem into an unconstrained one that is eventually minimized. Design tools, on the other hand, do not directly perform minimization. Use of the tools offer alternate means for understanding design space and the behavior of the dependent variables. Methods and tools are discussed in the sections that follow.

20.2 Single–Loop Analysis Tool

This is a simple and very direct tool for understanding design space. It is not necessary but it may be useful to compute values of state variables or the objective function. The design variables are all explicitly defined by the user. A single loop is equivalent to one complete finite element analysis (FEA) (i.e., one or more entries into /**PREP7**, /**SOLUTION**, /**POST1**, and /**POST26** analyses), and is selected with the **OPTYPE**,RUN command.

At the beginning of each iteration, the user defines design variable values,

$$\mathbf{x} = \mathbf{x}^* = \text{design variables defined by the user} \qquad (20.2\text{–}1)$$

and executes a single loop or iteration. If either state variables or the objective function are defined, corresponding g_i^*, h_i^*, w_i^*, and f^* values will result.

20.3 Random Tool

This design tool will fill the design variable vector with randomly generated values each iteration.

$$\mathbf{x} = \mathbf{x}^* = \text{vector generated at random} \tag{20.3–1}$$

in which case f^*, g_i^*, h_i^*, and w_i^* (if defined) will take on values corresponding to $\mathbf{x}^*$. The objective function and state variables do not need to be defined, but it can be useful to do so if actual optimization is intended to be performed subsequently. Each random design iteration is equivalent to one complete analysis loop. Random iterations continue until either one of the following conditions is satisfied. This method is selected with the **OPTYPE**,RAND command.

$$n_r = N_r \tag{20.3–2}$$

$$n_f = N_f \quad \text{if } N_f \geq 1 \tag{20.3–3}$$

where:

n_r = number of random iterations performed per each execution

n_f = total number of feasible design sets (including feasible sets from previous executions)

N_r = maximum number of iterations (input as NITR on the **OPRAND** command)

N_f = desired number of feasible design sets (input as NFEAS on the **OPRAND** command)

20.4 Sweep Tool

The sweep tool is used to scan global design space that is centered on a user–defined, reference design set. The method is selected via the **OPTYPE**,SWEEP command. Upon execution, a sweep is made in the direction of each design variable while holding all other design variables fixed at their reference values. The state variables and the objective function are computed and stored for subsequent display at each sweep evaluation point.

A sweep execution will produce n_s design sets calculated from

$$n_s = nN_s \qquad (20.4\text{–}1)$$

where: n = number of design variables

N_s = number of evaluations to be made in the direction of each design variable (input as NSPS on the **OPSWEEP** command)

For example, consider a portion of a sweep that is performed for design variable k. For simplicity, let the resulting designs sets be number as m+1, m+2, etc., where m is all the sets that existed prior to this part of the sweep. The design variables of a given design set m+i would be expressed as:

$$\mathbf{x}^{(m+i)} = \mathbf{x}^{(r)} + (i-1)\,\mathbf{\Delta}\, x_k\mathbf{e}^{(k)} \qquad (i = 1,\ 2,\ 3\ .\ .\ .\ N_s) \qquad (20.4\text{–}2)$$

where: $\mathbf{x}^{(r)}$ = reference design variables with $\underline{x}_k$ in the kth component and fixed, reference values in all other components. The rth reference design set number is input as DSET on the **OPSWEEP** command.

$\mathbf{e}^{(k)}$ = vector with 1 in its kth component and 0 for all other components

The increment of the sweep for design variable k is

$$\Delta\, x_k = (\overline{x}_k - \underline{x}_k) \,/\, (N_s - 1) \qquad (20.4\text{–}3)$$

20.5 Factorial Tool

This is a statistical tool that can be used to sample all extreme points in design space. This tool is selected by means of the **OPTYPE**,FACT command. Factorial methods are also referred to as *design of experiment* since this technology stems from the technology associated with the interpretation of experimental results. A complete review of the mathematics of this tool is not given here, and the reader is referred to Box, Hunter, and Hunter(191) for details.

Using the **OPFACT** command, the user specifies a two–level, full or a fractional factorial evaluation of design space. A full factorial evaluation of n design variables will create n_f design sets, where:

$$n_f = 2^n \tag{20.5–1}$$

Every component of the design variable vector will take two extreme values; that is:

$$x_i = \overline{x}_i \text{ or } \underline{x}_i \tag{20.5–2}$$

So in a full factorial evaluation, every combination of design variable extreme values are considered in n–dimensional design space.

The number of generated design sets associated with a fractional factorial evaluation is expressed as:

$$n_f = 2^n / M \qquad (M = 2,\ 4,\ 8\ .\ .\ .\ ) \tag{20.5–3}$$

Hence, a 1/2 fractional factorial evaluation (M=2) will yield half the number of design sets of a full evaluation.

Results from a factorial tool consist of printed output (**OPRFA** command) and bar chart displays (**OPLFA** command), showing main effects, and 2–variable interactions (n>1), and 3–variable interactions (n>2). These effects and interactions are calculated for the state variables and the objective function (if defined). Once again, consult Box, Hunter, and Hunter(191) for further details.

20.6 Gradient Tool

The gradient tool computes the gradient of the state variables and the objective function with respect to the design variables. This tool is selected by means of the **OPTYPE**,GRAD command. A reference design set is defined as the point of evaluation for the gradient. The reference design set is defined by the DSET quantity on the **OPGRAD** command. Focusing on the objective function, for example, let the reference state be denoted as:

$$f_r\ (\mathbf{x}) \ = f\ (\mathbf{x}^{(r)}) \qquad (20.6\text{–}1)$$

The gradient of the objective function is simply expressed as:

$$\nabla f_r \ = \left\lfloor \frac{\partial f_r}{\partial x_1}\ ,\ \frac{\partial f_r}{\partial x_2}\ .\ .\ .\ \frac{\partial f_r}{\partial x_n} \right\rfloor \qquad (20.6\text{–}2)$$

With respect to each design variable, the gradient is approximated from the following forward difference.

$$\frac{\partial f_r}{\partial x_i} \ = \frac{f_r\ (\mathbf{x} + \Delta\ x_i\mathbf{e}) - f_r\ (\mathbf{x})}{\Delta\ x_i} \qquad (20.6\text{–}3)$$

where: **e** = vector with 1 in its ith component and 0 for all other components

$\Delta\ x_i$ = $\frac{\Delta\ D}{100}\ (\overline{x}_i - \underline{x}_i)$

Δ D = forward difference (in %) step size (input as DELTA on **OPGRAD** command)

Similar calculations are performed for each state variable.

20.7 Subproblem Approximation Method

This method of optimization can be described as an advanced, zero–order method in that it requires only the values of the dependent variables (objective function and state variables) and not their derivatives. The dependent variables are first replaced with approximations by means of least squares fitting, and the constrained minimization problem described in Section 20.1 is converted to an unconstrained problem using penalty functions. Minimization is then performed every iteration on the approximated, penalized function (called the subproblem) until convergence is achieved or termination is indicated. For this method each iteration is equivalent to one complete analysis loop. This method is selected with the **OPTYPE**,SUBP command.

Since the method relies on approximation of the objective function and each state variable, a certain amount of data in the form of design sets is needed. This preliminary data can be directly generated by the user using any of the other optimization tools or methods. If not defined, the method itself will generate design sets at random.

20.7.1 Function Approximations

The first step in minimizing the constrained problem expressed by equations (20.1–3) to (20.1–2) is to represent each dependent variable by an approximation, represented by the ^ notation. For the objective function,

$$\hat{f}(\mathbf{x}) = f(\mathbf{x}) + \text{error} \tag{20.7–1}$$

And similarly for the state variables,

$$\hat{g}(\mathbf{x}) = g(\mathbf{x}) + \text{error} \tag{20.7–2}$$

$$\hat{h}(\mathbf{x}) = h(\mathbf{x}) + \text{error} \tag{20.7–3}$$

$$\hat{w}(\mathbf{x}) = w(\mathbf{x}) + \text{error} \tag{20.7–4}$$

The most complex form that the approximations can take on is a fully quadratic representation with cross terms. Using the example of the objective function,

$$\hat{f} = a_0 + \sum_{i}^{n} a_i x_i + \sum_{i}^{n} \sum_{j}^{n} b_{ij} x_i x_j \tag{20.7–5}$$

The actual form of each fit varies from iteration to iteration and are determined by the program. Some user control, however, is offered by the **OPEQN** command. A weighted least squares technique is used to determine the coefficient, a_i and b_{ij}, in equation (20.7–5). For example, the weighted least squares error norm for the objective function has the form

$$E^2 = \sum_{j=1}^{n_d} \phi^{(j)} \left(f^{(j)} - \hat{f}^{(j)} \right)^2 \qquad (20.7\text{–}6)$$

where: $\phi^{(j)}$ = weight associated with design set j

n_d = current number of design sets

Similar E^2 norms are formed for each state variable. The coefficients in equation (20.7–6) are determined by minimizing E^2 with respect to the coefficients. The weights used above are computed in one of the following ways (refer to the KWGHT input on the **OPEQN** command):

1. Based on objective function values, where design sets with low objective function values have high weight.
2. Based on design variable values, where the design sets closer to the best design receive high weight.
3. Based on feasibility, where feasible sets have high weight and infeasible sets low weights.
4. Based on a combination of the three weights described above.
5. All weight are unity: $\phi^{(j)} = 1$, for all j.

A certain number of design sets must exist in order to form the approximations; otherwise random designs sets will be generated each until the required number is obtained. This can be expressed as

$$\begin{aligned} n_d &< n + 2 \rightarrow \text{generate random design sets} \\ n_d &\geq n + 2 \rightarrow \text{form the approximations} \end{aligned} \qquad (20.7\text{–}7)$$

where: n = number of design variables

n_d = number of design sets

As more data (design sets) is generated, the terms included in equation (20.7–5) increase.

20.7.2 Minimizing the Subproblem Approximation

With function approximations available, the constrained minimization problem is recast as follows.

Minimize

$$\hat{f} = \hat{f}(\mathbf{x}) \tag{20.7–8}$$

subject to

$$\underline{x}_i \leq x_i \leq \overline{x}_i \qquad (i = 1,2,3,...,n) \tag{20.7–9}$$

$$\hat{g}_i(\mathbf{x}) \leq \overline{g}_i + \alpha_i \qquad (i = 1,2,3,...,m_1) \tag{20.7–10}$$

$$\underline{h}_i - \beta_i \leq \hat{h}_i(\mathbf{x}) \qquad (i = 1,2,3,...,m_2) \tag{20.7–11}$$

$$\underline{w}_i - \gamma_i \leq \hat{w}_i(\mathbf{x}) \leq \overline{w}_i + \gamma_i \qquad (i = 1,2,3,...,m_3) \tag{20.7–12}$$

The next step is the conversion of equations (20.7–8) to (20.7–12) from a constrained problem to an unconstrained one. This is accomplished by means of penalty functions, leading to the following subproblem statement.

Minimize

$$F(\mathbf{x}, p_k) = \hat{f} + f_0 p_k \left[\sum_{i=1}^{n} X(x_i) + \sum_{i=1}^{m_1} G(\hat{g}_i) + \sum_{i=1}^{m_2} H(\hat{h}_i) + \sum_{i=1}^{m_3} W(\hat{w}_i) \right] \tag{20.7–13}$$

in which *X* is the penalty function used to enforce design variable constraints; and G, H, and W are penalty functions for state variable constraints. The reference objective function value, f_0, is introduced in order to achieve consistent units. Notice that the unconstrained objective function (also termed a response surface), F(**x**,p_k), is seen to vary with the design variables and the quantity p_k, which is a response surface parameter. A sequential unconstrained minimization technique (SUMT) is used to solve equation (20.7–13) each design iteration. The subscript k above reflects the use of subiterations performed during the subproblem solution, whereby the response surface parameter is increased in value ($p_1 < p_2 < p_3$ etc.) in order to achieve accurate, converged results.

All penalty functions used are of the extended–interior type. For example, near the upper limit, the design variable penalty function is formed as

$$X(x_i) = \begin{Bmatrix} c_1 + c_2/(\overline{x} - x_i) & \text{if } x_i < \overline{x} - \epsilon\,(\overline{x} - \underline{x}) \\ c_3 + c_4\,(x_i - \overline{x}) & \text{if } x_i \geq \overline{x} - \epsilon\,(\overline{x} - \underline{x}) \end{Bmatrix} \qquad (i = 1,2,3,...,n) \tag{20.7–14}$$

where:

c_1, c_2, c_3, and c_4 = constants that are internally calculated ε = very small positive number

State variable penalties take a similar form. For example, again near the upper limit,

$$W(w_i) = \begin{Bmatrix} d_1 + d_2/(\overline{w}_i - \hat{w}_i) & \text{if } \hat{w}_i < \overline{w}_i - \epsilon\,(\overline{w}_i - \underline{w}_i) \\ d_3 + d_4\,(\hat{w}_i - \overline{w}_i) & \text{if } \hat{w}_i \geq \overline{w}_i - \epsilon\,(\overline{w}_i - \underline{w}_i) \end{Bmatrix} \quad (i = 1,2,3,\ldots,m_1) \tag{20.7–15}$$

where:

d_1, d_2, d_3, and d_4 = constants that are internally calculated

and similarly for G and H. See Figure 20.7–1 for a visualization of the effect.

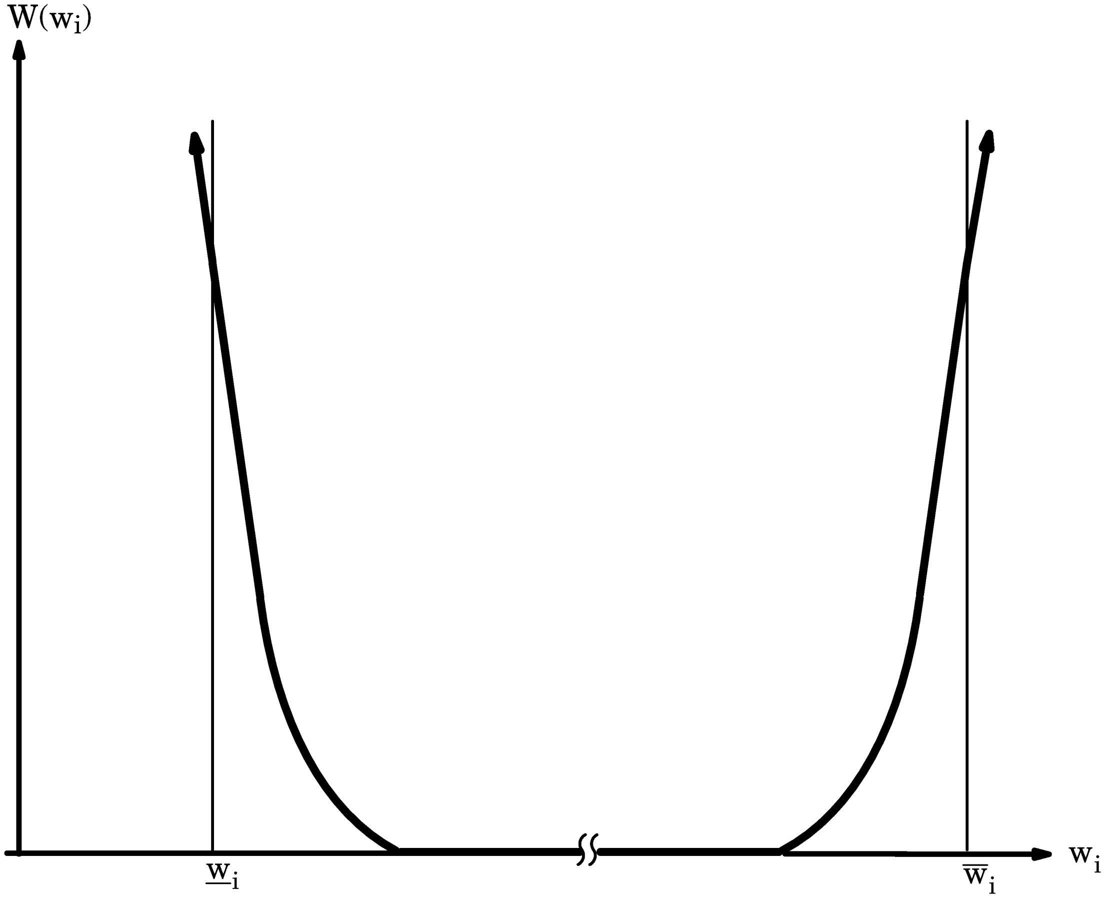


Figure 20.7–1 Extended Interior Penalty Function in State Variable Space

The SUMT algorithm is employed to reach the minimum unconstrained objective function, $\tilde{F}^{(j)}$, at design iteration j; that is,

$$\mathbf{x}^{(j)} \rightarrow \tilde{\mathbf{x}}^{(j)} \quad \text{as} \quad F^{(j)} \rightarrow \tilde{F}^{(j)} \tag{20.7–16}$$

where: $\tilde{\mathbf{x}}^{(j)}$ = is the design variable vector corresponding to $\tilde{F}^{(j)}$

The final step performed each design iteration is the determination of the design variable vector to be used in the next iteration (j+1). Vector $\mathbf{x}^{(j+1)}$ is determined according to the following equation.

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(b)} + C(\tilde{\mathbf{x}}^{(j)} - \mathbf{x}^{(b)}) \tag{20.7–17}$$

where: $\mathbf{x}^{(b)}$ = best design set (see Section 20.2) constants

C = internally chosen to vary between 0.0 and 1.0, based on the number of infeasible solutions

20.7.3 Convergence

Subproblem approximation iterations continue until either convergence is achieved or termination occurs. These two events are checked only when the current number of design sets, n_d, equals or exceeds the number required for the approximations (see equation (20.7–7)).

Convergence is assumed when either the present design set, $\mathbf{x}^{(j)}$, or the previous design set, $\mathbf{x}^{(j-1)}$, or the best design set , $\mathbf{x}^{(b)}$, is feasible; and one of the following conditions is satisfied.

$$| f^{(j)} - f^{(j-1)} | \leq \tau \tag{20.7–18}$$

$$| f^{(j)} - f^{(b)} | \leq \tau \tag{20.7–19}$$

$$| x_i^{(j)} - x_i^{(j-1)} | \leq \rho_i \qquad (i = 1,2,3,\ldots,n) \tag{20.7–20}$$

$$| x_i^{(j)} - x_i^{(b)} | \leq \rho_i \qquad (i = 1,2,3,\ldots,n) \tag{20.7–21}$$

where: τ and ϱ_i = objective function and design variable tolerances (input as TOLER on **OPVAR** command)

Equations (20.7–18) and (20.7–19) correspond to differences in objective function values; equations (20.7–20) and (20.7–21) to design variable differences.

If satisfaction of equations (20.7–18) to (20.7–21) is not realized, then termination can occur if either of the below two conditions is reached.

$$n_s = N_s \tag{20.7–22}$$

$$n_{si} = N_{si} \tag{20.7–23}$$

where: n_s = number of subproblem iterations

n_{si} = number of sequential infeasible design sets

N_s = maximum number of iterations (input as NITR on the **OPSUBP** command)

N_{si} = maximum number of sequential infeasible design sets (input as NINFS on the **OPSUBP** command)

20.8 First Order Optimization Method

This method of optimization calculates and makes use of derivative information. The constrained problem statement expressed in Section 20.1 is transformed into an unconstrained one via penalty functions. Derivatives are formed for the objective function and the state variable penalty functions, leading to a search direction in design space. Various steepest descent and conjugate direction searches are performed during each iteration until convergence is reached. Each iteration is composed of subiterations that include search direction and gradient (i.e., derivatives) computations. In other words, one first order design optimization iteration will perform several analysis loops. Compared to the subproblem approximation method, this method is usually seen to be more computationally demanding and more accurate. This method is selected with the **OPTYPE**,FIRST command.

20.8.1 The Unconstrained Objective Function

An unconstrained version of the problem outlined in Section 20.1 is formulated as follows.

$$Q(\mathbf{x},q) = \frac{f}{f_0} + \sum_{i=1}^{n} P_x(x_i) + q\left[\sum_{i=1}^{m_1} P_g(g_i) + \sum_{i=1}^{m_2} P_h(h_i) + \sum_{i=1}^{m_3} P_w(w_i)\right] \quad (20.8\text{–}1)$$

where: Q = dimensionless, unconstrained objective function

P_x, P_g, P_h, and P_w = penalties applied to the constrained design and state variables

f_0 = reference objective function value that is selected from the current group of design sets

Constraint satisfaction is controlled by a response surface parameter, q.

Exterior penalty functions (P_x) are applied to the design variables. State variable constraints are represented by extended–interior penalty functions (P_g,P_h,P_w). For example, for state variable constrained by an upper limit (equation (20.1–8)) the penalty function is written as:

$$P_g(g_i) = \left(\frac{g_i}{g_i + \alpha_i}\right)^{2\lambda} \quad (20.8\text{–}2)$$

where: λ = large integer so that the function will be very large when the constraint is violated and very small when it is not.

The functions used for the remaining penalties are of a similar form.

As search directions are devised (see below), a certain computational advantage can be gained if the function Q is rewritten as the sum of two functions. Defining

$$Q_f\,(\mathbf{x}) \;=\; \frac{f}{f_0} \tag{20.8–3}$$

and

$$Q_p(\mathbf{x},q) \;=\; \sum_{i=1}^{n} P_x(x_i) \;+\; q\left[\sum_{i=1}^{m_1} P_g(g_i) \;+\; \sum_{i=1}^{m_2} P_h(h_i) \;+\; \sum_{i=1}^{m_3} P_w(w_i)\right] \tag{20.8–4}$$

then equation (20.8–1) takes the form

$$Q(\mathbf{x},q) \;=\; Q_f(\mathbf{x}) \;+\; Q_p(\mathbf{x},q) \tag{20.8–5}$$

The functions Q_f and Q_p relate to the objective function and the penalty constraints, respectively.

20.8.2 The Search Direction

For each optimization iteration (j) a search direction vector, $\mathbf{d}^{(j)}$, is devised. The next iteration (j+1) is obtained from the following equation.

$$\mathbf{x}^{(j+1)} \;=\; \mathbf{x}^{(j)} \;+\; s_j\mathbf{d}^{(j)} \tag{20.8–6}$$

Measured from $\mathbf{x}^{(j)}$, the line search parameter, s_j, corresponds to the minimum value of Q in the direction $\mathbf{d}^{(j)}$. The solution for s_j uses a combination of a golden–section algorithm and a local quadratic fitting technique. The range of s_j is limited to

$$0 \;\le\; s_j \;\le\; \frac{S_{max}}{100}\,s_j^* \tag{20.8–7}$$

where: s_j^* = largest possible step size for the line search of the current iteration (internally computed)

S_{max} = maximum (percent) line search step size (input as SIZE on **OPFRST** command)

The key to the solution of the global minimization of equation (20.8–5) relies on the sequential generation of the search directions and on internal adjustments of the response surface parameter (q). For the initial iteration (j=0), the search direction is assumed to be the negative of the gradient of the unconstrained objective function.

$$\mathbf{d}^{(0)} = -\nabla Q(\mathbf{x}^{(0)}, q) = \mathbf{d}_f^{(0)} + \mathbf{d}_p^{(0)} \qquad (20.8\text{–}8)$$

in which q = 1, and

$$\mathbf{d}_f^{(0)} = -\nabla Q_f(\mathbf{x}^{(0)}) \quad \text{and} \quad \mathbf{d}_p^{(0)} = -\nabla Q_p(\mathbf{x}^{(0)}) \qquad (20.8\text{–}9)$$

Clearly for the initial iteration the search method is that of steepest descent. For subsequent iterations (j >0), conjugate directions are formed according to the Polak–Ribiere (More and Wright(186)) recursion formula.

$$\mathbf{d}^{(j)} = -\nabla Q\left(\mathbf{x}^{(j)}, q_k\right) + r_{j-1}\mathbf{d}^{(j-1)} \qquad (20.8\text{–}10)$$

$$r_{j-1} = \frac{\left[\nabla Q\left(\mathbf{x}^{(j)}, q\right) - \nabla Q\left(\mathbf{x}^{(j-1)}, q\right)\right]^T \nabla Q\left(\mathbf{x}^{(j)}, q\right)}{|\nabla Q\left(\mathbf{x}^{(j-1)}, q\right)|^2} \qquad (20.8\text{–}11)$$

Notice that when all design variable constraints are satisfied $P_x(x_i) = 0$. This means that q can be factored out of Q_p, and can be written as

$$Q_p\left(\mathbf{x}^{(j)}, q\right) = qQ_p\left(\mathbf{x}^{(j)}\right) \quad \text{if} \quad \underline{x}_i \le x_i \le \overline{x}_i \quad (i = 1, 2, 3, \ldots, n) \qquad (20.8\text{–}12)$$

If suitable corrections are made, q can be changed from iteration to iteration without destroying the conjugate nature of equation (20.8–10). Adjusting q provides internal control of state variable constraints, to push constraints to their limit values as necessary, as convergence is achieved. The justification for this becomes more evident once equation (20.8–10) is separated into two direction vectors:

$$\mathbf{d}^{(j)} = \mathbf{d}_f^{(j)} + \mathbf{d}_p^{(j)} \qquad (20.8\text{–}13)$$

where each direction has a separate recursion relationship,

$$\mathbf{d}_f^{(j)} = -\nabla Q_f(\mathbf{x}^{(j)}) + r_{j-1}\mathbf{d}_f^{(j-1)} \qquad (20.8\text{–}14)$$

$$\mathbf{d}_p^{(j)} = -q\nabla Q_p(\mathbf{x}^{(j)}) + r_{j-1}\mathbf{d}_p^{(j-1)} \qquad (20.8\text{–}15)$$

The algorithm is occasionally restarted by setting $r_{j-1} = 0$, forcing a steepest decent iteration. Restarting is employed whenever ill–conditioning is detected, convergence is nearly achieved, or constraint satisfaction of critical state variables is too conservative.

So far it has been assumed that the gradient vector is available. The gradient vector is computed using an approximation as follows:

$$\frac{\partial Q\left(\mathbf{x}^{(j)}\right)}{\partial x_i} \approx \frac{Q\left(\mathbf{x}^{(j)} + \Delta\, x_i\, \mathbf{e}\right) - Q\left(\mathbf{x}^{(j)}\right)}{\Delta\, x_i} \qquad (20.8\text{–}16)$$

where:

e = vector with 1 in its *ith* component and 0 for all other components

Δx_i = $\frac{\Delta D}{100}\left(\overline{x}_i - \underline{x}_i\right)$

ΔD = forward difference (in percent) step size (input as DELTA on **OPFRST** command)

20.8.3 Convergence

First order iterations continue until either convergence is achieved or termination occurs. These two events are checked at the end of each optimization iteration.

Convergence is assumed when comparing the current iteration design set (j) to the previous (j–1) set and the best (b) set.

$$| f^{(j)} - f^{(j-1)} | \leq \tau \qquad (20.8\text{–}17)$$

and

$$| f^{(j)} - f^{(b)} | \leq \tau \qquad (20.8\text{–}18)$$

where:

τ = objective function tolerance (input as TOLER on **OPVAR** command)

It is also a requirement that the final iteration used a steepest descent search. Otherwise, additional iterations are performed. In other words, a steepest descent iteration is forced and convergence rechecked.

Termination will occur when

$$n_i = N_1 \qquad (20.8\text{–}19)$$

where:

n_i = number of iterations

N_1 = allowed number of iterations (input as NITR on **OPFRST** command)

20.9 Topological Optimization

Topological optimization is a special form of shape optimization. It is sometimes referred to as layout optimization in the literature. The goal of topological optimization is to find the best use of material for a body such that an objective criteria (i.e. global stiffness, natural frequency, etc.) takes out a maximum or minimum value subject to given constraints (i.e. volume reduction).

Unlike traditional optimization (see Sections 20.1 to 20.8), topological optimization does not require the explicit definition of optimization parameters (i.e. independent variables to be optimized). In topological optimization, the material distribution function over a body serves as optimization parameter. The user needs to define the structural problem (material properties, FE model, loads, etc.) and the objective function (i.e. the function to be minimized or maximized) and the state variables (i.e. constrained dependent variables) must be selected among a set of predefined criteria.

20.9.1 The General Optimization Problem Statement

The theory of topological optimization seeks to minimize or maximize the objective function (OBJ) subject to the constraints (CON_j) defined. The design variables (η_i) are internal, pseudodensities that are assigned to each finite element (i) in the topological problem. The pseudodensity for each element varies from 0 to 1; where $\eta_i \approx 0$ represents material to be removed; and $\eta_i \approx 1$ represents material that should be kept. Stated in simple mathematical terms, the optimization problem is as follows:

$$\text{OBJ} = \text{a minimum/maximum w.r.t. } \eta_i \tag{20.9–1}$$

subject to

$$0 < \eta_i \leq 1 \qquad (i = 1, 2, 3 \ldots . N) \tag{20.9–2}$$

$$\underline{\text{CON}_j} < \text{CON}_j \leq \overline{\text{CON}_j}\ (j = 1, 2, 3, \ldots, M) \tag{20.9–3}$$

where:

N = number of elements

M = number of constraints

CON_j = computed jth constraint value

$\underline{\text{CON}_j}$ = lower bound for jth constraint

$\overline{\text{CON}_j}$ = upper bound for jth constraint

20.9.2 “Maximum Static Stiffness” Design Subject to Volume Constraint

In the case of “maximum static stiffness” design subject to a volume constraint, which sometimes is referred to as the standard formulation of the layout problem, one seeks to minimiuze the energy of the structural static compliance (U_C) for a given load case subject to a given volume reduction. Minimizing the compliance is equivalent to maximizing the global structural static stiffness. In this case, the optimization problem is formulated as a special case of (20.9–1), (20.9–2) and (20.9–3), namely,

$$U_C = \text{a minimum w.r.t. } \eta_i \tag{20.9–4}$$

subject to

$$0 < \eta_i \leq 1 \qquad (i = 1, 2, 3, \ldots, N) \tag{20.9–5}$$

$$V \leq V_0 - V^* \tag{20.9–6}$$

where:

V = computed volume
V_0 = original volume
V^* = amount of material to be removed

Topological optimization may be applied to either a single load case or multiple load cases. For the latter, given K different load cases, the following weighted function (F) is defined:

$$F\left(U_C^1, U_C^2, \ldots, U_C^k\right) = \sum_{i=1}^{k} W_i U_C^i, \; W_i \geq 0 \tag{20.9–7}$$

where:

W_i = weight for load case with energy U_C

The functional minimization equation (20.9–4) is replaced with:

$$F = \text{a minimum w.r.t. } \eta_i \tag{20.9–8}$$

and equations (20.9–4) and (20.9–8) are clearly identical for the special case of k = 1.

20.9.3 Minimum Volume Design Subject to Stiffness Constraints

In constrast to the formulation to section 20.9.2, it sometimes might be desirable to design for minimum volume subject to a single or multiple compliance (energy) constraint(s). In this case, given k different load cases, the optimization problem is formulated as:

$$V = \text{a minimum w.r.t. } \eta_i \tag{20.9–9}$$

subject to

$$0 < \eta_i \leq 1 \qquad (i = 1, 2, 3, \ldots, N) \tag{20.9–10}$$

$$\underline{U^j_C} \leq U^j_C \leq \overline{U^j_C} \qquad (j = 1, 2, 3, \ldots, M) \tag{20.9–11}$$

where:
V = computed volume
M = number of constraints
U^j_C = computed compliance of load case j
$\underline{U^j_C}$ = lower bound for compliance of load case j
$\overline{U^j_C}$ = upper bound for compliance of load case j

Additionally, it is allowed to constrain the weighted compliance function (F) as of equation (20.9–8). In this case the k constraints (equation (20.9–11)) are substituted by only one constraint of the form:

$$\underline{F} \leq F \leq \overline{F} \tag{20.9–12}$$

where:
F = computed weighted compliance function
$\underline{F}$ = lower bound for weighted compliance function
$\overline{F}$ = upper bound for weighted compliance function

20.9.4 "Maximum Dynamic Stiffness" Design Subject to Volume Constraint

In case of the "Maximum Dynamic Stiffness" design subject to a volume constraint one seeks to maximize the ith natural frequency ($\overline{\omega}_i > 0$) determined from a mode–frequency analysis subject to a given volume reduction. In this case, the optimization problem is formulated as:

$$\overline{\omega}_i = \text{a maximum w.r.t. } \eta_i \tag{20.9–13}$$

subject to

$$0 < \eta_i \leq 1 \qquad (i = 1, 2, 3, \ldots, N) \tag{20.9–14}$$

$$V \leq V_0 - V^* \tag{20.9–15}$$

where: ϖ_i = ith natural frequency computed

V = computed volume

V_0 = original volume

V^* = amount of material to be removed

Maximizing a specific eigenfrequency is a typical problem for an eigenfrequency topological optimization. However, during the course of the optimization it may happen that eigenmodes switch the modal order. For example, at the beginning we may wish to maximize the first eigenfrequency. As the first eigenfrequency is increased during the optimization it may happen, that second eigenmode eventually has a lower eigenfrequency and therefore effectively becomes the first eigenmode. The same may happen if any other eigenfrequency is maximized during the optimization. In such a case, the sensitivities of the objective function become discontinuous, which may cause oscillation and divergence in the iterative optimization process. In order to overcome this problem, several mean–eigenfrequency functions (Λ) are introduced to smooth out the frequency objective:

Weighted Formulation:

Given m natural frequencies (ω_i, . . ., ϖ_m), the following weighted mean function (Ω_W) is defined:

$$\Omega_W = \sum_{I=1}^{M} W_i \ \varpi_i \tag{20.9–16}$$

where: ϖ_i = ith natural frequency

W_i = weight for ith natural frequency

The functional maximization equation (20.9–13) is replaced with

$$\Omega_W = \text{a maximum w.r.t. } \eta_i \tag{20.9–17}$$

Reciprocal Formulation:

Given m natural frequencies (ϖ_i, . . ., ϖ_m), a shift parameter ϖ_0, the following reciprocal mean function (Ω_R) is defined:

$$\Omega_R = \varpi_o + \left[\sum_{i=1}^{m} \frac{W_i}{\varpi_i - \varpi_o} \right]^{-1} \tag{20.9–18}$$

where: ϖ_i = ith natural frequency

W_i = weight for ith natural frequency

The functional maximization equation (20.9–13) is replaced with

$$\Omega_R = \text{a maximum w.r.t. } \eta_i \qquad (20.9\text{–}19)$$

As shown in equation (20.9–18), the natural frequency which is the closest to the shift parameter $\overline{\omega}_o$ has the largest contribution to the objective function Ω_R, assuming all of the weights W_i are the same. In the special case, $\overline{\omega}_i = 0$, the lowest natural frequency in $(\omega_i, \ldots, \overline{\omega}_m)$ has the largest contribution to the objective function. Thus, the natural frequency that is the closest to $\overline{\omega}_o$ will be the major object of the optimization problem. This implies that this natural frequency will experience the largest change. When two modes whose natural frequencies occur in equation (20.9–18) exchange their order during optimization, the change in the objective Ω_R will be smooth because the contributions of these modes have already been accounted for in the objective function. To intensify this effect, the weighting coefficients W_i can be adjusted accordingly.

Euclidean Norm Formulation:

Given m natural frequencies $(\overline{\omega}_i, \ldots, \overline{\omega}_m)$, m frequency target values $(\overline{\overline{\omega}}_i, \ldots, \overline{\overline{\omega}}_m)$, the following Euclidean Norm function (Ω_E) is defined:

$$\Omega_E = \left[\sum_{i=1}^{m} \left(\overline{\omega}_i - \overline{\overline{\omega}}_i \right)^2 \right]^{\frac{1}{2}} \qquad (20.9\text{–}20)$$

The functional maximization equation (20.9–13) is replaced with

$$\Omega_E = \text{a maximum w.r.t. } \eta_i \qquad (20.9\text{–}21)$$

This formulation can be used to shift up single or multiple natural frequencies to given target values by minimizing the Euclidean distance between actual frequencies and the desired target values. All the specified frequencies $(\overline{\omega}_i, \ldots, \overline{\omega}_m)$ will approach to their desired target values $(\overline{\overline{\omega}}_i, \ldots, \overline{\overline{\omega}}_m)$, respectively, and the frequency which is the farthest from its target value will the fasted approach to its desired value.

20.9.5 Element Calculations

While compliance, natural frequency, and total volume are global conditions, certain and critical calculations are performed at the level of individual finite elements. The total volume, for example, is calculated from the sum of the element volumes; that is,

$$V = \sum_{i} \eta_i V_i \qquad (20.9\text{–}22)$$

where: V_i = volume for element i

The pseudodensities effect the volume and the elasticity tensor for each element. That is,

$$[E_i] = [E(\eta_i)] \qquad (20.9\text{–}23)$$

where the elasticity tensor is used to equate the stress and strain vector, designed in the usual manner for linear elasticity:

$$\{\sigma_i\} = [E_i]\{\varepsilon_i\} \qquad (20.9\text{–}24)$$

where: $\{\sigma_i\}$ = stress vector of element i

$\{\varepsilon_i\}$ = strain vector of element i

The exact dependence of the elasticity tensor, the compliance, and the natural frequency with respect to density is expressed in detail elsewhere (see Vogel(233), Mlejnek and Schirrmacher(234), Bendsoe and Kikuchi(235), and Diaz and Kikuchi(273).

The equations above directly apply to elastic solid elements (PLANE2 in plane stress, PLANE82, SOLID92, and SOLID95). Shells are treated in a slightly different manner. For SHELL93, the elasticity tensor is that used for 2–D plane stress.

References

Theory Reference Index

1. Ahmad, S., Irons, B. M. and Zienkiewicz, O. C., “Analysis of Thick and Thin Shell Structures by Curved Finite Elements”, *International Journal for Numerical Methods in Engineering*, Vol. 2, No. 3, pp. 419–451 (1970).

2. Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice–Hall, Englewood Cliffs (1982).

3. Biot, M. A., *Mechanics of Incremental Deformation*, John Wiley and Sons, New York (1965).

4. Chen, L. H., “Piping Flexibility Analysis by Stiffness Matrix”, ASME, *Journal of Applied Mechanics* (December, 1959).

5. Cook, R. D., *Concepts and Applications of Finite Element Analysis*, Second Edition, John Wiley and Sons, New York (1981).

6. Cook, R. D., “Two Hybrid Elements for Analysis of Thick, Thin and Sandwich Plates”, *International Journal for Numerical Methods in Engineering*, Vol. 5, No. 2, pp. 277–288 (1972).

7. Cuniff, D. F., and O’Hara, G. J., “Normal Mode Theory for Three–Directional Motion”, *NRL Report 6170*, U. S. Naval Research Laboratory, Washington D. C. (1965).

8. Denn, M. M., *Optimization by Variational Methods*, McGraw–Hill, New York (1969).

9. Henshell, K. D. and Ong, J. H., “Automatic Masters for Eigenvalue Economization”, *Earthquake Engineering and Structural Dynamics*, Vol. 3, pp. 375–383 (1975).

10. Imgrund, M. C., *ANSYS® Verification Manual*, Swanson Analysis Systems, Inc. (1992).

11. Flügge, W., *Stresses in Shells*, Springer Verlag, Berlin (1967).

12. Fritz, R. J., “The Effect of Liquids on the Dynamic Motions of Immersed Solids”, *ASME Journal of Engineering for Industry* (February, 1972)

13. Galambos, T. V., *Structural Members and Frames*, Prentice–Hall, Englewood Cliffs (1968).

14. Guyan, R. J., “Reduction of Stiffness and Mass Matrices”, *AIAA Journal*, Vol. 3, No. 2 (February, 1965).

15. Hall, A. S., and Woodhead, R. W., *Frame Analysis*, John Wiley and Sons, New York (1961).

16. Rajakumar, C. and Rogers, C. R., “The Lanczos Algorithm Applied to Unsymmetric Generalized Eigenvalue Problem”, *International Journal for Numerical Methods in Engineering*, Vol. 32, pp. 1009–1026 (1992).

17. Irons, B. M., “A Frontal Solution Program for Finite Element Analysis”, *International Journal for Numerical Methods in Engineering*, Vol. 2, No. 1, January, 1970, pp. 5–23 (Discussion May, 1970, p. 149).

18. Wilkinson, J. H., *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, pp. 515–569 (1988).

19. Kohnke, P. C., and Swanson, J. A., “Thermo–Electric Finite Elements”, *Proceedings, International Conference on Numerical Methods in Electrical and Magnetic Field Problems*, June 1–4, 1976, Santa Margherita Liqure (Italy).

20. Kohnke, P. C., “Large Deflection Analysis of Frame Structures by Fictitious Forces”, *International Journal of Numerical Methods in Engineering*, Vol. 12, No. 8, pp. 1278–1294 (1978).

21. Kollbrunner, C. F., and Basler, K., *Torsion in Structures*, Springer–Verlag, Berlin, (1969).

22. Konopinski, E. J., *Classical Descriptions of Motion*, Freeman and Company, San Francisco (1969).

23. Kreyszig, E., *Advanced Engineering Mathematics,* John Wiley and Sons, Inc., New York (1962).

24. Lekhnitskii, S. G., *Theory of Elasticity of an Anisotropic Elastic Body*, Holden–Day, San Francisco (1963).

25. Melosh, R. J., and Bamford, R. M., "Efficient Solution of Load–Deflection Equations", *ASCE Journal of the Structural Division*, Vol. 95, No. ST4, Proc. Paper 6510, Apr., 1969, pp. 661–676 (Discussions Dec., 1969, Jan., Feb., May, 1970, Closure, Feb., 1971).

26. Kanok–Nukulchai, "A Simple and Efficient Finite Element for General Shell Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 14, pp. 179–200 (1979).

27. Oden, J. T., *Mechanics of Elastic Structures*, McGraw–Hill, New York (1968).

28. Przemieniecki, J. S., *Theory of Matrix Structural Analysis*, McGraw–Hill, New York (1968).

29. Schnobrich, W. C., and Suidan, M., "Finite Element Analysis of Reinforced Concrete", *ASCE Journal of the Structural Division*, ST10, pp. 2109–2122 (October, 1973).

30. Seide, P., "Large Deflection of Rectangular Membranes Under Uniform Pressure", *International Journal of Non–Linear Mechanics*, Vol. 12, pp. 397–406.

31. Skjelbreia, L., and Hendrickson, J. A., "Fifth Order Gravity Wave Theory", *Proceedings, Seventh Conference on Coastal Engineering*, Ch. 10, pp. 184–196 (1961).

32. Timoshenko. S., and Woinowskey–Kreiger, S., *Theory of Plates and Shells*, McGraw–Hill, New York (1959).

33. Tracey, D. M., "Finite Elements for Three Dimensional Elastic Crack Analysis", *Nuclear Engineering and Design*, 26 (1973).

34. Vanmarcke, E. H., "Structural Response to Earthquakes", *Seismic Risk and Engineering Decisions*, Elsvier Scientific Publishing Co., Amsterdam–Oxford, New York, (edited by C. Lomnitz and E. Rosemblueth), pp. 287–337 (1976).

35. Wheeler, J. D., "Method of Calculating Forces Produced by Irregular Waves", *Journal of Petroleum Technology*, Vol. 22, pp. 359–367 (1970).

36. Willam, K. J., University of Colorado, Boulder, (Private Communication) (1982).

37. Willam, K. J., and Warnke, E. D., "Constitutive Model for the Triaxial Behavior of Concrete", *Proceedings, International Association for Bridge and Structural Engineering*, Vol. 19, ISMES, Bergamo, Italy, p. 174 (1975).

38. Wilson, E. L., Taylor, R. L., Doherty, W. P., and Ghaboussi, J., "Incompatible Displacement Models", *Numerical and Computer Methods in Structural Mechanics*, edited by S. J. Fenves, et al., Academic Press, Inc., N. Y. and London, pp. 43–57 (1973).

39. Zienkiewicz, O. C., *The Finite Element Method*, McGraw–Hill Company, London, (1977).

40. *ASME Boiler and Pressure Vessel Code,* Section III, Division 1, Subsection NC, Class 2 Components (1974).

41. "Regulatory Guide", Published by the U. S. Nuclear Regulatory Commission, *Regulatory Guide* 1.92, Revision 1 (February 1976).

42. Tamma, Kumar K. and Namburu, Raju R., "Recent Advances, Trends and New Perspectives Via Enthalpy–Based Finite Element Formulations for Applications to Solidification Problems", *International Journal for Numerical Methods in Engineering*, Vol. 30, pp. 803–820 (1990).

43. *Shore Protection Manual*, Published by the U. S. Army Coastal Engineering Research Center, Vol. I, Third Edition (1977).

44. Beer, F. P., and Johnston, R. E., *Vector Mechanics for Engineers, Statics and Dynamics*, McGraw–Hill, New York (1962).

45. Hinton, E., Rock, A., and Zienkiewicz, O., "A Note on Mass Lumping and Related Processes in the Finite Element Method", *International Journal of Earthquake Engineering and Structural Dynamics*, Vol. 4, pp. 245–249 (1976).

46. Krieg, R. D., and Krieg, D. B., "Accuracies of Numerical Solution Methods for the Elastic–Perfectly Plastic Model", *Journal of Pressure Vessel Technology,*

Vol. 99, No. 4, Series J, *Transactions of the ASME,*November, pp. 510–515 (1977).

47. Thomson, William T., *Theory of Vibrations with Applications*, Prentice Hall, pp. 343–352 (1971).

48. Roark, R. J., and Young, W. C., *Formulas for Stress and Strain*, McGraw–Hill, New York (1975).

49. Taylor, R. L., Beresford, P. J., and Wilson, E. L., “A Non–Conforming Element for Stress Analysis”, *International Journal for Numerical Methods in Engineering*, Vol. 10, pp. 1211–1219 (1976).

50. Hill, R., *The Mathematical Theory of Plasticity*, Oxford University Press, New York (1983).

51. Shih, C. F., and Lee, D., “Further Developments in Anisotropic Plasticity”, *Journal of Engineering Materials and Technology*, Vol. 100, pp. 294–302 (July 1978).

52. Valliappan, S., “Nonlinear Analysis for Anistropic Materials”, *International Journal for Numerical Methods in Engineering*, Vol. 10, pp. 597–606 (1976).

53. Besseling, J. F., “A Theory of Elastic, Plastic, and Creep Deformations of an Initially Isotropic Material Showing Aisotropic Strain–Hardening Creep Recovery and Secondary Creep”, *Journal of Applied Mechanics*, pp. 529–536 (December 1958).

54. Owen, R. J., Prakash, A., and Zienkiewicz, O. C., “Finite Element Analysis of Non–Linear Composite Materials by Use of Overlay Sytems”, *Computers and Structures*, Pergamon Press, Vol. 4, pp. 1251–1267.

55. Holman, J. P., *Heat Transfer*, Fourth Edition, McGraw–Hill, New York (1976).

56. Batoz, J. L., Bathe, K. J., and Ho, L. W., “A Study of Three–Node Triangular Plate Bending Elements”, *International Journal of Numerical Methods in Engineering*, Vol. 15, pp. 1771–1812 (1980).

57. Razzaque, A., “On the Four Noded Discrete Kirchhoff Shell Elements”, Robinson, J. (ed.), *Accuracy Reliability Training in FEM Technology*, pp. 473–483 (1984).

58. Gresho, P. M., and Lee, R. L., “Don’t Suppress the Wiggles – They’re Telling You Something”, *Finite Element Methods for Convection Dominated Flows*, ASME Publication AMD, Vol. 34, pp. 37–61 (1979).

59. Dean, R. G., *Evaluation and Development of Water Wave Theories for Engineering Application*, prepared for U. S. Army Corp of Engineers, Coastal Engineering Research Center, (November 1974).

60. *ASME Boiler and Pressure Vessel Code*, Section III, Division 1–1974, Subsection NB, Class 1 Components.

61. *American National Standard Code for Pressure Piping*, Power Piping, ANSI B31.1–1977, Published by the American Society of Mechanical Engineers.

62. Orris, R. M., and Petyt, M., “Finite Element Study of Harmonic Wave Propagation in Periodic Structures”, *Journal of Sound and Vibration*, pp. 223–236 (1974).

63. Gordon, J. L., “OUTCUR: An Automated Evaluation of Two–Dimensional Finite Element Stresses” according to ASME, Paper No. 76–WA/PVP–16, *ASME Winter Annual Meeting* (December 1976).

64. Powell, M. J. D., “An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives”, *Computer Journal*, Vol. 7, pp. 155–162 (1964).

65. Wilson, E. L., Der Kiereghian, A., and Bayo, E., “A Replacement for the SRSS Method in Seismic Analysis”, *Earthquake and Structural Dynamics*, University of California, Berkeley, Vol. 9, No. 2, p. 187 (March 1981).

66. Rankin, C. C., and Brogan, F. A., “An Element Independent Corotational Procedure for the Treatment of Large Rotations”, *Journal of Pressure Vessel Technology*, Vol. 108, pp. 165–174 (May 1986).

67. Argyris, J., “An Excursion into Large Rotations”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, pp. 85–155 (1982).

68. Tse, S., Morse, I. E., and Hinkle, R. T., *Mechanical Vibrations*, Allyn and Bacon, Boston (1963).

69. Chari, M. V. K., “Finite Element Solution of the Eddy Current Problem in Magnetic Structures, *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS–93 pp. 62–72 (1974).

70. Brauer, J. R., “Finite Element Analysis of Electromagnetic Induction in Transformers”, paper A77–122–5, *IEEE Winter Power Meeting*, New York City (1977).

71. Tandon, S. C., and Chari, M. V. K., “Transient Solution of the Diffusion Equation by the Finite Element Method”, J*ournal of Applied Physics* (March 1981).

72. Silvester, P. P., Cabayan, H. S., and Browne, B. T., “Efficient Techniques for Finite Element Analysis of Electric Machines”, *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS–92, pp. 1274–1281 (1973).

73. Chari, M. V. K., and D’Angelo, J., “Finite Element Analysis of Magneto–Mechanical Devices”, *Fifth International Workshop in Rare Earth–Cobalt Permanent Magnets and Their Application*, Roanoke, VA, 7–10, Paper No. V1–1 (June 1981).

74. Anderson, O. W., “Transform Leakage Flux Program Based on the Finite Element Method”, *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS–92, No. 2 (1973).

75. Zienkiewicz, O. C., Lyness, J., and Owen, D. R. J., “Three–Dimensional Magnetic Field Determination Using a Scalar Potential – A Finite Element Solution”, *IEEE Transactions on Magnetics*, Vol. MAG–13, No. 5, pp. 1649–1656 (1977).

76. Coulomb, J. L., and Meunier, G., “Finite Element Implementation of Virtual Work Principle for Magnetic for Electric Force and Torque Calculation, *IEEE Transactions on Magnetics*, Vol. Mag–2D, No. 5, pp. 1894–1896 (1984).

77. Moon, F. C., *Magneto–Solid Mechanics*, New York, John Wiley and Sons (1984).

78. Baker, A. J., *Finite Element Computational Fluid Mechanics*, McGraw–Hill Book Company, New York, pp. 266–284 (1983).

79. Yuan, S. NW., *Foundations of Fluid Mechanics*, Prentice–Hall International, Inc., London, pp. 71–102 (1976).

80. Clough, Ray W., and Penzien, Joseph, *Dynamics of Structures*, McGraw–Hill, New York, p. 559 (1975).

81. Allik, H., and Hughes, J. R., “Finite Element for Piezoelectric Vibration”, *International Journal Numerical Methods of Engineering*, No. 2, pp. 151–157 (1970).

82. Eer Nisse, N. P. “Variational Method for Electroelastic Vibration Analysis”, *IEEE Transactions on Sonics and Ultrasonics*, Vol. SU–14, No. 4 (1967).

83. Sato, J., Kawabuchi, M., and Fukumoto, A., “Dependence of the Electromechanical Coupling Coefficient on the Width–to–Thickness Ratio of Plant–Shaped Piezoelectric Transducers Used for Electronically Scanned Ultrasound Diagnostic Systems”, *Journal of Acoustics Society of America* No. 66 (6), pp. 1609–1611 (1979).

84. Kinsler, E. L. et. al., *Fundamentals of Acoustics*, John Wiley and Sons, New York pp. 98–123 (1982).

85. Craggs, A., “A Finite Element Model for Acoustically Lined Small Rooms”, *Journal of Sound and Vibration*, Vol. 108, No. 2, pp. 327–337.

86. Zienkiewicz, O. C., and Newton, R. E., “Coupled Vibrations of a Structure Submerged in a Compressible Fluid”, *Proceedings of the Symposium on Finite Element Techniques*, University of Stuttgart, Germany (June 1969).

87. Malvern, Lawrence E., *Introduction to the Mechanics of a Continuous Medium*, Prentice–Hall, Inc., Englewood Cliffs, NJ (1969).

88. Siegal, R. and Howell, J. R., *Thermal Radiation Heat Transfer*, Second Edition, Hemisphere Publishing Corporation (1981).

89. Rivlin, R. S., “Forty Years of Nonlinear Continuum Mechanics”, *Proceedings of the IX International Congress on Rheology*, Mexico, pp. 1–29 (1984).

90. Kao, B. G., and Razgunas, L., “On the Determination of Strain Energy Functions of Rubbers”, *Proceedings of the VI International Conference on Vehicle Structural Mechanics*, Detroit, MI, pp. 145–154 (1986).

91. Mooney, M., “A Theory of Large Elastic Deformation”, *Journal of Applied Physics*, Vol. 6, pp. 582–592 (1940).

92. Blatz, P. J., and Ko, W. L., “Application of Finite Elastic Theory to the Deformation of Rubbery Materials”, *Transactions of the Society of Rheology*, Vol. 6, pp. 223–251 (1962).

93. Tsai, Stephen W., *Composites Design*, Third Edition, Section 11.6, Think Composites, Dayton, Ohio (1987).

94. Weiss, J., “Efficient Finite Element Solution of Multipath Eddy Current Problems”, *IEEE Transactions on Magnetics*, Vol. MAG–18, No. 6, pp. 1710–1712 (1982).

95. Garg, V. K., and Weiss, J, “Finite Element Solution of Transient Eddy–Current Problems in Multiply–Excited Magnetic Systems”, *IEEE Transactions on Magnetics*, Vol. MAG–22, No. 5, pp. 1257–1259 (1986).

96. Dvorkin, E. N., “On Nonlinear Finite Element Analysis of Shell Structures”, Ph.D Thesis, Massachusetts Institute of Technology (1984).

97. Dvorkin, E. N., and Bathe, K. J., “A Continuum Mechanics Based Four–Node Shell Element for General Nonlinear Analysis”, *Engineering Computations*, Vol. 1, pp. 77–88 (1984).

98. Bathe, K. J., and Dvorkin, E. N., “A Formulation of General Shell Elements – The Use of Mixed Interpolation of Tensorial Components”, *International Journal for Numerical Methods in Engineering*, Vol. 22, pp. 697–722 (1986).

99. Hoit, M., and Wilson, E. L., “An Equation Numbering Algorithm Based on a Minimum Front Criteria”, *Computers and Structures*, Vol. 16, pp. 225–239 (1983).

100. Cuthill, E., and McKee, J., “Reducing the Band Width of Sparse Symmetric Matrices”, *Proceedings of the ACM National Conference*, New York (1969).

101. Georges, A., and McIntyre, D., “On the Application of the Minimum Degree Algorithm to Finite Element Systems”, *SIAM Journal of Numerical Analysis*, Vol. 15 (1978).

102. Zienkiewicz, O. C., and Zhu, J. Z., “A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis”, *International Journal for Numerical Methods in Engineering*, Vol. 24, pp. 337–357 (1987).

103. Babuska, I., and Rheinboldt, W. C., “Analysis of Optimal Finite Element Meshes in R”, *Mathematics of Computation*, Vol. 33, pp. 431–463 (1979).

104. Carnegie, W., “Vibrations of Rotating Cantilever Blading”, *Journal of Mechanical Engineering Science*, Vol. 1, No. 3 (1959).

105. Bergan, P. G., and Mollestad, E., “An Automatic Time–Stepping Algorithm for Dynamic Problems”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 49 (1985).

106. Paris, P. C., and Sih, G. C., “Stress Analysis of Cracks”, *Fracture Toughness and Testing and its Applications,* American Society for Testing and Materials, Philadelphia, STP 381, pp. 30–83 (1965).

107. O’Hara, G. J. and Belsheim, R. O., “Interim Design Values for Shock Design of Shipboard Equipment”, U.S. Naval Research Laboratory, Washington D.C., *NRL Memorandum Report 1396* (1963).

108. Markovsky, A., Soules, T. F. and Vukcevich, M. R., “Mathematical and Computational Aspects of a General Viscoelastic Theory”, G. E. Lighting and Research and Technical Services Operation, *Report No. 86–LRL–2021* (February 1986).

109. Scherer, G. W. and Rekhson, S. M., “Viscoelastic–Elastic Composites: I, General Theory”, *Journal of the American Ceramic Society*, Vol. 65, No. 7 (1982).

110. Narayanaswamy, O. S., “A Model of Structural Relaxation in Glass”, *Journal of the American Ceramic Society*, Vol. 54, No. 10, pp. 491–498 (1971).

111. Zienkiewicz, O. C., Watson, M. and King, I. P., “A Numerical Method of Visco–Elastic Stress Analysis”, *International Journal of Mechanical Science*, Vol. 10, pp. 807–827 (1968).

112. Taylor, R. L., Pister, K. S. and Goudreas, G. L., “Thermochemical Analysis of Viscoelastic Solids”, *International Journal for Numerical Methods in Engineering*, Vol. 2, pp. 45–59 (1970).

113. Allman, D. J., “A Compatible Triangular Element Including Vertex Rotations for Plane Elasticity Analysis”, *Computers and Structures*, Vol. 19, pp. 1–8 (1984).

114. Cook, R. D., “On the Allman Triangle and a Related Quadrilateral Element”, *Computers and Structures*, Vol. 22, pp. 1065–1067 (1986).

115. MacNeal, R. H. and Harder, R. L., “A Refined For–Noded Membrane Element with Rotational Degrees of Freedom”, *Computers and Structures*, Vol. 28, No. 1, pp. 75–84.

116. Garvey, S. J., “The Quadrilateral Shear Panel”, *Aircraft Engineering*, p. 134 (May 1951).

117. Yunus, Shah M., Pawlak, Timothy P., and Cook, R. D., “Solid Elements with Rotational Degrees of Freedom Part 1 and Part 2”, *International Journal for Numerical Methods in Engineering*, Vol. 31, pp. 573–610 (1991).

118. Mohammed, O. A., “Magnetic Vector Potential Based Formulation and Computation of Nonlinear Magnetostatic Fields and Forces in Electrical Devices by Finite Elements”, *Ph.D. Dissertation*, Virginia Polytechnic Institute and State University, Blacksburg, VA (May 1983).

119. Mayergoyz, I. D., “A New Scalar Potential Formulation for Three–Dimensional Magnetostatic Problems”, *IEEE Transactions on Magnetics*, Vol. MAG–23, No. 6, pp. 3889–3894 (1987).

120. Bíró, Oszkár and Preis, Kurt, “On the Use of the Magnetic Vector Potential in the Finite Element Analysis of Three–Dimensional Eddy Currents”, *IEEE Transactions on Magnetics*, Vol. 25, No. 4, pp. 3145–3159 (July 1989).

121. Robinson, J., *Basic and Shape Sensivity Tests for Membrane and Plate Bending Finite Elements*, Robinson and Associates (January 1985).

122. Kagawa, Y., Yamabuchi, T. and Kitagami, S., “Infinite Boundary Element and its Application to a Combined Finite–Boundary Element Technique for

Unbounded Field Problems”, *Boundary Elements VIII*, ed. C. A. Brebbia, Springer–Verlag, New York, NY (1986).

123. Oden, J. T. and Kikuchi, N., “Finite Element Methods for Constrained Problems in Elasticity”, *International Journal for Numerical Methods in Engineering*, Vol. 18, No. 5, pp. 701–725 (1982).

124. Sussman, T. and Bathe, K. J., “A Finite Element Formulation for Nonlinear Incompressible Elastic and Inelastic Analysis”, *Computers and Structures*, Vol. 26, No. 1/2, pp. 357–409 (1987).

125. Zienkiewicz, O. C., Liu, Y. C. and Huang, G. C., “Error Estimates and Convergence Rates for Various Incompressible Elements”, *International Journal for Numerical Methods in Engineering*, Vol. 28, No. 9, pp. 2191–2202 (1989).

126. Huang, H. C. and Lewis, R. W., “Adaptive Analysis for Heat Flow Problems Using Error Estimation Techniques”, Paper presented at the *6th International Conference on Numerical Methods in Thermal Problems*. Also University of Wales, University College of Swansea Internal Report CR/635/89 (April 1989).

127. Weber, G. G., Lush, A. M., Zavaliangos, A., and Anand, L., “An Objective Time–Integration Procedure for Isotropic Rate–Independent Elastic–Plastic Constitutive Equations”, *International Journal of Plasticity*, Vol. 6, pp. 701–749 (1990).

128. Eggert, G. M. and Dawson, P. R., “A Viscoplastic Formulation with Plasticity for Transient Metal Forming”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 70, pp. 165–190 (1988).

129. Narayanaswami R. and Adelman, H. M., “Inclusion of Transverse Shear Deformation in Finite Element Displacement Formulations”, *American Institute of Aeronautics and Astronautics Journal*, Vol. 12, No. 11, pp. 1613–1614 (1974).

130. Kaljević, I, Saigal, S., and Ali, A., “An Infinite Boundary Element Formulation for Three–Dimensional Potential Problems”, *International Journal for Numerical Methods in Engineering*, Vol. 35, No. 10, pp. 2079–2100 (1992).

131. Simo et al., “Finite Deformation Post–Buckling Analysis Involving Inelasticity and Contact Constraints”, *International Journal for Numerical Methods in Engineering*, Vol. 23, pp. 779–800 (1986).

132. Parisch, H. “A Consistent Tangent Stiffness Matrix for Three–Dimensional Non–Linear Contact Analysis”, *International Journal for Numerical Methods in Engineering*, Vol.28, pp. 1803–1812 (1989).

133. Bayo, “A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, pp. 183–195 (1988).

134. Jiang and Rodgers, “Combined Lagrangian Multiplier and Penalty Function Finite Element Technique for Elastic Impact Analysis”, *Computers and Structures*, Vol. 30, pp. 1219–1229 (1988).

135. Giannakopoulos, “The Return Mapping Method for the Integration of Friction Constitutive Relations”, *Computers and Structures*, Vol. 32, pp. 157–167 (1989).

136. Ridic and Owen, “A Plasticity Theory of Friction and Joint Elements”, Computational Plasticity: Models, Software, and Applications, Part II, *Proceedings of the Second International Conference*, Barcelona, Spain, Pineridge Press, Swansea, pp. 1043–1062 (Editors Owen, Hinton, Ornate) (1989).

137. Wriggers, VuVan, and Stein, “Finite Element Formulation of Large Deformation Impact–Contact Problems with Friction”, *Computers and Structures*, Vol. 37, pp. 319–331 (1990).

138. Stein, Wriggers and VuVan, “Models of Friction, Finite–Element Implementation and Application to Large Deformation Impact–Contact Problems”, Computational Plasticity: Models, Software, and Applications, Part II, *Proceedings of the Second International Conference*, Barcelona, Spain, Pineridge Press, Swansea, pp. 1015–1041, (Editors Owen, Hinton, Ornate) (1989).

139. Yunus, S. M., Kohnke, P. C. and Saigal, S. “An Efficient Through–Thickness Integration Scheme in an Unlimited Layer Doubly Curved Isoparametric

Composite Shell Element", *International Journal for Numerical Methods in Engineering*, Vol. 28, pp. 2777–2793 (1989).

140. Geddes, E. R. "An Analysis of the Low Frequency Sound Field in Non–Rectangular Enclosures Using the Finite Element Method", *Ph.D Thesis*, Pennsylvania State University (1982).

141. Gyimesi, M., Lavers, D., Pawlak, T. and Ostergaard, D., "Application of the General Potential Formulation in the ANSYS®Program", *IEEE Transactions on Magnetics*, Vol. 29, pp. 1345–1347 (1993).

142. Rajakumar, C. and Ali, A., "A Solution Method for Acoustic Boundary Element Eigenproblem With Sound Absorption Using Lanczos Algorithm", *Proceedings of 2nd International Congress on Recent Developments in Air– and Structure–Borne Sound and Vibration*, Auburn University, AL, pp. 1001–1010 (March 4–6, 1992)

143. Nishimura, H., Isobe, M., and Horikawa, K., "Higher Order Solutions of the Stokes and the Cnoidal Waves", *Journal of the Faculty of Engineering*, The University of Tokyo, Vol. XXXIV, No. 2, Footnote on page 268 (1977).

144. Mahinthakumar, G. and Hoole, S.R.H., "A Parallelized Element by Element Jacobi Conjugate Gradients Algorithm for Field Problems and a Comparison with Other Schemes", *Applied Electromagnetics in Materials*, Vol. 1, pp. 15–28 (1990).

145. Hughes, T.J.R., Analysis of Transient Algorithms with Particular Reference to Stability Behavior, *Computation Methods for Transient Analysis*, Vol. 1, North–Holland, Amsterdam, Eds. T. Belytschko and K. J. Bathe, pp. 67–155 (1983).

146. Anand, L., "Constitutive Equations for the Rate–Dependent Deformation of Metals at Elevated Temperatures", Journal of Engineering Materials and Technology,Transactions of the ASME, Vol. 104, pp. 12–17 (1982).

147. Brown, S. B., Kim, K. H., and Anand, L., "An Internal Variable Constitutive Model for Hot Working of Metals", *International Journal of Plasticity*, Vol. 5, pp. 95–130 (1989).

148. Analysis & Design Application Co., Ltd. (ADAPCO), *Sample Cyclic Symmetry Modal Analysis Using ANSYS* (March 1986).

149. Gyimesi, M. and Lavers, J. D., “Generalized Potential Formulation for 3–D Magnetostatic Problems”, *IEEE Transactions on Magnetics*, Vol. 28, No. 4 (1992).

150. Smythe, W. R., *Static and Dynamic Electricity*, McGraw–Hill Book Co., New York, NY (1950).

151. Demerdash, N. A., Nehl, T. W., Fouad, F. A. and Mohammed, O. A., “Three Dimensional Finite Element Vector Potential Formulation of Magnetic Fields in Electrical Apparatus”, *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS–100, No. 8, pp. 4104–4111 (1981).

152. Eggert, G. M., Dawson, P. R., and Mathur, K. K., “An Adaptive Descent Method for Nonlinear Viscoplasticity”, *International Journal for Numerical Methods in Engineering*, Vol. 31, pp. 1031–1054 (1991).

153. Schweizerhof, K. H. and Wriggers, P., “Consistent Linearization for Path Following Methods in Nonlinear FE Analysis”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 59, pp. 261–279 (1986).

154. Zienkiewicz, O. C. and Cormeau, I. C., “Visco–plasticity – Plasticity and Creep in Elastic Solids – A Unified Numerical Solution Approach”, *International Journal for Numerical Methods in Engineering*, Vol. 8, pp. 821–845 (1974).

155. Simo, J. C. and Taylor, R. L., “Consistent Tangent Operators for Rate–Independent Elastoplasticity”, C*omputer Methods in Applied Mechanics and Engineering*, Vol. 48, pp. 101–118 (1985).

156. Hughes, T. J. R., “Numerical Implementation of Constitutive Models: Rate–Independent Deviatoric Plasticity”, published in *Theoretical Foundation for Large–Scale Computations for Nonlinear Material Behavior* (eds. S. Nemat–Nasser, R. J. Asaro and G. A. Hegemier), Martinus Nijhoff Publishers, Dordrecht, The Netherlands (1984).

157. Hughes, T. J. R. and Carnoy, E., “Nonlinear Finite Element Shell Formulation Accounting for Large Membrane Strains”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 39, pp. 69–82 (1983).

158. Nedelec, J.,“Mixed finite elements in R3”, *Numer. Math.*, Vol.35, pp. 315–341, (1980).

159. Anand, L., “Constitutive Equations for Hot–Working of Metals”, International *Journal of Plasticity*, Vol. 1, pp. 213–231 (1985).

160. Abramowitz, M. and Stegun, I. A., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards Applied Mathematics Series 55, p. 966 (1972).

161. Swain, C. G. and Swain, M. S., “A Uniform Random Number Generator That is Reproducible, Hardware–Independent, and Fast”, *Journal of Chemical Information and Computer Sciences*, pp. 56–58 (1980).

162. Kreyszig, Edwin, *Advanced Engineering Mathematics*, 3rd Edition, John Wiley & Sons, Inc. (1972).

163. Hoel, Paul G., *Introduction to Mathematical Statistics*, 3rd Edition, Johnn Wiley & Sons, Inc., p. 196 (1962).

164. Neter, John et al., *Applied Statistics*, Allyn and Bacon, Inc., Boston, MA (1978).

165. Hughes, T. J. R., *The Finite Element Method Linear Static and Dynamic Finite Element Analysis*, Prentice–Hall, Inc., Englewood Cliffs, NJ (1987).

166. Wilson, E. L. and Itoh, Tetsuji, “An Eigensolution Strategy for Large Systems”, *Computers and Structures*, Vol. 16, No. 1–4, pp. 259–265 (1983).

167. Yokoyama, T., “Vibrations of a Hanging Timoshenko Beam Under Gravity”, *Journal of Sound and Vibration*, Vol. 141, No. 2, pp. 245–258 (1990).

168. Coulomb, J. L., “A Methodology for the Determination of Global Electromechanical Quantities from a Finite Element Analysis and its Application to the Evaluation of Magnetic Forces, Torques and Stiffness”, *IEEE Transactions on Magnetics*, Vol. MAG–19, No. 6, pp. 2514–2519 (1983).

169. Zienkiewicz, O. C., Emson, C., and Bettess, P., “A Novel Boundary Infinite Element”, *International Journal for Numerical Methods in Engineering*, Vol. 19, pp. 393–404 (1983).

170. Damjanic, F. and Owen, D. R. J., “Mapped Infinite Elements in Transient Thermal Analysis”, *Computers and Structures*, Vol. 19, No. 4, pp. 673–687 (1984).

171. Marques, J. M. M. C. and Owen, D. R. J., “Infinite Elements in Quasi–Static Materially Nonlinear Problems”, *Computers and Structures*, Vol. 18, No. 4, pp. 739–751 (1984).

172. Li, Hui, Saigal, Sunil, Ali, Ashraf, and Pawlak, Timothy P., “Mapped Infinite Elements for 3–D Vector Potential Magnetic Problems”, *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 343–356 (1994).

173. Gyimesi, M., Lavers, J., Pawlak, T., and Ostergaard,D., “Biot–Savart Integration for Bars and Arcs”, *IEEE Transactions on Magnetics*, Vol. 29, No. 6, pp. 2389–2391 (1993).

174. Forde, W. R. B. and Stiemer S. F., “Improved Arc Length Orthogonality Methods for Nonlinear Finite Element Analysis”, *Computers & Structures*, Vol. 27, No. 5, pp. 625–630 (1987).

175. Nour–Omid B. and Rankin C. C., “Finite Rotation Analysis and Consistent Linearization Using Projectors”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 93, pp. 353–384 (1991).

176. Emson, C.R.I. and Simkin, J., “An Optimal Method for 3–D Eddy Currents”, *IEEE Transactions on Magnetics*, Vol. MAG–19, No. 6, pp. 2450–2452 (1983).

177. Viollet, P.L., “The Modelling of Turbulent Recirculating Flows for the Purpose of Reactor Thermal–Hydraulic Analysis”, *Nuclear Engineering and Design*, 99, pp. 365–377 (1987)

178. Launder, B.E., Spalding, D.B, “The Numerical Computation of Turbulent Flows”, *Computer Methods In Applied Mechanics and Engineering*, Vol. 3, pp 269–289 (1974).

179. Rice, J.G., Schnipke, R.J.,“A Monotone Streamline Upwind Finite Element Method for Convection–Dominated Flows”, *Computer Methods in Applied Mechanics and Engineering*, vol 48, pp.313–327 (1985).

180. Harlow, F.H., Amsden, A.A., “A Numerical Fluid Dynamics Calculation Method for All Flow Speeds”, *Journal of Computational Physics*, Vol 8. (1971).

181. White, F.M., *Viscous Fluid Flow*, Second Edition, McGraw–Hill, New York (1991).

182. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere, New York (1980).

183. Hestenes, Magnus R, and Stiefel, Eduard, “Methods of Conjugate Gradients for Solving Linear System”, J*ournal of Research of the National Bureau of Standards*, Vol 49, No.6 (1952).

184. Reid, J.K.,“On the Method of Conjugate Gradients for the Solution of Large Sparse Sets of linear Euqations”, *Proceedings of the Conference on Large Sparse Sets of Linear Equations* (Ed. J.K. Reid). Academic Press, pp. 231–254 (1971) .

185. Elman, H.C., “Preconditioned Conjugate–Gradient Methods for Nonsymmetric Systems of Linear Equations”, *Advances In Computer Methods For Partial Differential Equations IV*, Vichnevetsky, R., Stepleman, ed., IMACS, pp. 409–413 (1981).

186. More, J.J. and Wright, S.J., *Optimization Software Guide*, SIAM, Philadelphia, p. 13 (1993).

187. Bilger, R.W., “A Note on Favre Averaging in Variable Density Flows”, *Combustion Science and Technology*, Vol. 11, pp. 215–217 (1975).

188. McCalla, M. C., *Fundamentals of Computer–Aided Circuit Simulation*, Kluwer Academic (1988).

189. Vermeer, P.A. and Verrujit, A., “An Accuracy Condition for Consolidation by Finite Elements”, *International Journal for Numerical and Analytical Methods in Geomechanics*, Vol. 5, pp. 1–14 (1981).

190. Tsai, Stephen W. and Hahn, H. Thomas, *Introduction to Composite Materials*, Section 7.2, Technomic Publishing Company (1980).

191. Box, G.E.P., Hunter, W.G., and Hunter, J.S., *Statistics for Experimenters*, John Wiley & Sons, Chapter 10 (1978).

192. Szabo, Barna and Babuska, Ivo, *Finite Element Analysis*, John Wiley & Sons (1991)

193. Chen, M.T. and Ali, A., Random Vibration Analysis: Closed–Form Solutions for Piecewise Linear PSD in Log–Log Scale, *Computers and Structures*, (under preparation).

194. Harichandran, R.S., Closed–Form Solutions for Random Vibration Under Propagating Excitation, *Earthquake Engineering and Structural Dynamics*, Vol. 15, No. 7, pp. 889–899 (1991).

195. Grimes, R.G., Lewis, J.G., and Simon, H.D., A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems, *SIAM Journal Matrix Analysis Applications*, Vol. 15 (1), pp. 228–272 (1994).

196. Rajakumar, C. and Rogers, C.R., The Lanczos Algorithm Applied to Unsymmetric Generalized Eigenvalue Problems, *International Journal for Numercial Method in Engineering*, Vol. 32, pp. 1009–1026 (1991).

197. Gartling, D.K., "*Finite Element Methods for Non–Newtonian Flows*", report SAND92–0886, CFD Dept., Sandia National Laboratories, Albuquerque, NM (1992).

198. Crochet, M.J., Davies, A.R., and Walters, K., *Numerical Simulation of Non–Newtonian Flow*, Elsevier Science Publishers B.V. (1984).

199. Hallquist, John O., *LS–DYNA3D Theoretical Manual*, Livermore Software Technology Corporation (1994)

200. Biro, O., Preis, K., Magele, C., Renhart, W., Richter, K.R., Vrist, G., "Numerical Analysis of 3D Magnetostatic Fields", *IEEE Transaction on Magnetics*, Vol. 27, No. 5, pp. 3798–3803 (1991).

201. Gyimesi, M. and Ostergaard, D., ”Non–Conforming Hexahedral Edge Elements for Magnetic Analysis”, (ANSYS, Inc. internal development), submitted to COMPUMAG, Rio (1997).

202. Gyimesi, M. and Lavers, D., ”Application of General Potential Formulation to Finite Elements”, Second Japan Hungarian Joint Seminar on Electromagnetics, Sapporo, Japan (1992). *Applied Electromagnetics in Materials and Computational Technology*, ed. T. Honma, I. Sebestyen, T. Shibata. Hokkaido University Press (1992).

203. Preis, K., Bardi, I., Biro, O., Magele, C., Vrisk G., and Richter, K. R., “Different Finite Element Formulations of 3–D Magnetostatic Fields”, *IEEE Transactions on Magnetics*, Vol. 28, No. 2, pp. 1056–1059 (1992).

204. Nedelec, J.C., “Mixed Finite Elements in R3”, *Numerical Methods*, Vol. 35, pp. 315–341 (1980).

205. Van Welij, J.S., “Calculation of Eddy Currents in Terms of H on Hexahedra”, *IEEE Transactions on Magnetics*, Vol. 18, pp. 431–435 (1982).

206. Kameari, A., “Calculation of Transient 3D Eddy Current Using Edge Elements”, *IEEE Transactions on Magnetics*, Vol. 26, pp. 466–469 (1990).

207. Jin, J., *The Finite Element Method in Electromagnetics*, John Wiley and Sons, Inc., New York (1993)..

208. Whitney, H., *Geometric Integration Theory*, Princeton U. P. (Princeton) (1957).

209. Stratton, J.A., *Electromagnetic Theory*, Section 1.14, McGraw–Hill, New York (1941).

210. Mitzner, K.M., “An Integral Equation Approach to Scattering From a Body of Finite Conductivity”, *Radio Science*, Vol. 2, pp. 1459–1470 (1967).

211. Mittra, R. and Ramahi, O., “Absorbing Boundary Conditions for the Direct Solution of Partial Differential Equations Arising in Electromagnetic Scattering Problems”, *Finite Element Finite Difference Methods in Electromagnetic Scattering*, Vol. II, pp. 133–173 (1989).

212. Peric, D. and Owen, D.R.J., "Computational Model for 3–D Contact Problems with Friction Based on the Penalty Method", *International Journal for Numercial Method in Engineering*, Vol. 35, pp. 1289–1309 (1992).

213. Cescotto, S. and Charilier, R., "Frictional Contact Finite Elements Based on Mixed Variational Principles", *International Journal for Numercial Method in Engineering*, Vol. 36, pp. 1681–1701 (1992).

214. Cescotto, S. and Zhu, Y.Y., "Large Strain Dynamic Analysis Using Solid and Contact Finite Elements Based on a Mixed Formulation – Application to Metalforming", *Journal of Metals Processing Technology*, Vol. 45, pp. 657–663 (1994).

215. Simo, J.C. and Laursen, T.A., "An Augmented Lagrangian Treatment of Contact Problems Involving Friction", *Computers and Structures*, Vol. 42, pp. 97–116 (1992).

216. Laursen, T.A. and Simo, J.C., "Algorithmic Symmetrization of Coulomb Frictional Problems Using Augmented Lagrangians", *Computers Methods in Applied Mechanics and Engineering*, Vol. 198, pp. 133–146 (1993).

217. Barry, A., Bielak, J., and MacCamy, R.C., "On absorbing boundary conditions for wave propagations", *Journal of Computational Physics.,* Vol. 79(2), pp. 449–468 (1988).

218. Kallivokas, L.F., Bielak J. and MacCamy, R.C., "Symmetric Local Absorbing Boundaries in Time and Space", *Journal of Engineering Mechanics,* Vol. 117(9), pp. 2027–2048 (1991).

219. Hughes, T.J.R., "Generalization of Selective Integration Procedures to Anisotropic and Nonlinear Media", *International Journal for Numerical Methods in Engineering*, Vol. 15, No. 9, pp. 1413–1418 (1980).

220. Nagtegaal, J.C., Parks, D.M., and Rice, J.R., "On Numerically Accurate Finite Element Solutions in the Fully Plastic Range", *Computer Methods in Applied Mechanics and Engineering*, Vol. 4, pp. 153–178 (1974).

221. Gyimesi, Miklos and Ostergaard, Dale, "Mixed Shape Non–Conforming Edge Elements", *CEFC '98*, Tucson, AZ (1998).

222. Ostergaard, Dale and Gyimesi, Miklos, “Analysis of Benchmark Problem TEAM20 with Various Formulations”, *Proceedings of the TEAM Workshop*, COMPUMAG Rio, pp. 18–20 (1997).

223. Ostergaard, Dale and Gyimesi, Miklos, “Magnetic Corner: Accurate Force Computations”, *Analysis Solutions*, Vol 1, Issue 2, pp. 10–11 (1997–98).

224. Brooks, A.N. and Hughes, T.J.R., “Streamline Upwind/Petro–Galkerin Formulation for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier–Stokes Equations”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, pp. 199–259 (1982).

225. Demerdash, N.A. and Arkadan, A.A., “Notes on FEM Modeling of Permanent Magnets in Electrical Devices”, *FEM for Electromagnetic Applications*,Section 3, p.26–7 (17),(19) (1981).

226. Demerdash, N.A. and Nehl, T.W., “Determination of Inductances in Ferrite Type Magnet Electric Machinery by FEM”, *IEEE Trans. on MAG*, Vol.18, pp.1052–54, (1982).

227. Nehl, T.W., Faud, F.A. and Demerdash, N.A., “Determination of Saturated Values of Rotation Machinery Incremental and Apparent Inductances by an Energy Perturbation Method”, *IEEE Trans. on PAS*, Vol.101. pp.4441–51 (1982).

228. Gyimesi, Miklos, Zhulin, Vladimir and Ostergaard, Dale, “Particle Trajectory Tracing in ANSYS”, Fifth International Conference on Charged Particle Optics, Delft University, Netherlands. To be Published in *Nuclear Instruments and Methods in Physics Research*, Section A (1998).

229. Gyimesi, Miklos and Ostergaard, Dale, “Inductance Computation by Incremental Finite Element Analysis”, *CEFC 98*, Tucson, Arizona (1998)

230. Gyimesi, M. and Lavers, D., “Electromagnetic Particle Trajectory Tracing” Unpublished Report, Available Upon Request from ANSYS, Inc. (1994).

231. Demerdash, N.A. and Gillott, D.H., “A New Approach for Determination of Eddy Currents and Flux Penetration in Nonlinear Ferromagnetic Materials”, *IEEE Trans. on Magnetics*, Vol. 10, pp. 682–685 (1974).

232. Flanagan, D.P. and Belytschko, T., “A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control”, *International Journal for Numerical Methods in Engineering*, Vol. 17, pp. 679–706 (1981).

233. Vogel, F., “Topological Optimization of Linear–Elastic Structures with ANSYS 5.4.”, *NAFEMS Conference on Topological Optimization* (1997).

234. Mlejnek, H.P. and Schirrmacher, R., “An Engineer’s Approach to Optimal Material Distribution and Shape Finding”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 106, pp. 1–26 (1993).

235. Bendsoe, M.P. and Kikucki, N., “Generating Optimal Topologies in Structural Design Using a Homogenization Method”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, pp. 197–224 (1988).

236. Bonet, Javier and Wood, Richard D., *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press (1997).

237. Simo, J.C. and Vu–Quoc, L., “A Three Dimensional Finite Strain Rod Model. Part II: Computational Aspects”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 58, pp. 79–116 (1986).

238. Ibrahimbegovic, Adnan, “On Finite Element Implementation of Geometrically Nonlinear Reissner’s Beam Theory: Three–dimensional Curved Beam Elements”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 122, pp. 11–26 (1995).

239. Vago, Istvan and Gyimesi, Miklos, *Electromagnetic Fields*, Published by Akademiiai Kiado, Budapest, Hungary (1998).

240. Flugge, S., *Encyclopedia of Physics*, Vol. 16, “Electric Fields and Waves”, Springer, Berlin (1958).

241. Lagally, M., Vorlesungen uber Vektorrechnung, Geest u. Portig, Peipzip (1964).

242. Flanagan, D.P. and Belytschko, T., “A Uniform Strain Hexahedron and Quadrilateral with Orthogonal Hourglass Control”, *International Journal for Numerical Methods in Engineering*, Vol. 17, pp. 679–706 (1981).

243. Calllen, H.B., *Thermodynamics and Introduction to Thermostatistics*, 2nd Edition, Wiley & Sons, New York, NY, p. 84 (1985).

244. Chaboche, J.L., “Equations for Cyclic Plasticity and Cyclic Viscoplasticity”, *International Journal of Plasticity*, Vol. 7, pp. 247–302 (1989).

245. Chaboche, J.L., “On Some Modifications of Kinematic Hardening to Improve the Description of Ratcheting Effects”, *International Journal of Plasticity*, Vol. 7, pp. 661–678 (1991).

246. Timoshenko, *“Theory of Elastic Stability”*, McGraw Hill Book Company (1961).

247. Schulz, M. and Fillippou, F. C.., *“Generalized Warping Torsion Formulation”*, Journal of Engineering Mechanics, pp. 339–347 (1998).

248. Gyimesi, M. and Ostergaard, D., “Electro–Mechanical Capacitor Element for MEMS Analysis in ANSYS”, Proceedings of Modelling and Simulation of Microsystems Conference, Puerto Rico, pp. 270 (1999).

249. Gyimesi, M. and Ostergaard, D., “Capacitance Computation with Ammeter Element”, University of Toronto, Department of Electrical Engineering, Unpublished Report (available upon request from ANSYS, Inc.) (1993).

250. Gyimesi, M. and Lavers, D., “Open Boundary Problems”, University of Toronto, Department of Electrical Engineering, Unpublished Report (available upon request from ANSYS, Inc.) (1992).

251. Hieke, A., Siemens and IBM, “ANSYS APDL for Capacitance”, Proceedings from ‘Second International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators’ San Juan, Puerto Rico, pp. 172, (1999).

252. Simo, J.C. and Hughes, T.J.R., *Computational Inelasticity*, Springer–Verlag (1997).

253. Voce, E., “Metallurgica”, Col. 51, pp. 219 (1955).

254. Press, W.H., *Numerical Recipes in C: The Art of Scienfitic Computing*, Cambridge University Press (1993).

255. Gyimesi, M., Lavers, D., Ostergaard, D., and Pawlak, T., “Hybrid Finite Element – Trefftz Method for Open Boundary Anslysis”, COMPUMAG, Berlin 1995, *IEEE Transactions on Magnetics*, Vol. 32, No. 3, pp. 671–674 (1996).

256. Gyimesi, M. and Lavers, D., “Application of the Trefftz Method to Exterior Problems”, University of Toronto, Department of Electrical Engineering, unpublished report. Available upon request from ANSYS, Inc. (1992).

257. Gyimesi, M. and Lavers, D., “Application of the Trefftz Method to Exterior Problems”, University of Toronto, Department of Electrical Engineering, unpublished report. Available upon request from ANSYS, Inc. (1993).

258. Gyimesi, M. and Lavers, D., “Implementation to the Exterior Trefftz Element”, University of Toronto, Department of Electrical Engineering, unpublished report. Available upon request from ANSYS, Inc. (1993).

259. Trefftz, E., “Ein Gegenstuck zum Ritz’schen Verfahren”, Proceedings of the Second International Congress on Applied Mechanics, Zurich (1926).

260. Trefftz, E., “Mechanik det elastischen Korper”, In Vol. VI of Handbuch der Physik, Berlin (1928). Translated from Matematicheskais teoriia Uprognosti, L. GTTI (1934).

261. Herrera, I., “Trefftz Method” (in progress), *Boundary Element Methods*, Vol. 3, (Wiley), New York (1983).

262. Zienkiewicz, O.C., “The Generalized Finite Element Method and Electromagnetic Problems”, COMPUMAG Conference (1978).

263. Zielinski, A.P. and Zienkiewicz, O.C., “Generalized Finite Element Analysis with T–Complete Boundary Solution Function”, *International Journal for Numerical Methods in Engineering*, Vol. 21, pp. 509–528 (1985).

264. Zienkiewicz, O.C., Kelly, D.W., and Bettess, P., “The Coupling of the Finite Element Method and Boundary Solution Procedures”, *International Journal for Numerical Methods in Engineering*, Vol. 11, pp. 355–375 (1977).

265. Zienkiewicz, O.C., Kelly, D.W., and Bettess, P., “Marriage a la mode – The Best of Both Worlds (Finite Element and Bpoundary Integrals)”, in *Energy Methods in Finite Element Analysis*, John Wiley, New York (1979).

266. Jirousek, J. and Guex, L., “The Hybrid–Trefftz Finite Element Model and its Application to Plate Bending”, *International Journal for Numerical Methods in Engineering*, Vol. 23, pp. 651–693 (1986).

267. Mayergoyz, I.D., Chari, M.V.C., and Konrad, A., “Boundary Galerkin’s Method for Three–Dimensional Finite Element Electromagnetic Field Computation”, *IEEE Transactions on Magnetics*, Vol. 19, No. 6, pp. 2333–2336 (1983).

268. Chari, M.V.K., “Electromagnetic Field Computation of Open Boundary Problems by Semi–Analytic Approach”, *IEEE Transactions on Magnetics*, Vol. 23, No. 5, pp. 3566–3568 (1987).

269. Chari, M.V.K. and Bedrosian, G., “Hybrid Harmonic/Finite element Method for Two–Dimensional Open Boundary Problems”, *IEEE Transactions on Magnetics*, Vol. 23, No. 5, pp. 3572–3574 (1987)

270. Arruda, E.M. and Boyce, M.C., “A Three–dimensional Constitutive Model for the Large STretch Behavior of Rubber Elastic Materials”, *Journal of the Mechanics and Physics of Solids*, Vol. 41 (2), pp. 389–412 (1993).

271. Bergstrom, J.S. and Boyce, M.C., “Constitutive Modeling of the Large Strain Time–dependent Behavior of Elastomers”, *Journal of the Mechanics and Physics of Solids*, Vol. 45 (5), pp. 931–954 (1998).

272. Glass, M.W., “Chaparral – A library package for solving large enclosure radiation heat transfer problems”, Sandia National Laboratories, Albuquerque, NM (1995)

273. Diaz, A.R. and Kikucki, N., “Solutions to Shape and Topology Eigenvalue Optimization Problems using a Homogenization Method”, *International Journal for Numerical Methods in Engineering*, Vol. 35, pp 1487–1502 (1992).

274. Ladeveze, P. and Leguillon, D., “Error estimation procedure in the finite element method and applications”, *SIAM Journal of Numerical Analysis*, Vol. 20 (3), pp. 483–509 (1983).

275. Synge, J.L., The Hypercircle in Mathematical Physics, Cambridge University Press (1957).

276. Cohen, M.F. and Greenberg, D.P., “The Hemi–Cube: A Radiosity Solution for Complex Environments”, *Computer Graphics*, Vol. 19, No. 3, pp. 31–40 (1985).

Index

Theory Reference Index