ORNL/TM-13071

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

TERRAIN:
A Computer Program To Process Digital Elevation Models for Modeling Surface Flow

Paul M Schwartz
Sidey P. Timmins
Daniel A. Levine
Carolyn T. Hunsaker

Environmental Sciences Division
Publication No.4455

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-13071

Environmental Sciences Division

TERRAIN: A Computer Program to Process Digital Elevation Models for Modeling Surface Flow

Paul M. Schwartz[1], Sidey P. Timmins[2],
Daniel A. Levine[3], and Carolyn T. Hunsaker[4]

1 Environmental Sciences Division, ORNL, P.O. Box 2008, Oak Ridge, TN 37831-6038, ztn@ornl.gov.
2 Analysas Corporation, 151 Lafayette Drive, Suite 310, Oak Ridge, TN 37830, stq@ornl.gov.
3 Environmental Sciences Division, ORNL, P.O. Box 2008, Oak Ridge, TN 37831-6038, oki@ornl.gov.
4 Environmental Sciences Division, ORNL, P.O. Box 2008, Oak Ridge, TN 37831-6038, cth@ornl.gov.

Environmental Sciences Division
Publication No. 4455

Date Published: August 1995

Prepared for the
U.S. Department of Energy
Biofuels Systems Program

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6285
Managed by
LOCKHEED MARTIN ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

Contents

List of Figures

Acronyms and Abbreviations

DEC	Digital Equipment Corp.
DEM	digital elevation model
DOE	U. S. Department of Energy
HP	Hewlett Packard
I/O	input/output
USGS	United States Geological Survey

Acknowledgments

We are very grateful to Susan Jenson who wrote the programs that TERRAIN is based upon. The TERRAIN software itself was written and tested by Sidey P. Timmins for a variety of projects at the Environmental Sciences Division of Oak Ridge National Laboratory. The flopath portion of TERRAIN is based upon the doctoral research of Dan Levine.

This research was supported in part by an appointment to the U.S. Department of Energy (DOE) Laboratory Cooperative Postgraduate Research Training Program administered by the Oak Ridge Institute for Science and Education.

Abstract

This document provides a step by step procedure, TERRAIN, for processing digital elevation models to calculate overland flow paths, watershed boundaries, slope, and aspect. The algorithms incorporated into TERRAIN have been used at two different geographic scales: first for small research watersheds where surface wetness measurements are made, and second for regional water modeling for entire counties. For small areas methods based on flow distribution may be more desirable, especially if time-dependent flow models are to be used. The main improvement in TERRAIN compared with earlier programs on which it is based is that it combines the conditioning routines, which remove depressions to avoid water storage, into a single process. Efficiency has also been improved, reducing run times as much as 10:1 and enabling the processing of very large grids in strips for regional modeling. Additionally, the ability to calculate the nutrient load delivered any cell in a watershed has been added. These improvements make TERRAIN a powerful tool for modeling surface flow.

Chapter 1
Introduction

The landform may be represented by a grid of elevation values that may be used to model water flow over the land surface independent of time. The program TERRAIN is a major revision of earlier FORTRAN programs written by Jenson and Dominque (1988) to permit efficient processing of grid digital elevation models (DEMs). These programs were unique in their capability to model water flow over the landform using cells on a grid, each of which has eight nearest neighbor cells. In this model, water from any grid cell is routed, cell by cell, in one of the eight possible directions to the edges of the grid without storage. Although newer methods distribute flow from each cell to more than one neighboring cell, the routing to a single cell is satisfactory for large areas.

1.1 Theory

Viewed as a continuous surface, which can be used to model water flow over the land independent of time, the landform can be represented by a discreetly sampled grid. Despite the presence of variations in soils, sink holes, springs, and other natural irregularities that influence flow, water is assumed to flow over the surface under gravitational pull parallel to the topographic gradient. The grid of regularly spaced elevations representing the landform is termed a DEM. Each cell in the grid (which may be square or rectangular) represents an areal portion of the landform generalized to a particular elevation.

To perform flow modeling on the landform, flow directions must be determined at least as accurately as the source contour maps portray topographic gradient. Thus, the DEM must be prepared for this purpose. Unfortunately, models that represent area on a grid may be costly to create and to store for small cell sizes because grid creation and storage costs quadruple when cell size is halved. This practical consideration encourages the use of data at low resolutions.

1.2 Spatial Aliasing

Digital data sets such as DEMs must meet certain minimum resolution requirements if they are to represent adequately a continuous surface such as a landform. A fundamental requirement is the choice of a fine enough spatial sampling interval. Fourier analysis shows that an arbitrary curve may be approximated by the sum of an infinite series of sines and cosines, each with a particular amplitude (Fourier 1822). This implies that a digital sampling scheme will reconstruct an arbitrary curve to the limit that the digital samples can represent each frequency. The sampling theorem (Nyquist 1928) states that each period of a periodic waveform must be sampled at least twice to preserve the waveform (Hamming 1983).

An example can demonstrate these concepts. Consider a ridge and valley province with a dominant period (between consecutive peaks) of 1000 meters. It is desired to make a DEM at 1-km resolution to match Advanced Very High Resolution Radar data. If each sample is taken at 1-km spacing, then the landform will appear as a plane. To reconstruct the ridges and valleys from digital samples, the DEM must have a maximum spacing of 500 m.

To represent a curve or surface in a digital sense, we must have at least one sample in each peak and trough of the frequency we wish to keep. When this condition is not met, aliasing, a process by which high frequencies are misrepresented in a data set as smoother lower frequencies, will result. In practice aliasing can be avoided only by choosing a sufficiently fine spatial sampling interval to sample the ridges and valleys of interest.

The use of digital elevation samples on a conventional x-y grid is convenient for processing and viewing—provided the sampling theorem is obeyed. Each profile, row or column, in the grid samples the continuous surface at a regular interval—the grid spacing. This grid spacing must be chosen fine enough to avoid aliasing. For example, in the Walker Branch watershed the streams are less than 3 m wide in valleys ranging from 15 to 300 m wide. The grid spacing was chosen as 3 m to preserve the drainage network. Balce (1987) has shown that fine cell sizes improve the accuracy of slope preservation (i.e., higher frequencies in the landform shape).

1.3 Vertical Resolution

Both accuracy and resolution are required in the z-direction (i.e., elevation) because the determination of flow direction is based on the calculation of slopes. Since slope is a derivative, its calculation will magnify errors in the two component elevations. For example, two points exactly 1 m apart with elevations of 9 and 10 m [both ± .5 m] have a slope of 1 (± 1). If the vertical sample interval (unit) is too coarse, these differences will be obscured. For example, the net difference of 1 m for these two values is removed by rounding both elevations to 3 m. In some flat regions, such as Florida, even a meter vertical resolution is too coarse to reveal the drainage. For these reasons, one of the authors has typically generated DEM values from contours at a resolution of 0.03 m. The quality of U.S. Geological Survey (USGS) DEM data as reviewed by Moore et al. (1991) is found to be lacking for the purposes of flow modeling at some scales.

1.4 Advantages of Grid Models

There are significant advantages to representing the landform with a DEM. Representing the landform with a contour map, a common form of representation, is not easily visualized and should be replaced by a three-dimensional view. A DEM provides the regularly sampled surface that is necessary to draw 3-D views. A grid model is easily represented as a matrix, stored by column and row, and implicitly represents area and topology. Each cell (barring those along the edge) is surrounded by exactly eight neighbors, one on each of its four sides and one at each of its four corners. This simple cell arrangement is particularly useful for computer algorithms to model flow. As is shown later, this model does not have to constrain flow to only these eight directions.

The main disadvantage of grid DEMs vs the contour maps from which they are usually generated, is that the mapping (i.e., the transformation that describes a correspondence for information from one representation to another) is not necessarily one to one nor reversible. While a grid of elevation values is easily generated by interpolating between the contours to estimate the landform surface, the grid may not contain the same valleys and peaks as the contour map.

The digital elevation modeler can accept this limitation by demanding that water drainage over the model follow the main streams and rivers evident from the contour map. This requirement, known as drainage enforcement, removes spurious sinks and dams created by simple interpolation and permits DEMs to be used in hydrologic modeling (Hutchinson 1989). Of course, if a DEM is acquired from a source that does not guarantee drainage enforcement (such as USGS DEMs), then results for flow modeling will be variable.

1.5 Generation of Useful DEMs

This section describes the results that are obtainable using digital elevation models to represent the landform. These results seem to validate the modeling process and surpass what can be achieved using nongrid elevation models such as contours and triangular irregular networks.

The generation of useful DEMs for hydrologic modeling in x-y space requires custom interpolation algorithms. The goal is to create a digital sampling of the landform that will emulate the stream network and geometry of the landform itself. Because streams flow in the valleys where the shape of the land may change abruptly, this requirement for useful landform geometry affects flow modeling and cannot be ignored.

The simplest way to achieve valid landform geometry for flow modeling is to incorporate the stream network into the interpolation process (assuming that the stream network is valid). The stream network represents places where the relative slope is known very well (i.e., water flows from the immediate banks and down the stream). Four authors have created computer programs that require both contours and breaklines (i.e., stream networks and ridgelines) as inputs. Timmins, Huston, and Clapp (1989) used a simple approach that linearly interpolated between contours, contours and streams, or contours and ridgelines. This approach was relatively successful, because for a suitable grid interval it produced a DEM that contained most of the input stream network. However, this simple method cannot perform drainage enforcement, forcing upstream cells always to be at higher elevations than downstream cells. Hutchinson (1989) used an iterative finite-difference interpolation method and introduced the concept of drainage enforcement and minimum roughness to guarantee that the interpolation produced a suitable model. Hutchinson's program is not in the public domain, and this lack of availability has reduced its use, though it may be the best interpolation program available. Recently placed in the public domain is a program, S.SURF.TPS for the GRASS GIS system, that uses splines under tension to perform interpolation but has no provision for drainage enforcement (Mitasova and Mitas 1993a; Mitasova and Mitas 1993b). The future holds promise because ARC/INFO® 7.0 will have a TOPOGRID command that will interpolate DEMs using Hutchinson's algorithm.

The remaining two aspects of current use of DEMs depend on the method of modeling flow in map form (i.e., two dimensions) and on the introduction of time (i.e., three dimensions). In map form it is necessary to generate flow paths and to determine the drainage network, watershed boundaries, flow counts, and so on. Jenson and Dominque (1988) released a series of programs that perform a unique step called conditioning in which pits (i.e., areas that permit the storage of water to take place) are removed from the surface of the model. The product of this process is a DEM that is useful for forcing flow over the land surface, though it ignores that sinkholes and pits exist in the real landform. Once this step is performed, the other programs can model flow over the matrix in the eight directions. These programs were thoroughly tested on real data and shown to produce, for example, either watershed delineation in agreement with hand-drawn boundaries or errors in these boundaries.

The only caveat against most modeling of water in two dimensions is that usually only a single direction of flow is allowed out of each cell. The methods of noted hydrologists partition the flow and permit flow in multiple directions from each cell (Bevin and Kirkby 1979; Moore 1993). This has a significant effect on maps of surface wetness (i.e., overland flow), but its effect on other modeled processes has not been investigated.

One application of single-direction flow methods is water quality modeling. Levine (1992) calculated delivery ratios for nutrient loads on the Ray Roberts watershed in Texas. The DEMs were created from digitized contours and stream networks, and a flow path algorithm was written to accumulate cell delivery ratios as if the drainage network had impedances at each cell that acted in series. The model could predict mass of nutrients and sediment delivered from each cell. This model has been incorporated into the TERRAIN program.

The use of DEMs in three dimensions to produce overland flow maps in response to rainfall as a function of time was pioneered by Beven and Kirkby (1979) and others who were able to use a semidistributed approach to model watershed response over similar areas of the watershed. Beven's program, TOPMODEL, classifies topographically similar areas, on the basis of slope and whether water flow is either convergent or divergent, to model the watershed in a semidistributed fashion . Unlike the Jenson and Dominque programs, water flow is dispersed to more than one cell; then the entire landform is classified using a histogram of upslope drained area, per unit contour, per unit slope, at each cell. This topographic index views flow as a flux that drains an upslope area, converges or diverges at each cell, and is moderated by slopes. Each watershed has a characteristic histogram of topographic index that describes its propensity to generate overland flow. The results can be effective for animations, as papers by Huston and Timmins (1992) and Clapp, Timmins, and Huston (1992) show. In these papers they report how overland flow can be modeled and displayed on a map with hourly time steps.

1.6 The Grid Elevation Model

The grid elevation model is based on the neighborhood relationship of square (or rectangular) cells on a grid. This relationship is known to be anisotropic because the cells in the cardinal directions share edges with a center cell, while the corner cells do not. Using this model, one can hypothesize that water flow will be in one or more of these eight directions from the center cell shown in figure 1.1.

cell - numbering (i)

7	8	1
6	9	2
5	4	3

Jenson and Dominque flow direction codes (D)

64	128	1
32	X	2
16	8	4

Figure 1.1 Cell numbering and Jenson and Dominque flow direction codes for the *d8-method* of waterflow routing.

The most common method of routing water flow is the *d8-method* that routes water to one of these eight neighboring cells. If two (or more) cells are downslope from the center cell (9), flow could be partitioned between them (but is not in the Jenson and Dominque programs or TERRAIN). The Jenson and Dominque flow direction code D may be determined from the cell number to which the water flows:

$$D = 2^{(i-1)}.$$

To determine the cell number to which the water flows, the gradients from the center cell to cells 1 to 8 are calculated from the elevations of the cells (z) and distances between the cell centers (d). The maximum gradient is used to determine the downslope cell number. If two downslope cells have the same maximum gradient, one of them is arbitrarily chosen:

$$\text{downslope cell number} = i\,[1..8]\ \max\left[\frac{z(9)-z(i)}{d(i)}\right], \qquad (1.1)$$

where

$d(2) = d(6) =$ X-spacing on the grid,
$d(4) = d(8) =$ Y-spacing on the grid,
$d(1) = d(3) = d(5) = d(7) =$ (X-spacing^2 * Y-spacing^2)$^{1/2}$,
$z(i) =$ the elevations.

Although Jenson and Dominque's programs (1988) use the *d8-method*, recently Moore (1993) and Fairfield and Leymarie (1991) have shown that the incorporation of flow dispersion into the *d8-method* can produce more realistic drainage networks. In particular, the *d8* quantities flow in 45° increments that can produce very straight channels under certain circumstances. Flow dispersion is incorporated by making flow proportional to multiple downslope neighbor cells according to a fraction f(i):

$$f(i) = \frac{\max\left[0, \text{slope}(i)^v\right]}{\sum_{i=1}^{8} \max\left[0, \text{slope}(i)^v\right]} . \tag{1.2}$$

The exponent v was found to be 1.1 by Freeman (1991). With the exception of this exponent, this algorithm seems to be similar to the dispersion method used by Beven in TOPMODEL. To summarize, the TERRAIN program uses a single flow path model; it gives a somewhat different pattern of surface wetness from the dispersive method used by TOPMODEL (Quinn et al. 1991).

The objective of determining flow is to route water from any interior point of a DEM to its perimeter in a manner similar to the way natural water on the earth's surface flows. If one considers any large continent, it will be apparent that the drainage network routes most water from the interior to the ocean boundaries of the land. Water may be temporarily stored in lakes, but with only a few exceptions (i.e., where there are underground sinkholes or lakes below sea level), water is routed to the perimeter of land masses.

To determine flow directions, it is necessary to simulate the natural routing process without storage (i.e., lakes or underground sinks). Storage requires the introduction of time to this two dimensional model and hence is not permitted. To simulate routing without storage, the DEM is first *conditioned* to remove any depressions. In this process each depression is first checked for a pour point, the lowest elevation on the lip surrounding the depression. Then all elevations within the depression are raised to the pour point. Once flow has been successfully routed from all cells, other algorithms can be written to derive channel networks, watershed boundaries, and so on.

1.7 Objectives of Program TERRAIN

The programs of Jenson and Dominque (1988) were improved in efficiency by combining them into a single program, TERRAIN, with the ability to process very large DEMs. This was accomplished by rewriting all the neighborhood codes to use a disk utility to perform input/output (I/O) and to maintain on request at least three rows of the DEM in memory, in order. This permitted a code to be written that accessed these rows in their correct orientation (designated north, mid, and south). Thus, by processing the DEM in strips, arbitrarily large DEMs can be input.

An objective of this work was to improve the efficiency of the multiprogram conditioning step. Previously, this step required the use of five separate programs to fill depressions in the landform surface prior to the calculation of flow directions. Without this step, water cannot be routed from every interior point to the edges. Because this step was the most time consuming, the subroutines in this step were highly optimized.

Revision of Jenson and Dominque's code (1988) was a nontrivial matter seeing that they used neighborhood operations in loops to attain non-neighborhood goals such as growing regions. For example, a watershed's area is traced upstream from a confluence cell by marking all upstream cells as belonging to the watershed. This operation is applied, first, to cells adjacent to the confluence cell, and then, to all cells adjacent to those cells, and so on. To simulate the outward growth of this process, the algorithm makes multipasses from top and bottom and from left to right. Revisions included reducing disc I/O and the addition of right-to-left processing.

The result of this revision was a tenfold improvement in speed and the aggregation of over ten independent programs into a single 2500-line program; the addition of other functions has since added an additional 1550 lines. A secondary goal, that of improving the ease of use, was accomplished by adding on-line help, arranging the functions in a logical order, and providing a menu for choosing processing options. TERRAIN was tested on a test data set (Jenson and Dominque 1988) and on a large DEM for the Little Tennessee River watershed (i.e., 3600 columns by 1351 rows) against the old code on a Digital Equipment Corp. (DEC) ALPHA 300 workstation. In both cases the new code got the correct answers, and for the Little Tennessee DEM, TERRAIN reduced the cost of the condition step from several hours to 5 min.

1.8 Summary

This chapter gives a brief introduction on the theory of representing landform as a DEM and identifies relevant literature. One should always evaluate the suitability and scale of data with regard to the objectives and acceptable uncertainty for a study. Therefore, this chapter gives the user guidance about some of the issues such as cell size and algorithms that need to be considered when processing DEMs. It explains how flow direction is addressed in TERRAIN. The following chapters walk the reader through DEM acquisition and the steps available in TERRAIN for processing DEMs. The appendices provide instructions for compiling TERRAIN and instructions on some relevant computer operations.

Chapter 2
DEM Acquisition and Creation

DEMs may be acquired from the USGS by binary transfer over the Internet. Two resolutions may be acquired: 3 arc-seconds or 30 m. The 3 arc-second DEMs are not projected and cover a rectangle 2° E-W and 1° N-S. The 30 m DEMs are projected in Universal Transverse Mercator and cover a 7.5′ quadrangle. The coverage, consequently, is a quadrilateral that does not have north-south and east-west boundaries. Neither of these DEMs has drainage enforcement and the vertical sampling interval is also coarse and can be problematic for flat areas. In addition, DEMs are not currently available for all areas of the United States. DEMs as compressed ASCII data files may be downloaded from an anonymous ftp server such as the USGS's at `edcftp.cr.usgs.gov`. Such a session might appear as follows:

```
unix> ftp edcftp.cr.usgs.gov

Connected to dg3.cr.usgs.gov.
220 dg3 FTP server (Version wu-2.4(3) DATE TIME CDT 1994) ready.
Name (edcftp.cr.usgs.gov:UID): anonymous

331 Guest login ok, send your complete e-mail address as password.
Password: me@ornl.gov

230-This is the USGS/EROS Data Center anonymous FTP server.
230-
230 Guest login ok, access restrictions apply.

ftp> cd /pub/data/DEM/250

250-This directory contains the 1:250, 000-scale Digital Elevation Models (DEM).
250-The files have been compressed with the GNU "gzip" utility. See the
250-file README for details about the directory structure, file compression
250-format, and general information about the data set.
250-
250-*** NOTE: Make sure to set the file transfer mode to BINARY.***
250-*** IF you are using MOSAIC turn on the "load to local disk" option***
250- (located under the OPTIONS menu) before downloading files.
250-
250- The DEM files do NOT contain record delimiters. Once you down load the
250- files, you can add delimiters using the following UNIX command:
250-
250- dd if=inputfilename of=outputfilename ibs=4096 cbs=1024 conv=unblock
250-
250-
250 CWD command successful.
```

Because the directories are arranged alphabetically, the user should change the directory to A, B, C, ..., X, Y, Z as appropriate.

If the quad names are not know, the command

```
ftp> dir * dem.list
```

will produce a file called `dem.list` on the host machine. This file will contain a list of the directories and the quads contained in each:

```
ftp> binary
200 Type set to I.
ftp> cd K
250 CWD command successful.
ftp> mget knox*
```

`knox*` is as specific as the user needs to be in naming the quad. Whether or not the user wants to download each file meeting the specified file name, a prompt will be given. In this case, the user would be able to download the files `knoxville-e` and `knoxville-w`. These files will be used in the example cases throughout this document. The user should choose file names carefully when downloading because some quad names may be misspelled (e.g., ap(p)alachicola).

```
ftp> quit
```

The text of the `00README` file can be found in Appendix C.

DEMs may be created from contours in-house using software in GRASS or EARTHVISION® (by Dynamic Graphics). These programs fit splines under tension through the contour lines to produce a grid. A better solution to converting the landform as represented by contours to a grid DEM is to use the software developed by Hutchinson (1989) that provides drainage enforcement. In summary, when working with DEM data, that the user should have a visualization package to verify the integrity of the data before using the TERRAIN program.

Chapter 3
TERRAIN Program Documentation

Though competent use of DEMs for flow modeling requires experience, this documentation is designed to guide the novice user and to warn of potential pitfalls. The novice user is advised to begin learning the program with small data sets (i.e., about 500 × 500 cells) so that computer runs take a short time and the output grids can be easily visualized on X-window screens.

3.1 Setting Up To Run TERRAIN

Set up an alias in the user's `.cshrc` for TERRAIN and whatever GIS and/or visualization packages are to be used for the watershed modeling. The user will then be able to invoke these programs by typing their name. The examples in this document will use TERRAIN and ARC/INFO (NOTE: the following settings are workstation specific, please check with the system adminisrator before making any modifications to the .cshrc file):

`alias terrain '~stq/for/terrain'`	(alias the location of TERRAIN)
`setenv ARCHOME /arcexe61`	(establish environment variable for ARC/INFO's home directory)
`alias arc "setenv LM_LICENSE_FILE $ARCHOME/sysgen/license.dat; \$ARCHOME/programs/arc"`	(establish environment variable for software licensing information and start ARC/INFO)

These changes will be effective the next time that the user logs into the system. They can be made active during the current session by the following command:

unix> **source .cshrc** (or **source .login** on DECs)

Create a working directory containing a binary 2-byte integer DEM file (for this example the directory name `jenson` and file name `knoxville` will be used):

```
unix> mkdir jenson
unix> cd jenson
unix> cp /usr3/public/knoxville.gis ~me/jenson/knoxville.gis
```

The following command removes the header from an ERDAS ".gis" file to *convert* it to the format required by TERRAIN. Various methods for converting DEMs to the ".raw" format are described in Section 3.2:

```
unix> dd if=knoxville.gis of=knoxville.raw skip=1 bs=128
```

Run TERRAIN:

```
unix> terrain
```

If these commands are unfamiliar, consult Appendix A for a basic explanation of common Unix commands.

3.2 Data Preparation

Data preparation consists generally of two or three steps: converting the source elevation information into a grid with a binary file format, optionally aggregating the grid files to form the desired watershed, and preparing an elevation grid file without any headers or compression (i.e., a so-called ".raw" file). Sometimes these steps can be combined or are trivial depending on the source form and the orientation of the grid. A prepared file consists of a grid of binary integers with a north-west origin (row by row). This format is commonly used by GRASS's 2-byte uncompressed files.

Note: File name extensions are designed to convey format always and content sometimes. Thus, the extension ".gis" is an ERDAS (1991) file with a specific format and is not just a file of geographic information systems data.

The most important data preparation step is to aggregate the available grids into a single grid that includes the desired watershed entirely. Available data from USGS comes in quads, and even data prepared in-house from contours may have been created in pieces. Maps should be joined together, and the edges of the watershed should be sketched as a preliminary step. The geographical extent of the watershed can then be estimated. Using this extent as a guide, the data required to construct the watershed can then be determined. If this aggregation step is not performed correctly, then any work performed with TERRAIN will have to be repeated with a data set with the necessary extent.

ARC/INFO GRID can be used to edge match and to aggregate multiple DEMs into a single file. An example of aggregating and clipping USGS 3 arc-second quads with GRID is shown in the following paragraph.

Before running TERRAIN, the grid of elevation values has to be prepared as a binary file of 2-byte integers and placed in a working directory with the extension ".raw". In addition, the data on the grid must be arranged with a particular orientation, stored in rows, ordered from the northwest — like GRASS and ERDAS image files. On DEC computers (and other computers for which the `recl` parameter in the FORTRAN OPEN statement does not allow odd record lengths) the matrix *must* have an *even* number of columns.

The data format of Unix integers is identical except with respect to the order of the bytes themselves (which varies by type of machine). TERRAIN assumes that the data is ordered as if it was produced by a DEC computer. If TERRAIN is to be run on a different architecture, use the fact that DEC machines have a byte order that is opposite to SUN and Hewlett Packard (HP) computers. Thus, files prepared on a SUN or HP must be byte-swapped before they are used with TERRAIN. Bytes can be reordered using the Unix **dd** command with appropriate substitutions for `input.file` and `output.file`:

```
unix> dd if=input.file of=output.file bs=128 conv=swab
```

The prepared data file should be placed in a working directory with the suffix ".raw". All of TERRAIN's processing is done within the working directory with assigned file extensions so that the program knows the contents of each file. Examples of preparing a file are shown in the following subsections:

3.2.1 USGS 1° or 7.5′ ASCII DEM file

The USGS format contains padding bytes, a remnant from when the data was distributed on tape. The simplest way to read this format is using the ARC/INFO GRID command DEMLATTICE. First, however, the user must add record delimiters to the file as specified in the documentation for the files (Appendix C):

```
unix> dd if=knoxville of=knoxville.DEM ibs=4096 cbs=1024
      conv=unblock
Arc>  demlattice knoxville.DEM knoxville usgs
```

This produces an ARC/INFO GRID file which can be further processed.

3.2.2 ARC/INFO GRID files

Note: The second option shown below ensures that the file produced is a 2-byte ".gis" file, but it is being removed in ARC/INFO v7. The GRIDIMAGE command will try to make the smallest ERDAS file possible, which can lead to 1-byte and 4-bit files. In order to force the GRIDIMAGE command to make a 2-byte file, one or more value in the grid has to be greater than 255. TERRAIN reads only the 2-byte format:

```
Arc>  gridimage knoxville none knoxville.gis erdas
      or
Arc>  griderdas knoxville knoxville.gis
```

This produces an ERDAS 2-byte ".gis" file that can be further processed.

3.2.3 ERDAS 2-byte ".gis" file

A suitable intermediary binary file format is the ERDAS ".gis" format, which has headers and thus can retain essential information such as the origin, cell size, and number of columns and rows. This format is input and output by ARC/INFO and is fairly easily read by bypassing the 128-byte header. Unfortunately, this binary file format cannot be directly used by TERRAIN because it is unable to bypass the header. Consequently, the header must be removed:

```
unix> dd if=knoxville.gis of=knoxville.raw bs=128 skip=1
```

3.2.4 ERDAS 2-byte ".gis" file with opposite byte order

The byte swapping and header removal can be done in a single step:

```
unix> dd if=knoxville.gis of=knoxville.raw bs=128 skip=1
      conv=swab
```

3.2.5 GRASS 2 byte integer file

GRASS stores data in the required format (providing it is not compressed) without headers. It is necessary only to copy the data file to the working directory and, if needed, to swap the order of the bytes.

3.2.6 Making the number of columns even using ARC/INFO GRID

a) With the file already in the GRID format, issue the DESCRIBE command on the file:

```
GRID> describe knoxville
```

b) GRID returns a number of statistics about the file. If number of columns is even, then the file is acceptable to all architectures; if not, a column will have to be removed. The values in italics will have to be calculated by the user from the values returned by the DESCRIBE command:

```
GRID> mapextents xmin ymin (xmax - cellsize) ymax
```

If the describe command returned these values,

```
Cell size =      3.000
Xmin =        -303.500
Xmax =        -120.500
Ymin =         126.500
Ymax =         306.500
```

the user would enter:

```
GRID> mapextents -303.5 126.5 -123.5 306.5
GRID> newknox = knoxville
```

c) Continue as above using newknox in place of knoxville when writing the GRID file to an ERDAS file:

```
Arc>  griderdas newknox newknox.gis
unix> dd if=newknox.gis of=newknox.raw bs=128 skip=1
```

3.2.7 Aggregating 1° USGS quads

This example shows the user how to use ARC/INFO to aggregate the two USGS DEM quads, knoxville-e and knoxville-w, downloaded in Section 2.

a) Record delimiters must be added to the files:

```
unix> dd if=knoxville-e of=knoxville-e.DEM ibs=4096 cbs=1024
      conv=unblock
unix> dd if=knoxville-w of=knoxville-w.DEM ibs=4096 cbs=1024
      conv=unblock
```

b) The files are brought into ARC/INFO:

```
Arc>  demlattice knoxville-e.DEM eastknox
Arc>  demlattice knoxville-w.DEM westknox
```

c) Use the procedure MERGE to combine the grids:

```
Grid> allknox = merge(westknox, eastknox)
```

d) Continue as above using `allknox` in place of `knoxville` when writing the GRID file to an ERDAS file:

```
Arc>  griderdas allknox allknox.gis
unix> dd if=allknox.gis of=allknox.raw bs=128 skip=1
```

3.2.8 Clipping a section from an elevation grid

This example shows the user how to select only part of the file `allknox.raw` for processing.

a) Determine the size and extent of the file to select from:

```
Arc>  describe allknox

      Xmin =   -302401.500     Cell Size          3.000
      Xmax =   -295198.500     Number of rows     1201
      Ymin =    125998.500     Number of columns  2401
      Ymax =    129601.500
```

b) Determine a window that would encompass the area to be selected (in this case a 1000 × 1000 square in the middle of the grid starting 700 cells (2100 map units) from the west and 100 cells (300 map units) from the north) and write it to a new grid:

```
Arc>  setwindow -300301.5 126298.5 -297310.5 129298.5
Arc>  knoxville = allknox
```

c) Verify that the new grid is the correct size, and if so, convert it to a ".raw" file (in this case reversing the bytes as well):

```
Arc>  describe knoxville
Arc>  griderdas knoxville knoxville.gis
unix> dd if=knoxville.gis of=knoxville.raw bs=128 skip=1
      conv=swab
```

3.3 TERRAIN: User Instructions

TERRAIN is a simple program with menu choices for each available function. Each function reads an input grid from a file and produces an output grid in a different file. To understand what TERRAIN has done requires visualization of the output files with systems such as ERDAS, PV-WAVE, GRASS, or ARC/INFO GRID.

Initially, TERRAIN prompts the user to provide details about the input grid. The responses for these prompts regarding the data set name, size, location, and cell spacing are stored in a parameter file `params.dat` for future reference. The prompts and example responses for the file `knoxville.raw` are :

```
Enter DEM name (without a .extension): knoxville

Enter size (columns, rows): 1000 1000

Enter xorigin, yorigin:  -300310.5 126298.5

Enter xcell, ycell sizes: 3 3
```

Then the thirteen menu choices controlling each TERRAIN function are presented in a menu:

```
Choose option (or help or quit)
examples: 2 2help quit

Condition DEM       : 1
Direct              : 2
Count               : 3
Delta               : 4
Seed                : 5
Watershed           : 6
Overland            : 7
Network             : 8
Flopath             : 9
Slope               : 10
Aspect              : 11
Log(a/tan B)        : 12
Clip                : 13
```

Limited help is available on-line by typing:

```
help

TERRAIN: a hydrologic processing program
          for digital elevation models.

Further help is available by reading the
program documentation or by choosing a
number from the menu with the word
"help" appended; example: 1help.
```

Help for each step is obtained by appending "help" or "h" to the option number:

```
1help

CONDITION: fills depressions in a DEM.
           It makes it generally convex so each
           cell flows to the edge of the DEM.
           OVERWRITES the dem file.
```

The TERRAIN program will process the DEM in the working directory according to the user's choice provided the appropriate files are available from running previous steps. The DEM name (without the ".raw" extension) and its size (columns and rows) are taken from a file ('params.dat') with these entries, respectively, on the first three lines of the file. The required files for each step (all steps except condition, slope, aspect, and clip *require* the flow direction ".direct" file) are:

Desired Function		Required File(s)
Condition DEM	: 1	Data Preparation of ".raw" file
Direct	: 2	Conditioned ".dem" file
Count	: 3	Flow directions ".direct" file
Delta	: 4	Flow accumulation ".count" file
Seed	: 5	Flow difference ".delta" file
Watershed	: 6	Watershed seeded ".seed" file
Overland	: 7	Manually seeded ".seed" file
Network	: 8	Flow accumulation ".count" file
Flopath	: 9	Delivery ratio ".ratio" file
Slope	: 10	DEM ".dem" file (may be conditioned)
Aspect	: 11	DEM ".dem" file (may be conditioned)
Log(a/tan B)	: 12	DEM ".dem" file (may be conditioned)
Clip	: 13	DEM ".dem" file & ".watershed" file used as mask

Each step lists the time required to perform that procedure on the 1000 × 1000 cell `knoxville.gis` file, described in Subsection 3.2.8, using an HP PA-RISC workstation. Following most steps is a grayshade image of the 1000 × 1000 knoxville grid (figure 3.1) as it appears after that operation. The steps for producing the images are described in Subsection 3.4.1.

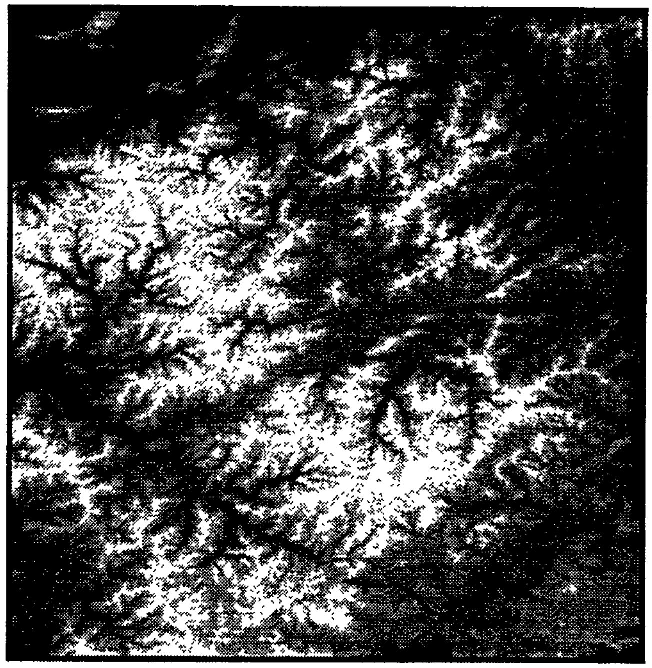


Figure 3.1 Knoxville 1000 × 1000 cell elevation grid.

3.3.1 Condition DEM

Purpose:

Conditioning removes depressions from the DEM so that flow directions can be determined without effects caused by storage in the depressions. The elevations in the depressions are adjusted to the height of the local pour point on the lip of the depression. Be aware that real depressions are filled in during this process.

Requirements:

A file ".raw" of grid data without headers and with the data stored as 2-byte integers by rows starting from the northwest, must have been prepared (Subsection 3.2).

Additional parameters: none..

Time required:

Condition is a multipass procedure and may require more than one run to condition the DEM completely (no more than four runs should be required). The knoxville file required three runs of times 2:02, 4:14, and 0:40.

Help printout:

```
CONDITION: fills depressions in a DEM.
           Makes it generally convex so each
           cell flows to the edge of the DEM.
           OVERWRITES the dem file.
```

Operation:

Condition copies the ".raw" file to the output ".dem" file and then fills the depressions in the ".dem" file (figure 3.2).

Output file:

The output file ".dem" is created by filling depressions according to the Jenson and Dominque (1988) technique so that the flow directions can be determined in the second step. Two temporary output files, ".diraw" and ".wtraw", are produced but are deleted provided the condition step is not interrupted.

Warning messages:

The message

```
COULD NOT SOLVE FOR ALL CELLS
```

signals that the DEM requires additional conditioning. This is not unusual, since cells raised during the first conditioning pass may cause other cells to appear as depressions. The partially conditioned DEM can be further conditioned by renaming it as the ".raw" file and then running **condition** again. Repeating **condition** without renaming the ".dem" file will merely repeat the previous condition step. If the file is not properly conditioned after three to four passes through **condition** the user may want to check if the bytes are ordered correctly. This can be done by using the utility **dd** on the *original* ".raw" file as described earlier.

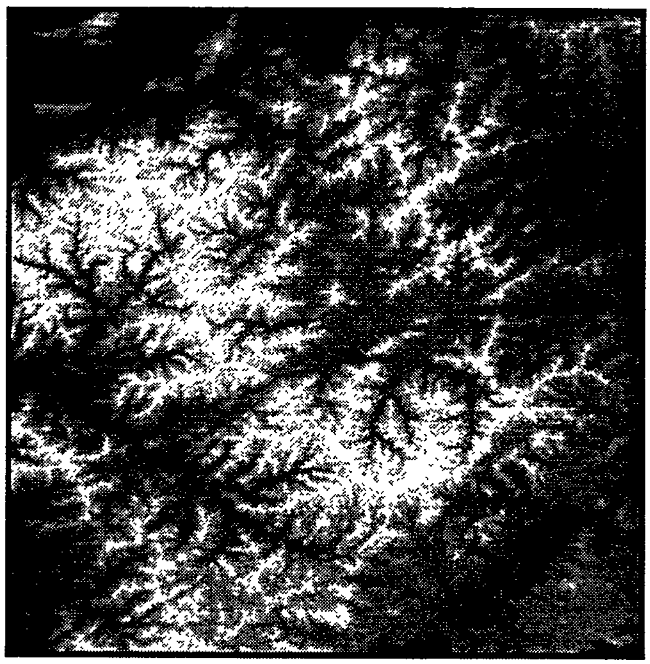


Figure 3.2 Conditioned elevation grid.

3.3.2 Direct

Purpose:

The **direct** step determines the directions water flows out from each cell of the conditioned DEM landform using the *d8*-algorithm of the eight cells surrounding each cell; water is routed to the one that has the maximum downslope gradient. (Gradients are measured using cell sizes for the cardinal cells and the hypotenuse for corner cells.) If two or more cells have the same downslope gradient, the choice is arbitrary. All succeeding steps depend on the successful completion of this step.

Requirements:

Prepare a conditioned ".dem" file by removing the depressions with the condition step.

Additional parameters: none..

Time required:

Direct is a multipass procedure that took 0:24 to process the knoxville file.

Operation:

For each cell, **Direct** chooses the steepest downhill gradient and writes the Jenson direction code (i.e., 1 = NE, 2 = E, 4 = SE, 8 = S, 16 = SW, 32 = W, 64 = NW, 128 = N) into the output grid which is written to a ".direct" file [see Jenson and Dominique (1988) for a description of how direction is resolved when two or more cells have the same and greatest slope].

Help Printout:

```
DIRECT: determines flow direction for each
        cell in a DEM file. Flow is routed
        across flat areas and each cell has
        a flow route to the edges.

Flow codes      64(NW)     128(N)     1(NE)
wrt             32(W)      x          2(E)
center cell.    16(SW)     8(S)       4(SE)
```

Output File:

The output file ".direct" contains the eight-way direction codes as shown above. This file (figure 3.3) is the most important grid that TERRAIN produces because it is used in most of the other steps. It can be visualized as an aspect map providing an appropriate color table is chosen for each discrete **direct** flow direction code. A suggested scheme is red = NE, purple = E, blue = SE, blue-green = S, green = SW, yellow-green = W, yellow = NW, and orange = north (Kimerling and Moellering, 1989). With gray shades, the tones should start from near white for north and become darker counterclockwise till they reach black in the northeast.

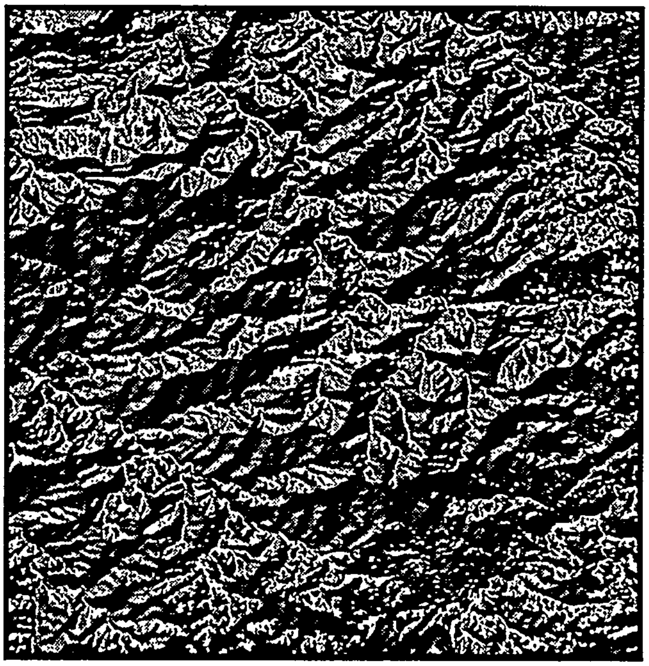


Figure 3.3 Direction grid.

3.3.3 Count

Purpose:

The flow **count** step accumulates the upstream flow to any cell by counting all the upstream cells that drain to it. Because the count may exceed 32,767 cells, 4-byte integers are used (which permit a maximum of more than 2 billion cells to be counted). The counts represent unit areas and can be converted to drained areas by multiplication by the cell area.

Requirements:

The **direct** step must have been completed successfully to create the flow direction file ".direct".

Additional parameters: none.

Time required:

Count is a multipass procedure that took 1:12 to process the knoxville file.

Help Printout:

```
COUNT: measure flow accumulation by counting
       the number of cells which flow to
       each cell. Thresholding the COUNT
       dataset produces the drainage network.
```

Output File:

The output file ".count" contains the accumulated flows from upstream cells stored as 4-byte integers. This file (figure 3.4) can be visualized best by taking logarithms, since the values have such a wide range. A suitable transformation schemes is:

$$output = \log[(max\ count) * (count + 0.001)]$$

where the value "0.001" is added to avoid taking the logarithms of zero.

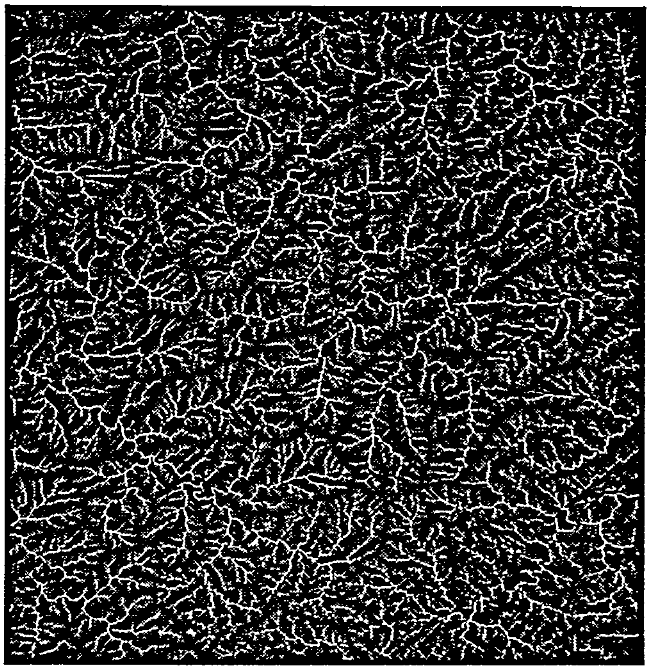


Figure 3.4 Count values grid.

3.3.4 Delta

Purpose:

The **delta** step computes the difference in accumulation value between each cell and its downstream cell:

delta value = accumulation value of downstream cell
- accumulation value of current cell

These values are positive except on the edges of a clipped watershed where they are negative.

Requirements:

The **count** step must have been completed successfully to create the flow accumulation file, ".count".

Additional parameters: none.

Time required:

Delta is a single-pass procedure that took 0:02 to process the knoxville file.

Help Printout:

```
DELTA: shows increase in flow accumulation
       values as each cell drains to its
       downstream neighbor.
```

Output File:

Delta creates a 4-byte integer file ".delta" containing the difference in accumulation value between each cell and its downstream cell. (These values are negative on the edges of a clipped watershed.) The values range from one to almost the entire size of watershed in cells; hence, a 4-byte integer is used. The output file can be visualized in a similar fashion as the count file (figure 3.5) by taking logarithms, since the values have a wide range. After removal of any negative values, a suitable transformation schemes is:

$$output = \log[(\max count) * (count + 0.001)]$$

where the value "0.001" is added to avoid taking the logarithms of zero.

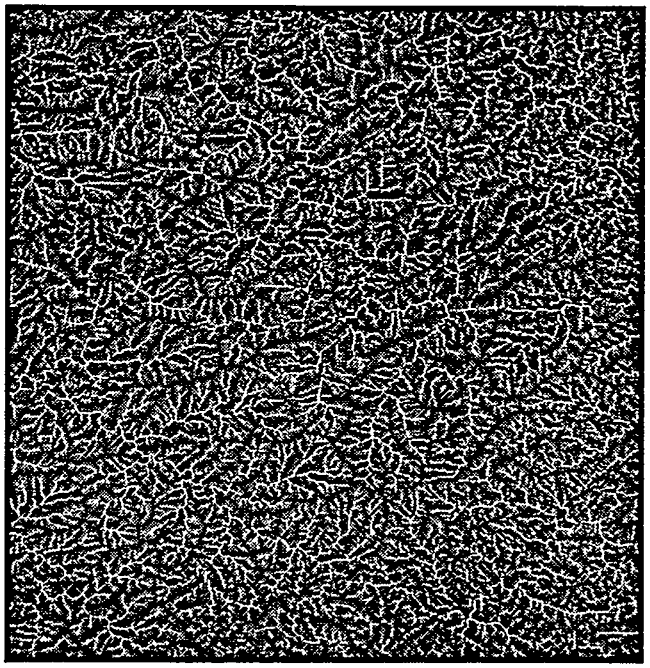


Figure 3.5 Delta values grid.

3.3.5 Seed

Purpose:

The **seed** step locates confluence points (cells) that meet or exceed a threshold value determined in the **delta** procedure. The resulting file is used for region growing to find the extent of each watershed in the **watershed** step.

Requirements:

The **delta** step must have been completed successfully to create the flow accumulation difference file ".delta". Both the ".direct" and the ".count" files are also input.

Additional parameters:

After invoking **seed**, the user must specify an area threshold (in cells) and decide whether to seed the edges of the grid. The area threshold is the minimum allowed size of the watershed. The significance of seeding the edges is not understood.

Example:

```
Enter area threshold, & whether to seed edges (T or F):

25000 F
```

Time required:

Seed is a single-pass procedure that took 0:01 to process the knoxville file.

Help Printout:

```
SEED: automatically seeds every major tributary
      where the area draining to the tributary
      exceeds the threshold. Seeds (1, 2, 3..)
      to build watersheds are placed in a
      are placed in background of -1.
```

Output :

Seed prints out the number of basins seeded. If the number is one or more, it creates a watershed file ".watershed" with certain cells seeded with unique watershed IDs (1, 2, 3, 4...). This file has no value by itself until it is updated to form entire watersheds by the **watershed** step.

Warning Messages:

If count threshold is too large, the following message appears:

```
WARNING, NO BASINS WERE SEEDED
```

Repeat SEED and reduce the threshold.

Appendix D lists instructions for manually seeding using GRID.

3.3.6 Watershed

Purpose:

The **watershed** step grows a region upstream from a confluence cell (which has been marked with a positive integer seed) to define the full extent of the watershed above that cell.

Requirements:

A correctly seeded watershed file ".watershed" must have been previously created either manually or automatically. If the user created it manually they may wish to keep a copy because **watershed** overwrites this input file.

Additional parameters: none.

Time required:

Watershed is a multipass procedure that took 3:05 to process the knoxville file.

Help Printout:

```
WATERSHED: builds watershed polygons from a dataset
           which has been seeded at the confluence
           cell (base of desired watershed).
           These positive seeds lie in a sea of -1s
           OVERWRITES this dataset.
```

Operation:

Watershed overwrites ".watershed" using neighborhood operations to extend the watershed(s) upstream from the seed cell(s).

Output File:

Watershed completes a watershed file ".watershed" by extending each uniquely seeded watershed seed cell to the far extent of its respective watershed (figure 3.6).

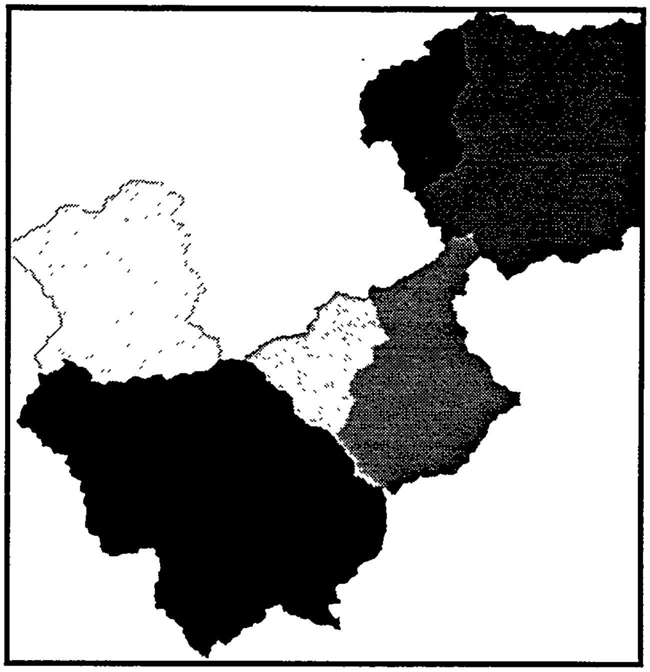


Figure 3.6 Watersheds.

3.3.7 Overland

Purpose:

> The **overland** step is analogous to the **watershed** step which traces all flow paths upstream to make a watershed. By tracing the path of pollutants downstream from one or more initial cells, **overland** defines a single downstream path for the seed cell(s).

Requirements:

> A correctly seeded file ".overland" must have been previously manually created. The file must consist entirely of -1's except for the cell(s) from which to start the trace from. Each trace cell must be uniquely identified. The creation of the ".overland" file is similar to the manual seeding of watersheds described in Appendix D.

Additional parameters: none.

Time required:

> **Overland** is a multipass procedure that took 1:33 to process the knoxville file.

Help Printout:

```
OVERLAND: traces overland paths for specific cells
          such as pollution sources, marked with
          a positive seed in background of -1s.
          OVERWRITES this seeded dataset.
```

Output File:

> **Overland** creates a file (figure 3.7) containing a grid with the overland path (for example, of a potential pollutant) from an upland seed cell to the nearest stream and thus down the stream to the confluence of the watershed.

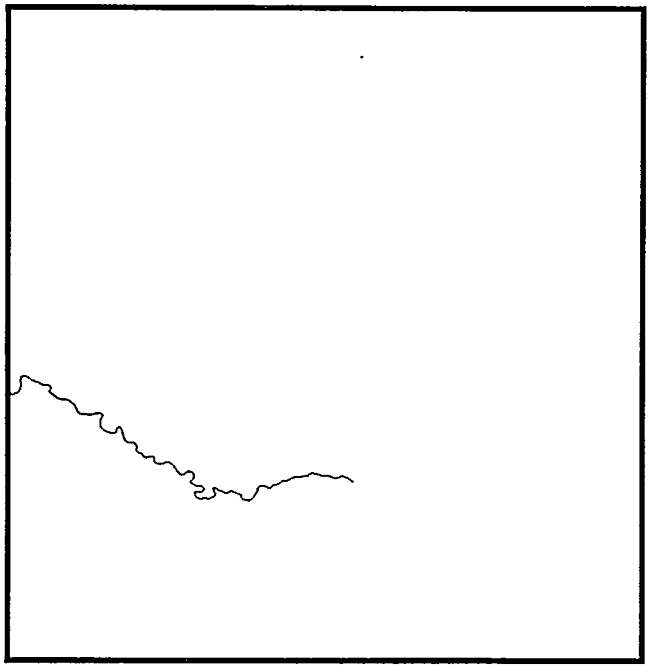


Figure 3.7 Overland path for the cell 544, 282.

3.3.8 Network

Purpose:

The **network** step determines the stream network from flow accumulation values that exceed a user specified threshold. For example, if 500 is the threshold value then a stream network is defined using all cells that have 501 or more cells flowing into them.

Requirements:

A flow accumulation file ".count" is required.

Additional parameters:

The onset value to begin a stream must be given in cells.

Example:

```
Enter flow threshold to begin a stream:

500
```

Time required:

Network is a single-pass procedure that took 0:01 to process the knoxville file.

Operation:

For each cell, **network** checks whether it exceeds the flow threshold and writes a 1 in the output grid if it does.

Help Printout:

```
NETWORK: converts a flow accumulation (count)
         file to a stream network.
```

Output File:

The output file ".network" represents the stream network with 1's against a background of zeros (figure 3.8). A simple black-on-white palette (0 = black, 1 = white) will reveal the stream network.

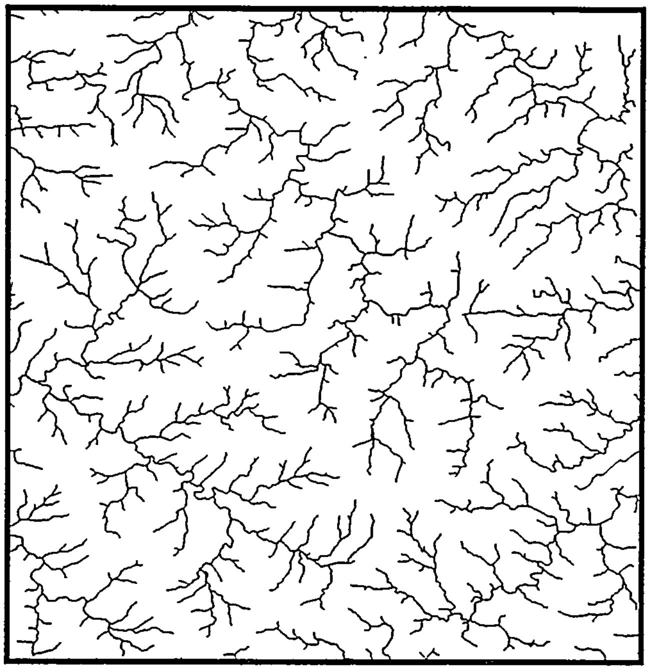


Figure 3.8 Network pathways for a threshold of 500.

3.3.9 Flopath

Purpose:

For measuring nutrient (or any non-point source pollutant) load delivered to each cell, it is necessary to construct a grid of total delivery ratios that can be multiplied (cell by cell) by a grid of potential nutrient load. The **flopath** step calculates total flow path delivery ratios for a stream network by three methods that cumulatively calculate the fraction transmitted to each cell.

Requirements:

Both a flow direction file ".direct" and a cell delivery ratio file ".ratio" (this file is *not* created by TERRAIN) must have been prepared. Please consult published data (Levine et al. 1993) for more information on the **flopath** process, including the nature of the ".ratio" file.

Calculating Delivery Ratios:

Delivery ratios are numbers that, starting from upstream points on overland flow paths, are to be cumulatively operated on to produce a total delivery ratio at each cell. Thus, delivery ratios are the inverse of impedances (Levine and Jones 1990). Because the delivery ratios can be combined to produce total delivery ratios in three different ways, the delivery ratios must be calculated correctly for the method to be used.

Although the **flopath** computation is performed with real numbers, the delivery ratio file must be 2-byte integers (in the range 0 - 32767). Since the values are integers, the delivery ratios (which represent real numbers) may have to be normalized *before* being converted to integers. For example, the real numbers 0.1, 0.2, 0.3, ... 1.0 could be converted to the integers 1, 2, 3, ... 10 by multiplication by 10. Because **flopath** does actually calculate delivery ratios as a function of path length, the user may wish to do a preliminary run with a matrix of integer 1's to determine the longest flowpath in their watershed. If the delivery ratios are assumed to go virtually to zero away from the streams, this test matrix could instead be a matrix of zeros with 1's buffering the streams. Use of a test matrix to determine effective flow path length can avoid problems with the total delivery ratio under flowing (to zero) at cells high in the watershed.

Additional parameters:

Like **watershed**, **flopath** needs to know the location of a seed cell (at a confluence) from which to trace upstream all flow paths in the watershed.

Example:

```
Number of cols= 564 Number of rows= 388
Enter seed cell column and row:

852, 176
```

Also the user must specify the method to combine the input delivery ratios. This can be done by addition of reciprocals (as in Ohm's Law for electric resistances in parallel), addition, or multiplication:

```
Choose method, Sum of 1/Ratio (1) or * (2) or + (3):

2
```

A numerator is required to multiply the reciprocals for method 1:

```
Enter numerator for sum 1/R:

12
```

A scalar (less than 1) is required to normalize the results for methods 2 and 3:

```
Enter data scalar (.01):

.15
```

At present there is little experience to guide the user to choose these scalars.

Time required:

Flopath is a multipass procedure which took 3:39 to process the knoxville file.

Help Printout:

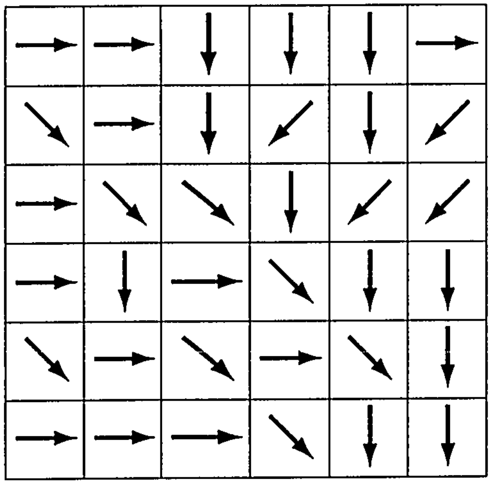
```
FLOPATH: FLOPATH computes the total flow path
         delivery ratio from the direction
         and cell delivery ratio files
```

Operation:

Flopath builds a tree describing the entire drainage network above the seed cell and then recursively goes up each limb computing total delivery ratios until the entire tree is processed.

Output File:

The output file ".flopath" contains the total delivery ratios scaled as 2-byte integers (figures 3.9 and 3.10).



A. Direction of flow

.75	.75	.75	.75	.75	.75
.60	.65	.80	.80	.80	.80
.60	.90	.95	.90	.85	.80
.60	.90	.90	1.0	.80	.80
.60	.90	.90	.90	1.0	.95
.60	.60	.90	.90	1.0	.95

B. Cell delivery ratio

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	0	0	0

C. Watershed seed

Figure 3.9 The flopath procedure, part 1. A. Direction of flow from each cell is calculated based on the digital elevation model. B. Cell delivery ratios are calculated by the user based on factors such as soil, slope, and land use. C. Watershed seed cells are determined based on the location of water quality sampling sites.

.32	.43	.57	.57	.51	
.48	.49	.76	.76	.68	.68
.48	.81	.95	.90	.85	.64
		.90	1.0	.80	
			.90	1.0	

D. Total flow path delivery ratio for each cell

10	10	10	10	10	
10	10	5	5	5	5
10	5	2	2	5	5
		2	2	5	
		2	1	1	

E. Potential nutrient load for each cell

3.2	4.3	5.7	5.7	5.1	
4.8	4.9	3.8	3.8	3.4	3.4
4.8	4.0	1.9	1.8	4.3	3.2
		1.8	2.0	4.0	
			0.9	1.0	

F. Actual nutrient load delivered from each cell

Figure 3.10 The flopath procedure, part 2. D. Total flow path delivery is calculated by starting at the seed cell (C) and using the flow direction grid (A) to "walk up" the hydrologic flow paths and multiplying (in this example) cell delivery ratios (B) together along the way. E. Potential nutrient loads are determined by land use and soil types using export coefficients from the literature. F. Actual delivered nutrient and sediment loads from each cell are calculated by multiplying the total flow path delivery ratio grid (D) with the potential nutrient on sediment load grid (E).

3.3.10 Slope

Purpose:

The **slope** step determines the maximum slope at each cell.

Requirements:

Prepare a conditioned ".dem" file by removing the depressions with the condition step.

Additional parameters:

Slope allows the user to specify an angle of correction if the grid is not already aligned to true north. This operation also requests the cell dimensions and a number to scale the DEM grid to the same units as the cell size.

```
Enter clockwise rotation angle from grid to true N (0.0):

0.0

Enter cell dimensions (dx, dy):

3 3

Enter number to scale elevations to same units as cell
dimensions (1.):

285
```

Time required:

Slope is a singlepass procedure that took 0:25 to process the knoxville file.

Operation:

For each cell **slope** chooses the steepest downhill gradient and writes the slope in percent times to the output file.

Help Printout:

```
SLOPE: SLOPE computes the local slope using
       a maximum slope algorithm.
```

Output File:

The output file ".slope" contains the slopes in percent with a maximum value of 999 (for 99.9%) (figure 3.11).

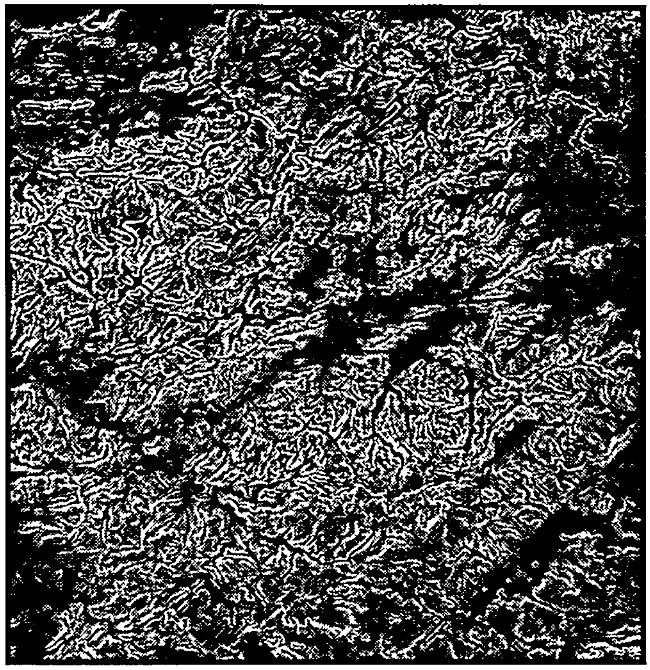


Figure 3.11 Slope.

3.3.11 Aspect

Purpose:

The **aspect** step determines the local aspect direction (i.e., the direction the cell faces) using a maximum slope algorithm.

Requirements:

Prepare a conditioned ".dem" file by removing the depressions with the condition step.

Additional parameters:

The same parameters as in **slope** are used for **aspect.**

Time required:

Aspect is a single-pass procedure that took 0:25 to process the knoxville file.

Operation:

For each cell **aspect** chooses the steepest downhill gradient and writes the azimuth of this direction to the output file.

Help Printout:

```
ASPECT: ASPECT computes the local aspect using
        a maximum slope algorithm.
```

Output File:

The output file ".aspect" contains the aspects in degrees from 1 to 360 (for north) (figure 3.12).

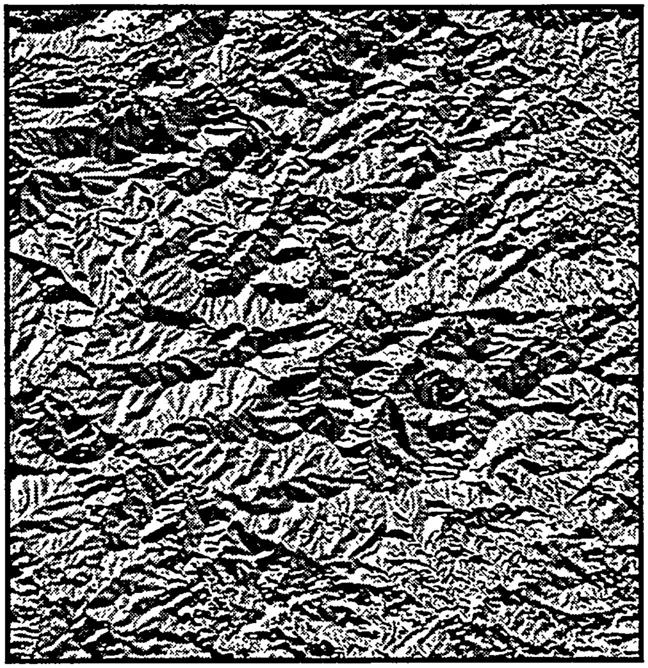


Figure 3.12 Aspect.

3.3.12 Log(a/tan β)

Purpose:

The **log(a/tan β)** step calculates the logarithm of the upslope drained area per unit contour divided by the local slope.

Requirements:

Prepare a conditioned ".dem" file by removing the depressions with the condition step.

Additional parameters:

```
Enter cell dimensions (dx, dy):

3 3

Enter number to scale elevations to same
units as cell dimensions (1):

285
```

Iterations of between 100 and 700 are appropriate for most landscapes; however, processing time increases dramatically for high iterations on large landscapes.

```
Enter number of iterations (1-1000):

200
```

Time required:

Log(a/tan β) is a multipass procedure that took 15:17 to process the knoxville file with 200 iterations.

Operation:

For each cell, log(a/tan β) determines the upslope drained cells and then, allowing flow distribution to multiple cells, divides by the estimated contour length and the local slope.

Help Printout:

```
Log(a/tan B): computes the natural logarithm of
              the upslope area per unit contour
              divided by the local slope.
```

Output File:

The output file ".logatanb" contains the log(a/tan β)'s multiplied by 100 and stored as integers in the range 0 to 32767 (figure 3.13).

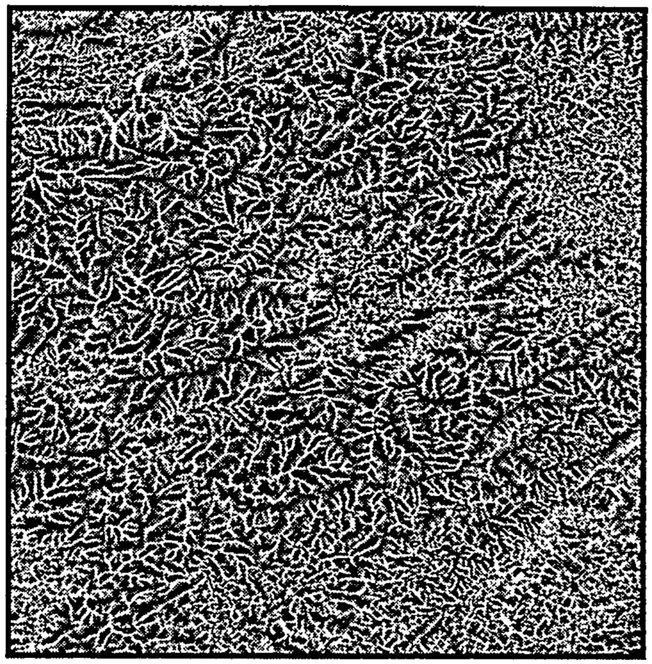


Figure 3.13 Log(a/tan β).

3.4 Other Modeling Steps

3.4.1 Visualizing TERRAIN's output

Each of the files output by TERRAIN can be easily read into GRID by appending a ".gis" header to the beginning of the file and then translating the file into GRID's native format (actually, the ".count" and ".delta" files cannot be visualized in this manner. See Appendix E for information on getting 4-byte files into GRID):

First, the ".gis" header must be extracted from the original ".gis" file

```
unix> dd if=knoxville.gis of=header.gis bs=128 count=1
```

Next, append the ".gis" header to one of the files output by TERRAIN (in this example, the file's bytes are reversed before attaching the header):

```
unix> dd if=knoxville.dem of=knox.dem bs=128 conv=swab
unix> cat gis.header knox.dem > knox_dem.gis
```

Finally, bring the file into GRID.

```
arc>  erdasgrid knox_dem.gis knox_dem
```

Visualize the grid with the following commands (an explanation of these commands can be found in the GRID Command Reference, (ESRI 1991)):

```
grid> shadecolorramp 1 256 white black
grid> mapextent knox_dem
grid> gridshade knox_dem # linear nowrap
```

3.4.2 Clipping a grid with one or more watersheds

After the **watershed** step has completed, the user may wish to clip the elevation grid (or other files) to contain information for only the watershed(s) of interest:

a) Bring the file to be clipped into GRID as described above (e.g. `knoxville.dem` to `knox_dem`).

b) Bring the ".watershed" file into GRID as described above (e.g., `knoxville.watershed` to `knox_water` (the name `knox_watershed` would be too long for ARC/INFO)).

c) Visualize the watershed grid:

```
grid> watersheds = setnull(knox_water eq 0, knox_water)
grid> mapextent watersheds
grid> gridshade watersheds
```

d) Determine the value(s) of the desired watershed(s) using the interactive form of the GRID command CELLVALUE:

```
grid> cellvalue watersheds *
```

Place the cursor over each watershed to be kept and click the mouse button. The value associated with that watershed will be displayed in the text window. Press 9 on the keyboard to end the selection process.

e) For each of the *undesired* watersheds, construct a logical condition for its exclusion using the **setnull** command:

```
grid> newsheds = setnull(watersheds lt 2 or watersheds gt 5
      or watersheds eq 3, watersheds)
```

e) Set this new grid to the logical mask and use it to clip any grid:

```
grid> setmask newsheds
grid> clip_dem = knox_dem
grid> setmask off
```

f) OPTIONAL: In some situations (especially when the clipped area is much smaller than the input), the user may wish to reduce the size of the grid to match the size of the clipped DEM. This can be done by using GRID's SETWINDOW interactively:

```
grid> mapextent clip_dem
grid> gridshade clip_dem
grid> setwindow *
grid> new_dem = clip_dem
```

If the grid is to be processed on a machine requiring an even number of columns, refer to Subsection 3.2.6.

g) Write the clipped dem to a file:

```
Arc>  griderdas new_dem new_dem.gis
unix> dd if=new_dem.gis of=new_dem.raw bs=128 skip=1
```

Of course, this DEM has to be placed in its own working directory, have its own params.dat file, and has to be reprocessed by TERRAIN because it is a different size than the original.

Chapter 4
References

Balce, A. E. 1987. Determination of optimum sampling interval in grid digital elevation models (DEM) data acquisition. Photogramm. Eng. Remote Sens. 53(3):323–30.

Beven, K. J. and M. J. Kirkby. 1979. A physically based variable contributing area model of basin hydrology. Hydrol. Sci. Bull. 24:43–69.

Clapp, R. B., S. P. Timmins, and M. A. Huston. 1992. Visualizing the surface hydrodynamics of a forested watershed. In Mathematical Modeling in Water Resources, Vol 2. Elsevier, N.Y.

ERDAS 1991. Field Guide. ERDAS, Atlanta.

Environmental Systems Research Institute. 1991. GRID Command Reference. ESRI, Redlands, Calif.

Fairfield, J. and P. Leymarie. 1991. Drainage networks from grid digital elevation models. Water Resour. Res. 27(5):709–17.

Fourier, Joseph. 1822. Theorie analytique de la chaleur. F. Didot, Paris.

Freeman, G. T. 1991. Calculating catchment area with divergent flow based on a regular grid. Comput. Geosci. 17:413–22.

GRASS. 1991. Geographic Resources Analysis Support System Users Guide. U.S. Army Corps of Engineers, Champaign, Ill.

Hamming, R. W. 1983. Digital Filters. Prentice Hall, Englewood Cliffs, N.J.

Hunsaker, C. T., S. P. Timmins, and D. A. Levine. 1990. Development of methods to analyze landscape processes important to water quality. Oak Ridge National Laboratory.

Huston, M., and S. P. Timmins. 1992. Walker Branch Watershed Model. Videotape 1071, length: 9:04. Oak Ridge National Laboratory.

Hutchinson, M. F. 1989. A new procedure for gridding elevation and stream line data with automatic removal of spurious pits. J. Hydrol. 106:211–32.

Jenson, S. K., and J. O. Dominque. 1988. Extracting topographic structure from digital elevation data for geographic information system analysis. Photogrammetric Eng. Remote Sens.. 54(11):1593–1600.

Kimerling, A. J., and H. Moellering. 1989. The development of digital slope-aspect displays. pp 241–44. Auto-Carto 9: Ninth International Symposium on Computer

Assisted Cartography. Baltimore. Society for Photogrammetry and Remote Sensing [and] American Congress on Surveying and Mapping.

Levine, D. A., and W. W. Jones. 1990. Modeling phosphorus loading to three Indiana reservoirs: A geographic information system approach. Lake Reservoir Manage. 6(1):81–91.

Levine, D. A. 1992. A geographic information system approach to modeling nutrient and sediment transport. Ph D. dissertation. Indiana University.

Levine, D. A., C. T. Hunsaker, S. P. Timmins, and J. J. Beauchamp. 1993. A geographic information system approach to modeling nutrient and sediment transport. ESD 3993. Oak Ridge National Laboratory.

Mitasova, H., and L. Mitas. 1993a. Interpolation by regularized spline with tension: I Theory and implementation. Math. Geology. 25(6)641–55.

Mitasova, H., and L. Mitas. 1993b. Interpolation by regularized spline with tension: II Application to terrain modeling and surface geometry analysis. Math. Geology. 25(6)657-69.

Mitasova, H., W. M. Brown, D. P. Gerdes, I. Kosinovsky, and L. Mitas. 1993. Modeling spatially and temporally distributed phenomena: New methods and tools for open GIS. In Press. In Proceedings of second International Conference/Workshop on Integrating GIS and Environmental Modeling, Breckenridge, Colo. GIS World.

Moore, I. D. 1993. Hydrologic modelling and GIS. In Press. In Proceedings of 2nd International Conference/Workshop on Integrating GIS and Environmental Modeling, Breckenridge, Colo. GIS World.

Moore, I. D., R. B. Grayson, and A. R. Ladson. 1991. Digital terrain modelling: A review of hydrological, geomorphological, and biological applications. Hydrol. Process. 5(1):3–30.

Nyquist, H. 1928. Certain topics in telegraph transmission theory. AIEE Trans. 47:617-28.

Quinn, P., K. Beven, P. Chevallier, and O. Planchon. 1991. The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. Hydrol. Process. 5(1):3–30.

Timmins, S.P., M.A. Huston, and R.B. Clapp. 1989. Generation of digital elevation models from contours and comparison of topographic indices. Poster presented at American Geophysical Union fall meeting, San Francisco.

U.S. Geological Survey. 1987. Digital Elevation Models, Data Users Guide 5. USGS, Reston, Va.

Appendix A
Unix

It is assumed that the user has some familiarity with Unix. All commands for Unix are in lowercase. The user can get help for any particular command by typing `man` followed by the command. For example:

```
unix> man more
```

which produces:

```
more(1)                                                          more(1)

 NAME
      more, page - file perusal filter for crt viewing

 SYNOPSIS
      more [-n] [-cdflsu] [+linenumber] [+/pattern] [name ...]

      page [-n] [-cdflsu] [+linenumber] [+/pattern] [name ...]

 REMARKS:
      pg is preferred in some standards and has some added functionality,
      but does not support character highlighting (see pg(1)).

 DESCRIPTION
      more is a filter for examining continuous text, one screenful at a
      time, on a soft-copy terminal.  It is quite similar to pg, and is
      retained primarily for backward compatibility.  more normally pauses
      after each screenful, printing --More-- at the bottom of the screen.
      To display one more line, press Return.  To display another screenful,
      press the space bar.  Other possibilities are described later.
```

Pressing the space bar will show additional information on the `more` command.

The common commands the user will require are:

`cd`	change directory
`mkdir`	make directory
`ls`	list directory contents
`mv`	rename a file or directory
`cp`	copy a file
`more`	type to the screen
`lp (or lpr)`	print a file
`rm`	remove a file
`dd`	special file manipulations
`cat`	concatenate file data (must use >> between filenames)

The user should be particularly careful with the remove (`rm`) command because once a file is deleted, it cannot be recovered unless a backup exists. Many problems with `rm` can be avoided by creating the following alias:

```
unix> alias rm 'rm -i file'
```

which prompts before removing file.

Appendix B
Program TERRAIN Compilation Instructions

To compile this FORTRAN 77 program, use the static option for variable storage:

1) HP unix> **f77 -k terrain.f -o terrain**

2) SUN unix> **f77 terrain.f -o terrain**

3) DEC unix> **f77 -static terrain.f -o terrain**

Linking to build an executable (TERRAIN) is automatic under Unix.

Appendix C
Text of the file **00README** from the USGS DEM ftp site

```
This directory contains the 1:250, 000-scale Digital Elevation Models (DEM).

*** NOTE: Make sure to set the file transfer mode to BINARY.          ***
*** If you are using MOSAIC turn on the "load to local disk" option   ***
    (located under the OPTIONS menu) before downloading files.
*** If you have any questions or suggestions, please contact Customer ***

Services at custserv@edcserver1.cr.usgs.gov

DIRECTORY STRUCTURE
Contiguous United States:
Cd to the directory (A-Z) that represents the first character of the
1:250, 000-scale map name you wish to download. Data files under these
directories are named after the maps with the east or west portion
specified.

Alaska
Cd to the Alaska directory then move to the directory (A-Z) that represents
the first character of the 1:250, 000-scale map name you wish to download.
Data files under these directories are named after the maps with the 1-degree
portion specified.

RECORD DELIMITERS:
The DEM files do not contain record delimiters. You can add delimiters by
using the following UNIX command:

dd if=inputfilename of=outputfilename ibs=4096 cbs=1024 conv=unblock

FILE COMPRESSION
The files have been compressed with the GNU "gzip" utility. If you
do not have access to gzip, the FTP server will uncompress the file
as you retrieve it. To do this, simply leave off the ".gz" extension
when retrieving the file (NOTE: This option is not available through MOSAIC).
For example, to retrieve the file "aberdeen-w.gz" without compression just use
"get aberdeen-w". Note that the uncompressed files are typically five times
larger than the compressed versions and so will take five times longer to
transmit. The gzip program is available via anonymous FTP at the following
sites:

        prep.ai.mit.edu:/pub/gnu
        wuarchive.wustl.edu:/systems/gnu

The data files are stored on a robotic mass-storage device so there
may be a short (~10 seconds) delay while the media is retrieved.

DATA FORMAT
For a complete explanation of the 1:250, 000-scale DEM format see the USGS
National Mapping Program, Technical Instructions, Data Users Guide 5, "Digital
Elevation Models". This booklet can be ordered, for a small fee, from the
Earth Science Information Center at the following address:

Earth Science Information Center
U.S. Geological Survey
507 National Center
Reston, VA 22092
Tel: 783-648-6045 or 800-USA-MAPS
Fax: 783-648-5548

Information about the DEM data set can also be found through the World Wide
Webb (WWW) at the following URL address:
http://edcwww.cr.usgs.gov/glis/hyper/guide/1_dgr_dem

The Global Land Information System (GLIS) also provides on-line data set
```

documentation.

BACKGROUND INFORMATION
A digital elevation model (DEM) consists of an array of elevations for ground positions that are usually at regularly spaced intervals.

The 1-degree DEM provides coverage in 1- by 1-degree blocks and is available for all of the contiguous United States, Hawaii, and most of Alaska. The basic elevation model is produced by the Defense Mapping Agency (DMA) using cartographic and photographic sources.

The 1-degree DEM consists of a regular array of elevations referenced horizontally on the geographic coordinate (latitude/longitude) system of the World Geodetic System 1984 Datum. Elevation data located on the degree lines (all four sides) correspond with the same profiles on adjoining DEM blocks.

Elevations are in meters relative to mean sea level. Spacing of the elevations along and between each profile is 3 arc-seconds with 1, 201 elevations per profile. The only exception are DEMs in Alaska, where the spacing and number of elevations per profile varies depending on the latitudinal location of the DEM. Latitudes between 50 and 70 degrees north have spacings at 6 arc-seconds with 601 elevations per profile and latitudes greater than 70 degrees north have spacings at 9 arc-seconds with 401 elevations per profile.

Appendix D
Manual Watershed Specification

NOTE: The process described below will become much easier with the release of ARC/INFO 7, because the new GRID will be able to address individual cells.

The *exact* location of the cell to be seeded must be found.

The 4-byte ".count" file must be converted to ASCII before it can be brought into ARC/INFO. This can be done in a variety of ways, including short programs in C or FORTRAN. Here is an example using the visualization package PV-WAVE:

```
wave> count = lonarr (columns, rows)
wave> openr, 1, 'knoxville.count'
wave> readu, 1, count
wave> openw, 2, 'knox.count'
wave> printf, 2, count
wave> close, 1
wave> close, 2
```

An ASCII header must be appended to the beginning of the ASCII file. This can be done in a text editor. Create a small file with the following information (the numbers in parentheses are user options) and use the UNIX CAT utility to join it with the ASCII file:

```
ncols          (number of columns)
nrows          (number of rows)
xllcorner      (x and y coordinates of the
yllcorner      (lower left corner of the grid)
cellsize       (cellsize)
NODATA_value   -9999
```

The file can now be brought into GRID using the **asciigrid** command:

```
arc>  asciigrid knox.count knox_count
```

If the grid is large, Arc will return an warning that VAT was not created for the file. One can be built with the **buildvat** command:

```
arc>  buildvat knox_count
```

Visualize the grid:

```
grid> shadecolorramp 1 256 black white
grid> mapextent knox_count
grid> gridshade knox_count # linear nowrap
```

Cells receiving large amounts of water flow will appear in light gray and white. Using a map of the areas, zoom in on the desired watershed confluence by setting the **mapextent** and then redisplaying the grid:

```
grid> mapextent *
grid> gridshade knox_count # linear nowrap
```

This process can be repeated multiple times to zoom in closer. Once the area around the desired confluence is found, the cell with the greatest influx of flow can be determined as follows:

```
grid> setwindow *
```

Select the entire graphics window making sure not to exceed the grid boundaries:

```
grid> tempgrid = knox_count
grid> list tempgrid.vat
```

This produces a listing of the count values for the cells in the selected window from lowest to highest.

Select only the cell with the highest value (or any value the user chooses):

```
grid> seed_grid = con(tempgrid eq [max count value], 1, -1)
```

Create a grid consisting entirely of -1's:

```
grid> setwindow knoxville
grid> minus1 = knoxville * 0 - 1
```

Insert the grid containing the confluence cell to the grid of -1's:

```
Arc>  gridinsert seed_grid minus1 knox_seed
```

Verify that the grid knox_seed is the same size as the knoxville grid:

```
Arc> describe knox_seed
```

If it is, write out the grid:

```
Arc>  griderdas knox_seed knox_seed.gis
unix> dd if=knox_seed.gis of=knoxville.watershed bs=128
      skip=1 <conv=swab>
```

If not, the grid can be restricted to the size of the knoxville grid:

```
grid> setwindow knoxville
grid> fixed_seed = knox_seed
```

Write the grid as above, substituting fixed_seed for knox_seed.

Execute TERRAIN and issue the **watershed** command.

It is possible to select multiple seed points if each is numbered differently. Try to select the smallest window possible for each the **seed_grid** and remember to insert each additional **seed_grid** into the **knox_seed** grid, not the **minus1** grid.

Distribution

1. L. D. Bates, K-1001, MS-7169
2. B. A. Berven, 4500S, MS-6124
3. J. H. Cushman, 1503, MS-6352
4. V. H. Dale, 1505, MS-6035
5. D. E. Fowler, 1505, MS-6035
6. R. L. Graham, 1505, MS-6038
7. S. G. Hildebrand, 1505, MS-6037

8–22. C. T. Hunsaker, 1505, MS-6038

23. P. Kanciruk, 1507, MS-6407

24–25. D. A. Levine, 1505, MS-6038

26. J. M. Loar, 1504, MS-6351
27. D. E. Reichle, 4500N, MS-6253
28. M. E. Sale, 1505, MS-6036

29–38. P. M. Schwartz, 1507, MS-6407

39. F. E. Sharples, 1505, MS-6036
40. S. H. Stow, 1505, MS-6038

41–42. S. P. Timmins, 1505, MS-6038

43. Central Research Library

44–58. ESD Library

59–60. Laboratory Records Dept.

61. Laboratory Records, ORNL-RC
62. ORNL Patent Section
63. ORNL Y-12 Technical Library

64. S. Bolton, University of Washington, College of Forest Resources, ARV4,33,10, Seattle, WA 98195
65. M. Broido, Acting Director, Environmental Sciences Division, Department of Energy, 19901 Germantown Road, Germantown, MD 20874
66. K. L. Dickson, Institute of Applied Sciences, P.O. Box 13078, Denton, TX 76203-3078
67. F. A. Donath, Director, Institute for Environmental Education, Geological Society of America, 1006 Las Posas, San Clemente, CA 92673
68. B. Engel, School of Agricultural Engineering, Purdue University, West Lafayette, IN 47907
69. R. N. Farvolden, Professor, Department of Earth Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
70. D. W. Freckman, Director, College of Natural Resources, 101 Natural Resources Building, Colorado State University, Fort Collins, CO 80523
71. C. A. Johnston, Natural Resources Research Institute, University of Minnesota, Duluth, MN 55811
72. G. Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
73. D. Maidment, Balcomes Research Center, University of Texas at Austin, Austin, TX 78712

74-75. L. Mertes, Geography Department, University of California, Santa Barbara, CA 93108

76. R. J. Naiman, Center for Streamside Studies, AR-10, University of Washington, Seattle, WA 98195
77. A. Patrinos, Director, Environmental Sciences Division, Office of Health and Environmental Research, ER- 74, U.S. Department of Energy, Washington, DC 20585
78. S. Paulsen, U.S. EPA/ERL, 200 SW 35th Street, Corvallis, OR 97333
79. J. C. Randolph, SPEA, Indiana University, Bloomington, IN 47405
80. G. S. Sayler, Professor, 10515 Research Drive, Suite 100, The University of Tennessee, Knoxville, TN 37932-2567
81. D. E. Weller, Smithsonian Environmental Research Center, Box 28, Edgewater, MD 21037
82. F. J. Wobber, Environmental Sciences Division, Office of Health and Environmental Research, ER-74, U.S. Department of Energy, Washington,DC 20585
83. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy Oak Ridge Operations, P.O. Box 2001, Oak Ridge, TN 37831-8600

84-85. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831