CRWR Online Report 95-3

A GIS Procedure for Merging NEXRAD Precipitation Data and Digital Elevation Models to Determine Rainfall-Runoff Modeling Parameters

by

Seann M. Reed
Research Assistant

and

David R. Maidment, Ph.D.

Principal Investigator

September, 1995

CENTER FOR RESEARCH IN WATER RESOURCES

Bureau of Engineering Research .The University of Texas at Austin

J.J.Pickle Research Campus . Austin, TX 78712-4497

This document is available online via World Wide Web at

http://civil.ce.utexas.edu/centers/crwr/reports/online.html

ACKNOWLEDGEMENT

This research was supported by the United States Army Corps of Engineers Hydrologic Engineering Center in Davis, CA. Information provided by Norm Bingham and Dale Lillie at the Arkansas-Red Basin River Forecast Center (ABRFC) and Dong-Jun Seo and Dennis A. Miller at the NWS Hydrologic Research Laboratory at Silver Springs, Maryland, is also appreciated.

ABSTRACT

The National Weather Service (NWS) Next Generation Weather Radar (NEXRAD) radar program generates a product called StageIII which offers gridded precipitation estimates spatially averaged over grid cells of approximately 16 km^2 and temporally averaged over 1 hour. Hydrologists need to consider how such distributed precipitation estimates may be translated into improved streamflow forecasts. Researchers at the U.S. Army Corps of Engineers Hydrologic Engineering Center (HEC) have proposed using a modified version of the Clark unit hydrograph method to incorporate NEXRAD rainfall data into their streamflow forecasts. The proposed method requires information about the area of each rainfall cell falling within each modeled subbasin and the average flow length from each rainfall cell to the corresponding subbasin outlet. A set of programs was written to obtain this information using Arc/Info GIS and USGS digital elevation models. Properly positioning NEXRAD rainfall cells relative to digital elevation model cells is an important issue. A fundamental problem is that NEXRAD estimates are referenced to a spherical earth datum while data sets describing the land surface (i.e. digital elevation models) are most commonly referenced to an ellipsoidal earth datum. A study of the equations required to transform NEXRAD cells and digital elevation model cells into a common ellipsoid-based map projection is presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0. INTRODUCTION

Accurate rainfall estimates during a storm event are invaluable to a forecaster responsible for flood warnings or reservoir operation. Typically concerned with "real-time" forecasting, a forecaster needs to predict actual flows within the next several hours as opposed to simulating a probabilistic design flood. Traditionally, rain gages have been used for measuring precipitation and telemetry for transmitting real-time records from remote gages to a forecast office. A significant drawback associated with gage information is that data is collected at a point and an interpolation scheme is required to produce a rainfall surface and to calculate watershed-average rainfall. As the time period for analysis diminishes, interpolation from gaging stations becomes less and less reliable because short duration rainfall data are less spatially correlated than are long duration data. In more recent years, ground-based radar has emerged as an effective tool for generating a rainfall surface with high temporal resolution (Smith, 1993). The National Weather Service (NWS) produces gridded precipitation estimates as part of its Next Generation Weather Radar (NEXRAD) program. Traditional lumped runoff model formulations need to be adapted to incorporate this new spatial rainfall information. This report describes a procedure for combining a gridded NEXRAD precipitation surface with a gridded description of surface topography from a digital elevation model (DEM) to generate input for a spatially distributed runoff model. Geographic information systems (GIS) provide standardized functions used to accomplish this task.

NEXRAD data are a significant improvement over traditional methods of estimating rainfall in both space and time. Traditional modeling with the unit hydrograph approach involves using watershed-average values for rainfall. Before NEXRAD, watershed-average rainfall values were as good an estimate as any because rain gage networks were typically sparse and the rainfall distribution between gages could not be accurately determined. National Weather Service River Forecast Centers typically spatially averaged precipitation inputs over areas of 750 to 2600 square kilometers and temporally averaged values over 6 to 24 hours (Lindsey, 1993). The NEXRAD StageIII product (described further below) offers precipitation estimates spatially averaged over grid cells of approximately 16 square kilometers and temporally averaged over 1 hour.

USGS Digital Elevation Models (DEMs) provide a description of the land surface. Arc/Info Grid processing of DEMs yields watershed boundaries and gridded estimates of travel length from each DEM cell in a watershed to the outlet. Arc/Info commands allow

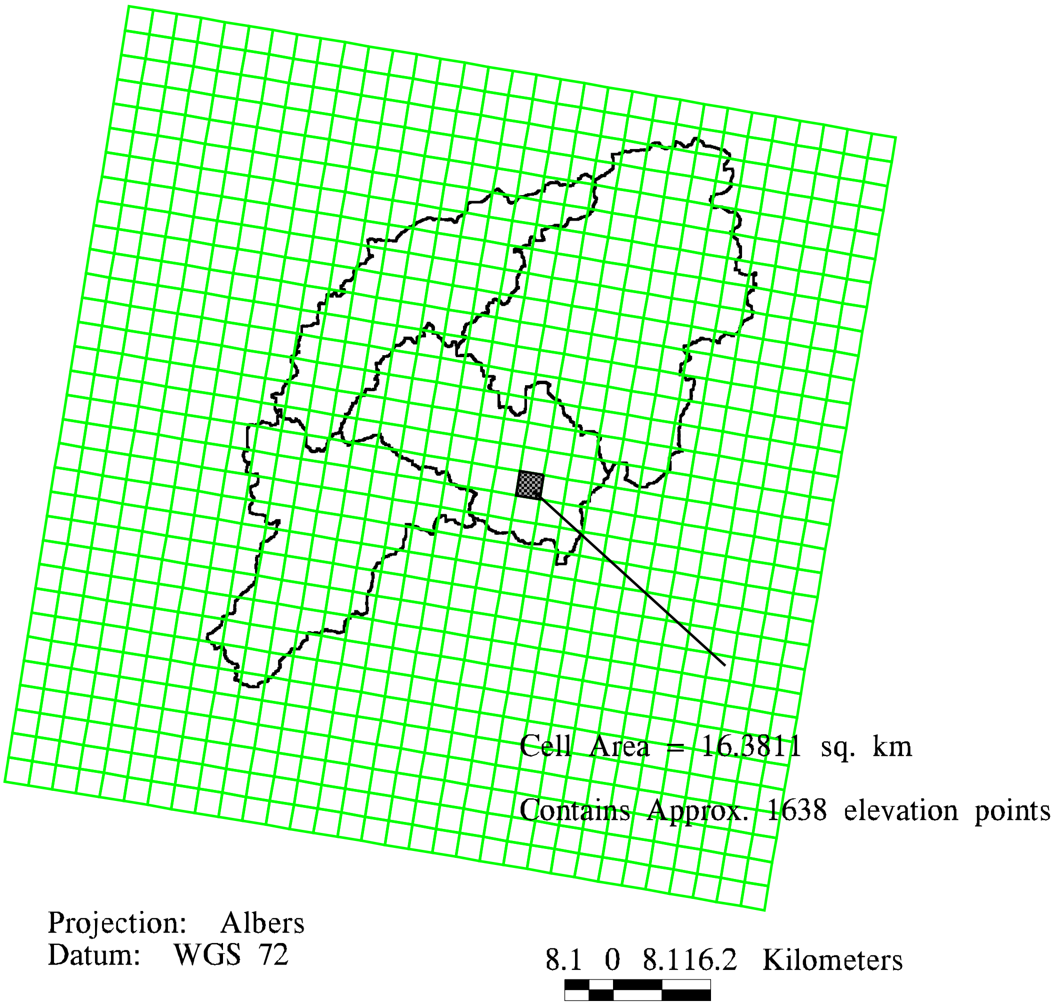

Cell Area = 16.3811 sq. km
Contains Approx. 1638 elevation points
Projection: Albers
Datum: WGS 72
8.1 0 8.116.2 Kilometers

Figure 1.1: StageIII Cells Overlaid on Watersheds at Tenkiller

intersection of watershed boundaries with rainfall cells and subsequent computation of travel length statistics on DEM cells that lie within a given rainfall cell. NEXRAD StageIII rainfall cells are much larger than the DEM cells. For the location and projection parameters selected in this study, a rainfall cell contains about 1600 elevation values as indicated on Figure1.1. Figure1.1 shows StageIII cells overlaid upon our study area — watersheds draining to the Tenkiller Reservoir in eastern Oklahoma and western Arkansas.

The United States Army Corps of Engineers Hydrologic Engineering Center (HEC) is developing "Next Generation Software" designed to replace HEC1, HEC2, and other codes. The Hydrologic Engineering Center is actively evaluating methods for using the StageIII product as input to the "Next Generation Software." The initial approach proposed by the Hydrologic Engineering Center is to use a modification of the Clark conceptual runoff model dubbed "modClark" (HEC, 1995). The modClark method involves computing rainfall excess for each rainfall cell, translating the rainfall excess to a watershed outlet, and routing flow through a linear reservoir at the outlet to generate a direct runoff hydrograph. Inputs required to run the modClark procedure that can be obtained with GIS are the area of each rainfall cell contained within a watershed and a measure of travel length or travel time to the outlet. Additional parameters such as average slope or travel time for each rainfall cell can be easily computed but do not play a role in modClark at this time. An attractive feature of the modClark approach is that the topographic parameters derived from GIS need to be computed only once. This report details the spatial analysis pre-processing needed for the modClark model.

1.1 EXECUTIVE SUMMARY

An Arc/Info GIS procedure has been developed to create a parameter file of information for input to the modClark distributed runoff program (HEC, 1995). This parameter file contains rainfall cell-ID, rainfall cell area, and average travel length from the rainfall cell to a watershed or subwatershed outlet. Three basic tasks required to obtain cell information are (1) to obtain and process DEM for watershed delineation and calculate travel lengths, (2) to properly define the geographic position of rainfall cells relative to the land surface, and (3) to determine the contributing area and travel length from each precipitation cell to watershed outlet(s). The procedure developed here is generic enough to be applicable anywhere in the conterminous United States; the results for a sample study area, the Tenkiller Reservoir drainage basin are presented in this report. Proper geo-positioning of NEXRAD radar estimates relative to GIS data sets describing land surface features is important and is sufficiently complicated to merit special attention. A working method for positioning NEXRAD radar estimates is presented.

Executing the commands and programs described in this report requires Arc/Info, Version 7.0 or higher with the Grid module, a FORTRAN 77 compiler, and the UNIX utilities *gunzip* and *dd*. The ArcViewII program is also useful but not essential. In addition to verbal descriptions of key commands, the syntax for key commands are printed in bold face as a user would enter them; computer responses are printed in italics. Particularly for tasks 2 and 3 above, command sequences have been automated using Arc Macro Language (AML) and FORTRAN programs. Listings of the codes for all programs are attached in the Appendix to this report. AML provides a framework through which this procedure could be further automated and made more user friendly. In fact, the Hydrologic Engineering Center has already modified the attached programs to develop a user friendly training module for generating a modClark parameter file (HEC, June 9, 1995); yet, the functionality remains the same.

1.2 DESCRIPTION OF DATA

Table1.1 summarizes the data sources used in this study and provides Internet addresses for obtaining the data.

Table 1.1: Internet Addresses for Data Sources

Data Source	Internet Address
ABRFC StageIII	http://gopherpc.abrfc.noaa.gov/abrfc
DEMs	http://sun1.cr.usgs.gov/eros-home.html
HUCs	http://h2o.er.usgs.gov/nsdi/wais/water/huc250.HTML
RF1	http://h2o.er.usgs.gov/nsdi/wais/water/rf1.HTML

In this report, both the raster and vector data structures are discussed. A raster data layer stores values in rectangular array of uniform cells and is referred to as a grid. A DEM is an example of raster data. A vector data layer stores points, lines, or polygons and is referred to as a coverage. A point is a single coordinate value, a line is a series of coordinate values, and a polygon is a connected sequence of lines. A point might be used to represent a stream gaging station, a line to represent a stream, and a polygon to represent a watershed. The RF1 and HUC files described below are examples of vector data. Key concepts used to communicate between these two data structures are that a single cell in a grid is equivalent to a point in a coverage, a line of cells in a grid is equivalent to a line in a coverage, and a zone of cells in a grid is equivalent to a polygon in a coverage. Arc/Info offers several functions that convert data layers between the raster and vector data structures.

1.2.1 NEXRAD Data

As part of the NEXRAD program, the National Weather Service, in conjunction with the Departments of Defense and Transportation, is in the process of deploying WSR-88D (Weather Surveillance Radar - 1988 Doppler) weather radars throughout the country. By 1996, 120 NEXRAD radars will be deployed in the United States (Smith, 1993). A single WSR-88D beam has an effective range of approximately 230 km (US Army Corps of Engineers, 1994), covering more than 166,000 km^2. A three stage data processing procedure incorporating information from radars, rain gages, and satellites has been developed by the National Weather Service to generate high quality hourly precipitation estimates. Different stages of processing are designed for different tasks ranging from flash-flood warnings to stream flow forecasting and reservoir operation. Stage I algorithms use radar information and a limited number of rain gage records to generate 1-hour and 3-hour storm totals on a 2 km rectilinear grid. Stage II processing incorporates additional gage data, resulting in a rainfall surface based on gage data alone which is merged with the radar surface. Surface temperature and satellite information are also used in StageII to eliminate anomalous radar echoes. In StageIII, estimates from several radars are mosaiced into a common grid system so that basin-wide stream flow forecasts can be made. StageIII also incorporates interactive quality control by the forecaster. Both StageII and StageIII products provide hourly estimates in the Hydrologic Rainfall Analysis Project (HRAP) grid system, a 4 km grid in a polar Stereographic map projection (Shedd and Fulton, 1993). StageIII hourly estimates for the Arkansas-Red River Basin can be obtained through Internet within 45 minutes after the hour of estimation (Lillie, personal communication, 1994).

Thirteen National Weather Service River Forecast Centers (RFCs) are responsible for forecasts in major river basins of the United States. The StageIII processing procedure is currently operational only at the National Weather Service Arkansas-Red Basin River Forecast Center (ABRFC) in Tulsa, Oklahoma, but will eventually operate at all thirteen River Forecast Centers. The Arkansas-Red Basin River Forecast Center's StageIII data cover all of Oklahoma, and parts of Kansas, Colorado, Arkansas, New Mexico, Texas, and Missouri (Shedd and Fulton, 1993). Although the focus of this study is on the use of StageIII data, StageII data is also useful in watersheds that are covered by the beam from a single radar, especially where StageIII data is not yet available. Because the StageII and StageIII products are both defined in the HRAP grid system, the procedures for positioning StageII or StageIII cells relative to land surface features in other coordinate systems are identical.

The Hydrologic Rainfall Analysis Project (HRAP) grid as defined by Greene and Hudlow (1982) is used to define the location of each average precipitation value in a StageII or StageIII data set. Flat map coordinate systems like HRAP are defined using a datum and a map projection. A datum consists of a three-dimensional mathematical surface (typically a sphere or spheroid) approximating the shape of the earth and a point of origin. A projection transforms features on a three-dimensional surface into a two-dimensional plane. Some projections can be visualized by imagining a beam of light passing through the earth and producing an image on the projection surface; other projections are purely mathematical. The HRAP coordinate system is defined in the polar Stereographic map projection with a spherical, earth-centered datum of radius 6371.2 km. A regular mesh in the polar Stereographic plane defines the HRAP cells. Rainfall estimates are referenced to the lower-left corner of HRAP cells. While the HRAP system is defined on a spherical datum, GIS data sets describing the land surface, including DEMs, are typically defined on an ellipsoidal datum. Chapter 3 of this report describes the transformations required to properly geo-reference HRAP cells and DEMs.

1.2.2 Digital Elevation Models

3-arc second (3”) DEMs, created by the Defense Mapping Agency (DMA) and distributed by the United States Geological Survey (USGS), are also readily available on Internet. These elevation data were generated by the Defense Mapping Agency from cartographic or photographic sources. For cartographic sources, hypsographic features from maps ranging in scale from 1:24,000 to 1:250,000 were digitized and then elevation data were processed into the desired matrix form. The DEMs are stored according to the names of the USGS 1 : 250,000 map sheets. These USGS map sheets cover a 1° x 2° area but elevation data are stored in 1°x 1° blocks. A user can download data for either the eastern or western half of a USGS map sheet. An elevation data point can be found every 3-arc seconds in both the North-South and East-West directions; thus, a 1° x 1° block of data contains 1201 rows and 1201 columns. As illustrated in “Digital Elevation Models” (USGS, 1990), 3” spacing does not correspond to uniform distances between elevation points on the surface of the earth. Because meridians of longitude converge at the poles, the spacing between data points along a parallel of latitude decreases as one moves north or south away from the equator. The spacing along a meridian of longitude is nearly constant and varies only slightly with the curvature of the earth. With a spherical datum, the length along the surface of the earth per radian of latitude (L_ϕ) and per radian of longitude (L_λ) are given by $L_\phi=R$ and $L_\lambda=R\cos\phi$ respectively. In meters, 3” spacing along a parallel of latitude or a meridian of longitude is equivalent to ($L_\lambda * \pi/180°)/1200$ and ($L_\phi * \pi/180°)/1200$ respectively. For example, at 30° N (Houston, Texas) the spacing between elevation points along a parallel of latitude is about 80 meters, but the spacing along a parallel of latitude at 40°N (Philadelphia, Pennsylvania) is about 70 meters. The spacing between data points along a meridian of longitude is about 92 meters at both locations. More complicated equations for an ellipsoidal datum given by Snyder, (1987), p.25, yield similar results. DEM data downloaded from Internet are typically defined on the WGS 72 datum.

In order to process digital elevation data for a hydrologic study, it must first be projected into a flat map coordinate system so that the coordinates are measured in units of distance rather than degrees. This is necessary because GIS functionality for computing area, distance, and slope depends upon the data being in a Cartesian coordinate system. Using Arc/Info functions to compute a slope where the horizontal dimensions are in degrees of latitude and longitude and the vertical dimension is elevation in meters makes no sense because the distance on the ground for a unit change of 3” of latitude is different from the corresponding distance for 3” of longitude and because the horizontal and vertical dimensions in the slope computation are different. During the map projection process, a new grid is created by resampling data from the original grid at uniform spacing in the projected domain, usually 100 m spacing for grids derived from 3” digital elevation data.

1.2.3 Gage Locations, HUCs, and RF1 Files

In addition to DEMs, watershed delineation requires specifying outlet locations. For the Tenkiller study, geographic coordinates of stream gaging stations provided by the Hydrologic Engineering Center were used to locate outlets.

Although not essential for the spatial analysis described in this study, USGS Hydrologic Cataloging Units (HUCs) and the Environmental Protection Agency's River Reach Files (RF1) provide a useful reference frame and a basis for checking watersheds and streams delineated from the DEMs. HUCs in digital form at 1:250,000 scale and RF1 digital line representation of streams in the United States at approximately 1:500,000 scale are available through Internet at the addresses given in Table1.1.

1.3 SELECTING A STUDY AREA

Consultation with John Peters at the Hydrologic Engineering Center and engineers at the Corps of Engineers office in Tulsa, Oklahoma, led to the selection of the 4163 km^2 (1607 mi^2) Tenkiller Reservoir watershed as an appropriate study area for testing this GIS procedure. Several considerations went into choosing this watershed. Most importantly, the watershed to be studied had to be located where the StageIII product was available, limiting the study to within the Arkansas-Red River Basin. It was also important to choose a watershed large enough so that it could be broken into subwatersheds and each of these subwatersheds would be large enough to contain several precipitation cells — otherwise the effects of improved spatial precipitation resolution on runoff computations could not be tested. In addition, subwatersheds free of control structures were chosen to simplify routing considerations. An important question to consider is: Into how many subwatersheds should a watershed be divided? The Arc/Info Grid model allows flexibility in delineating watersheds so that subdivision can range from modeling the Tenkiller watershed as a single lumped unit down to treating each digital elevation model cell as a separate watershed.

1.4 SCALES OF ANALYSIS

As shown in Figure1.1, there are four spatial scales at which hydrologic analysis can be conducted: the 100 m digital elevation cell (0.01 km^2 in area), the NEXRAD 4 km rainfall cell (approximately 16 km^2 in area), the four subwatersheds draining to flow gaging or estimation points (average area = 1040 km^2), or the watershed (4163 km^2) taken as a whole. In this study, the NEXRAD rainfall cell is taken as a hydrologic response unit and its properties are estimated by averaging the corresponding properties of the approximately 1600 digital elevation model cells within the rainfall cell. In particular, the geographic flow distance from each NEXRAD cell to the watershed or subwatershed

subwatershed outlet is estimated by averaging the flow lengths of all digital elevation model cells within the NEXRAD cell boundaries. Where a watershed or subwatershed boundary cuts through a NEXRAD cell, that cell is partitioned into components whose properties are calculated separately. Two test cases have been run — computing average travel lengths from each NEXRAD cell to the watershed outlet for the entire Tenkiller watershed and for Tenkiller divided into four subwatersheds. This report focuses on the more complicated case of analysis with subwatersheds.

1.5 CHOOSING A MAP PROJECTION AND DATUM FOR THE STUDY

Because this study required overlaying data from several different sources, it was necessary to select a standard map projection and datum for analysis. The hydrologic data sets listed in Table1.1 are available at different map scales and in different coordinate systems. Arc/Info GIS allows maps at different scales to be merged easily provided the maps have a common datum, projection, and map units. Arc/Info has built-in capability to transform data sets among many map projections and datums, although some of these transformations are approximate.

A national Albers Equal-Area Conic projection with the parameters listed in Table1.2 was chosen as the standard map projection for this study. This projection is commonly used for maps of the conterminous United States at scales of 1:2,500,000 and smaller (Snyder, 1987). The United States Department of Agriculture uses these same parameters for mapping its State Soil Geographic (STATSGO) database. Using these parameters, the scale error will be slightly less than 1 percent at the center of the United States with a maximum scale error of 1.25 percent at the northern and southern borders (Snyder, 1987). An equal-area projection seems appropriate for hydrologic modeling because drainage area on the globe is preserved in the projected space; therefore, precipitation depth-volume relationships are also preserved. Choosing standard parallels at a regional scale for Oklahoma and surrounding states, different than those listed in Table 1.2, could reduce scale error for the study, but the scale error of the national projection is not large relative to the uncertainty in hydrologic fluxes (i.e., precipitation, evaporation, loss rates). Also, using the national standard map projection allows for flexibility in expanding or contracting the study region without requiring additional coordinate transformations.

Table 1.2: Parameters of Albers Map Projection

First standard parallel	29.5° N
Second standard parallel	45.5° N
Longitude of central meridian	96° W
Latitude of the projection origin	23° N
False Easting	0.0
False Northing	0.0

Geographic datums are defined by an origin, a reference ellipsoid, and the orientation of the ellipsoid relative to the geoid. The two main ellipsoids used in the United States are the Geodetic Reference System of 1980 ellipsoid (GRS 80) which is used in the North American Datum of 1983 (NAD 83) for civilian mapping and in the World Geodetic System of 1984 (WGS 84) for military mapping, and the Clarke (1866) ellipsoid which is used in the earlier civilian datum, the North American Datum of 1927 (NAD 27). The WGS 72 datum based on the WGS 72 ellipsoid is also used; the DEMs downloaded from Internet for the Tenkiller area are defined on the WGS 72 datum. For the Tenkiller study, the decision was made to keep the DEM data in its original datum (WGS 72) and to transform other data sets into this datum. Although transformations of horizontal locations between coordinate systems can be made easily with Arc/Info, no easy adjustment can be made for vertical location. Although not a major concern for present purposes, transforming elevation values between datums might degrade the quality of the information.

For now, choice of an ellipsoidal datum when working with NEXRAD data is a moot point because a spherical earth datum is associated with the rainfall cells. Transformation between spherical and an ellipsoidal datums is discussed in depth in Chapter 3. No transformations between ellipsoidal datums were required at the scale of this study because horizontal shifts were small enough so that required information could be obtained by overlaying data sets defined on different ellipsoidal datums without having to make a datum conversion. For example, in the Tenkiller analysis, coordinates of the stream gaging stations provided by the Hydrologic Engineering Center were most likely taken from a map defined on the NAD 27 datum. An approximate transformation to WGS 72 could be made in Arc/Info by first transforming from NAD 27 to NAD 83 and then from NAD 83 to WGS 72. Experiments showed that horizontal differences between NAD 27 and NAD 83 points are on the order of 55 meters and differences between NAD 83 and WGS 72 are on the order of 10 meters in the Tenkiller area. Knowing the exact location of gaging stations is not required because their locations are only used to choose an outlet cell from a conceptual drainage network defined on a 100 m grid. Incidentally, the geographic coordinates of gaging station locations were reported in decimal degrees to four decimals, implying a North-South position of plus or minus 11 meters — there are 3600 arc seconds in a degree or 0.000277 degrees per arc second; therefore, 0.0001 degrees is equivalent to about 0.36 arc seconds and an arc second is equivalent to about 30 m on the earth’s surface so 30 * 0.36 ~ 11 m. Further discussion of map projection and datum issues can be found in Chapter 3 which discusses transforming precipitation cells to the common map projection and datum.

2.0 METHODOLOGY

As discussed above, the methodology can be divided into three major tasks: (1) DEM processing — delineating streams and watersheds and computing flowlength, (2) generating HRAP cells and properly defining their geographic position relative to the streams and watersheds, and (3) merging DEM data with HRAP cells to determine the contributing area and mean travel length from each HRAP cell to the watershed outlet(s). Figure 2.1 summarizes these tasks and provides a more detailed listing of the steps required to accomplish each task.

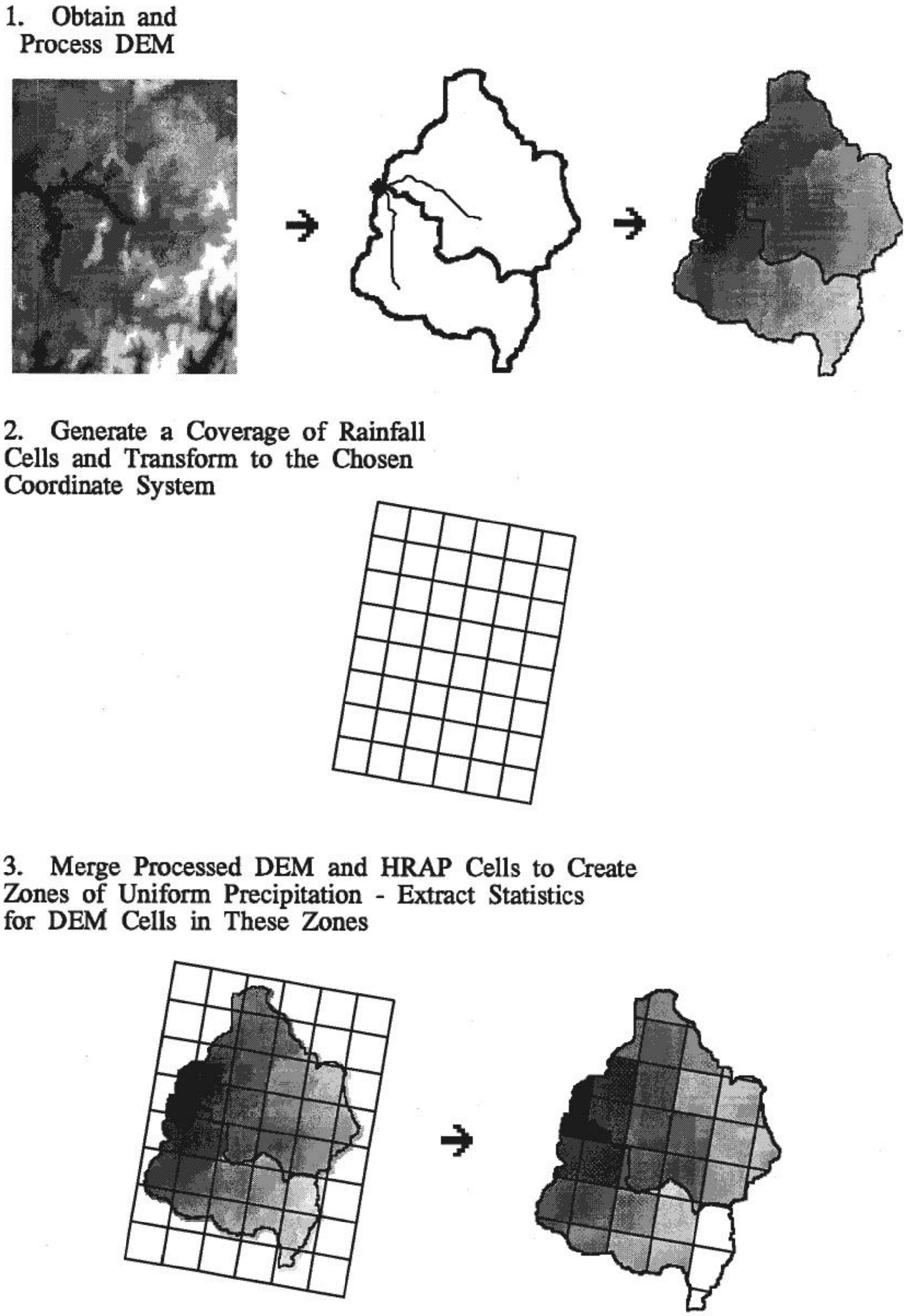
2.1 OBTAIN AND PROCESS DIGITAL ELEVATION MODEL

2.1.1 Create a "Hydrologic" DEM for the Region of Interest

2.1.1.1 Locating Study Region

Working with 3" DEMs consumes a considerable amount of disk storage (approximately 9.8 megabytes for a 1° by 1° block in raw form) and processing time; therefore, several steps are recommended to eliminate large portions of DEM data outside of the study area.

Since watersheds most often define the boundaries of a hydrologic study, an easy first step in establishing a digital reference frame is to identify and cut out USGS Hydrologic Cataloging Units (HUCs) of interest. A hydrologic unit is uniquely identified by an eight digit code. The first two digits of this code identify the hydrologic region, the third and fourth digits identify the hydrologic subregion, the fifth and sixth digits identify the accounting unit, and the seventh and eighth digits identify the cataloging unit. An Arc/Info coverage of the HUCs in the United States at 1:250,000 scale (huc250) contains the first eight digits (HUC), the first six digits (HUC6), the first four digits (HUC4), and the first two digits (HUC2) of the hydrologic unit codes as separate attributes. ArcViewII was used to identify the Tenkiller basin by viewing the huc250 coverage of United States basin outlines and zooming in on the region of interest. Simply double-clicking on the watershed of interest yielded the unit code for the Tenkiller basin (HUC = 11110103). Figure 2.2 illustrates different levels of HUC classification. The Tenkiller basin, with its unique cataloging unit, is highlighted.



1. Obtain and Process DEM

2. Generate a Coverage of Rainfall Cells and Transform to the Chosen Coordinate System

3. Merge Processed DEM and HRAP Cells to Create Zones of Uniform Precipitation - Extract Statistics for DEM Cells in These Zones

Figure 2.1: Procedure Overview

1. Obtain and Process DEM

Create a “hydrologic” DEM for the region of interest

1 Identify and cut out HUCS of interest
2 Create a buffer around the HUCS to extract relevant portions of DEM’s
3 Obtain 3” DEM’s from Internet and convert to ARC/INFO Grid
4 Transform buffered HUCS and DEM to a common map projection
5 Clip the relevant DEM region with buffered HUCS
6 Create “Hydrologic” DEM (i.e., FILL)

Process DEM for stream and watershed delineation.

1 Compute Flowdirection
2 Compute Flowaccumulation
3 Identify “streams” by threshold
4 Label stream links
5 Generate point coverage of relevant gaging stations
6 Identify outlet cells
7 Delineate watersheds
8 Convert Grids to Coverages as needed

Process DEM for Travel Length or Travel Time Parameter

1 Compute Flowlength (on sub-watershed or watershed level)
2 Compute integrated “time-index” parameter (includes computation of slope and FLOWACCUMULATION on a sub-watershed level)

2. Generate a coverage of rainfall cells in the chosen coordinate system.

1 Given the vertical and horizontal extent of the study area in latitude and longitude, write a file of HRAP coordinates that will cover the study area.
2 Transform the HRAP coordinates into geocentric coordinates and write a file that can be used to generate a polygon coverage.
3 Write a file of polygon ID’s and HRAP coordinates that can be attached to the polygon attribute table (PAT) of the HRAP coverage.
4 Generate a polygon coverage.
5 Transform the geocentric HRAP polygon coverage into the common coordinate system.
6 Attach HRAP coordinates as attributes to the PAT of the HRAP coverage.

3. Merge processed DEM and HRAP Cells to Create Zones of Uniform Precipitation — extract statistics for DEM cells in these zones..

1 Intersect HRAP cells with sub-watershed boundaries to create homogeneous precipitation zones.
2 Convert the result of step 1 to a grid of zones.
3 Compute statistics for values defined in the parameter grids (i.e. flowlength or “time-index” grids) based on zones defined in 2.
4 Establish Info Relates between sector_cov.pat and the statistics files.
5 Write HRAP cell parameters to a file that can be used as input to modClark: (i.e. write: hrapx, hrapy, mean flowlength to outlet, cell area).

Figure 2.1(cont.) Step Summary

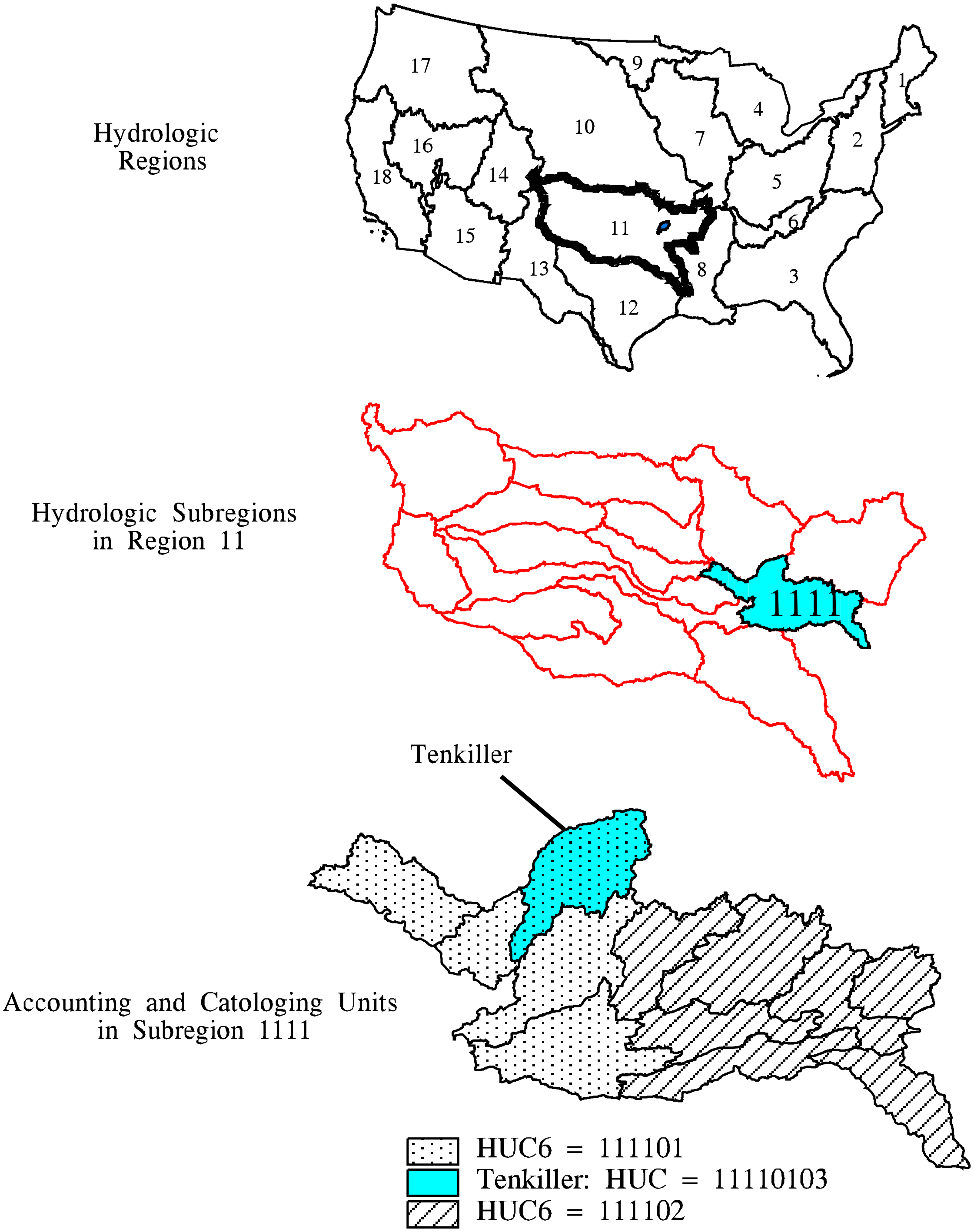

Hydrologic Regions
17
9
1
4
2
10
7
16
5
18
14
6
11
15
3
8
13
12
Hydrologic Subregions in Region 11
1111
Tenkiller
Accounting and Catologing Units in Subregion 1111
HUC6 = 111101
Tenkiller: HUC = 11110103
HUC6 = 111102

Figure 2.2: Identifying the Tenkiller Hydrologic Cataloging Unit

Although the ArcViewII Query Builder used to generate Figure 2.2 was useful for creating displays of selected polygons (watersheds), the actual coverage (huc250) remained unchanged at this point. To create a new HUC coverage containing only the Tenkiller basin, the Arc Reselect command was used. For this study, Reselect extracted a set of map features from the input coverage huc250 based upon the eight digit HUC attribute to produce an output coverage huctk

```
RESELECT <in_cover> <out_cover> {in_feature_class}
Arc: reselect huc250 huctk poly
  Reselecting POLYGON features from HUC250 to create HUCTK
  Enter a logical expression. (Enter a blank line when finished.)
>: res huc = 11110103
>: ~
n
n
```

Next, a buffer was placed around the coverage huctk to ensure that all relevant portions of the DEM were extracted. Because a watershed boundary delineated from a DEM does not exactly coincide with the HUC boundary, creating a buffer around huctk ensures that all DEM cells draining to Tenkiller Reservoir are selected. The Buffer command in Arc/Info was used for this purpose. Figure 2.3 illustrates a 10 km buffer generated for the Tenkiller HUC with the syntax below. Figure 2.3 also shows RF1 features within the Tenkiller basin.

```
BUFFER <in_cover> <out_cover> {buffer_item} {buffer_distance} {fuzzy_tolerance} {LINE | POLY | POINT |NODE}
Arc: buffer huctk huctkbuff # # 10000.0 # poly
```

The necessary buffer size cannot be predicted with certainty. In this example, a 10,000 meter buffer was used, but comparison between the Tenkiller HUC boundary and the Tenkiller boundary delineated from the DEM in a later step reveals that a 5,000 meter buffer would have been adequate for Tenkiller.

2.1.1.2 Obtaining Digital Elevation Models

3" DEMs can be obtained from Internet and converted into Arc/Info Grid format. Using Mosaic, the DEMs can be reached by choosing "Open URL" from the "File" menu and entering the address — http://sun1.cr.usgs.gov/eros-home.html — this is the “EROS Home Page.” At the “Home Page,” select “US Geodata (FTP File Access).” At this point, DEM files may be downloaded “via State”, “via Graphics”, or “via Alphabetical

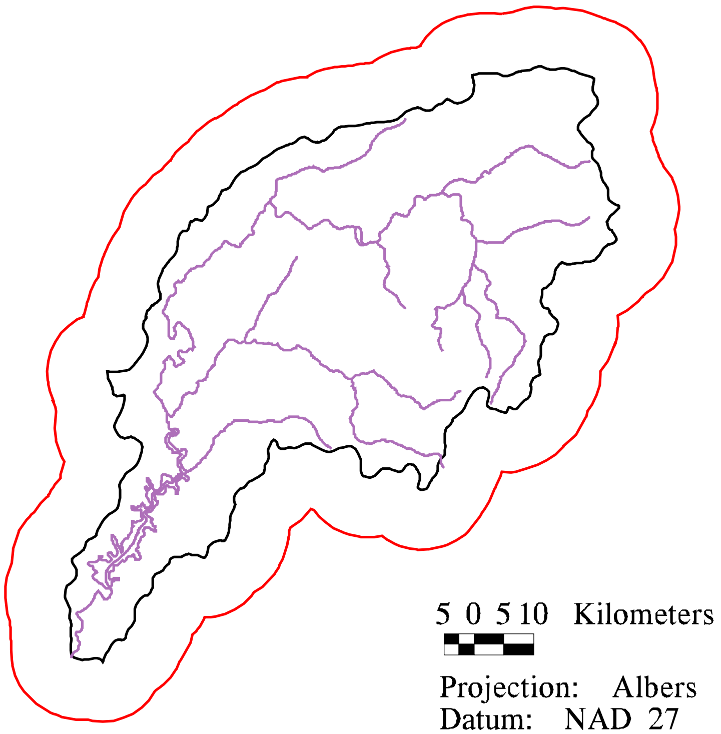


Figure 2.3: Buffered Tenkiller HUC with RF1 Features

List." In any case, the user needs to know the names of the names of the USGS 1:250,000 Index Maps that cover his or her study area. The 1° by 1° blocks downloaded for the Tenkiller analysis were Fort Smith (east and west) and Tulsa (east and west). 3" DEMs may also be obtained by anonymous ftp at edcftp.cr.usgs.gov in the directory /pub/data/DEM/250. DEM files can be downloaded in compressed or uncompressed form. Compressed files can be uncompressed using the UNIX utility gunzip. For example,

```
Unix: gunzip -v tulsae.gz
```

The United States Geological Survey distributes DEM files in a format that is imcompatible with Arc/Info because the record lengths are too long. A file's record length can be modified using the UNIX command dd. With the syntax printed below, dd adds a carriage return at the end of every 1024 bytes.

```
Unix: dd if=tulsae of=tulsaedel ibs=4096 cbs=1024 conv=unblock
   {if — input file; of — output file; ibs — input block size in bytes;
   cbs — conversion buffer size; conv=unblock — convert fixed length records to variable length}
```

A DEM file with appropriate delimiters can be converted into an Arc/Info Grid using the Arc command demlattice.

```
DEMLATTICE <in_dem> <out_lattice> {USGS | TAME} {z_factor}
Arc: demlattice tulsaedel tulsaeg USGS
```

For Tenkiller, the grids tulsaeg, tulsawg, fsmitheg, and fsmithwg were merged into one grid.

```
Arc: grid
Grid: totgrid = merge (tulsaeg, tulsawg, fsmitheg, fsmithwg)
```

2.1.1.3 Projecting the DEM

In order to extract relevant portions of the DEM for processing, the buffered Tenkiller HUC (huctkbuff) and the DEM grid (totgrid) were transformed to a common map projection. For this step, HUCs (already defined in the standard Albers projection) could be approximately transformed from NAD27 to WGS72; however, this transformation is unnecessary because horizontal shifts due to converting between two ellipsoidal datums are very small relative to the size of the buffer. Therefore, the only transformation required in this step was to project the DEM grid from geographic coordinates into the common Albers projection.

```
(*)  PROJECT ( <grid>, {projection_file}, { NEAREST | BILINEAR | CUBIC },
       {out_cellsize} )
Grid:  totalbg = project (totgrid, geoalb.prj, bilinear, 100.0)
```

With this syntax, the projection parameters are specified in the file geoalb.prj. The contents of this file are listed below:

```
input
projection geographic
units ds
datum wgs72
parameters
output
projection albers
units meters
datum wgs72
parameters
29 30 00
45 30 00
-96 00 00
23 00 00
0.0
0.0
end
```

During projection, Arc/Info resamples the elevation points at 3” spacing to generate a grid (totalbg) with uniform spacing of 100 meters. This cell size was chosen to make subsequent area computations simple. Arc/Info offers nearest, bilinear, and cubic resampling algorithms. Either the bilinear or cubic algorithms, in which a weighted distance average of the values in surrounding input cells is taken to determine the value of an output cell, are most appropriate for continuous data such as DEMs. The option for bilinear resampling was specified in this case. When a new elevation point is determined with bilinear sampling, the four nearest elevation values to the desired point in the output projection are used. The units of the input DEM grid are decimal seconds and the units of the output projection are meters. Decimal seconds (DS) and decimal degrees (DD) are common units for geographic coordinates. The conversion is simply, DS = DD * 3600.

Next, the desired region of DEM cells were clipped with the buffered Tenkiller HUC. The Grid Setwindow command reduces the analysis window to the mapextent of a specified coverage (huctkbuff). The mapextent of a coverage is always rectangular in shape. The snap_grid is totalbg which means that any grid created in this analysis window is aligned with totalbg.

```
SETWINDOW <GRID | COVERAGE | xmin ymin xmax ymax | *> {snap_grid}
Grid: setwindow huctkbuff totalbg
```

Once the analysis window was set, a new grid was created (tkalbg) that contains the values of totalbg within the analysis window.

```
Grid:  tkalbg = totalbg
```

These two commands reduced the grid totalbg containing 4,091,049 data points to a more manageable 1,340,932 data points.

2.1.2 Filling Sinks

Raw DEMs downloaded from Internet may contain one or more sinks. A sink is a grid cell or group of cells that has an elevation no higher than each of its immediate neighbors. Sinks need to be removed for runoff modeling with current Arc/Info algorithms. The removal of sinks creates a so-called "hydrologic" DEM. Two methods for removal of sinks available in Arc/Info are the Grid Fill command and the Arc Topogrid command. The Fill command simply alters the elevation of the sink point(s) to the height of its/their boundary cell(s) with the lowest elevation. The Fill command should be performed after grid projection because data resampling during the projection creates artificial sinks. The Fill command was used in the Tenkiller analysis.

```
FILL <in_grid> <out_grid> {SINK | PEAK} {z_limit} {out_dir_grid}
Grid: fill tkalbg tkalbgf
```

The Arc command Topogrid provides a more refined method for generating a hydrologically correct grid. Following Hutchinson, 1989, Topogrid uses a more sophisticated drainage enforcement algorithm to fill sinks and it can incorporate digital streamline data into its interpolation scheme. Topogrid is written to generate a hydrologic DEM given a point coverage of elevation values (expected to be more sparse than a DEM with sinks) or a line coverage of elevation contours. Although cumbersome, Topogrid can be used to modify raw USGS DEMs, enforcing drainage and forcing delineated streams to coincide more closely with digitized streams. Elevation points in USGS DEMs are rounded to the nearest meter, often resulting in large flat areas in stream valleys for which no accurate flow direction can be inferred. By incorporating digital streamline data, Topogrid more accurately represents the land surface in stream valleys. Use of Topogrid requires obtaining digital streamline data, obtaining DEM or hypsography data, transforming these data into a common coordinate system, and making sure that all stream arcs are oriented downstream. Experiments using Topogrid in the Allegheny watershed were successful on a region of approximately 90,000 cells but unsuccessful on a larger region of approximately 530,000 cells. Because of difficulties working with large numbers of elevation points, Topogrid was not used in the Tenkiller analysis.

An AML script could have been used to execute the Arc/Info and UNIX commands described in this section. The box below contains a simple script that could have been used to create a hydrologic DEM for the region of interest. Of course, a more sophisticated AML with error checking, flexibility to accommodate an arbitrary number of DEM files, input files located in different directories, messages to the user, menus, etc., could be written.

Table 2.1: Summary Script for Creating a Hydrologic DEM

```
reselect huc250 huctk poly  /* Select relevant HUCs
  res huc = 11110103
  ~
  n
  n
buffer huctk huctkbuff # # 10000.0 # poly
/*  Assuming that 4 DEM files have been downloaded from internet
/*  using Mosaic, Netscape, ftp, etc. and given the names
/*  tulsae.gz, tulsaw.gz, fsmithe.gz, and fsmithw.gz
&sv dem1 = tulsae; &sv dem2 = tulsaw
&sv dem3 = fsmithe; &sv dem4 = fsmithw
&sv count = 1
/*  Uncompress, reblock, and import all of the DEM files into Arc/Info format.
&do &while %count% le 4
   &sv filename = [value dem%count%]
   &sys gunzip -v %filename%.gz
   &sys dd if=%filename% of=%filename%del ibs=4096 ~
   cbs=1024 conv=unblock
   demlattice %filename%del %filename%g USGS
   &sv count = %count% + 1
&end
grid    /* Start Grid subprogram.
totgrid = merge (%dem1%g, %dem2%g, %dem3%g, %dem4%g)      /* Merge DEM
grids.
totalbg = project (totgrid, geoalb.prj, bilinear, 100.0)
setwindow huctkbuff totalbg    /* Set analysis window.
tkalbg = totalbg    /* Define a new grid of reduced size.
fill tkalbg tkalbgf    /* Fill sinks.
q    /* Quit grid.
&return
```

2.2 PROCESS DEM FOR STREAM AND WATERSHED DELINEATION

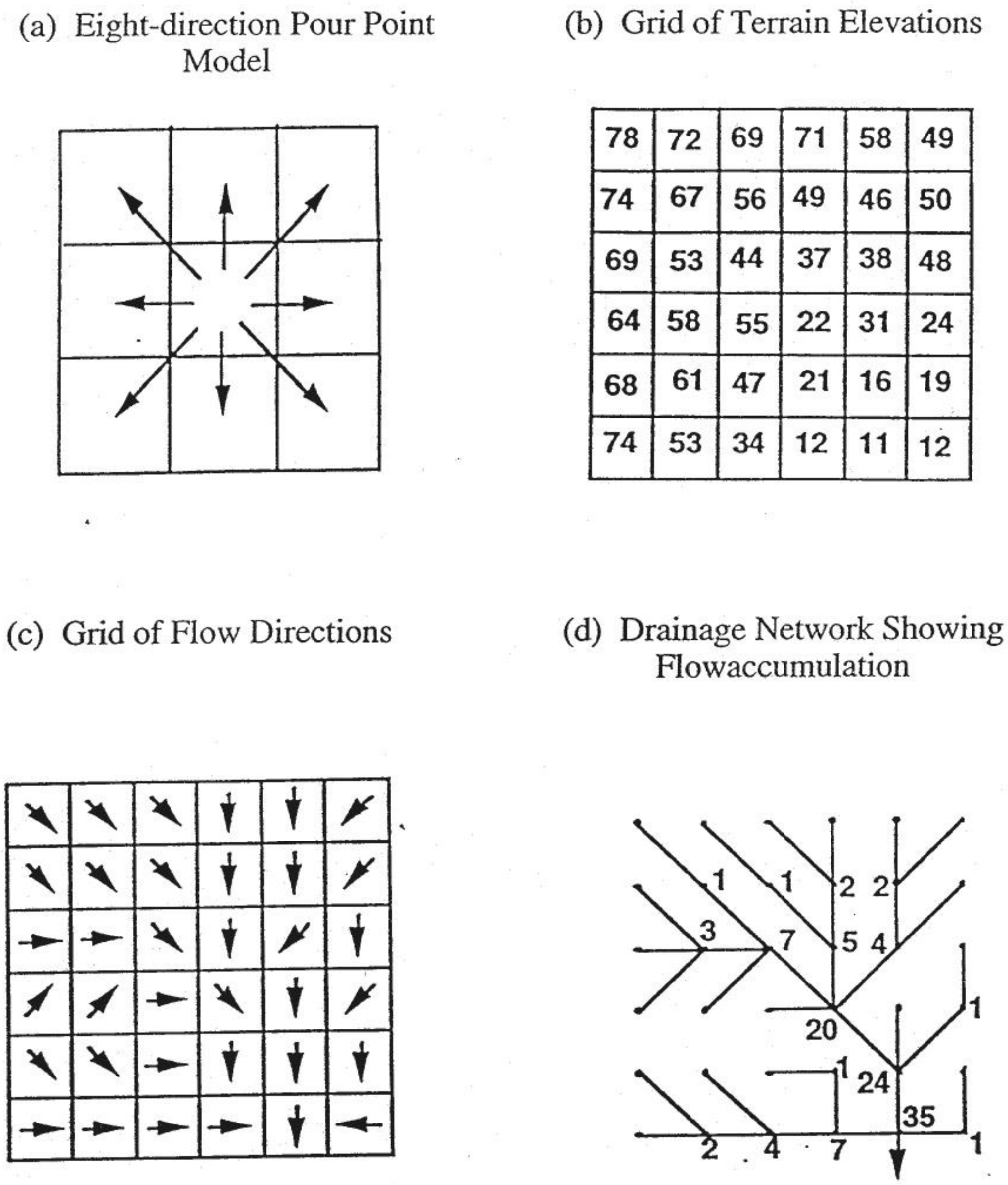
Arc/Info Grid functions can be used to delineate stream networks and watersheds. The first two steps in the watershed delineation process involve assigning a “flow direction” and “flow accumulation” to each grid cell. The Flowdirection function assigns a unique value to a cell indicating to which of its eight neighboring cells it will flow — assuming that water will only flow in one of eight possible directions. Figure 2.4a depicts this eight direction pour point model used to define flow direction. The Flowdirection function takes a grid of terrain elevations as input and produces a grid of flow directions as output as shown in Figure 2.4b and c . By assuming a flow direction to and from each cell, a drainage network has been developed (Figure 2.4d). The number of cells that are upstream of any given cell in a drainage network is termed the Flowaccumulation. The Flowaccumulation function takes the Flowdirection grid as input and stores in each output cell the number of cells that flow into that cell. Streams or major drainage paths can be defined as cells with high Flowaccumulation and a threshold Flowaccumulation value can be used to delineate a stream network. The density of the stream network will depend on the chosen threshold (Maidment and Mizgalewicz, 1993). The syntax for Flowdirection, Flowaccumulation, and stream delineation are:

```
Grid: tkfd = flowdirection(tkalbgf)
Grid: tkfa = flowaccumulation(tkfd)
Grid: tkst = con (tkfa > 10000, 1)
```

In this example the threshold chosen to define “streams” was 10,000, a somewhat arbitrary choice. In words, the third command can be translated as — define a grid named tkst and if the corresponding cell in tkfa has a value greater than 10,000 then assign a value of 1 to the cell in tkst; otherwise, assign the cell NODATA.

2.2.1 Creating a Point Coverage of Watershed Outlets

The Grid Watershed command traces upstream from selected outlet points in a drainage network and identifies all cells flowing to specified outlets. The Watershed command



(a) Eight-direction Pour Point Model

(b) Grid of Terrain Elevations

78	72	69	71	58	49
74	67	56	49	46	50
69	53	44	37	38	48
64	58	55	22	31	24
68	61	47	21	16	19
74	53	34	12	11	12

(c) Grid of Flow Directions

↘	↘	↘	↓	↓	↙
↘	↘	↘	↓	↓	↙
→	→	↘	↓	↙	↓
↗	↗	→	↘	↓	↙
↘	↘	→	↓	↓	↓
→	→	→	→	↓	←

(d) Drainage Network Showing Flowaccumulation

Figure 2.4: Creating a Drainage Network

takes a grid of Flowdirection and a grid of outlet cells as input and produces a grid in which each cell contains the value of the outlet cell to which it flows. Therefore, each cell in the grid of outlet cells must contain a unique value. Coordinates of flow measurement locations provided by the Hydrologic Engineering Center were used to select outlet cells for the division of Tenkiller into subwatersheds. Table 2.2 lists these flow measurement locations.

Table 2.2: Flow Measurement Locations

Location ID	Name	Latitude	Longitude
1	Illinois R. Near Watts, OK	36.1300	-94.5700
2	Illinois R. Near Tahlequah, OK	35.9214	-94.9208
3	Baron F. at Eldon, OK	35.9211	-94.8383
4	Tenkiller Ferry Reservoir	35.5967	-95.0492

As a first step, a point coverage of these outlets was generated.

```
Arc: generate tkout
Generate: input tkout.gen
Generate: points
Generate: quit
```

The ASCII file used as input (tkout.gen) is listed below. The file contains ID#, longitude, and latitude.

```
1,-94.5700,36.1300
2,-94.9208,35.9214
3,-94.8383,35.9211
4,-95.0492,35.5967
end
```

The point coverage (tkout) was then projected from geographic coordinates into the common Albers projection described in Table 1.2. The datum for the outlet point coordinates was not provided, although they were almost certainly taken from a map with an ellipsoidal rather than a spherical datum; the WGS 72 datum was assumed. As discussed in Section1.4**,** the magnitude of the errors associated with this assumption did not affect further analysis. The point coverage was projected using:

```
Arc: project cover tkout tkoutalb albdd.prj
Arc: build tkoutalb point
```

The projection file albdd.prj is the same as the projection file geoalb.prj listed in Section 2.1.1.3 except the units on the input file are decimal degrees (DD) instead of decimal seconds (DS).

2.2.2 Selecting Grid Cells for Watershed Outlets

In order to delineate watersheds, the outlet points need to be grid cells that lie on the stream network (tkst). Because the outlets in the point coverage tkoutalb do not fall precisely on the stream network, a simple conversion of the point coverage tkoutalb to a grid was not sufficient. An interactive procedure was used to select outlet cells that do lie on the stream network. Also, the values in different outlet cells needed to be unique; therefore, the values for outlet cells were selected from a grid of stream "links," not the grid tkst in which all cells contain the value 1. Given a gridded stream network and a Flowdirection grid, the Streamlink function produces an output grid of the stream network such that each cell in a given stream reach (or link) contains a unique value.

```
Grid:  tklink = streamlink (tkst,tkfd)
```

Before stream outlet points could be selected, the stream network and outlet points were displayed on the screen.

```
Grid: display 9999          {If no display}
Grid: mape tklink
Grid: gridpaint tklink      {paint streams}
Grid: points tkoutalb       {display outlet points}
```

The mapextent was reduced to the smallest window that contained all outlet points.

```
Grid: mape *
Define the box
Grid: clear
Grid: gridpaint tklink              {refresh streams}
Grid: points tkoutalb               {refresh outlet points}
```

In order to zoom in on multiple outlet cells at once, at a high enough resolution to resolve individual cells, the PAN/ZOOM feature of the Arc/Info graphics window was used to create four new windows. An alternative to manually creating each new window is to execute an AML that automatically displays the vicinity of outlet cells. The code for an AML called make_win.aml that automatically creates new windows in the vicinity of outlets is presented in the Appendix. After the new graphics windows were created, the Gridpaint and Points display commands were repeated again to redraw the same features in the new windows.

Grid: **gridpaint tklink**

Grid: **points tkoutalb**

The Selectpoint function allows choice of outlet points. With the * option, cells can be selected with the mouse.

Grid: **tkallout = selectpoint (tklink,*)**

Define the points <9 to END>

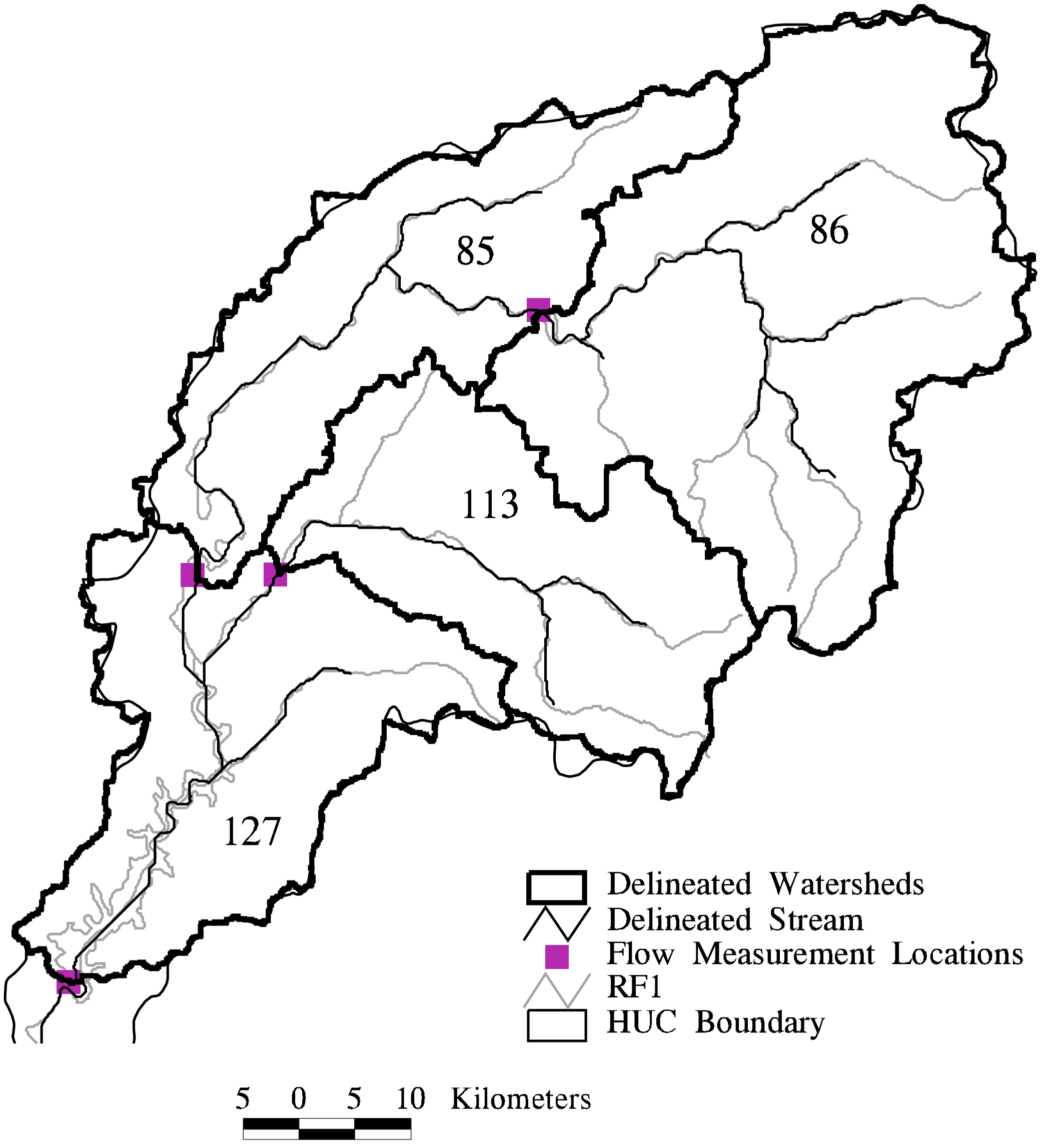
A new grid (tkallout) with four cells was created. With a large number of outlets, it might be too difficult to define all the outlet cells in one application of the Selectpoint function. In that case, several outlet grids created with repeated application of the Selectpoint function may be combined using the Merge function. For Tenkiller, the new grid tkallout contains four cells selected from the stream network that were closest to the outlet points in tkoutalb; all other cells in tkallout contain NODATA. The values of outlet cells in tkallout correspond to values in the streamlink grid tklink — these values are 85, 86, 113, and 127. The downstream end of the links with these values correspond to the flow measurement locations 1, 2, 3, and 4 in Table 2.2. If tkst were used as the input grid to Selectpoint, all the outlet cells would have the same value (1) and the Watershed function would delineate only one watershed.

2.2.3 Delineating Watersheds

With the grid of outlet points properly defined, subwatersheds were delineated.

Grid: **tksubsheds = watershed (tkfd,tkallout)**

Each of the cells in tksubsheds has one of four values, inherited from the cell values in tkallout, identifying the watershed in which that particular cell is located. The values for



Projection: Albers
Datum: RF1 - NAD27
Delineated Streams and Watershed Boundaries - WGS72

Figure 2.5: Delineated Tenkiller Streams and Watersheds with RF1 Streams and HUC Boundary

each of the Tenkiller watersheds are listed in the value attribute table (VAT) below. One record is listed for each value in the subwatershed grid. The count column lists the number of cells in the grid that contain a particular value; the watershed area (km^2) is equal to count times 0.01 since each DEM cell is 0.01 km^2 in area.

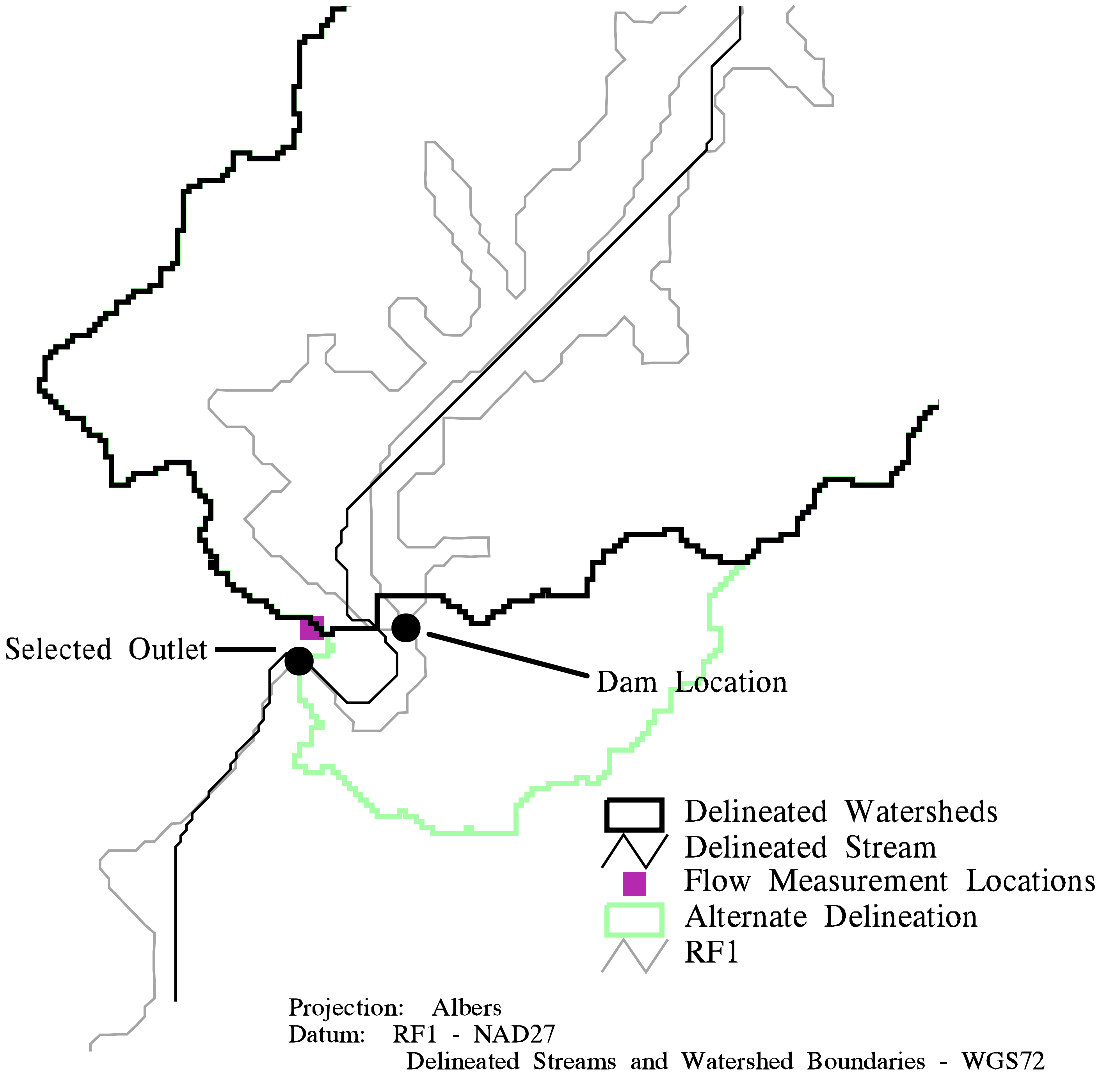
```
Arc: list tksubsheds.vat
Record     VALUE    COUNT
   1         85      84515
   2         86      155692
   3         113       82495
     4         127   93628
```

Figure 2.5 shows delineated Tenkiller streams and watersheds. HUC boundaries and RF1 streams are also included for comparison. Once again, the values 85, 86, 113, and 127 used to label subwatersheds correspond to numbers selected from the Streamlink grid tklink.

Selecting outlet cells in the stream network that are closest to the gaging stations in tkalbout could be automated; however, some degree of engineering judgement may be required. For example, at a stream confluence the user may want to delineate one branch of the stream while the computer might identify a stream cell that is downstream of both branches as the closest to an outlet point, resulting in an incorrect delineation. An example of the need for user judgement or the need for additional information arose in the Tenkiller delineation. The position of the furthest downstream station (Tenkiller Dam) occurs just upstream of a large bend in the delineated stream. This station, along with the stream network delineated from the DEM and streams from EPA's RF1 files is shown in Figure2.6. The RF1 file shows that the outlet should be selected above the stream bend — information that could not be obtained from the DEM alone. The original delineation for Tenkiller was made without this knowledge and is indicated by the lighter boundary in Figure 2.6 labeled "alternate delineation." The difference in area between the two delineations is 25 km^2 or 0.6% of the total watershed area.

2.2.4 Creating a Vector Coverage of Watersheds and Streams

In this procedure, it is necessary to make a raster to vector conversion on the watershed grid so that an intersection can be made with a coverage of rainfall cells in a



Projection: Albers
Datum: RF1 - NAD27
Delineated Streams and Watershed Boundaries - WGS72

Figure 2.6: Close-up of Dam Location

later step. Raster to vector conversion is also useful for display purposes. Two functions used to convert raster data into vector data are Gridline and Gridpoly. A newer function called Streamline is also available.

Grid: **tkstc = gridline (tkst)**

Grid**: tksubshedsc = gridpoly (tksubsheds,0.0)**

The VALUE item from tksubsheds.vat gets stored as item GRID-CODE in the polygon attribute table of tksubshedsc (tksubshedsc.pat). The second argument to the Gridpoly function specifies a weed tolerance of zero. Weed tolerance is the minimum distance between arc vertices in map units. Choice of weed tolerance is rather subjective. When the weed tolerance is less than the size of one cell, cell boundaries are precisely maintained. With the weed tolerance chosen, there is a potential for small errors in converting cells to polygons if two or more cells identified as belonging to the same watershed do not share a cell-side. An example of this error showed up during the raster to vector conversion at Tenkiller. Figure2.7 shows a close-up of three cells in the grid tksubsheds that drain to the gray watershed (113) but are surrounded on all sides by cells of another watershed (85).

During raster to vector conversion with Gridpoly, these three cells formed their own polygon. As a result, tksubshedsc.pat had one more row entry than expected. Two rows contain the GRID-CODE 113 as shown below.

Arc: **list tksubshedsc.pat**

Record	AREA	PERIMETER	TKSUBSHEDS2C#	TKSUBSHEDS2C-ID	GRID-CODE
1	-4163299584.0	520200.000	1	0	-9999
2	1556919808.0	297599.96	2	1	86
3	845149888.0	260599.984	3	2	85
4	824919808.0	216999.969	4	3	113
5	30000.0	800.000	5	4	113
6	936280000.0	244200.000	6	5	127

In this listing, area has units of meters squared and perimeter has units of meters. The polygon with GRID-CODE = -9999 is the “universe” polygon or the polygon outside of all watersheds. An Avenue code is being developed to fix the problem of small dangling polygons illustrated in Figure2.7. For now, the small replicate polygon (113) does not impair further execution or affect the final results of this analysis.

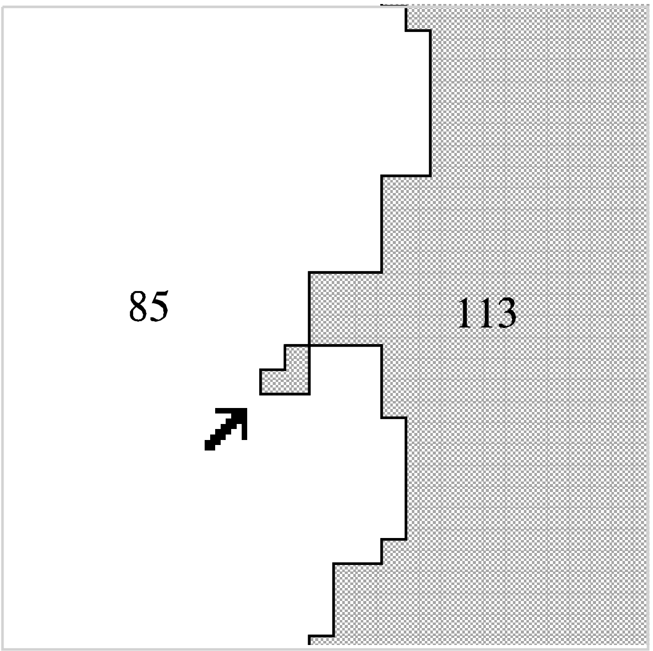


Figure 2.7: A Few Misplaced Cells During Raster to Vector Conversion

The script printed in the box below summarizes the steps described in this section.

Table 2.3 Summary Script for Stream and Watershed Delineation

```
/* Generate a point coverage of flow measurement locations.
/* This is done before all grid processing so that it is not
/* necessary to enter, exit, and re-enter the Grid subprogram.
generate tkout    /* Start Generate subprogram.
input tkout.gen    /* Requires input file of flow measurement locations.
points
quit    /* End Generate subprogram.
project cover tkout tkoutalb albdd.prj
build tkoutalb point
grid    /* Start Grid subprogram.
tkfd = flowdirection(tkalbgf)
tkfa = flowaccumulation(tkfd)
tkst = con (tkfa > 10000, 1)    /* Identify streams.
tklink = streamlink (tkst,tkfd)
&if [extract 1 [show display]] ne 9999 &then    /* If there is not display window
  display 9999    /* then bring one up.
mape tklink
gridpaint tklink
points tkoutalb
mape *    /* Zoom in on an area that contains all four outlets.
clear
gridpaint tklink    /* Paint streams in white.
points tkoutalb    /* Display outlet points.
/* Use the PAN/ZOOM menu to manually create a closeup window for each outlet
/*or type &r make_win tklink tkoutalb
gridpaint tklink
points tkoutalb
tkallout = selectpoint (tklink,*)    /* Select outlet cells.
/* Define the points <9 to END>
tksubsheds = watershed (tkfd,tkallout)
/* Create Arc coverages from grids for future use.
tkstc = gridline (tkst)
tksubshedsc = gridpoly (tksubsheds,0.0)
q   /* Exit Grid subprogram.
&return
```

2.3 PROCESS DEM FOR TRAVEL LENGTH OR TRAVEL TIME PARAMETER

To facilitate distributed modeling with a NEXRAD rainfall cell as the basic modeling unit, a measure of the flow distance or flow time from a rainfall cell to the point at which the flow is to be predicted (most often a watershed outlet) is needed. The development of a drainage network described in Section 2.2 can easily be extended to the computation of travel length from any cell in a DEM grid to a given outlet cell. Arc/Info offers the grid function Flowlength which computes, for each cell in a watershed, the length along the drainage network to either the lowest point in that watershed (downstream option) or to the highest point in that watershed (upstream option). The Flowlength function takes a grid of Flowdirection as input and also provides options for specifying a weight grid. Flowlength computations account for the fact that although all cells in the grid data model are of equal size, three different flow path lengths are possible through a given cell depending on which of eight possible directions the flow enters and leaves the cell. Although a cell-based drainage network clearly does not represent the true flow path that a water droplet falling on the land surface follows to the outlet, it provides a good conceptual model for computing relative travel lengths.

In addition to travel length, the Flowlength option to specify a weight grid opens up further possibilities for estimating a travel time or at least a travel time index for each cell. Maidment *et al.* (1995) proposed assigning a velocity to each cell and computing travel time to the outlet using (1/cell velocity) as the weight grid. Further work by Maidment *et al.* suggested that a model, conceptually similar to Manning's channel flow equation, would be suitable for assigning cell velocities.

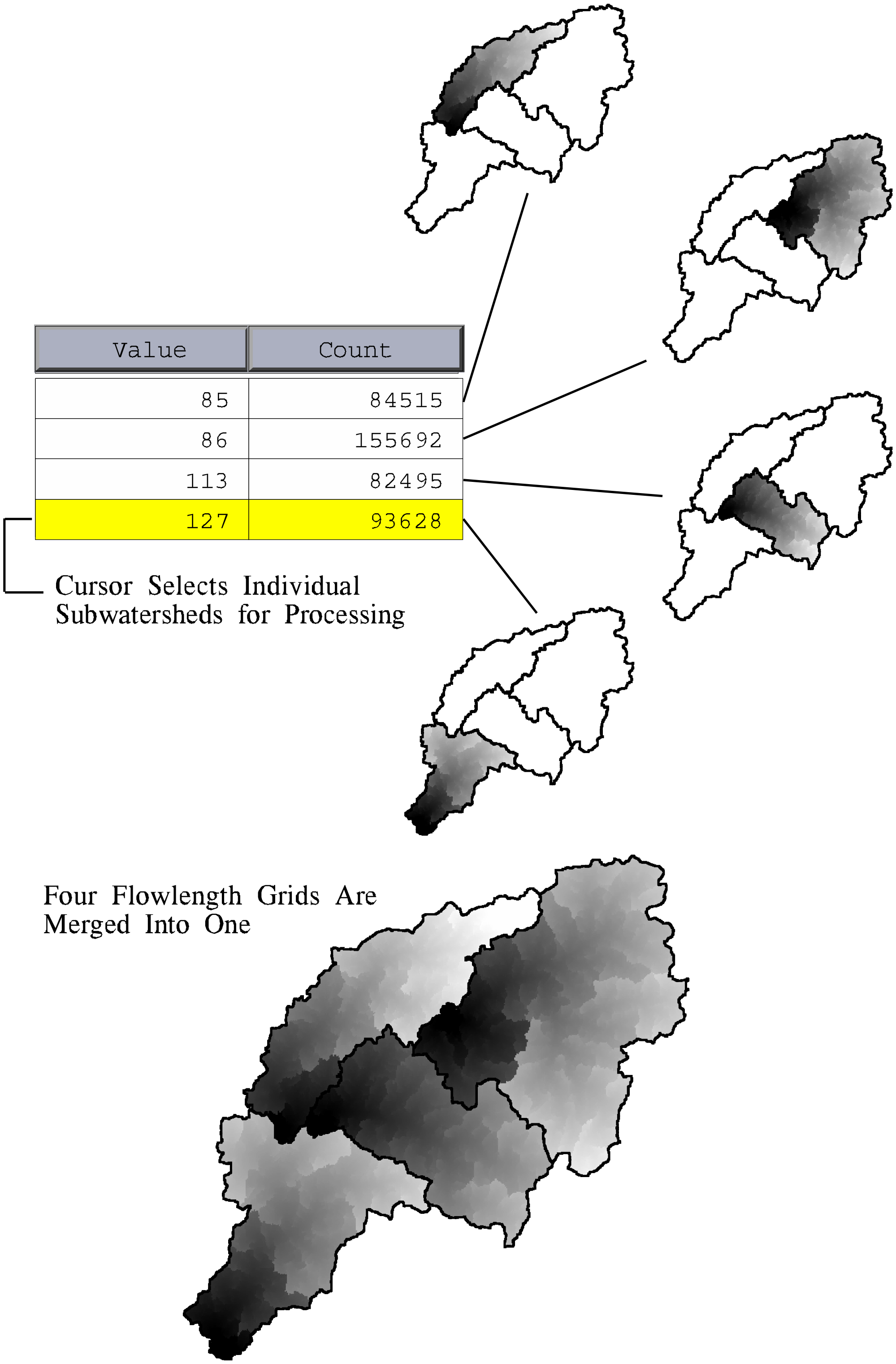
$$v = v_{avg} * (S^b A^c) / (S^b A^c)_{avg} \qquad (2.1)$$

S is local slope at a cell; A is upstream drainage area to a cell; b and c are coefficients determined by calibration (values b = c = 0.5 proved to be appropriate); and the subscript avg denotes an arithmetic average over the cells in a particular watershed. It follows from Equation 2.1 that by computing $1/S^bA^c$ for each cell and using this as a weight grid, the Flowlength function would yield a grid of travel time index values. At this point, the physical significance of a time index parameter has not been fully explored. In its draft version, the modClark formulation only requires a travel length measure for each rainfall cell as input; however, the procedure to compute a time index grid has been included in the codes in the Appendix to this report.

Flow_length.aml, presented in the Appendix, computes the travel length from each cell in a subwatershed to the outlet of that subwatershed given an input grid defining subwatersheds and the Flowdirection grid for the same area. A filled DEM is an optional input, required if a time index parameter is to be computed. The names of required input grids are passed to flowlength.aml at the command line. The command syntax used to run flowlength.aml for the Tenkiller watersheds was as follows:

```
Arc: &r <aml_name> <subshed_grid> <dir_grid> {filled_dem}
Arc: &r flow_length tksubsheds tkfd tkalbgf
```

The main loop in flow_lengtha.aml that controls which cells in the Flowdirection grid are currently being processed uses a programing concept called a Cursor to access individual records in the value attribute table (VAT) of the subwatershed grid tksubsheds.vat — a copy of this table was printed in Section 2.2.3. A Cursor is a mechanism in INFO for accessing a selected set of records in a table. As the Cursor points to each subwatershed (record) in tksubsheds.vat, a conditional statement is tested — if a cell is in the current subwatershed, its Flowdirection is passed to the Flowlength function. Temporary grids are created that contain the Flowlength values for each subwatershed. The names of the temporary grids are stored as character strings; these strings are later concatenated to serve as an argument to the Grid function Merge. The end result is flmerge_grid, a grid of flowlengths for all subwatersheds. Flow_length.aml also graphically displays the results of the Flowlength computations as each subwatershed is processed. The structure of flow_length.aml was modeled after an AML written by Beavers (1993). This structure is useful for automating any Grid functions that need to be performed on individual zones or watersheds. Flowlength.aml calls a "canned" program and its associated menu described by Beavers (1993), msworking2.aml and msworking2.menu; these programs only supply messages to the user and do not affect grid processing. In addition, slope.aml and time_weight.aml may be called in the main loop of flow_length.aml to compute a time index parameter for each cell. Since a time index was not computed for this report, the relevant lines in flow_length.aml have been commented out. Figure 2.8 illustrates several steps in the creation of flmerge_grid and shows a gray-shaded image of the final product.



Value	Count
85	84515
86	155692
113	82495
127	93628

Cursor Selects Individual Subwatersheds for Processing

Four Flowlength Grids Are Merged Into One

Figure 2.8: Computing Flowlength for Each Sub-basin

3.0 POSITIONING THE NEXRAD STAGEIII CELLS RELATIVE TO THE DIGITAL ELEVATION MODEL CELLS

In order to define the flow length from each NEXRAD precipitation cell to the watershed outlets, a mesh of NEXRAD cells must be created in the same coordinate system as the digital elevation data. This apparently simple task is complicated by the fact that the original NEXRAD mesh is defined on a spherical earth datum but the locations of all the radars are recorded in latitudes and longitudes on an ellipsoidal earth datum. The distinction between these two datums is sufficiently important to require a careful examination of their definitions and how to make transformations between them.

3.1 GEODETIC (ELLIPSOIDAL) AND GEOCENTRIC (SPHERICAL) COORDINATES

In a spherical coordinate system, the latitude of a point on the earth's surface is the angle between the equatorial plane and a line drawn from a point on the earth's surface to the center of the earth, and the longitude is the angle a plane drawn through a particular meridian makes with a similar plane drawn through the Greenwich meridian. These are called geocentric coordinates. But the earth is an oblate spheroid rather than a sphere, being flattened slightly at the poles compared to the equator such that the polar radius (6357 km) is about 1/300 shorter than the equatorial radius (6378 km). If the earth is represented as a sphere of constant radius, such as 6371 km, the "earth surface" or equivalent of mean sea level is about 7 km below the ocean surface at the equator, and about 14 km above the ocean surface in the polar regions. To avoid this discrepancy, geographers represent the earth mathematically by an ellipsoid of rotation with the major axis of the ellipse in the equatorial plane and the minor axis between the poles. This change does not alter longitudes but it means that latitudes normally used for mapping are not defined by a line passing through the center of the earth.

Snyder (1987, p. 13) gives the following definition: "the geographic or geodetic latitude, which is normally the latitude referred to for a point on the Earth, is the angle which a line perpendicular to the surface of the ellipsoid at the given point makes with the plane of the equator. It is slightly greater than the geocentric latitude, except at the equator and the poles, where it is equal. The geocentric latitude is the angle made by a line to the center of the ellipsoid with the equatorial plane." For a sphere, the geodetic and geocentric latitudes are identical because a line normal to the surface always passes through the center but for an ellipsoid they are different.

Figure 3.1 is a conceptual diagram illustrating geodetic (ϕ) and geocentric latitude (ϕ_g) of a point B on the earth's surface where the center of the earth is at point O.

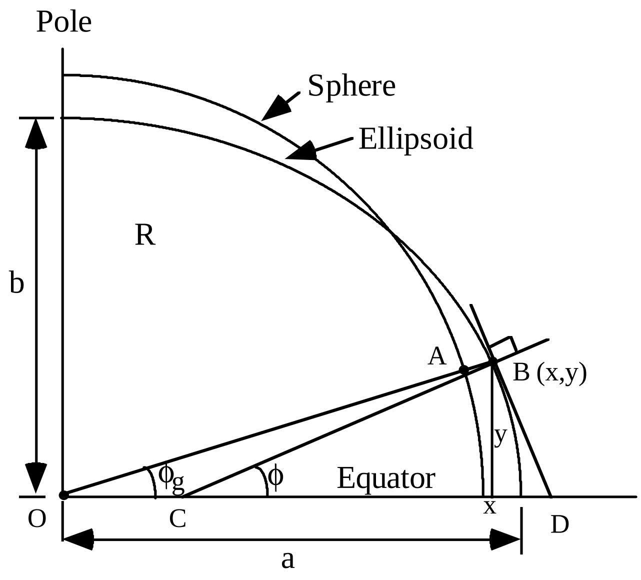


Figure 3.1: Geocentric and Geodetic Latitud

Geometrically, the geocentric latitude of a point B on the earth's surface is given by ϕ_g = angle BOD, and the geodetic latitude by ϕ = angle BCD, where the line BC is normal to the surface at the point B and the line BD is tangent to the ellipsoid at point B. An equivalent point with geocentric latitude ϕ_g is located at point A on a sphere of radius R. It can be seen that the shape of the curve of the ellipse produces a geodetic latitude ϕ greater than the geocentric latitude ϕ_g. The relation between these two latitudes can be derived in the following manner. The equation for an ellipse is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \qquad (3.1)$$

where the coordinates (x,y) of any point B are measured from the center O. By differentiating Eq. (3.1), the gradient of the surface at B is given by

$$\frac{2x}{a^2} + \frac{2y}{b^2}\frac{dy}{dx} = 0$$

or $$\frac{dy}{dx} = -\frac{b^2}{a^2}\frac{x}{y} \qquad (3.2)$$

which is equal to the slope of the tangent line BD. The slope of the line CB which is normal to the surface and perpendicular to BD is given by the negative inverse of the slope of BD, or

$$\tan \quad = \frac{a^2}{b^2}\frac{y}{x} \qquad (3.3)$$

The line OAB has slope y/x, so the geocentric latitude satisfies the relation

$$\tan \quad {}_g = \frac{y}{x} \qquad (3.4)$$

Thus the geodetic and geocentric latitudes are related by

$$\tan \quad = \frac{a^2}{b^2}\tan \quad {}_g \qquad (3.5)$$

Snyder (1987, p. 17) presents an equivalent equation to Eq. (3.5) in the form

$${}_g = \arctan[(1-e^2)\tan \quad] \qquad (3.6)$$

where the eccentricity e is given by

$$e^2 = 1 - \frac{b^2}{a^2} \qquad (3.7)$$

It can be verified that Eqs (3.5) and (3.6) are equivalent.

3.1.1 Conversions Between Geodetic and Geocentric Latitudes

As discussed in Section 1.5, three ellipsoids used for mapping in the United States are GRS 80, Clarke 1866, and WGS 72. The values of the major and minor axis lengths and the corresponding eccentricities of these three ellipsoids are as follows (Snyder, 1987, p.12-13)

GRS 80: a = 6378.137 km, b = 6356.7523 km, e^2 = 0.00669438

Clarke 1866: a = 6378.2064 km, b = 6356.5838 km, e^2 = 0.006768658

WGS 72: a = 6378.135 km, b = 6356.7505 km, e^2 = 0.00669432

By substituting values for a and b for GRS 80 into Eq. (3.5), the following relations are found between GRS 80 geodetic latitude, ϕ , and geocentric latitude, ϕ_g:

$$= \tan^{-1}(1.00673950 \tan \quad {}_g) \qquad (3.8)$$

$$_g = \tan^{-1}(0.99330562 \tan \quad) \qquad (3.9)$$

The simplest connection between the sphere and an ellipsoid is to use either the GRS 80 ellipsoid or the WGS 72 ellipsoid because the NAD 83, WGS 84, and WGS 72 datums use their respective ellipsoids in a geocentered position. In NAD 27, the Clarke ellipsoid is offset from a geocentric position so that it more nearly fits the earth's surface in North America. Applying Eq. (3.8) to compute ϕ from ϕ_g, Table 3.1 is derived,

Table 3.1: Differences Between Geodetic and Geocentric Latitudes for the GRS 80 Ellipsoid

Geocentric latitude, ϕ_g	**Geodetic latitude, ϕ**	**ϕ -ϕ_g**
0°	0°	0
10°	10° 3' 58"	3' 58"
20°	20° 7' 26"	7' 26"
30°	30° 10' 01"	10' 01"
40°	40° 11' 23"	11' 23"
50°	50° 11' 21"	11' 21"
60°	60° 9' 59"	9' 59"
70°	70° 7' 24"	7' 24"
80°	80° 3' 56"	3' 56"
90°	90°	0

from which it can be seen that the difference between the two values in the mid-latitudes is approximately 10', or (10/60)*(π/180) = 0.00291 radians. For a constant earth radius of R = 6371.2 km, this corresponds to 6371.2 * 0.00291 = 18.5 km or approximately 12 miles shift on the earth's surface. The shift is always in the same direction — a point on a spherical geocentric grid having 30° geocentric latitude corresponds to a geodetic latitude of 30° 10' 01" so the "map" location of this mesh point is actually about 10' further North than its geocentric latitude would indicate if the geocentric latitude is interpreted as a map latitude. Likewise, a measurement station at 30°N geodetic latitude corresponds to a location on a spherical grid of approximately 29°50' N.

Snyder (1987, p. 18) presents a similar latitude comparison table for the Clarke (1866) ellipsoid. Comparing values of ϕ_g - ϕ in Snyder's table with the above values shows that the two sets of values are within 9" (0.15') of one another in all cases. Thus, it appears, that differences in the geodetic latitudes of a point on the earth as a function of the eccentricity of the ellipse are considerably smaller than is the difference between the geocentric and geodetic latitudes of that point. However, the offset from an earth-centered position of the Clarke (1866) ellipsoid in NAD 27 will also have some effect on NAD 27 latitude and longitude values which is not accounted for by the calculations presented here. Experiments with Arc/Info datum conversions discussed in Section 1.5 show that differences in projected coordinates among ellipsoidal datums are on the order of tens of meters in the Tenkiller area where 1” is about 31 m.

3.2 THE HRAP COORDINATE SYSTEM

The Hydrologic Rainfall Analysis Project (HRAP) grid as defined by Greene and Hudlow (1982) is used to define the location of each average precipitation value in a NEXRAD StageIII data set. The HRAP cell coordinates are defined in the image plane of a secant polar Stereographic map projection on a spherical, earth-centered datum of radius 6371.2 km. The secant polar Stereographic projection has a standard (true) latitude of 60° North and a standard longitude (longitude of the projection center) of 105° West. In other words, features on the earth’s surface are projected onto a plane perpendicular to the axis of rotation and passing through the earth at $\phi_g = 60°$. This map projection is used by the Air Force Global Weather Central (AFGWC) and also by the National Meteorological Center (NMC) to define several reference grids (Hoke *et al.*, 1981). The Air Force Global Weather Central defines a whole-mesh reference grid in this map projection and several finer-resolution grids relative to the whole mesh grid. By definition, the whole-mesh grid length is 381 km. Finer resolution grids are defined as one-half, one-quarter, one-eighth, and one sixty-fourth of the whole-mesh grid. The half-mesh grid length is 190.5 km (Hoke *et al.*, 1981). The HRAP grid size is equivalent to one-eightieth of the whole-mesh grid, resulting in a grid length of 381/80 = 4.7625 km. However, the standard longitude of the Air Force Global Weather Central reference grids (10° E) is different than the standard longitude for HRAP (105° W) which means that the orientation of HRAP cells relative to map features is different than that of whole-mesh grid cells. In other words, HRAP cells do not fall in alignment with the whole-mesh grid cells. The grid cell lengths cited above are in the projected plane. Lengths in the projected plane are only equivalent to lengths on the surface of the spherical earth datum at 60° N.

3.2.1 Forward Transformation from Geocentric to HRAP Coordinates

The procedure for projecting spherical coordinates in degrees on the earth's surface to Cartesian coordinates in meters on a flat surface can be thought of in two parts. First, convert the geographic coordinates in latitude and longitude (ϕ_g,λ) into equivalent polar coordinates (R,λ) on a flat plane. Second, convert the polar coordinates (R,λ) to Cartesian coordinates (x,y) in the same plane. Finally, a grid mesh is defined in the projected plane by specifying a point of origin (x_o,y_o) and then a mesh size and orientation about that point. The steps are the same when an ellipsoidal datum is used but the conversion from ellipsoidal coordinates to polar coordinates in a plane is more complex because the radius of curvature on an ellipsoidal earth varies continuously with latitude and also varies continuously with direction at any point.

3.2.1.1 From Geographic to Polar Coordinates

The polar Stereographic projection is an azimuthal projection (projection onto a plane). Figure3.2 illustrates the geometry of the polar Stereographic map transformation with a standard latitude of 60° N. A light shining from point C will project point E on the Earth's surface onto the image plane at point B. In the polar Stereographic projection, the point B on the image plane can be defined by two parameters, the distance from the pole R and the longitude (λ) measured from the Greenwich Meridian. An expression for R as a function of latitude (ϕ_g) can be derived from Figure 3.2.

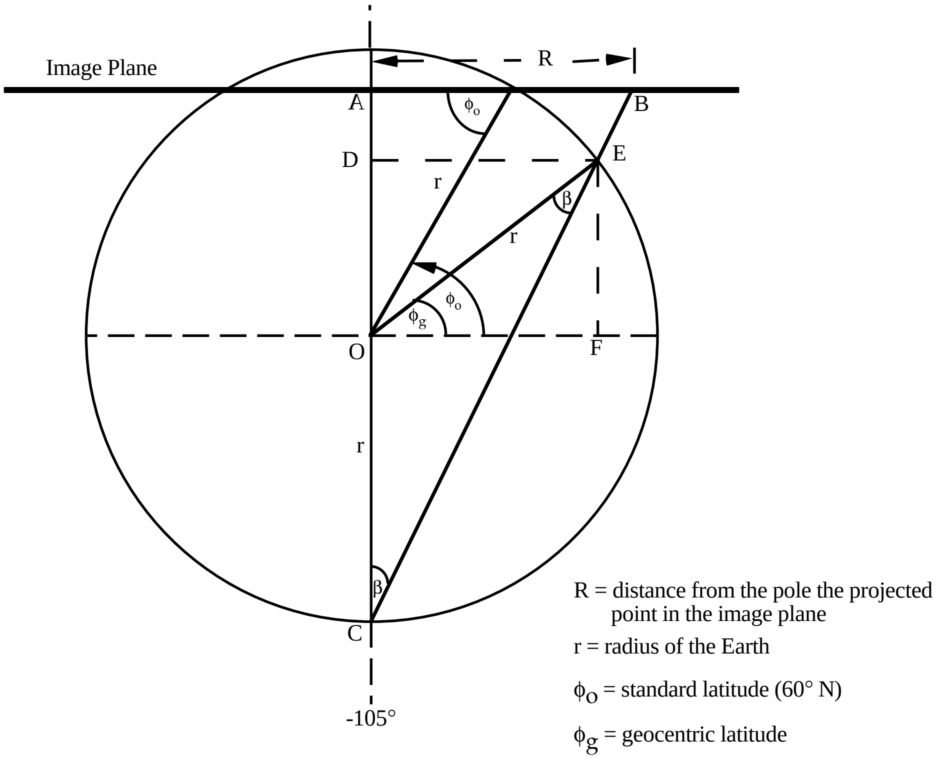
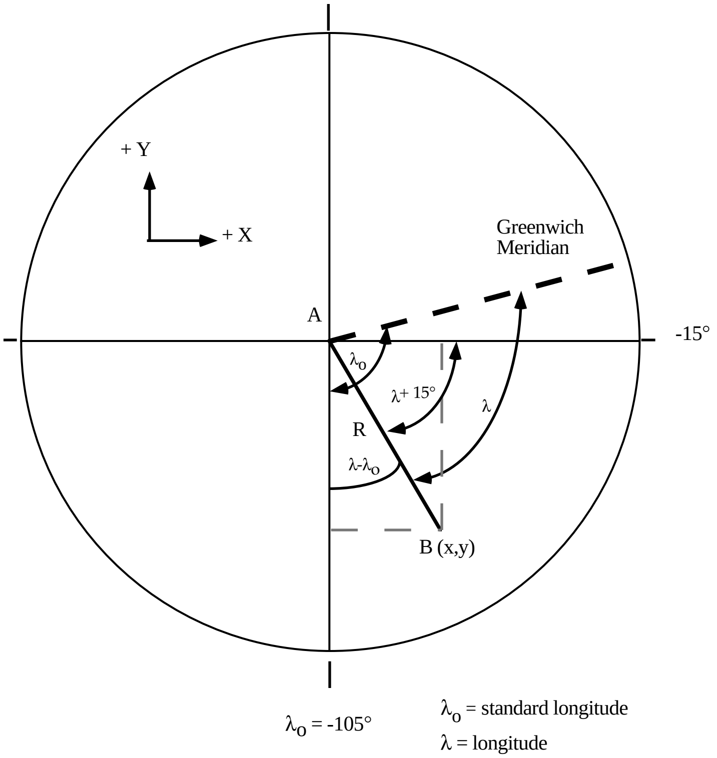


Image Plane
R
A
B
D
E
r
β
r
ϕo
ϕo
ϕg
O
F
r
β
C
-105°
R = distance from the pole the projected point in the image plane
r = radius of the Earth
ϕO = standard latitude (60° N)
ϕg = geocentric latitude

(Adapted from Greene and Hudlow, Figure 1)

Figure 3.2: Elevation View of a Polar Stereographic Map Projection



(Adapted from Green and Hudlow, Figure 2)

Figure 3.3: Plan View of a Polar Stereographic Map Projection

$$\phi \quad = \frac{}{} = \frac{}{} \qquad (3.10)$$

$$\sin\phi_g = \frac{EF}{r} = \frac{DO}{r} \qquad (3.11)$$

$$\sin\phi_o = \frac{AO}{r} \qquad (3.12)$$

$$CD = DO + r = r\sin\phi_g + r \qquad (3.13)$$

$$CA = AO + r = r\sin\phi_o + r \qquad (3.14)$$

Noting that triangles CDE and CAB are similar triangles,

$$\frac{CD}{CA} = \frac{DE}{AB} \qquad (3.15)$$

Further manipulation and substitutions from Eqns. (3.10) - (3.15) yields:

$$AB = \frac{DE * CA}{CD} = r\cos\phi_g \frac{r(1+\sin\phi_o)}{r(1+\sin\phi_g)} \qquad (3.16)$$

$$R = AB = r\cos\phi_g \frac{(1+\sin\phi_o)}{(1+\sin\phi_g)} \qquad (3.17)$$

3.2.1.2 From Polar to Cartesian Coordinates

The point B is located in the projected plane as shown in Figure 3.3 which is from the vantage point of an observer at point C of Figure 3.2 (the South Pole). Meridians of constant longitude (λ) on the globe become radial lines in the projected plane. The United States lies between longitudes 65° W and 125° W or ($-65 < \lambda < -125$) and a central meridian of 105° W or λ_o = -105° was chosen as the vertical axis (standard latitude) in the HRAP plane, as shown in Figure 3.3. This meridian lies approximately at the west longitude of Denver, Colorado. Thus, points east of Denver have positive x-values while those west of Denver have negative x-values.

The displacement, $\lambda - \lambda_o$, of a projected point from the central meridian is further modified by subtracting 90° to rotate the coordinate system and the Cartesian coordinates in the projected domain are computed with:

$$
\begin{aligned}
&x = R^* \cos(\lambda - \lambda\, 90^o) \\
&y = R^* \sin(\lambda - \lambda\, 90^o) \\
&\text{or} \\
&x = R^* \cos(\lambda + 15^o) \\
&y = R^* \sin(\lambda + 15^o) \qquad (3.18)
\end{aligned}
$$

For these equations, values of λ west of Greenwich are negative and values of λ east of Greenwich are positive.

3.2.1.3 From Cartesian to HRAP Coordinates

A mesh of HRAP coordinates is then defined in terms of the Cartesian coordinates as follows. Since HRAP is defined in grid units and not kilometers, the radius of the model earth (r) is scaled by the mesh length (4.7625 km).

$$a = \frac{r}{\text{meshlength}} = \text{scaled radius} \sim 1{,}337.78 \text{ units} \qquad (3.19)$$

In order to keep all the HRAP coordinate values within the United States positive, the North Pole is assigned coordinates (401,1601). Taking into account the scaled radius and the origin shift, HRAP coordinates can be written in terms of polar Stereographic coordinates (with x and y in kilometers).

$$\text{hrapx} = \frac{x}{4.7625} + 401 \qquad (3.20)$$

$$\text{hrapy} = \frac{y}{4.7625} + 1601 \qquad (3.21)$$

3.2.2 Reverse Transformation from HRAP to Geocentric Coordinates

A slightly different geometric derivation is required to go from HRAP to geocentric coordinates. The inverses of Equations (3.20) and (3.21) can be used to determine the polar Stereographic coordinates (x,y) given HRAP coordinates (hrapx,hrapy). To go from polar Stereographic (x,y) to polar coordinates (R,λ), determining R is simple.

$$R = \sqrt{x^2 + y^2} \tag{3.22}$$

Computation of the longitude (λ) requires the use of an inverse tangent function. In FORTRAN 77, the function ATAN2D(y,x) returns the value of the angle in degrees that the line segment passing through the origin and the point (x,y) makes with the x-axis. The range of angles returned by atan2(y,x) is 0° to 180° in the counter-clockwise direction and 0° to -180° in the clockwise direction. Measured in the clockwise direction with a range 0° to 360°, the angle(λ′) between the Greenwich Meridian and a line segment passing through the origin and a point (x,y) can be computed as follows:

If y>0, then

$$\lambda' = 270° - \lambda_o - a\tan 2d(y,x)$$

If y<0, then

$$\lambda' = -90° - \lambda_o - a\tan 2d(y,x) \tag{3.23}$$

(use λ_o = -105°)

To be consistent with the convention that west longitude values range from 0° to -180° and east longitude values range from 0° to 180°,

If λ′ < 180°, then

$$\lambda = (-1) * \lambda'$$

If λ′ > 180°, then

$$\lambda = 360° - \lambda' \tag{3.24}$$

Referring again to Figure 3.2 , an expression for ϕ_g in terms of R can be derived. Since the triangle EOC is isosceles,

$$90° + \phi_g + 2\beta = 180° \tag{3.25}$$

With manipulation,

$$\beta = 45° - \frac{\phi_g}{2} \tag{3.26}$$

From trigonometry and substitution of Equation (3.14),

$$\tan\beta = \frac{AB}{AC} = \frac{R}{r(1+\sin\phi_o)} \tag{3.27}$$

Substituting (3.26) into (3.27) and manipulating yields:

$$\phi_g = 90° - 2\text{atand}\left[\frac{R}{r(1+\sin\phi_o)}\right] \tag{3.28}$$

3.3 USING NEXRAD DATA WITH AN ELLIPSOIDAL DATUM

All discussion of the HRAP coordinate system in previous sections assumes that the earth is a sphere, but when locating radars or other ground control points, the latitudes and longitudes read from a map are in geodetic coordinates referenced to an ellipsoidal datum. It follows that if the HRAP grid is formally defined on a spherical earth datum, a shift to an ellipsoidal datum is needed to correctly register a radar to the rainfall in the HRAP cell that contains other map features. This involves an approximately 10’ shift in latitudes as specified in Table 3.1. With further information described subsequently, we found that this datum shift was apparently not made when the National Weather Service prepared software using the HRAP grid. Hence, a mathematical transformation appropriate for data in spherical coordinates was applied to map locations in ellipsoidal coordinates when radars and other map reference points were entered into NEXRAD map displays. This action introduces a further distortion, beyond that created by the HRAP projection itself, to the size and shape of HRAP cells relative to ellipsoid-based locations. To treat these subjects in sequence, the correct method for carrying out the transformations is now presented. Figure 3.4 is a conceptual diagram of the correct steps to transform coordinates from a sphere-based map projection into an ellipsoid-based map projection. The correct steps are (1) transform HRAP coordinates into latitude/longitude geocentric coordinates, (2) convert geocentric latitudes to geodetic latitudes using a datum shift from sphere to ellipsoid, and (3) perform datum transformation between ellipsoids if necessary and project geodetic coordinates into Albers.

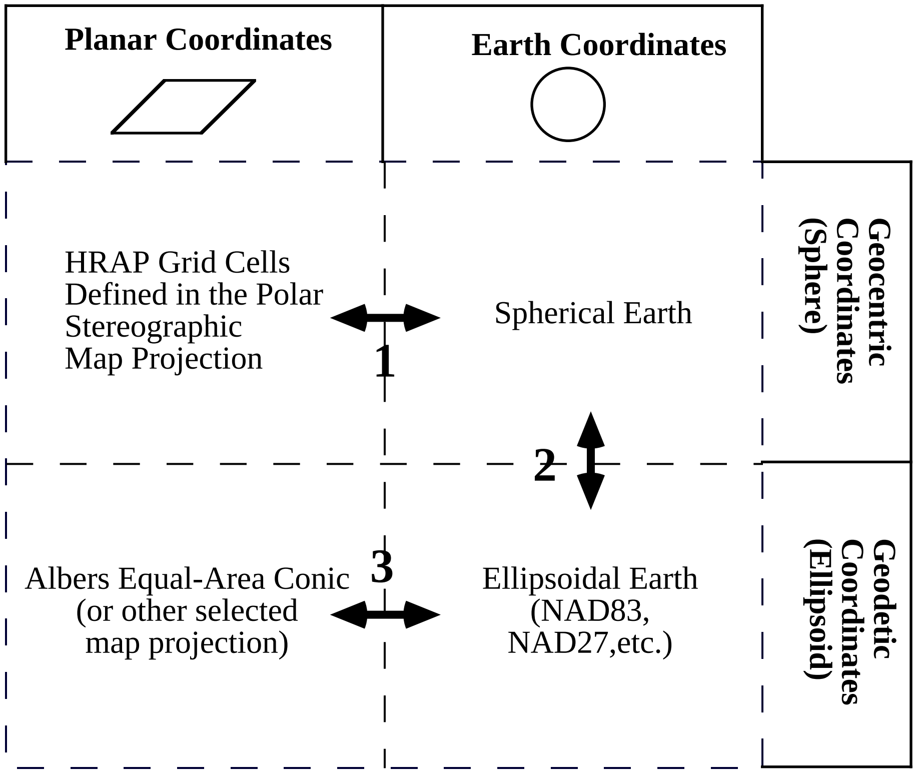

Planar Coordinates
Earth Coordinates
HRAP Grid Cells Defined in the Polar Stereographic Map Projection
1
Spherical Earth
Geocentric Coordinates (Sphere)
2
3
Albers Equal-Area Conic (or other selected map projection)
Ellipsoidal Earth (NAD83, NAD27,etc.)
Geodetic Coordinates (Ellipsoid)

Figure 3.4: Conceptual Diagram of the Correct Steps in the HRAP to Albers Transformation

Evidence provided by Arkansas-Red Basin River Forecast Center and correspondence with the National Weather Service at Silver Spring, Maryland, (Seo and Miller, personal communication, 1995) revealed that the National Weather Service has used geodetically defined coordinates to position radars and other map features (i.e. hydrologic features or political features) in the HRAP plane. The ellipsoidal datum used to define these map features is not known at this time. In other words, geodetic coordinates (ϕ,λ) were treated as geocentric coordinates (ϕ_g,λ) or equations for a spherical earth were used to project geodetically defined features. Therefore, in order to reproduce (as closely as possible) the positions of the HRAP cells relative to the radar sites used by the Arkansas-Red Basin River Forecast Center in radar computations, step 2 in Figure 3.4 should be omitted. That is, the treatment of geodetic latitudes as geocentric latitudes should be emulated in reverse by interpreting geocentric latitudes for HRAP cells as geodetic latitudes.

For convenience in discussion, the transformation involving all steps (1-3) will be referred to as the "true" transformation and the transformation omitting step 2 will be referred to as the "matching" transformation. Using the matching transformation appears to be the most accurate use of the data available at this time. The matching transformation should precisely replicate the relative position of a radar site and the HRAP grid used by Arkansas-Red Basin River Forecast Center at individual radar sites. The situation becomes complicated when information from several radars is merged because the sphere-ellipsoid distortion varies with latitude. If one begins with geodetic coordinates (ϕ,λ) and forward transforms these points to planar coordinates, the values (x',y') so produced are different than the values (x,y) that would have been produced if geocentric coordinates (ϕ_g,λ) were used. Although the reverse transformation applied to (x',y') will recover (ϕ,λ) in geodetic coordinates, there is a systematic difference between (x,y) and (x',y') that involves not only a shift in North-South location but also a rearrangement of the relative locations of points in the domain.

3.4 DISTORTIONS INVOLVED WITH USING THE HRAP COORDINATE SYSTEM

It is difficult for us to assess the magnitude of radar mapping errors without more knowledge about how distances traced out by a NEXRAD radar beam are converted to distances in the HRAP plane. It is also recognized that discussion of errors in this section is strictly limited to distortion due to map transformations without considering other errors, such as beam refraction, associated with radar detection of rainfall — other errors may be considerably larger than map transformation distortions (Seo and Miller, personal communication, 1995).

There are two types of distortions inherent in the mapping of NEXRAD products in the HRAP plane and using a spherical transformation on ellipsoidal coordinates: (1) variation of the scale factor (defined below) with latitude; (2) distortion of the scale factor because of the spherical transformation applied to ellipsoidal coordinates.

3.4.1 Scale Factor

When a map is produced, the dimensions on the earth's surface are first reduced to dimensions on a globe in proportion to the map scale, where

$$\text{map scale} = \frac{\text{globe distance}}{\text{earth distance}} \tag{3.29}$$

For example, a map scale of 1:100,000 means that 1 cm on the globe corresponds to 100,000 or 1 km on the earth. For HRAP, the map scale is 1 unit = 4.7625 km. The area on the earth's surface that an HRAP cell represents varies with latitude. In the HRAP image plane, all HRAP cells are square and have a side length equal to one unit. When HRAP cell coordinates are converted to polar Stereographic coordinates, the cells remain square in the map plane and the map distance of each cell side is 4.7625 km but this is not the length of a cell side measured on the globe. When a portion of the globe is projected onto a flat plane, the distance between two points on the globe is distorted by a scale factor:

$$\text{scale factor} = \frac{\text{map distance}}{\text{globe distance}} \qquad (3.30)$$

The scale factor is defined as the map distance divided by the distance on the globe. Snyder (1987, p. 21) denotes the scale factor along meridians of longitude with "h" and the scale factor along parallels of latitude with "k." The polar Stereographic projection is "conformal," which means that the scale factor does not vary with direction, so h = k. The Albers projection is "equal area" which means that h = 1/k so that area is preserved even though distances are distorted. For HRAP, the scale factor is equal to 1.0 only along the standard latitude (60°). At other latitudes, the scale factor can be computed with:

$$h{=}k = \frac{1 + \sin\phi_o}{1 + \sin\phi_g} \qquad (3.31)$$

which is equal to the ratio AB/DE in Figure 3.2.

At latitudes less than 60° the scale factor is greater than 1.0 and at latitudes greater than 60° the scale factor is less than 1.0. Table 3.2 lists the scale factors at several latitudes. A scale factor greater than 1.0 means that a distance measured on a map is larger than the actual distance on the earth's surface. Thus, one HRAP unit corresponds to 4.7625 km at 60° N and to smaller distances at lower latitudes. The North-South extent of the United States is from approximately 25° N to 49° N, so HRAP cells range in size from 3.6 km in Miami, Florida, to 4.4 km in Minneapolis, Minnesota, and the corresponding range in HRAP cell areas is from approximately 13 km^2 in Miami to 19 km^2 in Minneapolis.

The HRAP cell size is often stated to be 4 km. This statement is accurate for a scale factor of 4.7625/4.00 = 1.1906 which applies at ϕ_g= 34.56° N, approximately the latitude of Little Rock, Arkansas. If a nominal cell area of 16 km^2 is assumed everywhere in the United States, the assumed cell area is 18% larger than the true area in Miami and 19% smaller than the true area in Minneapolis. Significant variation of the scale factor also occurs within the scanning range (230 km) of a WSR-88D radar. For a radar located at 37.05° N latitude, Joplin, Missouri, the scale factor varies from 1.1475 at the northern limit of the radar range to 1.1820 at the southern limit of the radar range. This amounts to a 3.45% variation. The corresponding earth areas of an HRAP cell are 17.22 km^2 at the northern limit and 16.23 km^2 at the southern limit. It seems likely that some accounting for variations in scale factor is included in radar algorithms, but we don't know if that is the case. In general, the polar Stereographic projection is not well suited for local radar mapping due to large variations in the scale factor over short distances.

Table 3.2: Approximate Scale Factor at Different Latitudes

	Representative		HRAP Cell on the Earth's Surface	
Latitude (ϕ_g)	Location	Scale Factor	Side Length (km)	Area (km^2)
25	Miami, FL	1.3117	3.63	13.18
30	Houston, TX	1.2440	3.83	14.66
35	Memphis, TN	1.1858	4.02	16.13
40	Indianapolis, IN	1.1359	4.19	17.58
45	Minneapolis, MN	1.0931	4.36	18.98
50	Winnipeg, Manitoba	1.0566	4.51	20.32

3.4.1.1 The Shape of HRAP Cells

It is clear from this discussion that cell sizes vary with latitude. In addition, each cell represents an area of unique size and shape on the earth's surface. This occurs because the HRAP grid is defined in a projected plane rather than the earth's surface itself. An HRAP cell at 60° N latitude has an approximate globe length and width of 4.7625 km. The cell side is not precisely 4.7625 km in globe length because no side of an HRAP cell coincides exactly with the 60° latitude line. Strictly speaking, all sides of an HRAP cell have slightly different lengths in globe distance. Figure 3.5 illustrates the size and shape an HRAP cell in planar and earth coordinates. This particular HRAP cell is located at approximately (92°20'W, 32°37'N) in the southeast corner of the Arkansas-Red Basin River Forecast Center study area. Note that the apparent area of an HRAP cell in the polar Stereographic plane (22,681,406 m^2) differs significantly from that in the Albers Equal-Area plane (15,369,703 m^2) which is the true earth area because, by definition, for an equal-area projection the product h*k = 1.0 at all points. Since h=k for a polar Stereographic projection, the ratio of the polar Stereographic area to the Albers area is very nearly equal to the square of the polar Stereographic scale factor at the average geocentric latitude of the cell. This means that it is critical that when NEXRAD StageIII data are used, the map cell area should not be literally interpreted as a true area

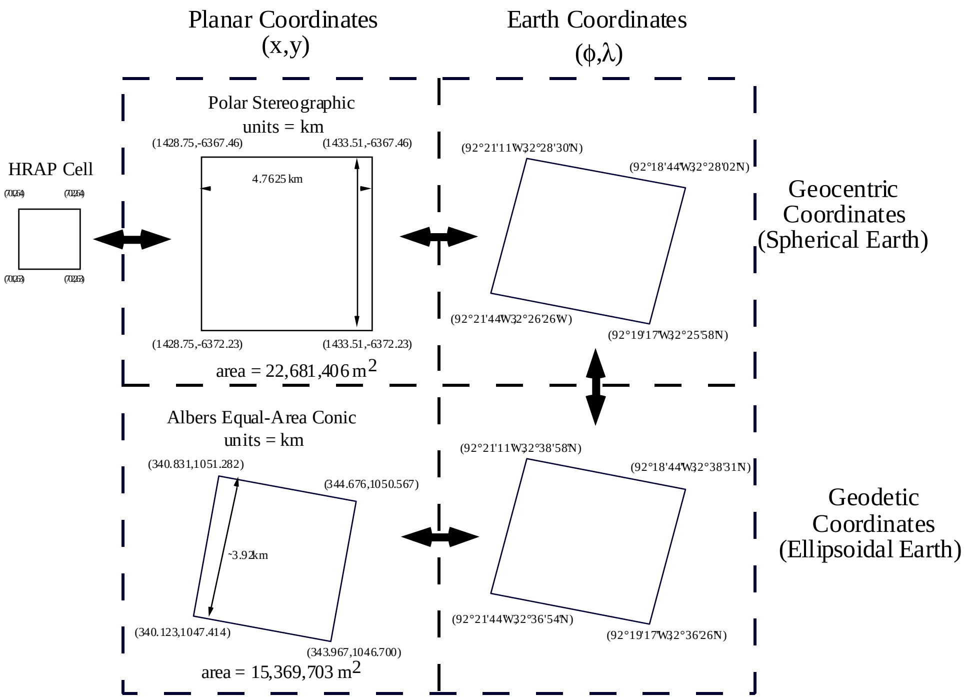


Figure 3.5: Size and Shape of an HRAP Cell (701, 263) in Several Coordinate Systems

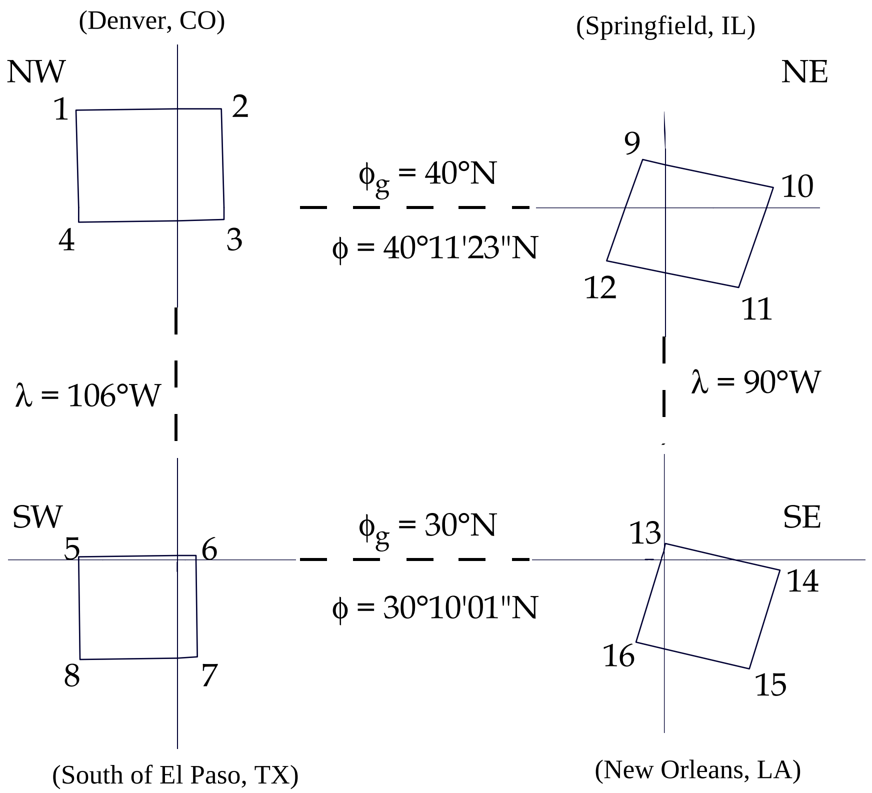


Figure 3.6a: Four HRAP Cells Plotted in Geographic Coordinates

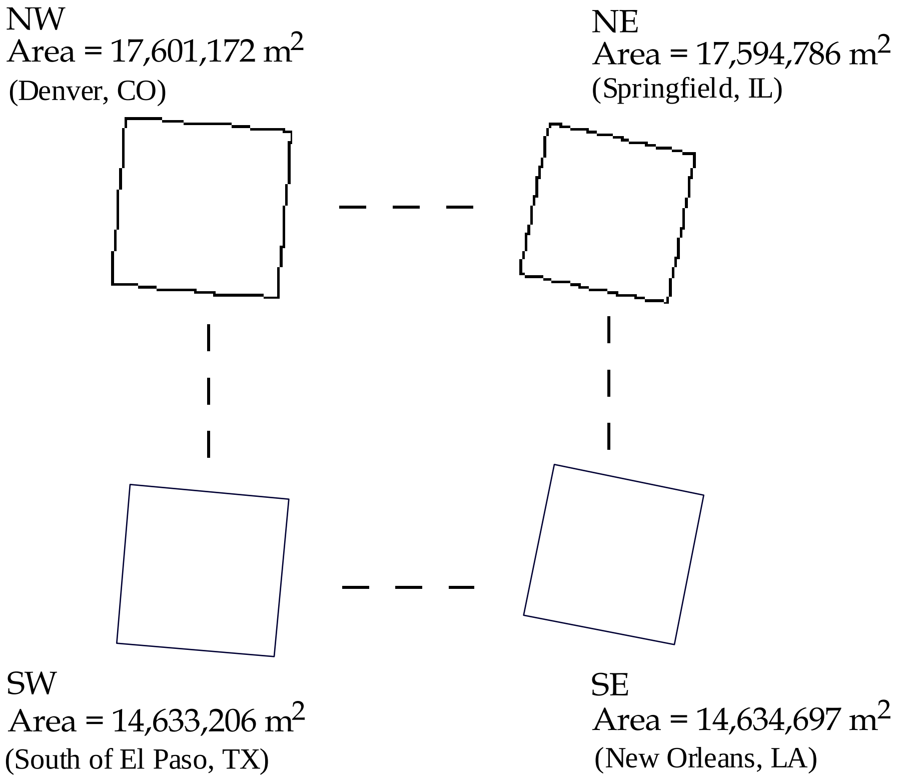


Figure 3.6b: Four HRAP Cells Transformed into Albers Equal-Area Projection

on the ground, particularly if a polar Stereographic projection is used. A more reliable approach is to project the cells into an equal-area projection so that the true earth area is closely approximated even if each cell still has a unique size and shape.

To demonstrate the uniqueness in size and shape of each HRAP cell, Figure 3.6a shows four HRAP cells that contain the intersection of a meridian of longitude and a parallel of latitude. For example, the upper left cell contains the intersection of the 106° W meridian of longitude and the 40° N parallel of latitude (geocentric). Table 3.3 lists the geocentric and geodetic (NAD83) coordinates computed for the cells of Figure 3.6a. Figure 3.6b illustrates the same four cells in the Albers projection used for this study. The differences in areas between cells can be explained by the scale factor of the polar Stereographic projection. Clearly the NW and NE cells have a smaller scale factor than the SW and SE cells and therefore represent a larger area on the earth's surface. The reason that the NE cell contains less area than the NW cell can be understood by looking back at Figure3.6a. More of the NE cell lies below the 40° N parallel; therefore, the scale factor is larger.

Table 3.3: Coordinate Values for Figure 3.6a

						NAD83 Coordinates					
	HRAP		Geocentric			Longitud			Geodetic Latitude		
	Coordinates		Latitude			e					
NW	hrapx	hrapy	deg	min	sec	deg	min	sec	deg	min	sec
1	380	438	40	1	58	106	2	4	40	13	21
2	381	438	40	2	0	105	59	6	40	13	23
3	381	437	39	59	45	105	59	3	40	11	7
4	380	437	39	59	42	106	2	0	40	11	5
SW											
5	375	160	30	0	3	106	2	1	30	10	4
6	376	160	30	0	5	105	59	38	30	10	6
7	376	159	29	58	1	105	59	35	30	8	2
8	375	159	29	57	59	106	1	58	30	8	0
NE											
9	702	477	40	1	2	90	0	29	40	12	24
10	703	477	40	0	26	89	57	38	40	11	49
11	703	476	39	58	15	89	58	24	40	9	38
12	702	476	39	58	50	90	1	15	40	10	13
SE											
13	774	209	30	0	19	89	59	57	30	10	20
14	775	209	29	59	47	89	57	39	30	9	47
15	775	208	29	57	47	89	58	16	30	7	47
16	774	208	29	58	19	90	0	34	30	8	20

3.4.2 Shape Factor, C_s

A shape distortion is introduced due to the use of a spherical transformation on ellipsoidal coordinates as described by Snyder, (1987), pp. 24-27. For a Stereographic projection, this distortion means that (h/k) is not equal to one as it should be for a conformal projection. The actual value of the ratio (h/k) can be determined using the local shape factor (C_s). C_s obeys the relationship,

$$\left(\frac{h}{k}\right)_s = C_s\left(\frac{h}{k}\right)_e \tag{3.32}$$

where s = sphere and e = ellipsoid. This means that if the ratio $(h/k)_e$ is calculated using spherical map projection equations applied to ellipsoidal coordinates then the actual scale factor $(h/k)_s$ may be computed using Equation 3.32. Since $(h/k)_e$ is forced to unity by the polar Stereographic projection equations, the relationship between h and k on the spherical version is simply $h_s = k_s* C_s$. C_s can be computed for different ellipsoids using Equation 3.33 which corresponds to Equation (4-31) of Snyder, 1987, p. 26.

$$C_s = \frac{\left(1-e^2\sin^2\phi\right)}{\left(1-e^2\right)} \tag{3.33}$$

where ϕ is the geodetic latitude and e is the eccentricity of the ellipse. For Houston, Texas, at approximately 30° N latitude, $C_s = 1.00511$ based on the Clarke 1866 ellipsoid with a = 6378.206, $e^2 = 0.006768658$. Due to this distortion, a circle on the earth maps to an ellipse in a projected plane whose major (East-West) axis and minor (North-South) axis have the ratio 1.00511:1. This distortion would cause a 230 km radius circle associated with a radar beam to map as an ellipse with a difference between the lengths of its major and minor axis as much as 230 km* 0.00511 or 1.175 km. This difference arises because the radius of curvature of an ellipsoidal earth varies continuously with latitude and also with direction at any point on the earth's surface. Using the Clarke 1866 ellipsoid, the radius of curvature at 30° N in the North-South direction is 6351.148 km and the corresponding radius of curvature in the East-West direction is 6383.609 km. These values were computed using Equations 3.34 and 3.35 which correspond to Equations (4-18) and (4-20) given by Snyder (a = 6,378,206 m).

$$R' = \frac{a(1-e^2)}{(1-e^2\sin^2\phi)^{3/2}} \tag{3.34}$$

R' = radius of curvature in the plane of the meridian

$$N = \frac{a}{(1 - e^2 \sin^2 \phi)^{1/2}} \qquad (3.35)$$

N = radius of curvature in the plane perpendicular to the meridian and also perpendicular to the tangent surface

The ratio of the two radii of curvature at 30° N is 6383.609/6351.148 = 1.00511 which is how the shape factor C_s is determined.

Although the mapping error associated with using a spherical transformation on ellipsoidal coordinates might seem large (about 0.5% in the example above), it is important to keep in mind that maps extending over large areas typically introduce larger variations in the scale factor (Snyder, p.27). For example, with the national Albers projection used in this study both h and k vary by 0.8% between 30° N and 40° N.

3.5 RECONSIDERING THE MAPPING PROBLEM

Figure 3.7 shows circles representing the 230 km beam coverage of radars under the jurisdiction of the the Arkansas-Red Basin River Forecast Center in an Albers Equal-Area projection. Each circular coverage is a plane, traced out by the radar beam as it rotates. The angle of elevation of the beam changes during rotation to better sense the rainfall at different distances from the radar, but essentially each radar coverage is like a flat map, drawn in the plane of the beam, in which distance and bearing are measured in polar coordinates relative to the center of rotation, which is the radar location.

There is a class of map projections called azimuthal projections formed by a flat map tangent to the earth at a given location. The Stereographic projection is one of these (polar Stereographic means that the projection plane is perpendicular to the polar axis). If each radar is treated as an individual map and the mosaicing of radar data to form a composite map is desired, the operation in GIS terms is very straightforward. Each input "map" is in an azimuthal projection with the projection origin at the radar location and is forward transformed to a common map reference frame. If the Lambert Azimuthal Equal-Area projection is used on the input side and Albers Equal-Area on the output side, the area of radar cells will be preserved in this process. The overlay and compositing of the precipitation from various radars on the output map is a standard GIS

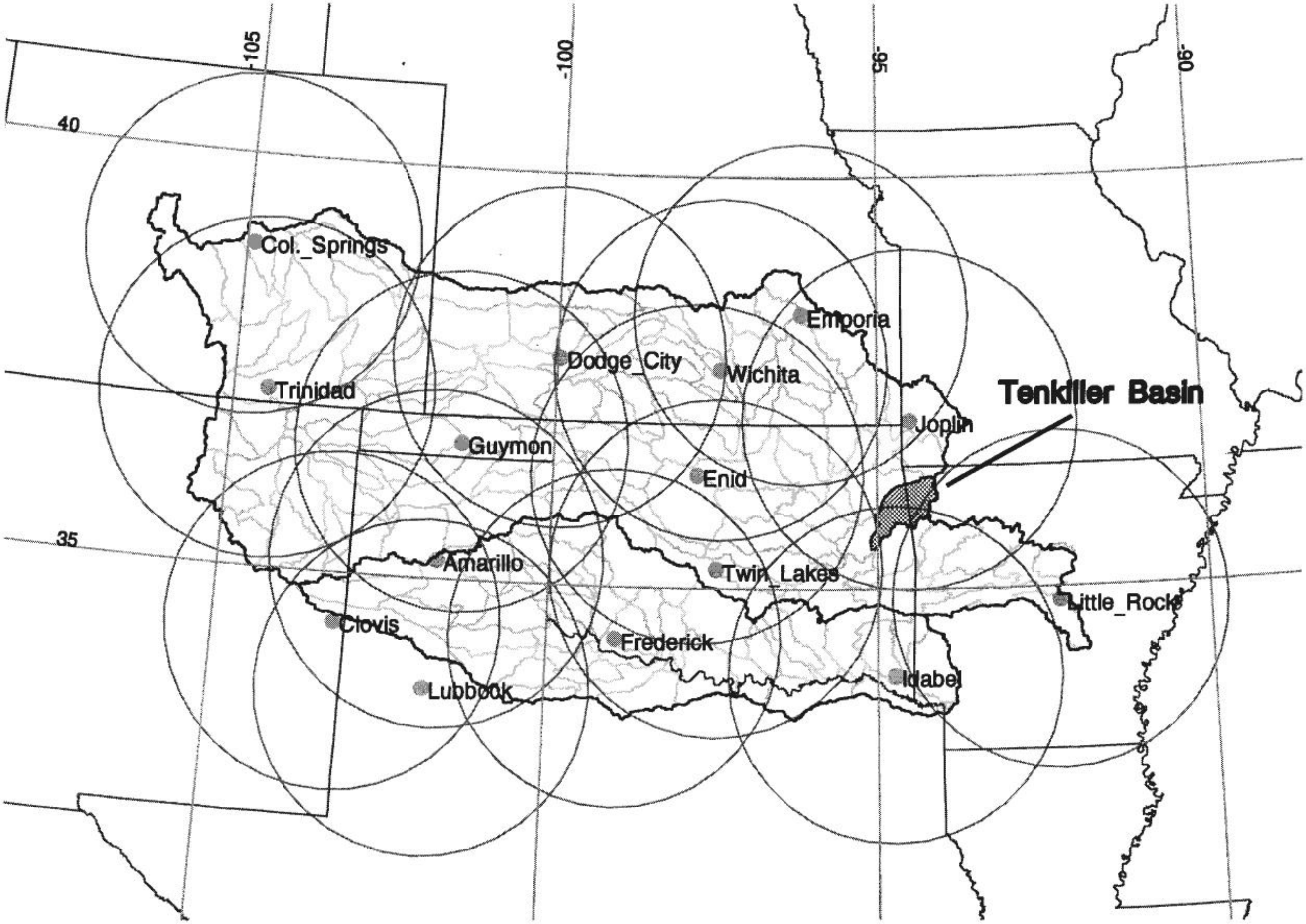


Figure 3.7: Arkansas-Red River Basin

operation. This GIS-based radar data transformation includes all of the required steps in Figure 3.4 and eliminates the shape factor distortion discussed above.

3.6 VERIFYING CONSISTENCY WITH ARKANSAS-RED BASIN RIVER FORECAST CENTER HRAP CELLS

Because of the complications involved with reproducing the HRAP cells in an ellipsoid-based coordinate system, some checks were made at control points provided by the Arkansas-Red Basin River Forecast Center to show that the sphere-ellipsoid transformation (Eqn. 3.8) should not be used for a matching transformation. Norm Bingham at the Arkansas-Red Basin River Forecast Center provided five "GIF" snapshots from their software depicting HRAP cells, state boundaries, and a few USGS gaging stations in the HRAP plane. The five snapshots are in the vicinity of state corners including the northwest corner of the Texas panhandle, the southeast corner of Colorado, the northwest corner of Louisiana, the northeast corner of Oklahoma, and the southwest corner of Missouri. Figure 3.8 (a-e) shows HRAP cells generated in GIS along with state boundaries from Environmental Systems Research Institute's ArcUSA CD-ROM in comparison with Arkansas-Red Basin River Forecast Center snapshots. The state boundaries from ArcUSA were transformed from geodetic coordinates into the HRAP coordinate system using Equations 3.17, 3.18, 3.20, and 3.21 for a spherical earth. After aligning the HRAP cells in our reproduction (first column in Figure 3.8) and those depicted in the ABRFC snapshots (second column in Figure 3.8), the distances between the corners of the state boundaries given by the two sources were approximated. The largest discrepancy occurred at the northeast corner of Oklahoma and was about 3.3 km measured in the polar Stereographic plane which corresponds to about 2.8 km on the earth's surface. A likely explanation for the discrepancy is that the two sets of state boundaries came from different sources and are inconsistent which is obvious in Figures 3.8b and e where the two sets of state boundaries clearly have a different shape. The ArcUSA data is at a relatively large scale, 1:2 million.

In addition to state boundaries, three USGS gaging stations were identified on the Arkansas-Red Basin River Forecast Center snapshots, two in Missouri and one in Oklahoma. The geodetic latitude and longitude for these stations were obtained via Internet at the address http://h2o.usgs.gov:81/swdata.html. Again, these points were transformed into HRAP coordinates using Equations 3.17, 3.18, 3.20, and 3.21. These gaging stations, along with their geographic and HRAP coordinates are listed in

Table 3.4. Figure 3.9 shows that our GIS reproduction of the gaging station HRAP coordinates listed in Table 3.4 along with our generated HRAP cell boundaries compares well with the position of these stations shown by the Arkansas-Red Basin River Forecast Center snapshots. The positions of the gaging stations in our reproduction are close enough to those in the ABRFC snapshots such that if the two maps were overlaid, with HRAP cells aligned, the map symbols used to identify gaging stations would overlap. The length of a side on the triangles used to represent gaging stations in the ABRFC snapshots is approximately 0.9 km in the polar Stereographic plane. The noted discrepancies between our reproduction and the ABRFC snapshots are far less than the differences between a geodetic and geocentric latitudes at the locations in Figures 3.8 and 3.9.

Table 3.4: USGS Gaging Stations Identified from Arkansas-Red Basin River Forecast Center Snapshots

Gaging Station	**λ**	**ϕ**	**hrapx**	**hrapy**
TIFM7: Elk River Near Tiff City, MO	-94.5867	36.6314	627.779	366.993
JOPM7: Shoal Creek Above Joplin, MO	-94.5161	37.0231	627.358	377.766
QUA02: Spring River Near Quapaw, OK	-94.7469	36.9344	622.858	374.490

3.8 DESCRIPTION OF FORTRAN AND AML CODES TO GENERATE CELLS AND TRANSFORM TO THE COMMON COORDINATE SYSTEM

Executing a FORTRAN code, genhrap.f, and an AML, genhrap.aml, (both listed in the Appendix) generates an Arc/Info coverage of HRAP cells in the chosen Albers projection. The geographic extent to be covered by these HRAP cells may be specified by the user. The user may specify the extent by geodetic latitudes and longitudes of the corners of the study region or by specifying the HRAP coordinate of the lower left hand corner of the study region and the number of columns and rows of cells to be created. Because the rainfall cells only need to be defined once, in whatever map projection a study is being made, it might be convenient at some point in the future to generate the HRAP grid for the entire United States and store this in a location from which users can cut out desired pieces. However, the Arkansas-Red Basin River Forecast Center radar coverage alone includes 53,365 cells and running simple codes has been easier than working with such a large file up to this point.

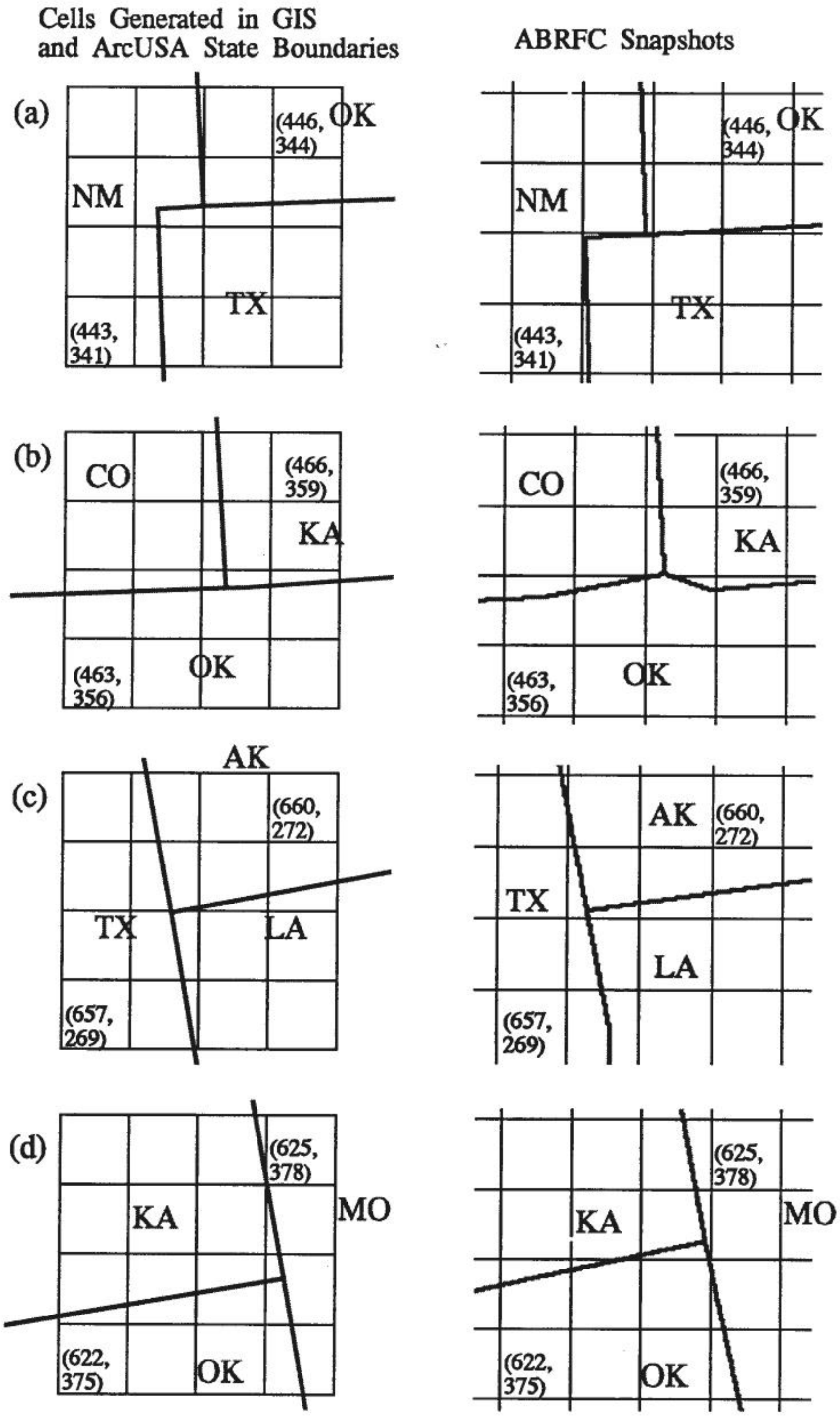
The code for genhrap.f consists of a main program and four subroutines. If the user chooses to specify the study extent with latitudes and longitudes, the subroutine llinput computes the corresponding extent of the area in HRAP coordinates. Llinput uses equations 3.17, 3.18, 3.20, and 3.21 to compute the HRAP coordinate corresponding to each geographic coordinate specified by the user. From the computed HRAP coordinates, llinput assigns the minimum hrapx and hrapy coordinates to be the lower left corner of the study area, computes the approximate number of rows and columns to span the geographic extent, and returns these values to the main program.

Given the geographic extent of the study area, genhrap.f first writes a file listing the HRAP coordinates of all corner points to be created beginning with the lower-left corner of the study area. Coordinates are written for the bottom row moving left to right, followed by the next row up and so on. This task is performed in the main program and the file generated is hrap.*cod* where *cod* is a user-defined code that is unique for a given run. For Tenkiller, the file hrap.tk3 was generated. HRAP coordinates from hrap.*cod* are converted to geocentric coordinates by the subroutine wll and written to geoc.*cod*. The subroutine wll uses the inverses of Equations 3.20 and 3.21, and Equations 3.22, 3.23, 3.24, and 3.28. The subroutine topoly reads the list of corner points from geoc.*cod* and creates a file (inputgc.*cod*) in the appropriate format for generating a polygon coverage. The last subroutine called by genhrap.f, crdat, writes a file, hrap.*cod*.dat, used to attach the correct HRAP-IDs as attributes to the final coverage in Albers. The hrapx and hrapy coordinates of its lower-left hand corner serve as the ID for an HRAP cell. Precipitation estimates available on Internet are listed according to the HRAP coordinates of a cell's lower-left hand corner; therefore, the HRAP-IDs are the only link between the geographic position of a cell and a precipitation depth obtained from Arkansas-Red Basin River Forecast Center.

Given the files inputgc.*cod* and hrap.*cod*.dat, genhrap.aml generates a polygon coverage called *cod*geocc, projects this coverage into the chosen projection producing *cod*geoccalb, creates an INFO data file (hrapxy.dat) and adds data from hrap.*cod*.dat to this file, and joins the newly created INFO file to the PAT of *cod*geoccalb. The file inputgc.*cod* is set up so that the polygons in geocentric coordinates that are created from this file are numbered from left to right starting with the bottom row followed by the next row up and so on. The numbering system is important because this is the key to joining the HRAP-IDs to the correct polygons. These polygon numbers get stored in the field *COD*GEOCCALB-ID of *cod*geoccalb.pat. The values in *COD*GEOCCALB-ID correspond to the values in the first column of the file hrap.*cod*.dat. Figure 3.10 illustrates the sequence of data files created by genhrap.f and genhrap.aml and how they are linked. A sample session showing execution of genhrap.f and genhrap.aml for Tenkiller is provided in Table 3.5.

Table 3.5: Sample Session for Generating a NEXRAD Mesh

```
unix% genhrap
 Enter 1 if you wish to specify the region by latitudes and longitudes of the corners of the study region.
Enter 2 if you   would like to specify region by hrap coordinates and number of   columns and rows.
2
 Enter the hrap(x,y) for the lower left hand corner of  the region of interest:
614 331
 Enter the number of grid columns and rows to be created:
32
32
 Enter a 3 character suffix to uniquely identify your grid:
tk3
nile.crwr.utexas.edu% arc
Arc: &r genhrap
Enter the 3 character suffix used to ID hrap files:: tk3
```

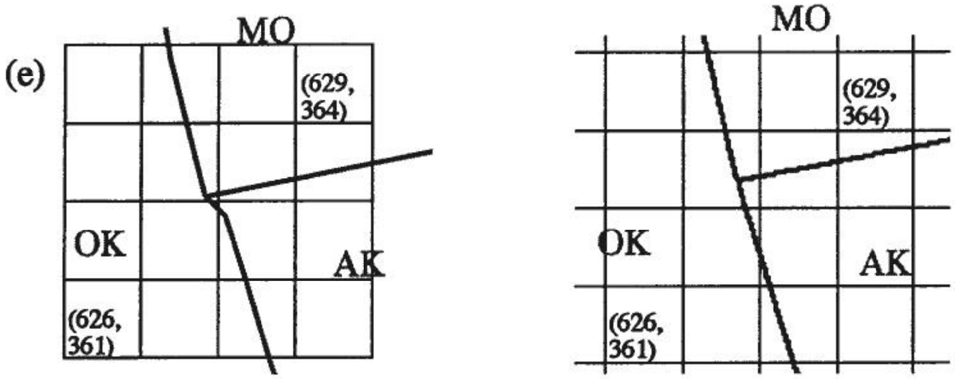


Cells Generated in GIS and ArcUSA State Boundaries

ABRFC Snapshots

Figure 3.8: Control Points at the Corners of State Boundaries

Figure 3.8 (cont.)



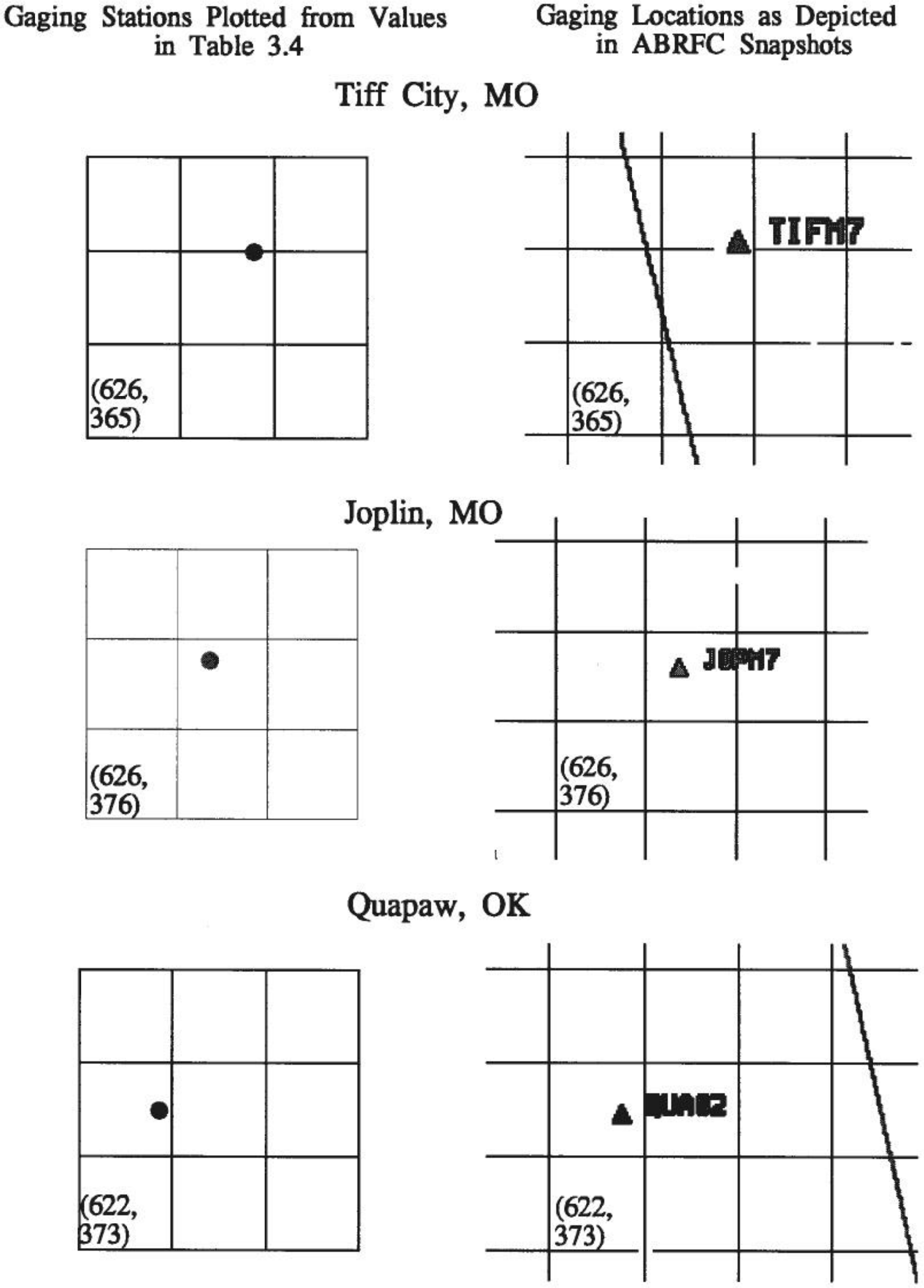
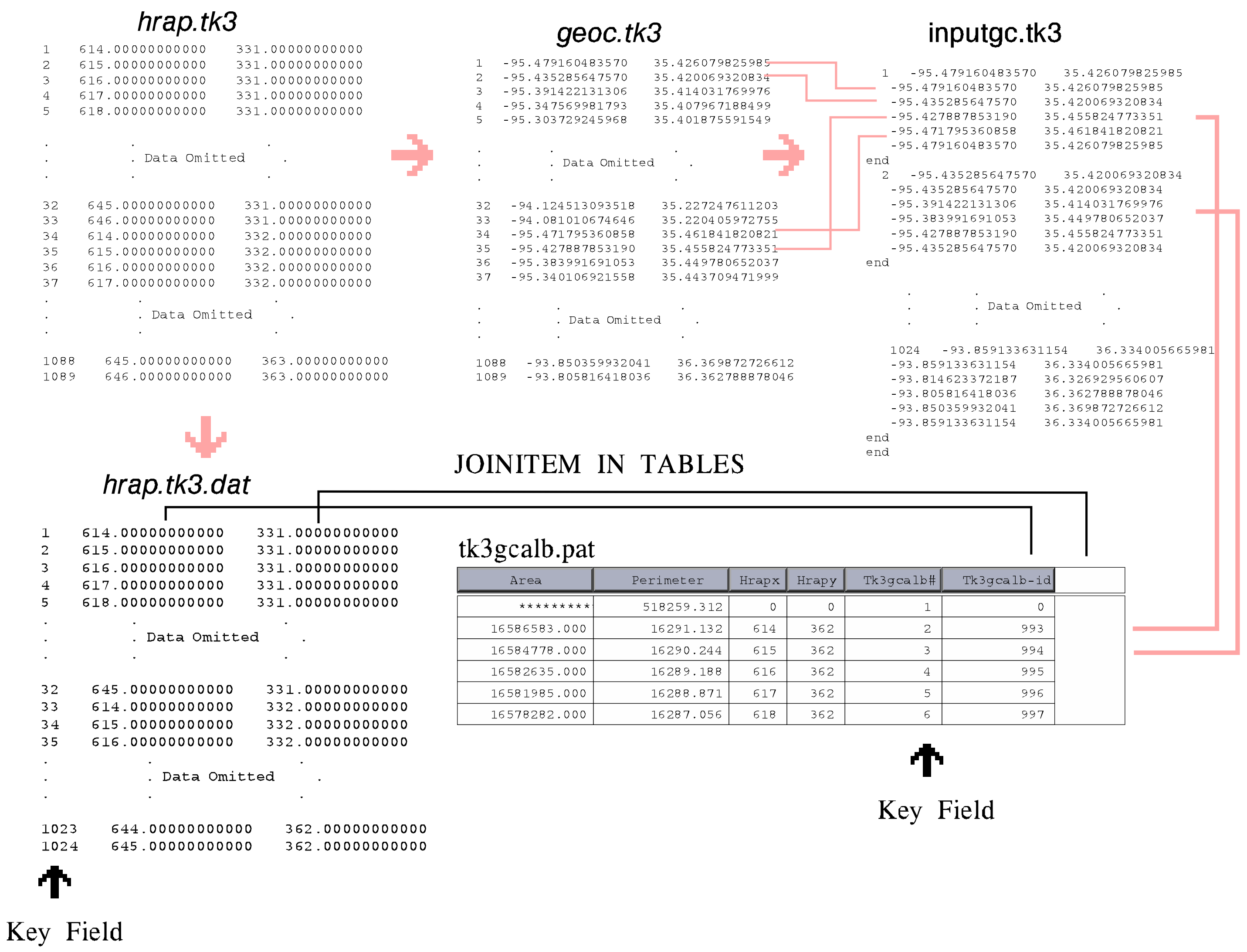


Figure 3.9: USGS Gaging Stations in the HRAP Plane



hrap.tk3

```
1    614.00000000000    331.00000000000
2    615.00000000000    331.00000000000
3    616.00000000000    331.00000000000
4    617.00000000000    331.00000000000
5    618.00000000000    331.00000000000

.            .                     .
.            . Data Omitted          .
.            .                     .

32    645.00000000000    331.00000000000
33    646.00000000000    331.00000000000
34    614.00000000000    332.00000000000
35    615.00000000000    332.00000000000
36    616.00000000000    332.00000000000
37    617.00000000000    332.00000000000
.             .                     .
.             . Data Omitted          .
.             .                     .

1088    645.00000000000    363.00000000000
1089    646.00000000000    363.00000000000
```

geoc.tk3

```
1   -95.479160483570    35.426079825985
2   -95.435285647570    35.420069320834
3   -95.391422131306    35.414031769976
4   -95.347569981793    35.407967188499
5   -95.303729245968    35.401875591549

.           .                    .
.           . Data Omitted         .
.           .                    .

32   -94.124513093518    35.227247611203
33   -94.081010674646    35.220405972755
34   -95.471795360858    35.461841820821
35   -95.427887853190    35.455824773351
36   -95.383991691053    35.449780652037
37   -95.340106921558    35.443709471999

.            .                    .
.            . Data Omitted         .
.            .                    .

1088   -93.850359932041    36.369872726612
1089   -93.805816418036    36.362788878046
```

inputgc.tk3

```
   1   -95.479160483570    35.426079825985
    -95.479160483570    35.426079825985
    -95.435285647570    35.420069320834
    -95.427887853190    35.455824773351
    -95.471795360858    35.461841820821
    -95.479160483570    35.426079825985
end
   2   -95.435285647570    35.420069320834
    -95.435285647570    35.420069320834
    -95.391422131306    35.414031769976
    -95.383991691053    35.449780652037
    -95.427887853190    35.455824773351
    -95.435285647570    35.420069320834
end

      .          .                    .
      .          . Data Omitted         .
      .          .                    .

    1024   -93.859133631154    36.334005665981
    -93.859133631154    36.334005665981
    -93.814623372187    36.326929560607
    -93.805816418036    36.362788878046
    -93.850359932041    36.369872726612
    -93.859133631154    36.334005665981
end
end
```

hrap.tk3.dat

```
1    614.00000000000    331.00000000000
2    615.00000000000    331.00000000000
3    616.00000000000    331.00000000000
4    617.00000000000    331.00000000000
5    618.00000000000    331.00000000000
.          .                    .
.          . Data Omitted         .
.          .                    .

32    645.00000000000    331.00000000000
33    614.00000000000    332.00000000000
34    615.00000000000    332.00000000000
35    616.00000000000    332.00000000000
.             .                    .
.             . Data Omitted         .
.             .                    .

1023    644.00000000000    362.00000000000
1024    645.00000000000    362.00000000000
```

tk3gcalb.pat

Area	Perimeter	Hrapx	Hrapy	Tk3gcalb#	Tk3gcalb-id
*********	518259.312	0	0	1	0
16586583.000	16291.132	614	362	2	993
16584778.000	16290.244	615	362	3	994
16582635.000	16289.188	616	362	4	995
16581985.000	16288.871	617	362	5	996
16578282.000	16287.056	618	362	6	997

Figure 3.10: Files Used to Create a Coverage of HRAP Cells

4.0 DETERMINE THE AREA AND MEAN TRAVEL LENGTH FOR EACH HRAP CELL IN A SUBWATERSHED

4.1 INTERSECT PROCESSED DIGITAL ELEVATION MODEL WITH THE COVERAGE OF HRAP CELLS

Once the HRAP cell mesh is generated for the region of interest, the mesh is merged with the digital elevation model of the watershed produced in Chapter 2 to generate parameters for each HRAP cell. Specifically, what is determined here is the average flow distance of each HRAP cell to a watershed outlet. The program hrap_int.aml in the Appendix intersects a vector coverage of HRAP polygons (tk3geoccalb) with a vector coverage of watershed boundaries (tksubshedsc), computes Flowlength statistics for each resulting sector based on the values in flmerge_grid, and writes selected statistics to an output file. The input requirements are a watershed coverage, a coverage of HRAP polygons, a value (Flowlength) grid, and a watershed grid. The names of the inputs and an output file name are passed as arguments at the command line. For example,

```
Arc: &r <aml_name> <wshed_cov> <hrap_cov> <value_grid> <wshed_grid> <out_file>
Arc: &r hrap_int tksubshedsc tk3geoccalb flmerge_grid tksubsheds tk3file.out
```

Use of the Arc Intersect command, the Grid Polygrid function, the Grid Zonalstats function, a Tables Relate, and the Tables Unload command are key lines in hrap_int.aml.

(1) INTERSECT <in_cover> <intersect_cover> <out_cover>

```
intersect %.subshed_cov% %.hrap_cov% sector_cov
```

For the Tenkiller watershed, %.subshed_cov% = tksubshedsc and %.hrap_cov% = tk3geoccalb. Intersect computes the geometric intersection of the subwatershed coverage and the HRAP coverage generating sector_cov which contains only areas common to the two input coverages. Sector_cov is shown in Figure 4.1. Sector_cov.pat is built automatically after intersection and retains all unique items from %.subshed_cov%.pat and %.hrap_cov%.pat. Most importantly, the grid-code from %.subshed_cov%.pat and the hrapx and hrapy attributes from %.hrap_cov%.pat are retained. The items in tksubshedsc.pat, tk3geoccalb.pat, and sector_cov.pat are listed below.

TKSUBSHEDSC-ID
GRID-CODE

TK3GEOCCALB-ID
HRAPX
HRAPY

SECTOR_COV-ID
TKSUBSHEDSC#
TKSUBSHEDSC-ID
GRID-CODE
TK3GEOCCALB#
TK3GEOCCALB-ID
HRAPX
HRAPY

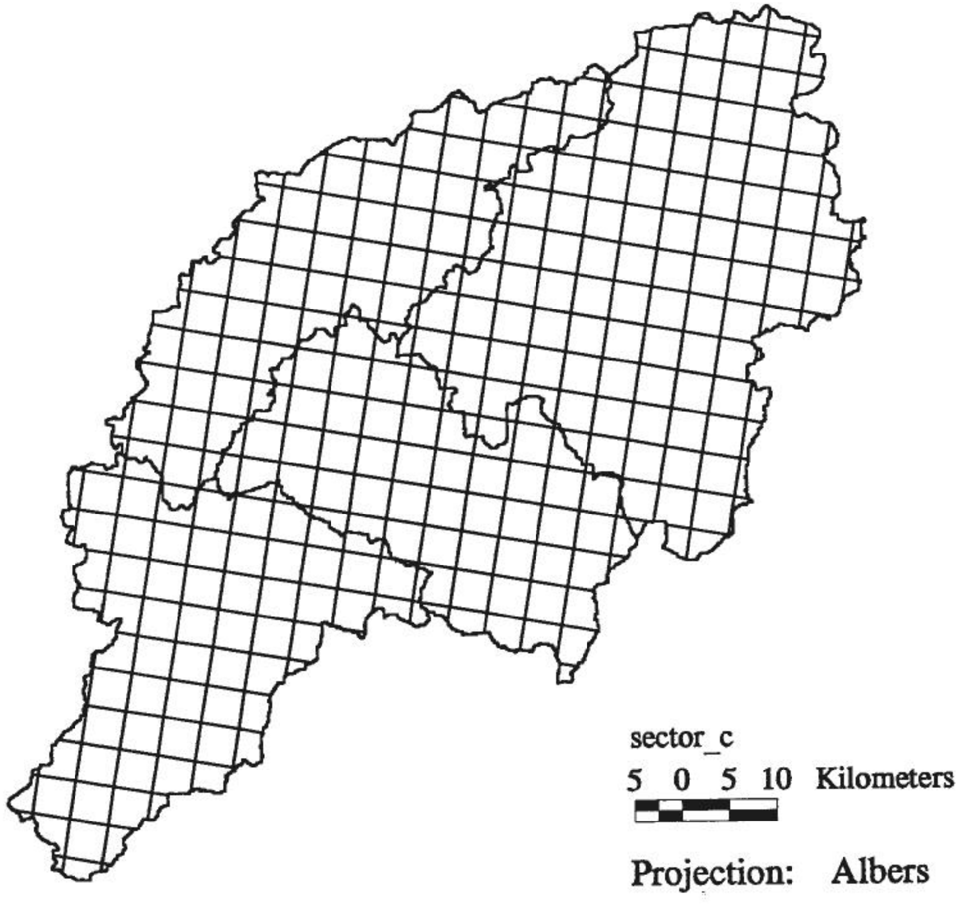


Figure 4.1: HRAP Coverage Intersected with a Subwatershed Coverage

tksubshedsc.pat	tk3geoccalb.pat	sector_cov.pat
AREA	AREA	AREA
PERIMETER	PERIMETER	PERIMETER
TKSUBSHEDSC#	TK3GEOCCALB#	SECTOR_COV#
TKSUBSHEDSC-ID	TK3GEOCCALB-ID	SECTOR_COV-ID
GRID-CODE	HRAPX	TKSUBSHEDSC#
	HRAPY	TKSUBSHEDSC-ID
		GRID-CODE
		TK3GEOCCALB#
		TK3GEOCCALB-ID
		HRAPX
		HRAPY

(2) POLYGRID(<cover>,{item},{lookup_table},{weight_table},{cell size})

sector_grid = polygrid (sector_cov,#,#,%cellsize%)

The Grid function Polygrid converts the coverage sector_cov into a grid in which all cells in a given sector (or zone) are assigned the value of the internal polygon number, sector_cov# by default. The cell size argument is read from the grid flmerge_grid and for the Tenkiller watershed is equal to 100 m. It should be noted that some polygons in sector_cov may be smaller than the size of one grid cell; these polygons do not retain a unique GRID-CODE during the vector to raster conversion. As a result, sector_grid.vat contains fewer records than sector_cov.pat and these small polygons in sector_cov.pat do not have an associated mean flowlength. The total area of these small polygons in the Tenkiller example with four subwatersheds was 0.0725 km^2 — only 0.002% of the total watershed area. This small area does not literally disappear during raster to vector conversion because the total number of cells in the raster representation is adjusted to reflect the total area in the vector representation as closely as possible. For example, the area of each 100 m raster cell at Tenkiller is 0.01 km^2 ; therefore, the area of sector_grid is equal to that of sector_cov to within 0.01 km^2.

(3) ZONALSTATS(<zone_grid>, <value_grid>, {stats_name}, {DATA |
NODATA})

```
flength.stat = zonalstats(sector_grid,flmerge_grid)
```

With the grid of zones established from polygrid, the Zonalstats function creates an INFO file that stores the statistics on cells in the <value_grid> that belong to a common zone in the <zone_grid>. There will be one record in the INFO file flength.stat for each sector with items VALUE, COUNT, MEAN, MIN, and MAX. VALUE is equal to sector_cov# and COUNT is equal to the number of cells in that sector. By default, when the user does not specify a statistic, Zonalstats will compute MEAN, MIN, and MAX. A sample of the file flength.stat is printed here. The units for MEAN, MIN, and MAX Flowlength are meters.

```
Enter Command: sel flength.stat
315 Records Selected.

Enter Command: list
Record    VALUE    COUNT     MEAN         MIN          MAX
   1       2        474      173727.594   172363.859   175119.500
   2       3        609      174331.188   172570.969   176250.875
   3       4        61       172292.359   171042.531   172698.172
   4       5        1593     169481.859   166014.094   172449.641
   5       6        9        172387.531   172198.172   172522.438
   6       7        1650     170554.547   167101.109   174770.969
   7       8        1071     169612.250   166496.938   172798.172
   8       9        159      176609.047   175526.609   177499.406
   .       .        .          .            .            .
   .       .        .          .            .            .
```

Note that the cells in the input grids sector_grid and flmerge_grid to the Zonalstats function may not be aligned precisely with one another but such misalignment will not alter computational results to a hydrologically significant degree.

(4) Establishing and using a Relate in TABLES.

```
sel sector_cov.pat
relate add                /* Add a relate to sector_cov.pat
relfl                     /* Relation name.
flength.stat              /* Name of table to be related.
info                      /* Type of table to be related.
sector_cov#               /* Relate item
value                     /* Related item
ordered                   /* Type of sorting (i.e, linear, ordered)
```

```
ro                                      /* Relate access (ro = read only)
.
.
.
reselect sector_cov# = relfl//value     /* Reduce the selection to rows in
                                        /* sector_cov.pat and flength.stat for which
                                        /* sector_cov# = value.  This is all rows
                                        /* except small polygons that were dropped.
reselect grid-code = [value basin%loops%]     /* Select rows in both tables that
                                        /* belong to a given watershed.
```

Once the statistics file has been written, a Relate is set up in the TABLES subprogram so that the mean Flowlength and area for sectors with a common grid-code (or sectors that are within the same watershed) can be unloaded to a data file. The Relate is set up between sector_cov.pat and flength.stat and the Relate items that link these two INFO tables are SECTOR_COV# and VALUE respectively. Thus, as a set of rows with a common GRID-CODE in sector_cov.pat is selected, the related rows in flength.stat are also selected.

```
(5)     UNLOAD <out_file> {item...item} {DELIMITED | COLUMNAR
       <format_file>} {INIT}
        unload %.outfile% grid-code hrapx hrapy relfl//mean area delimited
```

A loop controls the selection of columns and rows to be written to an ASCII file. The total number of subwatersheds is written at the top of the file followed by data for subwatershed1, subwatershed2, etc. The items GRID-CODE, HRAPX, HRAPY, MEAN (Flowlength), and AREA are written for each selected record. Because it was difficult to unload data in the exact format requested for the modClark parameter file using AML, a simple FORTRAN code, moutput.f, rewrites the ASCII file into the desired form. In this study, moutput.f read tk3file.out and created tk3modc.in.

4.2 RESULTS

The final product of this procedure is a file named tk3modc.in that lists HRAP cell characteristics required as input for the modClark (HEC, 1995) program. A sample from an output file is printed below. The characteristics listed for each cell are hrapx, hrapy,

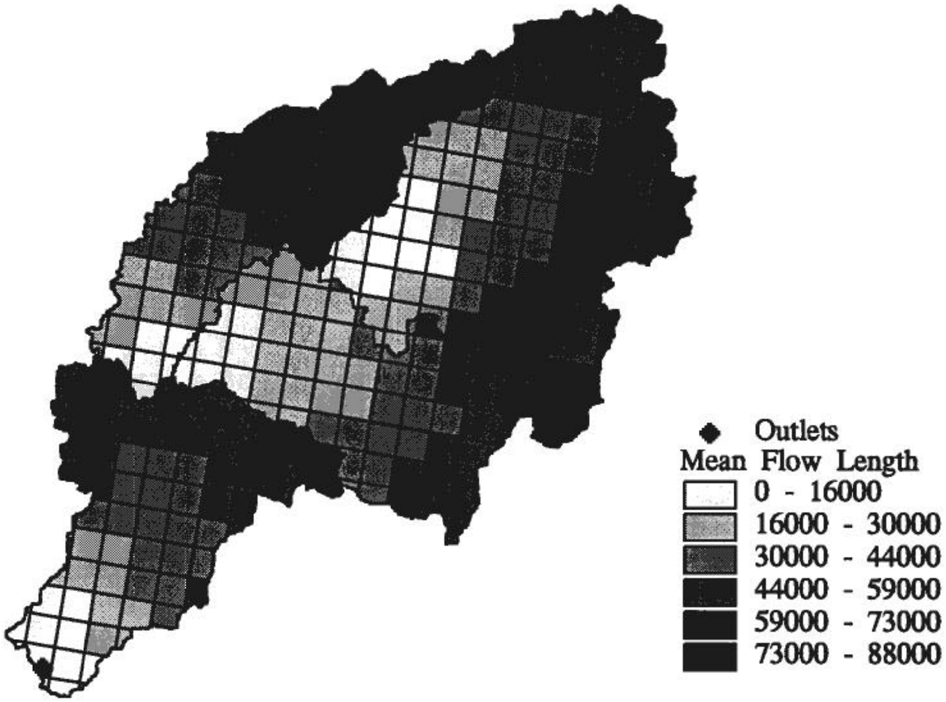


Figure 4.2: Mean Flow Length from HRAP Cells to Subwatershed Outlets

mean Flowlength to watershed outlet, and contributing area. Note that an HRAP cell may be split by a watershed boundary and therefore may be listed under more than one subwatershed. It is also possible that a watershed boundary may enter a cell, exit a cell, and re-enter a cell along a single side. With this scenario, the same HRAP-ID could be listed more than once for the same subwatershed.

```
     SUBBASIN:   85
     GRIDCELL:  633 359  86.8365  2.8668
     GRIDCELL:  634 359  87.4707  0.3250
     GRIDCELL:  632 358  81.4101 12.4244
     GRIDCELL:  633 358  84.4681 15.7702
     GRIDCELL:  634 358  87.3691  2.9036
     GRIDCELL:  630 358  71.8893  0.7228
     GRIDCELL:  631 358  76.4309  2.7685
     GRIDCELL:  629 357  67.4486  9.0327
     .    .    .
     .    .    .
     .    .    .
     GRIDCELL:  625 346   7.3406  0.0158
      END:
     SUBBASIN:   86
     GRIDCELL:  637 361  57.7422  4.7428
     GRIDCELL:  638 361  58.3458  6.0854
     .    .    .
     .    .    .
     .    .    .
     GRIDCELL:  636 361  56.3070  0.6148
.    .    .
```

Figure 4.2 is a shaded map of the resulting travel lengths for each sector.

5.0 CONCLUSIONS

The NEXRAD StageIII product offers high quality, hourly rainfall estimates with an approximate spatial resolution of 4 km by 4 km cells. This data provides much more information about how weather systems behave in space and time than can be inferred from rain gages alone. Use of this data should benefit the United States Army Corps of Engineers flood prediction and reservoir control operations. A repeatable GIS procedure consisting of a series of executable programs has been developed to merge NEXRAD radar data with DEMs and to extract hydrologic parameters. Provided with a reasonable UNIX platform and ARC/INFO software, a user could easily sit down and perform the procedure described here in a few hours for any site in the conterminous United States.

Proper geo-positioning of NEXRAD data turned out to be more complicated than expected. As discussed in Chapter 3, radar computations are performed in the HRAP coordinate system which is defined on a sphere, but when this system was originally set up by the National Weather Service the placement of radar sites in the HRAP coordinate system was done incorrectly. Geodetically defined latitude and longitude coordinates were interpreted as geocentric coordinates and projected onto a plane using equations for a spherical earth. Checking positions of USGS gaging stations (with known geodetic latitudes and longitudes) displayed on National Weather Service software confirmed the treatment of geodetic coordinates as geocentric coordinates. The procedure outlined in this report for positioning HRAP cells in an Albers projection defined on an ellipsoidal datum simply emulates this error in reverse. In the big picture of flood forecasting map transformation errors are much smaller than radar navigational errors and errors in estimating other hydrologic parameters such as losses, but GIS provides a mechanism for reducing map transformation errors. An approach to radar mapping has been described that could nearly eliminate map transformation errors, although some mapping errors inherent to the chosen projection for analysis would still be present.

Very likely, a large number of hydrologists around the country will be interested in the using publicly available NEXRAD estimates, and it seems prudent to make geo-positioning of rainfall estimates as simple as possible. Therefore, making a polygon coverage of HRAP cells in geodetic coordinates available on Internet in standard formats like ARC/INFO, GRASS, and USGS Digital Line Graph could save users a lot of time and eliminate transformation questions.

This report describes a technique for extracting HRAP cell characteristics but it is not clear that an HRAP cell is the most appropriate basic unit to drive a runoff model. One shortcoming of developing a model based on HRAP cells is that each cell has a unique area due to the scale factor associated with the polar Stereographic map projection; consequently, the conversion between precipitation depth and water volume is unique for every cell. One solution to this problem would be to map NEXRAD data onto a regular grid in an equal-area projection. Radar algorithms would not need to be modified to perform this mapping. An area-weight matrix, based upon the geometric intersection of HRAP cells with the regular array of cells in the equal-area plane, could transform currently available precipitation depths with associated HRAP-IDs into a gridded equal-area system.

Further thought should be put into whether or not the precipitation cell should be the computational basis for a runoff model, especially if other digital data sets with irregularly shaped polygons such as soils and landuse prove to be valuable for loss predictions. The watershed is traditionally the most studied and logical choice for a modeling basis. DEM analysis with GIS makes subdividing a watershed into smaller and smaller units an easy task. Thus, lumping precipitation or loss information over smaller watersheds may be an alternative approach for dealing with improved spatial estimates.

REFERENCES

Beavers, Mark, “Arc/Info - HEC-1 Interface: Working Papers,” Center for Research in Water Resources, Austin, TX, August, 1993.

Bingham, Norman, Personal Communication, Arkansas-Red Basin River Forecast Center, Tulsa, Oklahoma (1995).

ESRI, ArcDoc Version 7.0 (CD-ROM), Environmental Systems Research Institute, Redlands, CA, 1994.

Greene, Douglas R., and Hudlow, Michael D., "Hydrometeorologic Grid Mapping Procedures," AWRA International Symposium on Hydrometeorology, Denver, Colorado, June 13-17, 1982 (DRAFT).

HEC, "modClark User’s Manual," U.S. Army Corps of Engineers Hydrologic Engineering Center, Davis, CA, 1995.

HEC, “GridParm-DEM2HRAP: A Procedure for Evaluating Runoff Parameters for HRAP Cells from USGS Digital Elevation Models (DRAFT),” U.S. Army Corps of Engineers Hydrologic Engineering Center, Davis, CA, June, 9, 1995.

Hoke, J. E., Hayes, J. L., Renninger, L.G., "Map Projections and Grid Systems for Meteorological Applications,"Air Force Global Weather Central, March, 1981.

Hutchinson, M.F., “A New Procedure for Gridding Elevation and Stream Line Data with Automatic Removal of Spurious Pits,” *Journal of Hydrology*, 106, 211-232, 1989.

Lillie, Dale G., personal communication, Arkansas-Red Basin River Forecast Center, Tulsa, Oklahoma, 1994.

Lindsey, S.D., "Strategy for Utilizing Radar-based Precipitation Estimates for River Forecasting," *Engineering Hydrology : Proceedings of the Symposium Sponsored by the Hydraulics Division of the American Society of Civil Engineers*, San Francisco, CA, July 25-30, 1993.

Maidment, D.R., “A Grid-Network Procedure for Hydrologic Modeling,” Report Prepared for the Hydrologic Engineering Center, U.S. Army Corps of Engineers, Davis, CA, Contract DACW05-92-P-1983, October 14, 1992.

Maidment, D.R., “Grid-based Computation of Runoff : A Preliminary Assessment,” Report Prepared for the Hydrologic Engineering Center, U.S. Army Corps of Engineers, Davis, CA, Contract DACW05-92-P-1983, August 7, 1992.

Maidment, D.R., and Mizgalewicz, P., “Digital Delineation of Watersheds and Stream Networks in the Allegheny Basin,” Report Prepared for the Hydrologic Engineering Center, U.S. Army Corps of Engineers, Davis, CA, January 4, 1993.

Maidment, D.R., Olivera, J.F., Calver, A., Eatherall, A., and Fraczek, W., “A Unit Hydrograph Derived From a Spatially Distributed Velocity Field,” accepted for publication in *Hydrologic Processes*, 1995.

McDonnell, Porter W., Jr, *Introduction to Map Projections*, Marcel Dekker, Inc., New York, New York, 1979.

Shedd, R.C., and Fulton, R.A., "WSR-88D Precipitation Processing and its Use in National Weather Service Hydrologic Forecasting," *Engineering Hydrology : Proceedings of the Symposium Sponsored by the Hydraulics Division of the American Society of Civil Engineers*, San Francisco, CA, July 25-30, 1993.

Seo, D., and Miller, D.A., personal communication, National Weather Service Hydrologic Research Center, Silver Spring, Maryland, 1995.

Smith, J.A., “Precipitation,” Chapter 3, *Handbook of Hydrology*, Maidment, D.R. (editor), McGraw-Hill, New York, 1993.

Snyder, J.P., "Map projections — a working manual", Professional Paper 1395, US Geological Survey, Washington, D.C., 1987.

US Army Corps of Engineers, *Flood-Runoff Analysis: Engineering Manual*, EM 1110-2-1417, August 31, 1994.

USGS, “Digital Elevation Models,” USGeoData, Data Users Guide 5, United States Geological Survey, Reston, VA, 1990.

USGS, “Digital Line Graphs from 1:2,000,000-Scale Maps,” USGeoData, Data Users Guide 3, USGS, Reston, VA, 1990.

APPENDIX

Listing of Programs in the Order that They are Mentioned in the Text

1 MAKE_WIN.AML : Automatically creates windows and displays the the vicinity of outlet cells

2 FLOW_LENGTH.AML : Determines the flowlengths from each cell in a subwatershed to the outlet of that subwatershed

3 SLOPE.AML : If a time-index grid is being computed, this program is called by flow_length.aml to compute slope to the power b.

4 TIME_WEIGTH.AML : Computes a time-weight value when called by flow_length.aml.

5 MSWORKING2.AML : Called by flow_length.aml to provide messages to the user.

6 GENHRAP.F : Writes a file of coordinate values used to create a polygon coverage of NEXRAD cells in geographic coordinates given a user specified geographic extent.

7 GENHRAP.AML : Creates a polygon coverage of HRAP cells in geographic coordinates given the output from genhrap.f, projects these cells into Albers, and attaches the appropriate HRAP-ID values.

8 HRAP_INT.AML : Intersects a coverage of HRAP cells with a subwatershed coverage creating a number of sectors; computes mean flow length from each of these sectors to the appropriate subwatershed outlet.

9 MOUTPUT.F : Reformats the statistics file generated by hrap_int.aml.

** Codes are listed in order of their use in the procedure.
** Note on AMLs: in their current form, all output grids, coverages, and files will be killed if the procedure is run a second time without changing their names.

1 MAKE_WIN.AML

```
/************************************************************************
/************************************************************************
/* Name:  make_win.aml
/*
/*** Purpose:  This AML paints the vicinity of outlet locations in a point
/*** coverage so that the user can select the outlet cell from the
/*** streamlink grid which is closest to that point as a watershed outlet.
/*** Several new graphics windows are created.  The number of outlet
/*** locations that can be selected in one execution is influenced by the
/*** number of new windows that can fit on the screen.

&args linkgrid outlets
&type running make_win.aml
&messages &off &all
&if [iteminfo %outlets% -point X-COORD -exists] = .FALSE. &then
  &do
  &sys arc addxy %outlets% point
  &end
/*grid

&if [extract 1 [show display]] ne 9999 &then
  &do
   display 9999
  &end
mape %linkgrid%
describe %linkgrid%
&sv cellsize = %grd$dx%
units map
&sv mapxmin = [extract 1 [ show mape ] ]
&sv mapymin = [extract 2 [ show mape ] ]
&sv mapxmax = [extract 3 [ show mape ] ]
&sv pagxmin = [extract 1 [ show mape page ] ]
&sv pagxmax = [extract 3 [ show mape page ] ]
&sv mapfactor = ( %pagxmax% - %pagxmin% ) / ( %mapxmax% - %mapxmin% )
&sv mapxoffset = %mapxmin%
&sv mapyoffset = %mapymin%
&sv cellrange = 20.0

&sv end_of_points = .FALSE.
cursor out_cur declare %outlets%.pat info ro
cursor out_cur open

&sv count = 0
/*** Processing loop ***

&do &until %end_of_points% = .TRUE.
   &sv count = %count% + 1

   &sv x = %:out_cur.X-COORD%
   &sv y = %:out_cur.Y-COORD%
/*   &type %x%
/*   &type %y%
   &sv xmin = ( %x% - %cellrange% * %cellsize% - %mapxoffset% ) * %mapfactor%
   &sv xmax = ( %x% + %cellrange% * %cellsize% - %mapxoffset% ) * %mapfactor%
   &sv ymin = ( %y% - %cellrange% * %cellsize% - %mapyoffset% ) * %mapfactor%
   &sv ymax = ( %y% + %cellrange% * %cellsize% - %mapyoffset% ) * %mapfactor%
/*   &type %xmin% %ymin% %xmax% %ymax%
   &if %count% eq 1 &then
     windows create win%count% %xmin% %ymin% %xmax% %ymax% ~
        SIZE 350 350 POS UL DISPLAY UR
   &if %count% eq 2 &then
     windows create win%count% %xmin% %ymin% %xmax% %ymax% ~
```

```
        SIZE 350 350 POS UL WINDOW win1 LL
   &if %count% eq 3 &then
     windows create win%count% %xmin% %ymin% %xmax% %ymax% ~
        SIZE 350 350 POS UL DISPLAY LL
   &if %count% eq 4 &then
      windows create win%count% %xmin% %ymin% %xmax% %ymax% ~
        SIZE 350 350 POS UL WINDOW win3 UR

cursor out_cur next
&if %:out_cur.AML$NEXT% = .FALSE. &then
  &do
    &sv end_of_points = .TRUE.
    cursor out_cur remove
/*    &type %end_of_points%
  &end
&end   /*End of Main Processing Loop
&messages &on
/*&type quitting grid
/*q

&return
```

2 FLOW_LENGTH.AML

```
/*************************************************************************
/*************************************************************************
/* Name:  flow_length.aml
/*
/* Purpose:  Determine the flowlengths from each cell in a subwatershed to the
/*           outlet of that subwatershed.
/*
/*          Can also be used to compute the flowaccumulation for each cell in
/*            each subwatershed based only on flow originating in that
/*          subwatershed or to compute an integrated time-index parameter
/*          (requiring a call to time_weight.aml).
/*
/*          In its current form, this program also calls two "canned" programs
/*          described in "Arc/Info - HEC-1 Interface : Working Papers" by Mark
/*          Beavers (msworking2.aml and msworking2.menu).  These programs only
/*          supply information to the user and do not affect grid processing:
/*          the relevant lines can be commented out if desired.
/*
/* Inputs:   Two grids: (1) a projected grid of the subwatershed masks and
/*          (2) a grid of flowdirection for these subwatersheds.  Names of
/*            these input grids are supplied as arguments at the command line.
/*
/* Outputs:  The grid flmerge_grid contains the flowlengths from each cell in
/*           a subwatershed to the outlet of that sub-watershed.  If computed,
/*           the grid ftmerge_grid contains the time index value for each
/*           cell in a subwatershed based on flow originating in that
/*           subwatershed.
/*
/*************************************************************************
/*************************************************************************

/* Read in the names of the watershed grid and the direction grid as
/* global variables.
&args .subshed_grid .dir_grid .dem_grid

/* Initialize control variables.
&sv  first_time_thru = .TRUE.
&sv  end_of_subsheds = .FALSE.
&sv  mergelist1      = ' '
&sv  count           =  1
&sv  temp_count      =  1                /* TEMP
&sv  first_wshed = .TRUE.
```

```
/* Enter the grid module where processing will occur.
grid
&if [extract 1 [show display]] ne 9999 &then
   display 9999
ap gridnodatasymbol transparent
mape %.subshed_grid%

gridshades %.subshed_grid%

/*  Declare a cursor for the subshed grid, and open it.
/*  Also, check to make sure that there is something
/*  in the file to read.  If not, set a flag.
/*
cursor subshed_cur declare %.subshed_grid%.vat info ro
cursor subshed_cur open
&if %:subshed_cur.AML$NEXT% = .FALSE. &then
  &sv end_of_subsheds = .TRUE.


/******* Main processing loop. **************************************
/*
&do &while %end_of_subsheds% = .FALSE.

  &type loop begins [date -time]
  &if [exists temp_l%:subshed_cur.value% -grid] &then
    kill temp_l%:subshed_cur.value% all
  &if [exists temp_fa%:subshed_cur.value% -grid] &then
    kill temp_fa%:subshed_cur.value% all
  &if [exists temp_t%:subshed_cur.value% -grid] &then
    kill temp_t%:subshed_cur.value% all

  &if [exists length_grid -grid] &then
    kill length_grid all
  &if [exists time_grid -grid] &then
    kill time_grid all


  length_grid = flowlength (con (%.subshed_grid% == %:subshed_cur.value%, ~
                %.dir_grid%), #, downstream)

  /*  At the time this AML was first written,
  /*  the flowlength function returned zero values
  /*  instead of NODATA values at all points outside a watershed but
  /*  inside the mapextent. That is the reason for the inclusion of the
  /*   next line.  This problem may have been fixed in a later version.
```

```
 temp_l%:subshed_cur.value% = con (length_grid ne 0, length_grid)

/* <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
/* If it is desired to compute a time-index parameter, run time_weight.aml
/* followed by a weighted flowlength command.
/*
/*temp_fa%:subshed_cur.value% = flowaccumulation (con (%.subshed_grid% ~
/*                              == %:subshed_cur.value%, %.dir_grid%))
/*&if %first_wshed% = .TRUE. &then
/*   &do
/*     &r slope %.dem_grid%
/*     &sv first_wshed = .FALSE.
/*   &end
/*&r time_weight temp_fa%:subshed_cur.value%

/*time_grid = flowlength (con (%.subshed_grid% == %:subshed_cur.value%, ~
/*              %.dir_grid%), tweight, downstream)
/*temp_t%:subshed_cur.value% = con(time_grid ne 0, time_grid)
/*<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

/***********************************************************************
 /*  The next section creates a long variable which contains the NAMES
 /*  of all the temporary "fl" and "ft" grids created above.  These will be
 /*  used in a MERGE statement after loop completes.  This
 /*  is done to minimize MERGE executions - this way it will only happen
 /*  once, at the very end.

 &if [length mergelist%count%] ge 1000 &then
   &do
     &sv count = %count% + 1
     &sv first_time_thru = .TRUE.
   &end
 &if %first_time_thru% = .TRUE. &then
   &do
     &sv mergelist%count% = temp_l%:subshed_cur.value%
/*      &sv mlistft%count% = temp_t%:subshed_cur.value%
     &sv first_time_thru = .FALSE.
   &end
 &else
   &do
     &sv mergelist%count% = [value mergelist%count%], temp_l%:subshed_cur.value%
/*      &sv mlistft%count% = [value mlistft%count%],temp_t%:subshed_cur.value%
   &end

 /*  Display the current subshed, so the user will know something
 /*  is happening.
```

```
  gridpaint temp_l%:subshed_cur.value% value linear nowrap gray

  cursor subshed_cur next
  &if %:subshed_cur.AML$NEXT% = .FALSE. &then
    &do
      &sv end_of_subsheds = .TRUE.
      cursor subshed_cur remove
    &end
  &else
    &do
      &sv msg1 = 'Longest Streamlength Determination'
      &sv msg2 = Processing Subwatershed %:subshed_cur.value%
      &r msworking2 update %msg1% [quote %msg2%]
    &end
&end
/******* END OF MAIN PROCESSING LOOP **************************

/*  This kill was moved down here so that if the user bailed out of the program
/*  early, the merge_grid would still be intact (if it existed from a previous run).

&if [exists flmerge_grid -grid] &then
  kill flmerge_grid all
&if [exists ftmerge_grid -grid] &then
  kill ftmerge_grid all

/*  Merges all the flowlists and musklists created above.  Only three lists are
/*  coded for here, but any number is possible - three should be sufficient.
&sv msg1 = 'Longest Streamlength Determination'
&sv msg2 = Creating longest streamlength grid MERGE_GRID...
&r msworking2 update %msg1% [quote %msg2%]
&if %count% = 1 &then
  &do
    flmerge_grid  = merge ( %mergelist1% )
/*    ftmerge_grid = merge ( %mlistft1% )
  &end
&if %count% = 2 &then
  &do
    flmerge_grid  = merge ( %mergelist1%, %mergelist2% )
/*    ftmerge_grid = merge ( %mlistft1%, %mlistft2% )
  &end
&if %count% = 3 &then
  &do
    flmerge_grid  = merge ( %mergelist1%, %mergelist2%, %mergelist3% )
/*    ftmerge_grid = merge ( %mlistft1%, %mlistft2%, %mlistft3% )
  &end
```

```
/******** FILE CLEANUP: REMOVE ALL TEMPORARY GRIDS CREATED
&if [exists length_grid -grid] &then
  kill length_grid all
&if [exists time_grid -grid] &then
  kill time_grid all
&sv end_loop = .FALSE.

cursor subshed_cur declare %.subshed_grid%.vat info ro
cursor subshed_cur open
&do &until %end_loop% = .TRUE.

    &if [exists temp_l%:subshed_cur.value% -grid] &then
     kill temp_l%:subshed_cur.value% all
/*   &if [exists temp_ft%:subshed_cur.value% -grid] &then
/*     kill temp_ft%:subshed_cur.value% all
/*    &if [exists temp_fa%:subshed_cur.value% -grid] &then
/*     kill temp_fa%:subshed_cur.value% all
     cursor subshed_cur next
     &if %:subshed_cur.AML$NEXT% = .FALSE. &then
       &do
          &sv end_loop = .TRUE.
          cursor subshed_cur remove
       &end

&end                /* End of loop

quit  /* quit Grid subprogram
&return
```

3 SLOPE.AML

```
/**********************************************************************
/**********************************************************************
/* Name: slope.aml
/*
/* Purpose:  If desired, called by flowlength.aml to compute slope to the power b.
/*
/* Read in the names of the dem_grid and the flowaccumulation grid
&args .dem_grid
&if [exists slope1 -grid] &then
   kill slope1 all
&if [exists slope_grid -grid] &then
   kill slope_grid all
slope1 = slope( %.dem_grid%, percentrise )
slope_grid = slope1 div 100

/*** Compute S^b
&sv b = 0.5

&if [exists sb -grid] &then
   kill sb all
&if [exists slope_plus -grid] &then
   kill slope_plus all
/* Adjust the slope value by 0.0001 to avoid dividing by zero.
slope_plus = slope_grid + 0.0001
sb = pow( slope_plus, %b% )
&return
```

4 TIME_WEIGHT.AML

```
/************************************************************************
/************************************************************************
/* Name: time_weight.aml
/*
/* Purpose:  Generates a grid in which the value (1/S^bA^c ) is computed for
/* each cell in a watershed.  This grid can be used as a weight grid to
/* compute an integrated time index value using the flowlength function.
/* Called by flow_length.aml.  Assumes b = c = 0.5.
/*
/* Inputs:   Two grids: (1) a projected grid of the DEM used to compute the
/* slope and (2) a grid that contains the flowaccumulation values.  Depending
/* on how the flow routing is to be done, the flowaccumulation values might be /*
computed on a per-subwatershed basis or on a basin basis
/* -- in these two cases, flowaccumulation values would only differ along the
/* main stream stem. Both cases could be easily implemented within the
/* framework of this procedure. In its current form flowaccumulation is
/* computed on a basin basis.  The name of the projected DEM grid and the
/* flowaccumulation grid are passed as arguments at the command line.
/*
/* Output:   A grid named tweight.
/*
/************************************************************************
/************************************************************************

/* Read in the names of the dem_grid and the flowaccumulation grid
&args .fa_grid

/*** Compute S^bA^c

&sv c = 0.5

&if [exists ac -grid] &then
  kill ac all
&if [exists sbac -grid] &then
  kill sbac all
&if [exists fa_plus -grid] &then
  kill fa_plus all

/* Adjust the flowaccumulation value by 0.5 to avoid dividing by zero.
fa_plus = %.fa_grid% + 0.5
ac = pow( fa_plus, %c% )

sbac = sb * ac
/** Creating an index of travel time to the outlet
```

```
&if [exists tweight -grid] &then
   kill tweight all
tweight = 1 / sbac
/*quit /* Do not quit out of grid if called from fl_arg.aml.
&return
```

5 MSWORKING2.AML

```
/*------------------------------------------------------------------------
/*          Environmental Systems Research Institute
/*------------------------------------------------------------------------
/*   Program: MSWORKING2.AML
/*   Purpose: Display a menu with information that an action is taking
/*          place  (let the user know that something is happening).
/*          The message can be updated by using the UPDATE routine.
/*
/*------------------------------------------------------------------------
/*     Usage: msworking {INIT} <'message_1'> {'message_2'} {'position'} {'stripe'}
/*     Usage: msworking <routine_name>
/*
/* Arguments: routine - routine to be run
/*
/*          message_1 - The first line of the message to be displayed
/*          message_2 - The second line of the message to be displayed
/*          position  - (quoted string) menu position
/*          stripe    - (quoted string) menu stripe
/*
/*   Globals:
/*------------------------------------------------------------------------
/*     Calls: MSWORKING.MENU
/*------------------------------------------------------------------------
/*     Notes: All arguments must be quoted, and each of the message
/*          arguments should contain no more than 80 characters.
/*------------------------------------------------------------------------
/*     Input:
/*    Output:
/*------------------------------------------------------------------------
/*   History: Matt McGrath - 02/14/92 - Modified INFORM tool
/*          bernie szukalski - 09/16/92 - added UPDATE routine, changed
/*                               variable naming.
/*          bernie szukalski - 01/21/93 - added position & stripe args
/*          mark beavers     - 08/04/93 - added icon_name variable
/*==============================================================
===========
/*
&args routine message_1 message_2 position stripe icon_name

&severity &error &routine bailout

/* Check arguments
&if [NULL %routine%] &then
  &call usage
```

```
/* Default to the init routine if no routine has been specified
/*
&set routinelist = INIT UPDATE EXIT CLOSE USAGE
&if [KEYWORD %routine% %routinelist%] > 0 &then
  /* A routine has been specified
  &do
    &if [LOCASE %routine%] = init &then
      &do
        &set .msworking$message1 = [UNQUOTE %message_1%]
        &set .msworking$message2 = [UNQUOTE %message_2%]
      &end
  &end
&else
  /* A routine has not been specified, default to init
  &do
    &set stripe              = %position%
    &set position            = %message_2%
    &set .msworking$message2 = [UNQUOTE %message_1%]
    &set .msworking$message1 = [UNQUOTE %routine%]
    &set routine = INIT
  &end
/*
&call %routine%
/*
&return

/*------------
&routine UPDATE
/*------------
&set .msworking$message1 = [UNQUOTE %message_1%]
&set .msworking$message2 = [UNQUOTE %message_2%]

&thread &synchronize tool$msworking
&return

/*------------------
&routine USAGE
/*------------------
/* &type Usage: msworking <routine_name>
&type Usage: msworking2 INIT   <'"msg_1"'> {'"msg_2"'} {'"position"'} {'"stripe"'}
{icon-filename}
&type Usage: msworking2 UPDATE <'"msg_1"'> {'"msg_2"'}
&type Usage: msworking2 EXIT
&return &warning
```

```
*-------------------
&routine INIT
/*-------------------
/*
/* Check arguments
&if [NULL [VALUE .msworking$message1]] &then
  &call usage
/*
&if [NULL %.msworking$message2%] OR ~
    [QUOTE [UNQUOTE %.msworking$message2%_]] = [QUOTE #_] &then
  &set .msworking$message2
/*
&if [NULL %position%] OR %position%_ = #_ &then
  &set position = &cc &screen &cc
&if [NULL %stripe%] or %stripe%_ = #_ &then
  &set stripe = Working...
/*
/* Set the icon to be displayed in the menu
/* &set iconname = hourgls32.icon   /* Replaced with variable
&set iconname = %icon_name%
/*
/* Size the message menu based on the message string length
/*&set xsize = [LENGTH [QUOTE %message%]] * 10 + 60
/*&if %xsize% lt 250 &then &set xsize = 250
/*&set size = %xsize% 125
/*
&if not [SHOW &thread &exists tool$msworking] &then
  &thread &create tool$msworking ~
    &menu msworking2 ~
    &position [UNQUOTE %position%] ~
    &stripe [QUOTE [UNQUOTE %stripe%]]  ~
    &pinaction '&run msworking exit'

&thread &synchronize tool$msworking
/*
&return

/*-------------------
&routine EXIT
/*-------------------
/* Clean up
/*
&dv .msworking$*
&if [SHOW &thread &exists tool$msworking] &then
  &thread &delete tool$msworking
/*
```

```
&return

/*-------------------
&routine CLOSE
/*-------------------
/* Clean up
&call exit
/*
&return

/*------------------
&routine BAILOUT
/*------------------
&severity &error &ignore
&severity &warning &ignore
/*&call exit
&return &warning An error has occurred in routine: %routine% (MSWORKING.AML)

/*-----------------
&routine SAFETY_NET
/*-----------------
&return
```

WORKING2.MENU

```
7
/*--------------------------------------------------------------------------
/*            Environmental Systems Research Institute
/*--------------------------------------------------------------------------
/*    Menu: WORKING2.MENU
/* Purpose: Display a message while some action is executing.
/*--------------------------------------------------------------------------
/* Globals:
/*--------------------------------------------------------------------------
/*   Calls:
/*--------------------------------------------------------------------------
/*   Notes:
/*--------------------------------------------------------------------------
/* History: Matt McGrath - 02/10/92 - Mofified from the inform tool.
/*==============================================================
===========

 %icn      %msg1
          %msg2
%icn display iconname 8 ICON
%msg1 display .msworking$message1 65
%msg2 display .msworking$message2 65
```

6 GENHRAP.F

```
c*****************************************************************************
c Name and Location:  /export/home1/seann/hrapamls/crhrap/genhrap.f
c
c Purpose:  Write the HRAP coordinates for a selected region of cells to a
c file and create a subsequent file in geographic coordinates in a suitable
c format to serve as input to the GENERATE (polygon) command in ARC/INFO.
c This program is designed to be followed by genhrap.aml.
c
c Two options are available for defining the region of cells to be created --
c (1) Define the latitude and longitude extent of the region to be mapped, or
c (2) Specify the SW corner of the grid to be created and the number of colums
c and rows of cells to be created.  With either option, the program computes
c the HRAP coordinates of the SW corner (if necessary) and generates grid
c cells starting with the bottom row, moving from left to right, and then
c moving to the next row up and repeating.
c
c Comments:  Only the output files hrap.COD.dat and inputgc.COD are required
c as input to genhrap.aml.  Intermediate files and optional files that were
c created in an earlier version are also listed below.
c
c  Calls subroutines:  wll, topoly, crdat(numx,numy,xstart,ystart)
c
c Inputs: none
c Output: "COD" is a user defined suffix
c     hrap.COD = file of hrap coordinates  /*temporary
c     geoc.COD = file of geocentric coordinates  /*temporary
c     hrap.COD.dat = file containing HRAP coordinates in a format that can
c        be attached to the polygon attribute table
c     *pster.COD = file of polar stereographic coordinates
c     inputgc.COD = input file of geoc. coordinates to make a polgon
c        coverage
c     *inpster.COD = input file of polar stereographic coordinates to make
c     a polygon coverage
c     *inhrap.COD = input file of HRAP coordinates to make a polygon coverage
c
c A * denotes optional files -- the relevant lines have been commented out
c in this version.
c*****************************************************************************

      program genhrap

c     <<< Variable Declaration >>>
      parameter (maxcol = 336, maxrow = 160)
c    *** maxcol and maxrow are limited to the extent of HRAP cells for which
```

```
c    *** data is available in the Arkans.-Red River Basin

      integer xstart,ystart,numx,numy,numpts,numx1,numy1
      double precision xhrap(maxcol), yhrap(maxrow)
      integer count,bool,rfunit,wfunit
c    *** rfunit and wfunit store the readfile unit number and the
c    *** writefile unit number to be passed to the subroutine topoly.
      character suff*3,file1*8,file2*8,file3*12,file4*9,file5*11
      character file6*11,file7*10
c    ***
c    <<< End of variable declaration >>>

c    *** Allow two options for defining the study region.
      print*, 'Enter 1 if you wish to specify the region by latitudes and
     1 longitudes of the corners of the study region.  Enter 2 if you \
     2 would like to specify region by hrap coordinates and number of \
     3 columns and rows.'

      read*, bool

      if (bool.eq.1) then
         call llinput(xstart,ystart,numx1,numy1)
        else

        print*, 'Enter the hrap(x,y) for the lower left hand corner of
     1the region of interest:'

        read*, xstart,ystart
        print*, 'Enter the number of grid columns and rows to be
     1created:'
        read*, numx1,numy1
      endif

c    *** Number of points to write is one greater than the number of
c    *** columns or rows.  The name numx1 can be thought of as number of
c    *** x coordinates - 1.

      numx = numx1 + 1
      numy = numy1 + 1
      print*, 'Enter a 3 character suffix to uniquely identify \
     1your grid:'
      read*, suff

c    ***Create names for all of the output files.
c    *** file1 = file of hrap coordinates
c    *** file2 = file of geocentric coordinates
```

```
c    *** file3 = file containing HRAP coordinates in a format that can be
c              attached to the polygon attribute table
c    *** file4 = file of polar stereographic coordinates
c    *** file5 = input file of geoc. coordinates to make a polgon coverage
c    *** file6 = input file of p. stereographic coordinates to make a polgon
c              coverage
c    *** file7 = input file of hrap coordinates to make a polgon coverage

      file1 = 'hrap.'//suff
      file2 = 'geoc.'//suff
      file3 = 'hrap.'//suff//'.dat'
c      file4 = 'pster.'//suff
      file5 = 'inputgc.'//suff
c      file6 = 'inpster.'//suff
c      file7 = 'inhrap.'//suff


      open(unit = 10, file = file1, status = 'unknown')
      open (unit = 20, file = file2, status = 'unknown')
      open (unit = 30, file = file3, status = 'unknown')
c       open (unit = 40, file = file4, status = 'unknown')
      open (unit = 50, file = file5, status = 'unknown')
c       open (unit = 60, file = file6, status = 'unknown')
c       open (unit = 70, file = file7, status = 'unknown')

c    *** Compute the total number of cell corners
      numpts = numx*numy

      xnew = xstart

      do 100 i=1,numx
        xhrap(i) = xnew
        xtemp = xnew + 1.0
        xnew = xtemp
 100  continue

      ynew = ystart
      do 200 j=1,numy
        yhrap(j) = ynew
        ytemp = ynew + 1.0
        ynew = ytemp
 200  continue

      count = 1
      do 300 j=1,numy
        do 400 i=1,numx
```

```
c         ** with "count" in the file, this file can be used as input
c         ** for creating an ARC/INFO point coverage
           write(10,*) count,xhrap(i),yhrap(j)
           count = count + 1
 400     continue
 300     continue
c    write(*,*) ystart

      call wll(numpts)
      rfunit = 20
      wfunit = 50
      call topoly(numx,numy,rfunit,wfunit)
c     rfunit = 40
c     wfunit = 60
c     call topoly(numx,numy,rfunit,wfunit)
c     rfunit = 10
c     wfunit = 70
c     call topoly(numx,numy,rfunit,wfunit)
      call crdat(numx,numy,xstart,ystart)
      close(10)
      close(20)
      close(30)
      close(50)
      stop
      end

c<<<<<<<<<<<<<<<<<<<<<<<<SUBROUTINES>>>>>>>>>>>>>>>>>>>>>>>>>>
c *********************************************************************
c
c Purpose: Convert HRAP coordinates contained in file "hrap.COD" to lat-long
c coordinates based on a spherical earth and write them to an output file
c that can be used to generate a point coverage.  The parameter "numpts"
c stores the number of entries that will be expected from "hrap.COD"
c
c Inputs: file hrap.COD
c Outputs: file geoc.COD
c
c*********************************************************************
      subroutine wll(numpts)
      double precision xhrap, yhrap, x, y
      double precision bigr, arg, latd, lond, ang
      double precision stlatd,earthr,mesh,stlond
      integer rec,numpts
c
c***    Define constants
      stlond = -105.0
      stlatd = 60.0
c*** earthr,mesh, x, and y are in meters.
      earthr = 6371200.0
      mesh = 4762.5

      rewind(unit=10)

      do 100 i=1,numpts

        read(10,*) rec,xhrap, yhrap
        x = (xhrap - 401.0)*mesh
        y = (yhrap - 1601.0)*mesh

        bigr = (x*x + y*y)**0.5
        arg = bigr/(earthr*(1 + dsind(stlatd)))
        latd = 90.0 - 2*datand(arg)

        ang = datan2d(y,x)

        if (y.gt.0) then
          ang = 270.0-stlond-ang
        else
          ang = -90.0-stlond-ang
        endif
        if (ang.lt.180) then
          lond = -1 * ang
        else
          lond = 360.0 - ang
        endif
c***    Write polar stereographic coordinates and geocentric
c***    coordinates to a file.
c        write(40,*) i,x,y
        write(20,*) i,lond, latd
 100  continue
      return
      end

c**********************************************************************
c Purpose: Given a list of corner points for a grid (can be (ID,x,y) or
c          (ID, lon,lat) in which the coordinates for the bottom row are
c          listed one per line followed by the coordinates for the next row
c          up, create a file that can be used to generate a polygon coverage
c          of the grid cells.
c
c Input: File of corner points (ID,x,y),
c Ouput: File with lines: "poly-id, ll,lr,ur,ul,ll,end" -- repeated for
```

```
c        each polygon.  ll = lower left, lr = lower right, ur = upper right,
c        ul = upper left
c
c*********************************************************************
      subroutine topoly(numx,numy,rfunit,wfunit)

c       <<< Variable Declaration >>>
c       parameter (numx = 20, numy = 20)
c***    The old number of x-coordinates was 336.
c***    The old number of y-coordinates was 160.

      double precision xrowa(336),yrowa(336),xrowb(336),yrowb(336)
c        ** xrowa, yrowa are x and y coordinates of points in row a
      character*3 end
      integer i,l,rcount,r,polynum,numx,numy
      integer rfunit,wfunit
c     <<< End of Variable Declaration >>>

      end = 'end'

      rewind(unit=rfunit)
      rcount = 1
      polynum = 1

      do 200 i=1,numx
         read(rfunit,*) rec,xrowa(i),yrowa(i)
 200  continue

 100  if (rcount.lt.numy) then

      do 250 i=1,numx
         read(rfunit,*) rec,xrowb(i),yrowb(i)
 250  continue

      l = 1
 300  if (l.lt.numx) then
         r = l + 1
         write(wfunit,*) polynum, xrowa(l), yrowa(l)
         write(wfunit,*) xrowa(l),yrowa(l)
         write(wfunit,*) xrowa(r),yrowa(r)
         write(wfunit,*) xrowb(r),yrowb(r)
         write(wfunit,*) xrowb(l),yrowb(l)
         write(wfunit,*) xrowa(l),yrowa(l)
         write(wfunit,*) end
         l = l + 1
         polynum = polynum + 1
```

```
        goto 300
        endif

        rcount = rcount + 1
        do 350 i=1,numx
          xrowa(i) = xrowb(i)
          yrowa(i) = yrowb(i)
 350     continue
       goto 100
      endif
      write(wfunit,*) end

      return
      end

c **********************************************************************
c Purpose: This subprogram will create a data file that can be joined to the
c projected "hrap" polygon coverage so that "hrap" coordinates of the lower
c left hand corner of each polygon will be added to the appropriate line in
c the PAT.
c
c Note: The only difference between "hrap.COD.dat" produced by this
c subroutine and "hrap.COD" produced by the main program is that
c hrap.COD.dat does not contain entries for the last column and last
c row of points.
c **********************************************************************

      subroutine crdat(numx,numy,xstart,ystart)
c***    Old value of numx was 336
c***    Old value of numy was 160
      double precision xhrap(336), yhrap(160)
      integer count,numx,numy,xstart,ystart,numx1,numy1

      numx1 = numx - 1
      numy1 = numy - 1
      xnew = xstart

      do 100 i=1,numx1
        xhrap(i) = xnew
        xtemp = xnew + 1.0
        xnew = xtemp
 100  continue

      ynew = ystart
      do 200 j=1,numy1
        yhrap(j) = ynew
        ytemp = ynew + 1.0
        ynew = ytemp
 200  continue

      count = 1
      do 300 j=1,numy1
        do 400 i=1,numx1
          write(30,*) count,xhrap(i),yhrap(j)
          count = count + 1
 400    continue
 300  continue
      return
      end

c**************************************************************
c   At user's request, allow the user to input the latitude and
c   longitude of the four corners that are of interest in the
c   study.
c
c   Note: The user should input geodetic coordinates.  These
c   geodetic coordinates will be interpreted as geocentric coordinates
c   to be consistent with methodology used by the
c   National Weather Service.
c**************************************************************
      subroutine llinput(xstart,ystart,numx1,numy1)

c   <<< Variable Declaration >>>
      parameter (stlat = 60.0)
c*** clon is a constant used to account for the standard longitude
c*** see eqn. in "Geographic Positioning of the HRAP"
      parameter (clon = 15.0)
      parameter (rad = 6371.2)

      integer xstart,ystart,numx1,numy1
      real lon(4), lat(4)
      real sfactor,R,x,y,hrapx(4),hrapy(4)
c*** Declare variables llhrapx and llhrapy to pick the hrap coordinates of
c*** the lower left hand coordinates desired.
      real minhx,minhy,maxhx,maxhy
c   <<< End Variable Declaration >>>

      print*, 'Enter the latitudes and longitudes of four corners of a
     1 rectangle that encloses the study region (in decimal degrees). \
     2 Enter a longitude value and then a space and then a latitude \
     3 value. Hit return after each coordinate.  Remember to input West \
     4 longitude values as negative numbers.'
```

```
      do 100 i = 1,4

        read*, lon(i),lat(i)
        sfactor = (1+sind(stlat))/(1+sind(lat(i)))
c** x and y are in km
        R = rad*cosd(lat(i))*sfactor
        x = R*cosd(lon(i)+clon)
        y = R*sind(lon(i)+clon)
        hrapx(i) = x/4.7625 + 401
        hrapy(i) = y/4.7625 + 1601
        write(*,*) 'hrapx, hrapy:', hrapx(i), hrapy(i)
100   continue
        minhx = hrapx(1)
        minhy = hrapy(1)
        maxhx = hrapx(1)
        maxhy = hrapy(1)

      do 200 j = 2,4

         if (hrapx(j).lt.minhx) then
            minhx = hrapx(j)
         endif
         if (hrapy(j).lt.minhy) then
            minhy = hrapy(j)
         endif
         if (hrapx(j).gt.maxhx) then
            maxhx = hrapx(j)
         endif
         if (hrapy(j).gt.maxhy) then
            maxhy = hrapy(j)
         endif

 200  continue
      xstart = minhx
      ystart = minhy

      numx1 = maxhx - minhx
      numy1 = maxhy - minhy
      write(*,*) 'Lower left, num rows, num columns'
      write(*,*) xstart,ystart,numx1,numy1
      return
      end


c*******************************************************************
```

7 GENHRAP.AML

```
/***********************************************************************
/***********************************************************************
/*  Name and Location: /export/home1/seann/hrapamls/crhrap/genhrap.aml
/*  Purpose:  Generate polygon coverage(s) from user specified input file(s)
/*  (i.e. inputgc.COD)
/*  generated by genhrap.f, project the polygon coverage into chosen
/*  projection.  Create an INFO file with HRAP-IDs (given hrap.COD.dat), and
/*  join this INFO file to the PAT of the projected polygon coverage.
/***********************************************************************
/***********************************************************************

&sv suff = [response 'Enter the 3 character suffix used to ID hrap files:']
&sv covgc = %suff%geocc
&sv inputgc = inputgc.%suff%

&if [exists %covgc% -cover] &then
   kill %covgc% all
generate %covgc%
&if [exists %inputgc% -file] &then
input %inputgc%
&else &type Can't find input file.
polys
/* must quit out of the GENERATE sub-program
quit

clean %covgc%
&sv covgcprj = %covgc%alb
&if [exists %covgcprj% -cover] &then
   kill %covgcprj% all
project cover %covgc% %covgcprj% albdd.prj
clean %covgcprj%

tables
&if [exists hrapxy2.dat -info] &then
   &sv delvar = [delete hrapxy2.dat -info]

/* Add data to the INFO file hrapxy2.dat from the file hrap.***.dat
/* created by the FORTRAN program create.f
&sv addfile = hrap.%suff%.dat
/*add from %addfile%

define hrapxy2.dat
%covgcprj%-id
```

```
5
5
i
hrapx
4
4
i
hrapy
4
4
i
~
add from %addfile%
quit

/* Join the newly created INFO file to the PAT, creating two new colums
/* in the HRAP polygon coverage
joinitem %covgcprj%.pat hrapxy2.dat %covgcprj%.pat %covgcprj%-id %covgcprj%-id
~
        ordered
&return

/* Listing of albdd.prj
/*input
/*projection geographic
/*units dd
/*datum wgs72
/*parameters
/*output
/*projection albers
/*units meters
/*datum wgs72
/*parameters
/*29 30 00
/*45 30 00
/*-96 00 00
/*23 00 00
/*0.0
/*0.0
/*end
```

8 HRAP_INT.AML

```
/***********************************************************************
/***********************************************************************
/* Name: hrap_int.aml
/*
/* Purpose:  Intersect polygons representing subwatersheds and a radar
/* rainfall grid.  For the resulting coverage, determine
/* the mean, max, and min and median Flowlengths to the outlet from each of
/* the polygons and record this in the PAT of that coverage. Also compute
/* mean, max, and min values of the time_index parameter if desired.
/*
/* Execution: &r hrap_int <wshed_cov> <hrap_cov> <value_grid> <wshed_grid>
/*  <outfile>
/*
/* Inputs:   (1) a projected polygon coverage of subwatersheds, (2) a polygon
/*             coverage of an HRAP grid to intersect with the subwatershed
/*             coverages, (3) a value grid, (4) and a grid of the
/*           subwatershed.
/*
/* Outputs:   An output file unloaded from sector_cov.pat containing the
/*           following information for each subbasin: hrapx, hrapy,
/*           travel length to a subbasin outlet, and area of that cell
/*           draining to that subbasin.
/*
/* Comments: Polygons in sector_cov may be smaller than the size of one grid
/*          cell.  In this case, sector_grid.vat will contain fewer entries
/*          than sector_cov.pat because these small polygons were dropped.
/*          The precipitation and flowlength values written to sector_cov.pat
/*            for these polygons is zero.
/*            Before running this program, make sure that the HRAP polygons have
/*            been cleaned and projected into the same projection as the
/*            subwatershed coverage.  Also, make sure the coverage contains
/*            hrapx and hrapy values in its PAT.
/*
/**********************************************************************
/**********************************************************************

&args .subshed_cov .hrap_cov .valu_grid .subshed_grid .outfile

&if [exists sector_cov -cover] &then
  kill sector_cov all

intersect %.subshed_cov% %.hrap_cov% sector_cov

grid
```

```
&type what
&if [exists sector_grid -grid] &then
  kill sector_grid all

/*specify the cell size below
&describe %.valu_grid%
&sv cellsize = %grd$dx%
&sv max_fl = %grd$zmax%

sector_grid = polygrid (sector_cov,#,#,#,%cellsize%)
&type what

/***
/*  Create an INFO table that contains VALUE, COUNT, MEAN, MAX, MIN, and
/*  MEDIAN.
/*  VALUE = values of zones defined by sector_grid
/*  COUNT = number of cells in zones defined by sector_grid
/*  MEAN = mean of values from flowlength grid in zone defined by VALUE
/***

&if [exists flength.stat -info] &then
   &sv delvar = [delete flength.stat -info]
&if [exists flength.med -info] &then
   &sv delvar = [delete flength.med -info]
&if [exists sbac.stat -info] &then
   &sv delvar = [delete sbac.stat -info]
&if [exists sbac.med -info] &then
   &sv delvar = [delete sbac.med -info]
&if [exists flmerge_int -grid] &then
   kill flmerge_int all
&if [exists sbac_int -grid] &then
   kill sbac_int all
&if [exists time_ind_int -grid] &then
   kill time_ind_int all
&if [exists time_ind.stat -info] &then
   &sv delvar = [delete time_ind.stat -info]
&if [exists time_ind.med -info] &then
   &sv delvar = [delete time_ind.med -info]

/*flmerge_int = int (%.valu_grid%)
/*buildvat flmerge_int

flength.stat = zonalstats(sector_grid,%.valu_grid%)

/*flength.med = zonalstats(sector_grid,flmerge_int,median)
```

```
/*sbac_int = int (sbac)
/*sbac.stat = zonalstats(sector_grid,sbac)
/*sbac.med = zonalstats(sector_grid,sbac_int,median)

/*time_ind_int = int (time_ind)
/*time_ind.stat = zonalstats(sector_grid,time_ind)
/*time_ind.med = zonalstats(sector_grid,time_ind_int,median)

/* quit out of grid: joinitem cannot be used at the grid prompt
quit

/*********************************************************************
/* Join the info files created by zonalstats so that only one relate between
/* the PAT and the INFO files needs to be created.
/*********************************************************************

/*joinitem flength.stat flength.med flength.stat value max ordered

/*joinitem sbac.stat sbac.med sbac.stat value max ordered

/*joinitem time_ind.stat time_ind.med time_ind.stat value max ordered

/** Combining the three statistics tables was considered so that only one
/** "relate" would have to be established for the purpose of unloading data;
/** however, this would have required changing at least four of the item names.

/*joinitem flength.stat sbac.stat all.stat value median ordered

/*"Cursor" is not a valid command in TABLES


/* Declare a cursor named basin_cur on .subshed_grid.vat
/*&messages &off &all
&sv end_of_subsheds = .FALSE.
&sv count = 0
/*  &sv temp = 0

cursor subshed_cur declare %.subshed_grid%.vat info ro

cursor subshed_cur open

/*** Use a loop to count the number of subwatersheds.  Store the value
/*   in the variable 'count.'  The variable count will be used to control
/*   the loop that unloads data to an ASCII file.

&if %:subshed_cur.AML$NEXT% = .FALSE. &then
```

```
  &sv end_of_subsheds = .TRUE.
/* Make sure that 'temp' is the same item type as ':subshed_cur.value.'
&sv temp = %:subshed_cur.value% - 1
&do &while %end_of_subsheds% = .FALSE.
   &if %temp% ne %:subshed_cur.value% &then
      &do
         &sv count = %count% + 1
         &sv basin%count% = %:subshed_cur.value%
         &sv temp = %:subshed_cur.value%
         /* The variable 'temp' is used so that a subwatershed will not be
         /* counted more than once if it is listed twice in the VAT.
      &end

/*  Read next record from .subshed_grid.vat
  cursor subshed_cur next
  &if %:subshed_cur.AML$NEXT% = .FALSE. &then
    &do
      &sv end_of_subsheds = .TRUE.
      cursor subshed_cur remove
    &end
&end

/*******************************************************************************
/*  Enter tables to perform two tasks: (1) establish two relations: (a) between
/*  sector_cov.pat and flength.stat - call it "relfl" (b) between sector_cov.pat
/*  and time_ind.stat - callit "relti"; (2)  Using a loop and simple relates
/*  unload desired output for each of the sub-watersheds from the tables
/*  sector_cov.pat, flength.stat, and time_ind.stat.
/********************************************************************************

tables
sel sector_cov.pat
relate add
relfl
flength.stat
info
sector_cov#
value
ordered
ro
/*relsb
/*sbac.stat
/*info
/*sector_cov#
/*value
/*ordered
```

```
/*ro
/*relti
/*time_ind.stat
/*info
/*sector_cov#
/*value
/*ordered
/*ro
~

&if [exists %.outfile% -file] &then
   &sv delvar = [delete %.outfile% -file]

/*** Open the output file for writing.

&sv wfunit = [open %.outfile% openstat -append]
  &if %openstat% ne 0 &then
    &do
      &type openstat = %openstat%
      &stop Cannot open the output file %.outfile%
    &end
  &else &type File %.outfile% opened succussfully for writing.

/** Write the number of sub-watersheds being processed to the output file.
   &if [write %wfunit% %count%] ne 0 &then
     &do
       &type Error in writing to output file.  Exiting AML.
       &return
     &end

/***   Do not need to leave the output file open if using the "unload"
/*   function in tables because this function automatically opens and
/*   closes the file to which it writes.

&if [close %wfunit%] = 0 &then
   &type %.outfile% closed successfully

&sv loops = 1
&do &while %loops% le %count%
   /*  Update user on status.
   &type Processing watershed [value basin%loops%]

   select sector_cov.pat
   /** Reduce the selection to all of the polygons that are larger than one
   /** one grid cell.
   reselect sector_cov# = relfl//value
   reselect grid-code = [value basin%loops%]
   /** The unload command closes the file "hec.out."

   unload %.outfile% grid-code hrapx hrapy relfl//mean area ~
              delimited

/*   unload %.outfile% grid-code hrapx hrapy area relfl//mean relfl//max ~
/*              relfl//min relti//mean relti//max relti//min delimited

/*   unload %.outfile% grid-code hrapx hrapy area relfl//mean relfl//max ~
/*              relfl//min relfl//median relti//mean relti//max relti//min ~
/*              relti//median delimited

   &sv loops = %loops% + 1
&end   /*End of unloading real data.

/** Unload a list of the polygons (and their respective areas) that were
/** dropped during polygrid due to the fact that they had an area smaller
/** than the size of one grid cell.  To file "dropped.out"
   select sector_cov.pat
   reselect sector_cov# = relfl//value
   nselect
   unload dropped.out hrapx hrapy area

/*****************
/* Drop any "relates" before ending.
/*****************

relate drop
relfl
~
/*relate drop
/*relsb
/*~
/*relate drop
/*relti
/*~

/*&messages &on
/** Exit tables
quit
&return
```

9 MOUTPUT.F

```
/***************************************************************************
/***************************************************************************
/* Name: moutput.f
/*
/* Purpose: Translate data into the input file format for modClark
        program moutput
c       ** Modifies the ascii file created by hrap_int.aml to the form
c       ** requested by HEC.

        character sb*10, gc*10, e*4
c       ** dat(3) stores the first three data items for the current gridcell
c       ** dat(9) stores the last nine data items for the current gridcell
        integer count,dati(3)
c       ** with the median, the length of the datr array will  be 9 instead
c       ** of 7
        real datr(2)

        sb = 'SUBBASIN:'
        gc = 'GRIDCELL:'
        e = 'END:'


        open (unit = 20, file = 'tk3file.out', status = 'unknown')
        open (unit = 40, file = 'tk3modc.in', status = 'unknown')
        read(20,*) count
        read(20,*) dati, datr
        do 100 i = 1,count
          write(40,110) sb,dati(1)
 110      format (A,2x,I3)
          temp = dati(1)
c       ** If the first entry in the current row is different from the
c       ** first entry in the previous row, then the current cell
c       ** is in the same watershed and write the characteristics for the
c       ** grid-cell.
 115      if (dati(1).eq.temp) then

c***    Convert flowlength from meters to kilometers and
c***    area from meters^2 to km^2
          datr(1) = datr(1) / 1000.0
          datr(2) = datr(2) / 1000000.0
c       ** with median k = 1,9
            write(40,120) gc, (dati(j), j=2,3), (datr(k), k=1,2)
 120        format(A,1x,I3,1x,I3,1x,f8.4,1x,f7.4)
```

```
c         ** next line is the format to be used with median
c 120         format(I3,1x,I3,1x,f10.1,1x,3f9.1,1x,f8.0,1x,3f8.1,1x,f7.0)
              read(20,*,END=200) dati,datr

            goto 115
            endif
            write(40,*) e
 100      continue
 200      write(40,*) e

          end
```