

Incorporating level set methods in Geographical Information Systems (GIS) for land-surface process modeling

D. Pullar

Geography Planning and Architecture, The University of Queensland, Brisbane QLD 4072, Australia

Received: 1 August 2004 – Revised: 1 November 2004 – Accepted: 15 November 2004 – Published: 9 August 2005

Abstract. Land-surface processes include a broad class of models that operate at a landscape scale. Current modelling approaches tend to be specialised towards one type of process, yet it is the interaction of processes that is increasing seen as important to obtain a more integrated approach to land management. This paper presents a technique and a tool that may be applied generically to landscape processes. The technique tracks moving interfaces across landscapes for processes such as water flow, biochemical diffusion, and plant dispersal. Its theoretical development applies a Lagrangian approach to motion over a Eulerian grid space by tracking quantities across a landscape as an evolving front. An algorithm for this technique, called level set method, is implemented in a geographical information system (GIS). It fits with a field data model in GIS and is implemented as operators in map algebra. The paper describes an implementation of the level set methods in a map algebra programming language, called MapScript, and gives example program scripts for applications in ecology and hydrology.

1 Introduction

Over the past decade there has been an explosion in the application of models to solve environmental issues. Many of these models are specific to one physical process and often require expert knowledge to use. Increasingly generic modeling frameworks are being sought to provide analytical tools to examine and resolve complex environmental and natural resource problems. These systems consider a variety of land condition characteristics, interactions and driving physical processes. Variables accounted for include climate, topography, soils, geology, land cover, vegetation and hydro-geography (Moore et al., 1993). Physical interactions include processes for climatology, hydrology, topographic landsurface/sub-surface fluxes and biological/ecological sys-

tems (Sklar and Costanza, 1991). Progress has been made in linking model-specific systems with tools used by environmental managers, for instance geographical information systems (GIS). While this approach, commonly referred to as loose coupling, provides a practical solution it still does not improve the scientific foundation of these models nor their integration with other models and related systems, such as decision support systems (Argent, 2003). The alternative approach is for tightly coupled systems which build functionality into a system or interface to domain libraries from which a user may build custom solutions using a macro language or program scripts. The approach supports integrated models through interface specifications which articulate the fundamental assumptions and simplifications within these models. The problem is that there are no environmental modelling systems which are widely used by engineers and scientists that offer this level of interoperability, and the more commonly used GIS systems do not currently support space and time representations and operations suitable for modelling environmental processes (Burrough, 1998) (Sui and Magio, 1999).

Providing a generic environmental modeling framework for practical environmental issues is challenging. It does not exist now despite an overwhelming demand because there are deep technical challenges to build integrated modeling frameworks in a scientifically rigorous manner. It is this challenge this research addresses.

1.1 Background for Approach

The paper describes a generic environmental modeling language integrated with a Geographical Information System (GIS) which supports spatial-temporal operators to model physical interactions occurring in two ways. The trivial case where interactions are isolated to a location, and the more common and complex case where interactions propagate spatially across landscape surfaces. The programming language has a strong theoretical and algorithmic basis. Theoretically, it assumes a Eulerian representation of state space,

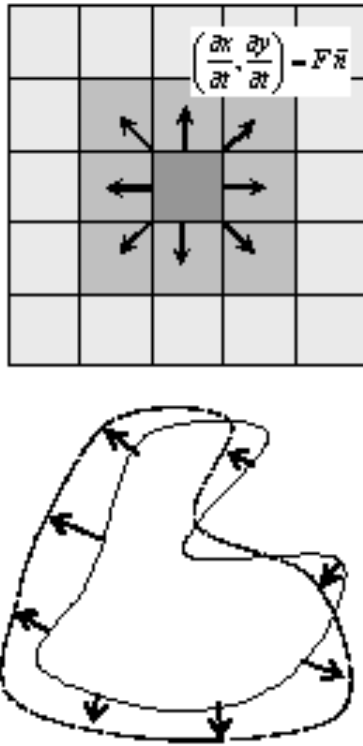


Fig. 1. Shows a) a propagating interface parameterised by differential equations, b) interface fronts have variable intensity and may expand or contract based on field gradients and driving process.

but propagates quantities across landscapes using Lagrangian equations of motion. In physics, a Lagrangian view focuses on how a quantity (water volume or particle) moves through space, whereas an Eulerian view focuses on a local fixed area of space and accounts for quantities moving through it. The benefit of this approach is that an Eulerian perspective is eminently suited to representing the variation of environmental phenomena across space, but it is difficult to conceptualise solutions for the equations of motion and has computational drawbacks (Press et al., 1992). On the other hand, the Lagrangian view is often not favoured because it requires a global solution that makes it difficult to account for local variations, but has the advantage of solving equations of motion in an intuitive and numerically direct way. The research will address this dilemma by adopting a novel approach from the image processing discipline that uses a Lagrangian approach over an Eulerian grid. The approach, called level set methods, provides an efficient algorithm for modeling a natural advancing front in a host of settings (Sethian, 1999). The reason the method works well over other approaches is that the advancing front is described by equations of motion (Lagrangian view), but computationally the front propagates over a vector field (Eulerian view). Hence, we have a very generic way to describe the motion of quantities, but can explicitly solve their advancing properties locally as propagating zones. The research work will adapt this technique for modeling the motion of environmental variables across time

and space. Specifically, it will add new data models and operators to a geographical information system (GIS) for environmental modeling. This is considered to be a significant research imperative in spatial information science and technology (Goodchild, 2001).

The main focus of this paper is to evaluate if the level set method (Sethian, 1999) can:

- provide a theoretically and empirically supportable methodology for modeling a range of integral landscape processes,
- provide an algorithmic solution that is not sensitive to process timing, is computationally stable and efficient as compared to conventional explicit solutions to diffusive processes models,
- be developed as part of a generic modelling language in GIS to express integrated models for natural resource and environmental problems?

The outline for the paper is as follow. The next section will describe the theory for spatial-temporal processing using level sets. Section 3 describes how this is implemented in a map algebra programming language. Two application examples are given – an ecological and a hydrological example – to demonstrate the use of operators for computing reactive-diffusive interactions in landscapes. Section 4 summarises the contribution of this research.

2 Theory

2.1 Introduction

Level set methods (Sethian, 1999) have been applied in a large collection of applications including, physics, chemistry, fluid dynamics, combustion, material science, fabrication of microelectronics, and computer vision. Level set methods compute an advancing interface using an Eulerian grid and the Lagrangian equations of motion. They are similar to cost distance modeling used in GIS (Burroughs and McDonnell, 1998) in that they compute the spread of a variable across space, but the motion is based upon partial differential equations related to the physical process. The advancement of the interface is computed through time along a spatial gradient, and it may expand or contract in its extent. See Fig. 1.

2.2 Theory

The advantage of the level set method is that it models motion along a state-space gradient. Level set methods start with the equation of motion, i.e. an advancing front with velocity F is characterised by an arrival surface $T(x, y)$. Note that F is a velocity field in a spatial sense. If F was constant this would result in an expanding series of circular fronts, but for different values in a velocity field the front will have a more contorted appearance as shown in Fig. 1b. The motion of this

interface is always normal to the interface boundary, and its progress is regulated by several factors:

$$F = f(L, G, I) \quad (1)$$

where L =local properties that determine the shape of advancing front, G =global properties related to governing forces for its motion, I =independent properties that regulate and influence the motion. If the advancing front is modeled strictly in terms of the movement of entity particles, then a straightforward velocity equation describes its motion:

$$|\nabla T| F = 1 \quad \text{given } T_0 = 0 \quad (2)$$

where the arrival function $T(x, y)$ is a travel cost surface, and T_0 is the initial position of the interface. Instead we use level sets to describe the interface as a complex function. The level set function ϕ is an evolving front consistent with the underlying viscosity solution defined by partial differential equations. This is expressed by the equation:

$$\phi_t + F |\nabla \phi| = 0 \quad \text{given } \phi(x, y, t = 0) \quad (3)$$

where ϕ_t is a complex interface function over time period $0..n$, i.e. $\phi(x, y, t)=t_0..tn$, $\nabla \phi$ is the spatial and temporal derivatives for viscosity equations. The Eulerian view over a spatial domain imposes a discretisation of space, i.e. the raster grid, which records changes in value z . Hence, the level set function becomes $\phi(x, y, z, t)$ to describe an evolving surface over time. Further details are given in Sethian (1999) along with efficient algorithms. The next section describes the integration of the level set methods with GIS.

3 Map algebra modelling

3.1 Map algebra

Spatial models are written in a map algebra programming language. Map algebra is a function-oriented language that operates on four implicit spatial data types: point, neighbourhood, zonal and whole landscape surfaces. Surfaces are typically represented as a discrete raster where a point is a cell, a neighbourhood is a kernel centred on a cell, and zones are groups of cells. Common examples of raster data include terrain models, categorical land cover maps, and scalar temperature surfaces. Map algebra is used to program many types of landscape models ranging from land suitability models to mineral exploration in the geosciences (Burrough and McDonnell, 1998; Bonham-Carter, 1994).

The syntax for map algebra follows a mathematical style with statements expressed as equations. These equations use operators to manipulate spatial data types for point and neighbourhoods. Expressions that manipulate a raster surface may use a global operation or alternatively iterate over the cells in a raster. For instance the GRID map algebra (Gao et al., 1993) defines an iteration construct, called *docell*, to apply equations on a cell-by-cell basis. This is trivially performed on columns and rows in a clockwork manner. However, for environmental phenomena there are situations

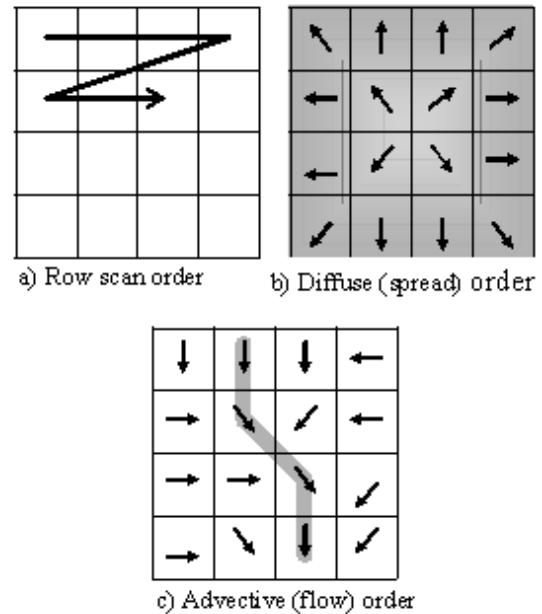


Fig. 2. Spatial processing orders for raster.

where the order of computations has a special significance. For instance, processes that involve spreading or transport acting along environmental gradients within the landscape. Therefore special control needs to be exercised on the order of execution. Burrough (1998) describes two extra control mechanisms for diffusion and directed topology. Figure 2 shows the three principle types of processing orders, and they are:

- row scan order governed by the clockwork lattice structure,
- spread order governed by the spreading or scattering of a material from a more concentrated region,
- flow order governed by advection which is the transport of a material due to velocity.

Our implementation of map algebra, called MapScript (Pullar, 2001), includes a special iteration construct that supports these processing orders. MapScript is a lightweight language for processing raster-based GIS data using map algebra. The language parser and engine are built as a software component to interoperate with the IDRISI GIS (Eastman, 1997). MapScript is built in C++ with a class hierarchy based upon a value type. Variants for value types include numerical, boolean, template, cells, or a grid. MapScript supports combinations of these data types within equations with basic arithmetic and relational comparison operators. Algebra operations on templates typically result in an aggregate value assigned to a cell (Pullar, 2001); this is similar to the convolution integral in image algebras (Ritter et al., 1990). The language supports iteration to execute a block of statements in three ways: a) *docell* construct to process raster in a row scan order, b) *dospread* construct to process raster in a spread

```

while (time < 100)
  dospread
    pop = pop + (diffuse(kernel*pop))
    pop = pop +(r * pop * dt * (1 - (pop/K)))
  enddo
end
where the diffusive constant is stored in the kernel:

```

$$\text{kernel} = \begin{array}{|c|c|c|} \hline \cdot & D & \cdot \\ \hline D & D & D \\ \hline \cdot & D & \cdot \\ \hline \end{array}$$

Fig. 3. Map algebra script and convolution kernel for population dispersion. The variable *pop* is a raster, *r*, *K* and *D* are constants, *dt* is the model time step, and the kernel is a 3×3 template. It is assumed a time step is defined and the script is run in a simulation. The first line contained in the nested cell processing construct (i.e. *dospread*) is the diffusive term and the second line is the population growth term.

order, c) *doflow* to process raster by flow order. Examples are given in subsequent sections. Process models will also involve a timing loop which may be handled as a general *while(<condition>).end* construct in MapScript where the condition expression includes a system *time* variable. This time variable is used in a specific fashion along with a system time *step* by certain operators, namely *diffuse()* and *fluxflow()* described in the next section, to model diffusion and advection as a time evolving front. The evolving front represents quantities such as vegetation growth or surface runoff.

3.2 Ecological example

This section presents an ecological example based upon plant dispersal in a landscape. The population of a species follows a controlled growth rate and at the same time spreads across landscapes. The theory of the rate of spread of an organism is given in Tilman and Kareiva (1997). The area occupied by a species grows log-linear with time. This may be modelled by coupling a spatial diffusion term with an exponential population growth term; the combination produces the familiar reaction-diffusion model.

A simple growth population model is used where the reaction term considers one population controlled by births and mortalities is:

$$\frac{dN}{dt} = r \cdot N \left(1 - \frac{N}{K} \right) \quad (4)$$

where *N* is the size of the population, *r* is the rate of change of population given in terms of the difference between birth and mortality rates, and *K* is the carrying capacity. Further discussion of population models can be found in Jrgensen and

Bendoricchio (2001). The diffusive term spreads a quantity through space at a specified rate:

$$\frac{du}{dt} = D \frac{d^2u}{dx^2} \quad (5)$$

where *u* is the quantity which in our case is population size, and *D* is the diffusive coefficient.

The model is operated as a coupled computation. Over a discretized space, or raster, the diffusive term is estimated using a numerical scheme (Press et al., 1992). The distance over which diffusion takes place in time step *dt* is minimally constrained by the raster resolution. For a stable computational process the following condition must be satisfied:

$$2D \frac{dt}{dx^2} \leq 1 \quad (6)$$

This basically states that to account for the diffusive process, the term $2D \cdot dx$ is less than the velocity of the advancing front. This would not be difficult to compute if *D* is constant, but is problematic if *D* is variable with respect to landscape conditions.

This problem may be overcome by progressing along a diffusive front over the discrete raster based upon distance rather than being constrained by the cell resolution. The processing and diffusive operator is implemented in a map algebra programming language. The code fragment in Fig. 3 shows a map algebra script for a single time step for the coupled reactive-diffusion model for population growth.

The operator of interest in the script shown in Fig. 3 is the *diffuse* operator. It is assumed that the script is run with a given time step. The operator uses a system time *step* which is computed to balance the effect of process errors with efficient computation. With knowledge of the time step the iterative construct applies an appropriate distance propagation such that the condition in Eq. (3) is not violated. The level set algorithm (Sethian, 1999) is used to do this in a stable and accurate way. As a diffusive front propagates through the raster, a cost distance kernel assigns the proper time to each raster cell. The time assigned to the cell corresponds to the minimal cost it takes to reach that cell. Hence cell processing is controlled by propagating the kernel outward at a speed adaptive to the local context rather than meeting an arbitrary global constraint.

3.3 Hydrological example

This section presents a hydrological example based upon surface dispersal of excess rainfall across the terrain. The movement of water is described by the continuity equation:

$$\frac{\partial h}{\partial t} = e_t - \nabla \cdot q_t \quad (7)$$

where *h* is the water depth (m), *e_t* is the rainfall excess (m/s), *q* is the discharge (m/hr) at time *t*. Discharge is assumed to have steady uniform flow conditions, and is determined by Manning's equation:

$$q_t = v_t h_t = \frac{1}{n} h_t^{5/3} s^{1/2} \quad (8)$$

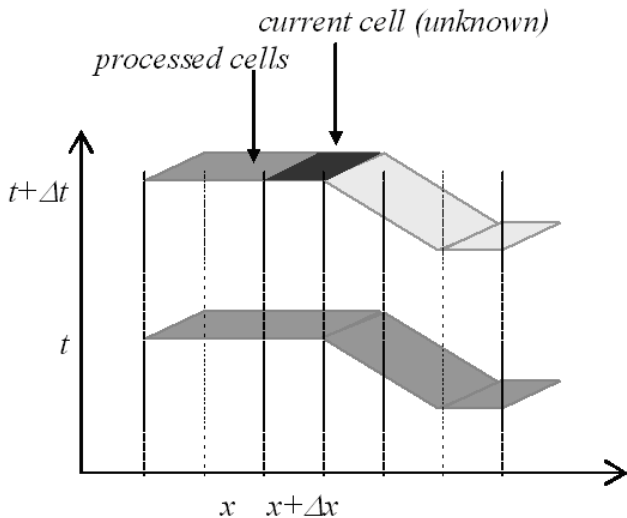


Fig. 4. Computation of current cell $(x + \Delta x, t + \Delta t)$.

where q_t is the flow velocity (m/s), h_t is water depth, and s is the surface slope (m/m). An explicit method of calculation is used to compute velocity and depth over raster cells, and equations are solved at each time step. A conservative form of a finite difference method solves for q_t in Eq. (5). To simplify discussions we describe quasi-one-dimensional equations for the flow problem. The actual numerical computations are normally performed on an Eulerian grid (Julien et al., 1995).

Finite-element approximations are made to solve the above partial differential equations for the one-dimensional case of flow along a strip of unit width. This leads to a coupled model with one term to maintain the continuity of flow and another term to compute the flow. In addition, all calculations must progress from an uphill cell to the down slope cell. This is implemented in map algebra by a iteration construct, called *doflow*, which processes a raster by flow order. Flow distance is measured in cell size Δx per unit length. One strip is processed during a time interval Δt (Fig. 4). The conservative solution for the continuity term using a first order approximation for Eq. (5) is derived as:

$$h_{x+\Delta x, t+\Delta t} = h_{x+\Delta x, t} - \frac{q_{x+\Delta x, t} - q_{x, t}}{\Delta x} \Delta t \quad (9)$$

where the inflow $q_{x, t}$ and outflow $q_{x+\Delta x, t}$ are calculated in the second term using Equation 6 as:

$$q_{x, t} = v_{x, t} \cdot h_t \quad (10)$$

The calculations approximate discharge from previous time interval. Discharge is dynamically determined within the continuity equation by water depth. The rate of change in state variables for Equation 6 needs to satisfy a stability condition where $v \cdot \Delta t / \Delta x \leq 1$ to maintain numerical stability. The physical interpretation of this is that a finite volume of water would flow across and out of a cell within the time step Δt . Typically the cell resolution is fixed for the raster, and adjusting the time step requires restarting the simulation

```
while (time < 120)
  doflow(dem)
    fvel = 1/n * pow(depth,m) * sqrt(grade)
    depth = depth + (depth * fluxflow(fvel))
  enddo
end
```

Fig. 5. Map algebra script for excess rainfall flow computed over a 120 minute event. The variables *depth* and *grade* are rasters, *fvel* is the flow velocity, *n* and *m* are constants in Manning's equation. It is assumed a time step is defined and the script is run in a simulation. The first line in the nested cell processing (i.e. *doflow*) computes the flow velocity and the second line computes the change in depth from the previous value plus any net change (inflow – outflow) due to velocity flux across the cell.

cycle. Flow velocities change dramatically over the course of a storm event, and it is problematic to set an appropriate time step which is efficient and yields a stable result.

The hydrological model has been implemented in a map algebra programming language Pullar (2003). To overcome the problem mentioned above we have added high level operators to compute the flow as an advancing front over a landscape. The time *step* advances this front adaptively across the landscape based upon the flow velocity. The level set algorithm (Sethian, 1999) is used to do this in a stable and accurate way. The map algebra script is given in Fig. 5. The important operator is the *fluxflow* operator. It computes the advancing front for water flow across a DEM by hydrological principles, and computes the local drainage flux rate for each cell. The flux rate is used to compute the net change in a cell in terms of flow depth over an adaptive time step.

4 Conclusions

The paper has described an approach to extend the functionality of tightly coupled environmental models in GIS (Argent, 2004). A long standing criticism of GIS has been its inability to handle dynamic spatial models. Other researchers have also addressed this issue (Burrough, 1998). The contribution of this paper is to describe how level set methods are: i) an appropriate scientific basis, and ii) able to perform stable time-space computations for modelling landscape processes. The level set method provides the following benefits:

- it more directly models motion of spatial phenomena and may handle both expanding and contracting interfaces,
- is based upon differential equations related to the spatial dynamics of physical processes.

Despite the potential for using level set methods in GIS and land-surface process modeling, there are no commercial or research systems that use this approach. Commercial systems such as GRID (Gao et al., 1993), and research systems such as PCRaster (Wesseling et al., 1996) offer flexible and

powerful map algebra programming languages. But operations that involve reaction-diffusive processing are specific to one context, such as groundwater flow. We believe the level set method offers a more generic approach that allows a user to program flow and diffusive landscape processes for a variety of application contexts. We have shown that it provides an appropriate theoretical underpinning and may be efficiently implemented in a GIS. We have demonstrated its application for two landscape processes – albeit relatively simple examples – but these may be extended to deal with more complex and dynamic circumstances.

The validation for improved environmental modeling tools ultimately rests in their uptake and usage by scientists and engineers. The tool may be accessed from the web site <http://www.geosp.uq.edu.au/projects/mapscript/> (version with enhancements available April 2005) for use with IDRIS GIS (Eastman, 1997) and in the future with ArcGIS. It is hoped that a larger community of users will make use of the methodology and implementation for a variety of environmental modeling applications.

Edited by: P. Krause, S. Kralisch, and W. Flügel

Reviewed by: anonymous referees

References

- Argent, R.: An Overview of Model Integration for Environmental Applications, *Environmental Modelling and Software*, 19, 219–234, 2004.
- Bonham-Carter, G. F.: *Geographic Information Systems for Geoscientists*, Elsevier Science Inc., New York, 1994.
- Burrough, P. A.: Dynamic Modelling and Geocomputation, in: *Geocomputation: A Primer*, edited by: Longley, P. A., et al., Wiley, England, 165–191, 1998.
- Burrough, P. A. and McDonnell, R.: *Principles of Geographic Information Systems*, Oxford University Press, New York, 1998.
- Gao, P., Zhan, C., and Menon, S.: An Overview of Cell-Based Modeling with GIS, in: *Environmental Modeling with GIS*, edited by: Goodchild, M. F., et al., Oxford University Press, 325–331, 1993.
- Goodchild, M.: A Geographer Looks at Spatial Information Theory, in: *COSIT – Spatial Information Theory*, edited by: Goos, G., Hertmanis, J., and van Leeuwen, J., LNCS 2205, 1–13, 2001.
- Jørgensen, S. and Bendoricchio, G.: *Fundamentals of Ecological Modelling*, Elsevier, New York, 2001.
- Julien, P. Y., Saghafian, B., and Ogden, F.: Raster-Based Hydrologic Modelling of Spatially-Variied Surface Runoff, *Water Resources Bulletin*, 31 (3), 523–536, 1995.
- Moore, I. D., Turner, A., Wilson, J., Jenson, S., and Band, L.: GIS and Land-Surface-Subsurface Process Modeling, in: *Environmental Modeling with GIS*, edited by: Goodchild, M. F., et al., Oxford University Press, New York, 1993.
- Press, W., Flannery, B., Teukolsky, S., and Vetterling, W.: *Numerical Recipes in C: The Art of Scientific Computing*, 2nd Ed. Cambridge University Press, Cambridge, 1992.
- Pullar, D.: MapScript: A Map Algebra Programming Language Incorporating Neighborhood Analysis, *GeoInformatica*, 5 (2), 145–163, 2001.
- Pullar, D.: Simulation Modelling Applied To Runoff Modelling Using MapScript, *Transactions in GIS*, 7 (2), 267–283, 2003.
- Ritter, G., Wilson, J., and Davidson, J.: Image Algebra: An Overview, *Computer Vision, Graphics, and Image Processing*, 4, 297–331, 1990.
- Sethian, J. A.: *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, 1999.
- Sklar, F. H. and Costanza, R.: The Development of Dynamic Spatial Models for Landscape Ecology: A Review and Progress, in: *Quantitative Methods in Ecology*, Springer-Verlag, New York, 239–288, 1991.
- Sui, D. and R. Maggio: Integrating GIS with Hydrological Modeling: Practices, Problems, and Prospects, *Computers, Environment and Urban Systems*, 23 (1), 33–51, 1999.
- Tilman, D. and P. Kareiva: *Spatial Ecology: The Role of Space in Population Dynamics and Interspecific Interactions*. Princeton University Press, Princeton, New Jersey, USA, 1997.
- Wesseling C. G., Karssenber, D., Burrough, P. A., and van Deursen, W. P.: Integrating Dynamic Environmental Models in GIS: The Development of a Dynamic Modelling Language, *Transactions in GIS*, 1 (1), 40–48, 1996.