NUMERICAL SIMULATION OF NONLINEAR FREE-SURFACE WAVES ON A TURBULENT OPEN-CHANNEL FLOW

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF CIVIL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ben Rockwood Hodges

March 1997

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert L. Street
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jeffrey R. Koseff

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Stephen G. Monismith

Approved for the University Committee on Graduate Studies:

Utram bibis? Aquam an undam?

- John Fowles

for Evalyn

Abstract

The study of free-surface flows covers a wide range of engineering and environmental flows, including such areas as small-scale bubble dynamics, liquid film deposition, solidification, wave mechanics, flow about a ship or offshore structure, open-channel flows, and ocean-atmosphere interactions. In the present work, we are interested in examining the kinematics of the interaction between surface waves and a turbulent current. This is a basic research task that is directed at increasing our understanding of the processes that occur where surface waves and turbulent undercurrents interact. Our primary motivation is two-fold: (1) to investigate the physical processes of wave-turbulence interaction in the near-surface region that effects both the "signature" of turbulence at the free surface and the mixing beneath the surface, and (2) to develop a numerical simulation method that can be used for future investigations of free-surface phenomena where finite-amplitude waves, turbulence, and structures (such as ship hulls) interact with viscous and nonlinear effects.

Through the use of a newly-developed method of numerical simulation, in this dissertation we show that turbulence in the near-surface region is enhanced by the interactions between non-breaking waves and a turbulent shear current. The irrotational strains of the wave-induced velocity field serve to rotate and stretch the sheared turbulence in the current. As a wave crest passes over a turbulent current, it pulls the turbulent structures up toward the crest and into the region above the level of the trough. This can be considered an enhancement of the turbulent "stirring" of the fluid. As the trough approaches, the turbulence becomes trapped in and near the free-surface boundary layer. Successive crests and troughs appear to have a "pumping" effect on the turbulence trapped in the this near-surface region. This intensification of the turbulent structures is due to the stretching and compression of vortex lines, which can be seen in data animations of the fluctuating enstrophy. These effects are demonstrated to occur with non-breaking finite-amplitude waves, and possibly contribute to the persistence of turbulent structures at a free-surface in the wake of a ship.

This dissertation includes a detailed study of the numerical method developed for a time-accurate simulation of an unsteady turbulent free-surface flow in three

space-dimensions with a progressive, finite-amplitude, free-surface wave. The simulation uses the time-dependent Navier-Stokes equations with the nonlinear kinematic and dynamic boundary conditions. The method is demonstrated to be accurate in the simulation of two-dimensional and three-dimensional flows of laminar standing waves. Comparisons with the experimental studies validate the method for turbulent free-surface flows, and provide insight into the flow behavior seen in laboratory flumes. Visualization of instantaneous and phase-averaged flow variables provide insight into the dynamics of the of the wave-turbulence interactions.

From this study several conclusions can be drawn: (1) simulation of three-dimensional, unsteady turbulent free-surface flows is practical using the numerical method developed in this research project; (2) a finite-amplitude surface wave can cause vertical stirring of the fluid where a surface wave propagates over a current with a strong shear; (3) turbulence in the near-free surface region is enhanced by interaction of the wave straining field acting on turbulent structures; and (4) rapid distortion of the turbulence by the wave straining field plays an important role in the wave/turbulence interaction.

Acknowledgments

At the beginning of a graduate study program, a new student should be handed a ledger and told to keep a list of all the people to whom debts are owed - the list will grow to be much too long to remember without a written record. Unfortunately, I don't have such a ledger, so I'm sure to leave somebody out.

I owe many thanks to my colleagues at the Environmental Fluid Mechanics Laboratory at Stanford. Many of their comments, conversations and insights have found their way into my thoughts and onto these pages (mostly without attribution). In particular, the development and implementation of the numerical method would have remained a Sisyphean task without the concurrent numerical work of Ron Calhoun, Lester Yuan, and Rajat Garg. Their knowledge of the complexities of numerics and skilled bug identification and elimination has been vital to the successful development of the numerical code. The analysis of the simulation results has gone through several iterations, thanks to the prodding of Todd Cowen. His experimental work and insights have been invaluable in developing my understanding of the physics of wave/current interactions. My special thanks are to those who went above and beyond the call of duty (including Ron, Todd, Rajat and my advisor, Bob Street) and provided me a bed (or floor) when I was on the Seattle/Stanford commute.

My thanks to Professors Harry Yeh, Catherine Petroff and Stephen Burges and the University of Washington for their help in getting around the hurdles of being a resident visiting graduate student. I also am indebted to the staff of the University of Washington Visualization Laboratory at the Locke Computer Center. Their knowledge and skills were indispensable in helping me put together video animations of the simulation results.

At Stanford, I am particularly indebted to Dr. Yan Zang and Professors Jeff Koseff and Steve Monismith. Not only did they agree to read and critique this dissertation, but they also have been vital sources of information and assistance during my studies and research.

As for my advisor, what can I say? Professor Robert L. Street is the one who brought me to the cave where the big hairy monster resides. He told me not to worry; I could defeat the monster. Well, the monstrous problem of practical free-surface

simulations for the Navier-Stokes equations is not slain, but I believe it is severely wounded. Perhaps others (or I) will give it the *coup de grace* in the near future. I never could have made it this far without the help and support of Bob (but then, he's the one who threw me into the cave in the first place). All such recriminations aside, it has been a pleasure and a privilege to be a student of Bob Street. I have a great debt to him for his faith in me and for giving me the opportunity to study at Stanford.

I owe special thanks to our research sponsor: Dr. Edwin P. Rood, scientific officer at the Fluid Dynamics Program, Mechanics and Energy Conversion Division of the Office of Naval Research, (support provided through Grant N00014-94-1-0190). I appreciate greatly the long-term support through the completion of this dissertation, even though my temporal completion estimates always turned out on the low side due to the numerical complexities of moving-grid simulations.

Finally, I could not have done this without Evalyn. Her insistence that I speak of graduation as a *'when'* rather than an *'if'* was often the only difference between success and Burger King. Her ability to hold down a stressful job, put up with a stressed-out husband, and still find time to bring two wonderful children into the world (all during the course of my studies!) makes her nothing short of phenomenal.

Table of Contents

List of Tables

List of Figures

Nomenclature

This dissertation uses tensor notation with the Einstein summation convention implied unless otherwise specifically noted.

a	wave amplitude
$\mathcal{A}$	wave amplitude scale
B_i	discrete operator for pressure gradient
c	wave speed
C_i	discrete operator for convective terms
C_r	scale-similarity coefficient
C_α^α	cross term in filtered kinematic boundary condition
$\mathcal{C}_\lambda$	relative capillarity
D	depth
$\mathcal{D}$	depth scale
D_α	: $\alpha = 1, 2, 3$; discrete operators for approximate-factorized diffusive terms
D_I	implicit discrete operator for diagonal diffusive terms
D_E	explicit discrete operator for off-diagonal diffusive terms
e_{ij}	rate-of-strain tensor
$\bar{e}_{ij}$	convective rate-of-strain tensor
$\tilde{e}_{ij}$	wave-induced rate-of-strain tensor
$\hat{e}_{ij}$	turbulent rate-of-strain tensor
F_{ij}	flux tensor in Cartesian momentum equation
$\mathcal{F}_i^q$	flux tensor in curvilinear momentum equation
F	function representing position of free surface
g_i	or $\mathbf{g}$, gravitational force vector
g'	magnitude of gravity in z direction
$\mathcal{G}$	gravity scale (magnitude)
G^{qr}	contravariant metric tensor
G_{qr}	covariant metric tensor
Δh	hydrostatic head (in units of length)
H	free surface height measured in physical space
$\mathcal{H}$	free surface height measured in curvilinear space
I	identity matrix
J	Jacobian
k	or K, wave number
$L_{ij}^{(m)}$	modified Leonard tensor

$\mathcal{L}$	length scale
$\mathcal{L}_\alpha$	Leonard stress operator in filtered kinematic boundary condition.
L^α_α	Leonard term in filtered kinematic boundary condition
M	mean curvature of surface
n_i	normal unit vector
p	modified pressure
p_d	dynamic pressure
P	pressure
Q_i	discrete operator for grid motion
R^α_α	Reynolds term in filtered kinematic boundary condition
$\mathcal{R}_u$	velocity ratio scale
R_1, R_2	radii of curvature of surface
S_i	source of momentum equation
$\mathcal{S}_i$	source of curvilinear momentum equation
$\mathcal{S}$	shallowness scale
S^q_j	surface metric tensor
t	time
t_i	tangent unit vector
T	wave period
$\mathcal{T}$	time scale
u_τ	bottom boundary shear velocity
u_b	bulk velocity
u_i	Cartesian velocity
u^*_i	intermediate Cartesian velocity or non-dimensional velocity
U^q	contravariant velocity
U^{*q}	intermediate contravariant velocity
$\mathcal{U}$	characteristic velocity
$\mathcal{U}_\tau$	characteristic velocity of turbulence
$\mathcal{U}_c$	characteristic velocity of convection
$\mathcal{U}_w$	characteristic velocity of wave
x_i	Cartesian (physical space) coordinates
$\dot{X}^q$	contravariant grid velocity
$\mathcal{X}$	surface-parallel grid scale
$\mathcal{Z}$	surface-normal grid scale
z	Cartesian vertical coordinate
z^*	stretched vertical grid coordinate

Greek Symbols

α	velocity scale factor
β	Stokes layer scale
δ_{ij}	Kronecker delta
η	free surface height
ϵ	wave slope scale
ε	wave slope (ak)
Δ	discrete difference
γ	surface tension coefficient
λ	wavelength
μ	dynamic viscosity
ν	kinematic viscosity
ν_T	eddy viscosity
θ	open-channel slope
ϕ	discrete pressure variable
Ψ	modified pressure in streamwise direction
ρ	density
τ	time (at fixed location in computational space)
τ_{ij}	subgrid-scale stress tensor
ξ^q	curvilinear coordinates
σ	or Σ, wave frequency
ω	vorticity
Ω	mean vorticity

Subscripts

i, j, k	indices for Cartesian or covariant vectors
$s+$	quantity computed in air at free surface
$s-$	quantity computed in water at free surface

Superscripts

q, r, s, t	indices for contravariant vectors
n, $n+1$	discrete time step

Abbreviations

ADI	Alternating Direction Implicit
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy stability condition
DBC	Dynamic Boundary Condition
DNS	Direct Navier-Stokes (simulation)
DPTV	Digital Particle Tracking Velocimetry
DTM	Dynamic Two-parameter Model
Fr	Froude number
LES	Large-Eddy Simulation
KBC	Kinematic Boundary Condition
MAC	Marker-And-Cell (numerical method)
N-S	Navier-Stokes (equations)
RWCF	Recirculating Wave-Current Flume
RMS	Root-Mean Square (variance)
TWCF	Tilting Wave-Current Flume
Re	Reynolds number
We	Weber number

Chapter 1

Introduction

The study of wind, wave, swell, and current interactions is an area where the interests of a number of disciplines collide. For oceanographers, the water's surface is the upper boundary of their domain, so parameterization of the interactions at the surface are important for developing accurate boundary conditions for ocean circulation simulations. Atmospheric scientists are interested in parameterizations of the surface for the lower boundary of their simulations. Researchers in global climate change are concerned with the mixing and transfer of heat and greenhouse gases across the water's surface. Biologists are interested in the manner in which mixing in the near surface region enhances productivity (which may also effect CO_2 absorption). Naval architects are interested in the kinematics and dynamics of the free surface up to the scale of the largest wavelengths. Aerospace, mechanical, ocean, and naval engineers are interested in the small-scale structures at the surface which hold the signature of a ship's wake that can be seen using synthetic aperture radar (or visually from the space shuttle when the sun glints off the waves). Civil/environmental engineers are interested in such areas as the surface effects on river and estuarine systems, lakes, sediment transport, and beach erosion. If you broaden the area of study to include all free-surface flows, then mechanical, industrial, chemical, and aerospace engineering become intertwined. Most of these disciplines are also interested in the study of free-surface flows as a fundamental flow of fluid dynamics: a complex and ubiquitous flow in engineering and the environment that is yet to be fully understood and has not been fully captured by solvable equations, detailed experimentation, or practical numerical simulations.

In the study of global climate change, it has become apparent that the linkage between the atmosphere and the oceans may be one of the most important (and least understood) parameters that govern long-term oscillations of the climate. The ocean is a potential sink for large amounts of CO_2 and thus may serve to mitigate the effects of increased CO_2 production caused by the progressive industrialization of developing

countries. Accurate calculation of the long-term effect of the ocean on the climate requires prediction of the rate at which absorption of CO_2 will occur. The answer to this question is linked to the small-scale dynamics and mixing processes at the ocean surface. To determine the rate of mixing at the ocean's surface, we must first obtain a better understanding of the mixing processes in the interactions between wind, waves and currents.

This dissertation is directed toward increasing our knowledge of how non-breaking waves on the the water's surface interact with turbulent currents below. This is a basic research task that has practical application in the study of near-surface ocean dynamics as well as in examining the persistence of turbulent ship wakes.

1.1 Motivation and objectives

In the study of geophysical flows, the scales of interest are so large that the interface between oceanographic circulations and atmospheric circulations can only be provided by simple modeling assumptions. The processes that control the interactions at the water's surface are inherently nonlinear and small scale. These processes are as yet not fully understood due to the difficulty in modeling the problem in a solvable and accurate manner.

A landsman's view of the ocean's surface tends to be of a vast expanse of breaking waves. Wave-breaking is both visually and dynamically dramatic and results in augmented mixing near and across the ocean's surface. The study of this mixing is a challenging problem that is being attacked on many fronts, but will not be addressed in this thesis. Instead, we are motivated by the problem of the majority of the ocean's surface, where the waves may not be breaking, but are long rolling swells with smaller parasitic waves. These waves and swells are moving over and interacting with turbulent currents in the near-surface layer. It is important to understand the interactions between turbulent currents and surface motions since this can effect the transfer of heat, mass, and momentum near the ocean surface. The area of the ocean surface covered by non-breaking waves is large, so even small-scale effects can have global implications.

The study of turbulent interactions, mixing, and stirring of a fluid near a free surface requires an understanding of the kinematics and dynamics of the surface. This proves challenging due to (1) the inherent nonlinearity of finite-amplitude water waves, (2) the dissipative and anisotropic effects of turbulence with a mean shear, and (3) the fine scale over which viscous effects are important beneath a water wave. This motivates our primary objective for this research: the development and application of a time-accurate numerical simulation method for three space dimensions that is capable of (1) simulating non-steady-state wave motions and (2) resolving the viscous boundary layer beneath a finite-amplitude wave traveling over a turbulent current. Our goal is to develop a research tool that can be used to investigate the turbulent structure beneath a free-surface wave, and then use the tool to develop a framework for understanding the interactions between waves and a turbulent current.

One of the difficulties with free-surface numerical methods presented in the literature is that they are generally so complex that their use and adaption by other research centers is difficult (*e.g.* the Dommermuth [1993] method designed for massively parallel machines). As noted by Sarpkaya (1996):

> *...the modeling of free-surface phenomena still poses difficulties, not only because of an insufficient understanding of the physics of the vorticity/free-surface interaction, but also because of the necessity to devise and use mathematical formulations, numerical schemes, and physical-property experiments of far greater complexity than had hitherto been used...*

This provides the motivation for our secondary objectives: (1) to develop a numerical method that is straightforward in its implementation, using the least-complicated algorithms consistent with maintaining second-order accuracy in time and space; and (2) to investigate the numerical requirements and limitations for practical simulations of free-surface flows.

1.2 Approach

This study of the turbulent wave/current interactions uses a numerical simulation of the Navier-Stokes equations with the dynamic and kinematic boundary conditions. Large-Eddy Simulation (LES) techniques are applied to keep the number of grid

cells required to a minimum while still attaining reasonable resolution of the large turbulent scales of motion. The flow studied is a temporally-evolving open-channel flow driven by a constant pressure gradient in the streamwise direction. The bottom boundary of the channel is a no-slip boundary, producing turbulence with coherent structures that interact with the free surface. A train of finite-amplitude waves with a wavy-flow field is superposed over the open-channel flow as an initial condition and the system is allowed to evolve with time.

1.3 Background

This work is concerned with surface waves, turbulence, open-channel flows, and numerical simulation methods for free-surface flows: an extensive range that makes it impossible to provide an in-depth background for each area. This section provides a look at a few key recent developments in research of (1) surface waves and turbulent currents, (2) open-channel flows, (3) interactions between turbulence and a free surface, and (4) numerical simulations of viscous free-surface flows.

1.3.1 Finite-amplitude waves and turbulent currents

The non-breaking long waves on the ocean's surface (commonly called "swell" to distinguish from local wind waves) have kinematic behavior that is generally well-described by classic irrotational finite-amplitude wave theory. However, such theory is of limited value in the study of (1) nonlinear interactions between waves, (2) nonlinear interactions with currents, or (3) viscous interactions with turbulence. The latter two areas are of primary concern in the present work. Viscous effects for laminar flow with small-amplitude waves were derived by Longuet-Higgins (1953). For finite-amplitude waves, Dore (1977) proposed a viscous model that has yet to be conclusively demonstrated either experimentally or numerically. A number of theories of wave-turbulence interaction have been proposed[1], but none have been proven due to the difficulties in obtaining data either experimentally or numerically.

[1]See Cowen (1996) for a detailed discussion of wave-turbulence theories.

Investigating water waves and turbulence with either physical or numerical experiments presents a similar set of problems: (1) the motion of the free surface is rapid, causing stability problems in numerical simulations and data collection problems in laboratory experiments, (2) the viscous boundary layer beneath a water gravity wave is small, $O(10^{-4}\ m)$, yet it must be resolved both in numerical and laboratory experiments to obtain a clear picture of the turbulence near the free surface, and (3) waves on a turbulent flow are a macroscopic phenomena that evolve in space and time, creating problems in obtaining a reasonable ensemble for computing turbulent statistics. More details of the problems associated with gathering experimental data under the free surface in the laboratory will be found in Cowen (1996). Further exposition of problems in numerical simulations will be found in chapter 4 of this dissertation.

Cowen's (1996) study provides experimental data within the viscous boundary layer beneath a water-wave that is unique in the literature. The ability of his digital particle tracking velocimetry (DPTV) technique to provide data in the region between the still-water level and the crest allows the first experimental look at the fine-scales of turbulence beneath a wave. Cowen's work shows that the capillary waves in laboratory channels become the dominant fluctuating effects in the near surface region, making it *"...difficult to differentiate between the effects of the gravity wave on turbulence and the effects of capillary waves."* The present numerical work has the complementary problem: resolution of capillary waves on gravity waves requires an extremely fine grid in the streamwise, spanwise and surface normal directions, which is impractical with current computational power for a four-dimensional simulation. As a consequence, experimental techniques such as Cowen's are suited for the investigation of combined capillary/gravity effects, while numerical work is ideal for studying either capillary or gravity waves, but not the combination.

1.3.2 Turbulent open-channel flows

The systematic study of turbulent open-channel flows has a history that begins with Leonardo da Vinci's studies of the behavior of rivers, a necessary part of his civil engineering works in supplying water to cities (Tokaty [1971]). The engineering-oriented nature of da Vinci's inquiries sets the tone for most of the open-channel flow studies conducted until the development of hot-film anemometers and advanced flow visualization techniques in the 1960's (Nezu and Nakagawa [1993]). With these

techniques came the ability to study directly the turbulence in water flows. Most open-channel flow literature is directed at the study of flows without significant wave motion (supercritical flows being a prominent exception). The open channel provides an ideal test laboratory for studying the evolution of turbulent boundary layers in incompressible flows. A comprehensive review of experimental and numerical work on open channel flows can be found in the recent monograph by Nezu and Nakagawa (1993).

Recent works not discussed in Nezu and Nakagawa (1993) are numerical simulations of open-channel flows by Lam and Banerjee (1992), Pan and Banerjee (1995), Komori *et al.* (1993), McDonald (1994), Borue *et al.* (1995), Salvetti *et al.* (1996), and Garg (1996). Banerjee's research group studied turbulent open-channel flows using the classic "rigid, shear-free lid" approximation where the free-surface is treated as a flat surface that cannot support a shear. This provides a domain for efficient numerical simulations of turbulence in the bottom boundary layer and the free stream. However, the loss of the true dynamic and kinematic boundary conditions makes this approach unsuitable for studying the small-scale viscous realm in the free-surface boundary layer. Furthermore, the implementation of a flat lid as a model for the free-surface prevents this approach from being used to study the effects of waves on the turbulence. A similar approach was taken by Salvetti *et al.* (1996), where the decay of the turbulent field generated by the Direct Navier-Stokes (DNS) simulation of Pan and Banerjee (1995) was duplicated using several subgrid-scale turbulence models. Garg (1996) provides a detailed numerical study of stratified open-channel flow with a flat, free-slip lid. McDonald (1994) developed a method which used a rigid, shear-free lid with a fixed surface wave that propagates over the domain. This allows interactions of free-stream turbulence and wave-induced fluctuations to be analyzed, but not the effects of the viscous boundary layer. The work of Borue *et al.* (1995) provides for movement of the free surface using linearized boundary conditions. The linearization has the unfortunate side-effect of limiting the range of applicability of the simulation method to surface deformations that are less than the free-surface boundary layer thickness, $O(10^{-4}\ m)$ for water. The method can be used to investigate interactions of turbulence with a free surface, but cannot be used with any significant progressive waves. Komori *et al.* (1993) have the only published turbulent open-channel flow with the nonlinear kinematic and dynamic boundary conditions imposed on a true free surface. However their simulations have been limited to free-surface deformations

caused by the turbulent channel flow, which remain small-amplitude and appear to have little effect on the turbulence[2].

For the purposes of the present work, the open-channel flow can be thought of as a convenient testing ground for studying wave/turbulence interaction. The solid boundary at the bottom of the open channel provides the production of turbulence and coherent structures that can interact with the free-surface and surface waves. Our goal is not to study the structures and turbulence produced by the bottom boundary, but to understand how these interact with the flow produced by a progressive train of waves.

1.3.3 Interactions of turbulence and a free surface

Due to the interests of the Office of Naval Research, a significant amount of recent work has addressed the interactions of coherent vortices with a quiescent free surface. A full review will be found in Sarpkaya (1996). The primary aim of research in this area has been to understand how coherent vortex rings (typically produced in the wake of a submarine or ship) interact with a free surface and provide a "signature" that is detectable at the surface. Much of the work in understanding vortex/free surface interactions has been done in laminar regimes (*e.g.* Lught and Ohring [1992], Dommermuth [1993], Wang and Leighton [1990]). A compendium of recent work with turbulent free-surface flows will be found in Rood and Katz (1994). With the exception of Dommermuth *et al.* (1994), the numerical work in this area has been with linearized or rigid-lid shear-free boundary conditions on the surface, which limits the ability to model significant free-surface deformations. Dommermuth *et al.* (1994) studied the interaction of a surface wave packet with grid-stirred turbulence. Using this simulation they were able to elucidate effects of turbulent roughening of the surface, turbulent scattering of the wave packet, and vortical structures induced by the waves. Their approach relies on a modeling assumption for the free-surface viscous boundary layer (which was not resolved in the simulations).

[2] As of this date, there has been no sign in the indexed literature of an open-channel simulation with progressive waves using the Komori *et al.* method. It may be (1) such work has been published in a report not generally available or cited in the literature indexes, (2) the method has proven unsuitable for larger waves, or (3) they simply have not had the time or funding for such research.

Rigid, shear-free lid simulations such as Pan and Banerjee (1995), Salvetti *et al.* (1996) and Walker *et al.* (1996) have provided significant insights into the interactions of turbulence with an idealized free surface. Walker *et al.* (1996) have investigated the decay of initially homogeneous and isotropic turbulence in the presence of a rigid, shear-free surface. Of particular interest is their results which show that the anisotropy of the turbulence caused by the free surface extends to a depth of the turbulent length scale, while the surface effect on vorticity is one-tenth of the turbulent length scale. With the rigid shear-free lid they noted a decrease in the dissipation and an increase in the turbulent kinetic energy near the surface which they attributed to a reduction in tangential enstrophy at the surface. However, Cowen (1996) found sharp increases in both the fluctuating vorticity and dissipation near the free surface in physical experiments, which he attributed to small capillary waves on the free-surface. Gharib *et al.* (1994) also found an increase in enstrophy near the surface in a laboratory experiment, which seems to indicate that the near-surface enstrophy decay in rigid-lid, shear-free surface simulations may be a result of neglecting small surface deformations.

1.3.4 Numerical methods

In order to place our simulation method in the appropriate context, the following section briefly reviews several approaches to viscous free-surface simulation. The references provided are a representative sample of types of simulations found in the literature, and should not be considered an exhaustive listing. A comprehensive review of free-surface simulations will not be attempted as the field has grown too large, and recent review papers by Tsai and Yue (1996) and Floryan and Rasmussen (1989) are available, as well as a textbook on moving-grid/free-surface simulations by Shyy (1996).

We have neglected several areas of free-surface simulation: (1) numerical methods with rigid-lid shear-free boundary conditions, which can generally use standard non-moving grid solution techniques (an exception being McDonald [1994], where a moving shear-free rigid-lid was used as a free-surface model); (2) inviscid free-surface numerical methods, such as those of Tulin's research group (*e.g.* Yao *et al.* [1996]). While these methods have much to offer in the study of free-surface phenomena, they are not directly comparable to the approach in the present work.

Fixed grid *vs.* moving grid

Simulation methods for viscous free-surface flows can roughly be grouped into two categories: (1) "fixed-grid" simulations where the governing equations are discretized in physical space on a fixed (usually Cartesian) grid while the free surface moves within the grid; and (2) "moving-grid" simulations where a new, boundary-conforming grid that moves with the free surface is generated at each time step.

The fixed-grid approach is advantageous for handling changes in the topology of a free surface, such as the merging of two fluids, or the pinching of a single bubble into two bubbles. The two major approaches are (1) variants on the marker-and-cell (MAC) approach – in its recent incarnation typified by the work of Raad *et al.* (1995), and (2) the level-set approach which directly simulates two-fluid interactions (*e.g.* Sussman *et al.* [1994]). However, it is difficult to accurately model the free-surface viscous boundary layer and the interactions of turbulence with a free surface in a fixed-grid approach. Additionally, conservation of mass and physical continuity of the free surface can be problems for fixed-grid simulations.

Moving-grid simulations tend to be more computationally intense than fixed-grid simulations, but are capable of accurately resolving the free-surface boundary layer and turbulent fluid interactions under free-surface waves. Prior to the current work, this capability had not been demonstrated for a turbulent flow with significant surface motion. Numerical conservation of mass and smoothness of the free-surface can be readily achieved in moving-grid simulations as long as the free-surface does not have slope discontinuities. The introduction of discontinuities in surface slope or changes in surface topology are challenges for moving-grid simulations that are a good subject for future research. It is likely that some hybrid technique combining a fixed-grid approach (such as the level-set method) with a moving grid simulation will prove the most effective technique for such a problem.

Coupling between grid generation and flow solution

The numerical techniques used to advance the free surface in moving-grid simulations of viscous free-surface problems fall into two categories: (1) coupled-grid, or (2)

uncoupled-grid. The coupled-grid techniques simultaneously solve implicit discretizations of a coupled set of equations consisting of the Navier-Stokes equations, the kinematic and dynamic boundary conditions, and one or more grid generation equations. The advantage of the coupled-grid technique is that the free-surface motion determined by the kinematic boundary condition is coupled directly to the flow solution and the dynamic boundary condition, which is arguably the most accurate and numerically stable approach. The disadvantage is that significant computational complexity is involved in the implicit coupling of grid generation and free-surface movement to the flow solution. The complexity of this approach can be appreciated by considering that the system of equations may consist of (1) the predominantly-elliptic, incompressible Navier-Stokes equations, (2) a combined pressure/velocity gradient dynamic boundary condition, (3) the hyperbolic kinematic boundary condition, and (4) an elliptic grid generation equation whose source term must be re-evaluated during the iterative solution process to control grid quality. Furthermore, once such a code is developed, the coupling of the equations makes it difficult (if not impossible) to experiment with new algorithms or different numerical methods. Unfortunately, the field of numerical grid generation is not sufficiently mature to provide a set of general-purpose algorithms that can provide efficient implicit coupling with Navier-Stokes flow solvers. To date, the coupled-grid approach for the Navier-Stokes equations has only been implemented in a maximum of three dimensions: either steady-state with three space dimensions such as Farmer *et al.* (1994), or two space dimensions and unsteady in time such as Loh and Rasmussen (1987), Lugt and Ohring (1992), and Hino *et al.* (1993). It appears unlikely that any of these methods could be practically extended to a four-dimensional simulation.

Uncoupled-grid techniques integrate the kinematic boundary condition in time without implicit reference to velocities at the $(n+1)$ time step. This allows the kinematic boundary condition to be solved with only the time (n) velocity data, effectively uncoupling the free surface advance and the grid generation from the solution of the Navier-Stokes equations. Uncoupled techniques sequentially compute the free-surface position and the curvilinear grid prior to advancing the flow solution in time, allowing significant flexibility in the design of a numerical simulation code. Since the flow solver, grid generation, and free surface advance are all separate modules, different numerical approaches can be implemented with minimal effect on other modules. Using an uncoupled approach, we have worked with three grid generation methods

(including two packages designed by other research teams), along with two different numerical algorithms for the free surface advance and two methods for flow solution. The ability to easily change and adapt the numerical algorithms makes the uncoupled technique preferred in a research code. A number of simulation methods in the literature have effectively used uncoupled-grid approaches for unsteady flows with two space dimensions (*e.g.* Park *et al.* [1993] and Hinatsu [1992]). Four-dimensional flows have also proven tractable for uncoupled-grid techniques in the works of Dommermuth (1993), Miyata, *et al.* (1992), Wang and Leighton (1990), and Kassinos and Prusa (1990).

Free-surface advance

The integration of the kinematic boundary condition to advance the free-surface in moving-grid simulations in the literature has been accomplished through five different forms of the Eulerian kinematic boundary condition: (1) numerical integration of the physical space Eulerian kinematic boundary condition (Hino *et al.* [1993], Loh and Rasmussen [1987], and Farmer *et al.* [1994]); (2) numerical integration of a curvilinear transformation of the physical space Eulerian kinematic boundary condition (Park *et al.* [1993], Miyata *et al.* [1992], Hinatsu [1992], Lugt and Ohring [1992], and Komori *et al.* [1993]); (3) rotation of the Cartesian space frame and integration of the physical space Eulerian kinematic boundary condition in a more desirable orientation in Kassinos and Prusa (1990); (4) Helmholtz decomposition of the flow solution, which provides the kinematic boundary condition in terms of the solenoidal velocity potential rather than the Cartesian or curvilinear velocity in Dommermuth (1993); and (5) integration of a curvilinear derivation of the kinematic boundary condition (Hino [1989], and Hodges *et al.* [1996]). The first two methods are unusable for waves which do not remain single-valued in the physical space. The third method, rotation of the coordinate system, allows computation of a free-surface that is multiple-valued in physical space; however, the surface must remain single-valued in the rotated Cartesian space frame, which eliminates the method from use with overturning waves. The fourth approach is unique to Dommermuth's work and has only been implemented in a form suitable for single-valued waves. We have been able to find only two examples of the fifth approach, which were developed independently. In Hino's work, the curvilinear kinematic boundary condition was used without derivation and was applied in a fixed-grid rather than a moving-grid simulation. Section 2.4.2 presents

a derivation of the curvilinear kinematic boundary condition used in our free-surface advance (Hodges *et al.* [1996]). The derivation is based upon a Taylor-series expansion in curvilinear space, and is similar to the derivation of the Eulerian kinematic boundary condition in physical space by Mei (1983). This curvilinear approach is advantageous because it allows the simulation of a free-surface that does not remain single-valued in physical space coordinates. Furthermore, steepening of the surface wave does not degrade the accuracy of the free-surface advance. Most of the approaches to the kinematic boundary condition track the surface by the movement of particles that are required to move vertically in physical space; this has some drawbacks in the accuracy of the free surface advance for steep waves since increasing the slope will increase the skewness of the grid cells.

Moving-grid flow solvers

There has been a wide range of numerical flow solvers used in free-surface simulations. Most of the flow solvers (present work included) are developed through a time-dependent curvilinear transformation of the Eulerian Navier-Stokes equations. This provides an additional convective term to account for the motion of the grid. One of the drawbacks of this approach is that the introduction of the convective term for the grid motion creates a Courant-Friedrichs-Lewy (CFL) stability limit on the movement of the grid. This limits the local deformation of the grid from one time step to the next and can require a time step smaller than that set by the convective CFL limit (see section 4.9.3). An approach which overcomes this limitation has been developed for use in adaptive gridding with internal flows by Laflin and McRae (1996). Instead of applying the grid motion to the convective term, their adaptive approach interpolates the time (n) solution to the time $(n+1)$ grid, where the Navier-Stokes equations can be solved in the standard Eulerian form, thus accommodating large local changes in the grid between time steps. This method has not been applied to a free-surface flow and there remain some open questions as to appropriate interpolation of a 3D flow field in the immediate vicinity of the free surface.

Numerical approaches in the literature

To put our method in perspective, it is worthwhile to briefly discuss some of the numerical approaches that other authors have taken to address moving-grid free surface problems. We limit our discussion to finite-difference techniques and spectral methods which have proved effective in capturing turbulent flows. Finite-element techniques are routinely used for laminar flows, but have not been demonstrated to be practical in turbulent simulations.

Dommermuth (1993) has implemented a sophisticated three-dimensional technique that uses fourth- and sixth-order finite differences with a third-order Runge-Kutta scheme for the unsteady Navier-Stokes equations. Dommermuth's method is unique in that (1) it applies Helmholtz decomposition to split the governing equations into irrotational and vortical flow parts, and (2) it is designed for implementation on massively parallel machines. Komori *et al.* (1993) also uses a Runge-Kutta scheme to integrate the Navier-Stokes equations, however their approach has a second-order RK scheme with fifth-order upwind discretization for nonlinear terms, second-order discretization for other derivatives, and Crank-Nicolson discretization in time for the kinematic boundary condition. Hino *et al.* (1993) have developed an artificial-compressibility multigrid method that is suitable for steady-state solutions about ship hull forms. Lugt and Ohring's (1992) coupled-grid method uses artificial compressibility to solve for unsteady flow in two dimensions. Wang and Leighton (1990) have developed a spectral method that discretizes the Navier-Stokes equations in a vertical-velocity, vertical-vorticity formulation that is suitable for periodic boundary condition problems. Park *et al.* (1993), Hinatsu (1992), Miyata *et al.* (1992), and Yeung and Ananthakrishnan (1992) use fractional-step methods with discretization on staggered grids. The approach of Yeung and Ananthakrishnan is unique in that it uses an additional predictor-corrector method within the fractional-step algorithm so that the corrected free-surface can be advanced with the predicted velocities. This approach requires the solution of two pressure Poisson equations in each time step, but arguably should allow a larger time step while maintaining simulation accuracy.

The present works uses a fractional-step method that is similar to others in the literature with the following major differences: (1) the free surface is advanced by a method that is not limited to single-valued waves; (2) a non-staggered grid is used to reduce storage requirements for metric terms and increase computational efficiency;

(3) an efficient multigrid solver is utilized for solution of the pressure Poisson equation, and (4) the viscous free-surface boundary layer is resolved beneath a finite-amplitude wave.

Limitations of previous works

Free-surface methods in the literature have had a range of limitations in their ability to capture free surface deformations. Recent work in turbulent channel flows has been limited to linearized boundary conditions with infinitesimal waves (Borue *et al.* [1995]), non-deforming surfaces (Pan and Banerjee [1995], Walker *et al.* [1996], McDonald [1994], Salvetti *et al.* [1996]), or small-deformation surfaces with non-linear boundary conditions (Komori *et al.* [1993]). Fixed grid methods have demonstrated their ability to capture large surface deformations from the very start of the genre with the simulations of Harlow and Welch (1965) and (1966) and later improvements such as Chan and Street (1970a) and (1970b). However, the ability of fixed-grid methods to accurately capture the physics of turbulence under a free surface remains an open question. Moving-grid methods for laminar flows have been used to simulate relatively large free-surface deformations with only two space dimensions by a number of authors (Lugt and Ohring [1992], Ananthakrishnan and Yeung [1994], Wang and Leighton [1994], Loh and Rasmussen [1997]). Smaller deformations have been simulated with three space dimensions by several authors, including Miyata *et al.* (1992), and Dommermuth (1993).

For inviscid fluids, large wave deformations (including overturning waves) have been simulated using boundary-element methods for the potential flow problem. Longuet-Higgins and Cokelet (1976) is the classic paper in this area. More recently, the research group under M.P. Tulin (*e.g.* Wang *et al.* [1995]) has made important contributions in this area. However, the methods remain constrained to inviscid flow solutions.

Of particular interest to the extension of free-surface simulations to turbulent regimes is the resolution of the viscous boundary layer beneath the free surface. This boundary layer is extremely thin, on the order of $(2\nu/\sigma)^{1/2}$, where ν is the kinematic viscosity and σ is the surface-wave radian frequency. It appears that only two simulation methods in the recent literature (Dommermuth [1993] and Borue *et al.* [1995]) explicitly consider the thickness of the free-surface boundary layer when

setting up the computational grid. However, it is apparent that a number of the low Reynolds number simulations also have sufficient grid points to resolve the boundary layer, even though it is not explicitly stated or investigated. Numerical simulations of turbulent flows in the literature which resolve the free-surface boundary layer have done so for fluids with a viscosity significantly higher than water (*e.g.* Borue *et al.* [1995] used a kinematic viscosity of 3.3×10^{-4}, about 300 times more viscous than water). While most methods could theoretically be extended to the fine resolution needed for a low-viscosity fluid, the computational cost will be high. The question of resolving the boundary layer is not merely an academic exercise, since the dynamic boundary condition derived from first principles[3] is only appropriate when the free surface boundary layer is resolved[4].

[3]A different dynamic boundary condition can be derived from the Bernoulli equation. This does not require boundary layer resolution, but is not appropriate as a boundary condition on the Navier-Stokes equations.

[4]This requirement is similar to the numerical restriction on applying a no-slip boundary condition for a Dirichlet boundary. Unless the viscous sub-layer is resolved on a no-slip boundary, it is more appropriate to use an approximate matching condition to the log-law profile to get the correct boundary condition.

Chapter 2

Mathematical formulation

2.1 Introduction

The flow of a fluid beneath a free surface is governed by the Navier-Stokes equations subject to conservation of mass in the fluid volume along with kinematic and dynamic boundary conditions at the free surface. In addition, for a temporally-evolving turbulent channel flow, we require a no-slip bottom boundary condition and periodic boundary conditions in the spanwise and streamwise directions. This set of equations is considerably more complex than the equation set for internal flows due to the nonlinear effects of the boundary conditions at the free surface. The kinematic boundary condition is nonlinear in the the velocity and spatial gradients of the surface and is formally a hyperbolic equation. The dynamic boundary condition is generally considered to have a nonlinear effect from its linkage to the nonlinear momentum equations through the surface pressure.

This chapter presents the governing equations for turbulent free-surface channel flow. In section 2.2 equations are non-dimensionalized to extract the relevant physics. Sections 2.3 through 2.5 present derivations of a set of filtered curvilinear equations that are suitable for discretizing using the numerical method presented in chapter 3.

2.2 Dimensional analysis of turbulent channel flows with progressive waves

The Navier-Stokes equations, kinematic boundary condition, and dynamic boundary condition can be presented in a Cartesian tensor form as:

$$\text{momentum} \qquad \rho\frac{Du_i}{Dt} \;=\; \rho g_i \;-\; \frac{\partial P}{\partial x_i} \;+\; \frac{\partial}{\partial x_j}\left\{2\mu\left(e_{ij} \;-\; \frac{1}{3}e_{kk}\delta_{ij}\right)\right\} \tag{2.1}$$

$$\text{continuity} \qquad \frac{1}{\rho}\frac{D\rho}{Dt} \;+\; \frac{\partial u_j}{\partial x_j} \;=\; 0 \tag{2.2}$$

$$\text{kinematic b.c.} \qquad \frac{\partial H}{\partial t} \;=\; u_3 \;-\; u_1\frac{\partial H}{\partial x} \;-\; u_2\frac{\partial H}{\partial y} \tag{2.3}$$

$$\text{normal dynamic b.c.} \qquad P_{s+} \;-\; P_{s-} \;=\; -2\mu\, e_{ij}n_in_j \;+\; \gamma\left(R_1^{-1} \;+\; R_2^{-2}\right) \tag{2.4}$$

$$\text{tangential dynamic b.c.} \qquad e_{ij}t_in_j \;=\; 0 \tag{2.5}$$

where $u_i \;:\; i = 1,3$ are the Cartesian (physical space) components of velocity, $x_i \;:\; i = 1,3$ are the Cartesian coordinates, D/Dt is a material derivative, g_i is gravity vector, P is the total pressure, ρ is the density, μ is the dynamic viscosity, δ_{ij} is the Kronecker delta, H is the surface height above a reference baseline, the subscripts $s+$ and $s-$ indicate the pressure on the upper and lower sides of the free surface, n_i and t_i are the unit normal and tangent vectors, γ is the surface tension coefficient, R_1 and R_2 are the principal radii of curvature of the surface, and e_{ij} is the rate-of-strain tensor defined as:

$$e_{ij} \;\equiv\; \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} \;+\; \frac{\partial u_j}{\partial x_i}\right) \tag{2.6}$$

For purposes of non-dimensionalization, we can decompose the pressure into a static pressure and a dynamic pressure:

$$P \;\equiv\; \rho\, p_d \;+\; \rho|\mathbf{g}|\Delta h \tag{2.7}$$

where Δh is the static head at a point in the fluid, measured as the distance from the point to the free surface vertically over the point. For incompressible flow, the

momentum and continuity equations become:

$$\text{momentum} \qquad \frac{Du_i}{Dt} \;=\; g_i \;-\; |\mathbf{g}|\frac{\partial}{\partial x_i}(\Delta h) \;-\; \frac{\partial p_d}{\partial x_i} \;+\; \frac{\partial}{\partial x_j}\left\{2\nu e_{ij}\right\} \tag{2.8}$$

$$\text{continuity} \qquad \frac{\partial u_j}{\partial x_j} \;=\; 0 \tag{2.9}$$

where ν is the kinematic viscosity.

If the total pressure (P) outside the free surface is equal to a constant (zero is convenient), then the normal component of the dynamic boundary condition, equation (2.4) can be written as a condition on the dynamic pressure (p_d):

$$(p_d)_s \;=\; 2\nu\, e_{ij} n_i n_j \;-\; \frac{\gamma}{\rho}\left(R_1^{-1} \;+\; R_2^{-2}\right) \tag{2.10}$$

where the subscript s indicates the dynamic pressure on the water side of the surface. The static head (Δh) at the surface is (by definition) exactly zero.

For a turbulent channel flow with a progressive surface wave, there are three primary components to the fluid motion: (1) the turbulence produced by the shear at the bottom boundary, (2) the convection in the streamwise direction, and (3) the oscillations produced by the surface waves. This makes a general non-dimensionalization difficult since the choice of the correct scales depends on *a priori* assumption of which scales are the most important. To complicate matters, we also have four length scales for the flow: (1) a turbulent length scale, (2) a convective length scale, (3) a wave-induced orbital length scale of particle motion, and (4) a length scale of the largest surface wave. Finally, we have time scales based upon (1) the turbulent eddy turnover time, (2) the convective flow-through time, and (3) the wave period.

2.2.1 Non-dimensionalization of the momentum equation

Scaling based on turbulence

If we assume that the turbulent scales of motion are dominant in all directions, then the appropriate non-dimensionalization of the momentum equation uses the turbulent velocity scale ($\mathcal{U}_\tau$). In the simulations we have conducted, the maximum flow depth

is such that free-surface is in the log-law region of the bottom boundary layer, so that appropriate turbulent length scale is the flow depth ($\mathcal{D}$), and the time scale is $\mathcal{D}/\mathcal{U}_\tau$. The dynamic pressure is non-dimensionalized with the turbulent velocity scale. The body force (gravity) and the gravity coefficient of the static pressure term are non-dimensionalized by the magnitude of gravity. Viscosity and the rate-of strain tensor are non-dimensionalized by the turbulent length and velocity scales. Using a (*) notation to indicate non-dimensional variables, we have

$$u_i = \mathcal{U}_\tau u_i^* \tag{2.11}$$

$$t = \frac{\mathcal{D}}{\mathcal{U}_\tau} t^* \tag{2.12}$$

$$g_i = \mathcal{G} g_i^* \tag{2.13}$$

$$\Delta h = \mathcal{D} \Delta h^* \tag{2.14}$$

$$x_i = \mathcal{D} x_i^* \tag{2.15}$$

$$p_d = \mathcal{U}_\tau^2 p_d^* \tag{2.16}$$

$$\nu = \mathcal{U}_\tau \mathcal{D} \nu^* \tag{2.17}$$

$$e_{ij} = \frac{\mathcal{U}_\tau}{\mathcal{D}} e_{ij}^* \tag{2.18}$$

This results in the non-dimensionalized momentum equation:

$$\frac{Du_i^*}{Dt^*} = \frac{\mathcal{G}\mathcal{D}}{\mathcal{U}_\tau^2} \left\{ g_i^* - \frac{\partial}{\partial x_i^*} (\Delta h^*) \right\} - \frac{\partial p_d^*}{\partial x_i^*} + \frac{\partial}{\partial x_j^*} \left\{ 2\nu^* e_{ij}^* \right\} \tag{2.19}$$

We can define the turbulent Reynolds number and the bulk Froude number as:

$$Re_\tau \equiv \frac{1}{\nu^*} \tag{2.20}$$

$$Fr_b \equiv \frac{\mathcal{U}_c}{\sqrt{\mathcal{G}\mathcal{D}}} \tag{2.21}$$

where $\mathcal{U}_c$ is a convective velocity scale. If our axes are aligned such that gravity is in the negative direction along the x_3 axis, then $g_i^* = -\delta_{i3}$ and the non-dimensionalized momentum equation can be written as:

$$\frac{Du_i^*}{Dt^*} = \frac{1}{Fr_b^2} \left(\frac{\mathcal{U}_c}{\mathcal{U}_\tau} \right)^2 \left\{ -\delta_{i3} - \frac{\partial}{\partial x_i^*} (\Delta h^*) \right\} - \frac{\partial p_d^*}{\partial x_i^*} + \frac{\partial}{\partial x_j^*} \left\{ \frac{2}{Re_\tau} e_{ij}^* \right\} \tag{2.22}$$

If $i = 3$ then

$$\frac{\partial}{\partial x_3^*}\left(\Delta h^*\right) \;=\; -1 \tag{2.23}$$

and the balance between gravitational and hydrostatic forces is exactly zero. If $i = 1$ or $i = 2$, the gravitational force is zero, and the gradient of the hydrostatic force has to be balanced by convective and turbulent motion. Note that the gradient of the hydrostatic pressure in the x_1 and x_2 directions should be of the order of the surface wave steepness, not of order one as would be indicated by this non-dimensionalization. This indicates our initial assumption of the appropriate non-dimensional scales is incorrect. However, we can obtain some insight into the effect of the the hydrostatic pressure force from this non-dimensionalization. For a subcritical flow, the Froude number is less than one, and the convective velocity scale is generally at least an order of magnitude larger than the turbulent velocity scale. Thus, the coefficient multiplying the gradient of the hydrostatic pressure is (at minimum) in a range of 10^2 to 10^3. This implies that a small perturbation of the gradient of the hydrostatic pressure causes a rapid change in the velocity field. A large perturbation (such as caused by a finite-amplitude wave) results in this term dominating the equations, confirming that this is not an appropriate non-dimensionalization for a flow with a significant surface wave.

Scaling based on surface wave

If we begin with the assumption that the wave-induced effects dominate the velocity field, then our non-dimensionalization must proceed in a different, more complicated manner. Since the irrotational velocity field produced by a progressive wave moving in the streamwise (x) direction is two-dimensional, the scaling in the spanwise direction should proceed as before, based on the turbulent scales ($\mathcal{U}_\tau$) and ($\mathcal{D}$). In the streamwise direction, the length scale is the wavelength (λ), while in the vertical direction the length scale is the flow depth ($\mathcal{D}$). The streamwise and vertical velocity scale of the wave-induced motions ($\mathcal{U}_w$) is the initial wave amplitude ($\mathcal{A}$) divided by the wave period. The time scale is the wave period (the initial wave amplitude divided by the wave velocity scale). The different scales require the momentum equation to be written as three separate equations rather than a single tensor equation. It will be convenient from the outset to define gravity as aligned with the x_3 axis in the

negative direction. The non-dimensional variables can be defined such that:

$$u_1 = \mathcal{U}_w u_1^* \tag{2.24}$$
$$u_2 = \mathcal{U}_\tau u_2^* \tag{2.25}$$
$$u_3 = \mathcal{U}_w u_3^* \tag{2.26}$$
$$t = \frac{\mathcal{A}}{\mathcal{U}_w} t^* \tag{2.27}$$
$$g_1 = 0 \tag{2.28}$$
$$g_2 = 0 \tag{2.29}$$
$$g_3 = -\mathcal{G} \tag{2.30}$$
$$|\mathrm{g}| = \mathcal{G} \tag{2.31}$$
$$\Delta h = \mathcal{A}\Delta h^* \tag{2.32}$$
$$x_1 = \lambda x_1^* \tag{2.33}$$
$$x_2 = \mathcal{D} x_2^* \tag{2.34}$$
$$x_3 = \mathcal{D} x_3^* \tag{2.35}$$
$$p_d = \mathcal{U}_w^2 p_d^* \tag{2.36}$$
$$\nu = \mathcal{U}_w \mathcal{D} \nu^* \tag{2.37}$$
$$\gamma = \rho\, \mathcal{U}_w^2 \lambda \gamma^* \tag{2.38}$$
$$H = \mathcal{A}\, H^* \tag{2.39}$$

We will leave the non-dimensionalization of the rate-of strain tensor for the next section, and write the *partially* non-dimensionalized momentum equations as:

$$\frac{Du_1^*}{Dt^*} = - \left(\frac{\mathcal{G}\mathcal{A}^2}{\mathcal{U}_w^2\,\lambda}\right)\frac{\partial}{\partial x_1^*}(\Delta h^*) - \left(\frac{\mathcal{A}}{\lambda}\right)\frac{\partial p_d^*}{\partial x_1^*} + \left(\frac{\mathcal{A}\mathcal{D}}{\lambda\mathcal{U}_w}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^* e_{11}\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^* e_{12}\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^* e_{13}\right\} \tag{2.40}$$

$$\frac{Du_2^*}{Dt^*} = - \left(\frac{\mathcal{G}\mathcal{A}^2}{\mathcal{U}_w\mathcal{U}_\tau\mathcal{D}}\right)\frac{\partial}{\partial x_2^*}(\Delta h^*) - \left(\frac{\mathcal{A}}{\mathcal{D}}\frac{\mathcal{U}_w}{\mathcal{U}_\tau}\right)\frac{\partial p_d^*}{\partial x_2^*} + \left(\frac{\mathcal{A}\mathcal{D}}{\lambda\mathcal{U}_\tau}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^* e_{12}\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_\tau}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^* e_{22}\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_\tau}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^* e_{23}\right\} \tag{2.41}$$

$$\frac{Du_3^*}{Dt^*} = - \left(\frac{\mathcal{A}}{\mathcal{D}}\right)\frac{\partial p_d^*}{\partial x_3^*} + \left(\frac{\mathcal{A}\mathcal{D}}{\lambda \mathcal{U}_w}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^* e_{13}\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^* e_{23}\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^* e_{33}\right\} \tag{2.42}$$

Non-dimensionalization of e_{ij} for waves

The rate-of-strain tensor is the most problematic term to correctly non-dimensionalize in a flow with waves and turbulence. To do so, it will be convenient to decompose the rate-of-strain into convective ($\bar{e}_{ij}$), wave-induced ($\tilde{e}_{ij}$), and turbulent ($\hat{e}_{ij}$) components, so that

$$e_{ij} = \bar{e}_{ij} + \tilde{e}_{ij} + \hat{e}_{ij} \tag{2.43}$$

where the convective velocities are similarly decomposed:

$$\bar{e}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \tag{2.44}$$

and similar relations exist for the wave-induced and turbulent rate-of-strain components. We define our decomposition such that (1) the convective rate-of-strain is zero except for the $\bar{e}_{13}$ component, (2) the wave-induced rate-of-strain is zero for any spanwise component ($i = 2$ or $j = 2$), and (3) the turbulent rate-of-strain has contributions in all dimensions.

This approach is not practical for implementation in a simulation method. However, it allows us to obtain a better understanding of the physics of the flow by providing non-dimensionalization of the separate components of the rate-of-strain using different scales. For the convective term, the appropriate scales are the convective velocity ($\mathcal{U}_c$), and the flow depth ($\mathcal{D}$) so that:

$$\bar{e}_{13} = \frac{\mathcal{U}_c}{\mathcal{D}}\,\bar{e}_{13}^* \tag{2.45}$$

For the turbulent rate-of strain, the appropriate scales are the shear velocity ($\mathcal{U}_\tau$) and the flow depth ($\mathcal{D}$):

$$\hat{e}_{ij} = \frac{\mathcal{U}_\tau}{\mathcal{D}}\,\hat{e}_{ij}^* \tag{2.46}$$

For the wave-induced components, the correct non-dimensionalization depends upon the particular (i,j) term under consideration. The wave-induced length scale depends upon direction; the appropriate vertical length scale is the channel depth ($\mathcal{D}$), while the horizontal length scale is the wavelength of the surface wave (λ). So we find that

$$\tilde{e}_{11} = \frac{\mathcal{U}_w}{\lambda}\,\tilde{e}^*_{11} \tag{2.47}$$

$$\tilde{e}_{33} = \frac{\mathcal{U}_w}{\mathcal{D}}\,\tilde{e}^*_{33} \tag{2.48}$$

$$\tilde{e}_{13} = \frac{1}{2}\left(\frac{\mathcal{U}_w}{\mathcal{D}}\,\frac{\partial \tilde{u}^*_1}{\partial x^*_3} + \frac{\mathcal{U}_w}{\lambda}\,\frac{\partial \tilde{u}^*_3}{\partial x^*_1}\right) \tag{2.49}$$

where (as before)

$$x_1 \equiv \lambda\, x^*_1 \tag{2.50}$$

$$x_3 \equiv \mathcal{D}\, x^*_3 \tag{2.51}$$

We now can write the rate-of-strain decomposition as:

$$e_{11} = \frac{\mathcal{U}_w}{\lambda}\tilde{e}^*_{11} + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}^*_{11} \tag{2.52}$$

$$e_{22} = \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}^*_{22} \tag{2.53}$$

$$e_{33} = \frac{\mathcal{U}_w}{\mathcal{D}}\tilde{e}^*_{33} + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}^*_{33} \tag{2.54}$$

$$e_{12} = \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}^*_{12} \tag{2.55}$$

$$e_{13} = \frac{\mathcal{U}_c}{\mathcal{D}}\bar{e}^*_{13} + \frac{1}{2}\frac{\mathcal{U}_w}{\mathcal{D}}\left(\frac{\partial \tilde{u}^*_1}{\partial x^*_3} + \frac{\mathcal{D}}{\lambda}\frac{\partial \tilde{u}^*_3}{\partial x^*_1}\right) + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}^*_{13} \tag{2.56}$$

$$e_{23} = \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}^*_{23} \tag{2.57}$$

Non-dimensional momentum equations

Substituting the relations for the non-dimensional rates-of-strain into equations (2.40) through (2.42) gives:

$$\begin{aligned}\frac{Du_1^*}{Dt^*} &= -\left(\frac{\mathcal{G}\mathcal{A}^2}{\mathcal{U}_w^2\,\lambda}\right)\frac{\partial}{\partial x_1^*}\left(\Delta h^*\right) - \left(\frac{\mathcal{A}}{\lambda}\right)\frac{\partial p_d^*}{\partial x_1^*} \\ &+ \left(\frac{\mathcal{A}\mathcal{D}}{\lambda\mathcal{U}_w}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^*\left(\frac{\mathcal{U}_w}{\lambda}\bar{e}_{11}^* + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{11}^*\right)\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^*\frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{12}^*\right\} \\ &+ \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^*\left(\frac{\mathcal{U}_c}{\mathcal{D}}\bar{e}_{13}^* + \frac{1}{2}\frac{\mathcal{U}_w}{\mathcal{D}}\left[\frac{\partial\tilde{u}_1^*}{\partial\tilde{x}_3^*} + \frac{\mathcal{D}}{\lambda}\frac{\partial\tilde{u}_3^*}{\partial\tilde{x}_1^*}\right] + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{13}^*\right)\right\}\end{aligned} \tag{2.58}$$

$$\begin{aligned}\frac{Du_2^*}{Dt^*} &= -\left(\frac{\mathcal{G}\mathcal{A}^2}{\mathcal{U}_w\mathcal{U}_\tau\mathcal{D}}\right)\frac{\partial}{\partial x_2^*}\left(\Delta h^*\right) - \left(\frac{\mathcal{A}\mathcal{U}_w}{\mathcal{D}\mathcal{U}_\tau}\right)\frac{\partial p_d^*}{\partial x_2^*} \\ &+ \left(\frac{\mathcal{A}\mathcal{D}}{\lambda\mathcal{U}_\tau}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^*\frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{12}^*\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_\tau}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^*\frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{22}^*\right\} \\ &+ \left(\frac{\mathcal{A}}{\mathcal{U}_\tau}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^*\frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{23}^*\right\}\end{aligned} \tag{2.59}$$

$$\begin{aligned}\frac{Du_3^*}{Dt^*} &= -\left(\frac{\mathcal{A}}{\mathcal{D}}\right)\frac{\partial p_d^*}{\partial x_3^*} \\ &+ \left(\frac{\mathcal{A}\mathcal{D}}{\lambda\mathcal{U}_w}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^*\left(\frac{\mathcal{U}_c}{\mathcal{D}}\bar{e}_{13}^* + \frac{1}{2}\frac{\mathcal{U}_w}{\mathcal{D}}\left[\frac{\partial\tilde{u}_1^*}{\partial x_3^*} + \frac{\mathcal{D}}{\lambda}\frac{\partial\tilde{u}_3^*}{\partial x_1^*}\right] + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{13}^*\right)\right\} \\ &+ \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^*\frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{23}^*\right\} + \left(\frac{\mathcal{A}}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^*\left(\frac{\mathcal{U}_w}{\mathcal{D}}\tilde{e}_{33}^* + \frac{\mathcal{U}_\tau}{\mathcal{D}}\hat{e}_{33}^*\right)\right\}\end{aligned} \tag{2.60}$$

Cleaning these up a little gives:

$$\begin{aligned}\frac{Du_1^*}{Dt^*} &= -\left(\frac{\mathcal{G}\mathcal{A}^2}{\mathcal{U}_w^2\,\lambda}\right)\frac{\partial}{\partial x_1^*}\left(\Delta h^*\right) - \left(\frac{\mathcal{A}}{\lambda}\right)\frac{\partial p_d^*}{\partial x_1^*} \\ &+ \left(\frac{\mathcal{A}}{\lambda}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^*\left(\frac{\mathcal{D}}{\lambda}\tilde{e}_{11}^* + \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{11}^*\right)\right\} + \left(\frac{\mathcal{A}}{\mathcal{D}}\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^*\hat{e}_{12}^*\right\} \\ &+ \left(\frac{\mathcal{A}}{\mathcal{D}}\frac{\mathcal{U}_c}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^*\left(\bar{e}_{13}^* + \frac{1}{2}\frac{\mathcal{U}_w}{\mathcal{U}_c}\left[\frac{\partial\tilde{u}_1^*}{\partial\tilde{x}_3^*} + \frac{\mathcal{D}}{\lambda}\frac{\partial\tilde{u}_3^*}{\partial\tilde{x}_1^*}\right] + \frac{\mathcal{U}_\tau}{\mathcal{U}_c}\hat{e}_{13}^*\right)\right\}\end{aligned} \tag{2.61}$$

$$\frac{Du_2^*}{Dt^*} = -\left(\frac{\mathcal{G}\mathcal{A}^2}{\mathcal{U}_w\mathcal{U}_\tau\mathcal{D}}\right)\frac{\partial}{\partial x_2^*}(\Delta h^*) - \left(\frac{\mathcal{A}}{\mathcal{D}}\frac{\mathcal{U}_w}{\mathcal{U}_\tau}\right)\frac{\partial p_d^*}{\partial x_2^*} + \left(\frac{\mathcal{A}}{\lambda}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^*\hat{e}_{12}^*\right\} + \left(\frac{\mathcal{A}}{\mathcal{D}}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^*\hat{e}_{22}^*\right\} + \left(\frac{\mathcal{A}}{\mathcal{D}}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^*\hat{e}_{23}^*\right\} \tag{2.62}$$

$$\frac{Du_3^*}{Dt^*} = -\left(\frac{\mathcal{A}}{\mathcal{D}}\right)\frac{\partial p_d^*}{\partial x_3^*} + \left(\frac{\mathcal{A}}{\lambda}\frac{\mathcal{U}_c}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_1^*}\left\{2\nu^*\left(\bar{e}_{13}^* + \frac{1}{2}\frac{\mathcal{U}_w}{\mathcal{U}_c}\left[\frac{\partial\tilde{u}_1^*}{\partial x_3^*} + \frac{\mathcal{D}}{\lambda}\frac{\partial\tilde{u}_3^*}{\partial x_1^*}\right] + \frac{\mathcal{U}_\tau}{\mathcal{U}_c}\hat{e}_{13}^*\right)\right\} + \left(\frac{\mathcal{A}}{\mathcal{D}}\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\right)\frac{\partial}{\partial x_2^*}\left\{2\nu^*\hat{e}_{23}^*\right\} + \left(\frac{\mathcal{A}}{\mathcal{D}}\right)\frac{\partial}{\partial x_3^*}\left\{2\nu^*\left(\tilde{e}_{33}^* + \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{33}^*\right)\right\} \tag{2.63}$$

2.2.2 Non-dimensional continuity and boundary conditions

In a fashion similar to the preceding section, we find the incompressible non-dimensional continuity equation can be derived from equation (2.9) as:

$$\frac{\partial u_1^*}{\partial x_1^*} + \frac{\lambda}{\mathcal{D}}\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\frac{\partial u_2^*}{\partial x_2^*} + \frac{\lambda}{\mathcal{D}}\frac{\partial u_3^*}{\partial x_3^*} = 0 \tag{2.64}$$

From equation (2.3), the non-dimensional kinematic boundary condition is:

$$\frac{\partial H^*}{\partial t^*} = u_3^* - \frac{\mathcal{A}}{\lambda}u_1^*\frac{\partial H^*}{\partial x_1^*} - \frac{\mathcal{A}}{\mathcal{D}}\frac{\mathcal{U}_\tau}{\mathcal{U}_w}u_2^*\frac{\partial H^*}{\partial x_2^*} \tag{2.65}$$

From equation (2.10) and the strain rates in equations (2.52) to (2.57), the normal component of the dynamic boundary condition becomes:

$$(p_d^*)_s = 4\nu^*\Bigg\{ n_1n_1\left(\frac{\mathcal{D}}{\lambda}\tilde{e}_{11}^* + \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{11}^*\right) + n_2n_2\left(\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{22}^*\right) + n_3n_3\left(\tilde{e}_{33}^* + \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{33}^*\right) + n_1n_2\left(\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{12}^*\right) + n_1n_3\left(\frac{\mathcal{U}_c}{\mathcal{U}_w}\bar{e}_{13}^* + \frac{1}{2}\left(\frac{\partial\tilde{u}_1^*}{\partial x_3^*} + \frac{\mathcal{D}}{\lambda}\frac{\partial\tilde{u}_3^*}{\partial x_1^*}\right) + \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{13}^*\right) + n_2n_3\left(\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}_{23}^*\right)\Bigg\} - \gamma^*\left(\frac{1}{R_1^*} + \frac{1}{R_2^*}\right) \tag{2.66}$$

From equation (2.5) and the strain rates in equations (2.52) to (2.57), the tangential component of the dynamic boundary condition is:

$$\begin{aligned} & 2\,t_1 n_1 \left(\frac{\mathcal{D}}{\lambda}\tilde{e}^*_{11} \;+\; \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}^*_{11}\right) \;+\; 2\,t_2 n_2 \left(\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}^*_{22}\right) \;+\; 2\,t_3 n_3 \left(\tilde{e}^*_{33} \;+\; \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}^*_{33}\right) \\ +\;\; & (t_1 n_2 \;+\; t_2 n_1)\left(\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}^*_{12}\right) \;+\; (t_2 n_3 \;+\; t_3 n_2)\left(\frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}^*_{23}\right) \\ +\;\; & (t_1 n_3 \;+\; t_3 n_1)\left(\frac{\mathcal{U}_c}{\mathcal{U}_w}\bar{e}^*_{13} \;+\; \frac{1}{2}\left(\frac{\partial \tilde{u}^*_1}{\partial x^*_3} \;+\; \frac{\mathcal{D}}{\lambda}\frac{\partial \tilde{u}^*_3}{\partial x^*_1}\right) \;+\; \frac{\mathcal{U}_\tau}{\mathcal{U}_w}\hat{e}^*_{13}\right) \;=\; 0 \end{aligned} \tag{2.67}$$

2.2.3 Analysis of non-dimensionalization

It is apparent that the relative scales of the mean flow, turbulence, surface wave, and domain size determine what terms in the governing equations are important. To provide some clarification, we can define six non-dimensional parameters (from the analysis thus far, these would seem to be independent):

$$\epsilon \;\equiv\; \frac{\mathcal{A}}{\lambda} \tag{2.68}$$

$$\mathcal{S} \;\equiv\; \frac{\lambda}{\mathcal{D}} \tag{2.69}$$

$$\mathcal{R}_u \;\equiv\; \frac{\mathcal{U}_\tau}{\mathcal{U}_w} \tag{2.70}$$

$$Re_w \;\equiv\; \frac{1}{\nu^*} \;=\; \frac{\mathcal{U}_w \mathcal{D}}{\nu} \tag{2.71}$$

$$Fr_w \;\equiv\; \frac{\mathcal{U}_w}{\sqrt{\mathcal{G}\,\mathcal{A}}} \tag{2.72}$$

$$We_w \;\equiv\; \frac{1}{\gamma^*} \;=\; \frac{\rho\,\mathcal{U}_w\,\lambda}{\gamma} \tag{2.73}$$

Note that ϵ is the wave ak divided by 2π. The parameter $\mathcal{S}$ is a measure of the shallowness of the wave (which approaches zero asymptotically for deep water waves). $\mathcal{R}_u$ is a relative measure of the velocity scales for turbulence and the wave-induced motions. Re_w, Fr_w and We_w are Reynolds, Froude, and Weber numbers based on the wave-induced velocity field. The convective flow velocity in an open-channel flow is dependent on the turbulent shear velocity, so it will be convenient to write:

$$\frac{\mathcal{U}_c}{\mathcal{U}_w} \;=\; \alpha\,\mathcal{R}_u \tag{2.74}$$

where α is a scaling constant that is a function of the flow depth and the turbulent shear velocity.

If the waves can be approximated by second-order Stokes theory then the non-dimensional parameters in definitions (2.68) to (2.73) are not all independent. Gravity waves on the surface can be expected to have a dispersion relation of

$$\sigma^2 \;=\; gk \tanh kD \tag{2.75}$$

where σ is the wave frequency, g is gravity, k is the wave number, and D is the domain depth. Noting that σ is 2π divided by the wave period, and from the definition of the wave-induced velocity scale as the wave amplitude divided by the wave period:

$$\sigma \;=\; \frac{2\pi}{T} \;=\; 2\pi \frac{\mathcal{U}_w}{\mathcal{A}} \tag{2.76}$$

The wave number is defined as $2\pi/\lambda$, so non-dimensional form of the dispersion relation is:

$$2\pi \frac{\mathcal{U}_w^2}{\mathcal{A}^2} \;=\; \frac{\mathcal{G}}{\lambda} \tanh\left(2\pi \frac{\mathcal{D}}{\lambda}\right) \tag{2.77}$$

We can write this as

$$\mathcal{U}_w \;=\; \mathcal{A}\sqrt{\frac{\mathcal{G}}{2\pi\lambda} \tanh\left(2\pi\mathcal{S}^{-1}\right)} \tag{2.78}$$

The Froude number, equation (2.72), can then be written as:

$$Fr_w \;=\; \sqrt{\frac{\epsilon}{2\pi} \tanh\left(2\pi\mathcal{S}^{-1}\right)} \tag{2.79}$$

The non-dimensional equation set consists of:
(1) x momentum equation:

$$\begin{aligned}
\frac{Du_1^*}{Dt^*} \;=\;& - \frac{2\pi}{\tanh\left(2\pi\mathcal{S}^{-1}\right)} \frac{\partial}{\partial x_1^*}\left(\Delta h^*\right) \;-\; \epsilon \frac{\partial p_d^*}{\partial x_1^*} \\
& + \epsilon \frac{\partial}{\partial x_1^*}\left\{ \frac{2}{Re_w}\left(\frac{1}{\mathcal{S}}\hat{e}_{11}^* \;+\; \mathcal{R}_u \hat{e}_{11}^*\right)\right\} \\
& + \epsilon\,\mathcal{S}\,\mathcal{R}_u \frac{\partial}{\partial x_2^*}\left\{ \frac{2}{Re_w}\hat{e}_{12}^*\right\} \\
& + \epsilon\,\mathcal{S} \frac{\partial}{\partial x_3^*}\left\{ \frac{2}{Re_w}\left(\alpha\,\mathcal{R}_u \bar{e}_{13}^* \;+\; \frac{1}{2}\left[\frac{\partial \tilde{u}_1^*}{\partial \tilde{x}_3^*} \;+\; \frac{1}{\mathcal{S}}\frac{\partial \tilde{u}_3^*}{\partial \tilde{x}_1^*}\right] \;+\; \mathcal{R}_u \hat{e}_{13}^*\right)\right\}
\end{aligned} \tag{2.80}$$

(2) y momentum equation:

$$\begin{aligned}\frac{Du_2^*}{Dt^*} &= -\frac{2\pi,\mathcal{S}}{\mathcal{R}_u\,\tanh\left(2\pi\,\mathcal{S}^{-1}\right)}\frac{\partial}{\partial x_2^*}\left(\Delta h^*\right) - \frac{\epsilon\,\mathcal{S}}{\mathcal{R}_u}\frac{\partial p_d^*}{\partial x_2^*} \\ &+ \epsilon\frac{\partial}{\partial x_1^*}\left\{\frac{2}{Re_w}\hat{e}_{12}^*\right\} + \epsilon\,\mathcal{S}\frac{\partial}{\partial x_2^*}\left\{\frac{2}{Re_w}\hat{e}_{22}^*\right\} + \epsilon\,\mathcal{S}\frac{\partial}{\partial x_3^*}\left\{\frac{2}{Re_w}\hat{e}_{23}^*\right\}\end{aligned} \tag{2.81}$$

(3) z momentum equation:

$$\begin{aligned}\frac{Du_3^*}{Dt^*} &= -\epsilon\,\mathcal{S}\frac{\partial p_d^*}{\partial x_3^*} \\ &+ \epsilon\frac{\partial}{\partial x_1^*}\left\{\frac{2}{Re_w}\left(\alpha\,\mathcal{R}_u\,\bar{e}_{13}^* + \frac{1}{2}\left[\frac{\partial\tilde{u}_1^*}{\partial x_3^*} + \frac{1}{\mathcal{S}}\frac{\partial\tilde{u}_3^*}{\partial x_1^*}\right] + \mathcal{R}_u\,\hat{e}_{13}^*\right)\right\} \\ &+ \epsilon\,\mathcal{S}\,\mathcal{R}_u\frac{\partial}{\partial x_2^*}\left\{\frac{2}{Re_w}\hat{e}_{23}^*\right\} \\ &+ \epsilon\,\mathcal{S}\frac{\partial}{\partial x_3^*}\left\{2\nu^*\left(\tilde{e}_{33}^* + \mathcal{R}_u\,\hat{e}_{33}^*\right)\right\}\end{aligned} \tag{2.82}$$

(4) continuity:

$$\frac{\partial u_1^*}{\partial x_1^*} + \mathcal{S}\,\mathcal{R}_u\frac{\partial u_2^*}{\partial x_2^*} + \mathcal{S}\frac{\partial u_3^*}{\partial x_3^*} = 0 \tag{2.83}$$

(5) kinematic boundary condition:

$$\frac{\partial H^*}{\partial t^*} = u_3^* - \epsilon\,u_1^*\frac{\partial H^*}{\partial x_1^*} - \epsilon\,\mathcal{S}\,\mathcal{R}_u\,u_2^*\frac{\partial H^*}{\partial x_2^*} \tag{2.84}$$

(6) normal component of the dynamic boundary condition:

$$\begin{aligned}\left(p_d^*\right)_s &= \frac{4}{Re_w}\Bigg\{ n_1 n_1\left(\frac{1}{\mathcal{S}}\tilde{e}_{11}^* + \mathcal{R}_u\,\hat{e}_{11}^*\right) \\ &+ n_2 n_2\left(\mathcal{R}_u\,\hat{e}_{22}^*\right) \\ &+ n_3 n_3\left(\tilde{e}_{33}^* + \mathcal{R}_u\,\hat{e}_{33}^*\right) \\ &+ n_1 n_2\left(\mathcal{R}_u\,\hat{e}_{12}^*\right) \\ &+ n_1 n_3\left(\alpha\,\mathcal{R}_u\,\bar{e}_{13}^* + \frac{1}{2}\left[\frac{\partial\tilde{u}_1^*}{\partial x_3^*} + \frac{1}{\mathcal{S}}\frac{\partial\tilde{u}_3^*}{\partial x_1^*}\right] + \mathcal{R}_u\hat{e}_{13}^*\right) \\ &+ n_2 n_3\left(\mathcal{R}_u\hat{e}_{23}^*\right)\Bigg\} - \gamma^*\left\{\frac{1}{R_1^*} + \frac{1}{R_2^*}\right\}\end{aligned} \tag{2.85}$$

(7) tangential component of the dynamic boundary condition:

$$\begin{aligned} & 2\,t_1 n_1 \left(\frac{1}{\mathcal{S}}\tilde{e}^*_{11} \;+\; \mathcal{R}_u\,\hat{e}^*_{11}\right) \;+\; 2\,t_2 n_2\,(\mathcal{R}_u\,\hat{e}^*_{22}) \;+\; 2\,t_3 n_3\,(\tilde{e}^*_{33} \;+\; \mathcal{R}_u\,\hat{e}^*_{33}) \\ +\;\; & (t_1 n_2 \;+\; t_2 n_1)\,(\mathcal{R}_u\,\hat{e}^*_{12}) \;+\; (t_2 n_3 \;+\; t_3 n_2)\,(\mathcal{R}_u \hat{e}^*_{23}) \\ +\;\; & (t_1 n_3 \;+\; t_3 n_1)\left(\alpha \mathcal{R}_u\,\bar{e}^*_{13} \;+\; \frac{1}{2}\left[\frac{\partial \tilde{u}^*_1}{\partial x^*_3} \;+\; \frac{1}{\mathcal{S}}\frac{\partial \tilde{u}^*_3}{\partial x^*_1}\right] \;+\; \mathcal{R}_u \hat{e}^*_{13}\right) \;=\; 0 \end{aligned} \tag{2.86}$$

This non-dimensionalization provides a wealth of information about the relationships among the turbulence, waves, and flow domain. With the exception of the hydrostatic pressure term, all the terms on the right hand side of the momentum equations are of the order of the wave slope (ϵ) and can be neglected for small amplitude waves. Note that none of the terms is order ϵ^2, so all terms must be retained for finite-amplitude waves. Some simplification can be obtained by considering only shallow-water waves (which we could define here as $S \gg 1$). For deep-water waves ($S \ll 1$) or when the turbulent velocity scale is small compared to the wave velocity scale ($\mathcal{R}_u \ll 1$) the number of terms that are needed is significantly reduced[1].

One of the most complicated cases is examined in the results of this dissertation (chapters 5 through-7):

$$\mathcal{R}_u \;=\; O\left(10^{-2}\right) \tag{2.87}$$

$$\mathcal{S} \;=\; O\left(1\right) \tag{2.88}$$

$$Re_w \;=\; O\left(10^{2}\right) \tag{2.89}$$

$$\epsilon \;=\; O\left(10^{-1}\right) \tag{2.90}$$

$$\alpha \;=\; O\left(10\right) \tag{2.91}$$

$$\gamma^* \;=\; 0 \tag{2.92}$$

The only terms neglected for this case are the surface tension terms.

[1]Consideration of boundary layer scaling in the near-surface region may mandate the retention of some of the viscous terms that could otherwise be eliminated from deep-water or shallow-water waves.

2.3 Filtered, curvilinear Navier-Stokes equations

The non-dimensional equations presented in section 2.2 provide insight into the physics of the important terms in the interactions between waves and a turbulent flow. However, the number of non-dimensional parameters (due to the multiple scales in the flow) makes numerical implementation impractical for a simulation method that is intended to be adaptable for use with other types of flows. One of the advantages of Zang's (1993) numerical method and code is its ability to be used or adapted to almost any form of interior flow. In a similar fashion, we want our method to be adaptable to free-surface flows which do not non-dimensionalize in accordance with the previous section. For example, in Appendix C we provide a simulation of three-dimensional sloshing of a free surface along a diagonal axis of the domain. This is a case with finite-amplitude wave motion in both x and y directions, which significantly changes the dimensional analysis of the flow. In general, for an adaptable numerical method, it is more practical to modify the the flow domain size, viscosity, and wave amplitude rather than adjusting non-dimensional scale parameters.

In this section, we derive a set of governing equations that is suitable for implementation in our numerical method and are applicable to any free-surface flow with gravity and capillary waves. The development of the equations for the numerical method starts from our initial statement of the governing equations in section 2.2.

2.3.1 Cartesian space N-S equations

In the following analysis, the x_1 (or x) coordinate is the streamwise direction, the x_2 (or y) coordinate is in the spanwise direction, and the x_3 (or z) coordinate is normal to the bottom boundary (a plane surface). Note that we are being careful not to define the z coordinate as vertical since an open-channel flow is easier to analyze if the coordinate system is rotated to match the slope of the channel. If we confine the slope of the channel to the x direction, then the gravity vector $\mathbf{g}$ has components only in the x and z directions. In the literature of open-channel flows, the z component of gravity is generally referred to as g' and is equal to $(-|\mathbf{g}|\,cos\,\theta)$, where θ is the slope of the channel. The body force in the streamwise direction is $\rho\,|\mathbf{g}|\,sin\,\theta$ which we can

redefine as a gradient of a modified pressure Ψ that drives the open-channel flow:

$$\rho \frac{d\Psi}{dx_1} = \rho \, |\mathbf{g}| \, sin\,\theta \tag{2.93}$$

This allows Ψ at any point to be defined by integrating to obtain:

$$\Psi = |\mathbf{g}| \, x_1 sin\,\theta \tag{2.94}$$

which is useful when dealing with curvilinear grids that are not aligned with the Cartesian coordinate system.

The g' component of the gravitational body force is included in the pressure term by defining the modified pressure as:

$$p \equiv \frac{P}{\rho} + g' \, x_3 \tag{2.95}$$

where x_3 is the vertical Cartesian coordinate and can be measured from any baseline. For numerical implementation, it is convenient to use the bottom of the domain as the $x_3 = 0$ baseline.

Using equations (2.1), (2.6), (2.9), (2.93), and (2.95), we can write the time-dependent, constant-density, incompressible Navier-Stokes equations in conservation-law form in physical space as:

$$\frac{\partial u_i}{\partial t} + \frac{\partial F_{ij}}{\partial x_j} = S_i \tag{2.96}$$

$$\frac{\partial u_j}{\partial x_j} = 0 \tag{2.97}$$

where the Cartesian momentum flux tensor (F_{ij}) and the momentum source (S_i) are:

$$F_{ij} = u_j u_i + p\delta_{ij} - \nu \frac{\partial u_i}{\partial x_j} \tag{2.98}$$

$$S_i = \frac{\partial \Psi}{\partial x_i} \tag{2.99}$$

2.3.2 Filtered Navier-Stokes equations

By discretizing the equations on a grid, we are implicitly filtering the equations and removing the subgrid-scale effects that our grid cannot capture. The filtering process is the basis for the definition and development of subgrid-scale turbulence models. The grid-scale filter in a finite-volume method is effectively a "box" or "top-hat" filter, such that the filtered value of u is:

$$\overline{u} \;=\; \frac{1}{\Delta_1 \Delta_2 \Delta_3} \int_{\Delta} u \, d\Delta_1 \, d\Delta_2 \, d\Delta_3 \tag{2.100}$$

and Δ_1, Δ_2, Δ_3 are the grid spacings in the x, y, and z directions.

It is generally assumed that filtering commutes with differentiation so that the difference between the filtered equations and the unfiltered equations lies in the filtering of nonlinear terms. However, as demonstrated by Ghosal and Moin (1995), this is strictly true only for uniform filters (*i.e.* uniform grids). For computational grids with stretching and curvature to accommodate complex domain shapes, the commutation of filtering with derivatives introduces an error that is second-order in the grid filter width. Our method uses second-order algorithms so that commutation of the filtering operation is numerically consistent. However, future extensions of this work to higher-order methods must consider the filtering error introduced by commutation or develop a filtering approach that commutes to a higher order of accuracy.

The filtered Navier-Stokes equations can be presented as:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{F}_{ij}}{\partial x_j} \;=\; \overline{S}_i \tag{2.101}$$

$$\frac{\partial \overline{u}_j}{\partial x_j} \;=\; 0 \tag{2.102}$$

where the momentum flux and source are:

$$\overline{F}_{ij} \;=\; \overline{u}_j \overline{u}_i + \overline{p}\delta_{ij} - \nu \frac{\partial \overline{u}_i}{\partial x_j} + \tau_{ij} \tag{2.103}$$

$$\overline{S}_i \;=\; \frac{\partial \overline{\Psi}}{\partial x_i} \tag{2.104}$$

The term τ_{ij} has been introduced to represent the additional subgrid scale terms that arise due to the filtering of the nonlinear advection terms. This is necessary because the filtering of a product of two variables is not identical to the product of the filtered variables. Although τ_{ij} is a subgrid-scale advection term, its effects are similar to those of a viscous stress. So, much like the misnamed[2] Reynolds stress in statistical turbulence literature, τ_{ij} is generally referred to as the subgrid-scale stress. By definition, τ_{ij} is:

$$\tau_{ij} \equiv \overline{u_i u_j} - \overline{u}_i \overline{u}_j \tag{2.105}$$

The subgrid-scale stress term contains both the interaction of subgrid-scales with themselves and the interaction of the subgrid-scales with the resolved scales. In our simulation, we apply the two-parameter dynamic model of Salvetti and Banerjee (1995) which was developed from the dynamic-mixed model of Zang *et al.* (1993). The implementation of the turbulence model in the present numerical method is identical to that of Salvetti *et al.* (1996) where subgrid-scale stress is modeled by:

$$\frac{\partial \tau_{ij}}{\partial x_j} = -\frac{\partial}{\partial x_j}\left(\nu_T \frac{\partial \overline{u}_i}{\partial x_j}\right) - \frac{\partial \nu_T}{\partial x_j}\frac{\partial \overline{u}_j}{\partial x_i} + C_r \frac{\partial L_{ij}^{(m)}}{\partial x_j} \tag{2.106}$$

This model introduces three terms to the filtered Navier-Stokes equations: the eddy viscosity ν_T, the scale-similarity coefficient C_r, and the modified Leonard tensor[3] $L_{ij}^{(m)}$. The eddy-viscosity is computed dynamically using the method that originated with Germano *et al.* (1991) and was subsequently improved by Lilly (1992), Zang *et al.* (1993), Salvetti and Banerjee (1995), and Salvetti *et al.* (1996). The purpose of the eddy-viscosity term is to implement the assumption that a portion of the unresolved subgrid-scale stresses can be modeled using the Smagorinsky (1963) approach in which

[2]The Reynolds stress is *not* a stress, it is an advection of temporal velocity fluctuations by the temporal velocity fluctuations that is developed in statistical treatments of the Navier-Stokes equations. The effect is generally treated in a fashion similar to viscous stresses, hence the name Reynolds stress. The use of the term *stress* to describe small-scale advection effects is an unfortunate stretching of technical grammar since it implies physics that are not rigorously supported by the mathematics of the derivation of the term. The Reynolds stress and the sub-grid scale stress are mathematical constructs of our measurement and simulation techniques and are not *real* stresses in the fluid.

[3]The modified Leonard tensor is generally presented as L_{ij}^{m} in the literature, but this nomenclature can be confused with that of a third-order tensor when curvilinear coordinates are being used. Therefore, we have adopted an unconventional notation with the m in parentheses to indicate that it is not a tensor superscript, but simply part of the variable description.

the unresolved stresses are assumed to be proportional to the resolved strain rates. The modified Leonard tensor represents the resolved portion of the subgrid-scale stress that can be computed explicitly. The scale-similarity coefficient (C_r) of the Leonard term is computed dynamically using the approach developed by Salvetti and Banerjee and implemented in Zang's numerical code by Salvetti *et al.* (1996). If $C_r = 1$, the Leonard tensor term is simply the computation of the resolved part of the subgrid-scale stress. For $C_r \neq 1$, the Leonard tensor term becomes a modeling term under the assumption that a part of the unresolved subgrid-scale stress is proportional to the resolved Leonard tensor. For more detail on the Leonard tensor and dynamic computation of ν_T and C_r, see Salvetti and Banerjee (1995) and Salvetti *et al.* (1996).

It is convenient to split the subgrid-scale stress so that part is contained in the momentum flux, and part is held in the source term. This requires that the momentum equations be rewritten as:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{\mathrm{F}}_{ij}}{\partial x_j} = \overline{\mathrm{S}}_i \tag{2.107}$$

where

$$\overline{\mathrm{F}}_{ij} = \overline{u}_j\overline{u}_i + \overline{p}\delta_{ij} - (\nu + \nu_T)\frac{\partial \overline{u}_i}{\partial x_j} \tag{2.108}$$

$$\overline{\mathrm{S}}_i = \frac{\partial \overline{\Psi}}{\partial x_i} + \frac{\partial \nu_T}{\partial x_j}\frac{\partial \overline{u}_j}{\partial x_i} - C_r\frac{\partial L_{ij}^{(m)}}{\partial x_j} \tag{2.109}$$

The above has a slight change in nomenclature when compared with equations (2.103) and (2.104). A portion of the subgrid-scale stress (τ_{ij}) that is included in the definition of the momentum flux ($\overline{F}_{ij}$) in equation (2.103) has been moved into the source term ($\overline{\mathrm{S}}_i$) in equation (2.109). It follows that $\overline{\mathrm{F}}_{ij} \neq \overline{F}_{ij}$ and $\overline{\mathrm{S}}_i \neq \overline{S}_i$.

2.3.3 Curvilinear transformation of N-S equations

Solution of numerical problems in complicated domains using boundary-fitted curvilinear coordinates is now a standard technique requiring little introduction. The method is based on concepts of tensor analysis and coordinate transformation found in textbooks such as Aris (1962). Numerical application of boundary-fitted curvilinear coordinates involves developing a coordinate system which matches the boundaries

in physical space and maps to a regular orthogonal grid in computational space. The distortions of the curvilinear coordinate system (as viewed from physical space) are measured as metric terms. The metric terms are created when the governing equations are transformed onto a regular orthogonal grid in computational space. Using this technique, an irregular domain in physical space can be discretized with a structured (*e.g.* finite difference) numerical method on a regular grid in computational space while maintaining the simulation boundary coincident with the physical boundary. The main disadvantages of this technique are (1) the governing equations are more complicated when transformed to computational space, and (2) the storage of metric terms can overwhelm computer memory. As computer memory capacity has increased, the latter disadvantage has almost disappeared for simulations with stationary grids. However, for moving-grid simulations, the computation of the grid movement can require holding two copies of the grid and/or some metrics in memory simultaneously, which significantly increases the memory requirements for the simulation. The development of curvilinear coordinate systems for complicated domains with fixed grids is covered extensively in Meakin and Street (1988).

The transformation techniques used for stationary grids are readily extended to moving grids by the addition of a curvilinear transformation of time derivatives. Although time measured in curvilinear space may be identical to time measured in physical space, their partial derivatives (in general) are not identical. A partial derivative with respect to time in physical space is taken at a fixed point in physical space, whereas a partial derivative with respect to time in curvilinear space is taken at a fixed point in curvilinear space (which may be moving through physical space). Extension of boundary-fitted curvilinear coordinate transformations from fixed to moving-boundary problems is straightforward and is described by Thompson *et al.* (1985) as well as in most other numerical grid generation textbooks.

Implementation of boundary-fitted curvilinear coordinates requires differential operators for transforming the governing equations from physical space to computational space. We can derive the transformation operators using the chain rule for partial differential equations, resulting in:

$$\frac{\partial}{\partial x_j} = \frac{\partial \xi^q}{\partial x_j}\frac{\partial}{\partial \xi^q} \tag{2.110}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \frac{\partial x_j}{\partial \tau}\frac{\partial \xi^q}{\partial x_j}\frac{\partial}{\partial \xi^q} \tag{2.111}$$

where x_j, with $j = 1,2,3$ are the physical space coordinates; ξ^q with $q = 1,2,3$ are the computational space coordinates; $\partial/\partial t$ is a time derivative taken at a fixed point in physical space; $\partial/\partial \tau$ is a time derivative taken at a fixed point in computational space, and repeated subscript/superscript combinations imply summation. We are using a formal general tensor representation with superscripts for the curvilinear coordinates so that covariant and contravariant tensor components can be properly distinguished. The plethora of terms of the form $\partial \xi^q/\partial x_j$ encourages us to adopt the notation used by a number of authors, where we define a surface metric tensor as a mixed tensor[4] such that:

$$S_j^q \equiv \frac{\partial \xi^q}{\partial x_j} \tag{2.112}$$

If we let a time derivative with respect to computational space be represented by a dot, then the transformation rules become:

$$\frac{\partial}{\partial x_j} = S_j^q \frac{\partial}{\partial \xi^q} \tag{2.113}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \dot{x}_j S_j^q \frac{\partial}{\partial \xi^q} \tag{2.114}$$

To simulate a flow with a free surface in boundary-fitted curvilinear coordinates, equations (2.113) and (2.114) are used to transform the physical space Navier-Stokes equations into computational space. Completing the transformation requires the metric identity noted in Thompson *et al.* (1985):

$$\frac{\partial}{\partial \xi^q}\left(J^{-1} S_i^q\right) \equiv 0 \tag{2.115}$$

along with the conservation of space explained by Demirdžić and Perić (1988):

$$\frac{\partial}{\partial \tau}\left(J^{-1}\right) - \frac{\partial}{\partial \xi^q}\left(J^{-1} S_j^q \dot{x}_j\right) \equiv 0 \tag{2.116}$$

[4]This is only a correct tensor representation when x_j is a Cartesian coordinate system so that $x_j \equiv x^j$.

Application of the transformation rules, (equations [2.113] through [2.116]) to the filtered Navier-Stokes equations ([2.102], [2.107], [2.108], and [2.109]) provides the unsteady, incompressible, constant-density, filtered Navier-Stokes equations in time-dependent boundary-fitted curvilinear coordinates as:

$$\frac{\partial}{\partial \tau}\left(J^{-1}\overline{u}_i\right) + \frac{\partial}{\partial \xi^q}\left(J^{-1}\overline{\mathcal{F}}_i^q\right) = \overline{\mathcal{S}}_i \tag{2.117}$$

$$\frac{\partial}{\partial \xi^q}\left(J^{-1}\overline{U}^q\right) = 0 \tag{2.118}$$

where the curvilinear momentum tensor $\overline{\mathcal{F}}_i^q$ and the curvilinear source vector $\overline{\mathcal{S}}_i$ are:

$$\overline{\mathcal{F}}_i^q = \left(\overline{U}^q - \dot{X}^q\right)\overline{u}_i + S_i^q\overline{p} - (\nu + \nu_T)\, G^{qr}\frac{\partial \overline{u}_i}{\partial \xi^r} \tag{2.119}$$

$$\overline{\mathcal{S}}_i = \frac{\partial}{\partial \xi^q}\left(J^{-1}S_i^q\overline{\Psi}\right) + J^{-1}S_j^q S_i^r\frac{\partial \nu_T}{\partial \xi^q}\frac{\partial \overline{u}_j}{\partial \xi^r} - C_r\frac{\partial}{\partial \xi^q}\left(J^{-1}S_j^q L_{ij}^{(m)}\right) \tag{2.120}$$

and other curvilinear quantities are defined as[5]:

$$\text{inverse Jacobian} \qquad J^{-1} = det\left|\frac{\partial x_i}{\partial \xi^s}\right| \tag{2.121}$$

$$\text{contravariant velocity} \qquad \overline{U}^q = S_j^q\overline{u}_j \tag{2.122}$$

$$\text{contravariant grid velocity} \qquad \dot{X}^q = S_j^q\dot{x}_j \tag{2.123}$$

$$\text{contravariant volume metrics} \qquad G^{qr} = S_j^q S_j^r \tag{2.124}$$

$$\text{contravariant surface metrics} \qquad S_i^q = \frac{\partial \xi^q}{\partial x_i} \tag{2.125}$$

2.3.4 Non-dimensional N-S equations

In section 2.2 we developed a detailed non-dimensionalization of the governing equations that reveals the scaling of the terms. For practical numerical implementation,

[5]Readers familiar with the work of Zang (1993) should be careful here: we have defined our contravariant metrics and velocities *without* the inverse Jacobian, whereas Zang defined contravariant fluxes and Jacobian normalized metrics. Admittedly, Zang's approach is cleaner in its presentation of the Navier-Stokes equations, but it complicates the presentation of the dynamic boundary condition which we provide in section 2.5.

it is sufficient to non-dimensionalize the entire set of equations using a single velocity scale, a single length scale, and the time scale that results from their quotient. However, it must be kept in mind that the other non-dimensional parameters of equations (2.80) through (2.86) are implicitly determined when setting the relationships between wavelength, wave amplitude, wave period, water depth, and shear velocity.

If we non-dimensionalize Navier-Stokes equations by a velocity scale $\mathcal{U}$, a length scale $\mathcal{L}$, a time scale $\mathcal{L}/\mathcal{U}$ and the density ρ (dropping the usual [*] nomenclature that goes with the non-dimensionalization), the momentum flux and source terms can be written as:

$$\overline{\mathcal{F}}_i^q = J^{-1}\left(\overline{U}^q - \dot{X}^q\right)\overline{u}_i + S_i^q\overline{p} - \left(Re^{-1} + Re_T^{-1}\right)G^{qr}\frac{\partial\overline{u}_i}{\partial\xi^r} \tag{2.126}$$

$$\overline{\mathcal{S}}_i = \frac{\partial}{\partial\xi^q}\left(S_i^q\overline{\Psi}\right) + \frac{1}{J^{-1}}S_j^qS_i^r\frac{\partial\overline{u}_j}{\partial\xi^r}\frac{\partial}{\partial\xi^q}\left(Re_T^{-1}\right) - C_r\frac{\partial}{\partial\xi^q}\left(S_j^qL_{ij}^{(m)}\right) \tag{2.127}$$

where the Reynolds numbers are:

$$Re \equiv \frac{\mathcal{U}\mathcal{L}}{\nu} \tag{2.128}$$

$$Re_T \equiv \frac{\mathcal{U}\mathcal{L}}{\nu_T} \tag{2.129}$$

and all physical space variables are non-dimensional.

The non-dimensional form of the definition of the modified pressure, equation (2.95), provides:

$$p \equiv \frac{P}{\rho} + \frac{1}{(Fr)^2}x_3 \tag{2.130}$$

where the Froude number is:

$$Fr \equiv \frac{\mathcal{U}}{\sqrt{g'\mathcal{L}}} \tag{2.131}$$

2.4 Kinematic boundary condition

The kinematic boundary condition is the Lagrangian condition that a particle on the surface must remain on the surface. If $F = 0$ is a function that describes the location

of the surface, then the kinematic condition requires that:

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0 \tag{2.132}$$

It is possible to use directly the Lagrangian condition, advancing the free surface by moving marker particles based upon their velocity at the free surface; however, Chan and Street (1970a) showed this method is unstable for long simulations where an explicit advance is used to integrate the free-surface position. For coupled-grid solutions (as discussed in section 1.3.4), where the kinematic boundary condition and grid generation are coupled to an implicit solution of the velocity and pressure, such instability should not occur with the Lagrangian boundary condition.

2.4.1 Physical space kinematic boundary condition

As an alternative to the implicit Lagrangian approach, the kinematic boundary condition in physical space can be written in a physical-space Eulerian form which can be obtained through a Taylor series expansion as shown in Mei (1983), resulting in:

$$\frac{\partial H}{\partial t} = u_3 - u_1 \frac{\partial H}{\partial x} - u_2 \frac{\partial H}{\partial y} \tag{2.133}$$

where H is the height (in the x_3 direction) of the free surface measured from some baseline in physical space. This boundary condition is enforced on surface particles that are restricted to vertical motion in physical space, and is therefore unsuited for overturning waves. The restricted curvilinear coordinate grid generation method (see section 3.2) shares the same limitations, so the two approaches are often seen together in numerical methods.

Note that if we non-dimensionalize with the same length and velocity scales used for the Navier-Stokes equations in section 2.3.4, then non-dimensional groups are not associated with the kinematic boundary condition. Under these conditions the equation is identical in dimensional and non-dimensional forms.

2.4.2 Curvilinear-space kinematic boundary condition

It was demonstrated by Hinatsu (1992) that curvilinear coordinate transformations, equations (2.113) and (2.114), can be applied to the physical-space Eulerian kinematic boundary condition, equation (2.133), for use in numerical simulations. This approach (also used by Park *et al.* [1993]) is workable in curvilinear coordinates, but retains the underlying vertical motion restriction on surface particles. The restriction makes the method unsuitable for waves which do not remain single-valued and brings into question the validity of the method for steep waves[6].

There is no reason that we must begin with the Cartesian form of an equation and transform it into computational space. A more general approach that does not have a single-valuedness restriction in physical space requires deriving the Eulerian kinematic boundary condition directly in curvilinear coordinates. This approach was used by Hino (1989). However it was applied in a fixed curvilinear system rather than in a moving-grid system, and appears to have been abandoned in the author's later work. A derivation is presented here because it does not appear elsewhere in the literature[7].

To obtain directly a curvilinear Eulerian kinematic boundary condition, we consider a *fixed*[8] curvilinear space (ξ^1, ξ^2, ξ^3) such that the free-surface is single-valued

[6]Steep waves that are defined by particles restricted to vertical motion have the surface motion at non-orthogonal angles to the surface. This generates additional error in the numerics due to the requirement to handle the skew-metric terms such as G^{13}.

[7]In some senses, this derivation of the curvilinear kinematic boundary condition is extension of the textbook derivation of the physical-space Eulerian kinematic boundary condition found in Mei (1983).

[8]To try to derive a useful curvilinear kinematic boundary condition with a *moving*, boundary-fitted grid would be counter-productive since the definition of moving curvilinear surface implies that

$$\frac{\partial \mathcal{H}}{\partial \tau} \equiv 0$$

That is, the reference space for $\mathcal{H}$ is moving along with $\mathcal{H}$ so it cannot be used to determine the movement of the space. For the purposes of derivation with a fixed grid, the boundary-fitted restriction is superfluous. For numerical discretization, the boundary-fitted requirement is used with the kinematic boundary condition in section 3.4.2.

in ξ^3. Define F as a scalar function for the free surface such that:

$$F\left(\boldsymbol{\xi}\,,\,t\right) \;=\; \xi^3 \;-\; \mathcal{H}\left(\xi^1\,,\,\xi^2\,,\,t\right) \;=\; 0 \tag{2.134}$$

where $\boldsymbol{\xi}$ is a vector representing the curvilinear coordinates of a surface position at time t, and $\mathcal{H}$ is the height of the free surface measured from $\xi^3 = 0$ along a line of constant ξ^1 and ξ^2 in fixed curvilinear space.

After some small time Δt, the free surface has moved, while the curvilinear coordinate system remains fixed. We require that Δt is small, so the free surface remains single-valued in ξ^3. A Taylor-series expansion gives:

$$F\left(\boldsymbol{\xi} \;+\mathbf{U}\Delta t\,,\,t+\Delta t\right) = \;F\left(\boldsymbol{\xi}\,,\,t\right) \;+\; \left(\frac{\partial F}{\partial t} \;+\; \mathbf{U}\cdot\nabla F\right)\Delta t \;+\; O\left(\Delta t\right)^2 \tag{2.135}$$

where $\mathbf{U}$ is the contravariant velocity vector of a point on the surface. It follows that:

$$\frac{\partial F}{\partial t} \;+\; \mathbf{U}\cdot\nabla F \;=\; 0 \tag{2.136}$$

Substitution of equation (2.134) into equation (2.136) provides the curvilinear kinematic boundary condition in *fixed* curvilinear coordinates as:

$$\frac{\partial \mathcal{H}}{\partial t} \;=\; U^3 \;-\; U^1\frac{\partial \mathcal{H}}{\partial \xi^1} \;-\; U^2\frac{\partial \mathcal{H}}{\partial \xi^2} \tag{2.137}$$

2.4.3 Filtered curvilinear kinematic boundary condition

Because the kinematic boundary condition is inherently a nonlinear condition, the filtering of the equation results in subgrid-scale terms. In this case, we can follow the approach used by Zang *et al.* (1993) for filtering the density terms. The filtered kinematic boundary condition can be presented as

$$\frac{\partial \overline{\mathcal{H}}}{\partial t} \;=\; \overline{U}^3 \;-\; \overline{U}^\alpha\frac{\partial \overline{\mathcal{H}}}{\partial \xi^\alpha} \;+\; \chi^\alpha_\alpha \qquad : \qquad \alpha = 1,2 \tag{2.138}$$

where the subgrid-scale terms are defined as:

$$\chi^\alpha_\alpha \;\equiv\; \overline{U^\alpha\frac{\partial \mathcal{H}}{\partial \xi^\alpha}} \;-\; \overline{U}^\alpha\frac{\partial \overline{\mathcal{H}}}{\partial \xi^\alpha} \tag{2.139}$$

We can follow the same approach that Germano *et al.* (1991) used to redefine the subgrid-scale stresses into the: (1) modified Leonard term, (2) modified cross term,

and (3) modified Reynolds term. The velocity and surface height are decomposed into resolved and unresolved parts so that:

$$U^\alpha = \overline{U}^\alpha + u^\alpha \tag{2.140}$$

$$H = \overline{\mathcal{H}} + h \tag{2.141}$$

where the overbars indicate resolved terms, and the lower-case letter represent subgrid-scale terms (and should not be confused with Cartesian variables in this instance). It follows that:

$$\chi^\alpha_\alpha \equiv \overline{\left(\overline{U}^\alpha + u^\alpha\right)\left(\frac{\partial\overline{\mathcal{H}}}{\partial\xi^\alpha} + \frac{\partial h}{\partial\xi^\alpha}\right)} - \left(\overline{\overline{U}^\alpha + u^\alpha}\right)\left(\overline{\frac{\partial\overline{\mathcal{H}}}{\partial\xi^\alpha} + \frac{\partial h}{\partial\xi^\alpha}}\right) \tag{2.142}$$

We can borrow and misapply the turbulence modeling nomenclature and define the modified "Leonard," "cross," and "Reynolds" terms as:

$$L^\alpha_\alpha \equiv \overline{\overline{U}^\alpha\frac{\partial\overline{\mathcal{H}}}{\partial\xi^\alpha}} - \overline{\overline{U}^\alpha}\frac{\partial\overline{\mathcal{H}}}{\partial\xi^\alpha} \tag{2.143}$$

$$C^\alpha_\alpha \equiv \overline{\overline{U}^\alpha\frac{\partial h}{\partial\xi^\alpha} + u^\alpha\frac{\partial\overline{\mathcal{H}}}{\partial\xi^\alpha}} - \left(\overline{\overline{U}^\alpha}\frac{\partial\overline{h}}{\partial\xi^\alpha} + \overline{u^\alpha}\frac{\partial\overline{\mathcal{H}}}{\partial\xi^\alpha}\right) \tag{2.144}$$

$$R^\alpha_\alpha \equiv \overline{u^\alpha\frac{\partial h}{\partial\xi^\alpha}} - \overline{u^\alpha}\frac{\partial\overline{h}}{\partial\xi^\alpha} \tag{2.145}$$

The modified Leonard term L^α_α is made up of resolved quantities and can be computed explicitly using the method of Zang *et al.* (1993). For the cross term and the Reynolds term a model is required. In the development of the dynamic mixed model for density variations, Zang could argue that the subgrid-scale density terms responded to the strain rates in a fashion similar to the subgrid-scale velocity terms. This allowed the development of a dynamic model where the subgrid-scale density variations were modeled with an eddy diffusivity term (similar to an eddy viscosity term) and a scale-similarity term. Both effects were assumed to be proportional to the magnitude of the resolved strain rate and were computed dynamically on the basis of a test-filter scheme and least-squares fit. However, the free surface will not support a shear stress, and the basis of a kinematic boundary condition model solely on the irrotational strain rates may be questionable. One can certainly make a scale-similarity argument that the cross terms should be proportional the the Leonard term, but there is an open question as to the appropriate constant of proportionality. Certainly when dealing

with the small-scale kinematics of the free surface it would be wise to consider the dynamics of the flow and the dynamic boundary condition. For the small scales of motion near the surface, pressure, viscous, and capillary effects in the dynamic boundary condition may all be of similar orders of magnitude and are more likely to drive the kinematics of the subgrid-scale flow than is the resolved velocity field. Further theoretical discussion of this issue will be delayed until section 4.10.

A simple approach suitable for initial investigations into LES modeling of the kinematic boundary condition is to compute the modified Leonard term directly using the method in Zang *et al.* (1993). If the cross terms and Reynolds terms are neglected the kinematic boundary condition can be written as:

$$\frac{\partial \overline{\mathcal{H}}}{\partial t} = \overline{U}^3 - \overline{U}^1 \frac{\partial \overline{\mathcal{H}}}{\partial \xi^1} - \overline{U}^2 \frac{\partial \overline{\mathcal{H}}}{\partial \xi^2} + \mathcal{L}_1\left(\overline{U}^1, \overline{\mathcal{H}}\right) + \mathcal{L}_2\left(\overline{U}^2, \overline{\mathcal{H}}\right) \tag{2.146}$$

where $\mathcal{L}$ is defined as a Leonard stress operator:

$$\mathcal{L}_\alpha\left(\overline{U}^\alpha, \overline{\mathcal{H}}\right) = \overline{\overline{U}^\alpha \frac{\partial \overline{\mathcal{H}}}{\partial \xi^\alpha}} - \overline{\overline{U}}^\alpha \frac{\partial \overline{\overline{\mathcal{H}}}}{\partial \xi^\alpha} \tag{2.147}$$

This approach recalls the initial development stages of LES methods for the Navier-Stokes equations, and is justifiable because we simply do not have enough information to do any better. The development of more sophisticated models requires well-resolved DNS experiments of nonlinear free-surface flows to provide a sound basis for examining the physics near the surface. Such DNS simulations do not yet exist. Laboratory experiments such as Cowen (1996) could provide some basis for testing LES models at the free surface, but have two major drawbacks: (1) the laboratory data collection techniques need to be extended to three-dimensions to obtain sufficient data to validate the three-dimensional terms in the LES models, and (2) the laboratory experiments necessarily have surface-tension effects which complicate model development and validation: for simplicity, it is preferable to first develop a model without surface tension effects that is directly comparable to a DNS simulation that neglects surface-tension.

The effects of the addition of the subgrid-scale term are not investigated in this dissertation as the available computational power and the memory requirements of the present simulation method make it difficult to run a set of validation simulations at a finer resolution. Without such simulations it would be difficult to analyze the effects of the subgrid-scale term on the kinematic boundary condition. To avoid

contaminating the results of our simulations by an untested algorithm, the Leonard terms ($\mathcal{L}_1$, and $\mathcal{L}_2$) of the filtered kinematic boundary condition at the free surface were not implemented for the simulations presented in chapters 5 through 7 of this dissertation. The previous derivation of the terms and the implementation presented in section 3.4.2 are provided for completeness.

2.5 Dynamic boundary condition

The dynamic boundary condition is generally obtained by assuming that (1) a free surface will only support the normal stress of surface tension and (2) tangential stresses must disappear. The result is the dynamic boundary condition for an incompressible fluid in its classic form (similar to that in Batchelor [1967]):

$$P_{s+} - P_{s-} = -2\mu\, e_{ij} n_i n_j + \gamma \left(R_1^{-1} + R_2^{-1} \right) \tag{2.148}$$

$$e_{ij} t_i n_j = 0 \tag{2.149}$$

where the subscripts $s+$ and $s-$ indicate the pressure on the upper and lower sides of the free surface, e_{ij} is the rate-of-strain tensor, equation (2.6), n_i and t_i are the unit normal and tangent vectors, respectively, γ is the surface tension coefficient, and R_1 and R_2 are the principal radii of curvature of the surface.

For most purposes, this form of the dynamic boundary condition is more than suitable. In fact, it is often approximated as simply $P = 0$. However, it should be noted that there are actually several other terms that should appear on the right-hand-side of the equation. A derivation of all the terms of the dynamic boundary condition requires the use of general curvilinear coordinates and can be found in Scriven (1960) and Aris (1962). To obtain equations (2.148) and (2.149) requires that we neglect: (1) gradients of surface tension, (2) inertia of the surface, (3) gradients of the dilational force, (4) force due to total curvature and velocity, (5) effects of varying normal velocity, (6) normal forces due to dilation and shear, and (7) the viscosity of the upper fluid.

Equations (2.148) and (2.149) do not provide for straightforward implementation in a boundary-fitted curvilinear coordinate numerical method; therefore, our approach will begin with the tensor form of the full equation from Scriven (1960). By applying the seven assumptions used to get equations (2.148) and (2.149) along with

the requirement that the curvilinear coordinate system be boundary orthogonal, the dynamic boundary condition can be presented as:

$$(P_{s+} - P_{s-}) = -2\mu U^3_{,3} + 2M\gamma \tag{2.150}$$

$$U^1_{,3} = -G_{33}\left\{G^{11}U^3_{,1} + G^{12}U^3_{,2}\right\} \tag{2.151}$$

$$U^2_{,3} = -G_{33}\left\{G^{22}U^3_{,2} + G^{12}U^3_{,1}\right\} \tag{2.152}$$

where G_{33} is the covariant metric:

$$G_{33} = \frac{\partial x_j}{\partial \xi^3}\frac{\partial x_j}{\partial \xi^3} \tag{2.153}$$

and M is the mean curvature, defined for a ξ^3 surface in a boundary-orthogonal coordinate system as:

$$M = \frac{1}{2}\frac{G^{\alpha\beta}}{G^{33}}\left(S^3_j\frac{\partial^2 x_j}{\partial\xi^\alpha\partial\xi^\beta}\right) \tag{2.154}$$

with α and β summed over $1, 2$ and j summed over $1, 2, 3$.

Note that the differentiation in equations (2.150), (2.151), and (2.152) is covariant tensor differentiation and requires the application of Christoffel symbols for deriving a discrete implementation. Because we have required boundary orthogonality, the covariant differentiation translates directly into a simple differentiation in the tangential terms (but adds additional complexity to the normal term). If boundary orthogonality is not provided, a number of additional terms will be required in both equations.

If we let the outside pressure (P_{s+}) equal zero, then apply equation (2.130) for the reduced pressure and non-dimensionalize, the normal component of the dynamic boundary condition can be written as:

$$p_{s-} = \frac{z_{s-}}{(Fr)^2} + \frac{2}{Re}U^3_{,3} - \frac{2}{We}M \tag{2.155}$$

where z_{s-} is the vertical Cartesian coordinate at the surface, the Weber number is defined as

$$We = \frac{\rho\mathcal{U}^2\mathcal{L}}{\gamma} \tag{2.156}$$

and the Reynolds and Froude numbers are defined in equations (2.128) and (2.131).

Applying some algebra and tensor manipulation, we reduce the dynamic boundary condition of equations (2.151), (2.152), and (2.155) to a form that can

be more readily implemented in a numerical method. For the present second-order method, metrics can be assumed to pass through the filter operation[9] then the dynamic boundary condition can be written as:

$$\overline{p}_{s-} = \frac{\overline{z}_{s-}}{(Fr)^2} - \frac{2M}{We} + \frac{1}{Re}\left\{2\frac{\partial \overline{U}^3}{\partial \xi^3} + G^{33}\left[\overline{U}^1 \frac{\partial}{\partial \xi^1}\left(G_{33}\right) + \overline{U}^2 \frac{\partial}{\partial \xi^2}\left(G_{33}\right)\right]\right\} \quad (2.157)$$

$$\frac{\partial \overline{U}^1}{\partial \xi^3} = -G_{33}\left\{G^{11}\frac{\partial \overline{U}^3}{\partial \xi^1} + G^{12}\frac{\partial \overline{U}^3}{\partial \xi^2}\right\} \quad (2.158)$$

$$\frac{\partial \overline{U}^2}{\partial \xi^3} = -G_{33}\left\{G^{22}\frac{\partial \overline{U}^3}{\partial \xi^2} + G^{12}\frac{\partial \overline{U}^3}{\partial \xi^1}\right\} \quad (2.159)$$

If our grid is sufficiently fine and the surface is sufficiently smooth, then the tangential derivatives of the G_{33} metrics on the right-hand-side of equation (2.157) will be small and can be neglected. However, if the free surface has oscillations on the order of the grid scale or slightly larger, then these terms cannot be neglected *a priori*.

[9] See section 2.3.2 for a discussion of problems with this assumption for higher order methods.

Chapter 3

Numerical method

3.1 General approach

The numerical approach in this dissertation is a free-surface/moving-grid adaption of the method developed by Zang *et al.* (1994) for fixed-grid simulations. This method has been shown to be second-order accurate in space and time for laminar and turbulent flows and is efficient for computation of four-dimensional turbulent flow problems. Zang's approach uses finite-volume discretization of the primitive variables in curvilinear coordinates. The Navier-Stokes solution is through a fractional-step (or projection) method with a multigrid solution of the pressure Poisson equation. The free-surface algorithm developed as a part of this research project has been designed to work within the framework of Zang's method, but is general enough to be adapted to any projection method. Our approach to the free-surface problem is to use a boundary-fitted grid with the free-surface advance and flow solver decoupled from the grid generation. The integration of the kinematic boundary condition for the movement of the free surface is through a Runge-Kutta 4th-order (RK4) method with 5th-order upwind discretization for spatial derivatives. The solution grid is generated with the Poisson equation method using a FORTRAN code adapted from the 3DGRAPE/AL code developed by Sorenson and Alter (1996) at NASA Ames and Langley laboratories.

3.2 Grid generation

As the grid generation community has grown rapidly in the past decade, a number of methods have become available for generating boundary-fitted curvilinear coordinate grids (for a selection of recent work, see Soni *et al.* [1996]). Not all the methods are suitable for a moving-grid simulation. For the boundary to track the free surface, a

new grid must be computed at each time step, so the ideal method for grid generation is fast and efficient in its use of computer memory and storage. Unfortunately, the grid generation research community appears to be driven by the aerospace industry, whose interest is in computing grids around aircraft. Much of the grid generation literature is devoted to handling the difficult aspects of generating grids around complicated shapes and the computer interface between the grid generation engineer and software applications. Because complicated grids are usually input only once, then modified only as an aircraft structural design changes, there has been little effort placed into developing fast and efficient methods for simple geometries such as we have with a rectangular domain having a single wavy surface. This may be changing since the CFD solvers for compressible flow have become sophisticated enough that the U.S. Air Force is interested in solving dynamic problems with changes in aircraft geometry (*e.g.* movement of aircraft control surfaces or the separation of ordnance from a wing rack). We can hope that by the time the environmental fluid mechanics world is ready to tackle viscous simulations of the physics of breaking waves, the grid generation researchers will have developed more efficient tools for the computation of grids.

Perhaps the simplest approach for grid generation under a wavy surface is to define one set of computational-space coordinate lines as vertical lines in physical space, then apply distribution functions to locate the coordinates smoothly along the set of vertical lines. This is the predominant approach among the moving-grid free-surface simulations in the literature. We might call this arrangement a "restricted" curvilinear grid because the free surface points are restricted to motion along the vertical coordinate lines. A simple 2D grid of this type is shown in figure (3.1).

The advantage of this type of grid is that it is easy to generate. Like most easy approaches, its disadvantages are legion, including: (1) limitation to single-valued wave surfaces (2) reduction in solution accuracy due to grid skewness, (3) increasing local errors near the free surface with increasing wave height, and (4) lack of orthogonality at the free-surface boundary. The last point implies the need for a more complicated dynamic boundary condition on the curvilinear velocities at the free surface: the covariant differentials have skew metric terms that do not disappear for non-orthogonal systems at the boundary.

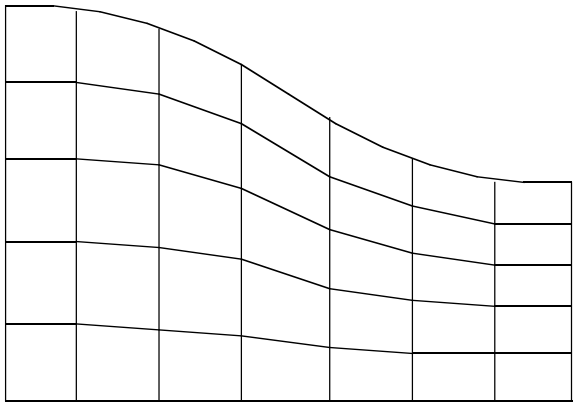


Figure 3.1: Restricted curvilinear coordinate grid

The disadvantages of the restricted curvilinear grid help point out our requirements for a good grid generation method, it should: (1) be computationally fast (vectorized), (2) use a modest amount of computer memory, (3) not require disk access during grid computation, (4) be able to generate boundary-orthogonal grids, (5) be able to handle a steep or overturning wave (no single-valuedness restriction), (6) minimize grid skewness, and (7) allow control of grid stretching.

Grids which are not subject to the disadvantages of the restricted grid method can be computed using a variety of approaches found in the numerical grid generation literature. The most mature approaches for computing three-dimensional grids are transfinite interpolation methods and Poisson equation methods. The transfinite interpolation approach involves defining bounding surfaces and performing three-dimensional iterpolation between surfaces to obtain the coordinates of grid points in the volume. This is sometimes called an "algebraic" grid generation method because the implementation requires direct solution of polynomial interpolating functions. While methods based on transfinite interpolation are generally fast, they cannot guarantee the resulting grid will be free from singularities and overlapping of grid lines for arbitrarily-shaped domains.

A characteristic of transfinite grid generation methods is that the solution is not dependent on an initial estimate of the grid in the interior. For applications where a grid is only generated once, this can be an advantage. For the present work, the grid only changes a small amount between time steps[1] so it can be argued that

[1]This is a requirement of the CFL condition with a fine grid discussed in section 4.9.3.

a transfinite interpolation method is doing extra work because it is always starting from the boundaries rather than from an interior solution that is close to the desired grid.

For the Poisson equation approach to grid generation, the relationship between curvilinear and Cartesian space is represented as a Poisson equation that is solved to obtain the Cartesian locations of the grid points associated with each point in curvilinear space. The source of the Poisson equation is used as a control function to set the spacing and approach angle[2] of the grid near the boundaries. The source term required for the desired spacing and approach angle cannot be determined *a priori*, but must be iteratively adjusted during the grid solution. The typical Poisson approach to grid generation requires: (1) an initial estimate of the grid coordinates and source-term control functions required to obtain the grid boundary conditions, (2) iterative smoothing of the Poisson equation holding the source term fixed, (3) evaluation of the *unconverged* grid to obtain a new estimate of the source term, (4) repeat steps (2) and (3) until a desired convergence is reached. The number of iteration steps between evaluations of the source term and the choice of method for estimating the correct source term will determine the rate at which the grid solution converges for a particular domain and initial conditions. Poor selection of the initial conditions (grid and control function estimate) or the iteration approach can cause the Poisson solution to diverge.

The Poisson equation method does not solve for the grid by simply solving a Poisson equation, instead it is solving *for* the Poisson equation that relates Cartesian and curvilinear space. Rather than thinking of the Poisson equation approach as solving an equation of the form $\nabla^2 \mathbf{x} = \mathbf{s}$, it is useful to think of the method as solving the equation $\nabla^2 \mathbf{x} - F(\mathbf{x}) = 0$ where $\mathbf{x}$ is the unknown location of the Cartesian grid points and F is a prescribed function with unknown coefficients that vary with $\mathbf{x}$. For more detail on Poisson equation and transfinite interpolation methods for grid generation see Thompson *et al.* (1985).

[2]Boundary orthogonality with fine grid spacing is usually desired.

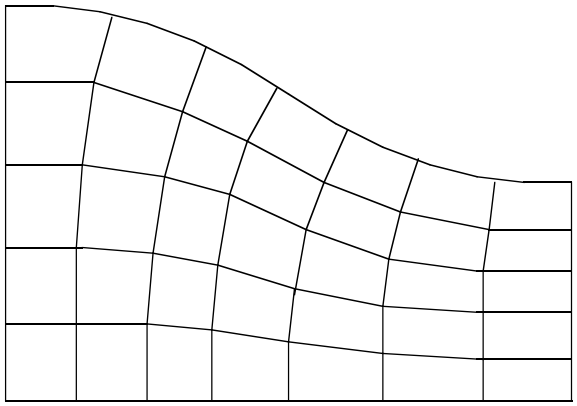


Figure 3.2: Generalized curvilinear coordinate grid

Both the transfinite interpolation approach and the Poisson equation method can produce grids suitable for computing steep or overturning waves and with boundary orthogonality. The grids produced by the such methods might be called "generalized grids" since they can be used in general topologies. An example of such a grid for a large wave deformation is shown in figure (3.2).

The most advanced grid-generation methods (such as the Poisson equation method) are capable of providing boundary-orthogonality, control over grid skewness, and specified point distributions near the boundaries. Control over point distributions is desirable so that grid points can be concentrated in boundary layers for efficient use of computational effort. There are also distinct advantages in the implementation of numerical techniques with boundary orthogonality. As shown by Zang (1993), to obtain consistent solutions on a fixed grid with a non-orthogonal boundary in a finite-volume method requires a pressure boundary condition as well as a velocity boundary condition for the pressure equation. The use of boundary orthogonality for the grid removes the requirement for the pressure boundary condition. Furthermore, in deriving a discrete form of the dynamic boundary condition in curvilinear space, boundary orthogonality provides significant simplification by removing two of the three skew metric terms at the free-surface. That is, for a free surface $\xi^3 = 1$, boundary-orthogonality provides $G^{13} = G^{23} = 0$ (where the G^{ij} terms are metrics defined by equation (2.124).

Some advanced grid generation methods provide for control over grid skewness, which is desirable because large grid skewness tends to produce less accurate results. Thompson *et al.* (1985) noted that the truncation error due to skewness

varies inversely with the sine of the angle between the curvilinear coordinate lines as viewed in physical space. A detailed investigation of the truncation errors induced by curvilinear coordinate transformations can be found in Lee and Tsuei (1992). Grid skewness has other consequences in addition to truncation error considerations. As shown by Zang (1993), if grid skewness can be limited *a priori*, the numerical algorithms can take advantage of this in the discretization of terms containing the skew metrics and provide a more efficient solution.

The uncoupled-grid numerical approach in the present work allowed experimentation with different numerical grid generation methods. We found restricted grids to be unacceptable due to the disadvantages outlined above. Initial experiments were conducted with the transfinite interpolation sections of the EAGLE grid generation code developed at Mississippi State University (Thompson [1990]). Unfortunately, this code had unacceptable computational overhead in CPU time, memory, and access to written temporary files. Our recent work has been with the 3DGRAPE/AL code developed jointly at NASA Ames and NASA Langley laboratories by Sorenson and Alter (1996). This code allows the control of boundary orthogonality and grid spacing near the boundaries through the choice of the source-term control functions of the Poisson equation[3]. We have adapted the Poisson grid generation routines from 3DGRAPE/AL to work as a callable module.

3DGRAPE/AL uses vectorized Successive Over-Relaxation (SOR) of the grid Poisson equation to smooth an initial grid estimate into a grid which satisfies the boundary orthogonality and grid spacing requirements. The control functions of the Poisson equation source term can be evaluated using one of several standard methods developed in the literature (for more detail see Sorenson and Alter [1996]). While it is accepted that multigrid techniques are faster than SOR for the solution of a Poisson equation, in this case the use of SOR is preferable due to the nature of the Poisson grid generation approach. Sorenson[4] attempted to apply a multigrid solution method to the Poisson grid generation equation and noted that there is a fundamental

[3]The user is required to choose (1) whether or not orthogonality is required on each boundary, (2) the grid spacing for the first point inside each boundary, and (3) the relative rate at which the boundary influence decays toward the interior of the domain. For the simulations presented in this dissertation, the 3DGRAPE/AL default value for the boundary influence decay is used.

[4]R.L. Sorenson, NASA Ames Laboratory, personal communication (1996).

incompatibility between multigrid methods and the Poisson grid generation method. The multigrid method is predicated on using a succession of fine and coarse grids to rapidly smooth out the difference between the Laplacian of the variable and the source term of the Poisson equation. However, Poisson grid generation methods require iterative evaluation of the source term to obtain the desired boundary configuration[5]. The iterative adjustment of the source term means that the multigrid solver is attacking a slightly different equation each time the source is adjusted. Furthermore, the method by which the source terms are generally computed (see Thompson *et al.* [1985]) introduces a slightly different error on each multigrid level. As a result, multigrid performance is severely degraded. Of course, there do not exist any numerical methods that are efficient with iterative adjustment of source terms; however, the SOR method appears to be more stable and efficient than the multigrid method in current implementations.

Generating a grid through the solution of a Poisson equation with iterative adjustment of the source term control functions can be a computationally intensive process when using only boundary information (on the order of several thousand smoothing sweeps through the domain). However, as pointed out by Raad[6], since the grid motion is restricted to small increments due to the grid CFL condition (see section 4.9.3), the time (n) grid is a good starting point for the Poisson smoother in developing the time ($n+1$) grid. This effectively reduces the number of Poisson iterations to a reasonable number. In the present work, less than 30 sweeps were required to smooth the grid after the free-surface advance. This proved to be faster than the direct solution of the grid using transfinite interpolation methods from the EAGLE grid generation code.

[5]This is one reason why Poisson grid generation is notoriously unstable and time-intensive. If the initial grid is not close enough to the desired grid, then the solution can diverge through poor adjustment of the source term.

[6]P.E. Raad, Dept. of Mech. Eng. Southern Methodist Univ., personal communication (1995).

3.3 Free-surface method

The approach used to advance the free surface in our method is an uncoupled-grid approach where the kinematic boundary condition is integrated forward in time without implicit reference to the time $(n+1)$ velocities. This approach is computationally efficient, but has numerical stability limitations (see section 4.9).

The method for free-surface advance requires that at time (n) there is a fixed curvilinear grid that is boundary-fitted in the ξ^3 coordinate. All three components of the contravariant velocity must be known at each grid point on the free surface. To find the change in the ξ^3 coordinates of the free surface at some small time later with reference to the same fixed curvilinear grid (see equation [3.15] ff.), a Runge-Kutta fourth-order method is applied with the fifth-order upwind discretization for spatial derivatives of Komori *et al.* (1993). The new curvilinear coordinates of the free surface are converted back to physical space coordinates. In this approach, the coordinates have a tendency to drift in the streamwise direction unless restrained or redistributed. Rather than using an artificial restraint, we let the grid points move in accordance with the kinematic boundary condition then redistribute using a two-dimensional cubic spline on the free surface. This maintains a smooth and even distribution of points on the surface in physical space.

The advantage of our free-surface method is that the kinematic boundary condition is enforced upon points which move along a line of constant ξ^1 and ξ^2 curvilinear coordinates rather than a line of constant x and y physical coordinates. Thus, the free surface motion is computed along lines that are locally orthogonal to the time n grid rather than at an angle that depends upon the steepness of the wave. As a result, the single-valuedness requirement in physical space is replaced by a single-valuedness requirement in curvilinear space, which is a less restrictive condition for a boundary-fitted coordinate system, requiring only that the slope of the free surface be continuous. A discontinuous slope implies wave breaking and a violation of the material condition of the free surface, so it can be said that the curvilinear form is generally valid and can be implemented numerically as long as the kinematic boundary condition itself is valid. The movement of the points along surface normal lines provides a simpler implementation than the curvilinear transformations of the physical space kinematic boundary condition used in the literature.

The ability of present method to simulate near-breaking and overturning waves is primarily a matter of the availability of computational power. The number of grid points required for accurate simulation of a wave shape increases as a wave steepens and overturns. Increasing the number of grid points along the surface affects the time step required to avoid numerical instabilities at the surface (see section 4.9). The study of near-breaking phenomena and demonstration of the numerical method for this type of problem remains a subject for future research.

3.4 Numerical discretization

3.4.1 Navier-Stokes equations

Our numerical approach follows the method developed by Zang *et al.* (1994) that descends from the methods of Kim and Moin (1985) and Harlow and Welch (1965). To discretize the momentum equation we apply the explicit 2nd-order Adams-Bashforth (AB2) algorithm to the convective terms and the off-diagonal viscous terms, with the implicit Crank-Nicolson scheme for the diagonal viscous terms. The addition of the free surface to Zang's method requires a convective grid-flux term which accounts for the convective motion of the grid. This term is discretized with a second-order approximation using the volume flux of the grid between the time (n) and $(n+1)$ physical space positions for each cell face and the time (n) velocity[7]. The pressure is removed from the momentum equation in the predictor stage of the fractional-step method and a numerical pressure variable (ϕ) is defined and computed in the solution of a Poisson equation. Second-order accurate approximate factorization is used on the left hand side of the discretized momentum equation for increased computational efficiency. The resulting system can be presented as[8]:

[7]Note that use of a multi-step discretization such as AB2 for the grid flux would not be consistent since the grid movement is an arbitrary numerical effect between two time steps and is not required to be continuous with previous time steps.

[8]For clarity, we will drop the overbar notation for filtered variables in discrete equations and use δ instead of ∂ to indicate discrete derivatives.

1. predictor step:

$$\left(I - D_1^{n+1}\right)\left(I - D_2^{n+1}\right)\left(I - D_3^{n+1}\right)\left(u_i^* - u_i^n\right) = S_i \tag{3.1}$$

2. pressure Poisson equation:

$$\frac{\delta}{\delta \xi^q}\left(J^{-1} G^{qr} \frac{\delta \phi}{\delta \xi^r}\right)^{n+1} = \frac{1}{\Delta t} \frac{\delta}{\delta \xi^q}\left(J^{-1} U^{*q}\right) \tag{3.2}$$

3. corrector steps:

   (a) for the Cartesian velocity (on cell centers):

$$u_i^{n+1} = u_i^* + \left[\frac{\Delta t}{J^{-1}} B_i\left(\phi\right)\right]^{n+1} \tag{3.3}$$

   (b) for the normal component of contravariant velocity (on cell faces):

$$\left(J^{-1} U^q\right)^{n+1} = J^{-1} U^{*q} - \Delta t \left(J^{-1} G^{qr} \frac{\delta \phi}{\delta \xi^r}\right)^{n+1} \tag{3.4}$$

The pressure variable (ϕ) is related to the reduced pressure (p) by:

$$B_i\left(p\right) = \left[J^{-1} - \frac{\Delta t}{2} D_I\right] \frac{B_i\left(\phi\right)}{J^{-1}} \tag{3.5}$$

and the source term of the predictor (equation [3.1]) is:

$$\begin{aligned} S_i = \frac{\Delta t}{\left(J^{-1}\right)^{n+1}} \Bigg\{ & \frac{3}{2}\left(C_i^n + D_E^n\left[u_i^n\right]\right) - \frac{1}{2}\left(C_i^{n-1} + D_E^{n-1}\left[u_i^{n-1}\right]\right) \\ & + \frac{1}{2}\left(D_I^n\left[u_i^n\right] + D_I^{n+1}\left[u_i^n\right]\right) + Q_i \Bigg\} + \left\{\frac{\left(J^{-1}\right)^n}{\left(J^{-1}\right)^{n+1}} - 1\right\} u_i^n \end{aligned} \tag{3.6}$$

The use of time $(n+1)$ metric terms in the source of the predictor, equation (3.6), is allowable as our numerical method solves for the time $(n+1)$ free surface position and the curvilinear grid prior to the solution of the predictor step. The last term in equation (3.6) is applied so that the velocity u_i^n on the left-hand-side of the predictor, equation (3.1), is multiplied by the time $(n+1)$ inverse Jacobian during the derivation of the discrete equations. This prevents the appearance of the time (n) Jacobian in the corrector step and the pressure Poisson equation, and provides for a simpler implementation.

Discrete operators from equations (3.1) through (3.6) are defined as:

$$D_\alpha ( ) = \frac{\Delta t}{2J^{-1}} \frac{\delta}{\delta \xi^\alpha} \left\{ \nu J^{-1} G^{\alpha\alpha} \frac{\delta}{\delta \xi^\alpha} ( ) \right\} \tag{3.7}$$

where $\alpha = 1, 2, 3$ with no summation.

$$D_E ( ) = \frac{\delta}{\delta \xi^q} \left\{ \nu J^{-1} G^{qr} \frac{\delta}{\delta \xi^r} ( ) \right\}_{q \neq r} \tag{3.8}$$

$$D_I ( ) = \frac{\delta}{\delta \xi^q} \left\{ \nu J^{-1} G^{qr} \frac{\delta}{\delta \xi^r} ( ) \right\}_{q = r} \tag{3.9}$$

$$C_i = -\frac{\delta}{\delta \xi^q} \left\{ J^{-1} U^q u_i \right\} \tag{3.10}$$

$$B_i ( ) = -\frac{\delta}{\delta \xi^q} \left\{ J^{-1} S_i^q ( ) \right\} \tag{3.11}$$

$$Q_i = \frac{\delta}{\delta \xi^q} \left\{ \left(J^{-1} \dot{X}^q \right)^{n+\frac{1}{2}} u_i^n \right\} \tag{3.12}$$

In the operator Q_i, we compute $\left(J^{-1} \dot{X}^q \right)^{n+1/2}$ as the volume swept out by the 'q' side of a cell as the grid moves from the time (n) to the time $(n+1)$ positions. An alternative approach is to use equation (2.123) to write the grid flux in terms of the surface metrics (S_j^q) and the motion of the grid points at the centers of a cell face such that:

$$\left(J^{-1} \dot{X}^q \right)^{n+\frac{1}{2}} = \frac{1}{2\Delta t} \left(S_j^{q(n)} + S_j^{q(n+1)} \right) \left(x_j^{n+1} - x_j^n \right) \tag{3.13}$$

The effect of the moving grid is carried in the grid velocity term (Q_i) which is the net contravariant flux of physical space through the sides of a control volume cell as viewed from computational space. The grid flux is a part of the source term for the computation of the intermediate (u^*) velocity, but does not explicitly appear in the pressure Poisson equation or the corrector steps. Therefore, the Poisson solver and corrector steps are only indirectly affected by the moving grid.

3.4.2 Kinematic boundary condition

The curvilinear kinematic boundary condition, derived in section 2.4 is:

$$\frac{\partial \mathcal{H}}{\partial t} = \overline{U}^3 - \overline{U}^1 \frac{\partial \overline{\mathcal{H}}}{\partial \xi^1} - \overline{U}^2 \frac{\partial \overline{\mathcal{H}}}{\partial \xi^2} + \mathcal{L}_1 \left(\overline{U}^1, \overline{\mathcal{H}} \right) + \mathcal{L}_2 \left(\overline{U}^2, \overline{\mathcal{H}} \right) \tag{3.14}$$

To discretize the kinematic boundary condition, we apply a 4th-order Runge-Kutta method for integration in time with the 5th-order upwind differencing of Komori *et al.* (1993) for spatial derivatives. We consider the time (n) curvilinear grid to be *fixed* with respect to time and require that it be boundary-fitted to the time (n) free surface. Thus, at the surface:

$$\mathcal{H}^{[n]} \;=\; \xi^{3[n]}\Big|_{surface} \;=\; \text{constant} \tag{3.15}$$

The gradients of the time (n) free-surface height relative to the ξ^1 and ξ^2 curvilinear coordinates will disappear, since:

$$\frac{\delta \mathcal{H}^{[n]}}{\delta \xi^q} \;=\; \frac{\delta \xi^{3\,[n]}}{\delta \xi^q} \;=\; 0 \qquad : \; q \;=\; 1,2 \tag{3.16}$$

The resulting RK4 discrete system is:

$$k \;=\; \Delta t U^3 \tag{3.17}$$

$$\bar{k} \;=\; \Delta t \left\{ U^3 \;-\; \frac{U^1}{2}\frac{\delta k}{\delta \xi^1} \;-\; \frac{U^2}{2}\frac{\delta k}{\delta \xi^2} \;+\; \frac{1}{2}\mathcal{L}_1\left(U^1, k\right) \;+\; \frac{1}{2}\mathcal{L}_2\left(U^2, k\right) \right\} \tag{3.18}$$

$$\tilde{k} \;=\; \Delta t \left\{ U^3 \;-\; \frac{U^1}{2}\frac{\delta \bar{k}}{\delta \xi^1} \;-\; \frac{U^2}{2}\frac{\delta \bar{k}}{\delta \xi^2} \;+\; \frac{1}{2}\mathcal{L}_1\left(U^1, \bar{k}\right) \;+\; \frac{1}{2}\mathcal{L}_2\left(U^2, \bar{k}\right) \right\} \tag{3.19}$$

$$\hat{k} \;=\; \Delta t \left\{ U^3 \;-\; U^1\frac{\delta \tilde{k}}{\delta \xi^1} \;-\; U^2\frac{\delta \tilde{k}}{\delta \xi^2} \;+\; \mathcal{L}_1\left(U^1, \tilde{k}\right) \;+\; \mathcal{L}_2\left(U^2, \tilde{k}\right) \right\} \tag{3.20}$$

$$\mathcal{H}^{n+1} \;=\; \mathcal{H}^n \;+\; \frac{1}{6}\left(k \;+\; 2\bar{k} \;+\; 2\tilde{k} \;+\; \hat{k}\right) \tag{3.21}$$

where a spatial derivative at location i is discretized as

$$\begin{aligned}\left(U\frac{\delta k}{\delta \xi}\right)_i \;&=\; \frac{U_i}{60}\left\{ k_{i+3} \;-\; 9k_{i+2} \;+\; 45\left(k_{i+1} \;-\; k_{i-1}\right) \;+\; 9k_{i-2} \;-\; k_{i-3} \right\} \\ &+\; \frac{|U_i|}{60}\left\{ k_{i+3} \;-\; 6k_{i+2} \;+\; 15k_{i+1} \;-\; 20k_i \;+\; 15k_{i-1} \;-\; 6k_{i-2} \;+\; k_{i-3} \right\}\end{aligned} \tag{3.22}$$

and the discrete Leonard stress operator is:

$$\mathcal{L}_\alpha\left(U^\alpha, k\right) \;=\; \overline{U^\alpha \frac{\delta k}{\delta \xi^\alpha}} \;-\; \overline{U}^\alpha \frac{\overline{\delta k}}{\delta \xi^\alpha} \tag{3.23}$$

To obtain the time $(n+1)$ physical space position of a particle on the surface after numerical solution of equations (3.17) through (3.21), we note that:

$$\Delta x_i \;=\; \Delta \xi^3 \frac{\partial x_i}{\partial \xi^3} \quad : \quad i \;=\; 1,3 \tag{3.24}$$

which can be discretized as:

$$x_i^{n+1} \;=\; x_i^n \;+\; \left(\mathcal{H}^{n+1} \;-\; \mathcal{H}^n\right) \left(\frac{\delta x_i}{\delta \xi^3}\right)^n \tag{3.25}$$

This system provides a method for explicitly updating the free surface position from time (n) to $(n+1)$. In the simulations presented in chapters 5 through 7, the Leonard stress operators were not used for reasons discussed in section 2.4.3.

3.4.3 Dynamic boundary condition

Tangential components of dynamic boundary condition

The tangential components of the dynamic boundary condition are used to obtain the tangential velocities on (1) the free surface (for use in integrating the kinematic boundary condition), and (2) the numerical ghost points outside the free surface (for use in the boundary conditions on the predictor step of the solution method). We experimented with linear, quadratic, and cubic implementations of the boundary conditions, and found the simple linear approach worked best when the boundary layer was well resolved. This follows the advice of MacCormack (1986) who cautions against the use of higher order interpolations/extrapolations onto a boundary for values that feed back into a simulation. Such extrapolations can amplify numerical oscillations in the solution and erroneously reintroduce the amplified error in following time steps.

The linear approach can be presented as:

$$U^\alpha_{surf} \;=\; U^\alpha_{surf-\frac{1}{2}} \;+\; \frac{1}{2}\frac{\delta U^\alpha}{\delta \xi^3}\bigg|_{surf} \qquad : \qquad \alpha \;=\; 1,2 \tag{3.26}$$

where the subscript notation $(surf)$ indicates the value at the free surface and the notation $(surf - 1/2)$ indicates the value at the center of a cell face for the first cell inside the surface. The gradient of U^α across the boundary is found from a discrete implementation (using central differences) of the tangential dynamic boundary condition, equations (2.158) and (2.159). The U^3 component at the center of the cell face on the free surface is computed directly in the corrector step, equation (3.4), from the computed pressure field and the U^{*^3} value at the surface.

Computation of contravariant velocities on the faces and centers of the ghost cells outside the free-surface is accomplished in a similar fashion. Once the contravariant velocities are computed, the three components of the Cartesian velocity must be calculated at the centers of the ghost cells outside the free surface. These values provide part of the boundary condition for the u^* estimated velocity computation using the approach of Kim and Moin (1985). The Cartesian velocities for each ghost point are obtained by inverting the 3×3 matrix of metrics that relates the Cartesian and contravariant velocities. Since

$$U^q = \frac{\partial \xi^q}{\partial x_i} u_i \qquad \text{summation on } i \text{ only} \tag{3.27}$$

it follows that

$$u_i = \left[\frac{\partial \xi^q}{\partial x_i}\right]^{-1} U^q \qquad \text{summation on } q \text{ only} \tag{3.28}$$

Normal component of dynamic boundary condition

The normal component of the dynamic boundary condition, equation (2.157) is discretized using central difference operators. This provides a Dirichlet condition on the modified physical pressure (p). A subtle point that is overlooked in some of the literature is that the numerical pressure variable (ϕ) is an approximation of the physical pressure that may not be of the same order of accuracy as the solution method. The relationship between the numerical and physical pressure variables is a function of the discretization method, and is given by equation (3.5) for the present approach. In general, it is not mathematically rigorous to simply substitute ϕ for p in the dynamic boundary condition (or any other pressure computation). However, our experience has been that it is in keeping with the order of accuracy of the simulation method for the flows investigated. In test simulations we computed the difference between the right-hand and left-hand sides of the p/ϕ relation, equation (3.5), with the result that the difference was always of the order Δt^2 or smaller. This conclusion was also reached by Calhoun[9], who conducted a more detailed analysis by numerically integrating the p/ϕ relation in a flow with turbulent separation over a wavy boundary.

[9] R. Calhoun, Env. Fluid Mech. Lab., Stanford Univ., personal communication (1996).

The normal component of the dynamic boundary condition is used as a Dirichlet boundary condition on the pressure in the solution of the Poisson pressure equation. To be numerically consistent in the discretization, all the terms in the normal component of the dynamic boundary condition, equation (2.157) should be time $(n+1)$ values. However, unlike a no-slip boundary, the velocity on the free surface at time $(n+1)$ is not known *a priori*. A precise discretization would involve substituting the contravariant corrector step, equation (3.4) into the dynamic boundary condition to change the U^{n+1} values into U^* values plus ϕ^{n+1} gradients. This boundary condition on ϕ includes both ϕ and second derivatives of ϕ, changing the form of the boundary condition and making implementation significantly more complicated (especially for use with a multigrid solver).

Two approaches could be used to obtain formally 2nd-order accurate discretizations of the dynamic boundary condition, equation (2.157), in terms of time (n) variables while retaining a simple Dirichlet pressure boundary form. First, we could use the contravariant corrector step, equation (3.4), along with a Taylor series expansion for the pressure to write:

$$(U^q)^{n+1} \;=\; U^{*q} \;-\; \Delta t \left(\frac{G^{qr}}{J^{-1}} \frac{\delta \phi}{\delta \xi^r} \right)^n \;+\; O\left(\Delta t\right)^2 \tag{3.29}$$

Substituting equation (3.29) into a discretization of equation (2.157) provides a Dirichlet boundary condition on the pressure in terms of the U^* velocities and second derivatives of the time (n) pressure variable. This has the disadvantage of feeding numerical errors in the time (n) pressure computation back into the computation of the time $(n+1)$ pressure boundary condition. Such feedback can induce an undesirable numerical oscillations in the pressure field.

An approach (used in the present work) that does not have feedback of the time (n) pressures into the boundary condition requires a Taylor-series expansion in for each velocity term on the right-hand side of equation (2.157), so that the resulting dynamic boundary condition is:

$$\begin{aligned}
\phi_{S-}^{n+1} &= \frac{z_{s-}}{(Fr)^2} + 2M^{[n+1]}\gamma \\
&+ \frac{1}{Re}\left\{ 2\frac{\delta U^{3\,[n]}}{\delta\xi^3} + \frac{G^{33}}{J^{-1}}\left[U^1\frac{\delta}{\delta\xi^1}\left(J^{-1}G_{33}\right) + U^2\frac{\delta}{\delta\xi^2}\left(J^{-1}G_{33}\right)\right]^{[n]}\right\} \\
&+ \frac{\Delta t}{Re}\Bigg\{ 2\frac{\delta}{\delta\xi^3}\left(U^{3[n]} - U^{3[n-1]}\right) \\
&\qquad + \frac{G^{33}}{J^{-1}}\left(U^{1[n]} - U^{1[n-1]}\right)\frac{\delta}{\delta\xi^1}\left(J^{-1}G_{33}\right)^{[n]} \\
&\qquad + \frac{G^{33}}{J^{-1}}\left(U^{2[n]} - U^{2[n-1]}\right)\frac{\delta}{\delta\xi^2}\left(J^{-1}G_{33}\right)^{[n]}\Bigg\} + O\left(\Delta t\right)^2 \qquad (3.30)
\end{aligned}$$

In our simulations, $1/Re \leq O\left(\Delta t\right)$. Without loss of accuracy, we can neglect terms of order $(\Delta t/Re)$ in our second-order method[10]. The discrete normal dynamic boundary condition becomes:

$$\begin{aligned}
\phi_{S-}^{n+1} &= \frac{z_{s-}}{(Fr)^2} + 2M^{[n+1]}\gamma \\
&+ \frac{1}{Re}\left\{ 2\frac{\delta U^{3\,[n]}}{\delta\xi^3} + \frac{G^{33}}{J^{-1}}\left[U^1\frac{\delta}{\delta\xi^1}\left(J^{-1}G_{33}\right) + U^2\frac{\delta}{\delta\xi^2}\left(J^{-1}G_{33}\right)\right]^{[n]}\right\}
\end{aligned} \qquad (3.31)$$

Implementation of the Dirichlet pressure boundary condition in the multigrid solver is accomplished using a linear approach that is consistent with the linear prolongation/restriction operators of the multigrid method developed by Zang (1993). In Zang's approach, the velocities on physical boundaries are considered to be known *a priori* so pressure boundary conditions are only required when a grid is not boundary-orthogonal[11]. To implement the pressure boundary condition, we first obtain the contravariant estimated velocity normal to the free surface (U^{*3}) using a linear average of the interior and ghost point u_i^* velocities and the boundary metrics. This provides the U^* on the boundary needed for the pressure Poisson equation source term

[10] This is an improvement over the $P = 0$ boundary condition (used by many authors) that is $O\left(1/Re\right)$ or $O\left(\Delta t\right)$ accurate.

[11] Where the boundary velocity is known and the grid is not boundary orthogonal, the pressure boundary condition is an $O\left(\Delta t\right)$ effect.

(see equation [3.1]). The primary difficulty in implementing the pressure boundary condition is that the boundary condition is defined on the edges of computational cells while the pressures in the interior are defined at the centers of computational cells. Our implementation uses the pressure at the center of the first cell inside the boundary and the boundary condition pressure to compute a ghost point pressure using linear extrapolation. The ghost point pressure appears in the discrete stencil for the left-hand-side of the pressure Poisson equation and thus provides for efficient implementation in the multigrid solver.

3.5 Summary of the numerical method

1. Use RK4 and 5th-order upwind discretization of the kinematic boundary condition, equations (3.17) through (3.25) to advance the free surface from time (n) to time $(n+1)$.

2. Compute grid and metrics for time $(n+1)$ grid using the Poisson equation method from the 3DGRAPE/AL code.

3. Solve for u_i^* at center of cells using AB2 with quadratic upwind interpolation (QUICK, Leonard [1979]) discretization for convective terms, CN2 for diagonal viscous terms, and approximate factorization for the implicit solution. This applies a vectorized tridiagonal solver to equation (3.1).

4. Use linear interpolation to obtain the normal component of U^* on each cell face[12].

5. Solve the Poisson pressure equation for the pressure variable ϕ using the 3D vectorized multigrid solver developed by Zang *et al.* (1994) which has been adapted for use with a Dirichlet pressure boundary condition. The normal component of the dynamic boundary condition is used to provide a Dirichlet pressure boundary condition on the free surface.

[12]Higher order interpolation methods were tested, but did not significantly improve the accuracy of the method. Upwind-biased interpolations are a problem for the U^* interpolations since they require a significant number of "IF - THEN" statements in the implementation. This reduces the ability of the interpolation method to be vectorized and slows the computation of each time step.

6. Using the Cartesian corrector step, equation (3.3), compute the time $(n+1)$ Cartesian velocity (u_i) at cell centers.

7. Using the contravariant corrector step, equation (3.4), compute the time $(n+1)$ contravariant velocity components normal to cell surfaces, U^q.

8. Using the tangential components of the dynamic boundary condition, equations (2.158) and (2.159) along with equation (3.26), compute the tangential components of contravariant velocity U^q on free surface and ghost points. This provides the velocities needed to advance the free surface in the next time step. Transform the contravariant velocities into Cartesian velocities for use in the u^* boundary condition in the next time step.

Chapter 4

Numerical implementation issues

4.1 Introduction

The development and implementation of a moving-grid simulation method presents a number of obstacles to the numerical analyst. The solutions to some problems are obvious; in other cases the solutions may only be found through laborious numerical experimentation. This chapter will present a summary of some of the issues encountered in the development of the present simulation method, along with a description of how the problems were addressed.

4.2 Grid storage and interpolation

A complication to the grid generation process is the requirement to produce a double-dense grid for a finite-volume simulation. In general, whether using a staggered or a non-staggered grid in the discretization of the Navier-Stokes equations, some grid metrics are needed on the control volume faces as well as at the center of the control volume cells. This requires the computation of a field of $2N \times 2N \times 2N$ grid points for a simulation with $N \times N \times N$ computational cells. The obvious (but wrong) solution is to compute a single-dense grid and the metrics for the centers of the control volumes, then obtain the metrics on the faces by interpolation from the center metrics. Experience in the numerical grid-generation community has shown this leads to unacceptable errors (Thompson *et al.* [1985]). It is likely that this approach could be successfully applied when combined with higher-order stencils for computing the metrics. However, metric computation in the present work (and most others) is through second-order central differences, which cannot be interpolated and maintain second-order accuracy in the simulation. An acceptable alternative that reduces the grid generation effort is used in the present work: a single-dense grid is generated

using the Poisson method, while the double-dense grid is computed using cubic[1] interpolation from the single-dense grid. Then, the metrics are computed directly on the double-dense grid without interpolation. This substantially reduces CPU time and storage requirements for the grid generation code, without degrading the accuracy of the simulation. Cubic interpolation from the single-dense to the double-dense grid ensures that the grid is smooth while limiting the overall computational effort. On highly stretched grids, it might be necessary to use even higher orders of interpolation. Note that the order of interpolation (for a moderately stretched grid) used to produce the double-dense grid should be at least as high as the order of accuracy of the simulation method. This follows from the grid generation requirement of Thompson *et al.* (1985) that the metrics should be computed using at least of the same order as the differences that are used in the flow solver.

4.3 Grid smoothness at boundary edges

One of the difficulties in generating a good grid for an arbitrary free-surface deformation is that, in general, the free surface is not required to be orthogonal to the boundaries of the domain. Thus, grid generation codes have difficulty near the boundaries as they try to make the interior grid lines orthogonal to the surface, while the boundary grid lines are not. The grid generation process has a choice of either (1) accepting a non-smooth grid near the boundaries while retaining boundary-orthogonality up to the last point, or (2) relaxing the boundary-orthogonality condition and obtaining grid smoothness. These approaches are predicated on the usual assumption of a grid-generation method that the boundaries are fixed *a priori*. Both approaches have a tendency to introduce numerical oscillations in the free surface position at the boundaries. In effect, inlet and outlet boundaries become a source of small amplitude wave forcing which can dominate the turbulent surface deformations.

The simplest approach to controlling this effect is to compute simulations in a wave-following system such that the wave crest or trough is fixed at a boundary. Our experience shows that it is preferable to fix the wave trough at the boundary since it

[1]This ensures that the point locations are more accurate than the desired metric accuracy. It might also be acceptable to use quadratic interpolation, but this could lead to a directional bias in the grid.

has smaller curvature than the crest. A more flexible approach has been developed in the present work for use in simulations with periodic boundary conditions. This new approach is based upon the recognition that the grid motion terms in the discrete Navier-Stokes equations can be applied at a moving boundary in the flow as well as at the moving free-surface. Thus, we are free to adjust periodic inflow and outflow boundaries so that boundary orthogonality and grid smoothness at the free-surface are obtained. The only limitations to this method are (1) the motion of a set of periodic boundaries must be identical, and (2) the grid motion term at the boundaries must be included in the solution method. Although it would be preferable to have the motion of the boundaries computed in the Poisson grid generation method, this did not seem practical for the 3DGRAPE/AL code. Instead, the standard grid generation method was applied using flat surfaces on periodic boundaries, and a *post hoc* adjustment was made to add curvature to the periodic boundaries in order to obtain the desired grid smoothness and boundary orthogonality. The implementation of this method involves interpolating a smooth boundary position from the grid points on the inside of the domain.

4.4 Grid stretching

The "correct" choice of grid spacing depends on (1) the number of grid points in the simulation, (2) the minimum spacing required to resolve the boundary layer, and (3) the maximum allowable stretching of the grid. The first of these is generally fixed by the available computational power; the second is determined by the choice of flow; the third requires numerical investigations to determine the effect of the stretching that results from the input conditions. For the numerical method developed by Zang *et al.* (1994), grid stretching has a direct effect on the truncation error through the multigrid pressure Poisson solver (which uses linear interpolation of metrics between multigrid levels). Grid stretching must be carefully chosen to prevent the truncation error in the multigrid interpolation from becoming higher than second order. Grid stretching problems are manifest in the convergence rate for the multigrid pressure Poisson solver. Inconsistency between the stretched grid metrics and the linear interpolations between multigrid levels causes a small (but different) error signal to be introduced at each multigrid level. Because the error signal is different on each level, the multigrid

method cannot efficiently smooth it out. Additional problems can arise due to ill-conditioning of the pressure Poisson equation. The coefficients of the pressure Poisson equation (see eq. [3.2]) are the G^{qr} metric terms defined in equation (2.124). Grid stretching can result in a matrix for the Poisson solver with large disparities in the magnitude of the metric terms between fine grid and coarse grid areas. This ill-conditioning can result in slow convergence of the Poisson solver.

Rai and Moin (1991) derived the allowable stretching for a grid in a second-order accurate finite-volume code. Their results showed that stretching was limited by a requirement that the second derivative of the grid spacing must be smooth and continuous. Our experience has been that in practice, the grid stretching is also limited by the magnitude of local stretching relative to the grid size. Armfield[2] has shown that a first-order truncation error can arise for stretching that exceeds 10% of the local grid spacing. This limit is consistent with experience of numerical researchers at the Environmental Fluid Mechanics Laboratory at Stanford University[3].

4.5 Wave-following coordinate system

In simulating a turbulent channel flow with a progressive wave, it is desirable to use a wave-following coordinate system to (1) minimize boundary motion for stability reasons described in section 4.9, and (2) so that phase-averages of flow variables at different points in time can be easily compared. Implementation of a wave-following coordinate system requires that the no-slip boundary at the bottom of the domain be given a velocity exactly equal to and opposite of the wave speed. The difficulty in implementing a wave-following coordinate system is that the exact velocity of the wave is not known *a priori*. If we are conducting a simulation of a wave without a current, we can get close to the wave speed using Stokes theory taken to the third order[4]. Unfortunately, obtaining a close approximation of the wave speed does not ensure that the wave will remain fixed in the coordinate system. If the wave does

[2]S. Armfield, Dept. of Mech. Eng., Sydney Univ., Australia, personal communication (1997).

[3]L. Yuan, Env. Fluid Mech. Lab., Stanford Univ., personal communication (1997).

[4]To the second-order, the wave speed using Stokes expansions is identical to first-order theory.

not remain fixed in the domain, then computing temporal means of phase-averages of simulation data requires spatial interpolation since the position of the grid points is changing relative to the wave phase. This problem becomes more complicated when a surface wave is propagating over a turbulent current. Superposition of the wave speed without a current and the bulk current speed provides a rough approximation of the actual wave speed, but again, this is not sufficient to ensure that the wave will remain fixed in the domain.

Our problem is that we need to move the domain at the exact numerical wave speed. The solution taken in the present work is to compute the wave speed at each time step (based on the position of the wave crest) and adjust the bottom boundary speed to coincide with the wave speed. In effect, this is an adjustment of the speed of our moving frame of reference. This approach is potentially a source of trouble since it implies that we are no longer in an inertial reference frame and our governing equations are not formally valid. In actual use, the change in the speed of the reference frame is initially of the order of a few percentage points per time step. Over the first few wave periods, the reference frame undergoes an oscillatory decaying adjustment about the actual wave speed as the wave-induced velocity field adjusts from the irrotational initial conditions. After the first three wave periods, the changes in the reference frame speed in each time step are less than 0.01% of the wave speed. Since the effect remains oscillatory rather than cumulative, the small changes from one time step to the next do not appear to be of any practical significance in the flow computation. When sufficient computational power is available, a fixed coordinate system with a translating wave can be simulated and compared to the wave-following system to quantify the effects of the reference-frame oscillations. This is not currently possible due to stability limitations on simulations using a fixed reference frame (see section 4.9.3).

4.6 Boundary condition on the pressure

In section 2.3.1 we defined the modified pressure as

$$p \equiv \frac{P}{\rho} + g' x_3 \tag{4.1}$$

so that our pressure Poisson solver uses a boundary condition, equation (3.30), which contains: (1) the hydrostatic pressure, (2) the dynamic pressure, and (3) the gravity body force. This approach presents a numerical problem for simulations with finite-amplitude surface waves: the difference between the hydrostatic pressure and the gravitational body force can become the predominant pressure effect in the flow (see equation [2.80] and accompanying discussion). This creates extra work for the Poisson solver as it must propagate the hydrostatic pressure effect through the domain as a part of the solution method.

As demonstrated by Mahadevan (1994), a preferable approach is to separate the dynamic pressure from the hydrostatic pressure and the gravitational force so that the Poisson equation only works on the dynamic pressure. The difference between the hydrostatic term and the gravitational term can be computed directly from the grid. Mahadevan's implementation of this approach was on a simulation domain with vertical grid lines so that the hydrostatic pressure was easily computed at any point in the flow from the height of the point on the surface directly above. The use of a boundary-orthogonal grid complicates the implementation of Mahadevan's method because the curvature of the grid means that there is not a grid point on the surface located exactly over a grid point in the flow. Because the hydrostatic/gravitational force term can be a primary driver for the velocity field in a flow with a finite-amplitude wave, one must be careful of any approximations made in computation of the hydrostatic/gravitational balance. Errors in computing this term can become a direct source of spurious advective motion in the simulation.

Because of these problems, the present work has not implemented Mahadevan's method. However, experience with the slowness of the Poisson solver for the modified pressure leads us to the conclusion that Mahadevan's method would be advantageous. Accurate implementation of Mahadevan's method for three-dimensional motion of the free surface with a boundary-orthogonal grid is not a trivial matter since it requires interpolation of the free surface position to compute a hydrostatic

height for the center of every grid cell in the flow[5]. Linear interpolation of the free surface will introduce an unacceptable error into the hydrostatic term, while quadratic interpolation will introduce a directional bias into the surface interpolation. Local cubic interpolation has the potential of introducing unwarranted numerical oscillations into the free surface. It would seem that the only reasonable approach is to use two-dimensional cubic splines that will ensure the smoothness of the surface and the required order of accuracy. The splines themselves need only be created once (and in fact are already a part of the present code used in the evaluation of the surface motion). The difficult part of the implementation is developing an efficient, vectorizable, computational method to decide which splines are to be used to evaluate the surface position above each point. The simplest algorithm that could be used (checking each point in the volume and searching the surface for the appropriate spline location) would require an order of N^5 operations with "IF - THEN" statements that would not vectorize.

4.7 Conservation of space

In an interior flow simulation, we are typically concerned that our simulation should conserve mass according to the continuity equation. Non-conservation of mass (even at the level of numerical truncation error) results in spurious source terms in the momentum equation that can destroy numerical accuracy. For a simulation with a moving grid we must recognize that we are changing our computational space both on a local scale (as grid cells expand, contract, and translate), and on a global scale (as the free surface moves in response to the flow kinematics). It is obvious that on a global scale, any net change in the total volume of the domain must be exactly balanced by a net inflow or outflow of fluid. It is less obvious that in any change to a local grid cell, the flux of the sides of the grid cell (an important convective term) must be exactly balanced by the change in the volume of the grid cell.

[5]The ability to compute directly the hydrostatic effects on a grid with vertical grid lines is an advantage in the use of a non-boundary orthogonal grid that has not been explored in this research. It would appear that for waves of limited steepness, the ability to easily compute a restricted grid and decompose the hydrostatic pressure term should compensate for the disadvantages caused by the complications in the dynamic and kinematic boundary conditions and grid skewness for a non-boundary-orthogonal system.

4.7.1 Local space conservation

As discussed by Thompson *et al.* (1985) and Demirdzic and Peric (1988), local conservation of space can be derived in the curvilinear form shown in equation (2.116), which could be written in a discrete form as:

$$\frac{\delta}{\delta\tau}\left(J^{-1}\right) \equiv \frac{\delta}{\delta\xi^q}\left(J^{-1}\dot{X}^q\right) \tag{4.2}$$

This requires that the change in the inverse Jacobian of a cell must be identical to the divergence of the grid fluxes. The appropriate implementation of this can be written as:

$$\left(J^{-1}\right)^{n+1} = \left(J^{-1}\right)^{n} + \Delta\tau\frac{\delta}{\delta\xi^q}\left(J^{-1}\dot{X}^q\right)\Bigg|_n^{n+1} \tag{4.3}$$

Once we have computed the grid volume flux $(J^{-1}\dot{X}^q)$ on each cell face for the motion from the time n to $n+1$ grids, the inverse Jacobian is updated from equation (4.3) rather than recomputed directly from the grid. Failure to use this approach introduces a spurious source term into the momentum equation due to the inconsistencies between the computed cell volumes and the grid fluxes. In essence, the local conservation of space requires that any numerical approximation in computing the grid flux must be *exactly* duplicated in the computation of the grid cell volume. If this condition is not met, a moving grid simulation cannot maintain the desired order of accuracy. To demonstrate that our moving grid method is second-order accurate in both space and time, numerical simulations of a decaying vortex (without a free surface) have been conducted and are reported in Appendix A.

4.7.2 Global conservation of volume

In the temporal advance of the free surface using the kinematic boundary condition (described in section 3.4.2) we do not make any global limitations on the movement of the surface. As pointed out by Ferziger[6]: an undesirable numerical event that is not explicitly prevented can be expected to always occur. Indeed, we find that our simulation method may have a tendency to either globally increase or decrease its total volume in each time step in a small amount that is consistent with the numerical error

[6] J.H. Ferziger, Dept of Mech. Eng., Stanford Univ., personal communication (1993).

of the method. We monitored this effect in test simulations and the increase/decrease in each time step was less than $O(10^{-6})$ times the total volume of the grid. If the change in the surface height simply oscillates around the true mean, then it can be argued that global conservation is achieved in the mean. Such local oscillations do not seem to have any significant kinematic effects on the flow field. However, where the surface was deformed by finite-amplitude progressive waves, our method had a tendency to produce a mean rise in the surface elevation. For a 6000 time step simulation, this could result in a 0.6% net change of the mean free surface height for the type of domain simulated in this work. We believe this bias in the free surface advance is due to the upwind biased discretization used in the kinematic boundary condition combined with the natural asymmetry between the forward and trailing sides of a finite-amplitude wave. In simulations of standing waves, and free-surface open-channel flows without finite-amplitude waves, we did not observe accumulations of error in the surface position. Instead, the small error oscillated about the mean.

This problem could be fixed in a formal manner by invoking a Green's function for global volume conservation as part of the governing equations. This creates significant complication in the solution of the kinematic boundary condition, and does not appear to be necessary where an explicit method is used for the free-surface advance[7]. We argue that the change in the mean free surface height is simply a numerical artifact that is in the range of the truncation error for our free-surface advance. Therefore, with no loss in accuracy (and no change in the kinematical behavior of the surface), we are free to adjust the mean position of the free surface *after* the solution of the kinematic boundary condition to obtain global conservation. That is, once we have computed the new free surface position, we can integrate the volume under the surface and compute the global non-conservation. This effect can be distributed evenly over the entire surface so that the surface shape is not changed, merely its mean position. This is done to adjust the time $(n+1)$ surface *before* the time $(n+1)$ velocity field is computed, so the flow solver always sees a wave shape that satisfies the discrete kinematic boundary condition and a flow volume that numerically satisfies a global

[7]Note that for a fully-implicit method where the kinematic boundary condition is solved in an implicitly-coupled system with the Navier-Stokes equations, the Green's function approach would be necessary. This is because the kinematics at the time $(n+1)$ step are being used in the free surface advance, so a change in the global volume is reflected in the kinematics of the time $(n+1)$ flow field.

conservation constraint. This approach is consistent with our study of periodic and wall-bounded motions; the issue is not so clear for non-periodic inflow and outflow boundaries.

4.8 Bottom boundary-layer resolution

The simulation of open-channel free-surface flows with surface waves is not the proper venue for investigating the performance of subgrid-scale turbulence models with regards to resolving boundary-layer structure at the channel bottom. Due to the computational demands of the moving grid method it is simply not possible to apply a large number of grid points and compute simulation runs for a range of conditions and turbulence models as would be required for a complete investigation. For the purposes of the present research, the bottom boundary layer can be considered as simply the mechanism by which we produce sheared turbulence that interacts with the wave field, so that the precision with which the production and dissipation of turbulence in the bottom boundary-layer is simulated is not of primary importance. However, it is worthwhile to record our experience and observations of the performance of the subgrid-scale model as a reference for future work on the interaction of waves and boundary layers.

Our simulations placed the greatest number of grid points in the vertical direction to allow resolution of the shear layers at the free-surface and bottom boundaries. The number of grid points in the free-surface boundary layer varied from 6 to 14 depending on the speed of the wave. For the bottom boundary layer, we were able to place 8 grid cells within five z^+ units (where $z^+ \equiv z/Re_\tau$), and three grid cells within $z^+ = 1$. While this resolution is close to DNS scales in the vertical direction, it was achieved at the expense of spanwise and streamwise resolution. In the spanwise direction the grid spacing was about 17 z^+ units, while the streamwise grid spacing was approximately 34 z^+ units. Similar grid spacing was used by Salvetti *et al.* (1996) in their simulations of decaying turbulence in an open-channel with a rigid, free-slip lid.

The coarse resolution in the spanwise and streamwise directions presented problems in accurately resolving the slow-speed streaks in the bottom boundary layer that are a ubiquitous characteristic of turbulent flat-plate boundaries (see Nezu and

Nakagawa [1993] for a comprehensive discussion of this phenomena). Both experiments and well-resolved DNS simulations have found that the slow-speed streaks in the bottom boundary layer are typically: (1) transient in time, (2) finite in streamwise length (on the order of $1000\,z^+$ units), (3) narrow in the spanwise direction, and (4) spaced approximately 100 z^+ units apart in the spanwise direction. With a spanwise resolution of 17 z^+, the physical spacing of the slow-speed streaks means that they are only five grid cells apart in our computational domain. Given the gradients involved in the slow-speed streaks, it is impossible for a second-order finite-volume method to accurately reproduce this phenomena in a space of only five grid cells.

In reviewing data from the simulations, we noted four problems related to the bottom boundary layer: (1) wide streak spacing, (2) infinite streak length, (3) temporal persistence of streaks, and (4) reduced dissipation. The spanwise spacing of the slow-speed streaks in our simulations was approximately 10 grid cells (170 z^+ units). This is better than the 250 z^+ unit spacing seen in the large-eddy simulations of channel flows by Moin and Kim (1982), and Horiuti (1987), but not as good as the physically-accurate spacing seen in DNS simulations (*e.g.* Kim *et al.* [1987]). The resulting slow-speed streaks were wider than those typically seen in the bottom boundaries of well-resolved channel flows. The coarse resolution (relative to the turbulent scales) in the spanwise and streamwise directions and the short domain length ($1074\,z^+$ units) served to make the slow-speed streaks in our simulation infinite in length. There was insufficient space in the computational domain for the streaks to develop significant spanwise "wandering" as seen in DNS simulations and physical experiments. As a result, when a streak crossed the periodic boundary, it wrapped around on its tail and effectively re-initialized itself. The streaks became infinite in length, temporally persistent, and relatively stable in position.

Along with the instantaneous behavior of the slow speed streaks, we observed that the mean flow was faster than the results of either laboratory experiments or DNS simulation would indicate for our applied pressure gradient. Since our simulation produced the correct wall shear, it appears that the simulated flow is not sufficiently dissipative to match experiments and DNS simulations. The dynamic two-parameter model (DTM) used in this simulation is currently being tested by Shah[8], whose preliminary results show that the model is not sufficiently dissipative. While the

[8]K. Shah, Dept. Mech. Eng., Stanford Univ., personal communication (1997).

results of Salvetti *et al.* (1996) indicate that the DTM handles dissipation in decaying sheared turbulence reasonably, they have shown (Salvetti *et al.* [1997]) that there are differences between their LES and the Pan and Banerjee (1995) DNS that are consistent with Shah's conclusion. Problems in the DTM related to its ability to simulate turbulent production in the boundary layer are under investigation.

Since the focus of the research has not been on the bottom boundary layer effects of subgrid-scale modeling, we have not developed the simulation data that would provide a conclusive evaluation of the DTM subgrid-scale model[9]. However, at this point we can offer a hypothesis as to why the DTM does not seem to be sufficiently dissipative in our simulation. In the DTM, the scale-similarity term dominates the Smagorinsky term in the dynamic modeling process. The scale-similarity term relies on the grid-scale turbulent structures to produce an estimate of the subgrid-scale dissipation and and energy backscatter. Thus it can argued that the ability of the DTM to capture the correct dissipation is a function of the ability of the DTM to obtain the correct grid-scale structures.

In the simulations of Salvetti *et al.* (1996), the DTM was used in a simulation of decaying turbulence where the grid-scale structures were part of the initial conditions[10]. With the correct grid-scale structures the method performed well. However, with a bottom boundary layer, the model must be able to *create* the grid-scale structures as well as dissipate them. We have already seen that a coarse grid cannot generate the correct slow-speed streaks in the bottom boundary layer[11]. To the extent that we understand the production of turbulent structures through the "burst" and "sweep" process, it appears that such production is tied to the slow speed streaks and the development of hairpin vortices (Nezu and Nakagawa [1993]). If our grid resolution is too coarse to obtain the correct number and types of structures in the boundary layer, then we are unlikely to produce the correct numbers and types of

[9] It appears the ongoing work of Shah as well as that of Salvetti may provide the answers.

[10] The initial velocity field for Salvetti *et al.* (1996) was an instantaneous velocity field from the DNS simulation of an open channel flow by Pan and Banerjee (1995).

[11] This is true with any coarse grid and is not related to the turbulence model used.

grid-scale turbulent structures in the flow[12]. This brings us back to the DTM: without the correct grid-scale structures, the DTM is unlikely to produce the correct dissipation. Thus, it would seem that DNS levels of resolution are a requirement for regions of turbulent production when using the DTM.

The success of other subgrid-scale models in obtaining the correct dissipation in channel flows may be a result of their reliance on the Smagorinsky eddy-viscosity model rather than the Bardina-type scale-similarity model. In essence, the DTM may be a victim of its own success. The results of Salvetti *et al.* (1996) leave no doubt that the scale-similarity model can accurately reproduce both turbulent dissipation and backscatter effects for a given grid-scale turbulence structure, and do so better than a Smagorinsky-type model. However, Smagorinsky models are tied directly to the strain rates through an eddy-viscosity rather than to the structure of the strain-rates used in the scale-similarity term. Thus, it can be argued that for a Smagorinsky model to obtain the correct dissipation requires the existence of the strain-rates, but not the coherence of the structures. A Smagorinsky-based model that is not producing enough turbulent structures may still obtain sufficient dissipation through eddy-viscosity amplification of the strain rates in the bottom boundary layer. This would seem to indicate that application of the DTM requires DNS resolution in the regions where turbulent structures are being produced, while allowing coarse resolution in the mean flow where the structures are being dissipated[13]

4.9 Simulation scaling limits

Most numerical analyses proceed by defining the characteristics of the flow to be investigated, followed by a determination of the appropriate grid and time scales suitable to resolve the flow. This in turn leads to a statement of the required computer memory and speed to conduct a simulation. We will take an unconventional approach and first define the capabilities of our computer and then investigate the limits that

[12]This is not to say turbulent structures will not be produced, but rather that there will not be enough of them or perhaps they will be of the wrong scale.

[13]The fine/coarse grid resolution requirement is probably better obtained using domain decomposition methods rather than grid stretching.

this places on our flow simulations. This approach is desirable because a free-surface simulation sees a number of limitations that do not occur in an interior flow, and it is not obvious how the limitations interact. The flow domain, mean flow, wave height and gravitational force can all be varied independently, and it is not a trivial matter to find a combination which can be practically and accurately simulated.

For the present work we had available a Cray J90 mini-supercomputer with 128 mega-words (MW) of memory. Because we shared this computer with a number of other users, there were limitations on the amount of memory we could access. In theory, we had available up to 48 MW, which would allow us to conduct a simulation with approximately $50 \times 50 \times 66$ grid points. The large memory requirements were partly due to the research nature of the code which (for debugging purposes) made it desirable to retain some variables that could have been dropped from memory after use. However, even the elimination of these variables would leave the code demanding almost twice the memory of an equivalent non-moving grid code. The extra storage requirements for two time levels of the grid and metrics is the source of the memory problem for moving grid codes. Future research should include development of innovative methods to handle computation of grid motion so as to minimize the storage required.

In theory we could run simulations as large as $50 \times 50 \times 66$, but in practice we were limited to simulations of $34 \times 34 \times 66$. There were two reasons for this limitation, the first being a matter of computer architecture and system usage: this was the largest grid we could run and keep our memory usage below 16 MW, which allowed us to run our simulations in a queue on the Cray J90 that had fast turnaround times and very little time spent swapped out of memory during job sharing. The second reason to keep our grid scale at the $34 \times 34 \times 66$ scale was a matter of computational time. The combination of the Poisson solver for the pressure term in the Navier-Stokes equations and the Poisson solver for the grid made for a simulation method whose computational-time per time-step increased dramatically as the number of grid points increased. This was a special problem for the grid generation method because the Poisson solution for the grid requires iterative evaluation of the source terms of

the grid Poisson equation[14]. The computational time required for the grid generation method was a function of the number of grid points, the local and global curvatures of the free-surface, and the amount of free surface motion from one time step to the next. Unfortunately, there does not seem to be any practical way around these limits, so larger simulations simply require larger computers.

4.9.1 Summary of scaling limitations

Because our numerical approach is explicit in the convective terms, the simulation scaling is limited by the convective Courant-Friedrichs-Lewy (CFL) condition which can be written in a general form as:

$$\mathcal{U}\frac{\Delta t}{\Delta \mathcal{X}} \; < \; C \tag{4.4}$$

where $\mathcal{U}, \Delta t, \Delta \mathcal{X}$ are velocity, time, and grid scales in the simulation, and C is a constant that is generally $O(1)$, but whose exact value depends upon the numerical method used for the convective terms.

In addition, our simulation scale is limited by: (1) a convective CFL condition on the grid motion; (2) a gravitational stability condition on the minimum Froude number; (3) a minimum grid-spacing requirement to resolve free-surface boundary layer; and (4) a maximum limit on the bulk Froude number. Each of these limits is discussed in detail in the following sections.

4.9.2 Surface wave characteristics

Before considering the simulation scaling limits, it is useful to define the characteristics of large and small-scale progressive waves on a free surface.

Consider an open-channel of depth D with a progressive surface wave having a wavenumber K, a frequency Σ, and an amplitude A that are imposed at $t = 0$ with

[14] A full investigation of convergence rates *vs.* number of grid points has not been completed, but our experience has shown that the number of Poisson iterations required to adequately smooth the grid increases with the number of grid points. Thus, there are more computations per iteration, and more iterations per time step. It could be argued that this is nominally an N^4 relationship.

an appropriate fluctuating velocity field. As the numerical solution develops, smaller parasitic waves (gravity and/or capillary waves depending on scaling) will appear on the surface. The smallest resolvable parasitic waves on the free surface may be said to have characteristic amplitudes a, wavenumbers k, and frequency σ.

The simulation domain is scaled so that the primary waves are gravity waves which obey the dispersion relation:

$$\Sigma^2 = gK \tanh KD \tag{4.5}$$

The secondary waves on the surface may be gravity or capillary waves depending on the wavenumber and surface tension coefficient (γ). These waves are small in relation to the depth so they can be considered deep-water waves, with the dispersion relation:

$$\sigma^2 = \left(gk + \frac{\gamma k^3}{\rho} \right) \tag{4.6}$$

If we non-dimensionalize by a length scale $\mathcal{L}$, the gravitational constant g, a time scale $\mathcal{T}$, and take the velocity scale to be the ratio of the length scale to the time scale, then

$$\Sigma = \frac{\Sigma^*}{\mathcal{T}} \tag{4.7}$$

$$\sigma = \frac{\sigma^*}{\mathcal{T}} \tag{4.8}$$

$$K = \frac{K^*}{\mathcal{L}} \tag{4.9}$$

$$k = \frac{k^*}{\mathcal{L}} \tag{4.10}$$

$$D = D^* \mathcal{L} \tag{4.11}$$

$$\frac{\gamma}{\rho} = \gamma^* \frac{\mathcal{L}^3}{\mathcal{T}^2} \tag{4.12}$$

Froude and Weber numbers can be defined as:

$$Fr \equiv \frac{\sqrt{\mathcal{L}}}{\mathcal{T}\sqrt{g}} \tag{4.13}$$

$$We \equiv \frac{1}{\gamma^*} = \frac{\rho \mathcal{L}^3}{\gamma \mathcal{T}^2} \tag{4.14}$$

Then in non-dimensional terms (but dropping the stars) equations (4.5) and (4.6) become:

$$\Sigma^2 = \frac{K}{Fr^2} \tanh KD \tag{4.15}$$

$$\sigma^2 = \frac{k}{Fr^2}\left(1 + \frac{Fr^2}{We}k^2\right) \tag{4.16}$$

The numerical simulation is governed by three non-dimensional parameters: the Reynolds number, the Froude number, and the Weber number. However, since a clean free surface with a constant surface tension is assumed and the velocity and length scales are fixed by the Reynolds number and the Froude number (see section 4.9.6), the Weber number can be considered a function of the fluid rather than the flow. In a full simulation with gravity and capillary effects it is important to ensure that the Weber number and the Froude number are scaled properly so as not to introduce an unnecessary variable into the problem. This can be done by defining a "relative capillarity" for the smallest resolved waves. The square of the phase speed (c) of a gravity-capillary wave can be written as:

$$(c_{grav-cap})^2 = \frac{g}{k} + \frac{\gamma}{\rho}k \tag{4.17}$$

For a pure gravity wave we have simply:

$$(c_{grav})^2 = \frac{g}{k} \tag{4.18}$$

so the relative capillarity ($\mathcal{C}$) of a particular wave (λ) could be measured as:

$$\mathcal{C}_\lambda \equiv \left(\frac{c_{grav-cap}}{c_{grav}}\right)^2 - 1 \tag{4.19}$$

$$= \frac{\gamma}{\rho g}(k_\lambda)^2 \tag{4.20}$$

In non-dimensional terms:

$$\mathcal{C}_\lambda = \frac{Fr^2}{We}(k_\lambda)^2 \tag{4.21}$$

where $\mathcal{C}_\lambda \gg 0$ indicates a predominantly capillary wave, while $\mathcal{C}_\lambda \to 0$ indicates a predominantly gravity wave.

This allows the dispersion relation for gravity-capillary waves, equation (4.16), to be written as:

$$\sigma^2 = \frac{k}{Fr^2}\left(1 + \mathcal{C}_\lambda \frac{k^2}{k_\lambda^2}\right) \tag{4.22}$$

The subscript λ is used so that the relative capillarity is a function of a particular wavelength that is used for scaling purposes.

4.9.3 Grid motion CFL condition

As might be expected from the formulation of the grid motion in the convective term of the momentum equation (see equation [2.119]), stable numerical solution of the free surface motion has an associated convective CFL condition. The normal approach would be to consider the CFL condition as a limit on the time step and grid size for a particular wave. The difficulty with this approach is that in setting up a simulation of a turbulent channel flow with a wave, both the CFL on the grid (associated with the wave motion) and the CFL of the turbulent flow need to be considered. Finding a simulation set-up that satisfies both conditions for limited computer resources can be a challenge. Our approach is to use the physics of wave theory to determine the maximum wave slope that can be simulated for a given grid resolution and time step. Of course, these are not the limits of the simulation method, but are the practical limits associated with finite computer resources and the convective stability of the method. The limits are derived separately for (1) a fixed reference frame (where the waves translate through the domain) and (2) a wave-following reference frame.

Fixed reference frame with translating waves

Consider the movement of the primary wave in relation to the numerical grid where the wave moves through the domain (rather than a wave-following coordinate system). Because we have formulated the moving-grid problem with a convective grid term, there is a limit on the movement of the grid similar to the velocity CFL number ($\mathcal{U}\Delta t/\Delta\mathcal{X}$). Experience has shown that to avoid instability we require that a grid CFL number be less than 1/4:

$$\dot{\mathcal{X}}\,\frac{\Delta t}{\Delta\mathcal{X}} \;<\; \frac{1}{4} \tag{4.23}$$

where $\dot{\mathcal{X}}$ is the scale of the grid velocity.

Because of the refined mesh near the free surface, this becomes an important limit on the time step for movement of the free surface in a surface-normal direction. The important scale of grid motion can be taken as the surface-normal velocity of the free surface. In terms of a boundary-orthogonal coordinate system where $\mathcal{U}_n$ is the normal velocity and $\mathcal{Z}$ is the surface-normal coordinate:

$$\mathcal{U}_n\,\frac{\Delta t}{\Delta\mathcal{Z}_{min}} \;<\; \frac{1}{4} \tag{4.24}$$

In order to obtain an estimate of the surface-normal velocity we can apply linear wave theory and a little trigonometry to write

$$\mathcal{U}_n \approx A\Sigma \cos\left(\tan^{-1} AK\right) \tag{4.25}$$

or

$$\mathcal{U}_n = \frac{A\Sigma}{\sqrt{1+A^2K^2}} \tag{4.26}$$

where A is the wave amplitude. The introduction of the wave amplitude into our analysis of the grid motion CFL condition allows us to define the maximum wave slope AK that can be simulated with any particular combination of grid spacing and time step. Since we have used linear (small-amplitude) wave theory, this can still be only considered a rough guide to the grid resolution required to simulate a particular wave. For large wave slopes or overturning waves, the grid scale and time step size will undoubtedly be required to be smaller than this analysis would predict.

Applying equation (4.26) to equation (4.24) gives the grid CFL condition in terms of the wave amplitude, wave number and wave frequency:

$$\frac{A\Sigma}{\sqrt{1+A^2K^2}} \frac{\Delta t}{\Delta \mathcal{Z}_{min}} < \frac{1}{4} \tag{4.27}$$

The CFL condition is an approximation so the denominator is ignored (being of O[1]). If the primary waves are not shallow-water waves the hyperbolic tangent term can be dropped from the dispersion relation. The grid CFL can then be written as:

$$A \frac{\sqrt{K}}{Fr} \frac{\Delta t}{\Delta \mathcal{Z}_{min}} < \frac{1}{4} \tag{4.28}$$

This can be seen as a limitation on the wave slope (AK) for a given set of simulation conditions:

$$AK < \frac{Fr\sqrt{K}}{4} \frac{\Delta \mathcal{Z}_{min}}{\Delta t} \tag{4.29}$$

The above inequality shows that for a practical simulation of finite-amplitude waves in a fixed frame of reference (where waves move through the domain), we require a small time step, a large Froude number, or a coarse grid. In section 4.9.4 it is demonstrated that boundary-layer considerations mandate a fine grid in the surface normal direction. In section 4.9.5 it is demonstrated that the Froude number has a maximum size limitation for subcritical flows. It follows that for large wave slopes at a given grid resolution, the time step will have to finer than that required for small wave slopes.

Wave-following reference frame

The previous derivation is applicable to a computational domain that is *not* wave-following, and demonstrates that there are strict time step limitations associated with the fixed frame of reference and progressive wave motion. For a wave-following domain the situation is different since the primary movement of the grid is due to small-scale waves. The previous limitation on the slope of the primary wave becomes a limitation on the slope (ak) of the small-scale waves:

$$ak \;<\; \frac{Fr\sqrt{k}}{4}\frac{\Delta\mathcal{Z}_{min}}{\Delta t}\left(1 \;+\; \mathcal{C}_\lambda\frac{k^2}{k_\lambda^2}\right)^{\frac{1}{2}} \tag{4.30}$$

In practice, this limitation is not severe since the ak of the resolved parasitic waves is small[15]. Thus, the wave-following domain has a significant advantage over the fixed domain with regards to the grid motion CFL. Of course, there is no free lunch: a wave-following system has velocities along the streamlines beneath the wave that are the order of the primary wave speed (c). Thus, the wave-following domain has a convective CFL that can be written as:

$$c\frac{\Delta t}{\Delta\mathcal{X}_{min}} \;<\; O\left(1\right) \tag{4.31}$$

where $\mathcal{X}_{min}$ is a minimum spacing scale in the streamwise coordinate system. Since $c = \Sigma/K$, we can apply the dispersion relation, equation (4.5). Dropping the hyperbolic tangent term (which is generally O[1]), this becomes

$$\sqrt{\frac{g}{K}}\frac{\Delta t}{\Delta\mathcal{X}_{min}} \;<\; O\left(1\right) \tag{4.32}$$

In non-dimensional terms:

$$\frac{1}{Fr\sqrt{K}}\frac{\Delta t}{\Delta\mathcal{X}_{min}} \;<\; O\left(1\right) \tag{4.33}$$

This is a limitation on the minimum Froude number in a wave-following system:

$$Fr_{(wave-following)} \;>\; \frac{1}{\sqrt{K}}\frac{\Delta t}{\Delta\mathcal{X}_{min}} \tag{4.34}$$

[15] If parasitic waves are steep or near breaking (as may be the case with pure capillary waves on a wind-ruffled surface) extremely fine grid resolution in the streamwise direction will be required to resolve the capillary waves, and a grid CFL with respect to the streamwise coordinate spacing must be derived. This will result in an extremely small time step.

For comparison to a fixed domain with translating waves, equation (4.29) could be written as:

$$Fr_{(fixed)} > 4A\sqrt{K}\frac{\Delta t}{\Delta \mathcal{Z}_{min}} \tag{4.35}$$

The ratio of equation (4.34) to equation (4.35) is:

$$\frac{Fr_{(limit\,wave-following)}}{Fr_{(limit\,fixed)}} = \frac{1}{4AK}\frac{\Delta \mathcal{Z}_{min}}{\Delta \mathcal{X}_{min}} \tag{4.36}$$

For resolution of small parasitic waves with $\Delta\mathcal{X}$ and $\Delta\mathcal{Z}$ of similar magnitudes, the R.H.S. of equation (4.36) may be greater than 1, indicating the fixed domain is preferable (having a lower limit on the minimum Froude number). Conversely, for simulating large waves with $\Delta\mathcal{X} \gg \Delta\mathcal{Z}$ (as in the present work), a wave-following system is preferred.

Gravitational stability condition

An important consideration in a free-surface simulation is the response of the numerical method to small perturbations of the free surface. In particular, it can be shown that small-amplitude, short-wavelength perturbations can cause numerical instabilities due to violation of the grid motion CFL condition unless the simulation parameters are chosen to satisfy a gravitational stability limit. This applies whether the simulation is in a wave-following or a fixed coordinate system.

Consider a flat free surface with a non-dimensional vertical coordinate spacing of Δz^*. The surface is given a small perturbation with an amplitude (a) and a non-dimensional wavelength of $2\Delta x^*$. Then for a predominantly gravity wave, equation (4.30) can be written as

$$\frac{(\Delta t^*)^2}{\Delta x^*} < \left(\frac{\Delta z^*}{4a}\right)^2 \frac{Fr^2}{\pi} \tag{4.37}$$

Again, we can write this as a lower limit on the Froude number such that:

$$Fr^2 > \pi \left(\frac{4a}{\Delta z^*}\right)^2 \frac{(\Delta t^*)^2}{\Delta x^*} \tag{4.38}$$

This implies that the minimum Froude number has a dynamic dependence on the size of small perturbations to the free surface. If we set the Froude number of the simulation, then the condition on the allowable perturbation amplitude is:

$$a^2 < \frac{1}{16\,\pi} Fr^2 \Delta z^2 \frac{\Delta x^*}{(\Delta t^*)^2} \tag{4.39}$$

Thus, if our flow causes a local $(2\Delta x)$ perturbation with an amplitude greater than allowed by equation (4.39), it can be expected that the grid motion CFL condition will be violated and the simulation may become unstable.

It is instructive to consider a perturbation having an amplitude of one-fourth the vertical grid spacing so that $a \approx \Delta z^*/4$. Then equation (4.38) can be written as

$$Fr^2 > \pi \frac{(\Delta t^*)^2}{\Delta x^*} \tag{4.40}$$

If we return this equation to dimensional terms, we find that

$$\frac{1}{g} > \pi \frac{(\Delta t)^2}{\Delta x} \tag{4.41}$$

which we can write as:

$$g \frac{(\Delta t)^2}{\Delta x} < \frac{1}{\pi} \tag{4.42}$$

This form is similar to the general forms for the viscous stability condition and the convective CFL conditions, which are typically written as:

$$\nu \frac{\Delta t}{(\Delta x)^2} < O(1) \tag{4.43}$$

$$u \frac{\Delta t}{\Delta x} < O(1) \tag{4.44}$$

The former applies to numerical simulations that treat the viscous terms explicitly, while the latter applies to numerical simulations that treat the convective terms explicitly. From this viewpoint, the response of the simulation to small-amplitude perturbations can be viewed as being controlled by a gravitational stability condition when the free-surface is advanced using an explicit method[16].

In actual application, the gravitational stability condition is a function of the Froude number, grid size, time step, and the small-scale perturbations of the free-surface. If there are no small-scale perturbations, then the gravitational stability condition can easily be met. However, if turbulence in the flow or numerical oscillations in the solution induce small-scale perturbations, then the gravitational stability

[16] Where a fully-implicit method is used so that the free-surface motion, flow solution, and grid generation are solved as a coupled set of equations, the gravitational stability condition would not apply since it is based on the grid motion CFL condition (which is only applicable for explicit free-surface advance).

number can become important. Because of the dynamic nature of the stability limit illustrated in equation (4.39), it is possible to have a seemingly stable simulation suddenly become unstable due to a violation of the gravitational stability condition. Such an occurrence requires that the time-step, grid spacing, or Froude number be modified to obtain a stable simulation.

Stability effects of wave decay

There is one further limitation on grid motion that can occur in a wave-following system: since viscosity causes waves to decay, the relationship between the rate-of-decay and the time and grid scales must not violate the grid CFL condition. From Lamb's (1932) presentation of first-order theory, the amplitude decay-rate for viscous effects is:

$$\frac{\partial A}{\partial t} = -2K^2 \nu A \tag{4.45}$$

If we define the Reynolds number in terms of the same scales used in section 4.9.2, then

$$Re = \frac{\mathcal{L}^2}{\nu \mathcal{T}} \tag{4.46}$$

It follows from the grid CFL limitation that (in non-dimensional terms)

$$\frac{2K^2 A}{Re} \frac{\Delta t}{\Delta \mathcal{Z}_{min}} < \frac{1}{4} \tag{4.47}$$

or

$$AK < \frac{Re}{8K} \frac{\Delta \mathcal{Z}_{min}}{\Delta t} \tag{4.48}$$

This limitation is typically not constraining since Re is generally large. However, for application of this method to highly viscous flows this restriction could become important.

4.9.4 Boundary layer thickness

The thickness of the free-surface boundary layer is generally approximated by β^{-1}, where $\beta = (\sigma/2\nu)^{1/2}$, (see Lamb [1932], Longuet-Higgins [1953], or Batchelor [1967] for derivation of β). The smallness of β^{-1} presents challenges for both physical and numerical experiments. We would like to get at least five grid points in the free-surface

boundary layer, so:

$$\Delta \mathcal{Z}_{min} \; < \; \frac{1}{5} \left(\frac{2\nu}{\sigma} \right)^{\frac{1}{2}} \tag{4.49}$$

where $\mathcal{Z}_{min}$ is the minimum spacing of the surface-normal coordinate in physical space.

Using equations (4.15), and (4.22), the dispersion relations for both the primary wave (A, K, Σ), and the smallest resolved waves $(a_\lambda, k_\lambda, \sigma_\lambda)$, it follows (in non-dimensional terms) that either:

$$\Delta \mathcal{Z}^*{}_{min} \; < \; \frac{1}{5} \left(\frac{4\, Re^2\, Fr^2}{K \tanh KD} \right)^{\frac{1}{4}} \tag{4.50}$$

or:

$$\Delta \mathcal{Z}^*{}_{min} \; < \; \frac{1}{5} \left(\frac{4\, Re^2\, Fr^2}{k_\lambda \left\{ 1 \, + \, \mathcal{C}_\lambda \right\}} \right)^{\frac{1}{4}} \tag{4.51}$$

These relations can also be written as lower limits on the Froude number allowed while maintaining adequate boundary-layer resolution.

$$Fr \; > \; \frac{25}{2} \frac{\left(\Delta \zeta_{min} \right)^2}{Re} \left(K \, \tanh KD \right)^{\frac{1}{2}} \tag{4.52}$$

$$Fr \; > \; \frac{25}{2} \frac{\left(\Delta \zeta_{min} \right)^2}{Re} \left(k_\lambda \left\{ 1 \, + \, \mathcal{C}_\lambda \right\} \right)^{\frac{1}{2}} \tag{4.53}$$

Note that as k_λ gets larger (*i.e.* the smallest wavelength gets smaller), it becomes more difficult to resolve the free surface boundary layer. Since k_λ is related to the horizontal resolution of the simulation method, it follows that to resolve the viscous effects of the smallest resolvable waves becomes an increasingly difficult task as the grid is refined the streamwise direction.

4.9.5 Bulk Froude number restriction

In setting up the relationships between the governing parameters in an open-channel flow, we want to keep the bulk Froude number (Fr_b) subcritical to prevent the development of a bore. In one experiment we did run at a supercritical bulk Froude number. As the surface began to develop into a bore, the steep surface deformation could not be adequately resolved with the modest number of grid points we had available. To compute a smooth grid, the grid generation code was force to reduce the

grid spacing at the front of the bore and eventually the CFL condition was violated. From this experience we believe that our numerical method is suitable for use with supercritical flows as long as sufficient streamwise resolution and a small enough time step can be maintained.

The bulk Froude number is defined as:

$$Fr_b = \frac{u_b}{\sqrt{gD}} \tag{4.54}$$

where D is the domain depth, and u_b is the bulk velocity. The bulk velocity for a combined wave/current system in an open-channel flow can be defined as:

$$u_b \equiv \frac{1}{D\,\lambda}\int_{x=0}^{x=\lambda}\int_{z=0}^{z=\eta} u_1\,dx\,dz \tag{4.55}$$

where λ is the wavelength and η is the height of the free surface measured from the bottom of the channel. From equation (4.54) it follows that

$$g > \frac{u_b^2}{D\,Fr_b^2} \tag{4.56}$$

where Fr_b represents a limit on the bulk Froude number which is imposed to keep the simulation subcritical. If we apply the definition of our simulation Froude number based on the length and time scales $\mathcal{L}$ and $\mathcal{T}$ (equation [4.13]), and define the non-dimensional depth and bulk velocity as:

$$u_b \equiv \frac{\mathcal{L}}{\mathcal{T}}u_b^* \tag{4.57}$$

$$D \equiv \mathcal{L}D^* \tag{4.58}$$

then we can write equation (4.56) as

$$Fr < Fr_b\frac{\sqrt{D^*}}{u_b^*} \tag{4.59}$$

to ensure subcritical flows, we set the limit on the bulk Froude number (Fr_b) equal to one and write:

$$Fr < \frac{\sqrt{D^*}}{u_b^*} \tag{4.60}$$

4.9.6 Setting up a practical simulation

By applying the scaling limits derived in the previous sections one can obtain a simulation that is stable and practical. By now it may be obvious why it is preferable

to first pick a time step and a grid size then determine the flow that can be computed. If both the flow Reynolds number and Froude number are selected first, then one has the tedious task of trying to find a combination of time step size and grid spacings that do not violate any of the minimum Froude number limits, the maximum Froude number limit, or the CFL condition. However, if the grid size and time step are chosen to fit a desired Reynolds number, then it is a straightforward matter to compute the Froude number limit for each effect. One then arrives at a range of Froude numbers that can be practically simulated with the chosen domain, computational scales, and flow.

Unfortunately, in translating practical, non-dimensional flow scales back into a physical domain, it becomes apparent that for the present work we do not currently have the computational power to simulate a true water flow at normal laboratory scales[17]. The problem with resolving laboratory scale flows lies in the relationship between the Reynolds number and the Froude number. The Reynolds number is a function of the fluid properties and the flow conditions, while the Froude number is a function of the gravitational field and the flow conditions. Once the velocity and length scale of a flow have been decided, there is a fixed 'physical' relationship between the appropriate Reynolds number and Froude number, just as there is a fixed relationship between the viscosity of a fluid and gravity.

When the only non-dimensional parameter in a flow is the Reynolds number, it can be argued that a simulation at a fixed Re covers a range of physical scales. So a low Reynolds number is either due to a slow velocity flow with a large length scale, or a high speed flow with a small length scale (or some combination between these limits). However, when the Froude number becomes a parameter, it eliminates a degree of freedom so as to limit the range of applicability of a particular simulation. If we non-dimensionalize the Navier-Stokes equations using a single length scale, a single velocity scale, and a time scale that is their quotient, then our equations admit two non-dimensional groups, the Froude number and the Reynolds number:

$$Re = \frac{\mathcal{U}\,\mathcal{L}}{\nu} \tag{4.61}$$

[17] "Laboratory scales" might be reasonably defined as wavelengths and depths on the order of 0.1 m to 1.0 m. With the computational power currently available for the present work we are at the very low end of this range.

$$Fr = \frac{\mathcal{U}}{\sqrt{g\,\mathcal{L}}} \tag{4.62}$$

If we fix the Reynolds number, we find that the velocity scale $\mathcal{U}$, in terms of the viscosity, Reynolds number and length scale ($\mathcal{L}$), is:

$$\mathcal{U} = \frac{\nu\, Re}{\mathcal{L}} \tag{4.63}$$

The Froude number is then

$$Fr = \frac{\nu\, Re}{\mathcal{L}^{\frac{3}{2}} g^{\frac{1}{2}}} \tag{4.64}$$

Since viscosity and gravity have physical values and the Reynolds number has been fixed, the above equation implies that setting the Froude number serves to fix the length scale of the flow (or fixing the length scale serves to fix the Froude number). So a water flow that is defined by both a Reynolds number and a Froude number has exactly one physical length scale for which the flow is representative.

For simulation purposes, a sufficiently low Reynolds number is an Re_τ (based on the boundary shear velocity, u_τ) of $O(100)$. Unfortunately, a low Reynolds number in a laboratory water flow with a length scale $O(0.1\,m)$ implies a velocity scale of $O(10^{-3}\,m/s)$. Since gravity is $O(10\,m/s^2)$, the corresponding Froude number is $O(10^{-3})$. This illustrates the importance of the lower limits on the Froude number developed in the previous sections. As long as we are computationally constrained to the simulation of flows at low Reynolds numbers, the Froude number based on the turbulent scaling will be small[18]. This implies that gravity is extremely strong relative to the turbulent scales of the flow.

In the simulations presented in chapters 5 through 7 of this dissertation, the maximum physical wavelength that could be simulated was a 0.19 m wave on a domain depth of 0.03 m. Simulation of 50 wave periods of this wave took approximately one week on a Cray J90 mini-supercomputer. The simulation wave is about half of the wavelength of the smallest wave in the laboratory experiments of Cowen (1996), with a depth about one-third of the experimental flume. Simulations of longer waves (on a proportionally deeper domain) can be accomplished with the present method by

[18]Note that in the development of the simulation scaling arguments we did not specify what length scale and time scale were to be used. Thus, as long as the scales are applied consistently, we are free to use the turbulent scales of motion (or any other scales) to compute the flow Froude number and determine the upper and lower bounds based a chosen time step and grid size.

decreasing the simulation Froude number and decreasing the time step. Thus, the real limit on the ability to simulate laboratory-scale waves is the amount of computational power one is able to apply to a problem.

4.10 Foundations for an LES model of the kinematic boundary condition

The development and validation of a subgrid-scale model for the cross terms and Reynolds terms of the kinematic boundary condition is an interesting challenge that is a likely candidate for a Ph.D. dissertation on its own. To develop such a model, the dynamic boundary condition will have to be considered in conjunction with the kinematic boundary condition. When the kinematic boundary condition is viewed as only one part of the total boundary condition, it becomes obvious the advective subgrid-scale effects of the kinematic boundary condition cannot be modeled alone.

4.10.1 Linking the boundary conditions

The chain rule of partial differentiation and the continuity equation can be used to write the kinematic boundary condition, equation (2.137) in a quasi-conservative form:

$$\frac{\partial \mathcal{H}}{\partial t} = U^3 - \frac{\partial}{\partial \xi^1}\left(U^1 \mathcal{H}\right) - \frac{\partial}{\partial \xi^2}\left(U^2 \mathcal{H}\right) + \mathcal{H}\frac{\partial U^3}{\partial \xi^3} \tag{4.65}$$

This provides a new term, the product of the surface height and the surface normal gradient of the U^3 velocity. This velocity gradient is the basis of the viscous term in the dynamic boundary condition (see section 2.5), and therefore provides a linkage between the two equations. If we neglect the gradients of the covariant metric G_{33} in equation (2.157), then the kinematic boundary condition could be written as:

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial t} = U^3 &- \frac{\partial}{\partial \xi^1}\left(U^1 \mathcal{H}\right) - \frac{\partial}{\partial \xi^2}\left(U^2 \mathcal{H}\right) \\ &+ \mathcal{H}\, Re \left\{ \frac{p_{s-}}{2} - \frac{x_3}{2\,(Fr)^2} - \frac{M}{We} \right\}\end{aligned} \tag{4.66}$$

Filtering this equation provides nonlinear terms that include the surface pressure, displacement and curvature. The dynamic boundary condition term in equation (4.66)

is made up of the dynamic pressure at the free surface ($p_s - x_3/Fr^2$) and the surface tension effect (M/We). These terms are expected to be small relative to the other terms in the equation, but they are multiplied by the Reynolds number (which is large), and so cannot be neglected.

4.10.2 Comparison of different equation forms

The quasi-conservative form of the kinematic boundary condition provides a basis for another approach to modeling the subgrid-scale terms in the kinematic boundary condition. The filtered form of equation (4.65) is

$$\frac{\partial \overline{\mathcal{H}}}{\partial t} = \overline{U}^3 - \frac{\partial}{\partial \xi^\alpha}\left(\overline{U}^\alpha \overline{\mathcal{H}}\right) + \overline{\mathcal{H}}\frac{\partial \overline{U}^3}{\partial \xi^3} + \Lambda_3^3 + \frac{\partial \Gamma^\alpha}{\partial \xi^\alpha} \quad : \quad \alpha = 1,2 \tag{4.67}$$

where

$$\Lambda_3^3 \equiv \overline{\mathcal{H}\frac{\partial U^3}{\partial \xi^3}} - \overline{\mathcal{H}}\frac{\partial \overline{U}^3}{\partial \xi^3} \tag{4.68}$$

$$\Gamma^\alpha \equiv \overline{U^\alpha \mathcal{H}} - \overline{U^\alpha}\,\overline{\mathcal{H}} \tag{4.69}$$

One can also manipulate the non-conservative filtered kinematic boundary condition, equation (2.138), into the quasi-conservative form:

$$\frac{\partial \overline{\mathcal{H}}}{\partial t} = \overline{U}^3 - \frac{\partial}{\partial \xi^\alpha}\left(\overline{U}^\alpha \overline{\mathcal{H}}\right) + \overline{\mathcal{H}}\frac{\partial \overline{U}^3}{\partial \xi^3} + \chi_\alpha^\alpha \quad : \quad \alpha = 1,2 \tag{4.70}$$

A comparison of equations (4.67) and (4.70) leads to the identity

$$\chi_1^1 + \chi_2^2 \equiv \Lambda_3^3 + \frac{\partial \Gamma^1}{\partial \xi^1} + \frac{\partial \Gamma^2}{\partial \xi^2} \tag{4.71}$$

This demonstrates that the two approaches (non-conservative and quasi-conservative) should produce the same subgrid-scale effects using different terms. It remains to be seen whether test filtering of one or both of these equations in conjunction with the dynamic boundary condition will produce a scale-similarity method for dynamic modeling the cross and Reynolds terms of the kinematic boundary condition.

4.10.3 Linear wave theory

One interesting avenue for the development of a subgrid-scale model for the kinematic boundary condition is the use of linear wave theory. If subgrid-scale effects are small

enough to be modeled using linear wave theory[19], it follows that the subgrid-scale deformation, h, and subgrid-scale velocity u^α (from equations [2.140] and [2.141]) are:

$$h = a\cos(k\xi) \tag{4.72}$$

$$u^\alpha = \frac{gak}{\sigma}\cos k\xi \tag{4.73}$$

where a is a subgrid-scale wave amplitude, k is a subgrid-scale wave number, σ is the subgrid-scale frequency, g is gravity, and ξ is a distance measure from the start of a subgrid-scale wave. For ease of illustration, we shall neglect capillary effects, and develop a subgrid-scale model in terms of gravitational waves. The subgrid scale frequency is:

$$\sigma = \sqrt{gk} \tag{4.74}$$

where we have assumed any subgrid-scale deformation is a deep water wave. This gives us:

$$u^\alpha = a\sqrt{gk}\cos kx \tag{4.75}$$

The modified Reynolds term for the kinematic boundary condition, equation (2.145) can be written as:

$$R^\alpha_\alpha = \overline{a^2k^{\frac{3}{2}}g^{\frac{1}{2}}\cos(k\xi)\,\sin(k\xi)} - \left\{\overline{ak^{\frac{1}{2}}g^{\frac{1}{2}}\cos(k\xi)}\right\}\left\{\overline{ak\sin(k\xi)}\right\} \tag{4.76}$$

Similar derivations can be made for the cross term, equation (2.144). The importance of this approach lies in recognizing that the subgrid-scale deformation and velocity can be changed into an amplitude and a wavenumber (or length). This simplifies the search for an appropriate subgrid-scale model. If we assume that the subgrid-scale effects are dominated by the largest wavelengths that are not resolvable on the grid, then we can say $k \approx \pi/\Delta x$. The modeling problem is reduced to finding an appropriate amplitude (a) of the subgrid-scale effects.

Further work is this area is necessary to find an appropriate approach to dynamically modeling the subgrid-scale effects in the kinematic boundary condition. Any attempt to develop a sophisticated subgrid-scale model for the kinematic boundary condition needs to be done in tandem with well-resolved DNS experiments for identical flows. This level of development is beyond the scope of the current work.

[19] This is not necessarily a good assumption since small capillary waves may be very steep and may be breaking on a microscale. However, understanding and parameterizing such effects requires DNS-scale simulations.

Chapter 5

Simulation approach

5.1 Introduction

The simulation method developed in the preceding chapters is suitable for simulating almost any type of free-surface flow (assuming the availability of sufficient computational power). Our particular interest in this dissertation is in examining the interaction of a sheared turbulent velocity field with a progressive wave in and near the free-surface boundary layer. This has application to a wide number of flows in ocean, estuarine, and engineering systems. The most convenient way to generate a flow field of sheared turbulence is by using a turbulent bottom boundary layer. Our interest is not focused on the bottom boundary layer *per se*, but in the manner in which turbulence generated by the boundary layer interacts with a progressive wave in the near-surface region,

This chapter examines: (1) validation of the numerical method, (2) set-up of numerical experiments, (3) data collection, and (4) analysis techniques that are applied in chapter 6 to investigate the interactions of waves and sheared turbulence in an open-channel flow.

5.2 Validation of the numerical method

The simulation of unsteady, viscous, turbulent flows with a free surface is a computationally demanding exercise that provides ample room for the introduction of coding errors or degradation of accuracy due to poor selections of numerical algorithms. As such, the ability of free-surface method to simulate simpler flows (both free surface and internal flows) needs to be demonstrated before results of the more complicated simulations can be believed.

Our simulation method is built around the Navier-Stokes solver developed by Zang *et al.* (1994), which has been implemented for and tested on a number of different flows by researchers at the Stanford Environmental Fluid Mechanics Laboratory. Zang (1993) includes tests with a decaying vortex, flow under a lid-driven cavity, flow over a backward-facing step, duct flow with a 90 degree bend, and an axisymmetric cavity flow. Ongoing work with Zang's method at the laboratory includes turbulent flow over topography, simulation of a turbulent jet in a cross-flow, and upwelling in a rotating stratified annulus. From the work of Zang (1993) we can consider the fundamental Navier-Stokes solver to be proven and without need of further demonstration. However, there are two major parts to the present addition to Zang's method whose accuracy needs to be demonstrated: (1) the ability to correctly simulate a flow on a moving grid, and (2) the ability to correctly simulate flow with three-dimensional free surface motion. Validation of these abilities is presented in the appendices of this dissertation[1]. The accuracy of the grid movement algorithm is demonstrated using the decaying vortex flow with an arbitrary moving grid (Appendix A). The free-surface method is demonstrated in two space dimensions for a standing wave problem (Appendix B), and in three space dimensions for the oscillations of two standing waves superposed at a 90 degree angle (Appendix C).

5.3 Approach

5.3.1 Simulation set-up

Our simulations of turbulent channel flows with progressive surface waves are conducted on a three-dimensional rectangular domain with a free surface on the upper boundary. The dimensions of the domain are $2\pi \times \pi \times 1$ when non-dimensionalized by the channel depth. A typical simulation domain of $34 \times 34 \times 66$ grid points is shown in figure (5.1). In this figure it may appear that the coordinate system has vertical grid lines, but this is not the case. The curvature of the vertical grid lines to obtain boundary-orthogonality occurs in the near-surface region where the grid is compressed to resolve the free surface boundary layer. On any scale large enough to

[1]The results in the appendices are culled from Hodges *et al.* (1996) and Hodges and Street (1996).

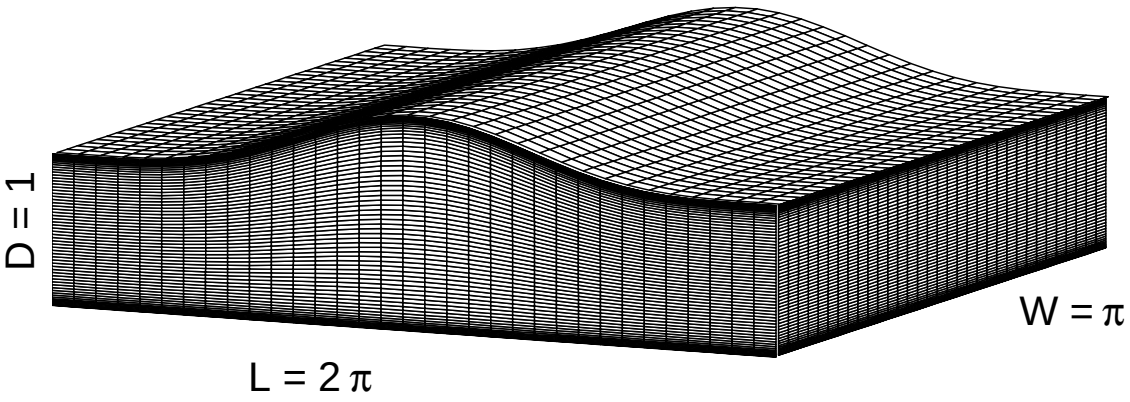


Figure 5.1: Computational domain.

view the entire domain, the curvature of the vertical grid lines seems to disappear.

The boundary conditions on the flow are (1) periodic in the streamwise and spanwise directions, (2) Dirichlet on the bottom boundary, and (3) the nonlinear kinematic and dynamic boundary conditions on the free surface. The Dirichlet condition on the bottom boundary is enforced as a no-slip condition on a boundary that is moving at the wave speed. The turbulent flow through the domain is driven by a streamwise pressure gradient, computed from the relation:

$$\frac{d\Psi}{dx_1} = -\frac{u_\tau^2}{D} \tag{5.1}$$

where u_τ is the shear velocity (or friction velocity) at the bottom boundary, D is the depth of the domain, and Ψ is the modified pressure discussed in section 2.3.1. The Reynolds number of the flow based on the bottom boundary shear is defined as

$$Re_\tau = \frac{u_\tau D}{\nu} \tag{5.2}$$

This dissertation examines three different nonlinear waves, each propagating over a turbulent current produced with the same (non-dimensional) streamwise pressure gradient. Each wave has a wave slope (or ak) of 0.18. For consistency with the work of Cowen (1996) and convenience of notation we refer to the ak as ε. In physical space, the waves are of different wavelengths and therefore propagate at different velocities. In a non-dimensional wave-following coordinate system, all the

waves are of the same wavelength, so the Froude number is changed to obtain different wave velocities[2]. An additional simulation of the turbulent current without a surface wave was conducted for comparison purposes. Tables (5.1), and (5.2) provide the simulation wave data in non-dimensional terms and in physical space.

The initial conditions for our simulations are (1) an instantaneous turbulent open-channel flow velocity field for case C, an open-channel flow without a surface wave, and (2) a linear combination of the turbulent current with a second-order Stokes wave profile and velocity field for the wave/current simulations, $W1$, $W2$, and $W3$. The velocity field for the open-channel flow was developed from an instantaneous velocity field produced in the DNS simulations of Pan and Banerjee (1995). This data was developed in a simulation with a rigid, shear-free lid, so we started by running a simulation with a similar upper boundary condition. When the mean flow appeared to have converged[3], the free surface boundary conditions were invoked and the simulation was run until it reached a statistically steady state. An instantaneous velocity field from the end of this simulation was used in the initial conditions for all the simulations reported in this dissertation.

5.3.2 Comparison to laboratory experiments

Cowen (1996) conducted a series of laboratory experiments studying turbulent open-channel flows with surface waves that have some similarities with the waves in the present numerical simulations. A comparison of the dimensional parameters[4] for the simulations and experiments is provided in Table (5.3). It can be seen that the simulation dimensions are close to being comparable to the TWCF experiments of Cowen (1996). The RWCF experiments used a significantly deeper laboratory flume so that comparisons with the simulations are more difficult. As discussed in section 4.9.6, the major difficulty in comparing free-surface numerical simulations

[2]The wave velocity is simply the negative of the bottom boundary velocity in a wave-following system, so the simulations have different bottom boundary velocities.

[3]As discussed in section 4.8, the mean flow converged at a larger bulk velocity than the Pan and Banerjee (1995) simulation due to insufficient dissipation in the subgrid-scale model.

[4]The dimensions in table (5.3) have been rounded off for ease of comparison. Exact values are found in table (5.2) and Cowen (1996).

	Current (C)	Wave 1 ($W1$)	Wave 2 ($W2$)	Wave 3 ($W3$)
PARAMETERS:				
Re_τ	171	171	171	171
Fr	0.0192	0.0333	0.0192	0.0105
DOMAIN:				
length (L)	2π	2π	2π	2π
width (W)	π	π	π	π
depth (D)	1	1	1	1
FLUID:				
viscosity (ν^*)	0.00585	0.00585	0.00585	0.00585
gravity(g^*)	2700	900	2700	9000
WAVE:				
(theory, w/o current)				
speed	n/a	26.2	45.3	82.8
frequency	n/a	26.2	45.3	82.8
period	n/a	0.234	0.138	0.0759
amplitude (a)	n/a	0.18	0.18	0.18
wave number (k)	n/a	1	1	1
wavelength (λ)	n/a	2π	2π	2π
slope (ε or ak)	n/a	0.18	0.18	0.18
β	n/a	47.3	62.2	84.1
δ (or β^{-1})	n/a	0.0211	0.0161	0.0119
CURRENT AND WAVE:				
(numerical results)				
U_{bulk}	19.19	19.49	19.59	19.79
wave speed (C)	n/a	46.5	67.1	113.
wave frequency (σ)	n/a	46.5	67.1	113.
wave period (T)	n/a	0.135	0.0937	0.0555
β	n/a	63.0	76.7	98.3
δ (or β^{-1})	n/a	0.0159	0.0132	0.0101

Table 5.1: Non-dimensional open-channel simulation parameters.

	Current (C)	Wave 1 ($W1$)	Wave 2 ($W2$)	Wave 3 ($W3$)
PARAMETERS:				
Re_τ	171	171	171	171
Fr	0.0192	0.0333	0.0192	0.0105
u_τ	8.54e-3	1.23e-2	8.54e-3	5.72e-3
DOMAIN:				
length (L), m	0.126	0.0874	0.126	0.188
width (W), m	0.0631	0.0437	0.0631	0.0942
depth (D), m	0.0201	0.0139	0.0201	0.0300
FLUID:				
viscosity (ν), mm^2/s	1.004	1.004	1.004	1.004
gravity (g), m/s^2	9.81	9.81	9.81	9.81
WAVE:				
(theory, w/o current)				
speed, m/s	n/a	0.323	0.387	0.473
frequency rad/s	n/a	23.2	19.3	15.8
period, s	n/a	0.271	0.326	0.398
amplitude (a), m	n/a	0.00251	0.00361	0.00540
wave number (k), rad/m	n/a	71.8	49.8	33.3
wavelength (λ), m	n/a	0.0874	0.126	0.188
slope (ε or ak)	n/a	0.18	0.18	0.18
β, $1/mm$	n/a	3.40	3.10	2.80
δ (or β^{-1}), mm	n/a	0.294	0.323	0.357
CURRENT AND WAVE:				
(numerical results)				
U_{bulk}, m/s	0.124	0.241	0.167	0.113
wave speed (c), m/s	n/a	0.573	0.573	0.647
wave frequency (σ), rad/s	n/a	41.1	28.5	21.6
wave period (T), s	n/a	0.153	0.220	0.291
β, $1/mm$	n/a	4.53	3.77	3.28
δ (or β^{-1}), mm	n/a	0.221	0.265	0.305

Table 5.2: Dimensional open-channel simulation parameters.

	ak	wave amplitude (mm)	wave length (mm)	wave speed (mm/s)	current speed (mm/s)	flow depth (mm)
Simulations:						
W1	0.18	2.5	90	570	240	14
W2	0.18	3.6	130	570	170	20
W3	0.18	5.4	190	650	110	30
Experiments:						
TWCF	0.093	5.5	370	870	130	100
RWCF I	0.14	24	1010	1300	60	350
RWCF II	0.066	16	1510	1520	60	350

Table 5.3: Comparison of simulation and experiment parameters.

with laboratory experiments is that the choice of a Reynolds number and a Froude number for a flow fixes the physical length scale of the flow. Thus, the simulation of a particular combination of mean flow and wave size is only directly comparable to exactly one laboratory experiment. However, based on the insights developed in the non-dimensionalization of the governing equations in section 2.2, we can expect that different aspects of the data will be comparable across various experiments.

Even if we had sufficient computational power to model a wave and current of exactly the same dimensions as the TWCF experiment of Cowen, this would not necessarily provide a perfect comparison of simulation and experiment. In the laboratory experiments, the data collected are for a spatially-decaying wave train propagating over either a developing turbulent boundary layer (TWCF), or decaying grid-generated turbulence (RWCF). In contrast, the numerical simulations are for a temporally-decaying wave over a temporally-developing boundary layer (which has already reached a statistically steady-state before introduction of the wave). The spatial decay of a wave train implies: (1) in a wave-following coordinate system, a single wave amplitude decays in time, and (2) the amplitude decay of the wave train over space is a continuous gradient, so the amplitude of a single wave decays over its wavelength. Thus, the spatially-decaying wave can have mean spatial gradients of velocity over a single wavelength caused directly by the wave decay. However, a temporally-decaying infinite train of waves is a series of waves that are identical

over every wavelength at a particular instant of time. Thus, simulation of a single wavelength of a temporally-decaying wave train will have mean temporal gradients of velocity due to the wave decay, without mean spatial gradients of velocity caused by the primary wave. The importance of this difference is shown in analysis of the wave-induced velocity and vorticity fields for the near free-surface region discussed in sections (6.2) and (6.3).

5.3.3 Data collection

The interaction of a decaying wave and a turbulent current is a temporally-evolving phenomena, so meaningful time-averages of flow variables cannot be computed during the simulation run. Instead, it is necessary to collect the instantaneous simulation data throughout the simulation run and use statistical detrending techniques (*e.g.* Bendat and Piersoll [1986]) to analyze the flow (see section 5.4.3). The need to store time-accurate data is a problem for a four-dimensional moving-grid simulation since a minimum of 7 pieces of data[5] per grid point for each time step is required to retain the full velocity and pressure fields for the flow. For a modest simulation run of 6000 time steps on a $34 \times 34 \times 66$ grid (about 140 hours of computational time on a Cray J90) storage requirements for full simulation data is 3.2 gigawords per simulation run. If held in binary form at 16 bits per word this is 51 gigabytes of data. If we increase the grid resolution to $66 \times 66 \times 66$, storage of 6000 time steps of data would require 193 gigabytes[6].

With the continued forward march of computational power, our ability to conduct time-accurate simulations of temporally-evolving flows is outstripping our ability to store all the data produced. This leads to a dilemma: when storage is limited, how does one decide what data is important to store prior to the analysis of the stored data? There does not seem to be a good answer to this question. In the present work, it has been necessary to re-run several simulations to collect data that

[5]Not only do we have to store the primitive flow variables, but we also have to store the grid at each time step. Significantly more storage is required if the subgrid-scale strain rates developed in the turbulence model are kept for analysis of subgrid-scale dissipation and turbulence budgets.

[6]Of course, increasing the number of grid points will require a reduced time step size, and more time steps to simulate the same number of wave periods. This can quickly lead to a requirement for the storage of terabytes of data.

was previously thought to be unnecessary[7]. The first practical step in data reduction is to cut down the number of time steps in which data is stored. For the present work, cases C, $W1$ and $W2$ stored data every 10 time steps, while case $W3$ (which used a smaller time step) stored data every 20 time steps. In doing this, we lost the ability to look at the small-time-scale fluctuations in the flow.

Reducing the number of time intervals for data storage dropped the theoretical storage requirements for the four simulations from 120 gigabytes to 10 gigabytes. Since this was still too large for the available disk storage, we were forced to reduce our data collection to a limited number of grid points in the domain. This could be done either by (1) collecting data on some coarser grid scale, such as a $16 \times 16 \times 32$ grid, or (2) collecting the full grid resolution on a limited number of two-dimensional surfaces. The first approach would reduce the data storage requirements to about 1.5 gigabytes (a difficult, but achievable strain on the available storage capacity), with the undesirable side-effect of smoothing out small-space-scale fluctuations. The second approach leaves large areas of the flow for which no instantaneous data is available, but allows analysis of small-space-scale fluctuations and provides a dramatic reduction in the storage requirements. We chose the second approach so as not to lose the ability to look at small-space-scale fluctuations while keeping our data storage requirements to a minimum.

The data from the simulation runs were stored in two ways: (1) time-evolving, spanwise-averaged flow statistics and grid coordinates were computed and stored for a vertical $(x : z)$ plane beneath the wave, and (2) instantaneous fields of velocity, pressure, vorticity, grid coordinates and turbulence model coefficients were stored on four curvilinear surfaces ($x\!:\!z$, $x\!:\!y$, and $y\!:\!z$ sections at at the middle of the domain plus the free surface) as illustrated in figure (5.2). One disadvantage of storing the flow data on a limited number of surfaces is that during the simulation we must compute the non-primitive variables which are dependent on non-stored grid locations. For example, to obtain the streamwise vorticity (ω_1) on a vertical $x-z$ plane requires the velocities and grid points on two planes parallel to the stored plane. This leaves the choice of either storing three surfaces for every surface on which data is desired, or computing and storing the vorticity in the data output. We chose the latter approach,

[7]This is a prime motivation for keeping simulations small and relatively short for initial investigations with new numerical methods (such as developed in the present work).

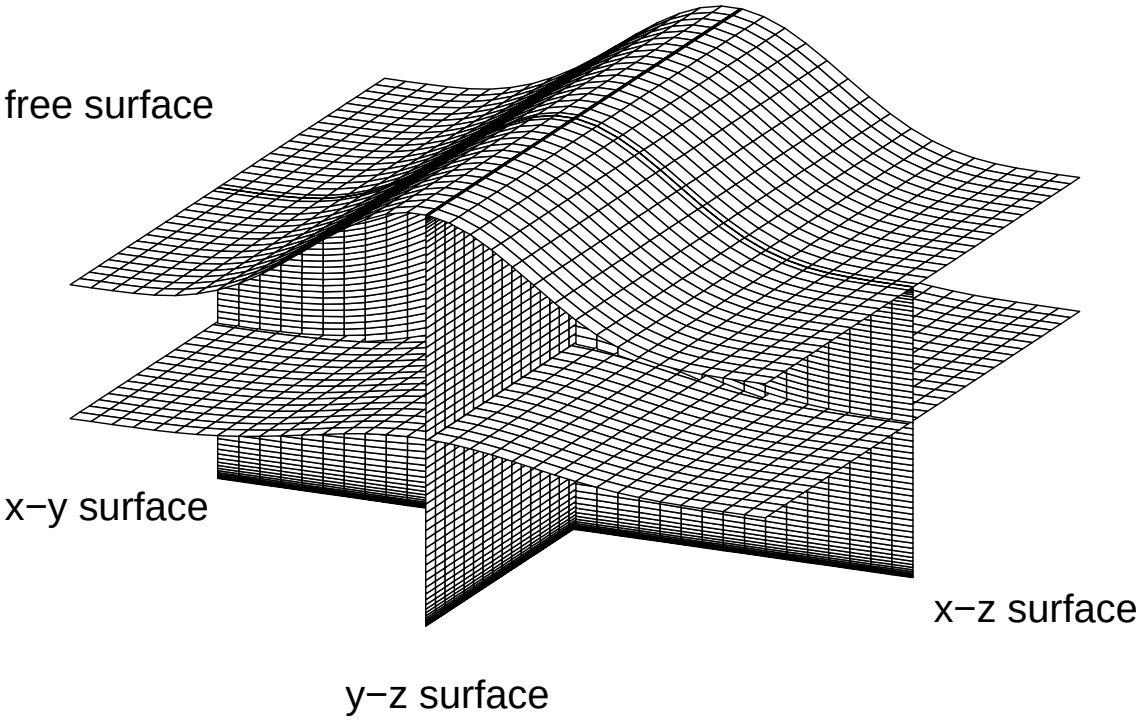


Figure 5.2: Instantaneous data collection surfaces.

which results in smaller storage requirements at the cost of extra computations in each data output time step.

5.4 Data analysis techniques

5.4.1 Computation of phase-averaged statistics

Because the open-channel flow and the monochromatic progressive wave are homogeneous in the spanwise direction, we can use the ergodic hypothesis to compute instantaneous spanwise statistics of the flow fluctuations from the full instantaneous flow field at each output time step. In the wave-following coordinate system these statistics are instantaneous averages beneath different phases of the wave and will be

referred to as *instantaneous phase-averaged data*[8].

The means of velocity and vorticity in the spanwise (y) direction for each (x, z) pair in the flow field are subtracted from the full instantaneous flow field to obtain the fluctuations from spanwise means at every point in the flow. The mean squares in the spanwise direction for these fluctuations are stored as the instantaneous, phase-averaged components of the fluctuating enstrophy and turbulence intensities[9] for each (x, z) position in the flow. This approach collapses the temporally-evolving three-dimensional spatial data into temporally-evolving two-dimensional spatial data. Using this data, we can develop animations of the temporal changes in phase-averaged effects. This allows us to look at the temporal changes of the large-scale structures in the flow (see chapter 7).

5.4.2 Computation of wave-induced effects

Wave-induced effects can be defined as the difference between the statistics for a wave-current simulation and the current-only simulation. By subtracting the current-only statistics from the wave/current statistics we can arrive at the wave-induced effect[10].

[8]The mean or average value of a quantity (u) over a distance W is defined as

$$\bar{u} \equiv \frac{1}{W} \int u \, dW$$

For the spanwise direction, the grid is essentially uniform and an average can be computed from a simple discrete sum:

$$\bar{u} \approx \frac{1}{W} \sum u \, \Delta W$$

In the streamwise direction, the stretching of the grid due to the wave requires computation of averages using the integral form. For the present work, this was done using trapezoidal numerical integration.

[9]This definition of the turbulence intensities as the difference between the instantaneous flow field and the instantaneous spanwise mean has implications on the flow effects that are included in the "turbulence." This is discussed in detail in sections 5.4.6 and 6.4.

[10]Subtracting the currently-only profile from the wave/current profile is based on the assumption that the current-only data follow the surface. This is a reasonable assumption that is often used in the study of wind-generated water waves (see Harris [1992]).

A formal decomposition of a quantity (such as a vorticity fluctuation component) into a current-effect and a wave-induced effect can be written:

$$\omega' = \omega'_c + \omega'_w \tag{5.3}$$

The phase-averaged product of the fluctuations is:

$$< \omega'\omega' > \quad = \quad < \omega'_c\omega'_c > \quad + \quad < 2\,\omega'_c\omega'_w > \quad + \quad < \omega'_w\omega'_w > \tag{5.4}$$

Since the current-only simulation contains only the term $< \omega'_c\omega'_c >$, it follows that subtraction of this term from the phase-averaged product (the definition of the wave-induced effects) leaves both the fluctuations caused directly by the wave, and the fluctuations that result from the interaction of the wave and the current. Since we are in a wave-following coordinate system, the wave-induced velocity field is essentially steady (except for decay effects due to wave attenuation, viscous conduction effects from the boundary layers, and small-scale perturbations near the free surface). The products of wave-induced fluctuations such as $< \omega'_w\omega'_w >$ are small, so the computed wave-induced effects are predominantly the effect of the cross-terms such as $< 2\omega'_c\omega'_w >$, which result from the interactions of the wave and the current[11].

5.4.3 Detrending data

Since there is no surface forcing, the simulated waves decay due to the viscous effects of (1) the bottom boundary, (2) free-stream turbulence and (3) the free-surface boundary layer, and (4) wave-induced shear in the flow interior. As such, we cannot temporally-average data from different time steps unless the decay trend is removed. To the extent that wave-induced effects are driven by the wave amplitude, it is appropriate to use the wave amplitude decay as the decay scale for detrending. However, where wave-current interactions are strong, the wave decay cannot be assumed to be the only temporal scale for the flow. In particular we find that the temporal evolution of wave-induced turbulence intensities and enstrophy do not scale simply on the wave decay. However, the wave-amplitude decay rate does provide a reasonable approximation of the temporal evolution of the mean wave-induced velocities and vorticities. This result is in accordance first-order wave theory (Phillips [1977]).

[11] The predominance of the $< 2\omega'_w\omega'_c >$ term could be demonstrated by simulating a wave over quiescent fluid and subtracting this from equation (5.4). This remains a subject for future investigations.

Since the wave-induced enstrophy components and turbulence intensities are mean square fluctuations from the mean vorticities and mean velocities, a naive expectation might be that they should decay as the square of the wave amplitude. Indeed, for the vertical velocity fluctuations this proved to be the case. However, the other terms either decay at a different rate, or have no decay at all. The wave-induced streamwise and spanwise turbulence intensities do not show any consistent decay trend, but instead have oscillations in the profiles over the course of the simulations. It appears that the wave-current interactions for the streamwise and spanwise turbulence intensities are likely a function of wave speed and/or the difference between the wave speed and the current speed (which remain constant over the course of the simulation). This is consistent with the developments of rapid distortion theory for waves that propagate significantly faster than the turbulent scales of motion (see section 7.3.3). A further complication to determining the appropriate decay trend for the wave-induced turbulence intensities is the velocity effects of smaller parasitic surface waves[12]. These waves are fed by nonlinear interactions in the flow and exhibit an initial growth in the first five to 10 wave periods of the primary wave, followed by decay in the remainder of the simulations. The decay rate for the amplitude of the parasitic waves is less than the decay rate for the primary wave.

The decay of the enstrophy components appears to scale on the wave amplitude, which implies that there is a growth trend in the wave-current enstrophy interactions that is smaller than the decay induced by the wave attenuation. As demonstrated analytically by Longuet-Higgins (1953), a viscous wave has a temporal conduction of vorticity from the boundaries into the interior of the fluid. Our initial conditions for the simulation method are for an irrotational wave field superposed on a turbulent current, so we can expect a temporal *growth* for the wave-induced vorticity in the interior of the fluid. Since our simulations are turbulent, the boundary vorticity will be convected into the interior rather than conducted, with a a time scale of the order D/u_τ (*i.e.* between 10 to 20 wave periods for the simulations conducted). Thus, the observed result (that the net decay of wave-induced enstrophy is less than the square of the wave amplitude) is reasonable.

To determine the wave amplitude decay rate, the instantaneous wave amplitude was computed for each simulation case, then fitted (using a least-squares

[12]See section 5.4.6 for a discussion of parasitic waves.

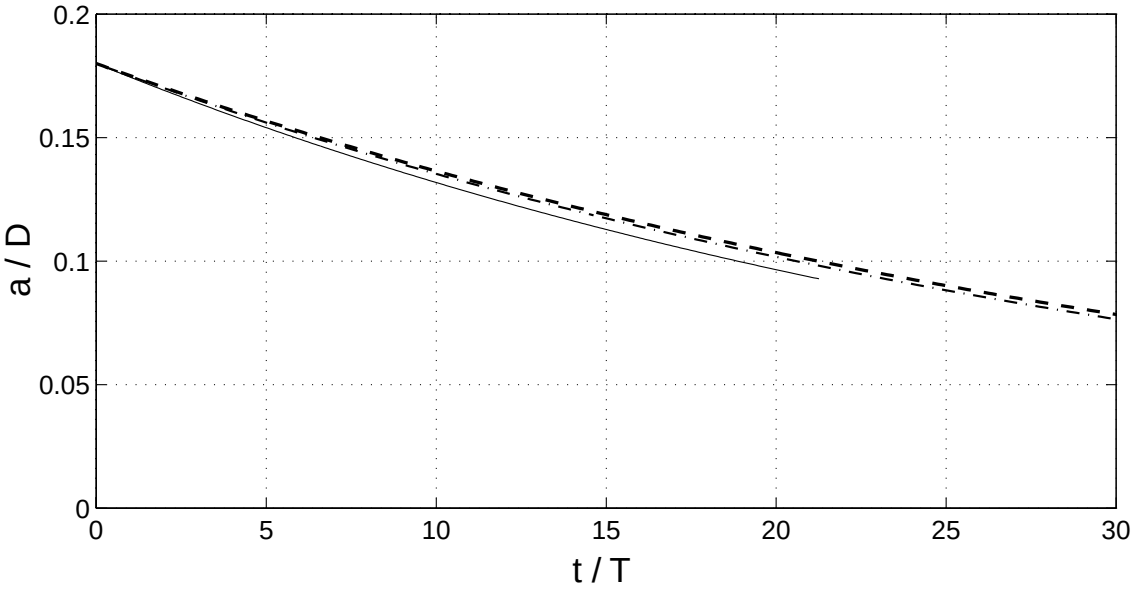


Figure 5.3: Wave attenuation. —— $W1$, – – – $W2$, –·–· $W3$,

	W1	W2	W3
α	0.0311	0.0277	0.0285

Table 5.4: Exponential wave amplitude decay rate.

approach) with an exponential curve of the form

$$\mathrm{a} \;=\; \mathrm{a}_0 \exp\left(-\alpha \frac{t}{T}\right) \tag{5.5}$$

where a_0 is the non-dimensional initial wave amplitude, t is the simulation non-dimensional turbulent time scale (D/u_τ), T is the non-dimensional wave period, and α is the decay rate determined by the equation fit. The resulting amplitude decay curves are plotted against the simulation time (non-dimensionalized by the wave period) in figure (5.3). The α coefficients of the decay trends are shown in table (5.4). Where the simulation data had an identifiable decay trend that was well represented by the decay of the wave amplitude, the data was detrended back to the original wave amplitude so that temporal averages could be computed (see Bendat and Piersoll [1986] for a discussion of data detrending).

5.4.4 Redefinition of the coordinate system

In chapters 2 and 3, it was convenient to define the origin of the z coordinate as the bottom boundary for purposes of numerical implementation. However, for comparisons to experiments and the literature on water waves, it is convenient to redefine the origin of z as the still water level so that the bottom boundary coincides with $z = -D$ (where D is the domain depth). This coordinate system is the one used by Cowen (1996), as well as in most of the literature on linear and finite-amplitude wave theory.

5.4.5 Computation of wave-averaged statistical profiles

The mean effect of the wave and the wave-current interactions can be reduced to a single vertical profile for each statistical quantity by computing a spatial average of the temporally-averaged (detrended) data across a wavelength. As discussed by both Cowen (1996) and McDonald (1994), a direct Cartesian average along planes of constant z becomes problematic in the region above the trough for computation of the streamwise velocity profile.

McDonald (1994) extended a method by Norris and Reynolds (1975) that maps the free-surface onto a flat plane so that the z coordinate is transformed into the z^* coordinate, where $z^* = 0$ is on the free surface (rather than at the still water level which is the $z = 0$ origin) and the coordinate spacing is stretched or compressed so that $z^*/D = -1$ at the bottom of the domain. Cowen (1996) adapted McDonald's numerical approach to interpret his laboratory-developed, experimental data. McDonald (1994) presents a relationship for z^* and z that is second order[13] in the wave slope (ε):

$$\begin{aligned} z^* \;=\; z \;&-\; \frac{\varepsilon}{k}\,\frac{\sinh k\,(D+z)}{\sinh kD}\,\cos kx \;+\; \frac{\varepsilon^2}{2k}\,\frac{\sinh 2k\,(D+z)}{2\sinh^2 kD}\,\cos^2 kx \\ &-\; \frac{\varepsilon^2}{4k}\,(\coth kD)\left(1 \,+\, \coth^2 kD \,+\, \frac{2}{\sinh^2 kD}\right)\frac{\sinh 2k\,(D+z)}{\sinh 2kD}\,\cos 2kx \\ &+\; O\left(\varepsilon^3\right) \end{aligned} \tag{5.6}$$

[13]The initial transformation presented by McDonald (1994) is an exact transformation from z^* to z, the reverse transformation that McDonald derived (our equation [5.6]) is an approximation.

where k is the wave number, and D is the domain depth. All variables are non-dimensionalized by the physical domain depth (so $D = 1$) and the equation is valid in a wave-following coordinate system.

McDonald (1994) showed that computation of the mean streamwise Eulerian velocity profile beneath a wave in a z^* coordinate includes a streamwise velocity effect that is due to the vertical grid stretching. A velocity profile for an irrotational wave in the z coordinate system that has a zero Eulerian mean (integrated beneath the wave on lines of constant z) will show a *positive* Eulerian mean velocity in a z^* coordinate system due to the effects of grid stretching. The profile of this effect (which Cowen[14] called $\bar{u}^*_{ES}$) can be presented as:

$$\bar{u}^*_{ES} = \frac{c}{2} \frac{\sinh^2 k\,(D + z^*)}{\sinh^2 kD} \tag{5.7}$$

Cowen (1996) obtained his experimental data in a z^* coordinate system, then subtracted $\bar{u}^*_{ES}$ from the profile of the z^* mean streamwise velocity to obtain a mean Eulerian velocity profile beneath the wave in the z coordinate.

Our numerical simulation uses a wave-following coordinate system with a stretching function developed by the Poisson grid generation system. This is similar to McDonald's (1994) approach in its effect, and makes it unnecessary to apply McDonald's (1994) formal transformation to obtain data in a z^* coordinate system. If we average the value of z along a curvilinear coordinate line with constant ζ, we obtain a mean value of z for that line, which we will call z_m. Figure (5.4) shows the difference between the z_m of the simulation and z^* computed from equation (5.6) over the depth of the domain at various phase angles beneath the wave. It can be seen that the difference between McDonald's z^* and the simulation z_m is third-order in the wave slope. Since our z_m corresponds to McDonald's z^*, we can use z^* in place of z_m to display the results of our simulations. This provides consistency with the work of Cowen (1996).

For direct comparisons to data developed by Cowen (1996) we can use equation (5.7) to compute the effect of grid stretching on the mean Eulerian velocity and subtract it from our computation of the mean velocity in the z^* coordinate system. This is done for the results presented in section 6.2.

[14]The presentation of the McDonald (1994) formula in Cowen (1996) contains a typographical error.

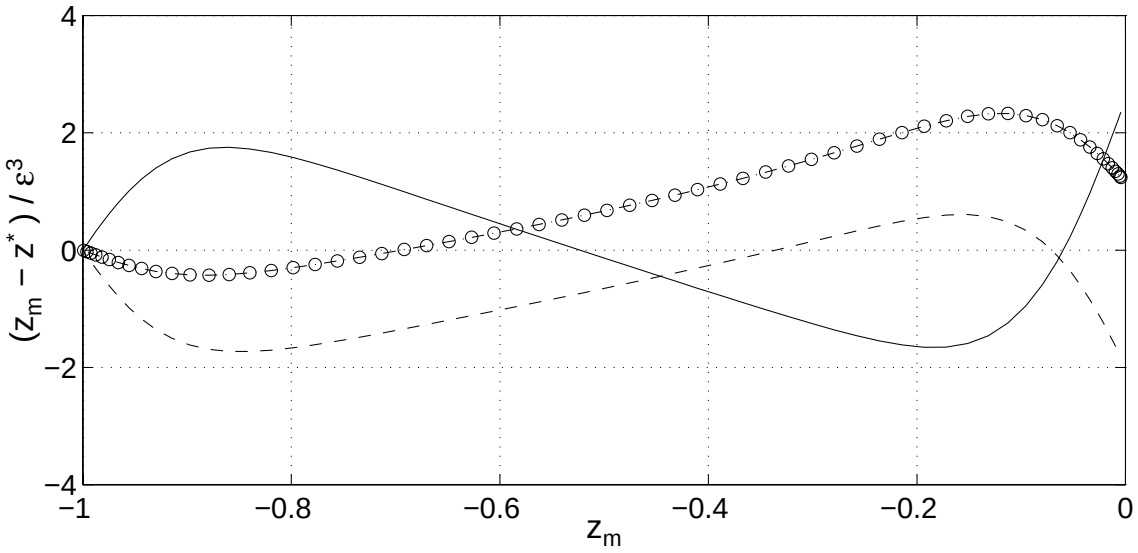


Figure 5.4: Comparison of McDonald's depth-normalized z^* to the simulation depth-normalized z_m averaged along a ξ_1 curvilinear coordinate line. The ordinate axis is scaled by the cube of the wave slope (ε). $---$ $kx = -\pi$ (trough), $\circ$ $kx = -\pi/2$, —— $kx = 0$ (crest), $-\cdot-$ $kx = \pi/2$

5.4.6 Computation of parasitic wave effects

The combination of wave and current produces a parasitic wave on the free surface that can be approximated using linear wave theory for short-crested waves. The parasitic wave shape can be defined as the instantaneous perturbation from a monochromatic wave for every point on the surface[15]. This process removes the primary monochromatic wave in the domain and leaves the surface perturbations. The short-crested parasitic wave shape is not seen in simulations of the wave without a current or in simulations of the current without a wave[16]. Thus, we can conclude that the parasitic waves are caused by interactions between the wave and current and/or nonlinear wave-wave interactions that may occur once the wave shape significantly

[15]This definition of the parasitic wave shape neglects monochromatic parasitic waves. However, the phase-averaged wave profile and the surface spectrum show that monochromatic parasitic waves do not occur in the simulations conducted.

[16]Simulations of a wave without a current have been conducted, but have not been fully analyzed and are not presented here.

departs from monochromatic. A typical example of the instantaneous surface perturbation is shown in figure (5.5). The perturbation amplitude ($\Delta\eta$) of the parasitic wave is normalized by the amplitude of the monochromatic wave, and can be seen to be on the order of 20% of the wave amplitude[17]. Since the wave amplitude[18] is 0.18, it follows that the parasitic waves are of second-order in the primary wave amplitude (*i.e.* $O[a^2]$).

The parasitic surface wave can be compared qualitatively to a wave shape obtained using linear theory[19] shown in figure (5.6). Linear and nonlinear short-crested wave theory was developed by Fuchs (1952); it can also be found in an abbreviated form in Wiegel (1964). The wave shape in figure (5.6) is developed from the linear surface deformation equation:

$$\Delta\eta \;=\; a_p \sin(mx)\,\cos(ny) \tag{5.8}$$

where a_p is the amplitude of the short-crested wave and m and n are wave numbers in the x and y directions (based on the streamwise and spanwise and distance between crests). For consistency, the amplitude, wave numbers and (x, y) coordinates of the wave are non-dimensionalized by the domain depth. For the surface perturbation shown in figure (5.5), the streamwise wave number[20] is $m = 2$ and the spanwise wave number is $n = 2$. Due to the periodicity of the boundary conditions, these wave numbers correspond to the largest wave lengths (besides the primary monochromatic wave) that are supportable in the domain.

A quantitative comparison between the parasitic waves and linear short-crested wave theory can be made by subtracting the wave forms shown in figures (5.5) and

[17] The maximum perturbation height does not exceed 30% of the wave amplitude in the simulations conducted.

[18] We have conveniently designed our non-dimensional domain such that the wave number (k) is 1, so our wave amplitude is equal to the wave ak.

[19] While it is possible to compare the surface perturbation to nonlinear theory, the additional complexity does not provide any "better" results. The comparison of theoretical waves to parasitic components of a nonlinear wave is a heuristic approach since a nonlinear wave is *not* a linear superposition of its components. However, to the leading order, we expect the behavior of the parasitic components of a nonlinear wave to have their analogue in the behavior of small-amplitude waves.

[20] The wave number (k) of the primary wave is 1 and the domain width is half of the domain length, so the resulting short-crested waves have identical wavelengths in the x and y directions.

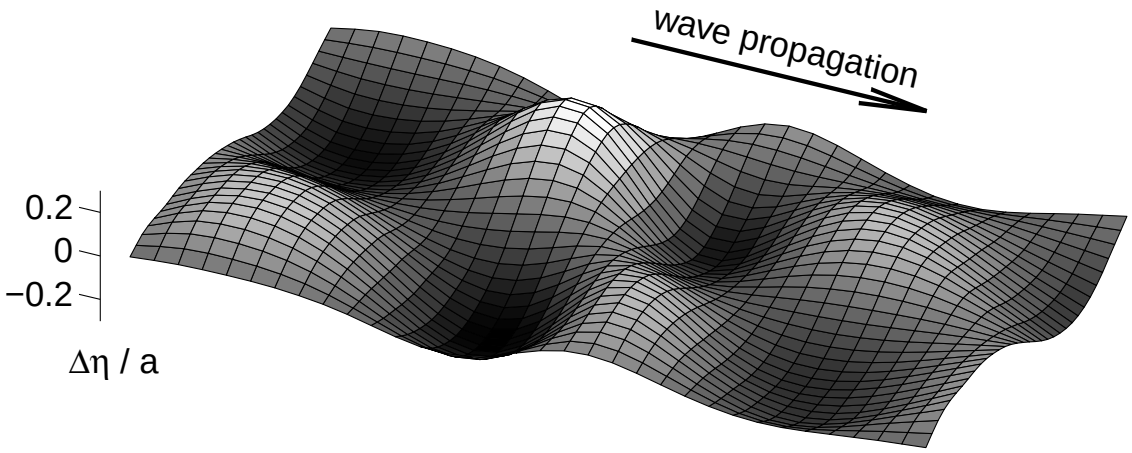


Figure 5.5: Instantaneous perturbation of surface height for monochromatic wave. Vertical axis normalized by monochromatic wave amplitude (horizontal axes not to scale).

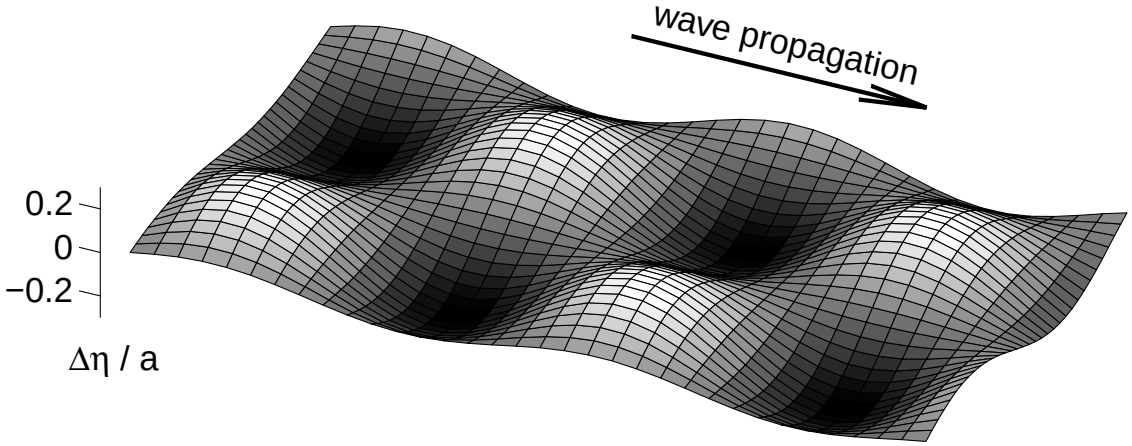


Figure 5.6: Short-crested wave shape using linear theory. Vertical axis normalized by monochromatic wave amplitude (horizontal axes not to scale).

(5.6). The standard deviation of the difference for the two figures shown is approximately 8% of the maximum perturbation height of the parasitic wave. It follows that linear short-crested wave theory is a reasonable approximation of the parasitic waves seen in the flow simulations.

The use of linear wave theory for the parasitic waves allows us to estimate the wave speed (c_p) and sub-surface velocity effects of the parasitic waves. The theoretical wave speed is obtained from:

$$c_p^2 = \frac{gr}{m^2} \tanh(rD) \tag{5.9}$$

where g is non-dimensional gravity, D is the domain depth ($D = 1$), and r is the magnitude of the resultant wave number vector, defined as $r^2 = m^2 + n^2$. As demonstrated by Fuchs (1952), this wave speed is larger than the wave speed of a monochromatic wave with the same streamwise wavelength. The speed of the short-crested waves is very close to the speed of the primary wave in the simulations conducted. This results in a surface perturbation that moves with the primary wave and appears almost stagnant in the wave-following coordinate system.

The velocities beneath an irrotational, small-amplitude, short-crested wave can be determined from:

$$u = \frac{ga_p}{c_p} \frac{\cosh r(D+z)}{\cosh rD} \sin(mx) \cos(ny) \tag{5.10}$$

$$v = \frac{ga_p n}{c_p m} \frac{\cosh r(D+z)}{\cosh rD} \cos(mx) \sin(ny) \tag{5.11}$$

$$w = \frac{ga_p r}{c_p m} \frac{\sinh r(D+z))}{\cosh rD} \cos(mx) \cos(ny) \tag{5.12}$$

where z is the vertical coordinate measured from the still water level. Of particular interest is the existence of a spanwise velocity v that is associated with a short-crested wave moving in the streamwise direction. This component does not have an analogue in the solution of an irrotational monochromatic wave.

Since the parasitic waves are not monochromatic, their velocity field will appear as fluctuations from the spanwise phase-average in the simulation results. As noted in section 5.4.2, we can obtain wave-induced effects by subtracting the time- and wave-averaged mean profiles of current-only simulation results from similar profiles for the wave-current case. For the turbulence intensities, we can also subtract the

mean effects of the parasitic waves (based on linear theory) to arrive at wave-induced effects due solely to the interaction of the waves and the current. This approach provides further insight into the wave-current interaction without the fluctuation signal caused by the parasitic waves[21].

The parasitic waves have a growth and decay behavior that is not consistent with the decay of the primary wave. To prevent the behavior of the parasitic waves from contaminating the detrending process, we computed the parasitic wave effect from the instantaneous data and subtracted this from the computed wave-induced data before the detrending. The parasitic wave velocity field (from equations [5.10] through [5.12]) was computed based on the instantaneous RMS amplitude of the parasitic surface perturbation. The square velocities of this field were averaged beneath the wave to obtain the mean effect of the parasitic waves.

[21]Removing the irrotational effect of the parasitic waves does not remove the interaction effects (if any) of the parasitic waves with the current.

Chapter 6

Time-averaged simulation results

6.1 Introduction

The data developed in our simulations are analyzed using the techniques outlined in chapter 5. In this chapter we concentrate on analyzing the simulation results for the time-averaged mean velocity, vorticity, turbulence intensities and enstrophy. By comparing this data to experimental results of Cowen (1996) we gain confidence in the validity of our simulation method and provide further insight into the mean interactions of waves and turbulence.

It will be convenient to divide the flow domain into three layers, defined as:

(1) bottom boundary layer: $-1.0 < z^*/D < -0.7$

(2) flow core: $-0.7 < z^*/D < -0.1$

(3) near-free-surface region[1]: $-0.1 < z^*/D < 0$

Our interest lies in analysis of the flow in the core region and the near-free-surface region. For the present work, the bottom boundary layer will not be discussed in detail.

[1]The near-free-surface region we have defined is approximately 5 to 10 times the thickness of the free surface boundary layer (β^{-1}, see Table [5.1]), depending upon the wave speed. A more precise definition of the near-free-surface region would use a consistent scaling from the free surface based on the boundary layer thickness rather than a non-dimensional depth. For our purposes, the choice of $-0.1 < z^*/D < 0$ provides an approximation of the region where the effects of the small surface waves and viscous forces are concentrated.

6.2 Wave-induced velocity

6.2.1 Introduction

Reports on experimental investigations of flows with surface waves have often noted a net-negative, wave-induced, mean Eulerian current with a negative shear profile[2]. Cowen (1996) demonstrated that the negative shear in the wave-induced velocity profiles seen in his experiments could *not* be attributed to advection from the horizontal boundaries since the arrival of the velocity deficit was based on the wave group velocity rather than the mean current velocity. In addition, he provides convincing arguments that the laboratory waves can be approximated as Gerstner waves (rotational waves with closed particle orbits, Kinsman [1984]). Monismith[3] has noted that the negative shear appears to be a fundamental feature of laboratory waves.

6.2.2 Comparison of simulation and experiments

We expected to confirm that a spanwise- and streamwise-unbounded numerical flow would *not* show a negative Eulerian-mean shear in the wave-induced velocity. This would indicate that the negative shear seen in laboratory experiments is due to (1) the convection of vorticity from the side and end walls, (2) the establishment of Gerstner waves in laboratory experiments as proposed by Cowen (1996), or (3) some unknown laboratory-scale effect. However, as illustrated in figure (6.1) the wave-induced mean streamwise velocity profiles for the numerical simulations show a distinct negative shear up to the lower edge of the near free-surface layer. For comparison purposes, the velocity profile developed by Cowen in the Tilting Wave-Current Flume (TWCF) at the Stanford Environmental Fluid Mechanics Laboratory is presented in figure (6.1) as a dashed line. The data is non-dimensionalized on the wave scale for second-order effects ($c\varepsilon^2$), and has had the effect of the grid stretching on the mean velocity profile removed (see section 5.4.5).

The negative shear in the core of the flow turns into positive shear in the

[2]See Cowen (1996) for a detailed discussion.

[3]S.G. Monismith, Dept. Civ. Eng., Stanford Univ. personal communication (1997)

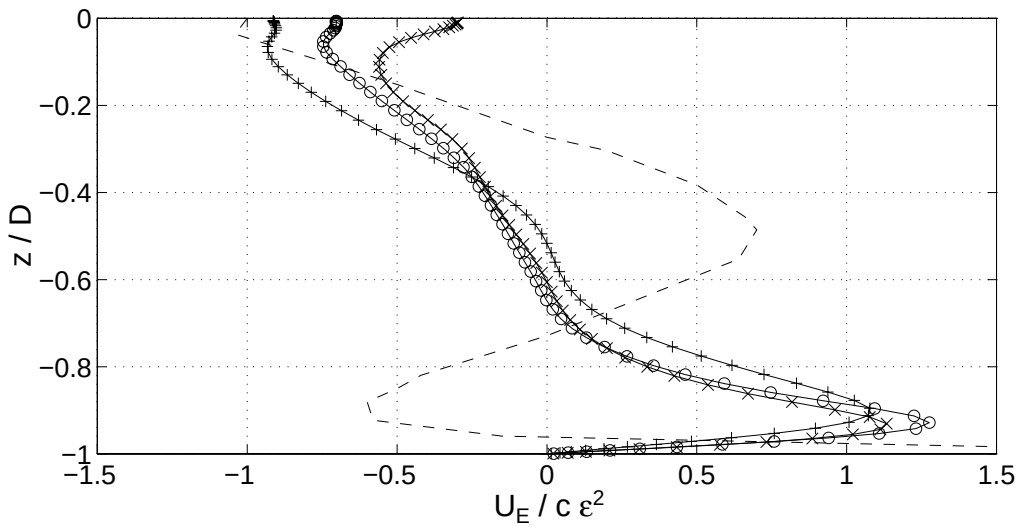


Figure 6.1: Wave-induced Eulerian-mean streamwise velocity compared to Cowen's data (TWCF). – – – Cowen, —|— $W1$, —○— $W2$, —×— $W3$

near-surface region, and then (for the simulations) into zero shear at the free surface. Figure (6.2) shows a close view of the velocity profiles in the near-surface region. As the free-surface is approached, the velocity shear for the numerical simulations goes to zero as required by the dynamic boundary condition for a flow that is *not* spatially decaying (see section 6.2.5). The profile for Cowen's (1996) TWCF experiment in figure (6.2) appears[4] to show a slight positive shear which is consistent with the computation of the mean profile beneath a spatially-decaying wave.

The results of Cowen (1996) presented in figure (6.1) show the data exceeding $1.5\,c\varepsilon^2$ as the bottom of the domain is approached. According to Cowen[5] the profile does reverse and return to zero at the bottom; however the experiment was not designed to look at bottom boundary effects, so the bottom boundary layer was not fully resolved in the data presented. Because the bottom boundary layer in the TWCF flume was significantly thinner than in the present numerical simulations, any comparisons in that region would be suspect.

[4]The TWCF data of Cowen is not well resolved in the near free surface region, so the magnitude of the shear should be considered approximate (E. Cowen, Env. Fluid Mech. Lab., Stanford Univ., personal communication [1997]). Cowen developed well-resolved data in the near-surface region for his RWCF I and RWCF II experiments, but the scaling of those experiments is significantly different than that of the numerical simulations.

[5]E. Cowen, Env. Fluid Mech. Lab., Stanford Univ., personal communication (1997).

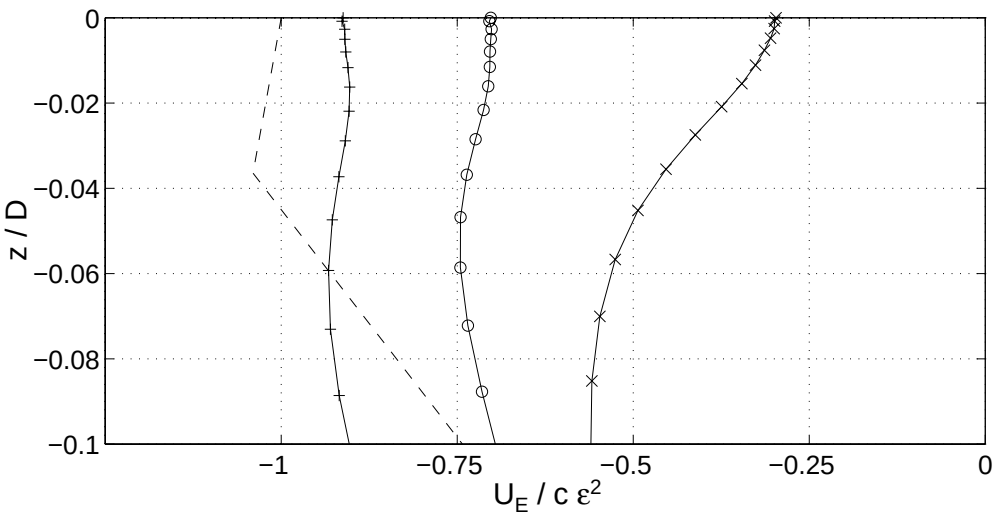


Figure 6.2: Wave-induced Eulerian-mean streamwise velocity in the near-surface region. – – – Cowen, —|— $W1$, —○— $W2$, —×— $W3$

6.2.3 Discussion of negative velocity shear

That both the simulations and the experiments result in negative shear is interesting, but may be coincidental. As pointed out by Cowen[6], a reasonable explanation for the negative shear in the simulations is a momentum deficit developed when slow-speed fluid is ejected out of the bottom boundary layer in bursting events, while the negative shear in the experiments of Cowen (1996) appears to be a property of the Gerstner waves developed at the wave maker. From temporal animations of the data (see chapter 7) it can be seen that one of the effects of the simulation waves is an increase in the turbulence intensity in the core of the flow due to entrainment of slow-speed fluid from the bottom boundary layer. Unfortunately, the present data cannot be used to conclusively demonstrate that the transport of the turbulent fluctuations into the near surface region is accompanied by a transport of low momentum fluid so as to provide a momentum deficit with a negative shear. Future simulations with scalar transport equations implemented in the numerical method may be able to determine if the negative shear is a momentum deficit effect.

[6] *ibid.*

Similar to the experimental results of Cowen (1996), Nepf (1992), and others[7], our results show a net-negative Eulerian-mean wave-induced velocity when the velocity profiles in figure (6.1) are integrated over the depth. From the data animations (chapter 7) and the analysis of the turbulence intensities (section 6.4) it can be argued that the net-negative result is caused by the rotation of streamwise fluctuations (and their momentum) into the spanwise and vertical directions as the result of rapid distortion effects. A task for future research is to determine conclusively whether this result is a coincidence due to the close proximity of the bottom boundary layer, or a necessary result for waves over a sheared current.

6.2.4 Evidence of simulation Gerstner waves?

The previous arguments provide a reasonable picture of the wave-induced streamwise velocity field as a result of the interaction between the waves and the bottom boundary layer. However, we can also conjecture that our simulation may have developed trochoidal Gerstner waves through the interactions of the rotational current with the irrotational Stokes wave used as the initial conditions. It is not clear how this would occur since a Gerstner wave requires a particle local rotation that is opposite to the orbital rotation. As noted by Wiegel (1964), this condition is most likely to be met where a wave is propagating *against* a sheared current; however, our simulations are of waves propagating *with* the sheared current.

The Eulerian velocity profile for a Gerstner wave (u_G) can be presented as (Wiegel [1964]):

$$u_G \;=\; -c\,\varepsilon^2 e^{2kz} \tag{6.1}$$

Since the simulation data for the wave-induced velocity profiles in the core of the flow collapse fairly well, we can define a "composite" velocity profile by averaging the profiles of the three waves. Figure (6.3) shows a plot of the Gerstner wave velocity profile against the composite velocity profile for the simulation waves in the flow core. It can be seen that the trend of the simulations is very similar to the trend of the Gerstner wave profile. The positive offset of the simulation profile may be attributable to the viscous streaming effects of the bottom boundary layer which are

[7]See Cowen [1996] for more discussion of experimental results.

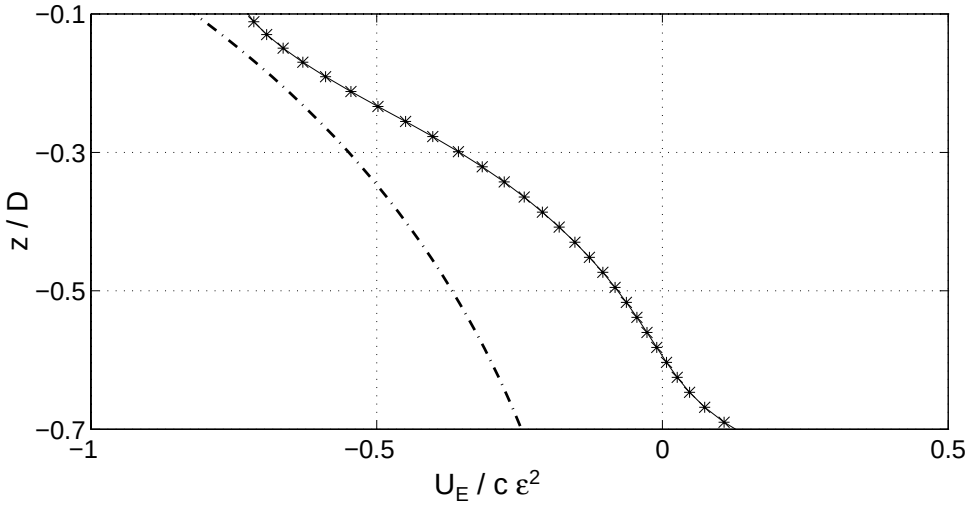


Figure 6.3: Wave-induced Eulerian-mean streamwise velocity in flow core ($-0.7 < z/D < -0.1$) for a composite average of numerical simulations compared to Gerstner wave velocity profile. $-\cdot-\cdot-$ Gerstner profile, $-\!*\!-$ composite simulation profile

not accounted for in the theory of Gerstner waves. While this evidence cannot be considered convincing on its own, in section 6.3.2 it is demonstrated that the vorticity field in the core of the flow also shows correspondence with the expected profile of a Gerstner wave. Further investigations are needed to determine (1) the mechanism of wave/current interaction that results in the negative shear for the Eulerian mean velocity profile, and (2) whether or not the simulation is developing Gerstner waves.

6.2.5 Discussion of near-free-surface velocity

Cowen's detailed RWCF results show a positive vertical shear of the streamwise velocity ($\partial \bar{u}_E / \partial z$) very close to the free-surface when averaged over a wave-length. Cowen attributes this to the spatial wave decay, which he analyzes through the vorticity expected to be generated by the decay. That the shear is attributable to spatial decay could also be argued from the tangential component of the dynamic boundary condition, which can be written for a two-dimensional, orthogonal, boundary-fitted curvilinear system (ξ, ζ) as:

$$\frac{\partial U}{\partial \zeta} + \frac{\partial W}{\partial \xi} = 0 \tag{6.2}$$

If a flow sees a mean vertical (Cartesian) shear in the horizontal (Cartesian) velocity, then it can be argued that the flow should also see a surface-normal shear in the surface-tangential velocity. If this is not the case, then the apparent mean Cartesian shear is an artifact of the coordinate system and does not indicate an actual strain rate in the flow in the near surface region[8]. A mean normal shear of the tangential velocity implies that:

$$\int \frac{\partial U}{\partial \zeta} d\zeta \; > \; 0 \tag{6.3}$$

In conjunction with equation 6.2 this implies that

$$\int \frac{\partial W}{\partial \xi} d\zeta \; < \; 0 \tag{6.4}$$

If a wave train decays spatially in amplitude while maintaining the same wavelength, then, at any particular phase of the wave, the magnitude of the vertical velocity (and thus the surface normal velocity) will be forced to decrease along the wavetrain, resulting in a mean negative gradient of the vertical and surface normal velocities. It follows that satisfaction of the inequality in equation (6.4) at the free surface implies that the wave is decaying. Thus a positive, mean, surface-normal shear of the tangential velocity is necessary for a spatially decaying wave that maintains a constant wavelength. A more succinct way of stating the above is:

> *The mean tangential stress on a free surface must vanish, which implies that a mean, negative, horizontal strain-rate of the vertical velocity must be balanced by a mean, positive, vertical strain-rate of the horizontal velocity.*

For the numerical simulations, a mean horizontal strain of the vertical velocity cannot be maintained with the periodic boundary conditions, so the vertical strain of the horizontal velocity must vanish. This effect is shown in the near-surface region for the streamwise velocity profiles in figure (6.2). This result indicates a fundamental difference between the near-surface behavior of a spatially-decaying wave in a laboratory and a temporally-decaying wave in a numerical simulation. This difference appears to be small; however, it may effect the generation of free-surface vorticity. This effect remains a subject for future investigation.

[8]Cowen's (1996) results account for the coordinate system, so that the shear appears to be physical rather than coordinate-based.

6.2.6 Phase-dependent mean velocity profiles

Figures (6.4) and (6.5) show the wave-induced, phase-averaged Eulerian velocity profiles for various phases of case $W3$ in the near surface layer. These figures provide a good qualitative comparison to similar graphs in Cowen (1996) (his figures [4.9a] and [4.9b]). Direct quantitative comparison is difficult due to the different scales of his RWCF I wave and the numerical simulations.

The strong negative shear in the streamwise velocity profile for the crest shown in figure (6.4) is a result of the requirement that the velocity at the free surface satisfies the tangential dynamic boundary condition. At the crest this can be written as:

$$\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} = 0 \tag{6.5}$$

Since the term $\partial w/\partial x$ is positive at the crest (for a wave moving in the positive x direction), it follows that a negative shear in the streamwise velocity is required. The same argument (but with opposite signs) explains the positive shear in the streamwise velocity beneath the trough. The magnitude of the negative shear beneath the crest is larger than the positive shear beneath the trough due to the steepening of the wave near the crest and the flattening of the trough. The nonlinear wave shape requires sharper horizontal gradients of the vertical velocity near the crest, which means a stronger shear in the streamwise velocity near the crest.

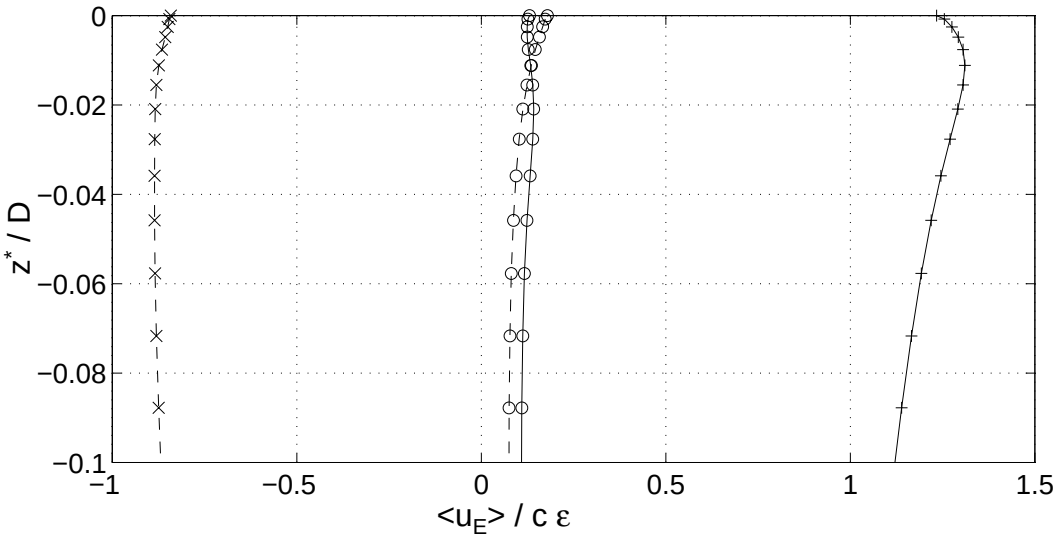


Figure 6.4: Phase-dependent, Eulerian streamwise velocity profiles in the near-surface region for case $W3$.
$--\times--$ $kx = -\pi$ (trough), $--\circ--$ $kx = -\pi/2$, $-|-$ $kx = 0$ (crest), $-\circ-$ $kx = +\pi/2$

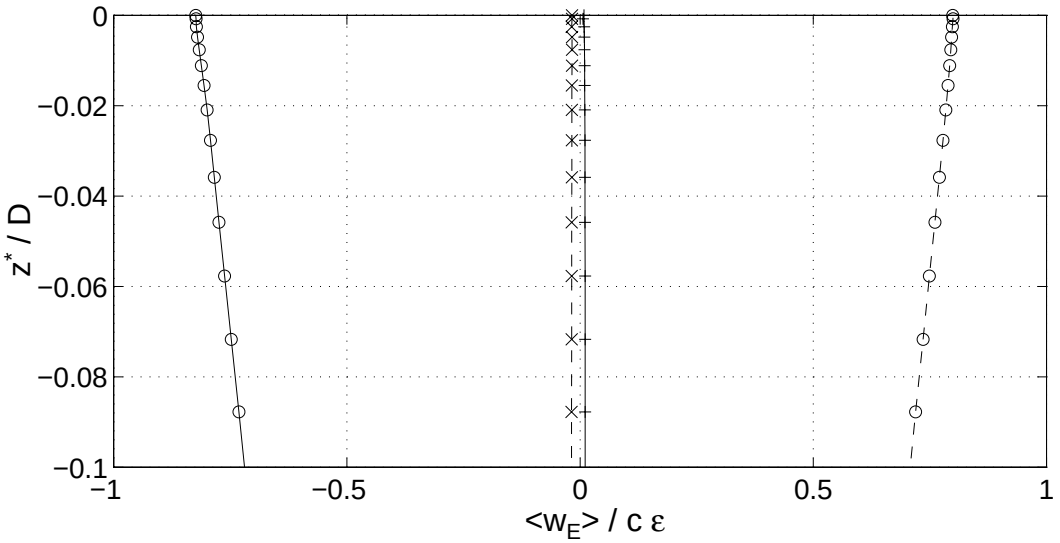


Figure 6.5: Phase-dependent, Eulerian vertical velocity profiles in the near-surface region for case $W3$.
$--\times--$ $kx = -\pi$ (trough), $--\circ--$ $kx = -\pi/2$, $-|-$ $kx = 0$ (crest), $-\circ-$ $kx = +\pi/2$

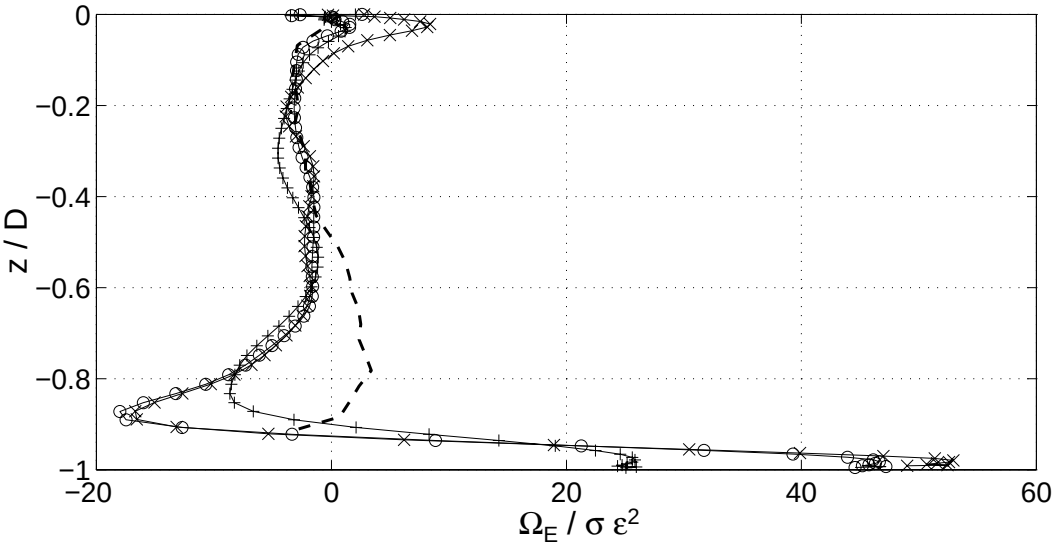


Figure 6.6: Wave-induced Eulerian-mean spanwise vorticity compared to Cowen's data (TWCF). – – – Cowen, —|— $W1$, —*— $W2$, —×— $W3$

6.3 Wave-induced vorticity

6.3.1 Comparison to experiments

Figure (6.6) shows a comparison of the wave-induced spanwise vorticity for the simulations and the TWCF flume data of Cowen. As with the velocity profiles, the agreement between experiment and theory is poor in the bottom boundary layer due to the different boundary layer thicknesses. A closer view of the data above the bottom boundary is provided in figure (6.7). This figure shows a distinct negative shear of the mean spanwise vorticity through the flow core with a sharp positive spike of the spanwise vorticity in the near-free-surface region.

6.3.2 Comparison with Gerstner waves

The good agreement of the experimental and numerical results in the core region was unexpected since the negative shear in the vorticity profile for the experiments

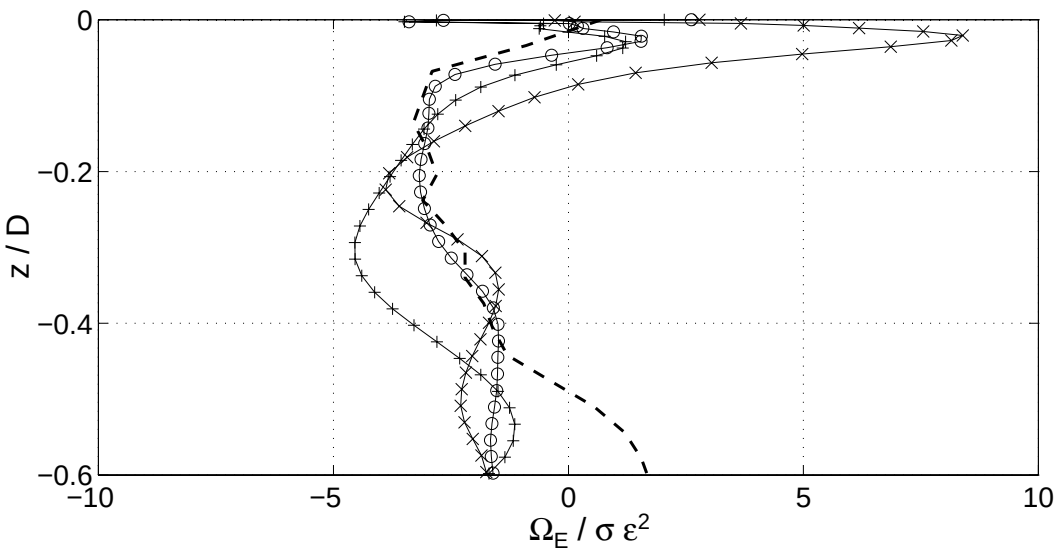


Figure 6.7: Wave-induced Eulerian-mean spanwise vorticity compared to Cowen's data (TWCF) above the bottom boundary layer. – – – Cowen, —|— $W1$, —*— $W2$, —×— $W3$

was attributed to the establishment of Gerstner waves at the mechanical wave maker. The correspondence between the simulation, experiments and Gerstner waves can be seen more clearly by graphing a composite wave-induced vorticity profile (the average of the three wave simulations) against the vorticity profile found in the experiments and the vorticity profile produced by Gerstner waves, as shown in figure (6.7). The vorticity profile for the Gerstner wave can be found from Kinsman (1984):

$$\omega_g \;=\; -\frac{2\sigma\varepsilon^2 e^{2kz}}{1 \,-\, \varepsilon^2 e^{2kz}} \tag{6.6}$$

where σ is the wave radian frequency, and ε is the wave slope (ak).

Based on the above results and the wave-induced velocity profiles in figure (6.3) it appears that the interaction between the initial Stokes wave and turbulent current in the numerical simulations may be producing rotational Gerstner waves. The mechanism that would produce this result is still unclear, and it may be that the correspondence between the experiment, simulation and Gerstner wave theory is coincidental. Alternatively, it may be that the interaction of a turbulent shear current and a waveform may produce Gerstner waves as a necessary consequence of interactions that are not yet understood. This remains a matter for future investigation.

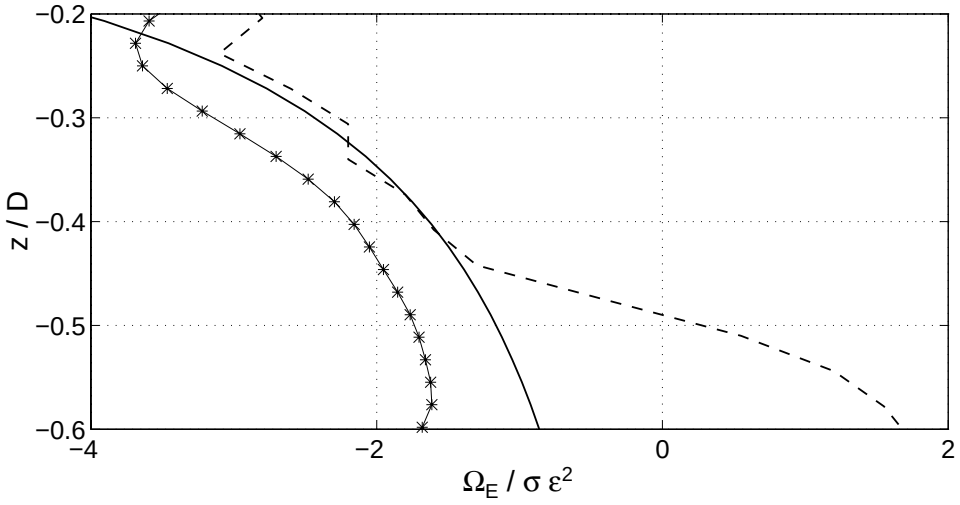


Figure 6.8: Wave-induced Eulerian-mean spanwise vorticity for composite-average of wave simulations compared to Cowen's data (TWCF) and Gerstner wave vorticity profile in flow core ($-0.6 < z/D < -0.2$) – – – Cowen, —*— composite simulation, —— Gerstner wave

6.3.3 Discussion of of near-surface vorticity

Vorticity profiles in figure (6.7) show a sharp spike in the wave-induced vorticity in the free-surface boundary layer that is not present in the TWCF experimental data of Cowen. In the simulation data, the spike in the vorticity can be seen to have a maximum that increases with the wave speed (case $W3$ having the highest wave speed), with the peak getting closer to the free surface. Since Cowen's experiments were run with significantly faster waves than the numerical simulations, we might expect the laboratory results to show a very strong vorticity spike quite close to the free surface. This does not show up in the TWCF experimental data. However, as noted by Cowen (1996):

> *The TWCF measurements are relatively coarse, and due to the uncertainty associated with the wave maker vorticity, it is difficult to evaluate the near-surface vorticity profile quantitatively.*

Because of these difficulties with the TWCF, Cowen used the Recirculating Wave-Current Flume (RWCF) at the Environmental Fluid Mechanics Laboratory at Stanford University to investigate the near-surface vorticity and velocity profiles. The

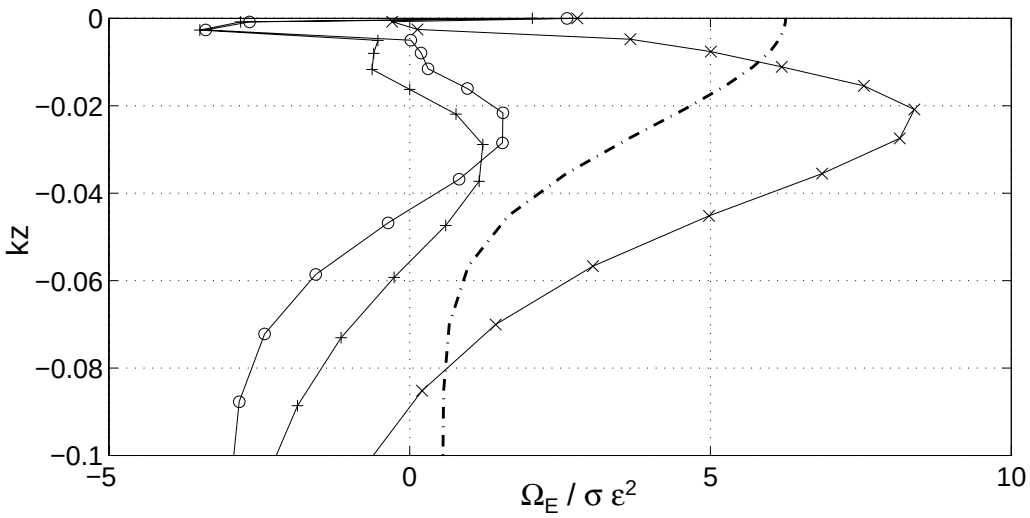


Figure 6.9: Wave-induced Eulerian-mean spanwise vorticity compared to Cowen's curve-fit to experimental data in the near-surface region. $-\cdot-$ Cowen (RWCF I, curve fit), $-|-$ $W1$, $-*-$ $W2$, $-\times-$ $W3$

results of these simulations are reported as RWCF I and RWCF II in Cowen (1996). In figure (6.9), the curve-fit computed by Cowen for the viscosity-induced vorticity in RWCF I is compared to the wave-induced vorticity in the simulations (in this figure the vertical scale is kz for consistency with the results of Cowen [1996]). Note that Cowen's experimental data (not shown) contains a sharp drop in the vorticity for $kz > -0.01$ that is not captured in the curve fit, but is consistent with the behavior of the numerical simulation.

6.3.4 Phase-dependent vorticity effects

The near-surface, phase-dependent, wave-induced vorticity profiles for case $W3$ shown in figure (6.10) provide reasonable qualitative comparison to Cowen's results (his figure [4.15b])[9]. The simulation data show a greater asymmetry of the vorticity profiles than the results of Cowen due to the larger ak in the simulation. The additional steepening near the crest and the flattening at the trough in the simulation causes the magnitude of the phase-averaged spanwise vorticity at the crest to be larger than

[9] We have scaled the vorticity by $\sigma\varepsilon$ rather than $\sigma\varepsilon^2$ used by Cowen. The former appears to be a better scale of the phase-averaged vorticity in the near-surface region for the numerical simulations.

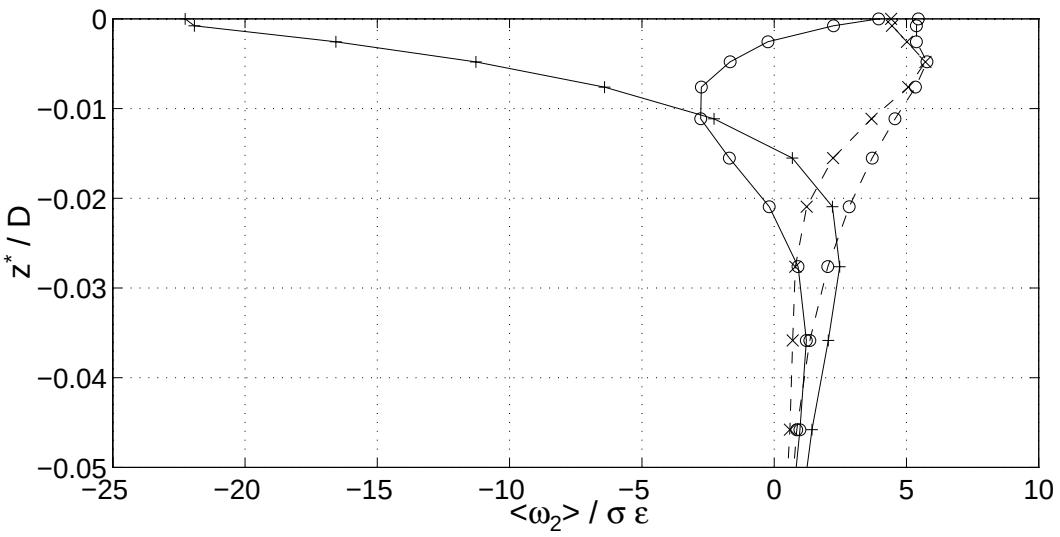


Figure 6.10: Phase-dependent, spanwise vorticity profiles for case $W3$ in near-surface region ($0.05 < z^*/D < 0$). $--\times--$ $kx = -\pi$ (trough), $--\circ--$ $kx = -\pi/2$, $—|—$ $kx = 0$ (crest), $—\circ—$ $kx = +\pi/2$

the magnitude of the vorticity at the trough.

A better picture of the spanwise vorticity generation on the surface can be obtained by plotting the surface vorticity versus the wave phase as shown in figure (6.11). For reference, the wave shape is shown at the top of the figure and an arrow indicates the direction of wave propagation. It is not clear why there is a small spike in the wave-induced vorticity at the trough near the boundary. It was observed in the simulation data that the low point of the trough did not actually occur on the boundary, but was instead located about three grid cells inside the boundary at the trailing edge of the wave. This is coincident with the start of the spike in the spanwise vorticity. While it is possible that this phenomena has a physical cause, the more likely explanation is that this is a numerical manifestation of the difficulties in obtaining a smooth boundary-orthogonal and periodic grid in a moving grid simulation. Further investigation is required to determine the cause of the vorticity spike.

By looking at the evolution of the vorticity beneath the wave, we can see that the effect of the spike in the surface vorticity is contained in the free-surface boundary layer and does not significantly affect the interior flow. Figure (6.12) provides a plot of the spanwise vorticity versus the wave phase for a series of curvilinear coordinate lines inside the free surface for case $W3$. The line with the largest magnitude is the

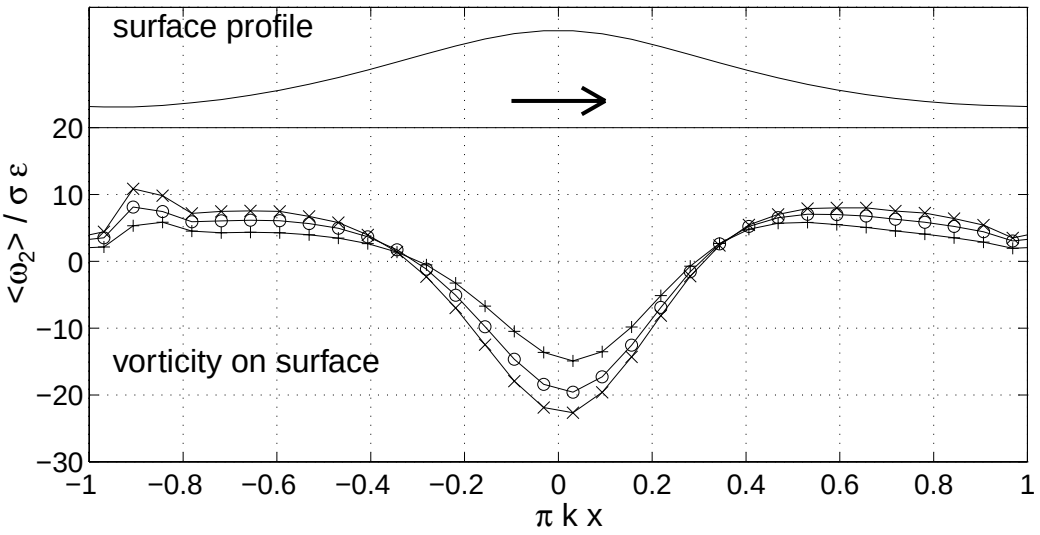


Figure 6.11: Wave-induced spanwise vorticity at the free surface plotted against the wave phase. Arrow indicates direction of wave propagation. —+— $W1$, —∗— $W2$, —×— $W3$

free surface, the next line is 1 $z+$ unit inside the free surface, and the following lines are the succeeding grid lines beneath the free surface. It can be seen that the wave-induced free-surface spanwise vorticity vorticity has decayed to zero within the first ten grid cells beneath the free surface, which is the approximate thickness of the free surface boundary layer.

Figure (6.13) provides a picture of the wave-induced vorticity in the bottom boundary layer in a fashion similar to figure (6.12). In the bottom boundary, the wave induces an oscillation in the generation of vorticity that has its maximum values before the crest and before the trough rather than at the crest and trough as is seen in the free-surface boundary layer.

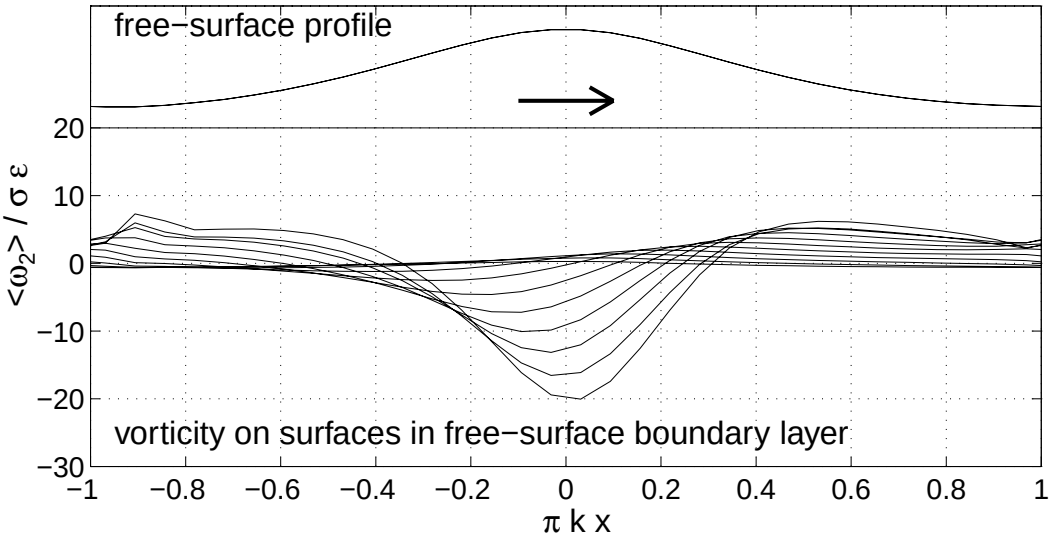


Figure 6.12: Wave-induced spanwise vorticity in free-surface boundary layer plotted against wave phase. Lines are vorticity on successive curvilinear coordinates inside the free surface. Arrow indicates direction of wave propagation.

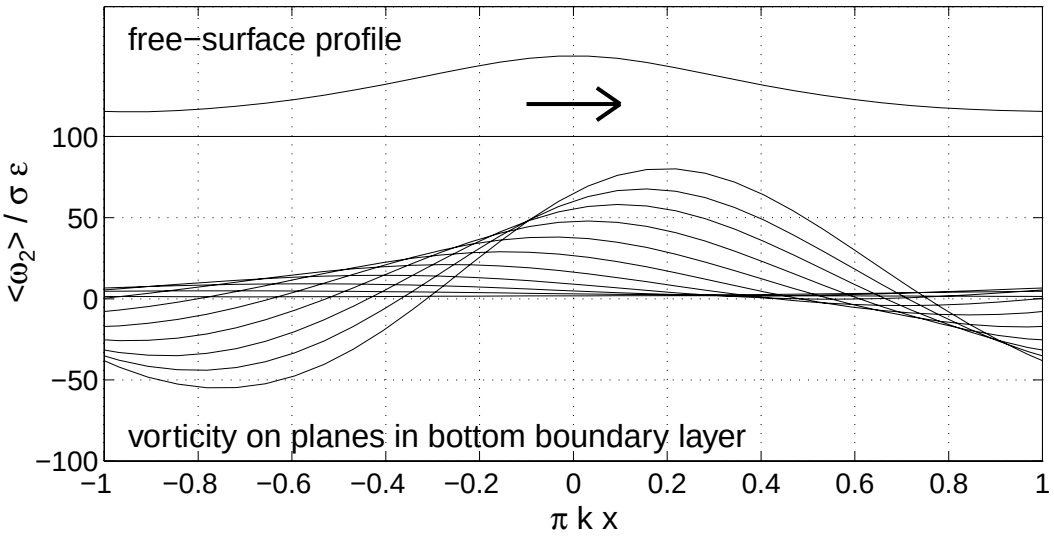


Figure 6.13: Wave-induced spanwise vorticity in bottom boundary layer plotted against wave phase. Lines are vorticity on successive curvilinear coordinates above the bottom boundary. Arrow indicates direction of wave propagation.

6.3.5 Generation of vorticity at crest and trough

We can demonstrate that the vorticity directions at the crest and the trough shown in figures (6.10) and (6.11) are correct by following the argument presented by Longuet-Higgins (1992)[10]. The definition of the spanwise vorticity is:

$$\omega_2 = \frac{\partial u_1}{\partial x_3} - \frac{\partial u_3}{\partial x_1} \tag{6.7}$$

If we note that the exact crest and trough have tangent lines that are horizontal, then the tangential component of the dynamic boundary condition can be written as:

$$\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1} = 0 \tag{6.8}$$

It follows that

$$\omega_2 = -2\frac{\partial u_3}{\partial x_1} \tag{6.9}$$

At the crest, in the direction of wave propagation along the x_1 axis, the u_3 component of velocity has a positive gradient, so the vorticity is necessarily negative at the crest. At the trough, the gradient is reversed, so the vorticity is necessarily positive.

[10]Note that Longuet-Higgins used an $x - y$ coordinate system rather than an $x - z$ coordinate system, which results in the vorticity being of the opposite sign when a right-hand rule is used. However, the actual direction of rotation associated with the surface vorticity will be the same.

6.4 Turbulence and velocity fluctuations

Turbulence is generally defined as the fluctuations of the velocity from a time-averaged mean. If this definition is applied by an observer in a fixed frame of reference for a wave/current system, then the entire wave-induced velocity field will be considered part of the turbulent fluctuations. However, the (mostly) irrotational velocity field produced by periodic wave motion is a different type of velocity fluctuation than is typically associated with turbulence.

Ideally, we would like to be able to decompose an instantaneous velocity field into (1) a temporal mean, (2) periodic fluctuations due to surface waves and (3) non-periodic fluctuations (which can be defined as turbulence). This approach was taken by Jiang, *et al.* (1990) in decomposing experimental results to examine the interaction of waves and turbulence for a spectrum of two-dimensional waves. Cowen (1996) extended the temporal linear filtration technique of Benilov and Filyushkin (1970) to spatial measurements using the assumption that the wave-induced velocity ($\tilde{\mathbf{u}}$) can be computed by averaging the velocity field at discrete phases of the wave. Neither of these techniques is entirely satisfactory for the present simulations due to the presence of three-dimensional short-crested waves on the surface that make significant contributions to the fluctuating velocity field.

In the present work, the use of a wave-following coordinate system allows the definition of turbulence as the fluctuations from the phase-averaged velocity beneath the wave. This follows the approach of Cowen (1996) in removing the periodic motions induced by the primary wave from the definition of turbulence. Using linear theory for short-crested waves (see section 5.4.6), we can estimate the mean velocity fluctuations caused by the largest parasitic waves and remove this effect from the computation of the mean turbulence averaged over a wavelength. An effective method of decomposing the velocity fluctuation effects for the entire spectrum of three-dimensional surface perturbations remains to be developed.

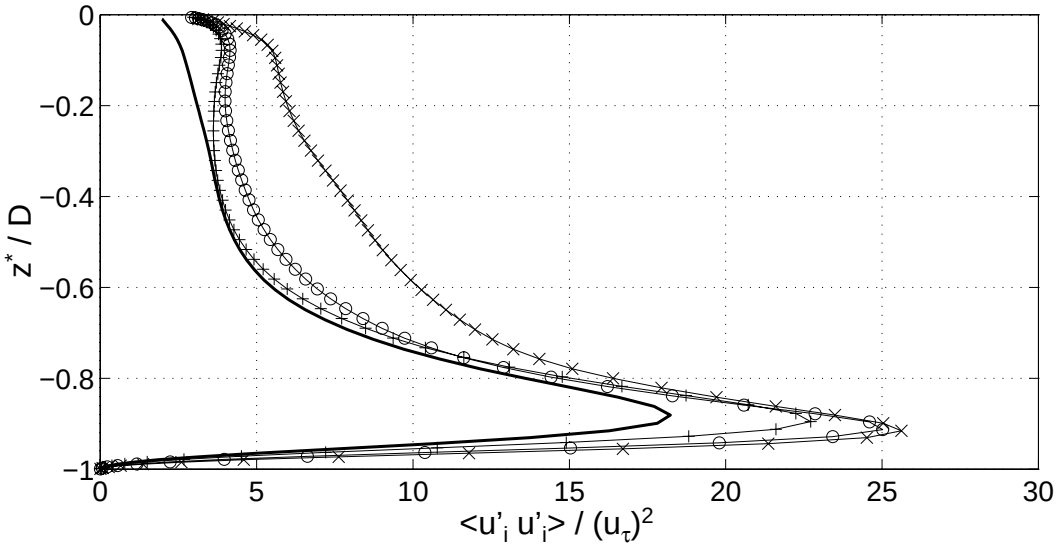


Figure 6.14: Mean total turbulent kinetic energy of velocity fluctuations. —|— $W1$, —o— $W2$, —×— $W3$, —— C (current only)

6.4.1 Total turbulent kinetic energy

Profiles of the total turbulent kinetic energy (averaged over a wavelength) are shown in figure (6.14). The profiles are normalized by the square of the turbulent shear velocity (u_τ) so an increasing trend with increasing wave speed[11] in the core of the flow is not unexpected: the higher-speed wave should have a more energetic interaction with the current.

Unfortunately, there is not a single scale by which the turbulence data can be normalized to obtain a collapse to a single line for various wave speeds. When the data in figure (6.14) is scaled by the appropriate wave scale ($c^2\varepsilon^2$), the relationship (not shown) between the profiles for the waves is reversed and the scaled data are not collapsed. Slightly better agreement occurs (not shown) when the scaling is on the square of the difference between the wave speed and the mean current. These results indicate that the appropriate scaling requires some combination of the turbulent and wave scales to properly collapse the data, and the proper scales may be a function of the vertical position in the flow. Scaling on the product of the wave speed and u_τ, was

[11]The slowest wave is $W1$, the fastest is $W3$. In relative terms, $c_{W2} = 1.5\,c_{W1}$, and $c_{W3} = 2.5\,c_{W1}$.

also investigated, but the data collapse was not consistently improved. The variations of wave-induced turbulence with wave speed seen in the simulations is consistent with Cowen's (1996) observation of increased viscous streaming in a turbulent boundary layer compared to the laminar solution. These effects can be attributed to three-dimensional interactions between the wave and the turbulent boundary layer.

By subtracting the turbulent profile of the current from the wave/current profiles, we arrive at the wave-induced turbulent profiles shown in figure (6.15). It can be seen the the wave induced effects do not go to zero at the surface, but collapse towards u_τ^2. It is interesting to compare the profiles in figure (6.15) to a profile of the turbulent fluctuations when the parasitic waves have *not* been removed, as shown in figure (6.16). The fluctuations due to the parasitic short-crested waves have a significant effect on the profiles for the upper third of the domain beneath the surface. From a comparison of these two graphs, it appears that there may be additional higher-mode parasitic waves on the surface that are responsible for some of the increased fluctuations in the range of $-0.2 < z^*/D < 0$. Such higher-mode waves would have velocity effects that decay faster than the effect of the low-mode parasitic waves.

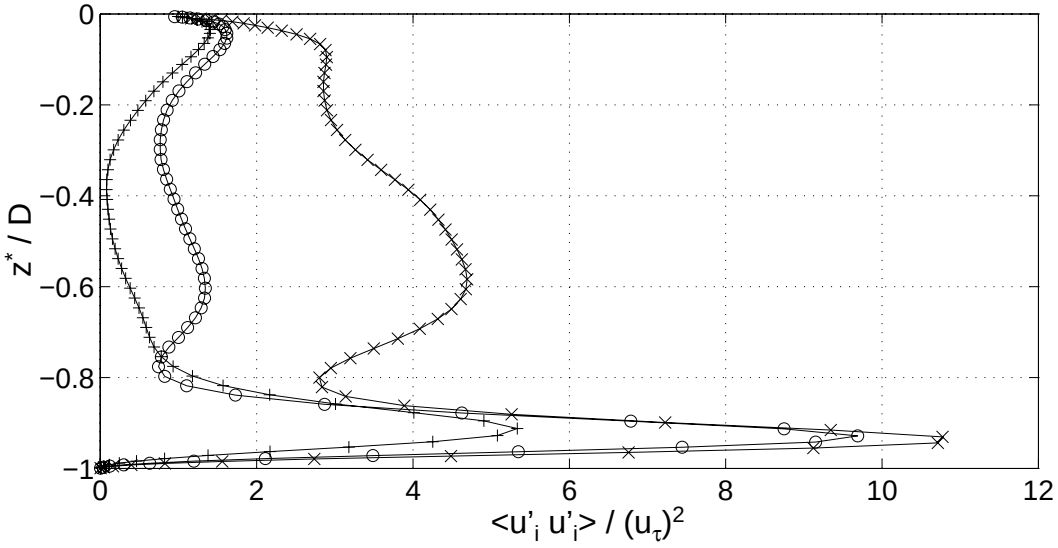


Figure 6.15: Mean wave-induced turbulent kinetic energy of phase-averaged velocity fluctuations (short-crested wave effects removed). —|— $W1$, —o— $W2$, —×— $W3$

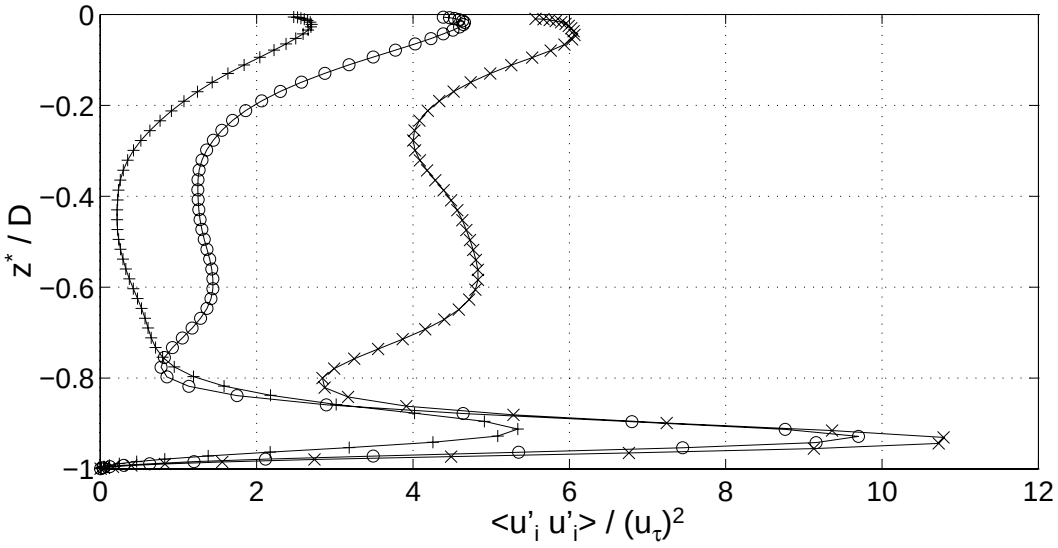


Figure 6.16: Mean wave-induced turbulent kinetic energy of phase-averaged velocity fluctuations (short-crested wave effects included). —|— $W1$, —o— $W2$, —×— $W3$

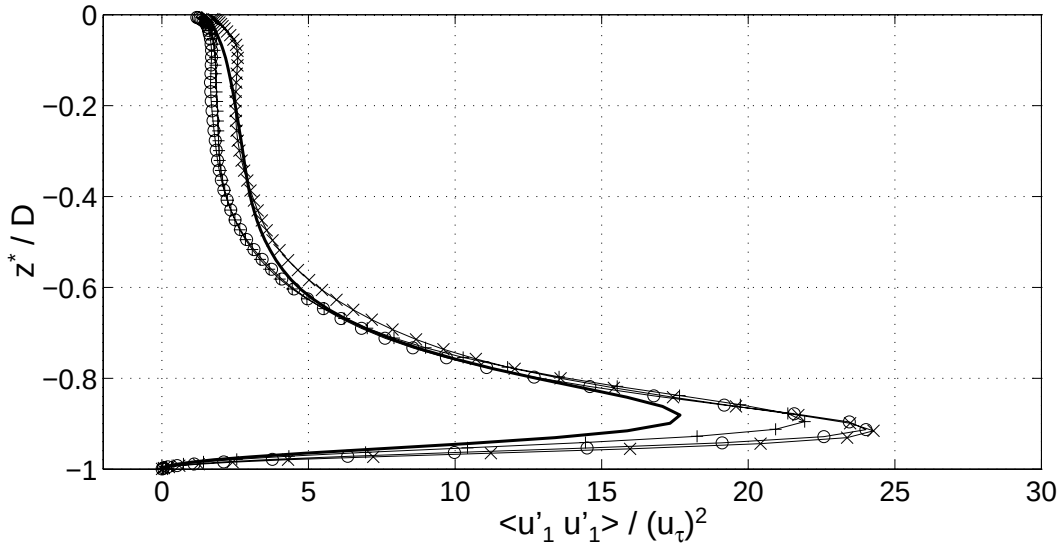


Figure 6.17: Mean total streamwise turbulent intensity of velocity fluctuations. —+— $W1$, —○— $W2$, —×— $W3$, —— C (current only)

6.4.2 Streamwise turbulence intensity

Comparison of the total turbulent kinetic energy in figure (6.14) and the streamwise contribution to the kinetic energy shown in figure (6.17) reveals that the streamwise fluctuations are the primary source of turbulence in both the current-only flow and the wave-current flows. The most obvious conclusion that can be drawn from figure (6.17) is that the primary effect of the wave on the streamwise fluctuation is in the viscous streaming effect in the bottom boundary layer. However, the behavior of the streamwise fluctuations in the flow core is also interesting and can be better seen in a plot of the wave-induced streamwise fluctuations shown in figure (6.18). In this figure the reduction of the intensity of the streamwise fluctuations in the core for flow $W1$ and $W2$ is obvious. Normally, a negative value for mean square fluctuations indicates an error in the data processing methods; however, in this case, the profiles in figure (6.18) are the difference between the fluctuations in the wave/current flow (minus the effect of the parasitic waves) and the current-only flow. As such, the region of negative net turbulent energy indicates that the monochromatic wave is redistributing a portion of the kinetic energy from the core of the flow.

The fastest wave, $W3$ seems anomalous in that the intensity in the core is larger than the intensity of the current-only simulation except for a very small region in the core. Further investigation is required to determine if the physical processes associated with the wave-turbulence interactions of the faster wave are different from that of the slower waves.

Figure (6.19) is a plot of the wave-induced streamwise turbulence effects with the short-crested wave fluctuations included. As before, we see that the effect is confined to the upper third of the domain where there is a significant increase in the computed streamwise fluctuations.

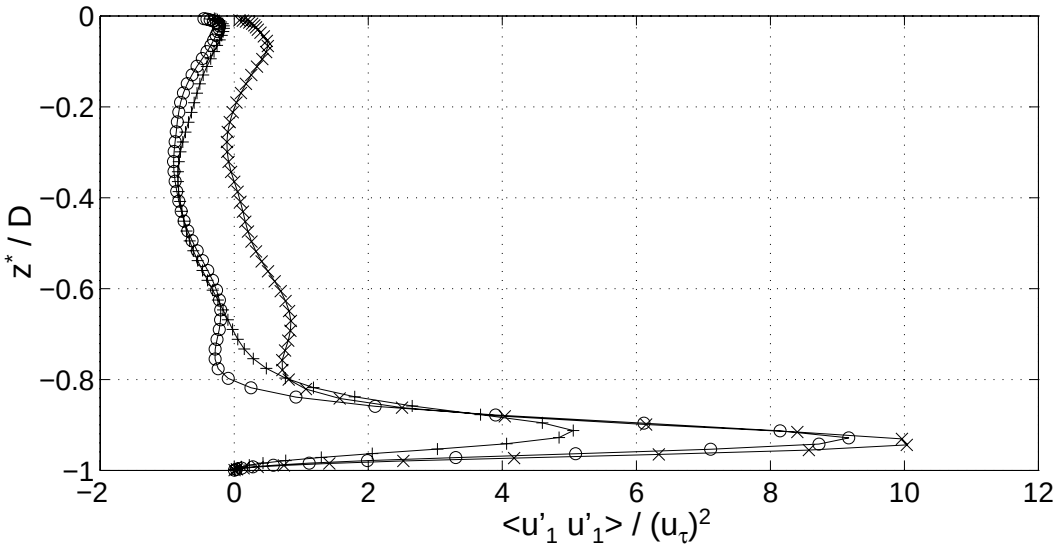


Figure 6.18: Mean wave-induced streamwise turbulence intensity of phase-averaged velocity fluctuations (short-crested wave effects removed). —+— $W1$, —o— $W2$, —×— $W3$

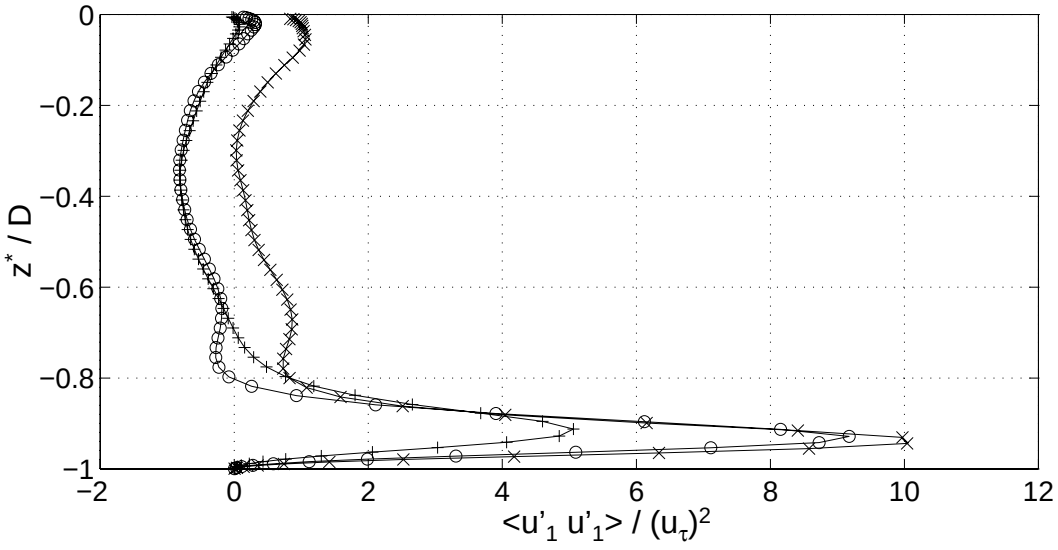


Figure 6.19: Mean wave-induced streamwise turbulence intensity of phase-averaged velocity fluctuations (short-crested wave effects included). —+— $W1$, —o— $W2$, —×— $W3$

6.4.3 Spanwise turbulence intensity

The total spanwise turbulence intensity for the wave and current is shown in figure (6.20). Note that the effect of the parasitic waves has been removed from these profiles, but there remains a significant increase in the turbulence intensity in the near surface region. In the bottom boundary layer we see that the wave in the streamwise direction causes an increase in the spanwise turbulence intensity. In figure (6.21) the wave-induced spanwise turbulence intensities are scaled by the product of the square of the turbulent shear velocity and the ratio of the wave speed to the current speed. This scaling collapses the data throughout the domain for this component of the turbulence[12]. From this scaling it can be argued that the spanwise intensification of the turbulence by the wave is at least partially due to three-dimensional interaction of the spanwise turbulence of the current with the streamwise and vertical strains of the wave. Additional contributions to the spanwise turbulence intensity appear to arise from the decay of the vertical turbulence intensity in the near-surface region shown in section 6.4.4. Figures (6.22) and (6.23) show the wave-induced portions of the spanwise turbulence intensity scaled by the square of the turbulent shear velocity.

[12] This scaling was applied to other turbulence components with no significant improvement in the collapse of the data.

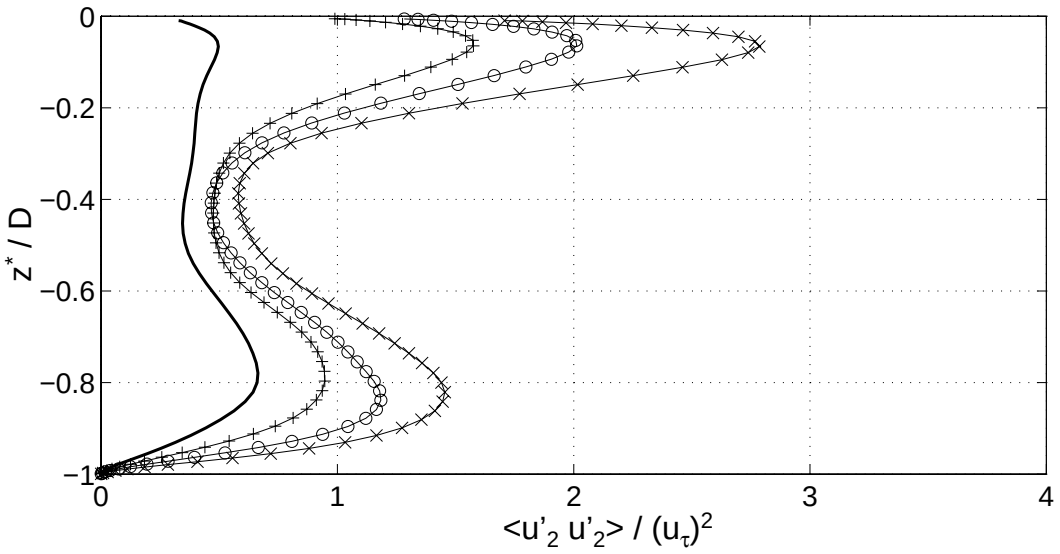


Figure 6.20: Mean total spanwise turbulent intensity of velocity fluctuations. —|— $W1$, —o— $W2$, —x— $W3$, —— C (current only)

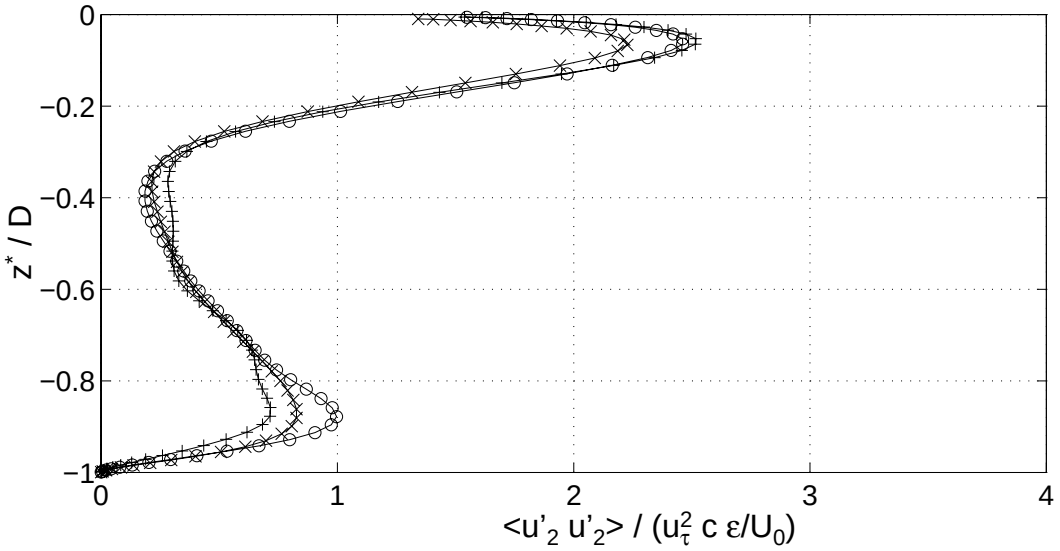


Figure 6.21: Mean wave-induced spanwise turbulence intensity of phase-averaged velocity fluctuations, scaled on product of the turbulence shear velocity and a ratio of the wave speed and current (short-crested wave effects removed). —|— $W1$, —o— $W2$, —x— $W3$

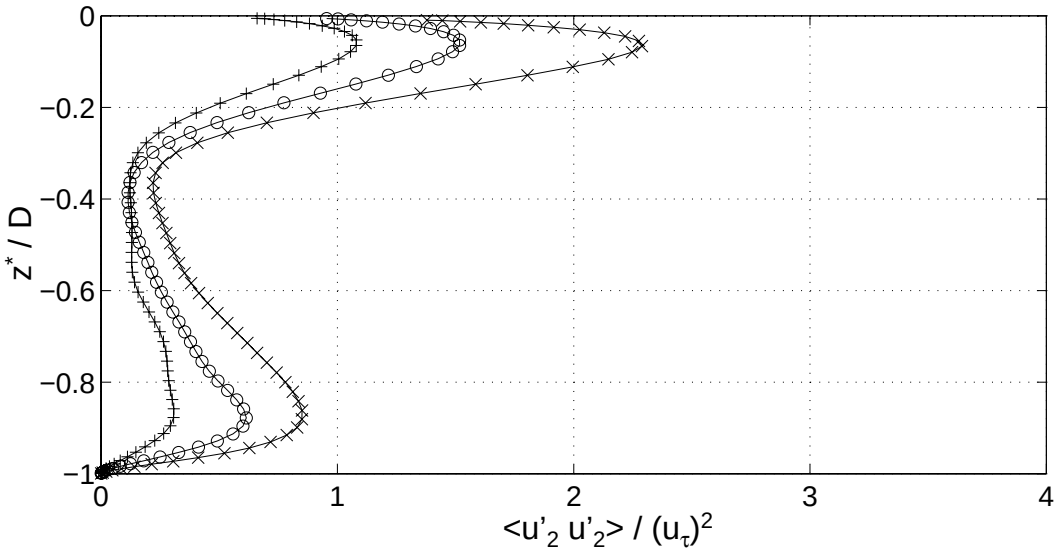


Figure 6.22: Mean wave-induced spanwise turbulence intensity of phase-averaged velocity fluctuations (short-crested wave effects removed). —|— $W1$, —o— $W2$, —×— $W3$

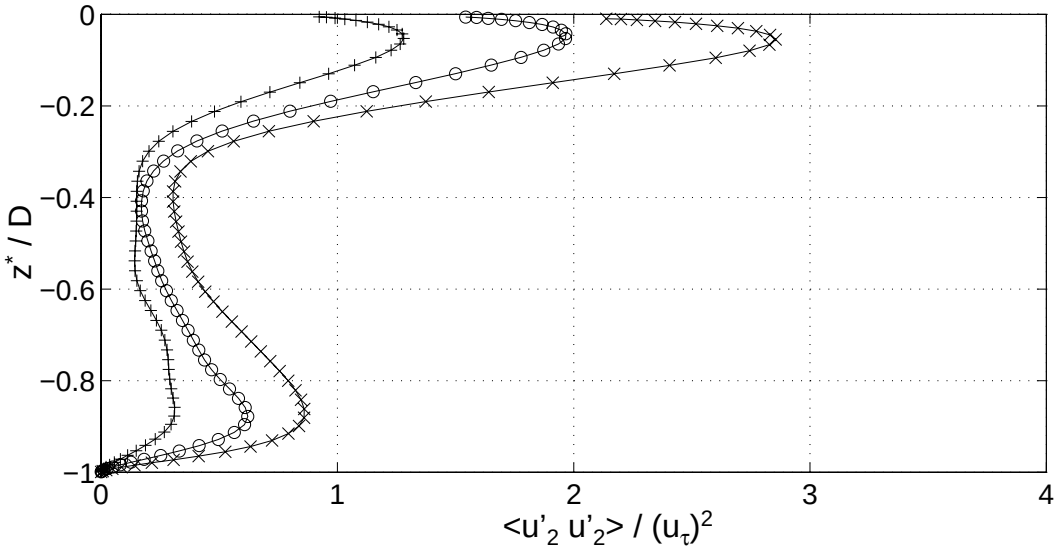


Figure 6.23: Mean wave-induced spanwise turbulence intensity of phase-averaged velocity fluctuations (short-crested wave effects included). —|— $W1$, —o— $W2$, —×— $W3$

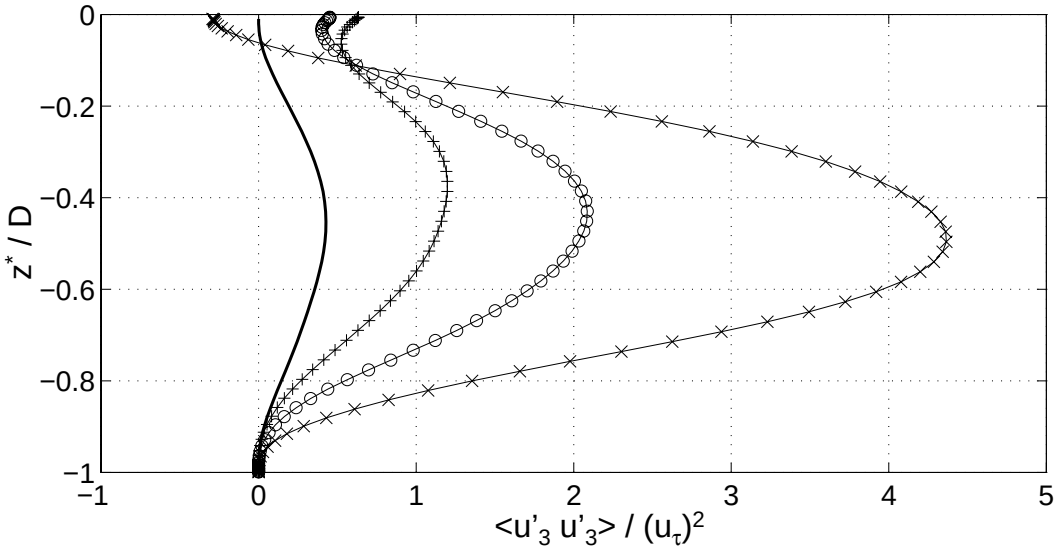


Figure 6.24: Mean total vertical turbulent intensity of velocity fluctuations. —+— $W1$, —o— $W2$, —×— $W3$, —— C (current only)

6.4.4 Vertical turbulence intensity

The total vertical turbulence intensity shown in figure (6.24) and the wave-induced vertical turbulence intensity shown in figure (6.25) are similar since the vertical fluctuations in the current-only case are small. From these graphs we can see that the wave/current interactions have a dramatic effect on the vertical turbulence intensity in the the core of the flow[13].

As one might expect, the vertical turbulence intensity decays as the surface is approached. The apparent negative turbulence intensity (relative to the current-only flow) shown in figure (6.25) is an artifact of the approximations used in removing the effect of the parasitic waves.

The increase in the maximum vertical fluctuations with increasing wave speed shown in figure (6.25) can be collapsed (to some extent) by scaling on the wave speed as shown in figure (6.26). In this figure we see that the scaling on the wave speed is

[13]Recall that our calculation of mean turbulence intensities is from phase-averaged data so the mean motions of the primary wave are removed.

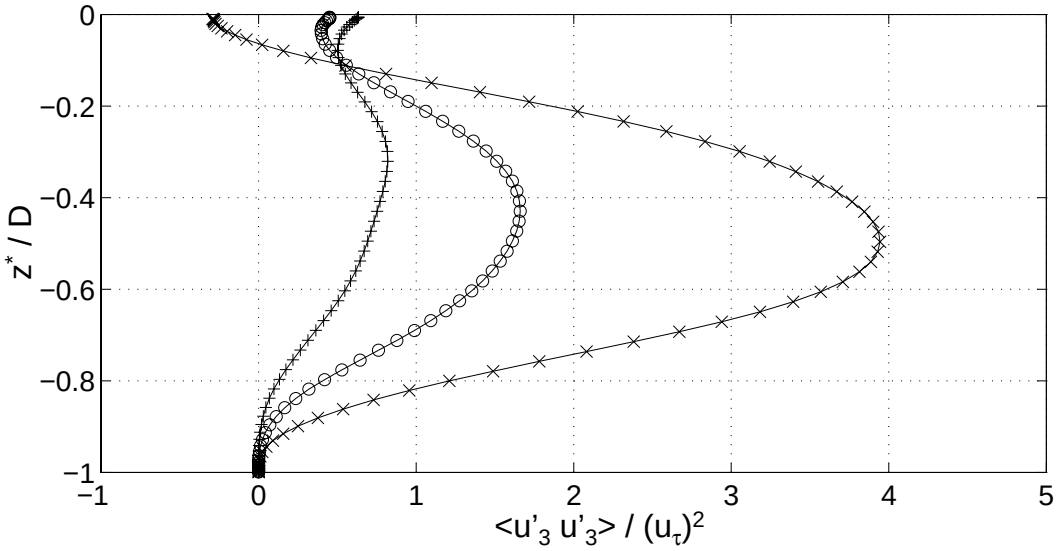


Figure 6.25: Mean wave-induced vertical turbulence intensity of phase-averaged velocity fluctuations (short-crested wave effects removed). —+— $W1$, —o— $W2$, —×— $W3$

not entirely satisfactory since since the trend in the upper half of the flow shows a reversal of the original trend shown in figure (6.25). Relative to wave speed scaling, the faster wave has less wave-induced turbulence than the slower wave. The use of wave scaling accentuates the near-surface positive shear of the vertical turbulence intensities. In figure (6.27), the wave-induced vertical turbulence intensities with the parasitic wave effects included are shown based on the same wave scaling used in figure (6.26). A comparison of figures (6.26) and (6.27) supports the conjecture that there are smaller parasitic waves on the surface that have not been removed and are causing the near-surface shear of the vertical turbulence intensity

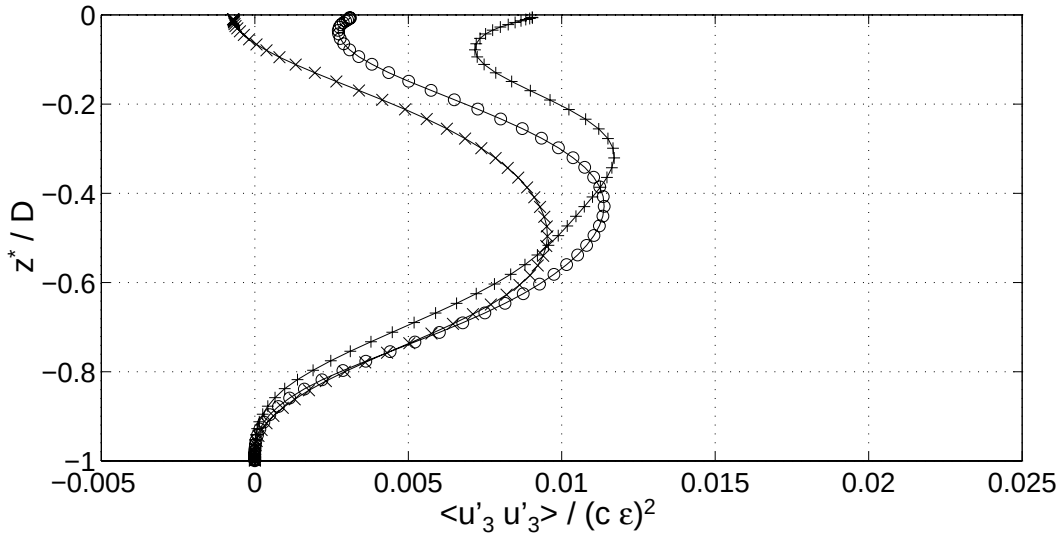


Figure 6.26: Mean wave-induced vertical turbulence intensity of phase-averaged velocity fluctuations, scaled on wave speed (short-crested wave effects removed). —+— $W1$, —○— $W2$, —×— $W3$

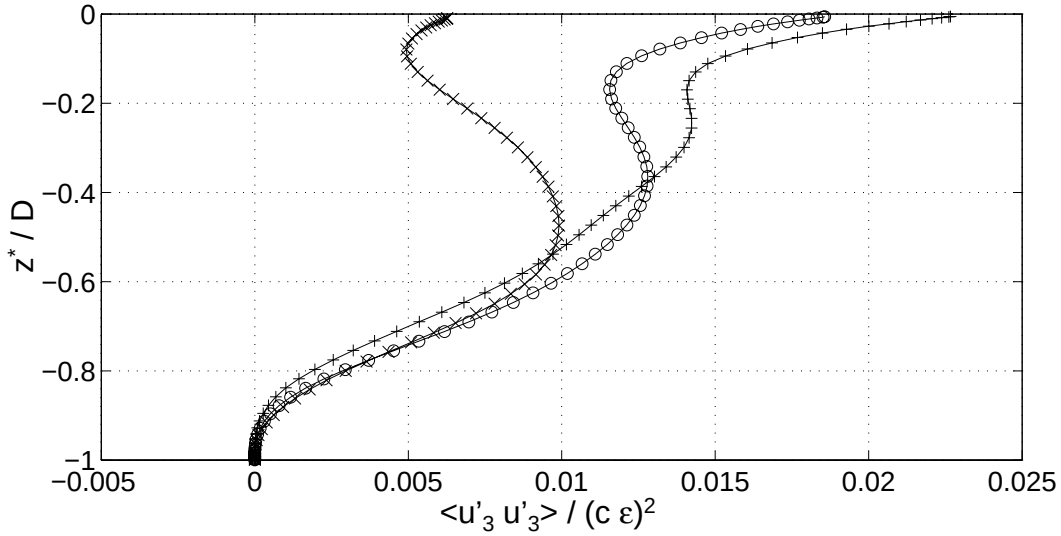


Figure 6.27: Mean wave-induced vertical turbulence intensity of phase-averaged velocity fluctuations, scaled on wave speed (short-crested wave effects included). —+— $W1$, —○— $W2$, —×— $W3$

6.5 Wave-induced fluctuating enstrophy

By subtracting the current-only fluctuating enstrophy components from the wave/current fluctuating enstrophy components for phase-averaged data at different points beneath the wave, we can examine the the effect of the waves on the vortical field in the flow (see section 5.4.2 for a discussion of the product of wave fluctuations and the product of wave fluctuations with current fluctuations that are present in the wave-induced field). Our primary interest in this dissertation lies in the wave-turbulence interactions in the area between the lower shear layer and the free-surface Stokes layer. This is roughly a range of $-0.7 < z^*/D < -0.1$, where z^*/D is the nondimensional distance measured from the free surface. In the following analysis, we will limit the vertical axes of our graphs to this range so that the strong enstrophy gradients in the boundary layers do not distract us from our analysis of the data in the core of the flow.

6.5.1 Wave-induced streamwise enstrophy

Figures (6.28), (6.29) and (6.30) show the wave-induced streamwise fluctuating enstrophy for the three wave/current cases. These profiles are similar for all three cases, showing that all the waves induce a net increase in the streamwise enstrophy, with the strongest increases being beneath the trough, and the weakest being beneath the crest. This indicates that as the flow goes beneath the trough, the vortex lines are stretched in the streamwise direction. As is discussed in most fluid dynamics texts (*e.g.* Batchelor [1967]), it can be argued that the stretching of vortex lines should cause an increase in the vorticity component in the direction of the stretching so as to conserve angular momentum. Since we are periodically expanding and contracting the flow as waves pass over a point in the flow, the stretching of the vortex lines should show up primarily as an increase in the enstrophy (rather than in the mean vorticity). This is consistent with our argument in section (5.4.2) that the product of the wave fluctuations with themselves $< \omega'_w \omega'_w >$ should be small and the primary wave-induced effect is in the product of the wave fluctuations with the current

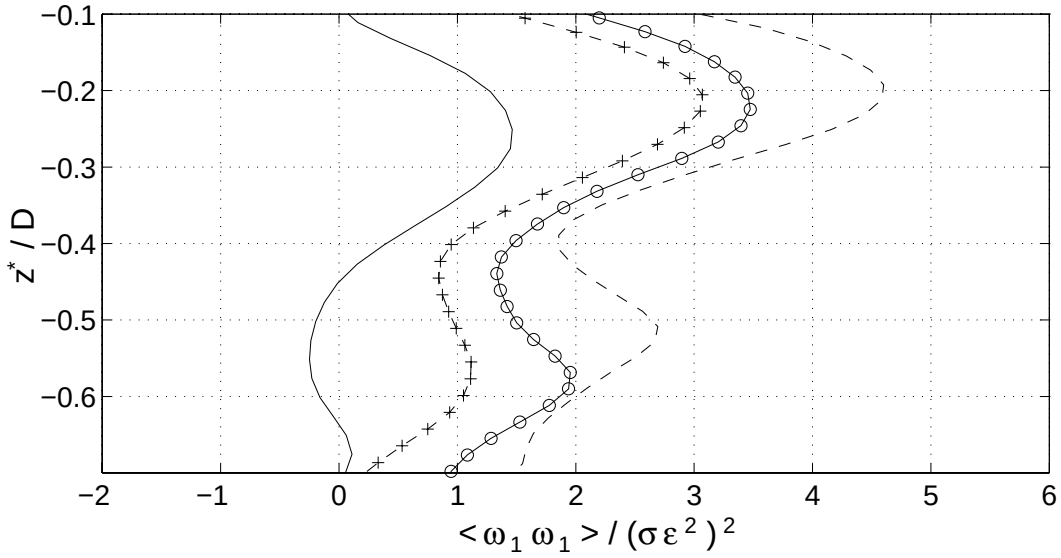


Figure 6.28: Wave-induced streamwise fluctuating enstrophy profiles in flow core for case $W1$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $\_\_$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

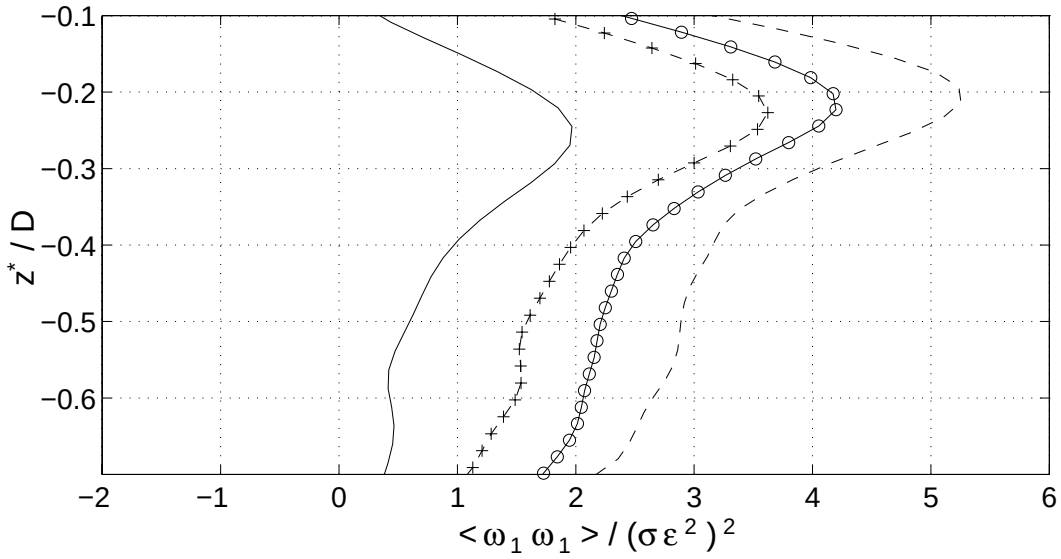


Figure 6.29: Wave-induced streamwise fluctuating enstrophy profiles in flow core for case $W2$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $\_\_$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

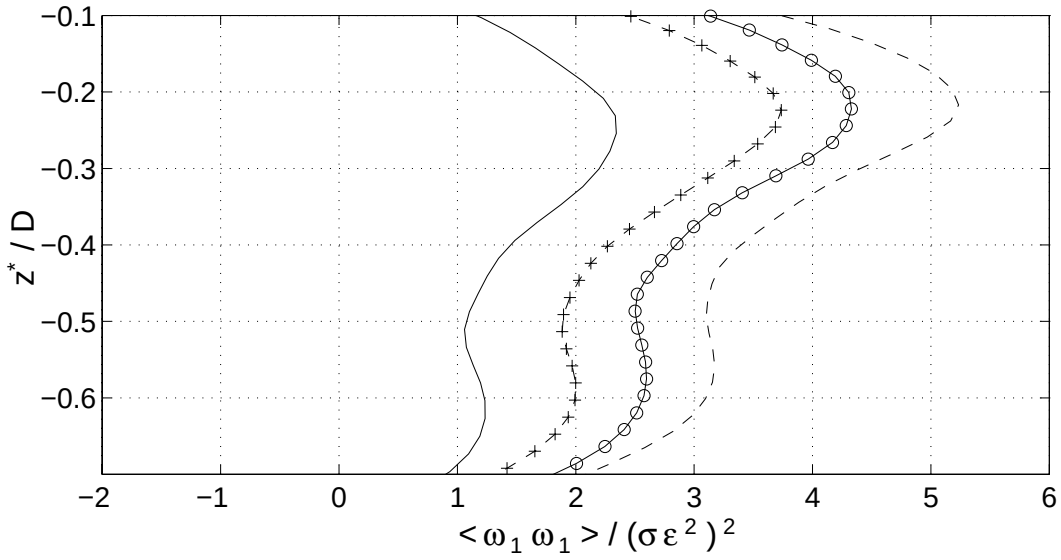


Figure 6.30: Wave-induced streamwise fluctuating enstrophy profiles in flow core for case $W3$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $\_\_$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

fluctuations, $< \omega'_w \omega'_c >$. The wave-straining field interacts with the vortical field produced by the bottom boundary; this interaction provides the wave-induced enstrophy effects.

There are two points that are particularly noteworthy in the streamwise enstrophy data: (1) there is a net streamwise enstrophy increase beneath the wave crest as well as beneath the trough (albeit the crest increase is smaller), and (2) the data shows an asymmetry about the crest in the $\pi/2$ and $-\pi/2$ lines. These observations are discussed in the following paragraphs.

The enstrophy increase beneath the wave crest is interesting because a vortex line compression argument beneath the crest would lead one to expect that the streamwise enstrophy should be reduced in this region when compared to the current-only flow (resulting in a negative value for the wave-induced vorticity). Since the wave-induced vorticity is positive beneath the crest, this implies one or more of the following causes: (1) the streamwise vortex stretching beneath the trough is stronger than the streamwise vortex compression beneath the crest, (2) streamwise enstrophy is transported from the lower or upper boundary layers into the near-surface region, (3) the wave strains are transforming spanwise or vertical enstrophy into streamwise vorticity. The most likely cause is (1); since the profiles through all phases of the wave

are the same and all are positive, it can be argued that the shape of the profile is due to dominant stretching effects in the streamwise direction beneath the trough. The offset of the profiles under other phases of the wave is due to the reduced stretching in these areas. These effects will be discussed in greater detail in conjunction with analysis of a time-series of the phase-averaged data in section (7.3).

The asymmetry of the streamwise enstrophy about the crest for the points $\pi/2$ and $-\pi/2$ is due to the asymmetry in the wave. Although it is not directly apparent in the animations, the initial wave shape (which is symmetrical) undergoes a steepening along the wave front so that there is a slight asymmetry to the wave shape. Although the asymmetry is small, we can see from figures (6.28), (6.29) and (6.30) that it has a significant effect on the wave-induced enstrophy profiles on either side of the crest.

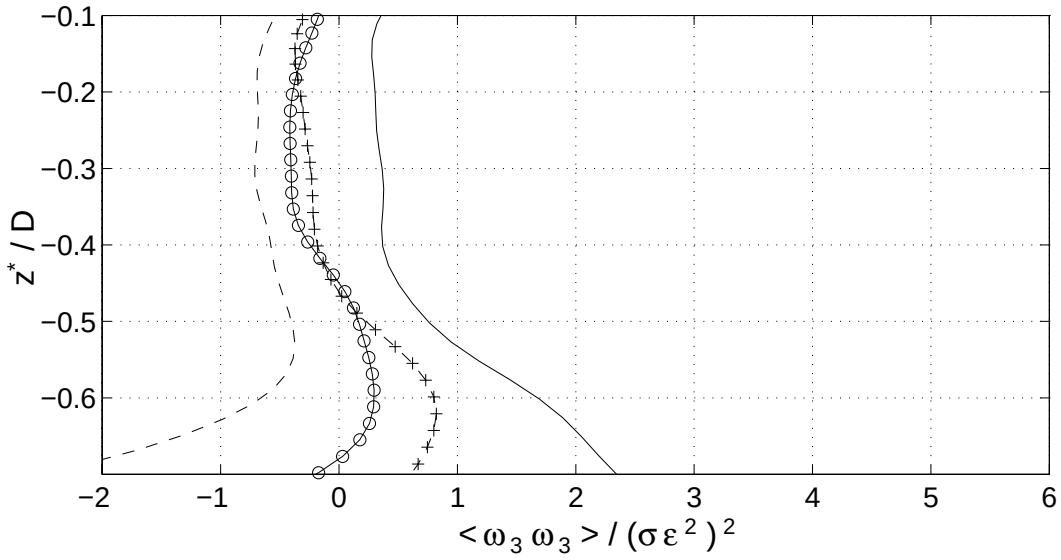


Figure 6.31: Wave-induced vertical fluctuating enstrophy profiles in flow core for case $W1$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $——$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

6.5.2 Wave-induced vertical enstrophy

In figures (6.31), (6.32), and (6.33) the mean, phase-averaged, wave-induced, vertical fluctuating enstrophy is shown for the wave/current simulations. Qualitatively the graphs are similar, showing that the vertical enstrophy is reduced under the trough and increased beneath the crest due to the passage of the wave. There is an interesting quantitative anomaly in figure (6.33) which shows the data for the fastest wave, case $W3$. For the other wave cases, the wave-induced enstrophy beneath the forward and reverse slopes of the wave goes to zero as the free-surface boundary layer is approached, and the effect of the trough is to reduce the vertical enstrophy below that seen in the current-only flow. However, for the fastest wave we see the vertical enstrophy goes to a positive value as the free-surface boundary layer is approached. This indicates that the higher speed wave has a greater vertical enstrophy beneath the trough than the current-only case, so there is another mechanism besides vortex stretching at work for this wave. This additional mechanism appears to be related to the spanwise wave-induced enstrophy discussed below.

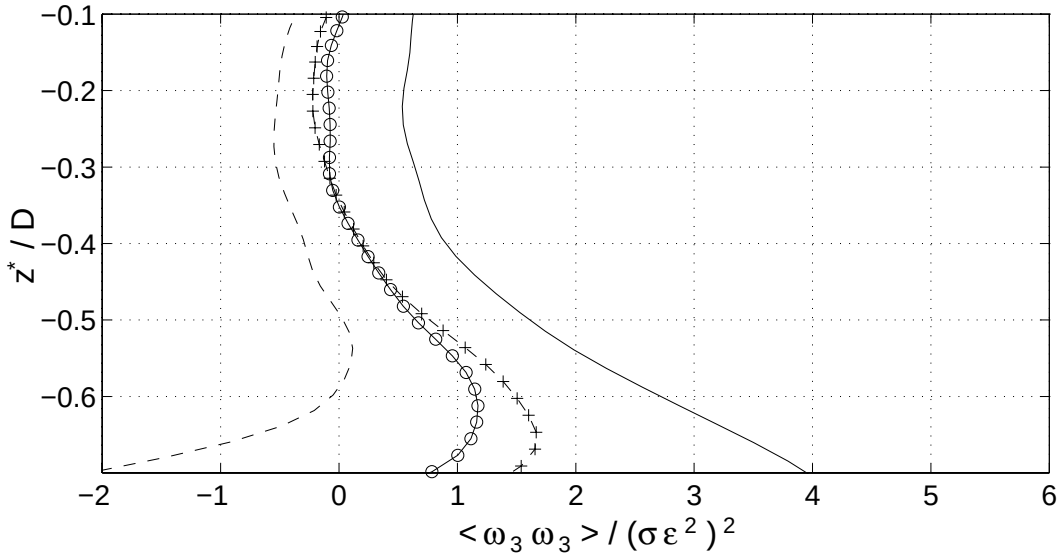


Figure 6.32: Wave-induced vertical fluctuating enstrophy profiles in flow core for case $W2$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $\_\_$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

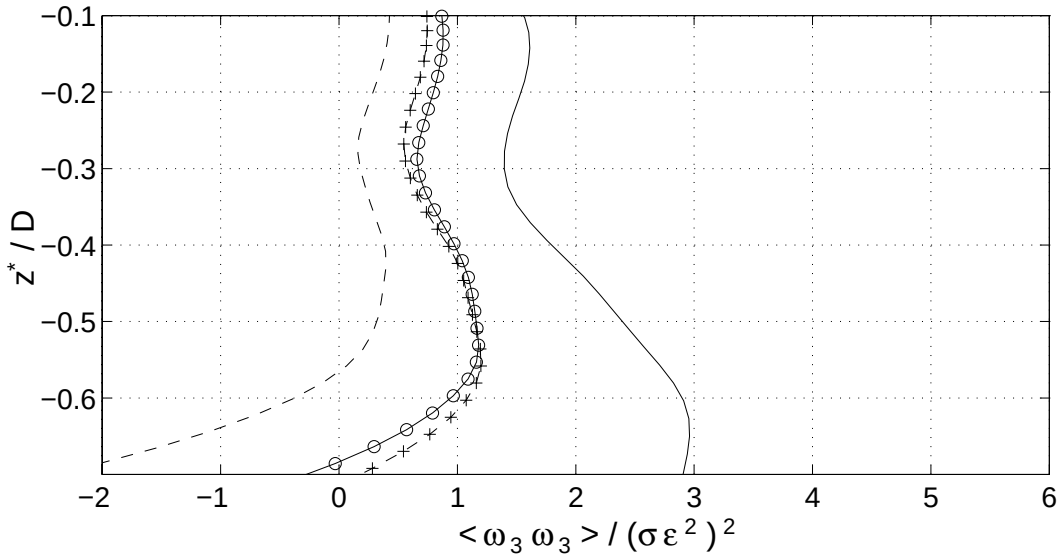


Figure 6.33: Wave-induced vertical fluctuating enstrophy profiles in flow core for case $W3$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $\_\_$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

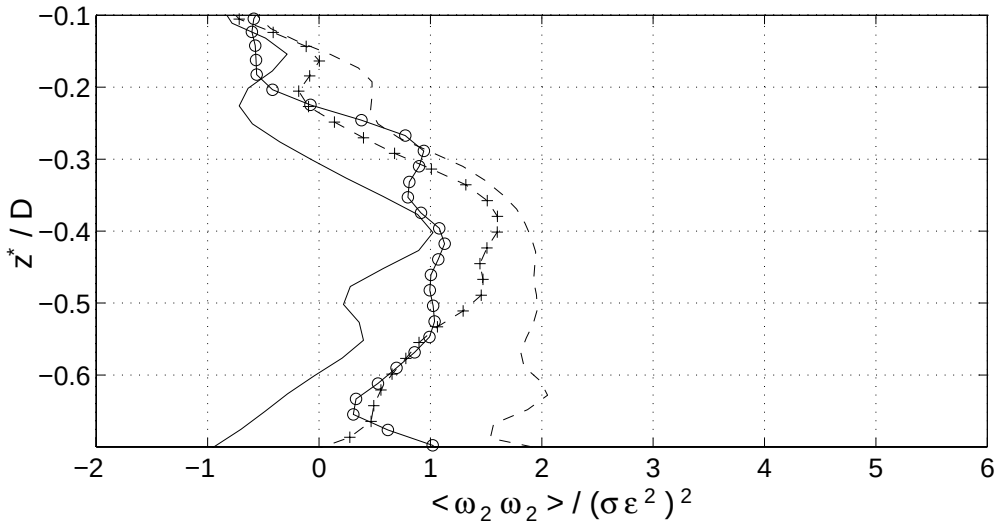


Figure 6.34: Wave-induced spanwise fluctuating enstrophy profiles in flow core for case $W1$. $---$ $kx = -\pi$ (trough), $—o—$ $kx = -\pi/2$, $——$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

6.5.3 Wave-induced spanwise enstrophy

Figures (6.34), (6.35) and (6.36) present the mean, phase-averaged, wave-induced spanwise enstrophy for the current/wave simulations. The wave-induced spanwise enstrophy shows some weak phase-dependent effects for the low and medium speed waves (cases $W1$ and $W2$), while the high speed wave does not have any significant phase-dependent effects. From these graphs it appears that the wave causes a redistribution of spanwise enstrophy from the upper edge of the bottom boundary layer into the core of the flow. This effect does not appear to scale on the wave vorticity scale, but shows an effect that increases with the wave speed. For case $W3$ the spanwise enstrophy transported from the bottom boundary connects with the spanwise enstrophy produced in free-surface boundary layer; thus, this is the only one of the three cases where the spanwise vorticity is non-zero at the lower edge of the bottom boundary layer.

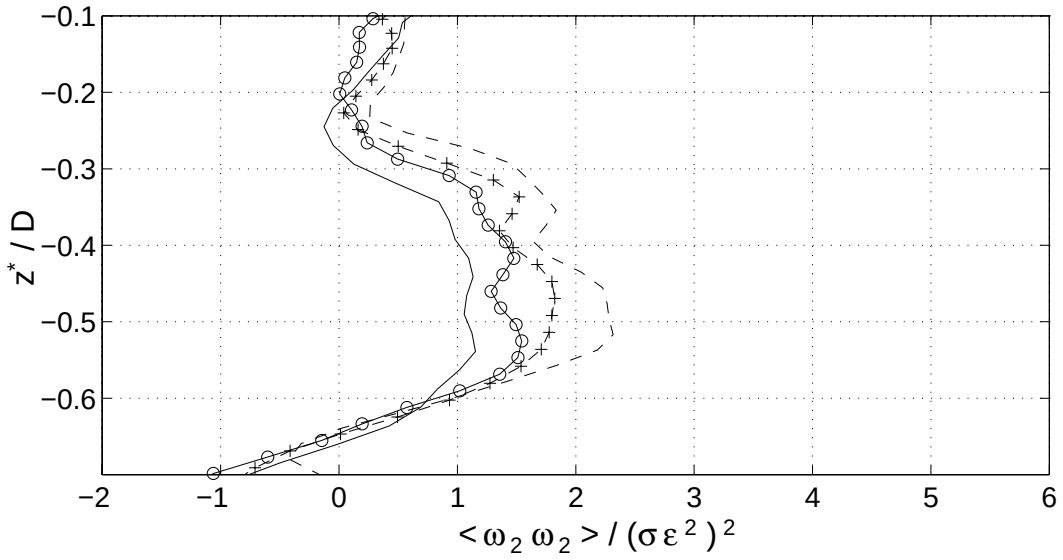


Figure 6.35: Wave-induced spanwise fluctuating enstrophy profiles in flow core for case $W2$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $——$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

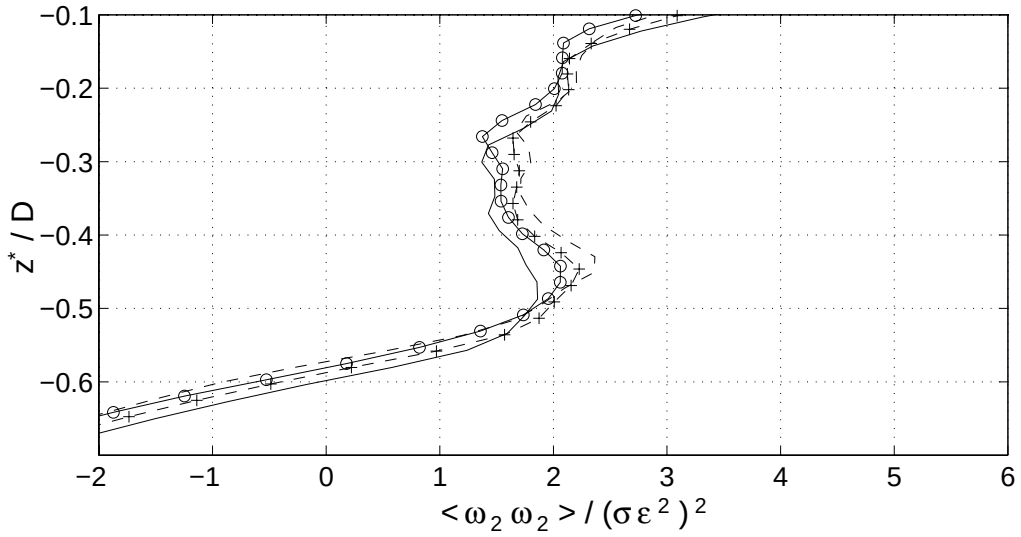


Figure 6.36: Wave-induced spanwise fluctuating enstrophy profiles in flow core for case $W3$. $---$ $kx = -\pi$ (trough), $-\circ-$ $kx = -\pi/2$, $——$ $kx = 0$ (crest), $-+-$ $kx = \pi/2$

Chapter 7

Analysis of instantaneous data

7.1 Introduction

The time-averages of the instantaneous data developed by the simulation provide a quantitative understanding of the mean interactions of the waves and current. However, the temporal kinematics and dynamics of mixing and stirring processes are lost in the temporal averaging. To obtain a better understanding of the temporal evolution of the flow field, we have developed a series of video animations of the simulation data. A VHS format video tape with 20 minutes of animations is included as an attachment to the hard-bound printings of this dissertation[1].

This chapter presents narrations of the animations followed by a discussion of insights gained from analyzing the temporal evolution of the flow field. To supplement the video, a selection of still frames has been extracted and are provided as color figures at the end of this chapter.

7.2 Data animations

7.2.1 Introduction

The video animations of selected simulation data are from wave/current case $W2$ and current-only case C. Computer animations of cases $W1$ and $W3$ were analyzed, and were found to be qualitatively similar to the results in case $W2$. The animations begin about one-third of the way through the simulation, when the wave and current

[1]Readers with a soft-bound copy of this work can obtain a copy of the video from the author for a nominal fee to cover copying and shipping.

have interacted for 10 wave periods. The animations run for 10 wave periods[2]. Unfortunately there is an error in the time counter on the video so that the clock runs at half the correct rate. Thus, the video shows the simulation beginning near $t/T = 5$ and ending at $t/T = 10$.

7.2.2 Observer's reference frame

The observer is in a wave-following system for all of the animations. To provide a similar visual reference, the current-only animations (case C) are shown in a reference frame moving at the speed of the wave used in case $W2$. For animations of data on the central $(x:z)$ plane, the wave is propagating to the right on the television screen. Because the wave is faster than the mean flow, the animations show the turbulent structures in the flow being swept to the left. The inlet and outlet conditions are periodic, so that a structure that moves out of the left side of the screen will appear on the right side. For $(y:z)$ cross-planes, the observer is riding the wave and looking in the negative (upstream) direction so that the positive y direction is to the right of the screen. A vortex structure that begins in the bottom boundary layer and rises to the free surface in the streamwise direction will be seen as a rising vortex as the cross-plane moves over the length of the vortex in the streamwise direction.

7.2.3 Data presentation

Data is presented in two forms: (1) phase-averaged statistics, where the averaging is done in the homogeneous spanwise direction to produce a single $(x:z)$ plane that is representative of the entire flow field, and (2) instantaneous primitive variables for selected data planes.

7.2.4 Color scales

For the phase-averaged data, the color spectrum ranges from dark blue (zero value) to red/brown (maximum value). For instantaneous velocity vectors, the color is

[2]The simulation ran for an additional 10 wave periods beyond the length of the video animations.

used to represent the velocity perpendicular to the viewing plane, and ranges from black (negative maximum) through red and yellow to white (positive maximum). For streamwise planes, the black areas of the screen represent a strong velocity moving towards the viewer ($-y$ direction), while the white areas represent a strong velocity moving away from the viewer ($+y$ direction). In spanwise planes, the black areas are slow-speed fluid (relative to the instantaneous phase-averaged velocity) and the white areas are high-speed fluid.

The temporal wave decay influences the magnitude of statistical effects over time so the color scale in the animations is adjusted to remove the decay trend for the animations of the statistical (turbulence/enstrophy) data for the wave/current simulations. This provides the animations with a consistent color scale so that the structures in the flow can be better visualized.

There was a limited range of color spectrum available for showing the animations. The effective visualization of flow structures is a function of the range represented by the color spectrum since the choice of where red/yellow/blue contrasts appear will define the structures that are visualized. Our interest in this dissertation lies in the wave/turbulence interactions above the bottom boundary layer, therefore the color scales were set to emphasize the structures in this region. As a result, where strong effects occur in the bottom boundary layer, the color scale saturates in a red/brown color and the detail of the boundary layer structures is generally lost. If the color scale were defined so that the maximum color only appears for the very largest values, then good resolution is obtained in the bottom boundary layer, but the structures in the core of the flow are lost[3].

The turbulence intensities and enstrophy intensities in the wave/current flow are larger than those in the current-only flow, so the color scales are not identical. In order to visualize the structures in the current-only flow, the color scale covers a range of approximately 75% of the wave/current flow.

In general, an attempt was made to keep the color scales consistent between similar animations (within the guidelines above). However, in some cases it was necessary to readjust the color scales to obtain better definition of the structures.

[3]One example of scaling to see the bottom boundary layer structures is provided to illustrate the structure in the current-only bottom boundary layer that is intensified by the wave.

In these cases, the video animations have labels such as: "1/2 color scale." This indicates that the range of the color scale was reduced by half when compared to the other similar components of data. For example, the spanwise turbulence intensity has an interesting structure, but is smaller than the streamwise turbulence intensity. Without rescaling the color the structures in the spanwise fluctuations would be lost. Colorbars were generally not included on the video, but are provided in figures (7.1) through (7.9) which are similar to the video.

In some of the animations (particularly those of the wave/current $x:z$ plane) there are anomalous pixels of color that appear fixed in relation to the wave and are inconsistent with the color contours in the flow. These spots are a "noise" signal in the graphics program that produced the PostScript^{TM} files that were the basis of the video animations. These anomalies are not part of the simulation data, but appear to be caused by the inability of the PostScript^{TM} driver in Matlab^{TM} to produce a precise color plot for data presented in the wavy domain. Note that these anomalies do not occur in the current-only domain where the underlying grid for the data is rectangular.

7.2.5 Contents of the video

Each animation segment lasts approximately 24 seconds and shows 10 wave periods of data. This represents approximately 2.2 s of real time for a 126 mm wave on a channel depth of 20 mm (wave/current case $W2$ shown in table [5.2]). For consistency, the time clock on the current-only case is non-dimensionalized by the same wave time scale as the wave/current case[4]. The video presents an animation of data for a wave/current case followed by an animation of data for a current-only case[5]. The animations are sequenced in the following order:

1. total turbulent kinetic energy,
2. streamwise turbulent fluctuations,

[4]As previously noted, the time scale on the clock is actually $t/2T$ rather than t/T as shown in the video.

[5]The video segment on free-surface deformation is reversed, showing the current-only data before the wave/current data.

3. spanwise turbulent fluctuations,
4. vertical turbulent fluctuations,
5. total fluctuating enstrophy,
6. streamwise fluctuating enstrophy,
7. spanwise fluctuating enstrophy,
8. vertical fluctuating enstrophy,
9. instantaneous velocity field in $x:z$ plane (3 components)
10. instantaneous velocity field in $y:z$ plane (3 components)
11. free-surface deformation
12. instantaneous horizontal velocity field on free surface (2 components)
13. instantaneous velocity field in $x:z$ plane (2 components)
14. instantaneous velocity field in $y:z$ plane (2 components)

7.2.6 Description of video animations

To describe the animations, it is useful to remember that we are in the reference frame moving with the wave, which is moving faster than the mean flow. Thus, the turbulent flow below the wave appears to be swept from right to left beneath the wave. Our intuition says that if things are moving from right to left, then the left side must be downstream. Of course, to observer that is fixed, the right side is downstream. This effect makes the terms upstream and downstream somewhat confusing, so they will not be used in our descriptions of the flow. It will be convenient to use two different approaches to discuss the animations. First, where the wave/turbulence interaction causes temporal events that are not periodic with the wave (such as bursting events or connections between the bottom boundary layer and the surface), it is useful to describe these events in terms of the effect of the wave passing over the turbulence (*i.e.* a reference frame moving with the mean flow). For turbulent structures that are persistent and are influenced by the passage of more than one wave, we see a periodicity that is linked to the wave. This makes it more convenient to discuss the flow in terms of turbulent structures that enter the domain beneath the leading trough, pass under the forward slope of the wave, the wave crest, the back side of the wave, and finally exit beneath the trailing trough (*i.e.* our wave-following system).

From this viewpoint, the turbulent structures undergo a rapid expansion from the leading trough to the crest, followed by a rapid contraction from the crest to the trailing trough. Since we are using periodic boundary conditions, a structure exiting beneath the trailing trough re-enters the domain at the leading trough.

7.2.7 Turbulent kinetic energy

The video animations for the turbulent kinetic energy include animations of the total turbulent kinetic energy, and the mean square fluctuations of the velocity (turbulence intensities) in the streamwise, spanwise and vertical directions.

The technique we developed for removing the mean wave-induced velocity-fluctuation effects of the short-crested parasitic waves (described in section 5.4.6) cannot be used to remove the phase-dependent fluctuation effects for the instantaneous turbulence intensities. Thus, the video animations of the wave/current flows include the effects of the parasitic waves.

Total turbulent kinetic energy: wave/current

The animation of the total turbulent kinetic energy for the wave/current case shows temporal connection events between the bottom boundary layer and the free surface that are the result of the wave passing over an intense patch of turbulent kinetic energy. Turbulent kinetic energy appears to be pulled off the bottom boundary layer as the turbulent structures in the bottom boundary approach the flow expansion caused by the forward slope of the wave. The connection to the surface is a short temporal event, and it leaves behind a turbulent patch in the near surface region. The connection occurs just before the crest. As the crest moves over the turbulence, the turbulent patch seems to disappear (or be reduced) then it reappears as the flow is contracted on the back side of the wave. After a surface connection event, the turbulent patch will be trapped in the near free surface region and will undergo alternate strengthening and reduction in the successive progression of waves. We see that the turbulent patch is intensified slightly before the trough and slightly before the crest, with reductions in intensity after the crest and after the trough.

Total turbulent kinetic energy: current-only

The animation of the total turbulent kinetic energy for the current-only case shows the slow speed of the turbulence produced by the mean flow at the bottom boundary. Initially, the turbulence field in the core of the flow is the decaying result of a previous bursting event from an earlier time. By the end of this animation segment, we see another bursting event occurring. While some weak connections to the free surface appear, there are no strong effects as seen in the the animation of the wave/current system.

Streamwise turbulence intensity: wave/current

This animation shows the temporal connections of the streamwise turbulence intensity with the forward slope of the wave. In a manner similar to the total turbulent kinetic energy, we see the streamwise fluctuations connect and are intensified on the forward slope, then are reduced beneath the crest. However, rather than reappearing on the back side of the wave (in the contraction) before the trailing trough (as is the case for the total turbulent kinetic energy), the streamwise fluctuations reappear directly beneath the trough. On some occasions, it can be seen that the streamwise fluctuations do not completely disappear on the back side of the wave. Instead they appear to be ejected from the crest and follow an arc down to the trough so that there remains an area with little or no streamwise fluctuations near the surface on the back side of the wave. Beneath the trough we see the streamwise fluctuations are intensified, then rapidly disappear at the start of the expansion on the forward slope of the wave. Figure (7.1a) shows a typical connection event between the free surface and the bottom boundary layer.

Streamwise turbulence intensity: current only

A comparison of this animation to the total turbulent kinetic energy for the current-only flow shows that the streamwise fluctuations are the primary component of the total turbulent kinetic energy. Again, we see the residual of a prior bursting event that has a weak connection with the surface. This is followed by another bursting event toward the end of this segment. The bursting event can also be seen in figure (7.2a).

Spanwise turbulence intensity: wave/current

The structure of the spanwise velocity fluctuations is shown with a color scale that is 50% of the color scale used for the total turbulent kinetic energy and the streamwise turbulence intensity. In this segment, we see that the front and back sides of the wave have periodic intensification of spanwise fluctuations. This effect is also shown in figure (7.1b). There are no strong connection events between the spanwise fluctuations in the bottom boundary (which are less intense) and the spanwise fluctuations near the free surface (which are more intense). Toward the end of this segment we see the periodic intensification of a patch of spanwise turbulent fluctuations near the bottom boundary as the patch passes beneath the wave trough.

Spanwise turbulence intensity: current only

The spanwise velocity fluctuations in the current-only flow are very weak and are only revealed by reducing the color scale to 12.5% of the scale used for the streamwise fluctuations. This segment shows the slow evolution of spanwise fluctuations from a prior bursting event. These fluctuations have weak connections with the surface that are shown in figure (7.2b). In the bottom boundary layer we see intensification of the spanwise fluctuations as the next bursting event is initiated.

Vertical turbulence intensity: wave/current

The vertical velocity fluctuations are shown on the 50% color scale used for the spanwise velocity fluctuations for the wave/current case. Here we see that the vertical fluctuations are periodically strong from the crest to slightly behind the crest, and from the trough to slightly behind the trough as seen in figure (7.1c). When the vertical turbulence intensity grows large beneath the crest or trough, it appears to be ejected into the core of the flow. In some cases the patches of vertical fluctuations reconnect with the trough, in other cases they appear to remain in the core of the flow, with alternating intensification (beneath the trough and crest) and reduction (beneath the front and back sides of the wave).

Vertical turbulence intensity: current only

In order to see any structure in the vertical velocity fluctuations for the current-only case, it was necessary to use a color scale that is only 5% of the color scale used for the streamwise fluctuations. The vertical fluctuations appear only in the core of the flow and are completely suppressed at the free surface and the bottom boundaries. Figure (7.2c) shows the structure of these fluctuations.

7.2.8 Fluctuating enstrophy

The video animations for the fluctuating enstrophy include animations of the total fluctuating enstrophy, and the mean square fluctuations of the vorticity in the streamwise, spanwise and vertical directions.

Total fluctuating enstrophy: wave/current

In viewing the total fluctuating enstrophy we see that the wave crest lifts the vortical fluctuations from the bottom boundary into the near-surface region. We see that the enstrophy is produced in the bottom boundary but is intensified beneath the crest and the trough. After the initial bursting event, the enstrophy appears to be dissipating due to viscous effects in the core of the flow while periodic intensification of patches of enstrophy continues beneath the crest and trough. Figure (7.3a) shows the distribution of enstrophy above the bottom boundary layer due to the vortex stretching effects of the wave.

Total fluctuating enstrophy: current-only

In the current-only flow, the total enstrophy remains trapped in the lower half of the domain. Even as the bursting event occurs near the end of the animation, regions of high enstrophy do not appear to approach the surface. This is shown in figure (7.3b).

Streamwise fluctuating enstrophy: wave/current

This animation shows strong patches of streamwise enstrophy that are intensified beneath the trough and reduced beneath the crest. In figure (7.4a) this effect can be seen with a strong dark red patch of streamwise enstrophy beneath the trough, a weak yellow patch beneath the forward slope of the wave, and a a slightly stronger red/orange/yellow patch on the reverse slope of the wave.

A thin area of strong streamwise enstrophy is seen in the free-surface Stokes layer and the viscous sublayer of the bottom boundary. These effects do not appear to connect directly to the streamwise enstrophy in the core of the flow. For the bottom boundary layer, this apparent lack of connection is likely due to the choice of color scaling. There does appear to be a relationship between the intense areas of streamwise enstrophy in the lower central section of the flow and fingers of streamwise enstrophy being ejected from the viscous sublayer.

Streamwise fluctuating enstrophy: current-only

The streamwise enstrophy in the current-only animation is initially the decaying remnants of a previous bursting event. We see the beginning of intensification of the enstrophy in the viscous sublayer of the bottom boundary as another bursting event occurs. There does not appear to be any significant interaction between streamwise enstrophy and the free surface. Figure (7.5a) shows the structure of the streamwise enstrophy, which is primarily in and near the bottom boundary layer.

Spanwise fluctuating enstrophy: wave/current

The ejection of spanwise enstrophy from the bottom boundary can be seen in the early stages of this animation. The enstrophy in the core of the flow either decays or is collected in a patch that becomes trapped in the near-surface region. This patch is stretched in the streamwise direction beneath the trough and compressed in the streamwise direction beneath the crest (these effects could also be considered as compression in the vertical direction beneath the trough and stretching in the vertical direction beneath the crest). These alternating effects appear to prevent the enstrophy from decaying in the near-surface region. Figure (7.4b) shows a typical

distribution of the enstrophy through the flow. The spanwise enstrophy in the Stokes layer can be seen in the early part of the animation, but it becomes obscured by the enstrophy convected from the bottom boundary.

Spanwise fluctuating enstrophy: current-only

The spanwise enstrophy generated by the prior bursting event is visible in the core of the flow, however it never reaches the surface with any significant magnitude. Some small structures of the spanwise enstrophy in the core of the flow can be seen in figure (7.5b).

Vertical fluctuating enstrophy: wave/current

The vertical enstrophy shows strong intensification as it passes beneath the crest. We see it develop into a periodically oscillating structure that is occasionally connected to the free surface. This structure can be seen in figure (7.4c).

Vertical fluctuating enstrophy: current-only

The vertical enstrophy in the current-only flow shows the typical slow evolution seen for other quantities in the open-channel flow without a surface wave. The connection of the vertical enstrophy with the free surface appears to be with very small magnitudes. The remants of the previous burst from the bottom boundary layer can be seen in figure (7.5c).

7.2.9 Instantaneous three-component velocity fields

The three-component video animations of instantaneous velocity fields on $x:z$ and $y:z$ cutting planes provide arrows illustrating the magnitude and direction of the two components of the fluctuating velocity in the plane with a color contour plot of the velocity component perpendicular to the plane. Color scales for the animations are described in section (7.2.4). The streamwise and vertical velocity fields have had the phase-averaged velocity removed at each point. This allows us to visualize the

velocity fluctuations without the distortion of the mean flow fields of the current and the wave.

Velocity fluctuations in streamwise plane: wave/current

The instantaneous velocity fluctuations at the streamwise mid-plane show the slow evolution of the turbulent structures produced by the bottom boundary layer. At first glance, it is difficult to see the effect of the wave. It is immediately obvious that the primary turbulent fluctuations in the flow in this plane are due to ejections of slow-speed fluid from the boundary layer. As can be observed in the cross-plane animations (which follow the streamwise animations on the videotape), the mid-plane in the $x:z$ direction is on the edge of a slow speed streak. Upon careful examination and comparison of this segment of animation to the current-only case (shown in the next segment), two wave-induced effects emerge: (1) the primary velocity structures near the bottom boundary (in the lower third of the domain) are alternately expanded and compressed in the vertical direction as they pass under a sequence of crest and trough regions, and (2) a small oscillating streamwise velocity field beneath the crest that moves with the wave can be seen. The expansion and contraction caused by the crest and trough appear to slightly compress and elongate the structures in the bottom boundary without significantly altering their shape. However, as a tall vertical structure passes below the crest, the streamwise fluctuations that move at the wave speed can be seen to be distorting the structure and forcing it up into the core of the flow. This appears similar to the classic "sweep/burst" events seen in bottom boundary layers, only in this case the sweeping velocity is periodic with the wave crest. Figure (7.6) provides a time series of velocity data on the streamwise mid-plane.

Velocity fluctuations in streamwise plane: current only

Without the wave forcing, the instantaneous velocities undergo a slow evolution. A portion of a coherent structure cut by the $x:z$ plane can be seen slowly rising toward the surface as part of a sweep/burst event in the lower boundary layer. A time series of the velocity vectors is shown in figure (7.7).

Velocity fluctuations in spanwise plane: wave/current

The instantaneous velocity vectors in the cross-plane appear to be more energetic than the velocity vectors seen in the streamwise $x:z$ plane. However, this is an effect produced by our wave-following coordinate system, which is moving at a rapid speed through a velocity field that is evolving slowly with time and has significant spatial structure[6]. The black and dark red areas of the color contours represent areas of slow speed fluid in the flow (relative to the instantaneous phase-averaged velocity), while the white areas represent high-speed fluid in the flow. We see that the right-hand side of the animation is dominated by a large slow-speed streak that provides a vertical upwelling of bottom-boundary fluid that connects with the free surface. On the left-hand side we see a corresponding downwelling and a dominant high-speed streak. A small, transient slow-speed streak occurs on the left-hand side of the domain and a medium-size persistent slow-speed streak occurs near the center of the domain. The spacing between the streaks is around 10 grid cells (which is approximately 170 z^+ units), with a transient bias that seems to move the medium and small streaks towards the large streak. There are two important points to note: (1) the shape of the slow-speed streaks seen in the cross-plane are the classic "mushroom cap" shapes that have been linked to the ejection of hairpin vortices from the bottom boundary layer (see Nezu and Nakagawa [1993]), and (2) the upwelling and downwelling provide a dominant counter-clockwise circulation in the center of the flow with a weaker clockwise circulation across the periodic boundary. The upwelling and downwelling are persistent and extend up to the free surface. Figures (7.8a) and (7.9a) are two typical instantaneous velocity plots for the spanwise plane.

Velocity fluctuations in spanwise plane: current only

The current-only data appears very similar to the wave/current data with one very important difference: The strong and persistent upwelling/downwelling effects that are seen in the near surface region for the wave/current velocity data appear only as weak transient effects that do not extend to the free surface. For the current-only case the slow-speed streaks appear more evenly spaced, and do not appear to be

[6] This is actually a useful way to get a picture of the entire turbulence field with a two-dimensional animation of a plane.

significantly drawn toward the large slow-speed streak. The velocity fields for the current-only case are also shown in figures (7.8b) and (7.9b).

7.2.10 Free-surface deformation

The deformations of the free surface are shown in a three-dimensional view looking from the *true* upstream side of the domain. Thus, in the wave-following coordinate system we see surface deformations due to the turbulence moving from right to left as the wave passes over the turbulent structures. The color scale in the animations is set by the height of the deformation so as to provide emphasis of the surface shape. Black areas are depressions in the surface while white or yellow areas are local positive increases in the surface deformation. Unfortunately, the colorbar scale shown on the video is actually off by a factor of 2 (dividing the scale by 2 gives the correct non-dimensional surface deformation relative to the flow depth).

Surface deformation: current-only

The current-only surface deformation is shown before the wave-current surface deformation in order to to illustrate the small nature of the surface deformations associated with the turbulence. This animation also shows the slow rate of evolution of the turbulence at the free surface.

Surface deformation: wave/current

Because the wave is large and deformations of the wave are small, an animation of the wave in the wave-following coordinate system does not provide any significant insights into the behavior of the wave/current system. For this segment of the video, the phase-averaged free-surface height was subtracted from the instantaneous free-surface height to leave the free-surface deformation. This phase-averaging removes the primary wave[7]. As is the case for the previous animation, the colorbar is incorrect by a factor of 2, so the actual non-dimensional deformations are half of what the color

[7]This would also remove any small plane waves (in the streamwise direction) that were at fixed phases on the primary wave; however, our analysis showed there were none in the simulations.

scale indicates. This animation is particularly interesting because it shows a persistent set of short-crested small-amplitude parasitic waves that are moving with the primary wave. The small oscillations of the waves are best seen by viewing the videotape in a fast-forward mode. As can be seen in animations of the surface velocity vectors (in the next video segment), the upper-left side of the domain is dominated by a slow-speed flow that is caused by the connection of the boundary-layer slow-speed streak to the free surface. The lower-right side of the domain is dominated by a high-speed streak. This shear in the surface velocities appears to interact with the primary wave to produce the short-crested waves.

7.2.11 Instantaneous surface velocity vectors

Surface velocity vectors: wave/current

As with previous velocity animations, the phase-averaged mean is removed from the instantaneous velocity vectors so the fluctuations in the velocity field due to the turbulence can be seen (in a wave-following system the phase averaging removes both the mean flow of the current and the mean flow induced by the wave). This animation clearly shows the dominant low-speed and high speed streaks at the free surface that account for the shape of the parasitic waves seen in the animation of the surface deformation. Note that behind the wave crest (which is at $x = \pi$) there is a narrow region across the width of the domain where the slow-speed and high-speed streaks appear to be reduced. This effect is seen more clearly by comparisons with the surface velocity vectors in the current-only animation that follows.

Surface velocity vectors: current-only

In comparison with the surface velocity vectors in the wave/current system, this animation shows velocity a field that does not undergo any rapid oscillations. Only if observers carefully follow a single vortex shape will they be convinced that this is an evolving velocity field. If our interest lies in examining the interactions of large-scale turbulent structures with a non-wavy free surface then the time-scale for the simulations and the animations need to be increased so that distortions of the surface velocity field can be seen. For the present work, the wave time-scale is large compared

to the time scale of the large eddies that connect with the free surface, Thus, if a wave is not included in the flow, the surface velocity field appears almost stagnant. The important observation that can be obtained by comparing the current-only and wave-current velocity fields is that the surface wave has a significant effect on the shape and oscillations of the surface velocity field.

7.2.12 Instantaneous two-component velocity fields

For reference purposes, velocity fields in the $x:z$ and $y:z$ planes without the color contours of the perpendicular velocity are provided as the last four segments of the video. The velocity vectors in these animations are identical to those shown in parts (9) and (10) as listed in section 7.2.5. The elimination of the color contours in the 2-component animations provides a clearer picture of some of the velocity effects in the planes.

7.3 Evolution of turbulence and enstrophy

7.3.1 Transport and transformation of turbulence

The most interesting phenomena seen in the animations of turbulence intensities is the manner in which turbulent fluctuations in the near surface region are transformed. We see initially a connection event (during the expansion under the leading slope of the wave) with transport of a discrete packet of streamwise turbulent fluctuations from the bottom boundary region up to the surface. The streamwise fluctuations are transformed into vertical fluctuations beneath the crest, and then into spanwise fluctuations as the packet encounters the back side of the wave. In the trough we see the transformation of the fluctuation packet repeated: streamwise, vertical, then spanwise. This transformation is not entirely clear-cut: there are significant overlaps between the sections, and the effect is periodic so that the intensities vary.

It can be argued that at least part of the mechanism for this effect is a rotation of the axis of the turbulent fluctuations to align with the wave-induced strain. The turbulent kinetic energy in the bottom boundary layer is carried primarily in the streamwise velocity fluctuations. The animations of the instantaneous flow field in a spanwise plane show that large streamwise velocity fluctuations are due to slow-speed streaks in the bottom boundary layer that throw bursts of slow-speed fluid out into the flow core and toward the free surface. As the slow-speed fluid in the core enters the region under the crest there is a tendency for the velocity fluctuations to align themselves along the axis of the primary strain. To first order, this is strongest at a 45° angle in the direction of the surface slope at a point that is $\pi/2$ in front of the crest (on the leading wave slope). The strain tends to rotate the strong velocity fluctuations from the streamwise direction toward the vertical axis. This accounts for the transformation of the fluctuations from streamwise to vertical seen in the animations, as well as the negative values of the mean wave-induced turbulent intensities illustrated in figure (6.18).

It is not entirely clear how the vertical fluctuations beneath the crest are transformed into spanwise fluctuations on the back side of the wave. It may be that this is due to the action of the spanwise shear in the streamwise velocity caused by the low-speed streak at the surface on one side of the domain in combination with

the high-speed streak on the other side. From the animation of the instantaneous velocities on the free surface it is evident that this spanwise shear is relatively small in the region immediately behind the crest, but intensifies on the back side of the wave as the fluctuations move from the crest to the trough. It could reasonably be argued that the intensification of the the spanwise fluctuations in this region is due to the action of this spanwise shear. Another plausible explanation is that the spanwise turbulence intensification is related to vortex stretching in the contraction that is strengthening streamwise vorticity.

7.3.2 Enstrophy transport and intensification

The enstrophy animations show that the wave induces two major changes to the vortical structure of the flow: (1) the expansion of the leading edge of the wave serves to pull vortical structures out of the bottom boundary layer and into the core of the flow, and (2) the vortical structures in the flow core are periodically intensified by the vortex stretching actions of the crest and trough. Without these stretching actions, vortical structures that are ejected into the core would rapidly decay (as shown by the current-only simulations after a bursting event). Thus, the wave (1) increases the vertical stirring of the turbulence from the shear layer in the bottom boundary toward the surface region, and (2) causes the vortical structures in the flow to persist for a longer time than would otherwise be expected.

7.3.3 Discussion of instantaneous velocity fluctuations

The instantaneous velocity fluctuations in the streamwise $x : z$ plane support the evidence of the turbulence and enstrophy animations that the wave effects the flow field by (1) stretching and compressing structures, and (2) by pulling structure on the bottom boundary up into the core of the flow.

Of particular interest is the strong upwelling and downwelling in the the spanwise $y : z$ plane that occurs in the wave/current case, but not in the current-only case. This gives the appearance of being a Langmuir circulation whose direction is initiated by the ejection of the slow-speed streak. However, we do not have enough data available from these simulations to demonstrate conclusively whether or not

this is a Langmuir circulation. Monismith and Magnaudet (1996) demonstrated that Craik-Leibovich (CL) forcing of Langmuir circulations can be interpreted with rapid distortion theory (RDT), and that CL forcing should be present if the conditions for RDT are satisfied. They also developed scaling arguments for the application of RDT based on the time scales of the turbulence and the distortions by the wave strains. Based on their analysis, RDT will describe the wave-turbulence interactions in the simulations when the turbulence length-scale L_τ is larger than a "Leibovich" length scale, L_L, defined as:

$$L_L \;=\; (ak)^{-3/2}\;\frac{\epsilon^{1/2}}{\sigma^{3/2}} \tag{7.1}$$

where ϵ is the dissipation and σ is the wave frequency. For our purposes, all variables can be considered to be non-dimensionalized by the bottom friction velocity (u_τ) and the domain depth, D. Since our simulation domain is only $171\,z^+$ units in depth, our entire flow is in the inner-layer of the bottom boundary layer and the turbulent length scale is $L_\tau \approx \kappa z$, where $\kappa = 0.4$ is the von Kármán constant. The dissipation can be approximated as

$$\epsilon \;\approx\; \frac{u_\tau^3}{\kappa z} \tag{7.2}$$

Using the full depth of the flow we have (in non-dimensional units) $L_\tau \approx 0.4$ and $\epsilon \approx 2.5$. For the different wave cases we have:

$$\begin{aligned} W1\text{:}\quad & L_L \;=\; 0.065 \\ W2\text{:}\quad & L_L \;=\; 0.037 \\ W3\text{:}\quad & L_L \;=\; 0.017 \end{aligned}$$

For all the wave cases we have $L_\tau \gg L_L$. Based on the arguments of Monismith and Magnaudet, this result indicates that RDT applies to the wave-turbulence interactions and the CL forcing should be present.

We can also use the scaling arguments of Belcher *et al.* (1994), to show that RDT should apply to the flow. They demonstrated that rapid distortion theory can be expected to be applied to the "outer" region of the flow beneath a wave if:

$$\frac{k\,u_{\tau(w)}\,T}{2\,\kappa^2} \ll 1 \tag{7.3}$$

where $u_{\tau(w)}$ is the friction velocity at the free surface boundary, T is the wave period, k is the wave number and $\kappa = 0.4$ is the von Kármán constant. The inner layer of

the flow (relative to the free surface) is defined by Belcher *et al.* (1994) as the *"... thin layer in the water near the interface where the turbulent stresses significantly affect flow perturbations..."*, and is shown to scale on $u_{\tau(w)}T$. The scaling arguments they developed are based on a coupled air/water flow where a turbulent shear stress is developed at the free-surface. As we demonstrated in section (6.2.5), the mean shear at the free surface must be zero for our temporally-decaying wave, so $u_{\tau(w)}$ does not exist at the surface. However, just below the free surface our simulations have a mean positive shear in the wave-induced streamwise velocity. We can use this shear to estimate the effective $u_{\tau(w)}$ in the near-surface regions from

$$u_{\tau(w)} = \sqrt{\nu \frac{\partial u_1}{\partial x_3}} \tag{7.4}$$

The non-dimensional velocity shear in the near-surface region can be approximated from figure (6.1) as: 0.5 for $W1$, 1.0 for $W2$, and 2.5 for $W3$. Using $\nu = 1/Re$ and the wave periods and wave numbers from table (5.1), the values for $u_{\tau(w)}$ are 0.054 for $W1$, 0.078 for $W2$, and 0.12 for $W3$. This results in the scaling parameter (computed from equation [7.3]) of approximately 0.02 for each of the simulations. Since this is much less than one, the inequality of equation (7.3) holds and RDT can be expected to apply outside of the near-surface region.

Based on these two entirely different scaling arguments, we can reasonably state that the wave-turbulence interactions in our simulations should behave according to rapid distortion theory. According to Monismith and Magnaudet (1996), this implies that the Craik-Leibovich forcing is present and we might expect to find Langmuir circulations in our flow. There is visual evidence in the velocity fields in the cross-plane beneath the crest that we are developing *some* type of persistent circulations. However, we have two problems that prevent us from concluding that these are Langmuir circulations: (1) we only have data for the cross-plane beneath the crest, and (2) the dominant effect of the persistent slow-speed streak ejected from the bottom boundary layer may be distorting the actual CL forcing. The lack of data in cross-planes beneath the trough and the wave slopes prevents us from seeing the full structure of the cross-plane circulations beneath a wave. Thus, we are unable to determine whether these circulations are seen beneath other parts of the wave or if they are a fluctuating field that is confined to the region beneath the crest.

It is possible (in fact, it is likely) that the circulations we are seeing are being driven by the slow-speed streak as it interacts with the wave-strain; however, it is

unclear as to whether this could be properly classified as a Langmuir circulation. As discussed in section (4.8) the dominant slow-speed streak in our flow is temporally persistent and infinite in length due to less-than-optimum streamwise and spanwise resolution in our domain. As such, its importance in the upwelling seen in the cross-plane velocity animation makes the resulting circulation suspect. A numerical investigation into Langmuir circulations with the bottom boundary layer producing the turbulence field would be best done in a domain that has good resolution in all directions and a domain depth large enough so that there is at least a depth of k^{-1} between the top of the constant stress layer and the still water level[8]. This would ensure that any Langmuir circulations would be in the log-law region of the bottom boundary layer and would not be affected by the strong shear near the bottom boundary.

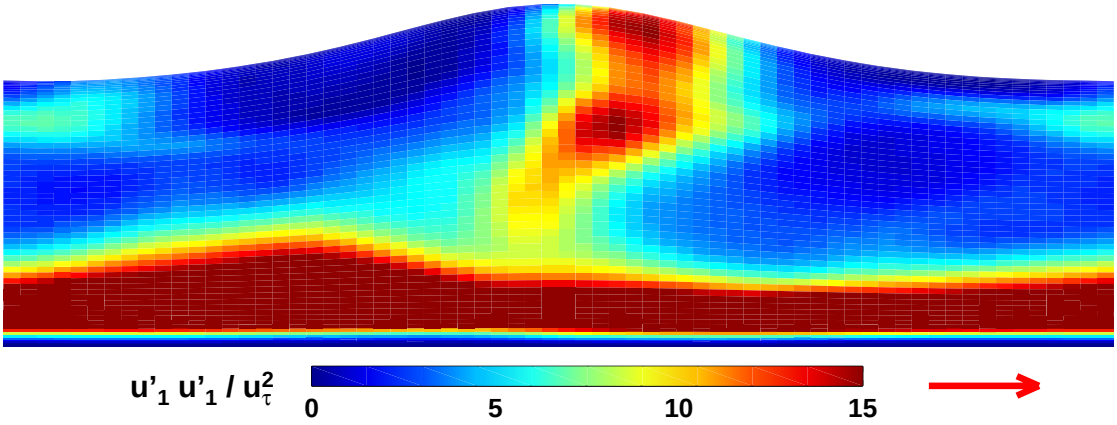
7.3.4 Wave-induced stirring

The results of the numerical simulations show that the wave-induced strains in the flow lift the turbulent structures out of the turbulent shear region of the bottom boundary into the core of the flow. This wave-induced stirring[9] is an important phenomena since it does not require wave-breaking. From the data available in the present work, it is unclear whether the stirring is being driven by (1) the circulation seen in cross-plane velocity field beneath the crest (which may or may not be a Langmuir circulation), or (2) by the interaction of the wave strains with the strong shear of the current at the bottom boundary. Langmuir circulations are usually seen as the primary agent of vertical stirring in a turbulent, wavy flow; however, in the present work there is some evidence that rapid distortion of the shear layer by the wave strains is actually transporting the turbulence up into the near-surface region. This effect seems to be tied to bursting events initiated in the flow expansion under the leading slope of the wave. The argument in support of this hypothesis is that transport caused by Langmuir circulations should not appear as strong bursting events that drag

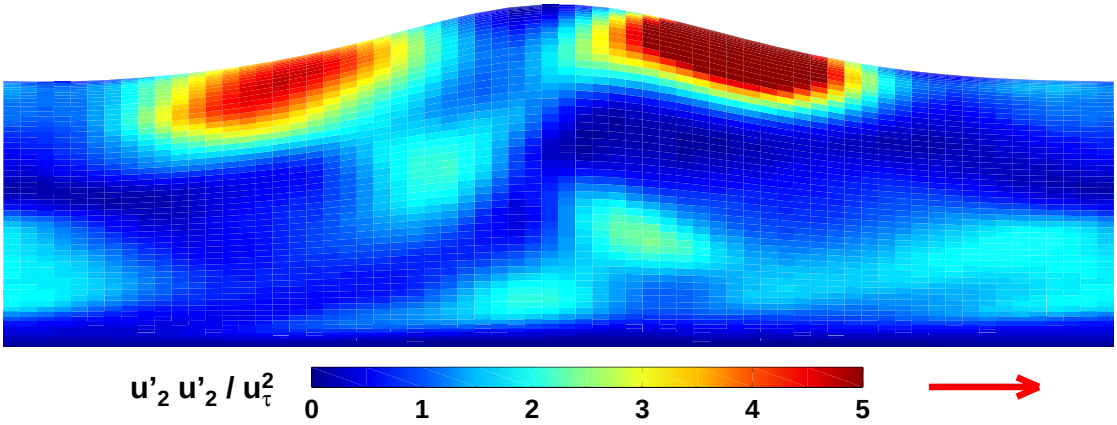
[8] In the present work, the domain depth provides only a distance of $0.7\,k^{-1}$ between the top of the constant-stress layer and the still water level.

[9] As pointed out by J. Koseff (Dept. Civil Eng., Stanford Univ., personal communication, [1997]), our present simulations do not provide sufficient data to analyze the actual *mixing* of the fluid due to the wave-current interactions, but we can provide qualitative analysis on the *stirring* of the fluid through the transports of the turbulence and enstrophy.

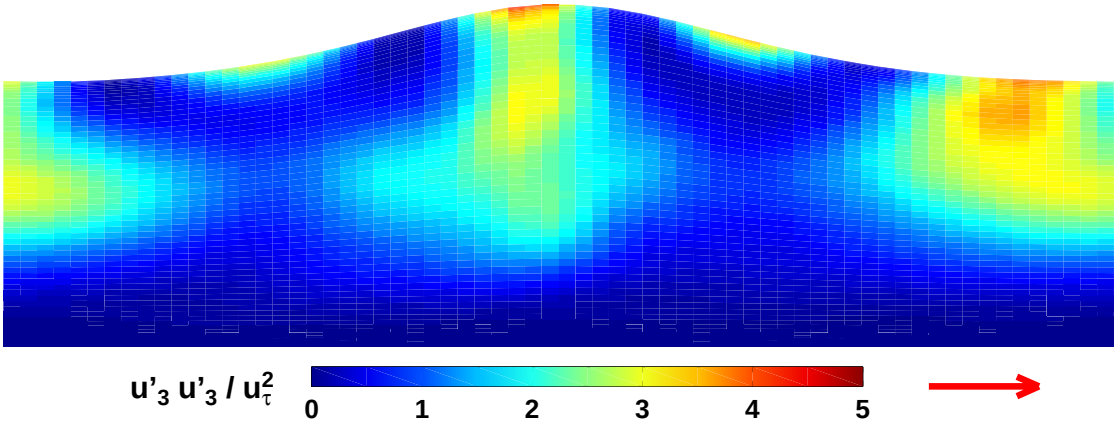
turbulence from the sheared current into the near surface region. Since the Langmuir circulations have both downwelling and upwelling zones, it can be argued that in a phase-averaged view the net effect of the Langmuir circulation on the distribution of the turbulence intensity should be close to zero. For this shallow domain, we expect that a Langmuir circulation would take turbulent fluctuations up into the near surface region through the upwelling zone, then return them in the downwelling zone. The difference between the turbulence intensities in the downwelling and upwelling should be the dissipation minus any turbulence intensification that occurs in the near surface region due to vortex stretching effects. Thus, the phase-averaged appearance of a Langmuir circulation should be a mild transport of the turbulent fluctuations since only the net transport would be seen. The strong, event-driven (bursting) transport of turbulence from the bottom boundary layer seen in the phase-averaged animations is evidence that the transport of the turbulent fluctuations may not be a Langmuir circulation effect.



(a) streamwise turbulence intensity

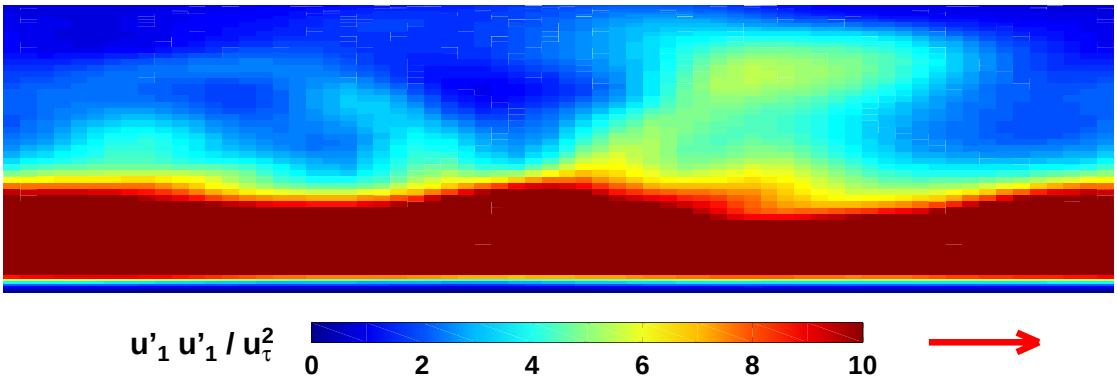


(b) spanwise turbulence intensity

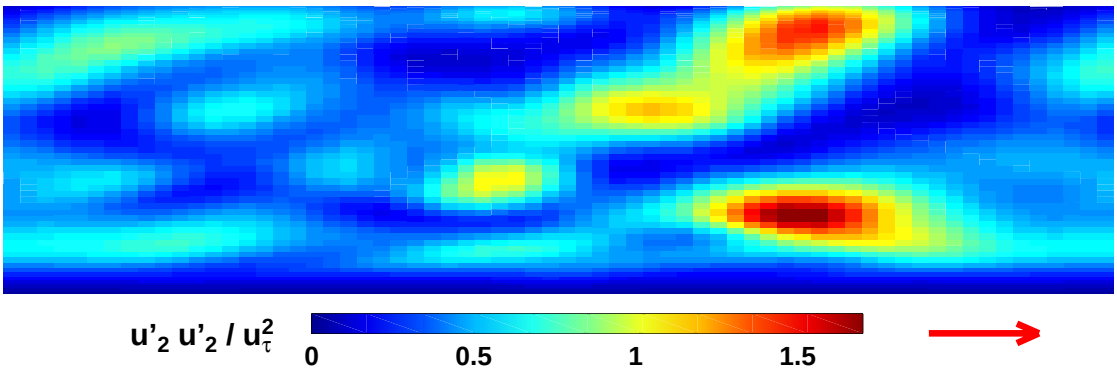


(c) vertical turbulence intensity

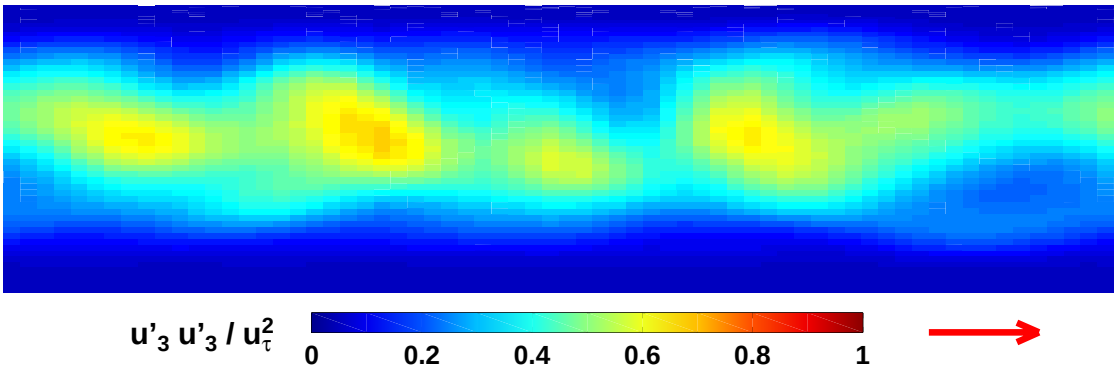
Figure 7.1: Instantaneous turbulence intensities for wave/current flow; case W2, t/T = 13.4



(a) streamwise turbulence intensity

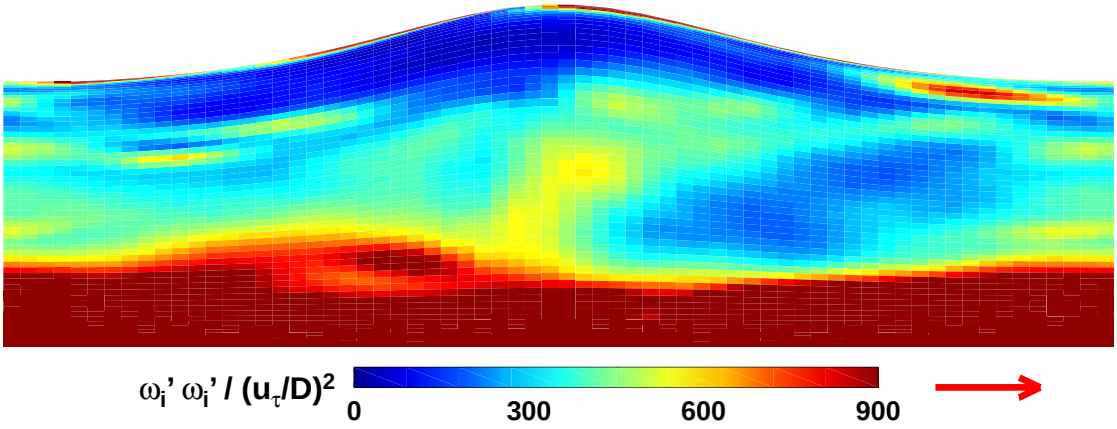


(b) spanwise turbulence intensity

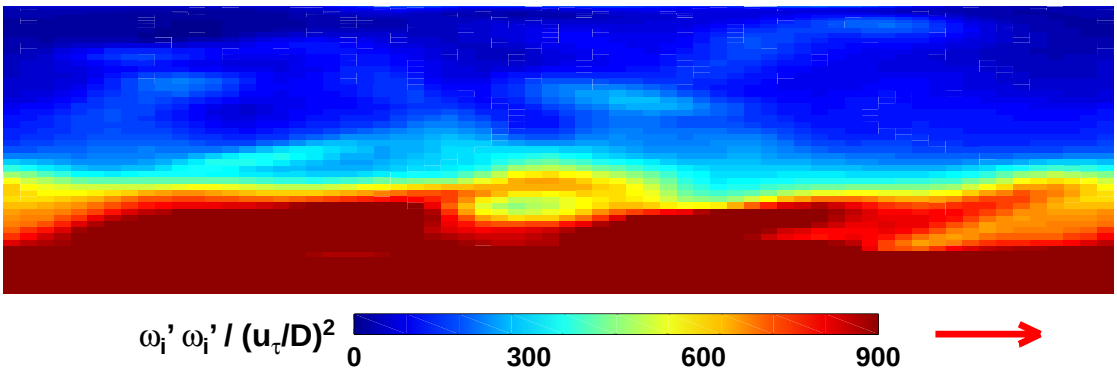


(c) vertical turbulence intensity

Figure 7.2: Instantaneous turbulence intensity for current-only flow; case C, t/T = 13.4

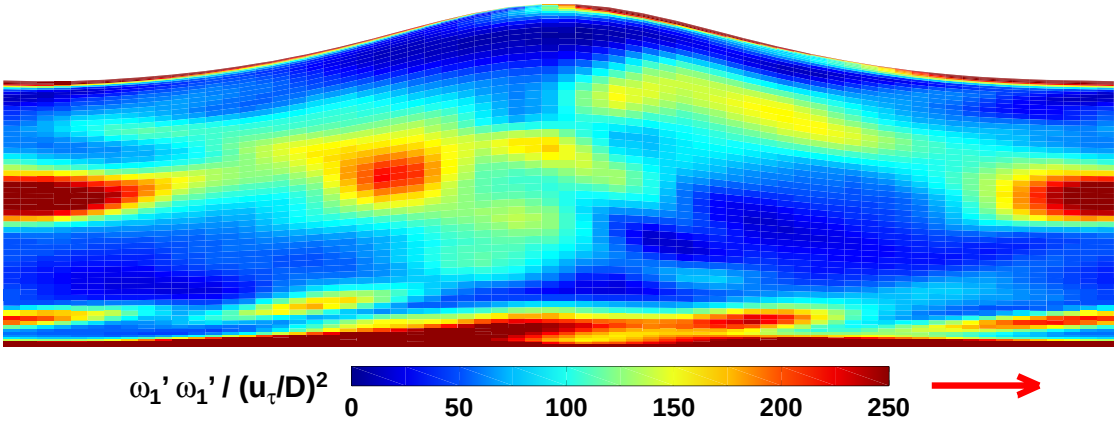


(a) wave/current flow, case W2

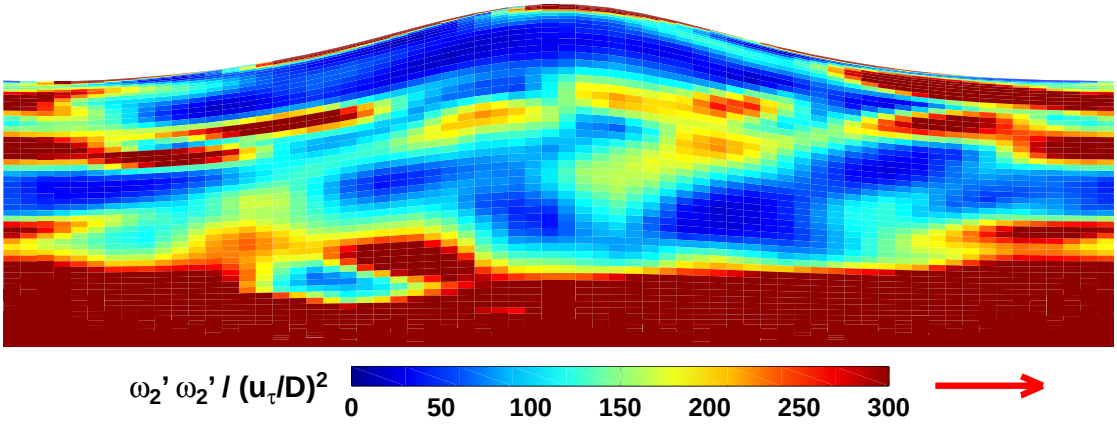


(b) current-only flow, case C

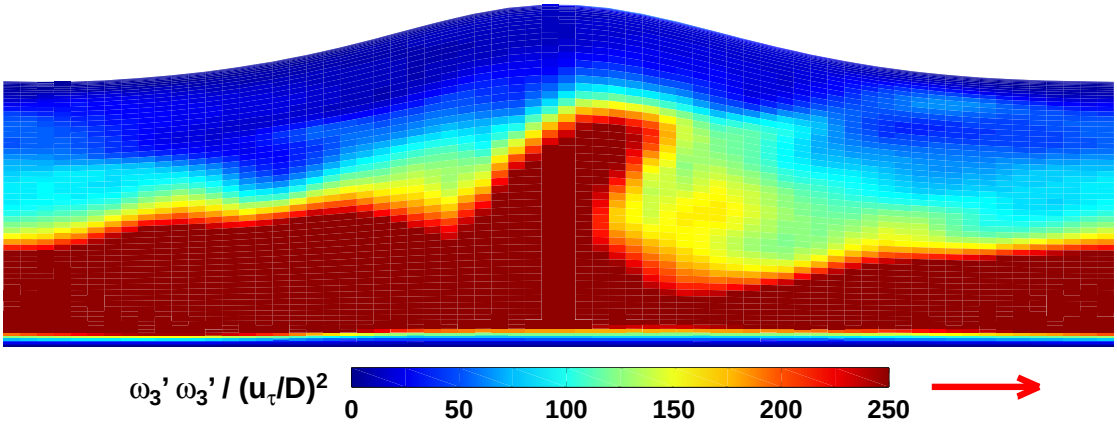
Figure 7.3: Instantaneous enstrophy, t/T = 13.4



(a) streamwise enstrophy component

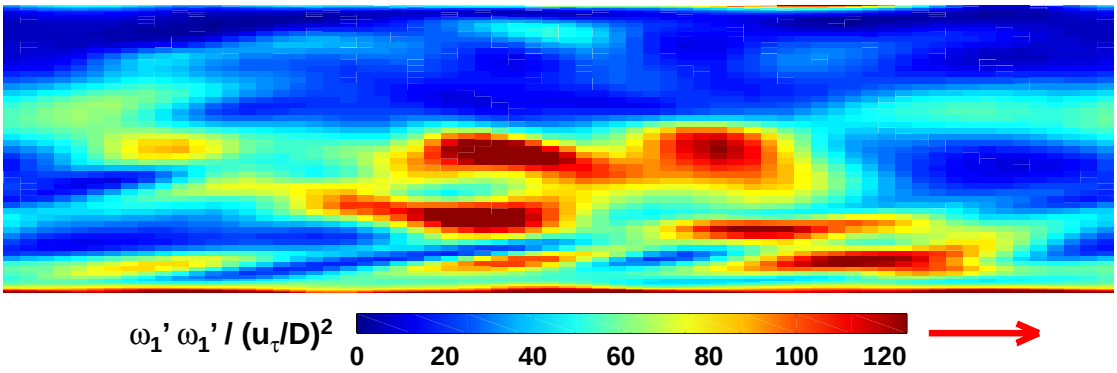


(b) spanwise enstrophy component

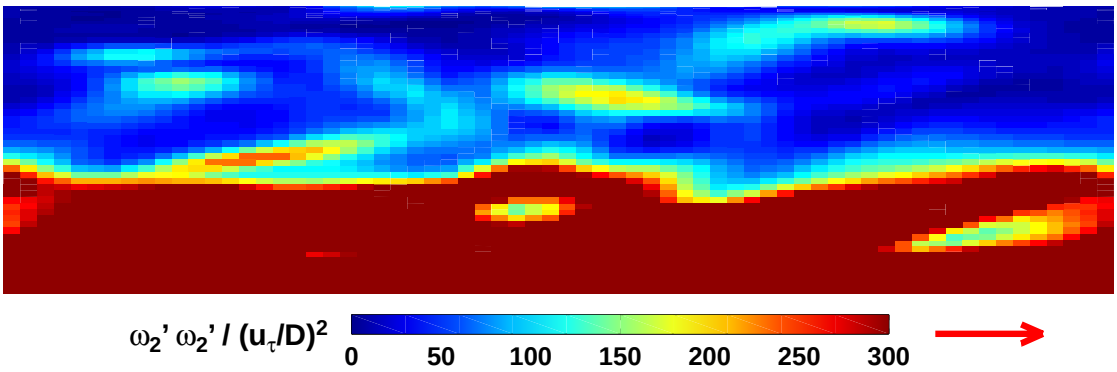


(c) vertical enstrophy component

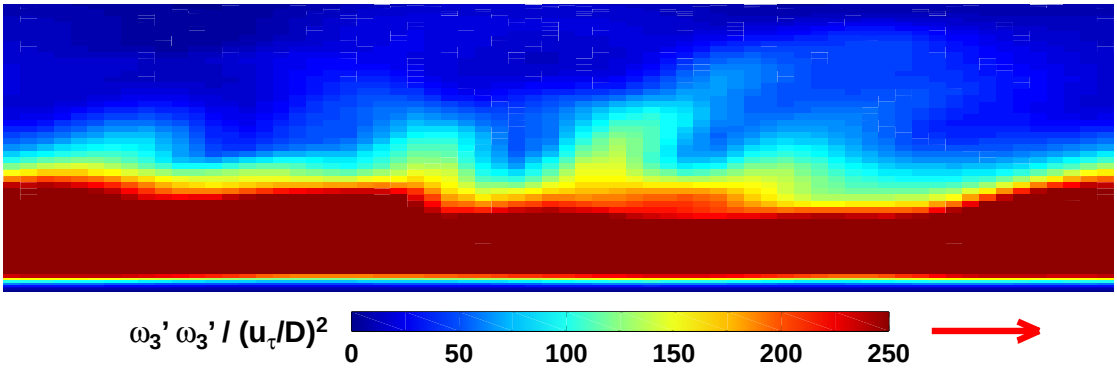
Figure 7.4: Instantaneous enstrophy components for wave/current flow; case W2, t/T = 13.4



(a) streamwise enstrophy component

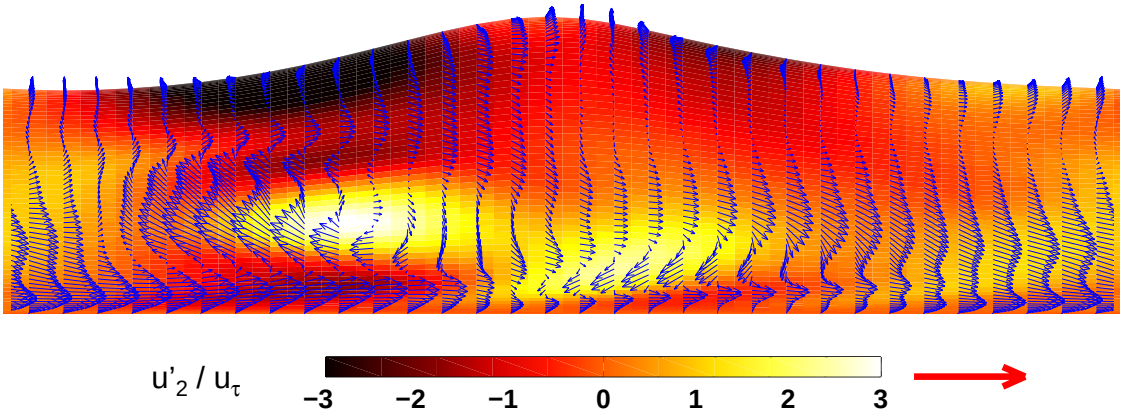


(b) spanwise enstrophy component

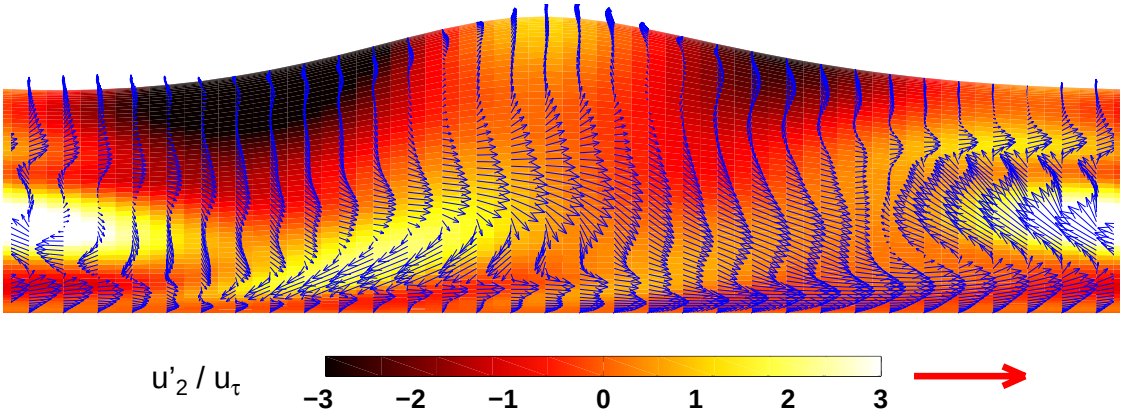


(c) vertical enstrophy component

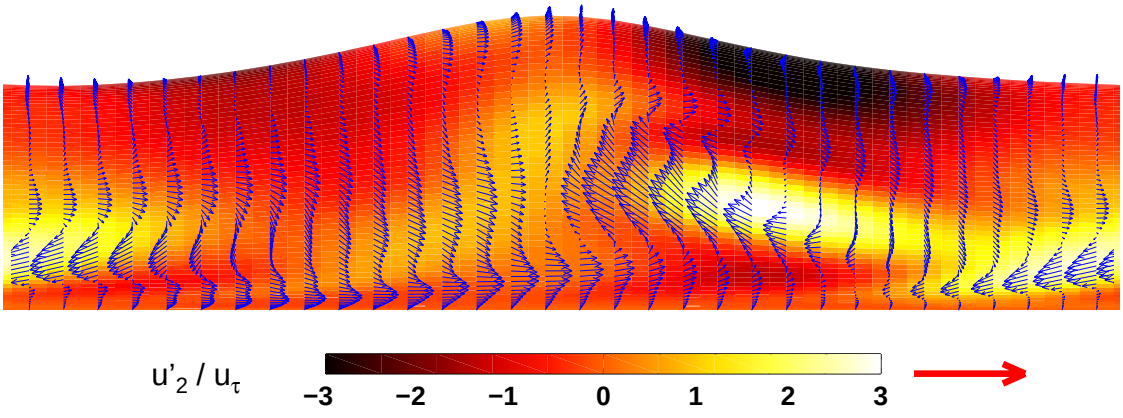
Figure 7.5: Instantaneous enstrophy components for current-only flow; case C, t/T = 13.4



(a) t/T = 9.9

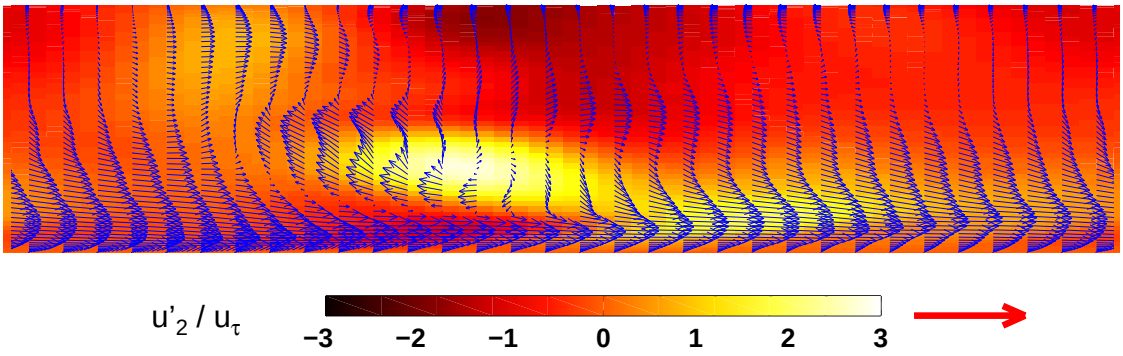


(b) t/T = 10.4

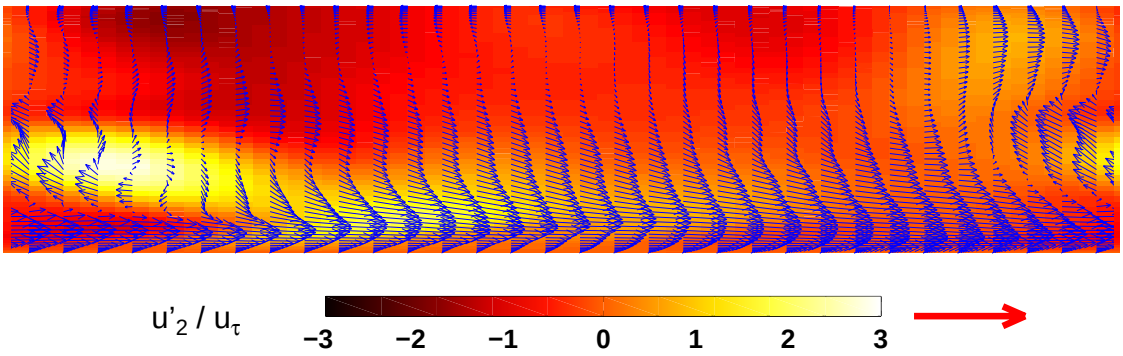


(b) t/T = 11.1

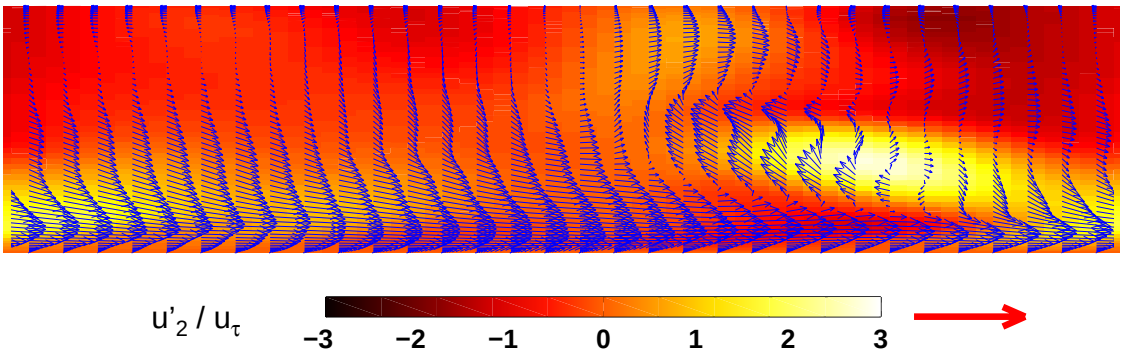
Figure 7.6: Instantaneous plots of velocity variation in vertical plane (with color scale showing spanwise velocity component variation); $x : z$ streamwise mid-plane for wave/current, case W2.



(a) t/T = 9.9

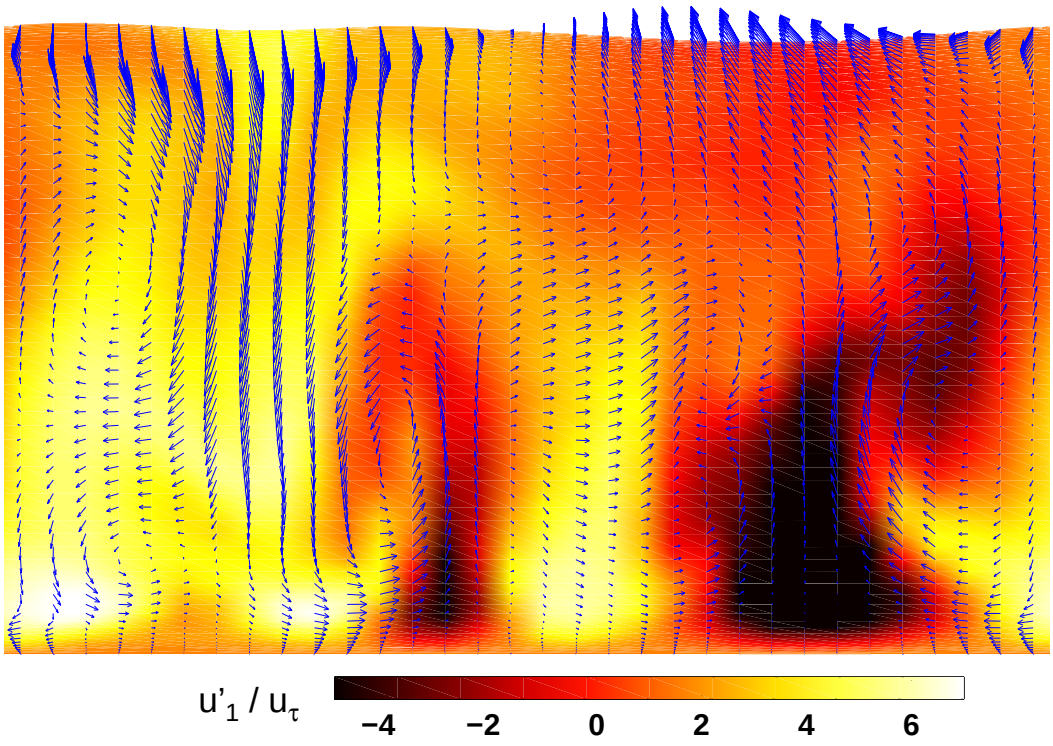


(b) t/T = 10.4

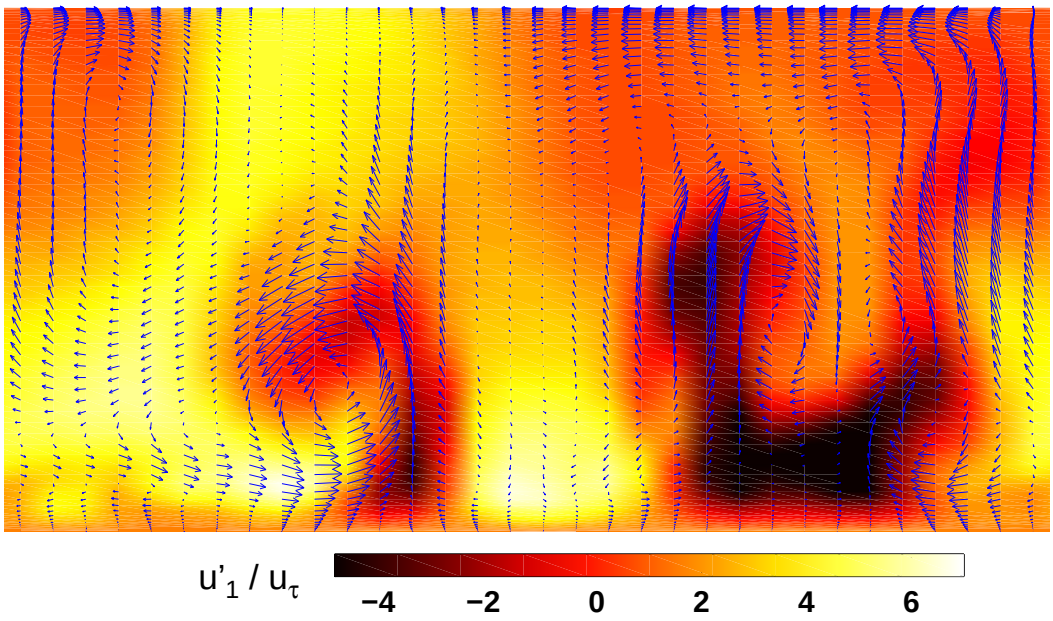


(b) t/T = 11.1

Figure 7.7: Instantaneous plots of velocity variation in vertical plane (with color scale showing spanwise velocity component variation); $x : z$ streamwise mid-plane for current-only, case C.

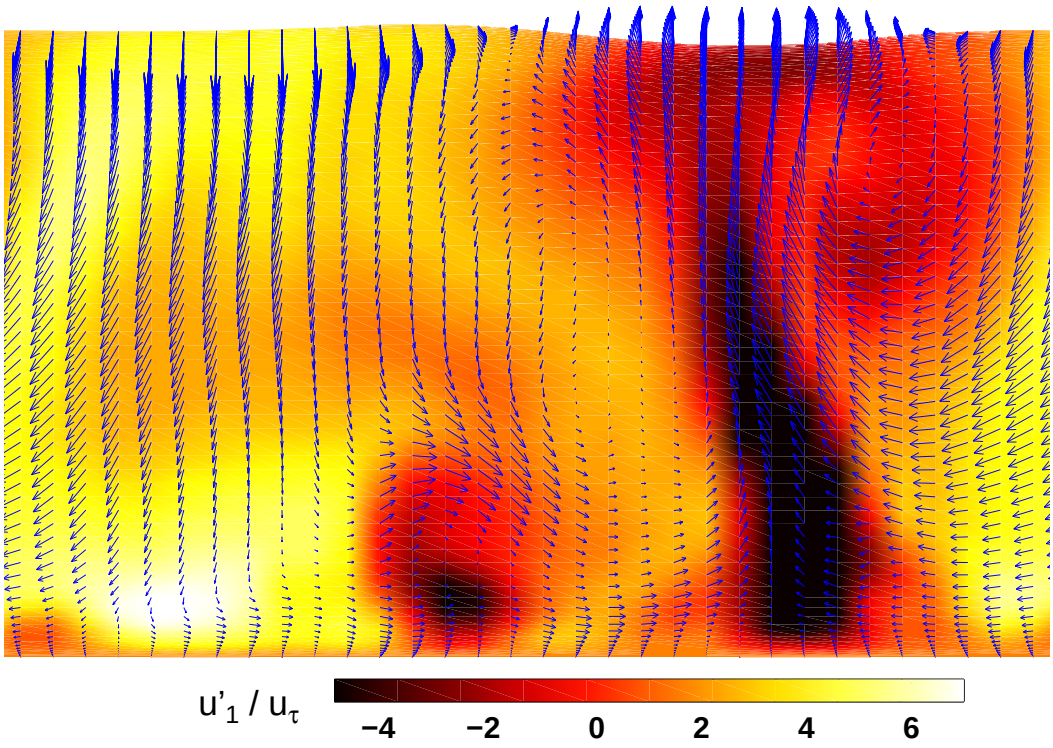


(a) wave/current flow, case W2

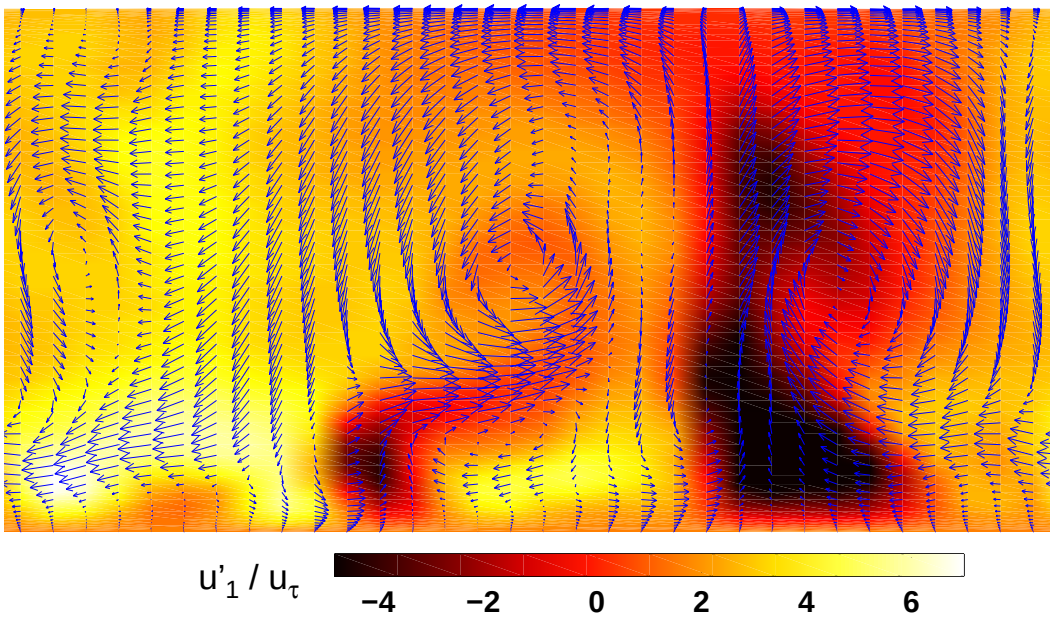


(b) current-only flow, case C

Figure 7.8: Instantaneous plots of velocity variation in vertical plane (with color scale showing streamwise velocity component variation); y:z cross-plane, t/T = 11.1



(a) wave/current flow, case W2



(b) current-only flow, case C

Figure 7.9: Instantaneous plots of velocity variation in vertical plane (with color scale showing streamwise velocity component variation); $y:z$ cross-plane, t/T = 11.6

Chapter 8

Conclusions

This dissertation has a major focus on numerical methods for viscous free-surface flows. The simulation results are less comprehensive, but perhaps are of interest to a wider audience than the details of the numerical work. This final chapter is an attempt to: (1) reduce our observations on the development of the numerical method to a few key points, (2) summarize our primary insights into wave-turbulence interactions, and (3) provide some recommendations for future improvements of the numerical method and avenues for research into wave-turbulence interactions.

8.1 Free-surface numerical methods

The complexities involved in the development and implementation of free-surface numerical methods were explored and discussed in detail in this dissertation. The analysis of stability conditions associated with free-surface simulations will hopefully prevent future researchers from being stalled by seemingly unexplainable catastrophic blow-ups in their simulation runs. The problems associated with grid-generation methods and the limitations of the present state-of-the-art were investigated and documented. The fundamental basis for LES filtering of the kinematic boundary condition was developed, along with a method for implementing the modified Leonard term in the free-surface advance. Possible approaches to defining subgrid-scale models for the Reynolds term and cross term in the filtered kinematic boundary condition were analyzed and discussed.

8.2 Demonstration of free-surface method

The results of the simulations presented in this dissertation demonstrate the effectiveness of the numerical method for accurately simulating free-surface flows with finite-amplitude waves in both laminar and turbulent regimes. Especially important is the good agreement attained with the experimental data of Cowen (1996) for the turbulent flow in the near-surface region of an open-channel flow with a finite-amplitude surface waves. Cowen's DPTV method and the present numerical method provide an effective set of tools for investigating the flow in and near the viscous Stokes' layer of finite-amplitude waves. The demonstrated agreement between the laboratory experiments and numerical simulations for the demanding near-surface flow regime has not been shown in the work of any other research laboratory[1].

8.3 Wave-induced stirring

The results of the simulations show that the wave causes turbulent fluctuations to be stirred up from the bottom boundary layer and into the near surface region. This is an important phenomena since it does not require wave-breaking. From the data available in the present work, it is unclear whether this stirring is being driven by (1) the circulation seen in cross-plane velocity field beneath the crest (which may or may not be a Langmuir circulation that extends through the length of the flow), or (2) by rapid distortion interaction of the wave-strains with the strong shear of the current at the bottom boundary. The effect of the wave-current interactions on the vertical transport and mixing of the fluid remains an area for future investigation.

[1] Dommermuth and Mui (1996) developed a two-dimensional simulation of gravity-capillary waves that showed qualitative agreement with the experiments of Duncan *et al.* (1994). However, as noted by Duncan in the discussion of Dommermuth and Mui, there was a 15:1 ratio between the wavelengths of the laboratory experiments (0.77 m) and the numerical experiments (0.05 m), which makes comparisons difficult. In contrast, the TWCF wave in Cowen's (1996) laboratory experiments was only twice the length of the $W3$ wave in our numerical simulation. This makes our quantitative comparisons possible.

8.4 Persistence of near-surface turbulence

Animations of the numerical simulations show that some of the turbulent, vortical structures which are extracted from the sheared current become trapped in the near-surface region. These structures can connect with the surface through the Stokes' layer and are intensified by vortex-stretching effects as successive wave troughs and crests are encountered. The effects of the stretching on the structures can be visualized by examining the fluctuating enstrophy components. Once a structure is trapped in the near surface region, the intensification by vortex stretching allows the structures to persist for long periods relative to more transient, wave-induced effects. Due to their persistence, these structures appear in wave-averaged data as large increases in the wave-induced turbulence intensities near the surface. The interaction of the turbulent structures with the wave strains in the near free-surface region may be part of the explanation for the persistence of ship wake signatures that remain on the ocean surface long after a ship has passed.

8.5 Recommendations for the numerical scheme

8.5.1 Coding improvements for more efficient operation

The code developed to implement the numerical scheme is effective, but it does not operate at its potential maximum efficiency. The free-surface algorithm was implemented on top of the code developed by Zang (1993), which was not optimized for a moving-grid simulation. Before any further modifications are attempted on the present code, it needs to be re-written in a form that minimizes the memory usage and CPU time required to perform grid-generation computations for each time step. Porting the code to FORTRAN 90 with the use of dynamic memory allocation is likely to provide significant improvement in this area.

An important improvement that could be made to the present implementation of the free-surface method is the development of an efficient approach to implement Mahadevan's (1994) separation of the hydrostatic pressure from the dynamic pressure (see section 4.6). This should significantly speed up the pressure Poisson solution.

Further work and investigation is needed to optimize the interface between the 3DGRAPE/AL grid-generation code and the Navier-Stokes solver. The present approach relies on written files to transfer data due to fundamental incompatibilities in the way in which Zang's code stores data and the approach used by Sorenson and Alter in 3DGRAPE/AL. This is another problem that could be addressed most easily by porting the code to FORTRAN 90 with a consistent set of data structures.

8.5.2 Extension of the code to simulate other flows

The first step towards extending this numerical code to practical applications in naval hydrodynamics is to develop simulations of the temporal evolution of a turbulent flow past an infinite surface-piercing flat plate. This type of flow maintains the present periodic boundary conditions in the streamwise direction while changing the spanwise boundary conditions to Dirichlet on one side and free-slip on the opposing boundary[2]. The bottom boundary could be made either Dirichlet or free-slip. The primary difficulties that will be encountered in implementing this flow are: (1) simulating the edge of the domain where the free surface and the flat plate intersect, (2) adapting the present implementation of the grid-generation method to handle grid stretching in multiple directions so that the boundary layers of the flat plate and the free surface can both be resolved, and (3) supplying the extra computational power required for this more complicated domain. The first problem is related to our relatively poor understanding of the physics and appropriate modeling of a contact line. This is an area of research unto itself. The second problem is related to the inherent difficulty of developing a smooth grid with boundary-orthogonality at the juncture of a three-dimensional wavy surface and a perpendicular flat surface. The third problem is related to the increased number of grid points that will be required to simulate this flow due to the need for coordinate stretching in two directions instead of one. In general, it is difficult to get a smooth stretching that provides good boundary layer resolution with less than 48 grid points (unless the domain is severely truncated). In a boundary-piercing flat plate simulation it is preferable to place the free-slip spanwise boundary as far from the Dirichlet spanwise boundary as is practical. Thus, it

[2] An interesting (but not trivial) task is the development of a spanwise boundary condition that would allow waves to propagate freely out of the spanwise boundary without reflection.

is likely that a surface piercing flat plate will require 64 grid points in the spanwise direction as well as 64 grid points in the vertical direction. Even if the present (less than optimum) streamwise resolution using 32 grid points is maintained, the number of grid points in the flow will have been doubled. Since the grid and pressure Poisson solvers have an execution time that is a function of the number of grid points, it can be argued that this will *at least* double the computational time per time step. In addition, while the memory requirements for the flow variables will also double, the memory requirements for the grid will go up by a factor of 8. For these reasons, it is recommended the simulation method be made more efficient before it is adapted to other flows.

Once the simulation method is coded in a more efficient form, it would be useful to develop a set of effective non-periodic inlet and outlet boundary conditions for the simulation of spatially-evolving flows. This would bring the simulations closer to laboratory experiments and allow greater control over the turbulent velocity field at the flow inlet. With effective inlet and outlet conditions, the turbulence-producing Dirichlet bottom boundary condition could be replaced by a free-slip boundary condition for the simulation of flows on a deep ocean current. Similarly, effective inlet and outlet conditions could allow the simulation of a finite-length surface-piercing flat plate for the study of the boundary-layer developing at the bow or separating at the stern of a ship.

The extension from surface-piercing flat plates to ship hull forms is primarily a problem of grid generation. The computation of a smooth grid with boundary-orthogonality to both the free surface and a curved ship hull for a continuously-varying free surface is not a straightforward matter. Since boundary orthogonality will undoubtedly be lost to some extent, the boundary conditions for the free-surface near the ship hull will require adjustment.

Further investigation into efficient implementation of a subgrid-scale model for the kinematic boundary condition is recommended. The approach proposed in the present work for the modified Leonard term requires computation of a new modified Leonard term for each step of the RK4 scheme used in the free-surface advance. Combining this approach with dynamic subgrid-scale models for the Reynolds term and the cross term could become both unwieldy (from a code writing and debugging point of view) and computationally expensive.

8.6 Recommendations for wave-current studies

The present results provide interesting qualitative and quantitative insights into the wave/turbulence interactions in an open-channel flow. However, it cannot be said that this dissertation has provided an exhaustive analysis of this flow. As demonstrated by the non-dimensionalization of the governing equations in section 2.2, the parameters that govern the an open-channel flow with a surface wave provide for a complicated set of interactions between the terms. The interactions depend on the scaling of the flow. To develop a full understanding of the flow requires an understanding of how each of these parameters affects the flow field. Such an undertaking will not be practical until some of the improvements in the code described in section 8.5 have been made.

In this work we have concentrated on examining the effects of the wave on the fluctuating enstrophy and the turbulence intensities in the core of the flow and the near-free surface region. This provides insight into the stirring and transport of turbulent kinetic energy from the sheared current into the near-surface region, and provides evidence of the intensification of vortical structures by vortex stretching. Further work remains to be done in examination of (1) wave-induced effects on the dissipation rate, (2) wave decay and its effect on the surface shear, (3) wave-induced effects in the bottom boundary layer, (4) connection of turbulent vortical structures to the free-surface, and (5) the mechanism that produces negative shear profile for the mean wave-induced streamwise velocity. The limited data collected in the present work is not sufficient to address any of these issues with any degree of qualitative or quantitative confidence. Due to the limitations on data collection and analysis outlined in section 5.3.3, the practical application of the numerical code requires developing a set of subroutines to collect selected data as the simulation is running rather than trying to dump and analyze all the data. It will not be practical to try to address all of issues outlined above in one or two simulation runs. Instead, it is likely that a number of runs obtaining different types of data and using different wave characteristics will be required to obtain a complete understanding of these effects.

It is recommended that future work be done in close conjunction with laboratory experiments. The choice of output data should be driven by (1) developing data which confirms agreement between the simulation and experiments, and (2) obtaining data which complements the experiments and provides insights that are not

obtainable in the laboratory. It is hoped that the video animations developed in this dissertation will enable future researchers to start work with a better understanding of the structure of wave/turbulence interactions. This should enable research to be directed more efficiently at the regions of interest.

Appendix A

Decaying vortex with a moving grid

The use of second-order accurate discretizations does not guarantee second-order accuracy in a numerical simulation (Kim and Moin [1985]). This is especially true with boundary-fitted curvilinear coordinates and moving grids. We have conducted several simulations of a decaying vortex to demonstrate that our Navier-Stokes solution method is second-order accurate in time and space with both fixed and moving grids The decaying vortex is an analytical solution of the two-dimensional Navier-Stokes equations over the domain of $(0 \leq x_1, x_2 \leq \pi)$that can be written as:

$$u_1 = -\cos(x_1)\sin(x_2)\, e^{-2t} \tag{A.1}$$

$$u_2 = \sin(x_1)\cos(x_2)\, e^{-2t} \tag{A.2}$$

$$p = -0.25\Big\{\cos(2x_1) + \cos(2x_2)\Big\} e^{-4t} \tag{A.3}$$

Figure (A.1) provides simulation results showing the reduction of the RMS velocity error with the increase in grid points from 8×8, to 16×16, to 32×32. Three different error lines are shown: the first represents the results for a fixed grid; the second is for a grid that is fixed in shape, but translates through the decaying vortex domain; the third is for a grid that has boundaries which remain fixed, but whose interior grid lines are stretched with each time step. It can be seen that accuracy is approximately second-order in all these cases.

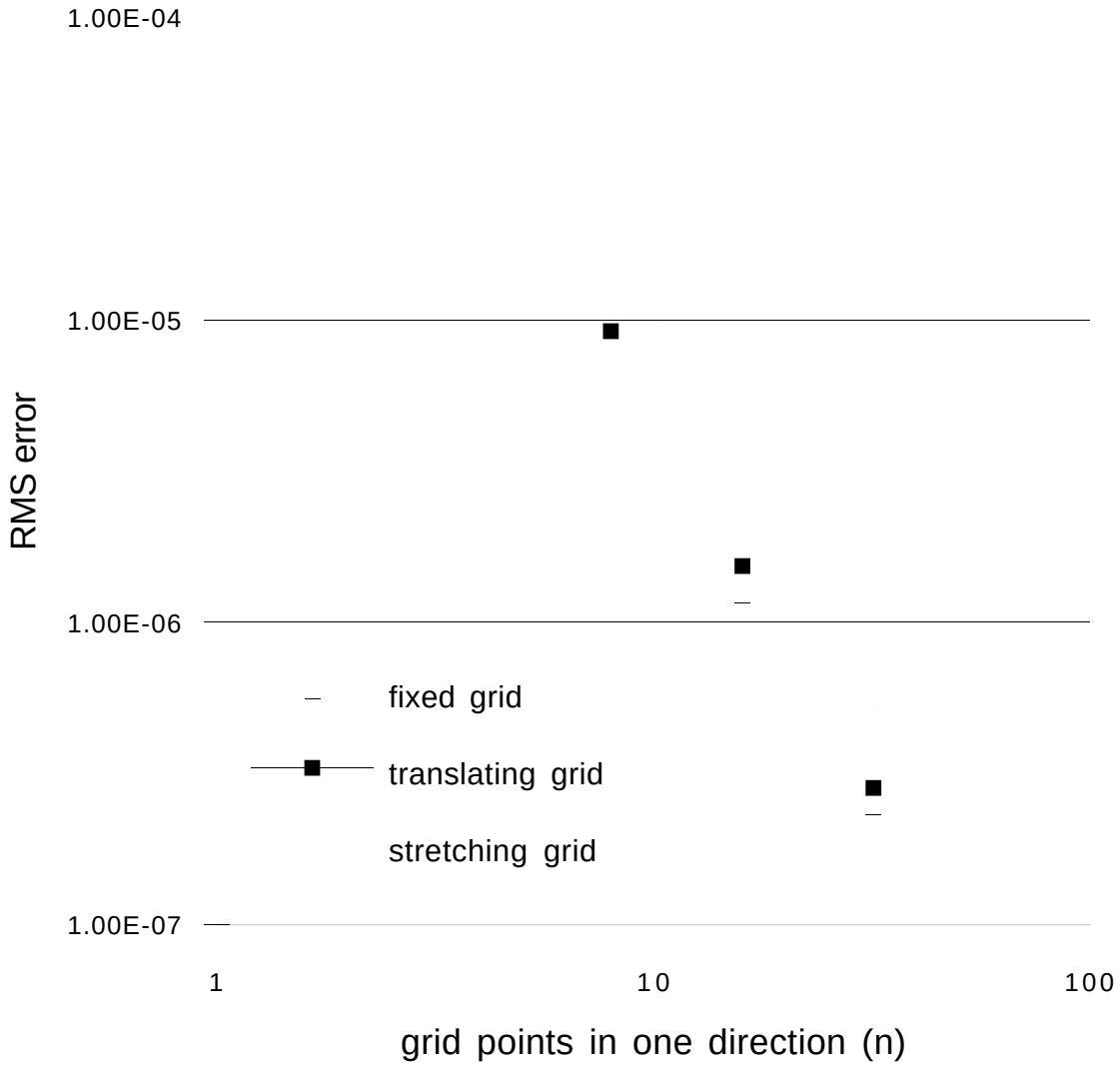


Figure A.1: Decaying vortex accuracy.

Appendix B

Monochromatic standing waves

To test the ability of our code to simulate a free surface, we performed simulations of standing waves in a two-dimensional rectangular basin. Free-slip boundary conditions were used for the sides and bottom of the 32 x 32 cell domain so that the simulations could be compared to nonlinear and linear standing wave theory. The grid points were distributed evenly in both directions. Using the wavelength $\lambda = 1$ to non-dimensionalize the domain, the length (L) is 0.5, and the depth (D) is 0.5. Simulations were run for small-amplitude waves with a non-dimensional wave height (or steepness) of 0.01 and for finite-amplitude waves with a non-dimensional wave height of 0.1. Using the wave amplitude a and the wave number k, these cases have ak values of 0.031 and 0.31, respectively. We conducted simulations at Reynolds numbers of 10, 100, and 1000 to illustrate capabilities of the simulation code with the relatively coarse grid. The Reynolds number is defined as:

$$Re = \frac{L\mathcal{U}}{\nu} \tag{B.1}$$

with ν as the kinematic viscosity, and $\mathcal{U}$ as the characteristic Cartesian fluid velocity based upon the wave amplitude (a) and the wave frequency (σ):

$$\mathcal{U} = a\sigma \tag{B.2}$$

From Lamb (1932), the damping of a free wave due to viscosity as a function of time can be approximated from:

$$a(t) = a(0)e^{-2\nu k^2 t} \tag{B.3}$$

Note that this is based upon an energy dissipation argument for linear waves in deep-water, so we can only expect this to provide a rough guide to our expected damping because the simulation waves are nonlinear with $\lambda/D = 2$, which places the waves in the intermediate water regime.

From Wiegel (1964) we can obtain predictions of wave period and shape for linear and non-linear standing waves. Both linear and non-linear irrotational wave theory predict the wave period (T) for an inviscid wave as:

$$T = \sqrt{\frac{2\pi\lambda}{g}\left(\tanh kD\right)^{-1}} \tag{B.4}$$

For our simulation domain this provides a theoretical period of 1.13 seconds.

According to linear theory for small amplitude waves, the wave shape should be a sinusoid, where the surface height (H) above the still water level is:

$$H\left(x,t\right) = a\sin\left(kx\right)\sin\left(\sigma t\right) \tag{B.5}$$

where σ is the wave frequency ($2\pi/T$). Nonlinear theory for finite amplitude standing waves predicts a wave shape given by:

$$\begin{aligned} H\left(x,t\right) &= a\sin\left(kx\right)\sin\left(\sigma t\right) \\ &- \frac{1}{2}ka^2\coth\left(kD\right)\cos\left(2kx\right)\left\{\sin^2\left(\sigma t\right) - \frac{3\cos\left(2\sigma t\right) + \tanh^2\left(kD\right)}{4\sinh^2\left(kD\right)}\right\} \end{aligned} \tag{B.6}$$

For a small amplitude standing wave, the wave shape predicted by linear and non-linear theory are practically indistinguishable.

We have run simulations using both the linear and non-linear free-surface shape as an initial condition in order to examine the ability of the method to move toward the correct free-surface shape. A summary of the simulation conditions and comparisons between simulation results and theory are presented in table (B.1). The results show excellent agreement for both the nonlinear wave shape and viscous damping of the wave height. In all cases, the wave shape is closer to the theoretical nonlinear shape than to the linear shape. The wave shape difference was computed independently of the effects of viscous damping by using the simulated wave amplitude in equations (B.5) and (B.6) instead of the theoretical amplitude from equation (B.3). Similarly, an adjustment was made for the difference between the theoretical period and the simulation period by applying a small time shift to adjust the theoretical crest to the same time as the simulation crest. The simulations show an increase in the wave period with increasing viscosity, which is a realistic physical result.

To provide a better picture of the simulation accuracy, we present two types of graphs which compare the results of typical simulations to linear and nonlinear theory.

case	ak	Re	initial wave shape	comparison of simulation and theory: period difference[1]	wave height difference[2]	linear wave shape difference[3]	nonlinear wave shape difference[3]
1a	0.031	10	linear	0.32 %	1.1 %	0.39 %	0.32 %
1b	0.031	10	non-linear	0.32 %	1.1 %	0.37 %	0.31 %
2a	0.031	100	linear	0.23 %	0.76 %	1.5 %	0.54 %
2b	0.031	100	non-linear	0.23 %	0.64 %	1.1 %	0.22 %
3a	0.31	10	linear	11 %	0.59 %	0.33 %	0.20 %
3b	0.31	10	non-linear	11 %	0.64 %	0.33 %	0.12 %
4a	0.31	100	linear	0.87 %	1.9 %	2.4 %	0.59 %
4b	0.31	100	non-linear	0.87 %	1.1 %	2.2 %	0.37 %
5a	0.31	1000	linear	0.32 %	6.4 %	1.2 %	0.53 %
5b	0.31	1000	non-linear	0.87 %	3.7 %	1.2 %	0.32 %

NOTES:
1. "period difference" is the mean difference between the simulation wave period and theoretical period for the four oscillations simulated, and is expressed as a percentage of the theoretical period.
2. "wave height difference" is RMS difference between the simulation and theoretical wave height for crests at $x = 0$ from the first through fourth periods, and is expressed as percent of wave height.
3. "wave shape difference" is RMS difference between the simulation wave shape and theory for one wave, expressed as a percentage of the wave height, and measured at the second wave period for case 3 and the fourth wave period for all other cases.

Table B.1: Two-dimensional standing wave simulation summary.

Figures (B.1), (B.3), (B.5), (B.7), and (B.9) show the height of the free surface at the wall $x = 0$ as a function of non-dimensional time (which is obtained using the theoretical period computed from equation B.4). The lines for linear and nonlinear theory are based on equations (B.5) and (B.6) with equation (B.3) used to compute the theoretical amplitude as a function of time. Figures (B.2), (B.4), (B.6), (B.8), (B.10), and (B.11) compare the wave shape for the simulation wave and theory.

Figures (B.1) and (B.2) present results for case 1b, showing that the simulation of a small amplitude standing wave at a low Reynolds number maintains the correct wave profile and is damped as predicted by theory. The results for case 1a (using a linear initial wave shape), are indistinguishable from the results from case 1b (using the non-linear initial wave shape). Figures (B.3) and (B.4) present results for the small amplitude standing wave at a Reynolds number of 100, case 2b. These graphs demonstrate that the simulation of viscous, small-amplitude wave still follows the linear and nonlinear theory both in wave shape and in period.

Figures (B.5) and (B.6) present results for the finite-amplitude standing wave at $Re = 10$, case 3b. In this case, the wave is rapidly damped out. It can be seen

that the period for the simulation is significantly greater than that for the theory. This result is not unreasonable, as one would expect that a highly viscous flow will oscillate at a slower period than that predicted by inviscid theory.

Figures (B.7) and (B.8) present results for the finite-amplitude standing wave at $Re = 100$ with an initial nonlinear wave shape, case 4b. It can be seen that linear and non-linear theory are not coincident and the simulation wave shape follows nonlinear theory very closely.

Figures (B.9) and (B.10) present results for case 5b ($Re = 1000$) for an initially nonlinear wave shape. It can be seen that this case is at the limit of our ability to resolve the viscous effects with the coarse grid used in the simulation.

To show that the simulation can evolve a nonlinear wave from an initially linear profile, figure (B.11) presents the wave shape for case 4a, which is similar to case 4b, (shown in figure [B.8]), except that the initial wave shape is a sinusoid from linear theory.

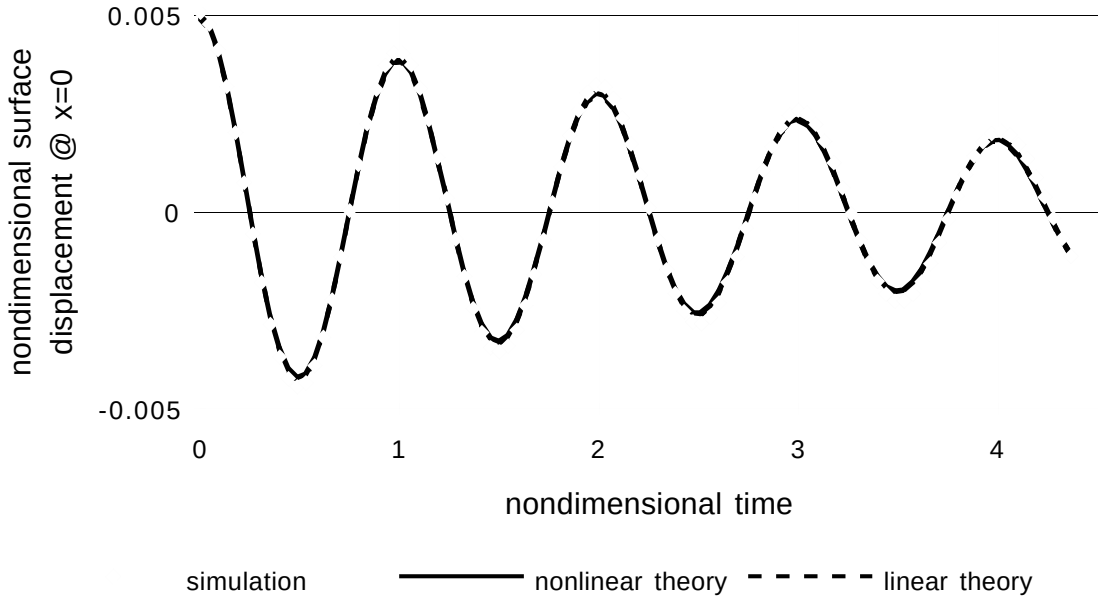


Figure B.1: Free surface wall height, case 1b; $Re = 10$, $ak = 0.031$

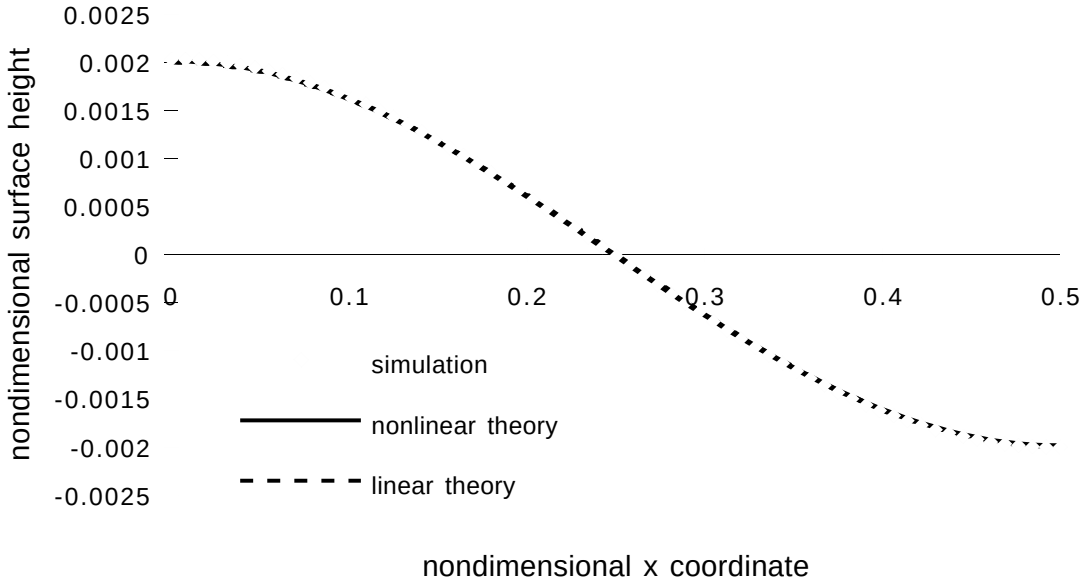


Figure B.2: Free surface wave shape, case 1b; $Re = 10$, $ak = 0.031$

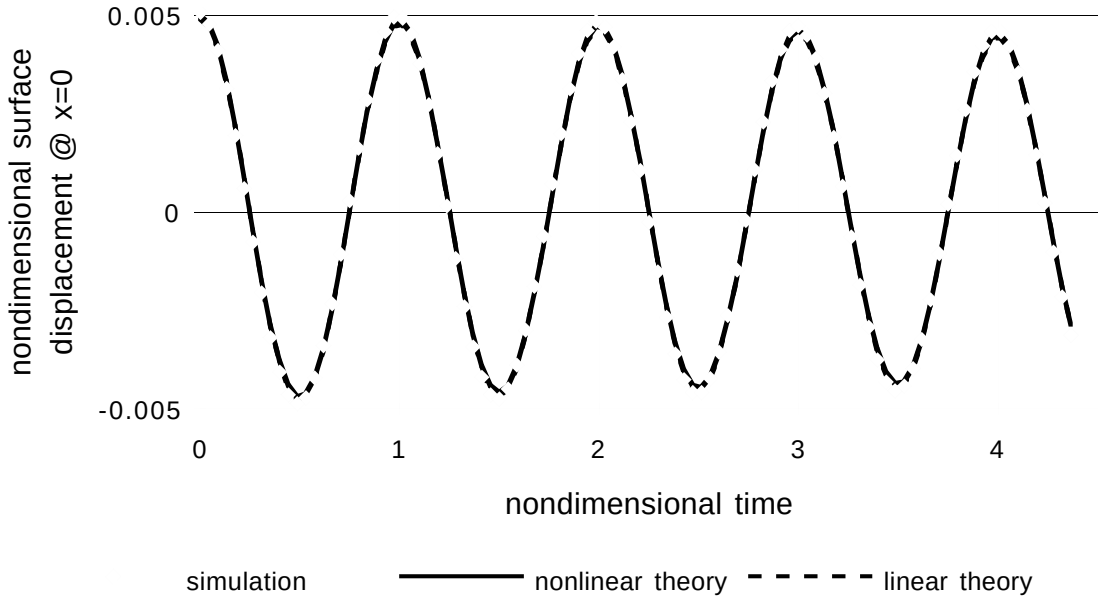


Figure B.3: Free surface wall height, case 2b; $Re = 100$, $ak = 0.031$

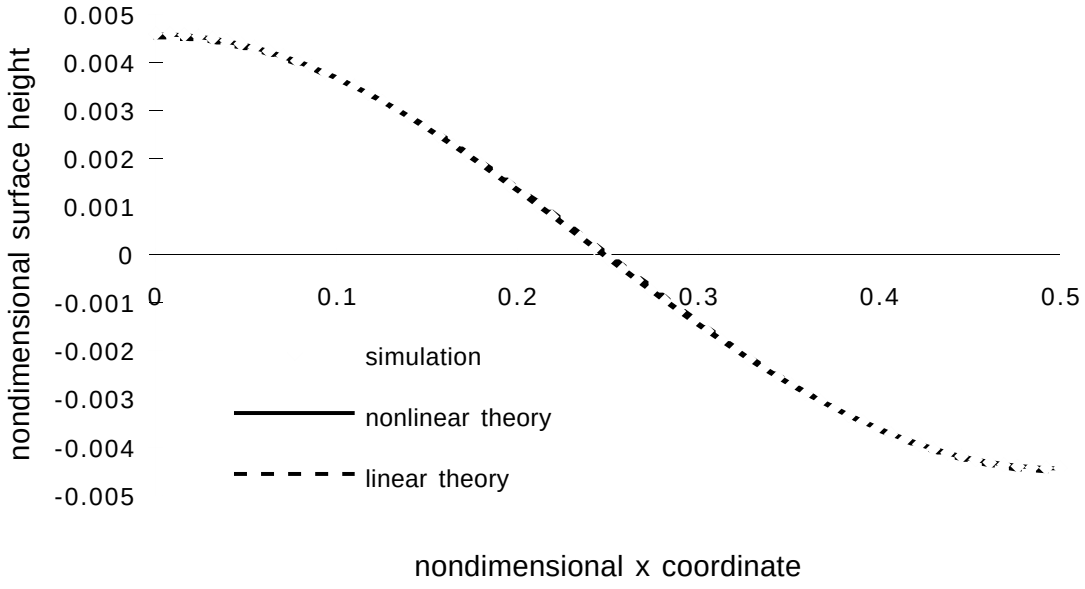


Figure B.4: Free surface wave shape, case 2b; $Re = 100$, $ak = 0.031$

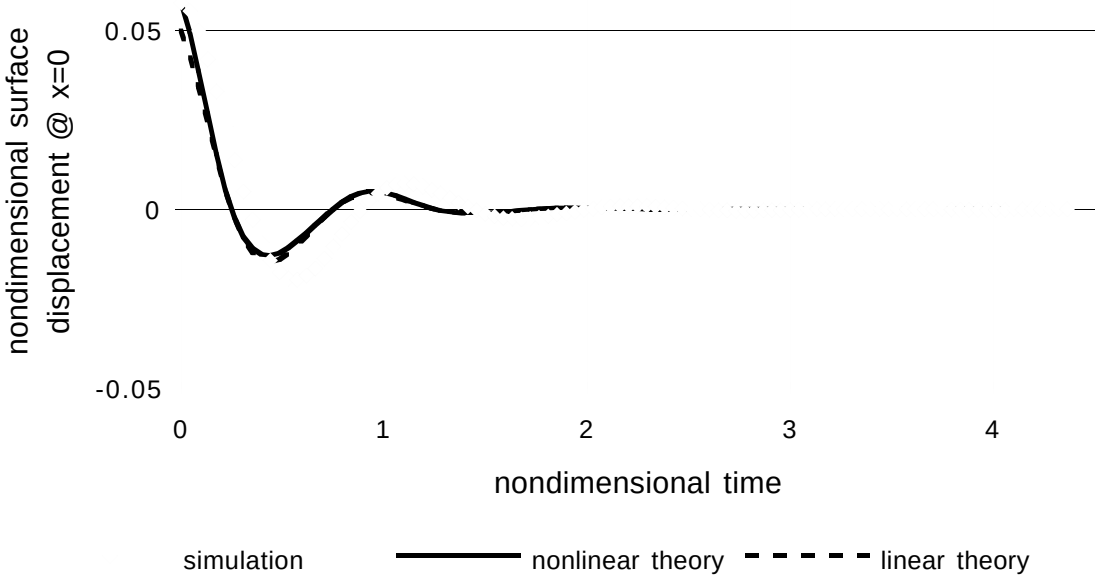


Figure B.5: Free surface wall height, case 3b; $Re = 10$, $ak = 0.31$

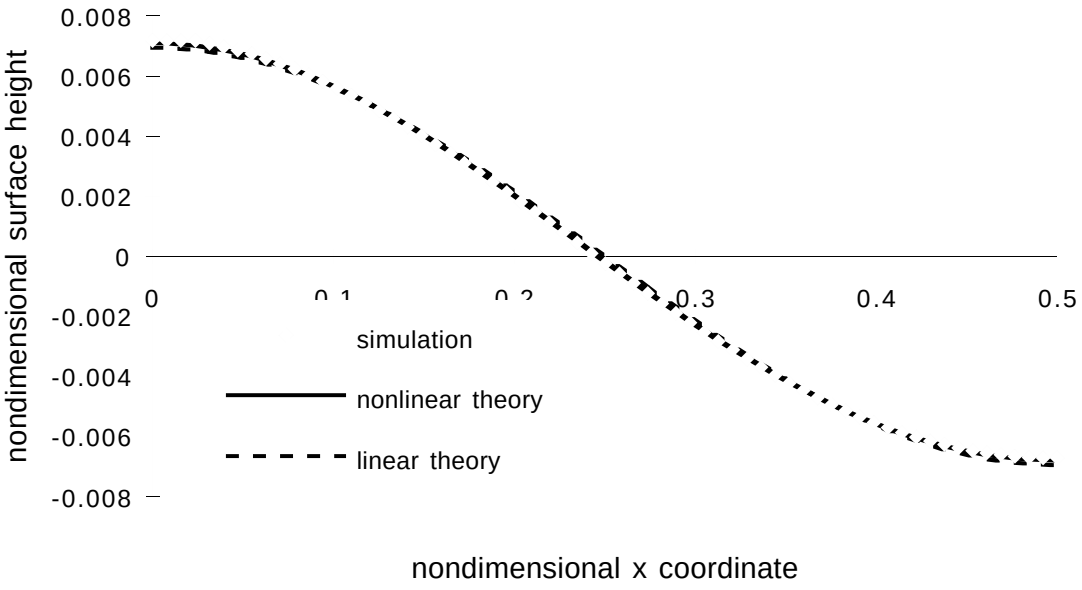


Figure B.6: Free surface wave shape, case 3b; $Re = 10$, $ak = 0.31$

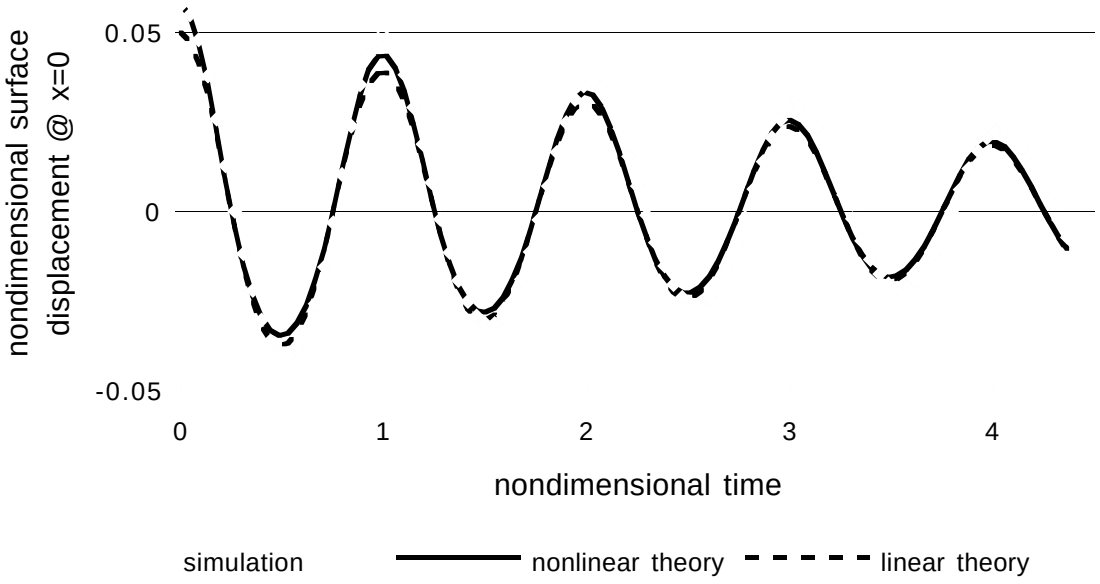


Figure B.7: Free surface wall height, case 4b; $Re = 100$, $ak = 0.31$

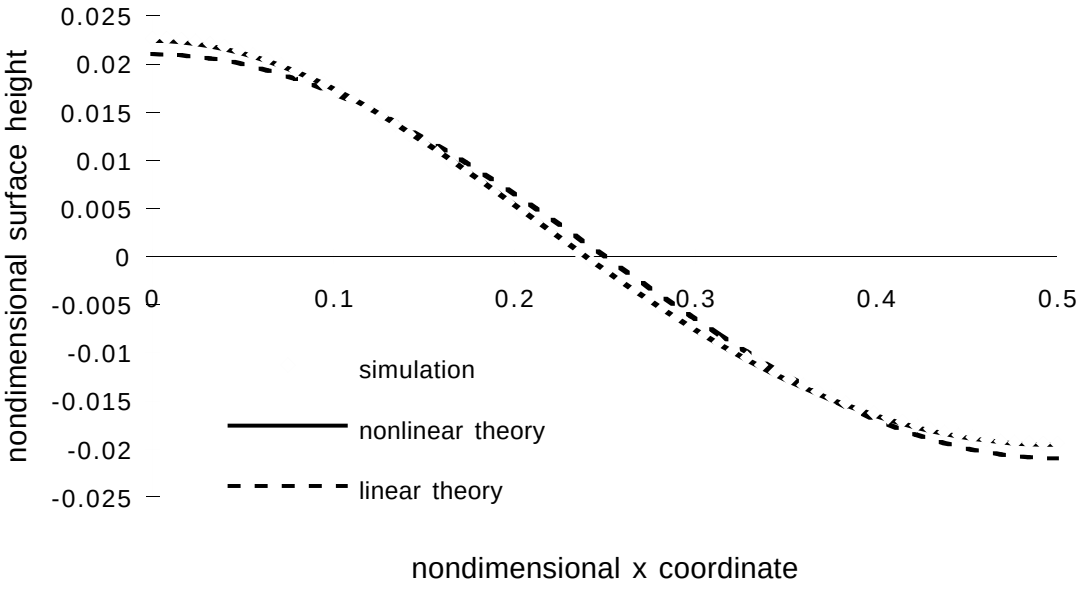


Figure B.8: Free surface wave shape, case 4b; $Re = 100$, $ak = 0.31$

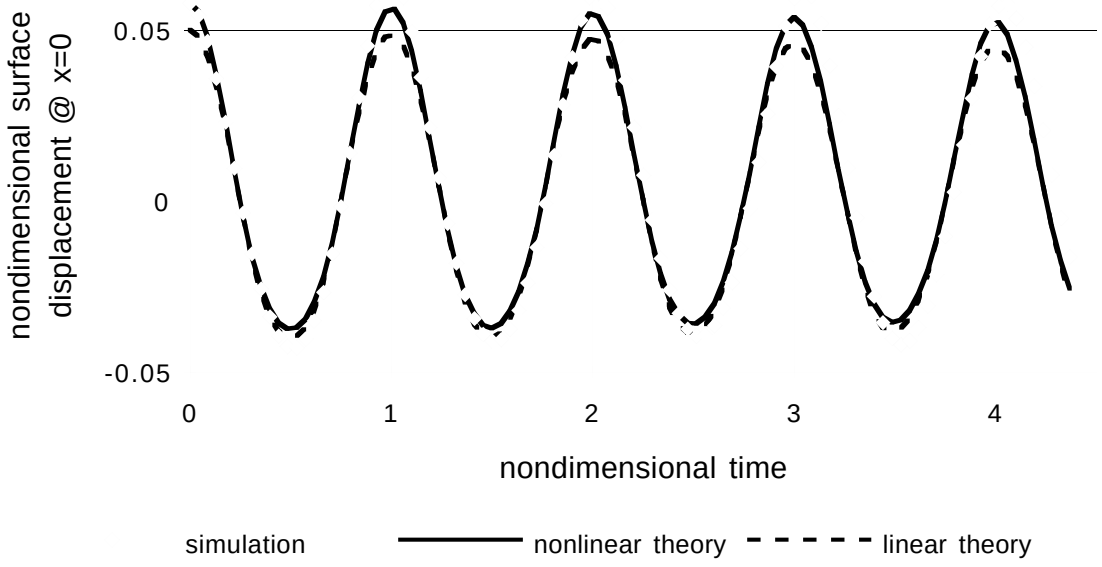


Figure B.9: Free surface wall height, case 5b; $Re = 1000$, $ak = 0.31$

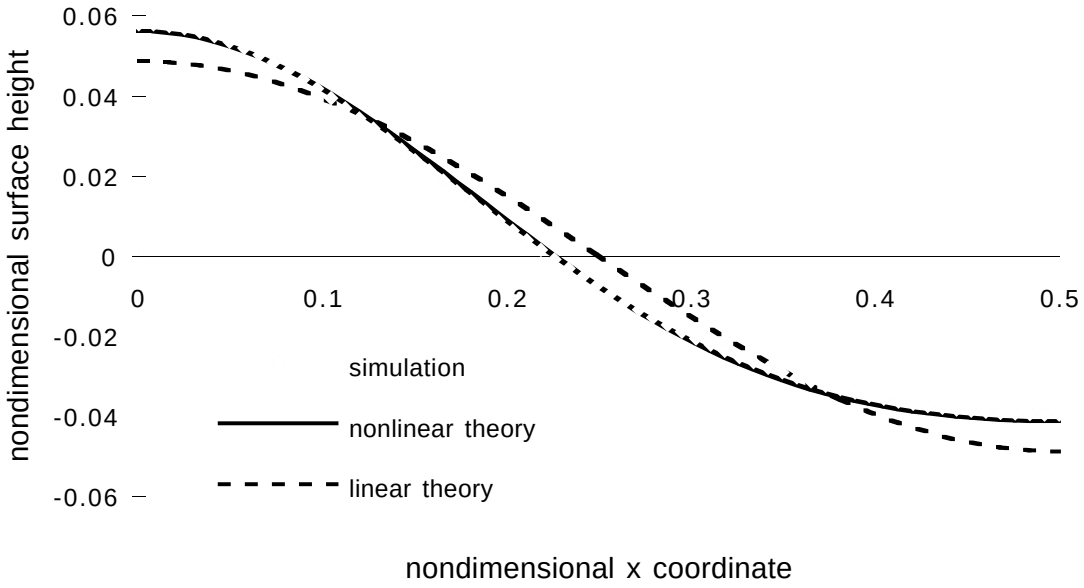


Figure B.10: Free surface wave shape, case 5b; $Re = 1000$, $ak = 0.31$

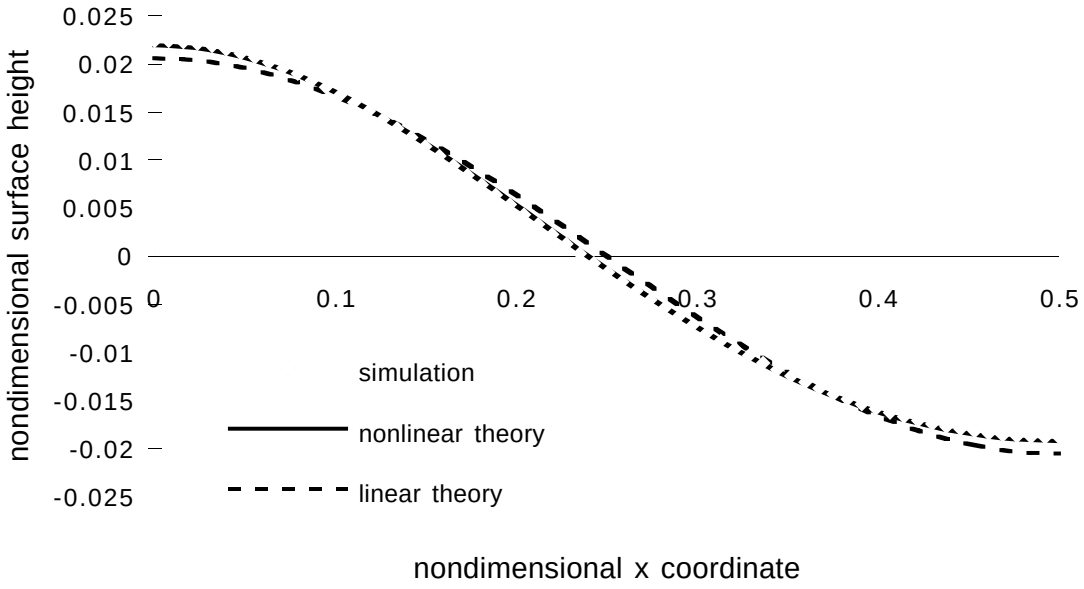


Figure B.11: Free surface wave shape, case 4a; $Re = 100$, $ak = 0.31$

Appendix C

Three-dimensional standing waves

C.1 Introduction

Simulations of standing waves in a rectangular basin were conducted to demonstrate the ability of the numerical method to simulate finite-amplitude surface deformations in three dimensions. The initial free surface position is computed from two identical standing waves that are linearly superposed at a 90 degree angle. Each individual wave is monochromatic, nonlinear, and of finite amplitude (using a second-order approximation from Wiegel, 1964). This provides the initial free surface shape shown in Figure (C.1).

The simulation grid contains $32 \times 32 \times 32$ grid cells (for clarity, only one-quarter of the grid cells on the surface are shown). The horizontal coordinates are non-dimensionalized by the lowest mode wavelength, while the surface deformation (η or eta , measured from still free-surface level) is non-dimensionalized by the combined amplitudes of the two superposed waves.

The simulation domain is one-half of a wavelength in length and breadth and one-tenth of a wavelength in depth ($0.5\,m \times 0.5\,m \times 0.1\,m$). The depth of the simulation was chosen in conjunction with the viscosity of the fluid ($7.32 \times 10^{-5}\,m^2/s$) so that the free-surface boundary layer could be resolved using five grid cells without requiring excessive computational time or memory. While the viscosity is still significantly larger than that of water, it is approximately one-fifth of that used in the linearized free-surface simulations of Borue *et al.* (1995). A proportionally scaled diagram of the domain is shown in figure (C.2)

The initial velocity field is the irrotational solution at the maximum displacement of a standing wave (*i.e.* zero for all velocity components). At the start of the simulation, the upper surface is set free and is allowed to evolve in response to the nonlinear kinematic and dynamic boundary conditions as applied to the Navier-Stokes

equations. Each of the superposed waves has the following initial characteristics: the wave number (k) is 6.28 $1/m$, the wave Reynolds number ($\sigma/\nu\, k^2$) is approximately 2000, the wave ak is 0.0314, the wavelength is 100 cm, and the individual component wave amplitude is $0.5\,cm$. The frequency (σ) of the component waves is 5.86 rad/sec, as computed from the dispersion relation (Wiegel [1964]):

$$\sigma^2 \;=\; gk\, tanh\ (kD) \tag{C.1}$$

where g is gravity, D is the depth, and k is the wave number.

The combination of the two waves has a sloshing amplitude of $0.01\,m$ along a diagonal axis of $0.707\,m$. Since the initial conditions are a linear superposition of the two waves, we expect that the primary oscillation of the system should be at the same frequency as the individual waves until the nonlinear interactions have had enough time to act upon the system. However, we should see some nonlinear effects occurring at wavelengths equal to the diagonal axis and twice the diagonal axis. These dimensions have wave k values of 8.89 $1/m$ and 4.44 $1/m$, respectively. The system ak based on the diagonal sloshing is 0.04. From the dispersion relation, we might expect nonlinear effects occurring at frequencies of 7.87 and 4.26 rad/sec, corresponding to the two primary diagonal modes. The sidewall and bottom boundary conditions are free-slip (no boundary layer). This is appropriate for this simulation since the presence of sidewall boundary layers would obscure the free-surface viscous effects which we seek to capture.

C.2 Approach

The simulation was carried out for seventeen wave periods (6700 time steps). For analysis, we ran an additional simulation of viscous 2D sloshing of one of the monochromatic initial component waves. Both the 3D and the 2D simulations oscillated with a primary period equal to the theoretical period (for a 2D finite-amplitude standing wave) with an uncertainty of 0.25%. The uncertainty was not surprising since the wave period was not evenly divisible by the simulation time step. There was no perceptible increase or decrease of the period over the course of the simulation, so we can conclude that the duration of the simulation was insufficient for nonlinear interactions to have any significant effect on the overall period of the sloshing. The waves are in

linear superposition to the first order; therefore it should require a relatively large number of wave periods for the higher-order nonlinear terms to have a large-scale effect on the dispersion relation.

The effect of wave-wave interactions on the surface deformation (Δ) in the 3D simulation is defined by subtracting the wave surface (η_{2D}) formed from a 90 degree superposition of two of the 2D monochromatic simulations from the wave surface of the 3D (η_{3D}) simulation:

$$\Delta^{[total]} \equiv \eta_{3D} - \eta_{2D} \tag{C.2}$$

Since the simulation of the 2D component wave is also nonlinear, this approach removes the nonlinear interactions of each wave with itself, leaving only the interactions between the two waves. Note that this analysis technique is designed to elucidate the differences between nonlinear 2D and nonlinear 3D waves, and should not be construed as a physical superposition of nonlinear effects. Figure (C.3) shows a plot of the difference between the 3D wave and the superposition of the 2D waves after 11.5 periods have been simulated. This is the total nonlinear effect of wave-wave interactions on the surface deformation (which we will call the "surface difference"). In Figure (C.3), the overall surface difference for the nonlinear interactions is less than 1% of the wave amplitude in either direction.

To provide a quantitative understanding and analysis of the nonlinear wave-wave interactions, we can decompose the total nonlinear effects (shown in figure C.3) using symmetries and asymmetries around four cutting planes illustrated in figure (C.4). These planes can be identified by the following axes:

- $d1$ the main diagonal axis along which the overall sloshing is approximately symmetrical
- $d2$ the secondary diagonal axis along which the overall sloshing is asymmetrical
- x an axis where x/wavelength = 0.25
- y an axis where y/wavelength = 0.25

The asymmetrical component of the surface deformation ($\Delta^{[a]}$) at a point (p) about a cutting plane (c_1) is defined as:

$$\Delta_p^{[a:c_1]} \equiv \frac{1}{2}\left(\eta_p - \eta_i^{[c_1]}\right) \tag{C.3}$$

where η_p is the surface deformation at point p, and $\eta_i^{[c_1]}$ is the surface deformation of the image point of p relative to the cutting plane c_1. The symmetrical component of the surface deformation ($\Delta^{[s]}$) is defined as:

$$\Delta_p^{[s:c_1]} \equiv \frac{1}{2}\left(\eta_p + \eta_i^{[c_1]}\right) \tag{C.4}$$

It follows that $\Delta_p^{[a:c_1]} + \Delta_p^{[s:c_1]} = \eta_p$. If we substitute $\Delta_p^{[s:c_1]}$ for η_p and the image point of $\Delta_p^{[s:c_1]}$ with respect to a second cutting plane (c_2) for $\eta_i^{[c_1]}$, then equations (C.3) and (C.4) become:

$$\left(\Delta_p^{[s:c_1]}\right)_p^{[a:c_2]} = \frac{1}{2}\left\{\Delta_p^{[s:c_1]} - \left(\Delta_p^{[s,c_1]}\right)_i^{[c_2]}\right\} \tag{C.5}$$

$$\left(\Delta_p^{[s:c_1]}\right)_p^{[s:c_2]} = \frac{1}{2}\left\{\Delta_p^{[s:c_1]} + \left(\Delta_p^{[s,c_1]}\right)_i^{[c_2]}\right\} \tag{C.6}$$

Using further recursions of equations (C.5) and (C.6) about each of the cutting planes decomposes the surface into sixteen components. The sum of the sixteen components is equal to the original surface deformation. Fortunately, it turns out that only three of the components are of any significance (with deformations larger than 0.01% of the wave amplitude). The surface difference and the three primary modes are shown in figure (C.5). The modes are:

1. First Damping mode: an oscillation which opposes the current wave position and can be seen as additional damping in the 3D simulation. This is a reasonable and expected effect since viscous effects should increase with wave steepness, and the 3D wave has steeper surface gradients along the diagonal than are seen in the 2D waves.
2. Second Mode: this is the second fundamental mode of oscillation supported by the domain and is a combination of standing waves along the diagonal.
3. First Cross Mode: an oscillation across the domain along the diagonal that is perpendicular to the initial direction of sloshing. The mode is one of the two lowest fundamental modes that are supported by the domain, but its existence in this simulation is purely numerical in origin.

C.3 Fundamental modes of oscillation

The sloshing in a square basin can be analyzed for an inviscid, irrotational, infinitesimal wave using the approach in Mei (1983). The equation that governs the surface deformation from still water (η) is:

$$\nabla^2\eta \; + \; k^2\eta \; = \; 0 \tag{C.7}$$

where k is the wave number and ∇^2 is the Laplacian differential operator.

The solution to equation (C.7) for $k^2_{m,n} = \frac{\pi^2}{L^2}(m^2 + n^2)$ is:

$$\eta \; = \; A_{nm} cos\frac{n\pi x}{L} cos\frac{m\pi y}{L} \qquad : \qquad n, m \; = \; 0, 1, 2, 3 \tag{C.8}$$

where n and m indicate linear wave modes in the x and y direction respectively, A_{nm} is the amplitude of the n, m mode, and L is the length of a side of the basin. We can see that the fundamental modes are $n = 1,\ m = 0$ and $n = 0,\ m = 1$, which correspond to 2D waves in the x and y directions. The modes of our numerical solution that we have identified as the first damping mode and the first cross mode are combinations of fundamental mode x and y waves that are oscillating out of phase. The largest second mode supported by the basin is for $n = 1, m = 1$. This corresponds to the second mode deformation shown in figure (C.5).

C.4 Analysis of second mode of oscillation

The solution to the Helmholtz equation found in equation (C.8) has modes of constant amplitude since there are no viscous effects or nonlinear interactions between modes. However a data animation of the second mode deformation from figure (C.5) shows that the maximum amplitude of the second mode is modulated by interactions with the other wave components. This can be seen also be seen by graphing the RMS displacement of the second mode as against time shown in figure (C.6). In this figure, the RMS displacement of the entire surface is shown on a reduced scale (divided by 170) as a reference to the actual wave simulation period. It can be seen that the second mode does not oscillate at the period of the free surface. This is because the free-surface period is that of two rectilinear modes, while the second mode is a true diagonal mode. The diagonal nature of the second mode is more apparent in

the expanded view of figure (C.7). The period of the second mode is based on the diagonal length rather than the rectilinear length. The resulting oscillation frequency of the second mode in the simulation is 7.33 rad/s, which is in reasonable agreement with the theoretical value of 7.87 rad/s for a wavelength measure on the diagonal.

Note in figure (C.6) that whenever the second mode would be precisely out of phase with the free surface, we see severe damping of the second mode. This indicates an energy exchange between the rectilinear and diagonal components of the wave. The existence of this second mode is physical, as attested to both by theory and the presented numerical results. We also ran simulations with varying convergence levels, different free-surface advance methods, and different grid resolutions: all resulted in the production of second modes that were both qualitatively and quantitatively comparable.

The second mode is attributable to the Navier-Stokes' vertical velocity nonlinear term ($w\ \partial w/\partial z$, where w is the vertical velocity and z is the vertical coordinate). The nonlinear vertical velocity term is included in the superposed component waves, but its 3D effect is different from the sum of the effects of the 2D component waves, since:

$$(w_1 + w_2)\frac{\partial}{\partial z}\big(w_1 + w_2\big) \neq w_1\frac{\partial w_1}{\partial z} + w_2\frac{\partial w_2}{\partial z} \tag{C.9}$$

where the subscripts 1 and 2 indicate the monochromatic waves in the x and y directions, respectively. We note that if $w_1 = w_2$ then the nonlinear term on the left-hand side of equation (C.9) is twice as big as the nonlinear term on the right-hand side. Thus, if the velocity field is linearly superposed to the first order, then the vertical velocity nonlinear terms at the $d1$ corners of the 3D simulation (see figure [C.4]) will be approximately twice that of the superposed waves (or four times that of a single wave). For example, in the first half-period of the sloshing, along the vertical line at the corner $(0, 0, z)$ we have $w < 0$ and $\partial w/\partial z < 0$, while at the corner where $(0.5, 0.5, z)$ we have $w > 0$ and $\partial w/\partial z > 0$. It follows that in the Navier-Stokes equations, along the line of $(0, 0, z)$, the nonlinear term is accelerating the free surface downward; while along the line of $(0.5, 0.5, z)$, the nonlinear term is retarding the upward motion of the free surface. This is exactly the effect illustrated in Figure (C.7). Note that in the corners of the $d2$ diagonal the w_1 and w_2 terms are approximately equal and in opposite directions. Therefore, effect of the nonlinear term in the 3D simulation is close to zero, whereas the nonlinear terms in a superposition of the 2D simulations are of the same sign and will accumulate rather than cancel. Thus, the

3D simulation has less damping in the corners of the $d2$ diagonal than there is in the 2D nonlinear waves, resulting in the surface deformation effect being positive at these corners. Again, this is clearly shown in figure (C.7).

As a further validation of the physical nature of the mode, we can look at the decay rate of the maximum amplitude of the second mode (figure [C.8]). The second mode decays at the rate predicted by 1st order damping theory (which admittedly can only be applied heuristically to an individual wave mode).

C.5 Analysis of cross mode

The cross mode shown in figure (C.5) is a numerically-induced mode of oscillation. Since we are starting with a perfectly symmetrical flow (with reference to a diagonal plane) we should expect that our solution to the Navier-Stokes equations should provide a symmetrical solution with no cross-mode. This would be so if the numerical methods were all applied isotropically. In the real world, we might expect the cross-mode to exist in an experiment due to the inability to get a perfectly symmetrical initial condition.

Our approach has an anisotropy in the numerics that is the result of using an approximate factorization technique in the computation of the estimated (u^*) velocity field of the fractional step method. The application of the approximate factorization is to change a 3D solution requiring a large sparse matrix into the successive inversion of 3 tridiagonal matrices. This is also known as an ADI, or alternating direction implicit method. Although this approach retains the formal order of accuracy of the simulation method, it results in the equations being treated anisotropically in the different coordinate directions. This allows the cross mode to be established from nonlinear interactions at the level of truncation error.

The important question with this numerical mode is whether or not its effect will grow, and dominate the simulation, or whether it will decay. In our initial simulations (reported in Hodges and Street [1996]), this mode grew dramatically. The cause of this growth was the initial numerical method used to solve the kinematic boundary condition and advance the free surface: a Crank-Nicolson method with a 2D approximate factorization. Although this approach retains the formal order of accuracy of

the simulation method, it results in the equations being treated anisotropically in the different coordinate directions. This allows the cross mode to be established from nonlinear interactions at the level of truncation error. We can see from figure (C.9) that this resulted in a linear growth rate of the cross mode. In comparison to the second mode decay, figure (C.8), it appears that the growth of the cross mode could eventually dominate the nonlinear effects. However, when the simulation was allowed to run out to complete damping (about 160 periods), it was seen that the cross mode would not grow indefinitely, but would begin to decay. Its decay rate was less than that of the free surface wave, so at about 120 periods, the cross mode became the dominant free surface mode.

Upon discovering the source of the cross-mode, we switched our free surface advance to an RK4 method that could be applied in an efficient isotropic manner. We can see from the data shown in the figures (C.8) and (C.9) that the cross-mode decays for the RK4 case and remains an order of magnitude smaller than the second mode nonlinear effects. For comparison we see that its decay rate is somewhat less than the theoretical damping of a wave (again heuristically applied). This is probably attributable to the continual feeding of the mode by the anisotropy of the numerics.

Figure (C.10) shows the RMS of the displacement of the cross mode as a function of time. As in figure (C.6) the RMS of the total free surface displacement is shown on a reduced scale for comparison of the wave periods. In this figure we see a rather confused adjustment period for the first 6 or 7 wave periods, then the cross mode settles down to be almost exactly in phase with the total free surface sloshing.

C.6 Analysis of first damping mode

The first damping mode shown in figure (C.5) is similar in appearance to the free surface shape, but opposes the free surface. This represents additional damping of the 3D free surface[1] that is caused both by nonlinear effects and by numerical effects. That this effect is physical can be argued from the nonlinear vertical velocity terms in the Navier-Stokes equations. Since the velocities at the diagonal corner crest and

[1]The additional damping is through comparison to the 2D superposed solution rather than to theory.

trough points are not exactly equal and opposite, their nonlinear terms will not be exactly equal and opposite. Thus, we expect that the 3D simulation should not damp identically to the 2D simulation.

Figure (C.11) provides a comparison of the magnitude of the damping effects of the first damping mode compared to the total damping of the wave. Here we can see that the Crank-Nicolson approach accrued more damping from this mode than the RK4 approach, indicating that there are numerical effects in the damping mode. It appears that the increasing magnitude of the cross-mode was driven by energy extracted (numerically) from the main sloshing through the damping mode. Figure (C.12) provides a closer view of just the Crank-Nicolson and RK4 method damping effects. We can see that the damping effect in the RK4 method appears to be fairly stable, small and linear, while the Crank-Nicolson method is damping at a significantly greater rate.

C.7 Viscous effects

To analyze the viscous effects, we compare the maximum crest-to-trough wave height (as a function of time) to the viscous-damping theory of Lamb (1932):

$$h(t) \;=\; h(0)\, e^{-2\nu k^2 t} \tag{C.10}$$

where $h\,(t)$ is the wave height as a function of time. Figure (C.13) shows the results for the 2D monochromatic sloshing, the 3D sloshing, and Lamb's theory. The results for the wave decay are in reasonable agreement with the theory for the 2D simulation and for the 3D simulations. Results are shown for two 3D simulations with different methods used for the free surface advance: (1) Crank-Nicolson with approximate factorization, and (2) 4th-order Runge-Kutta with 5th-order upwinding of Komori *et al.* (1993). We should not expect the simulations to agree perfectly with the theory since Lamb's derivation is based on an energy argument for a steady, linear, 2D monochromatic wave which is characterized by a single wave number, while we are applying it to a nonlinear 3D case that has at least three significant wave numbers with effects that are time-dependent. Despite this, our results are consistent with Lamb's theory, although not as dissipative.

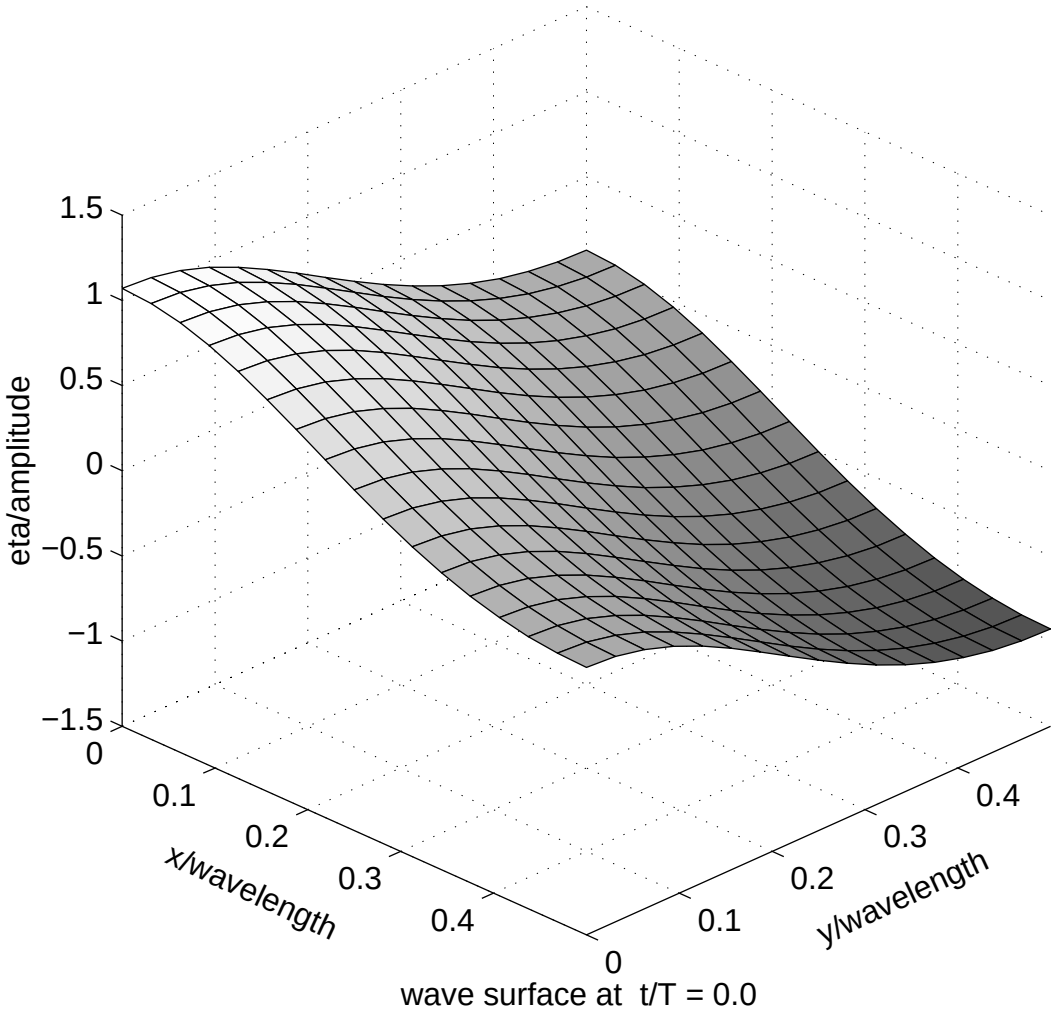


Figure C.1: Superposition of two 2D waves

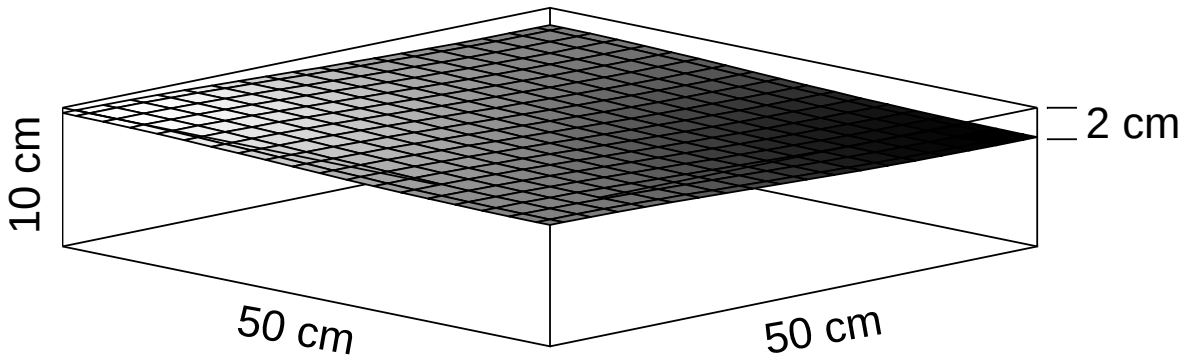


Figure C.2: Computational domain

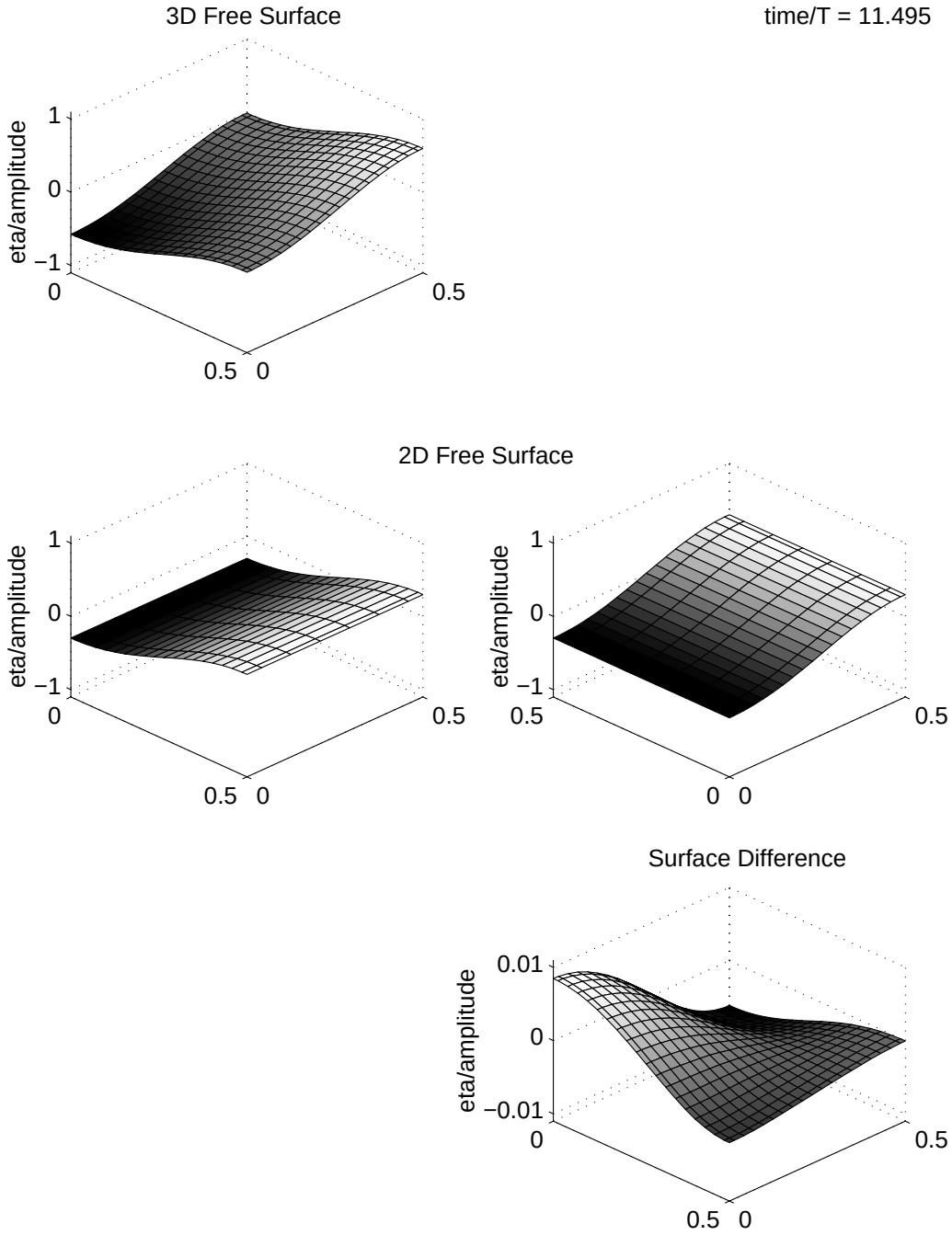


Figure C.3: Computation of surface difference

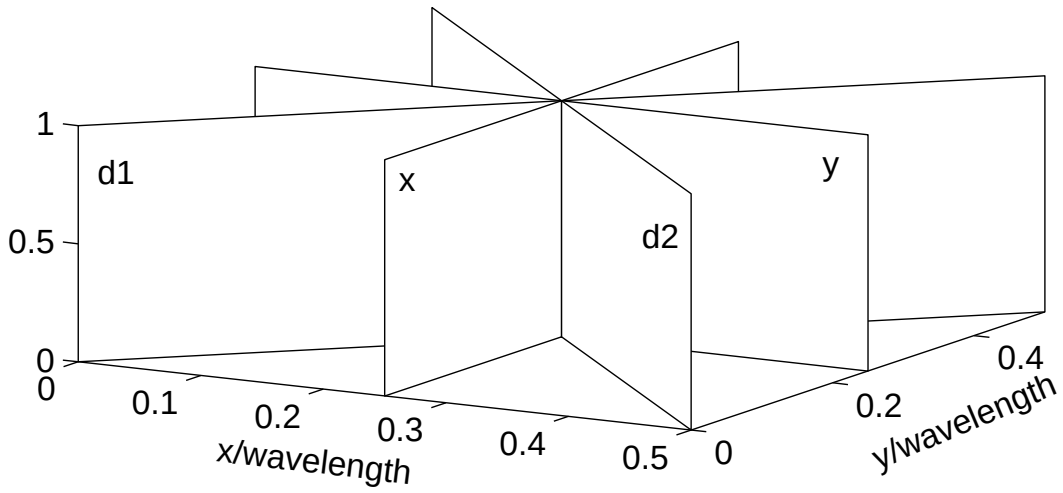


Figure C.4: Symmetry/asymmetry planes

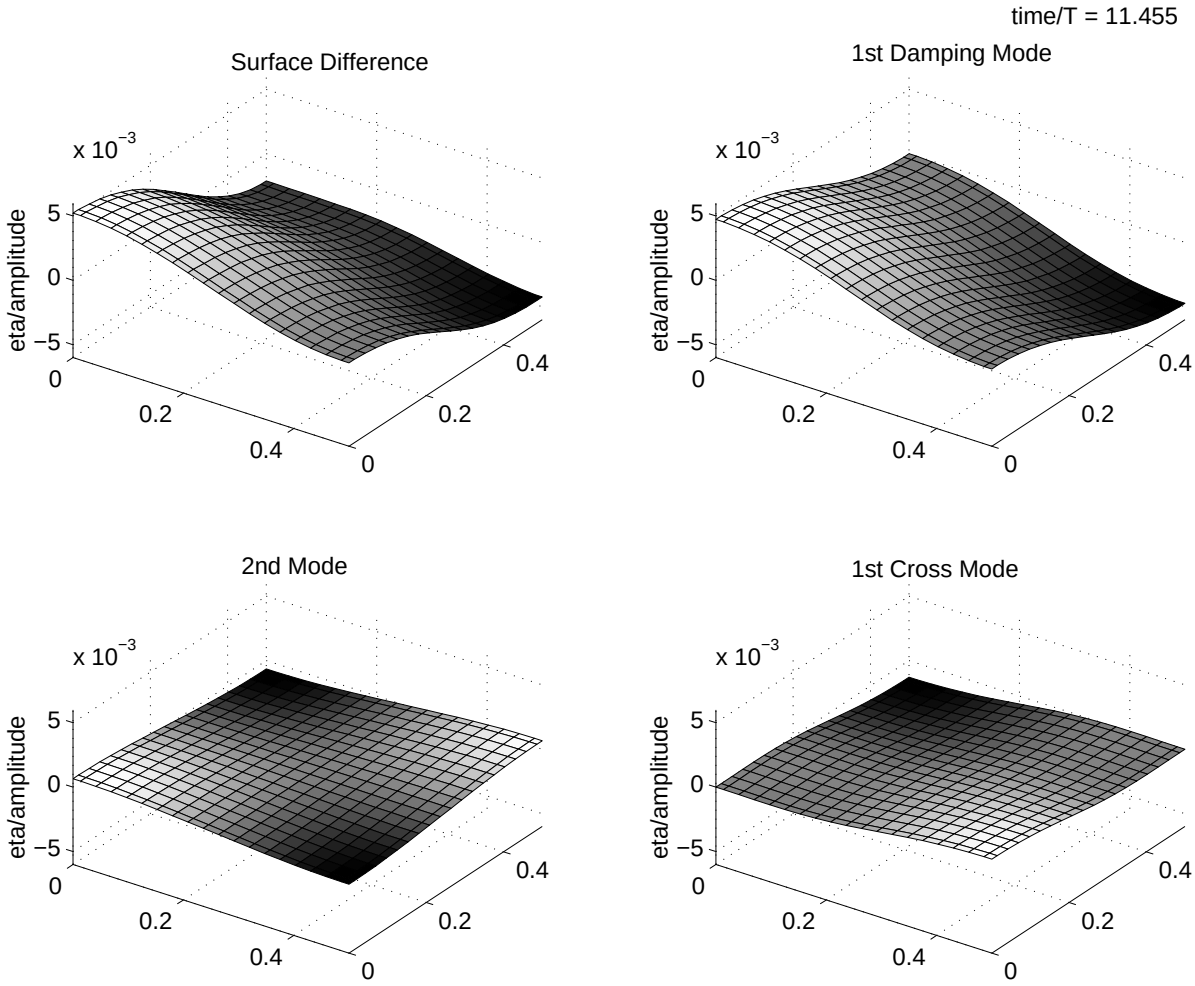


Figure C.5: Decomposition of surface difference

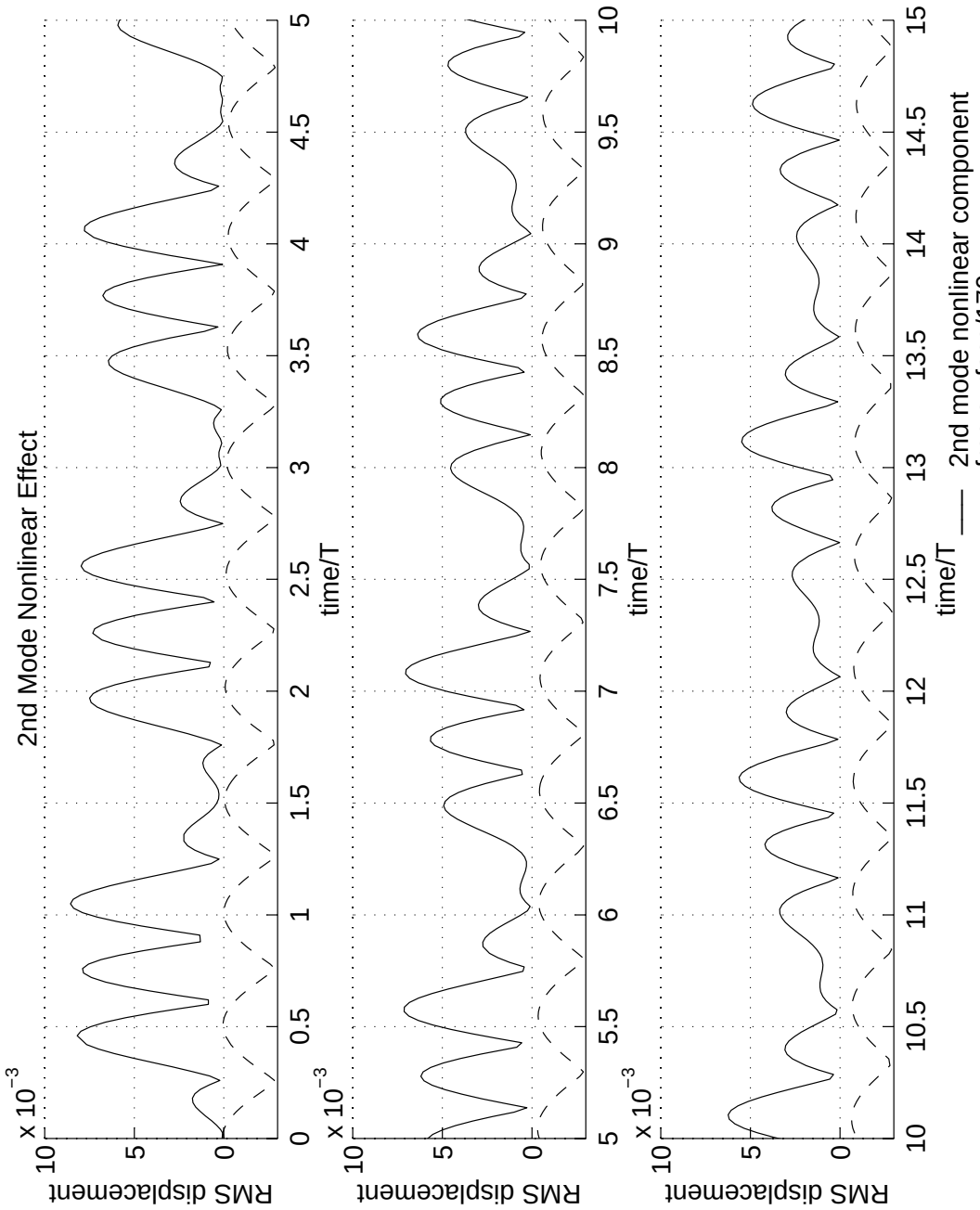


Figure C.6: RMS of second mode deformation

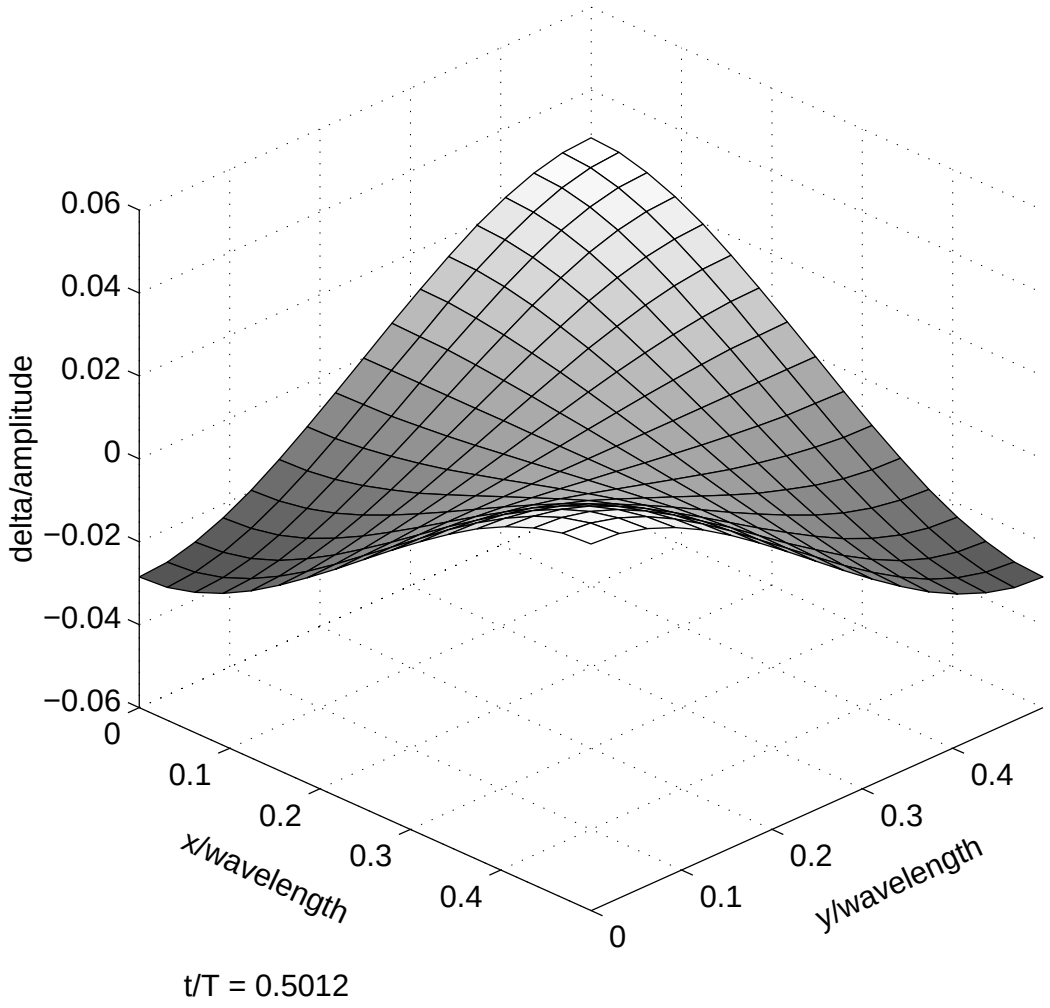


Figure C.7: Second mode deformation

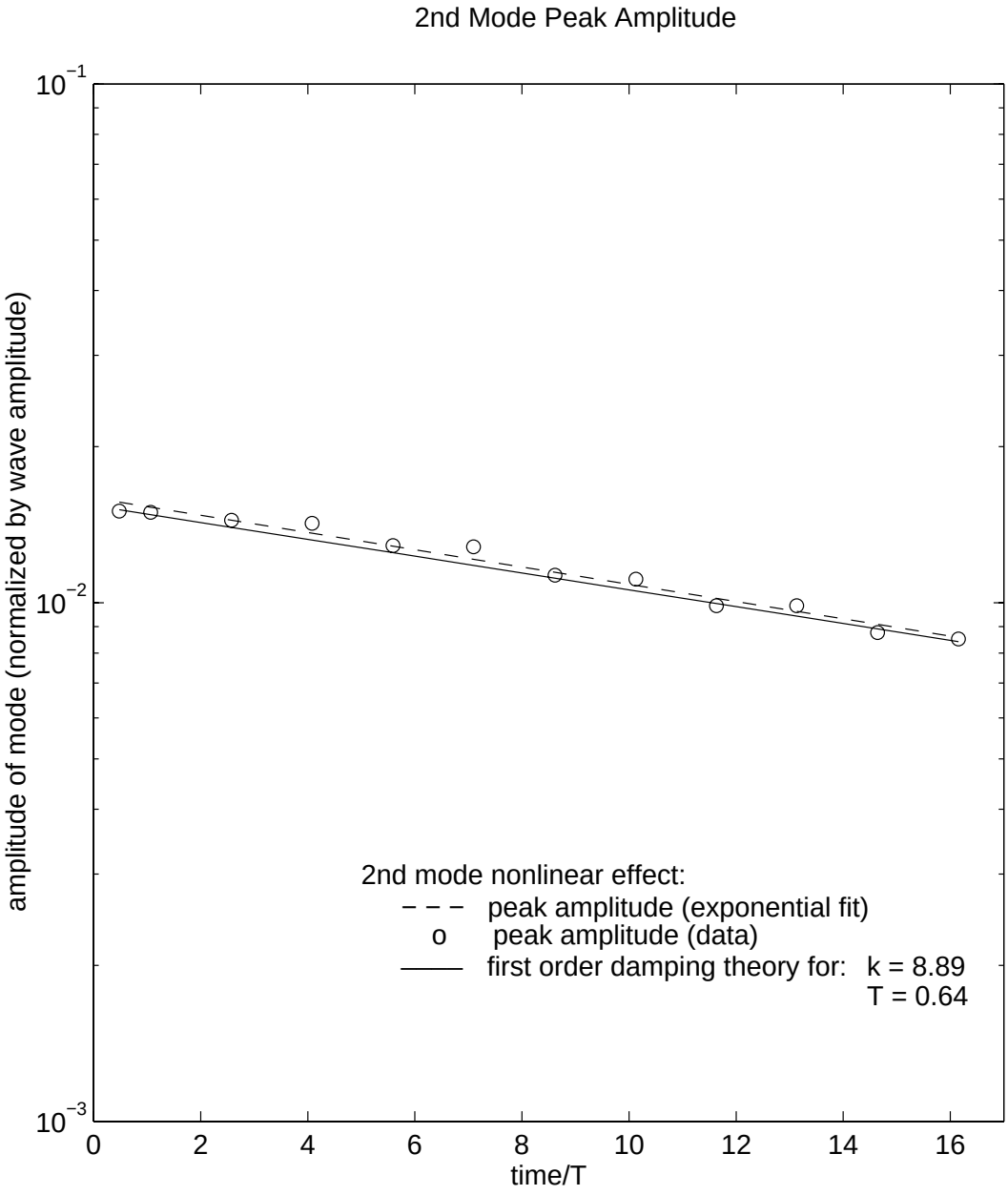


Figure C.8: Decay of second mode peak amplitude

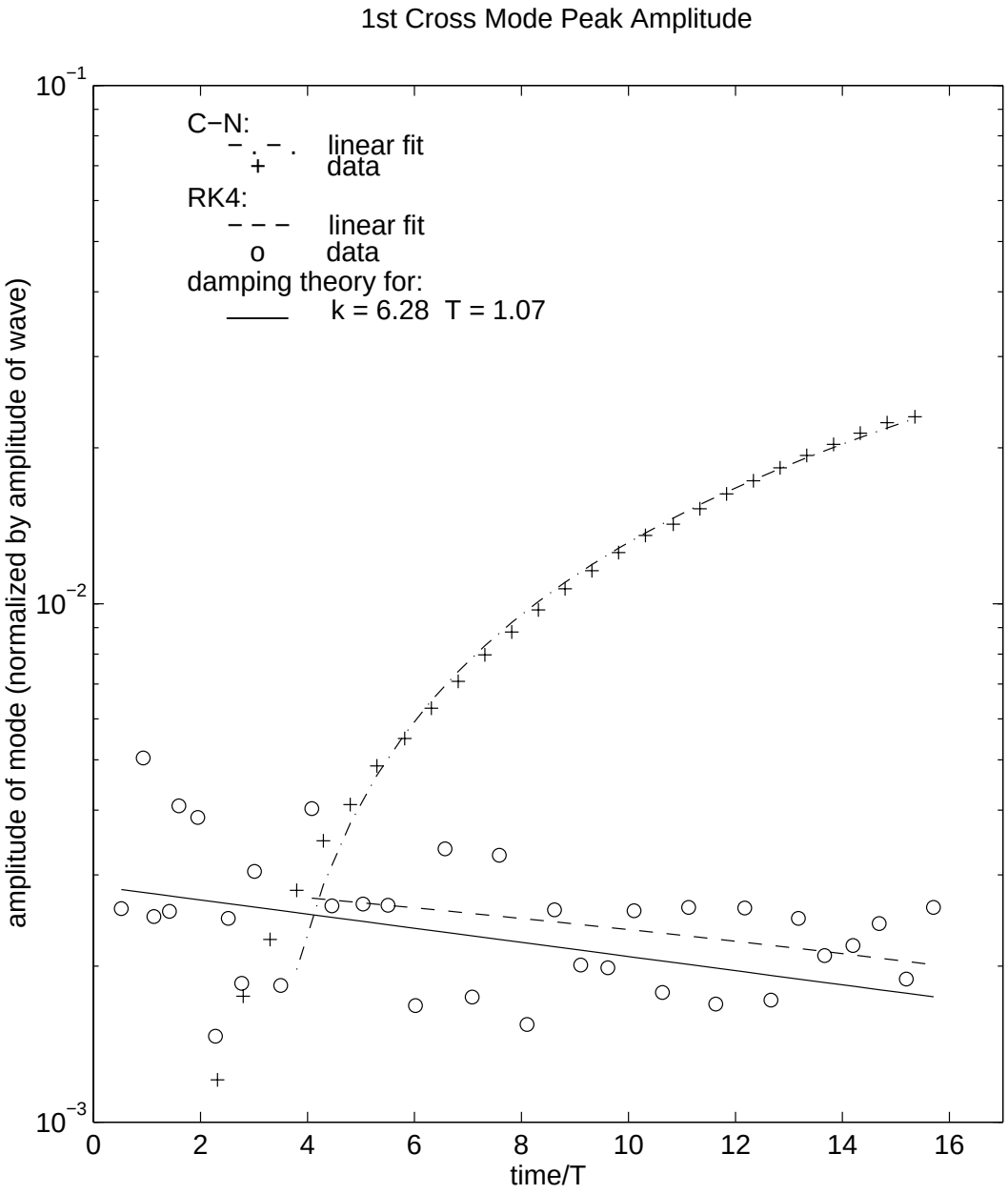


Figure C.9: Evolution of cross mode peak amplitude

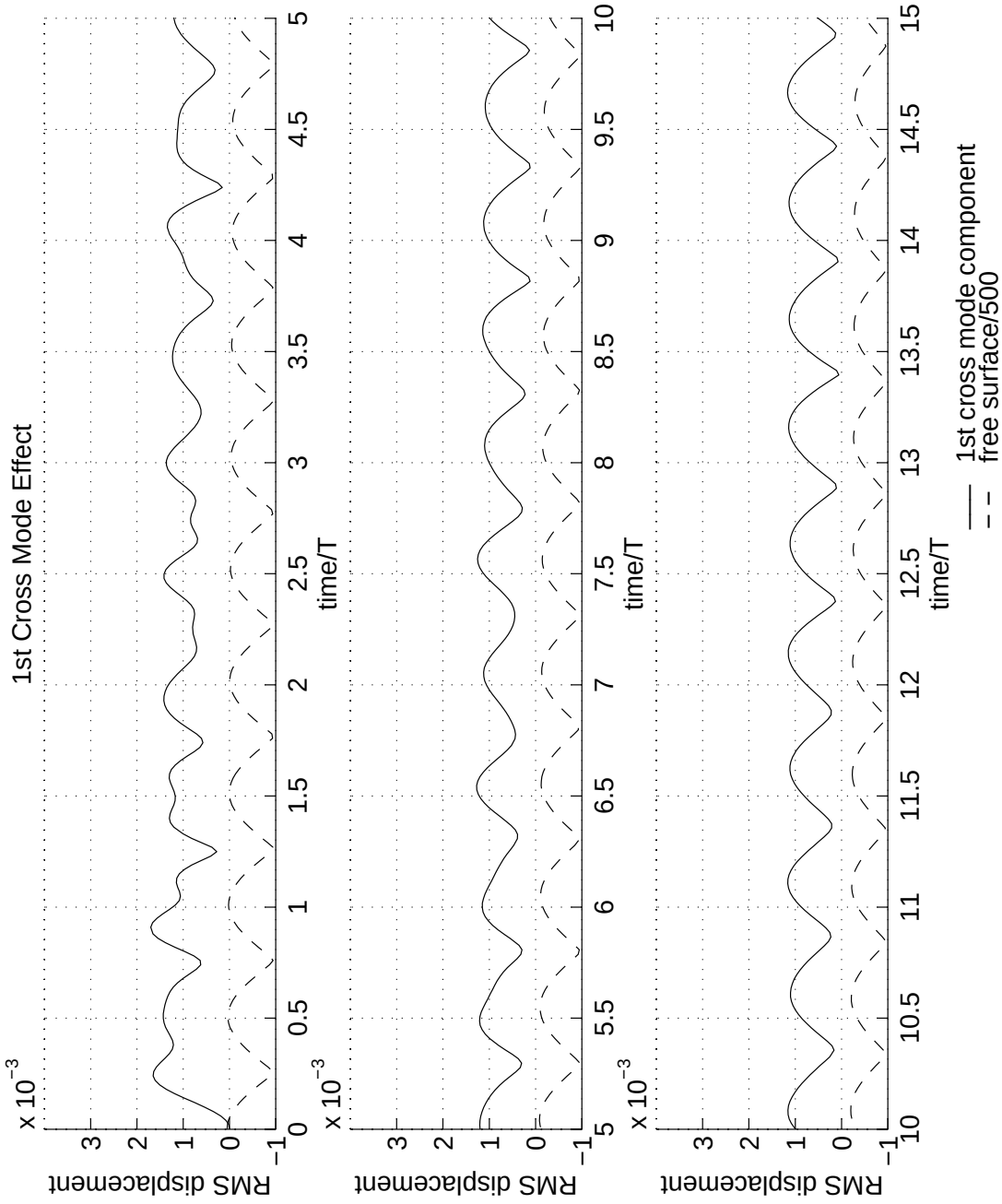


Figure C.10: RMS of cross mode deformation

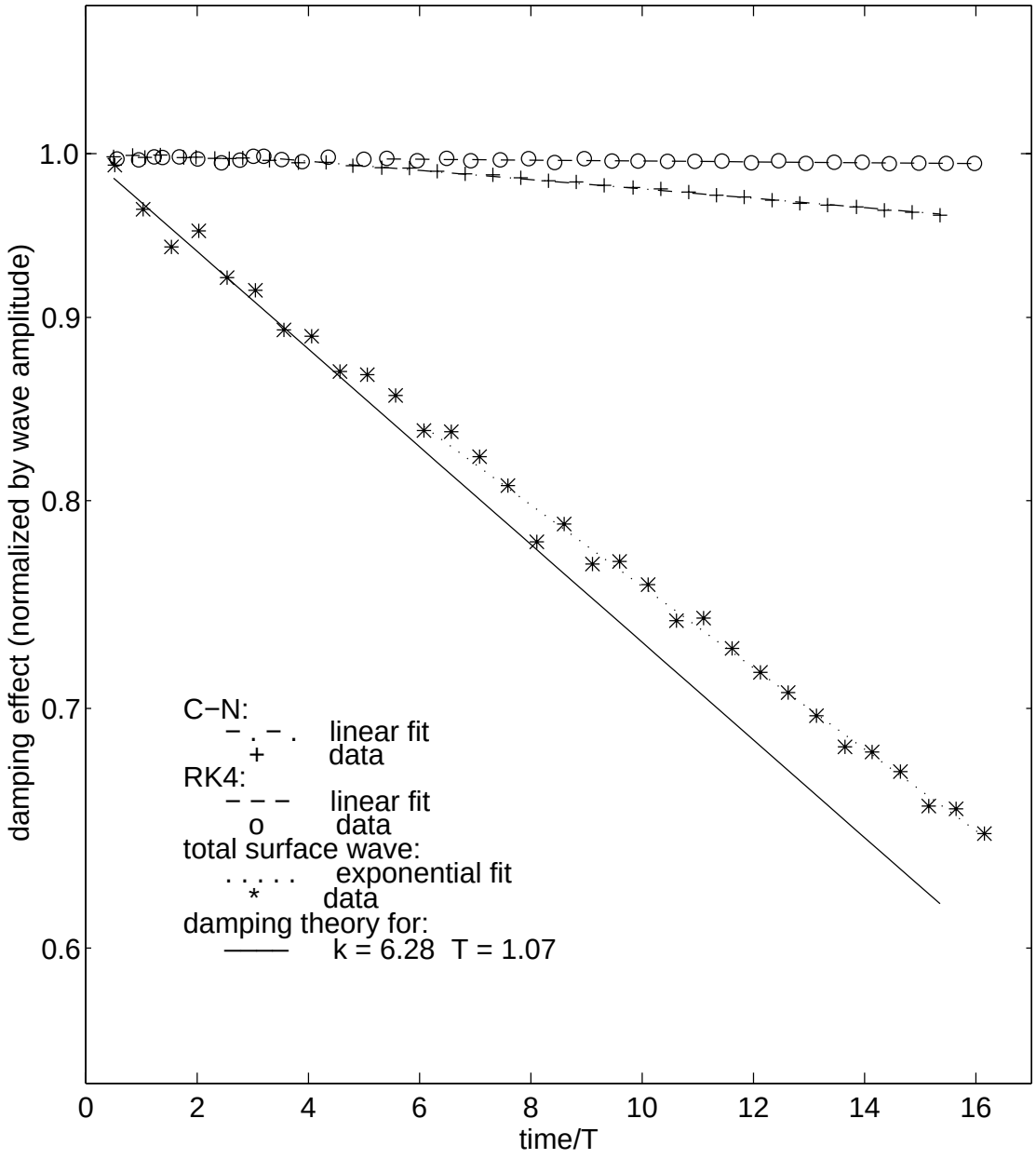


Figure C.11: Effect of first damping mode on wave height

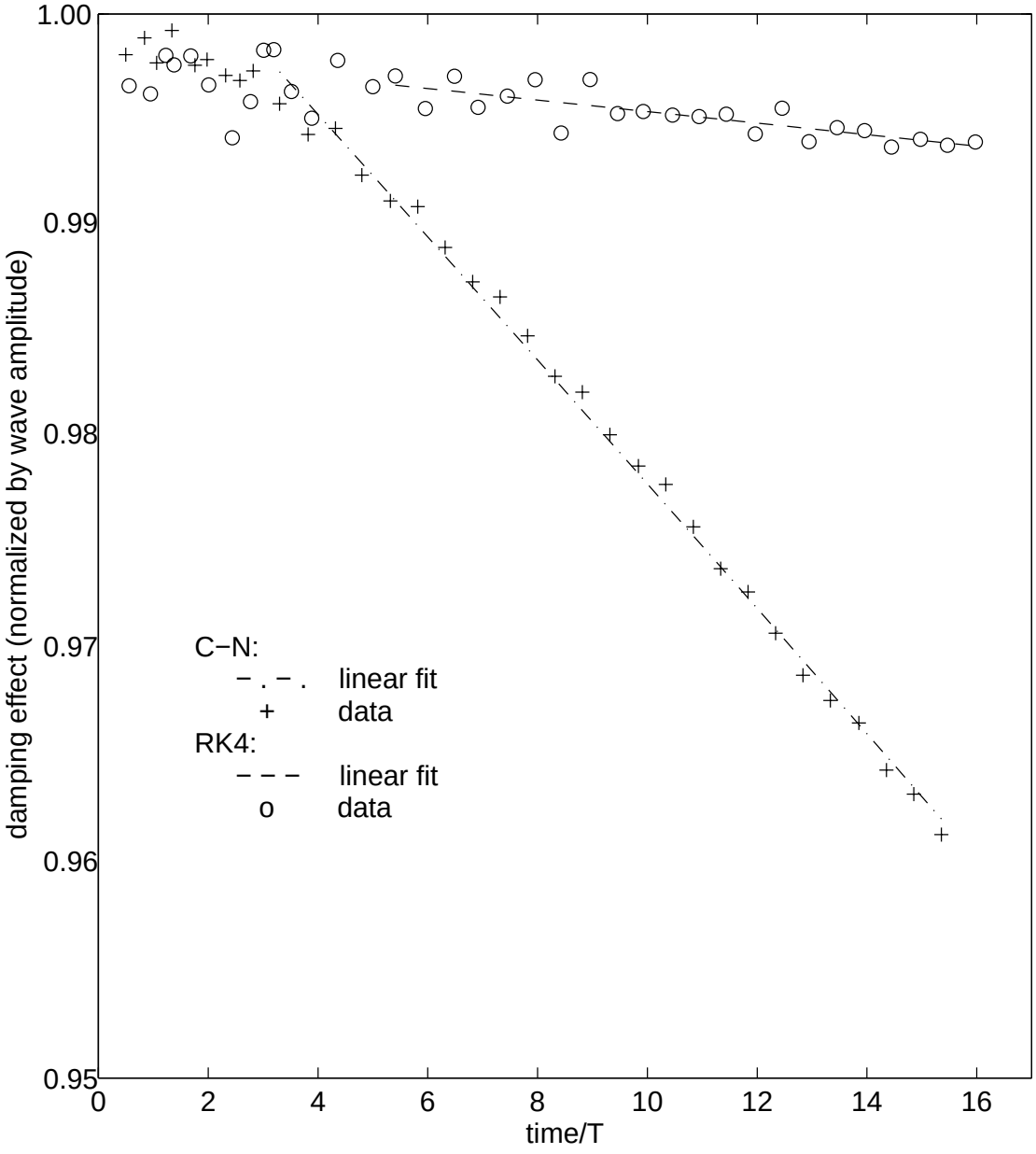


Figure C.12: Effect of first damping mode on wave height (small scale view)

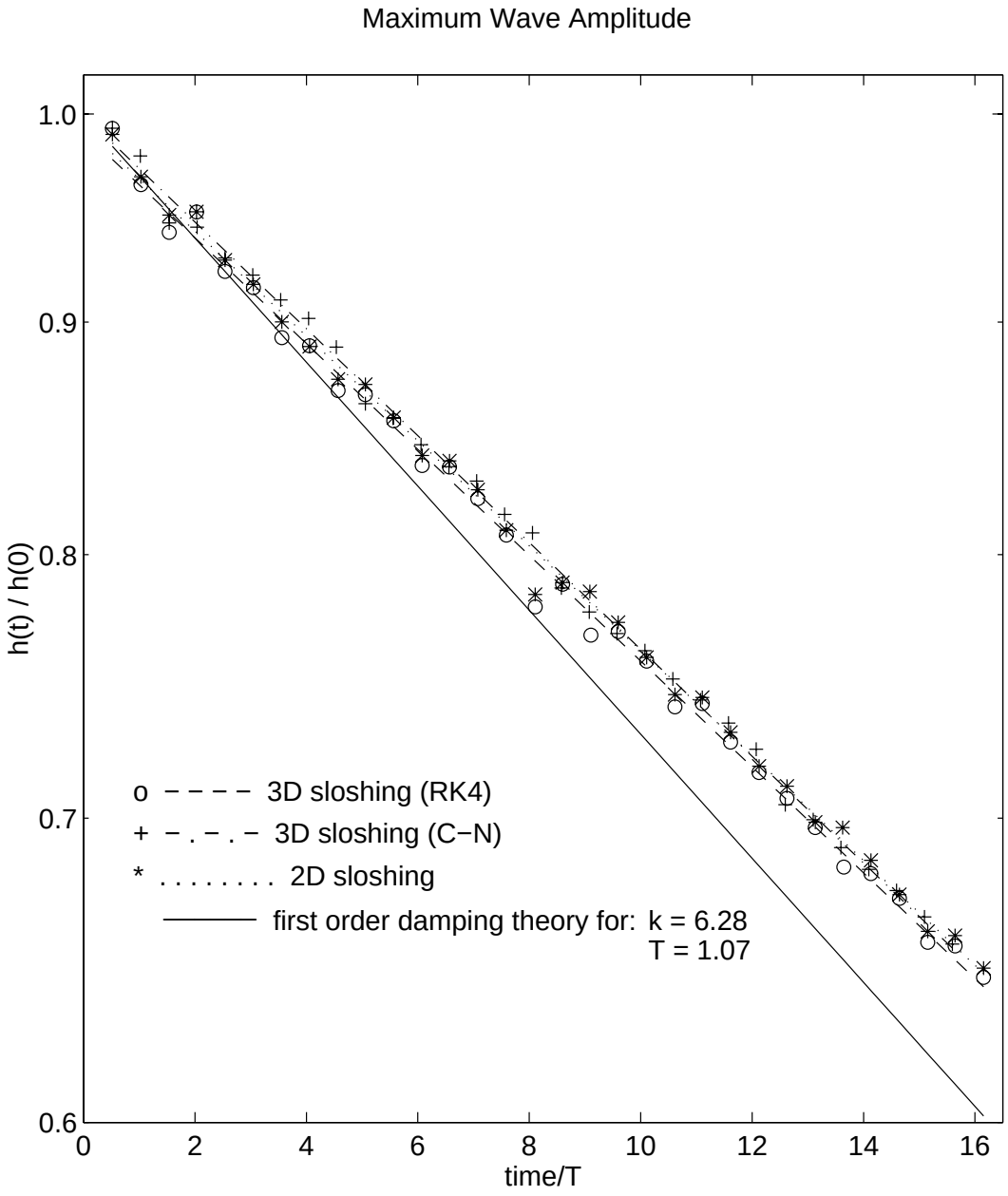


Figure C.13: Change in maximum crest-to-trough height

References

Ananthakrishnan, P., & R.W. Yeung (1994), "Nonlinear interaction of a vortex pair with clean and surfactant-covered free surfaces," *Wave Motion,* **19**, 343-365.

Aris, R. (1962), *Vectors, Tensors, and the Basic Equations of Fluid Mechanics,* Dover, 286 pages.

Batchelor, G.K. (1967), *An Introduction to Fluid Dynamics,* Cambridge, 615 pages.

Belcher, S.E., J.A. Harris, & R.L. Street (1994), "Linear dynamics of wind waves in coupled turbulent air-water flow. Part 1. Theory," *J. Fluid Mech.* **271**, 119-151.

Bendat, J.S., & A.G. Piersol (1986), *Random Data: Analysis and Measurement Procedures,* Wiley-Interscience.

Benilov, A.Y., & B.N. Filyushkin (1970), "Application of methods of linear filtration to analysis of fluctuations in the surface layer of the sea,' *Izv. Akad. Sci. USSR, Atmos. Oceanic Phys.,* **6**, 477-483.

Borue, V., S.A. Orszag, & I. Staroselsky (1995), "Interaction of surface waves with turbulence: direct numerical simulations of turbulent open-channel flow," *J. Fluid Mech.,* **286**, 1-23.

Chan, R.K.-C., & R.L. Street (1970a), *SUMMAC – A Numerical Model for Water Waves,* Technical Report No. 135, Dept. of Civil Eng., Stanford Univ., 155 pages.

Chan, R.K.-C., & R.L. Street (1970b), "A computer study of finite-amplitude water waves," *J. Comput. Phys.,* **6**, 68-94.

Chorin, A.J. (1967), "A numerical method for solving incompressible viscous flow problems," *J. Comput. Phys.,* **2**, 12-26.

Cowen, E.A. (1996), *An Experimental Investigation of the Near-Surface Effects of Waves Traveling on a Turbulent Current,* Ph.D. Dissertation, Dept. of Civil Eng., Stanford Univ., 175 pages.

Dean, R.G. (1965), "Stream function representation of nonlinear ocean waves," *J. Geophys. Res.,* **70**, 4561-4572.

Demirdžić, I., & M. Perić (1988), "Space conservation law in finite volume calculations of fluid flow," *Int. J. Num. Meth. Fluids,* **8**, 1037-1050.

Dommermuth, D.G. (1993), "The laminar interaction of a pair of vortex tubes with a free surface," *J. Fluid Mech.,* **246**, 91-115.

Dommermuth, D.G., & R.C.Y. Mui (1996), "The vortical structure of a near-breaking gravity-capillary wave," *Twentieth Symposium on Naval Hydrodynamics,* National Academy Press, 530-550.

Dommermuth, D.G., E.A. Novikov, & R.C.Y. Mui (1994), "The interaction of surface waves with turbulence," *Free Surface Turbulence*, Rood, E.P., & J. Katz (eds.), FED-Vol. 181, ASME, 123-139.

Dore, B.D. (1977), "On mass transport velocity due to progressive waves," *Q.J. Mech. Appl. Math.,* **30**, 157-173.

Duncan, J.H., V. Philomin, M. Behres, J. Kimmel (1994), "The formation of spilling breaking waves," *Phys. Fluids,* **6**, 2558-60.

Farmer, J., L. Martinelli, & A. Jameson (1994), "Multigrid solutions of the Euler and Navier-Stokes equations for a series 60 Cb=0.6 ship hull for Froude numbers of 0.160, 0.220 and 0.316 (program 1: Navier-Stokes formulations)," CFD Workshop, Tokyo.

Floryan, J.M., & H. Rasmussen (1989), "Numerical methods for viscous flows with moving boundaries," *Appl. Mech. Rev.,* **42**, 323-340.

Fuchs, R.A. (1952), "On the theory of short-crested oscillatory waves," in *Gravity Waves,* National Bureau of Standards Circular 521, Department of Commerce, 187-200.

Garg, R.J. (1996), *Physics and modeling of stratified turbulent channel flows,* Ph.D. Dissertation, Dept. of Mech. Eng., Stanford Univ., 326 pages.

Germano, M., U. Piomelli, P. Moin, & W.H. Cabot (1991), "A dynamic subgrid-scale eddy viscosity model," *Phys. Fluids A,* **3**, 1760-1765.

Gharib, M., D. Dabiri, & X. Zhang (1994), "Interactions of small scale turbulence with a free surface," *Free Surface Turbulence*, Rood, E.P., & J. Katz (eds.), FED-Vol. 181, ASME, 97-102.

Ghosal, S., & P. Moin (1995), "The basic equations for the large eddy simulation of turbulent flows in complex geometry," *J. Comput. Phys.,* **118**, 24-37.

Harlow, F.H., & J.E. Welch (1965), "Numerical calculation of time-dependent viscous incompressible flow of a fluid with a free surface," *Phys. Fluids,* **8**, 2182-2189.

Harlow, F.H., & J.E. Welch (1966), "Numerical study of large-amplitude free-surface motions," *Phys. Fluids,* **9**, 842-851.

Harris, J.A. (1992), *On the Growth of Water Waves and the Motions Beneath Them,* Ph.D. Dissertation, Dept. of Civil Eng., Stanford Univ., 341 pages.

Hinatsu, M. (1992), "Numerical simulation of unsteady viscous nonlinear waves using moving grid system fitted on a free surface," *J. of Kansai Society of Naval Architects, Japan,* **217**, 1-12.

Hino, T., L. Martinelli, & A. Jameson (1993), "A finite-volume method with unstructured grid for free surface flow simulations," *Sixth International Conference on Numerical Ship Hydrodynamics,* Ship Research Institute, Tokyo.

Hodges, B.R., R.L. Street, & Y. Zang (1996), "A method for simulation of viscous, nonlinear, free-surface flows," *Twentieth Symposium on Naval Hydrodynamics,* National Academy Press, 791-809.

Hodges, B.R., & R.L. Street (1996), "Three-dimensional, nonlinear, viscous wave interactions in a sloshing tank," *Proceeding of the Fluids Engineering Summer Meeting 1996, Vol. 3,* FED-Vol. 238, ASME, 361-367.

Horiuti, K. (1987), "Comparison of conservative and rotational forms in large eddy simulation of turbulent channel flow," *J. Comput. Phys.,* **71**, 343-370.

Jiang, J.-Y., R.L. Street, & S.P. Klotz (1990), "A study of wave-turbulence interaction by use of a nonlinear water wave decomposition technique," *J. Geophys. Res.,* **95**, No. C9, 16037-16054

Kassinos, A.C., & J. Prusa (1990), "A numerical model for 3-D viscous sloshing in moving containers," *Recent Advances and Applications in Computational Fluid Dynamics,* Vol. FED-103, ASME, 75-86.

Kim, J., & P. Moin (1985), "Application of a fractional-step method to incompressible Navier- Stokes equations," *J. Comput. Phys.,* **59**, 308-323.

Kim, J., P. Moin, & R. Moser (1987), "Turbulence statistics in fully developed channel flow at low Reynolds number," *J. Fluid Mech.,* **177**, 133-166.

Kinsman, B. (1984), *Wind Waves,* Dover, 676 pages.

Komori, S., R. Nagaosa, Y. Murakami, S. Chiba, K. Ishii, & K. Kuwahara, (1993), "Direct numerical simulation of three-dimensional open-channel flow with zero-shear gas-liquid interface," *Phys. Fluids A,* **5**, 115-125.

Laflin, K.R., & D.S. McRae (1996), "Stable, temporally accurate computations on highly dynamic moving grids," *5th International Conference on Numerical Grid Generation in Computational Field Simulations* , B.K. Soni, J.F. Thompson, J. Häuser, & P. Eiseman (eds.), NSF Engineering Research Center for Computational Field Simulations, Mississippi State Univ. 579-588.

Lam, K., & S. Banerjee (1992), "On the condition of streak formation in a bounded turbulent flow," *Phys. Fluids A,* **4**, 306-320.

Lamb, H.L. (1932), *Hydrodynamics,* 6th edition, Dover, 1945, 738 pages.

Lee, D., & Y.M. Tsuei (1992), "A formula for estimation of truncation errors of convection terms in a curvilinear coordinate system." *J. Comp. Phys.,* **98**, 90-100.

Leonard, B.P. (1979), "A stable and accurate convective modelling procedure based on quadratic upstream interpolation," *Computer Meth. in Appl. Mech. and Eng.,* **19**, 59-98.

Lilly, D.K. (1992), "A proposed modification of the Germano subgrid scale closure method," *Phys. Fluids A,* **4**, 633-635.

Loh, C.Y., & H. Rasmussen (1987), "A numerical procedure for viscous free surface flows," *Appl. Numer. Math.,* **3**, 479-495.

Longuet-Higgins, M.S., & E.D. Cokelet (1976), "The deformation of steep surface waves on water: Part 1: A numerical method of computation," *Proc. Roy. Soc. London A,* **350**, 1-26.

Longuet-Higgins, M.S. (1953), "Mass transport in water waves," *Phil. Trans. Roy. Soc.,* **A.245**, 535-581.

Longuet-Higgins, M.S. (1992), "Capillary rollers and bores," *J. Fluid Mech.,* **240**, 659-679.

Lught, H.J., & S. Ohring (1992), "The oblique ascent of a viscous vortex pair toward a free surface," *J. Fluid Mech.,* **236**, 461-476.

MacCormack, R.W. (1986), *Numerical Computation of Compressible Viscous Flow* Course reader for AA214B, Dept. of Aeronautics and Astronautics, Stanford Univ., 71 pages.

Mahadevan, A. (1994), *A non-hydrostatic mesoscale ocean basin model with a free surface,* Ph.D. Dissertation, Dept. of Civil Eng., Stanford Univ., 126 pages.

McDonald, B.K. (1994), *Modeling Laminar Flow Beneath a Prescribed Small-Amplitude Wavy Surface,* Eng. Dissertation, Dept. of Civil Eng., Stanford Univ., 88 pages.

Meakin, R.L., & R.L. Street (1988), "Simulation of environmental flow problems in geometrically complex domains. Part I: A general coordinate transformation," *Comp. Meth. in Appl. Mech. and Eng.,* **68**, 151-175.

Mei, C.C (1983), *The Applied Dynamics of Ocean Surface Waves,* Wiley-Interscience, 740 pages.

Miyata, H., M. Zhu, & O. Watanabe (1992), "Numerical study on a viscous flow with free-surface waves about a ship in steady straight course by a finite-volume method," *J. Ship Research,* **36**, 332-345.

Moin, P., & J. Kim (1982), "Numerical investigation of turbulent channel flow," *J. Fluid Mech.* **118**, 341-377.

Monismith, S.G., & J.J.M. Magnaudet (1996), "On wavy mean flows, Langmuir cells, strain, and turbulence," manuscript, 9 pages.

Nepf, H.M. (1992), *The production and mixing effects of Langmuir circulations,* Ph.D. Dissertation, Dept. of Civil Eng., Stanford Univ., 134 pages.

Nezu, I., & H. Nakagawa (1993), *Turbulence in Open-Channel Flows,* A.A. Balkema, 281 pages.

Norris, H.L., & W.C. Reynolds (1975), "Turbulent channel flow with a moving wavy boundary," *Dept. of Mech. Eng. Technical Report TF-7,* SUMET 7509, Stanford Univ., 213 pages.

Pan, Y., & S. Banerjee (1995), "A numerical study of free-surface turbulence in channel flow," *Phys. Fluids,* **7**, 1649-1664.

Park, J.-C., M. Zhu, & H. Miyata (1993), "On the accuracy of numerical wave making techniques," *J. of The Society of Naval Architects of Japan,* **173**, 35-44.

Phillips, O.M., (1977), *The Dynamics of the Upper Ocean,* Cambridge, 336 pages.

Raad, P.E., S. Chen, & D.B. Johnson (1995), "The introduction of micro cells to treat pressure in free surface fluid flow problems," *ASME J. Fluids Eng.,* **117**, 683-690.

Rai, M.M., & P. Moin (1991), "Direct simulations of turbulent flow using finite-difference schemes," *J. Comput. Phys.,* **96**, 15-53.

Rood, E.P., & J. Katz (eds.) (1994), *Free Surface Turbulence*, FED-Vol. 181, ASME, 169 pages.

Salvetti, M.V., & S. Banerjee (1995), "*A priori* tests of a new dynamic subgrid-scale model for finite-difference large-eddy simulations," *Phys. Fluids,* **7**, 2831-2847.

Salvetti, M.V., Y. Zang, R.L. Street, & S. Banerjee (1996), "Large-eddy simulation of decaying free-surface turbulence with dynamic/mixed subgrid-scale models," *21st Symposium on Naval Hydrodynamics,* Trondheim, Norway, 14 pages (in press).

Salvetti, M.V., Y.Zang, R.L. Street, & S. Banerjee (1997), "Large-eddy simulation of free-surface decaying turbulence with dynamic subgrid-scale models," *Phys. Fluids,* 58 pages (under review).

Sarpkaya, T. (1996), "Vorticity, free surface, and surfactants," *Annu. Rev. Fluid Mech.,* **28**, 83-128.

Scriven, L.E. (1960), "Dynamics of a fluid interface," *Chemical Engineering Science,* **12**, 98-108.

Shyy, W. (1996), *Computational Fluid Dynamics with Moving Boundaries,* Taylor & Francis, 185 pages.

Smagorinsky, J. (1963), "General circulation experiments with the primitive equations," *Monthly Weather Review,* **91**, 99-152.

Soni, R.K., J.F. Thompson, J. Häuser, & P. Eiseman (eds.) (1996), *5th International Conference on Numerical Grid Generation in Computational Field Simulations* , NSF Engineering Research Center for Computational Field Simulations, Mississippi State Univ. 1130 pages.

Sorenson, R.L., & S.J. Alter (1996), "3DGrape/AL: the Ames/Langley technology upgrade," *5th International Conference on Numerical Grid Generation in Computational Field Simulations* , B.K. Soni, J.F. Thompson, J. Häuser, & P. Eiseman (eds.), NSF Engineering Research Center for Computational Field Simulations, Mississippi State Univ. 343-352.

Sussman, M., P. Smereka, & S.J. Osher (1994), "A level set approach for computing solutions to incompressible two-phase flow," *J. Comput. Phys.,* **94**, 146-159.

Sutherland, J., C.A. Greated, & W.J. Easson (1995), "Variations in the crest kinematics of wave groups," *Applied Ocean Research,* **17**, 55-62.

Thompson, J.F., Z.U.A. Warsi, & C.W. Mastin (1985), *Numerical Grid Generation, Foundations and Applications*, Elsevier, 483 pages.

Thompson, J.F, (1990), "Current developments in grid generation for complex configurations," *Ocean Waves Mechanics, Computational Fluid Dynamics and Mathematical Modelling,* Southampton, 321-340.

Tokaty, G.A. (1971), *A History and Philosophy of Fluid Mechanics,* Dover, 241 pages.

Tsai, W.-T., & D.K.P. Yue (1996), "Computation of nonlinear free-surface flows," *Annu. Rev. Fluid Mech.,* **28**, 249-278.

Walker, D.T., R.I. Leighton, & L.O. Garza-Rios (1996), "Shear-free turbulence near a flat free surface," *J. Fluid Mech.,* **320**, 19-51.

Wang, H.T., & R.I. Leighton (1990), "Three-dimensional vortex interactions with a free surface," *Recent Advances and Applications in Computational Fluid Dynamics,* Vol. FED-103, ASME, 213-222.

Wang, P., Y. Yao., & M.P. Tulin (1995) "Efficient numerical tank of nonlinear water waves, based on the multi-subdomain approach with BEM," *Int. J. Num. Meth. Fluids,* bf 20, 1315-1336.

Wiegel, R.L. (1964), *Oceanographical Engineering,* Prentice-Hall, 532 pages.

Yao, Y., P. Wang, & M. Tulin (1996), "Wave groups, wave-wake interaction, and wave breaking: results of numerical experiments," *Twentieth Symposium on Naval Hydrodynamics,* National Academy Press, 551-567.

Yao, Y., M.P. Tulin, & A.R. Kolaini (1994) "Theoretical and experimental studies of three-dimensional wavemaking in narrow tanks, including nonlinear phenomena near resonance," *J. Fluid Mech.* **276**, 211-232.

Yeung, R.W., & P. Ananthakrishnan (1992), "Oscillation of a floating body in a viscous fluid," *J. Eng. Math.,* **26**, 211-230.

Zang, Y., R.L. Street, & J.R. Koseff (1993), "A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows," *Phys. Fluids A,* **5**, 3186-3196.

Zang, Y. (1993) *On The Development of Tools for the Simulation of Geophysical Flows,* Ph.D. Dissertation, Dept. of Mech. Eng., Stanford Univ., 224 pages.

Zang, Y., R.L. Street, & J.R. Koseff (1994), "A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates," *J. Comput. Phys.,* **114**, 18-33.

This may not seem
Very important, I know.
But it is. *So I'm bothering*
Telling you so.

- Theodor Seuss Geisel