Extending Map Algebra for Dynamic Spatial Models: A Hydrological Modelling Example

D. Pullar and **B. Fentie**

Geographical Sciences and Planning, The University of Queensland
Australia (d.pullar@mailbox.uq.edu.au)

Abstract: Dynamic spatial analysis addresses computational aspects of space-time processing. This paper describes the development of a spatial analysis tool and modelling framework that together offer a solution for simulating landscape processes. The spatial analysis tool extends representations of surface models used by traditional Geographical Information Systems (GIS) with special spatial operators and map algebra language constructs to handle dispersal and advective flows over terrain surfaces. This is developed as component technology and integrated into a modelling framework that performs dynamic simulation. The paper presents a hydrological application. The approach provides a modelling environment for scientists and land resource managers to write and to visualize spatial process models with ease.

Keywords: Spatial analysis, Map algebra, Simulation, Landscape modelling, Hydrology

1. INTRODUCTION

Landscape modelling is used to understand complex environmental systems and to codify our knowledge about natural systems. One challenge facing scientists and resource managers is overcoming the high learning curve associated with using modelling tools. Many tools have been developed as customised applications aimed at specialist end-users, but these rarely interoperate with other models to build integrated landscape models. This paper discusses modelling tools for hydrology. Hydrology, like other environmental disciplines, is a complex field with a bewildering set of tools used in practice. Even with background knowledge in hydrology, it often requires a significant effort to learn and use hydrological applications for real problems. Sometimes very different software tools are used for different aspects of hydrology; for instance surface runoff and ground flow modelling require different applications. The task of integrating hydrological models with other environmental disciplines, such as ecology and forestry, is more daunting.

This paper looks at new and innovative solutions for building integrated environmental models. It investigates: (i) generic *modelling frameworks,* and (ii) *analysis tools* for solving landscape modelling problems. We adopt a *divide-and-conquer* strategy to break complicated environmental problems into parts, and then solve each part individually. For instance, different aspects of the hydrological problem include precipitation, infiltration, overland flow, stream flow and material transport. The modelling framework provides the means to model each of these parts as components and couple them together in a coordinated fashion to build an application. We will discuss different options for modelling frameworks and analysis tools.

1.1 Modelling Frameworks

A modelling framework supports the coupling and coordination of components to build applications. Environmental modelling involves both spatial and dynamic analysis (Figure 1) and a number of approaches have developed to accommodate different analysis approaches. Geographical information systems (GIS) are increasingly used as a modelling framework. GIS, which is traditionally known for its spatial data management and visualisation capability, may be linked to specialised modelling applications. A GIS can be programmed to handle data transfers to and from external models, and to display results in a map-based view [Westervelt and Shapiro, 2000]. The main disadvantage of this approach is that conventional GIS do not include any dynamic analysis capability [Burrough, 1998] so coordinating the execution of components is difficult. Examples where GIS is integrated with hydrology models and ecology models generally compute time-averaged outputs.

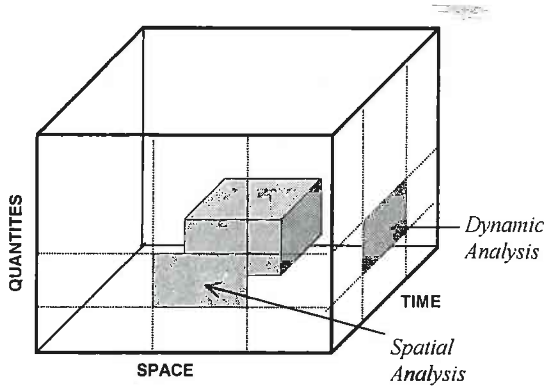


Figure 1. Conceptualisation of analysis tools for spatial dynamics.

Another approach is to use simulation to link models. Simulation software, like STELLA, is used to build deterministic systems as a functional network. The main disadvantage is that simulation software does not include any notion of space and expressing spatial dynamics is very cumbersome [Ford, 1999]. If we move away from conventional systems then there are some innovative modelling frameworks in development. One notable project is the integrated modelling architecture (IMA) at the Institute for Ecological Economics (IEE), University of Maryland [Villa, 2000]. IMA is a framework to construct and run models of natural systems. It supports the integration of component software which employ different representations and modelling techniques. The main improvement made by IMA over conventional simulation packages is that it extracts the semantic contexts for spatial dynamic models and expresses this in a high-level programming language. This provides a means to control and communicate space-time behaviour among separate model components. We adopt a similar approach built around a high-level specification language that describes hydrological model simulations. Other groups have also adopted modelling frameworks for hydrology, notably CMSS [Reed et al., 1999] who use an object-based representation as opposed the field-based representation used in this paper.

1.2 Spatial Analysis Tool

To perform environmental modelling it is important to use a data model that includes notions of time, space, and quantities. An appropriate logical data model for landscape processes is multivariate fields. Fields provide a continuous representation of landscape quantities and their variation across space; i.e. soils, slopes, and population density, etc. It is important to support a continuous representation of space as the physical behaviour of landscapes is often described by gradients, fluxes, and flows across space. Environmental models are then formulated in terms of inputs, governing parameters, and transformations. For practical reasons related to issues of data sampling and requirements for discrete computer representation, it is necessary to use a data structure that closely resembles the field data model. Map algebra is an example of a data representation for multivariate fields. The data structure in *map algebra* is based upon a uniform representation of space as discrete grids, which are organized into layers to represent various landscape quantities. Map algebra includes a functional notation to analyse distributions and patterns of spatial quantities [Tomlin, 1991]. It has proven to be popular for describing landscape models because it uses a data representation and processing constructs that are readily understood by typical end users.

The work in this paper adopts a map algebra approach with some enhancements for computing gradients, fluxes, and flows across space. Different aspects of the hydrological problem (precipitation, infiltration, overland flow, and stream flow) may be described using map algebra scripts. We encapsulate these scripts or programs as components to build complex applications. Other components, such as spreadsheets, may also be integrated into the final solution. The components perform their calculations at a point in time or over a time step, and it is up to the modelling framework described above to schedule the sequence of calculations and coordinate data parameter exchanges.

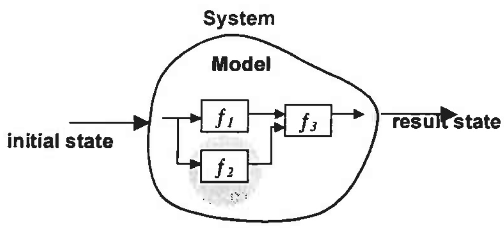


Figure 2. System of coupled components.

1.3 Objectives

This paper explores new modelling tools for describing hydrological applications. We apply a *divide-and-conquer* strategy to break hydrological problems into components for infiltration, surface flow and channel flow. From a hydrological modelling perspective we describe physical models as natural processes. The data descriptions and associated data processing are expressed in a map algebra language.

The main goals of our work are two-fold:

- To extend and refine map algebra for modelling material flows across landscapes. We have developed an implementation of the map algebra programming language called *MapScript* [Pullar, 2001] with additional processing constructs to model landscape flows.

- To provide modularity for map algebra programs using a modelling framework and to evaluate its application to hydrological models. MapScript programs are encapsulated as components with a simulator. The idea being that the simulator controls dynamic analysis, and MapScript performs the spatial analysis.

The outline for the paper is as follows. The next section discusses computer simulation as a modelling framework. We adopt a functional-network method that controls execution of components for discrete time steps. Our implementation uses a high-level specification language to express the contexts for running dynamic models. Section 3 explains the map algebra programming language and its extensions to handle flows. Section 4 describes aspects of hydrological models for infiltration, surface runoff, stream flow and erosion in conceptual terms. Section 5 presents our modelling implementation applied to a catchment, and presents qualitative modelling results for a storm event. Section 6 concludes by discussing how effective it was to express hydrological applications using map algebra language and modelling framework.

2. SIMULATION

This section discusses simulation and our use of a functional network as a modelling framework. Basic modelling formalisms have been described for discrete time, continuous, and discrete event systems. These differ by the way they handle dynamic changes in the model. Discrete time systems express state transitions through fixed time steps, continuous systems express changes through ordinary differential equations, and discrete event systems adopt a triggered execution mode where states change based upon previous states, conditions and inputs [Zeigler et al., 2000]. Discrete time systems are very appropriate for handling spatial dynamics. This is because change in environmental systems may be expressed using a stepwise model of execution over discrete time and spatial intervals (cells). A discrete time model for a single variable is shown in Figure 2. Systems may be decomposed into a set of coupled components. Specifications for system models define the issues of timing and state transition for components [Zeigler et al., 2000]. Figure 2 shows a set of components in a system model. A coupled component starts at a given event time and executes the model component for a specified time interval.

The coupled component framework may be ordered as a functional network. Components are represented as nodes with arcs defining transition dependencies. In our implementation a specification language is used to define the components and the topology for transitions. Similar components are allowed to share a common data space. Data and parameters may be defined in the specification. The specification language is written in the eXtensible Markup Language (XML). It bears some resemblance to the specifications used by other groups such as IMA [Villa, 2000] but currently is limited to a small subset of context semantics. This is because our main focus has been on dynamic analysis using simulations and spatial analysis.

The specification language allows users to define context parameters for dynamic aspects of the model. Timing defines the ordering for model execution. A time unit, such as 1 second, 1 minute, or 1 year, is chosen to correspond to each computational step covering the interval Δt. The process executes one computational step at a time with the system changing state from one instant to the next. There are many issues related to timing, such as initialisation, relation to physical time, quantisation, and synchronization between processes. The general syntax for our specification language is shown in Figure 3.

A model is composed of a set of modules which correspond to components. The only component types supported currently are MapScript and Excel. A universal clock defines the controlling parameters for execution and default time units. The functional network for modules is specified by a set of directed connections between named modules. Individual modules may define context parameters for spatial and dynamic analysis. This is currently limited to: (i) data parameters, (ii) timing parameters, and (iii) selections on spatial data.

```
<MODEL NAME="Main">
  <TIMER START=1 END=10 STEP=1 UNIT="hours"/>
  <MODULE NAME="a" COMPONENT="Excel"
    FILENAME="calc.xls" TIME_REFERENCE="A1" />
    <DATA NAME="k" REFERENCE="B1" />
  </MODULE>
  <MODULE NAME="b" COMPONENT="MapScript">
    <SELECT="REGION=Pine" />
    <DATA NAME="k" />
    <CODE>
      ... MapScript program ...
    </CODE>
  </MODULE>
  <NETWORK>
      <CONNECTION NAME="a_b" FROM="a" TO="b"/>
  </NETWORK>
</MODEL>
```

Figure 3. Syntax for the specification language.

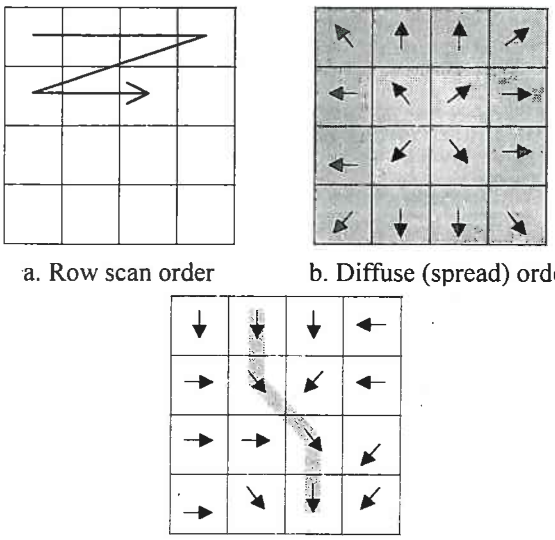
The modelling framework is implemented using an XML document. It is organised hierarchically as a tree view with cascading branches of *nodes*, and each node consists of an XML element. The XML editor is shown in Figure 6. Nodes can be expanded or collapsed, depending on whether or not the node has child nodes. Besides editing tasks, the only commands are to load and execute an XML document. Examples presented later in this paper will show working models.

3. MAPSCRIPT COMPONENT

Spatial models are expressed in a map algebra programming language. Map algebra is a function-oriented language that operates on four implicit spatial data types: local, neighbourhood, zonal, and whole landscape surfaces. It is used for tasks ranging from land suitability modelling to mineral exploration in the geosciences. Map algebra provides a high-level language interface to describe and manipulate surface data. Common examples of surface data include terrain models, categorical land cover maps, and scalar temperature surfaces.

Some implementations of map algebra allow models to be computed 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 where the order of computations has a special significance. For instance, processes that involve spreading or transport act 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 4 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 of as a medium.



a. Row scan order b. Diffuse (spread) order

c. Advective (flow) order

Figure 4. Spatial sweeping for processing grid map.

Our implementation of map algebra, called MapScript, includes a special iteration construct that supports these processing orders. MapScript is a lightweight language for processing grid-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. Determination of the type for a script variable is performed at run-time. MapScript supports basic arithmetic and relational comparison operators. The three basic language constructs are: functional assignment statement, iteration operator (*docell, doflow, dospread*) to iterate over cells in an image map, and an *if..else..end* conditional control program branching. Templates are supported with any arbitrary size and designation of a center. Basic operations on templates are also supported in a similar fashion to image algebras [Ritter et al., 1990], with the addition of template operations in cell iterations. Algebra operations on templates and a processing cell result in a new template, which is measured and assigned to a new processing cell. This is the typical way new cell values are computed from neighbouring cells using morphological operators, i.e. $y' = \mu(\Psi(x))$. Examples of the language are given in subsequent sections.

4. HYDROLOGICAL AND EROSION MODELLING

A model is an approximation of the actual system. Its inputs and outputs are variables which are linked by a set of equations. In the following three subsections three models are demonstrated using our modeling framework: a simple rainfall excess model, a runoff routing model, and a unit stream power based erosion/deposition model.

4.1 Constant Runoff Coefficient Model (RC)

Excess rainfall, or effective rainfall, is that portion of the rainfall which is neither retained on the land surface nor infiltrated into the soil. Under the assumption of Hortonian overland flow, excess rainfall becomes direct runoff.

In this model, the actual infiltration rate at the i[th] time step (f_i) is assumed to be proportional to the rainfall rate at the same time step (r_i) as:

$$f_i = (1 - R_c) r_i, \quad (1)$$

where R_c is runoff coefficient (i.e. runoff as a fraction of rainfall). The rainfall excess rate at the i[th] time step (e_i) is then estimated as the difference between rate of rainfall and infiltration rates, which is given by:

$$e_i = R_c r_i. \quad (2)$$

4.2 Runoff Routing

Once the rate of rainfall excess is determined using Equation (2), runoff rate at the i[th] time step (Q_i) is determined from [Yu, 2001]:

$$Q_i = \alpha\, Q_{i-1} + (1-\alpha)e_i \qquad (3)$$

in which α is related to the lag time (K) and the time interval used (Δt) as:

$$\alpha = \frac{K}{\Delta t + K} \qquad (4)$$

and the lag time is estimated from [Yu et al., 2001]:

$$K = 5/8\left(\frac{nL}{\sqrt{S}}\right)^{3\ 5} Q^{-2\ 5} \qquad (5)$$

where n is Manning's n ($m^{-1/3}$ s), L is slope length (m), and S is slope (m/m).

4.3 Erosion/Deposition Modelling with USPED Using GIS

Unit Stream Power based Erosion Deposition (USPED) [Mitasova et al., 1996] is a simple model, which predicts the spatial distribution of erosion and deposition rates for a steady state overland flow with uniform rainfall excess conditions for transport capacity limited case of erosion process. The model is based on the theory originally outlined by [Mitasova et al., 1996] with numerous improvements.

No experimental work was performed to develop parameters needed for USPED, therefore the USLE or RUSLE parameters are used to incorporate the impact of soil and cover and obtain at least a relative estimate of net erosion and deposition. It is assumed that sediment flow at sediment transport capacity can be estimated as:

$$T = RKCPA^m(\sin b)^n \qquad (6)$$

where RKCP are USLE factors, A is upslope contributing area (m^2), b is slope (deg), $m=1.6, and\ n=1.3$ for prevailing rill erosion while $m=n=1$ for prevailing sheet erosion.

Then the net erosion/deposition is estimated as:

$$ED = div(T \cdot a) \qquad (7)$$

where a is a gradient vector of the terrain surface, and div is the divergence which computes an indicative measure for loss or gain of sediment. If ED>0 the cell is an erosion source (more flows out than in) and if ED<0 then the cell is a deposition sink (more flows in than out).

Caution should be exercised when interpreting the results of Equation (7) because the USLE parameters were developed for simple plane fields and detachment limited erosion. Therefore, to obtain accurate quantitative predictions for complex terrain conditions, they need to be re-calibrated [Mitasova et al., 1996].

5. APPLICATION EXAMPLES

Both the simple constant runoff coefficient model and the erosion-deposition model have been implemented in MapScript.

5.1 Constant Runoff Coefficient Model

The model computes rainfall-runoff distribution – in a distributed sense – across a landscape. The rainfall data and observed runoff at the outlet are stored in an Excel file. Rainfall data is read from this file for each time step. The runoff model uses a simple formula based upon overland flow velocity. The MapScript component implements these formulae. Flow gains and losses are computed for cells in a flow order proceeding from higher cells to lower cells. The Excel and MapScript component are combined in a modelling framework using XML. The XML specification includes timing and shared parameter definition. Figure 5 shows the XML text code for the runoff module, and Figure 6 shows the module view in the XML editor.

```
<MODULE NAME="runoff" COMPONENT="MapScript">
  <DATA NAME="excess" />
  <CODE>
  // compute water depths
  doflow(dem)
     // accumulate inflows to cell
     gain = sum(inflows() * loss)
     // velocity for transmission across cell
     vel = pow(excess/(1000*60*60)*len,0.4) *
            pow(slope,0.3)/pow(n,0.6)
     // cell outflow rate given by velocity
     rate = (vel * 60 * step) / len
     loss = depth * rate
  enddo
  // net gain after event
  depth = depth - loss + gain + excess
  asVersion(depth)
  </CODE>
</MODULE>
```

Figure 5. Text view of net depth for rainfall-runoff model.

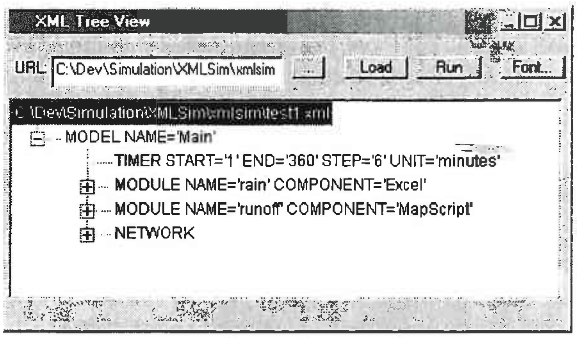


Figure 6. XML tree editor of the model.

5.2 Erosion/Deposition Model

The USPED computes erosion and deposition as an index, i.e. as an indicator of where this occurs in landscape. The script in Figure 7 shows how landscape values are computed from terrain slope derivatives, and Figure 8 shows output from model for an experimental plot in a grazing catchment.

```
// Constants
K = 0.5
C = 0.6
R = 120
// Model erosion-deposition from sediment transport capacity (USLE)
doflow(dem)
  accum = sum(inflows()) + sum(inflows() * accum)
enddo
flowtopo = pow(accum * 0.5,1.6) * pow(sin(slope(dem)),1.3) *K*C*R
erdep = divergence (dem, flowtopo )
```

Figure 7. Text view of erosion/deposition script.

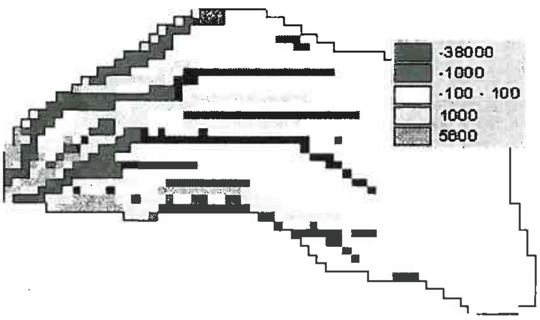


Figure 8. Map output of erosion/deposition model on experimental (hill slope) plot.

6. CONCLUSION

The paper has presented an application to integrate dynamic analysis and spatial analysis in a modelling framework, and to apply this to a hydrological application. A map algebra programming language, with special operators and constructs, is used for model surface flows. The program scripts are relatively easy to comprehend, especially when results are visualised. The scripts provide a high-level description of spatial analysis problems with flexibility for users to modify or write their own models. A secondary goal of the paper was to show how a map algebra language is used for dynamic analysis. Rather than augmenting the map algebra language we investigate its integration into a modelling framework to handle dynamics. The paper demonstrates this integration for hydrological models for surface and erosion modelling. The results of these models are presented qualitatively as maps. A future paper describes the application of these models to a small nested catchment.

7. REFERENCES

Burrough, P. A. Dynamic Modelling and Geocomputation, In: *Geocomputation - A Primer*, P. Longley et al. (eds.), John Wiley & Sons: 165-191, 1998.

Eastman, J. R., IDRISI for Windows Version 2.0, Clark University, Worcester, 1997.

Ford, A., *Modelling the Environment: An Introduction to Systems Dynamics Models of the Environment Systems*, Island Press, 401, Washington, D.C., 1999.

Mitasova, H., J. Hofierka, M. Zlocha and R. L. Iverson, Modeling topographic potential for erosion and deposition using GIS, *Int. Journal of Geographical Information Scienc, 10*(5), 629-641, 1996.

Pullar, D., MapScript: A Map Algebra Programming Language Incorporating Neighborhood Analysis, *GeoInformatica (In Press)*, 2001.

Reed, M., S. Cuddy and A. Rizzoli, A Framework for Modelling Multiple Resource Management Issues - an Open Modelling Approach, *Environmental Modelling and Software, 14*(6), 503-509, 1999.

Ritter, G., J. Wilson and J. Davidson, Image Algebra: An Overview, *Computer Vision, Graphics, and Image Processing, 4*, 297-331, 1990.

Tomlin, C. D. Cartographic Modelling. Geographical Information Systems. D. Maguire, M. Goodchild and D. Rhind. Essex, Longman Scientific & Technical. 361-374, 1991.

Villa, F., Design of multi-paradigm integrating modelling tools for ecological research, *Environmental. Modelling Software, 15*, 169-177, 2000.

Westervelt, J. and M. Shapiro. Combining Scientific Models into Mañagement Models. 4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4) CD-ROM, Banff, Canada, 2000.

Yu, B., C. W. Rose, C. C. A. Ciesiolka and U. Cakurs, The relationship between runoff rate and lag time and the effects of surface treatments at the plot scale, *Hydrological Sciences - Journal-des Science Hydrologiques, 45*(5), 709-725, 2001.

Zeigler, B. P., H. Praehofer and T. G. Kim, *Theory of Modeling and Simulation*, Academic Press, 2000.