PB99-155574

Final Report
K-TRAN Research Project KU-96-7

Utilization of Precipitation Estimates Derived from Composite Radar

by

C. Bryan Young
Bruce M. McEnroe
Rebecca J. Quinn

Department of Civil & Environmental Engineering
University of Kansas

for

Kansas Department of Transportation

January 1998

1. Report No.
K-TRAN: KU-96-7

2. Government Accession No.

3. Recipient Catalog No.

4 Title and Subtitle
UTILIZATION OF PRECIPITATION ESTIMATES DERIVED FROM COMPOSITE RADAR

5 Report Date
January 1998

6 Performing Organization Code

7. Author(s)
C. Bryan Young, Bruce M. McEnroe and Rebecca J. Quinn

8 Performing Organization Report No.

9 Performing Organization Name and Address
University of Kansas
Department of Civil and Environmental Engineering
Lawrence, Kansas 66045

10 Work Unit No. (TRAIS)

11 Contract or Grant No.
C-825

12 Sponsoring Agency Name and Address
Kansas Department of Transportation
Docking State Office Bldg.
Topeka, Kansas 66612

13 Type of Report and Period Covered
Final Report
May 1995 to January 1998

14 Sponsoring Agency Code
106-RE-0091-01

15 Supplementary Notes

16 Abstract
The National Weather Service river forecast centers produce grid-based estimates of hourly precipitation for NEXRAD radar data and precipitation measurements at selected gages. These NWS Stage III precipitation estimates can provide valuable information on the spatial and temporal distribution of storm rainfall over a watershed. The spatial resolution of the precipitation grid is four kilometers. State III data files for southern Kansas may be obtained from the archives of the NWS Arkansas-Red Basin River Forecast Center via the Internet. Stage III data files for the Missouri River basin, which includes northern Kansas, are produced by the NWS Missouri Basin River Forecast Center. These data files have been archived by our project team at the University of Kansas since July 1995.

An Arc/Info GIS application has been developed to analyze the Stage III precipitation data in a watershed context. This application, the KDOT Storm Analysis System, contains routines to develop precipitation contours and compute spatial-average precipitation over a watershed. It also includes routines to delineate watershed boundaries and calculate drainage areas, channel lengths and slopes from existing digital data sets. The project team has used this system to analyze several storms that caused flooding on Kansas highways.

17 Key Words
Drainage areas, geospatial, hydrologic, hypsography, precipitation, spatial, watershed

18 Distribution Statement
No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19 Security Classification (of this report)
Unclassified

20 Security Classification (of this page)
Unclassified

21 No. of pages
196

22 Price

Form DOT F 1700.7 (8-72)

r:\ktran\17KSU95.7doc

PREFACE

This research project was funded by the Kansas Department of Transportation K-TRAN research program. The Kansas Transportation Research and New-Developments (K-TRAN) Research Program is an ongoing, cooperative and comprehensive research program addressing transportation needs of the State of Kansas utilizing academic and research resources from the Kansas Department of Transportation, Kansas State University and the University of Kansas. The projects included in the research program are jointly developed by transportation professionals in KDOT and the universities.

NOTICE

DISCLAIMER

Acknowledgments

This project was supported by the Kansas Department of Transportation through the K-TRAN Cooperative Transportation Research Program. The authors sincerely appreciate this support. Special thanks are due to Mr. Jerome Younger, P.E., and Mr. Robert Reynolds, P.E., of KDOT for their contributions as project monitors.

Table of Contents

Chapter 1: Introduction 1
1.1 Background 1
1.2 Objective and Scope 1
1.3 The ARC/INFO GIS Package 2
Chapter 2: National Weather Service Stage III Precipitation Data 4
2.1 The Stage III Precipitation Product 4
2.2 Stage III Product Development 4
2.3 Limitations of Stage III Precipitation Estimates 7
2.4 Obtaining Stage III Data for Kansas 7
2.5 Using Stage III Data in a GIS Environment 8
2.5.1 Developing the HRAP Grid System in a GIS Environment 8
2.5.2 Reading the RFC Data Files 14
Chapter 3: Other Digital Geospatial Data for Hydrologic Studies 16
3.1 Digital Geospatial Data 16
3.2 The Data Access and Support Center 16
3.3 USGS Digital Line Graphs 17
3.3.1 Digital Line Graphs (1:100,000 scale) 17
3.3.2 Digital Line Graphs (1:2 million scale) 19
3.4 USGS Digital Elevation Models 19
3.4.1 Digital Elevation Models (1:24,000 scale) 20
3.4.2 Digital Elevation Models (1:250,000 scale) 21
3.5 NRCS Soil Surveys 22
3.6 KARS Land Cover 22
Chapter 4: Development of Geospatial Coverages for a Watershed with ARC/INFO GIS ... 23
4.1 Tools for Developing Geospatial Coverages 23
4.2 Setting Up the ARC/INFO Work Environment 23
4.3 Development of a Point Coverage 25
4.4 Development of 1:100k Digital Line Graph Coverages 28
4.4.1 Importing 1:100k Digital Line Graphs 28

4.4.2 Preprocessing Stream Coverages 31
4.5 Development and Processing of a Digital Elevation Model 34
4.5.1 Importing USGS Digital Elevation Models 34
4.5.2 Creating a Custom Digital Elevation Model 36
4.5.3 Hydrologic Analysis of DEMs 37
4.6 Development of a SSURGO Soils Coverage 41
4.6.1 Importing SSURGO Spatial Data 41
4.6.2 Processing SSURGO Attribute Data 42
4.7 Development of a Land Cover Coverage 43
4.8 User Control within the Menu System 43
Chapter 5: Analysis of Storm Rainfall over a Watershed 44
5.1 Rainfall Analysis Tools 44
5.2 Setting up the ARC/INFO Work Environment 44
5.3 Loading Rainfall Files to the Local Directory 44
5.4 Generating Hourly Precipitation Grids 45
5.5 Contouring the Precipitation Grids 50
5.6 Compiling Storm Statistics 52
5.7 Generating Rainfall Tables for One HRAP Cell 53
5.8 Practical Application for KDOT: Rainfall in Sedgwick County 53
Chapter 6: Summary and Conclusions 55
References 57
Appendix A 59
Appendix B: FORTRAN Source Code 61
Appendix C: ARC/INFO AML, Menu, and Projection Files 87

List of Tables

4-1 Listing of *indianlonlat.dat.* 25
4-2 Listing of indian statid.dat 26
4-3 Listing of flowgage.pri....................27
5-1 Listing of Sample VAT....................52
A-1 DASC Digital Data Sets....................59

List of Figures

2-1 WSR-88D Radar Umbrellas Covering Kansas 7
2-2 Profile View of Polar Stereographic Map Projection 10
2-3 Plan View of Polar Stereographic Map Projection 11
2-4 HRAP Grid over Douglas County, Kansas 14
3-1 Status Map for 1:100k DLGs of Hypsography 18
3-2 Sample of 1:100k DLGs 19
3-3 Status Map for 1:24k USGS DEMs 20
3-4 Areas of Zero Slope in the Lightning Creek 1:24K DEM after Filling of Sinks .. 21
4-1 Entering the Project Name 24
4-2 The Main and Data Preparation Menus 24
4-3 The DLG Preparation Menu 28
4-4 Choosing the DLG File to be Processed 32
4-5 Graphical Interface for Preprocessing Streams for Use with TOPOGRIDTOOL ... 33
4-6 The DEM Preparation Menu 35
4-7 The Process Dem Menu 37
4-8 USGS Stream Gage Offset from Flow Accumulation Network 39
4-9 Clipped and Filled DEM for Indian Creek 40
5-1 Entering the Storm Name 45
5-2 The Rainfall Analysis Menu 45
5-3 Defining the Storm 46
5-4 ARCEDIT Display for HRAP Coverage Generation 48
5-5 Grid of One Hour of Rainfall over the Bluff Creek Watershed 50
5-6 Contoured Storm Totals for the Bluff Creek Storm 51
5-7 Bar Graph of Rainfall Averages for Bluff Creek 53
5-8 Rainfall over Sedwick County from July 14, 1996, to September 1, 1996 ... 54

Chapter 1
Introduction

1.1 Background

The distribution of soils, land cover, and terrain features greatly affect watershed response to rainfall events. Traditional hydrologic models such as HEC1 (produced by the U.S. Army Corps of Engineers Hydraulic Engineering Center) model the spatial variation of these features by lumping, or averaging, parameters over portions of a watershed. Manual determination of these spatially averaged parameters can be difficult and time consuming.

Geographic Information Systems (GIS) are powerful computer software programs designed to store, manipulate, and analyze spatial data. The capabilities of these systems stand to greatly improve the accuracy, efficiency, and repeatability of hydrologic modeling.

The effectiveness of GIS to aid in describing watershed parameters hinges on the availability of data that is both inexpensive and accurate. During the last several years, federal and state government agencies have developed digital data sets describing many features relevant to hydrologic analysis. These data sets provide geospatial descriptions that may be used to develop input data for hydrologic models.

1.2 Objective and Scope

The objective of this research was to examine the use of digital geospatial data products for hydrologic study using the ARC/INFO GIS software package. The original goal was to develop tools to access, georeference, and analyze NEXRAD Stage III radar-based precipitation estimates. The scope of the project was later expanded to include other data sets that are useful for hydrologic studies. These data sets include United States Geological Survey (USGS) Digital Line Graphs of roads, elevation contours, and hydrologic features; USGS Digital Elevation Models; Natural Resources Conservation Service (NRCS) soil surveys; and a land-cover data set developed by the Kansas Applied

Remote Sensing program (KARS). Procedures to import and prepare each data set for hydrologic analysis were developed, and each data set was evaluated with respect to the needs of the Kansas Department of Transportation.

1.3 The ARC/INFO GIS Package

This project used the ARC/INFO GIS software of the Environmental Systems Research Institute (ESRI) on a Sun SPARC 20 UNIX workstation. This project uses ARC/INFO 6.0, available only for UNIX workstations. The full ARC/INFO software, version 7.1.1, is now also available for the Windows NT operating system. The ARC/INFO package was selected for this research for its versatile and powerful array of analysis tools. ARC/INFO consists of several integrated modules. The modules utilized in this research are ARC, GRID, TABLES, ARCEDIT, TIN, ARCPLOT, and AML.

The ARC module is the primary component of the ARC/INFO package. ARC contains a wide array of functions for importing, storing, retrieving, exporting, and analyzing spatial data. Analysis functions contained in ARC operate in the vector environment. The ARC vector environment consists of point, line, and polygon spatial features grouped in data layers called coverages. ARC relates descriptive information, called attributes, to the spatial features of a coverage through an attribute table, called point, arc, or polygon attribute tables (PAT, AAT, or PAT) depending on the nature of the coverage. Analysis in the vector environment is performed utilizing topology, or the spatial relationship between features in a coverage.

One of the attractive features of ARC/INFO is the availability of the GRID module. GRID is a full featured raster, or cell-based, GIS integrated with the ARC/INFO package. Raster GIS stores spatial data as a regular grid of numeric data values. GRID contains editing and analysis functions, including several built-in hydrologic operators. The TABLES module provides functions for performing operations on database files. ARC spatial coverages may be linked to several third-party database software systems or to data stored in INFO, the native ARC/INFO database. Point, arc, and polygon attribute tables are stored in the INFO database. TABLES provides functions necessary to establish links between, modify, analyze, and create database files.

ARCEDIT contains editing functions for ARC vector coverages. Commands for removing, adding, or editing points, lines, and polygons are available. ARCEDIT also serves as the digitizing interface for manual data input.

TIN provides tools for creating, analyzing, and storing triangulated irregular networks (TINs). TINs are composed of points connected by lines to form planar triangles. TIN commands are executed from the ARC prompt.

ARCPLOT is the graphics display module of ARC/INFO. ARCPLOT may be used to display data coverages or to create maps. Displays for this project were developed using the ARC/INFO **ARCTOOLS** graphical interface to ARCPLOT mapping functions.

The ARC Macro Language (or AML) is fully integrated with the ARC/INFO system. AML scripts (called AMLs) can execute ARC/INFO functions, submit commands to the operating system, and perform procedural program tasks. This project used AMLs extensively to automate repetitive operations.

Chapter 2

National Weather Service Stage III Precipitation Data

2.1 The Stage III Precipitation Product

The National Weather Service (NWS) River Forecasting Centers (RFCs) produce hourly estimates of precipitation amounts on a four-kilometer grid. These Stage III precipitation estimates are developed primarily from NEXRAD radar reflectivity data. The RFCs use these precipitation estimates as inputs to their river forecasting models. Two RFCs produce Stage III precipitation estimates for Kansas: the Missouri Basin RFC (MBRFC), which covers roughly the northern half of the state, and the Arkansas-Red Basin RFC (ABRFC), which covers the southern half.

2.2 Stage III Product Development

The National Weather Service (NWS) has operated a system of Weather Surveillance Radars (WSRs) for several decades. The early radars were operated manually, and were concentrated in the eastern and central United States. Over the past decade, these radars have been replaced by 1988 Doppler WSRs (WSR-88D). This new system of radars is also termed the Next Generation Weather radar (NEXRAD) system. The WSR-88D radars units have better resolution, better volume scanning, and greater sensitivity than the older WSR-57/74 radars (INTELLICAST 1996). The WSR-88D radar detects reflectivity and velocity on a one-kilometer grid (OSF 1996). Rainfall estimates are generated from the reflectivity data. The WSR-88D radar has an effective range of 230 kilometers. At the end of 1997, 137 WSR-88 units were operational. The umbrellas of these radars cover 95% of the land area of the coterminous United States.

The WSR-88D radar units collect data in one of three different modes, according to weather conditions. In the clear-air mode, the radar scans the atmosphere at five different elevation angles between 0.5 degrees and 4.5 degrees every ten minutes (INTELLICAST 1996). When the unit detects significant precipitation anywhere within its umbrella, it switches to the precipitation mode (OSF 1996). The radar performs nine scans every six minutes, at elevation angles from 0.5 to 19.5 degrees, in the precipitation mode. The third mode of operation is the severe weather mode, in which the unit completes fourteen scans every five minutes, at elevation angles from 0.5 to 19.5 degrees (INTELLICAST 1996). Radar scanning is performed by the Radar Data Acquisition (RDA) component of the WSR-88D radar, which relays the base reflectivity and velocity information to the unit's Radar Products Generator (RPG) for Stage I processing.

The precipitation estimates are developed from the NEXRAD radar data in three stages. The first two stages of data processing are performed automatically within the Radar Product Generator (RPG) component of the WSR-88D unit. The NWS River Forecast Centers (RFCs) produce the final Stage III precipitation estimates.

NEXRAD Stage I data processing converts WSR-88D base reflectivity readings to hourly rainfall estimates. This stage of processing is completed automatically within the WSR-88D RPG. As a first step in developing the Stage I product, the RPG corrects individual radar scans for bad or missing data, radar beam heights, partially obstructed beams, and range effects. The RPG then converts the corrected scans to rainfall rates using a reflectivity conversion equation. The basic equation is

$$Z_e = 300 \cdot R^{1.4} \qquad (2\text{-}1)$$

in which R is the rainfall rate in mm/hr and Z_e is the equivalent reflectivity factor in decibels. The RPG adds the resulting rainfall estimates to a running hourly total and checks that total for unreasonable rainfall rates and accumulations. In the future, a uniform multiplicative bias correction will be applied to the Stage I precipitation estimates. The bias correction factor will be computed from data transmitted from recording precipitation gages within the radar umbrella, according to the formula

$$\textit{bias correction factor} = \frac{\textit{average precipitation within umbrella from rain gages}}{\textit{average precipitation within umbrella from radar}} \qquad (2\text{-}2)$$

Currently, the WSR-88D radars are not equipped to receive data transmitted from rain gages, so the Stage I precipitation estimates are not corrected for bias. The Stage I product is used as input to the second level of NEXRAD processing (ABRFC 1996).

The Stage II process generates hourly precipitation estimates on a four-kilometer square grid, called the Hydrologic Rainfall Analysis Project (HRAP) grid, within the umbrella of the WSR-88D radar. The inputs to the Stage II calculations are the Stage I precipitation estimates, rain-gage data and surface temperature data. First, any bias applied during the Stage I analysis is removed so that more rain-gage data can be used. A new uniform bias is calculated for Stage II. Biases are computed for certain rain gages within the umbrella. The total radar bias is the average of the biases for the individual gages. Unlike Stage I processing, it uses data from preceding hours, including updates from rain gages, to create a product that agrees better with measured rainfall. The biases from the previous five hours are used in the computation of the current hour's bias. This process tracks underestimation and overestimation by the radar and keeps the bias close to 1.00 after correction.

The Stage III product is a mosaic of Stage II precipitation estimates from each WSR-88D radar in the RFC's region. The precipitation estimates may be revised to better match rain-gage data. Each day, the twenty-four-hour composite rainfall estimates are compared to observed values. If the overall bias is significant, the Stage III estimates are recalculated with a new bias correction. Fig. 2-1 shows the parts of the Missouri and Arkansas River basins within Kansas, with WSR-88D radar sites and umbrellas. The irregular line that splits the state roughly in half is the divide between the Missouri and Arkansas River basins.

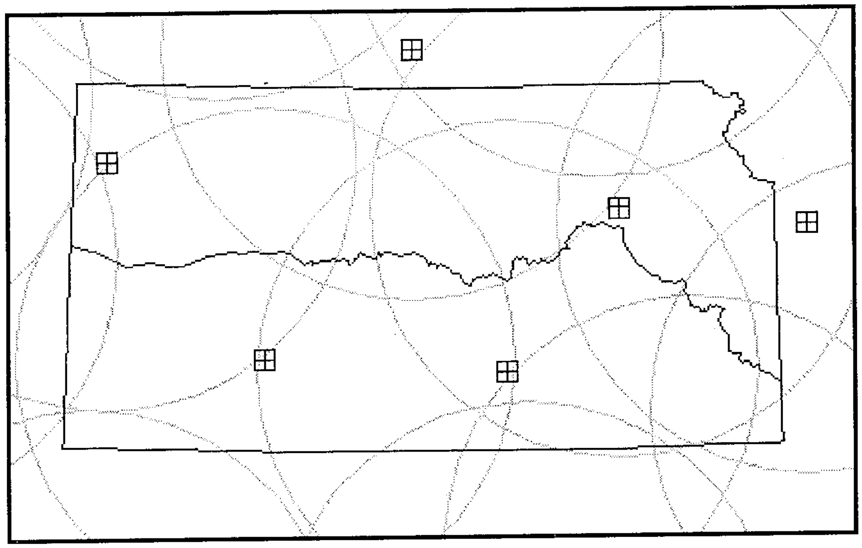


Figure 2-1: WSR-88D Radar Umbrellas Covering Kansas.

2.3 Limitations of Stage III Precipitation Estimates

Stage III precipitation estimates are better for rainfall than for other forms of precipitation. WSR-88D radars have difficulty detecting very small rain drops or snow; precipitation estimates for these conditions tend to be too low. For storms with large hail, the Stage III estimates tend to be too high. Range-related errors can be significant. The Stage I precipitation estimates from individual radar units tend to be too low within about 40 km of the radar and beyond 150 km from the radar (Smith et al. 1996). The algorithms used to produce Stage III precipitation estimates are still evolving, and will improve in the future. Fortunately, the NEXRAD system has been designed to allow both hardware and software improvements.

2.4 Obtaining Stage III Data for Kansas

The Arkansas-Red Basin RFC is currently the only NWS RFC distributing NEXRAD Stage III products to the public. The ABRFC produces hourly, six-hour, and twenty-four-hour and monthly composites of NEXRAD Stage III data, each available as a file or as a graphical image. Hourly Stage III data files are available from June 24, 1994, to the present. In addition, the ABRFC produces daily 24 frame movies (MPEGs) of precipitation over the Arkansas-Red Basin. All Stage III data products are easily accessed via the ABRFC Internet home page (http://info.abrfc.noaa.gov/).

The Missouri Basin RFC currently produces Stage III estimates, but does not possess the capability to distribute these files to the public. The MBRFC has been providing this research effort with Stage III hourly rainfall files via automatic FTP (File Transfer Protocol) since June 10, 1995. These files are archived on project computers and backup tapes.

2.5 Using Stage III Data in a GIS Environment

Using Stage III rainfall data in a GIS environment involves three steps. First, a coverage of the Hydrologic Rainfall Analysis Project (HRAP) grid must be developed in the GIS. Second, the data in the RFC Stage III files must be extracted with computer programs developed specifically for this purpose. Third, the rainfall data contained in those files must be linked to the HRAP grid in the GIS. This section describes the first two steps in this process, while Chapter 5 describes the tools developed to accomplish the third step.

2.5.1 Developing the HRAP Grid System in a GIS Environment

NEXRAD Stage III rainfall estimates are referenced to the Hydrologic Rainfall Analysis Project (HRAP) grid system. This system consists of a rectangular grid imposed on a polar stereographic map of the Northern Hemisphere. The polar stereographic map projection projects features from the Earth's surface onto a plane passed through the Earth at a given latitude, called the standard latitude, using an imaginary light source placed at the South Pole (Greene and Hudlow 1982). Figure 2-2 on page 10 demonstrates the polar stereographic projection. In this drawing, a feature at E is projected from a light source at C to B on the image plane. The standard latitude for the HRAP system is $\phi_0 =$ 60° North. At this latitude, the grid-cell length is exactly 4.7625 kilometers. The HRAP grid also specifies a standard projection longitude of $\lambda_0 = -105°$ (Reed 1995).

The HRAP coordinates of any point in the United States may be calculated using a series of conversion equations. These equations consider the Earth to be a geo-centered sphere with a radius of 6371.2 kilometers. In this approximate system, the longitude and latitude are referred to as the geocentric coordinates. The geocentric latitude differs slightly from the geographic (or geodetic) latitude, which is based on an ellipsoidal representation of the Earth. Figures 2-2 and 2-3 demonstrate the geometry of the transformation equations in the geocentric system (Reed 1995).

Equations 2-3 through 2-5 convert the geocentric coordinates of a point to polar stereographic coordinates

$$R = r \cdot \cos\left(\phi_g \frac{(1+\sin\phi_0)}{(1+\sin\phi_g)}\right) \tag{2-3}$$

$$x = R \cdot \cos(\lambda - \lambda_0 - 90^\circ) \tag{2-4}$$

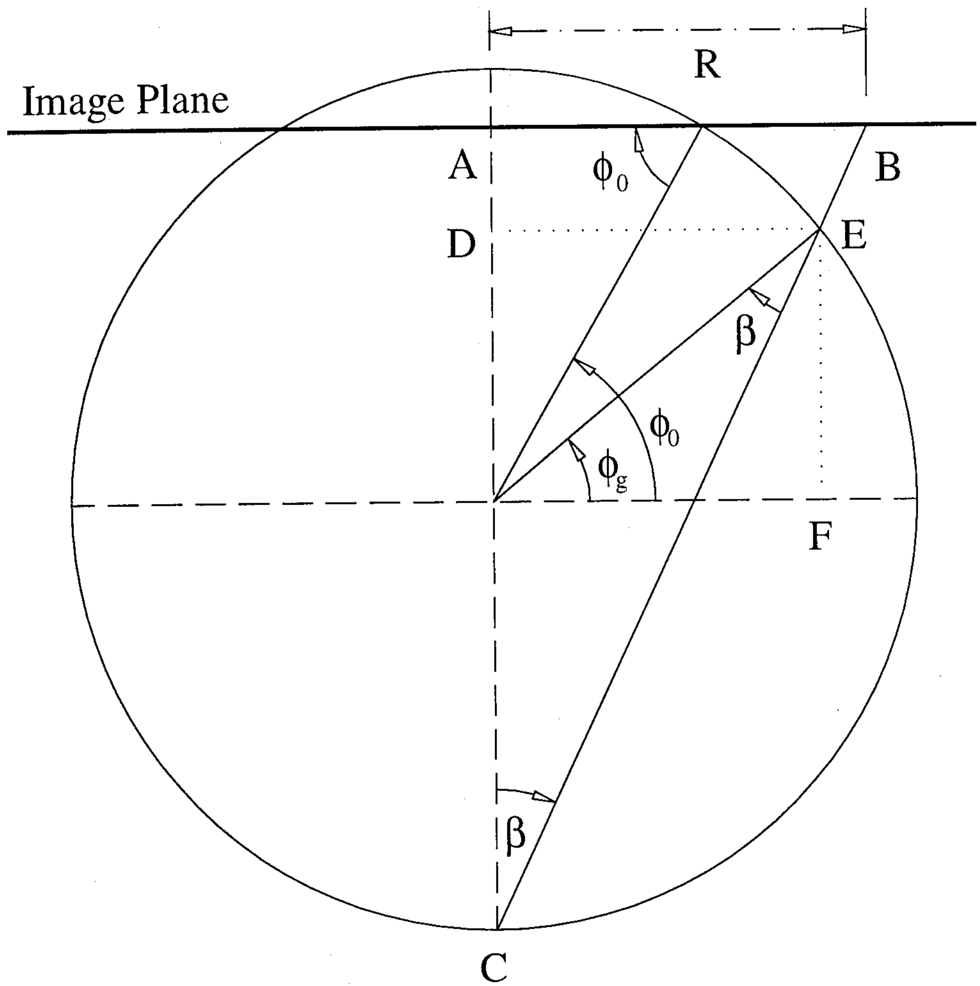
$$y = R \cdot \sin(\lambda - \lambda_0 - 90^\circ) \tag{2-5}$$

in which r is the radius of the Earth (km), R is the distance on the image plane to the axis (km), ϕ_0 is the standard latitude (60° North), ϕ_g is the geocentric latitude, λ_0 is the standard longitude (-105°), λ is the longitude, and x and y are the polar stereographic coordinates (km). The HRAP origin is defined so that all coordinates in the United States are positive, resulting in the coordinates of the North Pole being (401, 1601). Equations 2-6 and 2-7 convert the polar stereographic coordinates of a point to HRAP coordinates.

$$HRAPX = \frac{x}{4.7625} + 401 \tag{2-6}$$

$$HRAPY = \frac{y}{4.7625} + 1601 \tag{2-7}$$

These equations form the basis for the development of an HRAP coverage in GIS.



R = Distance from the pole to the location of the projected point in the image plane.

ϕ_0 = Standard Latitude (60^0 North).

ϕ_g = Geocentric Latitude.

Figure 2-2: Profile View of a Polar Stereographic Map Projection.

(Adapted from Reed, 1995).

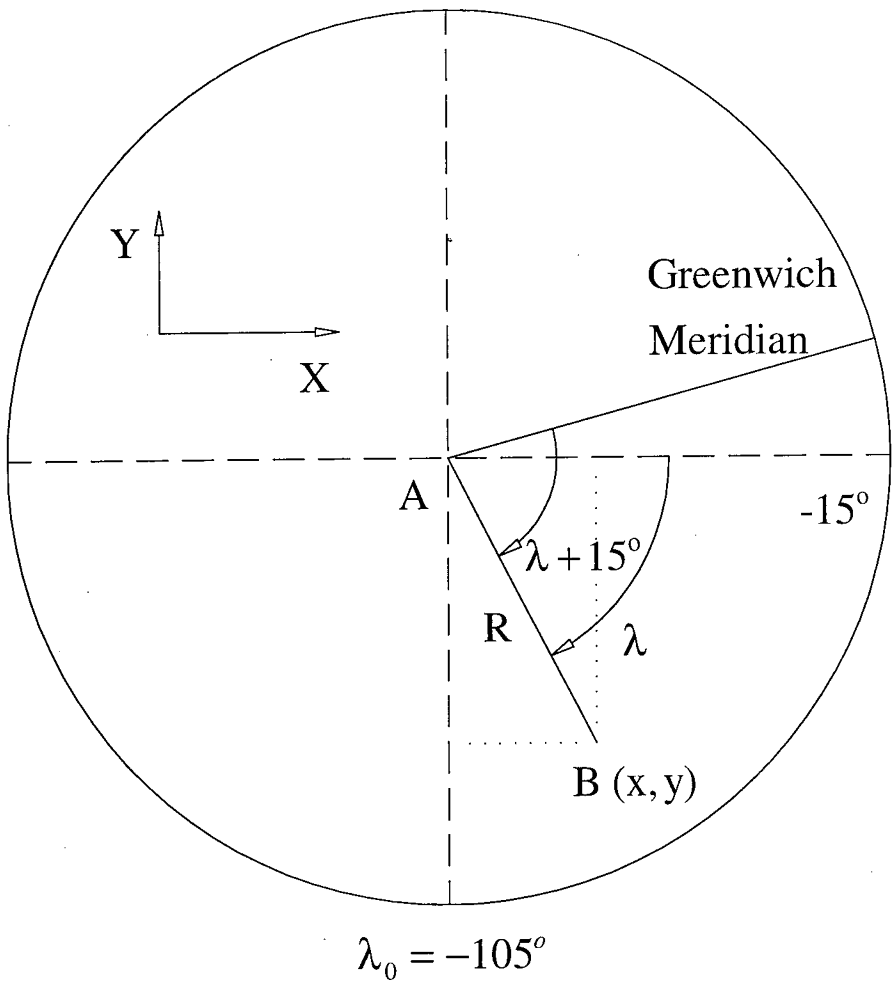


Figure 2-3: Plan View of a Polar Stereographic Map Projection.

(Adapted from Reed, 1995.)

Seann Reed of the University of Texas at Austin has developed a standard procedure for developing HRAP polygon coverages in ARC/INFO GIS (Reed 1995). This procedure uses two programs: *genhrap.for* (Appendix B) and *genhrap.aml* (Appendix C). The *genhrap.for* FORTRAN program generates a text file containing the geographic coordinates of the HRAP cells in a specified area in a form suitable for input to ARC/INFO. The program first prompts the user to define the extent of the HRAP grid desired by entering the geocentric coordinates of the four corners of the study area or by entering the HRAP coordinates of the lower left-hand corner of the area and the number of rows and columns desired. The program then creates a list of the HRAP coordinates contained in the specified area.

To calculate the geographic coordinates of each HRAP cell, *genhrap.for* first converts the HRAP coordinates to geocentric coordinates using Equations 2-3 through 2-7 in reverse (subroutine *w11* performs this transformation). Equations 2-8 through 2-10 demonstrate the computation of the polar stereographic coordinates (x,y) and the distance from the North Pole (R).

$$x = 4.7625 \cdot (HRAPX - 401) \tag{2-8}$$

$$y = 4.7625 \cdot (HRAPY - 1601) \tag{2-9}$$

$$R = \sqrt{x^2 + y^2} \tag{2-10}$$

The longitude angle λ' (from 0° to 360°) from the Greenwich Meridian to the location of (x,y) is calculated as follows:

If y > 0, then

$$\lambda' = 270^\circ - \lambda_0 - \tan^{-1}\left(\frac{y}{x}\right)$$

If y < 0, then

$$\lambda' = -90^\circ - \lambda_0 - \tan^{-1}\left(\frac{y}{x}\right) \tag{2-11}$$

This angle is converted to the standard convention of west longitude values ranging from 0° to -180° and east longitude values ranging from 0° to 180° using Equation 2-12.

If $\lambda' < 180°$, then

$$\lambda = -\lambda'$$

If $\lambda' > 180°$, then

$$\lambda = 360° - \lambda' \qquad (2\text{-}12)$$

The geocentric latitude of the polar stereographic coordinates is calculated using the following equation:

$$\phi_g = 90° - 2 \cdot \arctan\left[\frac{R}{r \cdot (1 + \sin \phi_0)}\right] \qquad (2\text{-}13)$$

After the subroutine *wll* calculates the geocentric coordinates, the subroutine *togeod* converts the geocentric coordinates to geodetic coordinates using standard conversions (Snyder 1987). The subroutine *topoly* writes the geodetic coordinates to a file in a form appropriate for generating a polygon coverage in ARC/INFO (Reed 1995).

This FORTRAN output file is the primary input to *genhrap.aml*, an ARC Macro Language (AML) script. This AML script executes the **GENERATE** command in ARC to develop the polygon coverage of the HRAP grid. **GENERATE** reads the output file from *genhrap.for* and creates the HRAP polygon coverage. The AML script then attaches the HRAP coordinates to the polygon attribute table of the new coverage and projects the coverage from geographic coordinates to the desired map projection (Reed 1995).

The *genhrap* AML script and FORTRAN programs were used to develop an HRAP coverage of the state of Kansas. Figure 2-4 displays a portion of this coverage over Douglas County, Kansas.

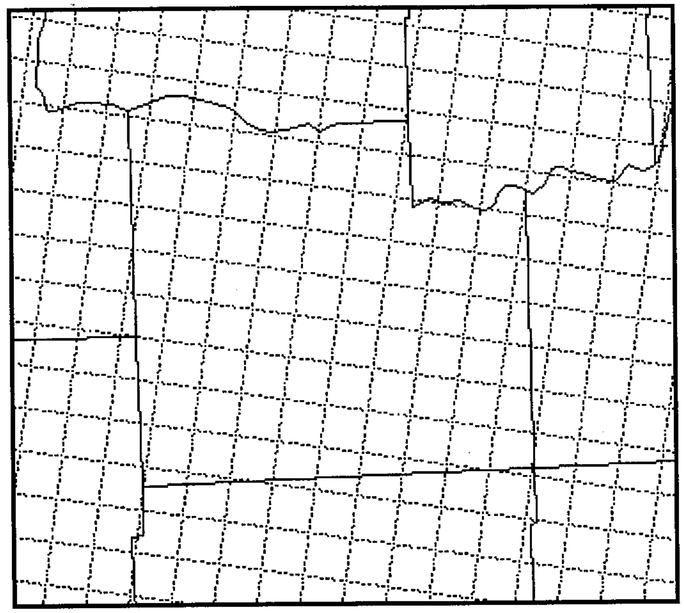


Figure 2-4: HRAP Grid over Douglas County, Kansas.

2.5.2 Reading the RFC Data Files

The ABRFC and MBRFC produce Stage III precipitation data in different file formats. These formats require two different processing procedures to make the data accessible to ARC/INFO.

The ABRFC hourly rainfall data files are stored in the netCDF (network common data form) file format, an array-oriented file system commonly used for the sharing of atmospheric data. The netCDF system consists of FORTRAN, C, C++ and Pearl interfaces to the netCDF file format. The use of programming language interfaces makes the netCDF format machine-independent. These interfaces contain netCDF function libraries, which are available freely over the Internet from UNIDATA (http://www.unidata.ucar.edu/packages/netcdf). These functions are implemented in program source code to open, read, write, and close netCDF files (UNIDATA 1996). For example, the *readabrfc.for* program in Appendix B uses the netCDF FORTRAN interface to load rainfall information from an ABRFC Stage III file into a FORTRAN array.

The NEXRAD Stage III data files from the MBRFC arrive as a bundled 'tar' file containing four days of compressed individual hourly rainfall files. The UNIX commands **tar** and **uncompress** make the hourly files accessible.

% **tar -xvf** ***mb0712.tar***

% **uncompress** ***xmrg07119620z***

The MBRFC stores the individual Stage III data as FORTRAN arrays in unformatted files. The program *readmbrfc.for* in Appendix B demonstrates how the MBRFC files are read into a FORTRAN array. The MBRFC rainfall arrays contain Stage III estimates for the entire Missouri River Basin. In order to save disk space, the rainfall data for Kansas have been extracted from these files using the AML script *monthly2.aml* (Appendix C) and FORTRAN program *reducembrfc.for* (Appendix B). The *monthly2.aml* script prompts the user for a start and end date (usually the first and last days of the month), executes the UNIX **tar** and **uncompress** commands on the MBRFC files, generates a list of the Stage III file names, stores that list to a file, and calls the *reducembrfc* program. The *reducembrfc* program extracts the array of rainfall values that covers Kansas from each Stage III file and saves these values as a new file. The original Stage III files are compressed again by the AML script to save space, while the reduced files are copied to /files/mbrfc/archives/reduced/. The program *readreduced.for* in Appendix B demonstrates how the reduced files are accessed in FORTRAN. Chapter 5 explains how the FORTRAN array of rainfall values is imported into ARC/INFO.

Chapter 3

Other Digital Geospatial Data for Hydrologic Studies

3.1 Digital Geospatial Data

Geographic information systems require the input of geospatial data in digital form. Information contained in maps, aerial photography, or other hard-copy product must either be digitized or scanned into a usable GIS format. Scanning and digitizing can be costly and time-consuming but are occasionally the only means of obtaining the required data. Often, however, digital forms of useful data already exist.

Over the past several years, government agencies have converted many national, state, and local data sets into digital form. These data sets contain information relevant to hydrologic analysis, including elevation information, location and description of hydrologic features, soil characteristics, and the type of land cover or use. Such information is readily accessible to GIS users in Kansas through the Data Access and Support Center.

3.2 The Data Access and Support Center

The State of Kansas Geographic Information Systems Policy Board established the Data Access and Support Center (DASC) in 1991 in order to facilitate state agency access to existing digital data sets. DASC is state-funded and is operated by the Kansas Geological Survey. DASC performs the following five services:

- Receive, archive, and catalog all core databases. Maintain associated documentation.
- Check and verify integrity of data to ensure they meet GIS Policy Board's database standards.
- Convert and transform databases to different software formats, computer architectures, coordinate unit measurements, and projection systems.
- Distribute databases as requested and handle inquiries for DASC services.
- Promote and assist the use of the core database and GIS technologies.

DASC currently administers over 30 digital data sets as part of the Kansas GIS Core Database. This database consists of a collection of geospatial data produced by federal, state, and local agencies as part of the Kansas GIS Initiative (DASC 1994). The data sets in the core database are listed in Table A-1 in Appendix A.

Of the data sets administered by DASC, four are of primary interest to the hydrologist: United States Geological Survey (USGS) Digital Line Graphs (DLGs), USGS Digital Elevation Models (DEMs), soil databases produced by the Natural Resources Conservation Service (NRCS), and a land cover data set developed by the Kansas Applied Remote Sensing (KARS) center. These data sets are available through anonymous file transfer protocol (ftp) over the Internet at http://gisdasc.kgs.ukans.edu. A detailed description of each data set follows.

3.3 USGS Digital Line Graphs

The United States Geological Survey provides digital versions of 1:24k, 1:100k and 1:2 million scale USGS topographic map sheets in Digital Line Graph (DLG) format. This format divides USGS map sheets into multiple DLG files, each representing one class of geographic features (e.g., elevation contours, hydrologic features, or political boundaries). DLGs contain attribute information in the form of codes.

DLG tiling schemes differ depending on scale, but the edges of all DLGs have been matched to ensure the compatibility of neighboring map sheets (USGS 1989, 1990, and 1994). The availability and accessibility of DLGs from DASC also varies according to scale; 1:24k DLGs are available for a limited portion of the state and were not investigated as part of this report.

3.3.1 Digital Line Graphs (1:100,000 scale)

The USGS produces seven layers of 1:100k scale digital line graphs from 30-minute by 60-minute topographic quadrangles. The seven layers contain boundaries (political and administrative), hydrology (streams and lakes), hypsography (elevation contours), public land survey system, railroads, roads, and miscellaneous (pipelines, transmission lines, and miscellaneous transportation) (USGS 1989). DLG coverage for the state of Kansas is complete at the 1:100k scale for all data layers except hypsography, boundaries and public land surveys. Figure 3-1 shows the availability of hypsography data in mid-1997. 1:100k DLGs of Kansas are available free of charge through anonymous FTP (File Transfer Protocol) from the DASC server.

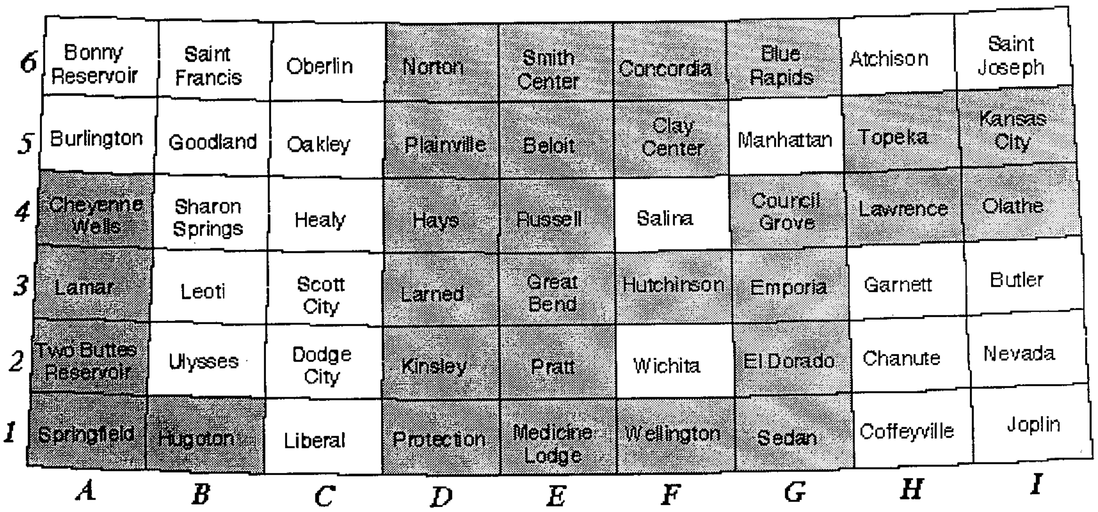


Figure 3-1: Status Map for 1:100k DLGs of Hypsography (Gray = Available). (DASC, 1997)

Of the seven DLG 1:100k data layers, the hypsography, hydrology, and roads layers are indispensable to hydrologists using GIS. These data provide the means to locate accurately a point of interest, identify the locations of streams and water bodies, and define the land surface of an area. Attribute information attached to these layers includes contour elevations for the hypsography layer (typically 5, 10, or 20 meter intervals), road classes and types, and types of water bodies. Figure 3-2 displays hypsography and roads layers in combination.

Figure 3-2: Sample of 1:100k DLGs.

3.3.2 Digital Line Graphs (1:2 million scale)

USGS 1:2 million scale DLGs provide five data layers: administrative boundaries and cultural features, counties, hydrology, roads, and railroads. DASC supplies USGS 1:2 million scale DLGs for the entire state; all are available via anonymous FTP. These small-scale DLGs are of limited use in hydrologic modeling, though they provide an excellent means for referencing locations in the state.

3.4 USGS Digital Elevation Models

In addition to the contoured elevation data provided by 1:100k scale hypsography DLGs, the USGS produces Digital Elevation Models (DEMs) at the 1:250k and 1:24k scales. A digital elevation model is a regular grid of elevation data points and is an efficient method for describing the land surface. Digital elevation models can be used within the ARC/INFO GRID module to calculate slopes, delineate watershed boundaries, and calculate other important hydrologic properties (see Section 4.5.3).

3.4.1 Digital Elevation Models (1:24,000 scale)

The USGS produces 1:24k DEMs from 7.5-minute quadrangle sheets. Data points for this scale are spaced regularly at 30 meters in Universal Transverse Mercator (UTM) coordinates (USGS 1994). 1:24k DEMs are available for a limited portion of Kansas in 7.5-minute blocks, as shown in Figure 3-3, upon request from DASC.

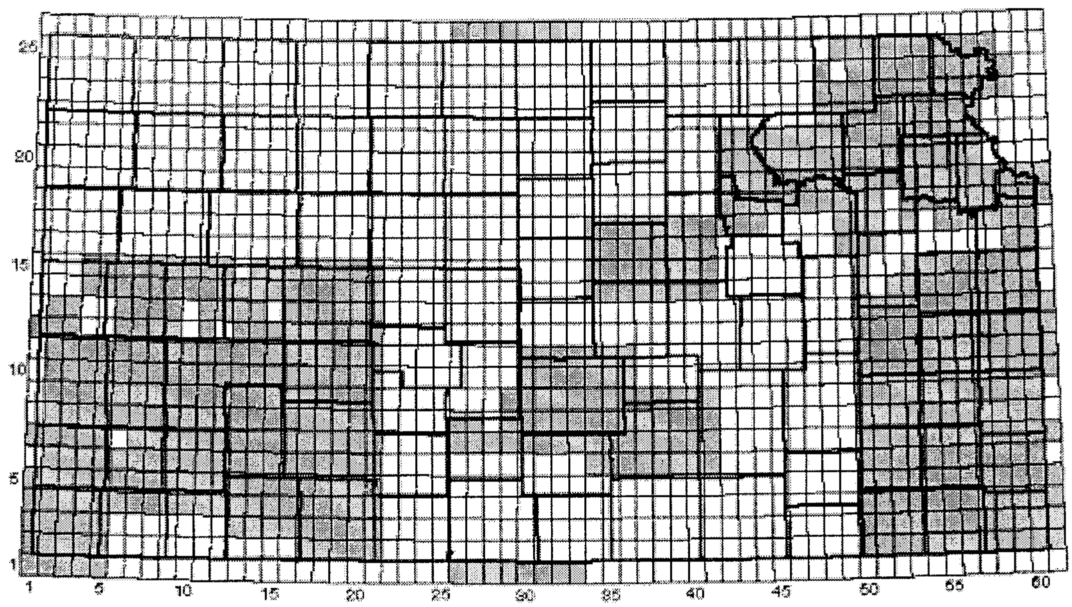


Figure 3-3: Status Map for 1:24k USGS DEMs (Shaded = Available).

(DASC, 1997)

1:24k DEMs are generally not edge-matched, meaning that map edges are apparent when multiple tiles are joined. The resulting systematic elevation differences may cause analytical errors when multiple map sheets are joined. In addition, 7.5-minute DEMs contain numerous sinks. A sink is a region of the DEM that has no flow outlet. Sinks are generally considered to be errors resulting from the sample spacing of the DEM, and may be filled to the level of the lowest neighboring cell to create a "hydrologically correct" DEM (ESRI 1995). ARC/INFO assigns fill areas a zero slope, which can cause problems in modeling, especially when large sinks are present in the

DEM (see Figure 3-4). 1:24k DEMs provide a detailed description of the land surface, but they may not be suitable for hydrologic analysis due to the prevalence of large sinks.

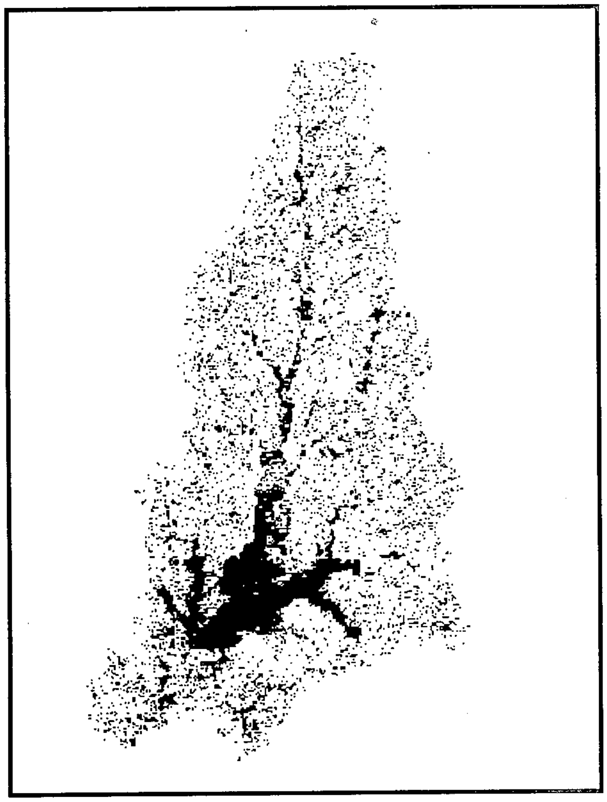


Figure 3-4: Areas of Zero Slope in the Lightning Creek Watershed 1:24k DEM after Sinks have been Filled.

3.4.2 Digital Elevation Models (1:250,000 scale)

The United States Geological Survey distributes 1:250k digital elevation models produced by or for the Defense Mapping Agency. Data point spacing is 3 arc-seconds in geographic coordinates, or about 100 meters depending on latitude (USGS 1994). 1:250k DEMs (also called 3-arc-second or 1-degree DEMs) are available from DASC in 1-degree square blocks via anonymous FTP.

Although 1:250k DEMs are edge-matched, the sample spacing of 100 m often results in the formation of large sinks. Three-arc-second DEMs may be used to delineate and describe large watersheds, but the use of these DEMs in hydrologic modeling requires caution.

3.5 NRCS Soil Surveys

The Natural Resources Conservation Service has developed three geospatial soil databases: the National Soil Geographic (NATSGO), State Soil Geographic (STATSGO), and the Soil Survey Geographic (SSURGO) data sets. These three data sets vary in detail and intended scale of use, with the SSURGO database being the most detailed and the most suited for use in hydrologic modeling of small watersheds. The NATSGO data set is intended for use at the national or regional scale, and is not discussed in this report.

The STATSGO and SSURGO databases consist of 22 different attribute tables and a polygon coverage of delineated map units, each uniquely identified by a map unit identification number (muid). The muid relates the geospatial description to the database files containing attribute information about that map unit. Attribute data provided for each soil layer in the map unit include particle size distribution, soil texture, bulk density, available water capacity, soil reaction, salinity, and organic matter content (NRCS 1995). The STATSGO database for Kansas is available via anonymous FTP; SSURGO data sets are supplied upon request. SSURGO spatial data are tiled by 7.5-minute blocks, and SSURGO attribute data are tiled by county. Both coverages are edge-matched.

3.6 KARS Land Cover

One digital data set exclusive to Kansas is a land-cover survey performed by the Kansas Applied Remote Sensing center. This survey interpreted LANDSAT Thematic Mapper multi-spectral images collected 1988-1990 using supervised remote sensing classification techniques. The survey classifies all of Kansas into 10 land-cover categories: residential, commercial/industrial, urban grasslands, urban woodlands, urban water, cropland, grasslands, woodlands, water, and other (DASC 1994). No update of the survey is planned. The KARS land-cover series is tiled by county, and tiles are edge-matched. It is available from DASC via anonymous FTP.

Chapter 4

Development of Geospatial Coverages for a Watershed Within ARC/INFO GIS

4.1 Tools for Developing Geospatial Coverages

The development of geospatial coverages for a watershed requires three general processing steps: importing the data layer into the GIS as a coverage or grid, projecting the data coverage into a common map projection, and manipulating the data coverage to obtain the desired information. ARC Macro Language scripts have been developed to perform these three processing steps on the four primary data sets discussed in Chapter 3. The AML scripts for processing geospatial data can be executed through a menu-driven Graphical User Interface (GUI) developed for this project. This chapter explains the AML script called by each menu option in the GUI. Coverages for the Indian Creek watershed in Johnson County, Kansas, are presented as examples. Complete listings of the AML scripts are provided in Appendix C.

4.2 Setting Up the ARC/INFO Work Environment

The GUI menu system is initialized by specifying the location of the interface AML files and executing *main.aml*.

Arc: **&amlpath/files/arcfiles/interface**

Arc: **&run main.aml**

The *main.aml* script prompts the user for the project name, as shown in Figure 4-1. In this application, the project name should not exceed six characters. Once the project name is defined, *main.aml* displays the main menu, as shown in Figure 4-2. The data preparation menu, also shown in Figure 4-2, is accessed from the main menu.

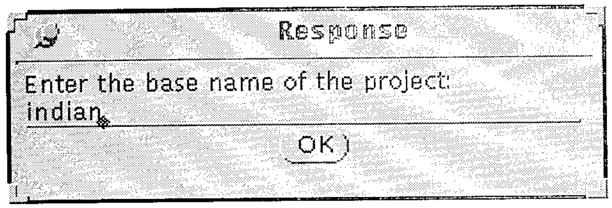


Figure 4-1: Entering the Project Name.

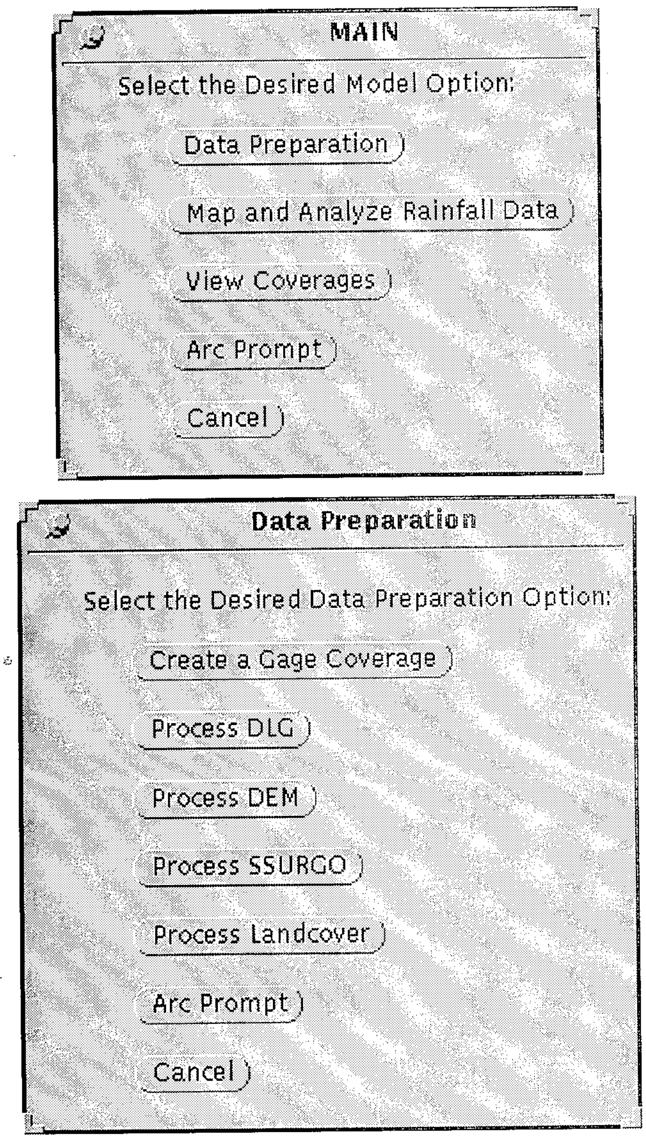


Figure 4-2: The Main and Data Preparation Menus.

4.3 Development of a Point Coverage

It is often important to develop a GIS data layer representing bridges, stream gages, or other key points in or near a study area. The *pointinput.aml* script creates such a layer by generating a point coverage from geographic coordinate data, adding attribute information to the points, and then converting the coverage into the map projection selected for the study.

The ARC **GENERATE** command produces an ARC/INFO coverage from coordinate information entered at the command line or as a text file. The *pointinput.aml* script creates an input file for **GENERATE** by prompting the user for point coordinates and then writing them to a file. Table 4-1 shows the coordinate file for the USGS stream gages on Indian Creek and the Blue River in Kansas.

Table 4-1 Listing of *indianlonlat.dat*.

```
1 -94.6694 38.9417
2 -94.6753 38.8125
End
```

The first column in Table 4-1 is an arbitrary number assigned to the points for purposes of identification in ARC/INFO. The second and third columns are the longitude and latitude of the gage in decimal degrees. After *pointinput.aml* generates the point coverage, it builds point topology.

Arc: **generate *indianstats***

Generate: **input *indianlonlat.dat***

Generate: **points**

Generate: **quit**

Arc: **build *indianstats* points**

After building the topology, *pointinput.aml* defines attribute information for the point coverage in two ways. First, the **ADDXY** command adds the geographic coordinates of the points to the attribute table.

Arc: **addxy *indianstats* points**

Second, *pointinput.aml* adds the identifier of the point to the coverage. In this example, the point identifiers are the numbers and names of the USGS stream gages. Again, *pointinput.aml* prompts the user for the required information and saves it to a text file. Table 4-2 displays the contents of the Indian Creek attribute file.

Table 4-2 Listing of indianstatid.dat.

1 06893300 'Indian Creek'
2 06893080 'Blue River'

In order for the attribute information in the text file to be joined to the point coverage, the contents of this file must be moved to an INFO data file. The INFO table must first be defined.

Arc: **tables**

Enter Command: **define *indian.dat***

Item Name: ***indianstats-id***

Item Width: **4**

Item Output Width: **4**

Item Type: **I**

Item Name: ***stat-num***

Item Width: **10**

Item Output Width: **10**

Item Type: **C**

Item Name: ***stat-name***

Item Width: **15**

Item Output Width: **15**

Item Type: **C**

Item Name: ~

This procedure defines an INFO table with three items: indianstats-id (the ARC/INFO identification number), stat-num (the station number), and stat-name (a name for the station). The *pointinput.aml* program executes the **ADD FROM** command to transfer attribute information to this INFO table.

Enter Command: **add from** ***indianstatid.dat***

Enter Command: **quit**

The ARC command **JOINITEM** joins the contents of this INFO file to the point coverage polygon attribute table (PAT) using the ARC/INFO identification number to relate the two tables.

Arc: **joinitem** ***indianstats.pat indian.dat indianstat.pat indianstats-id~***

indianstats-id

The attribute definition of the point coverage is now complete.

The ARC command **PROJECT** translates the *indianstats* point coverage from geographic coordinates to the desired map projection as defined by a projection file. Table 4-3 lists the projection file used in this example.

Arc: **project cover indianstats indiangage flowgage.prj**

Table 4-3 Listing of flowgage.prj.

```
input
projection geographic
units dd
datum wgs72
parameters
output
projection albers
units meters
datum wgs72
parameters
37 30 00
39 30 00
-98 00 30
37 00 00
0.0
0.0
end
```

The *indiangage* point coverage is now ready to use (Maidment 1995).

4.4 Development of 1:100k Digital Line Graph Coverages

Digital line graphs are useful for locating bridges and other points of interest along streams. Hypsography and hydrology DLGs may also be used as input to create a custom DEM using the ARC **TOPOGRIDTOOL**. **TOPOGRIDTOOL** accepts point or contoured elevation data, a streams coverage, and a boundary coverage as input and interpolates a grid of elevation points. Before DLGs may be used in a GIS environment, however, they must be imported, projected, and processed. Figure 4-3 displays the menu interface to the DLG processing tools.

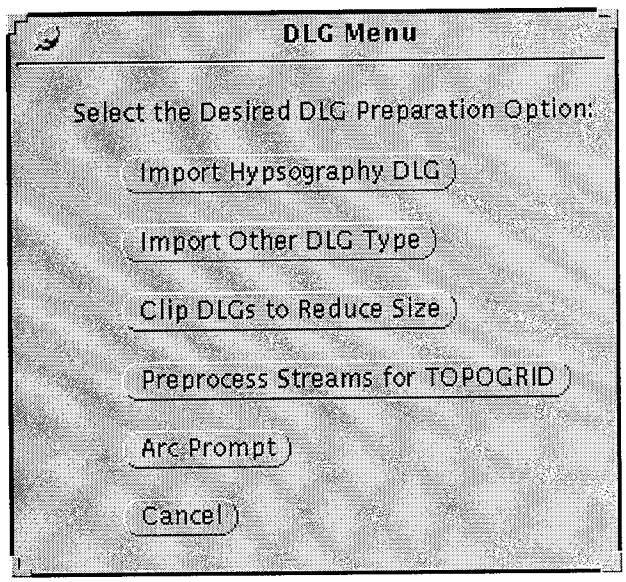


Figure 4-3: The DLG Preparation Menu.

4.4.1 Importing 1:100k Digital Line Graphs

Two AML scripts have been developed for importing DLGs: *dlg100hp.aml* and *dlg100other.aml*. The script *dlg100hp.aml* imports 1:100k hypsography DLGs, while *dlg100other.aml* imports all other 1:100k DLG layers. These programs differ in the way they process DLG attribute information. *Dlg100hp.aml* retains the attribute item containing contour elevations, while *dlg100other.aml* does not access any attribute information. This section demonstrates the use of *dlg100hp.aml* to develop a coverage of elevation contours for the Indian Creek watershed. The Indian Creek watershed traverses two 1:100k DLG map sheets: 4i_hp1 and 4i_hp2.

The digital line graphs arrive in a compressed format. *Dlg100hp.aml* executes the UNIX 'unzip' utility from the command prompt to decompress each map sheet intersected by the study area. The 'unzip' utility is available from DASC via FTP for both UNIX and MSDOS operating systems.

% **unzip** ***4i_hp1.zip***

Next, the **DLGARC** command converts the uncompressed DLG file into an ARC/INFO coverage. The *dlg100hp.aml* script then builds topology for the newly created coverage.

Arc: **dlgarc optional** ***4i_hp1.dlg 4i_hp1***

Arc: **build** ***4i_hp1*** **line**

The **DLGARC** command generates a line coverage and an INFO table with the extension *.acode. One such INFO file exists for each DLG tile. Before multiple map sheets are merged, information contained in *.acode must be joined to the arc attribute file (AAT) of the DLG coverage using the **JOINITEM** command. The *.acode file contains ten attribute categories. The AML dlg100hp.aml transfers only the contour elevations to the arc attribute table by first transferring the identification item (4i_hp1-id) and the elevation item (minor2) to a new INFO file using the **PULLITEMS** command. The newly created INFO file is then joined with the AAT.

Arc: **pullitems** ***4i_hp1.cont***

Enter the 1st item: ***4i_hp1-id***

Enter the 2nd item: ***minor2***

Enter the 3rd item: **end**

Arc: **joinitem** ***4i_hp1.aat 4i_hp1.cont 4i_hp1.aat 4i_hp1-id 4i_hp1-id***

Each DLG tile contains a border arc enclosing the map sheet. These borders must be deleted before the tiles are joined. The *dlg100hp.aml* program invokes the **RESELECT** command to select all but the border arcs, and then saves the selected arcs as a new coverage (Maidment 1995).

Arc: **reselect** ***4i_hp1 r4i_hp1*** **line # line**

Enter a logical expression. (Enter a blank line when finished)

>: **res rpoly# > 1**

>: ~

Do you wish to re-enter expression (Y/N)? **N**

Do you wish to enter another expression (Y/N)? **Y**

Enter a logical expression. (Enter a blank line when finished)

>: **res lpoly# > 1**

>: ~

Do you wish to re-enter expression (Y/N)? **N**

Do you wish to enter another expression (Y/N)? **N**

Once the multiple tiles have been processed, *dlg100hp.aml* combines them using the **APPEND** command.

Arc: **append *indianhputm* line**

Enter the 1st coverage: ***r4i_hp1***

Enter the 2nd coverage: ***r4i_hp2***

Enter the 3rd coverage: **end**

Next, the appended coverage is projected using a format that is slightly different than the one used for point coverages. The 1:100k DLGs' native map projection is Universal Transverse Mercator (UTM). Kansas traverses three UTM zones (13, 14 and 15), thus making the use of a projection file impractical. The AML enters projection information at the command line, permitting ARC/INFO to use the geographic information stored in the UTM coverage to define the input projection.

Arc: **project cover *indianhputm indianhp***

Project: **output**

Project: **projection albers**

Project: **units meters**

Project: **parameters**

1st standard parallel [0 0 0.0]: **37 30 00**

2nd standard parallel [0 0 0.0]: **39 30 00**

central meridian [0 0 0.0]: **-98 00 30**

latitude of projection's origin [0 0 0.0]: **37 00 00**

false easting (meters) [0.000]: **0.0**

false northing (meters) [0.000]: **0.0**

Project: **end**

Arc: **build *indianhp* line**

Next, *dlg100hp.aml* changes the name of the item containing the contour elevation data from minor2 to elev.

Arc: **tables**

Enter Command: **select *indianhp.aat***

Enter Command: **alter**

Enter item name: **minor2**

Item name: **elev**

Item output width: ~

Item type: ~

Alternate item type: ~

Enter item name: ~

Enter Command: **quit**

This completes the processing of the hypsography DLG. Digital line graphs of roads, hydrology, and other data types are processed similarly using *dlg100other.aml.*

The scripts *dlg100hp.aml* and *dlg100other.aml* both access a form menu to help the user select DLG files for processing. This menu, displayed in Figure 4-4, lists all the digital line graphs in the working directory. The user can select one or more files from the list.

4.4.2 Preprocessing Stream Coverages

TOPOGRIDTOOL requires that stream input coverages contain only arcs representing flowing water, and that these arcs form a branching network with no loops. Thus, all polygons in the DLG coverage representing lakes, marshes, braided streams, etc., must be removed. In addition, all stream arcs that were digitized in the wrong direction must be reversed (ESRI 1995). The scripts *preproc.aml* and *preproc.menu* furnish a graphical interface, for the interactive editing of hydrology DLGs. This interface is shown in Figure 4-5.

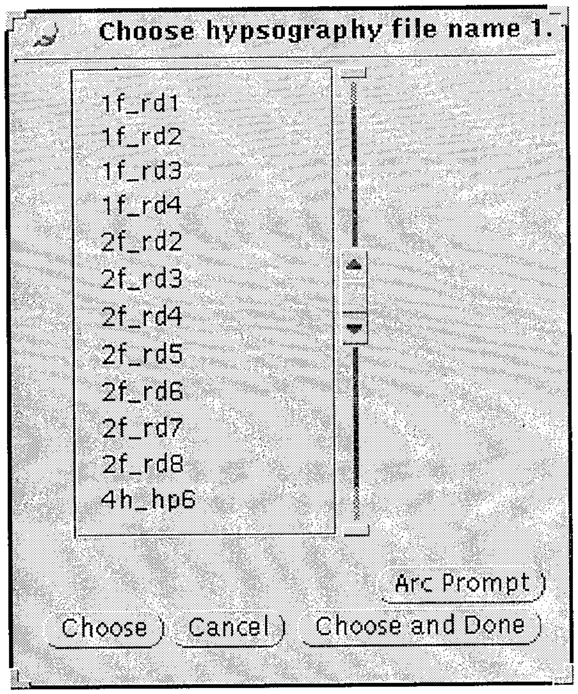


Figure 4-4: Choosing the DLG Files to be Processed.

Editing of ARC/INFO coverages is performed in the ARCEDIT module. The edit session begins by setting the display device and map extent, and specifying the coverage and feature to be edited. The **DRAWENVIRONMENT** and **DRAW** commands display the edit coverage.

Arc: **arcedit**

Arcedit: **display 9999**

Arcedit: **mape** ***streams***

Arcedit: **editcoverage** ***streams***

Arcedit: **editfeature arc**

Arcedit: **drawenvironment arc on arrows**

Arcedit: **draw**

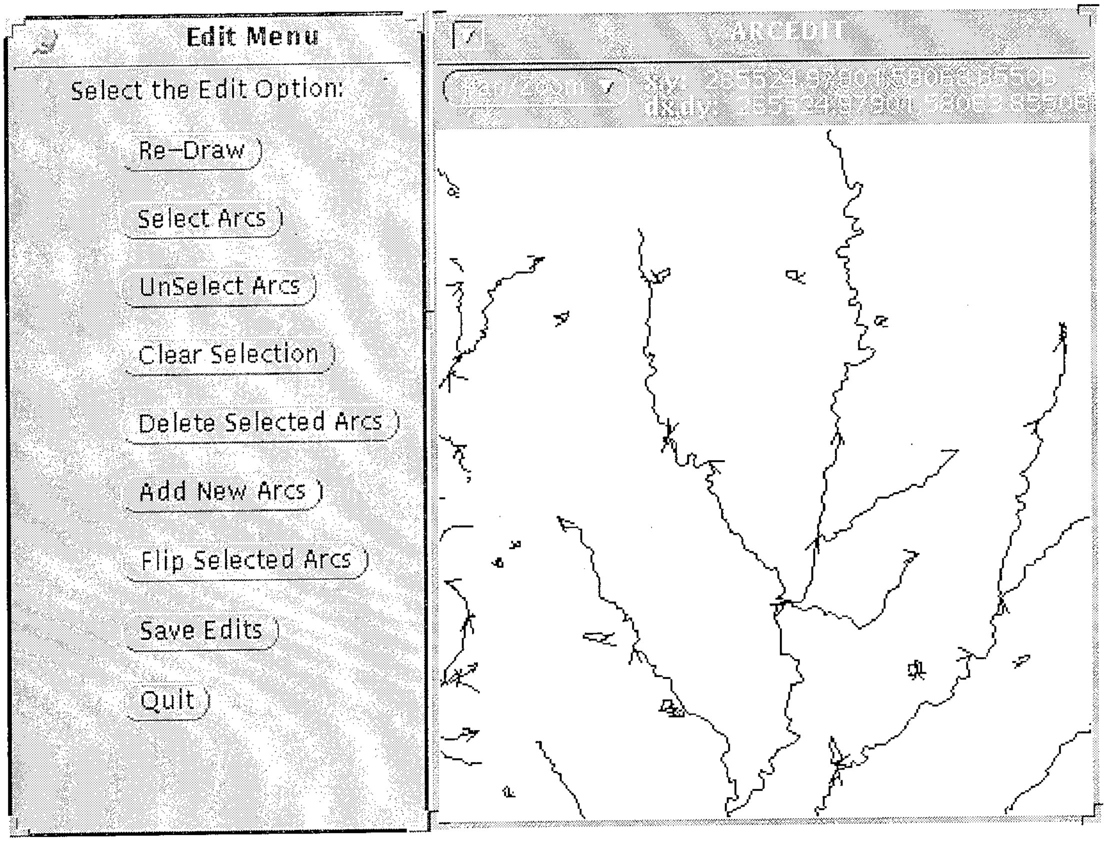


Figure 4-5: Graphical Interface for Preprocessing Streams for use with TOPOGRIDTOOL.

The first step in stream coverage preprocessing is to remove arcs from braided streams and polygon features. The user must select these arcs and then remove them.

Arcedit: **select many**

Arcedit: **delete**

Each deleted polygon must be replaced by a single arc.

Arcedit: **add**

Once all polygons have been removed, all stream arcs pointing upstream must be redirected. The **DRAWENVIRONMENT** option ARROWS displays the direction of each arc. Arcs pointing upstream are selected and then reversed using the **FLIP** command.

Arcedit: **select many**

Arcedit: **flip**

The above steps may be repeated as many times as necessary. When editing is complete, the user saves and then quits the session.

Arcedit: **save**

Arcedit: **quit**

The edited stream coverage is ready for input to **TOPOGRIDTOOL**.

In the future, a module may be developed that automates parts of this time-consuming process. All arcs that have the same "to" and "from" nodes, representing lakes or ponds, could be deleted automatically. Also, an algorithm to associate "to" and "from" nodes with the stream direction could be used to make sure that all stream arcs point in the correct direction.

4.5 Development and Processing of a Digital Elevation Model

The user can develop a digital elevation model for a study area from USGS digital elevation models, or create a custom DEM with the ARC/INFO **TOPOGRIDTOOL**. This section demonstrates the development of a DEM by each of these methods, then illustrates how the DEM is processed to yield useful hydrologic information. Figure 4-6 shows the options in the DEM menu.

4.5.1 Importing USGS Digital Elevation Models

ARC/INFO procedures for importing 1:24k and 1:250k USGS DEMs are similar. The scripts *dem24.aml* and *dem250.aml* have been developed to import these types of DEMs. This section explains the use of *dem250.aml* to import a 3-arc-second DEM for the Indian Creek area. If a watershed covers multiple DEM map sheets, *dem250.aml* repeats some of the processing steps listed below for each sheet. The Indian Creek watershed falls entirely within the Lawrence East one-degree block.

Digital elevation models downloaded from DASC arrive in a compressed format and must be uncompressed using the 'unzip' utility. Once the DEM has been uncompressed, the ARC command **DEMLATTICE** converts the USGS Digital Elevation Model file into an ARC/INFO grid.

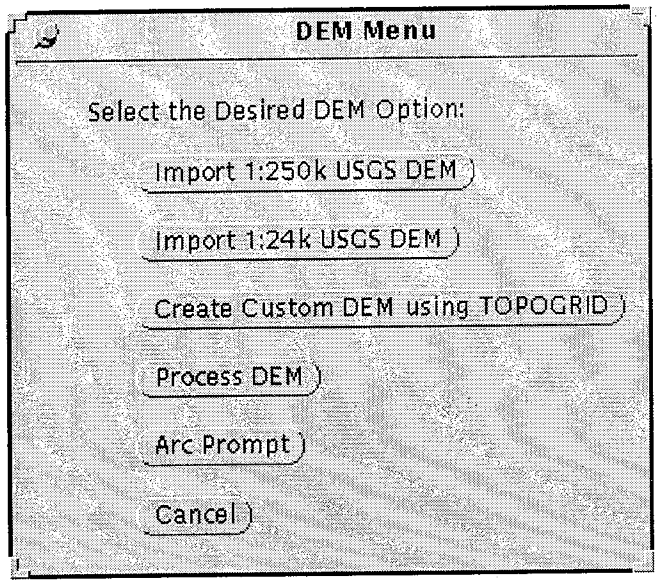


Figure 4-6: The DEM Preparation Menu.

% **unzip** *lawren_e.zip*

Arc: **demlattice** *lawren_e.dem lawren_e* **usgs**

Next, *dem250.aml* executes the GRID **MERGE** command to combine multiple map sheets. If only one map sheet exists, the sole effect of this operation is to copy the grid to a new name.

Arc: **grid**

Grid: **indian = merge(*lawren_e*)**

Due to the large size of the map sheet, the merged DEM may be much larger than the area of interest. The **SETWINDOW** command permits the user to zoom in on an area of interest. In order to set the window, the DEM is first displayed. The user then defines the analysis window. The AML saves the portion of the merged grid contained within the specified window under a new name.

Grid: **display 9999**

Grid: **mape** ***indian***

Grid: **gridp** ***indian*** **# linear nowrap gray**

Grid: **setwindow ***

Define the box

Grid: ***indiandem* = *indian***

Now that the grid has been reduced in size, *dem250.aml* projects it from geographic coordinates to the map projection of choice using the *geoalb.prj* projection file.

Grid: ***indianprj* = project (*indiandem, geoalb.prj*, #, 100)**

The projected DEM may now be processed as described in Section 4.6.3 (Maidment 1995).

4.5.2 Creating a Custom Digital Elevation Model

Frequently USGS DEMs do not provide the resolution necessary for hydrologic modeling. In this case, it is desirable to create a DEM from 1:100k DLGs of hypsography and hydrology or from digitized contour and stream coverages using **TOPOGRIDTOOL.**

Arc: **topogridtool *indianprj* 30**

TOPOGRIDTOOL is a menu-driven interface to the ARC **TOPOGRID** command. **TOPOGRID** accepts point or contour elevation, stream network, and boundary coverages as input and interpolates a DEM of specified cell size. The use of a stream coverage as input to **TOPOGRID** ensures that the interpolated DEM's drainage paths will closely resemble the true location of streams in the watershed. Before DLGs may be input to **TOPOGRID**, they must be processed as described in Section 4.4. Once **TOPOGRIDTOOL** has successfully completed the interpolation process, the generated DEM may be processed further.

The primary advantage of **TOPOGRIDTOOL** for the development of a watershed DEM is the **TOPOGRID** drainage enforcement option. When activated, this option attempts to develop a DEM free of sinks. The success of this option depends on three user-set tolerances: RMS, Tol1, and Tol2. These tolerances represent the root-mean-square residual, the maximum height of obstacle that will be removed from a drainage path, and the maximum height to a saddle that may be considered a possible exit from a sink. ARC/INFO ArcDoc provides guidance on selecting appropriate values for these tolerances (ESRI 1995). The interpolated DEM will generally contain far fewer sinks than a USGS DEM, minimizing the area of the watershed assigned a zero slope by the **FILL** command.

4.5.3 Hydrologic Analysis of DEMs

The ARC/INFO GRID module contains several built-in hydrologic functions. This section describes the use of **FILL**, **FLOWDIRECTION**, **FLOWACCUMULATION**, **WATERSHED**, and **FLOWLENGTH** commands on the Indian Creek **TOPOGRID** DEM. The menu shown in Figure 4-7 provides options for executing these commands.

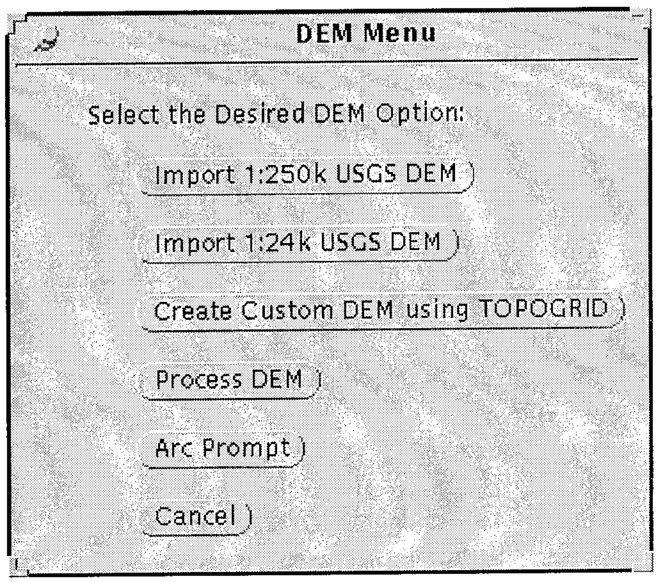


Figure 4-7: The Process DEM Menu.

The GRID **FILL** command fills sinks in a DEM to the elevation of the lowest possible outlet cell. All DEMs, even those created using **TOPOGRIDTOOL**, contain some sinks. Failure to remove these sinks results in a discontinuous stream network. The following usage of **FILL** removes all sinks from the DEM whose depth is less than 100 meters. The depth of a sink is defined as the difference in elevation of the lowest cell in the sink and the highest cell draining to that depression. Specification of a maximum depth is optional.

Grid: **fill *indianprj indianfil* sink 100**

The **FLOWDIRECTION** function in GRID determines the direction of flow in each cell of the DEM. **FLOWDIRECTION** operates on the eight-direction pour point model, in which flow through a cell is assumed to occur in the direction of steepest descent. This model considers the slope from a cell to each of its eight neighbors (Maidment 1995).

Grid: ***indianfdr*** **= flowdirection (*indianfil*)**

The GRID **FLOWACCUMULATION** command calculates the number of cells upstream of each cell in the DEM, in effect figuring each cell's contributing drainage area. The resulting flow accumulation grid may be used to create a stream coverage.

Grid: ***indianfac*** **= flowaccumulation (*indianfdr*)**

The following sequence of commands creates a line coverage of streams in the Indian Creek watershed which have a flow accumulation greater than 500 cells.

Grid: ***indianstr*** **= con (*indianfac* > 500, 1)**

Grid: ***indianlnk*** **= streamlink (*indianstr, indianfdr*)**

Grid: ***indianstream*** **= gridline (*indianlnk*)**

The resulting coverage may be compared to the USGS hydrology DLG to test the accuracy of the DEM.

The **WATERSHED** command delineates the watershed upstream of one or more points. The following procedure determines the watershed upstream of a USGS stream gage.

Grid: **mape** ***indianfac***

Grid: **gridpaint** ***indianfac*** **# linear # gray**

Grid: **markercolor 2**

Grid: **points** ***indiangage***

Grid: ***indianws*** **= watershed (*indianfdr*, selectpoint(*indianfac*, *))**

Grid: ***wsindian*** **= gridpoly (*indianws*)**

These commands display the flow accumulation grid and the point coverage of the USGS stream gages, then permit the user to select the watershed outlet from the display. The outlet must fall on a stream as delineated by the **FLOWACCUMULATION** command. The gage point might not lie exactly on the stream defined by the flow accumulation grid, as in Figure 4-8. In this case, the user must select the point on the stream nearest the gage. The final command in this series creates a polygon coverage of the watershed boundary.

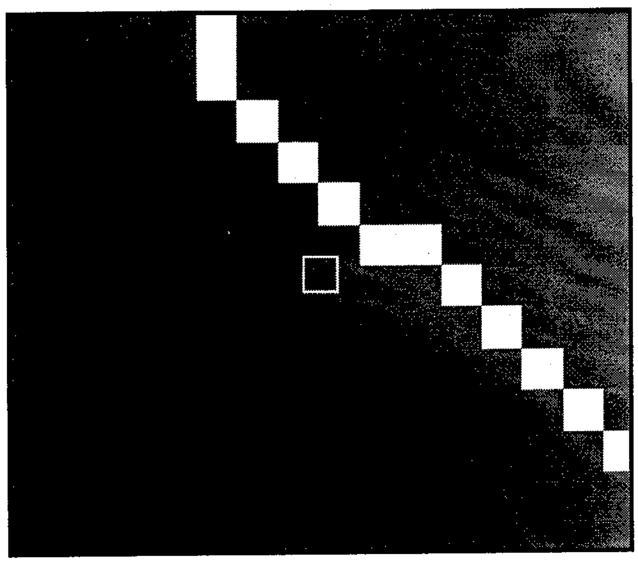


Figure 4-8: USGS Stream Gage Offset from Flow Accumulation Network.

The filled DEM, flow direction, and flow accumulation grids may be clipped to the watershed boundary by multiplying these grids by a grid of the watershed with each cell assigned a value of one.

Grid: ***indianclip = indianws / indianws***

Grid: ***wsindianfac = indianclip * indianfac***

Grid: ***wsindianfdr = indianclip * indianfdr***

Grid: ***wsindianfil = indianclip * indianfil***

Clipping the grids to the watershed boundary reduces computation time on subsequent operations. Figure 4-9 displays the clipped DEM for Indian Creek.

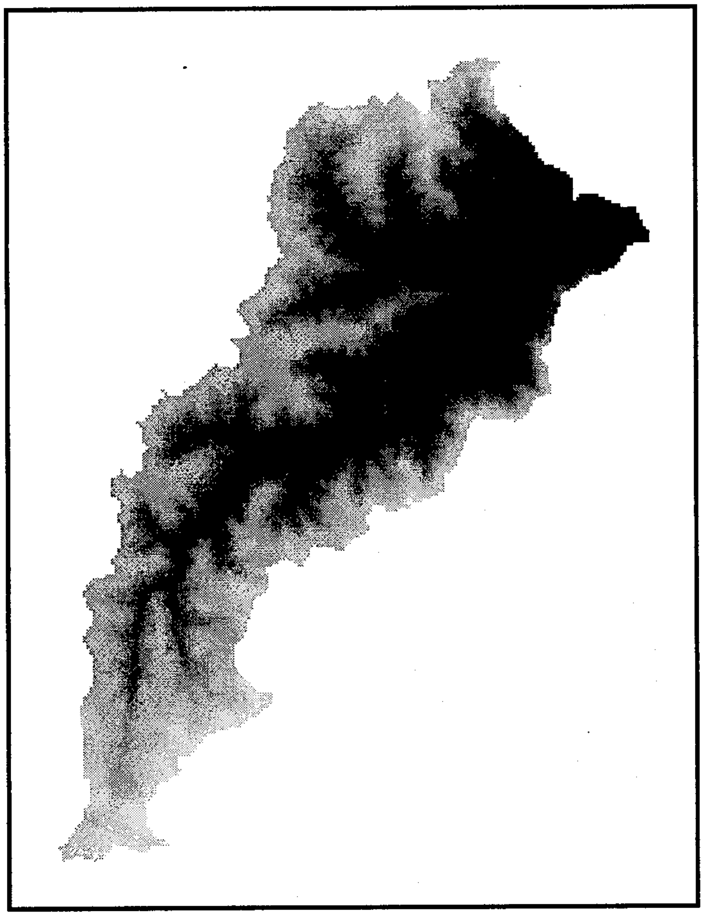


Figure 4-9: Clipped and Filled DEM for Indian Creek.

One last important hydrologic function built into GRID is the **FLOWLENGTH** command. The **FLOWLENGTH** command calculates the distance from each cell to the watershed outlet (using the DOWNSTREAM option) or the longest flow path to each cell (using the UPSTREAM option). Each calculation may be weighted by a grid of impedance to produce the travel time rather than distance. Impedance is defined as the reciprocal of the velocity, and the travel time through a cell is equal to the impedance multiplied by the length of the cell. These commands are applied as follows:

Grid: ***indianupstream* = flowlength (*indianfdr, #, upstream*)**

Grid: ***indiandwnstrm* = flowlength (*indianfdr, #, downstream*)**

The # sign occupies the location of the impedance grid when no such grid exists.

4.6 Development of a SSURGO Soils Coverage

The SSURGO database consists of polygon coverages of map units tiled by 7.5-minute quadrangle sheets and a series of attribute tables tiled by county. The procedure for implementing the SSURGO soils information in a GIS has two parts: importing the spatial data and processing the attribute data.

4.6.1 Importing SSURGO Spatial Data

The procedure for importing SSURGO spatial data is similar to that for 1:100k digital line graphs. The *ssurgo.aml* program describes the general procedure.

The Indian Creek watershed spans three 7.5-minute quadrangle sheets. The three SSURGO files for Indian Creek are jo1659.e00, jo1759.e00, and jo1760.e00. The *.e00 extension indicates that these are ARC/INFO export files. The ARC **IMPORT** command restores an export file to a coverage.

Arc: **import cover** ***jo1659.e00 jo1659soil***

Multiple polygon coverages are joined using the **MAPJOIN** command. **MAPJOIN** clips the merged coverages to a watershed boundary if one is specified. The watershed coverage must first be projected into UTM coordinates.

Arc: **mapjoin** ***joincov*** **poly #** ***wsclip***

Enter the 1st coverage: ***jo1659soil***

Enter the 2nd coverage: ***jo1759soil***

Enter the 3rd coverage: ***jo1760soil***

Enter the 4th coverage: **end**

The *ssurgo.aml* script executes the **DISSOLVE** command on the new soil coverage to remove map sheet borders.

Arc: **dissolve** ***joincov discov*** **muid poly**

This formulation of the **DISSOLVE** command removes all arcs separating polygons with the same map unit identification number (muid). Next, *ssurgo.aml* projects and builds topology for the dissolved coverage, which is then linked to attribute data using *ssurgatt.aml*.

4.6.2 Processing SSURGO Attribute Data

Use of the NRCS SSURGO database is complicated by the large number of attributes available for each soil unit. Typically, only a few of these attributes are of interest to the hydrologist, such as the USDA texture class and the depth of the surface soil layer. The *ssurgatt.aml* script adds the SSURGO items texture:c (USDA texture classification for the surface soil), laydeph:i (the depth of the surface soil layer), and musym:c (the soil unit symbol, which is lost from the original PAT when the DISSOLVE command is issued) to the SSURGO polygon attribute table.

Of the 22 different SSURGO attribute files, *ssurgatt.aml* accesses only the comp and layer INFO tables.

Arc: **import info *jo_layer.e00 jo_layer***

Arc: **import info *jo_comp.e00 jo_comp***

Often, the watershed of interest intersects multiple counties. If this is the case, the comp and layer INFO files for each county must be imported. To accomplish this, *ssurgatt.aml* defines an INFO table (indian_layer) with the following items: muid, texture, depth, and musym. Before attribute information is added to this new INFO file, the item definition for laydeph:i in each tile's layer INFO file must be changed from a character to a numeric value using the TABLES **ALTER** command. The *ssurgatt.aml* script then relates the comp and layer INFO files using the **RELATE** command in TABLES.

Enter Command: **relate add**

Relation Name: ***jo***

Table Identifier: ***jo_comp***

Database Name: **info**

INFO Item: ***muid:c***

Relate Column: ***muid:c***

Relate Type: **linear**

Relate Access: **rw**

Relation Name: ~

Once the relate environment has been set, *ssurgatt.aml* selects all records in the layer INFO file containing information about the top soil layer.

Enter Command: **sel** ***jo_layer***

Enter Command: **reselect**

Enter Command: **aselect** ***layernum:i*** **con '1'**

The **UNLOAD** command saves the desired attribute items for the selected records to a text file, which is in turn added to the INFO file defined above. The jo//musym:c option in the **UNLOAD** command syntax retrieves the musym:c item from the comp file via the relate environment defined above.

Enter Command: **unload** ***jodata muid:c texture:c laydeph:i jo//musym:c***

Enter Command: **sel** ***indian_layer***

Enter Command: **add from** ***jodata***

The AML repeats the above steps for each county intersected by the watershed. Once all of the counties have been processed, *ssurgatt.aml* quits the TABLES module and joins the new INFO file containing the desired attributes to the PAT of the watershed soils coverage.

4.7 Development of a Land Cover Coverage

The KARS land cover files are simple to implement in a GIS environment. These coverages are tiled by county and compressed as ARC/INFO export files. All attribute information is contained within the PAT. The *lc.aml* script may be used to import and prepare the land cover coverage. This script **UNZIP**s, **IMPORTS**s, **MAPJOINS**s, **DISSOLVE**s, **PROJECTS**s, and the **BUILDS**s topology for the land cover coverage.

4.8 User Control within the Menu System

Most menus is the GUI contain an 'Arc Prompt' option. This option allows the user to exit the AML script temporarily, execute ARC/INFO or system commands, and then return to the AML script. The command **&return**, executed at the ARC prompt, returns control to the AML script at the point of departure. Every menu also includes a 'Cancel' option, which returns control to the previous menu.

Chapter 5

Analysis of Storm Rainfall over a Watershed

5.1 Rainfall Analysis Tools

The ARC Macro Language has been used to develop several storm rainfall analysis tools. These tools generate ARC/INFO grids of NEXRAD Stage III hourly or daily rainfall estimates, contour these grids, compute the hourly or daily spatial average of rainfall over a watershed, and determine the temporal distribution of rainfall for any one HRAP cell. This chapter describes the use of these tools to analyze a rainfall event occurring April 29, 1995, over a 355-square-mile watershed on Bluff Creek in Harper County, Kansas.

5.2 Setting up the ARC/INFO Work Environment

The rainfall analysis tools are accessed from the main menu of the graphical user interface discussed in Chapter 4. When the 'Map and Analyze Rainfall Data' option is selected from the main menu, the interface first asks for a name for the storm event (Figure 5-1) and then displays the rainfall analysis menu (Figure 5-2).

5.3 Loading Rainfall Files to the Local Directory

Before analysis of rainfall begins, rainfall files for a particular storm must be copied to the working directory. Files for southern Kansas must be downloaded manually to the local directory via anonymous FTP from the Arkansas-Red Basin River Forecasting Center (ABRFC) and uncompressed at the system prompt using the 'gunzip' utility. Preparation of rainfall files for northern Kansas has been automated by the 'Load MBRFC Rainfall Files to Local Directory' menu option. This option calls copymb.aml, which prompts the user for the start and end dates and times of the storm and then retrieves all hourly files between those dates from the /files/mbrfc/archives/reduced directory.

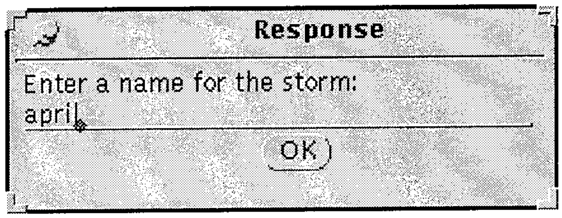


Figure 5-1: Entering the Storm Name.

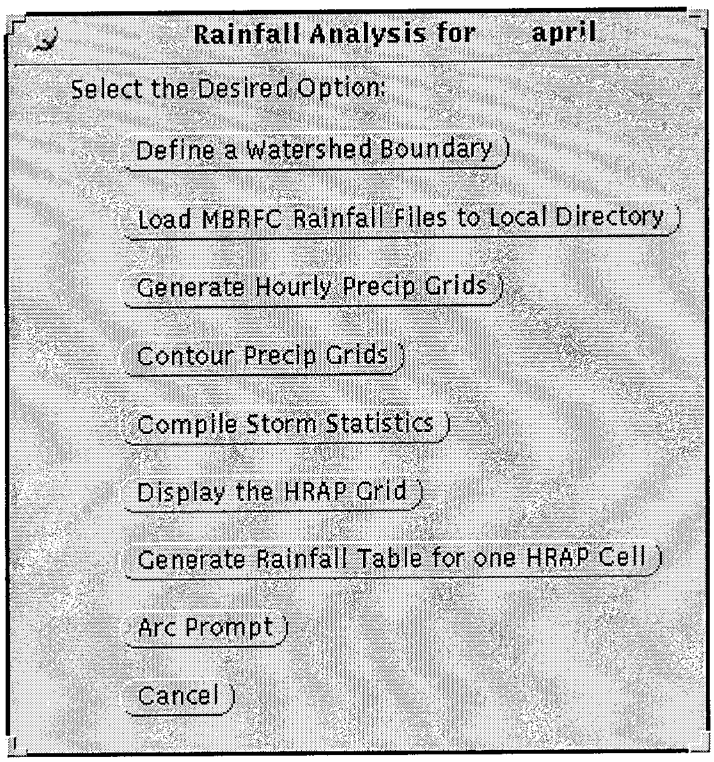


Figure 5-2: The Rainfall Analysis Menu.

5.4 Generating Hourly Precipitation Grids

Before NEXRAD Stage III precipitation data is analyzed in ARC/INFO, the information contained in the River Forecasting Center (RFC) files must be georeferenced. The 'Generate Hourly Precip Grids' option on the rainfall analysis menu links the NEXRAD estimates to an HRAP coverage of the area of interest and then generates an

ARC/INFO grid for each time period of rainfall. These grids serve as input to the 'Contour Precip Grids' and 'Compile Storm Statistics' analysis options.

The precip.aml program initializes the storm definition menu, which requires the user to select the river basin (Missouri or Arkansas), specify the start and end rainfall file names, choose a cell size for the hourly rainfall grids, and select the temporal resolution of the grids. Figure 5-3 shows this menu.

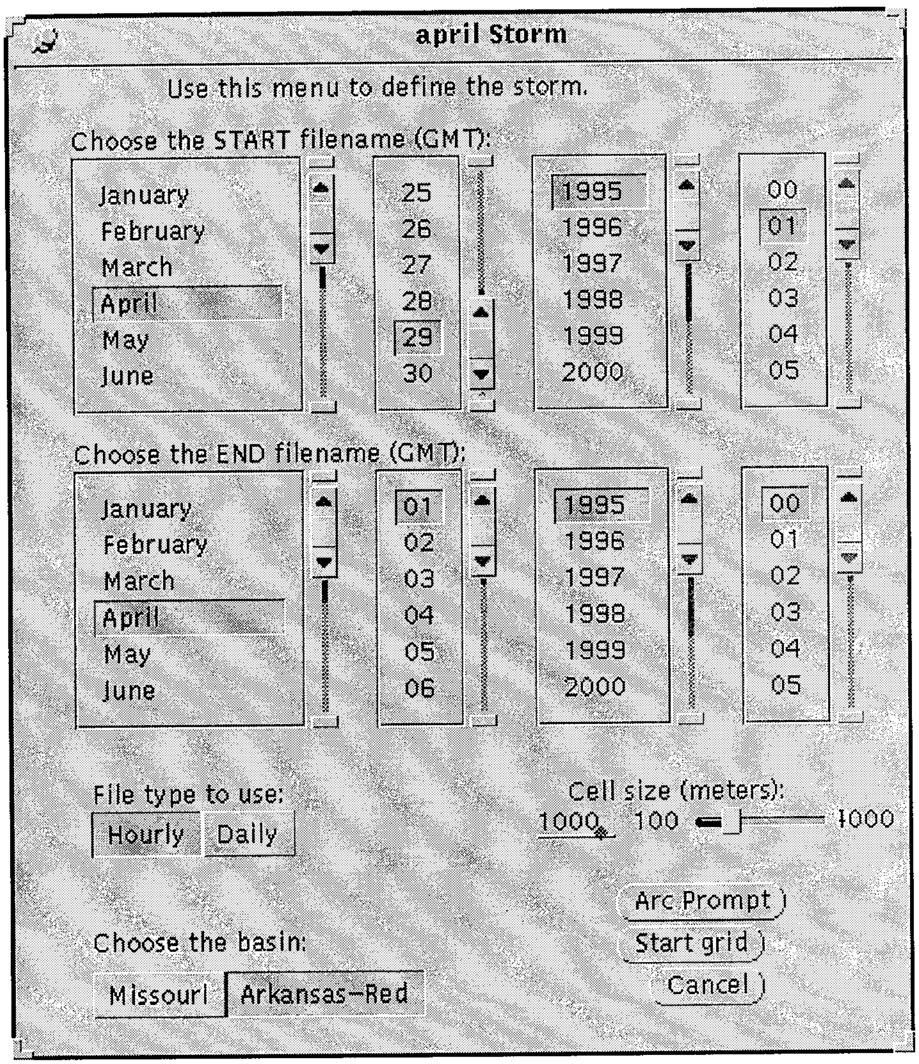


Figure 5-3: Defining the Storm.

The 'start' and 'end' filenames are the names of the first and last rainfall files to be processed. Hourly files are identified by the Greenwich Mean Time and date at the end of the hour. The daily files for the Arkansas basin contain rainfall data for

Greenwich Mean calendar days. Greenwich Mean Time is six hours earlier than Central Standard Time and five hours earlier than Central Daylight Time.

After the storm has been defined and the user has exited the storm menu, the user then identifies an area that encompasses the watershed of interest. The precip.aml program displays the watershed boundary and the HRAP grid, creates a new coverage, sets the feature to be edited, and then asks the user to draw a polygon around the area of interest. These operations are performed within the ARCEDIT module.

Arc: **arcedit**

Arcedit: **display 9999**

Arcedit: **mape** ***wsbluff***

Arcedit: **backcoverage** ***wsbluff*** **3**

Arcedit: **backcoverage** ***/files/arcfiles/kan_hrap/kangeoccalb*** **2**

Arcedit: **backenvironment poly on**

Arcedit: **drawenvironment poly on**

Arcedit: **create** ***aprilarea wsbluff***

Arcedit: **build**

Arcedit: **editfeature poly**

Arcedit: **draw**

Arcedit: **add one**

Figure 5-4 shows the ARCEDIT display at this point. Once the user has drawn the polygon that encompasses the watershed, precip.aml saves and quits the edit session and then intersects this area with the HRAP coverage for Kansas. The result is an HRAP coverage of the watershed and surrounding area.

Arcedit: **save**

Arcedit: **quit**

Arc: **intersect** ***aprilarea /files/arcfiles/kan_hrap/kangeoccalb junehrap~***

poly

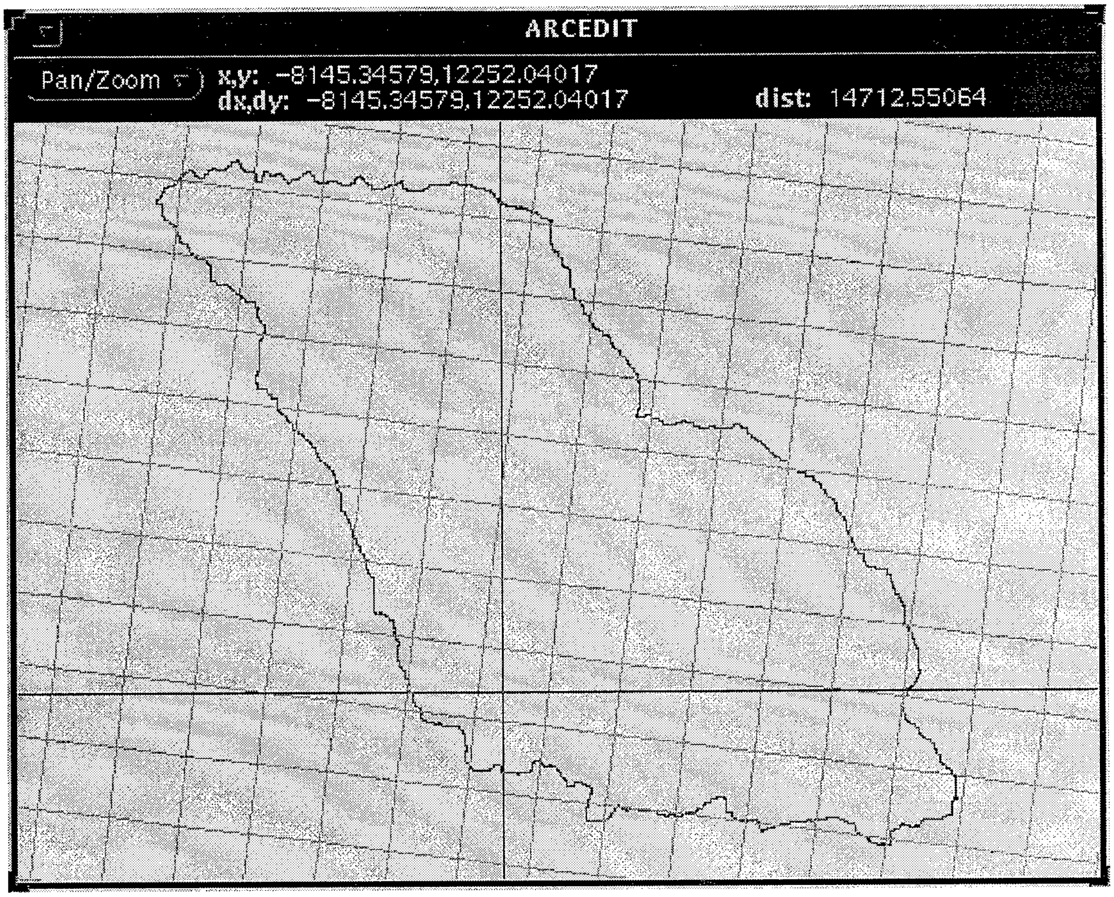


Figure 5-4: ARCEDIT Display for HRAP Coverage Generation.

The AML links rainfall data to this HRAP coverage in a multi-step process. First, precip.aml develops two text files. The first text file created by precip.aml is a list of the names of the hourly or daily precipitation files for the period of interest. The second text file contains the HRAP coordinates and ARC/INFO identification for all HRAP cells in the selected area. To generate this file, precip.aml enters TABLES, selects the coverage PAT, executes the **RESELECT** command to deselect all records, and uses **ASELECT** to select all polygons other than the universal polygon (the region outside of the area of interest). Next, precip.aml unloads the ARC/INFO identification number and HRAP coordinates for each selected polygon to a text file.

Arc: **tables**

Enter Command: **sel** ***aprilhrap.pat***

Enter Command: **reselect**

Enter Command: **aselect** ***aprilhrap-id*** **> 0**

Enter Command: **unload** ***hrapinput aprilhrap-id hrapx hrapy***

The second text file created by precip.aml is a list of the names of the hourly or daily precipitation files that cover the duration of the storm. These two lists serve as input to a FORTRAN program (mapab.for for the Arkansas basin or mapmb.for for the Missouri basin) that reads the rainfall data. This program generates a file that contains the ARC/INFO identification number for each HRAP cell and the corresponding rainfall value. If the user specifies a daily grid for a location in the Missouri basin, the FORTRAN program mabmbday.for reads the hourly RFC rainfall data, then sums the data and creates an output file for each Greenwich Mean Time (GMT) calendar day in the specified period.

Once the FORTRAN program has generated the output files, precip.aml transfers the contents of the first file to an INFO table and then adds that table to the PAT. After the rainfall values have been related to the HRAP coverage PAT, precip.aml executes the ARC **POLYGRID** command to develop a grid of the precipitation.

Arc: **polygrid** ***aprilhrap aprilhr1 precip***

Cell Size (square cell): **2000**

Convert the Entire Coverage? (Y/N): **Y**

Next, precip.aml executes the **DROPITEM** command to remove the precipitation values from the PAT.

Arc: **dropitem** ***junehrap.pat junehrap.pat precip***

The precip.aml program repeats the procedure for each rainfall file, saving each hour of data as a new grid. Figure 5-5 displays a grid of one hour of precipitation.

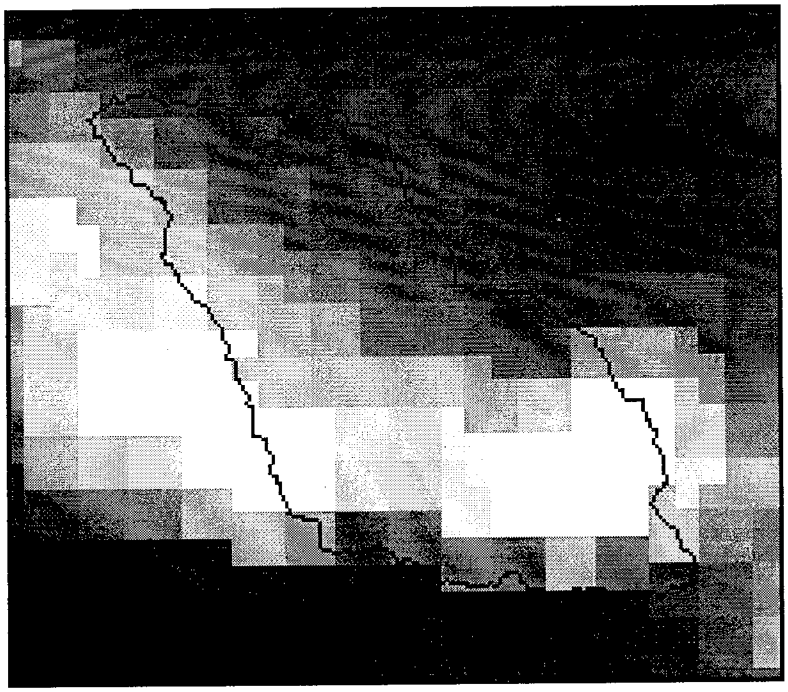


Figure 5-5: Grid of One Hour of Rainfall over the Bluff Creek Watershed.

5.5 Contouring the Precipitation Grids

Contours of precipitation amounts describe the spatial distribution of rainfall in a clear, understandable form. The 'Contour Precip Grids' option on the rainfall analysis menu executes contour.aml, which develops contours either for selected hours of rainfall or for storm totals over a watershed (Appendix C). This section outlines the basic contouring procedure.

The contour.aml program prompts the user to select contour units (millimeters or inches) and then develops rainfall contours in a two-step procedure. First, contour.aml executes the **FILTER** command to smooth anomalies in the rainfall grid, thus improving the appearance of the final contours.

Arc: **filter *aprilhr1 aprilhr1s* low**

Once the rainfall grid has been smoothed, contour.aml prompts the user to enter a contour interval and then executes the ARC **LATTICECONTOUR** command.

Arc: **latticecontour *junehr5s junehr5c* 635 1**

The above statement specifies a contour interval of a 6.35 millimeters (one-quarter inch) and a base contour of 1/100 of a millimeter. The default base contour is zero, but use of this default creates a rough contour at the storm's edge. The contour.aml program displays the contours and then permits the user to select a different interval if the results are unacceptable. Once the user accepts an interval, contour.aml clips the contour coverage to the watershed boundary.

Next, contour.aml creates a new item in the watershed contour coverage called 'precip'. The TABLES **CALCULATE** command uses this item to calculate the contour value in the user specified units.

Arc: **tables**

Enter Command: **additem *wsaprilhr5.aat precip* 4 5 n b**

Enter Command: **sel *wsaprilhr5.aat***

Enter Command: **calc *precip = contour* / 2540**

The above series of commands applies for contour units of inches. Figure 5-6 shows contoured precipitation values for the April storm totals.

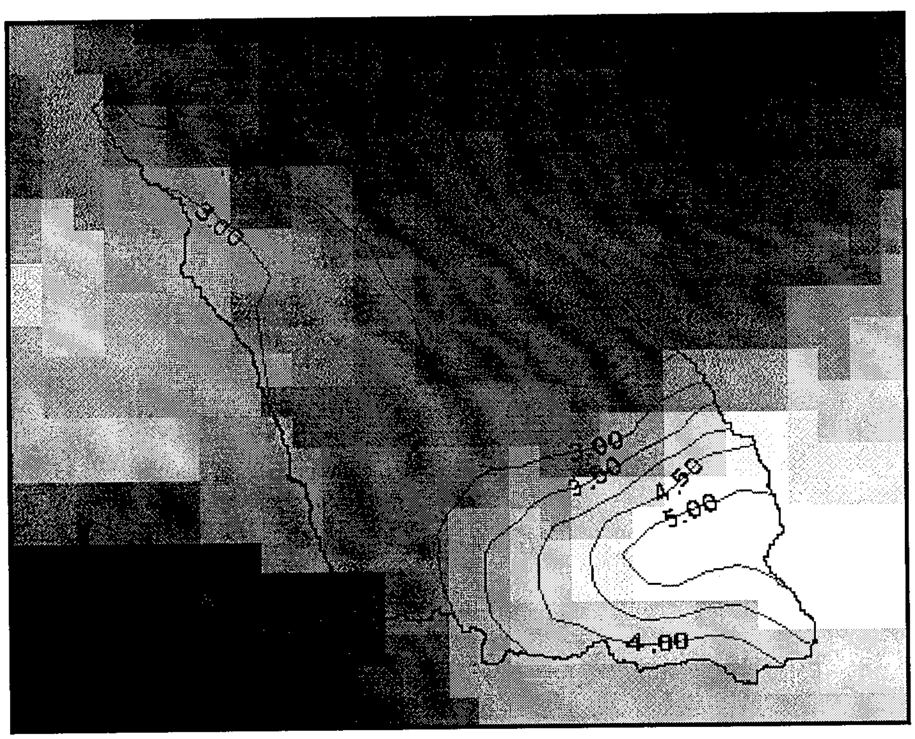


Figure 5-6: Contoured Storm Totals for the Bluff Creek Storm.

5.6 Compiling Storm Statistics

The 'Compile Storm Statistics' rainfall analysis tool (genstats.aml) computes that spatial-average rainfall depth over the watershed at hourly intervals. First, genstats.aml clips the rainfall grids to the watershed boundary using the clip grid developed in Section 4.6.3. The program then calculates the average rainfall from each clipped grid's value attribute table (VAT). Grid VATs list the cell values present in a grid and the number of cells having that value. Table 5-1 displays a sample VAT. The genstats.aml program uses the ARC command **STATISTICS** to calculate the area-weighted average rainfall in the grid VAT. The **UNLOAD** command in TABLES accumulates these hourly averages in a text file.

Arc: **statistics** ***wshr1.vat wshr1.stats***

Statistics: **mean** ***value count***

Table 5-1: Listing of a Sample VAT.

Record	VALUE	COUNT
1	0	42
2	1	5
3	2	3
4	4	2
5	7	4
6	9	4
7	12	3
8	14	4
9	17	4
10	18	2

Statistics: **end**

Arc: **tables**

Enter Command: **sel** ***wshr1.stats***

Enter Command: **unload** ***aprilstats MEAN-W-VALUE***

Figure 5-7 shows the temporal distribution of the area-weighted rainfall for the Bluff Creek storm.

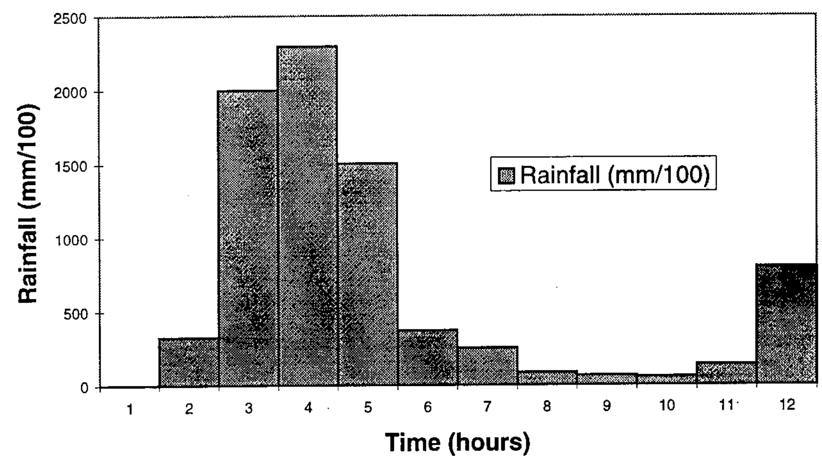


Figure 5-7: Bar Graph of Rainfall Averages for Bluff Creek.

5.7 Generating Rainfall Tables for One HRAP Cell

The temporal distribution of rainfall at a particular location can be obtained with the 'Display HRAP Grid' and 'Generate Rainfall Table for One HRAP Cell' menu options.

The 'Display the HRAP Grid' option executes hrapview.aml, which displays the HRAP grid over background coverages selected by the user and prompts the user to select an HRAP cell for which the coordinates are desired. Once a cell is selected, hrapview.aml displays the INFO record containing the coordinates for that polygon.

After the coordinates of the desired cell have been obtained, the user activates onecell.aml by selecting the 'Generate Rainfall Table for One HRAP Cell' button on the rainfall analysis menu. The onecell.aml program is similar to precip.aml, with the exception that the output file from the FORTRAN program (prepab.for or prepmb.for) is a list of hourly rainfall values for the specified cell rather than a series of files. This output file may be used to generate a bar graph of the temporal distribution of rainfall for that cell.

5.8 Practical Application for KDOT: Rainfall in Sedgwick County

At the request of the KDOT's District 5, Area 6 office, we used the rainfall analysis system to map the rainfall in Sedgwick County during late July and August of

1996. The "watershed" was defined as the entire county. The county boundary was entered as a polygon. A roads coverage was created from DLGs downloaded from DASC. Files of daily rainfall for the six-week period of interest were downloaded from ABRFC. Rainfall grids were generated for each day and for the entire period. Maps with rainfall contours were created from these grids. Figure 5-8 the shows total rainfall for the six-week period.

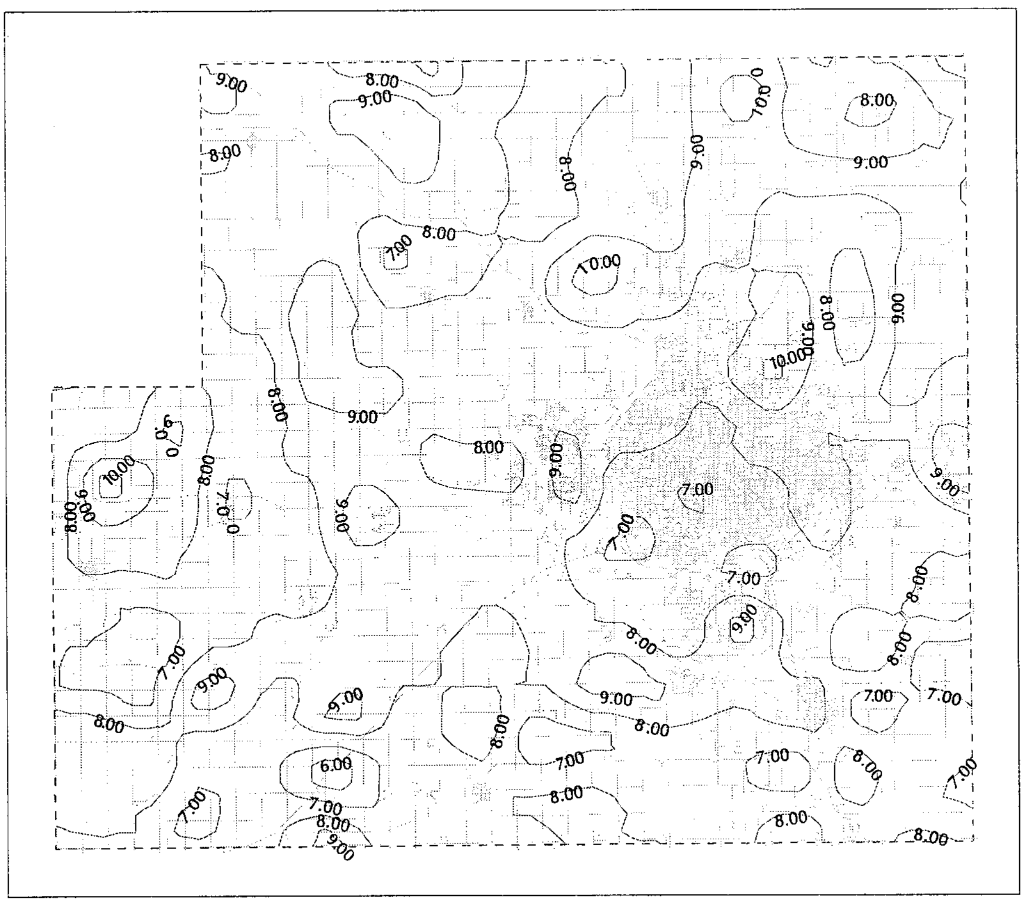


Figure 5-8: Rainfall over Sedgwick County from July 14, 1996, to September 1, 1996.

Chapter 6
Summary and Conclusions

The original objective of this study was to access, georeference, and develop tools to analyze NEXRAD Stage III radar-based precipitation estimates in a GIS environment. The scope of the study was expanded to include the development of procedures for creating ARC/INFO coverages of other available digital data sets including USGS Digital Line Graphs (DLGs), Digital Elevation Models (DEMs), and NRCS soil (SSURGO) and KARS land cover surveys.

NEXRAD Stage III radar-based rainfall estimates produced by the National Weather Service are an invaluable data source to hydrologists concerned with analyzing recent precipitation events. The Stage III estimates are developed hourly on the four-kilometer Hydrologic Rainfall Analysis Project (HRAP) grid using model 1988 Doppler Weather Surveillance Radar (WSR-88D). The use of NEXRAD rainfall estimates in a GIS environment requires development of an HRAP polygon coverage, obtaining and reading the Stage III data files, and linking the rainfall estimates to the HRAP GIS coverage. The HRAP system can be created within ARC/INFO using programs developed by Seann Reed of the University of Texas at Austin. These programs generate an ARC/INFO polygon coverage of the HRAP grid over a user specified area. Stage III data for Kansas must be obtained from one of two separate sources: the Arkansas-Red Basin River Forecasting Center (ABRFC) or the Missouri Basin River Forecasting Center (MBRFC).

Rainfall analysis tools have been developed to import Stage III estimates in ARC/INFO, create grids of rainfall, contour hourly grids or storm totals, calculate the hourly spatial average of rainfall over a watershed, and to list the temporal distribution of rainfall for any one HRAP cell. These tools have been integrated with a graphical user interface. The graphical user interface also contains tools for processing DLG, DEM, STATSGO, and KARS land cover data sets. These tools import and prepare each data set for hydrologic analysis.

DLGs are digital representations of USGS quadrangle sheets. Each DLG contains one geographic feature type, such as roads, hydrology, or hypsography. These maps may be used to locate important features or to develop high-quality DEMs using the ARC **TOPOGRIDTOOL**. DLGs are available for most of Kansas at the 1:100k scale.

DEMs describe the land surface using a regular grid of elevation points. The USGS produces DEMs at two scales: 1:24k and 1:250k. Each USGS DEM type must be used with caution due to the existence of large sinks (or unnatural depressions) in the DEM. The preferred source of hydrologic DEMs are those created with **TOPOGRIDTOOL** from USGS DLGs or from digitized contour and stream layers. The ARC/INFO GRID module contains several hydrologic functions for use with DEMs. These include commands for watershed delineation and for calculating slope, flow direction, flow accumulation, and flow length.

The NRCS SSURGO soil database provides detailed county soil surveys in digital form. The SSURGO database consists of spatial and attribute information in ARC/INFO export format. SSURGO attribute files contain useful hydrologic information such as soil texture, soil layer depths, soil drainage class, and available water capacity. These data may be used to generate input for spatially distributed infiltration models.

The KARS data set describes land cover for each county in Kansas. This survey classifies ten types of land cover, including a distinction between residential and commercial/industrial areas. KARS land cover data may be used to approximate the degree of development in areas for use in infiltration models.

The increasing availability of government digital databases is of great importance to hydrologists. When utilized in a GIS environment, these data sets are a quick and accurate source of input to exiting hydrologic models. In addition, the growing use of geographic information systems paves the way for the development of a new generation of hydrologic models designed to better utilize geospatial data. The ARC/INFO and FORTRAN programs developed in this project expedite the practical use of these data sets in hydrologic studies.

References

ABRFC (1996). Arkansas-Red Basin River Forecasting Center Internet Homepage (http://info.abrfc.noaa.gov/).

Bedient, Philip B., and Huber, Wayne C. Hydrology and Floodplain Analysis, Second Edition. Addison-Wesley Publishing Company. 1997.

DASC (1994). *Core Database Catalog.* Data Access and Support Center, Kansas Geological Survey, Lawrence, Kansas.

DASC (1996). Data Access and Support Center Internet Home Page (http://gisdasc.kgs.ukans.edu/).

ESRI (1995). *ArcDoc Version 7.0* (CD-ROM). Environmental Systems Research Institute, Inc., Redlands, California.

Fortune, Michael A., Labas, Kenneth, and Westcott, Nancy E. (1995). "Accuracy of point rainfall estimates from an 88-D radar in the cool season: the Illinois experience," *International Conference on Radar Meteorology*, Boston, Massachusetts.

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

INTELLICAST (1996). INTELLICAST Internet Home Page (http://www.intellicast.com/).

Maidment, David A. (1995). "GIS & Hydrology: A Workshop of the 15th Annual ESRI User Conference," *1995 ESRI User Conference*, Environmental Systems Research Institute, Inc., Redlands, California.

NRCS (1995). "Soil Survey Geographic (SSURGO) Data Base Data Use Information," *Miscellaneous Publication Number 1527*, Natural Resources Conservation Service.

OSF (1996). The WSR-88D Operation Support Facility Internet Homepage (http://www.osf.ou.edu/).

Reed, Seann M. (1995). "A GIS Procedure for Integrating NEXRAD Precipitation and Topographic Data Sets for Extraction of Rainfall-Runoff Modeling Parameters," thesis to be presented to the University of Texas at Austin (DRAFT).

Snyder, J.P. (1987). "Map Projections -- A Working Manual," *Professional Paper 1395*, United States Geological Survey, Washington, D.C.

UNIDATA (1996). UNIDATA Internet Home Page (http://www.unidata.ucar.edu/).

USGS (1989). "Digital Line Graphs from 1:100,000-Scale Maps," *Data Users Guide 2*, United States Geological Survey, Reston, Virginia.

USGS (1990). "Digital Line Graphs from 1:24,000-Scale Maps," *Data Users Guide 1*, United States Geological Survey, Reston, Virginia.

USGS (1994a). "Digital Line Graphs from 1:2,000,000-Scale Maps," *Data Users Guide 3*, United States Geological Survey, Reston, Virginia.

USGS (1994b). "Digital Elevation Models," *Data Users Guide 5*, United States Geological Survey, Reston, Virginia.

Appendix A

Table A-1: DASC Digital Data Sets (page 1 of 2)

Database:	Description:	Availability:
ADMIN-BOUND	Administrative boundaries of Kansas: State agency districts and contact persons.	All Kansas (FTP)
AQUIFERS	Borders & Isolines for aquifers: Alluvial, High Plains, Dakota, Ozark, Glacial drift.	All Kansas (FTP)
CONTAM-INATION	Identified contamination and environmental monitoring sites in Kansas by KDH&E.	All Kansas (FTP)
DEM-24K	7.5 minute Digital Elevation Model: 30 meter interval.	Limited (Request)
DEM-250K	One degree Digital Elevation Model: 3 arc second, elevation recorded every 240 feet.	All Kansas (FTP)
DLG-100K	Digital Cartographic/Geographic Data from USGS: 7 layers of topographic information.	Limited (FTP)
DLG-24K	Digital Cartographic/Geographic Data from USGS: 7.5 minute quadrangles.	Limited (Request)
DLG-2MIL	Digital Cartographic/Geographic Data from USGS of the entire United States.	All Kansas (FTP)
ENVIR-DATA -SOURCE	Locations of sites that collect environmental data for Kansas.	All Kansas (FTP)
EPA-RF-1	1:500,000 scale hydrology of Kansas.	All Kansas (FTP)
GEOLOGY	Generalized surface geology: Attributed by system, series, and group.	All Kansas (FTP)
GNIS	National Geographic Names Information System (GNIS) from USGS.	(Request)
GPS-BASE	Global Positioning System base station data (Salina, KS).	(FTP)
HUC-BOUND	USGS Hydrologic Unit Code Boundaries: 8 and 11 digit versions.	All Kansas (FTP)
HYPS-100	DLG 1:100,000 Hypsography (Elevation Contours)	All Kansas (FTP)
KCD	Kansas Cartographic Database: Major features digitized from 7.5 minute quadrangles.	All Kansas (FTP)
LANDCOVER	KARS Landcover identified by 10 cover types.	All Kansas (FTP)
LANDSAT TM	Landsat Thematic Mapper (TM): Seven bands of satellite data per scene.	(Request)
LEGS-DIST	Legislative Districts of the Kansas House and Senate, and US House.	All Kansas (FTP)
LULC-250K	Landuse and Landcover polygons from USGS: Nine major landuse classifications.	All Kansas (FTP)

Table A-1: DASC Digital Data Sets (page 2 of 2)

Database:	Description:	Availability:
ORTHO-PHOTOS	Digital aerial photos rectified for terrain and camera angles.	DG, OS, SN (Request)
RTK-HAZ	Right To Know Program - Hazardous facilities in metropolitan counties.	DG, JO, SG, SN, & WY
RWD-BOUND	Polygons of Rural Water District Boundaries.	All Kansas (FTP)
RWD-DISTR	Major Pipelines and Facilities of Rural Water District Distribution Systems.	All Kansas (FTP)
SSURGO	Digital County Soil Survey Data.	Eastern Kansas (Request)
STATSGO	State Soils Geographic database: Generalized soil surveys of Kansas.	All Kansas (FTP)
SWIMS	NPDES Permits, Stream, Effluent, & Lake Monitoring Networks, KATS Values, LEPP.	KS - Lower Rep. Basin (Request)
TIGER	US Census TIGER data for 1992: Fifteen theme layers created in ARC/INFO.	All Kansas (FTP)
UTILITIES	Railroad, Telephone, Electric, Gas distribution & transmission lines and territories.	Limited (Request)
WATER-QUALITY	Water quality data for approximately 1,200 monitoring sites in Kansas.	(Request)
WATERSHEDS	Watershed district boundaries in Kansas.	All Kansas (FTP)
WIMAS	Water Information Mgmt & Analysis System which displays and queries water rights.	All Kansas (Request)

Appendix B

FORTRAN Source Code

Files are listed in alphabetical order.

genhrap.for (page 1 of 10):

```
      program genhrap
C
C   GENHRAP.FOR DEVELOPS THE HRAP GRID SYSTEM COORDINATES
C     FOR INPUT TO GENHRAP.AML.  THIS PROGRAM WAS DEVELOPED
C     BY SEANN M. REED OF THE UNIVERSITY OF TEXAS AT AUSTIN
C     (REED 1995).
C
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 that 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,file2*8,file3*8,file4*8,file5*11,file6*12,file7*9
      character file8*11,file9*11,file10*10
c    ***
c    <<<End of variable declaration>>>

c    ***Allow two options for defining the study area.
      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 the region by hrap coordinates and number
     3 of 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 the
     1 region of interest:'
      read*, xstart,ystart
      print*,'Enter the number of grid columns and rows to be created:'
      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 the
c    ***number of x coordinates - 1.
      numx = numx1 +1
```

genhrap.for (page 2 of 10):

```
     numy = numy1 +1
print*,'Enter a 3 character suffix to identify your grid:'
     read*, suff
c    ***Create names for all of the output files.
c    ***file2 = file of hrap coordinates
c    ***file3 = file of geocentric coordinates
c    ***file4 = file of geodetic coordinates
c    ***file5 = input file of geod. coordinates to make a polygon coverage
c    ***file6 = file containing HRAP coordinates in a format that can be
c    ***       attached to the polygon attribute table
c    ***file7 = file of polar sterographic coordinates
c    ***file8 = input file of geoc. coordinates to make a polygon coverage
     file2 = 'hrap.'//suff
     file3 = 'geoc.'//suff
c    file4 = 'geod.'//suff
c    file5 = 'inputgd.'//suff
     file6 = 'hrap.'//suff//'.dat'
     file7 = 'pster.'//suff
     file8 = 'inputgc.'//suff
     file9 = 'inpster.'//suff
     file10 = 'inhrap.'//suff
     open (unit = 20, file = file2, status = 'unknown')
     open (unit = 30, file = file3, status = 'unknown')
c    open (unit = 40, file = file4, status = 'unknown')
c    open (unit = 50, file = file5, status = 'unknown')
     open (unit = 60, file = file6, status = 'unknown')
     open (unit = 70, file = file7, status = 'unknown')
     open (unit = 80, file = file8, status = 'unknown')
     open (unit = 90, file = file9, status = 'unknown')
     open (unit = 100, file = file10, 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
```

genhrap.for (page 3 of 10):

```
      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(20,*) count,xhrap(i),yhrap(j)
          count = count + 1
400   continue
300   continue
c     write(*,*) ystart

      call wll(numpts)
c     call togeod(numpts)
      rfunit = 30
      wfunit = 80
      call topoly(numx,numy,rfunit,wfunit)
      rfunit = 70
      wfunit = 90
      call topoly(numx,numy,rfunit,wfunit)
      rfunit = 20
      wrunit = 100
      call topoly(numx,numy,rfunit,wfunit)
      call crdat(numx,numy,xstart,ystart)
      stop
      end
c<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<SUBROUTINES>>>>>>>>>>>>>>>>>>>>>
>>>
c*************************************************************
c Purpose:  Convert HRAP coordinates contained in the file
c     "hrap.***" to lat-long coordinates based on a spherical
c     earth and write them to an output file that can be used
c     to generate a point coverage.  The parameter "numpts"
c     stores the number of entries that will be expected from
c     "hrap.***"
```

genhrap.for (page 4 of 10):

```
c Inputs: file hrap.***
c Outputs: file geoc.***
c
c*****************************************************************

      subroutine wll(numpts)
      double precision xhrap, yhrap, x, y
      double precision bigr, arg, latd, lond, ang
c      parameter (stlond = 105.0)
      double precision stlatd,earthr,mesh,stlond
      integer rec, numpts
c
c***  Define constants
      stlond = -105.0
      stlatd = 60.0
      earthr = 6371200.0
      mesh = 4762.5

      rewind(unit=20)

      do 100 i=1,numpts
c***  Modified to allow for the record number that was included in the
c***  hrap.** file.
c***  Define the polar stereographic coordinates for the same area.
c***  Multiply by 1000 to change from km to m
        read(20,*) 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)
c       write(50,*) ang
c       write(*,*) ang
c       write(*,*) x,y
c
        if (y.gt.0) then
          ang = 270.0-stlond-ang
        else
          ang = -90.0-stlond-ang
```

genhrap.for (page 5 of 10):

```
      endif
if (ang.lt.180) then
        lond = -1 * ang
      else
        lond = 360.0 - ang
      endif
c***  Write polar stereographic coordinates and geocentric coordinates
c***  to a file.
      write(70,*) i,x,y
      write(30,*) i,lond,latd
100   continue
     return
     end

c*********************************************************************
c Purpose: Convert geocentric coordinates to geodetic coordinates.
c Input: File containing rows: ID,lon,geoclat.
c Output:  File containing rows: ID,lon,geodlat
c*********************************************************************

      subroutine togeod(numpts)

      double precision grse2,cle2,const

      double precision lond,latd,alatd
      integer i,j,numpts

c*** Unused variables grse2 = eccentricity squared for GRS80
c*         cle2 = eccentricity squared for Clarke 1866
      grse2 = 0.0066943800
      cle2 = 0.006768658
      const = 1.00673950
      rewind(unit=30)
      do 100 i=1,numpts
       read(30,*) j,lond,latd
       alatd = datand(const*dtand(latd))
c      diff = alatd-latd
write (40,*) j,lond,alatd
100   continue
     return
     end
c*********************
```

genhrap.for (page 6 of 10):

```
c Purpose: Given a list of corner point for a grid (can be (ID,x,y) or
c  (ID,lat,lon) in which the coordinates for the bottom row are listed
c  one per line followed by the coordinates for the next row up,
c  create a file that can be used to generate a polygon coverage
c  of the grid cells.
c Input: File of corner points (ID,lat,lon),
c Output: 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**********************

      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 or 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
```

genhrap.for (page 7 of 10):

```
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
c  to the projected "hrap" polygon coverage so that "hrap" coordinates
c  of the lower left hand corner of each polygon will be added to the
c  appropriate line in the PAT
c Note:  The file "hrap.***.dat" contains the coordinates of all
c   corner 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
```

genhrap.for (page 8 of 10):

```
      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(60,*) 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 Note:  The user should input geodetic coordinates.  These
c   geodetic coordinates will be interpreted as geocentric
c   coordinates 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)
```

genhrap.for (page 9 of 10):

```
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
c***   of the lower left hand corner desired.
    real minhx,minhy,maxhx,maxhy
c     <<<End of 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
   4 West longitude values as negative numbers.'
    do 100 i = 1,4
      read*,lon(i),lat(i)
    sfactor = (1+sind(stlat))/(1+sind(lat(i)))

    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(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
```

genhrap.for (page 10 of 10):

```
      xstart = minhx
      ystart = minhy

      numx1 = maxhx - minhx
      numy1 = maxhy - minhy
      write (*,*) xstart,ystart,numx1,numy1
      return
      end

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

mapab.for (page 1 of 2):

```
    PROGRAM MAPAB
C
C  THIS PROGRAM WORKS WITH PRECIP.AML TO MAP
C   GRIDS OF STAGE III PRECIPITATION.
C
    PARAMETER (NVARS= 1) !NUMBER OF VARIABLES
C
    INTEGER*4 RCODE
C
C    VARIABLES FOR THIS NETCDF FILE
C
    INTEGER*2 AMOUNTOFPRECIP (335,153)
    CHARACTER*15 FLNM
    INTEGER HRAPX(1000), HRAPY(1000), ID(1000), PRECIP(1000)
    INTEGER N, NCELLS, I, X, Y
C
    INTEGER*4 START(10)
    INTEGER*4 COUNT(10)
    INTEGER VDIMS(10) !ALLOW UP TO 10 DIMENSIONS
    CHARACTER*31 DUMMY
C
C  OPEN THE TWO INPUT FILES
C
    OPEN (17,FILE='input',status='unknown',form='formatted')
    OPEN (18,FILE='hrapinput',STATUS='UNKNOWN',FORM='FORMATTED')
C
C READ THE ID'S AND COORDS FROM HRAPINPUT INTO THREE SEPARATE
C  ARRAYS.  COUNT THE NUMBER OF VALUES, SET THAT EQUAL TO
NCELLS.
C
    N = 1
  10 READ(18,*) ID(N), HRAPX(N), HRAPY(N)
    IF (ID(N).EQ.0) THEN
     NCELLS = N - 1
     GOTO 25
    ENDIF
    N = N + 1
    GOTO 10
C
C  READ IN THE FILENAMES.  IF FILENAME = DONE, THEN QUIT
C
  25 READ(17,*) FLNM
```

mapab.for (page 2 of 2):

```
      IF (FLNM.EQ.'done') THEN
        GOTO 100
      ENDIF
C
      NCID=NCOPN(FLNM,NCNOWRIT,RCODE)
C     statements to fill amountofprecip

      CALL NCVINQ(NCID, 1,DUMMY,NTP,NVDIM,VDIMS,NVS,RCODE)
      LENSTR=1
      DO  40 J=1,NVDIM
      CALL NCDINQ(NCID,VDIMS(J),DUMMY,NDSIZE,RCODE)
      LENSTR=LENSTR*NDSIZE
      START(J)=1
      COUNT(J)=NDSIZE
  40  CONTINUE
      CALL NCVGT(NCID, 1,START,COUNT,
     +amountofprecip,RCODE)
      CALL NCCLOS(NCID, RCODE)
C
C     MANIPULATION OF DATA
C
      OPEN (14, FILE='out.'//FLNM,STATUS='NEW',FORM='FORMATTED')
      DO 50 I=1,NCELLS
          X = HRAPX(I) - 366
          Y = HRAPY(I) - 262
          PRECIP(I) = AMOUNTOFPRECIP(x,y)
        WRITE(14,*) ID(I),',',PRECIP(I)
   50 CONTINUE
C
      CLOSE (14)
C
      GOTO 25
C
  100 STOP
      END
```

mapmb.for (page 1 of 2):

```
      PROGRAM MAPMB
C
C        THIS PROGRAM WORKS IN SYNC WITH PRECIP.AML.
C
      INTEGER*2 PRECIPDATA (166,69)
      INTEGER  HRAPX(1000), HRAPY(1000), ID(1000), PRECIP(1000)
      INTEGER  N, NCELLS, I, X, Y
      CHARACTER*15 FLNM
C
C OPEN THE TWO INPUT FILES, THE FIRST CONTAINING THE NAMES OF THE
C  PRECIP FILES, AND THE SECOND CONTAINING THE ID'S AND COORDS
C  OF THE CELLS IN THE WATERSHED.  ALSO OPEN AN OUTPUT FILE FOR
C  THE PRECIP ARRAY.
C
      OPEN (17,FILE='input',status='unknown',form='formatted')
      OPEN (18, FILE='hrapinput',STATUS='UNKNOWN', FORM='FORMATTED')
C
C READ THE ID'S AND COORDS FROM HRAPINPUT INTO THREE SEPARATE
C  ARRAYS.  COUNT THE NUMBER OF VALUES, SET THAT EQUAL TO
NCELLS.
C
      N = 1
   10 READ(18,*) ID(N), HRAPX(N), HRAPY(N)
      IF (ID(N).EQ.0) THEN
        NCELLS = N - 1
        GOTO 25
      ENDIF
      N = N + 1
      GOTO 10
C
C READ THE FILENAME FROM THE INPUT FILE.  IF THE FILENAME IS
C  'DONE', THEN STOP THE PROGRAM.
C
   25 READ(17,*) FLNM
      IF (FLNM.EQ.'done') THEN
        GOTO 100
      ENDIF
C OPEN THE PRECIP FILE, READ IN THE PRECIPDATA ARRAY.
      OPEN (13, FILE=FLNM, STATUS='OLD', FORM='UNFORMATTED')
      OPEN (14, FILE='out.'//FLNM, STATUS='NEW',FORM='FORMATTED')
      READ (13) PRECIPDATA
C SAVE THE PRECIPDATA VALUES FOR THE CELLS IN THE
```

mapmb.for (page 2 of 2):

```
C  WATERSHED INTO THE PRECIP ARRAY.  CALL THE PRINT
C  SUBROUTINE TO PRINT THE ARRAY TO THE OUTPUT FILE.
      DO 70 I = 1, NCELLS
        X = HRAPX(I) - 459
        Y = HRAPY(I) - 386
      PRECIP(I) = PRECIPDATA(X,Y)
        WRITE(14,*) ID(I),',',PRECIP(I)
   70 CONTINUE
C
      CLOSE (13)
      CLOSE (14)
      GOTO 25
C
  100 STOP
      END
```

mapmbday.for (page 1 of 2):

```
      PROGRAM MAPMBDAY
C
C        THIS PROGRAM WORKS IN SYNC WITH PRECIP.AML.
C
       INTEGER*2 PRECIPDATA (166,69)
       INTEGER  HRAPX(1000), HRAPY(1000), ID(1000), PRECIP(1000)
       INTEGER  DAILYSUM(1000)
       INTEGER  N, NCELLS, I, X, Y
       CHARACTER*15 FLNM,DATE
C
C OPEN THE TWO INPUT FILES, THE FIRST CONTAINING THE NAMES OF THE
C  PRECIP FILES, AND THE SECOND CONTAINING THE ID'S AND COORDS
C  OF THE CELLS IN THE WATERSHED.  ALSO OPEN AN OUTPUT FILE FOR
C  THE PRECIP ARRAY.
C
       OPEN (17,FILE='input',status='unknown',form='formatted')
       OPEN (18, FILE='hrapinput',STATUS='UNKNOWN', FORM='FORMATTED')
C
C READ THE ID'S AND COORDS FROM HRAPINPUT INTO THREE SEPARATE
C  ARRAYS.  COUNT THE NUMBER OF VALUES, SET THAT EQUAL TO
NCELLS.
C
       N = 1
    10 READ(18,*) ID(N), HRAPX(N), HRAPY(N)
       IF (ID(N).EQ.0) THEN
        NCELLS = N - 1
        GOTO 25
       ENDIF
       N = N + 1
       GOTO 10
    25 C = 1
C
C  CREATE AN OUTPUT FILE FOR EACH DAY THAT IS CALLED
C  OUTDAYdate
C
    30 OPEN (15,FILE='days',STATUS='UNKNOWN',FORM='FORMATTED')
        READ (15,*) DATE
        OPEN (14,FILE='out.'//DATE)
        DO 20 I=1,NCELLS
           DAILYSUM(I) = 0
    20 CONTINUE
        DO 70 H=1,24
```

mapmbday.for (page 2 of 2):

```
C
C READ THE FILENAME FROM THE INPUT FILE.  THIS IS DONE 24 TIMES
C FOR EACH DAILY OUTPUT FILE.  IF THE FILENAME IS
C  'DONE', THEN STOP THE PROGRAM.
C
        READ(17,*) FLNM
        IF (FLNM.EQ.'done') THEN

       GOTO 100


        ENDIF
C
C OPEN THE PRECIP FILE, READ IN THE PRECIPDATA ARRAY.
C
        OPEN (13, FILE=FLNM, STATUS='OLD', FORM='UNFORMATTED')
        READ (13) PRECIPDATA
C
C SAVE THE PRECIPDATA VALUES FOR THE CELLS IN THE
C  WATERSHED INTO THE PRECIP ARRAY.  CALL THE PRINT
C  SUBROUTINE TO PRINT THE ARRAY TO THE OUTPUT FILE.
C  THE RAINFALL VALUE FOR THE CURRENT HOUR (PRECIP)
C   IS ADDED TO THE CURRENT CUMULATIVE RAINFALL (DAILYSUM).
C  THIS NEW RAINFALL VALUE IS WRITTEN TO THE ARRAY IN
C   THE DAILY OUTPUT FILE.
C
        DO 50 I = 1, NCELLS
        X = HRAPX(I) - 459
        Y = HRAPY(I) - 386
        PRECIP(I) = PRECIPDATA(X,Y)
          DAILYSUM(I) = DAILYSUM(I) + PRECIP(I)
  50    CONTINUE
C
        CLOSE (13)
  70 CONTINUE
    DO 90 I = 1,NCELLS
     WRITE(14,*) ID(I),',',DAILYSUM(I)
  90 CONTINUE
    CLOSE (14)
    GOTO 30
C
 100 STOP
    END
```

prepab.for (page 1 of 2):

```
      PROGRAM PREPAB
C
C  THIS PROGRAM WORKS WITH ONECELL.AML TO GENERATE A LIST
C   OF RAINFALL VALUES FOR ONE HRAP CELL.
C
      PARAMETER (NVARS= 1) !NUMBER OF VARIABLES
C
      INTEGER*4 RCODE
C
C    VARIABLES FOR THIS NETCDF FILE
C
      INTEGER*2 AMOUNTOFPRECIP (335,153)
      CHARACTER*15 FLNM
      INTEGER HRAPX(1000), HRAPY(1000), ID(1000), PRECIP(1000)
      INTEGER N, NCELLS, I, X, Y
C
      INTEGER*4 START(10)
      INTEGER*4 COUNT(10)
      INTEGER VDIMS(10) !ALLOW UP TO 10 DIMENSIONS
      CHARACTER*31 DUMMY
C
C  OPEN THE TWO INPUT AND ONE OUTPUT FILE
C
      OPEN (17,FILE='input',status='unknown',form='formatted')
      OPEN (18,FILE='hrapinput',STATUS='UNKNOWN',FORM='FORMATTED')
      OPEN (14,FILE='precip',STATUS='UNKNOWN',FORM='FORMATTED')
C
C READ THE ID'S AND COORDS FROM HRAPINPUT INTO THREE SEPARATE
C  ARRAYS.  COUNT THE NUMBER OF VALUES, SET THAT EQUAL TO
NCELLS.
C
      N = 1
   10 READ(18,*) ID(N), HRAPX(N), HRAPY(N)
      IF (ID(N).EQ.0) THEN
        NCELLS = N - 1
        GOTO 25
      ENDIF
      N = N + 1
      GOTO 10
C
C  READ IN THE FILENAMES.  IF FILENAME = DONE, THEN QUIT
C
```

prepab.for (page 2 of 2):

```
  25 READ(17,*) FLNM
     IF (FLNM.EQ.'done') THEN
       GOTO 100
     ENDIF
     NCID=NCOPN(FLNM,NCNOWRIT,RCODE)
C    statements to fill amountofprecip
C
     CALL NCVINQ(NCID, 1,DUMMY,NTP,NVDIM,VDIMS,NVS,RCODE)
     LENSTR=1
     DO  40 J=1,NVDIM
     CALL NCDINQ(NCID,VDIMS(J),DUMMY,NDSIZE,RCODE)
     LENSTR=LENSTR*NDSIZE
     START(J)=1
     COUNT(J)=NDSIZE
  40  CONTINUE
     CALL NCVGT(NCID, 1,START,COUNT,
    +amountofprecip,RCODE)
     CALL NCCLOS(NCID, RCODE)
C
C    MANIPULATION OF DATA
C
     DO 50 I=1,NCELLS
         X = HRAPX(I) - 366
         Y = HRAPY(I) - 262
         PRECIP(I) = AMOUNTOFPRECIP(x,y)
  50 CONTINUE
C
     CALL PRINT(NCELLS, PRECIP)
C
     GOTO 25
C
 100 STOP
     END
C
C---------------------------------------------
C
     SUBROUTINE PRINT(NCELLS, P)
     INTEGER NCELLS, P(NCELLS)
C
     WRITE(14,*) P
     RETURN
     END
```

prepmb.for (page 1 of 2):

```
      PROGRAM PREPMB
C
C       THIS PROGRAM WORKS IN SYNC WITH ONECELL.AML TO
C         GENERATE A LIST OF RAINFALL VALUES FOR ONE
C         HRAP CELL
C
C  INPUT:  INPUT -- CONTAINS THE NAMES OF THE PRECIP FILES.
C     HRAPINPUT -- CONTAINS THE ID'S AND HRAP COORDINATES.
C
      INTEGER*2 PRECIPDATA (166,69)
      INTEGER  HRAPX(1000), HRAPY(1000), ID(1000), PRECIP(1000)
      INTEGER  N, NCELLS, I, X, Y
      CHARACTER*15 FLNM
C
C OPEN THE TWO INPUT FILES, THE FIRST CONTAINING THE NAMES OF THE
C  PRECIP FILES, AND THE SECOND CONTAINING THE ID'S AND COORDS
C  OF THE CELLS IN THE WATERSHED.  ALSO OPEN AN OUTPUT FILE FOR
C  THE PRECIP ARRAY.
C
      OPEN (17,FILE='input',status='unknown',form='formatted')
      OPEN (18, FILE='hrapinput',STATUS='UNKNOWN', FORM='FORMATTED')
      OPEN (14, FILE='precip', STATUS='NEW',FORM='FORMATTED')
C
C READ THE ID'S AND COORDS FROM HRAPINPUT INTO THREE SEPARATE
C  ARRAYS.  COUNT THE NUMBER OF VALUES, SET THAT EQUAL TO
NCELLS.
C
      N = 1
   10 READ(18,*) ID(N), HRAPX(N), HRAPY(N)
      IF (ID(N).EQ.0) THEN
        NCELLS = N - 1
        GOTO 25
      ENDIF
      N = N + 1
      GOTO 10
C
C READ THE FILENAME FROM THE INPUT FILE.  IF THE FILENAME IS
C  'DONE', THEN STOP THE PROGRAM.
C
   25 READ(17,*) FLNM
      IF (FLNM.EQ.'done') THEN
        GOTO 100
```

prepmb.for (page 2 of 2):

```
     ENDIF
C OPEN THE PRECIP FILE, READ IN THE PRECIPDATA ARRAY.
     OPEN (13, FILE=FLNM, STATUS='OLD', FORM='UNFORMATTED')
     READ (13) PRECIPDATA
C SAVE THE PRECIPDATA VALUES FOR THE CELLS IN THE
C  WATERSHED INTO THE PRECIP ARRAY.  CALL THE PRINT
C  SUBROUTINE TO PRINT THE ARRAY TO THE OUTPUT FILE.
C
     DO 70 I = 1, NCELLS
          X = HRAPX(I) - 459
          Y = HRAPY(I) - 386
       PRECIP(I) = PRECIPDATA(X,Y)
  70 CONTINUE
     CALL PRINT(NCELLS, PRECIP)
C
     CLOSE (13)
     GOTO 25
C
 100 STOP
     END
C
C---------------------------------------------------------
C
C  THE PRINT SUBROUTINE WRITES THE PRECIP ARRAY TO THE
C   OUTPUT FILE.  A SUBROUTINE IS USED TO PRINT ONLY NCELLS
C   VALUES TO THE FILE.
C
     SUBROUTINE PRINT(NCELLS, P)
     INTEGER NCELLS, P(NCELLS)
     WRITE(14,*) P
     RETURN
     END
C
C---------------------------------------------------------
C
```

readabrfc.for (page 1 of 1):

```
      PROGRAM READABRFC
C
C THIS PROGRAM DEMONSTRATES THE USE OF NETCDF LIBRARIES
C   TO READ AN ARRAY OF STAGE III DATA FROM THE ABRFC
C   HOURLY RAINFALL FILES.  IT SHOULD BE NOTED THAT THE
C   LOCATION OF 1,1 IN THIS ARRAY IS HRAPX = 367, HRAPY = 263.
      PARAMETER (NVARS= 1) !NUMBER OF VARIABLES
C
      INTEGER*4 RCODE
C
C    VARIABLES FOR THIS NETCDF FILE
C
      INTEGER*2 AMOUNTOFPRECIP (335,153)
      CHARACTER*15 FLNM
      INTEGER*4 START(10)
      INTEGER*4 COUNT(10)
      INTEGER VDIMS(10) !ALLOW UP TO 10 DIMENSIONS
      CHARACTER*31 DUMMY
C
C ENTER THE NAME OF THE STAGE III FILE.
C
      PRINT*, 'ENTER THE NAME OF THE STAGE III FILE'
      READ*, FLNM
      NCID=NCOPN(FLNM,NCNOWRIT,RCODE)
C
C    statements to fill amountofprecip
C
      CALL NCVINQ(NCID, 1,DUMMY,NTP,NVDIM,VDIMS,NVS,RCODE)
      LENSTR=1
      DO  40 J=1,NVDIM
      CALL NCDINQ(NCID,VDIMS(J),DUMMY,NDSIZE,RCODE)
      LENSTR=LENSTR*NDSIZE
      START(J)=1
      COUNT(J)=NDSIZE
   40 CONTINUE
      CALL NCVGT(NCID, 1,START,COUNT,
     +amountofprecip,RCODE)
      CALL NCCLOS(NCID, RCODE)
C
      STOP
      END
```

readmbrfc.for (page 1 of 1):

```
      PROGRAM READMBRFC
C
C THIS PROGRAM DEMONSTRATES HOW THE MBRFC FILES ARE READ
C  INTO AN ARRAY.  THE ORIGIN OF THE ARRAY (1,1) EQUALS
C  HRAPX = 235, HRAPY = 375.
C
      INTEGER*2 AMOUNTOFPRECIP (485,325)
      DIMENSION MOSAIC(485)
      INTEGER*2 MOSAIC
      INTEGER  XOR, YOR, MX, MY, JROW, I, J
      CHARACTER*13 FLNM
C
C ENTER THE FILE NAME
C
      PRINT*, 'ENTER THE FILE NAME'
      READ*, FLNM
C
C       OPEN THE XMRG FILE, READ THE INDIVIDUAL MOSAICS (1-D
C         ARRAYS) INTO AMOUNTOFPRECIP (2-D ARRAY).  NOTE:  THE
C         XMRG FILE STORES THE RAINFALL DATA IN AN ARRAY MOSAIC.
C         THIS 1-D ARRAY STORES DATA FOR A GIVEN HRAPY COORDINATE
C         FOR ALL 485 HRAPX COORDINATES.
C
      OPEN (13, FILE=FLNM,
     +     STATUS='OLD', FORM='UNFORMATTED')
      READ (13) XOR, YOR, MX, MY
      J=0
      DO 50 JROW=2,325,1
          J=J+1
          READ (13) MOSAIC
          DO 40 I=1,485,1
            AMOUNTOFPRECIP(I,J) = MOSAIC(I)
   40     CONTINUE
   50 CONTINUE
C
C MANIPULATE THE ARRAY AS DESIRED.
C
      STOP
      END
```

readreduced.for (page 1 of 1):

```
      PROGRAM READREDUCED.FOR
C
C       THIS PROGRAM READS THE REDUCED ARRAY OF RAINFALL
C        VALUES FROM REDUCED MBRFC FILES (COVERING KANSAS ONLY)
C        NOTE THAT THE ORIGIN OF THE ARRAY (1,1) EQUALS
C        HRAPX = 460, HRAPY = 387.
C
      INTEGER*2 PRECIPDATA (166,69)
      CHARACTER*15 FLNM
C
C  ENTER THE FILENAME
C
      PRINT*, 'ENTER THE NAME OF THE FILE'
      READ*, FLNM
C
C OPEN THE PRECIP FILE, READ IN THE PRECIPDATA ARRAY.
C
      OPEN (13, FILE=FLNM, STATUS='OLD', FORM='UNFORMATTED')
      READ (13) PRECIPDATA
C
C  MANIPULATION OF DATA
C
      STOP
      END
```

reducembrfc.for (page 1 of 2):

```
      PROGRAM REDUCEMBRFC
C
C       THIS PROGRAM WORKS IN SYNC WITH MONTHLY2.AML TO
C     REDUCE THE FILE SIZE OF THE MBRFC STAGE III FILES.
C
      INTEGER*2 AMOUNTOFPRECIP (485,325)
      INTEGER*2 PRECIPDATA (166,69)
      DIMENSION MOSAIC(485)
      INTEGER*2 MOSAIC
      INTEGER  XOR, YOR, MX, MY, JROW, I, J
      INTEGER  X, Y
      INTEGER  XMIN, YMIN, XMAX, YMAX, A, B
      CHARACTER*13 FLNM
C
C       OPEN THE FILE CREATED BY THE PRECIPGRID.AML.  ADJUST THE
C         MIN AND MAX COORDINATES TO THE LOCAL GRID SYSTEM
C         (AMOUNTOFPRECIP ARRAY SETS ORIGIN OF LOCAL GRID SYSTEM
C         AT 235,375)
      OPEN (16,FILE='reduce.input',status='unknown',form='formatted')
      print*, 'file opened'
      XMIN = 460
      YMIN = 387
      XMAX = 625
      YMAX = 455
      MINX=XMIN-234
      MAXX=XMAX-234
      MINY=YMIN-374
      MAXY=YMAX-374
    5 READ(16,*) FLNM
      IF (FLNM.EQ.'done') THEN
        GOTO 100
      ENDIF
C       OPEN THE XMRG FILE, READ THE INDIVIDUAL MOSAICS (1-D
C         ARRAYS) INTO AMOUNTOFPRECIP (2-D ARRAY).  NOTE:  THE
C         XMRG FILE STORES THE RAINFALL DATA IN AN ARRAY MOSAIC.
C         THIS 1-D ARRAY STORES DATA FOR A GIVEN HRAPY COORDINATE
C         FOR ALL 485 HRAPX COORDINATES.
      OPEN (13, FILE=FLNM,
     +    STATUS='OLD', FORM='UNFORMATTED')
      READ (13) XOR, YOR, MX, MY
      J=0
      DO 50 JROW=2,325,1
```

reducembrfc.for (page 2 of 2):

```
        J=J+1
        READ (13) MOSAIC
        DO 40 I=1,485,1
         AMOUNTOFPRECIP(I,J) = MOSAIC(I)
  40  CONTINUE
  50 CONTINUE
C
C       OPEN THE OUTPUT FILE AND DUMP THE DESIRED PRECIP DATA
C        INTO THIS FILE.
C
     OPEN (17,FILE='r.'//FLNM,STATUS='unknown',FORM='unFORMATTED')
     A=1
     B=1
     DO 70 Y=MINY,MAXY
       DO 60 X=MINX,MAXX
       PRECIPDATA(A,B)=AMOUNTOFPRECIP(X,Y)
         A=A+1
  60   CONTINUE
      A=1
        B=B+1
  70 CONTINUE
     WRITE (17) PRECIPDATA
     GOTO 5
C
 100 STOP
     END
```

Appendix C

ARC/INFO AML, Menu, and Projection Files

Files are listed in alphabetical order.

boundclipcov.menu (page 1 of 1):

```
7 boundclipcov.menu
/*
  Choose the coverage that
  contains the boundary to clip to:
   %datalist1
                    %button1
%datalist1 INPUT COV 14 TYPEIN NO SCROLL YES ROWS 4 COVER ~
 * -ALL -SORT
%button1 BUTTON KEEP 'Return' &return
```

calcslope.aml (page 1 of 1):

```
/* calcslope.aml

/* This aml calculates the slope from the ws%dem%fil grid

&term 9999
&messages &popup

/* Enter the base name of the DEM

&sv dem = %.name%

/* Check to see if the ws%dem%fil grid exists

&if [exists ws%dem%fil -grid] eq .FALSE. &then
 &do
  &type Clipped elevation grid does not exist
  &return
 &end

/* calculate the slopes

&messages &on
grid
&r msworking init 'Calculating Slope Grid' 'Waiting Time = SHORT'~
 '&cc' 'calcslope.aml'
 slopegrid = slope(ws%dem%fil, percentrise)

/* Convert to a decimal

 ws%dem%slope = slopegrid / 100

/* kill the slopegrid

 kill slopegrid

/* quit and return
&r msworking close
quit
&r msinform init 'The following Grid has been Created:' ws%dem%slope~
 '&cc' 'calcslope.aml'

&return
```

clipcov.menu (page 1 of 1):

```
7 clipcov.menu
/*

   Select a coverage to clip:
   %datalist1
                 %button1
%datalist1 INPUT COV 14 TYPEIN NO SCROLL YES ROWS 4 COVER ~
  * -ALL -SORT
%button1 BUTTON KEEP 'Return' &return
```

clipdem.aml (page 1 of 3):

```
/* clipdem.aml

/* This aml clips the dem to a smaller area.

&TERMINAL 9999
&MESSAGES &popup

/* Check for the existance of files necessary for the
/*  operation of this aml.

&if [exists %.name%prj -grid] ne .TRUE. &then
 &do
  &return &inform The %.name%prj grid does not exist
 &end

/* Clip the dem.

/* Enter arcedit, set the backcoverages (coverages
/*  to be viewed but not edited) and the backenvironment.

&messages &on

arcedit
  display 9999
  mape %.name%prj
  image %.name%prj
  image on
  drawenvironment poly on
/* Ask the user if any other coverages should be viewed.
&sv cover = [getchoice YES NO -prompt 'Do you wish to display a background~
 coverage?']

backe poly on
backe arc on
backe tic on

&do &while %cover% eq YES
 &sv covname = [getcov]
 &messages &pop
 &sv color = [response 'Enter the color (1=white, 2=red, 3=green, 4=blue):']
 &messages &on
 backcoverage %covname% %color%
```

clipdem.aml (page 2 of 3):

```
&sv cover = [getchoice YES NO -prompt 'Do you wish to display another~
 coverage?']
&end
/*  Create a new grid called clipgrid1.  First, ask the user which
/*    coverage the polygon will resemble.

&sv covtics = [getcov # # 'Select the Coverage whose Tic Cover will be~
 used for the new Polygon Coverage:']
 create gridclip1 %covtics%
 build
 editfeature poly

/*  Inform the user to draw the polygon.

&messages &pop

&type After quitting this window, draw a polygon that encompasses
&type the watershed of interest.  The dem will be clipped to this
&type polygon.  Enter 2 when done.

&messages &on

  draw
  add one
  save
  quit

/*  Convert the polygon to a grid

&do
 polygrid gridclip1 gridclip2
 30
 y
&end

/*  enter grid, set the value of gridclip2 to one, then multiply it
/*    by the prj grid

grid
 gridclip3 = (gridclip2 / gridclip2) * %.name%prj
 copy %.name%prj %.name%save
 kill %.name%prj
```

clipdem.aml (page 3 of 3):

```
%.name%prj = gridclip3
 kill gridclip3
 kill gridclip2
 kill gridclip1
quit

&return
```

clipgrid.aml (page 1 of 2):

```
/* clipgrid.aml


/* This aml uses the watershed grid to clip the fil,
/*   fdr, and fac grids.


&term 9999
&messages &popup

/* Enter the base name of the DEM

&sv dem = %.name%

/* Check to see if the watershed grid exists

&if [exists %dem%ws -grid] eq .FALSE. &then
 &do
  &type Flowdirection grid does not exist
  &return
 &end

/* Check to see if the flowdirection grid exists

&if [exists %dem%fdr -grid] eq .FALSE. &then
 &do
  &type Flowdirection grid does not exist
  &return
 &end

/* Check to see if the flowaccumulation grid exists

&if [exists %dem%fac -grid] eq .FALSE. &then
 &do
  &type Flowaccumulation grid does not exist
  &return
 &end

/* Check to see if the filled grid exists

&if [exists %dem%fil -grid] eq .FALSE. &then
```

clipgrid.aml (page 2 of 2):

```
&do
   &type Flowdirection grid does not exist
   &return
  &end
/* Enter grid, display the ws grid and zoom in the analysis window.
&messages &on
grid
  display 9999
  mape %dem%ws
  gridp %dem%ws
  &type Select the Analysis Window
  setwindow *

&r msworking init 'Clipping the FIL, FDR, and FAC grids to the WSHED Boundary.'~
 'Waiting Time = MEDIUM' '&cc' 'clipgrid.aml'

  %dem%wshed = %dem%ws

/* Convert the wshed grid to a clip grid by converting all the
/*    values in the wshed grid to 1.

%dem%clip = %dem%wshed / %dem%wshed

/* Clip the grids

ws%dem%fil = %dem%clip * %dem%fil
ws%dem%fdr = %dem%clip * %dem%fdr
ws%dem%fac = %dem%clip * %dem%fac

/* quit and return

&r msworking close

quit
&return
```

```
/* contour.aml

/* This aml individually contours each hour's or a storm totals
/*  grid, permitting the user to interactively
/*  select a contour interval.  Once the user
/*  has chosen the interval to be used, the
/*  lattice is contoured and clipped to
/*  the watershed boundary

&TERMINAL 9999
&MESSAGES &popup
&sv option = [getchoice 'INDIVIDUAL HOURS' 'STORM TOTALS' -prompt~
 'Contour Rainfall Data For:']

/* If individual hours are to be contoured, then enter the beginning
/*   and end hour you wish to contour.  If you are contouring
/*   storm totals, enter the number of hours in the storm.

&if %option% eq 'INDIVIDUAL HOURS' &then
  &do
        &if %.basin% = mbrfc and %time% = Daily &then &sv confile = days
        &else &sv confile = input
        &menu contour.menu &stripe 'Contour for '%.storm%''
        &if %done% = done &then &return
 &end
&else
&sv hours = [response 'Enter the numbers of hours in the storm:']

/* Select the Desired Units.

&sv units = [getchoice INCHES MM -prompt 'Select the Desired Units:']

/*-------------------------------------------------------------
/*       FOR THE INDIVIDUAL HOUR OPTION:
/* For each grid, check if it exists, smooth the grid, display the
/*   .sta file, prompt the user for a contour interval.
/*   if the units are inches, then convert the interval into
/*   hundreths of a mm.  If the units are mm, then multiply by
/*   100.  Contour the data, then display and ask the user
/*   if the chosen interval is okay.  If not, redo.  Otherwise,
/*   clip the contours to the watershed boundary and then
/*   alter the tables to the correct units.
```

```
/*------------------------------------------------------------
&if %option% eq 'INDIVIDUAL HOURS' &then
  &do

        &sv hour = 1
        &do &while [value file%hour%] ne done
          &messages &popup

          &if [exists %.storm%%.ext%%hour% -grid] ne .true. &then
          &return &inform Grid of precip does not exist.

          list %.storm%%.ext%%hour%.sta

           filter %.storm%%.ext%%hour% %.storm%%.ext%%hour%s low

          &sv okay = no

          &do &while %okay% ne YES

                &messages &pop

                &sv interval = [response 'Enter the contour interval desired (in~
                previously selected units)']

                &if %units% eq INCHES &then
                &sv interval = %interval% * 2540
                &else &sv interval = %interval% * 100

                latticecontour %.storm%%.ext%%hour%s %.storm%%.ext%%hour%c
 %interval% 1

                &messages &on

 grid
  display 9999
  mape %.storm%%.ext%%hour%c
  arclines ws%.name% 1
  arclines %.storm%%.ext%%hour%c 3
  &sv okay = [getchoice YES NO -prompt~
 'Is the contour interval acceptable?']
  &if %okay% eq NO &then kill %.storm%%.ext%%hour%c
 quit
```

```
        &end

        clip %.storm%%.ext%%hour%c ws%.name% ws%.storm%%.ext%%hour%c
line

        kill %.storm%%.ext%%hour%c
        kill %.storm%%.ext%%hour%s

tables
 &if %units% eq INCHES &then
  &do
   additem ws%.storm%%.ext%%hour%c.aat precip 4 5 n 2
   sel ws%.storm%%.ext%%hour%c.aat
   calc precip = contour / 2540
  &end
 &else
  &do
   additem ws%.storm%%.ext%%hour%c.aat precip 4 5 b
   sel ws%.storm%%.ext%%hour%c.aat
   calc precip = contour / 2540
  &end
quit

       &end
  &end

/*----------------------------------------------------------
/*          FOR THE STORM TOTALS OPTION

&else
  &do

/* Sum up the individual Hourly Grids

       &messages &on

grid

total2 = %.storm%%.ext%1

       &if %hours% gt 1 &then
         &do i = 2 &to %hours% &by 1
```

contour.aml (page 4 of 5)

```
                &sv n = %i% + 1
total%n% = %.storm%%.ext%%i% + total%i%
kill total%i%
            &end
        &else &sv n = 2

        %.storm%total = total%n%
kill total%n%

quit

list %.storm%total.sta

filter %.storm%total %.storm%totals low

        &sv okay = no

        &do &while %okay% ne YES

          &messages &pop

          &sv interval = [response 'Enter the contour interval desired (in~
          previously selected units)']

          &if %units% eq INCHES &then
          &sv interval = %interval% * 2540
          &else &sv interval = %interval% * 100

latticecontour %.storm%totals %.storm%totalc %interval% 1

          &messages &on

grid
 display 9999
 mape %.storm%totalc
 arclines ws%.name% 2
 arclines %.storm%totalc 3
 &sv okay = [getchoice YES NO -prompt 'Is the contour interval acceptable?']
 &if %okay% eq NO &then kill %.storm%totalc
quit

        &end
```

contour.aml (page 5 of 5)

```
clip %.storm%totalc ws%.name% ws%.storm%totalc line

kill %.storm%totalc
kill %.storm%totals

tables
 &if %units% eq INCHES &then
  &do
   additem ws%.storm%totalc.aat precip 5 5 n 2
   sel ws%.storm%totalc.aat
   calc precip = contour / 2540
  &end
 &else
  &do
   additem ws%.storm%totalc.aat precip 4 5 b
   sel ws%.storm%totalc.aat
   calc precip = contour / 2540
  &end
quit

   &end

&return
```

copymb.aml (page 1 of 3):

```
/* copymb.aml

/* This aml copies the required mbrfc rainfall files from the
/*   /files/mbrfc/archives/reduced folder to the current directory.

&TERMINAL 9999
&MESSAGES &popup

/*  Enter the start and end times of the storm.

&sv start = [response 'Enter the Start Date and Time (MM DD YY HH):']
&sv end = [response 'Enter the End Date and Time (MM DD YY HH):']

&messages &on

/* Call the precipfiles subroutine, which will create the
/*  array with the names of the precip files to be
/*  copied to the local directory.

&call precipfiles

/* For each hourly file, copy the file to the local directory.

&setvar number = 1

&do &while [value file%number%] ne done
 cp /files/mbrfc/archives/reduced/[value file%number%] [value file%number%]
 &sv number = %number% + 1
&end

&return

/*------------------------------------------------------------
/*------------------------------------------------------------
/* The following routine creates a file with all of the file names
/*  to be opened.  A similar list is stored in the array file%number%

&routine Precipfiles

/* Set the beginning and end month, day, year, and hour of the storm.
/*   Use the extract function to accomplish this.
```

copymb.aml (page 2 of 3):

```
&sv st = [unquote %start%]
&sv bMM = [extract 1 %st%]
&sv bDD = [extract 2 %st%]
&sv bYY = [extract 3 %st%]
&sv bHH = [extract 4 %st%]
&sv en = [unquote %end%]
&sv eMM = [extract 1 %en%]
&sv eDD = [extract 2 %en%]
&sv eYY = [extract 3 %en%]
&sv eHH = [extract 4 %en%]

/* Initialize the MM, DD, YY, and HH values.

&sv MM = %bMM%
&sv DD = %bDD%
&sv YY = %bYY%
&sv HH = %bHH%

/* Set the maximum number of days in the current month.

&if %MM% eq 02 &then
 &do
  &if %YY% eq 96 &sv maxDD = 29
  &else &sv maxDD = 28
 &end
&if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11 &then
  &setvar maxDD = 30
&else
  &setvar maxDD = 31

/* Set the prefix and suffix for the rainfall files.

&setvar prefix = r.xmrg
&setvar suffix = z

/* SET THE END DATE OF THE STORM

&setvar enddate = %eMM%%eDD%%eYY%%eHH%
/* GENERATE THE FILENAMES OF THE PRECIP DATA.  WRITE THESE
/* NAMES TO AN ARRAY.
&setvar number = 1
&do &until %date% eq %enddate%
```

copymb.aml (page 3 of 3):

```
 &setvar date = %MM%%DD%%YY%%HH%
 &setvar flnm = %date%
 &if %prefix% ne none &then &sv flnm = %prefix%%flnm%
 &if %suffix% ne none &then &sv flnm = %flnm%%suffix%
 &setvar file%number% = %flnm%
 &setvar number = %number% + 1
 &if %HH% eq 23 &then
  &do
   &setvar HH = 00
   &setvar DD = %DD% + 1
   &if %DD% lt 10 &then &setvar DD = 0%DD%
   &if %DD% gt %maxDD% &then
     &do
      &setvar DD = 01
      &setvar MM = %MM% + 1
      &if %MM% lt 10 &then &setvar MM = 0%MM%
      &if %MM% gt 12 &then
        &do
          &setvar MM = 01
          &setvar YY = %YY% + 1
          &if %YY% gt 99 &then &setvar YY = 0
          &if %YY% lt 10 &then &setvar YY = 0%YY%
        &end
      &if %MM% eq 02 &then
       &do
        &if %YY% eq 96 &then &sv maxDD = 29
        &else &sv maxDD = 28
       &end
      &if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11 &then
        &setvar maxDD = 30
      &else &setvar maxDD = 31
    &end
   &end
 &else
   &do
    &setvar HH = %HH% + 1
    &if %HH% lt 10 &then &setvar HH = 0%HH%
   &end
&end
&setvar file%number% = done
/*  Return to the Main Progam
&return
```

d250fac.aml (page 1 of 1):

```
/* d250fac.aml

/* This aml executes the flowaccumulation command

&terminal 9999
&messages &popup

/* Enter the base name of the DEM

&sv dem = %.name%

/* Check to see if the flowdirection grid exists

&if [exists %dem%fdr -grid] eq .FALSE. &then
 &do
  &type Flowdirection grid does not Exist
  &return
 &end

/* Compute the flowaccumulation grid

&if [exists %dem%fac -grid] &then
 &do
  &sv check = [response 'File *fac already exists.  Replace? (y or n)']
  &if %check% eq n &then
   &return
  &else kill %dem%fdr
 &end

&messages &on

grid

%dem%fac = flowaccumulation (%dem%fdr)

quit

&return
```

d250fdr.aml (page 1 of 1):

```
/* d250fdr.aml

/* This aml executes the flowdirection command

&terminal 9999
&messages &popup

/* Enter the base name of the DEM

&sv dem = %.name%

/* Check to see if base dem exists

&if [exists %dem%fil -grid] eq .FALSE. &then
 &do
  &type File does not Exist
  &return
 &end

/* Compute the flowdirection grid

&if [exists %dem%fdr -grid] &then
 &do
  &sv check = [response 'File *fdr already exists.  Replace? (y or n)']
  &if %check% eq n &then
   &return
  &else kill %dem%fdr
 &end

&messages &on

grid

%dem%fdr = flowdirection (%dem%fil)

quit

&return
```

d250snk.aml (page 1 of 2):

```
/* d250snk.aml

/* This aml first determines the depths of the existing sinks
/*    in the area of interest then permits the user to enter
/*    the depth of sink to be filled.

&terminal 9999
&messages &popup

/* Enter base name of DEM

&sv dem = %.name%

/* Compute the flowdirection grid for the DEM

&messages &on

grid
  sinkfdr = flowdirection (%dem%prj)

/* Determine the locations of all sinks
  sinks = sink (sinkfdr)
/* Determine the contributing area for each sink
  sinkarea = watershed (sinkfdr, sinks)
/* Determine the minimum and maximum elevations for each sink
  sinkmin = zonalmin (sinkarea, %dem%prj)
  sinkmax = zonalfill (sinkarea, %dem%prj)
/* Determine the depth of each sink
  sinkdepth = sinkmax - sinkmin
/* Report the statistics for the sinkdepth grid to the user
  list sinkdepth.sta
/* Prompt the user for the depth of sink to fill, then execute
/*    the fill command.  Delete the intermediate files.

  &messages &popup
  &sv depth = [response 'Enter the depth of sink to fill']
  &messages &on

  fill %dem%prj %dem%fil sink %depth% #
  kill sinkfdr all
  kill sinks all
  kill sinkarea all
```

d250snk.aml (page 2 of 2):

```
  kill sinkmin all
  kill sinkmax all
  kill sinkdepth all
quit
&return
```

dataprep.menu (page 1 of 1):

```
7 dataprep.menu
/*
 Select the Desired Data Preparation Option:

   %button1

   %button2

   %button3

   %button4

   %button5

   %button7

   %button6
%button1 BUTTON KEEP 'Create a Point Coverage' &r pointinput.aml
%button2 BUTTON KEEP 'Process DLG' &menu dlg.menu &stripe 'DLG Menu'
%button3 BUTTON KEEP 'Process DEM' &menu dem.menu &stripe 'DEM Menu'
%button4 BUTTON KEEP 'Process SSURGO' &menu ssurgo.menu &stripe 'SSURGO
Menu'
%button5 BUTTON KEEP 'Process Landcover' &menu lc.menu &stripe 'Landcover
Menu'
%button7 BUTTON KEEP 'Arc Prompt' &TTY
%button6 BUTTON KEEP 'Cancel' &return
```

definews.aml (page 1 of 3):

```
/* definews.aml

/* This aml permits the user to define an arbitrary watershed for
/*   use with the rainfall analysis tools.  The program enters arcedit,
/*   prompts the user for background coverages, prompts the user for
/*   a coverage to use as a basis for the new coverage, and then permits
/*   the user to delineate the watershed.  The new coverage is stored as
/*   ws%.name%.  The aml also creates a ws%.name%clip grid.

&TERMINAL 9999
&MESSAGES &ON

/* Enter arcedit, turn the messages to popup

arcedit

display 9999

&m mape.menu &stripe 'Map Extent Menu'

/* Ask if a grid is to be viewed.  If so, use getgrid to return
/*   the grid name.  Set the image to on, and then define the image
/*   to be viewed.

&sv grid = [getchoice YES NO -prompt 'Do you wish to display a background~
 grid?']
&if %grid% eq YES &then
 &do
  &sv grid = [getgrid]
  image on
  image %grid%
 &end

/* Ask if a coverage is to be viewed.  If so, use getcov to return
/*  the name of the grid, and then prompt for a color.  Set the
/*  backenvironment.

&sv cover = [getchoice YES NO -prompt 'Do you wish to display a background~
 coverage?']

backe poly on
backe arc on
```

definews.aml (page 2 of 3):

```
backe nodes on
backe point on
backe tic on
&do &while %cover% eq YES
 &sv covname = [getcov]
 &messages &pop
 &sv color = [response 'Enter the color (1=white, 2=red, 3=green, 4=blue):']
 &messages &on
backcoverage %covname% %color%
 &sv cover = [getchoice YES NO -prompt 'Do you wish to display another~
 coverage?']
&end

drawenvironment poly on

/* Create a new grid called ws%.name%. First, ask the user which
/*  coverage the polygon will resemble.

&sv covtics = [getcov # # 'Select the Coverage whose Tic Cover will be~
 used for the new Polygon Coverage:']

  create ws%.name% %covtics%
  build
  editfeature poly

/* Inform the user to draw the polygon.

&messages &pop

&type After quitting this window, draw the watershed using the left
&type hand mouse button. When done, enter 2 from the keyboard.

&messages &on

  draw
  add one
  save
  quit

/* Convert the polygon to a grid

&do
```

definews.aml (page 3 of 3):

```
polygrid ws%.name% %.name%ws
 30
 y
&end

/*  enter grid, create a clip grid.

grid

%.name%clip = %.name%ws / %.name%ws
quit
&return
```

dem.menu (page 1 of 1):

```
7 dem.menu
/*

   Select the Desired DEM Option:

     %button1

     %button2

     %button3

     %button4

     %button6

     %button5
%button1 BUTTON KEEP 'Import 1:250k USGS DEM' &r dem250.aml
%button2 BUTTON KEEP 'Import 1:24k USGS DEM' &r dem24.aml
%button3 BUTTON KEEP 'Create Custom DEM using TOPOGRID' ~
  &messages &popup;   &s og = %.name%prj; &s cs = [response 'Enter Cell Size ~
   (in meters):'];   topogridtool %og% %cs%
%button4 BUTTON KEEP 'Process DEM' &m demprocess.menu &stripe 'Process DEM'
%button6 BUTTON KEEP 'Arc Prompt' &tty
%button5 BUTTON KEEP 'Cancel' &return
```

dem24.aml (page 1 of 3):

```
/* dem24.aml


/* This aml processes 1:24k USGS DEMs from DASC

&TERM 9999
&MESSAGES &POPUP

/* Enter the number of Grids to be merged

&setvar numgrids = [response 'Enter the number of Grids']

&setvar n = 1

/* Enter each file name, save these names in an array.

&do &while %n% le %numgrids%
 &setvar grid%n% [response 'Enter the base name of the file (no extension)']
 &sv n = %n% + 1
&end

&sv gridname = %.name%

&setvar n = 1

/* Get rid of the POPUP menu option.

&messages &on

/* For each filename, convert the file into a lattice.

&r msworking init 'Converting the DEMs to Grids.' 'Waiting Time = LONG'~
 '&cc' 'dem24.aml
&do &while %n% le %numgrids%
 demlattice [value grid%n%].dem [value grid%n%] usgs
 &sv n = %n% + 1
&end
&r msworking close
/* Merge the individual lattices, display the merged grid, and
/*    set the analysis window.

&sv n = 2
```

dem24.aml (page 2 of 3):

```
&sv string = %grid1%

&do &while %n% le %numgrids%
 &sv string = %string%,[value grid%n%]
 &sv n = %n% + 1
&end

grid
 &r msworking init 'Merging the Grid Tiles' 'Waiting Time = LONG'~
  '&cc' 'dem24.aml'
 %gridname% = merge(%string%)
 &r msworking close
 display 9999
 mape %gridname%
 gridp %gridname% # linear nowrap gray

/*  Ask user if a gage coverage should be displayed.  If so,
/*    let the user pick the coverage and display it.

 &sv ask1 = [getchoice YES NO -prompt~
  'Do you wish to display an existing gage coverage?']
 &if %ask1% eq YES &then
  &do
   markercolor 3
   &sv ptgage = [getcover * -point 'Select the Gage Coverage']
   points %ptgage%
  &end
/*  Ask user if a line coverage should be displayed.  If so,
/*    let the user pick the coverage and display it.
 &sv ask2 = [getchoice YES NO -prompt~
  'Do you wish to display an existing line coverage?']
 &if %ask2% eq YES &then
  &do
   &sv line = [getcover * -line 'Select a Line Coverage']
   arclines %line% 4
  &end
 &messages &pop
 &type 'Set the analysis window on the area of interest.'
 &messages &on
 setwindow *
 &r msworking init 'Projecting the DEM' 'Waiting Time = LONG' '&cc' 'dem24.aml'
 %gridname%dem = %gridname%
```

dem24.aml (page 3 of 3):

```
%gridname%prj = project (%gridname%dem)
 output
 projection albers
 datum nad27
 units meters
 zunits meters
 parameters
  37 30 00
  39 30 00
  -98 00 30
  37 00 00
  0.0
  0.0
end

&r msworking close
quit

kill %gridname%dem
kill %gridname%

&r msinform init 'The Following Grid was Created:' %gridname%prj~
 '&cc' 'dem24.aml'

&return
```

dem250.aml (page 1 of 2):

```
/* dem250.aml


/* This aml processes 1:250k USGS DEMs from DASC

&TERM 9999
&MESSAGES &POPUP

/* Enter the number of Grids to be merged

&setvar numgrids = [response 'Enter the number of Grids']

&setvar n = 1

/* Enter each file name, save these names in an array.

&do &while %n% le %numgrids%
 &setvar grid%n% [response 'Enter the name of the file (no extension)']
/*[getfile *.zip -FILE -NOEXT 'Select a File']
 &sv n = %n% + 1
&end

&sv gridname = %.name%

&setvar n = 1

/* Get rid of the POPUP menu option.

&messages &on

/* For each filename, unzip and convert the file into a lattice.

&do &while %n% le %numgrids%
 unzip [value grid%n%]
 demlattice [value grid%n%].dem [value grid%n%] usgs
 &sv n = %n% + 1
&end

/* Merge the individual lattices, display the merged grid, and
/*    set the analysis window.
&sv n = 2
&sv string = %grid1%
```

dem250.aml (page 2 of 2):

```
&do &while %n% le %numgrids%
 &sv string = %string%,[value grid%n%]
 &sv n = %n% + 1
&end
grid
 %gridname% = merge(%string%)
 display 9999
 mape %gridname%
 gridp %gridname% # linear nowrap gray
&messages &popup
 &type 'Set the analysis window on the area of interest.'
&messages &on
 setwindow *
 %gridname%dem = %gridname%
 %gridname%prj = project (%gridname%dem,~
  /files/arcfiles/projection/geoalb.prj,#,100)
quit

&return
```

demprocess.menu (page 1 of 1):

```
7 demprocess.menu
/*

  Select the Desired DEM Operation:

     %button1

     %button2

     %button3

     %button4

     %button5

     %button6

     %button7

     %button9

     %button8
%button1 BUTTON KEEP 'Clip DEM to Reduce Size' &r clipdem.aml
%button2 BUTTON KEEP 'Process Sinks' &r d250snk.aml
%button3 BUTTON KEEP 'Compute Flowdirection' &r d250fdr.aml
%button4 BUTTON KEEP 'Compute Flowaccumulation' &r d250fac.aml
%button5 BUTTON KEEP 'Delineate Watersheds' &r wshed.aml
%button6 BUTTON KEEP 'Clip Grids to Watershed' &r clipgrid.aml
%button7 BUTTON KEEP 'Calculate Slopes' &r calcslope.aml
%button9 BUTTON KEEP ~
  HELP 'Type &return to return' ~
  'Arc Prompt' &tty
%button8 BUTTON KEEP 'Cancel' &return
```

dlg.menu (page 1 of 1):

```
7 dlg.menu
/*

 Select the Desired DLG Preparation Option:

   %button1

   %button2

   %button3

   %button4

   %button6

   %button5
%button1 BUTTON KEEP 'Import Hypsography DLG' &r dlg100hp.aml
%button2 BUTTON KEEP 'Import Other DLG Type' &r dlg100other.aml
%button3 BUTTON KEEP 'Clip DLGs to Reduce Size' ~
 &menu dlgclip.menu &stripe 'Clip DLGs to Reduce Size'
%button4 BUTTON KEEP 'Preprocess Streams for TOPOGRID' &r preproc.aml
%button6 BUTTON KEEP ~
 HELP 'Type &return to return' ~
 'Arc Prompt' &tty
%button5 BUTTON KEEP 'Cancel' &return
```

dlg100hp.aml (page 1 of 4):

```
/* dlg100hp.aml


/*
/* This aml will append and project all hypsography dlg's specified.
/*

&term 9999
&messages &popup


/* Enter the number of files to be appended. Enter the type of file,
/*   and enter a three character identifier for the final coverage.


&setvar type = hp
&setvar ext = %.name%

/*-----------------------------------------------------------------
/* Enter the file names. They are written to a text file, which
/* will later be displayed as a thread for confirmation.
/* The loop checks that each dlg file chosen contains a hypsography
/* coverage, and if it doesn't, asks the user to choose again.
/*-----------------------------------------------------------------
&if [exists dlgfiles.menu -file] &then rm dlgfiles.menu
&sv fileunit = [open dlgfiles.menu openstat -append]

&sv record = '7 dlgfiles.menu'
&setvar writestat = [write %fileunit% %record%]

&sv writestat = [write %fileunit% '/*']

&sv num = 0
&sv n = 1

&label more
&sv done = no
&sv another = not
&do &until %done% = done
  &label nothp
  &menu dlg100hp.menu &stripe 'Choose hypsography file name '%n%':'
  &if %done% = cancel &then
```

dlg100hp.aml (page 2 of 4):

```
 &do
        &sv close = [close %fileunit%]
        &return
  &end
  &sv e = [keyword hp [after [value file] '_']]
  &if %e% eq 0 &then &type This dlg file does not contain a hypsography coverage.
  &if %e% eq 0 &then &goto nothp
&sv name%n% = %file%
  &sv writestat = [write %fileunit% %n%'...'%file%]
  &sv num = %num% + 1
  &sv n = %n% + 1
&end
&sv closestat = [close %fileunit%]
&messages &off
&messages &popup
&sv stripe = 'Files for coverage '%type%':'
&thread &create display &menu dlgfiles.menu &stripe %stripe%
&sv another = [getchoice 'Add file' Continue Cancel -prompt 'Do you need to add another
dlg file?']

&if %another% = Cancel &then &return
&if %another% = 'Add file' &then
  &do
        &thread &delete display
        &sv fileunit = [open dlgfiles.menu openstat -append]
  &end
&if %another% = 'Add file' &then &goto more
&messages &off
&messages &on
&thread &delete display
&messages &off
&messages &popup


&r msworking init 'Processing Hypsography Files' 'Waiting Time = LONG'~
 '&cc' 'dlg100hp.aml'

&messages &on

/*---------------------------------------------------------------------
/*   Unzip and convert the dlg's to an arc file (using the optional format),
/*   build topology for the vector file, create a new info file from the
```

dlg100hp.aml (page 3 of 4):

```
/*   *.acode info file consisting of just the relate item and the minor2
/*   attribute, which is the elevation of the contour. Join this new info
/*   file to the *.aat file, and then remove the border from the vector
/*   coverage using the reselect command.
/*----------------------------------------------------------------------

&do n = 1 &to %num% &by 1
 unzip [value name%n%]
 &sv newname%n% = [after [after [after [after [before [value name%n%] .] /] /] /] /]
 dlgarc optional [value newname%n%].dlg [value newname%n%]
 build [value newname%n%] line
  pullitems [value newname%n%].acode [value newname%n%].cont
   [value newname%n%]-id
   minor2
   end
 joinitem [value newname%n%].aat [value newname%n%].cont [value
newname%n%].aat~
   [value newname%n%]-id [value newname%n%]-id
 reselect [value newname%n%] r[value newname%n%] line # line
  res rpoly# > 1
  ~
  N
  Y
  res lpoly# > 1
  ~
  N
  N
&end

/*   Append the existing coverages.

append %ext%%type%utm line
  &do n := 1 &to %num% &by 1
    r[value newname%n%]
  &end
end

/*   Project the appended coverages.

project cover %ext%%type%utm %ext%%type%
 output
  projection albers
```

dlg100hp.aml (page 4 of 4):

```
 units meters
 parameters
  37 30 00
  39 30 00
  -98 00 30
  37 00 00
  0.0
  0.0
end

/*  Build topology for the final coverage.

build %ext%%type% line

/*  Delete the intermediate coverages

kill %ext%%type%utm all
&do n = 1 &to %num% &by 1
 rm [value newname%n%].dlg
 kill [value newname%n%] all
 kill r[value newname%n%] all
&end
tables
 sel %ext%%type%.aat
 alter
  minor2
  elev
  ~
  ~

  ~
  ~
quit

&r msworking close

&r msinform init 'The Following Coverage was Created:' %.name%%type%~
 '&cc' 'dlg100hp.aml'

&return
```

dlg100hp.menu (page 1 of 1):

```
7 dlg100hp.menu
/*
  %datalist1
                %button4
 %b3    %b1    %button2
%datalist1 INPUT FILE 19 TYPEIN NO SCROLL YES ROWS 12 FILE ~
  *.zip -FILE -NOEXTENSION -SORT -other
%button4 BUTTON KEEP ~
  HELP 'Type &return to return' ~
  'Arc Prompt' &tty
%b3 BUTTON KEEP 'Choose' &return
%b1 BUTTON KEEP 'Cancel' ~
  &sv done = cancel;&sv closestat = [close %fileunit%];&return
%button2 BUTTON KEEP 'Choose and Done' &sv done = done;&return
```

dlg100other.aml (page 1 of 4):

```
/* dlg100other.aml


/*---------------------------------------------------------------
/* This aml will append and project all dlg's specified.
/*---------------------------------------------------------------

&term 9999
&messages &popup

/*---------------------------------------------------------------
/* Enter the number of files to be appended. Enter the type of file,
/*    and enter a three character identifier for the final coverage.
/*---------------------------------------------------------------

&setvar type = [response 'Enter the type of file (ie rd, bd, etc.)']
&setvar ext = %.name%

/*---------------------------------------------------------------
/* Enter the file names. They are written to a text file, which
/* will later be displayed as a thread for confirmation.
/*---------------------------------------------------------------
&if [exists dlgfiles.menu -file] &then rm dlgfiles.menu
&sv fileunit = [open dlgfiles.menu openstat -write]

&sv record = '7 dlgfiles.menu'
&setvar writestat = [write %fileunit% %record%]

&sv writestat = [write %fileunit% '/*']


&sv num = 0
&sv n = 1

&label more
&sv done = no
&do &until %done% = done
  &menu dlg100other.menu &stripe 'Choose file name '%n%':'
  &if %done% = cancel &then
  &do
        &sv closestat = [close %fileunit%]
        &return
```

dlg100other.aml (page 2 of 4):

```
  &end
  &sv name%n% = %file%
  &sv writestat = [write %fileunit% %n%'...'%file%]
  &sv num = %num% + 1
  &sv n = %n% + 1
&end

&sv closestat = [close %fileunit%]
&messages &off
&messages &popup
&sv stripe = 'Files for coverage '%type%':'
&thread &create display &menu dlgfiles.menu &stripe %stripe%
&sv another = [getchoice 'Add file' Continue Cancel -prompt 'Do you need to add another
dlg file?']
&if %another% = Cancel &then &return
&if %another% = Add &then
  &do
        &thread &delete display
        &sv fileunit = [open dlgfiles.menu openstat -append]
  &end
&if %another% = 'Add file' &then &goto more
&thread &delete display

&messages &on
&r msworking init 'Processing DLG' 'Waiting Time = LONG' '&cc' 'dlg100other.aml'

/*------------------------------------------------------------------------
/*   Unzip and convert the dlg's to an arc file (using the optional format),
/*   build topology for the vector file, create a new info file from the
/*   *.acode info file consisting of just the relate item and the minor2
/*   attribute, which is the elevation of the contour.  Join this new info
/*   file to the *.aat file, and then remove the border from the vector
/*   coverage using the reselect command.
/*------------------------------------------------------------------------


&do n = 1 &to %num% &by 1
 unzip [value name%n%]
 &sv newname%n% = [after [after [after [after [before [value name%n%] .] /] /] /] /]
 dlgarc optional [value newname%n%].dlg [value newname%n%]
 build [value newname%n%] line
```

dlg100other.aml (page 3 of 4):

```
/*----------------------------------------------------------------------
/* The commented out lines apply only to the hypsography layer, but could
/*   be adapted to extract attribute information for the hypsography,
/*   roads, or other layers.
/*----------------------------------------------------------------------

/* pullitems [value newname%n%].acode [value newname%n%].cont
/*   [value newname%n%]-id
/*   minor2
/*   end
/* joinitem [value newname%n%].aat [value newname%n%].cont [value
newname%n%].aat~
/*   [value newname%n%]-id [value newname%n%]-id
 reselect [value newname%n%] r[value newname%n%] line # line
  res rpoly# > 1

  ~
  N
  Y
  res lpoly# > 1

  ~
  N
  N
&end

/*   Append the existing coverages.

append %ext%%type%utm line
  &do n := 1 &to %num% &by 1
    r[value newname%n%]
  &end
end

/*   Project the appended coverages.

project cover %ext%%type%utm %ext%%type%
 output
  projection albers
  units meters
  parameters
   37 30 00
```

dlg100other.aml (page 4 of 4):

```
  39 30 00
 -98 00 30
  37 00 00
  0.0
  0.0
end

/*  Build topology for the final coverage.

build %ext%%type% line

/*  Delete the intermediate coverages

kill %ext%%type%utm all
&do n = 1 &to %num% &by 1
 rm [value newname%n%].dlg
 kill [value newname%n%] all
 kill r[value newname%n%] all
&end
&r msworking close
&r msinform init 'The Following Coverage was Created'~
 %.name%%type% '&cc' 'dlg100other.aml'

&return
```

dlg100other.menu (page 1 of 1):

```
7 dlg100other.menu
/*
  %datalist1
                 %button4
 %b3    %b1    %button2
%datalist1 INPUT FILE 19 TYPEIN NO SCROLL YES ROWS 12 FILE ~
 *.zip -FILE -NOEXTENSION -SORT -other
%button4 BUTTON KEEP ~
 HELP 'Type &return to return' ~
 'Arc Prompt' &tty
%b3 BUTTON KEEP 'Choose' &return
%b1 BUTTON KEEP 'Cancel' ~
 &sv done = cancel;&sv closestat = [close %fileunit%];&return
%button2 BUTTON KEEP 'Choose and Done' &sv done = done;&return
```

dlgclip.aml (page 1 of 2):

```
/* dlgclip.aml

/*
/* This aml will clip all dlg's specified to a user defined
/*   boundary.
/*

&term 9999
&messages &on

/* Enter arcedit, turn the messages to popup

arcedit
&messages &popup

/* Enter the number of dlg's to be clipped

&setvar num = [response 'Enter the number of dlg files to be clipped:']
&messages &on

/* Select the dlg coverages to be clipped.  Store the names in an
/*   array.  Make the dlg's into backcoverages.

&do n = 1 &to %num% &by 1
 &menu clipcov.menu
 &sv cov%n% = %cov%
 backcoverage [value cov%n%] %n%
&end

/* Set the map extent and the backenvironment.

display 9999
mape [value cov1]
backenvironment poly on
backenvironment arc on
backenvironment tic on
draw

/* Ask the user if additional backcoverages are necessary.  If so,
/*   add them.

&sv okay = [getchoice YES NO -prompt~
```

dlgclip.aml (page 2 of 2):

```
'Do you wish to add another back coverage?']

&sv i = %num% + 1

&do &while %okay% eq YES
 backcoverage [getcover] %i%
 &sv i = %i% + 1
 &sv okay = [getchoice YES NO -prompt~
  'Do you wish to add another back coverage?']
&end

/* Draw the backcoverages, set the drawenvironment, create the new
/*   polygon coverage, build topology for that coverage, set the
/*   editfeature to polygon, then add the polygon.

draw
drawe poly on
create %.name%dlgclip [value cov1]
build
editf polygon

&messages &popup
&type After quitting this message box, use the mouse to define the
&type polygon that encompasses the watershed. Be sure to include
&type enough area to permit accurate interpolation of the watershed
&type elevations by topogrid. Enter 2 when done.
&messages &on

add poly one

/* Save and quit.

save
quit

/* Clip the coverages to the boundary just specified.

&do n = 1 &to %num% &by 1
 clip [value cov%n%] %.name%dlgclip [value cov%n%]clip line
&end
&r msinform init 'The Specified Coverages were Clipped.'
&return
```

dlgclip.menu (page 1 of 1):

```
7 dlgclip.menu
/*

   %button1

   %button2

   %button3

   %button4
%button1 BUTTON KEEP 'Draw polygon to clip DLG' &r dlgclip.aml
%button2 BUTTON KEEP 'Clip DLGs to existing coverage' &r dlgclipcov.aml
%button3 BUTTON KEEP ~
  HELP 'Type &return to return' ~
  'Arc Prompt' &tty
%button4 BUTTON KEEP 'Cancel' &return
```

dlgclipcov.aml (page 1 of 1):

```
/* dlgclipcov.aml



/*
/* This aml will clip all dlg's specified to a user defined
/*   boundary that exists in a coverage (ie a watershed boundary).
/*

&term 9999
&messages &on

/* Enter arcedit, turn the messages to popup

&messages &popup

/* Enter the number of dlg's to be clipped

&setvar num = [response 'Enter the number of dlg files to be clipped:']

&messages &on

/* Select the dlg coverages to be clipped.  Store the names in an
/*  array.  Make the dlg's into backcoverages.

&do n = 1 &to %num% &by 1
 &menu clipcov.menu
 &sv cov%n% = %cov%
backcoverage [value cov%n%] %n%
&end

&menu boundclipcov.menu
&sv dlgclip = %cov%
/* Clip the coverages to the boundary in the coverage just specified.

&do n = 1 &to %num% &by 1
clip [value cov%n%] %dlgclip% [value cov%n%]clip line
&end

&r msinform init 'The Specified Coverages were Clipped.'

&return
```

genhrap.aml (page 1 of 2):

```
/* genhrap.aml

/* This program generates an HRAP polygon coverage using the file
/*   created by genhrap.for.  genhrap.aml was developed by
/*   Seann M. Reed of the University of Texas at Austin

&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% albnat.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
```

genhrap.aml (page 2 of 2):

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

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

genstats.aml (page 1 of 1):

```
/* genstats.aml

/* This aml generates hourly statistics on grids of NEXRAD
/*   rainfall estimates.

&TERMINAL 9999
&MESSAGES &popup

/* Enter the number of hours in the storm.
&sv num = [response 'Enter the number of hours/days in the storm:']
&messages &on
/*-------------------------------------------------------------
/* For each hourly output file, check to see if the grid exists.
/*   if the grid exists, then clip the grid
/*   to the watershed boundary to permit calculation of a spatial
/*   average (using the statistics command). The spatial average
/*   for each hour of data is compiled in a text file called
/*   %.storm%stats. Kill each hourly statistic file as well as
/*   the clipped grid, but retain the larger grids.
/*-------------------------------------------------------------

&do n = 1 &to %num%
 &if [exists %.storm%%.ext%%n% -grid] ne .TRUE. &then
   &return &inform The grid of precip does not exist.
 grid
  ws%.ext%%n% = %.name%clip * %.storm%%.ext%%n%
 quit
 statistics ws%.ext%%n%.vat ws%.ext%%n%.stats
  mean value count
  end
 tables
  select ws%.ext%%n%.stats
  unload %.storm%stats MEAN-W-VALUE
  kill ws%.ext%%n%.stats
 quit
 kill ws%.ext%%n%
&end

/* Make sure that all aml files are properly closed, then
/*   return.
&setvar closestat = [close -all]
&return
```

geoalb.prj (page 1 of 1):

```
input
projection geographic
units ds
datum wgs72
parameters
output
projection albers
units meters
datum wgs72
parameters
37 30 00
39 30 00
-98 00 30
37 00 00
0.0
0.0
end
```

gv.menu (page 1 of 1):

```
7 gv.menu
/*

 Select the GRID you wish to view:

 %datalist1
 %button1
%datalist1 INPUT grid 22 TYPEIN NO SCROLL YES ROWS 4 ~
  KEEP ~
  RETURN 'gridp %grid% # linear # gray; &return' ~
  GRID ~
  * -SORT
%button1 BUTTON KEEP 'Cancel' &return
```

hrapview.aml (page 1 of 2):

```
/* hrapview.aml

/* This aml permits the user to view the hrap coordinates for
/*    each cell in the watershed
/*

&term 9999
&messages &on

/* Enter arcedit, turn the messages to popup

arcedit

display 9999

&m mape.menu &stripe 'Map Extent Menu'

/* Ask if a grid is to be viewed.  If so, use getgrid to return
/*   the grid name.  Set the image to on, and then define the image
/*   to be viewed.


&sv grid = [getchoice YES NO -prompt 'Do you wish to display a background~
 grid?']
&if %grid% eq YES &then
 &do
  &sv grid = [getgrid]
  image on
  image %grid%
 &end
/* Ask if a coverage is to be viewed.  If so, use getcov to return
/*  the name of the grid, and then prompt for a color.  Set the
/*  backenvironment.
&sv cover = [getchoice YES NO -prompt 'Do you wish to display a background~
 coverage?']
backe poly on
backe arc on
backe tic on
backe point on
&do &while %cover% eq YES
 &sv covname = [getcov]
 &messages &pop
```

hrapview.aml (page 2 of 2):

```
&sv color = [response 'Enter the color (1=white, 2=red, 3=green, 4=blue):']
 &messages &on
 backcoverage %covname% %color%
 &sv cover = [getchoice YES NO -prompt 'Do you wish to display another~

 coverage?']
&end

/* Set the draw environment, and select the kangeoccalb coverage
/*   as the edit coverage.  Set the editfeature to poly

drawe poly on
editc /files/arcfiles/kan_hrap/kangeoccalb
editf poly

/* Draw the coverages, then prompt for the user to select a cell
/*  for which the HRAP coordinates are desired.

draw

&type After quitting this message, select the cell for which you desire
&type the HRAP coordinates.

/* Until the user says no, continue to select, list, and unselect cells.

&sv again = YES

&fullscreen &popup /*  makes the list appear in a popup menu

&do &while %again% eq YES
 select one
 list
 unselect all
 &sv again = [getchoice YES NO -prompt~
 'Do you need the HRAP Coordinates for another Cell?']
&end

&fullscreen &off

quit

&return
```

killstorm.aml (page 1 of 1):

```
/* killstorm.aml
&terminal 9999
&messages &popup
rm r.*
rm out.r.*
&sv storm = [response 'Enter name of storm']
&sv nhrs = [response 'Number of hours in storm']
&messages &on
&do n = 1 &to %nhrs% &by 1
kill %storm%hr%n% all
&end
kill %storm%hrap
kill %storm%area
&return
```

lc.aml (page 1 of 3):

```
/* lc.aml

/* This aml processes landcover coverages from DASC

&TERM 9999
&MESSAGES &POPUP

/* Enter the number of Counties intersecting the watershed.

&setvar counties = [response 'Enter the number of Counties Intersected']

&setvar n = 1

/* Enter each file name, save these names in an array.

&do &while %n% le %counties%
 &setvar county%n% [response 'Enter the two-letter county abbreviation']
 &sv n = %n% + 1
&end

/* Give a base name for the merged coverage

&sv cname = %.name%

/* Check if the watershed boundary exists for the coverage.

&if [exists ws%cname% -cover] eq .FALSE. &then
  &do
    &type The watershed boundary does not exist
    &return
  &end

/* Get rid of the POPUP menu option.
&messages &on

&setvar n = 1
/* For each filename: check if it exists, then unzip, change the permissions
/*   to remove the execute permission, and import the *.e00 file.
&do &while %n% le %counties%
 &if [exists [value county%n%]_landc.zip -file] eq .FALSE. &then
  &do
```

lc.aml (page 2 of 3):

```
   &messages &popup
   &type Zip file [value county%n%]_landc.zip does not exist
   &return
  &end
 unzip [value county%n%]_landc.zip
 chmod a-x [value county%n%]_landc.*
 import cover [value county%n%]_landc.e00 [value county%n%]lc
 &sv n = %n% + 1
&end

/* Project the existing polygon watershed boundary into the
/*   projection of the landcover files (geographic coordinates).
/*   Build topology for the projected coverage.

project cover ws%cname% wsclip /files/arcfiles/projection/wstolc.prj
build wsclip poly

/* If there is only one landcover coverage, then clip it to the
/*   watershed boundary, project it, and build topology.

&if %counties% eq 1 &then
 &do
  clip %county1%lc wsclip clipcov poly
  project cover clipcov ws%cname%lc /files/arcfiles/projection/flowgage.prj
  build ws%cname%lc poly
  kill clipcov
  kill wsclip
 &end
&else
 &do

/* If there are more than one counties:
/* Mapjoin the landcover coverages, clipping the final coverage to
/*   the watershed boundary.

  mapjoin joincov poly # wsclip
   &do n := 1 &to %counties% &by 1
    [value county%n%]lc
   &end
  end
/* Use the dissolve command to remove the border between the
/*  counties.
```

lc.aml (page 3 of 3):

```
 dissolve joincov discov grid-code poly
/* Project the dissolved coverage into the desired projection,
/*  build topology, and then kill all of the intermediary coverages.

 project cover discov ws%cname%lc /files/arcfiles/projection/flowgage.prj
 build ws%cname%lc poly
 kill discov
 kill joincov
 kill wsclip

&end

&do n := 1 &to %counties% &by 1
 kill [value county%n%]lc
&end

&return
```

lv.menu (page 1 of 1):

```
7 lv.menu
/*
 Select the Line Coverage you wish to view:

 %datalist1                 Line Color:

                        %symbollist1




     %button1          %button2
%datalist1 INPUT line 22 TYPEIN NO SCROLL YES ROWS 6 COVER ~
  * -LINE -SORT
%symbollist1 INPUT lc 14 TYPEIN NO SCROLL YES ROWS 4 SYMBOL ~
  -LINE
%button1 BUTTON KEEP 'Apply' arclines %line% %lc%; &return
%button2 BUTTON KEEP 'Cancel' &return
```

main.aml (page 1 of 1):

```
7 main.menu
/*
/*     %button2
/*
/*     %button3
/*
/* %button2 BUTTON KEEP 'Loss Model' &messages &pop; &sv .storm [response~
/* 'Enter a name for the storm:']; &menu lossmod.menu &stripe 'Loss Model'
/* %button3 BUTTON KEEP 'Flow Model' &m flowmod &stripe 'Flow Model'
  Select the Desired Model Option:

    %button1

    %button4

    %button5

    %button5_1

    %button6
%button1 BUTTON KEEP 'Data Preparation' ~
 &menu dataprep.menu &stripe 'Data Preparation'
%button4 BUTTON KEEP 'Map and Analyze Rainfall Data' ~
 &messages &pop;   &sv .storm [response 'Enter a name for the storm (max 5   ~
  characters):']; &m precip.menu   &stripe 'Rainfall Analysis for    ~
 '%.storm%"
%button5 BUTTON KEEP 'View Coverages' &r view.aml
%button5_1 BUTTON KEEP ~
 QUERY '&type Type &return to return' ~
 'Arc Prompt' &tty
%button6 BUTTON KEEP 'Cancel' &return
```

main.menu (page 1 of 1):

```
7 main.menu
/*
 Select the Desired Model Option:

   %button1

/*    %button2
/*
/*    %button3
/*
   %button4

   %button5

   %button6

%button1 BUTTON KEEP 'Data Preparation' ~
 &menu dataprep.menu &stripe 'Data Preparation'
/*%button2 BUTTON KEEP 'Loss Model' &messages &pop; &sv .storm [response~
/* 'Enter a name for the storm:']; &menu lossmod.menu &stripe 'Loss Model'
/*%button3 BUTTON KEEP 'Flow Model' &m flowmod &stripe 'Flow Model'
%button4 BUTTON KEEP 'Map and Analyze Rainfall Data' &messages &pop;~
 &sv .storm [response 'Enter a name for the storm:']; &m precip.menu~
 &stripe 'Rainfall Analysis'
%button5 BUTTON KEEP 'View Coverages' &r view.aml
%button6 BUTTON KEEP 'Cancel' &return
```

monthly2.aml (page 1 of 3):

```
/* monthly2.aml

/* This program handles the monthly processing of mbrfc stage III files.
/*  The aml tars and uncompresses the files sent to us by the mbrfc,
/*  and creates a reduced file for the HRAP cells covering Kansas.
/*  The hourly Stage III files are compressed again, but the reduced
/*  files are saved to /files/mbrfc/archives/reduced in an uncompressed
/*  form so that they are easy to access and read.


&TERMINAL 9999
&MESSAGES &ON

&setvar prefix = mb
&setvar suffix = .z

&setvar MM := [response 'Enter the month of the data (MM)']
&setvar YY := [response 'Enter the year (YY)']


/* SET THE DATE AND TIME VALUES EQUAL TO THE BEGINNING OF THE
STORM

&setvar bDD = [response 'Enter the first day of data to be processed (DD)']
&s maxDD = [response 'Enter the last day of data to be processed (DD)']
&setvar DD = %bDD%

/* SET THE END DATE OF THE STORM

&setvar enddate = %MM%%maxDD%

/* GENERATE THE FILENAMES OF THE PRECIP DATA.  WRITE THESE
/* NAMES TO THE OUTPUT FILE, AND STORE THEM IN AN ARRAY.
&do &until %date% eq %enddate%
  &setvar date = %MM%%DD%
  tar -xvf %prefix%%date%.tar
  rm %prefix%%date%.tar
  &setvar DD = %DD% + 1
  &if %DD% lt 10 &then &setvar DD = 0%DD%
&end
/* OPEN A FILE TO STORE THE FILE NAMES
```

monthly2.aml (page 2 of 3):

```
&setvar fileunit := [open reduce.input openstat -write]
&if %openstat% ne 0 &then
   &return &inform Could not open file.
/* SET THE PREFIXES AND SUFFIXES AFFIXED TO THE PRECIP DATA
&setvar prefix = xmrg
&setvar suffix = z

/* READ IN THE DATE AND TIME OF THE BEGINNING AND END OF THE
PERIOD
/* OF INTEREST

&s DD = %bDD%
&setvar HH := 00
&setvar eDD := %maxDD%
&setvar eHH := 23

/* SET THE END DATE OF THE STORM

&setvar enddate = %MM%%eDD%%YY%%eHH%
/* GENERATE THE FILENAMES OF THE PRECIP DATA. WRITE THESE
/* NAMES TO THE OUTPUT FILE, AND STORE THEM IN AN ARRAY.
&setvar number = 1
&do &until %date% eq %enddate%
  &setvar date = %MM%%DD%%YY%%HH%
  &setvar flnm = %date%
  &sv flnm = %prefix%%flnm%
  &sv flnm = %flnm%%suffix%
  uncompress %flnm%.Z
  &if [exists %flnm%] &then
   &do
    &setvar file%number% = %flnm%
    &setvar flnm = [quote ']%flnm%
    &setvar flnm = %flnm%[quote ']
    &setvar writestat = [write %fileunit% %flnm%]
    &setvar number = %number% + 1
   &end
  &if %HH% eq 23 &then
   &do
    &setvar HH = 00
    &setvar DD = %DD% + 1
    &if %DD% lt 10 &then &setvar DD = 0%DD%
   &end
```

monthly2.aml (page 3 of 3):

```
  &else
    &do
      &setvar HH = %HH% + 1
      &if %HH% lt 10 &then &setvar HH = 0%HH%
    &end
&end
&setvar writestat = [write %fileunit% done]
&setvar file%number% = done

/* CLOSE THE OUPUT FILE AND CALL THE FORTRAN PROGRAM
/* TO READ THE PRECIP DATA.  THE FORTRAN PROGRAM WILL
/* REDUCE THE FILES, LIMITING DATA TO KANSAS, AND OUTPUT
/* THE REDUCED FILE TO A SEPARATE FOLDER.

&setvar closestat = [close %fileunit%]

reducembrfc

/* COMPRESS THE ORIGINAL HOURLY FILES.

&setvar number = 1

&do &while [value file%number%] ne done
  compress [value file%number%]
  mv r.[value file%number%] /files/mbrfc/archives/reduced/r.[value file%number%]
  &sv number = %number% + 1
&end

rm reduce.input

&return
```

onecell.aml (page 1 of 5):

```
/* onecell.aml

/* This aml extracts NEXRAD precipitation estimates one cell.

&TERMINAL 9999
&MESSAGES &popup

/* Check if the file precip exists.  If so, ask the user to rename it.

&if [exists precip -file] &then
 &do
 &sv newname [response 'The existing precip file must be renamed.~
  Enter a new name:']
 mv precip %newname%
 &end

/* Select the river basin, enter the start and end times of the storm,
/*   and enter the HRAP coordinates.

&sv storm = %.storm%
&m selbasin.menu &stripe 'Select Basin'
&sv start = [response 'Enter the Start Date and Time (MM DD YY HH):']
&sv end = [response 'Enter the End Date and Time (MM DD YY HH):']
&sv coords [response 'Enter the HRAP Coordinates (HRAPX,HRAPY (no space)):']

/* Write the HRAP coordinates to a file, then write 0,0,0, which
/*   signals the fortran program that the end of the file
/*   has been reached.

&sv coords = [quote 1,[unquote %coords%]]
&s file1 = [open hrapinput openstat -write]
&s writestat = [write %file1% %coords%]
&s writestat = [write %file1% '0,0,0']
&s closestat = [close %file1%]

&messages &on

/* Call the precipfiles subroutine, which will create the
/*  'input' file with the names of the precip files to be
/*  opened by the fortran program.
&call precipfiles
```

onecell.aml (page 2 of 5):

```
/* CALL THE FORTRAN PROGRAM TO READ THE PRECIP DATA.
/* THE FORTRAN PROGRAM WILL READ THE INPUT AND THE HRAPINPUT
/* DATA FILES, OPEN THE APPROPRIATE RAINFALL FILES, AND
/* OUTPUT THE ASCII PRECIP DATA (ROWS = TIME STEPS, COLUMNS
/* = WS%HRAP%ID'S).

&if %.basin% eq abrfc &then
 &do
  cp /files/arcfiles/interface/prepab prepab
  prepab
 &end
&if %.basin% eq mbrfc &then
 &do
  cp /files/arcfiles/interface/prepmb prepmb
  prepmb
 &end

/* Delete the intermediary files.

rm input
rm hrapinput
rm prepab
rm prepmb

&r msinform init 'A File Named Precip has been Generated.'~
 'Rename this File Before Executing the Program Again.'

&return

/*--------------------------------------------------------------
/*--------------------------------------------------------------

/* The following routine creates a file with all of the file names
/*  to be opened.  A similar list is stored in the array file%number%

&routine Precipfiles
/* Set the beginning and end month, day, year, and hour of the storm.
/*   Use the extract function to accomplish this.
&sv st = [unquote %start%]
&sv bMM = [extract 1 %st%]
&sv bDD = [extract 2 %st%]
&sv bYY = [extract 3 %st%]
```

onecell.aml (page 3 of 5):

```
&sv bHH = [extract 4 %st%]
&sv en = [unquote %end%]
&sv eMM = [extract 1 %en%]
&sv eDD = [extract 2 %en%]
&sv eYY = [extract 3 %en%]
&sv eHH = [extract 4 %en%]
/* Initialize the MM, DD, YY, and HH values.
&sv MM = %bMM%
&sv DD = %bDD%
&sv YY = %bYY%
&sv HH = %bHH%

/* Set the maximum number of days in the current month.

&if %MM% eq 02 &then
 &do
  &if %YY% eq 96 &sv maxDD = 29
  &else &sv maxDD = 28
 &end
&if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11 &then
  &setvar maxDD = 30
&else
  &setvar maxDD = 31

/* OPEN A FILE TO STORE THE FILE NAMES OF THE PRECIPITATION
/*   DATA TO BE USED

&if [exists input -file] &then rm input
&setvar fileunit = [open input openstat -write]
&if %openstat% ne 0 &then
   &return &inform Could not open file input.

/* Set the prefix and suffix for the rainfall files according
/*   to the basin selected.

&if %.basin% eq mbrfc &then
  &do
   &setvar prefix = r.xmrg
   &setvar suffix = z
  &end
&else
  &do
```

onecell.aml (page 4 of 5):

```
   &sv prefix = none
   &sv suffix = z.nc
  &end
/* SET THE END DATE OF THE STORM
&setvar enddate = %eMM%%eDD%%eYY%%eHH%
/* GENERATE THE FILENAMES OF THE PRECIP DATA.  WRITE THESE
/* NAMES TO THE OUTPUT FILE, AND STORE THEM IN AN ARRAY.

&setvar number = 1

&do &until %date% eq %enddate%
  &setvar date = %MM%%DD%%YY%%HH%
  &setvar flnm = %date%
  &if %prefix% ne none &then &sv flnm = %prefix%%flnm%
  &if %suffix% ne none &then &sv flnm = %flnm%%suffix%
  &setvar file%number% = %flnm%
  &setvar number = %number% + 1
  &setvar flnm = [quote ']%flnm%
  &setvar flnm = %flnm%[quote ']
  &setvar writestat = [write %fileunit% %flnm%]
  &if %HH% eq 23 &then
    &do
      &setvar HH = 00
      &setvar DD = %DD% + 1

      &if %DD% lt 10 &then &setvar DD = 0%DD%
      &if %DD% gt %maxDD% &then
        &do
          &setvar DD = 01
          &setvar MM = %MM% + 1
          &if %MM% lt 10 &then &setvar MM = 0%MM%
          &if %MM% gt 12 &then
            &do
              &setvar MM = 01
              &setvar YY = %YY% + 1
              &if %YY% gt 99 &then &setvar YY = 0
              &if %YY% lt 10 &then &setvar YY = 0%YY%
            &end
          &if %MM% eq 02 &then
           &do
            &if %YY% eq 96 &then &sv maxDD = 29
            &else &sv maxDD = 28
```

onecell.aml (page 5 of 5):

```
        &end
        &if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11 &then
          &setvar maxDD = 30
        &else &setvar maxDD = 31
      &end
    &end
  &else
    &do
      &setvar HH = %HH% + 1
      &if %HH% lt 10 &then &setvar HH = 0%HH%
    &end
&end
&setvar writestat = [write %fileunit% done]
&setvar file%number% = done
/*  Close the input file
&sv closestat = [close %fileunit%]
/*  Return to the Main Progam
&return
```

pointinput.aml (page 1 of 3):

```
/* pointinput.aml

/* This aml generates a point coverage.

&terminal 9999
&messages &popup

/* Open a file to contain longitude and latitude data and
/*   one to contain station information.

&setvar file1 := [open %.name%lonlat.dat openstat -write]
&if %openstat% ne 0 &then
 &return &inform Could not open file.
&setvar file2 := [open %.name%statid.dat openstat -write]
&if %openstat% ne 0 &then
 &return &inform Could not open file.

/* Enter each point's coordinates and corresponding information.
/*   Write the coordinate information to %.name%lonlat.dat and the
/*   station information to %.name%statid.dat

&sv n = 1

&do &until %check% eq No

 &sv coords = [response 'Enter the point coords in DD~
 (ie -94.8694 38.7531).']
 &sv coords = [unquote %coords%]
 &sv coords = %n% %coords%
 &sv writestat = [write %file1% [quote %coords%]]
 &sv id = [response 'Enter a point ID# and~
 a point name (ie Little_Bull)']
 &sv id = [unquote %id%]
 &sv id = %n% %id%
 &sv writestat = [write %file2% [quote %id%]]
 &sv check = [getchoice Yes No -prompt 'Do you wish to input another point?']
 &if %check% eq No &then
  &do
   &sv writestat = [write %file1% end]
   &sv closestat = [close -all]
  &end
 &sv n = %n% + 1
```

pointinput.aml (page 2 of 3):

```
&end
&r msworking init 'Creating the Point Coverage.' 'Waiting Time = SHORT'~
 '&cc' 'pointinput.aml'
&messages &off &all

/* Build a point coverage.
generate %.name%stats
input %.name%lonlat.dat
points
quit

build %.name%stats points
addxy %.name%stats

/* Add attribute data to the point coverage
tables
define %.name%.dat

%.name%stats-id
4
4
i

stat-num
10
10
c

stat-name
15
15
c

~

add from %.name%statid.dat
quit

joinitem %.name%stats.pat %.name%.dat %.name%stats.pat %.name%stats-id
%.name%stats-id
project cover %.name%stats %.name%point /files/arcfiles/projection/flowgage.prj
```

pointinput.aml (page 3 of 3):

```
tables
kill %.name%.dat
q

kill %.name%stats

rm %.name%lonlat.dat
rm %.name%statid.dat

&r msworking close
&messages &popup
&r msinform init 'The Following Coverage was Created' %.name%point~
 '&cc' 'pointinput.aml'

&return
```

precip.aml (page 1 of 12):

```
/* precip.aml

/* This aml creates hourly grids of NEXRAD precip data.

&TERMINAL 9999
&MESSAGES &popup
/*&echo &on
/*-----------------------------------------------------------------
/* Check for the existance of files necessary for the
/*   operation of this aml.
/*-----------------------------------------------------------------

&if [exists %.name%clip -grid] ne .TRUE. &then
 &do
  &return &inform 'The %.name%clip grid does not exist'
 &end

&if [exists ws%.name% -cover] ne .TRUE. &then
 &do
  &messages &on
  &return &inform 'The ws%.name% coverage does not exist'
 &end

/*-----------------------------------------------------------------
/* Select the river basin, and enter
/*   the start and end times of the storm.
/*-----------------------------------------------------------------


&menu storm1.menu &form &stripe "%.storm%' Storm'
&messages &off
&messages &on

/*----------------------------------------------------------------------
/* Set file type to use in precipfiles.
/* Set the hour to 12 if daily files are being used.
/*----------------------------------------------------------------------

&if %.basin% = abrfc &then
   &do
        &if %time% = 'Hourly' &then &sv period = H
```

precip.aml (page 2 of 12):

```
        &if %time% = 'Daily' &then &sv period = D
/*      &if %time% = 'Monthly' &then &sv period = M
  &end
&if %.basin% = mbrfc &then &sv period = H
&if %time% = Hourly &then &sv .ext = hr
&if %time% = Daily &then &sv .ext = dy
&messages &on

/*------------------------------------------------------------
/*If the coverage %.storm%area exists, ask the user for permission
/*to kill it
/*------------------------------------------------------------

&if [exists %.storm%area -coverage] &then
  &do
        &sv getrid = [getchoice Yes Cancel -prompt 'Do you want to kill the
'%.storm%'area coverage?']
        &if %getrid% = Yes &then kill %.storm%area

        &if %getrid% = Cancel &then &return
  &end
&if [exists %.storm%hrap -coverage] &then
  &do
        &sv getrid = [getchoice Yes Cancel -prompt 'Do you want to kill the
'%.storm%'hrap coverage?']
        &if %getrid% = Yes &then kill %.storm%hrap
        &if %getrid% = Cancel &then &return
  &end

/*------------------------------------------------------------------
/* Select an area around the watershed boundary
/*   for which the precip values will be gridded.
/*   An area larger than the watershed boundary
/*   is used to ensure adequate information for
/*   contouring.
/* Enter arcedit, set the backcoverages (coverages
/*  to be viewed but not edited) and the backenvironment.
/*------------------------------------------------------------

arcedit
 display 9999
 mape ws%.name%
```

precip.aml (page 3 of 12):

```
 backcoverage ws%.name% 3
 backcoverage /files/arcfiles/kan_hrap/kangeoccalb 2
 backenvironment poly on
 drawenvironment poly on

/* Create a new coverage called %.storm%area. This coverage
/*   will have the same extents as the ws%.name% coverage.
/*   Build polygon topology for the new coverage, then set the
/*   editfeature to polygon.

 create %.storm%area ws%.name%
 build
 editfeature poly

/* Inform the user to draw the polygon.

&messages &pop
&type After quitting this window, zoom out from the watershed
&type and draw a polygon that encompasses the entire watershed
&type and enough HRAP cells around it to permit accurate contouring
&type of the precip data within the watershed. Use the mouse
&type button to draw the polygon, then press 2.
&messages &on

 draw
 add one
 save
 quit

/* Intersect the new polygon and the hrap coverage.

 intersect %.storm%area /files/arcfiles/kan_hrap/kangeoccalb %.storm%hrap~
 poly

/* Enter tables, select the %.storm%hrap.pat file, reselect to
/*   unselect all records, then use aselect to select all
/*   but the universal polygon. Unload the items %.storm%hrap-id,
/*   hrapx, and hrapy for the selected records to a file named
/*   hrapinput. Quit tables.

&if [exists hrapinput -file] &then
  rm hrapinput
```

precip.aml (page 4 of 12):

```
tables
 sel %.storm%hrap.pat
 reselect
 aselect %.storm%hrap-id >.0
 unload hrapinput %.storm%hrap-id hrapx hrapy
quit

/* Open the hrapinput file with the append option. Append the
/*  file with the line '0,0,0', which will signal the fortran
/*  program that the end of the file has been reached.

&s file1 = [open hrapinput openstat -append]
&s writestat = [write %file1% '0,0,0']
&s closestat = [close %file1%]

/* Call the precipfiles subroutine, which will create the
/*  'input' file with the names of the precip files to be
/*  opened by the fortran program.

&call precipfiles

/* CALL THE FORTRAN PROGRAM TO READ THE PRECIP DATA.
/* THE FORTRAN PROGRAM WILL READ THE INPUT AND THE HRAPINPUT
/* DATA FILES, OPEN THE APPROPRIATE RAINFALL FILES, AND
/* OUTPUT ONE ASCII FILE FOR EACH HOUR OF RAINFALL. EACH
/* FILE WILL CONTAIN A LIST OF THE %.STORM%HRAP-ID AND THE
/* APPROPRIATE PRECIP.

&if %.basin% eq abrfc &then
 &do
  cp /files/arcfiles/new/mapab mapab
  mapab
  rm mapab
 &end
&if %.basin% eq mbrfc &then
 &do
  &if %time% = Daily &then
   &do
      cp /files/arcfiles/interface/mapmbday mapmbday
      mapmbday
      rm mapmbday
   &end
```

precip.aml (page 5 of 12):

```
 &if %time% = Hourly &then
&do
        cp /files/arcfiles/interface/mapmb mapmb
        mapmb
        rm mapmb
  &end
&end

/* Check to see if an info file exists under the name of 'rainfall'
/*   if the file does exist, then proceed.  If it does not exist,
/*   create it.

&if [exists rainfall -info] &then
 &do
  tables
    kill rainfall
  quit
 &end
  tables
  define rainfall

  %.storm%hrap-id
  4
  5
  b
  precip
  5
  6
  i
  ~
  q

/* For each output file of the designated time
/*  period, read the data into the rainfall
/*   file, join that file with the %.storm%hrap.pat file, then
/*   grid the data to %.storm%hr%n%.

&setvar number = 1
&messages &on
&label again

/*&type [value %.storm%dy%number%]
```

```
&if [exists %.storm%dy%number% -grid] &then
  &do
        &sv getrid = [getchoice Yes Cancel -prompt 'Do you want to kill the
'%.storm%%.ext%%number%' grid?']
        &if %getrid% = Yes &then kill %.storm%%.ext%%number%
        &if %getrid% = Cancel &then &return
  &end
 tables
 select rainfall
 purge
 y
 add from out.[value file%number%]
quit
/*rm out.[value file%number%]
joinitem %.storm%hrap.pat rainfall %.storm%hrap.pat~
 %.storm%hrap-id hrapy
&do
  polygrid %.storm%hrap %.storm%%.ext%%number% precip
  %cs%
  y
&end
dropitem %.storm%hrap.pat %.storm%hrap.pat precip
&sv number = %number% + 1
&if [value file%number%] ne done &then &goto again
&label finish

/*-------------------------------------------------
/* The aml creates one coverage for the last hour
/* in the storm, which is not included in the
/* whole day. This coverage is killed here.
/*-------------------------------------------------

&if %time% = Daily and %.basin% = mbrfc &then
  &do
        &sv number = %number% - 1
        kill %.storm%%.ext%%number%
  &end
&messages &off
/* Make sure that all aml files are properly closed, then
/*  return.

&setvar closestat = [close -all]
```

```
rm input
rm hrapinput

&return

/*--------------------------------------------------------------
/* The following routine creates a file with all of the file names
/*  to be opened.  A similar list is stored in the array file%number%
/*--------------------------------------------------------------

&routine Precipfiles

/* Set the beginning and end month, day, year, and hour of the storm.
/*   Use the extract function to accomplish this.

/* Initialize the MM, DD, YY, and HH values.

&sv MM = %bMM%
&sv DD = %bDD%
&sv YY = %bYY%
&sv HH = %.bHH%

/* Set the maximum number of days in the current month.

&if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11 &then
  &setvar maxDD = 30
&else
  &setvar maxDD = 31
&if %MM% eq 02 &then
 &do
  &if %YY% eq 96 or %YY% = 04 &then &sv maxDD = 29
  &else &sv maxDD = 28
 &end
/*--------------------------------------------------------------
/* OPEN A FILE TO STORE THE FILE NAMES OF THE PRECIPITATION
/*   DATA TO BE USED
/*--------------------------------------------------------------

&if [exists input -file] &then rm input
&setvar fileunit = [open input openstat -write]
&if %openstat% ne 0 &then
   &return &inform Could not open file input.
```

precip.aml (page 8 of 12):

```
/*------------------------------------------------------------
/*  Set the prefix and suffix for the rainfall files according
/*   to the basin selected.
/*------------------------------------------------------------

&if %.basin% eq mbrfc &then
 &do
  &setvar prefix = r.xmrg
  &setvar suffix = z
 &end
&else
 &do
  &sv prefix = none
/*    &messages &popup
/*    &sv suffix = [response 'Enter the suffix for the ABRFC files']
/*    &messages &on
   &sv suffix = z.nc
 &end

/* SET THE END DATE OF THE STORM

&setvar enddate = %eMM%%eDD%%eYY%%eHH%
/*------------------------------------------------------------
/* GENERATE THE FILENAMES OF THE PRECIP DATA.  WRITE THESE
/* NAMES TO THE OUTPUT FILE, AND STORE THEM IN AN ARRAY.
/*------------------------------------------------------------

&setvar number = 1

&do &until %date% eq %enddate%
  &setvar date = %MM%%DD%%YY%%HH%
  &setvar flnm = %date%
  &if %prefix% ne none &then &sv flnm = %prefix%%flnm%
  &if %suffix% ne none &then &sv flnm = %flnm%%suffix%
  &sv file%number% = %flnm%
  &sv number = %number% + 1
  &setvar flnm = [quote']%flnm%
  &setvar flnm = %flnm%[quote']
  &setvar writestat = [write %fileunit% %flnm%]
   &if %period% eq D &then
    &do
       &if %DD% eq %maxDD% &then
```

precip.aml (page 9 of 12):

```
            &do
                &setvar DD = 1
                &setvar MM = %MM% + 1
                &if %MM% lt 10 &then &setvar MM = 0%MM%
                &if %MM% gt 12 &then
                  &do
                       &setvar MM = 01
                       &setvar YY = %YY% + 1
                       &if %YY% gt 99 &then &setvar YY = 0
                       &if %YY% lt 10 &then &setvar YY = 0%YY%
                  &end
                &if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11
&then &setvar maxDD = 30
                &else &setvar maxDD = 31
                &if %MM% eq 02 &then
                  &do
                       &if %YY% eq 96 or %YY% eq 04 &then &setvar maxDD = 29
                       &else &setvar maxDD = 28
                  &end
            &end
          &else &setvar DD = %DD% + 1
          &if %DD% lt 10 &then &setvar DD = 0%DD%
       &end
    &else
          &if %HH% eq 23 &then
             &do
               &setvar HH = 00
               &setvar DD = %DD% + 1
                 &if %DD% lt 10 &then &setvar DD = 0%DD%
                 &if %DD% gt %maxDD% &then
            &do
                       &setvar DD = 01
                       &setvar MM = %MM% + 1
                       &if %MM% lt 10 &then &setvar MM = 0%MM%
                       &if %MM% gt 12 &then
                  &do
                       &setvar MM = 01
                       &setvar YY = %YY% + 1
                       &if %YY% gt 99 &then &setvar YY = 0
                       &if %YY% lt 10 &then &setvar YY = 0%YY%
                  &end
```

precip.aml (page 10 of 12):

```
                        &if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM%
eq 11 &then &setvar maxDD = 30
                        &else &setvar maxDD = 31
                        &if %MM% eq 02 &then
                          &do
                        &if %YY% eq 96 &then &sv maxDD = 29
                        &else &sv maxDD = 28
                          &end

          &end
          &end
        &else
          &do
        &setvar HH = %HH% + 1
        &if %HH% lt 10 &then &setvar HH = 0%HH%
          &end
&end
&setvar writestat = [write %fileunit% done]
&sv .num%.storm% = %number%
&setvar file%number% = done

/*  Close the input file

&sv closestat = [close %fileunit%]

/*-------------------------------------------------------
/*Create a file containing the names of the daily files.
/*Initialize the date.
/*-------------------------------------------------------

&if %.basin% = mbrfc and %time% = 'Daily' &then
&do

&sv MM = %bMM%
&sv DD = %bDD%
&sv YY = %bYY%
&sv HH = %.bHH%

/*Check to see if the file containing the names of the daily
/*files exists.  If it does, remove it.

&if [exists days -file] &then rm days
```

precip.aml (page 11 of 12):

```
&sv fileunit = [open days openstat -write]
&if %openstat% ne 0 &then &return &inform Could not open file days.

/* Set the maximum number of days in the current month.
&if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11 &then
  &setvar maxDD = 30
&else

  &setvar maxDD = 31
&if %MM% eq 02 &then
 &do
  &if %YY% eq 96 or %YY% = 04 &then &sv maxDD = 29
  &else &sv maxDD = 28
 &end

/*Make a list of the dates and write to a file called days.
   &sv number = 0
   &do &until %date% eq %enddate%
         &sv date = %MM%%DD%%YY%%HH%
         &setvar flnm = %date%
         &if %prefix% ne none &then &sv flnm = %prefix%%flnm%
         &if %suffix% ne none &then &sv flnm = %flnm%%suffix%

         &setvar number = %number% + 1
         &setvar file%number% = %flnm%
         &setvar flnm = [quote']%flnm%
         &setvar flnm = %flnm%[quote']
         &setvar writestat = [write %fileunit% %flnm%]
         &if %DD% eq %maxDD% &then
             &do
                  &setvar DD = 1
                  &setvar MM = %MM% + 1
                  &if %MM% lt 10 &then &setvar MM = 0%MM%
                  &if %MM% gt 12 &then
                     &do
                           &setvar MM = 01
                           &setvar YY = %YY% + 1
                           &if %YY% gt 99 &then &setvar YY = 0
                           &if %YY% lt 10 &then &setvar YY = 0%YY%
                     &end
                  &if %MM% eq 04 or %MM% eq 06 or %MM% eq 09 or %MM% eq 11
&then &setvar maxDD = 30
```

precip.aml (page 12 of 12):

```
        &else &setvar maxDD = 31
        &if %MM% eq 02 &then
    &do
              &if %YY% eq 96 &then &setvar maxDD = 29
              &else &setvar maxDD = 28
          &end
     &end
   &else &setvar DD = %DD% + 1
   &if %DD% lt 10 &then &setvar DD = 0%DD%
 &end
&sv writestat = [write %fileunit% 'done']
&sv .num%.storm% = %number%
&setvar file%number% = done

/*Close the file containing the list of daily files.

&sv closestat = [close %fileunit%]
&end

/*  Return to the Main Program

&return
```

precip.menu (page 1 of 1):

```
7 precip.menu
/*
  Select the Desired Option:

    %button8

    %button7

    %button1

    %button2

    %button3

    %button4

    %button5

    %button9

    %button6
%button8 BUTTON KEEP 'Define a Watershed Boundary' &r definews.aml
%button7 BUTTON KEEP 'Load MBRFC Rainfall Files to Local Directory' &r copymb
%button1 BUTTON KEEP 'Generate Hourly Precip Grids' &r precip.aml
%button2 BUTTON KEEP 'Contour Precip Grids' &r contour.aml
%button3 BUTTON KEEP 'Compile Storm Statistics' &r genstats.aml
%button4 BUTTON KEEP 'Display the HRAP Grid' &r hrapview.aml
%button5 BUTTON KEEP 'Generate Rainfall Table for one HRAP Cell' &r onecell.aml
%button9 BUTTON KEEP ~
  HELP 'Type &return to return' ~
  'Arc Prompt' &tty
%button6 BUTTON KEEP 'Cancel' &return
```

preproc.aml (page 1 of 1):

```
/* preproc.aml

/*
/* This aml permits the interactive preprocessing of stream
/*   data for use in topogrid.
/*

&term 9999
&messages &on

/* Enter arcedit, turn the messages to popup

arcedit
display 9999 position ur

/* Select the stream coverage to be processed.  Set
/*   the map extent, the edit coverage, the edit feature
/*   and the draw environment.  Draw the coverage, then
/*   call the menu.

&sv cov = [getcov # -all 'Select the Streams Coverage to Edit:']

mape %cov%

editc %cov%

editf arc

drawe arc on arrows

draw

&m preproc.menu &position &left &display &stripe 'Edit Menu'

&return
```

preproc.menu (page 1 of 1):

```
7 preproc.menu
/*
 Select the Edit Option:

   %button1

   %button2

   %button3

   %button4

   %button5

   %button6

   %button7

   %button8

   %button9


%button1 BUTTON KEEP 'Re-Draw' draw
%button2 BUTTON KEEP 'Select Arcs' select many
%button3 BUTTON KEEP 'UnSelect Arcs' unselect many
%button4 BUTTON KEEP 'Clear Selection' unselect all
%button5 BUTTON KEEP 'Delete Selected Arcs' delete
%button6 BUTTON KEEP 'Add New Arcs' add
%button7 BUTTON KEEP 'Flip Selected Arcs' flip
%button8 BUTTON KEEP 'Save Edits' save
%button9 BUTTON KEEP 'Quit' &s save = [getchoice Y N -prompt~
 'Do you wish to save your edits?']; quit; %save%; &return
```

ptv.menu (page 1 of 1):

```
7 ptv.menu
/*
 Select the Point Coverage you wish to view:

 %datalist1              Marker Type:

                      %symbollist1




    %button1          %button2
%datalist1 INPUT point 24 TYPEIN NO SCROLL YES ROWS 6 COVER ~
  * -POINT -SORT
%symbollist1 INPUT mt 14 TYPEIN NO SCROLL YES ROWS 4 ~
  KEEP ~
  RETURN 'markersymbol %mt%' ~
  SYMBOL ~
  -MARKER
%button1 BUTTON KEEP 'Apply' points %point%; &return
%button2 BUTTON KEEP 'Cancel' &return
```

pv.menu (page 1 of 1):

```
7 pv.menu
/*

 Select the Polygon Coverage you wish to view:

 %datalist1              Select Item to Shade:
                       %datalist2


                       Lookup Table (if any):
                       %input1

 Select the Desired Symbol Sets:

 Outlines:        ^Shade:
                          %button1
 %symbollist1      %symbollist2
                          %button3

                          %button2
%datalist1 INPUT POLY 24 TYPEIN NO SCROLL YES ROWS 6 COVER ~
 * -POLY -SORT
%datalist2 INPUT SHDITEM 20 TYPEIN NO SCROLL YES ROWS 2 ITEM ~
 %poly%.pat -INFO -ALL -ALL
%input1 INPUT LUT 25 TYPEIN YES SCROLL NO CHARACTER
%button1 BUTTON KEEP 'Display Outlines' arclines %poly% %outline%
%symbollist1 INPUT OUTLINE 15 TYPEIN NO SCROLL YES ROWS 4 SYMBOL ~
 -LINE
%symbollist2 INPUT ITEM 14 TYPEIN NO SCROLL YES ROWS 4 SYMBOL ~
 -SHADE
%button3 BUTTON KEEP 'Display Shading' polygonshade %poly% %shditem% %lut%
%button2 BUTTON KEEP 'Return' &return
```

selbasin.menu (page 1 of 1):

```
7 selbasin.menu
/*
  Which River Basin Contains the Area of Interest?

        %button1

        %button2

%button1 BUTTON KEEP 'Arkansas-Red River Basin' &sv .basin = abrfc; &return
%button2 BUTTON KEEP 'Missouri River Basin' &sv .basin = mbrfc; &return
```

storm1.menu (page 1 of 2):

```
7 storm1.menu
/*
     Use this menu to define the storm.

  Choose the START filename (GMT):
  %datalist1      %l2      %datalist3   %datalist2_1







  Choose the END filename (GMT):
  %datalist1_1      %l2_1     %datalist3_1 %datalist2_2







   File type to use:           ^Cell size (meters):
   %choice1                    %slider1


                             %button3
   Choose the basin:               %button1
   %choice2                        %button2
%datalist1 INPUT MONTH 17 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED ~
 INITIAL 'January' ~
 CHOICE ~
 -file /files/arcfiles/interface/mos -display 1 -value 2 -var bMM -prompt Month
%l2 INPUT BDD 8 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED ~
 INITIAL '01' ~
 CHOICE ~
 -FILE /files/arcfiles/interface/daysofmo -DISPLAY 1 -SORT -prompt Day
%datalist3 INPUT YEAR 11 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED CHOICE ~
 -file /files/arcfiles/interface/years -display 1 -value 2 -var bYY -prompt Year
%datalist2_1 INPUT .BHH 8 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED ~
```

storm1.menu (page 2 of 2):

```
INITIAL '01' ~
 CHOICE ~
 -file /files/arcfiles/interface/hours -prompt Hour
%datalist1_1 INPUT MONTH2 17 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED ~
 INITIAL 'January' ~
 CHOICE ~
 -file /files/arcfiles/interface/mos -display 1 -value 2 -var eMM -prompt Month
%l2_1 INPUT EDD 8 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED ~
 INITIAL '01' ~
 CHOICE ~
 -FILE /files/arcfiles/interface/daysofmo -DISPLAY 1 -SORT -prompt Day
%datalist3_1 INPUT YEAR2 11 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED ~
 INITIAL '1996' ~
 CHOICE ~
 -file /files/arcfiles/interface/years -display 1 -value 2 -var eYY -prompt Year
%datalist2_2 INPUT EHH 8 TYPEIN NO SCROLL YES ROWS 6 ~
 REQUIRED CHOICE ~
 -file /files/arcfiles/interface/hours
%choice1 CHOICE TIME SINGLE ~
 INITIAL 'Hourly' ~
 'Hourly' 'Daily'
%slider1 SLIDER CS 22 TYPEIN YES ~
 INITIAL 1000 ~
 STEP 100 ~
 INTEGER 100 4000
%button3 BUTTON KEEP ~
 HELP 'Type &return to return' ~
 'Arc Prompt' &tty
%button1 BUTTON KEEP 'Start grid' &sv cancel = no;&return
%choice2 CHOICE .BASIN PAIRS ~
 INITIAL 'mbrfc' ~
 'Missouri' 'mbrfc' 'Arkansas-Red' 'abrfc'
%button2 BUTTON KEEP RETURN 'Cancel' &sv cancel = cancel;&return
```

surgatt.aml (page 1 of 3):

```
/* ssurgatt.aml

/* This aml processes SSURGO attribute data from DASC

&TERM 9999
&MESSAGES &POPUP

/* Check if the watershed soils coverage exists.

&if [exists ws%.name%soil -cover] eq .FALSE. &then
 &do
  &type The soils coverage does not exist.
  &return
 &end

/* Enter the number of Counties intersecting the watershed.

&setvar counties = [response 'Enter the number of Counties Intersected']
&setvar n = 1

/* Enter each file name, save these names in an array.

&do &while %n% le %counties%
 &setvar county%n% [response 'Enter the two letter county abreviation']
 &sv n = %n% + 1
&end

/* Get rid of the POPUP menu option.

&messages &on
&setvar n = 1
/* For each filename: check if it exists, then import the *.e00 file.
&do &while %n% le %counties%
 &if [exists [value county%n%]_layer.e00 -file] eq .FALSE. &then
  &do
   &messages &popup
   &type File [value county%n%]_layer.e00 does not exist
   &return
  &end
 &if [exists [value county%n%]_comp.e00 -file] eq .FALSE. &then
  &do
   &messages &popup
```

surgatt.aml (page 2 of 3):

```
 &type File [value county%n%]_comp.e00 does not exist
   &return
  &end

 import info [value county%n%]_layer.e00 [value county%n%]_layer

 import info [value county%n%]_comp.e00 [value county%n%]_comp

 &sv n = %n% + 1
&end
/* Enter tables.  Define a new attribute table %.name%_layer.
/*  This table will include the muid, texture, depth, and musym.
/*  For each county intersected, alter the laydeph:i item to make it a
/*  numeric item, relate the _layer and _comp files, select the
/*  _layer file, reselect, aselect the top layer of soil.
/*  unload the muid, texture, laydeph, and musym (from comp).
tables

 define %.name%_layer
  muid
   7
   7
   c
  texture
   10
   10
   c
  depth
   2
   2
   i
  musym
   5
   5
   c
  ~
 &do n = 1 &to %counties% &by 1
  sel [value county%n%]_layer
  alter
   laydeph:i
   laydeph:i
   7
```

surgatt.aml (page 3 of 3):

```
 n
  0
  ~
  ~
  relate add
   [value county%n%]
   [value county%n%]_comp
   info
   muid:c
   muid:c
   linear
   rw
   ~
  reselect
  aselect layernum:i cn '1'
  unload [value county%n%]data muid:c texture:c laydeph:i~
[value county%n%]//musym:c
 sel %.name%_layer
 add from [value county%n%]data
&end

quit

&do n = 1 &to %counties% &by 1
 rm [value county%n%]data
&end

joinitem ws%.name%soil.pat %.name%_layer ws%.name%soil.pat muid muid linear

&return
```

ssurgo.aml (page 1 of 4):

```
/* ssurgo.aml

/* This aml processes SSURGO spatial data from DASC

&TERM 9999
&MESSAGES &POPUP

/* Enter the number of tiles intersecting the watershed.

&setvar tiles = [response 'Enter the number of Tiles Intersected']

/* Enter the UTM Zone for the Data (saved as utmzone)

&sv junk = [getchoice -pairs 'West of 102 Degrees West' 13~
 'Between 96 and 102 Degrees West' 14 'East of 96 Degrees West' 15~
 -var utmzone -prompt 'Select the Location of the Watershed:']

&setvar n = 1

/* Enter each file name, save these names in an array.

&do &while %n% le %tiles%
 &setvar tile%n% [response 'Enter the base name of the tile (no extension):']
 &sv n = %n% + 1
&end

/* Check if the watershed boundary exists for the coverage.

&if [exists ws%.name% -cover] eq .FALSE. &then
  &do
    &type The watershed boundary does not exist
    &return
  &end


/* Get rid of the POPUP menu option.
&messages &on
&setvar n = 1
/* For each filename: check if it exists, then import the *.e00 file.

&do &while %n% le %tiles%
 &if [exists [value tile%n%].e00 -file] eq .FALSE. &then
```

ssurgo.aml (page 2 of 4):

```
 &do
  &messages &popup
  &type Tile [value tile%n%].e00 does not exist
  &return
&end
 import cover [value tile%n%].e00 [value tile%n%]soil
 &sv n = %n% + 1
&end

/* Project the existing polygon watershed boundary into the
/*   projection of the landcover files (geographic coordinates).
/*   Build topology for the projected coverage.

project cover ws%.name% wsclip
 output
  projection utm
  units meters
  zone %utmzone%
  parameters
 end

build wsclip poly
/* If there is only one soils coverage, then clip it to the
/*   watershed boundary, project it, and build topology.
&if %tiles% eq 1 &then
 &do
  clip %tile1%soil wsclip clipcov poly
  project cover clipcov ws%.name%soils
   input
    projection utm
    units meters
    zone %utmzone%
    parameters
   output
    projection albers
    units meters
    parameters
    37 30 00
    39 30 00
    -98 00 30
    37 00 00
    0.0
```

ssurgo.aml (page 3 of 4):

```
    0.0
   end
  build ws%.name%soils poly
  kill clipcov
  kill wsclip
 &end
&else
 &do
/* If there are more than one tiles:
/* Mapjoin the SSURGO coverages, clipping the final coverage to
/*   the watershed boundary.

  mapjoin joincov poly # wsclip
   &do n := 1 &to %tiles% &by 1
    [value tile%n%]soil
   &end
  end

/* Use the dissolve command to remove the border between the
/* counties.

  dissolve joincov discov muid poly

/* Project the dissolved coverage into the desired projection,
/*  build topology, and then kill all of the intermediary coverages.

  project cover discov ws%.name%soil
    input
     projection utm
     units meters
     zone %utmzone%
     parameters
    output
     projection albers
     units meters
     parameters
     37 30 00
     39 30 00
     -98 00 30
     37 00 00
     0.0
     0.0
```

ssurgo.aml (page 4 of 4):

```
 end
 build ws%.name%soil poly

 kill discov
 kill joincov
 kill wsclip

&end

&do n := 1 &to %tiles% &by 1
 kill [value tile%n%]soil
&end

&return
```

view.aml (page 1 of 1):

```
/*  view.aml

/*  This aml permits the user to view coverages.

&TERM 9999
&MESSAGES &ON

grid
 display 9999 position ur
 &menu view.menu &position &left &display &stripe 'Coverage Viewer'
quit

&return
```

wshed.aml (page 1 of 2):

```
/* The following aml delineates user-specified watersheds

&term 9999
&messages &popup

/* Enter the base name of the DEM

&sv dem = %.name%

/* Check to see if the flowdirection grid exists

&if [exists %dem%fdr -grid] eq .FALSE. &then
 &do
  &type Flowdirection grid does not exist
  &return
 &end

/* Check to see if the flowaccumulation grid exists

&if [exists %dem%fac -grid] eq .FALSE. &then
 &do
  &type Flowaccumulation grid does not exist
  &return
 &end
/* Display the flowaccumulation grid

grid
 display 9999
 mape %dem%fac
 gridp %dem%fac # linear # gray
/* Ask user if a gage coverage should be displayed. If so,
/*   let the user pick the coverage and display it.

 &sv ask1 = [response 'Do you wish to display an existing gage coverage~
 (y or n)']
 &if %ask1% eq y &then
  &do
   markercolor 3
   &sv ptgage = [getcover * -point 'Select the Gage Coverage']
   points %ptgage%
  &end
```

wshed.aml (page 2 of 2):

```
/* Ask user if a line coverage should be displayed. If so,
/*   let the user pick the coverage and display it.
 &sv ask2 = [response 'Do you wish to display an existing line coverage~
 (y or n)']
 &if %ask2% eq y &then
&do
  &sv line = [getcover * -line 'Select a Line Coverage']
  arclines %line% 4
 &end

/* Ask if the user wished to zoom in, if so prompt for the
/*   new map extent and redisplay the coverage.

 &sv ask3 = [response 'Do you wish to zoom in (y or n)']
 &if %ask3% eq y &then
  &do &until %ask3% eq n
   &messages &on
   &type Select the Mape Extent
   mape *
   gridp %dem%fac # linear # gray
   &if %ask1% eq y &then
    points %ptgage%
   &if %ask2% eq y &then
    arclines %line% 4
   &messages &popup
   &sv ask3 = [response 'Do you wish to zoom in again (y or n)']
  &end

/* Ask the user to specify the watershed outlet points, then
/*   execute the watershed command. Create a polygon coverage
/*   of the watershed boundary.

 &messages &on
 &type Select the watershed outlet points

 %dem%ws = watershed(%dem%fdr, selectpoint(%dem%fac, *))

 ws%dem% = gridpoly (%dem%ws)

 quit

 &return
```