LandMapper
Environmental Solutions

LandMapR Software Toolkit- C++ Version (2003)

Users Manual

R. A. MacMillan Ph.D., P.Ag.
LandMapper Environmental Solutions Inc.
Edmonton, Alberta

October, 2003

HOW TO CITE THIS DOCUMENT

This document may be cited as:

MacMillan, R. A. 2003. LandMapR© Software Toolkit- C++ Version: Users manual. LandMapper Environmental Solutions Inc., Edmonton, AB. 110 pp.

Copies of the publication may be requested from:

Attn: Dr. R. A. MacMillan
LandMapper Environmental Solutions Inc.
7415 118 St Street NW,
Edmonton, Alberta T6G 1V4

Cat. No. N/A
ISBN N/A

DISCLAIMER

The LandMapR© suite of programs is not commercial software and is NOT OFFERED FOR SALE. The programs are owned completely and solely by LandMapper Environmental Solutions Inc. and are viewed as an in-house toolkit that is used to provide services and products to clients of LandMapper.

Despite this restriction, the LandMapR© toolkit has been provided to a limited number of researchers in academic and public service organizations on an academic license basis. The software is provided to these researchers for the sole purpose of conducting academic research to evaluate the potential of the software to address issues of research interest. Academic researchers are provided with the software because it is in the interests of LandMapper to have interested and knowledgeable individuals apply and evaluate the software and provide their suggestions for improvements and their ideas for new applications. LandMapper also wishes to continue to provide the latest programs and support for the programs to those agencies that assisted with the initial development, evaluation and refinement of the programs.

Persons provided with the software on the basis of an academic license are strictly prohibited from applying the software for commercial purposes or monetary compensation. Academic license users are required to complete and submit annually a list of all locations where the software has been applied. The list should include an identifier for each area processed, an indication of the dimensions of the area (length and width), the grid dimensions of the DEM for each area (m) and the purpose for which the DEM was processed (see Appendix X for an example of the required reporting format).

The LandMapR© suite of programs is not guaranteed to be free of faults or bugs. In fact, several bugs are known to exist in the programs and these may restrict the ability of the programs to successfully process some data sets to completion, particularly large data sets containing many depressions. These known bugs are being identified and rectified as time and resources permit. The LandMapR© programs have been used to successfully process many hundreds of DEM data sets, so the known bugs do not prevent successful application of the programs in most instances.

LandMapper Environmental Solutions Inc. makes no guarantees as to the reliability of the software and accepts no responsibility for any problems that may arise from its use.

ACKNOWLEDGEMENTS

Numerous individuals, agencies and companies are acknowledged for their roles in developing, applying and evaluating the LandMapR toolkit software described in this manual.

The toolkit evolved from an initial set of programs created over the period 1987-1992 for Ph.D. research undertaken by the author under the supervision of Dr. Peter Furley and Dr. Richard Healey at the University of Edinburgh, Scotland. Portions of key pit-removing procedures were developed in 1988 with the assistance of Dr. Willem van Deursen, then at the University of Utrecht and now the principal with Carthago Consultancy, the Netherlands.

The initial Ph.D. programs were revised and re-configured in 1995-96 to automatically classify landform facets from digital elevation data under a contract with Agriculture and Agri-Food Canada initiated and supervised by Dr. W. W. Pettapiece. Several researchers applied and evaluated the original landform classification procedures in their regions. These included G. Manning, R. Eilers and Yann Pelcat in Manitoba, Dr. D. Pennock in Saskatchewan, D. Aspinal in Ontario, Dr. Dan Long at Montana State University and T. Piekutowski and E. Thibault in Quebec. They are thanked for their patience in learning to use a rather non-user friendly version of the program and for useful and generous comments on their experiences with the procedures and their opinion of the results produced by the procedures.

The inspiration, ideas and concepts for the original 15 unit LandMapR landform classification are acknowledged to have been derived directly from the published work of Dan Pennock and others in Saskatchewan (Pennock et al., 1987; 1994). The intent of the original LandMapR programs was initially to apply, improve and extend the original concepts of Pennock and his colleagues.

Contracts with Alberta Agriculture, Food and Rural Development (AAFRD) helped to add further functionality to the initial set of programs and provided the impetus for converting the programs from their original FoxPro platform into a more modern C++ environment. Individuals at AARFD who used and reviewed the original programs or contributed to the extension and re-coding of the programs include T. Martin, T. Goddard, S. Nolan, D. Spiess, Dr. L. Kryzanowski, Dr. L. Hall and Dr. T. Faechner. Contributions by this group of active users are acknowledged with gratitude.

The initial programs were extended once more in order to provide capabilities required to automatically classify ecological classes (Site Series) in British Columbia. Individuals who encouraged the author to modify the programs to map ecological entities include Dr. D. Moon of Core Decision Technologies Inc., Richmond, BC, Keith Jones of R. Keith Jones and Associates Ltd., Victoria, BC and Dr. David McNabb, formerly with Alberta Research Council, Edmonton. R. Coupé and D. Moon helped to define the extra computational capabilities that the program needed to successfully classify ecological entities. The expanded programs were applied and refined in projects funded by Forest Renewal BC (FRBC) and the BC Forest Investment Account (FIA) and managed by Tracy Earle of Lignum Ltd., Al Hicks of Weldwood of Canada Ltd and Earl Spielman of West Fraser Timber Ltd.

Conversion of the original Visual FoxPro programs into C++ was undertaken by Igor Kezhis of GISmo Solutions Ltd., Edmonton, Alberta.

Creation of the LandMapR toolkit was initiated in response to the efforts of Dr. W. W. Pettapiece prior to his retirement from Agriculture and Agri-Food Canada. They represent one final legacy of his efforts to promote the ideals and concepts of soil survey in Alberta and Canada.

INSTALLING THE LANDMAPR© SUITE OF PROGRAMS

Two different versions of the LandMapR© suite of programs currently exist.

The most recent version was written and compiled as a suite of stand-alone C++ programs. This manual applies to and describes operation of the C++ versions of the programs only.

The second, older, version of the programs was written for operation within the Visual FoxPro 97 operating environment. These Visual FoxPro versions of the programs are no longer supported and are not described in this document.

All programs and ancillary files required to run the C++ versions of the LandMapR suite of programs are present on the enclosed CD-ROM. The C++ programs do not modify the registry or install any new DLLs and therefore do not require a setup or install program to install. To install the full suite of 4 LandMapR C++ programs and 1 C++ utility program, simply copy all of the executable programs contained in the LandMapR\Programs\ folder on the distribution CD included with this manual into a program folder that you set up and name yourself.

It is recommended that users set up a folder or directory within which to place all LandMapR programs and related utilities. This folder can be in any location and have any desired name, but a recommended procedure would be to copy all LandMapR C++ executables to a folder named:

  - e.g. C:\Program Files\LandMapR\Programs\

Users who access and use the LandMapR programs frequently will likely find it convenient to place an icon on the desktop to provide a shortcut to each of the 4 LandMapR programs and 1 utility. Accessing and running any of the programs will therefore involve simply double clicking on the appropriate shortcut icon on the desktop.

It is recommended that users also set up a working directory within which to process all data.

- Set up a working directory in which all data is to be processed
  - e.g. D:\LandMapR\Work\

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This manual provides instructions for applying the LandMapR© toolkit of programs. These programs are mainly used to automatically segment landforms into landform elements using digital elevation data as the sole input. The programs can also be used to automatically extract hydrological spatial entities (stream networks, watershed boundaries, WEPP hillslopes) and to apply user-defined classifications of any type, including ecological and soil spatial classifications.

The suite of programs referred to collectively as the LandMapR© toolkit is currently organized into four separate programs. At various times, the individual components have been organized in various configurations ranging from 19 separate modules to a single program that implemented all 19 modules in a sequential fashion.

The current C++ programs still retain and exhibit characteristics of the initial Visual FoxPro database environment under which they were developed. The programs store all input and output data in database tables in DBF format. While the DBF file structure adopted by the LandMapR suite of programs is inefficient in terms of storage volume and access times for reading and writing data, it has the advantage of being a widely used standard for storing data that can be opened and read by many different types of programs, both spatial GIS programs and non spatial data base programs. For this reason, the DBF file structure used in the original Visual FoxPro programs has been retained in the newer C++ versions of the earlier programs.

Database tables are also used to store the rule bases used to define the criteria by which input data are converted into fuzzy output (landform) classifications. Storing rules in database tables makes it possible to define any desired new classes or to change the definitions of existing classes without having to alter the program code. Again, the DBF file structure used to store rule bases for FacetMapR can be read and opened by a large number of existing commercial programs. This means that existing programs such as Microsoft Excel, Access or FoxPro can be used to open and modify rule base files. This meant that there was no need to create a custom user interface for creating new rule tables or modifying existing rule files. LandMapper has tended to use Microsoft Excel as a tool for creating or editing rule files for use in the FacetMapR component of the LandMapR toolkit.

Users will likely find it convenient to have access to a copy of Visual FoxPro in order to be able to access and manipulate the various DBF files produced by the C++ programs or used to define the rule bases. The LandMapR C++ programs no longer absolutely require the presence of Visual FoxPro, but it continues to be both desirable and useful to have access to a copy of this program in order to interact conveniently and effectively with the various DBF files.

This manual was written to provide users with a complete and full description of all inputs required to operate any of the LandMapR programs and of the contents of all output files produced by application of the programs. The manual documents all algorithms used to compute any of the derived or intermediate output products and it explains why each derived output is produced and how it is used. It is hoped that the manual will act as a comprehensive guidebook for all licensed academic and research users of the LandMapR toolkit and will provide them with a reference that they can use to cite the programs in any papers or reports that they prepare in which the LandMapR programs are mentioned. This manual will also be provided to all clients and customers who purchase products or services from LandMapper Environmental Solutions that are produced using the LandMapR toolkit. This should provide clients with full and comprehensive documentation of the methods that LandMapper uses in producing landform classification or predictive ecosystem (PEM) maps.

The LandMapR toolkit is continually evolving in response to recognized improvements and new client needs. This manual reflects the state of development of the LandMapR toolkit as of September, 2003.

BACKGROUND

The LandMapR© toolkit was initially written to assist in developing and testing different concepts for automatically classifying landforms to define management units for precision agriculture and soil survey. It was intended as a convenient and customizable platform for prototyping and evaluating concepts and was never envisaged as commercial software. The toolkit provided a platform for testing, evaluating and revising landform segmentation concepts. Field research conducted to evaluate the utility of the procedures subsequently indicated that the LandMapR© classification was successful in explaining a significant amount of the total variation in stable soil properties and in crop yield at several research sites.

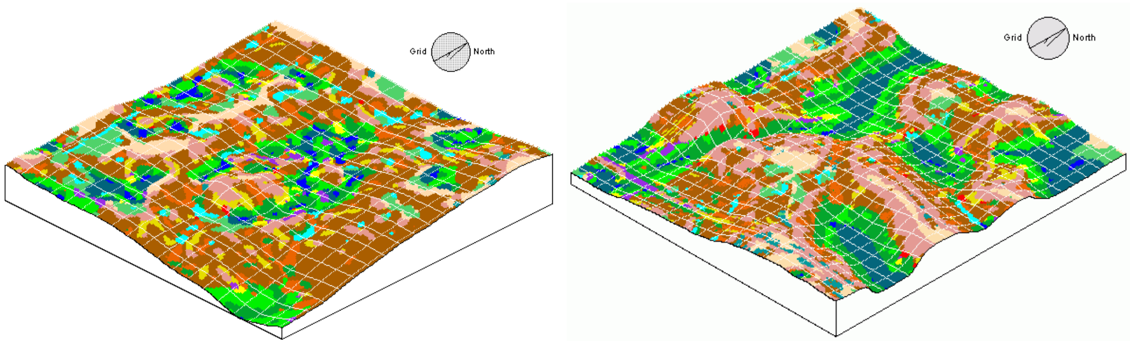


Figure 1. Illustration of the results of applying a LandMapR© classification to two quarter section sized agricultural parcels

All programs were written by Dr. R. A. MacMillan of LandMapper Environmental Solutions Inc. Their original purpose was to support a series of research projects undertaken by Agriculture and Agri-Food Canada (AAFC) in cooperation with Alberta Agriculture, Food and Rural Development (AAFRD) and private sector partners Agrium Inc., Westco Ltd. and Norwest Labs.

The aim of the original research projects was to develop and test a generic set of procedures for automatically segmenting a wide range of agricultural landforms into a consistent and repeating set of landform elements defined using expert judgment and fuzzy logic (MacMillan et al., 2000). The LandMapR programs were initially created because no single existing software product provided all of the integrated capabilities required to apply the computations and concepts which the research projects wished to implement and evaluate.

Underlying Assumptions

The LandMapR classification approach is based on the assumption that topography is the dominant factor in controlling the flow and accumulation of water, energy and matter in landscapes. The movement and accumulation of water in the landscape affects the development and properties of soils and of site environmental conditions in a variety of ways. It affects hill-slope forming processes by influencing the removal and deposition of soil materials by water, wind and mechanical means. It controls the amount and quality of moisture available for vegetative growth and therefore affects in-situ additions of organic carbon. It mediates processes involved in determining soil salinity, calcareousness, pH, leaching, root zone development, in-situ mineral formation and many other soil and site characteristics.

A second basic assumption is that a human devised and human-imposed classification of landforms is superior to one based on statistical analysis and ordination of numerical sample data. Human-devised classifications have the advantage of being generalizeable and flexible. Statistically imposed numerical classifications are designed to recognize classes with the greatest differences in class properties. Human-devised classifications both permit and encourage separation of classes with only subtle, but important, differences. This can be an important advantage for many applications, where subtle differences are important and where a single, universal classification system is required for large areas.

OVERVIEW

The LandMapR© suite of programs processes digital elevation data (DEMs) to automatically extract a variety of user-defined hydrological, ecological and landform spatial entities.

The LandMapR© suite of programs is presently organized as four separate C++ programs that each apply a sub-set of the overall operational procedures created to extract spatial entities.

The four currently available programs are:

- **FlowMapR:** This program processes an input DEM to compute flow topology for simulated surface water flow in both the down-slope and up-slope directions.
    - This program computes flow directions using a very basic implementation of the conventional D8 algorithm for assigning flow from each cell into its lowest down-slope neighbor. It does not utilize any of the more advanced and complex flow direction algorithms (e.g. Rho8) listed by Wilson (1996).
    - This program does possess powerful and unique capabilities for removing pits intelligently and selectively to compute the full topology under which depressions fill, overspill and connect to eventually produce fully integrated surface drainage.
- **FormMapR:** This program uses output from the FlowMapR program to compute a series of terrain derivatives for the input DEM.
    - This program computes common derivatives such as slope, aspect and curvatures, less common derivatives such as diffuse upslope area and compound topographic index (wetness index) and custom derivatives such as absolute and relative relief, absolute and relative slope length.
- **FacetMapR:** This program uses output from the FormMapR program, as well as any other user-supplied data in appropriately formatted DBF tables, to automatically apply fuzzy rules to classify landform or ecological spatial entities through reference to two user-constructed and user-supplied fuzzy rule files.
    - The first fuzzy rule file (ID#arule) identifies which existing input variables are to be used to define "fuzzy attributes" and the fuzzy models and thresholds that are to be used to convert hard input variables (e.g. slope) into fuzzy attribute values (e.g. likelihood of being steep). Any number and kind of available input variables can be used to define any number of desired fuzzy attributes.
    - The second fuzzy rule file (ID#crule) specifies the number, type and characteristics of the fuzzy output classes that the user wishes to define and extract. Users can define any number or type of desired output classes using any reasonable and convenient combination of previously defined fuzzy attributes.
    - The BC Direct-to-Site-Series option within FacetMapR permits users to define and apply different sets of fuzzy rules for different regions or zones within a grid data set of interest. This provides a capability to develop and apply hierarchical classifications with different rules and different output classes for different portions of any area of interest.
- **WeppMapR:** This program uses output from the FlowMapR program to automatically extract and document the spatial entities (channel segments, impoundments and hill-slopes) required for operation of the WEPP erosion and runoff model.
    - This program is an independent off-shoot of the original LandMapR toolkit and currently has no linkages to either of the FormMapR or FacetMapR programs.
    - This program will eventually be extended to automatically extract hydrological spatial entities required to populate the ArcGIS Hydro spatial data model.

FlowMapR

Purpose

The FlowMapR component processes an input DEM to compute flow topology for simulated surface water flow in both the down-slope and up-slope directions

Input Data

FlowMapR operates on input elevation data formatted as a regular (raster) grid. It does not operate on elevation data formatted as irregular x, y, z point data, as a triangular irregular network (TIN) or as vector contours. As the program relies solely on digital elevation data (DEMs) it is critically important that the program be provided with appropriate DEM data.

For DEM data to be suitable for use in the FlowMapR program, it must:

- Be organized as a regular grid of elevation values ordered by row from top (N) to bottom (S) and within rows by column from left (W) to right (E) and stored as a DBF format file.
- Be of a sufficient horizontal and vertical resolution to capture and describe the locations, dimensions and shape of all surface features of interest for a particular landscape at a particular scale.
- Be sufficiently abstracted, or generalized, that landform features of minor dimension do not overwhelm or confuse the classification procedures.

Suitable Format

FlowMapR does not contain any functionality for interpolating x, y, z point data to produce a regular raster grid. Users must therefore either obtain a DEM that is already organized as a regular raster grid or use an appropriate interpolation program to surface irregular x, y,z elevation data to a produce a regular grid.

Raster DEM data is becoming increasingly available from both government and private sector organizations. A considerable amount of coarse spatial resolution (>100 – 1000 m) raster DEM data is available free of charge for large portions of the world, mainly from US government web sites. Many US states now have GIS data clearing houses set up that provide raster DEM data at medium (25 – 100 m) to fine (10 m) grid resolutions for all or portions of a state. Shuttle Radar Topography Mission (SRTM) data have been collected for the entire world and are due to be available by the end of 2003 at 100 m resolution for the entire world and at 25 m resolution for approved users in the US.

Caution: Please be aware that many of the supposedly medium (25-100 m) to fine (5-10 m) spatial resolution raster DEM data sets currently available were created by scanning existing contour maps and running interpolation procedures to interpolate the contours to a raster grid. Many of these supposedly fine spatial resolution raster DEMs contain artifacts and errors that result in them being less than optimal for use in automated landform classification. Raster DEMs interpolated from already smoothed scanned contour lines will not portray any additional detail on topographic variation in areas between contour lines. It is very common for raster DEMs interpolated from contour lines to exhibit a conspicuous stepped pattern characterized by large flat benches separated by steep sharp changes in elevation. This is particularly true when the DEM data are provided in integer format and changes in the elevation values for adjacent grid cells must be at least 1 m in order to record a different elevation.

Users of the FlowMapR program are required to make their own decisions regarding the most appropriate software tools and algorithms to use in cases where there is a need to create a regular raster grid from irregular x, y, z point data. Typical options include using the interpolation procedures available in GIS software programs such as ArcInfo, ArcView (ESRI, 1996) or Idrisi or special purpose interpolation programs such as Surfer or Surface III.

Ultimately, FlowMapR requires elevation data to be read into a field named ELEV in a DBF format file that is always named ID#ELEV. The ELEV field requires that the elevation data be ordered in sequence, by row and column of the original raster input DEM, from top right to bottom left. To obtain the required sequence an original DEM of ***i*** rows and ***j*** columns needs to be converted into a single long string of elevations, one elevation per row, with the first elevation value representing record number 1 (top left) and the last record number ***n*** (where ***n = i*j***). The conversion of an original regular raster grid into a long sequential list of elevations as stored in the file ID#ELEV is illustrated below (Figure 2).

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

721.5	721.8	721.1	720.8	720.2
722.3	722.5	722.9	721.8	721.2
722.4	722.8	723.1	722.3	721.9
722.9	723.3	723.9	723.1	722.5

Record Numbers	Elevations (m)
1	721.5
2	721.8
3	721.1
4	720.8
5	720.2
6	722.3
7	722.5
8	722.9
9	721.8
10	721.2
11	722.4
12	722.8
13	723.1
14	722.3
15	721.9
16	722.9
17	723.3
18	723.9
19	723.1
20	722.5

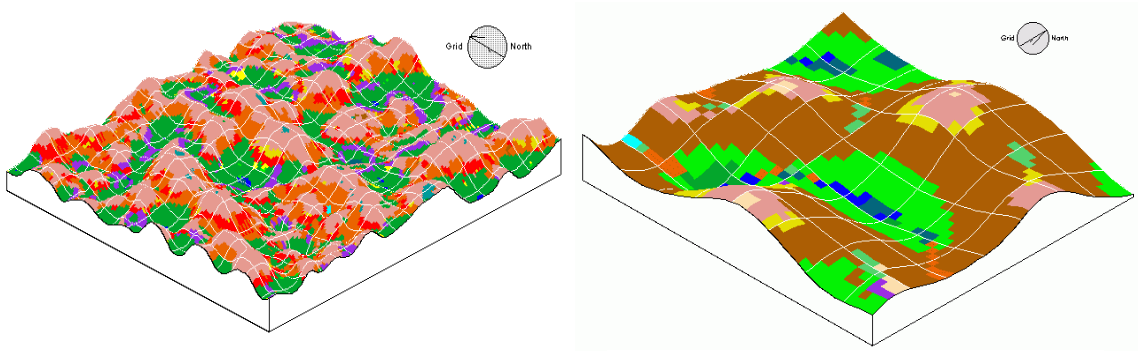
Figure 2. Illustration of reformatting of raster DEM data into a columnar string of elevation values in the DBF file ID#ELEV

FlowMapR was designed to access and read in elevation data from a file named ID#ELEV as illustrated in Figure 2. The most recent C++ software provides a utility (GridReadWriteUtility) to convert DEM data in ArcView binary export format (FLT) into a DBF file formatted as a single columnar string of elevation values in a file named ID#ELEV as illustrated in Figure 2. Other types of file formats that may be supported in future versions of this utility include:

- ArcView ASCII export files (ASC with header information embedded)
- Idrisi binary files (IMG with separate *.DOC header file)
- Surfer binary grid files (GRD with header information embedded)
- Surfer ASCII grid files (GRD with header information embedded)

Suitable Dimensions

It is the responsibility of the user to determine whether the DEM data available for an area of interest has a resolution in both the horizontal and vertical dimensions that is appropriate for capturing and portraying the landscape features of interest at the current scale of interest.



a) Landform classification using a 5 m DEM b) Landform classification using a 25 m DEM

Figure 3. Illustration of the difference in degree of spatial detail for DEMs of different horizontal and vertical resolution

Figure 3 illustrates the significant differences that are apparent when exactly the same location is portrayed using DEMs of different horizontal and vertical spatial resolution. The image on the left was constructed using a DEM with a 5 m horizontal grid resolution and a vertical accuracy of +/- 0.30 m. The image on the right was constructed using a DEM with a 25 m horizontal grid resolution and a vertical accuracy of +/- 10 m. The 5 m DEM (Figure 3a) correctly portrays the location, shape and dimensions of landform features (knobs and kettles) of interest for landform classification at a scale of 1:5,000 to 1:10,000. The 25 m DEM (Figure 3b) indicates that the landscape has a knob and kettle expression but is not able to correctly identify the location of individual knobs or kettles or to correctly describe their shape or dimensions. Terrain derivatives such as slope gradient, curvatures or slope length computed using the 25 m DEM will not provide an accurate representation of the slopes, slope lengths or curvatures associated with the actual landform features of interest at this scale.

Experience to date with application of the LandMapR programs suggests that DEMs with a horizontal grid resolution of 5-10 m and a relative vertical accuracy of +/- 0.5 m or better are most suitable for classifying landforms and extracting hydrological spatial entities (Figure 4). DEMs of coarser horizontal and vertical spatial resolution are suitable for recognizing larger physiographic features such as plains, plateau or valleys (Figures 5 & 6) but are generally not suitable for extracting individual landforms or landform components.

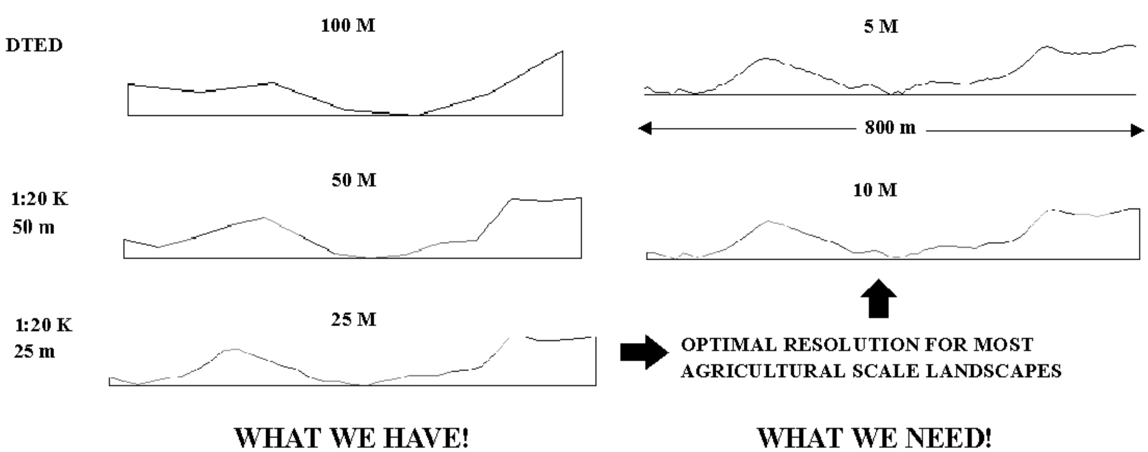


Figure 4. A comparison of the relative level of topographic detail in DEMs of different horizontal grid resolution

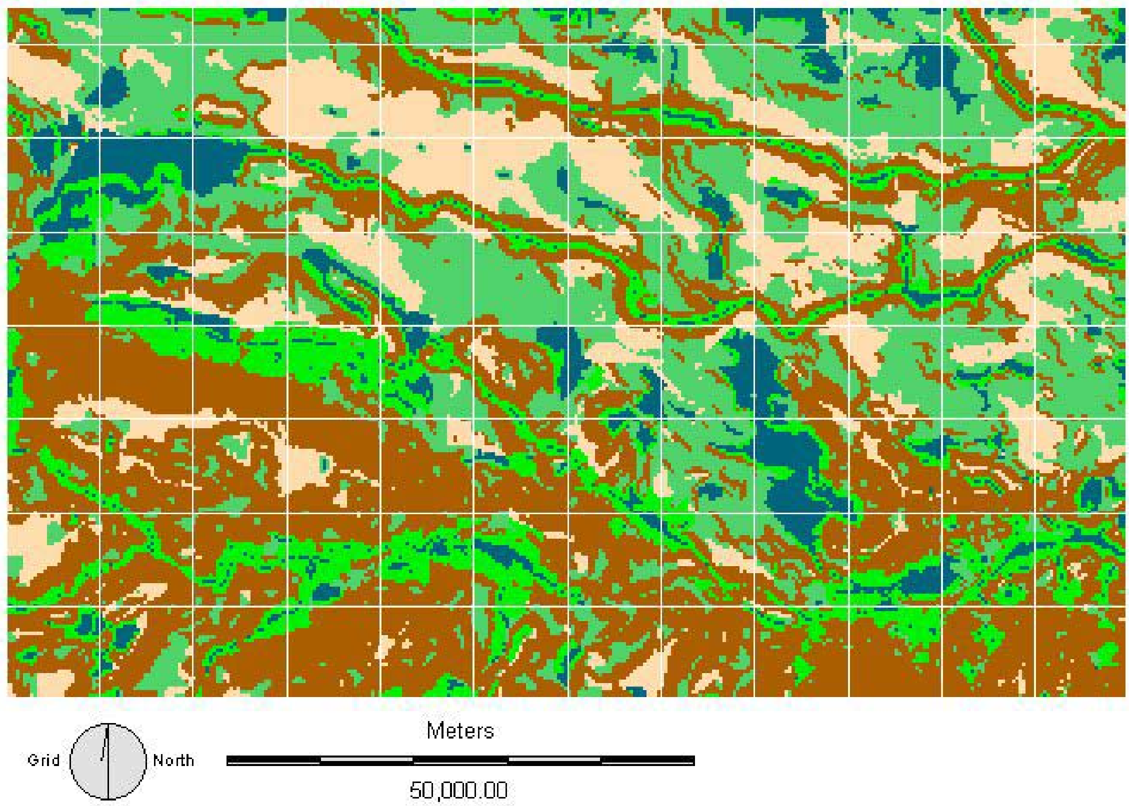


Figure 5. Illustration of how coarser resolution DEM data may be appropriate for recognizing large physiographic features

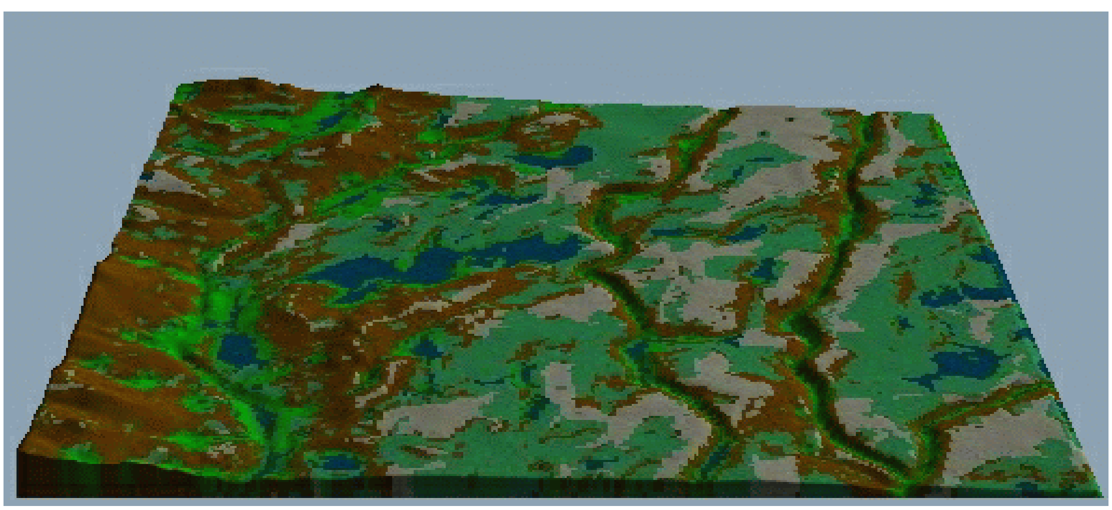


Figure 6. 3D Illustration of recognition of larger physiographic features using coarser resolution DEM data

Figures 5 and 6 illustrate how coarser resolution DEM data can be useful and appropriate. In this case, the desire was to identify and extract larger physiographic features corresponding to hills, plains, plateaus, piedmont foot-slopes, basins and valleys. The 3D image (Figure 6) portrays an area 70 km long and 70 km across looking from east to west. The DEM data used had a grid resolution of 500 m in the horizontal and 20 m in the vertical. If the DEM used here had been of a higher spatial resolution, it would have portrayed more local topographic detail than was necessary and would have obscured recognition of the larger physiographic features of interest.

Suitable Level of Abstraction

Part of the concept of using a DEM with a suitable level of abstraction is related to selection of a DEM with horizontal and vertical resolution appropriate to the size and scale of the current topographic features of interest (see previous section on Suitable Dimensions).

A second part of this equation is the necessity of applying smoothing or filtering operations in order to enhance the longer range signal and reduce shorter range noise. Experience in working with the LandMapR programs has led to the conclusion that virtually all DEM data sets benefit from application of smoothing filters if the DEMs are to be used to produce a LandMapR landform classification. Too much local detail suppresses the longer range signal and makes it difficult to "see the forest for the trees". Consider the following examples.

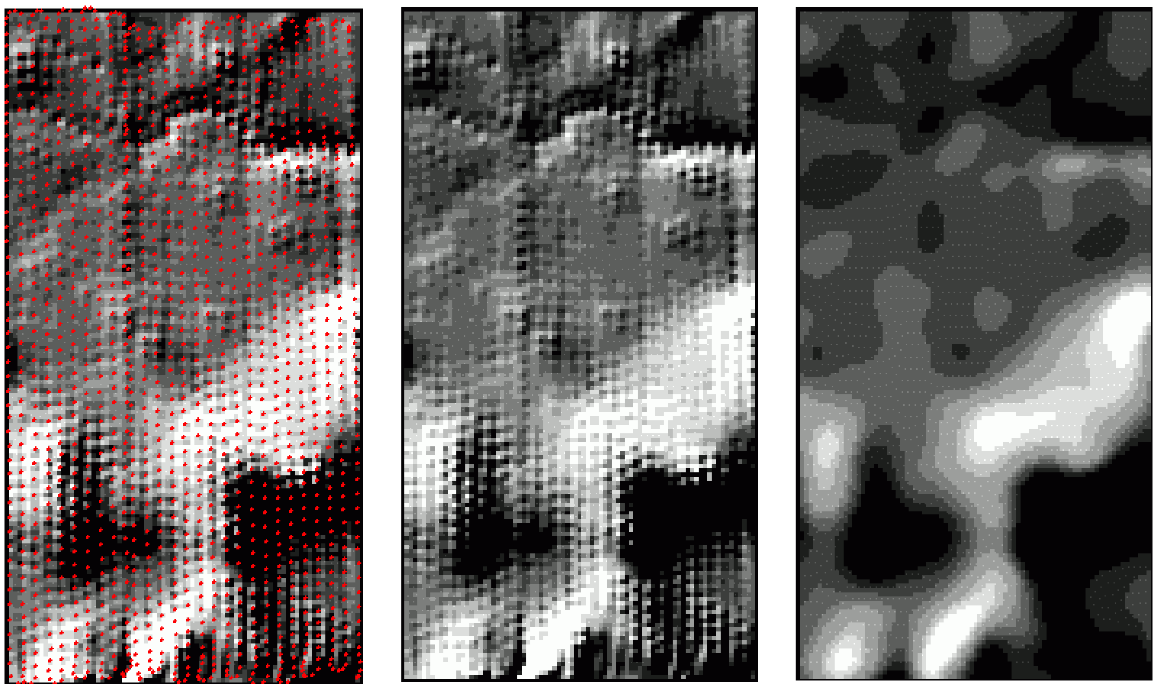


Figure 7. Illustration of local noise arising from inverse weighted interpolation procedures and its reduction by filtering

Figure 7 illustrates a high degree of local noise in an initial DEM arising from a combination of the regular pattern followed in collecting the original x, y, z input data (red dots) and from the inverse weighted distance (IWD) algorithm used to interpolate the x, y, z input data to a regular grid. The centre hillshade illustrates a very blocky, noisy DEM with many local errors. This local noise masks the bigger picture and makes it more difficult to recognize and classify the hillslopes and other landform features of interest. Application of a succession of mean filters (3x3, 3x3, 5x5) to the original noisy raster DEM surface noticeable improves the ability to recognize the larger hillslope features of interest.

Figure 8 shows obvious evidence of artificial patterns in an interpolated DEM that are clearly related to the trajectories along which the DGPS data used to create the DEM were collected. The artificial patterns cut across the grain of the landscape and dissect obvious ridges and draws. Filtering with successive applications of 3x3, 3x3 and 5x5 mean filters was able to reduce the amount of obvious error in this DEM but could never completely remove it. This type of error is encountered very frequently in custom, high resolution DEMs produced using DGPS data and it is invariably reduced by application of a succession of mean filters. Failure to reduce this type of local noise will invariably result in less than optimum landform classification results.

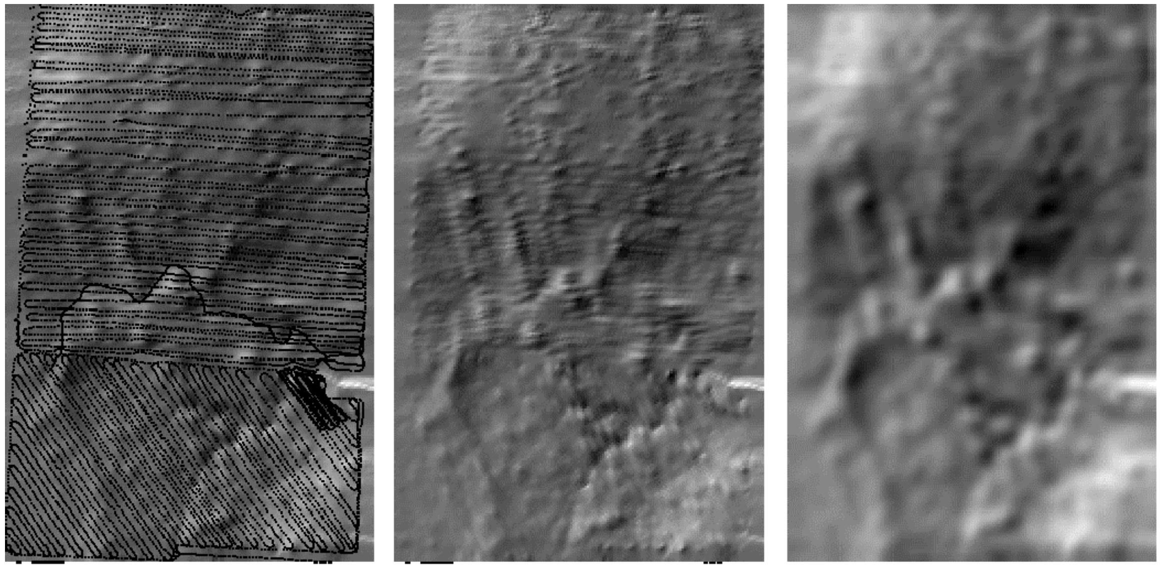


Figure 8. Illustration of clearly artificial patterns in a DEM and their reduction by filtering

High frequency local noise is not always strongly apparent in grayscale hillshade images of DEM data sets, but it may still exist nonetheless. Figure 9 shows a grayscale rendering of profile curvature computed for a DEM for which local noise was not evident in a hillshade rendering. The highly systematic checkerboard pattern evident in the image on the left was apparently a result of utilizing a thin plate spline algorithm to interpolate point data that had been collected using a very regular NS-EW sample spacing. The thin plate spline algorithm captured and amplified the harmonic inherent in the systematic pattern followed in collecting the data. This pattern was far more evident in renderings of second derivatives (curvatures) than it was in hillshade images. The default LandMapR landform classification assigns considerable weight to calculations of profile and plan curvature. Consequently, systematic patterns such as those illustrated in Figure 9 can have dramatic and adverse impacts on landform classifications computed using DEMs that have not been smoothed to reduce this type of systematic noise. The image on the right illustrates how filtering with successive passes of 3x3, 3x3 and 5x5 mean filters virtually eliminated the systematic pattern and improved recognition of legitimate terrain features of interest.

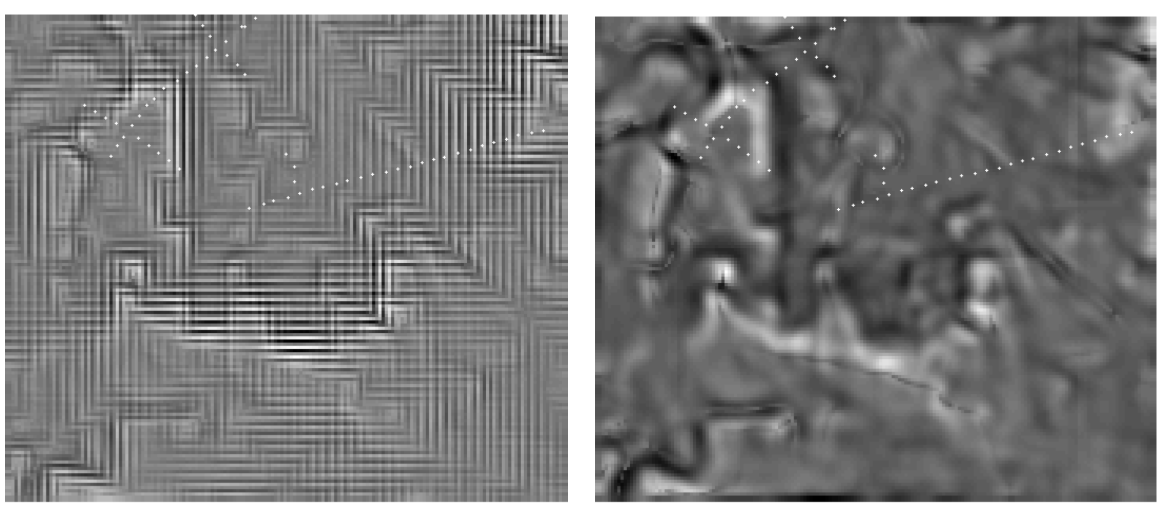


Figure 9. Systematic local noise revealed by gray scale renderings of plan curvature and its reduction by filtering

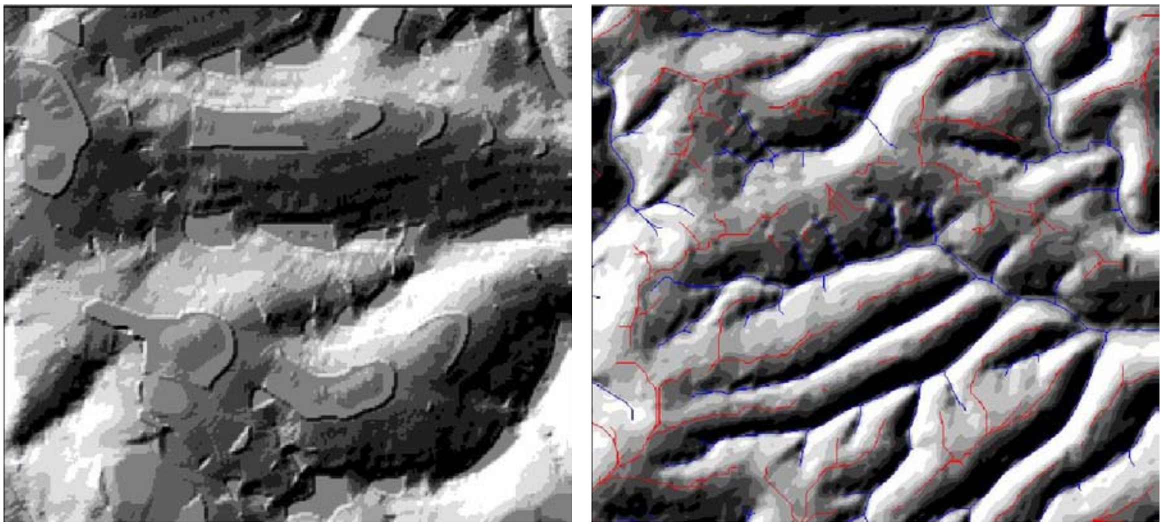


Figure 10. Obvious systematic error in a DEM interpolated from scanned contour data and its reduction by filtering

Figure 10 illustrates some of the strong and obvious systematic errors that are routinely encountered in raster DEMs produced by interpolation from scanned contour data. The image on the left clearly shows an obvious pattern of nearly level steps or terraces separated by sharp and rapid changes in elevation. The locations of the original contour lines are also clearly discernable. The image on the right illustrates a larger portion of the same DEM after application of a sequence of 3x3, 3x3 and 5x5 mean filters that reduced the obvious errors and made the DEM far more suitable for use in automated landform classification.

The message that users should extract from the preceding section is that most original DEM data are highly likely to contain obvious and systematic errors or local noise and that almost all DEMs will benefit from smoothing before being used to classify landforms using the LandMapR programs. In landform classification, original DEMs should not be treated as sacrosanct. Users are advised not to get too preoccupied with maintaining fidelity to absolute values for elevation at specific points. It is more important to produce a DEM that portrays smooth and continuous point to point relationships that reveal the shape and form of terrain features of interest at a specific scale of interest. Relative elevations and point-to-point relationships are more important than maintaining correct absolute elevations if maintaining correct absolute elevations masks or confuses recognition of terrain features of interest.

No consistent set of rules has been discovered to provide guidance for when and how to select and apply appropriate low pass filters to initial DEMs. In general, experience has shown that more filtering is better than less and that no filtering is rarely acceptable. For most landscapes the most effective approach to filtering appears to be to start with one or two passes of a rather small filter (e.g. 3x3) and to follow with a final pass of a larger filter (e.g. 5x5 up to 7x7).

For each new site, it is recommended that the initial raster DEM surface be examined visually by creating gray scale renderings of hillshaded images and second derivatives (slope curvature). If any strong, and clearly artificial, patterns are evident in these images, the initial DEM should be smoothed with a succession of low pass filters until such time as obvious patterns are no longer strongly apparent upon visual inspection. Selection of the most appropriate method for surfacing the data to create an initial raster surface and subsequent application of appropriate smoothing filters involves as much art as science and cannot be strictly specified in advance. It is, however, a critical requirement for achieving successful landform classification results.

Advances in filtering algorithms based on wavelet transforms promise to greatly refine and improve the options available for removing true noise whilst retaining all relevant details in DEMs. Until such time as wavelet filters become widely available and easy to use, we recommend applying conventional low-pass (mean) filters to smooth DEMs and reduce undesirable local noise.

Operation of the FlowMapR Program

Once you have prepared your input DEM data and are satisfied that it has a resolution suited to capturing the terrain features of interest and has had obvious errors and noise removed, actual operation of the FlowMapR program requires very little knowledge or effort. The FlowMapR interface is illustrated in Figure 11.

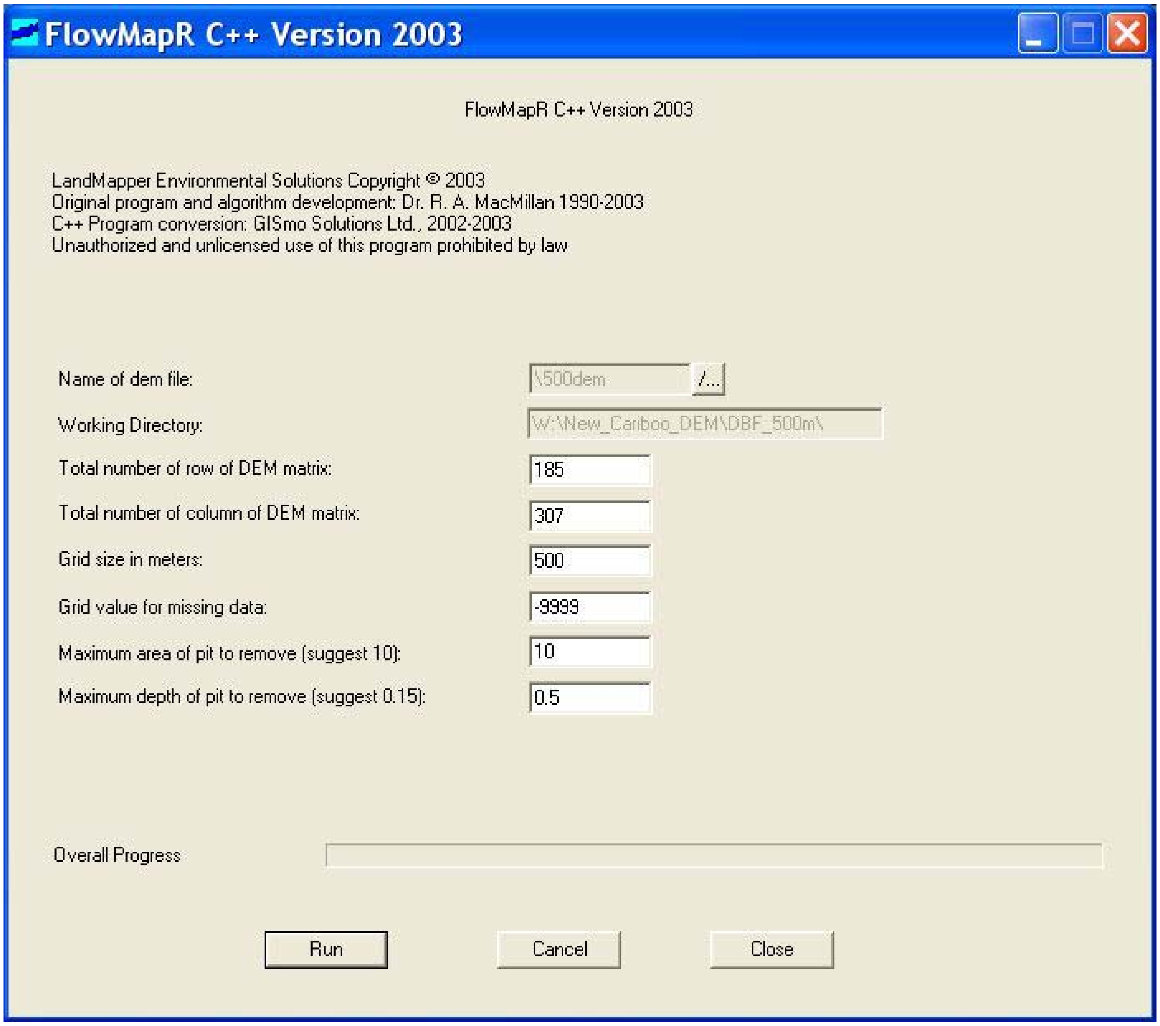


Figure 11. Illustration of the FlowMapR interface

Identification of the source file of DEM data

Users must first identify a source file that contains the elevation data for the area of current interest in a format that can be opened and read by FlowMapR. FlowMapR can currently only open and read DEM data in the default DBF format. The pre-formatted elevation data must exist as a DBF file named ID#ELEV that contains a field named ELEV that stores the elevation data in the required format as a single continuous column of elevation data ordered from top left to bottom right.

Specification of the working directory

The folder containing the source elevation data automatically becomes the default workspace within which FlowMapR will place all resulting output files. This default workspace folder can not currently be changed and must be accepted as the default workspace folder.

Specification of the input grid dimensions and extent

FlowMapR needs to know the number of rows and columns in the input DEM, the horizontal dimensions of the input grid in meters and the value used to identify missing data in the data set (if present). These values must be entered manually and interactively by the user. The FlowMapR program will not run and will not produce correct output if it is given incorrect values for the number of rows and columns, the grid dimensions or for the missing data value.

Specification of threshold values for pit removing

Operation of FlowMapR requires the user to enter two values that control how pits or depressions are treated by the pit removing procedures that are a fundamental feature of FlowMapR. These two values identify the maximum area of a pit (in grid cell units) and the maximum depth of a pit (in meters) above which pits will not be removed by the initial (stage 1) pit removing procedures.

The stage 1 pit removing procedures are designed to completely remove small or spurious pits that represent errors in the DEM or that are at least so small as to not be considered of interest for the current area. Users are advised to select rather small values for these two pit removing criteria. A good rule of thumb is to select a value for pit depth that is no more than ½ of the vertical accuracy of the input DEM. Thus, if the input DEM has a vertical accuracy of +/- 0.3 m then an appropriate threshold value for maximum pit depth might be 0.15 m. This threshold value recognizes that most pits that are less than 0.15 m deep are more likely to be related to error or noise in the input DEM than they are to be true landform features.

Any pits that are less deep than the depth threshold value or occupy fewer cells than the area threshold value will be completely removed by the initial stage 1 pit removing procedures. Complete removal means that all records for these pits are deleted from the pit documentation tables that are produced by the FlowMapR program and all information about these pits is lost. If users wish to obtain information on all possible depressions in a DEM, they only need to select a depth threshold of 0.00 m and an area threshold of 1 cell. This will result in calculation and recording of information on all possible pits in the DEM.

Initial removal of small pits in the stage 1 pit removing procedures saves considerable time and results in faster processing of the input DEM as the initial pit removing procedures are more efficient than the two subsequent pit removing stages. Pits removed in the initial pit removing stage do not have to be considered by the two subsequent pit removing stages that are designed to identify, characterize and then remove larger and more significant pits. Very small pits often constitute as much as 90% of the total number of pits, so initial removal of very small pits can result in a considerable improvement in processing time for this program.

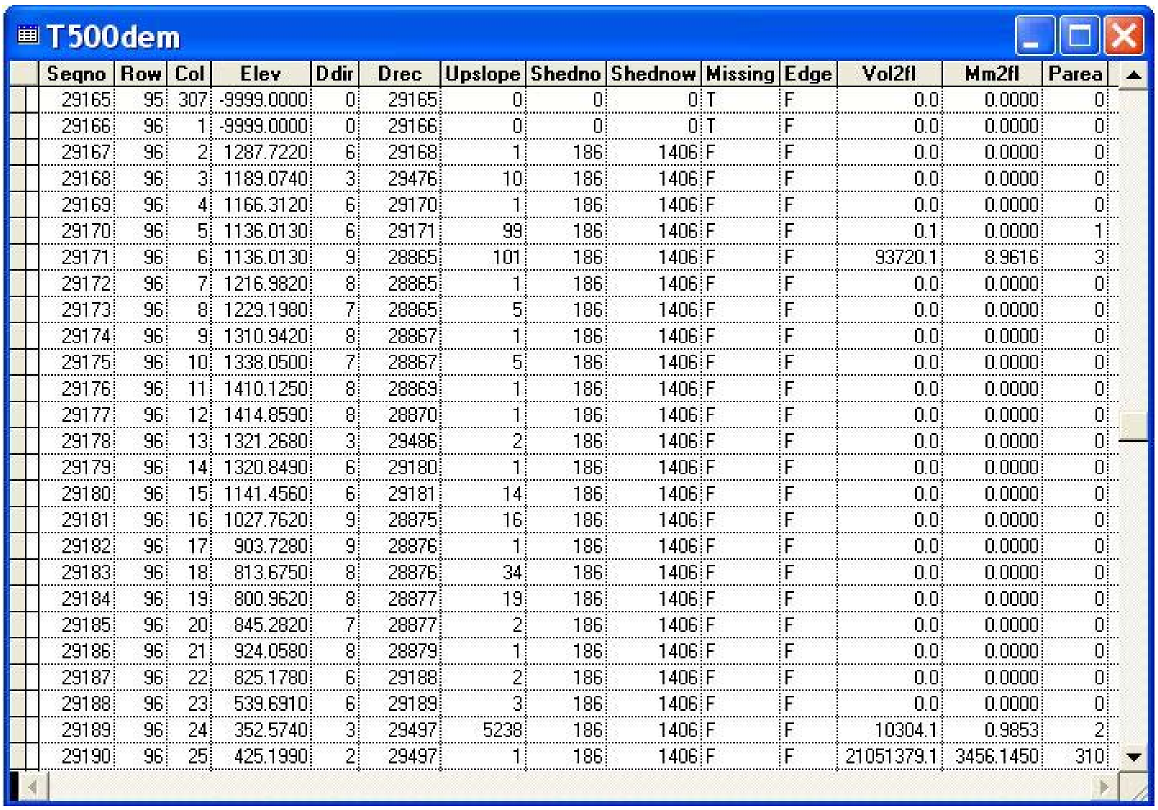
Selection of threshold values for pit area or pit depth that are too large will result in removal of pits that may very likely represent true structural depressions in the landscape. This will have negative consequences in terms of failure to identify and characterize pits that are highly likely to be significant for hydrological modeling purposes. Removal of significant pits also has adverse consequences during application of the default LandMapR landform classification procedures. These procedures use rules that use vertical distance to a pit cell to establish measures of relative landform position. These measures of relative landform position in turn influence the resulting landform classification. Any pit that is completely removed by stage 1 pit removing procedures becomes invisible to subsequent procedures and will not be available to be used to establish relative landform position. Thus, if thresholds are selected that remove valid pits, the measures of relative landform position that should relate to these pits will be in error and any resulting classifications may also be adversely impacted. For this reason, users are cautioned against selecting threshold values for pit depth and pit area that are too large and that will result in most pits being removed by the stage 1 pit removing algorithm.

FlowMapR Output Files

FlowMapR creates 8 different DBF output files (Table 1). Five of the files contain data that describe aspects of hydrological flow patterns for the original DEM and 2 of the files pertain to calculations of notional upslope flow made using an inverted DEM.

The main flow topology file (ID#DEM)

The main DBF file produced by the FlowMapR program is always named ID#DEM, where ID# is a user-assigned alpha-numeric code that is 3 to 4 characters in length (Figure 12).


T500dem

Seqno	Row	Col	Elev	Ddir	Drec	Upslope	Shedno	Shednow	Missing	Edge	Vol2fl	Mm2fl	Parea
29165	95	307	-9999.0000	0	29165	0	0	0	T	F	0.0	0.0000	0
29166	96	1	-9999.0000	0	29166	0	0	0	T	F	0.0	0.0000	0
29167	96	2	1287.7220	6	29168	1	186	1406	F	F	0.0	0.0000	0
29168	96	3	1189.0740	3	29476	10	186	1406	F	F	0.0	0.0000	0
29169	96	4	1166.3120	6	29170	1	186	1406	F	F	0.0	0.0000	0
29170	96	5	1136.0130	6	29171	99	186	1406	F	F	0.1	0.0000	1
29171	96	6	1136.0130	9	28865	101	186	1406	F	F	93720.1	8.9616	3
29172	96	7	1216.9820	8	28865	1	186	1406	F	F	0.0	0.0000	0
29173	96	8	1229.1980	7	28865	5	186	1406	F	F	0.0	0.0000	0
29174	96	9	1310.9420	8	28867	1	186	1406	F	F	0.0	0.0000	0
29175	96	10	1338.0500	7	28867	5	186	1406	F	F	0.0	0.0000	0
29176	96	11	1410.1250	8	28869	1	186	1406	F	F	0.0	0.0000	0
29177	96	12	1414.8590	8	28870	1	186	1406	F	F	0.0	0.0000	0
29178	96	13	1321.2680	3	29486	2	186	1406	F	F	0.0	0.0000	0
29179	96	14	1320.8490	6	29180	1	186	1406	F	F	0.0	0.0000	0
29180	96	15	1141.4560	6	29181	14	186	1406	F	F	0.0	0.0000	0
29181	96	16	1027.7620	9	28875	16	186	1406	F	F	0.0	0.0000	0
29182	96	17	903.7280	9	28876	1	186	1406	F	F	0.0	0.0000	0
29183	96	18	813.6750	8	28876	34	186	1406	F	F	0.0	0.0000	0
29184	96	19	800.9620	8	28877	19	186	1406	F	F	0.0	0.0000	0
29185	96	20	845.2820	7	28877	2	186	1406	F	F	0.0	0.0000	0
29186	96	21	924.0580	8	28879	1	186	1406	F	F	0.0	0.0000	0
29187	96	22	825.1780	6	29188	2	186	1406	F	F	0.0	0.0000	0
29188	96	23	539.6910	6	29189	3	186	1406	F	F	0.0	0.0000	0
29189	96	24	352.5740	3	29497	5238	186	1406	F	F	10304.1	0.9853	2
29190	96	25	425.1990	2	29497	1	186	1406	F	F	21051379.1	3456.1450	310

Figure 12. Illustration of the structure and content of the main ID#DEM file

The first four columns of the ID#DEM file store a unique record number (Seqno) for each grid cell in a DEM, the row number (Row) and column number (Col) of each grid cell in the data matrix and the original value for elevation for each cell (Elev) as read in from the input data file provided. The sequence number is used as a unique key field for joining this file to any other file for the same area that also contains values for sequence number in an identically named Seqno field. Values in the fields row and col are used to reference the locations of individual grid cells in the data matrix both in this ID#DEM file and in the pit files that document the location and characteristics of pits or depressions in the DEM. Missing data is identified by a value of -9999.

The next 2 columns store values computed to represent the local drainage direction (Ddir) for each grid cell in the DEM and the unique record number in the file (Drec) associated with the grid cell into which a particular cell drains. Recording the record number in the file of the cell into which each cell drains is an artifact of the FoxPro database implementation of the initial FlowMapR program. In the earlier FoxPro implementation, the entire data matrix was not read into memory and cell to cell flow was achieved by using direct access of sequential binary files to move from record number to record number according to the values stored in the field Drec.

Table 1. Listing and description of all DBF files produced by the FlowMapR program

File	Description of contents
ID#DEM	This is the main output file. It contains data on flow directions, initial and final catchments and amounts of water required to flood each cell located in a closed depression.
ID#Pit	This file is used to store attributes of all initial pits during stage 1 pit removing. After stage 3 pit removing, information for all pits except final or edge pits has been deleted and descriptions remain for only the attributes of final or edge pits.
ID#Pond	This file stores data on the attributes of all possible pits that remain after stage 1 pit removing. It details all possible ways in which pits could overspill and connect to one another. It is produced by sorting the input DEM file from lowest to highest elevation and computing and removing pits in order from the lowest pit to the highest.
ID#Fill	This file stores data on the attributes and most likely sequence of connectivity for all pits that remain after stage 1 pit removing. The most likely sequence of pit connectivity is estimated through reference to the pit attribute named varatio. This attribute estimates the number of mm of runoff that would have to be generated by all cells in a particular depressional catchment in order to fill the depression fed by the catchment to its overspill volume. Pits in this table have been removed, or coalesced, in order of the mm of runoff that is estimated to be required to fill them to overflowing.
ID#Vold	This is a backup file that stores data on each and every change made to key data fields in the ID#DEM file during stage 2 pit removing. During stage 2 pit removing, current data in the ID#DEM file is copied to the ID#Vold file before any change is made to the ID#DEM file. This copy provides an audit trail of all changes made to the ID#DEM file during Stage 2 pit removing in the exact order that the chages are made. It is therefore possible to copy the data stored in the audit file (ID#Vold) back into the main ID#DEM file to reconstitute the ID#DEM file to its status at any point during Stage 2 pit removing right back to the start.
ID#Mold	This is a backup file that stores data on each and every change made to key data fields in the ID#DEM file during stage 3 pit removing. During stage 3 pit removing, current data in the ID#DEM file is copied to the ID#Mold file before any change is made to the ID#DEM file. This copy provides an audit trail of all changes made to the ID#DEM file during Stage 3 pit removing in the exact order that the chages are made. It is therefore possible to copy the data stored in the audit file (ID#Mold) back into the main ID#DEM file to reconstitute the ID#DEM file to its status at any point during Stage 3 pit removing right back to the start.
ID#iDEM	This file is an exact copy of the main ID#DEM file except that the elevation data in the field ELEV has been inverted relative to the original input elevation data. Inverting the original elevation data permits the same algorithms and data files to be used to compute flow paths and catchments for notional upslope flow from each grid cell. Data in this file are used to support calculations of flow upslope from every cell to compute measures such as vertical and horizontal distance from a cell to the nearest grid cell classified as a ridge or peak cell.
ID#iPit	This file is an exact copy of the master pit file ID#Pit. It differs in that it contains data on pits computed for the inverted DEM, which are actually peaks. It also differs in that FLowMapR does not apply stage 2 or 3 pit removing procedures to calculations of upslope flow. This file therefore contains data for all initial pits that remain after stage 1 pit removing but does not contain any data on "higher order" pits that may subsume these initial first order pits. Data on these "higher order" pits can only be computed by removing lower order pits using the stage 2 and 3 pit removing procedures that are not used for the inverted DEM.

The column named Upslope stores a value for upslope area count for each grid cell. Upslope area is computed by flowing down from each grid cell into all of its down-slope neighbors and adding the value for accumulated upslope area to each down slope cell. The algorithm that does this first sorts all grid cells from highest to lowest elevation and then starts processing the highest elevation cell first. This ensures that all cells that are upslope of a particular cell and may possibly contribute flow into it are processed first, before the value for upslope area is computed for a particular cell. Upslope area count is computed along flow paths that are based on flow directions computed using the standard D8 flow direction algorithm of Morris and Heerdegen (1988). Use of this older D8 flow direction algorithm is also an artifact of the early development of the flow modeling procedures over the period 1988-1991 before any of several newer alternatives had been described (see Wilson, 1996 for a review of other algorithms).

The next two columns store data that identify the local and global catchments that each grid cell is assigned to in a fashion similar to that outlined by Martz and de Jong (1988). Local catchments (Shedno) are defined as the sum of all cells that flow into a local closed depression. In the current implementation of the FlowMapR program, local catchments represent the extent of each DEM that drains into one of the initial, or first order, depressional catchments that remain after stage 1 pit removing has been completed. Stage 1 pit removing is intended to remove very small pits that may be assumed to either arise from errors in the DEM or to be so small as to not warrant identification and documentation. Thus, no information is retained for these small pits, including no information on the location and extent of the catchments associated with them. Global catchments (Shednow) represent the extent of all cells that contribute surface flow to the final, integrated catchments drain to the edge of the DEM and from there allow water to escape to the outside world, beyond the known DEM.

Two columns store logical variables used to record whether a particular grid cell contains a missing data value for elevation (Missing) and whether the particular grid cell in question occurs at the edge of the data matrix (Edge). The Missing data field is redundant, as this information can be determined by comparing the value for elevation stored in the field Elev with the value entered by the user to identify missing data. The presence of this field is another artifact from the earlier FoxPro version of the program, when it was found to be convenient to be able to check for missing values by reference to the value stored for a logical variable rather then through direct comparison of the value stored for elevation. Use of the logical variable Edge is also a relict of the earlier FoxPro program and is likely not absolutely necessary. Initially, edge cells were recognized as cells located at the minimum and maximum row and column locations and these were considered to be the only cells that permitted drainage to occur to the outside world. Later refinements to the program involved adding features that recognized that areas of missing data in the DEM needed to be treated as if they also were edge cells that drained to the outside world. The earlier definitions of what constituted an edge cell were therefore no longer valid, but this column has not been updated to reflect this changed perception.

The last three columns in the ID#DEM file store values for calculations meant to describe the amount of water that would be required to fill each closed depression to the level at which it would first inundate each cell that is located within a closed depression. Only cells that are located within closed depressions can have non-zero values for these variables. The variable volume-to-first-flood (Vol2Fl) represents a calculation of the volume of water (m^3) that would be present in a closed depression at the point at which the surface of the pond first inundated a specific cell. The variable mm-to-first-flood (Mm2fl) represents an estimate of the amount of rainfall (in mm) that would have to fall on, and then run off from, each of the grid cells that can contribute flow into a closed depression in order to produce the volume of water that is computed to be required to fill the depression to the point where it first inundates a given grid cell of interest. This variable is an estimate only, as it is obvious that it is unlikely that every grid cell in a depressional catchment will convert all rain that falls onto it into runoff nor that all runoff will manage to flow all the way to an open water body in a closed depression. The variable is meant to provide an estimate of the relative likelihood that any grid cell will ever become inundated by water that accumulates in a closed depression, rather than providing an exact calculation of the amount of rainfall that will definitively result in filling a depression to a given level. The variable Parea stores a measure of the surface area of a pond (in grid cell units) at the point at which a pond first inundates a particular grid cell of interest.

The initial pit table (ID#Pit)

The FlowMapR program computes and stores a considerable amount of information about all pits (also referred to as depressions) that are identified by processing a DEM to compute flow topology (Figure 13). Data stored in the pit tables provides information on the row and column location of pits, on the depth, volume and area of all pits, on the identity of the adjacent depressional catchments into which each pit is most likely to overspill and on the grid cell locations at which overspill from one depression to another is most likely to occur.

T500pit

Shedno	Edge	Dr_2_mv	Final	Endpit	Shedarea	Pitrow	Pitcol	Pitrec	Pitelev	Pourelev	Prevol	Pitvol	Varatio	Pitarea	Drainsto	Nextpit	Becomes	Inrow	Incol	Inrec	Inelev	Outrow	Outcol	Outrec	Outelev
240	T	T	T	240	64	82	279	25146	1311.8390	1406.8880	0.000	1182.874	18482.41	20	339	1409	240	77	274	23606	1406.8880	76	273	23298	1406.6680
396	T	T	T	396	87	45	290	13798	1147.8060	1201.5680	0.000	101.621	1168.06	4	398	398	396	45	291	13799	1191.7990	44	292	13493	1201.5680
131	T	T	T	131	170	184	93	56274	1081.7140	1160.6960	0.000	133.000	782.35	2	37	37	131	184	92	56273	1106.6780	184	91	56272	1160.6960
807	T	T	T	807	86	7	95	1937	1140.2990	1144.0100	0.000	3.711	43.15	1	923	923	807	7	95	1937	1140.2990	7	94	1936	1144.0100
881	T	T	T	881	203	7	107	1949	1101.6280	1109.7560	0.000	20.276	99.88	3	1147	1147	881	8	108	2257	1104.0760	7	109	1951	1109.7560
992	T	T	T	992	165	7	68	1910	1087.5110	1087.8010	0.000	0.290	1.76	1	1293	1293	992	7	68	1910	1087.5110	8	67	2216	1087.8010
37	T	T	T	37	209	184	88	56269	908.7870	1160.6960	0.000	1416.010	6775.17	21	131	131	37	184	91	56272	1160.6960	184	92	56273	1106.6780
942	T	T	T	942	222	5	192	1420	992.3750	997.6310	0.000	8.247	37.15	3	1203	1203	942	5	191	1419	997.6310	5	190	1418	994.7750
554	T	T	T	554	185	20	30	5863	990.6190	1025.7310	0.000	247.358	1337.07	20	1341	1407	554	21	42	6182	1021.4710	21	43	6183	1025.7310
816	T	T	T	816	215	6	123	1658	969.4100	1009.4620	0.000	180.867	841.24	15	1334	1334	816	6	125	1660	1009.4620	6	126	1661	1001.5240
1311	T	T	T	1311	52	6	145	1680	900.5320	905.1860	0.000	4.654	89.50	2	1288	1288	1311	6	146	1681	905.1860	6	147	1682	902.6260
1360	T	T	T	1360	205	5	170	1398	915.0470	921.9810	0.000	6.934	33.82	1	1358	1405	1360	5	170	1398	915.0470	5	169	1397	921.9810
1288	T	T	T	1288	102	6	149	1684	882.9540	905.1860	0.000	135.265	1326.13	13	1311	1311	1288	6	147	1682	902.6260	6	146	1681	905.1860
1343	T	T	T	1343	418	181	208	55468	658.8680	889.0480	0.000	3769.056	9016.88	40	581	1375	1343	181	206	55466	862.8070	181	205	55465	889.0480
1375	T	T	T	1402	2841	180	202	55155	518.9270	889.0480	2071.23	22071.23	7768.82	168	1343	1343	1375	181	205	55465	889.0480	181	206	55466	862.8070
1377	T	T	T	1390	1108	11	236	3306	1070.4710	1132.4240	916.672	2131.394	1923.64	81	356	1397	1377	24	226	7287	1132.4240	23	225	6979	1128.0270
1379	T	T	T	1407	308	6	118	1653	1017.3390	1018.0650	0.726	0.726	122.25	1	816	816	1379	6	118	1653	1017.3390	6	119	1654	1018.0650
1383	T	T	T	1409	662	45	303	13811	482.4750	788.9940	102.064	5102.064	9217.62	50	1362	1362	1383	35	304	10742	788.9940	34	305	10436	786.5360
1397	T	T	T	1392	344	11	223	3293	1094.5270	1115.1010	174.099	742.574	2158.65	55	1388	1405	1397	23	215	6969	1112.5750	23	214	6968	1115.1010
1405	T	T	T	1404	3737	5	167	1395	877.5300	921.9810	563.487	563.487	466.32	48	1360	1360	1405	5	169	1397	921.9810	5	170	1398	915.0470
1406	T	T	T	1400	10458	183	76	55950	249.8830	427.1170	9695.00	19695.00	1883.25	227	1341	1407	1406	40	16	11989	386.7700	39	15	11681	427.1170
1407	T	T	T	1403	3302	30	14	8917	328.3180	427.1170	515.792	515.792	466.32	14	1406	1406	1407	39	15	11681	427.1170	40	16	11989	386.7700
1408	T	T	T	1408	745	22	305	6752	752.4440	1074.1200	739.260	6739.260	9045.99	95	60	60	1408	14	303	4294	1069.6900	14	304	4295	1074.1200
1409	T	T	T	1410	16613	182	142	55709	504.4580	825.1420	1738.09	41738.09	2512.38	424	600	600	1409	180	136	55089	825.1420	181	135	55395	797.0150
1410	T	T	T	1377	637	59	277	18083	967.4450	1149.0180	330.329	3277.927	5145.88	92	1405	1405	1410	46	246	14061	1142.7740	46	245	14060	1149.0180

Figure 13. Illustration of the structure and content of the first pit table (ID#Pit) produced by the FlowMapR program

Each pit in a DEM is identified by a unique integer ID number that refers to the watershed or catchment that contributes flow into the depression (Shedno). Procedures invoked by the FlowMapR program determine whether a pit may overspill at the edge of the data matrix (Edge) or whether it may overspill into the outside world through adjacency with an area of missing data (Dr_2_mv). The procedures also identify whether a particular pit is a final pit (Final) that overspills into the outside world and that is fed by a fully integrated catchment that consists of all possible cells that are upslope of the pit and that can possibly contribute flow into it. The column Endpit stores a unique integer ID that identifies the number of the final Endpit into which each non-final pit will eventually contribute flow once all pits are full to their overspill level and fully integrated surface flow has been achieved.

The total number of cells that are contained within an identified depressional catchment is stored in the column Shedarea. The lowest cell in each depression is referred to as the pit center cell. The location of the pit center cell is recorded in terms of the record number of the pit cell (PitRec) and the grid cell row (Pitrow) and column (Pitcol) locations. The elevation of the pit center cell is stored in the column Pitelev. The column Pourelev stores the elevation of the pour point, which is the elevation of the lowest cell at which water stored in a particular pit can overspill and contribute flow into an adjacent catchment. The pour elevation is the higher value of the elevation of the cell within the current depressional watershed at which overspill may occur (InElev) and the elevation of the cell in the adjacent catchment (OutElev) at the lowest point at which overspill can occur. The difference between the pit center elevation and the pour elevation will yield the maximum depth of the pit.

The pit table stores two different values for volume of a pit. The value stored in the column Prevol refers to the total volume of water stored in any previously identified pits than may underlie a larger pit formed by coalescence of two or more underlying pits. Initial, or first order, pits will always have a value of zero (0) for Prevol. The value stored in the column PitVol refers to the total volume of a pit including the previously computed volume of any pits that it covers and subsumes. Pit volumes are computed by subtracting the elevation of each cell in a pit from the elevation of the pour point and summing the resulting volume calculations for all grid cells within a depression. Pit volumes need to be multiplied by the area of a grid cell to convert them from relative volumes into true volume measurements in cubic meters (m^3).

The column named Varatio stores an estimate of the amount of rainfall (in mm) that would be required to fill a pit from completely empty to full, assuming that all cells within the catchment received an equal amount of rainfall and all rainfall received ran off from every cell and was transmitted to accumulate in the depression. Volume to area ratio (Varatio) is calculated simply following the formula Varatio = ((pitvol/shedarea)*1000). The absolute value for Varatio is not to be taken literally, but values may be interpreted to give a rough estimate of the relative amounts of rainfall (in mm) that might be required to completely fill one pit versus another. Two pits may both have the same volume, but if one has a much larger potential contributing area (catchment area) it is more likely to fill up sooner (with less rainfall) than the other.

The column Pitarea stores a value for the number of cells that are covered by surface water when a pit or depression is full to its overspill volume and depth. Area in given in terms of number of cells and must be multiplied by the unit area of a grid cell to convert to absolute area in m^2.

The columns Drainsto, NextPit and Becomes are pointers used to identify the number of the adjacent depressional catchment that a particular depression overspills into or becomes. DrainsTo stores the ID number of the adjacent catchment into which a given catchment is most likely to overspill and "drain into" once the depression for that given catchment has been completely filled to its overspill volume. NextPit stores the ID number of an alternate adjacent catchment that the current catchment of interest can overspill into if overspill into the "DrainsTo" catchment is returned to the current catchment because the "DrainsTo" catchment is already full and overspilling back into the current catchment of interest. These two pointers are needed to allow for the possibility that a given catchment might possibly have several neighbors into which it could overspill at the same overspill or pour elevation. The pit removing procedures allow for and check for circular flow from one depressional catchment into several adjacent depressional catchments. Flow is permitted to cycle from one depression into the next until all adjacent depressions that share a common overspill elevation are "filled". Only once all adjacent depressions that share a common overspill elevation are "full" will the pit removing procedures permit identification of a higher elevation pour point and calculation of the statistics associated with a new depression that over-rides and subsumes the underlying depressions that share a common overspill elevation. The column Becomes stores the ID number of the new catchment that is formed when 2 initial or "lower order" catchments coalesce and combine to form a new "higher order" catchment. This field is best understood by opening and investigating a pit file with the name ID#Fill produced by the stage 3 pit removing procedures. In the ID#Fill file data for adjacent pits is always arranged in pairs such that the first listed pit always drains into the second listed pit to become a new joined pit with an ID number identified by the field Becomes. The first pit always has a smaller volume to area ratio (Varatio) than the second and all pairs of pits are sorted in order of increasing volume to area ratio of the first pit in each pair. This is because the pairs of pits are sorted in order of the sequence in which they are considered most likely to fill and overspill which is inferred from the value computed for volume to area ratio (Varatio). It takes some effort to develop an understanding of how the pit removing procedures in FlowMapR work and how the information computed during these pit removing procedures can be read and interpreted. Users may wish to work through a few example DEMs that have only a few well defined depressions in order to appreciate how the pit removing procedures compute all information about how pits fill and overspill and store this information in the relevant pit tables.

The fields InRow, InCol, InRec, InElev and OutRow, OutCol, OutRec and OutElev record the location and elevation of the cells within the current (In) catchment and in the adjacent neighbor (Out) catchment at which overspill from one catchment into the other is judged most likely to occur. The overspill point (or pour point) is the lowest elevation along the boundary between any two adjacent depressional catchments. The record number of the pour cells is stored to facilitate going directly to those cells to initiate pit removing procedures. The elevations are stored to permit comparison of the elevations of potential pour points to all possible adjacent catchments in order to identify the lowest of all possible overspill locations. The lowest of all possible overspill locations is treated as the most likely pour point.

The second pit table (ID#Pond)

The stage 2 pit removing procedures produce a second pit table that is identical in structure with the first, but not exactly identical in content (Figure 14).

T500pond

Shedno	Edge	Final	Endpit	Shedarea	Pitrow	Pitcol	Pitrec	Pitelev	Pourelev	Prevol	Pitvol	Varatio	Pitarea	Drainsto	Nextpit	Becomes	Stage	Removed	Visited
390	F	F	0	507	176	76	53801	260.6850	346.7940	0.000	870.5480	999.90	16	1023	1348	1348	1	T	F
1348	F	F	0	2111	176	76	53801	260.6850	359.8240	3096.386	4157.3590	1969.38	84	432	1349	1349	0	T	F
1352	F	F	0	7942	176	76	53801	260.6850	427.1170	15726.292	17818.7390	2243.61	155	1341	1353	1353	0	T	F
1349	F	F	0	3395	176	76	53801	260.6850	383.7000	5104.717	7967.2370	2346.76	104	186	1351	1351	0	T	F
1351	F	F	0	5635	176	76	53801	260.6850	400.9060	10551.879	13744.9310	2439.21	168	343	1352	1352	0	T	F
1353	T	F	0	11375	176	76	53801	260.6850	1025.7310	18491.293	893429.9690	9999.90	3006	554	1354	1354	0	T	F
1354	T	F	0	11575	176	76	53801	260.6850	1043.4550	893692.960	949599.7830	9999.90	3281	69	1355	1355	0	T	F
1355	T	F	0	12659	176	76	53801	260.6850	1060.9350	950047.990	1008588.6580	9999.90	3349	475	1356	1356	0	T	F
1356	T	F	0	12828	176	76	53801	260.6850	1075.0560	1008884.948	1066331.3560	9999.90	4128	929	1357	1357	0	T	F
1357	T	F	0	13417	176	76	53801	260.6850	1085.3250	1066478.834	1111393.5630	9999.90	4449	1173	1358	1358	0	T	F
1358	T	F	0	13964	176	76	53801	260.6850	1088.5560	1111453.182	1127102.5040	9999.90	4794	951	1372	1372	0	T	F
1372	T	F	0	30709	176	76	53801	260.6850	1097.3000	1624383.243	1725897.7470	9999.90	10322	1211	1373	1373	0	T	F
1373	T	F	0	30753	176	76	53801	260.6850	1104.8690	1725951.661	1810149.8360	9999.90	11556	1096	1374	1374	0	T	F
1374	T	F	0	30959	176	76	53801	260.6850	1114.2340	1810187.201	1924550.2980	9999.90	12844	339	1375	1375	0	T	F
1375	T	F	0	31451	176	76	53801	260.6850	1122.5930	1924764.942	2036421.2670	9999.90	13631	1133	1376	1376	0	T	F
1376	T	F	0	31483	176	76	53801	260.6850	1122.7590	2036444.236	2039142.0980	9999.90	13898	1037	1377	1377	0	T	F
1377	T	F	0	31741	176	76	53801	260.6850	1131.5410	2039233.290	2167472.1730	9999.90	15161	911	1378	1378	0	T	F
1378	T	F	0	31867	176	76	53801	260.6850	1138.3470	2167514.006	2273628.1610	9999.90	15806	887	1390	1390	0	T	F
1390	T	F	0	38368	176	76	53801	260.6850	1139.8360	2480270.282	2508783.1760	9999.90	19087	1030	1391	1391	0	T	F
1391	T	F	0	39060	176	76	53801	260.6850	1146.1110	2508841.829	2624502.0650	9999.90	18570	648	1392	1392	0	T	F
1392	T	F	0	39985	176	76	53801	260.6850	1149.0180	2624565.794	2691268.1820	9999.90	20658	509	1395	1395	0	T	F
1395	T	F	0	40645	176	76	53801	260.6850	1154.4130	2694828.476	2798988.8320	9999.90	19792	1000	1396	1396	0	T	F
1396	T	F	0	40713	176	76	53801	260.6850	1162.3720	2799076.629	2986739.5480	9999.90	22918	822	1397	1397	0	T	F
1397	T	F	0	40850	176	76	53801	260.6850	1164.8130	2986764.030	3042238.7990	9999.90	22762	700	1398	1398	0	T	F
1398	T	F	0	40942	176	76	53801	260.6850	1169.1840	3042270.939	3146064.2410	9999.90	23749	773	1399	1399	0	T	F
1399	T	F	0	41206	176	76	53801	260.6850	1179.9860	3146101.297	3412082.7540	9999.90	25075	438	1401	1401	0	T	F
1401	T	F	0	42223	176	76	53801	260.6850	1231.6580	3413594.156	4820964.9780	9999.90	27600	37	1402	1402	0	T	F
1402	T	F	0	42652	176	76	53801	260.6850	1358.9450	4825405.572	9014863.2440	9999.90	34906	368	1409	1403	0	T	F
1403	T	T	0	42680	176	76	53801	260.6850	1362.6510	9015051.616	9144610.6200	9999.90	35010	333	1406	1407	0	T	F
1407	T	T	0	42782	176	76	53801	260.6850	1362.6510	9144610.620	9144940.5510	9999.90	35016	1406	1406	1408	0	T	F
1408	T	T	0	42848	176	76	53801	260.6850	1362.6510	9144940.551	9145368.5180	9999.90	35031	1406	1406	1409	0	T	F
1409	T	T	0	42959	176	76	53801	260.6850	1362.6510	9145368.518	9145368.6060	9999.90	35032	1406	1406	0	0	F	T
921	F	F	0	890	140	57	42730	262.3640	324.6030	0.000	341.8110	384.06	14	84	1346	1346	1	T	F
1346	F	F	0	1223	140	57	42730	262.3640	342.4290	775.215	741.8620	606.59	12	1023	1348	1347	0	T	F
1347	F	F	0	1604	140	57	42730	262.3640	346.7940	1321.381	2225.8380	1387.68	53	390	390	1348	0	T	F

Figure 14. Illustration of the structure and content of the second pit table (ID#Pond) produced by the FlowMapR program

The columns used to identify the locations and elevations of pour points have been excluded from the illustration of the pond file in Figure 14. The file is shown sorted as it is during the second stage of pit removing procedures. These procedures locate the lowest depression in a DEM data file (here 390) and remove into it all catchments that lie above it and can possibly drain into it. When all catchments lying above a low depression that can possibly drain into it have been removed into it, a global catchment has been defined. The procedures then locate the next lowest depression (here 921) that has not already been removed itself and try to remove into it all catchments above it that can possibly drain into it. This continues until all catchments that can drain into a lower or edge pit have been removed into a lower pit.

Most of the data illustrated in Figure 14 pertain to a single pit centered at the same row (176) and column (76) location. The pit centered at this location is the lowest pit in the example DEM data base (see Pitelev). Each successive record (line) in the pond table illustrated in Figure 14 pertains to a new, larger pit that is formed by the coalescence of two smaller adjacent pits when the higher pit is allowed to fill and overspill into the lower pit. Pits are filled from the bottom up, from the lowest to the highest. As a pit grows, its pit volume (Pitvol) increases as does the volume of previous pits that it overlies and subsumes (Prevol). The area of the new joined pit (Pitarea) becomes increasingly larger as does the area of the combined catchment (Shedarea) that now contributes flow to the new combined pit.

Pits are identified as Edge pits when they can overspill into a cell at the edge of the data set (or into a cell with a missing value) at an elevation that is lower than the elevation of the lowest pour point into an adjacent interior catchment. Pits are identified as Final pits when they can drain to the outside world at an elevation that is lower then the lowest overspill elevation of an interior catchment that is adjacent to them and into which they could overspill. Final pits should not exhibit an increase in pit volume or area as overlying pits overspill into them as they themselves overspill into the outside world, thereby preventing the flooded area from getting any larger or deeper.

The logical fields Visited and Removed are used as flags to keep track of the status or condition of specific pits during the pit removing procedures. The Visited flag is set to false at the start of each attempt to remove a particular pit. It is used to detect circular flow by identifing if flow that originates at a given pit ever returns to that pit. If flow returns to a pit then the program checks all pits along the connected flow path and locates the lowest pit along the path and the second lowest that can overspill into it. The flag Removed is used to identify if a given pit has already been removed during a pit removing exercise. Once a pit has been removed it is no longer available or eligible to be removed again.

The third pit table (ID#Fill)

The stage 3 pit removing procedures produce a third pit table that is identical in structure with the other two, but differs in its content (Figure 15).

T500fill

Shedno	Edge	Final	Endpit	Shedarea	Pitrow	Pitcol	Pitrec	Pitelev	Pourelev	Prevol	Pitvol	Varatio	Pitarea	Drainsto	Nextpit	Becomes
878	F	F	1409	118	66	173	20128	1068.2200	1070.0060	0.000	7.1750	60.81	16	1300	1300	1346
860	F	F	1409	1451	75	161	22879	1066.9400	1070.0060	0.000	24.4790	16.87	31	878	878	1346
1300	F	F	1409	136	65	179	19827	1067.2290	1070.0060	0.000	13.8860	102.10	19	878	1346	1347
1346	F	F	1409	1569	75	161	22879	1066.9400	1070.0060	31.654	31.6730	20.19	63	1300	1300	1347
792	F	F	1409	448	49	235	14971	1123.3540	1127.0440	0.000	11.4280	25.51	12	798	798	1348
798	F	F	1409	191	49	231	14967	1118.9420	1127.0440	0.000	37.3140	195.36	17	792	792	1348
951	F	F	1409	6187	138	130	42189	637.7190	693.7480	0.000	411.2690	66.47	20	1344	1344	1349
1344	F	F	1409	562	166	134	50789	546.3130	593.0490	0.000	310.3760	552.27	14	118	118	1349
118	F	F	1409	1600	182	142	55709	504.4580	593.0490	0.000	1636.8660	999.90	29	1344	1349	1350
1349	F	F	1409	6749	166	134	50789	546.3130	593.0490	310.376	310.3760	66.47	14	118	118	1350
648	F	F	1409	925	130	230	39833	1137.3130	1146.1110	0.000	63.7290	68.90	14	581	581	1351
581	F	F	1409	2142	180	202	55155	518.9270	889.0480	0.000	22829.0310	999.90	169	1343	1343	1351
438	F	F	1409	477	102	251	31258	1162.9540	1169.2450	0.000	57.9710	121.53	32	350	350	1352
350	F	F	1409	540	101	272	30972	1163.9610	1169.2450	0.000	38.4550	71.21	33	438	438	1352
887	F	F	1409	180	47	212	14334	1110.3340	1113.0000	0.000	15.8970	88.32	11	1119	1119	1353
1119	F	F	1409	271	40	208	12181	1109.0780	1113.0000	0.000	22.3560	82.49	22	887	887	1353
1030	F	F	1409	692	55	122	16700	1134.5570	1139.8360	0.000	58.6530	84.76	17	911	911	1354
911	F	F	1409	126	60	126	18239	1126.1720	1131.5410	0.000	41.8330	332.01	14	860	1347	1354
1347	F	F	1409	1705	75	161	22879	1066.9400	1079.9820	45.559	1598.8500	937.74	201	694	694	1355
1354	F	F	1409	818	60	126	18239	1126.1720	1131.5410	41.833	41.8330	84.76	14	1347	1347	1355
1173	F	F	1409	547	86	110	26205	1076.0290	1085.3250	0.000	59.6190	108.99	11	929	929	1356
929	F	F	1409	589	81	87	24647	1068.8000	1075.0560	0.000	147.4780	250.39	42	475	475	1356
475	F	F	1409	169	74	73	22484	1030.2010	1060.9350	0.000	296.2900	999.90	18	343	343	1357

Figure 15. Illustration of the structure and content of the third pit table (ID#Fill) produced by the FlowMapR program

Figure 15 does not illustrate all the columns (fields) contained in the ID#Fill pit table but it does illustrate the critical ones. Examination of Figure 15 reveals that pit records are grouped into pairs of records on adjacent lines. Each pair of records describes how one pit overspills into the other pit to form a new pit that is assigned the ID number stored in the field Becomes. The pairs of pits are listed in the order that they are most likely to fill up and overspill into one another. The order of filling and overspill is estimated through reference to the values stored in the field Varatio. The smaller the volume to area ratio (Varatio) the less runoff is likely to be required to fill a particular pit to its overspill volume. Thus, in the example given here (Figure 15) pit 860 overspills into pit 878 at a Varatio of 16.87 to form a new, joined catchment identified by the new shed ID number 1346. The higher pit (860) is always removed into the lower pit (878) to preserve correct pit topology. The second pair of depressional catchments sees pit 1346 over spilling into pit 1300 at a Varatio of 20.19 to create the new joined catchment numbered 1347. Users should be able to see and appreciate that pits are ordered in the ID#Fill pit table in order of the most likely sequence in which they will overspill into one another to form larger, joined pits. The order can be deduced by referring to values stored in the field Varatio and recognizing that the lowest value of Varatio for each pair of controls the sort order. This sort order differs from the order seen in the second pit (ID#Pond) in which pits are ordered by elevation of the pit center as low pits fill up from the bottom and connect with higher pits that drain into them.

Users do not have to understand the pit removing procedures in order to apply the programs or use the output produced by the programs but there is considerable potentially useful information contained in the pit tables and some users may wish to try to access and interpret these data.

The first audit trail file (ID#Vold)

The purpose of the audit trail files is to record a complete audit trail of all changes made to the original values stored in the ID#DEM file for drainage direction, drainage record number, upslope area and catchment ID number for each cell for which the value is changed during pit removing. Changes made during stage 2 pit removing are stored in an audit trail file that is given the name ID#Vold (Figure 16) which stands for old values associated with removal of pits by volume.

T500vold

Seqno	Ddir	Drec	Upslope	Shednow	Stage	Urec	Ds_area	Varatio
42732	3	43040	21	84	1346	0	0	384.1
43040	3	43348	56	84	1346	0	0	384.1
43348	2	43655	71	84	1346	0	0	384.1
43655	5	43655	333	84	1346	0	0	384.1
42731	4	42730	4	921	1346	0	0	384.1
42730	5	42730	890	921	1346	0	0	384.1
46730	2	47037	5	1023	1347	0	0	606.6
47037	2	47344	15	1023	1347	0	0	606.6
47344	3	47652	23	1023	1347	0	0	606.6
47652	3	47960	39	1023	1347	0	0	606.6
47960	3	48268	65	1023	1347	0	0	606.6
48268	2	48575	68	1023	1347	0	0	606.6
48575	5	48575	381	1023	1347	0	0	606.6
46422	8	46115	39	1346	1347	0	0	606.6
46115	8	45808	45	1346	1347	0	0	606.6

Figure 16. Illustration of the structure and contents of the ID#Vold audit trail DBF table

The FlowMapR pit removing procedures operate by locating the cell within a depressional catchment that occurs at the pour point within the catchment (Inrec) and tracing down the flow path from this pour point cell until flow terminates at the pit centre cell for the current catchment. As this flow path is traversed, the original flow direction for each cell is changed from pointing down slope towards the pit centre to pointing upslope towards the pour point cell (Figure 17). Changes are also made to the value stored for the upslope area count for each cell that lies along this flow path from the pour point cell to the pit centre cell. The value for upslope area count in each cell is reduced by an amount equal to the value of upslope area recorded for the previous cell immediately upslope of it along the flow path from the pour point cell. This has the effect of removing the contribution to the upslope area count of a cell that arises from flow contributed by the flow path that is being reversed.

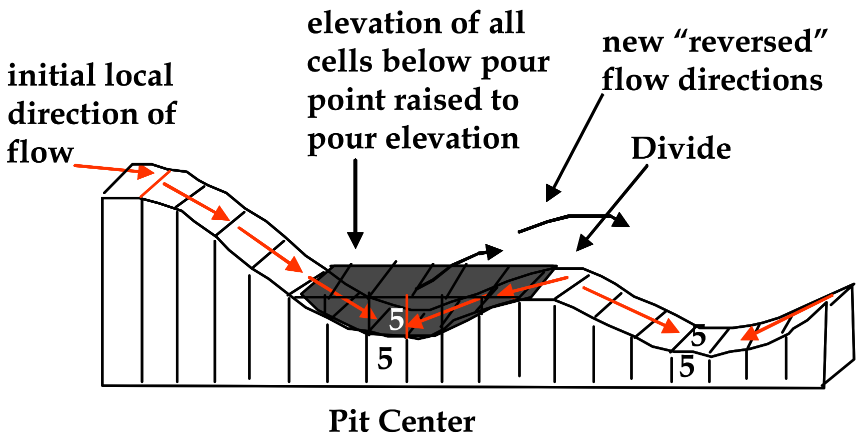


Figure 17. Illustration of the procedure used to "remove pits" by reversing flow directions from the pour point cell

When the flow path has been traversed from the pour point cell to the pit centre cell all cells along the path that flow has to take to escape from the pit being removed have had their drainage directions reversed to point up slope towards the pour point cell and have had their upslope area counts revised to remove the contributions to upslope associated with that flow path. The algorithm then traverses the new, reversed flow path again from the pit centre cell up slope to the pour point cell. As it traverses this upslope flow path it adds to the value stored for upslope area for each cell the accumulated area of the previous cell. Thus it will add the value for upslope area stored for the old pit centre cell to the value already stored for the first cell upslope from the pit centre cell and so on, until it reaches the interior pour point cell.

At this point, the algorithm changes the flow direction assigned to the interior pour point cell so that it now points into the previously computed exterior pour point cell (the Outrec). It then traces down the flow path from the exterior pour point cell until flow terminates at the pit centre cell for the adjacent (lower) depressional catchment into which the upper catchment is being removed. As it traces down this exterior catchment flow path it adds a value equal to the total area of the upper catchment that is being removed to the value stored for upslope area for each and every cell along the flow path. This reflects the fact that the entire area of the catchment being removed is now considered to be capable of contributing flow to this flow path from the overspill or pour point onwards.

The point of the audit trail file is that benefits can be realized by maintaining a full audit trail and not just throwing away all original values as they are changed. Before any value of any cell is changed by the pit removing procedures, the original values are stored in the relevant fields in the audit trail file. New values are then assigned to a cell that requires new values in order to permit a pit to be "removed". In the example given (Figure 16) one can see that the first pit removed by the stage 2 pit removing procedures was a pit associated with catchment number 84 and that the flow path from the pour point cell for this catchment to the pit centre cell was only four cells long. The original values for drainage direction (Ddir) drainage record number (Drec), upslope area count (upslope) and catchment ID number (Shednow) for each of these four cells were recorded in the first four lines of this DBF table before any changes were made by the pit removing procedures.

Maintaining a complete audit trail of all changes made during pit removing supports an ability to reverse any changes made during pit removing by restoring the original values to the appropriate cells in the correct order. If one goes to the bottom of the audit trail file and writes the stored data values back over top of the changed values in reverse order from bottom to top of the file, one is effectively reversing all changes in the reverse order in which they were made. If the entire extent of the audit trail file is written back into the original ID#DEM file in reverse order, one will have effectively restored the flow topology of the ID#DEM file to the state it was in when stage 2 pit removing began. If one were to only restore a portion of the changes back to a specified location in the audit trail file, one would have set the flow topology to the state it was in at a particular stage in the stage 2 pit removing exercise. Thus, if one wished to see what the flow network looked like after a certain stage of stage 2 pit removing, one would only have to restore the original values from the bottom of the ID#Vold audit trail file back up to the desired stage. This means that it is possible to be able to rapidly depict the flow network at any stage of pit removal from no pits removed (non-integrated drainage into all local depressions) to all pits removed (fully integrated drainage with all cells flowing to the edge of the data matrix) and any state in between these two end members.

Unlike most other pit removing approaches, none of the valuable information on the sequence of changes in flow topology associated with the filling and over spilling of pits is lost or thrown away in the FlowMapR approach. It is possible and practical to select any state between completely non-integrated and fully integrated drainage and to rapidly reconstitute and depict the cell to cell flow topology and the resulting drainage network that would be associated with this level of pit filling and drainage integration. This represents a major advantage of the FlowMapR approach to pit removing but it is one that has been little exploited or used to advantage to date.

The second audit trail file (ID#Mold)

The second audit trail file (ID#Mold) (Figure 18) is identical in structure and intent to the first (ID#Vold). The only difference is that the ID#Mold file stores backup data associated with changes made to the flow topology of the ID#DEM file during stage 3 pit removing.

T500mold

Seqno	Ddir	Drec	Upslope	Shednow	Stage	Urec	Ds_area	Varatio
20736	8	20429	1	878	1346	0	0	16.9
20429	9	20123	20	878	1346	0	0	16.9
20123	9	19817	7	878	1346	0	0	16.9
19817	3	20125	25	878	1346	0	0	16.9
20125	3	20433	0	878	1346	0	0	16.9
20433	8	20126	74	878	1346	0	0	16.9
20126	9	19820	-25	878	1346	0	0	16.9
19820	3	20128	111	878	1346	0	0	16.9
20128	5	20128	118	878	1346	0	0	16.9
21042	7	20734	5	860	1346	0	0	16.9
20734	2	21041	141	860	1346	0	0	16.9
21041	1	21347	143	860	1346	0	0	16.9
21347	7	21039	143	860	1346	0	0	16.9
21039	2	21346	122	860	1346	0	0	16.9
21346	1	21652	747	860	1346	0	0	16.9
21652	1	21958	98	860	1346	0	0	16.9
21958	6	21959	744	860	1346	0	0	16.9
21959	3	22267	127	860	1346	0	0	16.9
22267	1	22573	767	860	1346	0	0	16.9
22573	1	22879	788	860	1346	0	0	16.9
22879	5	22879	1451	860	1346	0	0	16.9
19515	9	19209	1	1300	1347	0	0	20.2
19209	3	19517	19	1300	1347	0	0	20.2
19517	3	19825	4	1300	1347	0	0	20.2
19825	6	19826	84	1300	1347	0	0	20.2
19826	6	19827	7	1300	1347	0	0	20.2

Figure 18. Illustration of the structure and contents of the ID#Mold audit trail DBF table

Stage 3 pit removing removes pits according to the sequence in which they are most likely to fill and overspill. This sequence is estimated through reference to the variable volume to area ratio (Varatio) which can be interpreted as the number of mm of runoff that would be required to fill a given depression to its maximum capacity, if one were to assume a completely impervious surface for all cells and 100% conversion of rainfall to runoff with no infiltration or evaporation.

In the example provided, it is clear that the stage 3 pit removing first removes pit 878 into pit 860 to produce a new depressional catchment numbered as 1346 after an estimated 16.9 mm of runoff. The flow path from the pour point cell to the pit centre cell in catchment 878 is 9 cells long and the flow path from the exterior pour point cell to the pit centre cell in catchment 860 is 11 cells long. These are the only cells in either catchment for which changes need to be made in order to reverse the flow paths and "remove" pit 878 into pit 860. Compare the data in the audit trail file (Figure 18) to the corresponding first two records in the ID#fill pit table (Figure 15) and once can see that the audit trail file provides confirmation of the pit table data. Each numbered stage in Figure 18 corresponds to the removal of one pit into a second, lower pit. In the case of stage 3 pit removing, the pits are removed in order of the mm of runoff estimated to be required to fill each pit to its overspill volume.

The ID#Mold file can be used to reconstruct the flow topology for a DEM to correspond to any desired state corresponding to any specified amount of notional runoff. One has only to write all the archived data back into the original master ID#DEM file in reverse order from bottom to top of the ID#Mold audit trail file and to stop restoring the old data at a specified value for the variable Varatio in the ID#Mold file. Varatio can be interpreted as the amount of runoff (in mm) that will result in a particular set of pits being filled to capacity and over spilling. Thus, one can reconstruct and "see" any desired drainage network corresponding to any level of rainfall runoff.

The inverted DEM file (ID#iDEM)

The inverted iDEM file (Figure 19) is identical in structure to the original ID#DEM file.

T500idem

Seqno	Row	Col	Elev	Ddir	Drec	Upslope	Shedno	Shednow	Missing	Edge	Vol2fl	Mm2fl	Parea
15997	53	33	1096.6770	9	15691	6	50	1341	F	F	2119504768.0	369251.7017	14730
15998	53	34	1077.5280	9	15692	2	50	1341	F	F	2659733504.0	463368.2063	20395
15999	53	35	1051.6270	9	15693	64	50	1341	F	F	3247683584.0	565798.5338	24484
16000	53	36	1034.9100	8	15693	5	50	1341	F	F	3678376448.0	640832.1338	26675
16001	53	37	1019.0130	7	15693	1	50	1341	F	F	4114380288.0	716790.9909	28116
16002	53	38	1000.2720	9	15696	4	191	1341	F	F	4653875712.0	810779.7408	29400
16003	53	39	973.3030	8	15696	25	191	1341	F	F	5472283136.0	953359.4314	30857
16004	53	40	921.2370	6	16005	1	1379	1341	F	F	7138141184.0	1243578.6035	33035
16005	53	41	863.9580	9	15699	2	1379	1341	F	F	9088652288.0	1583388.9003	34960
16006	53	42	823.3940	6	16007	1	172	1341	F	F	0.0	0.0000	0
16007	53	43	692.5760	8	15700	189	172	1341	F	F	0.0	0.0000	0
16008	53	44	740.2810	4	16007	164	172	1341	F	F	0.0	0.0000	0
16009	53	45	759.6870	7	15701	2	1379	1341	F	F	0.0	0.0000	0
16010	53	46	840.9820	4	16009	1	1379	1341	F	F	0.0	0.0000	0
16011	53	47	804.6840	8	15704	1	1341	1341	F	F	0.0	0.0000	0
16012	53	48	764.3810	8	15705	4	1341	1341	F	F	0.0	0.0000	0
16013	53	49	783.9760	8	15706	5	1341	1341	F	F	0.0	0.0000	0
16014	53	50	795.0930	7	15706	8	1341	1341	F	F	0.0	0.0000	0
16015	53	51	822.5130	7	15707	2	1341	1341	F	F	0.0	0.0000	0
16016	53	52	867.1670	7	15708	1	1341	1341	F	F	8976402432.0	1563833.1763	34871
16017	53	53	827.5250	8	15710	6	484	1341	F	F	0.0	0.0000	0
16018	53	54	860.8590	8	15711	1	484	1341	F	F	0.0	0.0000	0
16019	53	55	846.2300	7	15711	51	484	1341	F	F	0.0	0.0000	0
16020	53	56	930.8210	4	16019	1	484	1341	F	F	6823277568.0	1188724.3150	32681

Figure 19. Illustration of the structure and content of the inverted ID#iDEM file

The only difference between the inverted DEM file and the original DEM file is that the elevation values in the inverted DEM file have been reversed or inverted. The effect of this reversal is that pits in the original DEM file become peaks in the inverted DEM file and peaks in the original DEM file become pits in the inverted DEM file. Flow directions computed for the inverted DEM data therefore simulate flow upslope from each grid cell until flow terminates at a pit centre cell that is equivalent to a peak in the original DEM.

The approach of inverting the DEM and then computing flow topology for the inverted DEM was adopted so that all the algorithms and code developed to compute flow topology for a non-inverted, original DEM could be used unaltered to compute notional upslope flow for an inverted DEM. The need to compute upslope flow paths was related to a desire to compute a series of measures of relative slope position for each cell in a DEM to use in subsequent landform classification programs. Using the conventional D8 flow direction algorithm, it is quite common for as many as 50% of all cells in a DEM to receive no flow input from upslope cells. With only the information provided by down slope flow topology, there is no easy way to compute the distance or elevation change from a given cell upslope to the nearest peak or ridge cell to which it is connected by a defined flow path. Calculation of notional upslope flow using an inverted DEM solves this problem by providing explicit paths of steepest upslope flow from every cell in a DEM to the nearest peak (and ridge) cell to which it is connected by a defined flow path.

The example data provided were taken from a DEM for a fairly rugged mountainous area in the interior of British Columbia. It is therefore not surprising that inversion of the DEM leads to the recognition of a very large number of pits in the inverted DEM since the natural mountainous landscape had many local peaks in it (which became pits in the inverted DEM). This effect can be seen in the large number of cells that have non-zero values for the variables Vol2Fl, Mm2Fl and Parea in Figure 19 and in the very large absolute values associated with these variables. The large number of pits in an inverted DEM makes it impractical and un-necessary to apply all three of the pit removing stages used for the non-inverted DEM. Only stage 1 pit removing is applied to the inverted DEM by FlowMapR.

The inverted DEM pit file (ID#iPit)

Application of the flow calculation algorithms to the inverted DEM only produces a single pit table (ID#iPit) as only stage 1 pit removing procedures are applied to the inverted DEM.

T500ipit

Shedno	Shedarea	Pitrow	Pitcol	Pitrec	Pitelev	Pourelev	Prevol	Pitvol	Varatio	Pitarea	Drainsto	Inrow	Incol	Inrec	Inelev	Outrow	Outcol	Outrec	Outelev	Edge	Final
84	218	153	126	46790	801.0640	853.1690	0.000	298.2970	1368.33	11	0	149	124	45560	846.9760	148	124	45253	853.1690	F	F
545	67	60	186	18299	1059.5730	1095.8850	0.000	207.2860	3093.82	13	0	59	190	17996	1094.7930	59	191	17997	1095.8850	F	F
610	93	4	243	1164	605.9060	772.6300	0.000	2131.8980	9999.90	26	0	8	245	2394	740.7120	9	246	2702	772.6300	F	F
869	83	44	227	13428	982.4770	1067.7680	0.000	694.0400	8361.93	25	0	49	227	14963	1064.4080	50	228	15271	1067.7680	F	F
1235	32	104	280	31901	851.5000	956.5800	0.000	546.9110	9999.90	14	0	103	277	31591	956.5800	104	276	31897	952.5420	F	F
79	159	45	1	13509	1167.3520	1552.0900	0.000	7221.7540	999.90	69	53	38	1	11360	1552.0900	37	2	11054	1529.9860	F	F
53	119	30	1	8904	1263.6430	1552.0900	0.000	11662.0850	999.90	82	79	37	2	11054	1529.9860	38	1	11360	1552.0900	F	F
38	113	61	13	18433	1210.0920	1260.8480	0.000	581.2990	999.90	17	138	65	14	19662	1246.4000	66	13	19968	1260.8480	F	F
93	116	162	200	49627	927.0110	979.5630	0.000	452.6040	999.90	18	154	159	198	48704	979.5630	158	197	48396	975.3040	F	F
96	164	145	124	44332	799.0430	853.1690	0.000	210.8460	999.90	12	84	148	124	45253	853.1690	149	124	45560	846.9760	F	F
87	149	160	140	48953	1122.4960	1316.7820	0.000	4649.8680	999.90	62	132	157	142	48034	1316.7820	156	143	47728	1309.3500	F	F
204	54	6	161	1696	1183.7850	1235.8450	0.000	336.3260	999.90	13	231	8	163	2312	1231.1460	9	162	2618	1235.8450	F	F
231	52	9	159	2615	1204.7210	1235.8450	0.000	175.8730	999.90	15	204	9	162	2618	1235.8450	8	163	2312	1231.1460	F	F
41	133	73	22	22126	1184.4370	1353.6220	0.000	4090.4570	999.90	35	38	68	19	20588	1298.5770	67	18	20280	1353.6220	F	F
249	53	181	222	55482	1043.5460	1251.4650	0.000	1453.1570	999.90	22	274	180	224	55177	1251.4650	179	225	54871	1245.5330	F	F
292	146	30	64	8967	1118.7210	1128.2980	0.000	44.8710	307.34	12	1094	30	66	8969	1128.2980	31	67	9277	1126.8660	F	F
261	81	14	172	4163	1144.0160	1179.0400	0.000	250.4630	999.90	15	355	14	175	4166	1179.0400	15	176	4474	1178.7860	F	F

Figure 20. Illustration of the structure and content of the inverted pit file ID#iPit

The inverted DEM pit file (ID#iPit) is identical in structure to the original non-inverted pit file however several fields (columns) of data are not used by the inverted DEM pit file. No data are computed or stored for the fields Edge, Final, NextPit or Becomes. The fields Removed, Stage, Visited and Endpit are also not used for the inverted DEM pit removing and contain no valid data. Values computed for pit volume and pit area can get very large as most natural landscapes tend to have large peaks or domes that produce large values for pit volume when inverted and treated as pits.

Using and visualizing the FlowMapR Output Data

FlowMapR is mainly used to pre-process DEM data to compute flow topology for use by subsequent programs, specifically FormMapR and WeppMapR.

The FormMapR program uses both up-slope and down-slope flow topology computed by FlowMapR to assist in calculating flow distances and changes in elevation from each cell in a DEM to the closest cells recognized as pit or peak, channel or divide cells. Distance to peak and divide cells is measured by flowing up-slope along notional paths of upslope flow from every cell in a DEM to the nearest cell recognized to be a peak cell or a divide cell to which it is connected by a defined flow path. Similarly, distance to pit and channel cells is measured by flowing down-slope along paths of steepest descent from every cell to the nearest cell that is classified as a pit or channel cell.

Recognition of cells as belonging to ridges or channels is accomplished by selecting threshold values for upslope area count (for both down-slope and notional up-slope flow) that are believed to correspond well with the actual observed drainage network. It is therefore often advantageous to produce maps that depict the drainage network associated with different threshold values for upslope area count as computed by FlowMapR in order to provide guidance and confidence in selecting appropriate values of upslope area count to use to identify likely channel and divide networks. Different threshold values can be mapped and the resulting simulated drainage networks can be compared to available information that depicts the known drainage network to aid in selecting the most appropriate threshold values to use. Figure 21 illustrates the drainage networks simulated using three different threshold values for upslope area count of 100, 200 and 300 cells against a backdrop of the hillshade image produced from the DEM used to compute the simulated drainage network. The lower the threshold value selected for upslope area, the longer, more complex and more extensive will be the resulting simulated drainage network. The higher the selected threshold value the simpler and less complex the resulting drainage network will be.

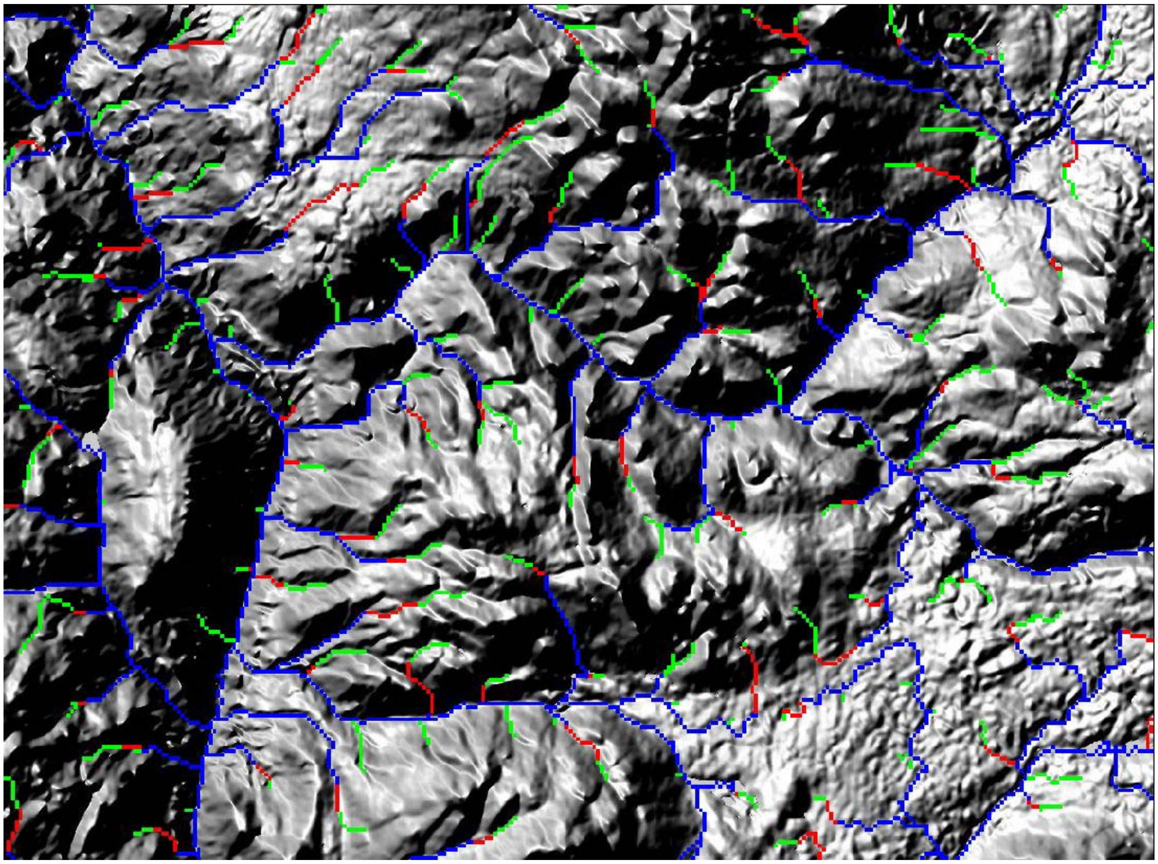


Figure 21. Ilustration of drainage networks simulated using threshold values of 100 (green), 200 (red) and 300 (blue) upslope cells

Several rules of thumb have been developed for selecting appropriate threshold values for upslope area count. These rules are based on the experience gained from processing hundreds of DEMs of varying resolution.

In general appropriate threshold values usually fall into the range of 300-1000 cells of upslope area count. A value of 300 has consistently worked well for high-resolution DEMs with a horizontal resolution of 5 m and a vertical accuracy of +/- 30 cm. I have had to use threshold values of 1000 up to 2000 to simulate drainage networks for DEMs with grid resolutions of 10 to 25 m that cover very large areas (10-50 km on a side) where I wanted to avoid simulating overly complex networks. Figure 21 depicts an area 28 km EW by 22 km NS for which a drainage network was simulated using a 100 m grid DEM. With a 100 m grid DEM, fewer cells are needed to produce an appropriate threshold since every 100 m cell contains an area equivalent to 100 10 m cells. Thus a threshold value of 100 cells for a 100 m grid DEM is equivalent in absolute terms to a threshold of 10,000 cells for a 10 m grid DEM.

Users are advised to make maps of the drainage networks produced by selecting different threshold values for upslope area for most areas that they process through FlowMapR, at least until they develop confidence that a particular threshold value will consistently produce appropriate simulated drainage networks for their particular areas and DEM resolutions. I must confess that I very frequently use a default threshold value of 300 for 5 m grid DEMs of relatively small areas (800-2000 m on a side) without first making maps to depict various simulated drainage networks. This has consistently worked well for me and I seldom check before using it, but I do check if I am working in a new or unfamiliar terrain or with unfamiliar DEM data or different horizontal grid resolution.

Users will be asked to enter appropriate threshold values for upslope area to use to identify channels and ridges when they run either the FormMapR or the WeppMapR programs. These programs will produce very different results depending upon the threshold values entered to identify channel and divide networks. If users enter inappropriate threshold values they will likely get inappropriate results from these two programs.

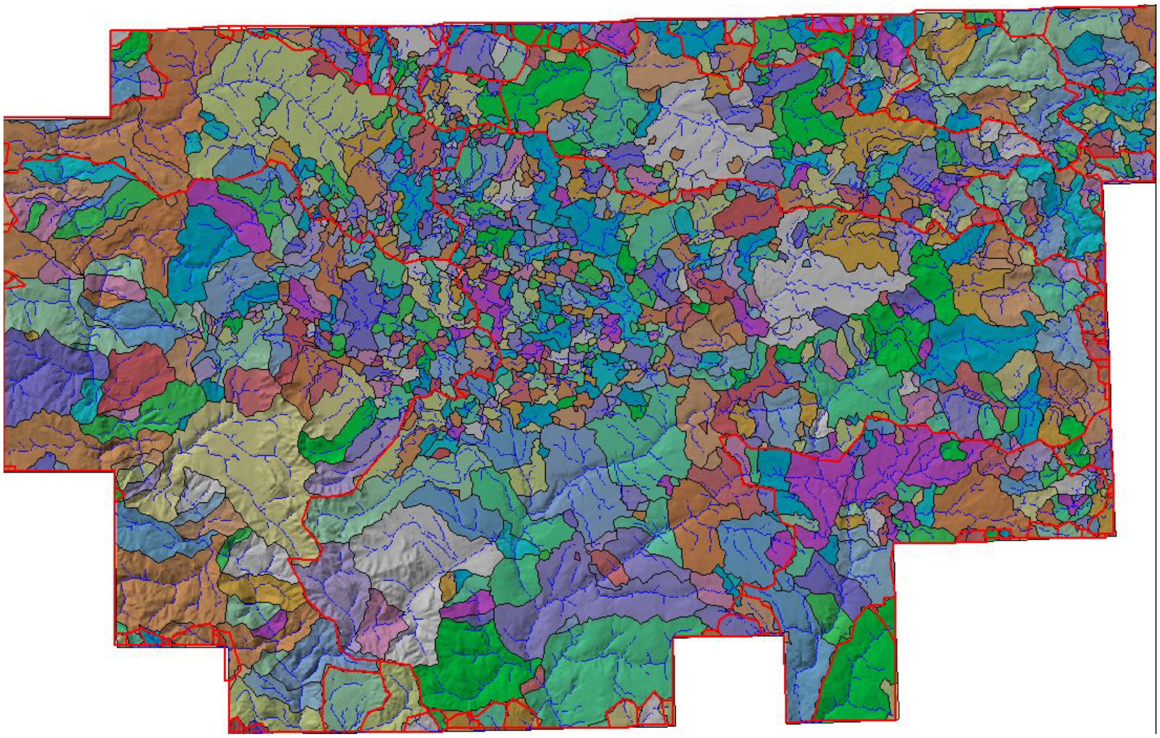


Figure 22. Illustration of local and global catchments computed for a 100 m DEM for an area 150 km EW by 95 km NS

Users may also find it instructive to create and review maps that depict the location and extent of local catchments (Shedno) and global catchments (Shednow) as computed by the FlowMapR program and recorded in the ID#DEM file (Figure 22). Review of these maps can help to identify possible processing errors or errors in the input DEM that lead to incorrect or inconsistent catchment delineation. Local catchments depict areas that contribute flow into local closed depressions as computed by FlowMapR at the end of stage 1 pit removing. Global catchments depict areas of fully integrated flow after all pits have been removed by stage 3 pit removing. Global catchments should flow to and connect to the outside edge of the DEM matrix and no global catchment should flow into and combine with any other global (edge) catchment. The thick red lines in Figure 22 depict the extent of the simulated global catchments while the black lines and various background colors depict the location and extent of local catchments. Each numbered catchment can be linked to its description in the appropriate ID#Pit file via the catchment ID number (shedno for local catchments and shednow for global catchments).

Users may also wish to make and review maps of the variables vol2fl, mm2fl and parea that are stored in the ID#DEM file. These variables provide a good indication of which portions of a DEM are contained within closed depressions and how much water is required to first flood each cell in a closed DEM in terms of both absolute volume (vol2fl) and relative amount of runoff (mm2fl). The data for mm2fl can be interpreted in terms of the relative likelihood that a given cell contained within a closed depression will ever receive enough runoff to become inundated. It may be considered to be a measure of relative likelihood of inundation from runoff water.

Users need to be patient with the FlowMapR program as it can take a long time to run to completion on large data sets, particularly if they contain many depressions or pits.

FormMapR

Purpose

The FormMapR component computes a number of terrain derivatives that describe surface form, orientation, wetness index, relative and absolute relief and relative and absolute slope lengths. These terrain derivatives are used as the primary (or sole) inputs to the subsequent FacetMapR program for classifying landforms or ecological spatial entities.

Input Data

The FormMapR component uses output from the FlowMapR program as well as the original input elevation data to perform its calculations. FormMapR will not work properly unless the FlowMapR program has been run successfully to completion prior to running FormMapR and unless all files produced by application of the FlowMapR program are present in the default folder that is identified as containing the required input data.

As with FlowMapR, FormMapR operates on input elevation data formatted as a regular (raster) grid and stored in a DBF format file. It does not operate on elevation data formatted as irregular x, y, z point data, as a triangular irregular network (TIN) or as vector contours.

Operation of the FormMapR Program

The FormMapR interface is illustrated in Figure 23.

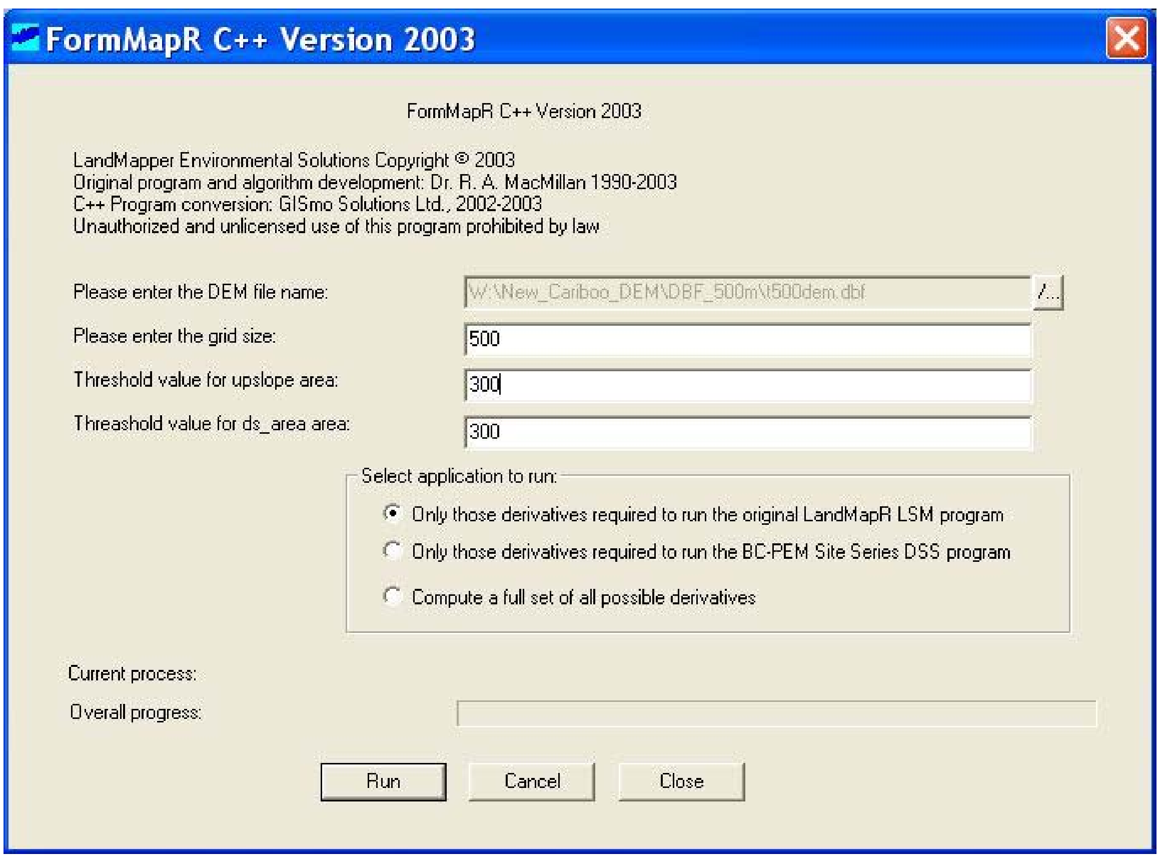


Figure 23. Illustration of the FormMapR interface and input requirements

Running FormMapR requires the user to make the following decisions and input the following user supplied information.

Identifying the input data location and working folder

First, the user must navigate to the folder that contains all of the required input data and to select the ID#DEM file that is associated with the site to be processed. This ID#DEM file must have been produced by a successful application of the FlowMapR program. Other files produced by the FlowMapR program (e.g. ID#Pit, ID#iPit) must also have been pre-computed by the FlowMapR program and must be present in the same folder as the selected ID#DEM file. The folder that you navigate to will automatically become the default folder into which all output from FormMapR will be placed.

Entering the horizontal dimensions of a grid cell

The user must then enter the horizontal dimensions of a grid cell in meters. The FormMapR program was never set up to handle measurement units other than meters. The user-entered value for grid dimension is used in calculations of slope gradient and curvature as well as to compute distance from each cell to cells identified as channels and ridges or pits and peaks.

There are occasions where it can be beneficial to lie to the program and to enter incorrect values for the horizontal dimensions of a grid cell. An example of such a situation would be the case of a DEM for a landscape that had very low and subtle relief. The existing landform classification rules (e.g. LM3Crule) would likely classify such areas into mostly three classes, namely level upper, level mid and level lower slopes, as most cells would not likely produce high enough values for slope gradient or slope curvature to be considered either sloping or strongly convex or concave. If the horizontal grid dimensions were actually 5 m and the user entered a value of 1 m for the horizontal grid dimensions, this would be equivalent to applying a 5 times vertical exaggeration to the input DEM data. This vertical exaggeration increases the values computed for slope gradient and curvatures by a factor of five, resulting in many more cells being considered to have strong slopes and curvatures.

From this example, it is clear that there may be situations where it is advantageous to enter an incorrect value for the horizontal grid dimensions in order to achieve a de facto rescaling of the input DEM data without having to actually make any changes to the input data. Entering values that are less than the actual grid dimensions will have the effect of exaggerating the relief and generating values for slope and curvature that are larger than they actually are. Entering values that are greater than the actual grid dimensions will have the effect of flattening the landscape and producing lower values for slope and curvatures. This approach permits users to, in effect, apply different classification rules to different types of landscapes without having to change the input rule files in any way. The rule files stay the same, but the input data sets are altered to exaggerate or subdue the landscape and alter the values for slope and curvature presented to subsequent classification procedures.

Entering threshold values for recognizing channels and ridges

Users must decide upon, and enter, threshold values for upslope area count for both down-slope and upslope flow. As previously discussed, these threshold values are used to identify which grid cells get recognized as belonging to channels (for down-slope flow) or ridges (for upslope flow). It is important to select threshold values that result in production of a channel and ridge network that is reasonable and realistic for the area, the terrain and the objectives of the subsequent landform classification. If the entered threshold values are too low, too many cells will be classed as belonging to channels or ridges. This can confuse and distort the calculations of absolute and relative relief and slope position performed by the FormMapR program. In the same vein, if threshold values are too high, too few cells may be classed as channels or ridges and calculations of distances to channels or ridges will not differ much from the parallel calculations of distances to pits and peaks.

A default value of 300 cells of upslope area count has been found to work well for many kinds of terrain and many different DEMs of different grid resolutions and extent. The 300 cell value seems to be particularly suitable when processing relatively high resolution (5-10 m horizontal grid cells) DEM data for field and farm sized areas (800 m to 1600 m on a side). For DEM data sets with a very small number of cells (e.g. less than 10,000) it may frequently be necessary to select lower threshold values (100-200 cells) as there are simply not enough cells in the DEM to produce large upslope area counts in the range of 300 or greater. For very large DEM data sets containing 1 million cells or more, it has often proven better to select threshold values closer to 1000 cells as lower thresholds can result in recognition of divide and channel networks that are overly detailed and confusing.

It is advisable for users to produce maps and visualizations of upslope area count immediately after running FlowMapR and before running FormMapR so that they can be confident they have selected and entered threshold values that will produce appropriate stream and channel networks. Having said this, I have very often selected and entered threshold values "blindly" without reference to maps of different threshold values. This approach seems to work if one has processed a large number of DEMs of similar resolution for similar terrain and if a particular threshold value has consistently proven to produce reasonable divide and channel network results.

Upslope area represents a count of numbers of cells and does not represent absolute upslope area in square meters. The number of cells needs to be multiplied by the area of a single grid cell to convert upslope area count into a measure of absolute upslope area. There are both advantages and disadvantages to storing upslope area as a cell count rather than an absolute area value. Cell counts produce smaller numbers and take up less space both in memory and on disk. This was an important consideration at the time the programs were first written. Cell counts are also relative, rather than absolute. This has turned out to be an advantage in many instances. DEMs with larger horizontal grid resolutions (e.g. 100-500 m) tend to be used to cover larger areas than DEMs with smaller grid resolutions (e.g. 5-10 m). One tends to be interested in lower order main channels when working with coarser resolution (100-500 m) DEMs that cover large areas and to be interested in higher order ephemeral channels when working with finer resolution DEMs for smaller areas. A threshold value of 300 cells tends to be appropriate across a variety of scales for different resolutions of DEM data and extent of area covered. Because upslope area count is a relative measure, it tends to support "scaling up" in that only larger and more conspicuous channels and ridges are recognized by the 300 cell threshold for coarse resolution DEMs and smaller and more ephemeral channels and ridges are recognized by the same 300 cell threshold for finer resolution DEMs covering a smaller area. Problems with this default threshold value appear to occur when working with fine resolution DEMs that cover very large areas or coarser DEMs that cover small areas.

Selecting which combination of derivatives to compute

FormMapR is set up to compute three different sets of output that comprise different subsets and combinations of the full range of possible terrain derivatives that are available. The user is required to identify which output subset is required to support which subsequent application.

There are several reasons for offering three different choices. Firstly, the FormMapR program has, to date, been used mainly to compute terrain derivatives for use as input into two different classification applications. These are the original LandMapR landform segmentation model (LSM) and a more recent Predictive Ecosystem Mapping (PEM) application in British Columbia, Canada. These two applications use different subsets of the total range of terrain derivatives that are presently available for computation by the full FormMapR program. A decision was taken to not compute those terrain derivatives that were not absolutely necessary for use by the two main applications. This approach saves considerable disk space by not having to store data for variables that are not used in a subsequent classification procedure. It also reduces memory requirements and speeds up the subsequent classification procedures as they are not required to read in data for terrain derivatives that they subsequently do not use. Avoidance of computing, storing and reading un-necessary data values can result in considerable savings of time and disk storage when processing large files containing several millions of cells.

A second reason for having different options is that there is a small, but significant, difference in the values computed for diffuse upslope area and compound topographic index (wetness index) in the output produced for the two main applications. When the initial LandMapR landform segmentation program (LSM) was developed diffuse upslope area was computed and recorded as a count of numbers of upslope cells that could contribute runoff to a down-slope cell according to the multiple descent algorithm of Quinn et al., (1991, 1995). Diffuse upslope area was reported in terms of relative cell count units and not absolute area. The rules developed to classify landforms using the LandMapR LSM procedures referenced diffuse upslope area reported in terms of these cell counts. When the BC PEM classification procedures were developed, it was decided to revise the calculations of diffuse upslope area and wetness index so that the results were reported in terms of absolute upslope area in square meters. Thus, the two applications reference different versions of the variables computed for wetness index and diffuse upslope area. Values computed in terms of cell units for use in the initial LandMapR landform classification procedures will not properly support the BC PEM application and vice versa. It is therefore necessary to maintain separate options within the FormMapR program that compute and report wetness index and diffuse upslope area in terms of either relative cell counts or absolute upslope area.

Users are cautioned that it is very important that they make the correct selection and compute the correct suite of terrain derivatives needed to support whichever subsequent classification application they propose to apply. Failure to select the proper output subset will result in an incorrect subset of variables being available for use by the subsequent classification program and with incorrect values for diffuse upslope area and wetness index that are not in synch with the classification rules being used.

The third option is to compute all possible terrain derivatives currently available for computation by the FormMapR program. This option will result in calculation and storage of all terrain derivatives that the FormMapR program can currently compute. Users are advised that diffuse upslope area calculations produced by this option are reported in terms of absolute upslope area in square meters. The values for wetness index and diffuse upslope area are therefore not compatible with the rule bases currently used to classify landform facets using the LandMapR LSM landform classification procedures. A small change to the LSM rule bases would permit these values to be used as input to the LSM classification procedures.

The third option produces a third file of output data that is not produced by either of the other two options. This third output file (ID#Len) contains data on absolute and relative slope lengths that is not computed or stored by the other two options.

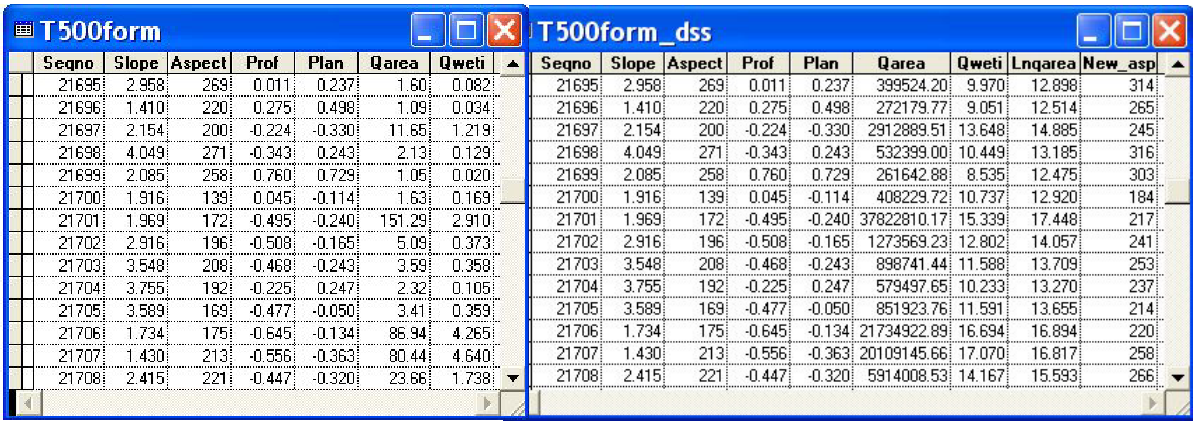
FormMapR Output Files

FlowMapR creates 3 different DBF output files (Table 2). Two of the files (ID#Form & ID#RelZ) are produced by all three options while one file (ID#Len) is only produced by option three.

Table 2. Listing and description of all DBF output files produced by the FormMapR program

File	Description of contents
ID#Form	This file contains data that describe the shape and form of the terrain surface as well as the spatial distribution of a relative moisture index. It contains data for slope gradient and aspect, profile and plan curvature, diffuse upslope area and wetness index or compound topographic index as per Quinn et al., (1991).
ID#RelZ	This file contains data on a number of custom measures of absolute relief and relative slope position measured as vertical distance from each cell to the closest cell to which it is connected by a defined flow path that is classified as pit or peak, channel or divide cell.
ID#Len	This file contains data on a number of custom measures of absolute slope length and relative slope position measured as horizontal distance from each cell to the closest cell to which it is connected by a defined flow path that is classified as pit or peak, channel or divide cell.

The ID#Form Output File


T500form

Seqno	Slope	Aspect	Prof	Plan	Qarea	Qweti
21695	2.958	269	0.011	0.237	1.60	0.082
21696	1.410	220	0.275	0.498	1.09	0.034
21697	2.154	200	-0.224	-0.330	11.65	1.219
21698	4.049	271	-0.343	0.243	2.13	0.129
21699	2.085	258	0.760	0.729	1.05	0.020
21700	1.916	139	0.045	-0.114	1.63	0.169
21701	1.969	172	-0.495	-0.240	151.29	2.910
21702	2.916	196	-0.508	-0.165	5.09	0.373
21703	3.548	208	-0.468	-0.243	3.59	0.358
21704	3.755	192	-0.225	0.247	2.32	0.105
21705	3.589	169	-0.477	-0.050	3.41	0.359
21706	1.734	175	-0.645	-0.134	86.94	4.265
21707	1.430	213	-0.556	-0.363	80.44	4.640
21708	2.415	221	-0.447	-0.320	23.66	1.738

T500form_dss

Seqno	Slope	Aspect	Prof	Plan	Qarea	Qweti	Lnqarea	New_asp
21695	2.958	269	0.011	0.237	399524.20	9.970	12.898	314
21696	1.410	220	0.275	0.498	272179.77	9.051	12.514	265
21697	2.154	200	-0.224	-0.330	2912889.51	13.648	14.885	245
21698	4.049	271	-0.343	0.243	532399.00	10.449	13.185	316
21699	2.085	258	0.760	0.729	261642.88	8.535	12.475	303
21700	1.916	139	0.045	-0.114	408229.72	10.737	12.920	184
21701	1.969	172	-0.495	-0.240	37822810.17	15.339	17.448	217
21702	2.916	196	-0.508	-0.165	1273569.23	12.802	14.057	241
21703	3.548	208	-0.468	-0.243	898741.44	11.588	13.709	253
21704	3.755	192	-0.225	0.247	579497.65	10.233	13.270	237
21705	3.589	169	-0.477	-0.050	851923.76	11.591	13.655	214
21706	1.734	175	-0.645	-0.134	21734922.89	16.694	16.894	220
21707	1.430	213	-0.556	-0.363	20109145.66	17.070	16.817	258
21708	2.415	221	-0.447	-0.320	5914008.53	14.167	15.593	266

Figure 24. Illustration of the format and content of the ID#Form DBF Tables produced by options a (left) and b (right)

FormMapR uses equations reported by Eyton (1991) to compute slope gradient (Slope), aspect (Aspect), profile (Prof) and plan (Plan) curvature. Aspect is not presently used by the default LSM classification rules, but the derivative is still computed and stored. Alternative algorithms for computing slope and curvature (Martz and de Jong, 1987; Zevenbergen and Thorne, 1987) were initially investigated but it was eventually concluded that Eyton's (1991) algorithms were more robust and more suitable for use in the landform classification procedures

FormMapR uses the multiple descent flow accumulation algorithm of Quinn et al., (1991) to compute diffuse upslope area (Qarea) and wetness index (Qweti). The algorithm of Quinn et al., (1991) differs from a conventional D8 algorithm in that drainage is allowed to flow from a given cell to all neighbor cells that are down-slope of it. The total amount of drainage from a cell to its down-slope neighbors is partitioned in proportion to the steepness of the slope from the cell to each of its neighbors. In this algorithm, more of the total upslope area count is placed into neighbors into which the slope is steep and less into neighbors with almost the same elevation as the cell being drained. Wetness index is the familiar $\ln(\alpha/\tan\beta)$ index. It is computed as simply the log of the diffuse upslope area count divided by the tan of the slope. This calculation reflects the assumption that a cell with a high upslope area but a steep slope will pass on more of the in-flow it receives than will a cell with equivalent upslope area but a low slope gradient (flat) where water is more likely to stand for some period. The wetness index derivative provides a useful measure of the relative likelihood of a cell being wetter or drier than normal, due to surface and near surface runoff from positions above it and higher in the landscape.

Figure 24 illustrates the similarities and differences in content between the ID#Form file produced using option a (derivatives required for the LSM classification) and options b and c (derivatives required for the BC PEM DSS classification and for the full set of derivatives). Values for slope, aspect, profile and plan curvature are identical in both sets of output. The values computed for diffuse upslope area (Qarea) and wetness index (Qweti) are much larger in the table on the right produced by selecting options b or c. This is because these options compute diffuse upslope area in terms of absolute area in square meters and not in terms of grid cell units as reported by the table on the left. Options b and c compute and store data for an additional field named New_asp, for new aspect. This is simply the value for aspect rotated 45° counter-clockwise to achieve an effect equivalent to having 0° and 360° oriented pointing NW (e.g. 315°) with true NE (45°) having a re-scaled value of 90°. This re-orientation of aspect was done to facilitate direct application of fuzzy logic equations to data for aspect to compute values of likelihood of having a NE or SW orientation. This version of the ID#Form table also computes and stores a value for log of diffuse upslope area as the transformed value is more suitable for display as a grid map.

The ID#RelZ Output File

T500relz_lsm

Seqno	Z2st	Z2cr	Z2pit	Z2peak	Z2top	Zcr2st	Zpit2peak	Ztop2pit	Pctz2st	Pctz2pit	Pctz2top	Pmin2max
21695	15.042	90.185	524.416	189.667	1062.874	105.227	714.083	1587.290	14	73	33	46
21696	15.649	75.654	538.947	175.136	1048.343	91.303	714.083	1587.290	17	75	33	46
21697	10.154	81.149	533.452	180.631	1053.838	91.303	714.083	1587.290	11	74	33	46
21698	22.931	68.372	546.229	167.854	1041.061	91.303	714.083	1587.290	25	76	34	47
21699	50.642	40.661	573.940	140.143	1013.350	91.303	714.083	1587.290	55	80	36	48
21700	25.462	47.983	566.618	147.465	1020.672	73.445	714.083	1587.290	34	79	35	48
21701	0.000	53.194	561.407	152.676	1025.883	53.194	714.083	1587.290	0	78	35	48
21702	3.895	50.568	564.033	150.050	1023.257	54.463	714.083	1587.290	7	78	35	48
21703	9.423	45.040	569.561	144.522	1017.729	54.463	714.083	1587.290	17	79	35	48
21704	12.987	57.305	580.633	133.450	1006.657	70.292	714.083	1587.290	18	81	36	49
21705	7.963	60.866	577.072	137.011	1010.218	68.829	714.083	1587.290	11	80	36	48
21706	4.882	57.281	573.991	140.092	1013.299	62.163	714.083	1587.290	7	80	36	48
21707	6.435	55.728	575.544	138.539	1011.746	62.163	714.083	1587.290	10	80	36	48
21708	12.667	66.456	581.776	132.307	1005.514	79.123	714.083	1587.290	16	81	36	49

Figure 25. Illustration of the format and content of the ID#RelZ DBF Table produced by option a (LSM Derivatives only)

Each of the three options provided by the FormMapR program generates a different version of the ID#RelZ file. The first option to "compute only those derivatives required to run the LSM classification program" produces the ID#RelZ output file illustrated in Figure 25. This is the option that most current users of the LandMapR suite of programs will most often be interested in and will most often select.

Custom algorithms are used in FormMapR to compute a large number of measures of absolute and relative relief, slope length and relative slope position. These algorithms have some similarities to procedures previously described by Skidmore (1990) but also possess some features not described or available in the Skidmore approach (Figures 26 & 27).

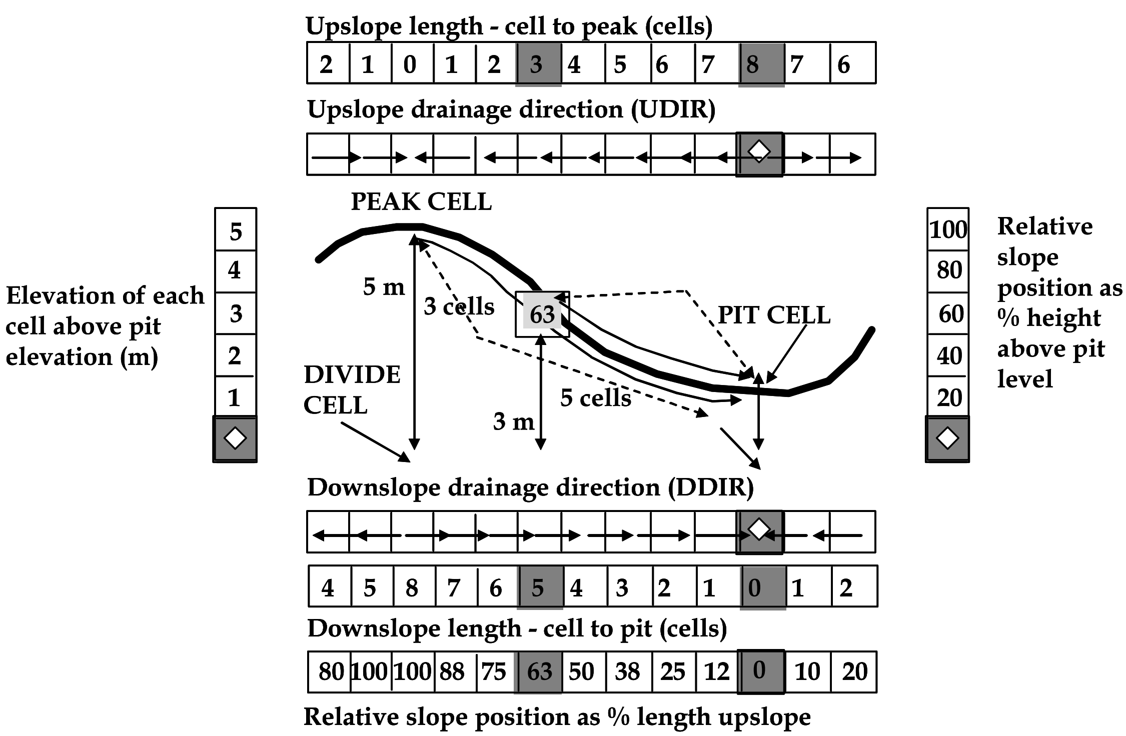


Figure 26. Illustration of the procedures used to compute various measures of relief, slope length and slope position

As illustrated in Figures 26 and 27, the algorithm traces down-slope from each cell along a defined flow path until it reaches a pit centre cell identified by a drainage direction value of 5. Along the way, it records the location and elevation of the first cell along the flow path that is classified as a channel cell. The first channel cell is the first cell encountered along the flow path that has a value for upslope area equal to or greater than the user-selected threshold value for identifying channels. The algorithm records the location of both the first channel cell and the pit cell in terms of row and column coordinates and the elevation of each target cell in meters. The algorithm returns to the initial start cell and traverses the flow path a second time. As it re-traverses the flow path it computes the difference in elevation between each cell along the flow path and the currently stored value for elevation of the nearest channel and pit cell. It also computes the horizontal distance from each cell to the stored locations of the nearest channel and pit cells. It also computes the length of the flow path from each grid cell to the nearest channel and pit cell in terms of total number of cells along the flow path from each cell to its associated channel and pit cell. These data on path length in cells is only written out for options b and c.

The column labeled Z2St in Figure 25 stores a value for a variable that describes the vertical elevation difference (or relief) between a given cell and the first channel cell to which it is connected by a defined flow path. Similarly, Z2Pit is the vertical change in elevation (relief) between each cell and the local (first order) pit cell at which flow would terminate if the pit were not removed.

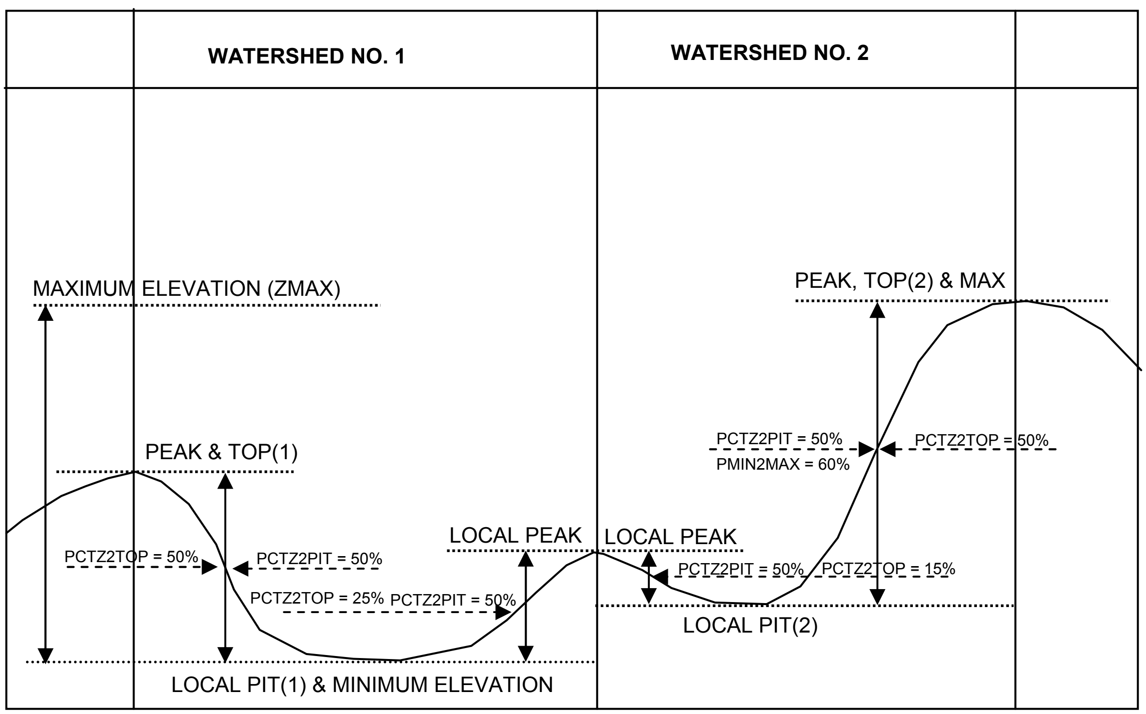


Figure 27. Illustration of the concept of relative relief as a function of elevation distance to significant landscape tie points

After completing tracing down-slope to identify vertical and horizontal distances to cells classified as channel and pit cells, the process is repeated for notional upslope flow. The algorithm traces upslope along a path of steepest ascent until it reaches a peak cell. As with the previously described down-slope flow algorithm, the location and elevation of the first cell classified as a divide cell that is passed through along the way is also recorded. Once the locations and elevations of these significant tie points have been determined for each cell it is possible to compute both the elevation difference (relief) and the horizontal distance between each cell and the identified divide and peak cells to which it is connected by a defined path of steepest ascent.

The column labeled Z2Cr in Figure 25 stores a value for a variable that describes the vertical elevation difference (or relief) between a given cell and the first divide (or crest) cell to which it is connected by a defined flow path. Similarly, Z2Peak is the vertical change in elevation (relief) between each cell and the local peak cell (e.g. a pit in the inverted DEM) at which up-slope flow terminates.

A number of measures of absolute and relative relief, slope length and slope position can be computed by reference to the previously computed values for horizontal and vertical distance from each cell to the respective pit, peak, channel and divide cells to which the cell is connected by a defined flow path.

One measure of absolute relief is the total change in elevation from pit to peak of the flow path that runs through a given cell (Zpit2Peak). A similar variable can be computed to express the change in elevation from divide to channel (Zcr2st) of the flow path that runs through a given cell. A third tie point not previously described is the elevation of the highest cell in a catchment. Since a catchment can possess a number of different local peaks, but only one local pit, total relief can also be expressed as the total change in elevation from the highest to lowest point in a given catchment to which a cell belongs (Ztop2Pit).

As with vertical relief, similar measures of the total length of a flow path through a cell from pit to peak (LPit2Peak) or divide to channel (Lst2Div) can also be computed. Flow distance can also be expresses as total length of the flow path in cells, rather than as total horizontal (as the crow flies) distance in meters.

The measures of absolute relief and slope length described above are then combined to compute a number of useful measures of relative relief and relative slope position. The variable PctZ2Pit provides an estimate of the percent vertical distance that a cell is upslope relative to the total change in elevation between the pit and peak to which it is connected to via defined flow paths. Similarly PctZ2St expresses the relative relief of a cell with respect to the nearest divide and channel cells to which it is connected by a defined flow path. PctL2Pit and PctL2St provide equivalent measures expressed in terms length upslope relative to the nearest pit and peak or channel and divide to which a cell is connected by a defined flow path.

Experience working with these measures of relative relief and relative slope length has led to the conclusion that the vertical measures of PctZ2St and PctZ2Pit provide superior measures of relative slope position than do the equivalent horizontal measures of PctL2St and PctL2Pit.

Finally, the FlowMapR program also computes relative relief with respect to the highest and lowest elevation in the watershed in which the cell is located (PctZ2Top) and with respect to the highest and lowest elevation in the entire DEM data set (PMin2Max).

These multiple measures of relative relief and relative slope position provide very useful data for establishing the contextual position of each cell in a landscape. They can act as a kind of glue, enforcing a certain amount of spatial continuity and cohesion onto any resulting landform classification.

To date, none of the classification approaches have made use of all of the various measures of absolute and relative relief, slope length and landform position, but they can all be computed and they can all be recorded and stored if the third option (compute a full set of all possible derivatives) is selected.

The DBF table produced when the third option is selected (compute a full set of all possible derivatives) contains all currently computed measures of absolute and relative relief. It is rather large and unwieldy and not easily illustrated as a single table. Figure 28 illustrates, as two separate halves, the entire contents of ID#RelZ table produced by selecting the third option. This table stores a great deal of data, some of which is redundant and unnecessary. The structure of the full table reflects the early algorithm development history when row and column locations and elevations were explicitly stored for each cell for the pit, peak, channel and divide cells to which each cell was connected by a defined flow path. These data were initially used to check and debug the algorithm but no longer have any valid function.

T500relz_all

Seqno	St_row	St_col	St_elev	Z2st	N2st	Cr_row	Cr_col	Cr_elev	Z2cr	N2cr	Pit_row	Pit_col	Pit_elev	Z2pit	Pk_row	Pk_col	Pk_elev
21695	71	204	1147.09300	15.042	1	69	214	1252.320	90.185	9	138	130	637.719	524.416	69	234	1351.802
21696	72	206	1161.01700	15.649	1	69	214	1252.320	75.654	8	138	130	637.719	538.947	69	234	1351.802
21697	72	206	1161.01700	10.154	1	69	214	1252.320	81.149	8	138	130	637.719	533.452	69	234	1351.802
21698	72	206	1161.01700	22.931	2	69	214	1252.320	68.372	6	138	130	637.719	546.229	69	234	1351.802
21699	72	206	1161.01700	50.642	3	69	214	1252.320	40.661	5	138	130	637.719	573.940	69	234	1351.802
21700	72	208	1178.87500	25.462	2	69	214	1252.320	47.983	4	138	130	637.719	566.618	69	234	1351.802
21701	71	211	1199.12600	0.000	0	69	214	1252.320	53.194	4	138	130	637.719	561.407	69	234	1351.802
21702	72	212	1197.85700	3.895	1	69	214	1252.320	50.568	4	138	130	637.719	564.033	69	234	1351.802
21703	72	212	1197.85700	9.423	1	69	214	1252.320	45.040	2	138	130	637.719	569.561	69	234	1351.802
21704	72	214	1205.36500	12.987	1	68	215	1275.657	57.305	3	138	130	637.719	580.633	69	234	1351.802
21705	72	215	1206.82800	7.963	1	68	215	1275.657	60.866	3	138	130	637.719	577.072	69	234	1351.802
21706	72	215	1206.82800	4.882	1	68	216	1268.991	57.281	3	138	130	637.719	573.991	69	234	1351.802
21707	72	215	1206.82800	6.435	2	68	216	1268.991	55.728	3	138	130	637.719	575.544	69	234	1351.802
21708	72	215	1206.82800	12.667	3	68	220	1285.951	66.456	3	138	130	637.719	581.776	69	234	1351.802

T500relz_all

Seqno	Pk_col	Pk_elev	Z2peak	N2peak	Z2top	Zpit2peak	Zcr2st	Zcr2pit	Ztop2pit	Ncr2st	Pmin2max	Pctz2st	Pctz2pit	Pctz2top	Pctn2st
21695	234	1351.802	189.667	30	1062.874	714.083	105.227	614.601	1587.290	10.000	46	14	73	33	10
21696	234	1351.802	175.136	29	1048.343	714.083	91.303	614.601	1587.290	9.000	46	17	75	33	11
21697	234	1351.802	180.631	29	1053.838	714.083	91.303	614.601	1587.290	9.000	46	11	74	33	11
21698	234	1351.802	167.854	27	1041.061	714.083	91.303	614.601	1587.290	8.000	47	25	76	34	25
21699	234	1351.802	140.143	26	1013.350	714.083	91.303	614.601	1587.290	8.000	48	55	80	36	37
21700	234	1351.802	147.465	25	1020.672	714.083	73.445	614.601	1587.290	6.000	48	34	79	35	33
21701	234	1351.802	152.676	25	1025.883	714.083	53.194	614.601	1587.290	4.000	48	0	78	35	0
21702	234	1351.802	150.050	25	1023.257	714.083	54.463	614.601	1587.290	5.000	48	7	78	35	20
21703	234	1351.802	144.522	23	1017.729	714.083	54.463	614.601	1587.290	3.000	48	17	79	35	33
21704	234	1351.802	133.450	23	1006.657	714.083	70.292	637.938	1587.290	4.000	49	18	81	36	25
21705	234	1351.802	137.011	23	1010.218	714.083	68.829	637.938	1587.290	4.000	48	11	80	36	25
21706	234	1351.802	140.092	22	1013.299	714.083	62.163	631.272	1587.290	4.000	48	7	80	36	25
21707	234	1351.802	138.539	22	1011.746	714.083	62.163	631.272	1587.290	5.000	48	10	80	36	40
21708	234	1351.802	132.307	18	1005.514	714.083	79.123	648.232	1587.290	6.000	49	16	81	36	50

Figure 28. Illustration of the format and content of the ID#RelZ DBF Table produced by option c (Compute all Derivatives)

The procedures used to compute the various measures of absolute and relative relief and slope length are perhaps most easily understood by examining the table produced when the option to compute all possible derivatives is selected (Figure 28). This table explicitly stores the row and column locations and the elevation of the key tie point cells (pits, peaks, channels and divides) to which each cell is connected by a set of defined down-slope and up-slope flow paths. It should be fairly clear that, knowing the location and elevation of each grid cell (not shown in Figure 28) it is relatively easy and straight forward to compute the vertical difference in elevation as well as the horizontal distance from each cell to the pit, peak, channel and divide cells to which it is connected.

The ID#Len Output File

T500len_all

Seqno	L2pit	L2peak	Lpit2peak	Ppit2peakl	L2str	L2div	Lstr2div	Pstr2divl
21695	266986.0	95940.8	362926.9	73	7537.6	94945.5	102483.1	7
21696	274193.8	88685.7	362879.5	75	7840.5	87665.0	95505.5	8
21697	271564.5	91324.4	362888.8	74	5126.0	90388.8	95514.8	5
21698	277911.5	84933.7	362845.2	76	11519.9	83986.6	95506.4	12
21699	291606.4	71184.7	362791.1	80	25370.3	70123.2	95493.5	26
21700	288073.3	74709.3	362782.6	79	12780.0	73766.4	86546.4	14
21701	285581.8	77205.8	362787.6	78	0.0	76359.3	76359.3	0
21702	286943.5	75833.7	362777.2	79	2010.7	75038.3	77049.0	2
21703	289731.7	73026.7	362758.4	79	4764.3	72269.6	77033.9	6
21704	295245.5	67477.6	362723.1	81	6512.7	66743.7	73256.5	8
21705	293566.9	69168.3	362735.2	80	4012.8	68521.9	72534.7	5
21706	292126.1	70628.9	362755.0	80	2541.4	70062.1	72603.4	3
21707	292962.8	69796.2	362759.1	80	3406.2	69287.5	72693.8	4
21708	296098.1	66643.0	362741.1	81	6527.9	66178.1	72705.9	8

Figure 29. Illustration of the format and content of the ID#Len DBF Table produced by option c (Compute all Derivatives)

The DBF table ID#Len (Figure 29) is produced by processing the full ID#RelZ file (Figure 28) when the option to compute all possible terrain derivatives is selected. The location of each grid cell in row and column coordinates is used to compute the horizontal distance from the cell to the cells identified in Table 28 as being the closest pit, peak, channel and divide cells. The total pit to peak (LPit2Peak) and channel to divide (LStr2Div) distances are computed by summing the individual pairs of distances (e.g. L2Pit + L2Peak = Lpit2peak). The measure of relative slope position in terms of slope length (Pstr2div) is computed, for example, by dividing the length from a cell to a channel (L2Srt) by the total channel to divide length (LStr2Div) and multiplying by 100 to express as a percent. Percent pit to peak length (Ppit2peakL) is computed in a similar way.

Using and visualizing the FormMapR Output Data

New users are encouraged to produce grid maps depicting most of the terrain derivatives computed by the FormMapR program in order to increase their appreciation and understanding of these variables (see Figure 30).

Grid maps of most of the terrain derivatives computed by FormMapR have regularly been displayed and visually interpreted to provide guidance on which variables appeared to have the greatest potential for achieving a desired classification and what threshold values were most appropriate to use if the variables were selected for use in a classification rule base.

Users are encouraged to experiment with using different combinations of the terrain derivatives computed by FormMapR to create their own classification schemes with their own input variables and rule bases. Visual review of the available terrain derivatives is an important step in devising and implementing new or revised classification rules.

Reformatting grid data from columns in DBF tables into ArcView GRID files

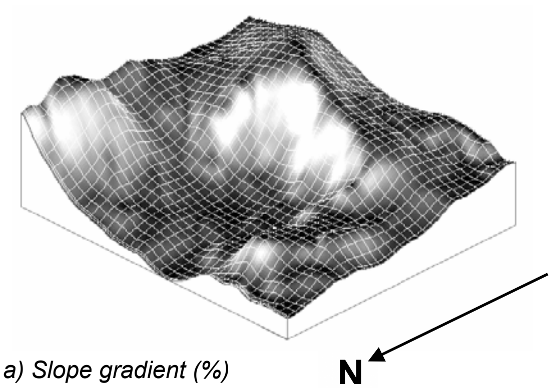
The GridReadWriteUtility can be used to convert data stored in any of the DBF tables produced by FormMapR into binary grid files in the FLT import format used by ArcView 3.2. The FLT file can be easily imported into ArcView 3.2 using the import functionality in ArcView 3.2.

With the exception of differences in the ASCII header file, the ArcView binary FLT file is identical to binary RST or IMG files used by Idrisi 32 so the GridReadWriteUtility can also be used to reformat DBF format data for display in Idrisi.

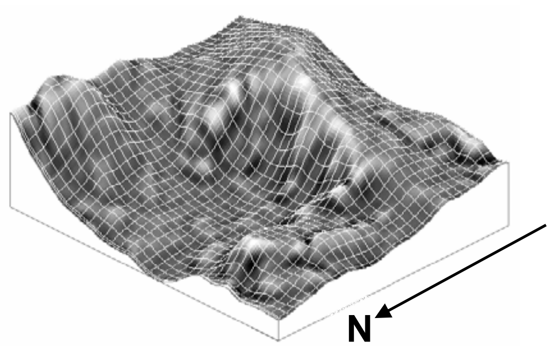
If the GridReadWriteUtility is used to convert an integer value in a DBF table into an FLT file for import into ArcView 3.2, the ArcView grid file that is produced will always treat the imported data as real numbers. It is therefore necessary to convert the imported grid from a real to an integer representation. This can be done by using the INT function in the ArcView 3.2 calculator to calculate and save an integer version of the initial real grid map. This represents an unwelcome extra step in importing integer data from a DBF table via the GridReadWriteUtility, but it is necessary if the imported grid data are to be properly displayed and used in ArcView 3.2.

Users of the original FoxPro versions of the LandMapR programs will be familiar with a FoxPro utility named Loc_2002 which was used to produce a DBF file named ID#_Loc. This location file computed and stored x and y coordinates in the relevant reference system linked to the unique sequence number (SEQNO) that appears in each DBF grid file produced by the LandMapR programs. This X,Y, SEQNO file could be linked in ArcView 3.2 to any of the DBF tables produced by the LandMapR programs (including FormMapR) to create a geo-registered ArcView file of vector point data. The ArcView function Save-as-Grid could then be used to save any column of data in the DBF table as an ArcView grid map.

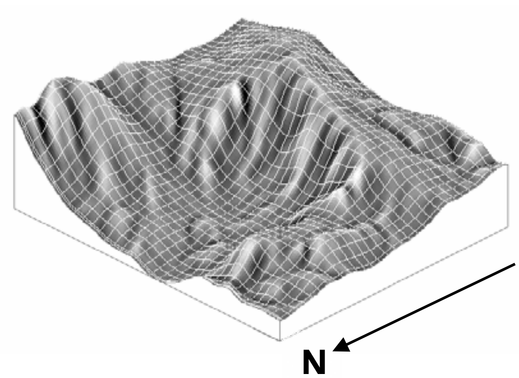
There is, as yet, no C++ equivalent of the FoxPro Loc_2002 program that created the ID#_loc file required to import DBF data into ArcView in this fashion. Users with FoxPro can continue to use the FoxPro version of the program to accomplish this task. A C++ implementation will be prepared if it proves to be necessary and desirable.



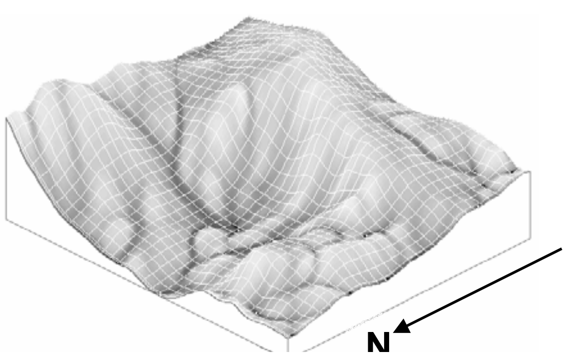
a) Slope gradient (%)



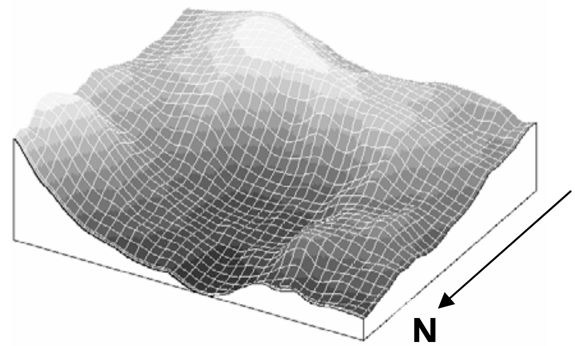
b) Profile (down-slope) curvature (deg/100 m)



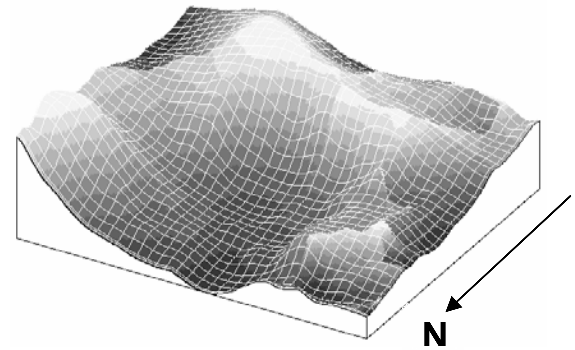
c) Plan (across-slope) curvature (deg/100 m)



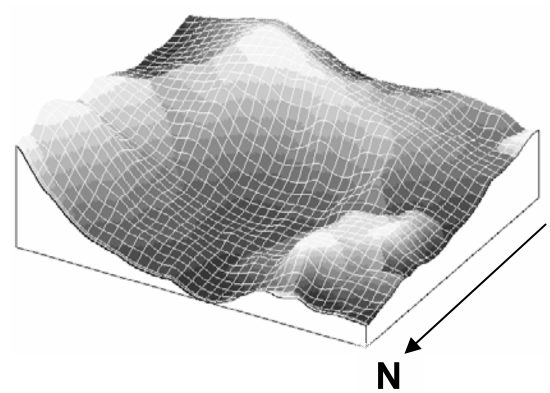
d) QWETI (wetness index as per Quinn et al., 1991)



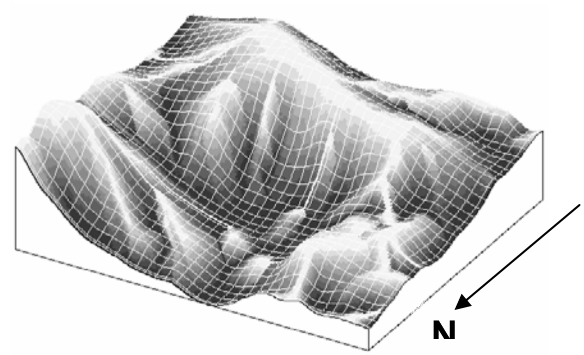
e) PMIN2MAX (Relief relative to min and max elev.)



f) PCTZ2TOP (Relief relative to watershed min and max)



g) PCTZ2PIT (Relief relative to local pit & peak elev.)



h) PCTZ2ST (Relief relative to nearest channel & divide)

Figure 30. 3D Illustrations of 8 of the terrain derivatives computed by FormMapR (source: MacMillan et al., 2000)

FacetMapR

Purpose

FacetMapR is a custom program written expressly to facilitate automated classification of landform, ecological or soil spatial entities using heuristic fuzzy logic rule bases.

The FacetMapR program reads in two control files that instruct it regarding what classes to define and what criteria to use to define each class. Neither the classes defined, nor the input variables used to define them are hard coded into the program. Therefore any kind or number of available input data sets can be used to define any number of user specified classes.

New users are likely to start by applying existing rule files to produce results based on existing classification rules. As users become more familiar with the program and how it defines and applies classification rules, they may wish to become more adventurous and to revise existing rule bases or define entirely new rule bases appropriate to their specific needs.

Input Data

The FacetMapR component mainly uses output from FormMapR to perform its calculations. At present, the options that compute the original LSM landform facets use as inputs only data computed from a single DEM input data set using the FormMapR program. The option used to compute ecological spatial entities for Predictive Ecosystem Mapping (PEM) in BC is set up to use other input data sets not computed from the input DEM data and, in fact, it requires several obligatory non-DEM data sets to be present or it will not run.

FacetMapR will not work properly unless FormMapR has been run successfully to completion prior to running FacetMapR and unless all the required files produced by FormMapR are present in the default folder that is identified as containing the source input data.

Users must ensure that they have run FormMapR with the correct option selected. If, for example, they were to have run FormMapR with the option to compute the variables needed to run the LSM classification selected and they were to subsequently try to run FacetMapR with the option to apply the PEM DSS classification approach selected, the FacetMapR program would fail. It would not be able to locate all of the input variables and input files that it requires to run this option. Similarly, the options to apply either the original LSM classification procedures or to apply a revised and condensed version of the original LSM classification procedures both require FormMapR to have been previously run using the option to compute only those variables required by the LSM procedures.

FacetMapR also needs to have access to previously prepared files that contain the rules that are used to determine what output classes to predict and what input variables to use to predict these output classes. The FacetMapR options that apply the original and condensed LSM landform classification procedures look for and need two (2) input files that provide the rules used to define fuzzy landform attributes and fuzzy landform classes. By convention these usually have names like LM3arule and LM3crule, but there is no enforced naming convention for these files.

The FacetMapR option that applies the new BC PEM Direct-to-Site-Series (DSS) classification procedures looks for and requires a large number of paired files that define fuzzy attributes and fuzzy classifications for each and every zone or region within the overall area for which different rules are required and defined. The DSS option in FacetMapR reads in and uses a DBF grid map that identifies each zone within an overall area for which different rules are meant to apply. Each zone has a unique integer ID number. FacetMapR then looks for and reads in a fuzzy attribute rule file with the name AruleXXXX (where XXXX is the unique integer zone ID) and a corresponding fuzzy classification rule file with the name CruleXXXX (where XXXX is the unique integer zone ID). If it does not find a correctly named pair of rule files for each unique integer zone in the DBF zone file, FacetMapR will not work properly and will terminate without completing its processing of the data. This naming convention is absolute and strictly enforced.

Operation of the FacetMapR Program

Once all required input data sets have been prepared, actual operation of the FacetMapR program is very simple and requires users to supply only a very limited amount of interactive input. The FacetMapR interface is illustrated in Figure 31.

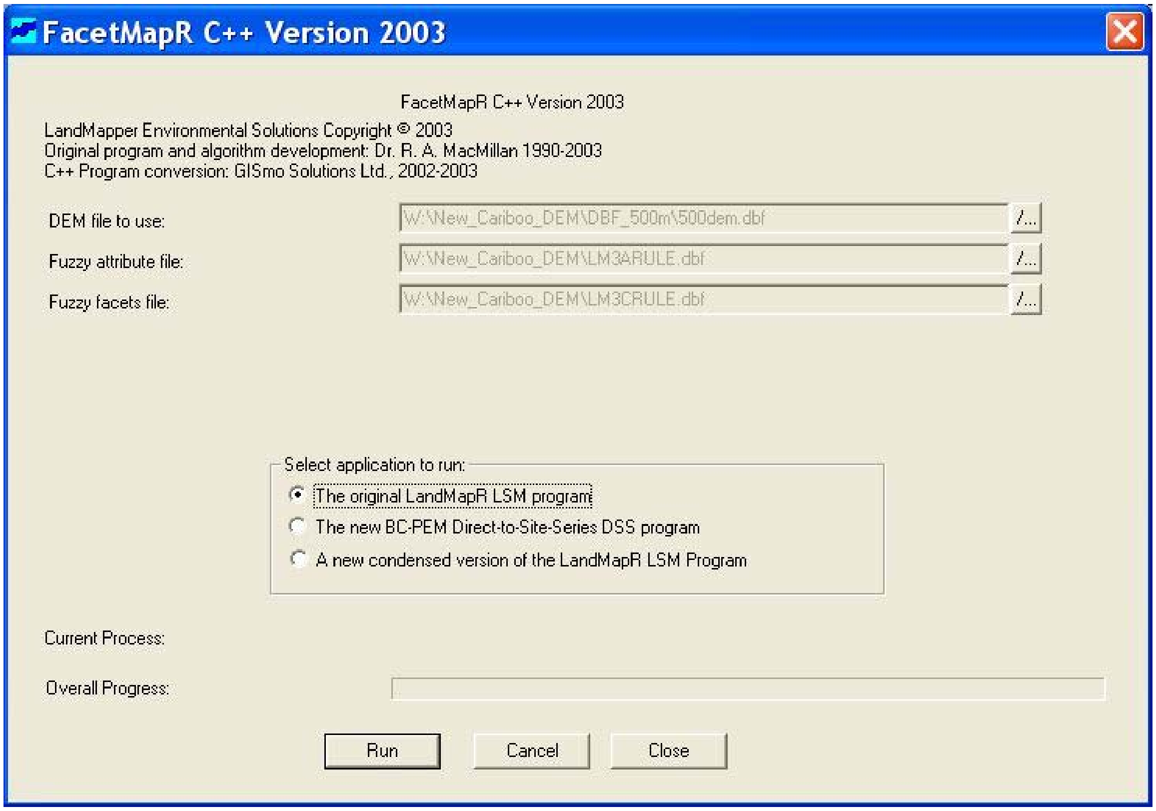


Figure 31. Illustration of the FacetMapR interface

Identifying the input data location and working folder

First, the user must navigate to the folder that contains all of the required input data and must select the ID#DEM file that is associated with the site to be processed. This ID#DEM file must have been produced by a successful application of the FlowMapR program. All necessary files produced by the FormMapR program (e.g. ID#Form, ID#RelZ) must also be present in the same folder as the selected ID#DEM file. The folder that you select here will automatically become the default folder into which all output from FacetMapR will be placed.

Identifying the name and location of the fuzzy attribute rule file

Users must identify the name and location of the DBF table that contains the rules required to define fuzzy attributes. This DBF table can have any name and be in any location as long as it retains the structure defined for an attribute rule table and as long as it references attributes that occur as columns in one of the DBF tables produced by the FormMapR program. Identification of a fuzzy attribute rule file is only required for options a (The original LandMapR LSM program) and c (A new condensed version of the LandMapR LSM program). If option b (The new BC-PEM Direct-to-Site_Series DSS program) is selected, this navigation box will be disabled and grayed out as the DSS procedures do not require users to identify the name of the DBF file(s) that are to be used to define fuzzy attributes. New users are supplied with two fuzzy attribute rule files (LM0Arule & LM3arule) that have been widely tested and applied and that represent the current standard for applying the original LandMapR LSM landform classification procedures.

Identifying the name and location of the fuzzy classification rule file

Users must identify the name and location of the DBF table that contains the rules required to define fuzzy landform facets (or any other desired output classification). This DBF table can have any name and be in any location as long as it retains the structure defined for a fuzzy classification rule table and as long as it references fuzzy attributes that are defined in the corresponding fuzzy attribute rule table and that can be computed using data that is present in available DBF tables.

Identification of a fuzzy classification rule file is only required for options a (The original LandMapR LSM program) and c (A new condensed version of the LandMapR LSM program). If option b (The new BC-PEM Direct-to-Site_Series DSS program) is selected, this navigation box will be disabled and grayed out as the DSS procedures do not require users to identify the name of the DBF file(s) that are to be used to define fuzzy ecological classes. New users are supplied with at two fuzzy classification rule files (LM0arule & LM3crule) that have been widely tested and applied and that represent the current standard for applying the original LandMapR LSM landform classification procedures.

Selecting which classification procedure (option) to run

FacetMapR is set up to compute three different sets of output that implement three different classification approaches. The user is required to identify which classification approach is desired and to select the appropriate option. Three options are presently available.

The first option (Option a) runs the LandMapR LSM program exactly as it was originally devised and set up in its original FoxPro implementation (MacMillan et al., 1997a,b; 1998; 2000). This option produces two large files that contain data for every fuzzy attribute computed and for every possible fuzzy classification for every grid cell in a matrix.

As the size of the areas to which the LSM procedures were applied increased it became evident that storage of all this unused data was wasteful and un-necessary. An alternative implementation of the LSM procedures was devised (Option c) that only stored data on the final classification results for each grid cell and did not store all of the intermediate calculations.

The second option (Option b) runs the new BC Direct-to-Site-Series (DSS) programs that apply different rules to different portions of a grid matrix and that also access and use external grid data stored in DBF tables. The external grid data represent data layers or grid coverages that were not produced by analysis of an input DEM data set by the FlowMapR and FormMapR programs but were converted from available secondary source data or were produced expressly to support the PEM DSS classification.

The third option (Option c) runs a new condensed version of the original LandMapR LSM landform classification procedures. This option is functionally equivalent to the original LSM program but it runs faster and uses less disk storage for the output results. This is because it only stores data for the final classification result for each grid cell and does not store data on the intermediate calculations of fuzzy landform attributes or individual fuzzy landform class likelihood values for each grid cell.

This approach saves considerable disk space by not having to store data for intermediate calculations that tend not to be used. It reduces memory requirements and speeds up the classification procedures as they are not required to write out data that are subsequently not used. Avoidance of computing, storing and reading un-necessary data values can result in considerable savings of time and disk storage when processing large files containing several millions of cells.

Most current users are expected to select and run option c (The new condensed version of the LandMapR LSM Program) most of the time. They may wish to run option a (The original LSM program) a few times to gain a better appreciation of how the classification approach works or to ensure that their results are identical to those produced by the original FoxPro LSM program. In general, however, option c will run faster and will produce and record all output data most users will be interested in obtaining and using.

Constructing and modifying rule files

FacetMapR applies fuzzy logic using a two step approach adapted from Burrough et al., 1992.

The first step is to convert numerical input data (both classed and continuous) into integer values that are referred to as "fuzzy attributes". Fuzzy attributes are continuous values scaled from 0 – 100 which express landform attributes in terms of fuzzy semantic constructs such as the degree to which a site (e.g. a grid cell) is considered to be steeply sloping or near a mid-slope. Rules for computing fuzzy attributes are defined in a DBF attribute rule file that, by convention, is always assigned a name that contains the root word "arule". Different prefixes and suffixes are appended to this root to identify different versions of the rules for computing fuzzy attributes.

In the second step, previously computed fuzzy attribute values are converted into continuous numbers scaled from 0-100 which express the likelihood that a given cell or site belongs to each of ***n*** defined output classes. Rules for defining "fuzzy output classes" are stored in a second DBF rule table that, by convention, is always assigned a name that contains the root word "crule". Different prefixes and suffixes are appended to this root to identify different versions of these fuzzy classification rule files.

Converting raw input variables into fuzzy landform attributes

A module in the FacetMapR program named Fuzz_calc converts raw input data values (usually terrain derivatives) into fuzzy landform attributes. The process by which this occurs is documented in MacMillan et al., (2000).

The Fuzz_calc procedure reads the fuzzy attribute rule file (Figure 32) to determine how many fuzzy attributes to compute, what to name them, what variables to use to compute them and what fuzzy models to apply to the input values to convert them into fuzzy attribute values.

Lm3arule

Sortorder	File_in	Attr_in	Class_out	Model_no	B	B_low	B_hi	B1	B2	D
1	formfile	PROF	CONVEX_D	4	5.0	0.0	0.0	2.5	0.0	2.5
2	formfile	PROF	CONCAVE_D	5	-5.0	0.0	0.0	0.0	-2.5	2.5
3	formfile	PROF	PLANAR_D	1	0.0	0.0	0.0	-2.5	2.5	2.5
4	formfile	PLAN	CONVEX_A	4	5.0	0.0	0.0	2.5	0.0	2.5
5	formfile	PLAN	CONCAVE_A	5	-5.0	0.0	0.0	0.0	-2.5	2.5
6	formfile	PLAN	PLANAR_A	1	0.0	0.0	0.0	-2.5	2.5	2.5
7	formfile	QWETI	HIGH_WI	4	7.0	0.0	0.0	3.5	0.0	3.0
8	formfile	QWETI	LOW_WI	5	0.5	0.0	0.0	0.0	3.5	3.0
9	formfile	SLOPE	NEAR_LEVEL	5	0.5	0.0	0.0	0.0	1.0	0.5
10	formfile	SLOPE	REL_STEEP	4	2.0	0.0	0.0	1.0	0.0	1.0
11	relzfile	PCTZ2ST	NEAR_DIV	4	90.0	0.0	0.0	75.0	0.0	15.0
12	relzfile	PCTZ2ST	NEAR_HALF	1	50.0	50.0	50.0	25.0	75.0	25.0
13	relzfile	PCTZ2ST	NEAR_CHAN	5	10.0	0.0	0.0	0.0	25.0	15.0
14	relzfile	PCTZ2PIT	NEAR_PEAK	4	90.0	0.0	0.0	75.0	0.0	15.0
15	relzfile	PCTZ2PIT	NEAR_MID	1	50.0	50.0	50.0	25.0	75.0	25.0
16	relzfile	PCTZ2PIT	NEAR_PIT	5	5.0	0.0	0.0	0.0	10.0	5.0
17	relzfile	Z2PIT	HI_ABOVE	4	2.0	0.0	0.0	1.0	0.0	1.0

Figure 32, Illustration of the fuzzy attribute rule file that is the current standard for computing LSM fuzzy landform facets

The example fuzzy attribute rule file (Figure 32) defines 17 different fuzzy landform attributes based on 7 different terrain derivative inputs. The field File_in tells the program to obtain data for the first 10 attributes from data stored in the formfile (ID#Form) DBF table and to obtain data to compute the next 7 fuzzy attributes from the relzfile (ID#Relz). The field Attr_in tells the program the field name of the column in the source file that contains the input variable to be used to compute the fuzzy attribute currently being defined. The field Class_out tells the program what name to give to the fuzzy attribute currently being defined. The field Model_no tells the program the kind of fuzzy model to use to re-scale the raw input value for each grid cell into a fuzzy likelihood value scaled from 0 to 100. The fields named B, B_low, B_high, B1, B2 and D are parameters used by the fuzzy model calculation to determine how to re-scale the raw input data into fuzzy likelihood values ranging from 0 to 100 (Figure 33).

For example, the fuzzy attribute for likelihood of being convex in the down-slope direction (CONVEX_D) is based on applying fuzzy model number 4 (Figure 33) to the value for the input variable profile curvature (PROF) that is located in the formfile source file in a field named PROF. The key parameter values are b = 5 and D = 2.5. B = 5 indicates that any value of profile curvature equal to or greater than 5 is considered to fully meet the criteria for being considered to be strongly convex in the down-slope direction (likelihood = 100). D = 2.5 is the dispersion index. It indicates that the value for likelihood of being convex in the down-slope direction decreases by ½ to 50 at a value for profile curvature that is 2.5 units less than the value assigned for B. This value works out to be 2.5 (e.g. B(5) – D(2.5) = 2.5).

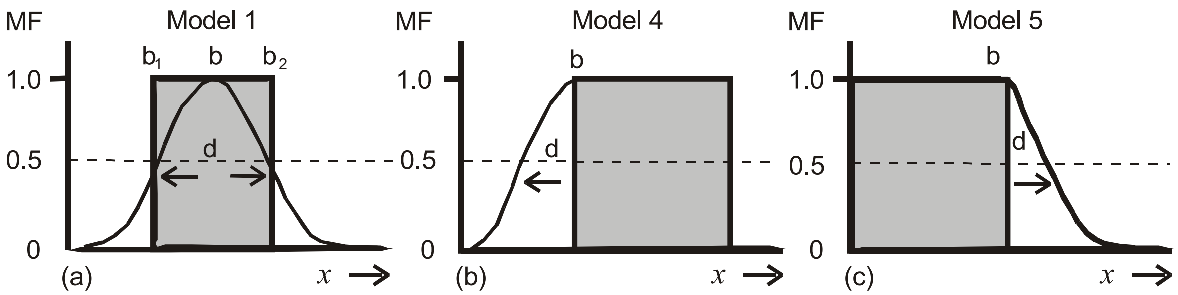


Figure 33. Illustration of the three main models used to compute fuzzy likelihood values from raw input values

The actual calculation used to convert a raw input value (x) into a fuzzy likelihood value for any defined fuzzy attribute is quite simple (see equation 1). The value for B read in from the fuzzy attribute rule file (Figure 32) is subtracted from the raw value for the input variable (X) and divided by the value for dispersion index (D) also read in from the fuzzy attribute rule file. The result is squared and added to 1 to form a denominator that is then divided into 1. For model 4, if X is greater than B then the fuzzy likelihood value is automatically set to 100 and equation 1 is not used. If X is less than B, then the more it differs from B the higher the value that will be computed for $((X\text{-}B)/D)^2$ and the lower the resulting value for fuzzy membership function will be.

$$MF_x = \frac{1}{\left[1+\left(\frac{x-b}{d}\right)^2\right]} * 100 \ldots\ldots\ldots for \ldots 0 \le x \le P \qquad \textbf{(1)}$$

Scaling of the fuzzy landform attribute values from 0 to 100 requires selection of an appropriate model and application of appropriate values for the boundary value or central concept value (***b***) and the dispersion index (***d***) for each terrain derivative. Of the five distinct model types defined by Burrough (1989) for conversion of terrain derivatives into fuzzy landform attributes only types one, four and five have yet been used by any of the landform classification rules defined to date. However, the FacetMapR program will support all 5 model types defined by Burrough (1989) should users wish to use model types two or three.

Model 1 (Figure 33 a) is used for computing attributes such as relative likelihood of being close to a mid-slope landscape position in which the central concept for mid-slope is taken to be a value of 50% for landscape position computed relative to the closest local pit and peak cells. In this model, the relative likelihood of occupying a mid-slope landscape position diminishes in a symmetrical manner as one moves outwards in both directions (up and down) from the central value. Model 4 (Figure 33 b) is applied for attributes such as relatively convex (in profile or plan) or relatively steep slope in which all values of an input variable greater than the specified upper boundary value (***b***) are considered to fully satisfy the requirements for class membership (MF = 100). Similarly, model 5 (Figure 33 c) is applied for all attributes for which some minimum value of input variable exists below which all values are considered to fully satisfy the criteria for class membership.

Expert judgment is generally used to select appropriate values for ***b*** and ***d*** for each fuzzy terrain attribute. If spatially referenced field sample data are available, they can certainly be analyzed to determine if there are natural clusters or breaks in the distribution of input variable values relative to classes of interest. In this way, the heuristic reasoning typically employed to create rule definitions can be replaced, or supplemented with, evidential reasoning based on quantitative analysis of available data.

In theory, values used for boundaries, central concepts (***b***) and dispersion indices (***d***) should be selected with reference to the relevant literature. In practice, however, most of the existing rule bases defined for use with FacetMapR were developed using personal judgments arrived at through successive application and evaluation of different rule bases that used different threshold values to define fuzzy attributes. The criteria used to decide which threshold values produced the most desirable classifications were a combination of visual review and assessment and field studies to document the classes that explained the greatest proportion of the observed variation in stable soil properties (e.g. organic carbon, topsoil depth), crop yields or ecological classes.

The current standard fuzzy attribute rules for use by the LandMapR LSM procedures (Table 32) use quite low threshold values to recognize strong curvatures (5.0 deg/100 m versus the more commonly used threshold value of 10 deg/100 m) and steep slopes (2% versus a more common value of 5%). We have found that the higher threshold values for slope gradient and curvature typically proposed in the literature (see Pennock et al., 1987, 1994) tend to place extensive portions of most typical agricultural landscapes into classes that are deemed to be both planar and level. The more demanding thresholds of Pennock et al., (1987) are not sensitive enough to the relatively low degree of profile and plan curvature and to the relatively low slope gradients that appear to be characteristic of most typical agricultural landscapes. They therefore do not seem to produce the most desirable classifications of landform elements using the fuzzy classification procedures employed by FacetMapR. The current default LandMapR LSM landform classification rule bases reflect our trial and error experience in the relatively low threshold values that they use to differentiate divergent and convergent from planar slopes and gentle from steeper slopes.

The BC PEM Direct-to-Site-Series classification procedures require different fuzzy attribute rule files and fuzzy classification rule files for many different zones or classification regions. The BC-PEM DSS approach also requires an ability to read in and consider data for any number and type of "external" input data layers that are not produced by analysis of a digital elevation model using the FlowMapR and FormMapR programs. Figure 34 provides an example of a fuzzy attribute rule base created for use with the BC PEM DSS classification procedures. This table illustrates how a different combination of input variables from that used for the conventional LSM landform facet classification can be used to define a considerably different set of fuzzy attributes. This rule base elects to use different ranges of the terrain variable for log of diffuse upslope area (LNQAREA) to define a series of fuzzy attributes that express the likelihood of a cell being in a particular slope position. Similarly, the terrain variable wetness index (QWETI) is used to define a series of fuzzy attributes that provide approximations of moisture status, slope is used to define classes of relative slope steepness and the terrain variable NEW_ASP is used to compute the likelihood that a cell has a NE or a SW aspect. Also evident in this rule base is a series of fuzzy attributes that are based on consideration of input data not derived solely from automated analysis of an input DEM. These data provide information on the depth and texture of surficial materials and on the presence of high ridges. They are read in from a new input file (Geofile) that is not produced by automated analysis of an input DEM and that was not referenced by the previously illustrated fuzzy attribute rule file used in computing LSM landform facets.

Arule1001

Sortorder	File_in	Attr_in	Class_out	Model_no	B	B_low	B_hi	B1	B2	D
1	formfile	LNQAREA	Crest	5	5.20	5.20	5.20	0.00	5.70	0.50
2	formfile	LNQAREA	Crest2Up	5	6.00	6.00	6.00	0.00	6.50	0.50
3	formfile	LNQAREA	Crest2Mid	5	6.50	6.50	6.50	0.00	7.00	0.50
4	formfile	LNQAREA	Crest2Toe	5	8.50	8.50	8.50	0.00	9.00	0.50
5	formfile	LNQAREA	Up2Mid	1	6.30	6.30	6.30	5.60	7.00	0.70
6	formfile	LNQAREA	Mid2Toe	1	8.00	8.00	8.00	7.00	9.00	1.00
7	formfile	LNQAREA	Toe	1	8.90	8.90	8.90	8.30	9.50	0.60
8	formfile	LNQAREA	Toe2Valley	1	9.50	9.50	9.50	8.50	10.50	1.00
9	formfile	LNQAREA	Valley	4	11.50	11.50	11.50	11.00	25.00	0.50
10	formfile	QWETI	Dry_Wl	5	5.20	5.20	5.20	0.00	5.40	0.20
11	formfile	QWETI	VDry2Wet	5	6.50	6.50	6.50	0.00	7.50	1.00
12	formfile	QWETI	Dry2SlDry	5	5.50	5.50	5.50	0.00	6.00	0.50
13	formfile	QWETI	Sl_Dry_Wl	1	5.50	5.50	5.50	5.00	6.00	0.50
14	formfile	QWETI	Dry2Med_Wl	1	6.50	6.50	6.50	5.50	7.50	1.00
15	formfile	QWETI	Med_Wl	1	7.50	7.50	7.50	6.50	8.50	1.00
16	formfile	QWETI	Med2Sl_Wet	1	8.50	8.50	8.50	8.00	9.00	0.50
17	formfile	QWETI	Sl_Wet_Wl	1	9.20	9.20	9.20	8.80	9.60	0.40
18	formfile	QWETI	SLWet_Wet	1	10.50	10.50	10.50	9.30	11.70	1.20
19	formfile	QWETI	Wet	1	11.60	11.60	11.60	11.10	12.10	0.50
20	formfile	QWETI	Wet2V_Wet	4	12.30	12.30	12.30	11.80	25.00	0.50
21	formfile	SLOPE	Steep	4	35.00	35.00	35.00	30.00	100.00	5.00
22	formfile	SLOPE	SlopeLT05	5	4.00	4.00	4.00	0.00	5.00	1.00
23	formfile	SLOPE	SlopeLT10	5	5.00	10.50	10.50	0.00	10.00	5.00
24	formfile	SLOPE	SlopeLT20	5	15.00	22.50	22.50	0.00	20.00	5.00
25	formfile	SLOPE	SlopeLT30	5	25.00	42.50	42.50	0.00	30.00	5.00
26	formfile	NEW_ASP	NE_Aspect	1	90.00	90.00	90.00	0.00	180.00	45.00
27	formfile	NEW_ASP	SW_Aspect	1	270.00	270.00	270.00	180.00	360.00	45.00
28	geofile	DEPTH	Deep	4	99.00	99.00	99.00	99.00	900.00	1.00
29	geofile	DEPTH	Shallow	5	51.00	51.00	51.00	0.00	50.00	1.00
30	geofile	TEXTURE	Coarse	4	55.00	55.00	55.00	50.00	100.00	5.00
31	geofile	TEXTURE	Medium	1	50.00	50.00	50.00	30.00	70.00	20.00
32	geofile	TEXTURE	Fine	5	31.00	31.00	31.00	0.00	30.00	1.00
33	geofile	RIDGE	OnRidge	4	0.90	0.90	0.90	0.90	10.00	0.10

Figure 34. Example of a fuzzy attribute rule file with that defines attributes for use in the BC-PEM DSS classification

The BC-PEM DSS application required an ability to edit existing fuzzy rule files and create new fuzzy rule files in an efficient and cost-effective manner, since there was a need to create and revise many different rule files for many different zones. The most convenient and efficient way to edit (or create) a fuzzy rule file is to open an existing rule file in DBF format using Microsoft Excel and to add, delete or change entries in the fuzzy rule file using the cell editing capabilities of Excel. The edited file can then be saved from Excel in DBF format and used by FacetMapR.

A Microsoft Excel worksheet that contains a paired combination of fuzzy attribute and fuzzy classification rule files on the same page was determined to be a very convenient and effective way to organize rule files for editing. When the fuzzy attribute and fuzzy classification rule files are viewed simultaneously, the effect of changes made to either of the rule files can be better reviewed and appreciated. This makes for more informed and efficient creation or editing of rule files and supports improved archiving and documentation of the history of development of rule bases. Figure 35 provides an example of an Excel workspace used in creating rule bases for the BC-PEM DSS application. It contains a cross sectional profile diagram that illustrates the conceptual classes that the rules are trying to capture in addition to working copies of the fuzzy attribute and fuzzy classification rule bases that pertain to the classes in the illustration.

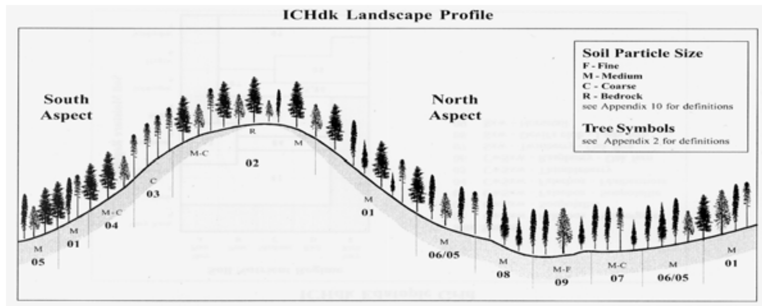
Rules for medium textured ecological units on low relief landforms in ICHdk (Zone 501)

CRULE500

Date 10-Mar-03
Version 1.0
Approval

Texture Medium
Relief Low

f_name	fuzattr	attrwt	facet_no	f_code
S501m	Crest2Toe	30	1	501
S501m	Dry2Med_WI	40	1	501
S501m	Hi_Above	20	1	501
S501m	SlopeLT30	10	1	501
S501n	Mid2Toe	40	2	501
S501n	Med2Sl_Wet	40	2	501
S501n	SlopeLT30	10	2	501
S501n	Gentle_NE	10	2	501
S502r	Crest2Up	30	3	502
S502r	Dry2Sl_Dry	30	3	502
S502r	Crs2Med	10	3	502
S502r	Shallow	30	3	502
S502a	Crest2Mid	30	4	502
S502a	Dry2Sl_Dry	30	4	502
S502a	Steep_SW	30	4	502
S502a	Crs2Med	10	4	502
S503n	Crest2Mid	30	5	503
S503n	Dry2Sl_Dry	30	5	503
S503n	Steep_NE	30	5	503
S503n	Crs2Med	10	5	503
S504m	Crest2Mid	25	6	504
S504m	Dry2Med_WI	25	6	504
S504m	Hi_Above	10	6	504
S504m	SlopeLT10	10	6	504
S504m	Medium	20	6	504
S504m	Deep	10	6	504
S505m	Toe	30	7	505
S505m	Sl_Wet2Wet	40	7	505
S505m	SlopeLT30	20	7	505
S505m	Low20_50	10	7	505
S506m	Toe2Valley	30	8	506
S506m	Sl_Wet2Wet	40	8	506
S506m	SlopeLT30	20	8	506
S506m	Bot5_20	10	8	506
S507m	Valley	30	9	507
S507m	Wet2V_Wet	40	9	507
S507m	SlopeLT10	20	9	507
S507m	Hi_Frost	10	9	507
S508m	Valley	30	10	508



NOTE: Diagram is incorrect! 03 should be on NE steep (Not 01). 02a is SW steep (Not 03).

ARULE500

sortorder	file_in	attr_in	class_out	model_no	b	b_low	b_hi	b1	b2	d	Used For
1	formfile	LNQAREA	Crest	5	5.20	6.00	6.00	0.00	6.50	0.50	
2	formfile	LNQAREA	Crest2Up	5	6.00	6.00	6.00	0.00	6.50	0.50	02r
3	formfile	LNQAREA	Crest2Mid	5	6.50	6.50	6.50	0.00	7.00	0.50	02r,02a,03
4	formfile	LNQAREA	Crest2Toe	5	8.00	8.00	8.00	0.00	8.50	0.50	01m
5	formfile	LNQAREA	Up2Mid	1	6.30	6.30	6.30	5.60	7.00	0.70	
6	formfile	LNQAREA	Mid2Toe	1	7.50	7.50	7.50	6.50	8.50	1.00	01n
7	formfile	LNQAREA	Toe	1	8.80	8.80	8.80	7.30	10.30	1.50	05m
8	formfile	LNQAREA	Toe2Valley	1	9.80	9.80	9.80	9.30	10.30	0.50	06m
9	formfile	LNQAREA	Valley	1	10.20	10.20	10.20	9.70	10.70	0.50	07m,08m
10	formfile	LNQAREA	Dep	4	11.20	11.20	11.20	10.70	50.00	0.50	08i,09
11	formfile	QWETI	Dry_WI	5	5.20	5.20	5.20	0.00	5.70	0.50	
12	formfile	QWETI	Dry2Sl_Dry	5	5.50	5.50	5.50	0.00	6.00	0.50	02r,02a,03
13	formfile	QWETI	Sl_Dry_WI	1	5.50	5.50	5.50	5.00	6.00	0.50	
14	formfile	QWETI	Dry2Med_WI	1	6.50	6.50	6.50	5.50	7.50	1.00	01m
15	formfile	QWETI	Med2Sl_Wet	1	7.50	7.50	7.50	6.50	8.50	1.00	01n
16	formfile	QWETI	Sl_Wet2Wet	1	8.80	8.80	8.80	7.30	10.30	1.50	05m,06m

Figure 35. Illustration of an Excel workspace used to create and edit fuzzy rules for the BC-PEM DSS application

Computing fuzzy landform classes based on fuzzy landform attributes

A module in the FacetMapR program named Calc_class computes likelihood values for fuzzy landform classes based on consideration of previously computed values for fuzzy landform attributes. The process by which this occurs is documented in MacMillan et al., (2000).

The fuzzy classification rule base for the current default LSM landform classification (LM3Crule) defines 15 fuzzy landform facets as described in Table 2.

Table 3. Names and general characteristics of the 15 landform facets in the default LSM landform classification

LM3crule default 15 LSM classes			**Slope Curvature**		**Slope**		**Generalized 3-4 classes**		
ID No.	**Code**	**Name**	**Profile**	**Plan**	**Gradient**	**Other**	**ID No.**	**Code**	**Description**
1	LCR	Level crest	Planar	Planar	Low	Near Peak/Div	1	UP	Upper Slopes
2	DSH	Divergent Shoulder	Convex	Convex	Any	Low WI	1		
3	UDE	Upper Depression	Concave	Concave	Low	High WI	1		
4	BSL	Back slope	Planar	Planar	High	Near Half/Mid	2	MID	Mid Slopes
5	DBS	Divergent Back slope	Planar	Convex	High	Near Half/Mid	2		
6	CBS	Convergent Back slope	Planar	Concave	High	Near Half/Mid	2		
7	TER	Terrace	Planar	Planar	Low	Near Half/Mid	2		
8	SAD	Saddle	Concave	Convex	Any	Near Half/Mid	2		
9	MDE	Mid Slope Depression	Concave	Concave	Low	Near Half/Mid	2		
10	FSL	Foot slope	Concave	Concave	High	Near Chan/Pit	3	LOW	Lower Slopes
11	TSL	Toe slope	Planar	Planar	High	Near Chan/Pit	3		
12	FAN	Fan	Planar	Convex	High	Near Chan/Pit	3		
13	LSM	Lower Slope Mound	Convex	Convex	Any	Near Chan/Pit	3		
14	LLS	Level Lower Slope	Planar	Planar	Low	Near Chan/Pit	4	DEP	Depression
15	DEP	Lower Depression	Concave	Concave	Low	Near Chan/Pit	4		

Lm3crule

F_name	Fuzattr	Attrwt	Facet_no	F_code
LCR	NEAR_PEAK	30	11	1
LCR	NEAR_DIV	20	11	1
LCR	HI_ABOVE	10	11	1
LCR	NEAR_LEVEL	20	11	1
LCR	PLANAR_D	10	11	1
LCR	PLANAR_A	5	11	1
LCR	LOW_WI	5	11	1
DSH	NEAR_PEAK	30	12	2
DSH	NEAR_DIV	20	12	2
DSH	HI_ABOVE	10	12	2
DSH	CONVEX_D	20	12	2
DSH	CONVEX_A	10	12	2
DSH	LOW_WI	10	12	2
UDE	NEAR_PEAK	30	13	3
UDE	NEAR_DIV	20	13	3
UDE	HI_ABOVE	10	13	3
UDE	NEAR_LEVEL	10	13	3
UDE	CONCAVE_D	10	13	3
UDE	CONCAVE_A	10	13	3
UDE	HIGH_WI	10	13	3
BSL	NEAR_HALF	20	21	4
BSL	NEAR_MID	10	21	4
BSL	HI_ABOVE	5	21	4
BSL	REL_STEEP	20	21	4
BSL	PLANAR_D	15	21	4
BSL	PLANAR_A	25	21	4
BSL	LOW_WI	5	21	4
DBS	NEAR_HALF	20	22	5
DBS	NEAR_MID	10	22	5
DBS	HI_ABOVE	5	22	5
DBS	REL_STEEP	20	22	5
DBS	CONVEX_A	20	22	5
DBS	PLANAR_D	15	22	5
DBS	LOW_WI	10	22	5
CBS	NEAR_HALF	20	23	6
CBS	NEAR_MID	10	23	6
CBS	HI_ABOVE	5	23	6
CBS	REL_STEEP	20	23	6
CBS	CONCAVE_A	20	23	6
CBS	PLANAR_D	15	23	6
CBS	HIGH_WI	10	23	6
TER	NEAR_HALF	20	24	7
TER	NEAR_MID	10	24	7
TER	HI_ABOVE	5	24	7
TER	NEAR_LEVEL	30	24	7
TER	PLANAR_D	15	24	7
TER	PLANAR_A	20	24	7
SAD	NEAR_HALF	20	25	8
SAD	NEAR_MID	10	25	8
SAD	HI_ABOVE	5	25	8

Lm3crule

F_name	Fuzattr	Attrwt	Facet_no	F_code
SAD	HI_ABOVE	5	25	8
SAD	NEAR_LEVEL	20	25	8
SAD	CONCAVE_D	20	25	8
SAD	CONVEX_A	20	25	8
MDE	NEAR_HALF	20	26	9
MDE	NEAR_MID	10	26	9
MDE	HI_ABOVE	5	26	9
MDE	NEAR_LEVEL	25	26	9
MDE	CONCAVE_D	10	26	9
MDE	CONCAVE_A	10	26	9
MDE	HIGH_WI	20	26	9
FSL	NEAR_CHAN	20	31	10
FSL	NEAR_PIT	10	31	10
FSL	REL_STEEP	10	31	10
FSL	CONCAVE_D	20	31	10
FSL	CONCAVE_A	20	31	10
FSL	PLANAR_A	10	31	10
FSL	HIGH_WI	20	31	10
TSL	NEAR_CHAN	20	32	11
TSL	NEAR_PIT	10	32	11
TSL	REL_STEEP	10	32	11
TSL	PLANAR_D	25	32	11
TSL	PLANAR_A	25	32	11
TSL	HIGH_WI	10	32	11
FAN	NEAR_CHAN	20	33	12
FAN	NEAR_PIT	10	33	12
FAN	REL_STEEP	10	33	12
FAN	CONVEX_A	25	33	12
FAN	PLANAR_D	25	33	12
FAN	LOW_WI	10	33	12
LSM	NEAR_DIV	10	41	13
LSM	NEAR_CHAN	20	41	13
LSM	NEAR_PIT	10	41	13
LSM	NEAR_PEAK	10	41	13
LSM	REL_STEEP	10	41	13
LSM	CONVEX_D	15	41	13
LSM	CONVEX_A	15	41	13
LSM	LOW_WI	10	41	13
LLS	NEAR_CHAN	20	42	14
LLS	NEAR_PIT	20	42	14
LLS	NEAR_LEVEL	40	42	14
LLS	PLANAR_D	5	42	14
LLS	PLANAR_A	5	42	14
LLS	HIGH_WI	10	42	14
DEP	NEAR_CHAN	20	43	15
DEP	NEAR_PIT	30	43	15
DEP	NEAR_LEVEL	20	43	15
DEP	CONCAVE_A	10	43	15
DEP	CONCAVE_D	10	43	15
DEP	HIGH_WI	10	43	15

Figure 36. Illustration of the fuzzy classification rule file (LM3Crule) for the default LSM landform classification

The Calc_class procedure reads a fuzzy classification rule file (Figure 36) to determine how many fuzzy output classes to compute values for, what names or codes to assign to each fuzzy class and what fuzzy attributes and fuzzy attribute weights to use to compute them. Users can set up rule files to define any number or type of desired output classes using any reasonable and convenient combination of previously defined fuzzy attributes.

The fuzzy classification rule file for the default LSM landform classification (Figure 36) defines 15 landform facets using 17 fuzzy attribute values as identified in Figure 32. Rules for defining a single facet must all be contiguous (grouped together). Each defined class is assigned a facet name (F_name), a facet number (Facet_no) and a facet code (F_Code). The likelihood that a given facet will occur is computed as the sum of the likelihoods of the individual defining fuzzy attributes (Fuzattr) times the attribute weight (Attrwt) associated with each attribute.

Crule1001

F_name	Fuzattr	Attrwt	Facet_no	F_code
S1001s	Up2Mid	30	1	1001
S1001s	Dry2Med_Wl	30	1	1001
S1001s	SlopeLT30	10	1	1001
S1001s	Gentle_SW	20	1	1001
S1001s	Deep	10	1	1001
S1001n	Up2Mid	30	2	1001
S1001n	Med_Wl	30	2	1001
S1001n	SlopeLT30	10	2	1001
S1001n	Gentle_NE	20	2	1001
S1001n	Deep	10	2	1001
S1001a	Crest2Mid	35	3	1001
S1001a	Dry2Med_Wl	35	3	1001
S1001a	SlopeLT10	20	3	1001
S1001a	Deep	10	3	1001
S1002	Crest	30	4	1002
S1002	Dry_Wl	30	4	1002
S1002	Coarse	10	4	1002
S1002	Shallow	30	4	1002
S1003	Crest2Mid	25	5	1003
S1003	Sl_Dry_Wl	25	5	1003
S1003	Steep_SW	30	5	1003
S1003	Coarse	10	5	1003
S1003	Deep	10	5	1003
S1004a	Crest2Up	35	6	1004
S1004a	Dry_Wl	35	6	1004
S1004a	SlopeLT10	20	6	1004
S1004a	Deep	10	6	1004
S1005	Crest2Mid	25	7	1005
S1005	Sl_Dry_Wl	25	7	1005
S1005	Steep_NE	30	7	1005
S1005	Coarse	10	7	1005
S1005	Deep	10	7	1005

Crule1001

F_name	Fuzattr	Attrwt	Facet_no	F_code
S1005	Deep	10	7	1005
S1006	Toe	25	8	1006
S1006	Sl_Wet_Wl	25	8	1006
S1006	SlopeLT30	10	8	1006
S1006	Gentle_SW	20	8	1006
S1006	Medium	10	8	1006
S1006	Deep	10	8	1006
S1007a	Toe	25	9	1007
S1007a	SLWet_Wet	45	9	1007
S1007a	SlopeLT20	10	9	1007
S1007a	Medium	10	9	1007
S1007a	Deep	10	9	1007
S1007b	Toe2Valley	25	10	1007
S1007b	SLWet_Wet	45	10	1007
S1007b	SlopeLT20	10	10	1007
S1007b	Medium	10	10	1007
S1007b	Deep	10	10	1007
S1007st	Valley	30	11	1007
S1007st	Wet2V_Wet	30	11	1007
S1007st	Medium	10	11	1007
S1007st	Deep	10	11	1007
S1008	Valley	30	12	1008
S1008	Wet2V_Wet	30	12	1008
S1008	SlopeLT05	20	12	1008
S1008	Fine	10	12	1008
S1008	Deep	10	12	1008
S1009	Valley	25	14	1009
S1009	Wet2V_Wet	25	14	1009
S1009	SlopeLT05	30	14	1009
S1009	Medium	10	14	1009
S1009	Deep	10	14	1009

Figure 37. Example of a fuzzy classification rule file with that defines output classes for the BC-PEM DSS classification

Figure 37 provides an example of a fuzzy classification rule base created for use with the BC PEM DSS classification procedures. This table illustrates how a different combination of fuzzy attributes from that used for the conventional LSM landform facet classification can be used to define a considerably different set of fuzzy ecological classes. This example table also shows how the same final Site Series output class (e.g. 1001) can occur in more than one landform position and under more than one set of defining conditions. The codes in the column F_name (e.g. S1001s, S1001n, S1001a) identify different occurrences or phases that are all ultimately recognized as belonging to the same final Site Series class (e.g. 1001). A similar situation is evident for Site Series 1007 which also has 3 different phases defined for it. As detailed later in this document, the output file produced by the FacetMapR DSS program records, for each cell in the output file, the codes stored in both the F_name and F_code fields so that grid maps can be made of either the separate phases or the rolled up classes.

The example DBF tables (Figures 36 & 37) also provide a good illustration of the logic that underlies the fuzzy classification approach used by FacetMapR. The approach utilizes a concept that is formally referred to as a semantic import (SI) model (Burrough, 1989; MacMillan et al., 2000). In less formal terms, it means that output classes are defined using easily understood combinations of words and word-like phrases, where the words and phrases have likelihood values or numbers associated with them and can therefore be manipulated in a quantitative fashion.

For example in Figure 37, unit S1001a is defined as most likely to occur in crest to mid slope positions that have a dry to medium moisture index, slopes less than 10% and deep parent materials. Each of these phrases (e.g. Crest2Mid) is a fuzzy attribute and each grid cell in a matrix has a numerical value between 0 and 100 computed for it for every fuzzy attribute that may be used to define any fuzzy class in a region of interest. So, if the value for the fuzzy attribute (e.g. Crest2Mid) is multiplied by the weighting factor for that attribute (Attrwt) and a weighted sum is computed for all fuzzy attributes used to define a class, one arrives at an overall likelihood value that a defined class will occur at given cell with a given set of attributes. The defined class with the highest likelihood of occurring at a particular site is selected as being the most likely correct classification for that site.

The overall weighted fuzzy average for each defined class for each grid cell in a matrix is formally referred to as a Joint Membership Function (JMF_A). The Joint Membership Function (JMF_A) for each defined class is computed as the weighted linear average of the individual Membership Functions (MF_{Aj}) of the fuzzy attributes used to describe a given class multiplied by the relative weighting factor (W_j) assigned to each fuzzy landform attribute according to equation 2.

The SI model was selected as the basis for FacetMapR because it has the advantage of

$$JMF_A = \sum_{j=1}^{k} W_j * MF_{Aj} \ldots\ldots where \ldots \sum_{j=1}^{k} W_j = 1, \ldots\ldots W_j > 0 \quad \textbf{(2)}$$

permitting formal recognition and incorporation of the "imprecise and overlapping semantics used to describe or classify data" (Burrough 1989). In the case of automated terrain and ecological classification, the SI model permits experts to identify any desired number of conceptual classes and to define each class using imprecise semantics (e.g. the class is relatively steep or relatively high in the landscape). Landform classes may be defined as having overlapping characteristics such that two or more elements may both be described as relatively steep but differentiated on the basis of another attribute such as being relatively convex or relatively concave. An added advantage is the ability to manage and resolve situations where the value of one or more input variables used to define a class falls just outside what would otherwise be a class boundary using a classed approach and Boolean logic. With fuzzy logic, the values of one or more input variables may fall outside a notional class boundary, but if the overall likelihood of the class occurring is computed to be greater than the likelihood of any other class, then a site will receive the most likely classification, regardless of the fact that one or more input variables lie outside the stated preferred range for the class. This is a very powerful and useful feature that helps to accommodate the conflict between high natural variation in input properties and rigid class boundaries in traditional Boolean classification approaches.

FacetMapR Output Files

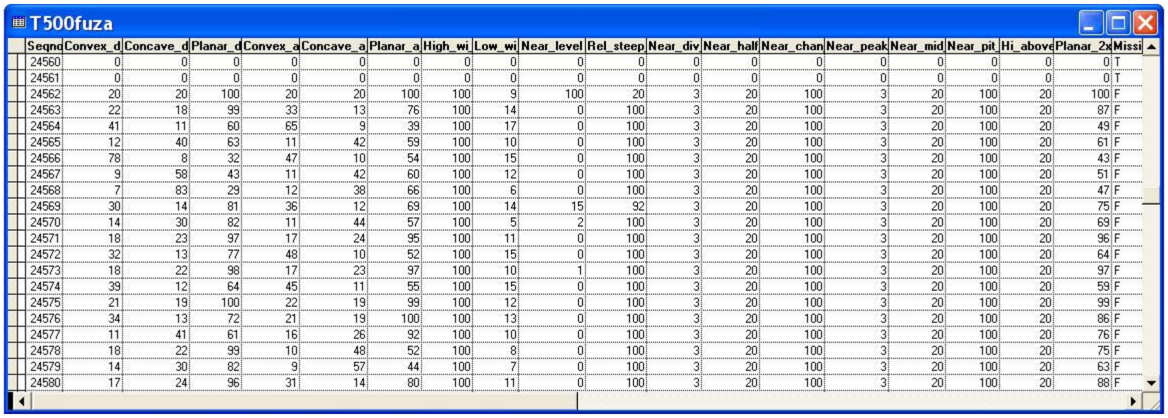
The three different options available for FacetMapR produce three different forms of DBF output files (Table 4) that contain different amounts and types of data pertaining to the resulting classifications.

Table 4. Listing and description of all DBF output files produced by the FacetMapR program

File	Description of contents
ID#Fuza	This file stores and reports a value for every grid cell for every fuzzy landform attribute that is defined in the fuzzy attribute rule file selected for use when option a of the FacetMapR program is run. This file is only produced by option a.
ID#Fuzc	This file stores and reports a value for every grid cell for every fuzzy landform class that is defined in the fuzzy classification rule file selected for use when option a of the FacetMapR program is run. This file is only produced by option a.
ID#DSS	This file stores output data on the most likely ecological classification for every grid cell in a matrix. It does not record or report the likelihood value for every possible fuzzy attribute or every fuzzy class. It is only produced by option b
ID#LSM	This file stores data for only the most likely landform classification produced by application of a condensed version of the full LandMapR LSM program. It only saves data for the most likely output classification and not for every possible fuzzy attribute value or fuzzy class value. It is only produced by option c.

Output files produced by selecting option a

Selecting option a (The original LandMapR LSM program) from the FacetMapR interface results in the production of two different output files (Figures 38 & 39). These files contain values for the fuzzy likelihood that each grid cell meets the criteria for membership in every possible fuzzy attribute class (ID#Fuza) and fuzzy landform class (ID#Fuzc) defined in the relevant rule bases selected to be applied to the input data sets.


T500fuza

Seqno	Convex_d	Concave_d	Planar_d	Convex_a	Concave_a	Planar_a	High_wi	Low_wi	Near_level	Rel_steep	Near_div	Near_half	Near_chan	Near_peak	Near_mid	Near_pit	Hi_above	Planar_2x	Missi
24560	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T
24561	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T
24562	20	20	100	20	20	100	100	9	100	20	3	20	100	3	20	100	20	100	F
24563	22	18	99	33	13	76	100	14	0	100	3	20	100	3	20	100	20	87	F
24564	41	11	60	65	9	39	100	17	0	100	3	20	100	3	20	100	20	49	F
24565	12	40	63	11	42	59	100	10	0	100	3	20	100	3	20	100	20	61	F
24566	78	8	32	47	10	54	100	15	0	100	3	20	100	3	20	100	20	43	F
24567	9	58	43	11	42	60	100	12	0	100	3	20	100	3	20	100	20	51	F
24568	7	83	29	12	38	66	100	6	0	100	3	20	100	3	20	100	20	47	F
24569	30	14	81	36	12	69	100	14	15	92	3	20	100	3	20	100	20	75	F
24570	14	30	82	11	44	57	100	5	2	100	3	20	100	3	20	100	20	69	F
24571	18	23	97	17	24	95	100	11	0	100	3	20	100	3	20	100	20	96	F
24572	32	13	77	48	10	52	100	15	0	100	3	20	100	3	20	100	20	64	F
24573	18	22	98	17	23	97	100	10	1	100	3	20	100	3	20	100	20	97	F
24574	39	12	64	45	11	55	100	15	0	100	3	20	100	3	20	100	20	59	F
24575	21	19	100	22	19	99	100	12	0	100	3	20	100	3	20	100	20	99	F
24576	34	13	72	21	19	100	100	13	0	100	3	20	100	3	20	100	20	86	F
24577	11	41	61	16	26	92	100	10	0	100	3	20	100	3	20	100	20	76	F
24578	18	22	99	10	48	52	100	8	0	100	3	20	100	3	20	100	20	75	F
24579	14	30	82	9	57	44	100	7	0	100	3	20	100	3	20	100	20	63	F
24580	17	24	96	31	14	80	100	11	0	100	3	20	100	3	20	100	20	88	F

Figure 38. An example of a fuzzy attribute DBF output file produced by application of option a of FacetMapR

Figure 38 is difficult to read, but it should be evident that it contains 17 columns that correspond exactly in name to the 17 different fuzzy landform attributes defined in the fuzzy landform attribute rule table (LM3arule) previously presented as Figure 32. When option a is selected, the Fuz_calc procedure of FacetMapR creates a new DBF table that has a column in it for every fuzzy attribute defined in the fuzzy attribute rule file (here LM3Arule) that is read in. The program then computes a fuzzy attribute likelihood value for every one of the defined fuzzy attributes for each grid cell in the matrix. It stores the fuzzy likelihood value for each fuzzy attribute for each grid cell in the field that has the same name as the defined fuzzy attribute.

Some readers may notice that Figure 38 contains an extra (18^{th}) column at the right end of the table that is given the name Planar_2x. This column is the result of a hard coded calculation built into the FacetMapR program. The value in this field represents an average of the values computed and stored in the fields planar down (Planar_d) and planar across (Planar_a). This value was deemed to be needed in order to properly recognize sites that were simultaneously planar in both the down slope and across slope directions. Because it is hard coded into the program, the program always looks for values for the variables Planar_d and Planar_a. If these fuzzy attributes have not been defined in the input fuzzy attribute rule base they will not be present for the program to access. In such cases, the program is likely to fail.

It is therefore important that users realize that any fuzzy attribute rule base that they define and use must include definitions for Planar_d and Planar_a. If these fuzzy attributes are not defined, the hard coded calculation in FacetMapR will not be able to find critical input data it requires and the program is likely to fail.

From the above explanation, it should also be clear that the number of fields in the ID#Fuza output file and the names of the fields are not fixed but will vary depending upon how many fuzzy attributes are defined in the fuzzy attribute rule file that is read in at the start of the program. If the fuzzy attribute rule file defines 6 fuzzy attributes then the resulting fuzzy attribute output file (ID#Fuza) will contain 6 fields plus 1 extra field for Planar_2x and 1 more extra field to identify the presence of cells with missing data (Missing). If the fuzzy attribute rule file defines 20 fuzzy attributes, the fuzzy attribute output file will contain 20 fields plus the 2 extra fields for Planar_2x and Missing.

It is not hard to imagine how the fuzzy attribute output file produced by application of option a can rapidly grow to a large and unmanageable size for large data sets with many fuzzy attributes defined for them. Option c was introduced to address this problem.

T500fuzc_cplus

Seqno	Missing	Lcr	Dsh	Ude	Bsl	Dbs	Cbs	Ter	Sad	Mde	Fsl	Tsl	Fan	Lsm	Lls	Dep	Max_facet	Next_facet	Fac4
24560	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24561	T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24562	F	84	60	60	67	51	43	83	49	51	22	55	43	32	53	27	1	7	1
24563	F	62	97	58	40	65	37	15	37	18	13	12	46	69	2	2	2	13	1
24564	F	65	100	62	38	63	35	13	35	16	13	12	46	71	2	2	2	13	1
24565	F	50	55	67	43	48	59	18	39	41	49	13	21	35	3	23	3	6	1
24566	F	49	84	46	43	68	39	18	40	21	13	13	46	65	3	3	2	5	1
24567	F	27	32	43	56	61	73	31	54	54	50	14	22	27	4	24	6	5	2
24568	F	22	27	39	58	63	75	33	56	57	52	15	23	26	5	25	6	5	2
24569	F	23	58	20	58	83	55	33	56	37	14	14	48	56	4	5	5	4	2
24570	F	22	27	39	59	64	77	35	57	59	52	15	22	25	5	25	6	5	2
24571	F	25	30	41	57	62	74	32	55	56	50	14	22	26	4	24	6	5	2
24572	F	30	65	26	51	76	48	26	48	29	13	13	47	57	3	3	5	2	2
24573	F	27	32	44	54	59	70	29	51	52	50	13	22	27	3	24	6	5	2
24574	F	29	64	26	51	76	47	26	48	29	13	13	46	57	3	3	5	2	2
24575	F	24	59	21	58	83	54	33	55	36	14	14	47	56	4	4	5	2	2
24576	F	21	56	18	60	85	56	35	58	38	14	14	48	55	4	5	5	4	2
24577	F	19	24	35	51	56	68	26	49	49	55	19	27	28	9	29	6	5	2
24578	F	18	23	34	45	50	61	20	42	43	61	26	34	35	16	36	10	6	3
24579	F	17	22	34	41	46	58	16	38	39	67	32	41	41	23	43	10	6	3
24580	F	17	32	24	42	67	38	17	60	30	47	30	64	55	21	31	5	12	2

Figure 39. An example of a fuzzy classification DBF output file produced by application of option a of FacetMapR

Figure 39 illustrates how option a of FacetMapR outputs data on fuzzy classification values for an example LSM landform classification. As with the fuzzy attribute output file, the number of fields in this file and the names assigned to the fields are controlled by the list of fuzzy classes present in the fuzzy classification rule file selected for use by the program.

The example output file provided (Figure 39) reflects the list of output classes contained in the current standard LandMapR LSM fuzzy classification rule file named LM3Crule (see Figure 36). This fuzzy classification rule file defines 15 different landform facet classes. Each landform class is identified by a 3 letter code in the field named F_name in the fuzzy classification rule file (Figure 36). The fuzzy classification output file contains a field with the same name as each 3 letter code for every unique code present in the input fuzzy classification rule file. The fields are ordered from left to right according to the order established by the value of the field Facet_no in the input rule file. Working from left to right in figure 39, the first field (other than the key field of Seqno) is named LCR for level crest. It is associated with a facet code (F_code) of 1 in the fuzzy classification rule file (Figure 36). The divergent shoulder field (DSH) is associated with facet code 2 and so on until the final defined class of depression (DEP) which has a facet code of 15. A different fuzzy classification rule base could clearly contain a different number of defined facets with different names. The basic principal remains, however, that this fuzzy classification output file will have one field for each defined fuzzy class and that field will have the name of the fuzzy class assigned to it and will be ordered in sequence in accordance with the facet number (Facet_no) assigned to that class in the input rule file.

The field Max_facet contains a number that identifies the output class that has the highest computed likelihood value and is judged to be the most likely facet to occur at that grid cell. Ties are settled by selecting the last class with the highest value. This favors recognition of landform classes that are lower in the landscape over classes that are higher in the landscape, at least given the order in which the default LandMapR LSM classification rule files consistently present their defined classes (from higher to lower landform classes). The field Next_Facet contains the facet code (F_Code) for the fuzzy output class considered to be next most likely to occur at a given grid cell site. This is an artifact of the original program development when it was considered desirable to be able to check the output results manually to confirm that reasonable results were being obtained. The field Fac4 just contains a re-classification of the most likely output class into the simplified 4 unit classification as indicated in Table 3. The field Missing is used to identify if a cell is considered to be a missing data cell.

Output files produced by selecting option b

Selecting option b (The new BC-PEM Direct-to-Site-Series DSS program) from the FacetMapR interface results in the production of a single output file that is always given the name ID#DSS (Figure 40).

P96dss

Seqno	Missing	Bec_zone	Max_class	Max_code	Max_value
1	F	1101	S1101A	1101	70
2	F	1101	S1101A	1101	93
3	F	1101	S1101A	1101	90
4	F	1101	S1101A	1101	88
5	F	1101	S1101A	1101	87
6	F	1101	S1101A	1101	88
7	F	1101	S1101A	1101	88
8	F	1101	S1101A	1101	90
9	F	1101	S1101A	1101	90
10	F	1101	S1101A	1101	90
11	F	1101	S1101A	1101	89
12	F	1101	S1101A	1101	89
13	F	1101	S1101A	1101	89
14	F	1101	S1101A	1101	92
15	F	1101	S1101A	1101	93

Figure 40. An example of a fuzzy classification DBF output file produced by application of option b of FacetMapR

The output file produced by selecting option b from the FacetMapR interface is vastly more compact than the two files produced by selecting option a. This file only contains values for the most likely fuzzy output classification for each grid cell. It does not record or report values for all defined fuzzy attributes or all possible fuzzy output classifications for each grid cell.

There are several good reasons for this move to a different, and more compact, file structure.

Firstly, the BC-PEM DSS application was designed to permit application of different classification rules to different zones or regions within an area of interest. The rules for each zone can, and do, define different types and numbers of both fuzzy attributes and fuzzy output classes for each zone. It is therefore physically impossible to maintain the practice of defining output DBF files with a dynamic structure determined by differing numbers of fuzzy attributes and fuzzy output classes with different names in each zone. Field names defined dynamically on the basis of the names of attributes or classes for the first zone read in would not match those for the next zone and so on. The output file structure would therefore become both unreasonably large and highly non-normal (e.g. it would contain many empty fields and many fields of redundant data). It was absolutely necessary to move away from the practice of dynamically creating an indeterminate number of fields with unpredictable names. It became necessary to define a structure for the output file that was fixed and predictable. The resulting structure is illustrated in Figure 40.

Secondly, even without the problems caused by the unwieldy and unpredictable file structure created by option a, it was necessary to move to a more compact file structure that stored less data, particularly less data that was unlikely to ever be used. The BC-PEM DSS application is focused on creating complex multi-level hierarchical classifications for very large areas (1 million ha and more) at quite fine levels of spatial resolution (10-25 m grid squares). The raster data sets for these very large areas have the potential to become extremely large (tens to hundreds of gigabytes). Any design that reduces the size of the output files is sure to result in efficiencies in both data storage and processing time.

It was concluded that it was neither necessary nor desirable to store all computed fuzzy attribute likelihood values and fuzzy classification likelihood values for every cell in a raster grid. Having access to all fuzzy attribute and fuzzy classification likelihood values for all cells was desirable and useful during the original stages of program development and testing. It was far less necessary or desirable once the program began to be used for routine operational mapping.

In Figure 40, the field Bec_zone contains the numeric identifier for the ecological zone within which a particular set of classification rules is considered to apply. The field Max_class stores the alpha-numeric code used to identify the phase of the most likely ecological class as computed by the program. The code assigned to each phase is the same one that is recorded in the field F_name in the fuzzy classification rule file that is used to define all potential output classes for the current zone. The field Max_code stores an integer value that identifies the most likely fuzzy output class. This integer value is equivalent to the value stored in the field F_code in the input fuzzy classification rule file used to define the list of possible fuzzy output classes for the current zone. It is meant to equate to the rolled-up Site Series classification in the BC-PEM system while the Max_class value is meant to equate to the un-rolled up phase or landscape specific occurrence of the defined Site Series. The field Max_Value records the likelihood value computed by the program for the reported most likely class. It is generally not used but can be referenced to provide an indication of the relative confidence that the predicted class is likely to occur. The higher the maximum likelihood value, the more likely it is that the site in question meets all of the criteria specified in the rule files for membership in the defined output class.

When option b is selected, none of the intermediate calculations pertaining to the individual values for fuzzy attributes or to individual fuzzy output classes are stored in the final output file. All of this potentially interesting, but not vital, information is discarded. The savings in processing time and disk space were judged to be more important than the ability to review the results of all intermediate calculations of fuzzy attribute values or fuzzy classification values for classes that were not computed to be the most likely to occur.

Output files produced by selecting option c

Selecting option c (A new condensed version of the LandMapR LSM program) from the FacetMapR interface results in the production of a single output file that is always given the name ID#LSM (Figure 41).

T500lsm_cplus

Seqno	Missing	Max_class	Max_code	Max_value
24560	T	NA	0	0
24561	T	NA	0	0
24562	F	LCR	1	84
24563	F	DSH	2	97
24564	F	DSH	2	100
24565	F	UDE	3	67
24566	F	DSH	2	84
24567	F	CBS	6	73
24568	F	CBS	6	75
24569	F	DBS	5	84
24570	F	CBS	6	76
24571	F	CBS	6	74
24572	F	DBS	5	76
24573	F	CBS	6	70
24574	F	DBS	5	76
24575	F	DBS	5	82
24576	F	DBS	5	85
24577	F	CBS	6	68
24578	F	CBS	6	61
24579	F	FSL	10	67
24580	F	DBS	5	67

Figure 41. An example of a fuzzy classification DBF output file produced by application of option c of FacetMapR

The output file produced by selecting option c from the FacetMapR interface is almost identical to that produced by selecting option b. The principal difference is that this output files does not have a column to store data for the zone within which the rules used to define the classes applied (BEC_Zone in Figure 40). The classification results produced by the two different versions of the same program are virtually identical, only the volume of extra, intermediate data is changed.

Extra DBF Format Input Files Required for the DSS Option of FacetMapR

Option 2 of the FacetMapR program implements the new BC-PEM Direct to Site Series (DSS) procedures. This option requires the prior creation of 2 specific DBF format files that are not required for running the original LandMapR landform classification but are required to run the DSS option. These 2 files are always given the names ID#Zone and ID#Geo (Table 5).

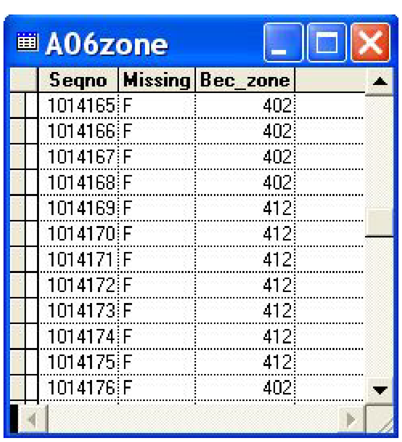
Table 5 Listing and description of the 2 extra DBF input files required to run the DSS option of the FacetMapR program

File	Description of contents
ID#Zone	This file stores and reports an integer value for every grid cell for a field named BEC_Zone. The BEC_Zone field identifies a specific zone or area within which a particular set of fuzzy rules is to be used by the FacetMapR program. This file is only required when option b is selected.
ID#Geo	This file stores and reports a value for every grid cell for any number of user-supplied attributes. The user-supplied attributes generally come from non-DEM data sources such as direct manual mapping, remotely sensed imagery or pre-existing maps of geology, vegetation or other spatially distributed phenomenon. This file is only required when option b is selected and non-DEM spatial attributes are used to define any of the desired fuzzy output classes.

Content and Purpose of the ID#Zone DBF File

The DBF file named ID#Zone was introduced to permit application of different fuzzy classification rules within different zones or regions of an area covered by a single DEM. This capability was first required in order to implement hierarchical concepts inherent in the British Columbia system of Biogeoclimatic Ecosystem Classification (BEC) Mapping used in the production of predictive ecosystem maps (PEM) in BC.

Initially, the BC-PEM DSS application envisaged setting up one zone for each BEC sub-zone or variant in any given region of interest. The idea was that each BEC zone would have a unique set of ecological classes that needed to be predicted and that a single set of fuzzy rules for each BEC zone would be sufficient to predict all ecological classes defined for any given BEC zone. A file named ID#Zone was created that contained a field named BEC_Zone to identify the BEC sub-zone within which a given set of classes defined by a given set of fuzzy rules was to occur (Figure 42).


A06zone

Seqno	Missing	Bec_zone
1014165	F	402
1014166	F	402
1014167	F	402
1014168	F	402
1014169	F	412
1014170	F	412
1014171	F	412
1014172	F	412
1014173	F	412
1014174	F	412
1014175	F	412
1014176	F	402

Figure 42. Illustration of the structure and content of the ID#Zone file used in the DSS option of FacetMapR

This concept evolved as experience was gained in actually applying the BEC ecological classification system in BC. It rapidly became apparent that the hierarchical nature of the BEC system extended below the level of BEC sub-zones and variants that was initially considered to be the lowest level of the hierarchy. Within each BEC sub-zone, it was observed that different sets of ecological classes with different sets of defining rules existed in different portions of the BEC sub-zone, depending upon additional environmental factors such as the texture of the parent material and the scale of relief in the landscape. It became apparent that the DSS version of the FacetMapR program would have to be able to read in and apply different sets of rules to define different sets of classes not only within different BEC sub-zones but also within further sub-divisions of the BEC Sub-zones.

For the BC DSS PEM application, BEC sub-zones were identified and further sub-divided according to a systematic procedure and numbering scheme (Table 6). The integer numbers used to identify classification regions or zones using the field BEC_Zone can be entirely arbitrary, as long as each zone has a unique number. However, it is easier to remember and interpret the numbering scheme if it follows some kind of logical order. In the BC DSS PEM procedures, a hierarchical numbering scheme documented in Table 6 was used.

Table 6. Example of hierarchical numbering scheme used to assign unique codes to sub-divisions of BEC sub-zones

Zone Code	BEC Sub-zone	BEC Code	Texture Class	Texture Code	Other Class	Other Code
100	Atp	100	Not Applicable	00	Exception	0
101	Atp	100	Medium	00	Low relief	1
102	Atp	100	Medium	00	High relief	2
111	Atp	100	Coarse	10	Low relief	1
112	Atp	100	Coarse	10	High relief	2
200	ESSFdc2	200	Not Applicable	00	Exception	0
201	ESSFdc2	200	Medium	00	Low relief	1
202	ESSFdc2	200	Medium	00	High relief	2
211	ESSFdc2	200	Coarse	10	Low relief	1
212	ESSFdc2	200	Coarse	10	High relief	2
300	ESSFwc3	300	Not Applicable	00	Exception	0
301	ESSFwc3	300	Medium	00	Low relief	1
302	ESSFwc3	300	Medium	00	High relief	2
311	ESSFwc3	300	Coarse	10	Low relief	1
312	ESSFwc3	300	Coarse	10	High relief	2
1200	ESSFwc3	1200	Not Applicable	00	Exception	0
1201	ESSFwc3	1200	Medium	00	Low relief	1
1202	ESSFwc3	1200	Medium	00	High relief	2
1211	ESSFwc3	1200	Coarse	10	Low relief	1
1212	ESSFwc3	1200	Coarse	10	High relief	2

Examination of Table 6 reveals that each of the 12 BEC Sub-zones in this map area was assigned an integer value starting at 100 and ending at 1200. The 12 BEC sub-zones were arranged and numbered in alphabetical order. Within each BEC sub-zone, regions that were deemed to be covered by coarse textured parent materials were identified by the number 10 which resulted in the presence of the numeral 1 in the 10's place for the final combined code for regions that were deemed to be coarse and the numeral 0 in the 10's place for all other parent material textures. The numeral in the 1's place was set to indicate whether the region of interest contained an exception class (0) or was deemed to exhibit low (1) or high (2) relief. Exception classes were classes that were directly mapped by manual methods and did not require prediction by the modeling procedures. Different rules were deemed to be required within areas of low (1) and high (2) relief within areas of both medium (00) and coarse textured (10) soils. The combined code, which is listed under the heading Zone Code in Table 6, was computed by adding the individual values for BEC code, texture code and other code. The sum of these individual codes represented a unique integer number that could be easily interpreted to rapidly identify the BEC zone, parent material texture and degree of relief of the region that it applied to. Using a systematic, cognitive numbering scheme such as this helps to organize the rule files and to maintain an understanding of the areas for which each numbered rule file is meant to apply.

Users who plan to use the BC-PEM DSS option to apply different rule bases to different portions of an area of interest should consider adopting a similar numbering strategy so as to make their zone numbers and the matching rule base numbers as easy as possible to understand and interpret. The field BEC_Zone in the file ID#Zone is presently defined to accept integer numbers up to a value of 99999. Users can therefore devise and assign any numbering scheme they may find convenient to use as long as the numbers are integers and the largest integer value is no larger than 99999.

The ID#Zone file must be created by users and formatted as a DBF table that contains the fields illustrated in Figure 45. A blank copy of an ID#Zone file is provided on the LandMapR distribution CD. Users will need to create a grid file of integer numbers that correspond to the BEC_Zones that they wish to identify and for which they propose to define and apply different sets of fuzzy attribute and fuzzy classification rule files. They must export this grid file that identifies which BEC_Zone each grid cell has been assigned to from whatever GIS they are using and reformat the grid data into a DBF format file named ID#Zone that is structured as depicted in Figure 45. In order to do this, they will have to have access to a program that can create DBF format files and populate these DBF format files with data. It may well be possible to use the database table editing features of ArcView 3.2 or ArcGIS 8.0 to create and populate these ID#Zone files. To date, LandMapper Environmental Solutions has created the grid files in ArcView 3.2, exported them as binary FLT files, used the GridReadWrite utility to reformat them into DBF format files and then used Visual FoxPro to open a blank ID#Zone file and append the data for BEC_Zone to this file from the reformatted DBF file. This is a somewhat cumbersome and inefficient procedure and should probably be improved. For the moment, users who propose to create zone files in order to use the BC-PEM DSS option of FacetMapR will have to follow similar inefficient and cumbersome procedures, until such time as improved tools and procedures for accomplishing this reformatting are created.

Content and Purpose of the ID#Geo DBF File

A second DBF file may be required by the BC-PEM DSS option of FacetMapR that is not required by the other options. This file is always named ID#Geo (Figure 43). The ID#Geo file was initially conceived of and devised to permit use of geological information on the texture and depth of parent material deposits in defining and mapping ecological classes in BC. The content of the ID#Geo file was subsequently expanded to include almost any kind of ancillary data layer not produced by the LandMapR programs FormMapR or FlowMapR that was deemed to be needed for classifying any ecological entity of interest.

A06geo

Seqno	Geocode	Elev	Depth	Texture	Water	Swamp	Brush	Meadow	Not_mapped	Seepage	Ridge	B3	B5	Z2wet	N2wet	L2wet
307348	102	2059.598	40	70	0	0	0	0	0	0	1	250	254	511.908	370	5860.2
307349	102	2059.051	40	70	0	0	0	0	0	0	1	252	254	511.361	370	5853.6
307350	102	2058.331	40	70	0	0	0	0	0	0	1	252	254	510.641	370	5845.6
307351	102	2057.477	40	70	0	0	0	0	0	0	1	252	254	509.787	370	5836.4
307352	102	2056.553	40	70	0	0	0	0	0	0	1	250	254	508.863	370	5826.7
307353	102	2055.627	40	70	0	0	0	0	0	0	1	250	254	507.937	370	5816.9
307354	102	2054.751	40	70	0	0	0	0	0	0	1	250	254	507.061	372	5807.5
307355	102	2053.950	40	70	0	0	0	0	0	0	1	248	254	506.260	372	5798.9
307356	102	2053.218	40	70	0	0	0	0	0	0	1	248	254	505.528	372	5790.9
307357	102	2052.531	40	70	0	0	0	0	0	0	1	248	254	504.841	372	5783.2
307358	102	2051.861	40	70	0	0	0	0	0	0	1	248	254	504.171	372	5775.8
307359	102	2051.189	40	70	0	0	0	0	0	0	1	248	254	503.499	372	5768.3
307360	102	2050.518	40	70	0	0	0	0	0	0	1	248	254	502.828	372	5760.8
307361	102	2049.873	40	70	0	0	0	0	0	0	1	211	254	502.183	372	5753.6
307362	102	2049.307	40	70	0	0	0	0	0	0	1	211	254	501.617	372	5747.1

Figure 43. Illustration of the structure and content of the ID#Geo file used in the DSS option of FacetMapR

The ID#Geo file is a convenient catch-all for storing any data for grid cells that are not computed automatically by the FlowMapR or FormMapR programs. The ID#Geo file illustrated in Figure 46 contains information for a variety of data layers not computed directly by the LandMapR toolkit programs.

The field Geocode contains a unique integer number that identifies the polygon number of a vector ArcView coverage that describes the spatial distribution of geological materials and exception classes in this area of interest. Each manually mapped polygon in the ArcView vector coverage had been assigned an estimated value for parent material texture (Texture) and depth (Depth). In this particular case, the assigned values were essentially codes with values of 20 for fine textured, 50 for medium textured, and 70 for coarse textured materials. Material depth had been reported as 40 for shallow materials and 200 for deep materials. Users can define any fields they wish, with any names they wish and may enter any values they wish. The example provided simply demonstrates the approach used in one particular application.

The fields labeled as Water, Swamp, Brush, Meadow, Not_Mapped and Seepage in Figure 43 are binary fields that contain data on what were termed "exception classes" for the purpose of DSS PEM mapping in BC. The BC DSS procedures elected to use a manually produced map of exception classes to map several spatially distributed phenomena that were judged to be more suited to direct mapping via human visual interpretation than to estimation via computer modeling. The rationale behind this approach is that some spatial entities can be identified far more easily, rapidly, accurately and cost-effectively using direct human visual interpretation than by any currently available means of automated modeling or computerized classification. Spatial entities such as open water (lakes and ponds), non-forested wetlands (e.g. bogs, fens, swamps and marshes) and non-forested uplands (e.g. meadows, pastures, brush, bare soil or rock) are amenable to rapid and accurate visual identification using available high resolution imagery. In the BC PEM DSS procedures, it was decided that these kinds of areas could be identified far more accurately and in a more cost effective manner using manual visual interpretation than by attempting to model them. In fact, both open water and non-forested wetlands had already been mapped as part of the provincial TRIM base mapping program and provincial forest cover mapping respectively. It therefore did not make sense to attempt to predict these previously mapped spatial entities using a modeling approach when they had already been mapped as well as could be expected at the scale of interest (1:20,000).

Areas directly mapped as Water, Swamp, Brush, Meadow or Not_Mapped are therefore viewed as exception areas for which it is neither necessary nor desirable to compute the likelihood of occurrence of any other spatial entities. These areas represent binary classes in that, if they occur at a given location, no other classes of spatial entity are considered to have any likelihood of occurring at the same location. Consequently, any grid cell identified on the map of exceptions as being open water (Water) will be immediately classified as water by the BC PEM DSS procedures with no computational effort being expended to compute the likelihood of that cell belonging to any other potential class. This approach effectively "cookie cuts" into the final resulting maps all areas that have been directly mapped as exceptions using manual visual interpretation of available imagery. The "cookie cut" procedure preserves the outlines and boundaries of all manually mapped areas exactly as they were drawn by the photo interpreter. Preserving such boundaries exactly helps to maintain the credibility of the resulting output map. Most users will have a tendency to rapidly check the classification of clearly identifiable features such as lakes, marshes and open meadows or bare rock that have sharply defined and easily interpreted boundaries. The credibility of any map of predicted ecological classes will be reduced if users detect discrepancies between the boundaries of readily visible features such as lakes or wetlands and the boundaries of mapped entities. It is therefore desirable to map these hard boundaries as accurately as possible (using manual visual interpretation) and to maintain these hard boundaries in any final output map.

The exception area identified by the designation of "seepage" highlights another situation in which data derived from manual interpretation were needed to recognize conditions that were not easily and accurately identified using derivatives of the DEM alone. This situation was one in which ecological site conditions were considerably changed due to sub-irrigation caused by the presence of a higher than normal water table. The higher than normal water table was initially related to the presence of seepage and the binary field was therefore initially given the name seepage. Similar conditions of a higher than normal water table were subsequently encountered in flood plain areas underlain by a high water table. The key feature of such areas is therefore not the mechanism by which an area comes to possess a higher than expected water table, but the simple presence of a higher than expected water table. While DEM data can be of some use in modeling the likely locations of high water tables, information on the configuration of the terrain surface is generally not sufficient to unambiguously and accurately identify the locations of all areas of anomalously high water tables.

In the BC PEM DSS procedures we elected to treat areas with anomalously high water tables as "exception areas" that were most effectively identified and outlined using manual visual interpretation, guided and informed by reference to both high resolution imagery and to grid maps of terrain derivatives and forest cover. The concept here was that areas that were "wetter" than expected for a given landform position or situation would stand out due to anomalous vegetation cover or other visual clues on the available imagery. Specifically, in BC such areas tended to exhibit a forest cover dominated by tree species that preferred wetter conditions. Stands dominated by tree species that preferred wetter conditions tended to stand out as "exceptions" relative to the distribution of normal tree species for a given area. These "exception areas" were recognized and isolated by manual visual interpretation. Subsequent modeling procedures were trained to recognize that different rules were required within these wetter exception areas so that different classes of ecological entities occupied each of the defined landform positions relative to the classes that would normally be expected to occupy each position.

The lesson learned from recognition and application of the "seepage" concept in the BC PEM mapping is instructive and can be generalized. It is clear that DEM data are often not sufficient to permit effective recognition of subsurface conditions that are not strongly and uniquely linked to terrain position or terrain configuration. The presence of an unexpectedly high water table is just one example of such a condition. Other sub-surface changes that might conceivably occur that could significantly alter site conditions might include a significant change in the mineralogy of the parent material (e.g. from base rich and fertile to acidic and less fertile) or the appearance of a shallow depth to bedrock in an unexpected location such as a valley bottom. If sub-surface conditions can be inferred from other available digital data sources (e.g. satellite or airborne radar, radiometric, electromagnetic or magnetic data) then these data can be used to model and isolate other kinds of "exception areas". If there are no digital data sources available that can be easily interpreted to infer sub-surface conditions, it may prove necessary to use manual visual interpretation of available imagery and maps to infer important changes in sub-surface conditions, as was done for the BC PEM mapping.

Users planning to develop and apply custom classifications are advised to consider whether their particular application may require information about sub-surface conditions that is not easily related to any of the terrain derivatives computed by the basic FormMapR or FlowMapR programs. If the existing suite of terrain derivatives is not able to easily and accurately infer a particular condition that may be vital to achieving a correct classification, users are encouraged to consider alternative means of obtaining the data required to achieve a satisfactory result. Preparation of alternative input layers via direct manual interpretation of available imagery and secondary source data sets is often a viable and cost-effective approach. Alternately, users may consider whether alternate non-DEM digital data sets may be able to provide them with a more effective means of inferring the sub-surface (or other) conditions that are not well predicted using available DEM derivatives.

The ID#Geo file illustrated in Figure 43 contains two fields named B3 and B5 that contain contrast stretched digital numbers associated with bands 3 and 5 of a LandSat7 TM image of the study area. These data were used to identify three different classes with 3 different densities of shrub or stunted tree cover in an alpine ecological zone. None of the DEM derivatives were very effective for predicting the spatial pattern of distribution of vegetation density in the alpine zone in this example. However, this pattern was readily visible and easily interpreted on the LandSat7 TM imagery. The LandSat7 TM image data were therefore added to the ID#Geo file for this area and were used as the key inputs for differentiating 3 different classes of differing density of shrub cover in the alpine zone.

In a similar fashion, the ID#Geo file illustrated in Figure 43 was expanded to include data for 3 new terrain derivatives (Z2Wet, N2Wet, L2Wet) that are not computed by the basic FormMapR program. These 3 derivatives represent the vertical (Z), flow length (N) and horizontal (L) distance from each grid cell to the closest wetland or body of open surface water to which a cell is connected by a defined path of overland surface flow. These are derivatives of the DEM, but they are not derivatives that are normally computed by the FlowMapR program. These extra, non-standard, derivatives were appended to the ID#Geo file, as it is this file that is intended to include all non-standard input values that the FacetMapR program needs to consider.

Users who wish to customize the FacetMapR program to classify entities of their own definition that vary in a hierarchical fashion will therefore need to define an ID#Zone file and will almost certainly need to create an ID#Geo file that can contain spatial data not derived from the DEM by application of the FormMapR or FlowMapR programs. The ID#Geo file does not have to be present if no rules are developed that reference input data that is not derived from application of the LandMapR procedures to DEM data. If users do wish to use non-DEM input data to define classes, they must obtain and map these data as grid cell values tied to each grid cell within an area of interest. They must then format these data into a DBF file named ID#Geo that can contain any number of fields of data with any names that the user finds to be convenient used to identify each field of data. The ID#Geo file can then be used to hold all non-standard input data that may be required to define any spatial entities using custom, user defined rule bases.

Using and visualizing the FacetMapR Output Data

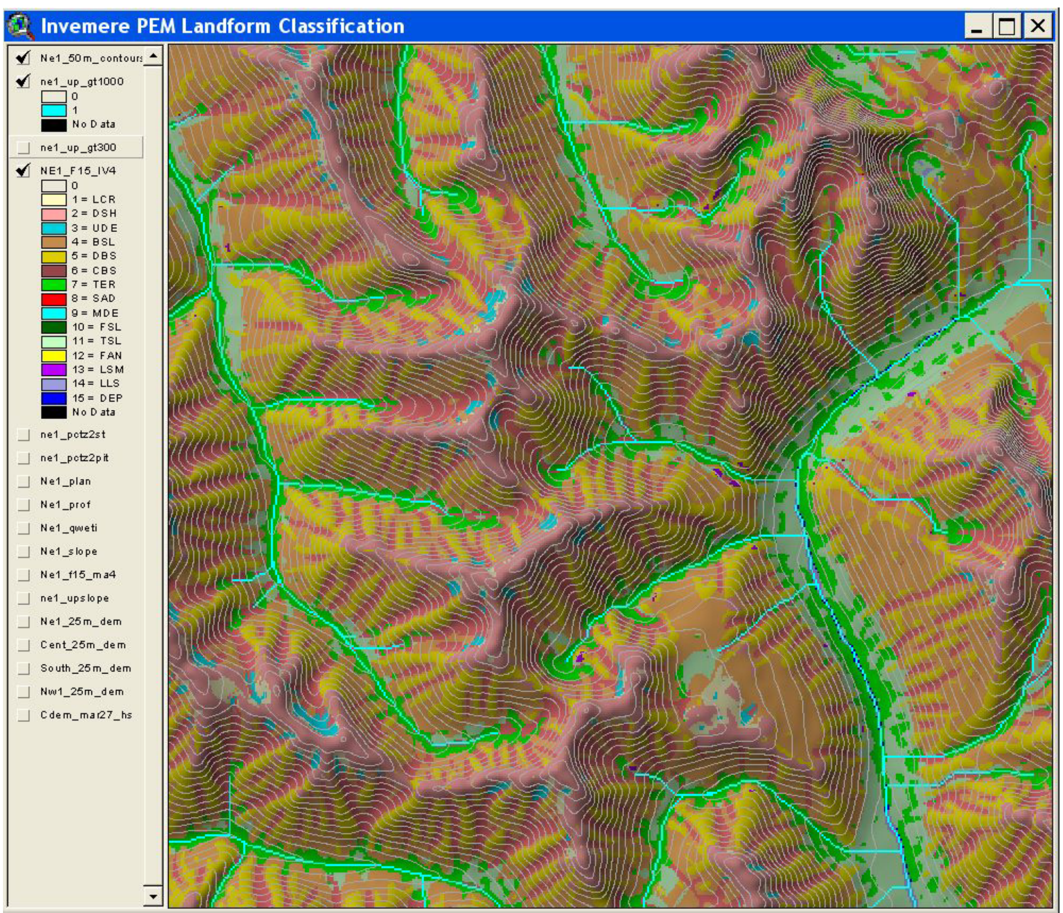


Figure 44. Illustration of a final "hard" LSM landform classification displayed in 2D ArcView 3.2

In general, most users will only be interested in producing maps and visualizations of the final "hard" classification produced by application of the FacetMapR program (Figure 44). They will need to export the integer numbers stored in the fields Max_facet (for option a) or Max_code for options b and c. It is also possible to export the alpha-numeric character code from the field Max_Class and to use this to produce a grid map of unique character values. Occasionally, users may wish to create maps that depict the likelihood value associated with the most likely output class as stored in the field Max_value in the files produced by options b and c. The output file produced by option a provides an opportunity to create grid maps of the relative likelihood of occurrence of each of N defined landform classes, where N is determined by the particular rule based read in and used to define landform classes.

Users can, of course, create their own legends with their preferred color assignments. Visual interpretation and evaluation of the final classification maps is improved by the addition of three dimensional visual clues such as are offered by adding hillshade illumination to the two dimensional map or overlaying vector contours on top of the raster rendering (see Figure 44).

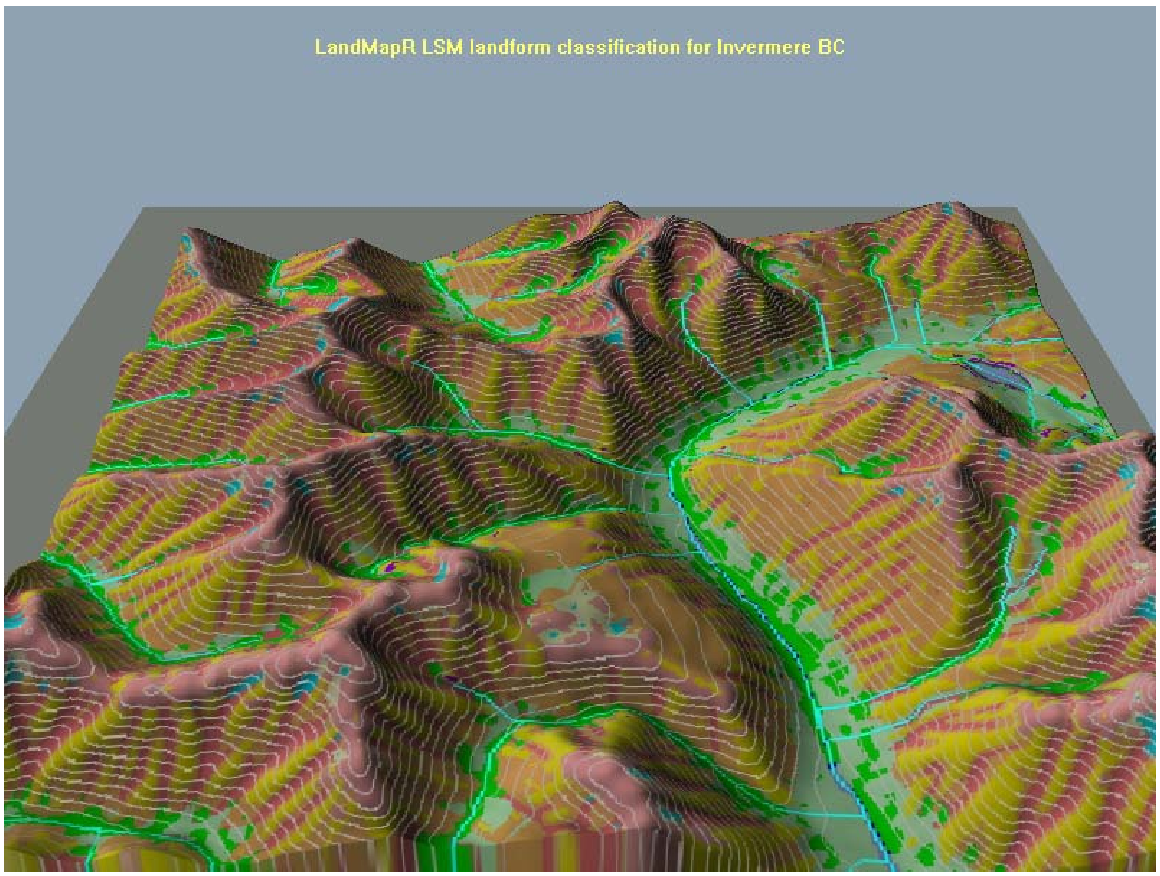


Figure 45. Illustration of a final "hard" LSM landform classification displayed in 3D using the program 3DEM

Over the years, several different software packages have been used to create 3-dimensional perspective views to illustrate LandMapR classification results (figures 45 and 46). For those unable to afford to purchase ArcView's 3D Analyst extension, freeware tools such as 3DEM (http://www.visualizationsoftware.com/3dem.html) and the public domain version of 3DMapper (http://solim.geography.wisc.edu/mapper/index.htm) provide excellent capabilities at no charge. Figure 45 was produced by exporting a 25 m DEM from the ArcView view illustrated in Figure 44 and capturing a screen image of the 2-dimensional view portrayed in Figure 44 using a screen capture utility. The ArcView binary export file (FLT) was easily imported into 3DEM as was the JPG image of the screen capture. 3DEM provides capabilities to create VRML worlds, fly-by animations and rotation animations as well as the static 3D views illustrated in figure 45. It is easy to obtain, easy to learn and is recommended and an excellent tool for visualizing the results produced by FacetMapR or any of the LandMapR suite of programs.

3DMapper represents another excellent and affordable option for creating and saving 3D views of the LandMapR LSM classification (Figure 46). 3DMapper provides easy to use capabilities for importing DEM and orthoimage data in ArcView ASCII grid format to create base files of DEM and image pairs. It is also quite easy to import LandMapR LSM landform classification files into 3DMapper, also as ArcView ASCII grids. Assigning appropriate colors to the integer values is accomplished by reading in an ASCII palette file. Several palette files have been prepared to associate appropriate colors with the integer numbers used by the current standard LandMapR LSM landform classification. These are included on the distribution CD.

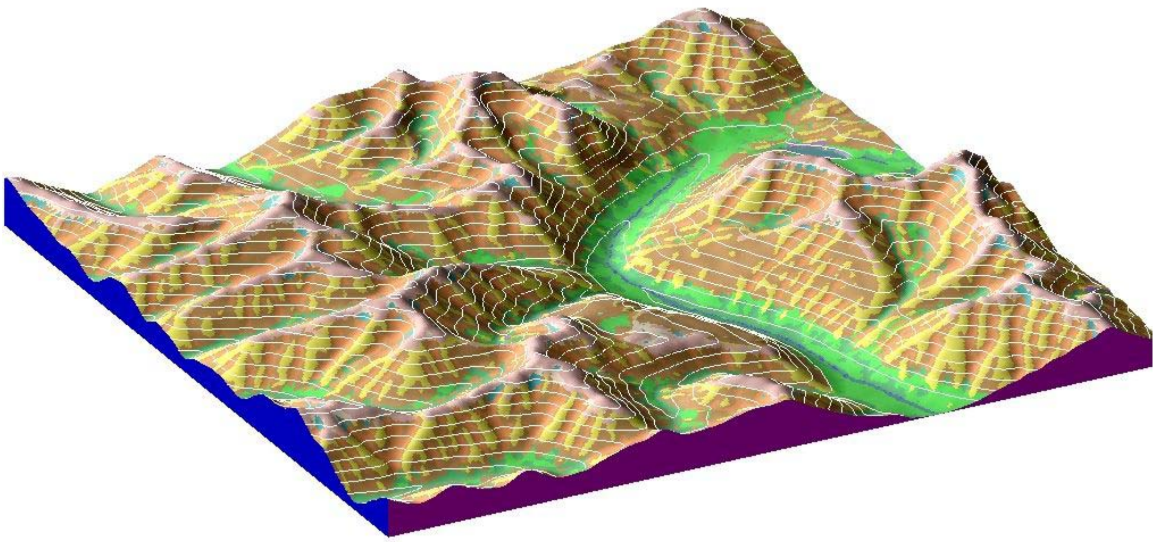


Figure 46. Illustration of a final "hard" LSM landform classification displayed in 3D using the program 3DMapper

It is beyond the scope of this manual to instruct users how to import data into the various programs available for displaying the maps and data that result from applying FacetMapR and related LandMapR programs. The utility program GridReadWriteUtility provided on the distribution disk can be used to convert back and forth between the DBF table format that is used to store all data used by the LandMapR programs and the raw binary format (FLT) used by ArcView and several other widely used GIS programs. Users will have to experiment with reformatting data from the LandMapR DBF format into whatever formats they use to display the data. Assistance and advice will be available to licensed users, if necessary, via e-mail and telephone support.

FacetMapR Summary

FacetMapR was initially created as a custom program for developing, evaluating and refining a single set of fuzzy rules for classifying landform facets. It was subsequently extended to permit automated recognition of a hierarchy of ecological classes in BC. The hierarchical classification uses different rules defined for different zones within an area of interest. The FacetMapR program was kept as general as possible throughout these developments and this has meant that it is quite flexible and can be used to define almost any kind and number of classes of spatial entity within any portion of a region of interest. FacetMapR does, however, retain idiosyncrasies and naming conventions that reflect its early development and initial uses. If users are willing to invest the time and effort to learn how to customize the classification rule tables, they should find that they are able to define almost any type and number of classes they desire for any zone or sub-region of any area of interest to them.

WeppMapR

Purpose

The WeppMapR component computes hydrological spatial entities, specifically those spatial entities required to provide the spatial structure for the WEPP water erosion model.

Input Data

The WeppMapR component uses output from the FlowMapR program as well as the original input elevation data to perform its calculations. WeppMapR will not work properly unless the FlowMapR program has been run successfully to completion prior to running WeppMapR and unless all files produced by application of the FlowMapR program are present in the default folder that is identified as containing the required input data.

As with FlowMapR, WeppMapR operates on input elevation data formatted as a regular (raster) grid. It does not operate on elevation data formatted as irregular x, y, z point data, as a triangular irregular network (TIN) or as vector contours.

Background to the WeppMapR Program

The WEPP model uses mathematical equations to model how much surface runoff will occur in a given segment of the landscape and how, when and where that runoff will travel. A key requirement for running the WEPP model is defining the landscape segments that are used as the basis for computing water budgets and routing water from one segment to the next (Figure 47). The WEPP model uses segments that represent individual hillslopes or portions of hillslopes referred to as overland flow elements (OFEs). Until recently, each hillslope or OFE segment had to be defined by manual means such as through field measurements or tedious interpretation of air photos or topographic maps.

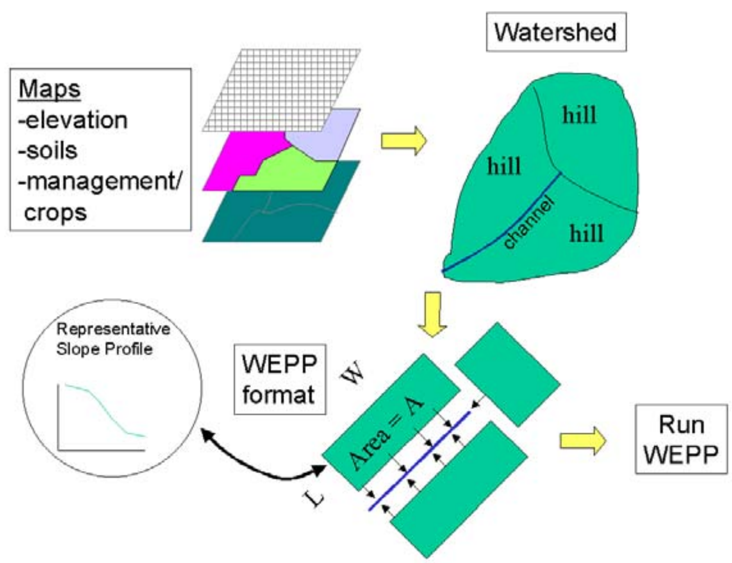


Figure 47. Illustration of the sub-division of space into hill slope segments for the WEPP model (Flanagan et al., 2000)

Renschler (2003) has produced a convenient and useful ArcView extension named GeoWEPP that is integrated seamlessly with the WEPP model and that automatically defines and extracts all of the structural information required to run WEPP from within ArcView. At the time the WeppMapR module described here was developed, the GeoWEPP extension had not yet been developed and its appearance was not anticipated. Consequently, the WeppMapR module described here was developed in order to automatically extract and document the spatial entities required to run the WEPP model for very large data sets covering very large areas. Users specifically interested in running the WEPP model would be best served by acquiring and running GeoWEPP. However, users specifically interested in classifying landforms may find the ancillary data produced by application of WeppMapR to be a useful extension to the basic output produced by the LandMapR toolkit.

Processing Steps Implemented by the WeppMapR Program

The WeppMapR program implements 22 separate steps organized into 22 different program procedures. The role and function of each of the procedures is listed and described below.

Step 1: Creating and populating the master WEPP table (MAKE_WEPP)

The first module of WeppMapR is the procedure Make_WEPP. This module creates an empty DBF table called ID#Wepp (Figure 48). It then copies into this table all of the relevant information on elevation and flow topology computed by the initial FlowMapR program. All processing done by the WeppMapR program is applied to this duplicate data copied from the previously processed DEM data table.

M14wepp

Seqno	Row	Col	Elev	Ddir	Drec	Upslope	Seedtype	Shed_no	Shed_side	Hill_no	Chan_no	Chan_side	Segment_no
1	1	1	-9999.0000	0	1	1	0	0	0	0	0		0
2	1	2	937.5941	6	3	2	0	5979	3	2340	0		0
3	1	3	937.3654	6	4	12	0	5979	3	2340	0		0
4	1	4	937.3209	6	5	13	0	5979	3	2340	0	3	0
5	1	5	937.2789	6	6	535	0	0	0	0	434		5979
6	1	6	937.2778	6	7	550	0	0	0	0	434		5979
7	1	7	937.2767	6	8	551	0	0	0	0	434		5979
8	1	8	937.2756	6	9	563	0	0	0	0	434		5979
9	1	9	937.2756	6	10	565	0	0	0	0	434		5979
10	1	10	937.1732	6	11	577	0	0	0	0	434		5979
11	1	11	937.0315	6	12	578	0	0	0	0	434		5979
12	1	12	936.8940	6	13	579	7	0	0	0	434		5979
13	1	13	936.8054	6	14	1078	2	0	0	0	434		6058

Figure 48. Illustration of the master DBF table (ID#WEPP) used to process grid data to define WEPP spatial entities

Duplicate data copied from the master DEM table (ID#DEM) consist of Seqno, Row, Col, Elev, Ddir, Drec and Upslope. The remaining fields illustrated in Figure 48 are empty when WeppMapR starts to run. These fields are filled with computed data by one or more subsequent WeppMapR modules.

Applying the FlowMapR program to very large data sets can require a very long processing time. Applying the WeppMapR program results in changes to both original elevation values and computed flow directions as stored in the master DEM table. It therefore makes sense to run the WeppMapR program on a copy of the data. This way, if the results from the WeppMapR program are not suitable and there is a desire to re-run the program using a different threshold value to identify channels, it is not necessary to start from scratch and have to re-run the FlowMapR program as well.

Step 2: Nominating cells as belonging to initial channels (MARK_CHANS)

The second module (MARK_CHANS) identifies and marks grid cells as initial channel cells. It also marks cells as channel start (1), end (6) or junction (2) seed types.

The master WEPP table (ID#WEPP) is indexed by elevation and upslope area count from highest to lowest elevation and from lowest to highest upslope area. The file is processed from top to bottom. The algorithm searches for cells that have an upslope area count in excess of a user specified threshold value. If a cell has a value for upslope area in excess of the user specified threshold, and the cell is not already marked as belonging to a channel, the cell is marked as a channel start cell (seed type 1). If a cell is marked as a channel start cell, the algorithm then traces down from the start cell, following the path of steepest downslope flow until it can proceed no further (it terminates at the edge of the data matrix or at a final pit center cell).

Each cell along the downslope flow path followed from the start cell to the end cell is marked with an integer ID number unique to that flow path. If a flow path intersects a previously marked flow path, the algorithm stops flowing down. The cell at which the current flow path intersected a previously marked flow path is marked as a junction cell seed type (2) and the unique ID number for channels is incremented by one to prepare for labeling the next new channel. The algorithm returns to the position in the file occupied by the start cell, skips one cell down from it and proceeds to search for the next potential channel start cell. This process is continues until the bottom of the file is reached and there are no more possible channel start cells.

Step 3: Cutting or burning channels into the DEM (CUT_CHAN)

This module (CUT_CHAN) alters the original DEM elevation values to force all flow in marked channels to follow a continuous down slope gradient. The pit removing procedures employed by the FlowMapR program simply reverse flow directions to direct flow from pit centers upslope to the pour points at which depressions were computed to over spill. This results in situations where flow along a marked channel could be upslope, against gravity. These situations confuse subsequent efforts to label channels in correct topological order, from top to bottom. These procedures use the elevation of the last cell in a channel as the criteria for sorting channels into correct topological order. In order to remove this problem, it is necessary to revise the original elevation values to force all flow to be down-slope in all channels.

The algorithm locates each marked channel start cell and flows down the channel to its terminus. It keeps track of the elevation of the previous cell along the flow path and compares it to the elevation of the current cell. Any time the elevation of the current cell is greater than that of the previous cell, a flag is raised to indicate that upslope flow has begun. The location and elevation of the cell just before the point at which upslope flow began is noted. The algorithm continues tracing along the flow path; checking elevations as it goes, until it encounters a cell with an elevation lower than that of the cell marked as preceding the start of upslope flow. The algorithm then returns to the cell marked as the start of upslope flow and replaces the original elevation with an interpolated value. The new value is based on a linear interpolation between the value of elevation at the cell just before upslope begins and the value of elevation at the next cell along the path that has an elevation lower than this start elevation.

When all start cells have been traced down their respective channels and all channels have been checked, the original elevation values along all channels have all been revised to ensure that all flow in channels is continuously down slope.

Step 4: Merging channels to remove parallel channels (MERGE_CHAN)

A common problem with the D8 algorithm used by FlowMapR to compute flow directions is its tendency to assign parallel flow directions to many adjacent cells. This tendency is particularly evident in landscapes with very subdued relief and consequently with very poorly defined drainage directions and flow regimes.

Initial efforts to compute WEPP channels and hillslopes were complicated by the presence of numerous initial channels that ran exactly parallel to and adjacent to another nominated channel. Having two channels exactly adjacent to one another for a considerable length confounds efforts to identify cells on the left and right sides of channels and to define left and right hillslope areas for all channel segments. It also complicates efforts to identify and mark cells as channel junctions or as cells immediately upslope of a channel junction. It was determined that subsequent procedures would benefit from application of a raster line thinning algorithm aimed at reducing, or entirely removing, (if possible) all occurrences of adjoining parallel channel segments (See Figure 49).

This algorithm traces down each marked channel changing flow directions and drainage record numbers to point into the current channel for any neighboring cell that is marked as belonging to a different channel and that has a flow direction parallel to the current cell. Once a cell has its flow direction changed, it is not permitted to be changed again. This ensures that all cells in one adjacent parallel channel are pointed into a neighbor channel, but that the flow directions are not be changed a second time to point into a different parallel neighbor. Any time a cell has its flow direction changed, its status is changed and it is unmarked as a channel. That way, when the algorithm later flows down along the neighbor channel it does not find any adjacent cells that are marked as belonging to a channel. It will therefore not attempt to change the flow directions of the neighbor cells to force flow into the current channel. This algorithm contains a procedure that checks for potential circular flow before permanently changing the local drainage direction of any cell. Any proposed change in drainage direction that produces circular flow (flow that returns back to the originating cell) is disallowed. This ensures that the channel thinning procedures do not result in the definition of flow networks with closed loops that can result in infinite loops in later programming stages.

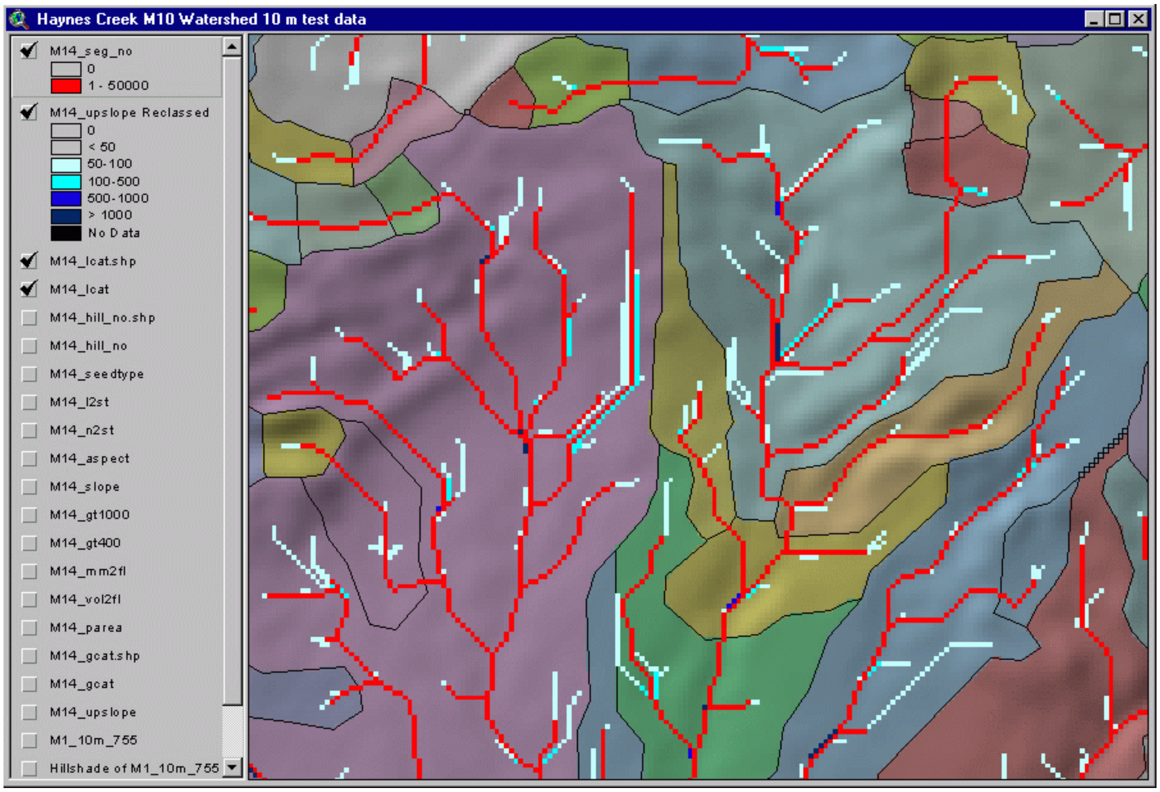


Figure 49. Illustration of the result of thinning and merging adjacent parallel channels to simplify the channel network

The raster line thinning algorithm was found to be effective at removing all obvious instances of adjacent parallel channels (Figure 49). It does, however, permit parallel channels separated by at least 1 cell width to remain. A more powerful line-thinning algorithm is probably desirable, but the current algorithm was judged to be sufficient for our initial purposes.

Step 5: Recomputing upslope area count for all cells (NEW_UPS)

Actions taken by step 4 produce numerous changes in flow directions for cells along or adjacent to channels. These changes invalidate the original values computed for upslope area for the altered cells and for any cells down slope of them along newly altered flow paths. Several subsequent modules require accurate knowledge of correct values for upslope area for all cells, particularly cells along defined channels. It is therefore deemed necessary to re-compute upslope area count for all cells at this point. The new value replaces the original value in the field upslope (Figure 48).

Step 6: Remarking cells as belonging to thinned channels (REMARK_CHAN)

The module (REMARK_CHAN) re-implements the previous MARK_CHANS module to remark cells as belonging to channels and to re-identify seed cells of types 1 (start cell), 2 (junction cell) and 6 (end cell). Cells that have previously been part of marked channels are no longer part of marked channels if the channel has been removed by thinning, or if forcing a channel into an adjoining channel has altered the flow path and reduced the value computed for upslope area for cells down gradient from the altered cells. The new data set of marked channels is always considerably cleaner and lacking in examples of multiple adjoining parallel channel cells.

An additional feature of this module is that it forces all cells that are adjacent to a channel and are not part of a marked channel themselves, to flow into the marked channel that they are adjacent to. This procedure explicitly forces flow to enter a channel as soon as flow arrives at a cell that is adjacent to a marked channel. These flow directions are not absolutely correct relative to known elevations and to down slope flow as computed by the D8 algorithm. These imposed, artificial drainage directions do, however, more accurately reflect human perceptions and models of hillslope and channel topology. If these flow directions were not arbitrarily changed to force flow into channels, most flow would parallel channels for long distances and enter the channel at a limited number of entry points. Such flow regimes are difficult to interpret to assign cells to hillslopes that drain into their associated channel segments. Forcing flow to enter the channel as soon as it approaches it results in definition of more logical and intuitively reasonable hillslopes that are directly connected to the channel segments which they border.

Another routine in this module checks for topological consistency at channel junctions. It goes to all marked channel junctions and identifies the location and number of cells upslope of the junction that belong to marked channels that flow into the junction cell. It tries to limit the number of upslope tributaries to a maximum of three. It forces flow from each upslope tributary to enter the down slope channel segment at the cell with the lowest elevation that is marked as belonging to the down slope channel segment. It then checks the upslope area count of each of the upslope channel cells that flow into each junction. The upslope cell with the greatest upslope area count is marked as a type 7 seed type. This identifies it as the upslope cell of the main branch of the channel upslope of the junction. All other cells on marked channels that flow into the junction are marked as type 3 seed types. These cells are the first cell upslope of a junction on a shorter tributary channel (not the main channel). Type 3 and 7 seed points mark end points for channel segments.

Step 7: Marking cells as representing pit center locations (MARK_PITS)

This module (MARK_PITS) uses data contained in a previously computed table of pit statistics (ID#FILL). This pit table lists all depressions that are computed to occur in the DEM data set and contains details of their location, morphology and hydraulic connectivity. For this module, the only information of interest is the location. Each unique pit center location is identified. The algorithm goes to the cell in the master WEPP table corresponding to the pit center location (PITREC) and marks this cell as being a pit center cell. A value of 5 is entered into the field SEEDTYPE to identify pit center cells. Pit center cells may occur either on a marked channel or off a marked channel. Pits that are on marked channels are treated differently, in later routines, than pits that are not. If a type 5 (pit) seed point is placed on a marked channel, the cell immediately down slope from it along the channel fed from the depression is marked as a type 8 seed point. Type 8 seed cells are cells immediately down stream from either a depression or a split-type seed cell (see below). They identify points at which it is necessary to recognize the start of a new channel segment.

Step 8: Splitting channels into segments of a specified length (SPLIT_SEGS)

This module checks each marked channel to measure the distance between seed points that have been inserted to identify any of start cells (1), pit cells (5), upslope of junction cells (3, 7) or end cells (6). Any channel segment that is longer than a user specified distance between marked seed points is split to create two or more channel segments that are shorter in length.

Originally channel splitting was done in response to an incorrect interpretation of the WEPP documentation by LandMapper. However, having split WEPP channel segments into lengths no longer than a user-specified distance, more or less in error, it was observed that this splitting resulted in smaller hillslope entities that were more internally homogeneous, particularly with respect to variation in shape and aspect. Constraints on hillslope size and morphological variation were deemed to be beneficial and the channel splitting process was therefore retained.

Any channel segment longer than a user specified distance (typically 200-400 m) is therefore spilt into n segments, where n is computed according to n = INT((channel segment length/ user specified maximum segment length)+1). The number 4 is inserted at a split cell to indicate that this needs to be considered as an end point for the channel segment that leads into it. The number 8 is inserted at the cell immediately down slope of the type 4 split cell to indicate that it is necessary to recognize the beginning of a new channel segment at this cell.

A more robust procedure for splitting channels into shorter segments is planned. This new procedure will consider channel shape and will attempt to locate split points at bends in the channel or other prominent changes in channel direction or shape.

If users do not wish to split channel into shorter segments they need only enter a very large value for maximum segment length such that no segment is ever likely to be longer than this entered length. An example might be to enter a maximum segment length of 10,000 m for an area that was only 5,000 m across. No channel in this area is likely to have any segments longer than 10,000 m and so no channel segments are likely to be split into shorter segments.

Step 9: Forcing all cells adjacent to a channel to flow into it (FLOW2CHAN)

It is not entirely clear why, but it was found to be necessary to run this procedure a second time at this point in the processing. The procedure forces all cells that are adjacent to a channel and are not marked as a channel cell to flow into the adjacent marked channel. It appears that some of the changes in flow direction implemented in steps 6 and 8 above overrode previous efforts to ensure that all cells adjacent to channels drained into a cell with a lower elevation that was part of a marked channel. Re-running this procedure at this juncture seemed to be required to reset this condition to true for all cells adjacent to marked channels.

Step 10: Identifying and labeling marked channel segments (CALC_SEGS)

This module (CLAC_SEGS) is a key component of the procedures to automatically create and describe WEPP spatial entities. This module traces down every marked channel again, starting with the channel segment with the highest elevation start cell (seed type 1) and following it to its terminus, then finding the next highest type 1 start cell and so on.

As it traces down each channel it identifies, numbers and stores data about every unique channel segment it encounters. A unique channel segment begins at any seed point labeled as a type 1, 2, 5, or 8 and ends at any cell labeled as a type 3, 7, 4, 5, or 6 (Figure 51). The procedure stores data on the location, elevation and seed type of the cell at the start of the segment and the end of the segment, the length of the segment, whether the segment ends at a pit and the initial unique identifier for the segment.

The segment table (Figure 50) records most of the information required to compute and store complete descriptions of the topological connectivity of the individual segments (e.g. which segments flow into which other segments) and several of the important attributes of each segment (e.g. length, overall slope from start to bottom). The field START_DDIR records the code for drainage direction assigned to the first (start) cell in each segment. This information was required to assist in computing whether upslope segments flowing into any given segment entered from the top, left or right. The field DOWN_SEG identifies the unique sequence number of the cell in the master WEPP table into which each segment end cell drains. A logical field (IMPOUND) identifies whether the current channel segment is actually an impoundment.

M14segs

Initial_id	Start_type	Start_cell	Start_row	Start_col	Start_elev	Start_ddir	End_type	End_cell	End_row	End_col	End_elev	Len_cells	Len_meters	Down_seg
1	5	39802	67	202	982.3011	3	5	39802	67	202	982.3011	1	10	4763
2	5	39802	67	202	982.3011	3	5	39802	67	202	982.3011	1	10	4775
3	8	40403	68	203	982.2990	6	3	30215	51	215	973.0126	20	200	4821
4	5	160	1	160	979.9965	5	5	160	1	160	979.9965	1	10	4765
5	5	160	1	160	979.9965	5	5	160	1	160	979.9965	1	10	4765
6	1	118299	198	99	979.8419	2	4	129099	216	99	976.9833	19	190	4804
7	1	106279	178	79	979.7014	8	4	94883	159	83	974.7014	20	200	4774
8	1	82238	138	38	978.7338	4	4	82220	138	20	965.7224	19	190	5649
9	1	118867	199	67	978.6370	2	7	125466	210	66	975.4341	12	120	4801
10	1	123679	207	79	978.4289	4	3	130265	218	65	969.6098	15	150	4877
11	1	16963	29	163	977.8193	4	4	18750	32	150	970.8517	14	140	4844
12	1	43972	74	172	977.3669	2	3	52973	89	173	969.9122	16	160	4803
13	1	25381	43	181	976.7680	8	7	22384	38	184	974.8232	6	60	4783
14	8	129699	217	99	976.7390	2	4	139901	234	101	969.0423	18	180	4857
15	8	121859	204	59	976.5333	3	3	125464	210	64	975.2784	8	80	4801
16	5	121858	204	58	976.4341	6	5	121858	204	58	976.4341	1	10	4768
17	5	121858	204	58	976.4341	6	5	121858	204	58	976.4341	1	10	4770
18	1	22378	38	178	975.9858	6	3	21784	37	184	974.7148	7	70	4783
19	1	99691	167	91	975.8422	9	4	91895	154	95	970.7034	14	140	4825

Figure 50. Illustration of the information about channel segments computed and stored in the segment table (ID#SEGS)

Figures 51 and 52 present examples of the results obtained by application of the procedures for locating seed cells (Figure 51) and labeling individual channel segments between identified seed cells with unique channel segment ID numbers (Figure 52). In Figure 51 it can be seen that start type seed cells (1) occur at the start of each unique channel and junction type seed cells (2) occur at each channel junction. Tributary branch (3) and main branch (7) seed cells occur immediately upslope from each junction type cell. Impoundment type cells (5) and split type cells (4) break up linear segments into shorter segments and always have type 8 start cells below them. Each linear channel segment has a unique color associated with it in Figure 52 and has a unique ID number associated with it in the channel segment table (Figure 50).

WEPP rules stipulate that impoundments can be fed by up to three channels or by a single hillslope, but not by more than one hillslope or by both channels and hillslopes. These rules necessitated enforcing some special conditions on channel segments identified as occurring at type 5 depression seed points. Duplicate records were entered into the segment table (ID#SEGS) for cells labeled as pit type seed cells. The first occurrence of a channel segment that began and ended at the same type 5 seed point was considered to describe a channel segment one cell in length. This short, isolated channel segment 1 cell in length was then considered to direct its outflow into a notional impoundment cell that occupied the same physical space as the channel segment cell. In this way, any channels or hillslopes computed to deliver flow to the impoundment seed point location were considered to be flowing into a WEPP channel segment. This WEPP channel segment, in turn, could only flow into the notional impoundment entity. Every impoundment entity is therefore fed by one, and only one, channel segment. No impoundments are ever fed by hillslopes as any potential hillslopes feed the channel segment that occupies the same physical space as the impoundment cell. These special mental gymnastics were required to avoid otherwise unavoidable situations in which impoundment cells could be fed by multiple hillslopes or by an illegal combination of channels and hillslopes.

This module performs another important calculation as it traces down each channel and labels and describes each channel segment. At each cell along each channel segment, all neighbor cells are scanned to identify neighbor cells that are not marked as channel cells and that flow into the current channel at the current cell. A procedure is invoked that determines the relative direction of each neighbor cell with respect to the orientation of the channel passing through the current cell. Every neighbor cell that is not marked as a channel cell and that drains into the current channel cell is marked as draining in from either the left (3), right (2) or top (1) of the channel (Figure 53). Only cells labeled as type 1 start cells can accept input from a top location, so all type 1 cells are marked as receiving input from the top. All other cells are marked as receiving inflow from the left or right. Data indicating which side of a channel a cell drains from (1 = top, 2 = right, 3 = left) are stored in the field CHAN_SIDE in the master WEPP table.

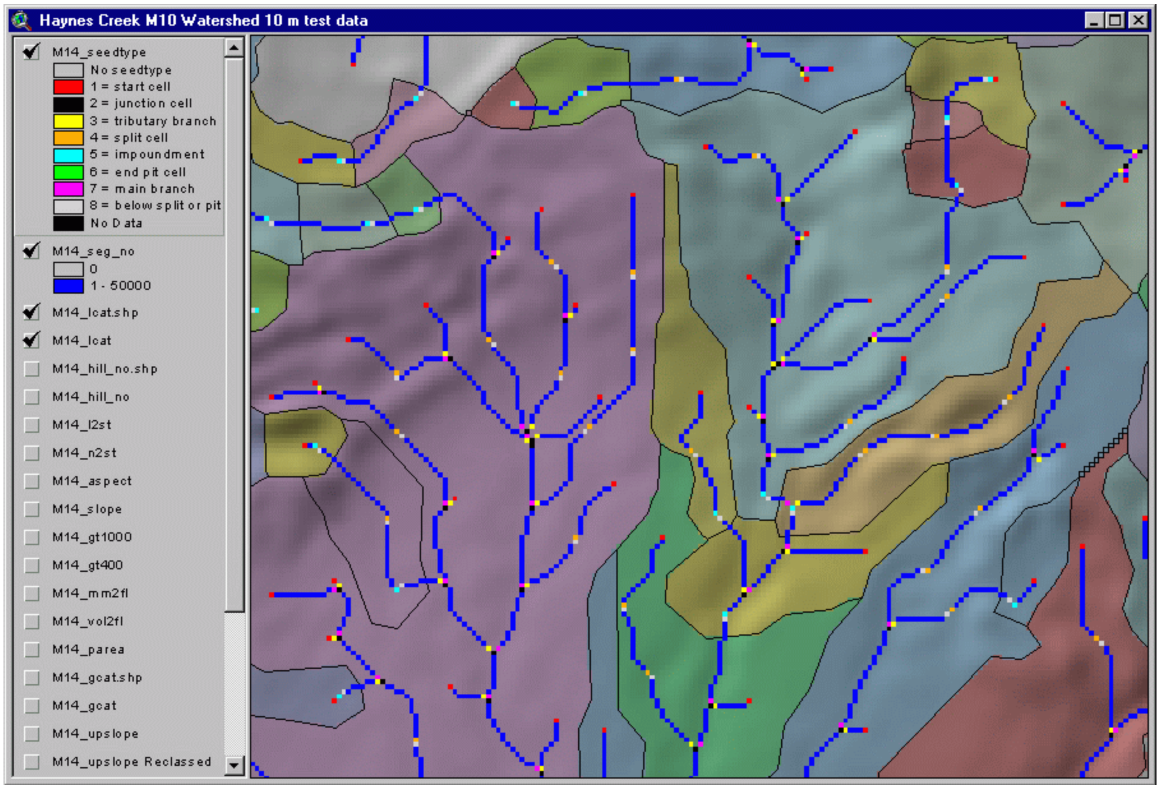


Figure 51. Illustration of the result of locating seed point identifiers at critical locations along a channel network

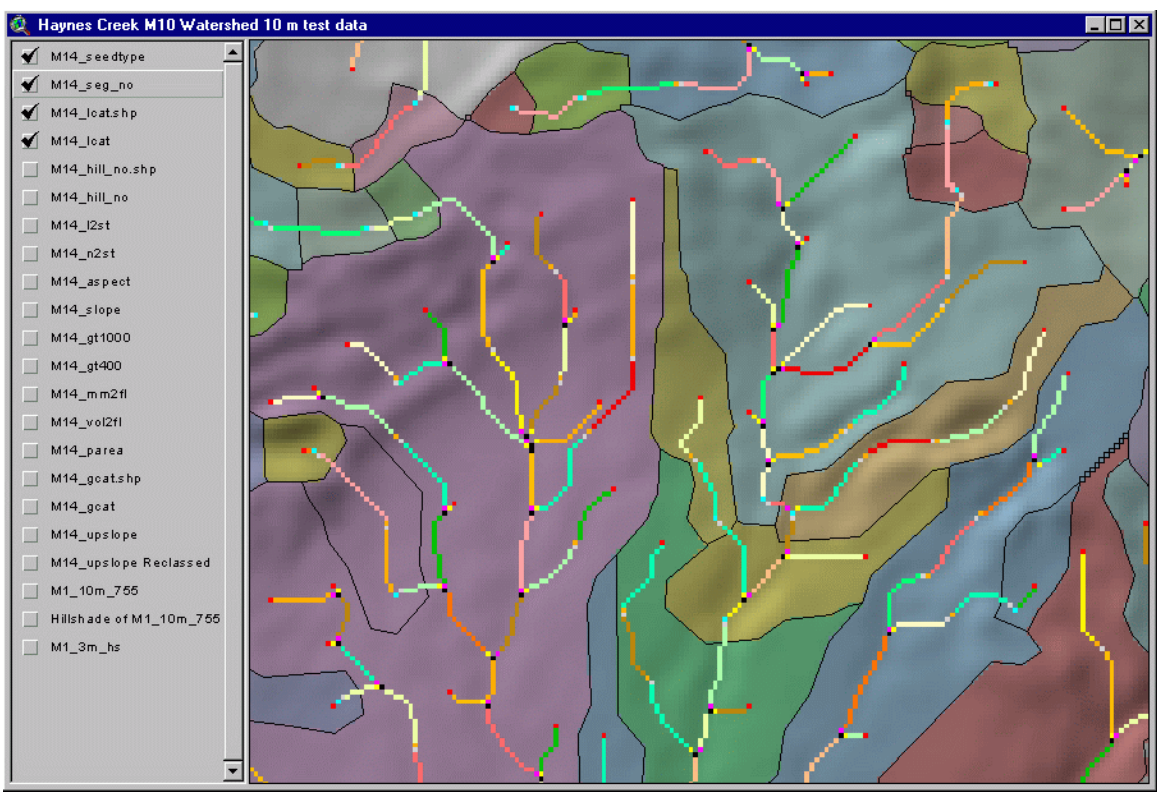


Figure 52. Illustration of results of identifying and labeling individual channel segments between seed cells

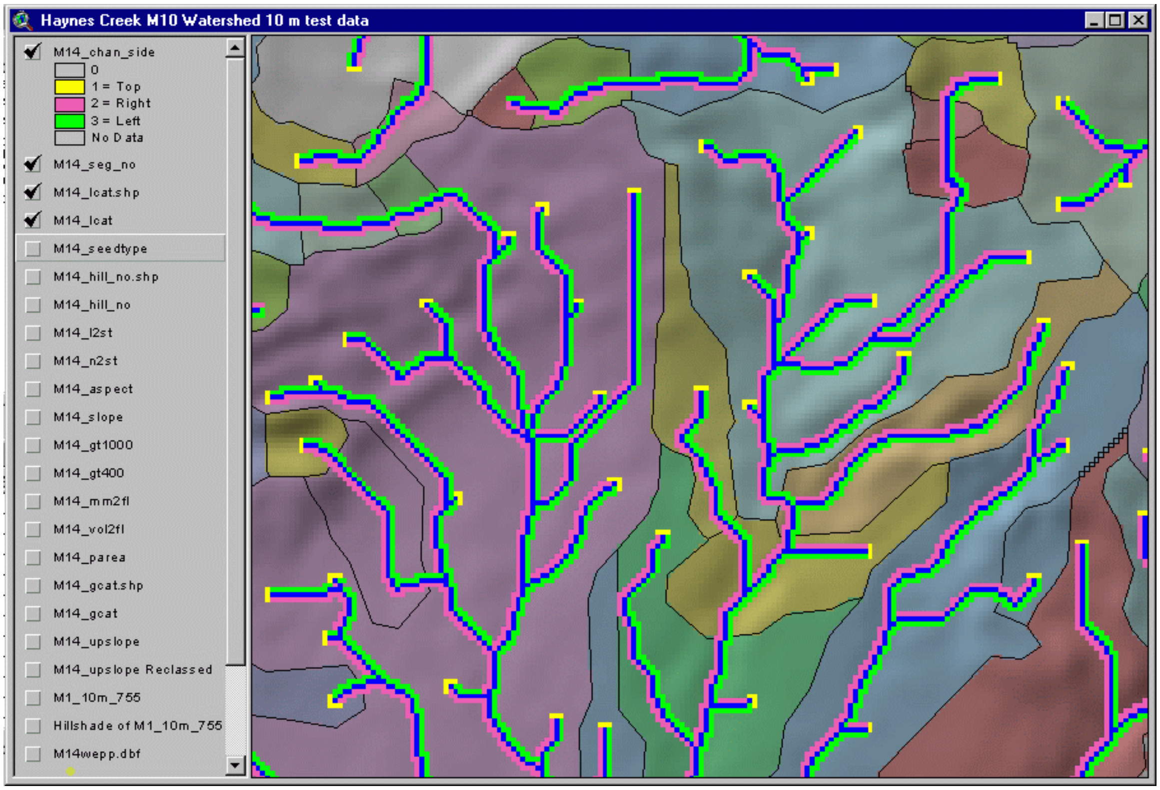


Figure 53. Illustration of the result of computing the direction from which flow enters a marked channel segment

Step 11: Sorting channel segments into topological order (ORDER_SEGS)

This module sorts the WEPP channel segments identified in step 10 above into the correct topological order. Channel segments in the original ID#SEGS table (Figure 50) are indexed in descending order of elevation from the segment with the highest elevation for its end cell to the segment with the lowest elevation. The original ID number for each segment (INITIAL_ID) is supplemented by a new number (SORT_ORDER) that is assigned based on the sequential position of any given segment in the indexed table (Figure 54). This ensures that the number now assigned to any channel segment is smaller than the ID number assigned to any other channel segment lower in the landscape into which the current segment might possibly contribute flow.

Ignoring the data in the field FINAL_ID for the present, it is clear from Figure 54 that channel segments are ordered from highest to lowest relative to the elevation of the end cell for every channel segment. At this point, the module labels every cell in the master WEPP table that lies along a marked channel with an initial channel segment ID number that corresponds to the value stored in the field SORT_ORDER. The algorithm goes directly to the record number of the start cell (START_CELL) associated with each numbered channel segment and traces down the flow path from that cell until it reaches the cell identified as the end cell. At each cell along the path, it enters the integer ID number corresponding to SORT_ORDER into the field SEGMENT_NO in the master WEPP table (Figure 48).

Again, special rules are followed in the case of channel segments that are labeled as impoundments. Cells are labeled with the ID number of only the first (virtual channel segment) of the pairs of channel segment ID numbers for the same physical location. This is true except if there is no channel defined for the location, in which case, the cell is labeled with the ID number of the impoundment.

M14segs

Initial_id	Sort_order	Final_id	End_elev	Down_seg	Drain_rec	Start_type	Start_cell	Start_row	Start_col	Start_elev	Start_ddir	End_type
1	1	4762	982.3011	4763	39802	5	39802	67	202	982.3011	3	5
2	2	4763	982.3011	4775	40403	5	39802	67	202	982.3011	3	5
4	3	4764	979.9965	4765	160	5	160	1	160	979.9965	5	5
5	4	4765	979.9965	4765	160	5	160	1	160	979.9965	5	5
6	5	4766	976.9833	4804	129699	1	118299	198	99	979.8419	2	4
16	6	4767	976.4341	4768	121858	5	121858	204	58	976.4341	6	5
17	7	4768	976.4341	4770	121859	5	121858	204	58	976.4341	6	5
9	8	4769	975.4341	4801	126065	1	118867	199	67	978.6370	2	7
15	9	4770	975.2784	4801	126065	8	121859	204	59	976.5333	3	3
13	10	4771	974.8232	4783	21785	1	25381	43	181	976.7680	8	7
18	11	4772	974.7148	4783	21785	1	22378	38	178	975.9858	6	3
7	12	4773	974.7014	4774	94282	1	106279	178	79	979.7014	8	4
24	13	4774	973.5568	4813	83475	8	94282	158	82	974.5986	7	4
3	14	4775	973.0126	4821	29616	8	40403	68	203	982.2990	6	3
29	15	4776	972.9985	4821	29616	1	33823	57	223	973.9446	7	7
35	16	4777	972.7165	4778	54745	5	54745	92	145	972.7165	3	5
36	17	4778	972.7165	4781	55346	5	54745	92	145	972.7165	3	5
40	18	4779	972.5387	4780	49940	5	49940	84	140	972.5387	7	5

Figure 54. Illustration of the relevant portion of the segment table showing segments ordered by decreasing elevation

At this point, the module processes the data in the segment table (Figure 54) to identify and record values for the down slope segment into which each segment drains (DOWN_SEG) and the record number of the first cell in the down slope segment into which each segment drains (DRAIN_REC). For all but impoundment cells, this is a simple matter of going to the cell in the master WEPP table identified by the value stored in the field END_CELL in the segment table (Figure 12). Once at this cell, the drainage direction is followed to move directly to the down slope cell that this end cell drains into. This is the DRAIN_REC. The recently added value for SEGMENT_NO is read for this cell and is transferred to the field DOWN_SEG in the segment table (Figure 54). Every channel segment now knows the integer ID value that identifies the channel segment that it drains to (DOWN_SEG) and knows the sequential record number that identifies the location of the first cell in the down slope channel segment that it drains into (DRAIN_REC).

Step 12: Recomputing drainage direction codes for all cells (REDO_DDIR)

Up until this point, all changes made to drainage directions are effectuated by changing the value stored in the field DREC for any given cell to direct its flow into a different cell with a different record number. It is desirable to update the value stored in the field DDIR for all cells for which drainage directions have been changed. A later procedure requires a correct value for the drainage direction assigned to the first cell in each channel segment (the start cell). This value is used to determine the direction of flow from the start cell of each channel segment into the next down slope cell. The direction of flow is needed to assess the relative direction from which upslope segments enter any given down slope segment at that segment's start cell. The REDO_DDIR procedure is applied to re-compute correct drainage directions for all cells in the master WEPP file, rather than just channel start cells. This ensures that all cells have correct codes for drainage direction after all of the changes that have been made to initial drainage directions by previous procedures.

Step 13: Adding a value for drainage direction for start cells (ADD_DDIR)

Once the values for drainage direction (DDIR) have been corrected in the master WEPP table, the value for START_DDIR is added to the segment table (Figure 54). This is done by going directly to the record number in the master WEPP table specified in the field START_CELL in the segment file (Figure 54). The value for DDIR for this cell is recorded and transferred to the field START_DDIR in the segment file.

Step 14: Linking upper to lower channel segments (FIND_UPSEGS)

This module (FIND_UPSEGS) processes the segment file to link each down slope segment with all upslope segments that drain into it. At the same time, segments that flow into a given down slope segment are identified as flowing in from the top, left or right. Additionally, if the segment flowing into the current down slope segment is marked as an impoundment, it is determined if the impoundment enters the current segment from the top, left or right and this information is stored in the appropriate field in the segment table (Figure 55).

M14segs

Initial_id	Sort_order	Final_id	Down_seg	Left_seg	Right_seg	Center_seg	Left_rec	Right_rec	Center_rec	Left_imp	Top_imp	Right_imp	Impound
1	1	4762	4763	0	0	0	0	0	0				F
2	2	4763	4775	0	0	4762	0	0	39802				T
4	3	4764	4765	0	0	0	0	0	0				F
5	4	4765	4765	0	0	4764	0	0	160				T
6	5	4766	4804	0	0	0	0	0	0				F
16	6	4767	4768	0	0	0	0	0	0				F
17	7	4768	4770	0	0	4767	0	0	121858				T
9	8	4769	4801	0	0	0	0	0	0				F
15	9	4770	4801	0	0	4768	0	0	121858		4768		F
13	10	4771	4783	0	0	0	0	0	0				F
18	11	4772	4783	0	0	0	0	0	0				F
7	12	4773	4774	0	0	0	0	0	0				F
24	13	4774	4813	0	0	4773	0	0	94883				F
3	14	4775	4821	0	0	4763	0	0	39802		4763		F
29	15	4776	4821	0	0	0	0	0	0				F
35	16	4777	4778	0	0	0	0	0	0				F
36	17	4778	4781	0	0	4777	0	0	54745				T
40	18	4779	4780	0	0	0	0	0	0				F

Figure 55. Illustration of relevant portions of the segment table indicating upper segments linked to lower segments.

Figure 55 illustrates how upper segments are linked to lower segments. For example, segment 4762 drains into segment 4763. It is a channel and is the only entity that drains into segment 4763 so it must enter from the top (see CENTER_SEG for FINAL_ID 4763).

In Figure 55, FINAL_ID 4763 is an impoundment entity and it flows to DOWN_SEG 4775. Segment 4763 is listed as being a TOP_IMP for FINAL_ID 4775. It is a relatively simple process to locate the FINAL_ID record that corresponds to the value stored in the field DOWN_SEG for an upper segment and to go to that record. It is then also relatively simple to compute whether the upslope segment enters the down slope segment from the top, left or right. Once this is computed, the value of FINAL_ID from the upslope segment is entered into the appropriate field for the down slope segment record (e.g. one of LEFT_SEG, RIGHT_SEG, CENTER_SEG). This is repeated for every record in the segment table, to identify and go to the record number for the segment that it drains to and to insert a reference to itself as an upslope segment that drains into this down slope segment.

It should be noted that all original calculations are done using identifying values that correspond to the value stored in the field SORT_ORDER. It is only later, after the total number of WEPP hillslopes is known that the FINAL_ID number can be assigned to all channel segments. Once the FINAL_ID number is assigned to all channel segments, the values in the various fields in the segment table (LEFT_SEG, CENTER_SEG, RIGHT_SEG, LEFT_IMP, TOP_IMP, RIGHT_IMP) are updated from referencing the initial SORT_ORDER number to referencing the FINAL_ID number.

Step 15: Computing and labeling WEPP hillslope segments (HILL_SHEDS)

The module HILL_SHEDS computes an integer ID number for WEPP hillslope for every cell in a raster DEM. This is another key module in the program WeppMapR. The process relies on previously computed results in which each cell along marked channels was labeled with an integer ID number corresponding to the sequential ID number for WEPP channel segments ordered from highest to lowest (see Step 11).

M14wepp

Seqno	Row	Col	Elev	Ddir	Drec	Upslope	Seedtype	Shed_no	Shed_side	Hill_no	Chan_no	Chan_side	Segment_no
1	1	1	-9999.0000	0	1	1	0	0	0	0	0		0
2	1	2	937.5941	6	3	2	0	5979	3	2340	0		0
3	1	3	937.3654	6	4	12	0	5979	3	2340	0		0
4	1	4	937.3209	6	5	13	0	5979	3	2340	0	3	0
5	1	5	937.2789	6	6	535	0	0	0	0	434		5979
6	1	6	937.2778	6	7	550	0	0	0	0	434		5979
7	1	7	937.2767	6	8	551	0	0	0	0	434		5979
8	1	8	937.2756	6	9	563	0	0	0	0	434		5979
9	1	9	937.2756	6	10	565	0	0	0	0	434		5979
10	1	10	937.1732	6	11	577	0	0	0	0	434		5979
11	1	11	937.0315	6	12	578	0	0	0	0	434		5979
12	1	12	936.8940	6	13	579	7	0	0	0	434		5979
13	1	13	936.8054	6	14	1078	2	0	0	0	434		6058

Figure 56. Illustration of how data in the master WEPP file are used to permit calculation of WEPP hillslope numbers

The module traces down flow paths following drainage directions (DREC) from every cell until the flow path reaches a cell marked as a channel segment. Such cells have non-zero values in the field SEGMENT_NO. The algorithm records the value stored in the field CHAN_SIDE for the last cell with a positive value for the variable CHAN_SIDE that it traversed before encountering a marked channel segment cell. The algorithm then knows the ID number of the channel segment that the start cell (and its entire flow path) drains to and the direction from which the flow path enters the channel (left, right or top). This information is recorded and the algorithm returns to the start cell and traces down the flow path again. It stores the value for the ID number of the channel segment that it drains to in the field SHED_NO and the code for the direction from which it enters the channel in the field SHED_SIDE. Any flow path that intersects a cell that is already labeled with a value for SHED_NO and SHED_SIDE simply stops going down slope. It records these stored values and assigns them to all previously unlabeled cells along the flow path.

A second algorithm is run once all cells have been linked to their respective channel segments and identified as entering the channel from the left, right or top. This algorithm sorts the master WEPP file by increasing SHED_NO and increasing SHED_SIDE value. WEPP hillslope ID numbers are then computed for each unique combination of assigned SHED_NO and SHED_SIDE values. WEPP hillslope ID numbers begin with one (1) for the first hillslope and progress sequentially from there. The first WEPP hillslope is the hillslope that drains into the lowest numbered WEPP channel segment (the segment with the highest end cell elevation) from the top (SHED_SIDE value 1). The next drains into the same channel segment from the right (SHED_SIDE value 2) and then the left (SHED_SIDE value 3) (if present). This process continues until all cells have been assigned WEPP hillslope ID values (stored in the field HILL_NO).

Step 16: Renumbering channel and impoundment segments (RENUM_SEGS)

It is only at this point that the total number of WEPP hillslopes in the DEM area is known. Once the total number of WEPP hillslope entities is known, it is necessary to renumber all previously numbered WEPP channel segment and impoundment entities. WEPP requires that channels and impoundments have ID numbers that are larger than the largest ID number assigned to a WEPP hillslope. The WEPP channel and impoundment entities are originally numbered using values that correspond to the field SORT_ORDER in the WEPP segment file. It is now a relatively simple process to renumber all fields that reference a SORT_ORDER ID number with a value of SORT_ORDER+MAX_HILLS, where MAX_HILLS is the maximum number of known WEPP hillslopes. Channel segments are now numbered starting with a value that is one greater than the maximum number of WEPP hillslopes in the data set and incremented by 1 for each subsequent down slope channel segment or impoundment entity.

The location and extent of all WEPP spatial entities of interest (hillslopes, channels and impoundments) has now been defined. What remains is to compute the morphological attributes of these spatial entities and to reformat the data into formats suitable for direct input into the WEPP model.

Step 17: Building the WEPP structure file for all WEPP entities (BUILD_STRU)

This module (BUILD_STRU) reformats data contained in the channel segment file into a format suitable for direct input into the watershed structure file required by WEPP.

Flanagan and Livingston (1995) provide the following description of a WEPP watershed file.

- The WEPP watershed structure file describes the watershed configuration. For each channel element or impoundment, it indicates what hillslopes, channels and/or impoundments are draining into it from the top or laterally from the left or right. Each element in the watershed is given an ID number. The numbers need to comply with the following rules:
    - All hillslope ID numbers are attributed first, i.e. channel or impoundment numbers are always greater than those of hillslopes.
    - Any upstream element of a channel or impoundment has a lesser ID number than the ID number of the channel or impoundment itself.

This module processes previously computed data to identify every channel or impoundment element using a unique, sequential ID number (ELEMENT_NO) (see Figure 57). Channel type elements (2) are distinguished from impoundment type elements (3) in the field ELE_TYPE. The unique ID numbers of any upslope WEPP hillslopes, channel segments or impoundments that drain into a given channel or impoundment entity are indicated in the fields left, right or top hill, channel or impoundment in the table (Figure 57).

M14stru

Element_no	Ele_type	Left_hill	Right_hill	Top_hill	Left_chan	Right_chan	Top_chan	Left_imp	Right_imp	Top_imp	Comment
4762	2	2	1	0	0	0	0	0	0	0	Element No: 4762 channel
4763	3	0	0	0	0	0	4762	0	0	0	Element No: 4763 impoundment
4764	2	5	4	3	0	0	0	0	0	0	Element No: 4764 channel
4765	3	0	0	0	0	0	4764	0	0	0	Element No: 4765 impoundment
4766	2	8	7	6	0	0	0	0	0	0	Element No: 4766 channel
4767	2	10	9	0	0	0	0	0	0	0	Element No: 4767 channel
4768	3	0	0	0	0	0	4767	0	0	0	Element No: 4768 impoundment
4769	2	13	12	11	0	0	0	0	0	0	Element No: 4769 channel
4770	2	15	14	0	0	0	0	0	0	4768	Element No: 4770 channel
4771	2	18	17	16	0	0	0	0	0	0	Element No: 4771 channel
4772	2	21	20	19	0	0	0	0	0	0	Element No: 4772 channel
4773	2	24	23	22	0	0	0	0	0	0	Element No: 4773 channel
4774	2	26	25	0	0	0	4773	0	0	0	Element No: 4774 channel
4775	2	28	27	0	0	0	0	0	0	4763	Element No: 4775 channel

Figure 57. Illustration of a portion of the database table containing data formatted as required for a WEPP structure file

At this point, the WeppMapR program has computed the location and extent of all spatial entities required by the WEPP model, namely hillslopes, channels and impoundments. It has also explicitly computed the hydrological connectivity from each WEPP hillslope into its associated channel or impoundment and from each channel or impoundment into the down slope channel or impoundment into which it drains. These data are stored in a database table in a format suitable for direct input into a WEPP structure file (Figure 57).

Subsequent modules are oriented towards computing and describing the morphological characteristics of the WEPP hillslopes and channels in terms of representative hillslope and channel profiles and impoundment stage-area-length relationships. Several fields are added to the master WEPP table to facilitate computation and storage of data pertaining to morphological attributes of WEPP spatial entities (see Figure 58).

Step 18: Computing slope and aspect for each DEM grid cell (WEPP_FORM)

This module (WEPP_FORM) computes slope gradient (SLOPE) (Figure 59) and aspect (ASPECT) for each grid cell in the DEM using algorithms described by Eyton (1991). The code for this module was extracted from the FormMapR program described previously.

M14wepp

Seqno	Row	Col	Elev	Drec	Ddir	Upslope	Segment_no	Hill_no	Slope	Aspect	N2st	L2st	Z2st
1	1	1	-9999.0000	1	0	1	0	0	0.000	360	0	0.0	0.000
2	1	2	937.5941	3	6	2	0	2340	0.000	360	3	30.2	0.315
3	1	3	937.3654	4	6	12	0	2340	4.316	43	2	20.0	0.087
4	1	4	937.3209	5	6	13	0	2340	2.777	34	1	10.0	0.042
5	1	5	937.2789	6	6	535	5979	0	1.929	11	0	0.0	0.000
6	1	6	937.2778	7	6	550	5979	0	1.827	354	0	0.0	0.000
7	1	7	937.2767	8	6	551	5979	0	1.805	2	0	0.0	0.000
8	1	8	937.2756	9	6	563	5979	0	1.815	15	0	0.0	0.000
9	1	9	937.2756	10	6	565	5979	0	1.717	31	0	0.0	0.000
10	1	10	937.1732	11	6	577	5979	0	1.971	38	0	0.0	0.000
11	1	11	937.0315	12	6	578	5979	0	2.124	42	0	0.0	0.000
12	1	12	936.8940	13	6	579	5979	0	1.978	35	0	0.0	0.000
13	1	13	936.8054	14	6	1078	6058	0	1.734	21	0	0.0	0.000
14	1	14	936.7719	15	6	1080	6058	0	1.630	6	0	0.0	0.000
15	1	15	936.7632	16	6	1094	6058	0	1.693	1	0	0.0	0.000

Figure 58. Illustration of the expanded master WEPP table with fields for storing morphological variables for each cell

Step 19: Computing distance to channel for each dem grid cell (WEPP_LEN)

This module (WEPP_LEN) computes several measures of distance to channel for each grid cell in a DEM. It simply traces down flow paths from each grid cell, following drainage directions, until the flow path reaches a cell marked as belonging to a WEPP channel segment (SEGMENT_NO). It notes the location and elevation of each cell along each flow path and the location and elevation of the cell at which the flow path first enters the marked channel. Using this information it computes several different values for distance from cell to channel (Figure 58). N2ST records the number of grid cells traversed in flowing from any given cell to the first cell marked as a channel cell. L2ST records the line of sight distance from each grid cell to the cell at which it enters a marked channel. The value for L2ST is modified to account for the vertical change in elevation between each grid cell and the cell at which the flow path from it enters a marked channel (Z2ST). All three measures of distance to channel are computed and stored, but only the variable N2ST is subsequently used to describe WEPP hillslopes (Figure 60).

Step 20: Computing hillslope profiles for each wepp hillslope (HILL_STATS)

This module (HILL_STATS) computes and stores all data required to describe notional hillslope profiles for each WEPP hillslope. The notional hillslope profiles do not reflect the geometry of any specific flow path within a WEPP hillslope. Rather they represent a statistical sample that portrays mean values for slope gradient and aspect at different relative distances along a notional, or representative, hillslope profile.

The module extracts to a temporary working file (Figure 61) data on slope, aspect and distance to channel (N2ST) for all cells in a given WEPP hillslope entity. The file is indexed by ascending value of distance to channel (N2ST). The largest value of N2ST for the hillslope is determined (MAX_N2ST). A value for relative distance along a notional hillslope profile (from top to bottom) is computed according to REL_N2ST = 1 - [(N2ST-1)/MAX_N2ST-1)]. In the example illustrated in Figure 59 the maximum distance to channel was 15 cells. The geometric mean of slope and the spherical mean of aspect are computed for specified intervals of relative profile distance as portrayed by the variable REL_N2ST. For all hillslopes that have a maximum value for N2ST less than 21 cells, a mean is computed for every unique value of REL_N2ST for the hillslope.

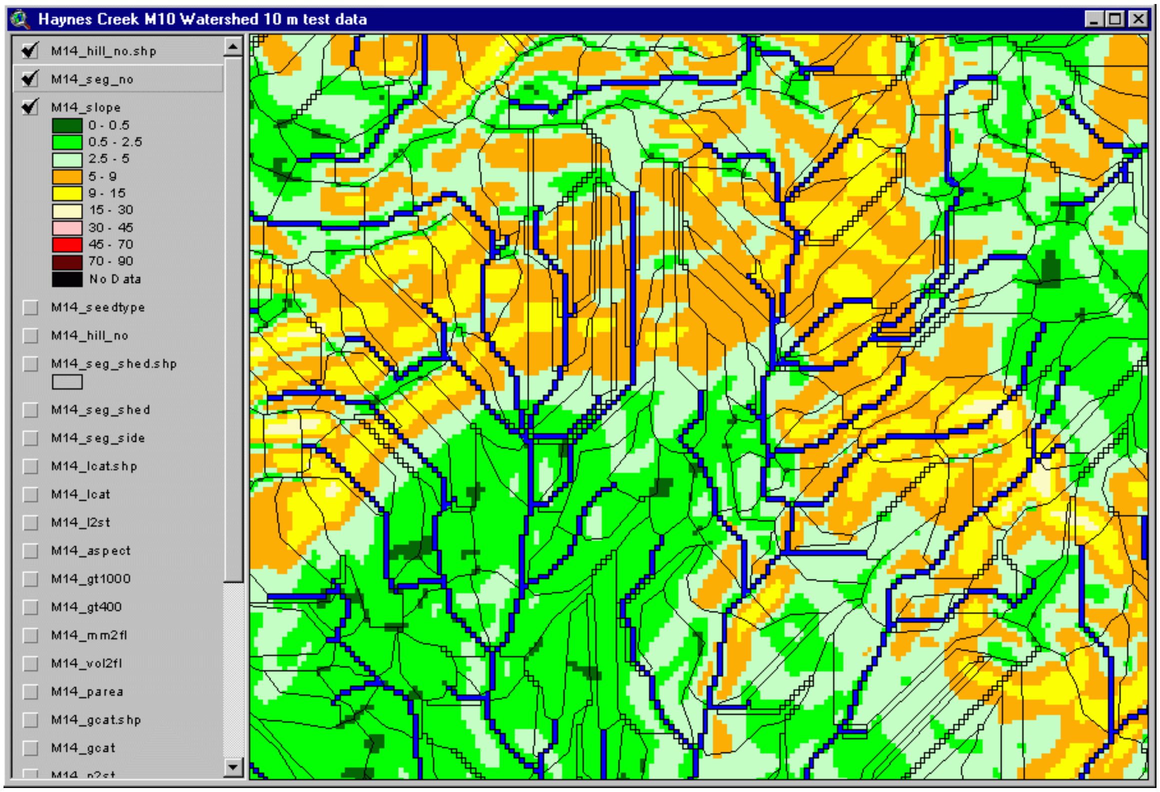


Figure 59. . Illustration of slope gradient (in CSSC slope classes) computed by WeppMapR for an example area

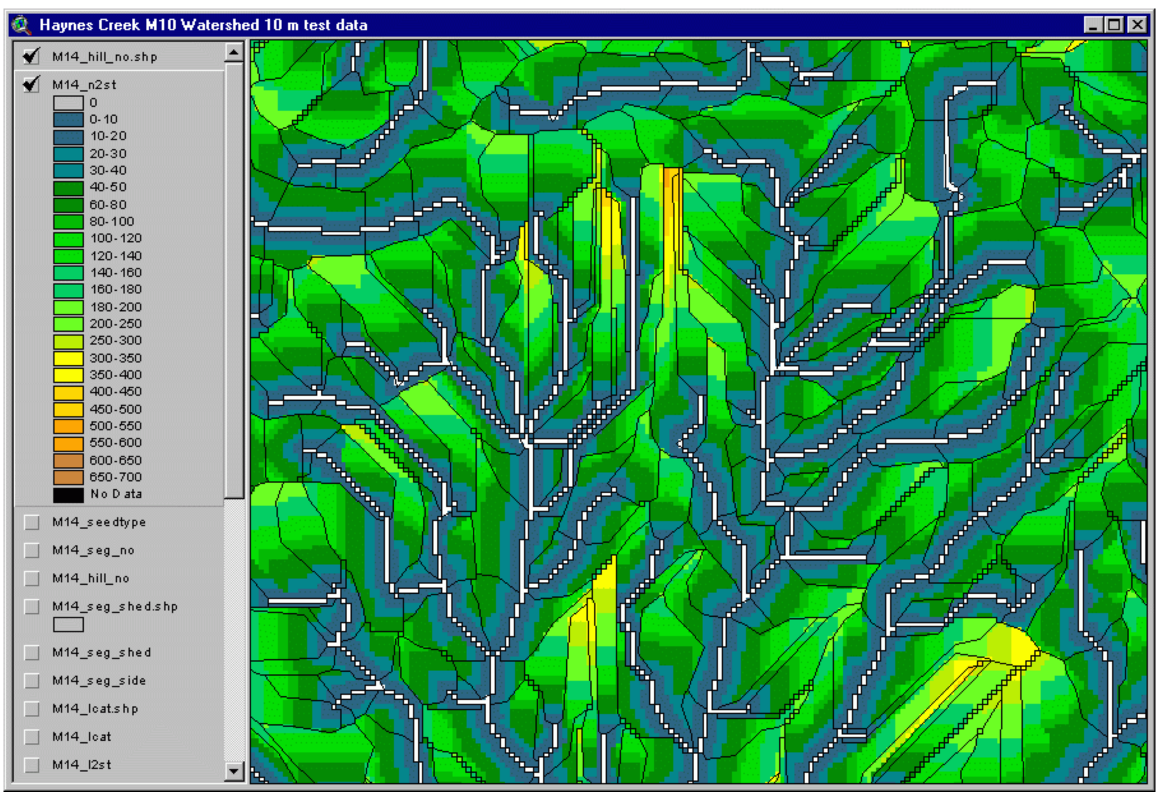


Figure 60. Illustration of distance to channel as measured by the variable N2ST for an example area

For hillslopes with maximum values for N2ST greater than 20 cells, average values are computed for no more than 18 equal intervals of relative profile distance. This limitation is imposed by the WEPP convention that hillslope profiles can be described by no more than 20 points located at specified relative distances along a representative WEPP hillslope profile.

Each WEPP hillslope is therefore described by computing mean values for slope gradient and aspect for all cells that are within a specified range of relative distance along a notional hillslope profile. The notional profile begins at the farthest point from the channel (N2ST = MAX_N2ST) and ends at cells immediately adjacent to the channel (N2ST = 1). As per WEPP conventions, the slope profile is described using pairs of data values. The first value in a pair gives the distance from the start (top) of the hillslope profile expressed as a relative proportion of the maximum total profile length (a decile value). The second value in a pair describes the slope gradient (in m/m) at that point. All profiles must be described by at least two points corresponding to the start (relative distance = 0.0) and the end (relative distance 1.0) of the hillslope. No profile may be described by more than 20 pairs of points (see Figure 62).

Tempstat

Slope	Aspect	N2st	Rel_n2st
0.026	111	1	1.00
0.027	44	1	1.00
0.027	45	1	1.00
0.027	86	2	0.93
0.029	86	2	0.93
0.032	85	2	0.93
0.034	93	2	0.93
0.036	81	2	0.93
0.040	77	2	0.93
0.043	73	2	0.93
0.046	70	2	0.93
0.048	68	2	0.93
0.050	69	2	0.93
0.051	74	2	0.93
0.035	95	3	0.86
0.040	95	3	0.86
0.043	102	3	0.86
0.047	93	3	0.86

Figure 61. Illustration of the temporary working file used to compute slope profile data for notional WEPP hillslopes.

M14hill

Hill_no	Hill_width	Hill_area	Max_len	Wepp_len	Num_points	Aspect	Profile
3340	51.4	11300	220.0	220.0	18	202	0.00,0.021 0.05,0.022 0.10,0.020 0.19,0.019 0.24,0.020 0.29,0.02
3341	130.0	11600	120.0	89.2	12	194	0.00,0.018 0.09,0.017 0.18,0.019 0.27,0.020 0.36,0.022 0.45,0.02
3342	130.0	8000	130.0	61.5	13	252	0.00,0.008 0.08,0.009 0.17,0.010 0.25,0.012 0.33,0.013 0.42,0.01
3343	180.0	2100	40.0	11.7	4	215	0.00,0.023 0.33,0.027 0.67,0.028 1.00,0.014
3344	180.0	11000	160.0	61.1	16	231	0.00,0.029 0.07,0.028 0.13,0.025 0.20,0.019 0.27,0.018 0.33,0.02
3345	110.0	8200	160.0	74.5	16	36	0.00,0.007 0.07,0.010 0.13,0.022 0.20,0.043 0.27,0.059 0.33,0.07
3346	110.0	4500	110.0	40.9	11	125	0.00,0.021 0.10,0.027 0.20,0.032 0.30,0.035 0.40,0.035 0.50,0.03
3347	53.6	11800	220.0	220.0	18	95	0.00,0.004 0.05,0.011 0.10,0.014 0.19,0.010 0.24,0.010 0.29,0.01
3348	170.0	8900	210.0	52.4	18	89	0.00,0.015 0.05,0.019 0.10,0.016 0.20,0.009 0.25,0.011 0.30,0.01
3349	170.0	7800	50.0	45.9	5	142	0.00,0.016 0.25,0.018 0.50,0.019 0.75,0.018 1.00,0.014
3350	170.0	9100	80.0	53.5	8	56	0.00,0.009 0.14,0.012 0.29,0.013 0.43,0.015 0.57,0.016 0.71,0.01

Figure 62. Illustration of the data table prepared to describe the geometry of WEPP hillslopes and their hillslope profiles

The data computed to describe each WEPP hillslope is stored in a DBF table called ID#HILL (Figure 62). In this table, the width of a WEPP hillslope (HILL_WIDTH) is set to equal the length of the WEPP channel that it drains to for left and right hillslopes. The area of a WEPP hillslope (HILL_AREA) is computed as the number of cells in the hillslope times the unit area of a grid cell in the DEM. The maximum length (MAX_LEN) refers to the maximum value for N2ST computed for the hillslope times the unit width of a grid cell in the DEM. The field WEPP_LEN stores a value for the allowable length for the notional WEPP hillslope. If the width of a left or right WEPP hillslope is set equal to the width of the channel segment that it drains to, then the length of the hillslope should be equal the area of the hillslope divided by the width (WEPP_LEN = HILL_AREA/HILL_WIDTH). The field ASPECT stores a value for the spherical mean of aspect for all cells included in the hillslope. The field NUM_POINTS identifies the number of discrete points along the notional profile for which pairs of values for relative distance and slope gradient are given in the field PROFILE. The PROFILE field contains a text string that describes the notional hillslope profile according to the format required by WEPP. Each pair of points gives a value for the relative distance along a profile (from the top) linked to a mean slope gradient for all cells within that profile length increment.

Step 21: Computing channel profiles for each WEPP channel (CHAN_STATS)

This module (CHAN_STATS) performs an operation to compute and describe channel morphology (see Figure 61) that is equivalent to that performed for hillslope morphology by HILL_STATS.

M14chan

Chan_no	Chan_len	Num_points	Mean_slope	Gen_slope	Aspect	Profile
6505	130.0	13	0.0144	0.0153	221	0.00,0.016 0.08,0.019 0.17,0.017 0.25,0.014 0.33,0.014 0.42,0.015 0.50,0.016
6506	180.0	18	0.0137	0.0155	216	0.00,0.016 0.06,0.014 0.12,0.009 0.18,0.004 0.24,0.003 0.29,0.004 0.35,0.007
6507	110.0	11	0.0198	0.0164	100	0.00,0.011 0.10,0.009 0.20,0.010 0.30,0.012 0.40,0.017 0.50,0.021 0.60,0.024
6508	170.0	17	0.0132	0.0131	112	0.00,0.021 0.06,0.019 0.13,0.016 0.19,0.012 0.25,0.009 0.31,0.007 0.38,0.007
6509	170.0	17	0.0109	0.0107	93	0.00,0.012 0.06,0.010 0.13,0.010 0.19,0.011 0.25,0.012 0.31,0.011 0.38,0.009
6510	130.0	13	0.0099	0.0076	134	0.00,0.011 0.08,0.019 0.17,0.017 0.25,0.014 0.33,0.006 0.42,0.004 0.50,0.004
6511	130.0	13	0.0127	0.0116	199	0.00,0.007 0.08,0.007 0.17,0.009 0.25,0.011 0.33,0.014 0.42,0.016 0.50,0.017
6512	170.0	17	0.0144	0.0184	122	0.00,0.004 0.06,0.005 0.13,0.008 0.19,0.012 0.25,0.018 0.31,0.022 0.38,0.022
6513	100.0	10	0.0096	0.0098	177	0.00,0.005 0.11,0.006 0.22,0.005 0.33,0.006 0.44,0.009 0.56,0.013 0.67,0.016
6514	40.0	4	0.0175	0.0119	204	0.00,0.024 0.33,0.021 0.67,0.016 1.00,0.010
6515	40.0	4	0.0453	0.0461	333	0.00,0.049 0.33,0.047 0.67,0.045 1.00,0.040
6516	150.0	15	0.0085	0.0076	159	0.00,0.017 0.07,0.017 0.14,0.017 0.21,0.015 0.29,0.011 0.36,0.008 0.43,0.005

Figure 63. Illustration of the DBF table created to store data on the morphology of WEPP channel elements

The field CHAN_NO contains the unique channel ID number for each channel. Channels are numbered sequentially starting with the channel with the highest elevation end cell and progressing to the channel with the lowest end cell. Channel numbers start at 1 greater than the number of WEPP hillslope elements. Channel length (CHAN_LEN) is computed by counting the number of cells traversed by the flow path that extends from the channel start cell to the channel end cell and multiplying by the width of a grid cell. NUM_POINTS gives the number of points along the channel flow path used to describe the geometry of the channel profile (PROFILE).

For channel segments less than or equal to 20 cells in length, every cell along the channel is included in the profile description. For channel elements that contain more than 20 cells, the channel is divided into up to 19 equal increments and a channel description is extracted for relative slope length and slope gradient at a point for each of up to 19 points along the channel. Mean slope is computed as the arithmetic mean for slope gradient (m/m) for all cells along the channel. General mean (GEN_MEAN) is computed as the absolute vertical drop divided by the linear run between the cell at the start of the channel element and the cell at the end. It provides a useful check of the mean slope gradient and could be used instead of mean gradient to describe the overall top to bottom slope of the channel element. Aspect is the spherical mean of the value for aspect in all cells along the channel element. Profile consists of a series of up to 20 ordered pairs of points that report the relative distance along the channel and the value for slope gradient associated with each of these points (Figure 61).

Step 22: Recomputing upslope area again for all cells (NEW_UPS)

Upslope area count is re-computed again at this point. This is done to ensure that all cells have a correct value for upslope area.

Step 23: Export of data from DBF tables into WEPP ASCII files

Data stored in the ID#Hill and ID#Chan DBF tables is formatted exactly as required for use in the ASCII hill and channel files required by the WEPP program. Each row of data in the ID#Hill or ID#Chan DBF tables can be exported as an ASCII file and read directly into the WinWEPP program. Similarly, the ID#Stru file (Figure 57) is formatted exactly as defined for a WEPP ASCII structure file. Data for any combination of WEPP hillslopes and the channel segments they drain into can be exported from the ID#Stru file into an ASCII structure file for use in the WEPP for Windows program.

Operation of the WeppMapR Program

The WeppMapR interface is illustrated in Figure 64. Running WeppMapR requires the user to make the following decisions and input the following user supplied information.

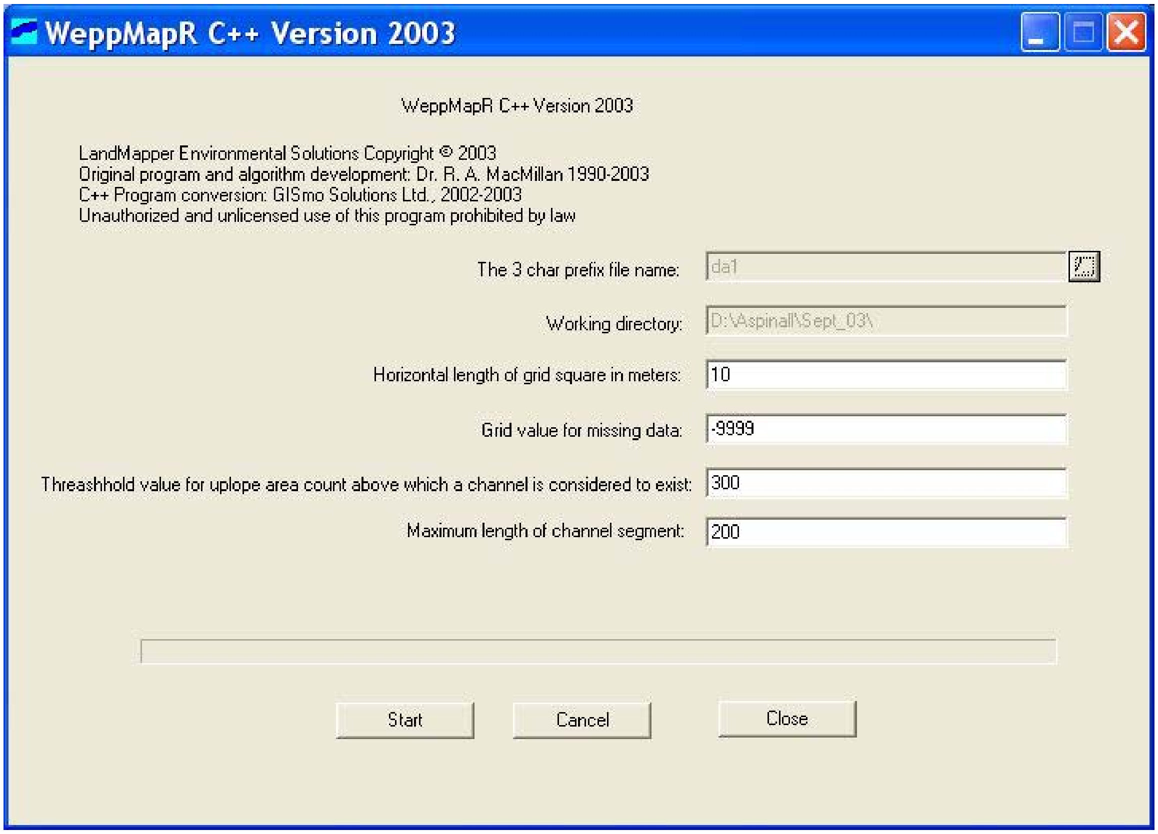


Figure 64. Illustration of the WeppMapR interface and input requirements

Identifying the input data location and working folder

First, the user must navigate to the folder that contains all of the required input data and to select the ID#DEM file that is associated with the site to be processed. This ID#DEM file must have been produced by a successful application of the FlowMapR program. Other files produced by the FlowMapR program (e.g. ID#Pit, ID#Pond) must also have been pre-computed by the FlowMapR program and must be present in the same folder as the selected ID#DEM file. The folder containing the ID#DEM file will automatically become the default working directory into which all output from FormMapR will be placed.

Entering the horizontal dimensions of a grid cell

The user must then enter the horizontal dimensions of a grid cell in meters. The WeppMapR program was never set up to handle measurement units other than meters or cells that were not square. The user-entered value for grid dimension is used in calculations of slope gradient and slope length as well as to compute lengths of channel segments and distance from each cell to cells identified as channels and ridges or pits and peaks.

Entering a value to identify missing data

The user must enter a value that identifies cells that have missing data. The user-entered value for missing data is used to tell the WeppMapR program which cells should not be processed.

Entering a threshold value for recognizing simulated channels

Users must decide upon, and enter, a threshold value for upslope area count for down-slope flow. As previously discussed, this threshold value is used to identify which grid cells are recognized as belonging to channels. It is important to select threshold values that result in production of a channel network that is reasonable and realistic for the area, the terrain and the objectives of the planned WEPP modeling. If the entered threshold value is too low, too many cells will be classed as belonging to channels. This can confuse and distort extraction of channel and hillslope spatial entities performed by the WeppMapR program. In the same vein, if the selected threshold value is too high, too few cells may be classed as channels and the resulting network of channel segments and the hillslopes that flow into them will be inadequate.

A default value of 300 cells of upslope area count has been found to work well for many kinds of terrain and many different DEMs of different grid resolutions and extent. The 300 cell value seems to be particularly suitable when processing relatively high resolution (5-10 m horizontal grid cells) DEM data for field and farm sized areas (800 m to 1600 m on a side). For DEM data sets with a very small number of cells (e.g. less than 10,000) it may frequently be necessary to select lower threshold values (100-200 cells) as there are simply not enough cells in the DEM to produce large upslope area counts in the range of 300 or greater. For very large DEM data sets containing 1 million cells or more, it has often proven better to select threshold values closer to 1000 cells as lower thresholds can result in recognition of channel networks that are overly detailed and confusing.

It is advisable for users to produce maps and visualizations of upslope area count immediately after running FlowMapR and before running FormMapR so that they can be confident they have selected and entered threshold values that will produce appropriate stream and channel networks. Having said this, I have very often selected and entered threshold values "blindly" without reference to maps of different threshold values. This approach seems to work if one has processed a large number of DEMs of similar resolution for similar terrain and if a particular threshold value has consistently proven to produce reasonable divide and channel network results.

Upslope area represents a count of numbers of cells and does not represent absolute upslope area in square meters. The number of cells needs to be multiplied by the area of a single grid cell to convert upslope area count into a measure of absolute upslope area.

Entering a value for maximum length of a channel segment

Users are required to enter a value to identify the maximum length that a channel segment can attain before it is split to produce two channel segments of shorter length.

This value will determine whether long channel segments are split into shorter segments. The advantage of defining shorter segments is that shorter segments tend to be associated with a more uniform set of conditions, both in terms of the shape and orientation of the channel segment and in terms of the orientation and diversity of the hillslopes that contribute flow into the defined channel segment.

If users do not wish to sub-divide long channel segments into shorter segments, they need only select an arbitrarily large value for maximum length of channel segment. A very large value will almost certainly mean that no channel segments exceed the maximum permitted length and so no channel segments will be sub-divided by insertion of a split type seed cell. It is unadvisable to define a maximum length of channel segment that is less than 5 cell units long and it is preferable to allow channel segments to be at least 20 cell units long.

Running the WeppMapR program

Click on Start to run the WeppMapR program. If the progress dialogue message stalls and does not update itself for a long period of time, it may well be that the program has entered an infinite loop. In this case, the program can be cancelled by clicking on the Cancel button. Press Close to close the program after it has successfully run to completion.

WeppMapR Output Files

WeppMapR creates 5 different DBF output files (Table 8). The file named ID#Wepp is the main file that contains one record for each grid cell in the original raster DEM (Figure 65). All other files are summary files that summarize the attributes of hydrological entities (channels and hillslopes) extracted from the raster DEM by the WeppMapR program (Figure 66). The file named ID#Segs is a working file that is used by WeppMapR in computing the topology of hydrological connectivity from hillslopes into channel segments and from upper into lower channel segments. The three other files (ID#Chan, ID#Hill and ID#Struc) are formatted exactly as defined for their exact counterparts in the WEPP model.

Table 7. Listing and description of all DBF output files produced by the WeppMapR program

File	Description of contents
ID#Chan	This file reports data for each unique channel segment extracted by the WeppMapR program. Each channel segment has a unique integer ID number and is described in terms of its length, mean slope, general slope, mean aspect, number of points used to create its representative profile and a series of paired points that define a representative channel profile.
ID#Hill	This file reports data for each unique WEPP hillslope extracted by the WeppMapR program. Each hillslope has a unique integer ID number and is described in terms of its width, area, maximum length, WEPP length, mean aspect, number of points used to create its representative profile and a series of paired points that define a representative hillslope profile.
ID#Segs	This is a working file used by the WeppMapR program to label each unique channel segment in a simulated drainage network and to work through the procedures that order channel segments and compute which channel segments connect to which other down slope channel segments.
ID#Struc	This file stores output data on the most likely ecological classification for every grid cell in a matrix. It does not record or report the likelihood value for every possible fuzzy attribute or every fuzzy class. It is only produced by option b
ID#Wepp	This file stores data for only the most likely landform classification produced by application of a condensed version of the full LandMapR LSM program. It only saves data for the most likely output classification and not for every possible fuzzy attribute value or fuzzy class value. It is only produced by option c.

M14wepp

Seqno	Row	Col	Elev	Ddir	Drec	Upslope	Seedtype	Shed_no	Shed_side	Hill_no	Chan_no	Chan_side	Segment_no
1	1	1	-9999.0000	0	1	1	0	0	0	0	0		0
2	1	2	937.5941	6	3	2	0	5979	3	2340	0		0
3	1	3	937.3654	6	4	12	0	5979	3	2340	0		0
4	1	4	937.3209	6	5	13	0	5979	3	2340	0	3	0
5	1	5	937.2789	6	6	535	0	0	0	0	434		5979
6	1	6	937.2778	6	7	550	0	0	0	0	434		5979
7	1	7	937.2767	6	8	551	0	0	0	0	434		5979
8	1	8	937.2756	6	9	563	0	0	0	0	434		5979
9	1	9	937.2756	6	10	565	0	0	0	0	434		5979
10	1	10	937.1732	6	11	577	0	0	0	0	434		5979
11	1	11	937.0315	6	12	578	0	0	0	0	434		5979
12	1	12	936.8940	6	13	579	7	0	0	0	434		5979
13	1	13	936.8054	6	14	1078	2	0	0	0	434		6058

Figure 65. An example of the main ID#Wepp DBF file produced by WeppMapR

M14chan

Chan_no	Chan_len	Num_points	Mean_slope	Gen_slope	Aspect	Profile
6505	130.0	13	0.0144	0.0153	221	0.00,0.016 0.08,0.019 0.17,0.017 0.25,0.014 0.33,0.014 0.42,0.015 0.50,0.016
6506	180.0	18	0.0137	0.0155	216	0.00,0.016 0.06,0.014 0.12,0.009 0.18,0.004 0.24,0.003 0.29,0.004 0.35,0.007
6507	110.0	11	0.0198	0.0164	100	0.00,0.011 0.10,0.009 0.20,0.010 0.30,0.012 0.40,0.017 0.50,0.021 0.60,0.024
6508	170.0	17	0.0132	0.0131	112	0.00,0.021 0.06,0.019 0.13,0.016 0.19,0.012 0.25,0.009 0.31,0.007 0.38,0.007
6509	170.0	17	0.0109	0.0107	93	0.00,0.012 0.06,0.010 0.13,0.010 0.19,0.011 0.25,0.012 0.31,0.011 0.38,0.009
6510	130.0	13	0.0099	0.0076	134	0.00,0.011 0.08,0.019 0.17,0.017 0.25,0.014 0.33,0.006 0.42,0.004 0.50,0.004
6511	130.0	13	0.0127	0.0116	199	0.00,0.007 0.08,0.007 0.17,0.009 0.25,0.011 0.33,0.014 0.42,0.016 0.50,0.017
6512	170.0	17	0.0144	0.0184	122	0.00,0.004 0.06,0.005 0.13,0.008 0.19,0.012 0.25,0.018 0.31,0.022 0.38,0.022
6513	100.0	10	0.0096	0.0098	177	0.00,0.005 0.11,0.006 0.22,0.005 0.33,0.006 0.44,0.009 0.56,0.013 0.67,0.016
6514	40.0	4	0.0175	0.0119	204	0.00,0.024 0.33,0.021 0.67,0.016 1.00,0.010
6515	40.0	4	0.0453	0.0461	333	0.00,0.049 0.33,0.047 0.67,0.045 1.00,0.040
6516	150.0	15	0.0085	0.0076	159	0.00,0.017 0.07,0.017 0.14,0.017 0.21,0.015 0.29,0.011 0.36,0.008 0.43,0.005

a) Example of a WEPP channel file produced by the WeppMapR program

M14hill

Hill_no	Hill_width	Hill_area	Max_len	Wepp_len	Num_points	Aspect	Profile
3340	51.4	11300	220.0	220.0	18	202	0.00,0.021 0.05,0.022 0.10,0.020 0.19,0.019 0.24,0.020 0.29,0.02
3341	130.0	11600	120.0	89.2	12	194	0.00,0.018 0.09,0.017 0.18,0.019 0.27,0.020 0.36,0.022 0.45,0.02
3342	130.0	8000	130.0	61.5	13	252	0.00,0.008 0.08,0.009 0.17,0.010 0.25,0.012 0.33,0.013 0.42,0.01
3343	180.0	2100	40.0	11.7	4	215	0.00,0.023 0.33,0.027 0.67,0.028 1.00,0.014
3344	180.0	11000	160.0	61.1	16	231	0.00,0.029 0.07,0.028 0.13,0.025 0.20,0.019 0.27,0.018 0.33,0.02
3345	110.0	8200	160.0	74.5	16	36	0.00,0.007 0.07,0.010 0.13,0.022 0.20,0.043 0.27,0.059 0.33,0.07
3346	110.0	4500	110.0	40.9	11	125	0.00,0.021 0.10,0.027 0.20,0.032 0.30,0.035 0.40,0.035 0.50,0.03
3347	53.6	11800	220.0	220.0	18	95	0.00,0.004 0.05,0.011 0.10,0.014 0.19,0.010 0.24,0.010 0.29,0.01
3348	170.0	8900	210.0	52.4	18	89	0.00,0.015 0.05,0.019 0.10,0.016 0.20,0.009 0.25,0.011 0.30,0.01
3349	170.0	7800	50.0	45.9	5	142	0.00,0.016 0.25,0.018 0.50,0.019 0.75,0.018 1.00,0.014
3350	170.0	9100	80.0	53.5	8	56	0.00,0.009 0.14,0.012 0.29,0.013 0.43,0.015 0.57,0.016 0.71,0.01

b) Example of a WEPP hill file produced by the WeppMapR program

M14segs

Initial_id	Start_type	Start_cell	Start_row	Start_col	Start_elev	Start_ddir	End_type	End_cell	End_row	End_col	End_elev	Len_cells	Len_meters	Down_seg
1	5	39802	67	202	982.3011	3	5	39802	67	202	982.3011	1	10	4763
2	5	39802	67	202	982.3011	3	5	39802	67	202	982.3011	1	10	4775
3	8	40403	68	203	982.2990	6	3	30215	51	215	973.0126	20	200	4821
4	5	160	1	160	979.9965	5	5	160	1	160	979.9965	1	10	4765
5	5	160	1	160	979.9965	5	5	160	1	160	979.9965	1	10	4765
6	1	118299	198	99	979.8419	2	4	129099	216	99	976.9833	19	190	4804
7	1	106279	178	79	979.7014	8	4	94883	159	83	974.7014	20	200	4774
8	1	82238	138	38	978.7338	4	4	82220	138	20	965.7224	19	190	5649
9	1	118867	199	67	978.6370	2	7	125466	210	66	975.4341	12	120	4801
10	1	123679	207	79	978.4289	4	3	130265	218	65	969.6098	15	150	4877
11	1	16963	29	163	977.8193	4	4	18750	32	150	970.8517	14	140	4844
12	1	43972	74	172	977.3669	2	3	52973	89	173	969.9122	16	160	4803
13	1	25381	43	181	976.7680	8	7	22384	38	184	974.8232	6	60	4783
14	8	129699	217	99	976.7390	2	4	139901	234	101	969.0423	18	180	4857
15	8	121859	204	59	976.5333	3	3	125464	210	64	975.2784	8	80	4801
16	5	121858	204	58	976.4341	6	5	121858	204	58	976.4341	1	10	4768
17	5	121858	204	58	976.4341	6	5	121858	204	58	976.4341	1	10	4770
18	1	22378	38	178	975.9858	6	3	21784	37	184	974.7148	7	70	4783
19	1	99691	167	91	975.8422	9	4	91895	154	95	970.7034	14	140	4825

c) Example of a WEPP segment file produced by the WeppMapR program

M14stru

Element_no	Ele_type	Left_hill	Right_hill	Top_hill	Left_chan	Right_chan	Top_chan	Left_imp	Right_imp	Top_imp	Comment
4762	2	2	1	0	0	0	0	0	0	0	Element No: 4762 channel
4763	3	0	0	0	0	0	4762	0	0	0	Element No: 4763 impoundment
4764	2	5	4	3	0	0	0	0	0	0	Element No: 4764 channel
4765	3	0	0	0	0	0	4764	0	0	0	Element No: 4765 impoundment
4766	2	8	7	6	0	0	0	0	0	0	Element No: 4766 channel
4767	2	10	9	0	0	0	0	0	0	0	Element No: 4767 channel
4768	3	0	0	0	0	0	4767	0	0	0	Element No: 4768 impoundment
4769	2	13	12	11	0	0	0	0	0	0	Element No: 4769 channel
4770	2	15	14	0	0	0	0	0	0	4768	Element No: 4770 channel
4771	2	18	17	16	0	0	0	0	0	0	Element No: 4771 channel
4772	2	21	20	19	0	0	0	0	0	0	Element No: 4772 channel
4773	2	24	23	22	0	0	0	0	0	0	Element No: 4773 channel
4774	2	26	25	0	0	0	4773	0	0	0	Element No: 4774 channel
4775	2	28	27	0	0	0	0	0	0	4763	Element No: 4775 channel

d) Example of a WEPP structure file produced by the WeppMapR program

Figure 66. Examples of all summary output files produced by the WeppMapR program

Using and visualizing the WeppMapR Output Data

WeppMapR presently exists as an interesting, but non-essential, addition to the LandMapR toolkit of programs. None of the output from WeppMapR is presently used in current procedures for classifying ecological or landform spatial entities.

The ability to extract hillslopes and portions of hillslopes from a DEM using WeppMapR opens up some interesting possibilities for computing integrated land and water spatial entities that possess attributes of both hydrological connectivity and characteristic landform position and surface configuration. It is hoped that the LandMapR toolkit will evolve to enable automated extraction of landform facets that act as integrated land and water spatial entities with hydrological connectivity being explicitly defined for each hillslope patch or facet, in addition to the shape and relative landform position for each facet.

For the present, however, the hillslope and channel spatial entities extracted by WeppMapR are mainly interesting curiosities that exist distinctly apart from the landform or ecological entities computed by the FacetMapR program.

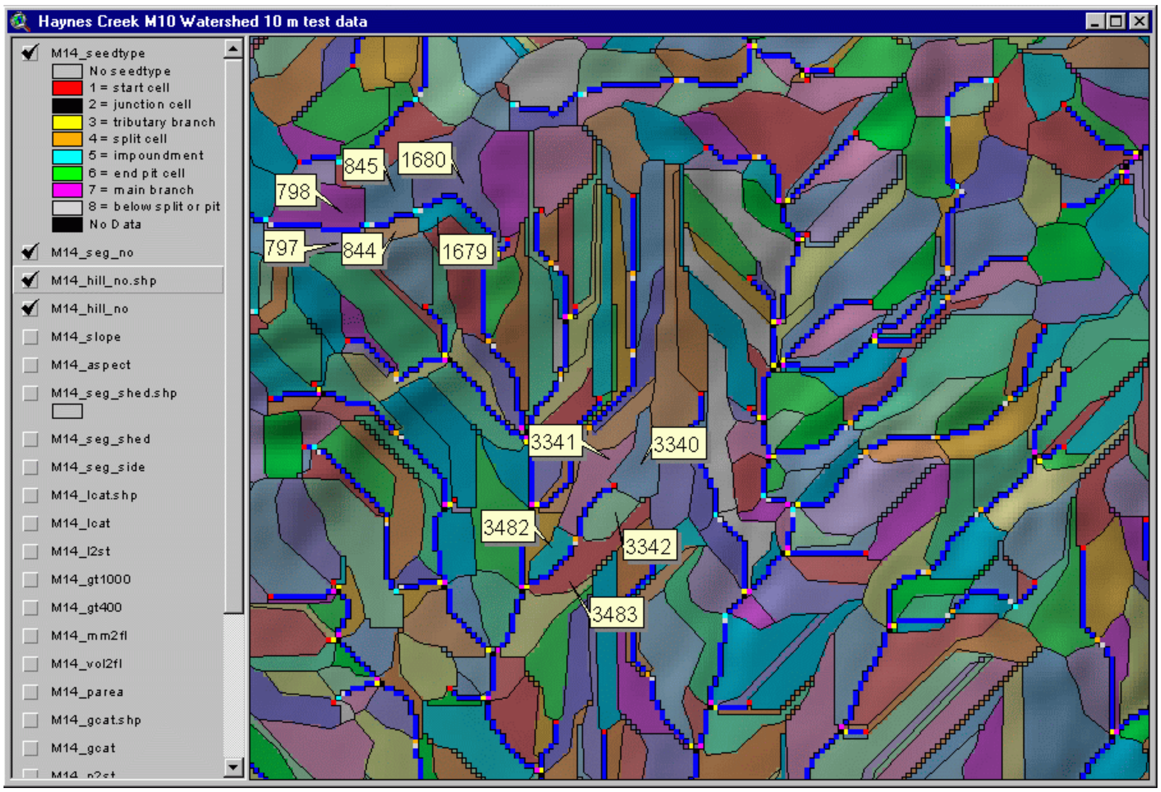


Figure 67. An example of hillslope and channel entities extracted using the WeppMapR program

Users are encouraged to run WeppMapR on their data sets and to create grid maps that depict the location and extent of WEPP hillslopes and channel segments (Figure 67). They are further reminded that each of the spatial entities depicted on these maps exists as a uniquely identified object in a relational table in DBF format. The DBF tables identify the ID number of the channel segment that each WEPP hillslope drains into and the ID numbers of the channel segments that drain into each uniquely numbered channel segment, as well as the channel (or impoundment) that each segment itself flows into. The WEPP hillslopes easily and rapidly identify which sections of which stream channels are likely to be impacted by any activity in any upslope area. Any activity within a defined WEPP hillslope will have its most immediate impact on the section of stream channel that the hillslope drains into and may have additional impacts on any down slope drainage channel segments. This kind of information may prove to be very useful for estimating relationships between activities in upslope areas and their down slope impacts.

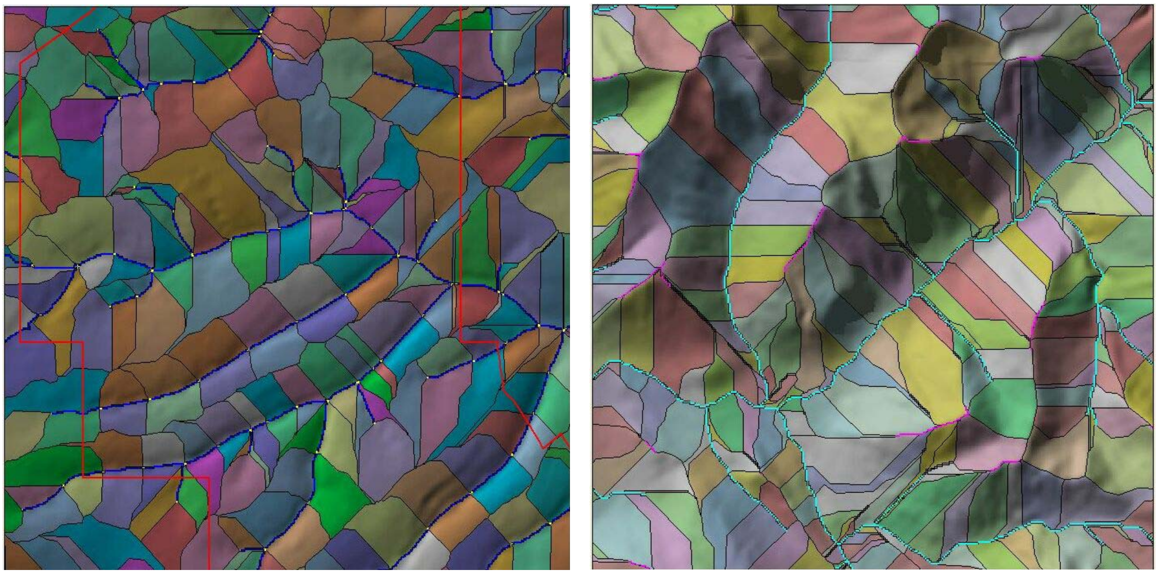


Figure 68. Examples of WEPP hillslopes and channels extracted for two very different types of terrain

Possible future development of the WeppMapR program

The WeppMapR program was initially developed as a one-of custom program to address specific needs of a specific client (Alberta Agriculture, Food and Rural Development). It was developed to take advantage of the flow topology computed by the FlowMapR program and to extend the utility of this topology. The program has proven capable of extracting meaningful and consistent hydrological networks and their associated hillslopes in a wide variety of types and scales of terrain (Figure 68).

The initial functionality of the WeppMapR program could be very readily generalized to compute and store hydrological entities and their topological relations for any number of other hydrological data models. One model of particular interest and relevance is the ArcGIS Hydro data model (Maidment, 2000). It is hoped that the underlying functionality of the WeppMapR program can eventually be expanded to develop fully automated procedures for extracting the full ArcGIS Hydro data model from DEM data.

Other possible applications of the basic WeppMapR capabilities may include extracting the cells from a particular WEPP hillslope and processing them to classify each hillslope area into components or patches that have both a unique set of topographic conditions (e.g. relative landform position, shape (concave/convex), slope gradient and orientation) and an explicitly defined connectivity to other hillslope patches that lie both above them and below them. This kind of classification approach would result in recognition of fully integrated land and water spatial entities that had significance from both geomorphological and hydrological perspectives.

LandMapper Environmental Solutions Inc. strongly believes that an integrated land and water data model will ultimately prove to be more powerful and more useful than separate independent data models for land (landform or soil facets) and water (hydrological flow networks). It is LandMapper's intention to continue to advocate for an integrated land and water spatial data model and to try to develop the LandMapR toolkit to support extraction of integrated land and water data models.

For the moment, however, WeppMapR will likely remain a bit of an orphan program in the LandMapR toolkit, until such time as client needs or external funding create an environment in which improvements or extensions are immediately necessary and desirable.

The GridReadWriteUtility

Purpose

The GridReadWriteUtility program is used to transfer spatial data back and forth between the DBF format files used by all LandMapR toolkit programs and the binary grid format used by ArcView 3.2 to import and export raster data.

Rationale for the GridReadWriteUtility

The GridReadWriteUtility program was created to help address inefficiencies in reformatting and transferring spatial data between the DBF format tables used by all LandMapR toolkit programs and the GIS software used to collate and display both input data sets and output results.

The original LandMapR programs were written to operate within the Visual FoxPro database environment. Users of the FoxPro version of the programs were therefore obliged to have a copy of FoxPro and so had at their disposal the means for defining DBF format tables, importing raster data in ASCII format into defined DBF tables and exporting data from DBF tables into ASCII files that could then be reformatted for import into GIS programs.

The new C++ versions of the LandMapR toolkit no longer run inside Visual FoxPro and therefore no longer absolutely require users to possess a copy of FoxPro. It therefore became desirable to have a utility program that would provide users who did not have a copy of FoxPro with a convenient alternative means of moving input data from their GIS into the LandMapR DBF format and moving output data from the LandMapR DBF tables into their GIS format.

Operation of the GridReadWriteUtility

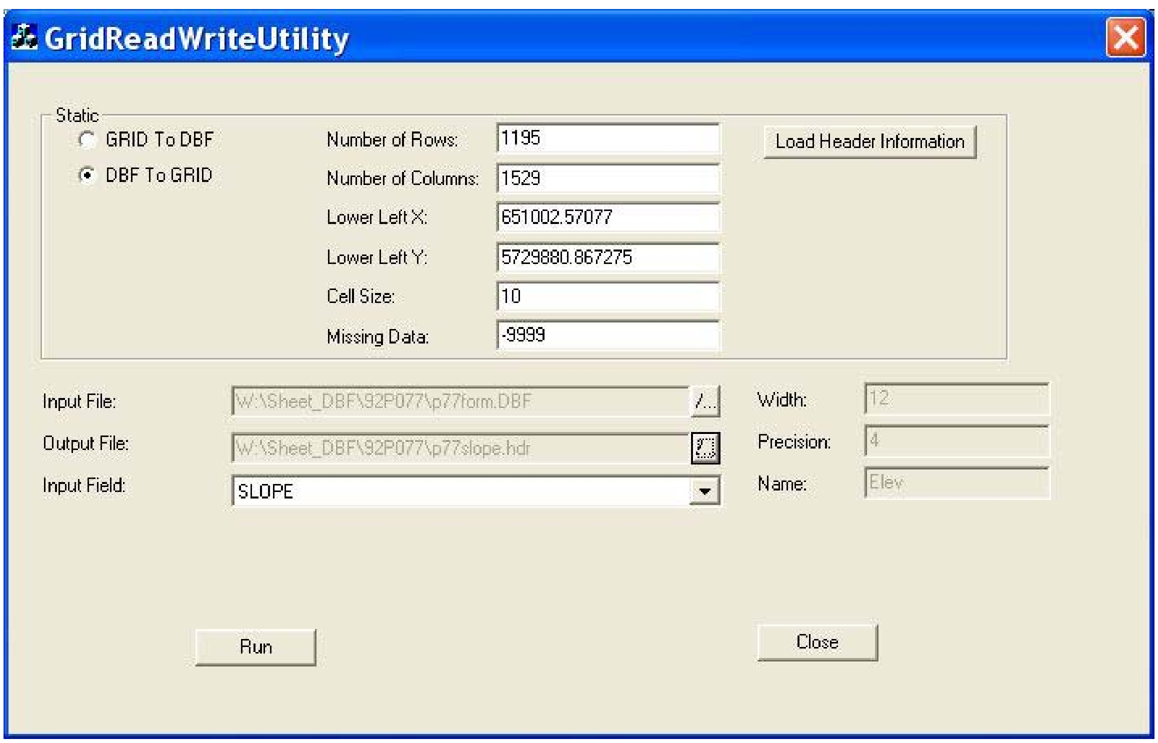


Figure 69. Illustration of the interface for the GridReadWriteUtility

The interface for the GridReadWriteUtility is illustrated in Figure 23.

This utility can be used to convert grid data exported from ArcView 3.2 in binary raster format (FLT & HDR files) into the DBF format used by the LandMapR toolkit. Alternately, it can be used to extract a single column of data from a DBF format file used by the LandMapR toolkit and format these data as a single binary raster in the format used by ArcView 3.2 to import (and export) binary raster data sets (FLT & HDR files).

Exporting from ArcView 3.2 and reformatting into DBF format

Most users will have a raster DEM data set that they will need to reformat for use in the LandMapR toolkit programs. It is assumed that the raster DEM data were created in ArcView 3.2 or at least reside in an ArcView 3.2 view. Those users who use other software packages to prepare their DEM data will have to adjust the procedures described here accordingly. Most commercial DEM software packages offer capabilities to export into the common ArcView binary raster format characterized by a raw binary file of data with an extension of FLT and an ASCII header file with an extension of HDR.

Assuming the user has their DEM data in an ArcView 3.2 project; it is first necessary to export the data (usually DEM data but not always) as an ArcView 3.2 binary export file. Users need to select Export Data Source from the ArcView File menu and then navigate to the location where the original ArcView data is stored in native ArcView GRID format. They select this file, select Export As binary format and then provide a name and folder location for the binary export file to be created. ArcView will export the data to a file with the user assigned name and will give it a file extension of FLT and will at the same time create an ASCII header file with the extension HDR.

Once the data have been exported you will have 2 files, one with a FLT extension and one with a HDR extension. At this point you can open the GridReadWriteUtility and select the Grid to DBF option.

In the navigation bar to the right of the input file box navigate to the location where you have the FLT and HDR ArcView export files and select the HDR file for the file you wish to reformat into a DBF table. In most cases, this will be a file of DEM elevation data that you will want to convert into a DBF file named ID#ELEV as this is the initial file required to run the FlowMapR program.

Next, click on the navigation bar to the right of the output file box and assign a name and folder location to the DBF file that you wish to create. Again, most users will want to create a file with the name ID#Elev that will contain the DEM elevation data they wish to process through the LandMapR toolkit programs.

The GridReadWriteUtility is set up to default to creation of a DBF output file with the name ID#Elev that contains a single field with the name Elev of width 12 and precision 4. Users can change the name to be assigned to the field in the DBF file they are creating and can change its width and precision, if the data they are reformatting are not the default DEM elevation data that the GridReadWriteUtility is expecting. Otherwise, users can accept the default field name, field precision and field width.

Users then click on the Run button and the utility will read in the ArcView FLT file data and then write out the same data as a DBF table with the name, field name, field width and field precision specified by the user. The program has no progress dialogue but will write out the message "Done" once the conversion is complete.

Reformatting from DBF format into ArcView binary raster import format

The GridReadWriteUtility can also extract data from a single field or column in a DBF table and convert it into GIS grid file in ArcView binary raster (FLT) format. Typically, most users will want to convert the final classification produced by application of the FacetMapR program into a grid map for display and review in ArcView Spatial Analyst. However, the utility can be used to convert any data present in any columns of a DBF table into a grid map, as long as the DBF table contains data for the same number and location of grid cells as the original gridded DEM file described by the ArcView header file (HDR) for an area of interest.

It is first necessary to have a header file (extension HDR) that describes the size (number of rows and columns) of the grid that is to be produced and gives the coordinates of the lower left corner in whatever coordinate system is applicable. The easiest way to produce such a file is to export an existing grid file from ArcView using the Export Data Source function (see above). Users who have exported their DEM data to a file named ID#Elev will already have created an appropriate header file (HDR) that describes the size and location of the grid data set that they wish to create. An example of the organization and content of an ArcView header file for a binary raster FLT file named ID#ELEV is provided below.

ID#ELEV.hdr

```
ncols         6872
nrows         7129
xllcorner     471725.46875
yllcorner     5505242.5
cellsize      25
NODATA_value  -9999
byteorder     LSBFIRST
```

The GridReadWriteUtility has been found to be sensitive to the order in which users interact with the interface to select input and output files. The program will fail if entries are selected in an incorrect order. Users are therefore advised to keep to the following sequence when entering their selections into the interface.

1. Select the option DBF to Grid
2. Click on Load Header Information and navigate to the location where there is a header file (HDR) that describes the size and location of the grid file that you wish to create. This header file does not have to have the same name as the file you wish to create and, in fact, should have a different name.
3. Click on the navigation bar to the right of the input file box and navigate to the folder location and file name of the DBF file that contains the data that you want to convert into an ArcView binary grid file. Click on this DBF file name (e.g. ID#LSM or ID#DSS).
4. Click on the down arrow to the right of the Input Field box and scroll through the list of fields in the DBF file you just picked to select the field that contains the data that you wish to convert into a binary FLT file. Click on this field name.
5. Click on the navigation bar to the right of the input file box and identify the folder location and file name for the new binary raster FLT file that you wish to create. Enter a name for this file that is has meaning for you such as ID#DSS_Run1.
6. Click on Run. The utility will read in the data from the selected field in the DBF file and write it out as a binary raster file suitable for import into ArcView 3.2.
7. Click on Close once you receive the notification that the processing is "Done".

One unfortunate characteristic of ArcView binary grid files in FLT format is that they are always assumed by ArcView to contain data expressed in terms of real numbers. Consequently, when data from these files are imported into ArcView via the Import Data Source function on the File menu, they are automatically created as files of real numbers.

If the data you are importing consist of integer numbers (as is the case with all classification results from FacetMapR) then it will be necessary to convert the imported data from real to integer format inside ArcView. This can be done easily enough by using the INT function in the ArcView Map Calculator (look under the Arithmetic option in the calculator). This necessitates extra work that would be better off avoided, but for now there is no option but to follow this extra step to convert real data into integer data fro any imported data sets that should be in integer format. No conversion is required, of course, for data that are meant to be expressed as real numbers.

Possible Future Developments for the LandMapR Toolkit

Developments and Applications to Date

The C++ version of the LandMapR toolkit represents a considerable evolution and improvement from the original Visual FoxPro version. The C++ programs are many times faster than the Visual FoxPro versions and can process data sets of much greater size.

Between the original FoxPro versions and the new C++ versions, the LandMapR toolkit has now been applied successfully to several hundreds of DEM data sets ranging in size from one or two hundred rows by one or two hundred columns up to a maximum of about 4,000 rows by 4,000 columns. The initial rule bases developed to classify a standard set of 15 landform classes (LM0Arule/LM0Crule and LM3Arule/LM3Crule) have produced very consistent and useful classification results for almost every DEM to which they have been applied. The more recent BC-PEM DSS version of FacetMapR has been used to create meaningful landform and ecological maps for areas of up to 2 million ha consisting of as many as 50 million grid cells.

Major Areas Requiring Improvement

Despite the overall satisfactory performance of the LandMapR toolkit in applications to date, there are a number of areas in which the toolkit could benefit from improvements. The major areas that would benefit from improvements have been identified as follows:

1. The current procedures for creating rule files to define new classes of spatial entities are inefficient and time consuming and need to be improved.
    a. Improved procedures for creating new rule files could reduce the time required to create effective, accurate rules from days or weeks down to hours.
    b. Identifying and implementing procedures for creating new rule files more efficiently is the highest priority area for improvement at the present time.
2. All components of the LandMapR toolkit are constrained in terms of the maximum size of data set that they can process in a timely and cost efficient manner and need to be improved to handle larger data sets more efficiently.
    a. This is especially true for the FlowMapR module whose output is required by all other modules.
    b. New flow algorithms are needed that can extract and process sub-watershed tiles from a master DEM and that use improved algorithms to enable processing of massive raster data files with 10's to 100's of millions of cells.
    c. The LandMapR toolkit is increasingly being applied to larger and larger areas described by ever larger DEM data sets. The next version of the LandMapR toolkit will need to be able to process raster data sets of at least 10,000-20,000 rows by 10,000-20,000 columns successfully and in a reasonable amount of time (hours not days).
3. The current LandMapR modules are not well linked to an easy to use GIS environment to facilitate the collation and preparation of input data or the rapid display of results and intermediate output products.
    a. It would be highly desirable to link the LandMapR programs more directly and seamlessly to a GIS shell for rapid and easy collation of input data and display of output data.
    b. An ideal solution would be to use an existing public domain GIS application programming interface (API) such as that provided by a GNU license GIS such as SAGA to provide a platform that could host all of the LandMapR toolkit functionality.

4. Data sets characterized by a preponderance of choropleth input maps of nominal or ordinal classed data associated with choropleth maps of the map classes that users wish to predict are not handled very well by the current LandMapR classification routines.
    a. The fuzzy classification procedures implemented by the current FacetMapR program are more suited to analyzing data sets of continuous variables to develop and apply fuzzy likelihood values. They are not well suited to analyzing nominal, classed data sets to construct predictive rules.
    b. It is proposed that a new set of routines be developed that use analysis of evidence procedures to compute Bayesian Maximum Estimated Likelihood values for each class to be predicted based on spatial co-occurrence of desired output classes relative to classed input maps of nominal or ordinal data.
    c. These new procedures would be specifically targeted to addressing situations in which users had small scale maps of soils or ecological entities supplemented by larger scale maps for scattered training areas and wished to develop rules using the large scale maps that could be extrapolated to areas where these maps did not exist.
5. The current version of FormMapR does not compute many potentially useful terrain derivatives for which algorithms have more recently been described and published.
    a. It is hoped that the FormMapR program can be extended to compute a larger number of these potentially useful terrain derivatives and indices including solar radiation inputs, exposure indices, a full suite of curvature calculations, measures of fractal dimension and measures of similarity distance or range.

Improved procedures for creating new fuzzy rules

The initial development of the LandMapR toolkit was aimed at providing a platform to develop, apply and evaluate a single set of rules for classifying any landforms anywhere in the world into a single, standard set of 15 landform element classes. This initial application did not require efficient procedures for creating and reviewing classification rules interactively, as only one set of rules was intended to be created.

Use of the LandMapR toolkit was later expanded to include preparing multiple sets of custom rule tables to capture and apply the logic embodied in the BC system of Biogeoclimatic Ecosystem Classification (BEC). This application did require creation of multiple rule files for each BEC sub-zone and even for sub-divisions of BEC sub-zones. Procedures were developed to edit rule tables using Microsoft Excel and this helped make revising rules somewhat more efficient. However, creation of all rule files was done completely by trial and error and no tools were developed to help inform or guide the creation of new rule tables.

Two quite different options have been identified that could help to make the creation of new rule sets faster and more correct.

The first involves creating tools that will support interactive adjustment of fuzzy rules accompanied by immediate visual display of the results. Sliders could be used to provide users with the ability to adjust threshold values and weights for the input variables used to defined fuzzy classes and to receive immediate visual feedback on the effects these changes have on a resulting classification. This kind of interactive gaming approach has real appeal as it expands the ability of a knowledgeable local expert to rapidly assess the effects of different rules with different weights and different thresholds. This would empower the ecological and soil experts and provide them with tools that would let them fully control which mapping concepts were captured and how. This would be the ideal approach for building heuristic rule bases for the BC-PEM DSS application. Initial pilot projects have clearly shown that capturing and applying heuristic beliefs is actually more effective and accurate than using a limited amount of field observation data to train "field based" rules. The errors resulting from incomplete or inaccurate field sampling compounded with field classification error (measurement error) appear to result in rule bases that are less accurate and effective than ones based purely on concepts and beliefs.

The second possible approach that might make the creation of new rule bases more efficient is one that would provide automated tools for analyzing spatial co occurrence between sites or areas identified as belonging to a class that one wished to predict and the full suite of potential input layers. This automated analysis would identify central values and dispersion patterns for all continuous input variables for each defined class that one wished to predict. It might conceivable analyze these patterns to automatically create and write out fuzzy rule files in the format required for use by the present FacetMapR program. This approach could provide a much faster and more systematic way of arriving at an initial set of classification rules for any area of interest that can then be applied, evaluated and refined using fewer iterations.

The fact that this approach would require spatially located training areas that had been classified either through field observation or visual interpretation of available data sets and imagery is both its attraction and its drawback. On the positive side, people tend to believe a classification more if it has been developed through reference to actual field observations and classifications. On the negative side, experience in the Cariboo PEM pilot has tended to support the conclusion that use of a limited amount of field data to train classification rules or guide in manual mapping may actually lead to achieving lower levels of accuracy than can be achieved by simply applying conceptual beliefs without the benefit of any field data. This is initially counterintuitive but the more one considers it, the more credible the conclusion becomes. Given the large number of different kinds of error that can occur in locating, collecting and classifying field observations, it is not surprising that use of a limited number of field classified sites may actually lead to a reduction in the accuracy of the resulting predicted classes. Since most sites are actually intergrades between two or more defined classes, it is not unexpected that many field sites classified into a particular class may actually have many attributes of an adjacent class and may even be judged to more correctly belong to another class by a different interpreter. Thus, the classified sites used to help build rules to recognize a particular class already contain a great deal of ambiguity and classification error. One ends up using suspect or invalid classifications for many sites to develop rules that will recognize other similar sites as belonging to this class. Then there is the whole issue of spatial concordance between the reported locations of the field classified sites and the various spatial layers of input data. The likelihood of spatial discrepancies is so high that it is very likely that the location of sites used to train rules are displaced relative to the input data sets used to develop the rules. This leads, inevitably to the creation of erroneous rules that classify sites into incorrect classes.

Of the two main approaches for improving the speed and accuracy of procedures for creating new rules, the first (interactive adjustment and immediate visual feedback) is the more difficult to implement at this time. This approach would involve embedding the logic applied by the FacetMapR program into a graphical GIS interface in such a way that users could interactively adjust weights and threshold values for all potential input layers and immediately see the results of these changes displayed as 2D maps or 3D perspective views. This would allow for very rapid development of rules that produced output classes consistent with an expert's expectations and experience. However, production of such an environment is not trivial and is beyond the programming capabilities possessed in-house at LandMapper. This approach will be pursued if an affordable option for implementing the idea can be identified and acquired.

The second approach of analyzing the distribution of input class values within areas or sites identified as belonging to classes that one wishes to predict is more easily accomplished using capabilities that are already in place in-house at LandMapper. A program to analyze input data layers relative to mapped instances of known classification could be produced quite rapidly. The limitations restricting the development of such a program are the current unavailability of mapped areas of known classification in the areas where LandMapper is currently working and the uncertainty about whether use of a limited amount of field classified training data will actually lead to improved classification rules and results.

Users of the LandMapR toolkit are encouraged to contribute their opinions and suggestions regarding how the procedures for creating new fuzzy classification rules could be improved and streamlined. It is clear that they could benefit from improvement, but not so clear how this should be done.

Improved ability to process very large data sets

The second major issue encountered in using the current LandMapR programs is related to limitations in use of the programs for processing very large data sets. Increasingly, the LandMapR programs are being used to process data for very large areas covered by very large DEM data sets. At present, the programs slow down to undesirable rates of progress when applied to data sets that are much larger than about 2,000 rows by 2,000 columns and the FlowMapR program runs out of memory and fails on data sets much larger than about 4,000 rows by 4,000 columns on Intel computers with 1 GB of RAM.

Increasingly, the LandMapR programs have needed to be applied to areas represented by DEMs of up to 10,000 rows by 10,000 columns (100 million data points). In order to process data for such large areas, the full DEM data sets have had to be subdivided into a number of smaller tiles not much larger than about 2,000 rows by 2,000 columns each. Utility programs have been created to automatically extract tiles from a master DEM such that each tile contains a core area that abuts exactly against all adjacent tiles but that also contains a buffer of several km of data around every core area so as to minimize errors arising from edge effects. These rectangular tiles extracted from a master DEM data set do permit the LandMapR programs to be applied to a set of tiles for an entire area of interest which are then stitched back together to produce a single seamless mosaic. However, such tiles represent only a partial solution to the issue of processing very large areas. Other options exist and should be investigated.

The first option for improving the ability of the LandMapR programs to process data for very large areas is simply to produce an improved, automated procedure for tiling large areas into smaller subset tiles. The optimum approach would be to extract tiles that corresponded to entire independent watersheds, or failing that, into tiles that represented discrete sub-watershed areas whose relationships to upslope and down slope watershed components could be established and recorded. The concept here would be to sub-sample a large DEM to a reduced grid resolution. This would entail extracting 1 in 2 cells or 1 in 4 cells from a master DEM file, whatever is required to produce a reduced grid not much larger than about 2,000 rows by 2,000 columns. This reduced grid could be processed by FlowMapR to identify the location and extent of major watershed areas and, if necessary sub-watershed catchments. The procedures would automatically record the row and column coordinates of the bounding rectangle that enclosed each watershed or sub-watershed area. The procedures would then extract from the full resolution DEM all data for the area of a bounding rectangle that enclosed a particular watershed or catchment. Data for catchments could be extracted and processed sequentially in such a manner that information on the outflow from upper catchments could be added to the relevant location in the next down slope catchment tile to be processed. Each watershed or sub-catchment would be processed independently, but in sequence from higher to lower catchments so that any information about higher catchments was available to be used to characterize the flow regimes of lower catchments. The tiling procedures would automatically place result data for each processed catchment tile back into the master grid, but only for those cells that occurred within the defined catchment boundaries. Thus, each watershed or sub-catchment would represent a semi independent tile and each catchment tile would be extracted and processed and the results replaced back into the master DEM tile automatically by the tiling procedures. This approach would permit very large areas to be broken down into a set of smaller, hydrologically independent, areas that could be processed more rapidly and efficiently using the existing LandMapR programs, especially the FlowMapR program.

The second option is to completely re-engineer all programs in the LandMapR toolkit, but especially FlowMapR so that the algorithms they use are more efficient and less demanding of memory and processing time. The initial Visual FoxPro programs were quite slow, because they involved a lot of read/write activities requiring disk access. However, they had to be able to work with very limited amounts of computer memory, having been originally designed to work on computers that had less than 1 MB of RAM. These programs first sorted an input DEM by elevation from lowest to highest elevation and then processed the DEM from lowest to highest cell (or for some procedures from highest to lowest cell) to compute flow directions and to remove pits.

Several recent projects have identified a need for more efficient algorithms for processing DEM data to compute flow topology and specifically to remove pits to compute fully integrated flow regimes. Perhaps the most comprehensive and encouraging of these is the TerraFlow project at Duke University (http://www.cs.duke.edu/geo*/terraflow/) (Toma et al., 2001). This project has demonstrated that more efficient algorithms for computing flow topology and removing pits are absolutely necessary in order to process very large DEM data sets efficiently, and indeed in order to be able to process them at all. The TerraFlow approach involves sorting an input DEM by elevation from lowest to highest elevation and then processing the DEM from lowest to highest cell. The approach only requires holding in memory information about the cells that occur up to the current elevation range along with very small amounts of data about specific critical points encountered in previously processed cells (e.g. pour points, exit points for lower watersheds and depressions). The algorithm effectively identifies and removes all depressions in one pass through the DEM going from bottom to top. It is therefore much more efficient than all currently used algorithms and is able to process very large DEM data sets in acceptable amounts of time. Also, since it does not have to hold an entire matrix of elevation data in memory at one time, it is not limited in the size of DEM data set that it can process. This is a very important point as one increasingly encounters requirements to process DEMs of 100's of millions of cells.

So one possibility is to return to the original FlowMapR approach used in the original FoxPro programs and to sort and process very large DEM data sets from bottom to top. The original FoxPro FlowMapR algorithms could be adapted to take advantage of some of the ideas and procedures suggested by the TerraFlow project researchers. It should be possible to re-engineer the FlowMapR algorithms so that they can handle DEM data sets of any size in a timely and efficient manner. The new approach would require only a single pass through the DEM data set and would not require an entire matrix to ever be held in memory at any one time. It would likely require giving up an ability to compute some of the data currently computed by the existing FlowMapR pit removing algorithm, but most of these data have not proven to be critical for application of the LandMapR classification procedures to date anyway.

An ideal solution might be to combine the procedures for tiling a large area into watershed or sub-catchment tiles with the faster and less memory constrained algorithms proposed by the TerraFlow project. This would permit a faster algorithm to be applied to a series of independent to semi-independent tiles that were much smaller than the full DEM and that therefore processed much faster as a series of smaller tiles than as a single large tile.

The FormMapR and FacetMapR components of the LandMapR toolkit were also less demanding of computer memory in their original FoxPro versions. The FoxPro FormMapR program never read in more than three rows of data from a DEM in implementing any of its algorithms. The FoxPro FacetMapR component only read in data for a single cell at a time and applied all required calculations to each cell and wrote out the results before moving on to the next cell. These original FoxPro processing approaches could be reintroduced for use by the C++ versions of FormMapR and FacetMapR. They would not necessarily speed up the C++ versions of the programs but they would remove memory restrictions resulting from the present C++ practice of reading in complete matrices containing all required input data for all cells in a grid in order to perform computations that do not require access to all cells in a DEM at any one time.

Improved GIS data management and display capabilities

All users of the current LandMapR toolkit will undoubtedly agree that the present programs are unwieldy in terms of their inability to easily exchange input data and output results with a GIS display environment.

Initial development of the LandMapR toolkit focused on developing algorithms and procedures for classifying geomorphic, ecological and hydrological spatial entities using mainly DEM data. No effort was placed on developing a graphical GIS interface for preparing and collating input data sets and displaying output results. Expanding applications of the LandMapR toolkit have led to the realization that it would be highly desirable to improve the integration of the current LandMapR toolkit programs with a graphical GIS interface. One option would be to integrate the current LandMapR programs more closely with ArcView 3,2 or ArcGIS 8.0. Another would be to embed the functionality of the current LandMapR toolkit inside a public domain GNU license GIS shell such as is offered by the SAGA GIS application programming interface (API). Users may wish to download and experiment with SAGA from (http://134.76.76.30/saga/html/index.php).

Both of these options represent a considerable amount of work and expense. Neither is likely to happen in the existing circumstances under which the LandMapR toolkit is being used. The present situation of acceptable when the programs are used by LandMapper Environmental Solutions Inc. to conduct commercial ecological mapping. It is also acceptable for the very few academic and research users. These users tend to need to apply the programs only once or twice and at a very limited number of sites. The benefits to them of having access to the programs inside a GIS shell are not sufficient to justify the costs. Should the LandMapR toolkit ever be released to others for widespread commercial application, then it will likely be necessary to embed the functionality within a GS interface.

New procedures for creating Bayesian Maximum Likelihood rules

The current commercial applications of the LandMapR toolkit for in-house use are adequately served by the existing FacetMapR procedures for creating and applying fuzzy logic rule bases.

These fuzzy logic procedures are not as well suited to the particular circumstances of some of the more recent academic and research users of the LandMapR toolkit. These more recent users find themselves in situations in which they have a large amount of potentially useful input data that exists as nominal or ordinal data extracted from classed (choropleth) maps of geology, soils, vegetation or similar secondary data sources. At the same time, these users also tend to have a substantial amount of spatially referenced "training data" often in the form of large scale maps of the classes they are interested in predicting that occupy scattered portions of the areas for which they wish to produce predictive maps.

Situations such as those described above are not well suited to addressing using the fuzzy logic procedures currently implemented in FacetMapR. The current FacetMapR procedures are best suited to accessing and interpreting continuous variables reported in terms of interval or ratio data values. For example, slope gradient in percent can be easily converted into a fuzzy attribute of likelihood of being steep using the current FacetMapR functionality. A choropleth map of slope classes will need to be manually interpreted before it can be used in the current FacetMapR procedures. A soil map of nominal classes of named soils is very difficult to use in the current FacetMapR procedures unless each soil entity is interpreted to produce ordinal or ratio numbers that reflect some defined attribute of the soil such as texture (coarseness scale of 0 to 100) or soil depth (depth range of 0-400 cm).

Situations in which users have a preponderance of classed input maps characterized by nominal or ordinal data values are perhaps better analyzed using analysis of evidence procedures based on extracting Bayesian Maximum Likelihood Estimates. LandMapper Environmental Solutions has developed and applied such analysis of evidence procedures for other projects that did not use the LandMapR toolkit (MacMillan and Marciak, 1999, 2000a,b, 2003). These procedures analyzed the spatial co-occurrence of any number of classed input maps against spatially distributed examples of nominal output classes for which a predicted map was desired. This approach does not require the input maps to be of continuous variables but rather prefers to treat all input data layers as if they were nominal classed data with no requirement that the classes possess any kind of structure or order. The procedures simply use the relative frequency of occurrence of the class to be predicted within each class of a given available input map to estimate the likelihood of occurrence of the output class given a particular class of a particular input map. The procedures can be expanded to compute the relative importance of each of ***j*** input layers in predicting a given output class so that a prediction for an output class can be made using a weighted average of the individual likelihoods of the class occurring relative to each of ***i*** classes on each of ***j*** input maps. Based on previous experience in computing Bayesian Maximum Likelihood values for other applications, LandMapper believes that it would be feasible to create a new set of procedures (BayesMapR) for extracting and applying Bayesian Maximum Likelihood estimates to create predictive maps that better utilized existing choropleth maps of nominal and ordinal data.

The proposed new procedures for implementing procedures for extracting and applying Bayesian Maximum Likelihood beliefs are a relatively low priority for LandMapper at the present time. Still, they are of interest because they could be useful and applicable to many current situations in which soil survey agencies have existing smaller scale maps and wish to "densify" these maps by using finer resolution DEM data and available secondary source choropleth maps to predict the most likely soil classes at some larger scale and finer resolution. This is usually done through documenting patterns extracted by analysis of finer resolution soil maps that exist for scattered "training areas" within a larger region of interest relative to all available input data sets.

LandMapper would be very interested to work with existing or potential users of the LandMapR toolkit to extend the current functionality of the FacetMapR procedures so that they were more suited to analyzing and using nominal and ordinal data from classed choropleth input maps.

Calculation of additional terrain derivatives and indices

In the period since the original LandMapR toolkit was first developed, LandMapper has become aware of a large number of additional terrain derivatives and other terrain indices that have considerable potential to prove useful as inputs to automated classification of landforms and soil or ecological entities. LandMapper would like to expand the current version of FormMapR to include calculation of as many of these additional terrain indices as possible.

Some terrain derivatives that are of interest that the current FormMapR program does not compute include measures of incoming solar radiation (averaged by hour, day, month, season and year), a more comprehensive set of calculations of terrain curvature (after Shary et al., 2002), a number of proposed methods of evaluating relative landform position, a proposed exposure index that provides an indication of the degree to which a cell is in a valley or on a ridge and several proposed measures of the relative range about a cell within which the terrain conditions are similar or the distance out to which variation in the terrain continues to increase (a kind of sill distance from geostatistics).

The current FormMapR program would benefit considerably from an update that would implement more recently published algorithms for many new derivatives of relevance for landform classification. These new derivatives are not required in order to implement the current commercial work available to LandMapR but would be very desirable in order to maintain the relevance and currency of the LandMapR toolkit.

Likelihood and Timing of Proposed Improvements

At the present time, all of the improvements and additions to the LandMapR toolkit proposed above are simply speculative. There are no immediate plans to implement these proposed improvements. LandMapper Environmental Solutions Inc. is currently fully committed to contracts for operational mapping using the existing suite of LandMapR programs. There is little time and no money or incentives to undertake these improvements at the present time.

The purpose of this section was simply to alert users of the LandMapR toolkit to the fact that LandMapper realizes that there are several areas where the current programs could significantly benefit from upgrades or changes. LandMapper has identified several areas in which changes were judged to be both desirable and necessary. Other users are encouraged to identify different areas where they feel improvements can be made. Ultimately no further improvements will be made until internal requirements of LandMapper's work and clients necessitate changes or new clients appear with specific requests for changes and the funding required to implement them.

REFERENCES

Burrough, P. A. 1989. Fuzzy mathematical methods for soil survey and land evaluation, J. Soil Sci. 40:477-492

Burrough, P. A., R. A. MacMillan and W. Van Deursen. 1992. Fuzzy classification methods for determining land suitability from soil profile observations and topography, Journal of Soil Science. 43:193-210.

ESRI, 1996. ArcView Spatial Analyst. Using the ArcView Spatial Analyst: advanced spatial analysis using raster and vector data. Environmental Systems Research Institute, Inc., Redlands, CA. 148 pp.

Eyton, J. R. 1991. Rate-of-change maps. Cartography and Geographic Information Systems. 18: 87-103

Flanagan DC., CS. Renschler and TA. Cochrane. 2000. Application of the WEPP model with digital topographic information. (in) Problems, Prospects and Research Needs. Proceedings of the 4th International Conference on Integrating GIS & Environmental Modeling (GIS/EM4); 2000 Sep 2-8; Banff, Alberta, Canada. URL: <http://www.colorado.edu/research/cires/banff/upload/330/index.htm>. Accessed 2001, Feb 1.

MacMillan, R. A., W. W. Pettapiece, L. D. Watson and T. W. Goddard. 1998. A Landform segmentation model for precision farming. ***In:*** Proceedings of the 4th International conference on Precision Agriculture, Radisson Hotel and Conference Center, Minneapolis, Minnesota. July 19-22, 1998.

MacMillan, R. A., W. W. Pettapiece, S. C. Nolan and T. W. Goddard. 2000. A generic procedure for automatically segmenting landforms into landform elements using DEMs, heuristic rules and fuzzy logic. Fuzzy Sets and Systems 113 (1): 81-109.

MacMillan, R. A. and W. W. Pettapiece. 1997a. Soil landscape models: Automated landform characterization and generation of soil-landscape models. Technical Bulletin No. 1997-1E. Research Branch, Agriculture and Agri-Food Canada, Lethbridge, AB. 75 pp.

MacMillan, R. A. and W. W. Pettapiece. 1997b. Landform segmentation procedures manual: Step by step instructions for processing DEM data to define landform segments. Research Branch, Agriculture and Agri-Food Canada, Lethbridge, AB. 65 pp.

MacMillan, R. A. and L. C. Marciak. 1999. Application of a hybrid methodology for estimating potential salinity hazard (PSH) to a large and diverse test region in Alberta, Canada (NTS Sheet 82P). prepared for Alberta Agriculture, Food and Rural Development, Environmentally Sustainable Agriculture Program (AESA) Soil Quality Program (SQP) component. 35 pp.

MacMillan RA. and LC.Marciak. 2000a. Modeling potential salinity hazard (PSH): predicting potential salinity hazard for a large and diverse test region in Alberta, Canada (NTS Sheet 82P). 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4): Problems, Prospects and Research Needs, Banff, Alberta, Canada, September 2-8, 2000. Available on-line at: <http://www.colorado.edu/research /cires/banff/upload/1/> Accessed September 15, 2000.

MacMillan, R. A. and L. C. Marciak, (2003). Estimating potential salinity hazard (PSH) using a hybrid of evidential reasoning and multi-criteria evaluation. Photogrammetric Engineering and Remote Sensing. (accepted for publication).

Martz, L. W. and E. DeJong. 1987. Using Cesium-137 to assess the variability of net soil erosion and its association with topography in a Canadian prairie landscape. Catena. 14: 439-451

Martz, L. W. and E. DeJong. 1988. CATCH: a FORTRAN program for measuring catchment area from digital elevation models. Computers and Geosciences. 14: 627-640.

Morris, D. G., and R. G. Heerdegen, 1988, Automatically derived catchment boundary and channel networks and their hydrological applications, *Geomorphology,* 1:131-141.

Pennock, D. J., B. J. Zebarth and E. DeJong. 1987. Landform classification and soil distribution in hummocky terrain, Saskatchewan, Canada. Geoderma. 40: 297-315.

Pennock, D. J., D.W. Anderson and E. DeJong. 1994. Landscape changes in indicators of soil quality due to cultivation in Saskatchewan, Canada. Geoderma 64:1-19.

Quinn, P., K. Beven, P. Chevallier and O. Planchon. 1991. The prediction of hillslope flow paths for distributed hydrological modelling using digital terrain models. Hydrological Processes. 5: 59-79

Quinn, P., K. J. Beven, and R. Lamb. 1995. The ln(a/tan β) index: How to calculate it and how to use it within the TOPMODEL framework. Hydrological Processes. 9: 161-182

Renschler, C.S. 2003. Designing geo-spatial interfaces to scale process models: The GeoWEPP approach. Hydrological Processes. 17:1005-1017

Renschler, C.S., D.C. Flanagan, B.A. Engel and J.R. Frankenberger. 2002. GeoWEPP - The Geo-spatial interface for the Water Erosion Prediction Project. Paper Number: 022171 An ASAE Meeting Presentation at the 2002 ASAE Annual International Meeting / CIGR XVth World Congress Sponsored by ASAE and CIGR Hyatt Regency Chicago Chicago, Illinois, USA July 28-July 31, 2002

Shary, P. A., L. S. Sharaya and A. V. Mitusova. 2002. Fundamental quantitative methods of land surface analysis. Geoderma, 107(1-2): 171-197.

Skidmore, A. K. 1990. Terrain position as mapped from gridded digital elevation data. Int. J. Geographical Information Systems. 4: 33-49.

Toma, Laura; Rajiv Wickremesinghe, Lars Arge, Jeffrey S. Chase, Jeffrey Scott Vitter, Patrick N. Halpin, and Dean Urban. Flow computation on massive grid terrains. *GeoInformatica, International Journal on Advances of Computer Science for Geographic Information Systems*, 2001. In preparation.

Wilson, J. P. 1996. GIS-based land surface/subsurface modeling: New potential for new models. *In:* 3rd International Conference/Workshop on Integrating GIS and Environmental Modeling. Santa Fe, New Mexico, Jan 21-25, 1996. National Centre for Geographic Information and Analysis, Santa Barbara, CA, USA. CD-ROM

Zevenbergen, L. W. and C. R. Thorne. 1987. Quantitative analysis of land surface topography. Earth Surface Processes and Landforms. 12: 47-56

Appendix 1

SOME USEFUL WEB SITES

DEM and GIS Software (Mostly Free or at least low cost)

3DEM 3D Perspective views: http://www.visualizationsoftware.com/3dem.html

3DMapper: Freeware version: http://solim.geography.wisc.edu/mapper/index.htm

3DMapper: Commercial version: http://www.terrainanalytics.com/

Global Mapper: http://www.globalmapper.com/

Map Window free GIS API Shell: http://www.mapwindow.com/index.html

SAGA GNU license GIS shell: http://134.76.76.30/saga/html/index.php

Irfanview Image Editor: http://www.irfanview.com/

CSIRO Flow Tube TOPOG Software: http://www.per.clw.csiro.au/topog/

Chris Maunder Australia CRCCH DEM Maker and DEM Viewer Add-ons to TOPOG: http://www.catchment.crc.org.au/products/models/the_models/DEMMaker/DEMMaker.htm

Moore, Gallant and Wilson's TAPES http://cres.anu.edu.au/outputs/tapes.html

Garbrecht and Martz's TOPAZ: http://grl.ars.usda.gov/topaz/TOPAZ1.HTM

Jo Wood's LandSerf: http://www.soi.city.ac.uk/~jwo/landserf/

John Fels TopoMETRIX: http://www.undersys.com/topometrix.html

MicroDEM: http://www.usna.edu/Users/oceano/pguth/website/microdem.htm

Mike Hutchinson's ANUDEM: http://cres.anu.edu.au/outputs/orderform-int.html

Keith Beven's TOPMODEL: http://www.es.lancs.ac.uk/hfdg/topmodel.html

AGWA Hydrological Extraction: http://www.tucson.ars.ag.gov/agwa/

Daylon Leveller DEM Editing software: http://www.daylongraphics.com/products/leveller/

AccuTrans 3D: http://www.micromouse.ca/landscapes.html

Surfer and Didger from Golden: http://www.goldensoftware.com/

DEM Data (Mostly Free)

Globe 1 km DEM data set at: http://www.ngdc.noaa.gov/seg/topo/gltiles.shtml

Geogateway SRTM: http://data.geocomm.com/dem/demdownload.html

ASTER DEM Data: http://edcdaac.usgs.gov/aster/dem_map.html

USGS SRTM Data: http://srtm.usgs.gov/index.html

USGS DEM Gateway: http://edc.usgs.gov/geodata/

USGS Seamless NED: http://gisdata.usgs.net/seamless/

USGS NED: http://edcnts12.cr.usgs.gov/neddsi/viewer.htm

USGA DEM Downloads from: http://seamless.usgs.gov/viewer.htm

USGS 1:250 DEM data: http://edcwww.cr.usgs.gov/glis/hyper/guide/1_dgr_demfig/index1m.html

GTOPO30, 1 km DEM for world: http://edcdaac.usgs.gov/gtopo30/gtopo30.html

CANADA TOPO: http://www.cits.rncan.gc.ca/cit/servlet/CIT/site_id=01&page_id=1-004.html

Canada TOPORAMA: http://toporama.cits.rncan.gc.ca/En/frame.html

Canada3D: http://www.cits.rncan.gc.ca/cit/servlet/CIT/site_id=01&page_id=1-005-002-005.html

Wyoming Data Warehouse: http://www.wygisc.uwyo.edu/24k/

Appendix 2

ORIGINAL RULE BASE FOR LANDFORM ATTRIBUTES (LM_ARULE)

No.	ATTR_IN	CLASS_OUT	MODEL NO.	B	B_LOW	B_HI	B1	B2	D
1	b.PROF	CONVEX_D	4	5.0000			2.5000		2.5000
2	b.PROF	CONCAVE_D	5	-5.0000			0.0000	-2.5000	2.5000
3	b.PROF	PLANAR_D	1	0.0000	0.0000	0.0000	-2.5000	2.5000	2.5000
4	b.PLAN	CONVEX_A	4	5.0000			2.5000		2.5000
5	b.PLAN	CONCAVE_A	5	-5.0000				-2.5000	2.5000
6	b.PLAN	PLANAR_A	1	0.0000	0.0000	0.0000	-2.5000	2.5000	2.5000
7	b.SLOPE	NEAR_LEVEL	5	0.5000				1.0000	0.5000
8	b.SLOPE	REL_STEEP	4	2.0000			1.0000		1.0000
9	QWETI	HIGH_WI	4	7.0000			3.5000		3.0000
10	QWETI	LOW_WI	5	0.5000				3.5000	3.0000
11	PMIN2MAX	NEAR_MAX	4	90.0000			75.0000		15.0000
12	PCTZ2TOP	NEAR_TOP	4	90.0000			75.0000		15.0000
13	PCTZ2ST	NEAR_DIV	4	90.0000			75.0000		15.0000
14	PCTZ2PIT	NEAR_PEAK	4	90.0000	0.0000	0.0000	75.0000	0.0000	15.0000
15	PCTZ2PIT	NEAR_MID	1	50.0000	50.0000	50.0000	25.0000	75.0000	25.0000
16	PCTZ2PIT	NEAR_PIT	5	10.0000				25.0000	15.0000
17	Z2PIT	HI_ABOVE	4	2.0000			1.0000		1.0000
18	PIT2PEAKZ	HI_RELIEF	4	10.0000			4.0000		6.0000
19	PMIN2MAX	NEAR_MIN	5	10.0000				25.0000	15.0000
20	PCTZ2TOP	NEAR_BOT	5	10.0000				25.0000	15.0000

Appendix 3

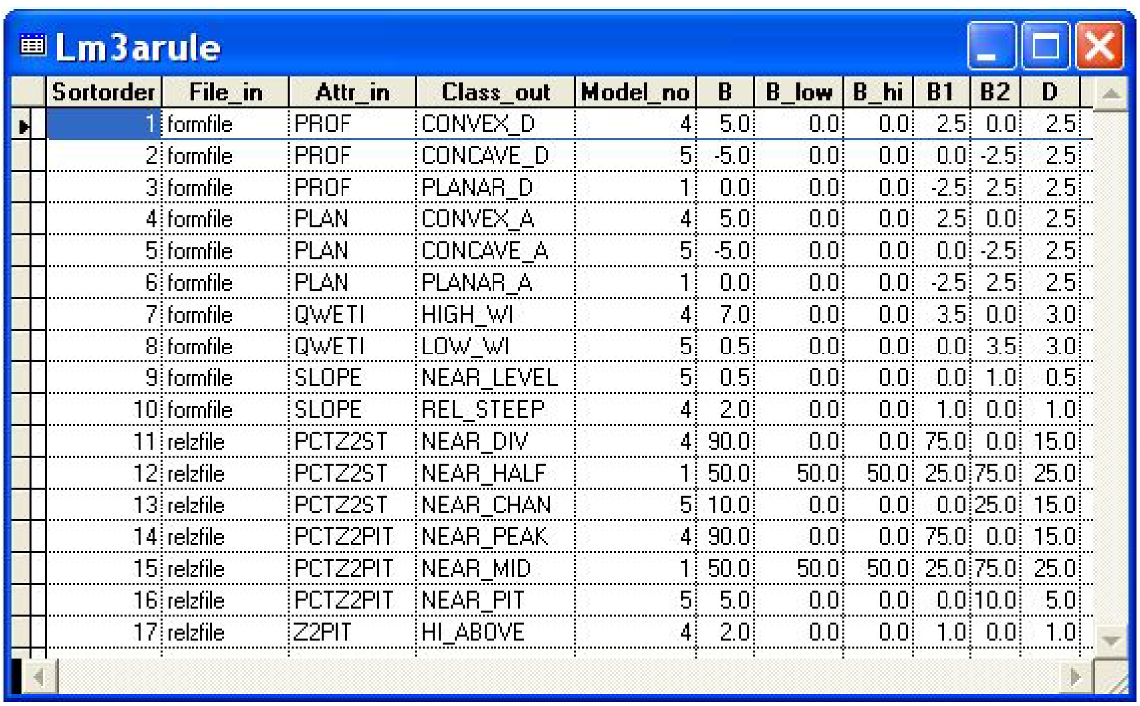
ORIGINAL RULE BASE FOR LANDFORM CLASSIFICATION (LM_CRULE)

FACET NAME	CODE	FUZZY ATTRIBUTE	WT	FACET NAME	CODE	FUZZY ATTRIBUTE	WT
Level Crest	LCR	NEAR-LEVEL	20	Saddle	SAD	CONCAVE_D	20
	LCR	NEAR_TOP	20		SAD	CONVEX_A	20
	LCR	NEAR_DIV	10		SAD	NEAR_MID	20
	LCR	PLANAR_2X	5		SAD	HI_ABOVE	10
	LCR	LOW_WI	5		SAD	HIGH_WI	5
	LCR	HIGH_ABOVE	5	Mid-slope	MDE	NEAR_MID	20
Divergent	DSH	REL_STEEP	20	Depression	MDE	CONCAVE_D	10
Shoulder	DSH	CONVEX_D	20		MDE	CONCAVE_A	10
	DSH	CONVEX_A	20		MDE	HIGH_WI	20
	DSH	NEAR_DIV	10		MDE	NEAR_LEVEL	20
	DSH	NEAR_TOP	10		MDE	HIGH_ABOVE	5
	DSH	HI_ABOVE	5	Foot-slope	FSL	NEAR_BOT	20
	DSH	LOW_WI	5		FSL	CONCAVE_D	20
Upper	UDE	NEAR_TOP	20		FSL	HIGH_WI	20
Depression	UDE	NEAR_MAX	10		FSL	CONCAVE_A	10
	UDE	HIGH_WI	10		FSL	REL_STEEP	5
	UDE	CONCAVE_D	10	Toe-slope	TSL	PLANAR_A	20
	UDE	CONCAVE_A	10		TSL	NEAR_BOT	20
	UDE	NEAR_LEVEL	10		TSL	REL_STEEP	10
	UDE	HI_ABOVE	5		TSL	PLANAR_D	10
Back-slope	BSL	PLANAR_D	20	Lower-slope Fan	FAN	NEAR_BOT	20
	BSL	PLANAR_A	20		FAN	PLANAR_D	20
	BSL	NEAR_MID	20		FAN	CONVEX_A	20
	BSL	REL_STEEP	10		FAN	REL_STEEP	10
	BSL	HI_ABOVE	5	Lower-slope	LSM	CONVEX_D	20
Divergent	DBS	PLANAR_D	20	Mound	LSM	CONVEX_A	20
Back-slope	DBS	CONVEX_A	20		LSM	REL_STEEP	20
	DBS	NEAR_MID	20		LSM	NEAR_BOT	20
	DBS	REL_STEEP	10		LSM	LOW_WI	10
	DBS	HI_ABOVE	5		LSM	NEAR_DIV	10
	DBS	LOW-WI	5	Level	LLS	NEAR_BOT	20
Convergent	CBS	CONCAVE_A	20	Lower-slope	LLS	NEAR_LEVEL	20
Back-slope	CBS	PLANAR_D	20		LLS	NEAR_PIT	10
	CBS	NEAR_MID	20		LLS	PLANAR_D	5
	CBS	REL_STEEP	10		LLS	PLANER_A	5
	CBS	HIGH_WI	5		LLS	HIGH_WI	5
	CBS	HI_ABOVE	5	Lower-slope	DEP	NEAR_PIT	20
Mid-slope	TER	NEAR_LEVEL	20	Depression	DEP	CONCAVE_A	10
Terrace	TER	NEAR_MID	20		DEP	CONCAVE_D	10
	TER	PLANAR_D	10		DEP	NEAR_LEVEL	10
	TER	PLANAR_A	10		DEP	NEAR_BOT	10
	TER	HI_ABOVE	5		DEP	HIGH_WI	10

Appendix 4

REVISED RULE BASE FOR LANDFORM ATTRIBUTES (LM3ARULE)

No.	ATTR_IN	CLASS_OUT	MODEL NO.	B	B_LOW	B_HI	B1	B2	D
1	PROF	CONVEX_D	4	5.0000			2.5000		2.5000
2	PROF	CONCAVE_D	5	-5.0000			0.0000	-2.5000	2.5000
3	PROF	PLANAR_D	1	0.0000	0.0000	0.0000	-2.5000	2.5000	2.5000
4	PLAN	CONVEX_A	4	5.0000			2.5000		2.5000
5	PLAN	CONCAVE_A	5	-5.0000				-2.5000	2.5000
6	PLAN	PLANAR_A	1	0.0000	0.0000	0.0000	-2.5000	2.5000	2.5000
7	QWETI	HIGH_WI	4	7.0000			3.5000		3.0000
8	QWETI	LOW_WI	5	0.5000				3.5000	3.0000
9	SLOPE	NEAR_LEVEL	5	0.5000				1.0000	0.5000
10	SLOPE	REL_STEEP	4	2.0000			1.0000		1.0000
11	PCTZ2ST	NEAR_DIV	4	90.0000			75.0000		15.0000
12	PCTZ2ST	NEAR_HALF	4	90.0000			75.0000		15.0000
13	PCTZ2ST	NEAR_CHAN	5	10.0000				25.0000	15.0000
14	PCTZ2PIT	NEAR_PEAK	4	90.0000	0.0000	0.0000	75.0000	0.0000	15.0000
15	PCTZ2PIT	NEAR_MID	1	50.0000	50.0000	50.0000	25.0000	75.0000	25.0000
16	PCTZ2PIT	NEAR_PIT	5	10.0000				25.0000	15.0000
17	Z2PIT	HI_ABOVE	4	2.0000			1.0000		1.0000



Lm3arule

Sortorder	File_in	Attr_in	Class_out	Model_no	B	B_low	B_hi	B1	B2	D
1	formfile	PROF	CONVEX_D	4	5.0	0.0	0.0	2.5	0.0	2.5
2	formfile	PROF	CONCAVE_D	5	-5.0	0.0	0.0	0.0	-2.5	2.5
3	formfile	PROF	PLANAR_D	1	0.0	0.0	0.0	-2.5	2.5	2.5
4	formfile	PLAN	CONVEX_A	4	5.0	0.0	0.0	2.5	0.0	2.5
5	formfile	PLAN	CONCAVE_A	5	-5.0	0.0	0.0	0.0	-2.5	2.5
6	formfile	PLAN	PLANAR_A	1	0.0	0.0	0.0	-2.5	2.5	2.5
7	formfile	QWETI	HIGH_WI	4	7.0	0.0	0.0	3.5	0.0	3.0
8	formfile	QWETI	LOW_WI	5	0.5	0.0	0.0	0.0	3.5	3.0
9	formfile	SLOPE	NEAR_LEVEL	5	0.5	0.0	0.0	0.0	1.0	0.5
10	formfile	SLOPE	REL_STEEP	4	2.0	0.0	0.0	1.0	0.0	1.0
11	relzfile	PCTZ2ST	NEAR_DIV	4	90.0	0.0	0.0	75.0	0.0	15.0
12	relzfile	PCTZ2ST	NEAR_HALF	1	50.0	50.0	50.0	25.0	75.0	25.0
13	relzfile	PCTZ2ST	NEAR_CHAN	5	10.0	0.0	0.0	0.0	25.0	15.0
14	relzfile	PCTZ2PIT	NEAR_PEAK	4	90.0	0.0	0.0	75.0	0.0	15.0
15	relzfile	PCTZ2PIT	NEAR_MID	1	50.0	50.0	50.0	25.0	75.0	25.0
16	relzfile	PCTZ2PIT	NEAR_PIT	5	5.0	0.0	0.0	0.0	10.0	5.0
17	relzfile	Z2PIT	HI_ABOVE	4	2.0	0.0	0.0	1.0	0.0	1.0

Appendix 5

REVISED RULE BASE FOR LANDFORM CLASSIFICATION (LM3CRULE)

Lm3crule

F_name	Fuzattr	Attrwt	Facet_no	F_code
LCR	NEAR_PEAK	30	11	1
LCR	NEAR_DIV	20	11	1
LCR	HI_ABOVE	10	11	1
LCR	NEAR_LEVEL	20	11	1
LCR	PLANAR_D	10	11	1
LCR	PLANAR_A	5	11	1
LCR	LOW_WI	5	11	1
DSH	NEAR_PEAK	30	12	2
DSH	NEAR_DIV	20	12	2
DSH	HI_ABOVE	10	12	2
DSH	CONVEX_D	20	12	2
DSH	CONVEX_A	10	12	2
DSH	LOW_WI	10	12	2
UDE	NEAR_PEAK	30	13	3
UDE	NEAR_DIV	20	13	3
UDE	HI_ABOVE	10	13	3
UDE	NEAR_LEVEL	10	13	3
UDE	CONCAVE_D	10	13	3
UDE	CONCAVE_A	10	13	3
UDE	HIGH_WI	10	13	3
BSL	NEAR_HALF	20	21	4
BSL	NEAR_MID	10	21	4
BSL	HI_ABOVE	5	21	4
BSL	REL_STEEP	20	21	4
BSL	PLANAR_D	15	21	4
BSL	PLANAR_A	25	21	4
BSL	LOW_WI	5	21	4
DBS	NEAR_HALF	20	22	5
DBS	NEAR_MID	10	22	5
DBS	HI_ABOVE	5	22	5
DBS	REL_STEEP	20	22	5
DBS	CONVEX_A	20	22	5
DBS	PLANAR_D	15	22	5
DBS	LOW_WI	10	22	5
CBS	NEAR_HALF	20	23	6
CBS	NEAR_MID	10	23	6
CBS	HI_ABOVE	5	23	6
CBS	REL_STEEP	20	23	6
CBS	CONCAVE_A	20	23	6
CBS	PLANAR_D	15	23	6
CBS	HIGH_WI	10	23	6
TER	NEAR_HALF	20	24	7
TER	NEAR_MID	10	24	7
TER	HI_ABOVE	5	24	7
TER	NEAR_LEVEL	30	24	7
TER	PLANAR_D	15	24	7
TER	PLANAR_A	20	24	7

Lm3crule

F_name	Fuzattr	Attrwt	Facet_no	F_code
SAD	NEAR_HALF	20	25	8
SAD	NEAR_MID	10	25	8
SAD	HI_ABOVE	5	25	8
SAD	NEAR_LEVEL	20	25	8
SAD	CONCAVE_D	20	25	8
SAD	CONVEX_A	20	25	8
MDE	NEAR_HALF	20	26	9
MDE	NEAR_MID	10	26	9
MDE	HI_ABOVE	5	26	9
MDE	NEAR_LEVEL	25	26	9
MDE	CONCAVE_D	10	26	9
MDE	CONCAVE_A	10	26	9
MDE	HIGH_WI	20	26	9
FSL	NEAR_CHAN	20	31	10
FSL	NEAR_PIT	10	31	10
FSL	REL_STEEP	10	31	10
FSL	CONCAVE_D	20	31	10
FSL	CONCAVE_A	20	31	10
FSL	PLANAR_A	10	31	10
FSL	HIGH_WI	20	31	10
TSL	NEAR_CHAN	20	32	11
TSL	NEAR_PIT	10	32	11
TSL	REL_STEEP	10	32	11
TSL	PLANAR_D	25	32	11
TSL	PLANAR_A	25	32	11
TSL	HIGH_WI	10	32	11
FAN	NEAR_CHAN	20	33	12
FAN	NEAR_PIT	10	33	12
FAN	REL_STEEP	10	33	12
FAN	CONVEX_A	25	33	12
FAN	PLANAR_D	25	33	12
FAN	LOW_WI	10	33	12
LSM	NEAR_DIV	10	41	13
LSM	NEAR_CHAN	20	41	13
LSM	NEAR_PIT	10	41	13
LSM	NEAR_PEAK	10	41	13
LSM	REL_STEEP	10	41	13
LSM	CONVEX_D	15	41	13
LSM	CONVEX_A	15	41	13
LSM	LOW_WI	10	41	13
LLS	NEAR_CHAN	20	42	14
LLS	NEAR_PIT	20	42	14
LLS	NEAR_LEVEL	40	42	14
LLS	PLANAR_D	5	42	14
LLS	PLANAR_A	5	42	14
LLS	HIGH_WI	10	42	14
DEP	NEAR_CHAN	20	43	15
DEP	NEAR_PIT	30	43	15
DEP	NEAR_LEVEL	20	43	15
DEP	CONCAVE_A	10	43	15
DEP	CONCAVE_D	10	43	15
DEP	HIGH_WI	10	43	15