Chapter 16

Environmental Modeling With PCRaster

PETER A. BURROUGH AND DEREK KARSSENBERG
UTRECHT CENTRE FOR ENVIRONMENT AND LANDSCAPE DYNAMICS, FACULTY OF GEOSCIENCES,
UTRECHT UNIVERSITY UTRECHT, THE NETHERLANDS.

WILLEM VAN DEURSEN
PCRASTER ENVIRONMENTAL SOFTWARE UTRECHT, THE NETHERLANDS

ABSTRACT

MODERN GIS ARE WELL-DEVELOPED for dealing with information on the spatial distribution of static objects and continuous fields but most lack the ability to deal with the change of patterns or collections of objects over time. Consequently, they may be poorly equipped to deal with dynamic space–time processes such as flooding, erosion, plant growth, tectonics, or diffusion of plants and animals. Although raster GIS are often used to link stand-alone dynamic models of spatial change to geographic databases, few incorporate the process modeling in the GIS itself.

Developments during the past ten years show that GIS toolkits can be developed to address many of the generic aspects of space–time modeling. Temporal changes in attributes of cell values can be computed for single cells or locations, for neighborhoods of varying size and shape, for transport of materials over a network, or for action at a distance. Extreme flexibility in space–time modeling is given by writing the basic operations in easy-to-understand commands. Once the model has been set up and run, the changes can be visualized as a film, providing the user with extra understanding of the processes being modeled.

This chapter discusses the basic ideas behind space–time modeling using embedded GIS generic commands, in particular the PCRaster toolkit. It starts with a simple model of rainfall–runoff interaction over a small catchment for which each attribute of the model is given by a single raster layer. The ideas are extended to large-scale landscape change ranging from landscape building (a volcano) to landscape degradation (gulley-forming and erosion). It also shows how the generic toolkit may itself be used as a computer modeling language and how simple and complex geographic models can be created. Finally, the chapter introduces new developments in PCRaster that provide special facilities for modeling the sedimentation and erosion of material on three-dimensional stacks of cells.

INTRODUCTION: A GENERIC APPROACH TO ENVIRONMENTAL MODELING

Modern geographical information systems (GIS) provide a consistent and mature technology for storing, organizing, retrieving, and modifying information on the spatial distribution of plants, natural resources, forest areas, land use, land parcels, utilities, and many other natural and anthropogenic features (Burrough and McDonnell 1998). Although GIS offers a well-established and well-defined framework for the analysis of the spatial component of geographic problems, support for the analysis of the temporal or dynamic component of these issues is largely lacking. This chapter deals with the development and applications of a generic approach for such a dynamic framework.

With respect to their representation of space, GIS are often divided into raster GIS (based on a tesselation of space) and vector GIS (detailed geographic representation of boundaries of well-defined objects). The raster representation is often ideal for spatial analysis because it provides a common and consistent basis for computing new attributes from existing data. This aspect of raster spatial data analysis was developed by Dana Tomlin (1990) in his Map Analysis Package, which forms the basis of the ArcView Spatial Analyst. The ideas behind the methods of Cartographic Algebra that were embodied in the Map Analysis Package are based on the view that space can be divided into regular, orthogonal cells or grids which can be subjected to well-defined, generic, and primitive transformations. As in ordinary algebra, these transformations can be linked together in sequence to compute more complex functions. For example, using a gridded digital elevation model, it is possible to compute a slope hazard map by computing the slope of the land at every cell as a function of the elevations of the cells surrounding it and then to select and display those cells for which the slope exceeds a given value.

Raster GIS are often used to supply data to and display results from stand-alone models that run on databases external to the GIS. This has often been deemed necessary for computer models of processes that involve the use or input of time-related data. Increasingly, however, people have perceived the need for GIS toolkits that are equipped with generic suites of algorithms for spatial and temporal analysis and modeling. These GIS toolkits (known variously as Map Algebra or Map Calculus) provide a computational scripting language based on simple, easy-to-understand mathematical operations for which various classes of primitive operation can be distinguished (Burrough and McDonnell 1998, Eastman 2003, Tomlin 1990, Pullar 2003, Takeyama and Couclelis 1997). The main classes of spatial operations found in these GIS toolkits are:

i. Operations on single cells or locations (point operations or local functions).

ii. Operations in which a value is computed for a cell that lies at the center of a surrounding circular or rectangular block of cells (neighborhood operations or focal functions).

iii. Operations that compute the flux of material over a topologically linked route (network functions; e.g., water over a river network).

iv. Operations that involve action at a distance (surface functions; e.g., a viewshed or area that can be seen from a given point or the amount of solar radiation falling on a surface at a given time of day and year).

v. Operations calculating statistics of an attribute for each area (zonal functions; e.g., the average soil contamination for each land-use type).

From space to time: the basic principles of dynamic spatial modeling

In many environmental problems (such as rainfall-runoff interactions, soil erosion, dispersion of plants and animals, water quality variations, land cover, and landform change) where patterns change in response to external forces, it is essential to include the time component in the analysis. This gives rise to a dynamic spatial model, which is defined as a mathematical representation of a real-world process in which the state of a geographic field or object changes in response to variations in the driving forces. Any system for modeling space–time processes must include procedures for discretizing space–time and for the computation of new attributes for the spatial and temporal units in response to the driving forces.

Like space, time can be divided in different ways. Computationally, the easiest way is to discretize time into equal steps, and that is the procedure followed in PCRaster. When discretizing both time and space, the choice of the size of the interval or cell is extremely important because variations that occur within the dimensions of the cell will not be registered by either the data or the process. Therefore, it is essential to choose the correct spatial and temporal resolution.

Dynamic modeling involves computing the temporal change of the state of an entity in response to information from driving forces (or inputs) and the processes that act in the system being modeled. In a dynamic modeling program, temporal change is represented by discrete time steps, applying the following equation for each time step:

$$\mathbf{z}_{t+1} = f(\mathbf{z}_t, \mathbf{i}_t) \qquad \text{for each t} \qquad (1)$$

The equation shows that the change in the state (**z**) of an entity from a specific time step *t* to the next time step *t*+1 is a function of the processes (*f*) in the system and the driving forces (**i**). The state of each entity ("object" or grid cell) is described by three kinds of information, namely what is it? where is it? and what is its relation to other entities? The nature of an entity is given by its attributes, its whereabouts by its geographic location or coordinates, and the spatial relations between different entities in terms of proximity and connectivity (topology).

In a dynamic model, the processes (f in Equation 1) in the system causing changes in z from each time step t to *t*+1 are represented in an iterative modeling program using standard spatial operations, combining operations from the five standard groups of spatial operations described in the introduction. Note that this set of operations is the same for each time step, since in most cases it can be assumed that the kind of processes occurring in the system does not change through time. These mathematical operations are described in detail by Burrough and McDonnell (1998) and are to be found in many standard GIS. A dynamic GIS expands the capabilities of these operations by allowing them to be placed in an interactive loop that represents the temporal behavior of the processes f.

In addition to these standard groups of spatial operations to represent f, additional operations are provided in dynamic modeling to represent the driving forces or inputs i. These timeinput operations are used to read driving forces as inputs to the model, for each time step and for each grid cell. In most cases, the driving forces are derived from time series of observations sampled at points or over areas. Dynamic modeling also requires efficient data storage and retrieval to access and use intermediate results and to report them to file. These files can be displayed as time series plots and animated 2-D maps or 2.5-D drapes to provide the user with dynamic visual output. In addition, the dynamic modeling of surface changes (transport of material from place to place) may not only require all of the usual raster GIS functionality but also the specific ability to derive surface topology and use that information to transfer data from cell to cell.

A TOOLKIT FOR DYNAMIC SPATIAL MODELING—PCRASTER

The PCRaster modeling language

Many scientists have approached the problem of modeling dynamic aspects of environmental processes by writing individual models using standard languages such as FORTRAN or C++ which could be interfaced with GIS. These models are often difficult to maintain and modify, particularly if the original author is no longer working in the team.

PCRaster (Van Deursen 1995, PCRaster 2004, Wesseling et al. 1996) was the first widely available raster GIS to incorporate a dynamic, generic modeling tool that takes Map Algebra beyond the static timeless spatial analysis model. This has been made possible by the development of a modeling language with a large number of basic analysis operations that can be linked in programs (i.e., scripts). Many of these operations extend existing Map Algebra concepts with the notion and control of time.

PCRaster is a dynamic modeling language that fills the gap between standard non-dynamic or non-temporal commercial GIS and the off-line dynamic model or the once-off program. It operates in the raster domain, providing a large selection (>150) of standardized, generic operations for spatial and temporal analysis, including a full set of mathematical tools for computing new attributes from the original attributes of each cell. It also includes a wide range of operations for modeling spatial dispersion (neighborhood interactions) and directed transport over topological networks. These operations can be driven by data from 1-D or 2-D time series to provide interactive dynamic models of spatial and temporal processes with feedback loops. Visual output of results is in the form of 1-D graphs or 2-D (and 2.5D) stacks of maps.

A model written in PCRaster should adhere to the four precepts of good modeling given by Casti (1997), namely simplicity, clarity, freedom from bias, and tractability. Figure 1 provides a conceptual overview of the structure of a PCRaster program, and further details are given in Box 1. The dynamic database of time series and stacks of grid maps can easily be obtained from conventional GIS such as ArcView, remote sensing imagery, or interpolation.

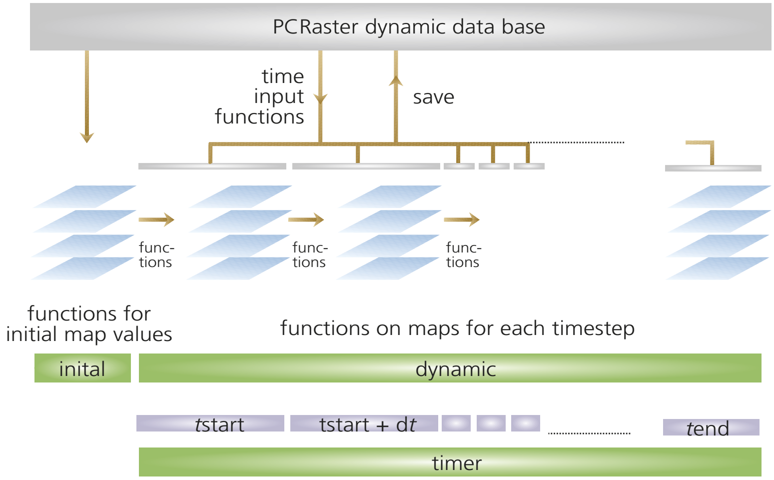


Figure 1. Conceptual overview of a PCRaster program.

A PCRaster program has 5 main sections called binding, areamap, timer, initial, and dynamic.

The binding section links the external file names or parameter values to the internal variable names used in the program; by simply changing file names the same model can be run with different data.

The areamap section defines the geographic extents of the gridded input files that will be addressed by the model—the default is the whole area.

The timer section specifies and controls the number of iterations for the dynamic section of the program.

The initial section defines the starting values of all attributes, either by directly reading from file or by creating derived attributes (e.g., the slope of a digital elevation surface) that will be used only once in the program. All of the allowed mathematical operators can be used to prepare the data in the initial section.

The dynamic section contains the code for all the mathematical operations for one cycle of the model. Intermediate results may be saved to time series or to stacks of gridded attribute files. The output of one cycle forms the input for the next. For example, when modeling erosion and sedimentation, each new sedimentary layer computed can be stored as a new data file so the whole set of maps forms a 3D sediment packet.

Box 1. The structure of a PCRaster program.

Once the model has been run, the results can be displayed as static or dynamic 2D or 3D (2.5D) displays; in the latter case. the display resembles a movie. This enables the user to see exactly how the dynamic model has created the patterns and how these patterns change with time. By changing the model parameters and re-running the program, the effect of changing the value of single parameters, or combinations of parameters, on the results can easily be seen and evaluated, as is demonstrated in the remainder of this chapter.

3. A SIMPLE EXAMPLE OF DYNAMIC MODELING WITH PCRASTER

This example illustrates the use of PCRaster for modeling surface water runoff. We have deliberately chosen a naïve model so that the functioning of PCRaster can be more easily understood. This example models the inputs of rain to a surface and its division into infiltration into the soil and overland flow. Rainfall, recorded at three rain gauges, is extrapolated to Thiessen polygons to provide water inputs to each cell. If the infiltration of water is too great, there is no runoff; if insufficient, then overland flow is generated. The topology of flow is computed to yield the directions for surface drainage. In the dynamic (iterative) part of the model, incident rainfall is partitioned into infiltration and overland flow, depending on the balance between inputs, infiltration, and outputs per cell. Finally, the state of the cells is recorded (report operation) every time step to provide a means to display the results of the model as a 2.5D + time visualization.

Box 2 contains the model script built on the structure given in Box 1. The first section is the binding, which links the internal variable names to the names of the external files or attributes supplying the data. Note that a 2D grid map has the extension ".map," a time series the extension ".tss," and a lookup table the extension ".tbl." Note also that the value of constants or starting values for single attributes can be defined in the binding. The second section is the areamap, which defines the grid used for mapping (location, resolution, direction of counting, numbers of rows and columns). The third section is the timer, which simply gives the number of iterations to be used in the model.

```
binding        # Defines inputs and outputs

 # Inputs

 RainStations = rainstat.map;    # map with location of
rain stations

 RainTimeSeries = rain.tss;      # timeseries with rain at
rainstations (mm/3h)

 SoilInfTable = infilcap.tbl;    # table with infil. cap. of
soil types (mm/3h)

 SoilType = soil.map;            # soil map
 DEM = dem.map;                  # Digital elevation map

 SamplePlaces = samples.map;     # map with runoff
sampling locations

 ConvConst = 36000;              # conversion constant,
mm/3hours -> m3/s

 # Outputs

SampleTimeSeries = runoff.tss;  # timeseries runoff at
sample loc's

areamap             # Defines area to be operated on by
model clone.map;

timer          # Defines number of time steps in model
1 56 1;

initial           # Computes initial values for whole area

# Compute surface topology

report Ldd = lddcreate(DEM, 1E34, 1E34, 1E34, 1E34);

 # Coverage of meteorological stations for the whole
area

 report RainZones = spreadzone(RainStations,0,1);

# Create a map of infiltration capacity (mm/3hours),
based on a soilmap

 InfiltrationCapacity = lookupscalar(SoilInfTable,SoilT
ype);

dynamic       # Iterations of the model

 # Add rainfall to surface (mm/3h)

 SurfWater = timeinputscalar(RainTimeSeries,RainZo
nes);

 # Runoff per time step as water input to cell minus infil-
tration,

# and actual infiltration (mm/3h)

report Runoff, Inf =

accuthresholdflux, accuthresholdstate(Ldd,SurfWater,In
filtrationCapacity);

# Runoff (converted to m3/s) at each timestep for
selected locations

report SampleTimeSeries = timeoutput(SamplePlaces,R
unoff/ConvConst);

# report log of Runoff over whole area for visualisation

 report LogRun = log10((Runoff+0.001)/ConvConst);
```

Box 2. Model for simple simulation of precipitation, infiltration, and overland flow for 56 timesteps of 3-hour intervals, modeling time one week

The initial section sets and computes initial map values. The example in Box 2 shows the derivation from the DEM of topology (LDD "local drain direction" map, fig. 2) for routing flow, the creation of Thiessen polygons (RainZones) to define the coverage for each rain station as a spread operation (fig. 3), and the use of a lookup table to derive potential soil infiltration (InfiltrationCapacity) from the soil type map shown in figure 2.

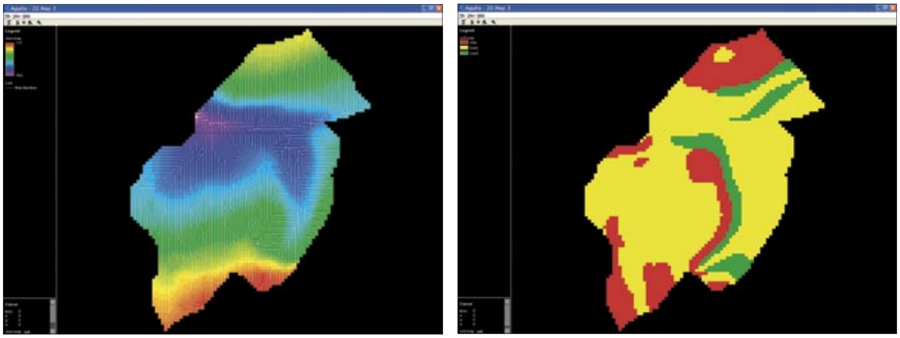


Figure 2. Left, digital elevation model (Dem, m, purple: low, red: high) with Local Drain Direction Map (LDD white lines); Right, soil map (SoilType), red: clay, yellow: loam, green: sand.

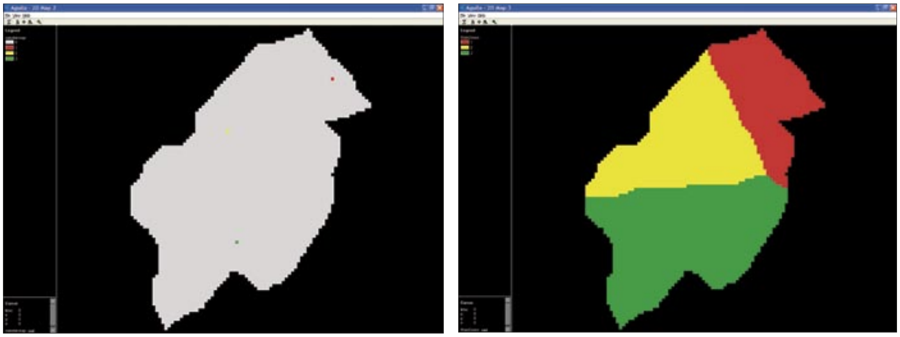


Figure 3. The map with the rain stations (RainStations) and the resulting Thiessen polygons (RainZones) created with the operation report RainZones = spreadzone(RainStations,0,1)

The dynamic section contains the model iterations representing the change through time. First the precipitation from the rain gauges (fig. 4) is extrapolated to the whole area of the Thiessen polygon surrounding any given rain gauge. This is done with the timeinputscalar operation, which for each time step and for each rain zone in RainZones reads a value from the time series file RainTimeSeries (fig. 5). The overland flux of runoff is determined by the accuthresholdflux operation, which computes flow to the next downstream cell only when the input of water, which is rain plus run-on from upstream cells, exceeds the infiltration capacity of the cell. Maps of the distribution of overland flow are obtained by saving the maps of runoff for each time step. These values are converted to logarithms for ease of display and the resulting series of maps can be shown as a movie using the PCRaster visualization software (fig. 6). The sequence of flux is written as an output time series file for the sites indicated on sample.map in the binding (fig. 7). Note that, to minimize run times, model results are only saved when the statement is preceded by a report command.

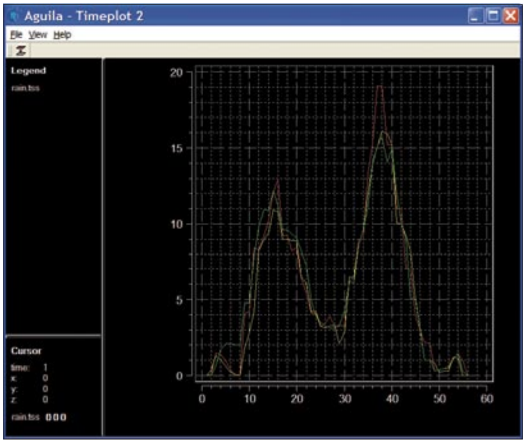


Figure 4. Rainfall time series. Rainfall measured at three rain stations shown on figure 3. X-axis shows time steps each with a duration of 3 hours.

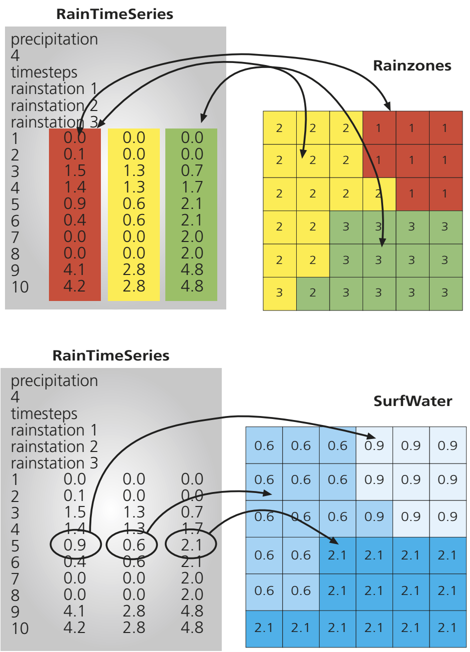


Figure 5. Left: heading and upper part of the time series file RainTimeSeries. First column: time steps, second up to fourth column: rainfall measured at first, second, and third rain station. Top-right: zoomed area of the RainZones map with three rain zones associated with the three rain stations. The arrows in the upper figure show that each column in the rainfall time series is linked to an area on the RainZones map. The operation

SurfWater = timeinputscalar(RainTimeSeries,RainZones);

reads for each time step a row of values from the time series file and assigns these values to the SurfWater map for that time step, as shown in the lower figure for Time Step 5.

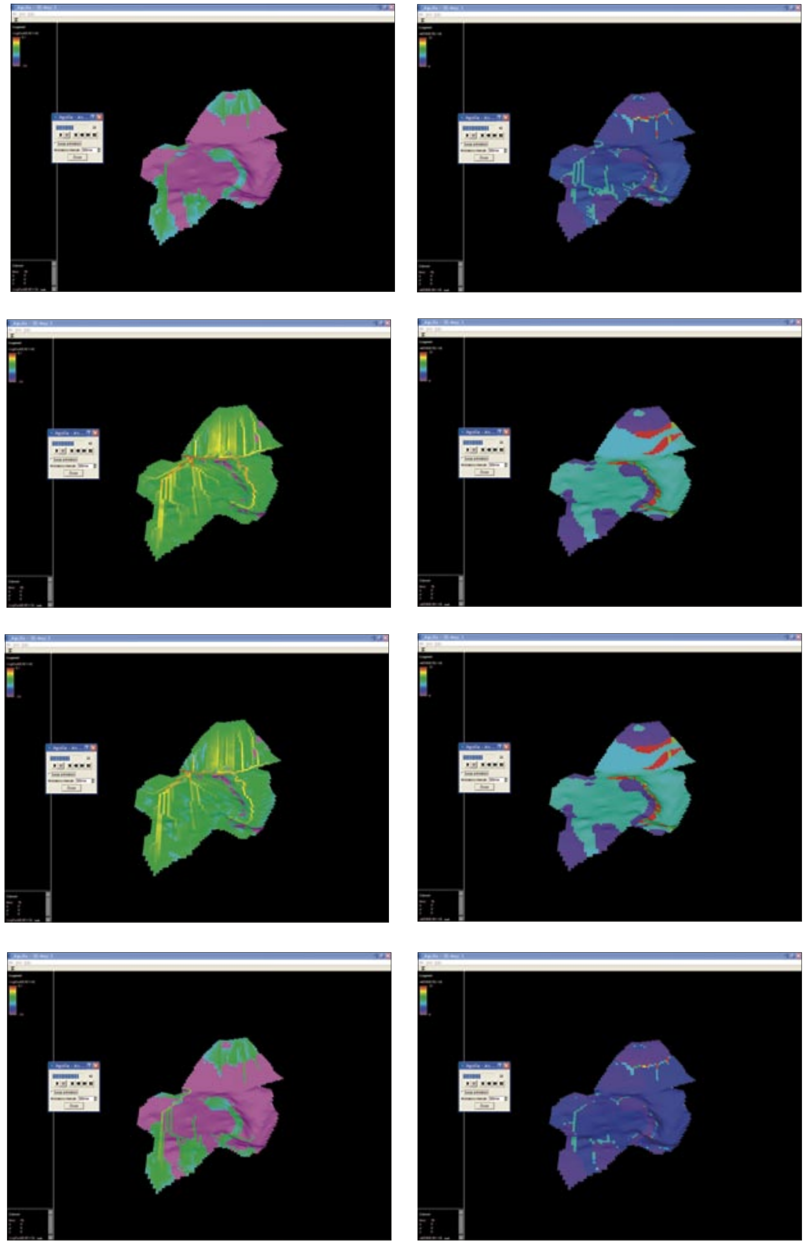


Figure 6. Slices from a PCRaster animation of the temporal change of runoff (LogRun, left), and actual infiltration (Inf, right). From top to bottom, the time steps 30, 35, 40, and 45 are shown, representing the second rainfall event on the rainfall time series (fig. 4) occurring over a period of 45 hours. Note that the areas with high actual infiltration rates are associated with sandy soils on the soil-type map shown in figure 2. At Time Steps 30 and 45, runoff occurs only on the clay soil type having the lowest infiltration capacity.

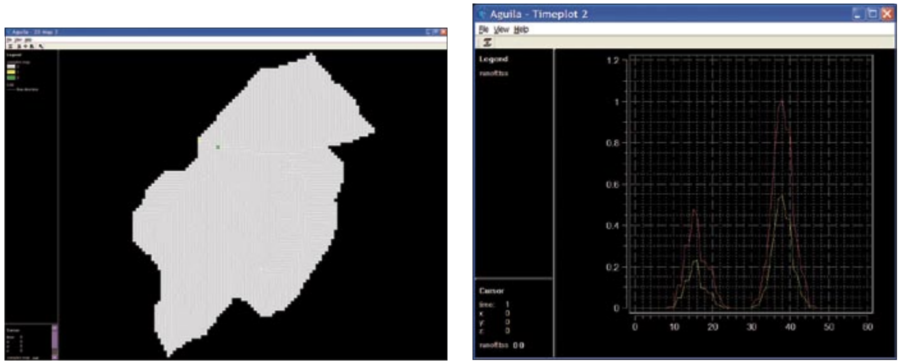


Figure 7. Left: sample locations (SamplePlaces) and right: runoff simulated at these sample locations as stored in a time series file (SampleTimeSeries) using a timeoutput operation. The sample locations are located in the main streams as can be seen from the local drain direction pattern (Ldd) draped over the SamplePlaces map.

In this example the soil water balance is represented in a simplified way. In reality, the inputs per cell would consist of precipitation and runon minus evapotranspiration. The amount of water held in the soil will also have an effect on infiltration and runoff. Consequently, for each time step there is a feedback between the moisture remaining in the soil after runoff and infiltration, which means that the moisture input to cells in later cycles of the model must take account of what happened in the previous cycle. The feedback loop may be computed by modifying the model script by adding two extra lines of code, as seen in Box 3. This model script represents a well-known approach to dealing with feedback in soil water infiltration and flow, known as the Green-Ampt equation (Green and Ampt 1911), which needs merely two additional lines of PCRaster code (see Box 3). First, the cumulative infiltration (InfCum) is recorded for each time step by adding the actual infiltration (Inf) for each time step to InfCum. Second, the potential infiltration is modeled as a function of this cumulative infiltration, with three parameters (Ks, B, and DTau) used in the Green-Ampt equation. Karssenberg (2002b) gives an example of a more complicated rainfall-runoff model and discusses the value of PCRaster for hydrological modeling.

```
dynamic     # Iterations of the model

 # Add rainfall to surface (mm/3h)

 SurfWater = timeinputscalar(RainTimeSeries,
RainZones);

 # cumulative infiltration (mm)

 InfCum=InfCum+Inf;

 # potential infiltration per time step (mm/3h)

 InfiltrationCapacity = Ks * ( (-B*Dtau+InfCum)
/ InfCum );

 # Runoff per time step as water input to cell
minus infiltration,

 # and actual infiltration (mm/3h)

 report Runoff, Inf = accuthresholdflux(Ldd,SurfWater,
InfiltrationCapacity)
```

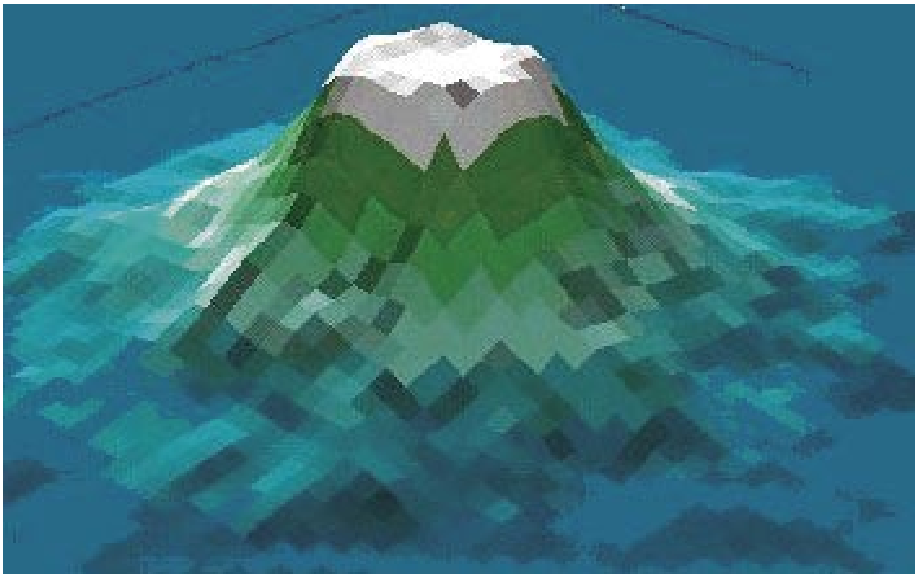
Box 3. Model component with feedback regarding infiltration

EXAMPLES OF PCRASTER APPLICATIONS

Modeling cumulative changes in landscapes: the growth of volcanoes and the effects of erosion and sedimentation.

Processes such as erosion and deposition in a landscape may be difficult to appreciate because they operate over geological, rather than human, time scales, but they are similar in concept to the phenomenon of surface water transport in which material is moved downslope along a path of least resistance. Consider the formation of volcanoes, for example. Essentially these landforms develop by discharge of volcanic lava from a central chamber, which then flows over a steepest downhill path. The lava cools, solidifies, and modifies the landform in such a way that the next lava discharge must take another path. As the paths accumulate, so does the characteristic conic form develop.

A PCRaster model may provide useful insights into processes of erosion and deposition. For example, formation of simple volcanoes can easily be modeled by starting with a random amount of lava discharged per cycle from a central vent. The paths taken by the lava are determined by the surface over which it flows so the ldd function may be used to route the lava flow over the cone. As the cone cools and solidifies, it modifies the form of the underlying DEM and hence, the possible surface over which lava may flow. Adding a small amount of uncertainty to the surface of the growing cone to indicate that deposition of molten lava is not confined to a single path ensures that the average growth of the cone is much the same, and a symmetrical feature evolves. Figure 8 illustrates some of the steps in the simulation and compares the final result with the processes operating on real volcanoes such as Arenal in Costa Rica and Vesuvius in Italy.



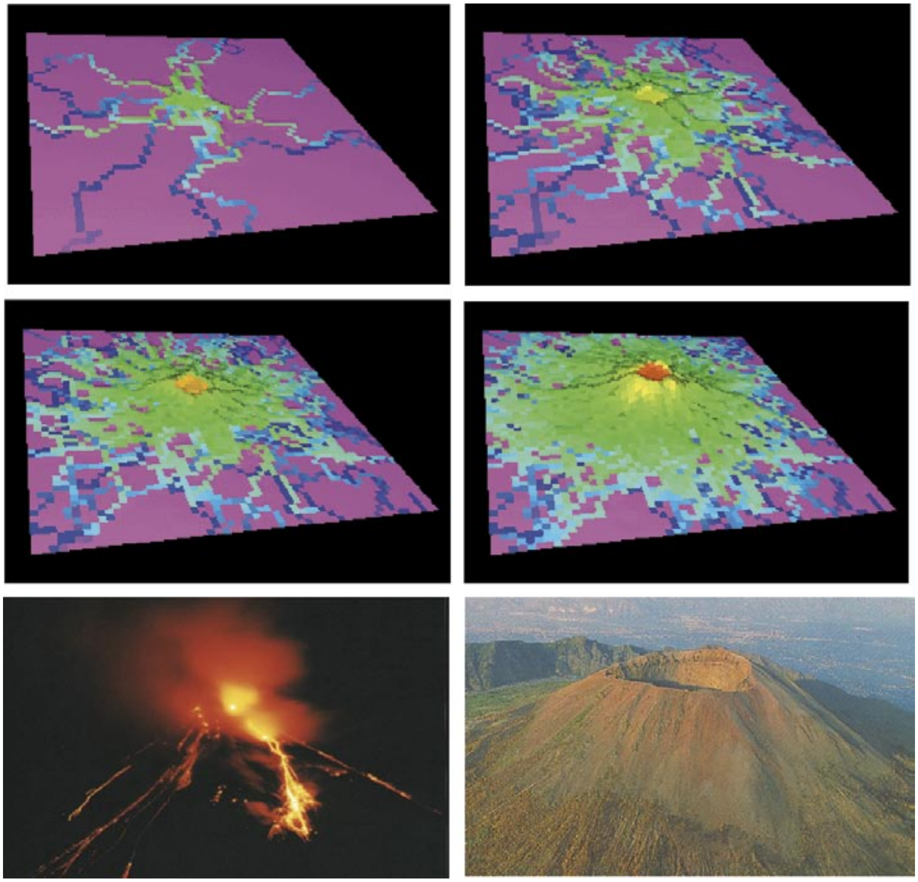


Figure 8. Modeling the development of a volcanic cone (4 steps selected from 250.) Compare process and results with Arenal (left bottom) and Vesuvius (right bottom). (Photo: Arenal – R. Sluiter)

The growth of a volcano only entails the addition and distribution of new material over the growing landscape. In many situations, the land surface is subject to both erosion and deposition. Figure 9 illustrates the formation of simple gullies on a uniform slope covered with a uniform soil. Clearly, the feedback in the model resembles the kinds of erosion seen in the field in which an initially linear system is transformed into a non-linear result (gullies varying in depth down the field). Figure 10 illustrates the same process, but this time for a landscape in which there is a buried, resistant rock layer, such as a basalt flow or a layer of limestone or hard slate. At the start of the simulation, the hard rock layer cannot be seen and has no effect on the process because it is buried under layers of softer sediments. After a certain period of time, however, the hard rock layer becomes exposed to the elements, but as it erodes much more slowly than the other sediments, the landform changes in response to its presence.

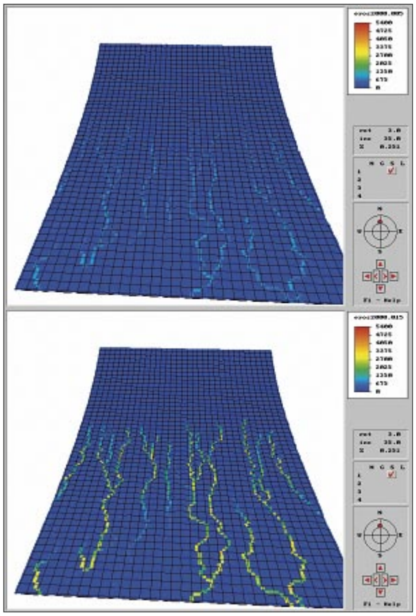


Figure 9. Example of modeling gully erosion on a sloping paddock.

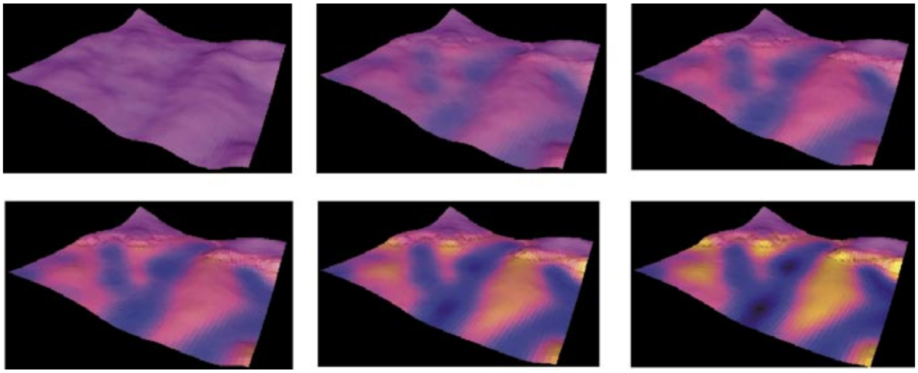


Figure 10. Selection from a series of simulations of landscape development as a buried hard rock layer becomes exposed. Light yellow colors indicate erosion, dark blue colors indicate deposition. Numbers indicate numbers of model cycles.

All these examples illustrate the value and the role of creating positive and negative feedback loops in PCRaster to deal with situations for which the end result can only be known by running the simulation–the results of these models are difficult to predict as the processes are non-linear, both in time and space.

User models built with PCRaster: the LISEM and LISFLOOD models

PCRaster has been used in several studies for flood management and erosion control. For water routing and erosion and deposition mentioned above, the PCRaster models use the "accu" family of functions, which route all available material in one time step through the catchment. The "accu" approach is insufficient when flood routing becomes important, when the time step of the model approaches the average travel time through a cell, or when the number of pixels along the transport path is large. In these cases, a more detailed approach is needed. This approach can be found in the conventional algorithms for flood routing such as the kinematic wave approximation and the diffusion wave model (Chow, Maidment, and Mays 1988, Fread 1993, Singh 1996). The kinematic wave was implemented as an extra PCRaster function using the following syntax:

Q2 = kinematic(Ldd, Q1, Input, Alpha, Beta, DtSecs, CellLengthMetres);

The new discharge Q2 at a certain location within the flow network (defined by Ldd) is calculated from the discharge Q1 from the previous time step at the same location and using Q2 from neighboring upstream pixels. The kinematic wave can be used for hydrologic catchment models for overland flow routing and for channel routing under certain conditions. Extensions of the kinematic wave, which include backwater effects, are found in the new dynamic wave equation modules, which are currently being tested in PCRaster. These functions further expand the PCRaster capabilities for physically based models.

The LISFLOOD distributed catchment model (De Roo, Wesseling, and van Deursen 2000) was written in PCRaster to investigate the origin and causes of flooding and the influence of land use, soil characteristics, and antecedent catchment saturation. LISFLOOD simulates runoff and flooding in large river basins as a consequence of extreme rainfall; it is a distributed rainfall–runoff model taking into account the influences of topography, precipitation amounts and intensities, antecedent soil moisture content, land use type, and soil type. LISFLOOD is a major extension of LISEM, an earlier PCRaster model that was developed for simulating soil erosion in Limburg, the Netherlands (De Roo 1996), and also in Mediterranean countries and China.

Processes that can be simulated in LISFLOOD are precipitation, interception, snowmelt, evapotranspiration, infiltration, percolation, groundwater flow, lateral flow, and surface runoff (fig. 11). LISFLOOD uses rainfall and temperature time series as input. Data from many meteorological stations can be used. Inundation extents may be computed by extrapolating predicted water levels onto a DEM, or LISFLOOD can be linked to an existing 2D or 3D model for detailed floodplain routing. Infiltration is simulated using a two-layer Green-Ampt equation. The LISFLOOD model contains around 200 lines of code, which is considerably less than a model written in FORTRAN or C++.

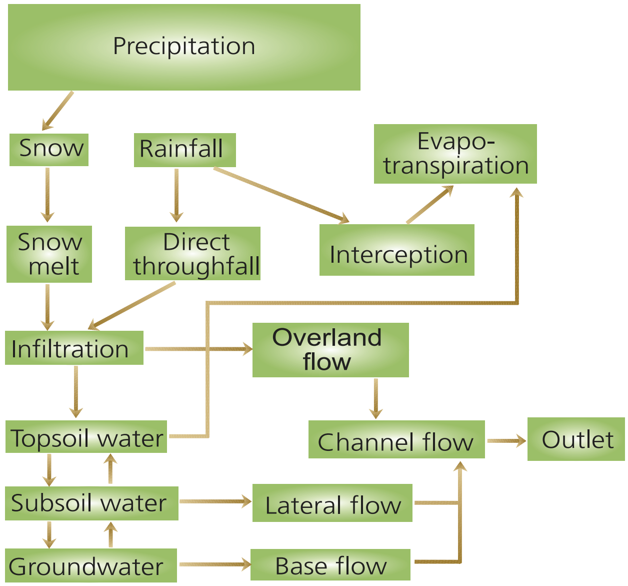


Figure 11. Flowchart of the LISFLOOD model.

Outputs of LISFLOOD are time series of discharge at user-defined catchment outlets and sub-outlets. Furthermore, final maps of source areas of water, total rainfall, total interception, total infiltration, etc. can be produced, as well as a series of maps showing changes in time of certain variables, such as the water depth in each pixel.

The LISFLOOD model has been used in pilot studies to investigate flooding problems in two transnational European river basins, namely the Meuse catchment, covering parts of France, Belgium, and the Netherlands, and the Oder basin, covering parts of the Czech Republic, Poland, and Germany. The Meuse suffered from extreme flooding in December 1993 and January/February 1995. The Oder area was flooded in July 1997 (De Roo and Schmuck 2002). In these catchments, LISFLOOD was tested, calibrated, and validated by stream flow data from gauging stations before a detailed analysis of the causes of flooding could be examined. SAR images were used to validate the (maximum) extent of the flooded area in the floodplain.

NEW APPROACHES TO MODELING RECENTLY ADDED TO THE TOOLKIT

Modules for simulating alluvial architecture.

As mentioned in Section 4.2, new modules can be added to the PCRaster software in order to extend the application field of the software to a wider range of environmental systems and modeling approaches. While the existing PCRaster software comes with many functions for hydrological modeling, several recent applications of PCRaster have also been in the field of large-scale sedimentological modeling. One example is the group of models referred to as alluvial architecture models that have been developed to understand the processes driving delta evolution and to predict the nature of hydrocarbon and water reservoirs consisting of deposits from rivers (Karssenberg, Törnqvist, and Bridge 2001, Mackey and Bridge 1995). This group of models simulates the landscape evolution of a floodplain or river delta over time periods of thousands to millions of years. The aim of these models is to simulate the temporal evolution of the channel network on a river delta, the spatial pattern of sedimentation and erosion as a function of this continuously changing channel network, and the resulting three-dimensional architecture of the different types of sediments deposited through time.

To provide model builders with the appropriate suite of tools to develop these kinds of models, several tools are currently being added to the PCRaster software. One of these is a new water (and sediment) routing algorithm that can deal with divergent flow patterns. Most standard GIS, including PCRaster, include functions to derive the local drain direction (LDD) network from a digital elevation model. While this network is very powerful for representing flow patterns in hilly catchments, it is insufficient to represent divergent flows in relatively flat terrain such as a river floodplain. This limitation of the LDD network is due to the use of single flow directions assigned to each cell (fig. 12A and 13A). As a result, channel confluences can be represented, but channel divergences cannot. To solve this problem, new, experimental functions for flow routing have been added that use multiple flow directions for each cell, resulting in channel networks that can include both channel confluences and divergences (fig. 12B and 13B). This group of functions can also be used for constructing hydrological and erosion models for hilly catchments in which multiple flow directions are currently widely applied (Burrough and McDonnell 1998) replacing the single flow direction algorithm. These functions are still in a prototype phase. In addition, a new module is currently being developed to deal with three-dimensional layering in sedimentological models (Karssenberg, de Jong, and Burrough 2000, Karssenberg 2002a, Karssenberg and de Jong Forthcoming). This module provides functions for operating on three-dimensional blocks of data consisting of stacks of voxels with variable thickness and attribute values (fig. 14). When sedimentation occurs, voxels are added at the top of this block; when erosion occurs, voxels are removed or sliced. Compaction of sediments can be simulated by decreasing the thickness of individual voxels. Although this module is still in the prototype phase, the aim is to provide a wide range of functions operating on this three-dimensional data type, integrated with existing PCRaster modeling tools using two dimensional data (maps).

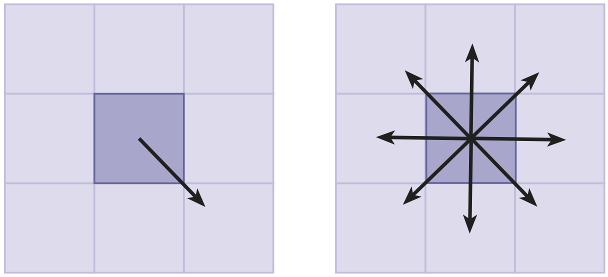


Figure 12. Flow direction(s) shown for a single cell on a map. Left: approach used in the local drain direction network of standard GIS; each cell has a single flow direction to one of its neighboring cells, and material is moved to that single cell only; right: approach where each cell may have flow to each of its 8 neighbors and where material can, therefore, flow in multiple directions.

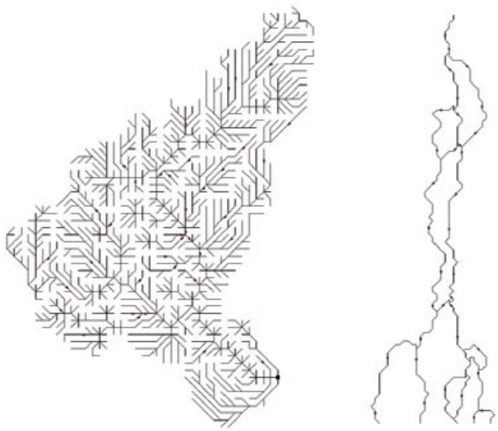


Figure 13. Resulting drainage networks from the approaches shown in figure 12. Left: approach used in local drain direction network of standard GIS, with single flow directions resulting in convergent flow only. Right: approach with multiple flow directions for each cell, used to represent convergent and divergent flow. Note that the right figure shows only the flow directions for the main channels.

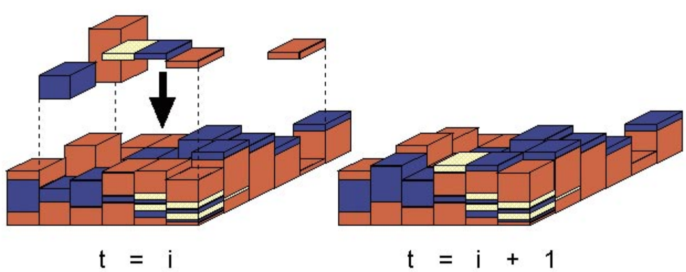


Figure 14. Three-dimensional block containing stacks of voxels with variable thickness. Deposition of material between t = i and t = i+1 is represented by addition of voxels at the top of the block.

Geostatistics and dynamic modeling.

There are many situations in dynamic modeling which are incompletely known and where a geostatistical approach is appropriate. PCRaster is linked to a full suite of methods for geostatistical interpolation (Gstat) and conditional or unconditional spatial simulation (Pebesma and Wesseling 1998) that is now also incorporated in the R suite of statistical methods (Bivand and Gebhardt 2000). The module for spatial simulation can be used for calculating error propagation in models built with PCRaster, using the approach of Monte Carlo simulation. To make error propagation modeling with Monte Carlo simulation available to a wider audience of modelers, prototype language extensions are being developed with new functions for Monte Carlo simulation with dynamic models (Karssenberg 2002a).

OpenMI

PCRaster was developed as a general-purpose simulation language, but it is becoming increasingly clear that many specialist models (including hydrodynamic models) have been written and maintained by other workers. These models might provide useful material for expanding PCRaster functionality, but it would require a prohibitively huge effort to translate them into PCRaster. This is a generic problem that extends beyond the PCRaster environment: several organizations have already invested quite some time and effort in the development of models, and although they don't want to lose their investments, they would like to offer their models to the outside world to be linked with other models. There are several ongoing international projects that are aimed at providing generic frameworks for linking models. As an example, PCRaster is participating in the HarmonIT framework, a research project funded by the European Commission aiming at the development and implementation of a European Open Modelling Interface and Environment (OpenMI) that will simplify the linking of hydrological and related models.

REFERENCES

Bivand, R., and A. Gebhardt. 2000. Using the R statistical data analysis language on GRASS 5.0 GIS data base files. Computers and Geosciences 26: 1043–52.

Burrough, P. A. 1998. Dynamic modeling and geocomputation. In Geocomputation: A primer, ed. P. A. Longley, S. M. Brooks, R. McDonnell, and B. MacMillan. Chichester: John Wiley and Sons.

Burrough, P. A., and R. A. McDonnell. 1998. Principles of geographical information systems. Oxford: Oxford University Press.

Casti, J. 1997. Would-be worlds. New York: John Wiley and Sons.

Chow, V. T., D. R. Maidment, and L. W. Mays. 1988. Applied hydrology. New York City: McGraw-Hill.

De Roo, A. P. J. 1996. The LISEM project: An introduction. In Hydrological processes 10: 1021–26.

De Roo, A. P. J., and G. Schmuck. 2002. ODER-LISFLOOD: Assessment of the effects of engineering, land use and climate scenarios on flood risk in the Oder catchment. European Commission, EUR 20276.

De Roo, A. P. J., C. G. Wesseling, and W. P. A. Van Deursen. 2000. Physically-based river basin modelling within a GIS: The LISFLOOD model. Hydrological Processes14: 1981–92.

Eastman, R. 2003 IDRISI. Clark Labs. Worcester, Mass.: Clark University.

Fread, D. L. 1993. Flow routing. In Handbook of hydrology, ed. D. R. Maidment, 10.1-10.36. New York City, McGraw Hill

Green, W. H., and G. A. Ampt. 1911. Studies on soil physics: Flow of air and water through soils. Journal of Agricultural Science 4:1–24.

Karssenberg, D., 2002a. Building dynamic spatial environmental models. Nederlandse Geografische Studies. 305. Utrecht: KNAG/Faculteit Ruimtelijke Wetenschappen Universiteit Utrecht. www.library.uu.nl/digiarchief/dip/diss/2003-0414-125023/inhoud.htm.

Karssenberg, D. 2002b. The value of environmental modelling languages for building distributed hydrological models. Hydrological Processes 16: 2751–66.

Karssenberg, D., and K. de Jong. Forthcoming. New concepts for dynamic environmental modelling in GIS: Modelling in three spatial dimensions. International Journal of Geographical Information Science.

Karssenberg, D., K. de Jong, and P. A. Burrough. 2000. A prototype computer language for environmental modeling in the temporal, 3D spatial and stochastic dimension. In International conference on integrating geographic information systems and environmental modeling: Problems, prospectus, and needs for research. www.colorado.edu/research/cires/banff/pubpapers/192.

Karssenberg, D., T. E. Törnqvist, and J. S. Bridge. 2001. Conditioning a process-based model of sedimentary architecture to well data. Journal of Sedimentary Research, 71: 868–79.

Mackey, S.D., and J. S. Bridge. 1995. Three-dimensional model of alluvial stratigraphy: Theory and application. Journal of Sedimentary Research B65: 7-31.

Pebesma, E., and C. G. Wesseling. 1998. GSTAT—A program for geostatistical modelling, prediction and simulation. Computers and Geosciences 24: 17-31.

PCRaster. 2004. pcraster.geog.uu.nl.

Pullar, D. 2003. Simulation modelling applied to runoff modelling using MapScript. Transactions in GIS 7: 267–83.

Singh, V.P.. 1996. Kinematic wave modeling in water resources. New York: John Wiley and Sons.

Stewart, I. 1989. Does God play dice? Oxford: Blackwell.

Takeyama, M., and H. Couclelis. 1997. Map dynamics: Integrating cellular automata and GIS through Geo-Algebra. International Journal of Geographical Science 11:73-92.

Tomlin, C.D. 1990. Geographic information systems and cartographic modeling.

Englewood Cliffs, N.J.: Prentice Hall.

Van Deursen, W. P. A. 1995. Geographical information systems and dynamic models: Development and application of a prototype spatial modelling language. PhD thesis, University of Utrecht. www.carthago.nl (under publications).

Wesseling, C. G., D.J. Karssenberg, P. A. Burrough, and Van Deursen, W. P. A. 1996. Integrating dynamic environmental models in GIS: The development of a Dynamic Modeling Language. Transactions in GIS 1: 40-48.