

Review Article

The PCRaster Software and Course Materials for Teaching Numerical Modelling in the Environmental Sciences

Derek Karssenbergh, Peter A. Burrough, Raymond Sluiter
and Kor de Jong

*Utrecht Centre for Environment and Landscape Dynamics,
Faculty of Geographical Sciences,
Utrecht University*

Abstract

Teaching numerical modelling in the environmental sciences not only needs good software and course material but also an understanding of how to program the models in the computer. Conventional environmental modelling procedures require computer science and programming skills, which may detract from the important understanding of the environmental processes involved. An alternative strategy is to build a generic toolkit or modelling language that operates with concepts and operations that are familiar to the environmental scientist. PCRaster is such a spatio-temporal environmental modelling language developed at Utrecht University, the Netherlands. It is used for teaching modelling in classrooms and over the Web (distance learning) at three levels: (1) explaining environmental processes and models, where models with a fixed structure of model equations are evaluated by changing model parameters, (2) teaching model construction, where students learn to program spatial and temporal models with the language, and (3) teaching all phases of scientific modelling related to field research. So far, we have received positive responses to these courses, largely because the software provides a set of easily learned functions matching the conceptual thought processes of a geoscientist that can be used at all levels of teaching.

Address for correspondence: Derek Karssenbergh, Utrecht Centre for Environment and Landscape Dynamics, Faculty of Geographical Sciences, Utrecht University, PO Box 80115, 3508 TC Utrecht, The Netherlands. E-mail: d.karszenbergh@geog.uu.nl

1 Introduction

As in many areas of science, numerical modelling is a means of expressing our understanding of complex, dynamic processes, such as those operating in natural or semi-natural environments. Not so long ago, it was sufficient for environmental scientists merely to name the forms created by frequently occurring interactions of many processes – soil series, vegetation types, climatic zones: then classification was an aim in itself. Today, modelling has largely replaced classification as a scientific activity as scientists attempt to find better and more quantitative explanations of how complex environmental systems work. In this respect, creating a numerical model is one way of expressing our scientific understanding in a communicable form. It is therefore completely natural, and important, that we should want to teach numerical modelling to our students.

Environmental systems are mostly open systems with many, poorly definable interactions between hard to describe processes. These complicated systems having both spatial and temporal variation at several levels of resolution are difficult to reproduce in the laboratory. Since analogue modelling requires large scale models, it is generally not feasible, especially for teaching, so numerical methods are currently most used for modelling environmental systems. Whereas engineering skills and tools are needed to build an analogue model, mathematical and computer programming skills and tools are necessary to build numerical models. Today, many environmental scientists have access to powerful computers, but they are not necessarily able to program them. Until recently, the creation of complex environmental models has required the efforts of teams of computer scientists – think of MODFLOW (McDonald and Harbaugh 1984) and related pollutant transport codes. This is mainly because such models need complex numerical algorithms, programmed by specialist programmers in system programming languages such as C or FORTRAN. For teaching environmental model construction, the use of system programming languages for constructing a model presents difficulties because many students in the geosciences are not skilled in programming computers. Instead, a modelling language is needed that matches the thinking level of an environmental researcher or geoscientist who thinks in terms of higher level functions that may be used to simulate real world processes.

A second issue in education is the computing load. It is not uncommon for some environmental models to require super-computers for their operation. While this is appropriate for highly specific problems, such methods are out of the question for routine modelling and instruction in class (Casti 1998). Clearly, there is a need for a generic environmental modelling language that operates in a way that is easily understandable for students and environmental scientists providing flexible, modular model construction on affordable computers. PCRaster is such an environmental modelling language (Van Deursen 1995; Wesseling et al. 1996). It is raster-based and supports construction of both static GIS-like cartographic models and dynamic spatio-temporal models (including cellular automata functions) using a modelling syntax similar to mathematical notation.

This paper explains how the PCRaster software may be used for teaching numerical modelling in a classroom situation and for distance learning (Web based teaching). At Utrecht University, course material has been written for both kinds of teaching environments, which supports individual learning without too much direct contact with the tutor. This absence of direct assistance by a tutor is crucial in Web

based teaching since learning over the Web does not allow direct assistance from a tutor. Instead, support has to be provided by the course material itself. The paper describes how this automatic assistance is provided as an integral part of the course material.

The Utrecht modelling curriculum has a learning path that starts with map overlay operations and static cartographic modelling (c.f. Burrough and McDonnell 1998), and simple numerical modelling for the construction and application of spatio-temporal environmental models (Burrough 1998, Wesseling et al. 1996). Geoscientific fields that use such spatio-temporal modelling include hydrology, ecology, pedology, sedimentology, land degradation research and geomorphology. The students are undergraduate, or postgraduate students in these sciences; they are usually unfamiliar with programming, but most have some basic knowledge of GIS. At the MSc level and above, students are offered courses in using PCRaster to construct models of environmental processes that they encountered in their major fieldwork course. One of the main principles behind all the material is that it should stimulate students to construct conceptual frameworks that can support the development of their own models, thereby finding solutions for their own specific problems.

2 The PCRaster Environmental Modelling Software

The effort and degree of specialism needed to create powerful environmental models has led to model building being seen as a scientific activity in its own right. The products of this work – the models – may be completely understood by their creators, but others often see them as ‘black box’ units in a data processing system. Such models are frequently linked to GIS as sources of data, and as a means of displaying the results or for combining them with other spatial information.

This loose coupling of GIS and off-the-shelf models may be ideal for many environmental scientists working in advisory or dedicated research institutes but it poses several problems for those wishing to experiment with modelling, or to teach it to university and post-academic students. These problems include:

- The aforementioned need for programming skills in a system programming language such as FORTRAN or C++
- A lack of insight into how mathematical representations of processes are expressed in algorithms in the standard models
- The difficulties or impossibilities of easily modifying the model code to explore new ways of modelling the process being studied
- A complete absence of a generic approach (apart from the mathematical basis) to dynamic spatial modelling because each program is written by a different team under different circumstances
- The need for means of linking models with spatial and temporal data in a GIS

In recent years there have been developments in both computational mathematics and GIS that provide answers to these problems. The first edition of the ‘Numerical Recipes’ books (Press et al. 1986) was a major advance that made many standard algorithms available to those without knowledge of numerical mathematics. But modellers still needed to be programmers, since the algorithms were still written in system programming languages. The main disadvantage of model construction in a

system programming language, however, is that all technicalities of a model have to be defined in the code, resulting in difficult to handle model programs. So there is a need for a higher level programming language with standard easy to use functions at the level of thinking of a modeller, that enables the representation of real world processes. This line of reasoning has been followed by programs such as Matlab (2000) and STELLA (2000) in which the user can set up and solve equations without any overt programming skills. Such programs are in effect higher level programming languages that operate with an interface at the level of scientific or technical communication.

In GIS there is a long tradition of using command languages for stringing together sets of well-defined operations to achieve given results. The command syntax of such languages is frequently of the form:

$$\text{Newmap} = \text{operation}(\text{Inputmap}_{1..n})$$

in which the resulting map is created by a certain operation on one or more input maps. The **operation** is the name of a generic process that the user understands at his or her level of working. The operation could be that of calculating slope from an elevation map, or of determining proximity zones to linear features such as roads or rivers (c.f. Burrough and McDonnell 1998), so the 'language' is very easy to use. By linking these operations together one can create what are in effect 'models', albeit static models of spatial patterns that are the 'bread-and-butter' of much GIS analysis.

This concept of higher level operations has been extended in PCRaster to include not just spatial operations but also temporal iterations. In addition, the set of spatial functions in the language is increased to cover many standard operations encountered in environmental studies. The result is a mathematical space-time modelling language that gives many advantages for environmental modelling and those wishing to teach it. Data entities in PCRaster are objects such as stacks of raster maps for spatio-temporal attributes, time series for temporal non-spatial data and lookup tables. The language contains 125 generic functions operating on these entities. Functions included are mathematically defined non-spatial (point) operations, network creation, transport and flow operations and window operations, as well as spatio-temporal time operations for reading and writing temporal data, e.g. hydrographs at specific locations. A GIS-like database and visualization software is integrated in the system.

3 Levels of Teaching

PCRaster is used at several levels in teaching environmental modelling:

- As a means of explaining environmental processes and models
- As a modelling language in its own right and as a means of teaching model construction
- As a means of teaching all phases of scientific modelling needed for field research.

3.1 *Explaining Environmental Processes and Models*

At the introductory level, students do not yet have the skills needed for constructing models. Teaching at this level has two aims. First, it is important to give students insight in processes occurring in the natural world. In a lecturing environment, models

can be used as virtual landscapes illustrating processes occurring in a real landscape and they supplement – but not replace! – field excursions and field studies. Model animations have been shown to be very useful for illustrating processes that are too slow or imperceptible to experience in the field, e.g. tectonic uplift or groundwater flow. In addition, models are useful for illustrating different scenarios of landscape evolution, for example, exploring the degrees of sedimentation and erosion that may occur if a dam is built.

Secondly, at a somewhat higher level, students can run models themselves by modifying certain model parameters through a graphical user interface (GUI). By clicking on the buttons in the user interface, the user invokes the simulation software under user defined scenarios or the standard visualization routines of PCRaster. This can be done without direct support of a lecturer, if good course material is provided. This helps students understand how landscape processes are represented by environmental models and how model components interact.

One of the Utrecht courses at this introductory level covers the long-term effects of slope processes on slope profile form. The temporal change in elevation for each location on a hillslope is a result of interaction between processes of weathering and supply or removal of material over the slope, driven by several factors, see Figure 1 (cf. Ahnert 1987). A PCRaster model simulating the development of a mountain range in geological times is used to explain these often difficult to understand interactions (Plate 1, see plate section). The student learns the basic principles of hill slope processes by changing the input parameters in the interface, such as base level lowering, levels of creep and wash, and comparing animated maps and time series for different scenarios. A course text is written that stimulates students to learn by ‘playing’ with the model.

An advantage of using PCRaster at this level of teaching is that lecturers can quickly and easily make models with student interfaces that will run on personal computers. PCRaster automatically generates the GUI from a PCRaster modelling script written by the lecturer, without extra programming. As a result, development

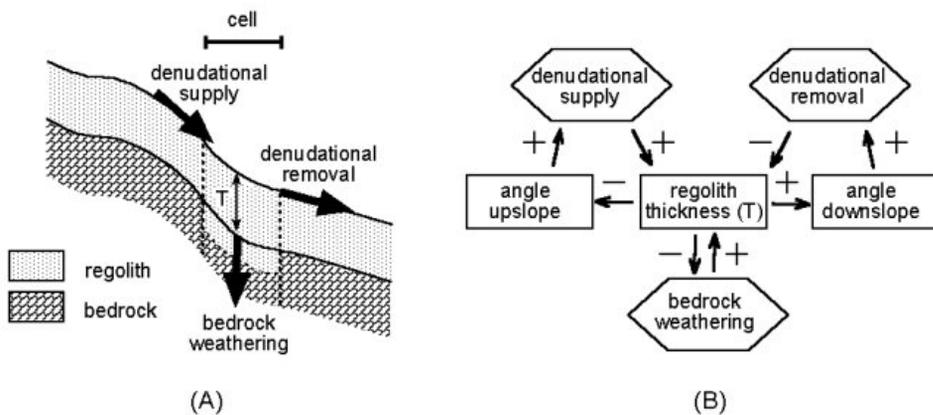


Figure 1 Factors and processes determining the rate of bedrock weathering (i.e. denudation) at a grid cell on a hillslope, (A) profile with processes, (B) positive and negative feedback loops in the cell (After Ahnert 1987)

times of models are short, and models can be tailored to a specific lecture or theme. In addition, evaluating models by interactively changing model parameters requires the students to think, and inspires them to make their own models.

3.2 *Teaching Model Construction*

At this second level, students are taught how to construct simple models using PCRaster. They learn how the language is used and construct models step by step. The course text gives the basic theory behind the models and explains how the models may be constructed with PCRaster. This level is taught in two phases: static modelling and dynamic modelling.

3.2.1 *Static Modelling*

Static modelling involves modelling in space with single PCRaster operations or combinations of operations in script files. The student learns the ideas behind the different groups of (non-spatial) point operations and spatial operations by applying them to a case study data set. Initially, students work from the command line, by typing single operations with a syntax similar to mathematical notation. For instance, runoff during a rainstorm is calculated by:

$$\text{pcrcalc RunoffMap} = \text{accuthresholdflux}(\text{LddMap}, \text{RainMap}, \text{InfilMap})$$

where LddMap contains the flow directions, and RainMap, InfilMap contain the amount of rain and potential infiltration respectively, under the assumption of steady state conditions (van Deursen 1995, Wesseling et al. 1996, Burrough 1998 – Plate 2, see plate section). In the next phase, students have to solve case study problems by combining operations in a PCRaster script file. Here it is not just the meaning of the single operation that counts, but also how functions can be combined, where the output of an operation can be the input of the next operation. Examples of problems which may be solved are finding the location of a sports fields complex in an urban area or finding the optimal route for a road through a mountainous terrain.

3.2.2 *Dynamic Modelling*

Dynamic modelling builds upon static modelling with script files that include the time component. The construction of dynamic models is accomplished in a PCRaster script file structured in different sections containing PCRaster operations written using the same syntax as in static modelling (Figure 2). Operations in the initial section define the initial state of the model at time step = 0. In an event-based runoff model, the initial section could define the initial soil moisture content, derived by crossing a soil type map with a lookup table of soil types and their hydrologic characteristics. The temporal behaviour is defined in the dynamic section. The same set of operations in this section is carried out for each time step. The results from the previous time step are the input for the next time step. For example, in a runoff model, the dynamic section would contain an iterative operation that for each time step and each cell, adds the amount of infiltrated water to the soil moisture content. This soil moisture map could be used in the next time step for calculating potential infiltration. Additionally, the dynamic section contains time operations that read temporal data such as rain time series or cloud cover maps from the database. Storing results is simply done by adding a 'report' keyword to operations that should be stored in the database. The results can

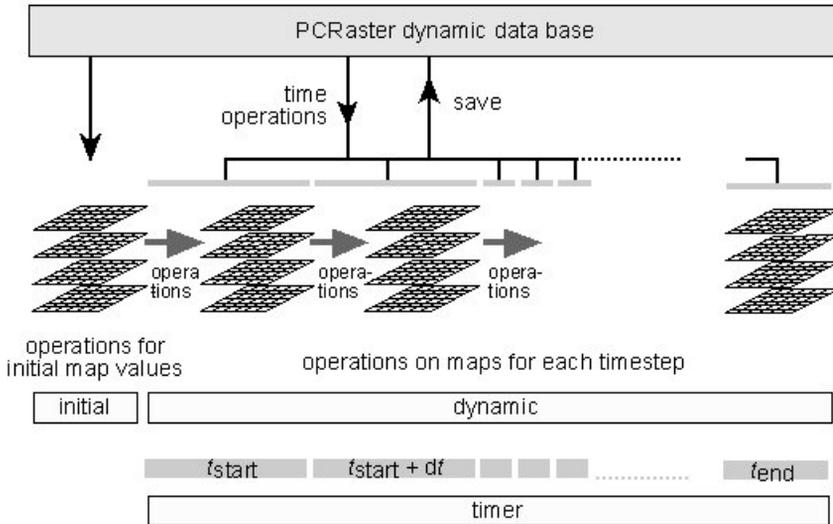


Figure 2 Concept of a dynamic modelling script with initial, dynamic and timer sections

be visualized as 4D (3D plus time) displays (Plate 3, see plate section). The current course material includes exercises for constructing large scale catchment models, event based runoff modelling, plant growth and dispersion modelling and pollutant dispersion modelling.

Our experience in Utrecht with these PCRaster-based courses suggest that PCRaster is an ideal environment for teaching model construction because:

- Its generic approach to modelling using a wide range of standard algorithms and functions avoids having to re-invent the wheel
- There is no need for students to learn a programming language such as FORTRAN or C++ before they start to think quantitatively about landscape change: if they wish, they can do this in a later stage of their study
- The modelling language is similar to that of scientific discourse – therefore it is easy to learn
- Its text-based interface supports both simple static modelling with single commands and the construction of space-time models with the same set of operations
- The language provides a means to embed modelling in the GIS rather than create once-off or clumsy links between models and GIS
- The language is portable and can be run on different platforms, including currently available, cheap PCs

As a result, the learning path is fairly short: after two weeks students are able to use the language without too many problems and are ready for the next stage of environmental modelling: the construction of models according to model concepts tailored to their own field research.

3.3 *Teaching All Phases of Scientific Modelling Related to Field Research*

At this level, equivalent to MSc and above, students perform a complete study by combining data gathered at their own field study site and knowledge of environmental processes with modelling. Their modelling study includes all phases of scientific modelling:

- identification of the structure of a model based upon the field study problem and available data, i.e. choice of processes to be simulated, mathematical representation of these processes, and the levels of temporal and spatial resolution,
- implementation of this model with PCRaster,
- calibration and validation with their own field data, and
- presentation of the results.

Students get some assistance from their tutor, especially in the first phase of model structure identification, but most of the work is accomplished without much support. Some studies performed by undergraduate students include event-based runoff and erosion modelling studies, water balance, vegetation growth and grazing modelling studies, heat balance modelling studies using remote sensing, and modelling of geomorphological processes.

Unlike the use of standard models in student research, requiring students to construct their own models has the advantage that they have to identify model structures themselves, which forces them to understand a landscape in a numerical form. In addition, students are directly confronted with the usual problems of modelling such as incorrect model structures, implementation errors, the effects of spatial and temporal resolution, and parameter identification problems, since they are responsible for everything. This responsibility is also the reason for their appreciation of this work: it is much more interesting and challenging to construct your own model, using your own field data, than pressing buttons in a model written by somebody else. A disadvantage of students constructing their own models is that they may use inappropriate concepts and methods, but this can be prevented by good support from the tutor. Another disadvantage is that the restricted set of standard functions in PCRaster limits the range of models, but this can usually be solved by programming extra functions in C++, which are called up by the plug-in function of PCRaster. This may help to give students some knowledge of system programming as well.

4 Courses in Classrooms and Distance Learning

4.1 *Structuring the Course Material*

Our course material for teaching in a classroom and over the internet includes a software manual and a course text for teaching different levels of environmental modelling. We have developed the material in a structured way resulting in a database with all written material based on the Extensible Markup Language (XML 2000). This approach has several advantages over developing course material as separate entities:

- Modularity and reuse of pieces of text. As the same operations may be repeated in different courses or different parts of a course, standard texts have been written

explaining these operations which can be included at any location in the course material. This prevents double work.

- Links between courses. By using one structured database of all written material, it is easy to include and maintain hyperlinks within or between different courses and the user manual. This increases the accessibility of the specific information a student needs while making the exercises.
- Recombining courses. Post-graduate students need courses tailored to their individual needs. A structured database allows course material to be extracted to meet specific learning paths.
- Standard output format for digital and hardcopy material. A standard style sheet is used after the text has been written resulting in a standard format for all material. In addition, it is possible to derive both digital (e.g. HTML) and hardcopy output from the same course text, with a standard format.

4.2 Distance Learning

People working at government institutes and private firms may have little chance to spend periods away from home to study. These people cannot attend courses at a university campus but need courses that can be followed part time. Such on-job training courses can be provided by web based training, or distance learning.

Distance learning courses are different from courses on the campus in the sense that apart from an initial seminar, there is little direct contact between the tutor and the student. Instead, all contact is through the internet (formerly the mail), mostly with time lags. PCRaster has been incorporated in a distance learning course system that is currently developed in the EU-sponsored MUTATE bundle (MUTATE 2000). The course database contains course content in XML, written by multiple authors, and a course management tool managing and storing all the data per student, such as subscription details per module, and scores on exercises (Figure 3). The students retrieve and navigate through the course material on a WWW server, over the internet. Since spatio-temporal data sets may be too large to be sent over the internet, PCRaster runs with datasets on the student's computer.

Since there is little direct contact between the student and tutor, the system provides alternative ways for student evaluation and support. The course material includes multiple choice, numerical and keyword questions. Students enter their answers in the digital course material and get an automatic feedback on their answer providing additional explanation, and a score, which is also used for final evaluation of the student by the tutor. This automatic evaluation is possible since the course material contains algorithms to compare students' answers with correct answers available in the course data base. In addition, open questions are given for more complicated problems, which are evaluated by the tutor. The system also stores a log of all actions of the student with the PCRaster software in the database. This information can be used by the tutor to check what the student does and provides a good reference for communication between the tutor and the student if the student has problems solving the exercises.

For courses at the second and third level that teach the construction of models, support from the tutor is very important, since students may easily get stranded while constructing a model. In a classroom situation, this support is provided by a tutor in the classroom. Distance learning needs alternative strategies to provide comparable support. Strategies followed in the PCRaster course material are:

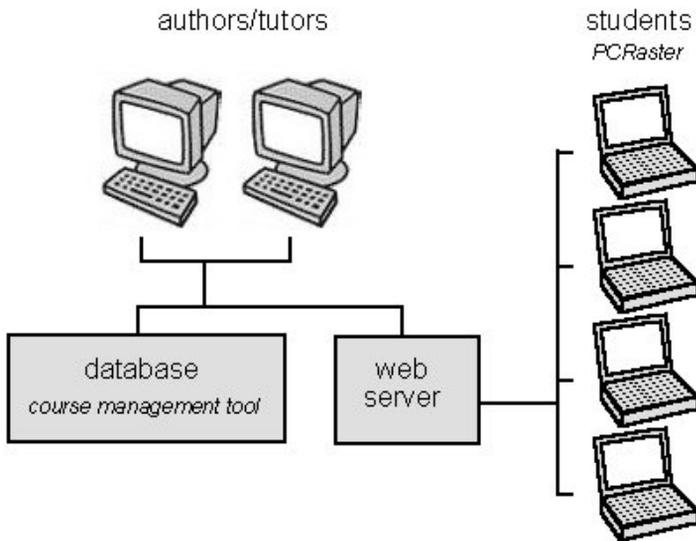


Figure 3 Distance learning environment.

- Provide the student with background knowledge at the start of each module. This is done by referring to pages in standard text books, links to the PCRaster manual and internet sites.
- Provide links to (parts of) courses, which the student has already done, to freshen up the skills needed for a specific exercise.
- Provide advice on the basis of answers to questions, which are automatically evaluated. If the student gives the wrong answer, the system explains why this answer is wrong and provides (parts of) the correct solution.
- Provide optional links to hints giving additional information that should help the student solve the problem.
- Provide methods for contacting other students, such as bulletin boards or mailing lists.
- Promote contact with the tutor via email and attachments containing all additional information (model scripts, commands) that may help the tutor to support the student.

5 Discussion and Future Work

Although the existing software and course content provides modelling courses in a wide range of fields, PCRaster does not support all types of modelling which could be the subject of teaching in one of the geosciences. Geostatistical modelling, interpolation and simulation is currently supported by the Gstat package (Pebesma and Wesseling 1998), but this package is only loosely linked to PCRaster through the same raster file structure. A useful combination of modelling and geostatistics would provide methods for error propagation in spatial-temporal models, assuming spatially correlated errors

in the inputs. Theory for error propagation modelling in GIS has been described in standard text books (e.g. Burrough and McDonnell 1998, Heuvelink 1998) but tools, which are easy to use for environmental modellers, are not generally available. The development of such a tool in PCRaster would need embedded linking of the Gstat random field simulation software with the PCRaster environmental modelling software (cf. Karssenberget al. 2000, Pebesma et al. 2000).

The disadvantage that PCRaster is only a 2.5D system is currently being worked on by extending the modelling language with 3D entities and functions (Karssenberget al. 2000). Whether the requirement for cheap computing will be met has yet to be seen, but given the still increasing processing power we expect that 3D methods will be soon available for both the classroom and research.

Acknowledgements

The authors wish to thank our colleagues in the PCRaster research team: Willem van Deursen, Cees Wesseling (PCRaster Environmental Software), Marlous van der Meer, Arko Lucieer (Utrecht University), colleagues of the Faculty of Geographical Sciences, Utrecht University and many other people who supported the development of PCRaster and associated course material through the years.

References

- Ahnert F 1987 Process-response models of denudation at different spatial scales. *Catena Supplement* 10: 31–50
- Burrough P A 1998 Dynamic modelling and GIS. In Longley P A, Brooks S M, McDonnell R and MacMillan B (Eds) *Geocomputation: A Primer*. Wiley: 165–92
- Burrough P A and McDonnell R A 1998 *Principles of Geographical Information Systems*. Oxford, Oxford University Press
- Casti J L 1998 *Would-Be Worlds: How Simulation Is Changing the Frontiers of Science*. New York, Wiley
- Heuvelink G B M 1998 *Error Propagation in Environmental Modelling with GIS*. London, Taylor and Francis
- Karssenberget D, de Jong K and Burrough P A 2000 A prototype computer language for environmental modelling in the temporal, 3D spatial and stochastic dimension. In WWW document, <http://www.colorado.edu/research/cires/banff> (Paper presented at the *Fourth International Conference on Integrating GIS and Environmental Modelling*, September 2–8, Banff, Canada)
- Matlab 2000 MATLAB®6. WWW document, <http://www.mathworks.com/products/matlab/>
- McDonald M G and Harbaugh A W 1984 *A Modular Three-dimensional Finite-difference Ground-water Flow Model*. Reston, VI, US Geological Survey
- MUTATE 2000. Multimedia Tools for Advanced GIS Training in Europe. WWW document, <http://mutate.chiron.pt/>
- Pebesma E and Wesseling C G 1998 Gstat: A program for geostatistical modelling, prediction and simulation. *Computers and Geosciences* 24 17–31
- Pebesma E J, Karssenberget D, and de Jong K 2001 The stochastic dimension in a dynamic GIS. In *Proceedings of Compstat 2000, The Fourteenth Conference of the International Association for Statistical Computing*, 21–25 August, Utrecht, the Netherlands
- Press W H, Teukolsky S A, Vetterling, W T and Flannery B P 1986 *Numerical Recipes in Fortran 77; The Art of Scientific Computing* (First Edition). Cambridge, Cambridge University Press
- STELLA 2000 HPS: The Systems Thinking Company. WWW document, <http://www.hps-inc.com/edu/stella/stella.html>

- Van Deursen W P A 1995 *Geographical Information Systems and Dynamic Models*. Ph.D. thesis, Utrecht University, the Netherlands, NGS Publication 190, 198 pp
- Wesseling C G, Karssenbergh D, Van Deursen W P A and Burrough P A 1996 Integrating dynamic environmental models in GIS: The development of a Dynamic Modelling language. *Transactions in GIS* 1: 40–8
- XML 2000 The XML Industry Portal. WWW document, <http://www.xml.org>