Geomorphometry in GRASS GIS

3 authors:

Jaro Hofierka
Pavol Jozef Šafárik University in Košice
70 PUBLICATIONS 2,445 CITATIONS
SEE PROFILE

Helena Mitasova
North Carolina State University
217 PUBLICATIONS 4,795 CITATIONS
SEE PROFILE

Markus Neteler
mundialis GmbH & Co. KG
179 PUBLICATIONS 3,791 CITATIONS
SEE PROFILE

CHAPTER 17

Geomorphometry in GRASS GIS

J. Hofierka, H. Mitášová and **M. Neteler**

how to set-up GRASS GIS · computing DEMs from various data sources · local and regional land-surface parameters · land-surface modelling and applications · DEM quality analysis · GRASS command examples with online database

1. GETTING STARTED

GRASS (Geographic Resources Analysis Support System) is a general-purpose Geographic Information System (GIS) for the management, processing, analysis, modelling and visualisation of many types of georeferenced data. It is Open Source software released under GNU General Public License (GPL, see http://www.gnu.org) and as such it provides a complete access to its source code written in ANSI C programming language. The main component of the development and software maintenance is built on top of highly automated web-based infrastructure sponsored by OSGeo Foundation, http://grass.osgeo.org in Trento, Italy with numerous worldwide mirror sites. This chapter is based on GRASS 6.2 version available for all commonly used operating systems. It includes 2D raster and 3D voxel data support, a new topological 2D/3D vector engine and capabilities for vector network analysis. Attributes are managed in a SQL-based DBMS.

1.1 Installing and running GRASS GIS

Complete information about GRASS GIS features, software installation and usage can be obtained from the GRASS homepage (http://grass.osgeo.org). Neteler and Mitášová (2008) provide detailed information about the use of GRASS including land-surface modelling and analysis, and various tutorials in several languages are available at http://grass.osgeo.org/gdp/tutorials.php. GRASS binaries for different architectures, source code, as well as the user's and programmer's manuals can be downloaded from the GRASS homepage: http://grass.osgeo.org/download/.

Developments in Soil Science, Volume 33 © 2009 Elsevier B.V.
ISSN 0166-2481, DOI: 10.1016/S0166-2481(08)00017-2. All rights reserved.

TABLE 1 GRASS commands naming convention

Prefix	Functional group	Example command
d.*	display, query	d.what.rast
r.*	2D raster	r.watershed
r3.*	3D raster (voxel)	r3.mapcalc
i.*	imagery	i.rectify
v.*	2D/3D vector	v.net
g.*	general	g.remove
ps.*	postscript maps	ps.map
db.*	database	db.select

The easiest way to learn GRASS is to start with an existing, ready-to-use data set. Several are available at the GRASS web site.[1] At geomorphometry.org, we provide the GRASS database for the Baranja Hill dataset used in this chapter along with the shell script file containing all GRASS commands used to produce the figures shown here and perform the described analysis.

GRASS data are stored in a directory referred to as *database* (also called `GISDBASE`), in our case the directory `grassdata`. Within this *database*, the projects are organised by *locations* (subdirectories of the *database*): the provided data set is therefore a *location* called `baranja`. It is important to know that each *location* is defined by its coordinate system, map projection and geographical boundaries. Each *location* can have several *mapsets* (subdirectories of the *location*) that are used to subdivide the project into different topics, subregions, or as workspaces for individual team members. Each *mapset* includes subdirectories for raster and vector data, attribute data and a working (current) spatial extent definition file `WIND`; all these subdirectories and files are hidden from the user. When defining a new *location*, GRASS automatically creates a special *mapset* called `PERMANENT` which is used to store the core data, default spatial extent and coordinate system definitions.

GRASS has over 350 modules, so it is helpful to get familiar with its naming convention — it is very intuitive, as shown in Table 1. The prefix indicates the functional group (type of operation that the command performs), the word after the dot describes what the command does or what type of data it works with.

After downloading and installing GRASS and the test data set, GRASS 6.2 can be launched from either the menu or from a terminal window by typing:

```
grass62
```

A GRASS startup window will open as shown in Figure 1. The path to the *database* goes into the first field, then `baranja` is selected as *location* and `topobook` as *mapset*. After clicking the *Enter GRASS* button at bottom left, basic information about the GRASS version and help access appears followed by the GRASS shell prompt and the GUI display manager (`gis.m`, see Figure 2). GRASS commands

[1] http://grass.osgeo.org/download/data.php.

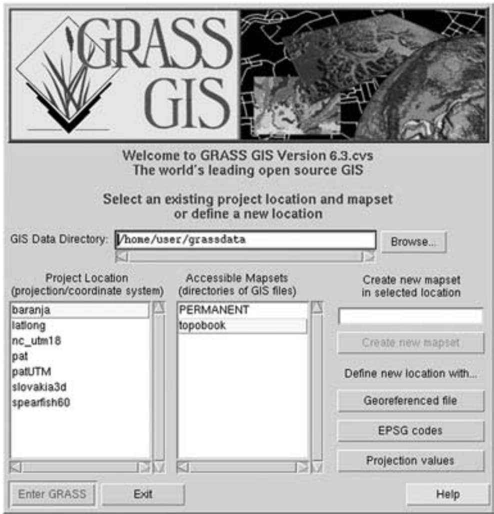

FIGURE 1 GRASS 6 startup screen with selection of database, location and mapset.

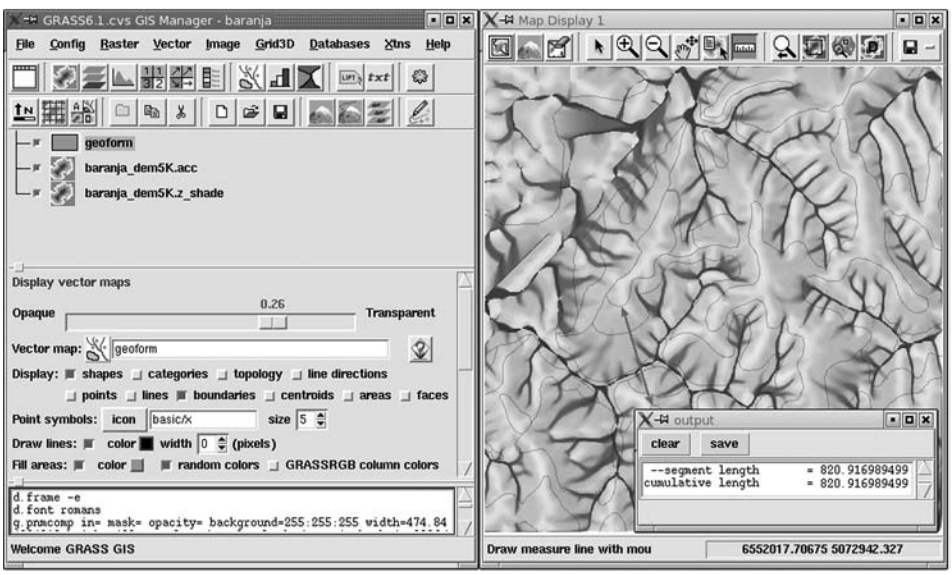

FIGURE 2 GRASS 6 graphical user interface `gis.m`.

can be entered either via GUI, or by typing the commands directly in the UNIX shell window. Execution of tasks that require a sequence of GRASS commands or operating system procedures can be automated by writing a shell script; a feature that is similar to ARC Macro Language or Avenue of ESRI GIS products.

GRASS includes a set of scripts, which behave like standard GRASS modules. In the next sections we provide a couple of examples that can be used directly in GRASS via the UNIX shell. Lines starting with a '#' indicate a comment that is not interpreted by the shell.

1.2 Importing, displaying and computing DEMs

Grid-based DEMs in various formats can be imported using the `r.in.gdal` command (refer to its manual page for the list of supported formats). Elevation data represented by digitised contours or measured points can be imported using the `v.in.ogr` command that supports numerous vector formats while `v.in.ascii` is used for data given as an ASCII list of (x, y, z) coordinates. Very dense ASCII point data, such as those acquired by LiDAR, can be directly converted to raster using `r.in.xyz` that performs a binning procedure based on different statistical measures (min, max, mean, range, etc.). For example, the data used in this book can be imported as follows:

```
# import contours from a SHAPE file
v.in.ogr -o dsn=contours5K.shp output=contours5K
# import raster DEM in Arc ASCII GRID format
r.in.arc input=DEM25m.asc output=DEM25m
# import Landsat imagery in LAN format
r.in.gdal -o input=bar_tm.lan output=bar_tm
```

Grid-based DEMs can be displayed as 2D raster maps and as 3D views (we use here command line but viewing is best handled through GUI, such as `gis.m`; `nviz`):

```
# zoom to raster map
g.region rast=DEM25m

# display 2D raster DEM
d.mon x0
d.rast DEM25m

# display shaded 2D raster DEM
r.shaded.relief DEM25m
d.rast DEM25m_shaded
d.his h=DEM25m i=DEM25m_shaded

# display 3D views
nviz elev=DEM25m vect=contours5K
```

In addition to the internal display tools, GRASS data can be viewed using external programs such as `QGIS` (http://www.qgis.org, see Figure 3) for 2D maps and `paraview` (http://www.paraview.org) for 3D visualisation. `QGIS` can read

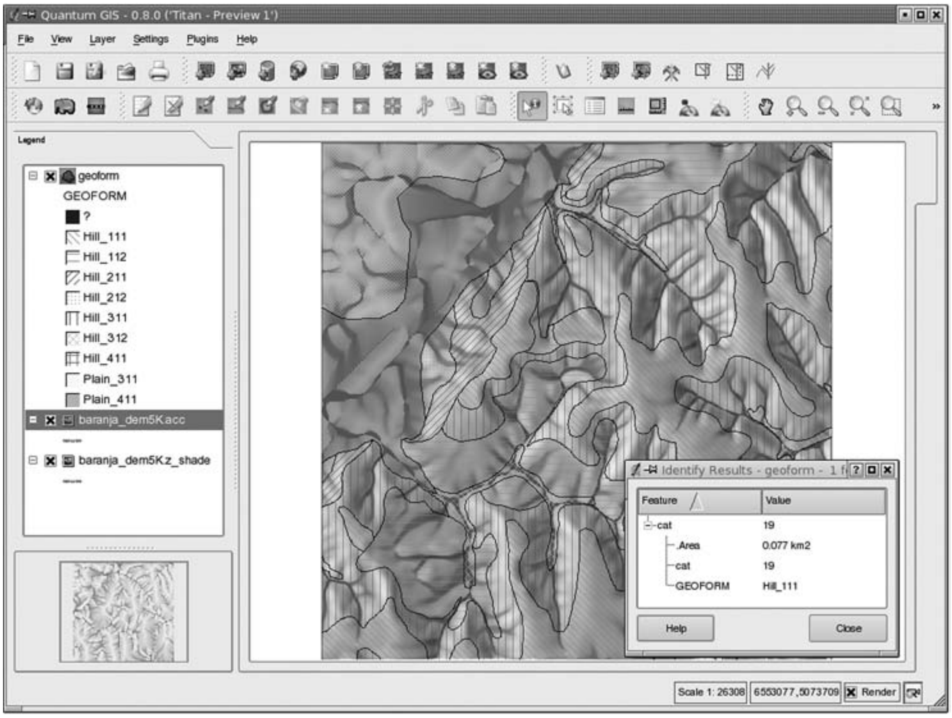

FIGURE 3 QGIS graphical user interface with GRASS 6 support.

the GRASS raster and vector data directly. Its GRASS plugin also offers a toolbox, providing GUI access to important GRASS commands for data analysis. The `r.out.vtk`, `r3.out.vtk` and `v.out.vtk` commands are used to export raster and vector data into VTK format readable by `paraview`.

1.3 Computing a DEM from contours or point data

If the data are given as contours or points we need to use spatial interpolation to first create a grid DEM. GRASS includes several modules for gridding irregularly spaced point or contour/isoline data. A simple inverse distance weighted interpolation is implemented as a `v.surf.idw` module. While this basic method is easy to use, it is not particularly suitable for elevation surfaces (Mitas and Mitášová, 1999). To compute DEM from rasterised contours, `r.surf.contour` can be used.

A more sophisticated interpolation method is based on the variational approach represented by *Regularised Spline with Tension* (RST, see more details in Mitas and Mitášová, 1999; Neteler and Mitášová, 2008). From the viewpoint of geomorphometric analysis, it is important that this interpolation function is differentiable to all orders (Mitášová et al., 1995). Using this property, topographic parameters can be computed simultaneously with interpolation. RST has been explicitly defined by Mitášová et al. (1995) up to four dimensions; for geomorphometric analysis the bivariate function implemented as `v.surf.rst` is the most relevant. The trivariate version of RST has been implemented in GRASS as `v.vol.rst`.

The behaviour of RST interpolation in the modules is controlled by the following parameters:

- *tension*;
- *smoothing*;
- *anisotropy*;
- *minimum and maximum distance between points*.

The parameters can be selected empirically, based on the knowledge of the modelled phenomenon and function, or automatically, by minimisation of the predictive error estimated by a cross-validation procedure (Hofierka et al., 2002).

The tension parameter controls the behaviour of the resulting surface — from a stiff steel plate to a thin, flexible membrane. Using a high tension, the influence of each point is limited to a relatively short distance, while with very low tension each point has a long range of influence. The RST method is scale dependent and the tension works as a rescaling parameter (Neteler and Mitášová, 2008).

Using the smoothing parameter, the RST behaves like an approximation function, i.e. the resulting surface does not pass through the given points, but approximates the input values. This parameter is useful in modelling noisy data, where higher smoothing can filter out the noise, or alternatively, when a phenomenon needs to be modelled at a lower level of detail. Tension and smoothing parameter are linked; lower tension automatically leads to increased smoothing (for more details see Mitášová et al., 2005).

The anisotropy parameter can be used for interpolation of spatially asymmetric data. Anisotropy is defined by orientation of the perpendicular axes characterising the anisotropy and a scaling ratio of the perpendicular axes (a ratio of axes sizes). These parameters scale distances (i.e. the value of tension) in the two perpendicular directions that should fit the spatial pattern of the anisotropic phenomenon.

Minimum and maximum distances between points control the number of points that are actually used in interpolation after reading the input data. The minimum distance allows the user to eliminate the points that are so close to each other that they can be considered identical for the given DEM resolution. The maximum distance can be used only for vector lines with a constant elevation value (e.g. contours) and it allows the user to automatically add points on the contour line if the points are farther apart than the given maximum. The distance parameters also influence the effect of the tension parameter, because tension works as a distance-scaling factor. Therefore, the tension can be set with or without spatial normalisation. The density of data does not affect the normalised tension parameter.

The RST method is appropriate for interpolation of various types of data — irregular data using, for example, `v.surf.rst`, regular (existing grid-based DEM) using `r.resamp.rst`, or alternatively converting grid data to points (`r.to.vect` or `r.random`) that can be re-interpolated using `v.surf.rst`.

2. DERIVING LAND-SURFACE PARAMETERS

GRASS includes an extensive set of modules for deriving land-surface parameters (shown here in Figures 4–20 for the Baranja Hill data set) and performing spatial analysis that involves elevation data.

2.1 Local parameters: slope, aspect, curvatures and derivatives

Topographic (or geomorphometric) analysis provides tools to compute a set of parameters that represent geometrical properties of the land surface. Local parameters describe land-surface properties both at a point and in its immediate surroundings. They can be computed based on the principles of differential geometry using partial derivatives of the mathematical function representing the surface. Local approximation methods are usually applied to estimate derivatives on a regular grid. A surface defined by the given grid point and its 3×3 neighbourhood is approximated by a second-order polynomial, and partial derivatives for the given centre grid point are computed using one of the common finite difference equations — e.g. the method of Horn (1981). This approach works well for smooth and non-flat areas. However, for high resolution data representing relatively flat areas with small differences in elevations or noisy surfaces, the small neighbourhood may not be sufficient to adequately capture the geometry of land-surface features. Also the approximation needs to be modified to estimate derivatives for grid cells on edges of the study area, where the complete 3×3 neighbourhood is not available.

A more general approach to estimation of partial derivatives is to use a differentiable function for DEM interpolation. Then the local surface parameters can be computed using an explicit form of the function derivatives, usually simultaneously with interpolation. However, this task is not trivial because the interpolation function must, at the same time, fulfil several important conditions necessary for reliable land-surface modelling.

GRASS provides both approaches for deriving land-surface parameters. The modules based on RST perform simultaneous interpolation and computation of partial derivatives including the following local land-surface parameters defined in Mitášová and Hofierka (1993):

- *slope* (steepest slope angle, a magnitude of gradient);
- *aspect* (slope orientation, direction of gradient, steepest slope direction, flow direction);
- *profile curvature* (surface curvature in the direction of gradient);
- *tangential curvature* (surface curvature in the direction of contour tangent);
- *mean curvature* (an average of the two principal curvatures).

Alternatively, a user can output first- and second-order partial derivatives instead of land-surface parameters and use them to compute additional maps, such as slope or curvature in any given direction. Plan curvature (contour curvature) can be derived from the tangential curvature and the sine of the slope angle (Mitášová and Hofierka, 1993).

It is important to note that land-surface parameters (especially curvatures based on second-order derivatives) are very sensitive to the quality of interpolation process. For example, interpolation from contours may lead to a false pattern of waves along the contours that can be visible only on a map of profile curvature. This is caused by a very heterogeneous distribution of input data — distances between points on the contours are relatively small, while they are large between the contours. These artifacts can be minimised by tuning the RST parameters (Neteler and Mitášová, 2008). The increase of minimal distance between points to a value that reflects an average distance between contours will reduce the heterogeneity of data density (points that are too close to each other will be removed from the interpolation). The decrease of tension and increase of smoothing will lead to a smoother surface with filtered-out small land-surface variations.

Using the RST method with properly set parameters, the DEM and land-surface parameters can be computed using a single command as follows:

```
v.surf.rst input=contours5K elev=b_dem5K.z
         slope=b_dem5K.s aspect=b_dem5K.a
         pcurv=b_dem5K.pc tcurv=b_dem5K.tc
         mcurv=b_dem5K.mc devi=b_dem5K_dev
         dmin=7.5 dmax=300
         tension=20 smooth=0.5
         zcolumn=VALUE
```

where `input` is the name of the vector data file with contours or elevation data points, `elev`, `slope`, `aspect`, `pcurv`, `tcurv`, `mcurv` are the output DEM and local parameters maps including profile, tangential, and mean curvatures, `devi` is the output deviations file that provides deviations of the resulting surface for each given point, `dmin`, `dmax` are the minimum and maximum distance between points (see explanation in the previous section), `tension`, `smooth` are RST function parameters, also explained in the previous section, and `zcolumn` is the name of the column if the elevation is stored as an attribute rather than as a z-coordinate. The resulting maps are shown in Figures 4–8. The resolution of the resulting maps can be set using `g.region` command before the calculation. In our example, we have used 5 m resolution, based on the data point density and size of the features represented by the given contours.

Computation of curvatures from densely sampled or noisy data, such as LiDAR and SRTM, poses a different type of challenge. Without adequate smoothing, the curvatures will reflect the noise rather than the land-surface features. Figures 9 and 10 show profile curvature maps computed from the original SRTM data using the `r.slope.aspect` command described below and re-interpolated with smoothing using `r.resamp.rst` command, respectively.

Quality of interpolation can be assessed using deviations between the interpolated surface and the given data that are stored in a deviation file. These interpolation errors can be evaluated by statistical measures (e.g., root mean squared error, mean absolute error, etc.). The resulting elevation surface is often a compromise between minimisation of predictive and interpolation errors and the application purpose of the DEM. For example, environmental applications usually require smoother surfaces, while technical applications prefer interpolation accuracy.

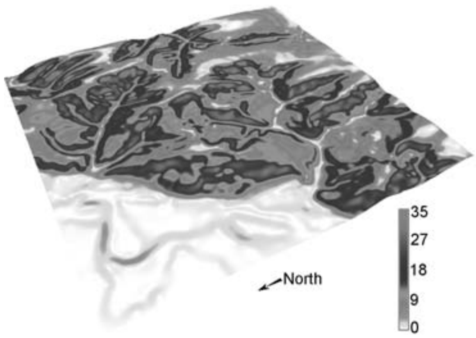

FIGURE 4 Slope steepness [°]. (See page 734 in Colour Plate Section at the back of the book.)

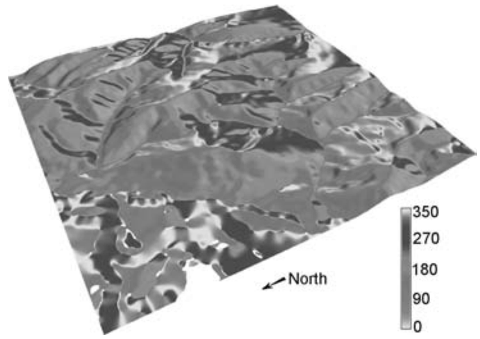

FIGURE 5 Aspect [°]. (See page 734 in Colour Plate Section at the back of the book.)

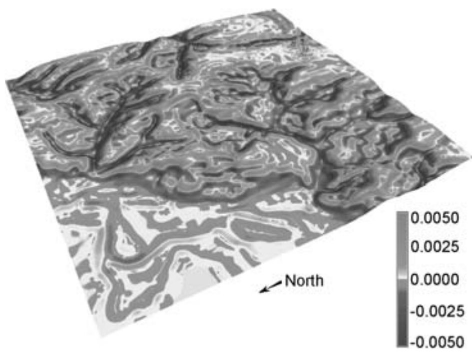

FIGURE 6 Profile curvature [m^{-1}]. (See page 734 in Colour Plate Section at the back of the book.)

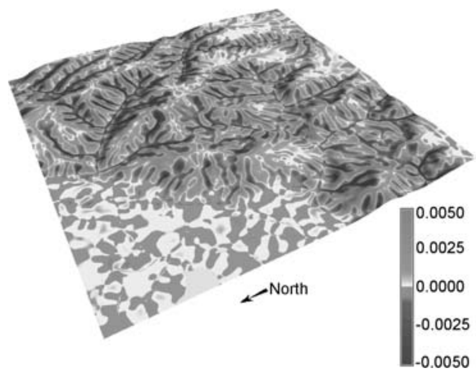

FIGURE 7 Tangential curvature [m^{-1}]. (See page 734 in Colour Plate Section at the back of the book.)

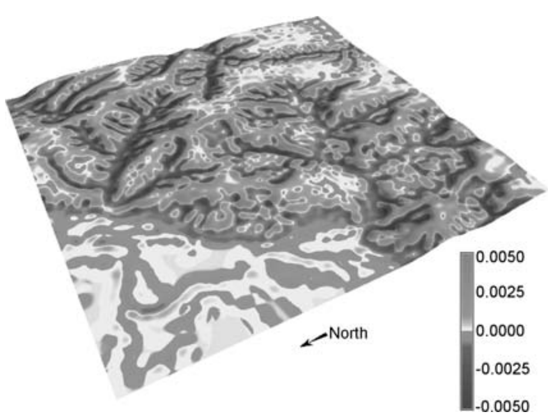

FIGURE 8 Mean curvature [m^{-1}]. (See page 734 in Colour Plate Section at the back of the book.)

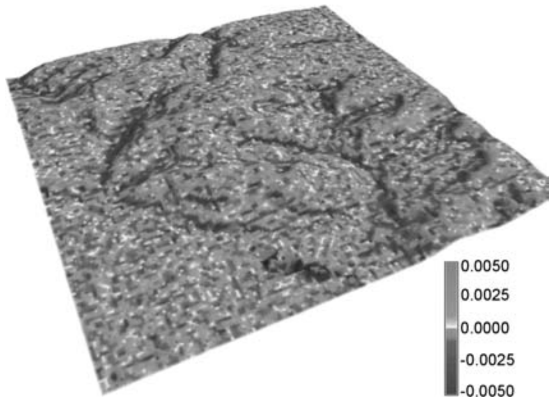

FIGURE 9 Profile curvature [m^{-1}] computed directly from SRTM data using `r.slope.aspect`. (See page 735 in Colour Plate Section at the back of the book.)

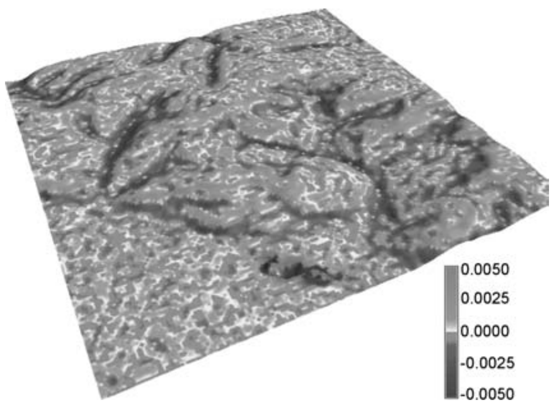

FIGURE 10 Profile curvature [m^{-1}] from smoothed SRTM data using `r.resamp.rst`. (See page 735 in Colour Plate Section at the back of the book.)

If re-interpolation of an existing DEM is not necessary, the local polynomial approximation method implemented in `r.slope.aspect` module can be used to compute local land-surface parameters. Mathematical definitions of the local parameters are identical to the RST modules mentioned above. In `r.slope.aspect` the following second-order polynomial approximation is used:

$$z(x, y) = a_0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot x \cdot y + a_4 \cdot x^2 + a_5 \cdot y^2 \tag{2.1}$$

By fitting this polynomial to 9 grid points (3×3 array), we can derive the coefficients of this polynomial using weighted least squares. First order partial derivatives are derived using Horn's formula (Horn, 1981; Neteler and Mitášová, 2008):

$$f_x = \frac{(z_7 - z_9) + (2z_4 - 2z_6) + (z_1 - z_3)}{8 \cdot \Delta x} \tag{2.2}$$

$$f_y = \frac{(z_7 - z_1) + (2z_8 - 2z_2) + (z_9 - z_3)}{8 \cdot \Delta y} \tag{2.3}$$

and the second order derivatives are as follows (Neteler and Mitášová, 2008):

$$f_{xx} = \frac{z_1 - 2z_2 + z_3 + 4z_4 - 8z_5 + 4z_6 + z_7 - 2z_8 + z_9}{6 \cdot (\Delta x)^2} \tag{2.4}$$

$$f_{yy} = \frac{z_1 + 4z_2 + z_3 - 2z_4 - 8z_5 - 2z_6 + z_7 + 4z_8 + z_9}{6 \cdot (\Delta y)^2} \tag{2.5}$$

$$f_{xy} = \frac{(z_7 - z_9) - (z_1 - z_3)}{4 \cdot \Delta x \Delta y} \tag{2.6}$$

where $z_3 = z_{i+1,j+1}$, $z_5 = z_{i,j}$, $z_7 = z_{i-1,j-1}$ are elevation values at row i column j, Δx is the east-west grid spacing and Δy is the north–south grid spacing (resolution). Computation of a similar set of parameters as in our RST example, but this time from a raster DEM using `r.slope.aspect`, is performed as follows:

```
r.slope.aspect elevation=b_dem5K.z
          slope=b_dem5Kr.s aspect=b_dem5Kr.a
          pcurv=b_dem5Kr.pc tcurv=b_dem5Kr.tc
```

For an additional example see Section 2.4.

2.2 Regional land-surface parameters

Many landscape processes are influenced by land-surface properties and, at the same time, change the land-surface geometry. Mass and energy flows transport water, air, sediment particles, heat, sound, gases and aerosols within and between landscape elements. Mass flows are influenced by the local land-surface parameters, as well as by landscape configuration that reflects broad-scale geometry of the terrain. The magnitude of the transporting agent (e.g. water) affects its carrying capacity or defines the occurrence of specific phenomena such as floods or gullying. It is often related to the spatial extent of the land surface from which the mass is accumulating while moving downslope. Thus, the movement can be traced by flowlines and currents. GRASS has several modules for computing regional parameters that can be used for analysis of mass flows over the land surface.

2.3 Flow parameters and watersheds

Topography has a profound influence on mass and energy fluxes in the landscape and is often a major factor in many geospatial models and applications. GRASS provides many tools for watershed and water flow analysis. Flow parameters are derived by flow tracing algorithms that approximate the route of water or other liquid over the surface represented by a DEM. Flow routing is based on

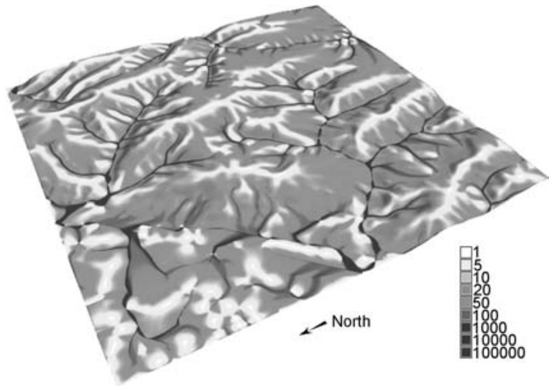

FIGURE 11 Flow accumulation [-] generated by `r.terraflow`. (See page 735 in Colour Plate Section at the back of the book.)

flow-lines — curved lines of descent perpendicular to contours in the direction indicated by aspect. The following basic flow parameters can be computed using GRASS (Neteler and Mitášová, 2008):

- *flow accumulation*;
- *upslope contributing area*;
- *stream network*;
- *watershed (basin) boundaries*;
- *flowpath length*.

Numerous algorithms have been developed for flow routing, based on the approach for estimation of the steepest slope direction and water movement to the downslope cells. In GRASS, the following algorithms have been implemented:

- single flow direction to eight neighbouring cells (SFD, D8) moves flow into a single downslope cell (`r.watershed`);
- single flow to any direction ($D\infty$) or vector-grid approach (`r.flow`);
- multiple flow direction (MFD) to two or more downslope directions (`r.terraflow`, `r.topmodel`);
- 2D water movement simulation based on overland flow differential equations (`r.sim.water`).

The single flow direction approach has the disadvantage that it discretises the flow into only one of eight possible directions. Therefore it produces artificial straight-line patterns especially in areas of flat terrain and on convex landforms with dispersed water flow. SFD is useful for stream network extraction where a single cell representation is needed. Multiple flow routing has the disadvantage that the flow from a cell is dispersed to all neighbours of lower elevation, resulting in a more diffuse flow of water, especially in valleys where concentrated water flow occurs. However, the resulting water flow accumulation surface is smoother, thus more appropriate for further differential geometry analysis (Figure 11).

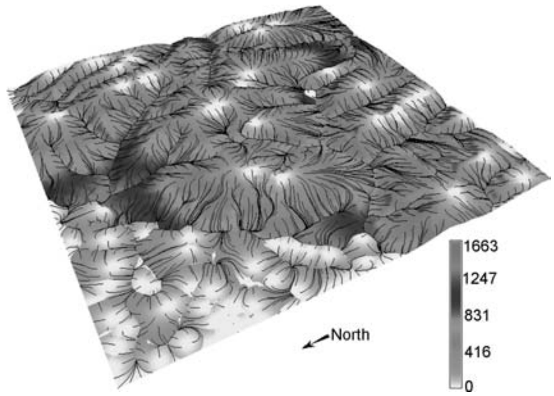

FIGURE 12 Flowpath lengths [m] and flowlines generated by `r.flow`. (See page 736 in Colour Plate Section at the back of the book.)

Many flow routing algorithms are negatively influenced by DEMs of poor quality. Numerous local depressions in valleys or flat areas interrupt the flow-tracing algorithms and create incorrect patterns of flow accumulation (upslope contributing areas), stream networks, flowpath lengths and of other flow parameters. Modules `r.fill.dir` and `r.carve` can be used to remove depressions (sinks) and lakes on DEMs. However, these depression-filling algorithms also introduce positional errors, create artificial features (e.g. flats leading to parallel streams) so that the flow parameters then do not fit with values of other land-surface parameters computed from the original DEM. In GRASS, however, `r.watershed` does not require prior filling of depressions to produce continuous flow accumulation maps, stream networks and other hydrologic parameters, as it uses the least-cost search algorithm to traverse the elevation surface to the outlet. In applications with a new type of DEMs, for example, based on LiDAR or radar-based surveys, this often leads to more accurate results compared to the traditional methods of depression removal (Kinner et al., 2005).

The choice of the module and operations depends on the application. For example, `r.flow` stops flow tracing on flat areas and depressions, so it is more suitable for estimation of flow on hillslopes, smaller watersheds, or DEMs without pits or flat areas. Flowlines, flow accumulation and flowpath lengths for hillslopes in the test region can be computed as follows (Figure 12):

```
r.flow elev=b_dem5K.z aspin=b_dem5K.a skip=15
       flout=b_dem5K_fl dsout=b_dem5K.dd
r.flow -u elev=b_dem5K.z aspin=b_dem5K.a
       lgout=b_dem5K.ul
```

Stream networks and watershed boundaries can be extracted more effectively with `r.watershed`:

```
r.watershed b_dem5K.z accum=b_dem5K.acc
         thresh=10000 basin=b_dem5K.bas
         stream=b_dem5K.st drainage=b_dem5K.dir
```

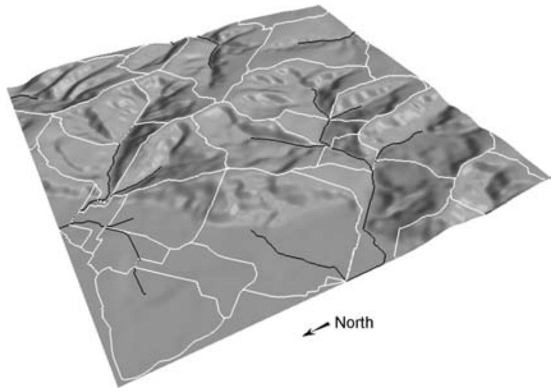

FIGURE 13 Stream network (black lines) and watershed boundaries (white lines) extracted using `r.watershed`.

Note that a more detailed stream network can be extracted from the accumulation map using `r.mapcalc` and a lower threshold than the one defined in the above command. It is often useful to convert the results of `r.watershed` into a vectorised stream network and watershed boundaries (Figure 13). First, the raster representation of the stream network is thinned into a single cell width using `r.thin` and then it is converted to vector lines using `r.to.vect`; watershed areas (basins) can be converted directly, without thinning:

```
r.thin b_dem5K.st out=b_dem5K.st.thin
r.to.vect -s b_dem5K.st.thin out=b_dem5K_st
r.to.vect -s b_dem5K.bas out=b_dem5K_bas
              feature=area
d.vect b_dem5K_st col=blue
d.vect b_dem5K_bas type=boundary
```

If a watershed (contributing area) draining to a given outlet `n,e` is needed, the flow direction map `b_dem5K.dir` generated by `r.watershed` can be used as input for `r.water.outlet`, for example:

```
r.water.outlet drainage=b_dem5K.dir
         basin=b_dem5K.basne easting=6552738
         northing=5071763
```

Specifically designed for handling of very large DEMs (thousands of rows and columns) is `r.terraflow` (Arge et al., 2003, Figure 11); here we show its application to computation of an MFD-flow map and the topographic wetness index:

```
r.terraflow elev=b_dem5K.z filled=b_dem5Kt.fil
           dir=b_dem5Kt.dir swat=b_dem5Kt.swat
           acc=b_dem5Kt.acc tci=b_dem5Kt.tci
```

A wide range of additional parameters can be computed using JGRASS (http://www.jgrass.org/); a Java based GIS built on top of GRASS that includes tools for the HORTON machine (Rigon et al., 2006).

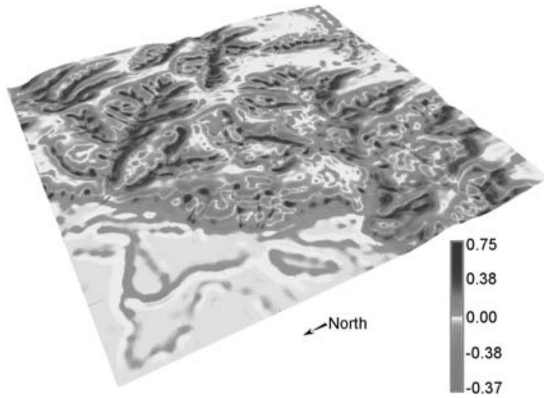

FIGURE 14 Topographic soil erosion index [-]. (See page 736 in Colour Plate Section at the back of the book.)

2.4 Land-surface analysis for modelling

Land-surface parameters play an important role in representing landscape processes. For example, Mitas and Mitášová (1998) have explicitly shown how profile and tangential curvatures influence soil erosion. However, there are applications that may require user-defined or otherwise specific land-surface parameters, or just partial derivatives. To compute specific land-surface parameters not included as outputs of existing modules, such as directional derivatives or flow divergence, a map algebra module `r.mapcalc` and partial derivatives computed by the RST modules or `r.slope.aspect` can be used. The following example demonstrates shell-scripting, map algebra and various land-surface parameters for calculating a topographic index of soil erosion and deposition using the modified LS factor for the Universal Soil Loss Equation; see also the Unit Stream Power-based Erosion Deposition (USPED) model in Mitášová and Mitas (2001). This topographic potential is expressed by a dimensionless index (Figure 14) calculated as a divergence of sediment flow transport capacity:

```
# modified LS factor
# using upslope contributing areas
r.mapcalc "flowtopo = 1.4 *
     exp((upslope_area/cell_size)/22.13,0.4)
     * exp(sin(slope)/0.0896,1.3)"

# sediment transport in x and y directions
r.mapcalc "flowtopo.dx=flowtopo * cos(aspect)"
r.mapcalc "flowtopo.dy=flowtopo * sin(aspect)"

# partial derivatives for sediment transport
r.slope.aspect elev=flowtopo.dx dx=qs.dx
r.slope.aspect elev=flowtopo.dy dy=qs.dy

# topographic potential for
# net erosion and deposition
```

```
r.mapcalc "topoindex = qs.dx + qs.dy"
```

Because the distribution of topographic potential values is skewed, it is helpful to replace a default colour table with a user-defined colour table by the `r.colors` command. This will reveal spatial differences in the values when displaying the resulting map by the `nviz` or `d.rast` commands:

```
r.colors topoindex col=rules
> -1500000 100 0 100
> -10 magenta
> -0.5 red
> -0.1 orange
> -0.01 yellow
> 0 200 255 200
> 0.01 cyan
> 0.1 aqua
> 0.5 blue
> 10 0 0 100
> 1500000 black
> end
```

The first value in every row represents a topoindex value to which a specific colour is attributed either by colour name, or by an RGB triplet. Yellow through red hues represent erosion while blue shades are used for deposition.

Flow parameters represent the potential of relief to generate overland water flow. These parameters do not take into account infiltration or land cover. Therefore, topographical indexes derived from these parameters often represent a steady-state situation or maximal values of overland flow, assuming uniform soil and land cover properties. At landscape scale, a uniform steady-state overland flow is a rare phenomenon occurring only during extreme rainfall events. Therefore resulting patterns of net erosion and deposition based on upslope contributing areas may contradict field observations.

Water and sediment flows are spatial and dynamic phenomena described by complex differential equations that are usually solved by approximation methods. The recently developed `r.sim` group of modules uses the Monte Carlo path sampling method to simulate spatial, dynamic landscape processes. The module `r.sim.water` simulates overland water flow (Figure 15), while `r.sim.sediment` produces sediment flow and erosion/deposition maps based on the Water Erosion Prediction Project (WEPP) theory (Mitas and Mitášová, 1998). The following example shows the application of `r.sim.water` module for the Baranja Hill data set using derived land-surface parameters and uniform *ad hoc* rainfall, soil and land cover properties:

```
r.sim.water -t elevin=b_dem5K.z
dxin=b_dem5K.dx dyin=b_dem5K.dy
rain=b_dem5K.rain infil=b_dem5K.infil
manin=b_dem5K.manning disch=b_dem5K.disch
nwalk=1000000 niter=2400 outiter=200
```

The output of these modules can be in the form of a time-series of maps showing evolution of the modelled phenomenon (available at geomorphometry.org).

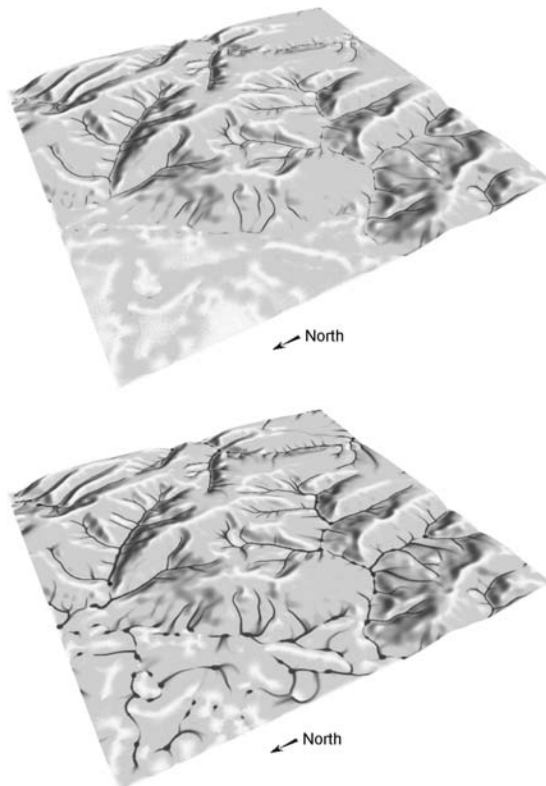

FIGURE 15 Overland water flow simulated by `r.sim.water` after 200 (above) and 2400 (below) seconds.

2.5 Landforms

The GRASS module `r.param.scale` extracts basic land-surface features from a DEM, such as peaks, ridges, passes, channels, pits and plains. This module is based on the work by Wood (1996). It uses a multi-scale approach by fitting a bivariate quadratic polynomial to a given window size using least squares. This module is a predecessor to the system described in Chapter 14 (e.g. Figure 11). In the following example (Figure 16), main land-surface features were identified using a 15×15 processing window:

```
r.param.scale in=b_dem5K.z out=b_dem5K.param
              param=feature size=15
```

2.6 Ray-tracing parameters

Solar radiation influences many landscape processes and is a source of renewable energy of interest to many researchers, energy companies, governments and consumers. GRASS provides two modules related to solar radiation: `r.sunmask`

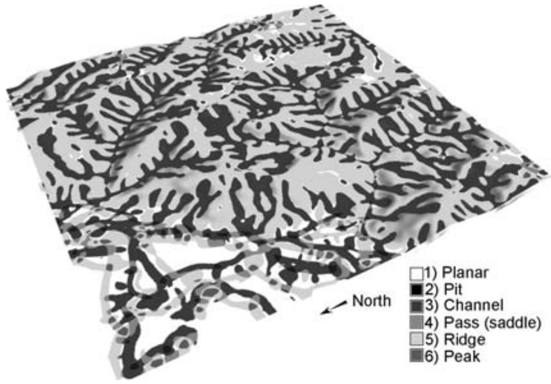

FIGURE 16 Basic land-surface features extracted using `r.param.scale`. (See page 736 in Colour Plate Section at the back of the book.)

calculates a Sun position and shadows map for specified time and Earth position using the SOLPOS2 algorithm from National Renewable Energy Laboratory, and `r.sun` calculates all three components of solar irradiance/radiation (beam, diffuse and reflected) for clear-skies as well as overcast conditions (Šúri and Hofierka, 2004). The clear-sky solar radiation model is based on the work undertaken for development of the European Solar Radiation Atlas (Scharmer and Greif, 2000; Rigollier et al., 2000). The model works in two modes. The irradiance mode is selected by setting a local time parameter; the output values are in W/m^2. By omitting the time parameter, the radiation model is selected; output values are in Wh/m^2.

The model requires only a few mandatory input parameters such as elevation above sea level, slope and aspect of the terrain, day number and, optionally, a local solar time. The other input parameters are either internally computed (solar declination) or the values can be overridden by explicitly defined settings to fit specific user needs: Linke atmospheric turbidity, ground albedo, beam and diffuse components of clear-sky index, time step used for calculation of all-day radiation from sunrise to sunset. Overcast irradiance/radiation are calculated from clear-sky raster maps by the application of a factor parameterising the attenuation of cloud cover (clear-sky index). The clear-sky global solar radiation for Baranja Hill data set, March 21 (spring equinox) has been calculated using `r.sun` as a sum of beam, diffuse and reflected radiation. The shadowing effects of relief were taken into account (Figure 17).

In practical applications related to evaluation of available solar radiation within a specific period of day or year we can use a shell script and the `r.mapcalc` command to compute a sum of available radiation values. Viewshed analysis can be performed using `r.los` that generates a raster map output in which the cells that are visible from a user-specified observer location are marked with integer values that represent the vertical angle (in degrees) required to see those cells (viewshed). A map showing visible areas (in blue) from the position of a man

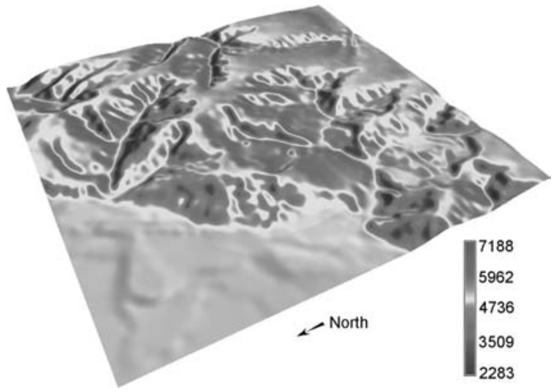

FIGURE 17 Global solar radiation for spring equinox [Wh/m^2]. (See page 737 in Colour Plate Section at the back of the book.)

standing on the hill crest depicted by a black dot in Figure 18 can be computed as follows:

```
r.los b_dem5K.z out=b_dem5K.los
      coor=6553202,5071538
```

An improved viewshed analysis program is available as GRASS extension.[2] Shaded relief maps enhance the perception of terrain represented by a DEM. In GRASS, they are generated using the `r.shaded.relief` module with parameters defining the sun position (sun altitude and azimuth) and vertical scaling

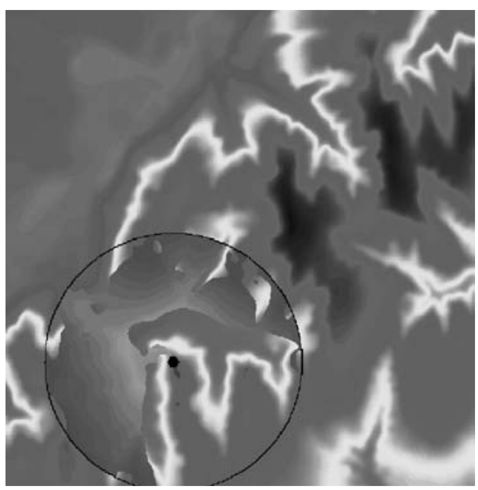

FIGURE 18 Visibility analysis using `r.los`. (See page 737 in Colour Plate Section at the back of the book.)

2 http://www.uni-kiel.de/ufg/ufg_BerDucke.htm.

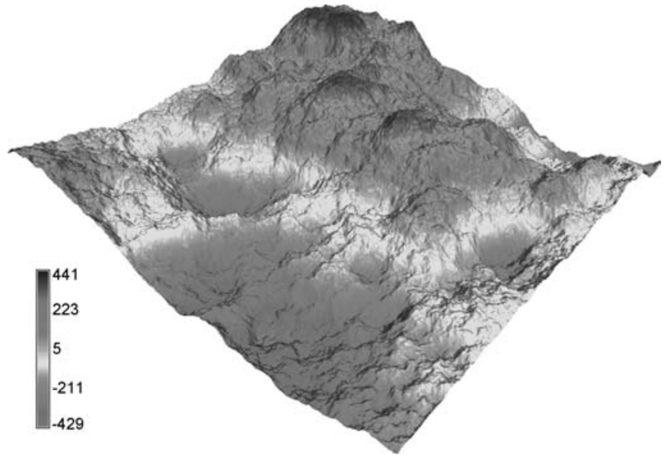

FIGURE 19 Random fractal surface generated by `r.surf.fractal`. (See page 738 in Colour Plate Section at the back of the book.)

(z-exaggeration). This shaded map can be used to transform colours of other thematic map using the IHS colour model. The resulting shaded, coloured map, displayed by command `d.his` provides enhanced perception of terrain and better orientation especially in hilly areas (see example in the displaying DEMs section).

2.7 Fractal surfaces

The concept of fractals has attracted the attention of scientists in many fields, including geomorphometry. According to many studies, most real land surfaces have a fractal dimension in the range of 2.2–2.6. However, Wood (1996) notes that landscapes usually do not possess a single fractal dimension, but a variety of values that change with scale. The concept of fractal surfaces and fractal dimension can be employed to generate synthetic, natural-looking surfaces with controllable topographic variation. There are numerous methods of generating fractal surfaces, but the one adopted in `r.surf.fractal` module uses the spectral synthesis approach described by Saupe (1988).

This technique involves selecting scaled (Gaussian) random Fourier coefficients and performing the inverse Fourier transform. It has the advantage over the more common midpoint displacement methods which produce characteristic artifacts at distances 2^n units away from a local origin (Voss, 1988). Wood (1996) has modified this technique so that multiple surfaces may be realised with only selected Fourier coefficients in the form of intermediate layers showing the buildup of different spectral coefficients. The result is that the scale of fractal behaviour may be controlled as well as the fractal dimension itself. In the example for the Baranja region (Figure 19) we have used the `r.surf.fractal` module with the fractal dimension set to 2.05:

```
r.surf.fractal out=b.fractal d=2.05
```

Other fractal-related modules are `r.surf.gauss` and `r.surf.random`. The module `r.surf.gauss` generates a fractal surface based on a Gaussian random number generator whose mean and standard deviation can be set by the user. The module `r.surf.random` uses a different type of random number generator and uniform random deviates whose range can be expressed by the user.

2.8 Summary parameters and profiles

GRASS provides various tools for querying and summarising maps of land-surface parameters. For example, the module `r.report` can be used to create a frequency distribution of map values in the form of a table containing category numbers, labels and (optionally) area sizes in units selected by a user. The command `r.stats` calculates the area present in each of the map categories. Alternatively, `d.histogram` can be used to visualise a distribution of the values in the form of a bar or pie chart. Polar diagrams can be used for displaying distributions of aspect values by the `d.polar` module. If the polar diagram does not reach the outer circle, no data (NULL) cells were found in the map. The vector in the diagram indicates the prevalent direction and vector length the share of this direction in the frequency distribution of aspect values.

The aspect map for the Baranja Hill DEM with a spatial resolution of 25 m and derived from the 1:5000 contours [Figure 20(a)] shows dominant spikes in the polar diagram [Figure 20(d)] indicating a suboptimal land-surface representation in DEM25m. The aspect map of the DEM25-SRTM [Figure 20(c)] does not show dominant spikes but mostly regular spikes representing relatively homogeneous noise typical for RADAR data [Figure 20(d)]. Finally, the aspect computed simultaneously with DEM interpolation from the Baranja Hill contour lines using `v.surf.rst` [Figure 20(b)] is relatively smooth and does not show any significant spikes [Figure 20(d)]. Lengths of the average direction vectors in the diagram are very short which indicates that DEMs for this region show no prevalent aspect direction.

The area of a surface represented by a raster map is provided by `r.surf.area` which calculates both the area of the horizontal plane for the given region and an area of the 3D surface estimated as a sum of triangle areas created by splitting each rectangular cell by a diagonal. More complex analysis is available in `r.univar` and `r.statistics`. The `r.univar` module calculates univariate statistics that includes the number of counted cells, minimum and maximum cell values, arithmetic mean, variance, standard deviation and coefficient of variation. The `r.statistics` module also calculates mode, median, average deviation, skewness and kurtosis. Using the `r.neighbors` module, a *local* statistics based on the values of neighbouring cells defined by a window size around the central cell can be computed. Available statistics include minimum, maximum, average, mode, median, standard deviation, sum, variance, diversity and inter-dispersion. Sophisticated statistics and spatial analysis are available via GRASS interface with the R statistical data analysis language (http://cran.r-project.org/).

Land-surface analysis often requires querying map values at a specific location. This can be done in GRASS either interactively with the mouse, or by a com-

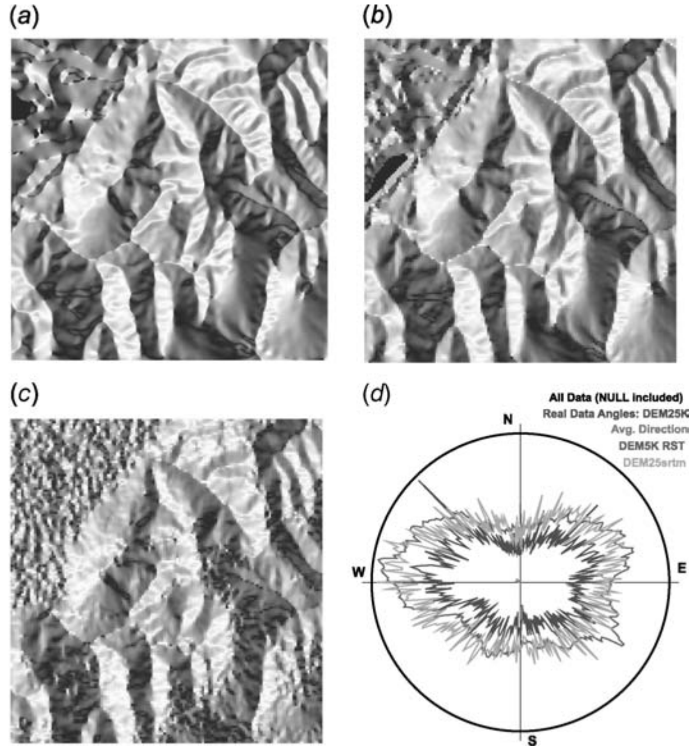

FIGURE 20 Baranja Hill aspect maps: (a) DEM25, (b) DEM5K (generated by `v.surf.rst`), (c) DEM25-SRTM, and (d) a combined polar diagram of all aspect maps from `d.polar`. (See page 738 in Colour Plate Section at the back of the book.)

mand with coordinates defining the location. The simplest command for interactive querying by mouse is `d.what.rast`. To generate profiles, a user can run `d.profile`. It allows one to interactively draw profiles over the terrain by mouse within the GRASS monitor.

Non-interactive query can be performed at specific points defined by coordinates (`r.what`) or along the user-defined profile (`r.profile` and `r.transect`). Similar query commands are available for vector maps as well.

2.9 Volume parameters

Land surface is a 2-dimensional contact between different landscape components (atmosphere vs. lithosphere, or hydrosphere vs. lithosphere). As such, it often represents the surface of a 3D object. To compute the volume of the object, the summary parameter `r.volume` can be used, for example, to estimate the amount of earth that must be excavated for a construction project.

Many landscape phenomena can be investigated using differential geometry tools extended to three dimensions (Hofierka and Zlocha, 1993). GRASS provides several tools for 3-dimensional (volume) modelling. For example, tri-variate Reg-

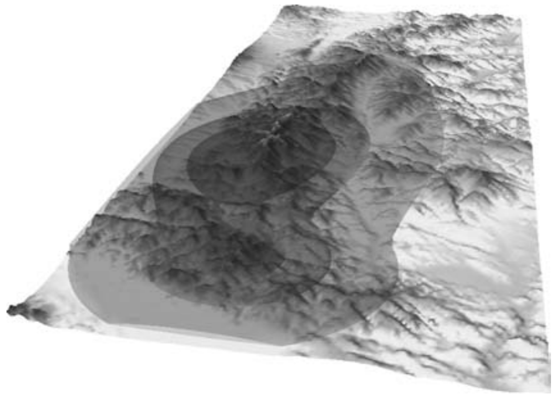

FIGURE 21 Volume interpolation and isosurface visualisation of precipitation (isosurfaces of 1100, 1200, 1250 mm/year are shown) using `v.vol.rst`. (See page 739 in Colour Plate Section at the back of the book.)

ularised Spline with Tension is implemented in `v.vol.rst` for spatial interpolation of volume data. `v.vol.rst` has similar properties and parameters as the bi-variate version of RST, so the principles described in the Introduction section are applicable here as well. Similarly to the bi-variate version, tri-variate RST can compute a number of geometric parameters related to the gradient and curvatures of the volume model: magnitude and direction of gradient, directional change of gradient, *Gauss–Kronecker* and mean curvatures. Mathematical definitions and explanation of volume parameters can be found in Hofierka and Zlocha (1993) and Neteler and Mitášová (2008).

Moreover, tri-variate interpolation can be helpful in spatial characterisation of natural phenomena influenced by land surface. For example, Hofierka et al. (2002) present an application of tri-variate RST in precipitation modelling. Elevation, aspect, slope, or other land-surface parameters can be incorporated in the tri-variate interpolation as a third variable. The approach requires 3D data (x, y, z, w) and a raster DEM. The phenomenon is modelled by tri-variate interpolation. Then, phenomenon values on the land surface are computed by intersection of the volume model with the land surface represented by a DEM. The volumetric visualisation of the precipitation volume model using `nviz` is presented in Figure 21.

3. LIMITATIONS OF GRASS

Although GRASS has rather comprehensive geomorphometry tools it is by no means complete. For example, support for TIN-based land-surface modelling and analysis, often used in engineering applications, is very limited. Also, modelling of terrain with faults and breaklines, although possible, is rather cumbersome as it requires additional pre- and post-processing. Some help is available in `r.surf.nnbathy`, which employs a natural neighbour interpolation library (http://www.marine.csiro.au/~sakov/) and supports interpolation with breaklines. It is provided as an add-on module at GRASS Wiki site (http://grass.osgeo.org/wiki/). The error of prediction can be analysed using a simple comparison of estimated and true values or using more sophisticated cross-validation. GRASS is currently evolving rather rapidly based on the needs of its developers, therefore new capabilities not included here could have happen during the production of this book. The most recent capabilities can be checked at the official GRASS web site.

4. SUMMARY POINTS AND FUTURE DIRECTION

GRASS is a mature, fully-featured open-source GIS capable of a broad spectrum of spatial calculations in geomorphometry. The ANSI C source code provides a comprehensive suite of modules and UNIX-shell scripts to manipulate DEMs, extract a variety of land-surface parameters and objects, and analyse hydrogeomorphological phenomena in both 2D and 3D. Surface-form data can be imported as grid DEMs, digitised contours, or as scattered point-measurements of elevation. Considerable automation has been built into the system, which features a graphical user interface and is readily available through a web-based infrastructure. The 6.2 version of GRASS illustrated in this chapter is available for all commonly used operating systems.

Advances in mapping technologies, especially the rapid evolution of airborne and ground-based laser scanning as well as satellite and airborne radar interferometry are bringing significant changes to geomorphic analysis. The point densities now exceed the level of detail required for most applications and DEMs with resolutions of 3 m and better are becoming common even for large areas. The high mapping efficiency makes repeated mapping at relatively short time intervals feasible, resulting in multi-temporal DEMs. These developments require new concepts and approaches in geomorphometry. In response, GRASS modules are being further enhanced to accommodate very large data sets produced by the new mapping technologies; new tools are added, for example, for efficient handling of very dense elevation or bathymetry data, hierarchical watershed analysis and quantification of land-surface change.

IMPORTANT SOURCES

http://grass.osgeo.org — The GRASS website.
http://www.jgrass.org — JGRASS.
http://skagit.meas.ncsu.edu/~helena/gmslab/viz/sinter.html — Multidimensional Spatial Interpolation in GRASS GIS.
http://skagit.meas.ncsu.edu/~helena/gmslab/viz/erosion.html — Land-surface analysis and applications.
http://skagit.meas.ncsu.edu/~helena/publwork/Gisc00/astart.html — Path sampling modelling.
http://www.cs.duke.edu/geo*/terraflow/ — Terraflow.
http://re.jrc.cec.eu.int/pvgis/ — PVGIS and solar radiation modelling using GIS.