GRASS Tutorial: watershed

Marji Larson

U.S. Army Construction Engineering Research Laboratory

ABSTRACT

This paper describes the use and syntax of the menu-driven program watershed. The series of programs accessed through this menu are used to derive geometric and morphologic information for drainage basins from digital elevation data. The goal of this procedure is to produce a drainage network and complementary divide network which would completely partition a drainage basin into sub-basins.

1. ORIENTATION

It is assumed that the reader is familiar with the concepts of grid cells, map layers, current mapset and location, and windows. For a review of these concepts refer to An Introduction to GRASS.

Land managers are given the job of protecting, maintaining, and developing areas of land. In order to perform this duty, information is often needed which is related to the delineation of a watershed and its stream network. Sometimes information related to stream and divide networks is available within a GIS system. If the channel and divide networks have to be manually interpreted from topographic maps or aerial photographs, with subsequent digitization to enter the information into a GIS system, the work could be quite tedious and time consuming. If this information could instead by derived from digital elevation data which is readily available for many locations, less work would be required, and the results might be more accurate. Accuracy is of course affected by the accuracy of the original elevation data, the resolution of the data used, and the effectiveness of the routines for extracting the information. The series of programs provided within the watershed menu is effective for the delineation and segmenting of channel networks, and for identifying a watershed boundary. Work is continuing to increase the effectiveness of delineating sub-basin boundaries as well. The logic for these routines is based primarily on code from Larry Band, Department of

Geography, University of Toronto. The original code was changed to include the use of GRASS routines for the accessing and storing of data, and some additions were made to provide additional output.

2. GETTING STARTED

The watershed command requires no arguments. The structure is interactive, and primarily menu-driven. Within the menus, the cursor can only be moved to lines requiring input. Movement from line to line can be made using the "RETURN" key. When "RETURN" is entered at the last line, the cursor returns to the top line of input. Mistakes can be overwritten, or control-h can be used to back up over errors on the current line. (These menus are created with the GRASS tool named *Vask*. See the manual entry for *Vask* for more information.) Checks for errors are made when possible.

Upon entry, the first input required from the user is a project name. A "project" file is created to hold information pertaining to map layer names and input parameter values. This is done in order to complete the process with a minimum of input required at each step, and to maintain a record of parameters used to obtain a particular result.

Upon entry of the watershed program, the user will first see the following:

```
         WELCOME TO THE WATERSHED SOFTWARE SYSTEM

  This program is designed to help you follow the correct
  steps to obtain a GRASS raster file depicting watershed
  geometry using a raster elevation file (with true elevation
  values) as a starting point.  All steps are recorded
  under the project name.



  Choose desired option:

  1.  Create new project
  2.  Create new project based on currently existing project
  3.  Work on an existing project

  4.  Remove old files
```

The user can exit from the program after completing any number of steps. Hence, the option to "Work on an existing project" allows a user to finish any remaining steps, or to redo particular steps, or both.

A new project may be created with all new inputs, or a new project can be created using inputs from a previous project.

The option to remove old files should be used to remove only miscellaneous files created as part of the watershed program. All map layers (cell files) created using this program should be removed in the usual way using command Gremove[2G]. The files which can be removed using option 4 above include those storing pit points and the miscellaneous file storing nodes information created as part of the step to code stream segments. These files should not be removed until the analysis of which they are a part is finished and over. An option is also provided to remove project files. Again, since this is a file which stores information to complete the project and which tells of the input parameters used to create a particular product, this should also not be removed prematurely. See the rest of this document for further explanation of these files and their usage.

Once an old or new project has been chosen, the user will be presented with the following menu and allowed to choose an option:

1. Filtering of elevation data
2. Locating pits
3. Calculating drainage accumulation/outlining watershed
4. Creating stream network
5. Coding stream segments/finding segment lengths
6. Finding subwatershed basins

7. Quit

The steps are presented in the order they should be used to complete the total watershed analysis for the purposes of creating all products. While input is being accepted for each individual step, the option is given whether to run the actual program. If the program has been run and completed without error, the word "COMPLETE" will appear on the above menu after the listing of that step.

An explanation of the steps are presented below, followed by an example which can be run on the elevation data available in the GRASS database under location spearfish.

2.1. FILTERING

This function smoothes elevation data using a particular filter.As a result of smoothing, the number of pit points is decreased significantly, eliminating many pits which are glitches in the data instead of true pits.

For the purposes of completing all the steps of the watershed analysis, this is the only optional step. If filtering of the elevation data is not done, it is likely that an abundance of pits will be found, and execution of the next couple of steps will be time consuming.

Theoretically, any data can be filtered with this program, but for the purposes of watershed analysis, the input layer must contain true elevation values. The user is asked for the input layer, an output layer name, and the desired number of filter iterations. It is normally sufficient to filter the data once. Some pit points will remain, but these are probably true pits. The filtering is done on 3x3 neighborhoods, using a weighted averaging technique. The filtering is done only on the input values, not on rows already averaged during the filtering process.

2.2. LOCATING PITS

This step generates a list of pit points designated by their grid cell coordinates. Pits are points which are characterized as having an elevation value less than or equal to that of all neighboring points. Input to this program should be a cell file containing true elevation values. It is usually helpful to filter the elevation file first (see "FILTERING" above). The pit points are sorted by decreasing elevation value, and stored in a miscellaneous file under the current location and mapset.

2.3. CALCULATING DRAINAGE ACCUMULATION

This step generates a drainage accumulation map layer from elevation data. The drainage accumulation map layer depicts a particular drainage basin, with each grid cell having a category value equivalent to its upstream drainage area (the number of cells within the drainage basin which would drain into it). This accumulation map layer is generated through recursion by starting at an outlet location and checking each adjacent grid cell for the criteria of being "upslope." An implied watershed boundary is generated, as all grid cells outside the watershed have category value zero (no data).

Also generated are an adjusted drainage direction map (adjusted to allow for the draining of water across lakes and flat areas), and an optional map layer product which denotes designated lakes, pits, and redirected routes from a pit point within each lake to the pouring point of the lake.

The elevation map layer used for input will be the same one used in the previous step to find pit points. Users must supply all output map names, the desired watershed outlet point, and threshold values for slope (what decimal slope value is considered "flat" for the purposes of including cells in the watershed basin) and for pit processing (number of meters to allow for error above a lake basin's pouring point when deciding which grid cells to include in the lake basin surrounding a pit point).

The watershed outlet acts as a starting place for this routine. The grid point chosen as the outlet is used as the first cell in the watershed, and then all adjacent cells are checked for the criteria of 'upslope.' Recursive calls are made to a 'hill-climbing' routine each time an adjacent upslope cell is found. These recursive calls continue until the edge of the watershed is reached. Thus, the chosen outlet point determines the accuracy of the product, and attempts should be made to find a point within a drainage channel. In order to choose an outlet point accurately, the user should follow these recommendations:

1. use the displaying techniques available in GRASS to "window in" on the area containing the desired outlet point (displaying the elevation file being used). 2. use "whereami" techniques within display or the Dwhere[2D] command to locate a point in an area where elevation category changes have sharp crenulations.

When looking for a point where elevation changes have sharp crenulations, it is important to avoid the use of a rescaled version of the elevation map layer where two or more elevation values have been clumped into one category. Depending on the range of elevation values in the map layer containing true elevation values, it may be impossible to display the map layer and have enough color distinction between different elevation values in order to find an area where the changes have sharp crenulations. In this case, a copy of the elevation map layer in the vicinity of the desired outlet point should be made. This can be done by first creating a small current window around the desired outlet point and then using the GRASS command resample[1] (or Gresample[2G]). The range of elevation values in the new map layer (created using resample[1]) should be small enough to provide clear color distinctions between elevations on the color monitor display. The map layer can then be displayed using Dcell[2D] and the outlet point chosen using Dwhere[2D].

The slope threshold is used to aid in decisions during the process of hill-climbing by defining "flatness." When a grid cell's neighbors are being checked for inclusion in the watershed, the decision is based on whether that neighbor drains into the current cell in question or on whether the slope magnitude of that neighbor is sufficiently small to justify traversing that cell on the way to finding the watershed boundary (thus defining "flatness"). This definition of what is meant by the slope magnitude being "sufficiently small" is left to the user through input of the slope threshold. Slope in this routine is calculated as a decimal value using a gradient calculation of the square root of the sum of the central difference in the x direction squared plus the central difference in the y direction squared. A reasonable value for the slope threshold is dependent on the terrain in question and the resolution of the elevation data provided. As a general rule, the slope threshold should be smaller for a finer resolution of elevation data, due to the fact that finer resolution data generally implies smaller differences in elevation values between two grid cells (again, dependent on the terrain begin depicted). The user may need to experiment with various threshold values until desired results are obtained, but as a suggested starting point, a slope threshold of about 2/1000 (.002) could be tried for 30 m. data, and about 2/100 (.02) for 100 m. data. If the results from this step include grid cells in the watershed which seemingly should not be included, a smaller slope threshold value may help create more accurate results.

As part of this step, "local drainage areas" (or "lakes") are defined for each pit. Starting at a pit point, surrounding grid cells are included in its local drainage area by satisfying one of two criteria: either the adjacent cell drains naturally to the pit point in question, or it has a slope magnitude of zero AND has an elevation value equal to that of the adjoining cell which is already assigned to the drainage area. (When assignment occurs due to this second set of criteria, the assigned cell has its drainage direction redirected to the adjoining cell, and thus ultimately towards the pit point.) Once all cells meeting the criteria have been included in the drainage area, a "lake area" (which may or may not represent an actual lake) is extracted from it. The lake area is determined by finding the pouring point of the local drainage area (the cell along the boundary with the lowest elevation), and including in the lake only the cells within the drainage area that have elevation equal to or lower than the pouring point plus some error tolerance defined by the user. This error tolerance is what is input as the <u>pit threshold</u>. In general, it also holds true here that the finer the resolution of the data, the smaller the value of the needed pit threshold. For 30 m. data, a reasonable pit threshold value is 5 m., and for 100 m. data, a pit threshold of 10 m. or more is probably needed. This threshold may also need to be adjusted according to the type of terrain being depicted. Grid cells which are assigned to lakes may have their drainage directions adjusted later in the routine as the lake is being traversed during an uphill climb.

In order to obtain the desired results, the user may need to experiment with various combinations of inputs. It is essential that the drainage accumulation output map layer contain meaningful data before its use in the next step, so the user should check the results by displaying the resultant map layer before continuing. A quick preliminary check for meaningful results can be done using Gdescribe[2G] on the output map layer. If Gdescribe[2G] gives the values of the output map layer as all small, then the results are not meaningful unless the user desires a very small watershed to be defined. The size of the values in the output layer are directly related to the size of the watershed depicted. Since the category values within the accumulation map layer may be quite large, it may be helpful to rescale the data (using either rescale[1] or Grescale) before displaying it as it is difficult to interpret the display of a large number of categories.

The current window setting is used in this step. The user should take care that the desired watershed is within the current window if possible. (The watershed boundary may extend beyond the edge of available elevation input data.) However, it is to the user's advantage to create as small a window as is reasonable since this step is time-consuming, and will operate more quickly on a smaller window size.

2.4. CREATING STREAM NETWORK

This step generates a stream network map layer from a drainage accumulation input layer. The user must supply an accumulation threshold (number of cells that must drain into a given cell before that cell is saved as part of the stream network) as well as the desired number of thinning iterations. The accumulation threshold will determine the density of the tributaries in the resulting stream network. The smaller the threshold, the more tributaries in the result. All cells within the input layer which have category values less than the threshold will be eliminated. Note that if the results of the previous step are not meaningful, it will be impossible to obtain a stream network product here. The threshold value should be chosen based on the values in the input map layer, and on the desired tributary density; but a reasonable threshold estimate for 30 m. data would be 60 or 70. (An equivalent threshold for 100 m. data would be smaller.) If the resulting stream network is to be used to complete the rest of the steps of this interface, it is essential that thinning be done to one pixel width. The user should begin by using a fairly small number of thinning iterations (3 to 5), and rerun this step with a higher number of iterations if the results do not meet this criteria. Display the map to check the results.

2.5. CODING STREAM SEGMENTS

This step generates a stream network which is coded by stream segment, given the one pixel wide stream network generated in the previous step. At each branching of the original network, a new category number is assigned. Thus, the resulting network has each channel segment denoted by a unique category number. The resulting map layer is an optional product.

A miscellaneous file is also created during this step. The file contains a listing of the nodes (beginning and ending points) for each segment, as well as the length of each segment, and is needed for the running of the next step.

2.6. FINDING SUBWATERSHED BASINS

This final step in the process generates an output layer containing a unique category number and area for each subbasin within a given watershed and for a given stream network. Map layer products from previous steps (adjusted drainage direction map layer and the miscellaneous nodes file) are used. The numbering of subbasins corresponds with the numbering of channel segments generated in the previous step.

Results of this step may not be completely accurate. Errors occur due to problems with the routing of water over lakes and flat areas, as well as possible problems with the thinning routine. To create a product using alternate means, see suggestions in Appendix A.

3. EXAMPLE SESSION

Following is an example session that can be used on the *spearfish* database. The watershed command can only be reached from the quick access version of GRASS. Once you have entered GRASS, simply enter the command watershed with no arguments.

Menus as you should see them are presented here. The step which calculates drainage accumulation may take a good deal of time to complete, depending on the size of the current window. In order to speed up this example, first set your current window to the following coordinates (these will include the entire watershed to be outlined in this example):

north:	4924000
south:	4914010
east:	600000
west:	590220

resolution: 30 by 30

Now enter the GRASS command name watershed and you should see the following on your screen.

```
              WELCOME TO THE WATERSHED SOFTWARE SYSTEM

  This program is designed to help you follow the correct
  steps to obtain a GRASS raster file depicting watershed
  geometry using a raster elevation file (with true elevation
  values) as a starting point.  All steps are recorded
  under the project name, which must be a legal GRASS
  system filename, unique within the current location and
  mapset.

  Choose desired option:

  1.  Create new project
  2.  Create new project based on currently existing project
  3.  Work on an existing project

       Desired option:   1_


        AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                     (OR <Ctrl-C> TO CANCEL)
```

You will need to start by choosing "Create new project." Then enter a project name. For this example, the project name entered is demo.

```
PROJECT NAME:    demo

Choose desired option:

1.  Filtering of elevation data
2.  Locating pits
3.  Calculating drainage accumulation/outlining watershed
4.  Creating stream network
5.  Coding stream segments/finding segment lengths
6.  Finding subwatershed basins

7.  Quit



    Choose desired option:   1__



     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
```

You can now begin the series of steps to complete a watershed analysis. The steps are presented in this menu in the order they should be completed, so begin with step 1 ("Filtering of elevation data"). Use as input the map layer elevation.

```
PROJECT NAME:    demo

Step1:  Filtering of elevation data

Input:
    Unscaled elevation map layer:             elevation_____________________

Output:
    Filtered elevation map layer:              filt.elev_____________________
    Number of filter iterations:                1____


            Run filter program?  yes




     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                  (OR <Ctrl-C> TO CANCEL)
```

Note that at each step you are given the option of running the program. If you enter "no" here, the map layer names and input parameters will be recorded under the project name, but the word "COMPLETE" will not appear on the main menu. If you choose to run the program by entering "yes", the next time you are returned to the main menu you will see the following provided the routine completed without error.

```
PROJECT NAME:    demo

Choose desired option:

1.  Filtering of elevation data                                 COMPLETE
2.  Locating pits
3.  Calculating drainage accumulation/outlining watershed
4.  Creating stream network
5.  Coding stream segments/finding segment lengths
6.  Finding subwatershed basins

7.  Quit



    Choose desired option:   2__



     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
```

Now run the next step ("Locating pits").

```
PROJECT NAME:    demo

Step2:  Locating pits

Input:
    Will use filtered elevation data from Step 1.

Output:
    Will create pits file under element watershed/pits,
    with name of [filtered] elevation data.


            Run pits program?  yes





     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                  (OR <Ctrl-C> TO CANCEL)
```

Before running the next step ("Calculating drainage accumulation/ outlining watershed") you need to know the grid coordinates for the outlet point. For this example, UTM coordinates have been supplied.

```
PROJECT NAME:    demo

Step3:  Drainage accumulation/watershed identification

Input:
    Will use [filtered] elev data and pits information
    Outlet coordinates:  easting:            595220.00___________
                    northing:            4923000.00__________
    Slope threshold (decimal):               0.0002_
    Pit threshold (meters):                 5______

Output:
    Drainage accumulation map layer:        accum.demo____________________
    Aspect map layer:                     ptr.demo_____________________
      (adjusted pointer map)
    Lakes map layer (optional):             lakes.demo____________________

            Run accumulation program?  yes


     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                  (OR <Ctrl-C> TO CANCEL)
```

Note that the lakes map layer is optional. There is no need for you to save this output in a cell file unless you are interested in seeing a display of the pit points, their surrounding lake areas, and the redirected route from the pit point to the pouring point for the lake.

This step may take quite some time to complete, depending on the size of the window and the number of pit points found. Processing of the pits in this routine is lengthy. Once the step is completed, you should exit the watershed program and check the results. If the results of this step are not meaningful, the rest of the process cannot be completed. Use Gdescribe[2G] to see if the values in the result file are large, or display the resulting map layer to see if it contains a depiction of the desired drainage basin. (This resulting map layer will need to have a color table assigned. This can be done through the use of the support[1] command. If the range of category values is large, the map layer may need to be rescaled first using the rescale[1] command in order to make a meaningful display on the color monitor screen.) If the results of this step are not satisfactory, attempt to choose a better watershed outlet point before reentering the program.

Once you have evaluated the results of this step, reenter the watershed command, and this time choose the option to "Work on an existing project". Again, enter the project name demo, and you should receive the main menu, with the choice of steps, and the word "COMPLETE" after the first three options. If the results from this routine were not satisfactory, you would rerun the third step. You should have gotten meaningful results with this example, so continue by choosing option 4.

```
PROJECT NAME:    demo

Step4:  Thinning of accumulation file

Input:
    Will use drainage accumulation map layer from Step 3

Output:
    Thinned network map layer:                thin.demo_____________________
    Number of thinning iterations:              4______
    Accumulation threshold:                    300____

            Run thin program?  yes




     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                  (OR <Ctrl-C> TO CANCEL)
```

Again, you need to check the results of this step to be sure that the stream network has been thinned to one pixel width. Once this has been accomplished, continue to the next step. If the stream network has not been thinned to one pixel width, an error message will be produced during the next step and it will not complete correctly.

```
PROJECT NAME:    demo

Step5:  Channel segment coding

Input:
    Extended thinned network map layer:         thin.demo_____________________

Output:
    Coded network map layer (optional):          marked.demo___________________
    Nodes file will have extended network name


            Run code program?    yes




     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                  (OR <Ctrl-C> TO CANCEL)
```

The cell file depicting the stream network with each channel segment "coded" (given a unique category number) is an optional product. You only need to provide a map layer name for this if you wish to save the results for displaying. However, this step must be completed in order to continue to the next and final step in the series. Note that the above menu asks for an extended thinned network map layer. If you are planning to use results from this series for input into a sediment runoff routine (as through the command grass.armsed[1]), the stream segments may need to be extended to ridge boundaries in order to provide the kind of information needed, such as the lengths of channel segments which completely divide subbasins. The extensions can be created in a manner similar to that used for creating incomplete ridge information as explained in Appendix A of this document.

```
PROJECT NAME:    demo

Step6:  Subwatershed designation

Input:
    Will use nodes file from Step 5.

Output:
    Subwatershed map layer:                   basin.demo____________________


            Run subbasin program?  yes





     AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
                  (OR <Ctrl-C> TO CANCEL)
```

Once this step is completed, you should now see the main menu with the word "COMPLETE" after each step (see below).

```
PROJECT NAME:    demo

Choose desired option:

1.  Filtering of elevation data                                 COMPLETE
2.  Processing pits                                          COMPLETE
3.  Calculating drainage accumulation/outlining watershed       COMPLETE
4.  Creating stream network                                  COMPLETE
5.  Coding stream segments/finding segment lengths              COMPLETE
6.  Finding subwatershed basins                               COMPLETE

7.  Quit



    Choose desired option:   0__



       AFTER COMPLETING ALL ANSWERS, HIT <ESC> TO CONTINUE
```

This concludes the example session. If the results of the last step (finding subbasin boundaries) are not satisfactory, and you wish to obtain a more complete product, follow the instructions for an alternate method given in Appendix A.

APPENDIX A: AN ALTERNATE METHOD FOR DELINEATING SUBBASINS WITHIN A WATERSHED

The following steps explain a method for delineating subbasins within a given watershed.

First, obtain a map layer depicting the watershed perimeter. If the result of the drainage accumulation step from the watershed analysis program is a map layer named drain, then the watershed perimeter map layer could be derived using the GRASS commands Ggrow[2G] and Gmapcalc[2G] in the following manner:

```
     Ggrow drain drain.big

     Gmapcalc boundary = drain.big - drain
```

The Ggrow[2G] command "grows" the drainage area by one cell, and then the subtraction of the original area using Gmapcalc[2G] leaves a resulting file that is the grown portion only. (The map layer drain.big can now be removed.)

Second, create a map layer depicting ridges within the watershed. This can be done by digitizing the ridges from a paper map (using the digitizing software available in GRASS) or using the digitizing available as part of the display software with the following steps as guidelines.

1. "Overlay" the boundary map layer and the streams map layer (derived from the watershed analysis software) onto a map layer which can be used to delineate ridges. For instance, the aspect map layer derived with the GRASS program slope.aspect[1] will imply ridges by the change in aspect. The command:
   ```
   Gmapcalc overlay = 'if(streams || boundary, 26, aspect)'
   ```
   would create a map layer (overlay) with the aspect categories (0-25) overlain by the streams and boundary in category 26. A color table must be assigned to the map layer overlay (the GRASS command support[1] can be used for this), and then the color for category 26 should be adjusted to cause the streams and boundary to stand out from the other categories (use the GRASS commands Dcell[2D] and d.colors[1] or do this within display[1]).
2. The ridges can now be drawn in with the mouse, by first using the GRASS command Dcell[2D] to display the map layer overlay and then the command d.digit[1] to draw the ridge lines. The lines drawn to represent ridges should be put into a map layer named ridges.
3. Create the final map layer by combining the ridges with the boundary using the following command:
   ```
   Gmapcalc ridges.all = '(boundary || ridges)'
   ```

The temporary map layers boundary, overlay, and ridges can now be removed.

Finally, the GRASS command basins.fill[1] can be used to generate a map layer depicting subbasins. As input, the command needs the ridges map layer (containing the watershed perimeter as well as the interior ridges as in ridges.all above) and a coded stream network with uniquely numbered stream segments which can be generated using the watershed analysis tool. A number of passes must also be supplied as input. This is the number of passes that will be made through the data in an attempt to fill in the subbasins with the category number corresponding to the category number of the channel segment passing through the given subbasin. It is usually sufficient to use a small number of passes, like 2 or 3. If the program basins.fill[1] is run, and the results are incomplete ("holes" exist in one or more subbasins), the program should be rerun with a higher number of passes.