24 Simulation and Reduced Complexity Models

James D.A. Millington

SYNOPSIS

The process of simulation modelling iterates through system conceptualization, data collection, model construction, evaluation, and model use, demanding continual reflection on the part of the modeller. Simulation models, a product of the simulation modelling process, are *representations of reality* that couple theory with data to dynamically represent processes, interactions and feedbacks (potentially across space). For example, spatial simulation models allow examination of feedbacks between spatial patterns and processes. Advances in computing mean that many modelling environments and programming languages/libraries are now available which enable relatively quick production of simple simulation models for investigating geographical systems. This chapter includes examples of how to use these readily available tools, providing an introduction to simulation modelling for geographers with a focus on Reduced Complexity and Agent-Based Models (ABM). The importance of *conceptualization* of key processes to represent in a model, and careful consideration for how they should be *operationalized in code* (via equations and/or rules), is also discussed.

This chapter is organized into the following sections:

- Introduction: Conceptualizing the geographical world
- Reduced complexity models in physical geography
- Simulating pattern-process feedbacks
- Developing your simulation model

24.1 Introduction: Conceptualizing the Geographical World

You may have read this chapter before, but repetition with new initial conditions is often a useful way to explore and understand a concept or system (although deciding when to stop and do something else is also important!). In trying to understand the geographical world, we often resort to simplified representations of it that we can experiment with. These simplified representations are often called models, and include the conceptual models we carry around in our brains and that enable us to navigate our way through the world without needing to observe and comprehend precisely every detail of what is going on around us. For example, imagine you are walking along a busy city street. Your understanding about how humans move and interact with one another (based on previous observations of other people's movements), combined with observations of the locations and rates of movement of the people on the particular street you are on, provides a mental model that allows your brain to anticipate the dynamics of the crowd and help you to avoid collisions. In a similar way, more formal simplified representations of the world can help us to understand and anticipate dynamics of physical systems over geographical scales that are harder to comprehend and observe given human sensory and cognitive capacities. For instance, in computer-simulation modelling, a conceptualization of target phenomena – such as erosion and deposition in a landscape over several hundred years – is specified in computer code that is then executed in a computer (i.e. simulated) to produce output that allows the logical consequences of the conceptualization to be examined. Whereas walking down the street your brain combines your conceptual model of human movement (e.g. walking humans move at about 5 km/hr, on the horizontal plane and not vertically, etc.) with observations of individual people to anticipate the dynamics of a particular crowd, a computer simulation of landscape evolution might combine a mathematical model of erosion and deposition with input data for a particular landscape to anticipate landscape dynamics. Beyond the difference in scales of processes represented by each (seconds and minutes walking down the street, years and decades in the landscape model), a key difference between our brains using mental models and computers simulating formal models is that we can use the computer to simulate many instances of the same system to explore how different input data and other variations produce different outcomes. In the real world we only get one chance to walk down a given street at any given point in history (and who knows who we'll bump into), but by representing pedestrians on the street in a computer simulation (e.g. Torrens, 2012) we might explore, for example, how different initial conditions (e.g., people starting in different places or moving at different speeds) results in different flows of pedestrians. Similarly, computer simulation model models of landscape evolution offer the chance to examine how different initial landscape states result in different patterns and trajectories of change (e.g. Wainwright, 2008). Asking such 'what if…' questions with simulation models is not restricted only to initial conditions, but can also be used to investigate alternative representations of processes (what if some pedestrians *want* to bump into another? How do different vegetation species influence sediment deposition differently?), and through repeated use, experimentation and reflection, simulation modelling allow us to improve our understanding of the world being represented.

Computer simulation models and modelling differ from other forms of model and modelling discussed elsewhere in this book. Quantitative statistical models (see Chapters 19, 28) make assumptions about the form of relationships between measured variables and then estimate parameters (numeric values) to describe those relationships. As such, although statistical models can quantify relationships in observed measurements they are heavily dependent on data and which do not necessarily represent dynamics of change well. In contrast, analytic models are driven primarily by theory and represent system dynamics and change through mathematical equations or expressions (e.g. based on differential equations). Although these models allow exact solutions based on their formulation (and therefore clear links to underlying theory), those formulations (and therefore the theory represented) is heavily constrained by mathematical possibilities and therefore may not represent as many influences on dynamics as might be desired. The advent of digital computers and the growth of their processing power now provides an alternative to these two older approaches (which could feasibly be executed by hand and brain power alone). Computer simulations models are able to use statistical relationship or mathematical expressions to represent the world, but can also use other means that have less restrictive assumptions. For example, analytic and statistical models tend to simplify representation of the world by aggregating system elements (people, animals, grains of sand etc.), assuming individuals are identical and uniform (analytic) and/or by representing them with population-level summaries (statistical). In contrast, computer simulation models now allow representation of the dynamics of interactions between disaggregated, differentiated and discrete individual elements (Bithell et al., 2008). Such discrete-element techniques are increasingly being applied to better understand how broad, general patterns in environmental and social systems are generated as a result of specific, local interactions between individual elements. Although the appropriate use of models in geography has a long history, the advent of discrete-element approaches only brings yet more questions and debate about their possibilities and limits, how they should be used (e.g. for prediction or for understanding?), and how their output can and should be interpreted (Clifford, 2008; Millington et al., 2012).

Although the discrete representation of system elements is possible in a computer simulation model (but not required), another form of *discretization* is always necessary. Whereas relationships found or expressed by equations are usually continuous in nature (e.g. differential equations describe rates of change between theoretically infinitesimal intervals) the digital nature of computers means that models implemented using them requires discretization *of time and space*. Space must be split into discrete areas that are assumed to be internally homogeneous in all characteristics (e.g. a lattice or grid of cells), and time must be split up into discrete steps ('timesteps', in which change is assumed to happen between but not within). Consequently, to represent change and dynamics over space and time, computer simulation requires the same calculations be repeated over and over again for each of these discrete chunks of space and/or time in order that interactions and change between them be represented. Each repeated calculation is known as an *iteration* and demands that to represent change in time and space, the code that the computer executes must be written so that it loops over on itself (Box 24.1).

Box 24.1 Loops and NetLogo

As we'll see throughout this chapter, 'loops' are important for representing change across space and through time in computer simulation. This chapter also challenges you to develop and use computer simulation models yourself to understand geographical systems, and provides example models for you to explore in the freely available modelling software NetLogo (Wilensky, 1999). As your first challenge, see if you can understand and implement the NetLogo code shown below. Download NetLogo [from http://ccl.northwestern.edu/netlogo/], install it on your computer (Windows or Mac), open the NetLogo programme and then type the code below into the Code tab (make sure it is *exactly* the same as shown above).

```
to go                                                  ;; line 1
  let population 4                                     ;; line 2
  let growth-rate 2                                    ;; line 3
  while [population < 1000]                            ;; line 4
   [                                                   ;; line 5
    set population (population * growth-rate)          ;; line 6
    print population                                   ;; line 7
   ]                                                   ;; line 8
end                                                    ;; line 9
```

This code demonstrates how a 'while' loop can be used to simulate the exponential growth of a population that starts with a size of 2 individuals (where in the code is this initial value specified?). To execute the code in NetLogo (after you've typed it in), go to the Interface tab and type 'go' at the bottom of the screen where is says 'observer>', then hit enter. You should see the value of the population as it grows; each timestep printed on the screen (if not check you have typed the code correctly; computers are stupid and will only do exactly what you tell them so make sure the code is correct!).

Here's what the computer does when it reads the code. First, the size of the population is checked (line 4). If less than 1000, the code between the second set of square brackets will be executed (lines 5–8). The size of the population is checked again (line 4). If the less than 1000, the code between the second set of square brackets will be executed again. This continues until the population is no longer less than 1000 (i.e. while the expression population < 1000 is true). In each iteration of the loop, population doubles (lines 3 and 6) and the current value of the population is printed (line 7). Inherently in this code we assume that each iteration of the loop is an advance in time of one unit (i.e. a single 'timestep'). How many timesteps are simulated (and why)? What is the last population value printed (and why)? Loops can work equally well across discrete areas of space and can also be nested within themselves so that they self-repeat (think what you would see if you held up two mirrors facing one another). This concept of nested loops – known as recursion – is often used in computer programming. Once you've tried the code above, have a run through the tutorials that come with NetLogo to learn more about how to use this flexible modelling environment (in Netlogo go to the Help menu then 'NetLogo Manual' then click Tutorial #1 on the left).

The idea of loops and looping seems to pervade simulation modelling and we'll see three types of loop in this chapter. First, are the loops of computer code that execute the same commands or calculations over and over again (Box 24.1) and which are well visualised through flow charts (as we will see below; see also Figure 24.4). Second are feedback loops in the real-world geographical systems that we might aim to represent with our models, which we will consider those further in Box 24.3. The third loop in computer simulation we will consider here is the process of modelling itself, from model conceptualization, through data collection, model construction, model evaluation and model use. We will look at how that first type of loop is related to the other two loops in the final section of the chapter in which provides advice for how you might go about using and developing computer simulation models yourself. Before those sections however, we'll consider in more detail one particular type of simulation model used in physical geography.

24.2 Reduced Complexity Models in Physical Geography

Reduced complexity models (RCMs) are simulations models used by geomorphologists to represent processes and change at 'intermediate' scales, generally 1s – 100s km^2 in extent and 10s – 100s years in duration. Such scales are intermediate between finer scale representations of physically-based models to understand processes of fluvial sediment transport and deposition, and broader scale representation of landscape evolution models to understand longer term impacts of climate and land-use change on channel dynamics (Brasington and Richards, 2007). These intermediate scales are also those that are potentially most useful, for example, to river and environmental managers (Stott, 2010). All computer simulation modelling of physical systems must negotiate the trade-off between representational detail on the one hand and the computational resources needed to simulate that detail across time and space on the other. The trajectory of much simulation modelling in geomorphology and hydrology through the late twentieth century followed the reductionist-deterministic perspective towards ever-more detailed and smaller-scale quantitative representation of physical processes using empirical or theoretical relationships in the form of mathematical equations (e.g. Navier–Stokes equations, see Reddy, 2011). In these physically-based models, increased representational detail demands increased numbers of equations, parameters and calculations, in-turn requiring increased computational resources. If larger spatial extents or temporal durations are to be simulated (e.g. up to the 100s km^2 and 1000s of years of 'intermediate' scales), computational resources increase yet further. Despite continuing rapid increases in computational power in the late twentieth and early twenty-first centuries, the need and motivation to investigate processes and change at 'intermediate' scales required a new approach with a reduced level of representational detail commensurate with available resources. Hence, the growth in use of 'reduced complexity' models with their relatively more simple representation of the laws of physics. For example, by relaxing some assumptions of equations determining fluid flow, fluvial geomorphological RCMs are able to provide rapid solutions to calculations of water depth and velocities (Coulthard et al., 2007).

Simplified equations are only one aspect of RCMs however, the other being the adoption of lattice structures (e.g., grids) to represent space discretely (e.g. Figure 24.1). Using a lattice of discrete elements ('cells'), each of which corresponds to some area of land (which is assumed to be internally homogenous), a more general or holistic representation of catchments and landscapes is possible. For example, while many hydrological and geomorphological modellers were developing ever-more refined equations to represent processes at finer scales, other geomorphologists interested in how landscapes are shaped over long time periods and large spatial extents (many thousands of years and hundreds of square kilometres) have used landscape evolution models (LEMs; Tucker and Hancock, 2010). To investigate how the shape of the landscape changes – due to the entrainment, transport and deposition of sediment as water flows from points of higher elevation to lower – LEMs use rules to determine which route water (and sediment) takes as it moves from one cell to another adjacent cell. Similarly, aeolian geomorphologists have experimented with the use of 'cellular automata' type approaches, in which each cell is assumed to be in a discrete state, to simulate creation of dune formations (e.g. Baas, 2002). By exploiting the computational efficiencies of a lattice structure, albeit with a finer spatial resolution (i.e. cells represent smaller areas of land surface), and combining it with simpler versions of the equations used in fine-scale physically-based models or entirely different conceptual abstractions (e.g. 'slabs' of sand instead of individual grains, see DECAL model below), RCMs are able to efficiently represent processes and change at intermediate scales. The lattice or cellular structure of these models means that they are often referred to as cellular models, while their simplified representation of physical processes means that they are often seen as useful for explanation or exploration (i.e. examining 'what if…' scenarios for management) rather than prediction (where physically-based models offer greater precision). In future, combinations of different types of models, exploiting the different strengths of the various approaches may become more prevalent (Nicholas et al., 2012; see Box 24.2).

Possibly the 'original' RCM was developed to investigate river form and process feedbacks in fluvial geomorphology (Murray and Paola, 1994). One of the primary assumptions of this model, and an important example of representational simplification, is that water is assumed to flow only in a pre-defined downstream direction (i.e. no eddies back upstream will be represented). This simplification allows iterative application of rules that determine water and sediment movement between cells, starting from the row of cells at the upstream of a simulated reach and finishing at the downstream end (before returning to the upstream end to begin the next model iteration). From this ground-breaking model, many other cellular models have been developed and applied across range of fluvial environments (Coulthard et al., 2007, 2002; Van De Wiel et al., 2007; Nicholas et al., 2012). Cellular models have also been developed to examine dynamics of aeolian systems. For example, Baas (2002) outlines how a 3-D cellular model enables the representation of aeolian sand entrainment,

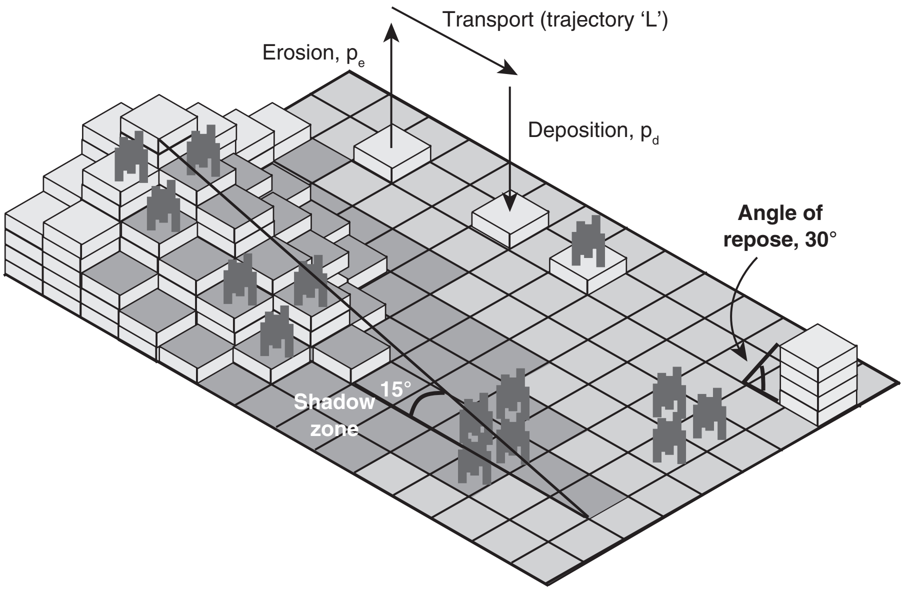

Figure 24.1 Illustration of the representation of simulated sand erosion, transport and deposition in the DECAL model (after Baas 2002, Figure 3). Flow charts describing the scheduling of the model can be found in Baas (2002) and Nield and Baas (2008)

transport and deposition processes by representing the topography (height) of sand dunes using stacked 'slabs' of sand on a lattice (Figure 24.1). Sand transport is simulated by moving slabs across the lattice between adjacent cells, with rules determining movement dependent on wind strength (with slabs in short stacks in the lee of taller stacks being less likely to be moved) and avalanching due to steep slope angles (to maintain an angle of repose at around 33°). Vegetation is represented dynamically in the model, with rules specifying locations and rates of growth dependent on dune height (i.e. number of sand slabs stacked in a cell). In turn, the presence and growth of vegetation in cells limits movement of sand slabs through those cells around that part of the lattice, and thereby influencing dune height and subsequent vegetation growth. Thus, feedbacks between vegetation and sand erosion, transport and deposition are represented and the dynamics of system interactions and dune morphology can be efficiently simulated. Nield and Baas (2008) used a model based on the structure described above, now named DECAL (Discrete ECogeomorphic Aeolian Landscape model), to show how multiple landforms (including Barchan and parabolic dunes) can be produced by the model which is built on simple local rules for movement of sand slabs between cells (aeolian entrainment, deposition, avalanches) and for vegetation growth. For example, if avalanching is only simulated in a given timestep between adjacent cells if the difference in height of slabs between those cells is greater than some predefined value. Thus, through the DECAL model, we see some of the primary characteristics of reduced complexity models: the ability to efficiently simulate the dynamics (due to feedbacks) of geomorphological systems by using rules of interactions between cells in a lattice structure.

Although the term 'Reduced Complexity Model' has been confined mainly to geomorphology, other sub-disciplines of physical geography have adopted similar cellular modelling approaches that enable simulation over larger spatial extents and temporal durations. In terrestrial plant ecology for example, models range from those that represent individual trees and their interactions with one another (e.g. competition for light and other resources) and their environment over areas on the order of 0.1 km^2 (Liu and Ashton, 1995; Pacala et al., 1996) to models that represent succession-disturbance dynamics across landscapes on the order of 1000 km^2 (Scheller and Mladenoff, 2007; Millington et al., 2009). At these broader scales, models usually use a cellular structure and represent plants at species- or community-level, and processes are represented by rules for vegetation change (due to succession or disturbance) rather than equations for growth and mortality. As for RCMs in geomorphology (Box 24.2), advances in computing are enabling more detailed representation over ever larger scales, and the primary use of the broader-scale cellular models is for exploration and understanding of landscape dynamics rather than prediction of system states at particular points in time or space.

Box 24.2 Reduced complexity models in geomorphology

The label 'Reduced Complexity Model' (RCM) is somewhat of paradox (Brasington and Richards, 2007). To see this we first need to understand that the term 'reduced' in this context is relative to the perspective that a 'standard' level of complexity in geomorphological models is the computational fluid dynamics (CFD) approach which uses physically-based partial differential equations (e.g. Navier-Stokes equations; see Lane et al., 1999). RCMs are simplified (or have 'reduced complexity') relative to CFD models in terms of the physical representation of fluid flows and processes of erosion, transport and deposition of solid material. The benefit of such simplification is that RCMs have vastly reduced computational demands compared to CFD models, enabling simulation of longer time durations and greater spatial extents that are more relevant to environmental management (i.e. 'intermediate' scales of 1–100s km^2 and 10s–100s years). However, the simplified representation of RCMs does not preclude the modelling issues of process conceptualization, parameterization, spatial and temporal resolution, and validation faced by developers of more complicated models. The reduced representation fidelity of RCMs can potentially lead to physically-inconsistent results and difficulties reproducing empirical observations (Coulthard et al., 2007). Future increases in computational power may mean that CFD approaches are possible at the scales of study that RCMs are now seen to be valuable for, but in the meantime it has been suggested that use of both in a hybrid, hierarchical manner might prove fruitful (Nicholas et al., 2012). The general view is that the promise of RCMs is in their use for explanation of system dynamics rather than prediction of static states: to understand system behaviour and change (Coulthard et al., 2007), to think differently and challenge assumptions about forms and processes (Odoni and Lane, 2011), and to investigate emergent properties of geomorphological systems (Brasington and Richards, 2007; Murray, 2007). Given that emergence lies at the heart of complexity theory (e.g. Harrison, 2001) this final point further compounds the paradoxical nature of the label 'reduced complexity model' – what is simple and useless for some, is complex and useful for others.

24.3 Simulating Pattern-Process Feedbacks

In geography we are often interested in areal differentiation and spatial patterns, such as how plant and animal species are differentially distributed across the earth's surface or how river channels vary in their planiform shape. Spatial patterns such as these are interesting in of themselves and identifying and mapping spatial distributions and patterns was at the heart of early geography. Although still important, contemporary geography is often interested as much in the processes producing observed patterns as much as the patterns themselves, sometimes simply to better understand, but at other times to be able to predict or make more informed management recommendations or decisions. Furthermore, as our understanding of the processes producing spatial patterns has improved, so we have often become interested in how the patterns themselves influence and modify or change the processes producing them. Consequently, an important reason simulation modelling has become popular in disciplines dealing with space (such as geography, landscape ecology, geomorphology, etc.) is because they provide a means to examine feedbacks between temporal processes and spatial patterns (see Box 24.3).

Box 24.3 Spatial simulation of pattern and process

Computer simulation models provide a means to represent spatial patterns and processes of change for investigation and experimentation: representation of processes may be modified, alternative measurements of patterns tested, and each done repeatedly. Providing numerous examples

(Continued)

(Continued)

that readers can explore for themselves using freely available software, O'Sullivan and Perry (2013) present and discuss spatial simulation for investigating pattern and process in geographical and ecological systems. In particular, O'Sullivan and Perry (2013) suggest that three processes underlie the majority of spatial simulation models currently available. The three spatial processes are:

- *Aggregation/Segregation:* Aggregation and segregation are two sides of the same coin, the former driven by the tendency of similar elements to group together in space, and the latter by the tendency of dissimilar elements to separate in space. If elements are unable to move in space, aggregation may occur as elements change their attributes to become more similar to their local neighbours, thereby likely becoming more dissimilar to elements farther away. A primary means to represent this process is through iterative local averaging, in which values at a location are updated through time as the mean of spatially-local neighbours' attributes.
- *Mobile Entities and Random Walks:* A spatial walk is a succession of 'steps' (i.e., movements), each of which moves an entity from one location in space to another. Spatial walks might be random (direction and length of each step is random) or influenced by attributes of the environment the entity is moving through, by attributes of the entity itself, or even by other entities moving through space. Examples include individual animals herding or flocking or pollutants moving through an environment. In some cases the 'walking' entity may change the environment or influence the walks of other entities and, in turn, this may reciprocally influence the entity's walk.
- *Spread:* Spread processes include diffusion, growth and percolation, and refer to the movement of material or phenomena in a more aggregated form than considered in a spatial walk. For example, the diffusion of a gas through a vacuum from a point source results in the gas becoming evenly distributed across a space, but this process could also be thought of at an atomistic level as being the aggregate result of random walks of all the individual gas particles. Growth, in the context of spread, refers to expansion at a common boundary or front. A prime example is the spread of fire across a landscape, leaving burned land behind as it moves into unburned areas. Percolation shares many characteristics of diffusion and growth, but the emphasis here is on how the environment through which a material or phenomenon moves influences spread (rather than the characteristics of the material or phenomenon itself).

O'Sullivan and Perry (2013), and suggest that these processes might provide the 'building blocks' from which to start simulating a wide variety of spatial patterns. They provide example NetLogo models to demonstrate this which you can explore yourself (see link on accompanying book website).

At its most fundamental, feedback is information about the state of an entity in a system communicated from that entity to another entity in the system. Commonly, feedback is understood to form reciprocal links between system entities, known as feedback loops. Thus, we meet the second kind of loop in simulation modelling. Positive feedback loops reinforce trajectories of change, whereas negative feedbacks act to stabilise and reduce change. For example, to illustrate how positive feedback loops can lead to areal growth or spatial clustering of vegetation, consider the relationship between vegetation and soil in semi-arid environments. In these environments, higher plant densities facilitate greater infiltration of water into the soil than at lower plant densities (HilleRisLambers et al., 2001). Where soil is devoid of plants, falling rain will barely infiltrate and will run off across the surface until it reaches a location where it can infiltrate more readily. Because plants facilitate infiltration this is likely to be near existing patches of vegetation, thus increasing soil moisture availability in the vicinity of the patch. Consequently, conditions for plant establishment are better near existing patches of vegetation, leading to the growth of the vegetation patch. This further raises plant density facilitating yet greater infiltration. In this feedback loop the information about soil (the original entity) causes a change in the extent of vegetation patches (the second entity), which in turn provides information which causes a change in the soil (original entity) at the periphery of the vegetation patch. This can be demonstrated by a simple simulation model (online Model 24.1 – download the code, examine it and test it yourself in NetLogo). The important point to highlight here is that changes are being caused at the periphery of an entity – soil conditions change at the periphery of the vegetation patch and in turn the periphery of the vegetation patch moves. It is because changes due to feedback do not occur at exactly the same point in space, but rather in rather spatially adjacent positions, that allows the areal growth of the vegetation patch.

Across larger spatial areas occupied by a single vegetation patch, HilleRisLambers et al. (2001) found that this positive feedback loop alone, without any spatial heterogeneity in the environment (e.g., slope), can lead to spatial patterns of vegetation patches alternating with bare soil. This can be demonstrated by a second simulation model (online Model 24.2 – try it yourself in NetLogo) in which rain drops fall randomly across an area and then run overland in random directions (due to lack of slope) in a random walk process (see Box 24.3). As the rain water runs overland it infiltrates into soil, increasing the moisture available for plants to grow, with infiltration rates influenced again by plant density. Even with random rainfall, the modifications to infiltrations rates due to plant density (as in the Model 24.1) results in clustering of plant patches across the simulated space. This second model is therefore similar is some ways to the DECAL model described above, in which sand is randomly entrained by wind, transported and deposited depending on the location of other sand (i.e. more likely to be deposited in the lee of higher piles of sand). Again, a simple simulation model (online Model 24.3) can demonstrate how randomness with some simple rules of interaction between landscape elements can lead to spatial pattern, because the processes implied by the interactions are dependent on existing spatial patterns.

In ecology, this concept of processes being shaped by their history (via spatial pattern) is known as the 'memory' of the process (Peterson, 2002). Feedback loops are created if the process has memory – the attributes of entities distributed across space record information about previous events (changes in state) caused by the process. For example, in landscapes that experience frequent fire, a mosaic of vegetation patches of varying age (i.e., due to varying time since last burn) can be produced (Figure 24.2). At landscape scales (e.g., 10 – 10,000 km^2) multiple factors influence how a fire spreads, including wind, physical relief and vegetation cover. Wind can be highly variable between fire events, whereas physical relief varies very little between individual events. Consequently, memory – and therefore feedbacks between spread and spatial pattern – has been conceptualized as being contained in vegetation flammability as a function of time since last the fire (Peterson, 2002). Peterson (2002) demonstrated this using a cellular model of vegetation growth and fire spread (online Model 24.4 is a NetLogo version of this model for you to try yourself). Debate continues about the importance of memory for wildfire in real world ecosystems, particularly in Mediterranean regions (Piñol et al., 2005; Keeley and Zedler, 2009). However, a positive feedback loop is known to exist between landscape homogeneity and fire spread (Loepfe et al., 2010). Homogenous landscapes are characterized by fewer, larger patches of similar vegetation meaning that fire spreads equally well through large areas of the landscape. A fire that spreads through a large area further homogenizes landscape vegetation pattern (by burning the same area and likely more), in turn facilitating spatially consistent fire spread in future events. In contrast, landscapes with greater spatial heterogeneity in land uses (i.e., more, smaller patches) compose a negative feedback loop with fire spread because greater spatial variation in vegetation means fire spreads non-uniformly spatially, producing further heterogeneity.

There are, of course, many simplifications and assumptions in the models described above. However, as we saw for RCMs, such assumptions (allied with a cellular structure) enable rapid simulation and exploration of system dynamics. Furthermore, when used in an appropriate manner, such models can provide an experimental toolkit to examine hypotheses about which processes are most important for producing patterns observed in the real world. For example, the 'Pattern-Oriented Modelling' (POM) approach has been advocated for using simulation models to understand ecological systems (Grimm et al., 2005; Grimm and Railsback, 2012). The POM approach examines different representations of alternative hypothesised processes influencing individual elements in the models (e.g. sand or vegetation in the examples above) and how they combine to reproduce patterns observed in the real world at

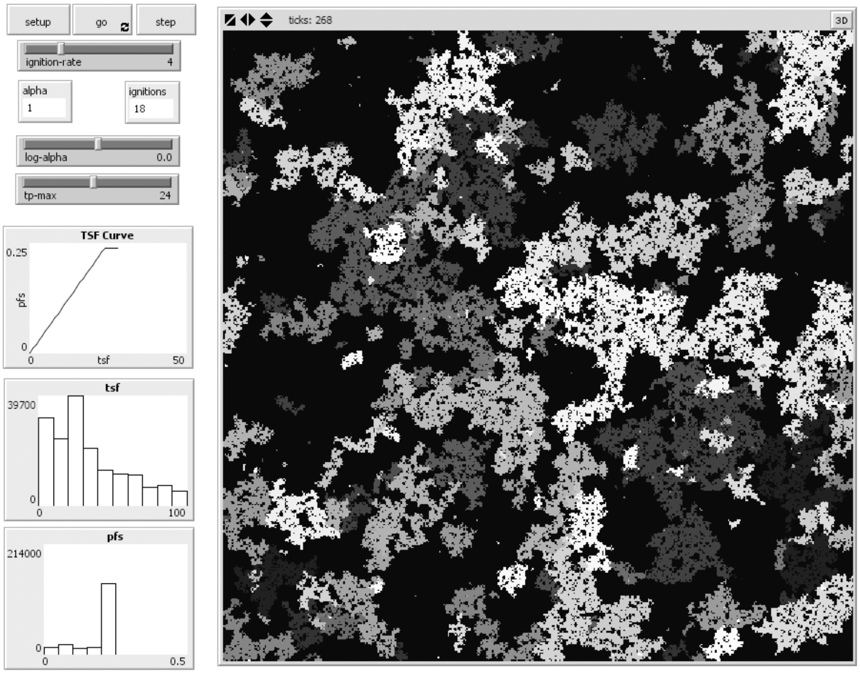

Figure 24.2 Spatial patterns produced by a NetLogo implementation of the Peterson (2002) model. This model is available as Model 24.4 from the book website

different levels of organization. As an experimental approach, different model structures or parameters (e.g. probabilities of sand entrainment due to wind in the DECAL model) are examined systematically to see how outputs vary. One of the keys to POM is that support is provided for the different model structures (or parameters) only if they are able to reproduce *multiple* patterns observed in the real world, preferably at different levels of organisation. For example, a model structure simulating the relationship between vegetation and soil in semi-arid environments may represent processes acting on individual plants (as online Model 24.2 does), but comparison of model output should be not only at that level but also the landscape level (e.g. clustering of vegetation) and also non-vegetation variables (e.g. soil moisture). Thus, careful and systematic use of such relatively simple simulations models can help advance understanding of geographical systems and move such models 'from animations to science' (Grimm and Railsback, 2013; and see below).

Before we turn to the use of these models in your own research, a final set of feedbacks that simulation models are currently being used to explore is those between human activity and environment processes. In particular, a form of simulation known as agent-based modelling (often abbreviated to ABM) can couple explicit representation of individual actors, their attributes, interactions and decisions with cellular representations of the physical environment to explore how the pervasive influence of human activity interacts with environmental processes. For example, Wainwright and Millington (2010) describe two models that link human activity with environmental processes described above. The CybErosion model (Wainwright, 2008) links a Landscape Evolution Model with an agent-based model that represents human and animal agents and their reciprocal influence on soil erosion over several hundred years. The SPASIM model links an agent-based model of contemporary agricultural decision-making (Millington et al., 2008) with a cellular model of Mediterranean-type vegetation succession and fire disturbance (Millington et al., 2009) to explore the reciprocal impacts of land use/cover change and wildfire regimes. As with RCMs, agent-based modelling is still relatively new for investigating geographical systems and offers great promise (Heppenstall et al., 2012), but will potentially produce new loops of interactions that need to be to be negotiated (Hacking, 1995; Millington et al., 2011).

24.4 Developing your Simulation Model

Deciding where to start when embarking on an environmental modelling project has been likened to the age-old Chicken-or-Egg dilemma; do you start by collecting data to then inform the construction of your model, or do you start with a conceptual model implemented in code and then collect data to identify parameter values needed for the model to represent the real world? (Mulligan and Wainwright, 2013). One way of breaking such an infinite loop (or circular reference) might be to envisage the development of simulation models as a *process of modelling*, a series of yet more feedback loops (Figure 24.3). The steps of the modelling process (as I conceptualize it) are:

- Identification of Objectives;
- System Conceptualization;

- Data Collection;
- Model Construction;
- Evaluation; and
- Model Use.

A brief overview of each step is provided below, but further discussion can be found in Mulligan and Wainwright (2013) and Grimm and Railsback (2013b:Chapter 1); a succinct example of model development can be found in O'Sullivan and Perry (2013b: Chapter 8).

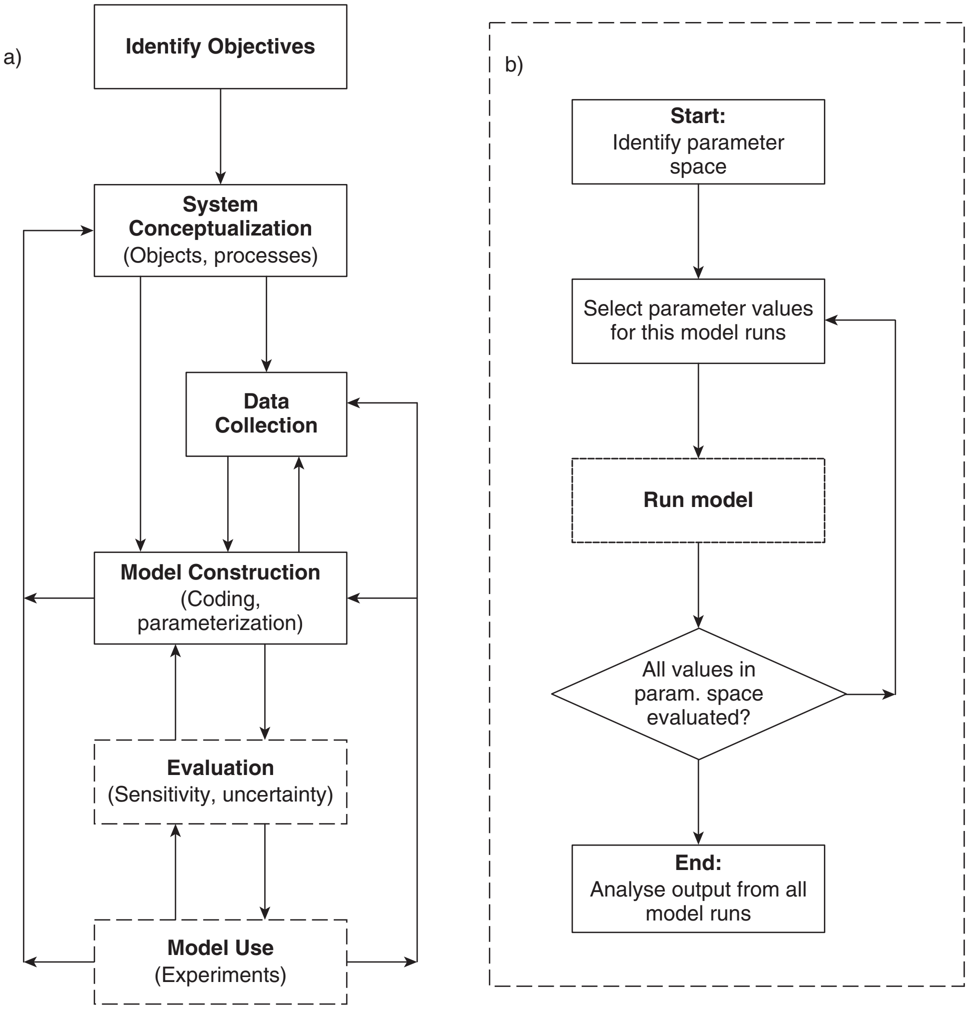

Figure 24.3 Loops in the modelling process. Evaluation and Model Use steps in the general modelling process (a) may have the process for multiple model runs (b) embedded within them. In turn, the 'run model' step in (b) may be represented by a further flowchart for a particular model (e.g. Figure 24.4)

Identify Objectives

Before stepping into the modelling process (loop), it is important to highlight that encompassing the entire processes is the need for a clearly defined objective for the modelling project. Models are sometimes distinguished between their use for prediction and explanation (e.g. Perry and Millington, 2008) and the debate about the how models should or can be used in geography has a long history (Clifford, 2008). Whereas data-driven modelling approaches like statistical models are more likely to aim to predict system states, simulation modelling is often used to improve explanation of a process or phenomena. If prediction is an ultimate goal of a simulation model it may be at a lower level of precision (e.g. more coarse scale) than other forms of modelling. Others may take a different view, but my emphasis here is on the use simulation for explanation and improving understanding. Therefore, before starting out on the development of your simulation model you should have a good idea about either the phenomena you wish to represent via simulation (e.g. dune forms) or the processes (e.g. vegetation-fire feedbacks) you wish to better understand. In either case, (prediction or explanation) you will ideally also have information about patterns and forms observed in the real world against which you can compare the consequences of simulated processes (e.g. frequency-size distributions, landscape pattern metrics, ranges of variability, and so forth).

Conceptualization

From problem identification, we will step into the modelling process (loop) at the conceptualization stage – largely because simulations are essentially abstracted representations of the world used to explore and understand patterns and processes (Odoni and Lane, 2011), but also because when learning how to develop a simulation model it might be better to identify the possibilities and limitations of such tools before spending a lot of time and effort collecting data (see also Box 24.4). Regardless, conceptualization is a key stage in any modelling endeavour, and in geography in particular the issue of identifying model or system boundaries for representation has been long recognized (Richards, 1990; Lane, 2001; Brown, 2004). 'Bounding' or 'closure' of a model is an important step in geographical modelling because the real world is open, in the sense that there is free flow of energy, material and information, but computer models are 'closed' in the sense that a boundary must be defined beyond which the model cannot represent such flows. Model 'closure' thus involves deciding which processes will be represented explicitly within the model, which processes will not be represented in the model but will be parameterised or provide 'boundary conditions', what the spatial extent and resolution of the model will be, what the initial state of the modelled environment will be, and when the simulation will stop. For example, consider the DECAL model described above (and the simple version available online). The model represents sand transport in the wind direction but not variations due to turbulence, it specifies wind direction and strength (boundary conditions) but not the atmospheric conditions producing that wind, and it models an arbitrarily area or space and length of time. Similarly, general circulation models for examining possible climate change must decide which processes to simulation, what area (global or regional), resolution (what grid-size for atmospheric representation) and usually do not represent processes causing changes in atmospheric composition (e.g. greenhouse gas emissions are model boundary conditions specified by scenarios of future human activity). Further decisions in model conceptualization are what objects will be represented and what processes or relationships will be represented by parameters within the model. For example, in the DECAL model 'slabs' of sand are represented, not individual grains, and deposition in non-shadow regions is probabilistic (e.g. see parameter *p* in the simple version online) avoiding the need to represent fine-detailed processes of aeolian entrainment. Finally, variables often must be defined to be able to 'measure' model output so that it can be compared to empirical observations. For example, in models of vegetation growth and disturbance by fire, fire sizes must be recorded (and output to the user) so that simulated and observed frequency-size distributions can be compared (Millington et al., 2009). Model description protocols, such as the Overview, Design concepts, and Details (ODD) protocol developed for individual and agent-based simulation models (Grimm et al., 2010) can be useful to aid model conceptualization and construction.

Data Collection

Data may be required for several different aspects of model development; to establish parameters, to provide boundary conditions (e.g. scenarios of change), to establish empirical patterns against which to compare model output (e.g. pattern-oriented modelling). The particular data required will vary from project to project depending on the subject, and methods for collecting data are covered in detail in several other chapters in this book (e.g. Chapters 30–32).

Model Construction

This step refers to the process of converting the conceptual model into code that a computer can then execute to simulate, often thought of as 'computer coding'. In the past this step of the modelling process required detailed knowledge of computer programming and how computers actually function to achieve their calculations. However, recently several 'modelling environments', programming languages and libraries have been created that simplify coding (e.g. by taking care of things like memory allocation and providing functions to automate certain tasks) so that those interested in modelling geographical systems can do so much more efficiently. The choice of modelling environment depends in part on the objectives of the modeller (see *Identify Objectives* above), their programming skills, and the characteristics of the modelling environment (or programming language). The benefits and constraints of several modelling environments and languages for developing simulation models of geographical systems depend on factors such as ease of use, speed of execution and types of representation of spatial interactions (these are discussed further on the accompanying book website. Example online models accompanying this chapter have been written in code that can be implemented in the freely available modelling environment, NetLogo (Wilensky, 1999). Designed for the creation of individual- and agent-based models, but also useful for developing geographic simulation models more generally, NetLogo is a good place to start for geographers learning simulation modelling because of its flexibility and simple syntax (see Box 24.1). These characteristics mean many people are now using this environment to develop simple, abstract models (O'Sullivan and Perry, 2013; Railsback and Grimm, 2012), but as model complexity increases computational overheads (a consequence of the architecture needed to keep the programming language simple) mean that NetLogo models can run slowly and alternatives may need to be sought. Whether using NetLogo or another environment (e.g. PCRaster (http://pcraster.geo.uu.nl; Windows, Linux), MASON (http://cs.gmu.edu/~eclab/projects/mason/; Cross-platform)) or language

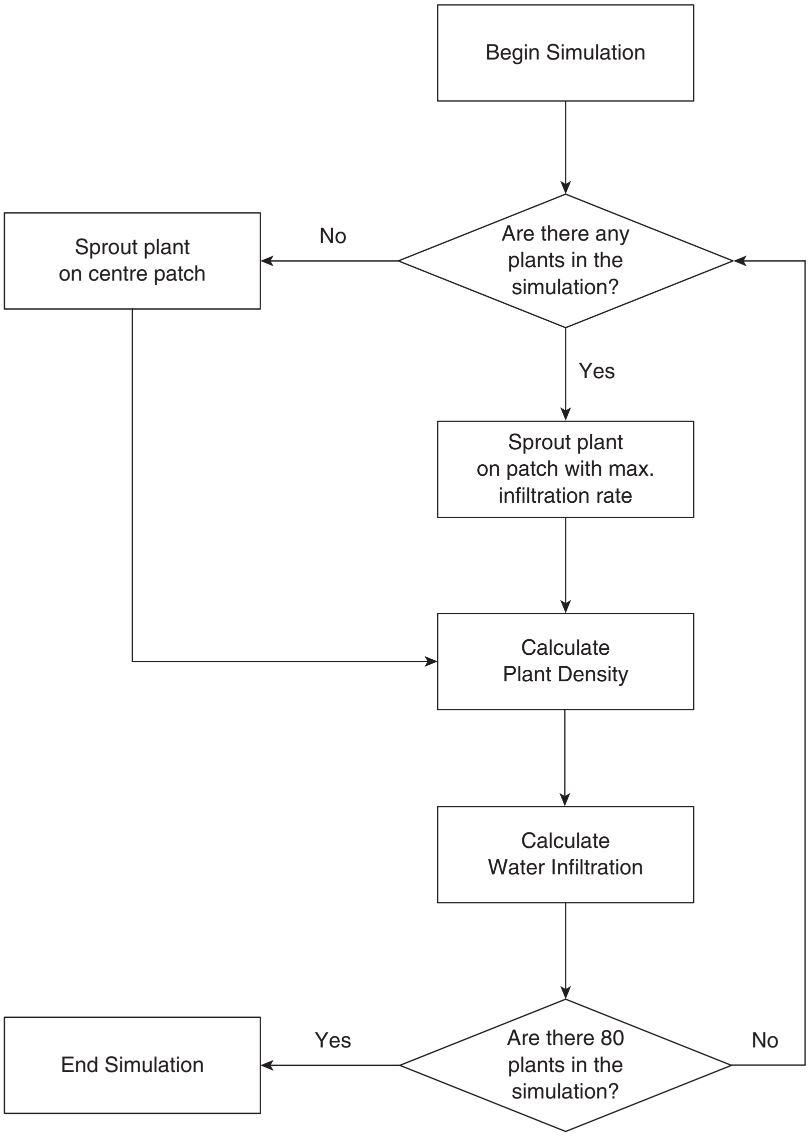

Figure 24.4 Flow chart describing scheduling of the code for a simple areal vegetation growth model (online Model 24.1, NetLogo code for which is available on the book website). The 'Calculate Plant Density' step could in turn be represented by another flow chart

(e.g. Python), some competence in programming will be needed. Introductory materials (including books, websites, Internet forums, other programmers) are available and should be sought out and used!

An important foundation for learning to program for simulation modelling is the ability to think computationally so that effective algorithms can be written. An algorithm is a step-by-step procedure that enables calculations and can be as simple as the population growth example in Box 24.1, or contain multiple algorithms to produce an entire model (e.g. the example models provided online). Because algorithms are step-by-step procedures, and often demand looping, they can be thought of as the coded equivalent of a flow chart. Indeed flows charts can be very useful when developing and presenting algorithms and models as they provide a visual means to trace the sequences of calculations or events required (Figure 24.3a and Figure 24.4). Other techniques may also be required in model construction, for example the use of regression modelling to establish relationships to be represented in the simulation model (Millington et al., 2013) but these techniques are beyond the scope of this chapter.

Evaluation

Once a model has been constructed in code ready for execution in the computer, several forms of evaluation are needed and which may send a modeller right back to the previous step in the modelling process (model construction) or earlier! Most generally, two forms of evaluation can be distinguished: verification and validation (Oreskes et al., 1994; Rykiel, 1996). Verification can be thought of as ensuring the model matches the intended conceptual model (by comparing output against expectations – 'building the model correctly'), whereas validation can be thought of as ensuring the model matches processes in the real world (by comparing output to empirical observations – 'building the correct model'). Validation is important if the objective of the modelling is prediction, but the fundamental difficulties of 'true' validation of (closed) models of real-world (open) geographical systems are well known, if often forgotten (Oreskes et al., 1994). Verification is more important when the objective of the modelling is explanation or understanding, as aspects of it can even overlap with the *Model Use* step. The three forms of verification that are often useful (particularly for finding 'bugs' or errors in your model code) are: model exploration, sensitivity analysis and uncertainty analysis (Malamud and Baas, 2013). Model exploration is a form of model testing that involves playing with the model code to check it does what you expect by comparing alterations in code to expected outcomes in model output. As well as helping to identify bugs it can help to explore appropriateness of the chosen model conceptualization by identifying unrealistic or empirically impossible system dynamics. Sensitivity analysis examines how variation in parameters or constants in a model influence outputs (e.g. what percentage change in model output does a given percentage change in parameter X produce?). Uncertainty analysis is very similar, but whereas sensitivity analysis is interested solely in the relationship between input and output, uncertainty analysis demands that the uncertainty in the input value is accounted for (e.g. via a probability distribution for the input value). Both sensitivity and uncertainty analyses are useful for identifying aspects of the model structure that are more or less important (or even irrelevant and therefore unnecessary) for understanding model outputs by tracing how different parameter or constant values influence processes and dynamics. They are also useful for identifying bugs in model code (when variations are unfeasible) and each can examine input values independently (vary values one at a time) or jointly (vary multiple values in a given model run). The importance of output for evaluation highlights the thought needed to establish what aspects of the modelled system will be 'measured' and reported via model code. Visualization of model output is also often important and the use of plots of variables, images and movies of model dynamics can be useful for model evaluation.

Model Use

As with model evaluation, model use depends on the objectives of the modelling. Whereas for predictive purposes a model may be used to simulate outcomes of different scenarios (of boundary conditions) for decision-making or management in particular instances, for explanatory purposes a model may be used to identify model structures that produce observed patterns (as in pattern-oriented modelling) or explore system dynamics more generally (e.g. existence of location of limits or thresholds). For example, some simulation models of wildfire model physical processes of ignition and combustions to represent fire behaviour as accurately as possible so that firefighters can anticipate where a fire will spread and therefore how to tackle it (e.g. Anderson et al., 2007). In contrast, ecologists may be more interested in understanding relationships between vegetation succession and fire as an ecological disturbance, and therefore represent processes differently to understand dynamics more generally (Peterson, 2002). However, a common aspect to both cases is the execution of the model multiple times (possibly up to thousands of times), each execution being known as a model run. Multiple model runs may use the same initial conditions to identify the range of likely outcomes (predictive mode) or parameters may be varied systematically in 'simulation experiments' to explore system dynamics (Peck, 2004; Railsback and Grimm, 2012). As we saw in the examples above, the ability to run a simulation model multiple times enables experimentation in systems that would not be possible empirically due to scale issues and exploration of alternative trajectories of change into the future. In NetLogo the 'BehaviorSpace' is useful for specifying multiple model runs with alternative parameter values (Peterson, 2002).

Key to appropriate use and interpretation of a model is understanding how output is related to input via simulated processes. Questions will often arise about those connections during *Model Use* (and *Construction* and *Evaluation* steps), due to mismatches between model output and empirical data or theoretical expectations, and modellers will often find themselves returning to the *Conceptualization* step for consideration of alternative models. In returning to conceptualization, we realize that modelling has the potential to be never-ending process. This may seem frustrating (see Box 24.4), but ultimately it is simple a reflection of the use of the scientific method to continually refine our understanding. Critics of simulation models may see them as incomplete representations of reality that have little bearing on our understanding of the real world (Goering, 2006; Simandan, 2010), but they fail to see the inductive and hermeneutic value of the process of modelling for understanding (Kleindorfer et al., 1998; Peck, 2008) and how models can be more or less useful (Box, 1979) or reliable (Winsberg, 2010) for understanding the world. Furthermore, innovative participatory modelling means such advantages may be appreciated by non-scientists and scientist alike (Lane et al., 2011; Souchère et al., 2010). Ultimately, all models (including maps and mental models) are simplifications of the real world and do not provide perfect representation of it. However, to the extent that we gain trust in a simulation model through the modelling process (e.g. by confronting it with data and our theoretical expectations), it allows us to explore structures needed to produce observations or expectations and how much scope for different outcomes there might be.

Box 24.4 Tips for developing a simulation model

Algorithms are step-by-step procedures at the heart of computer simulation models. Although simple algorithms describing the process of developing a simulation model in a step-by-step fashion are possible (e.g. Figure 24.4), modelling is as much an art as a science and no single set of instructions is appropriate in all circumstances. Identifying the key real-world objects, interactions and processes to represent, and carefully considering how they should be appropriately operationalized in computer code, demands imagination and experience as much as theory and knowledge. This experience must be gained personally, but a few tips might save the novice some heartache and time:

- Start with as simple a model (i.e. representation) as possible and build from there. This has the benefit of both adhering to Ockham's Razor (Wainwright and Mulligan, 2013) and will help to keep tests of your patience to a minimum (but accept that your patience *will* be tested).
- Be clear about your objectives right from the start. This will help keep you on track (and sane) when you find yourself wrestling with the implementation of an algorithm or playing curiously for hours with the behaviour of some interesting minor aspect of your model. Being clear about objectives will also help with your (simple!) system conceptualization.
- In your system conceptualization, be clear about your model boundaries (and the constraints they impose), what your key objects are (multiple of which produce patterns), what the vital processes are (that cause changes in states of objects) and how these are all related. In turn, this will allow you to understand what parameters, constants and variables you need and what data might be needed to provide values for them. For example, for the code in Box 24.1 we can see the following:
    - Object: Population
    - Process: Growth of the population
    - Initial conditions: Initial population size (in this case 4)
    - Parameter: Growth rate
- When using a new modelling environment or language, take a little time to learn about how it is designed to be most efficient and get yourself up to speed on the built-in commands and data structures (often known as 'primitives'). These will save time both in coding up your model and executing it in the computer by increasing processing speed.
- Do ask advice of those who have been where you are now. Look at how others have conceptualized similar systems and solved similar coding problems, and use the Internet as the prodigious programming resource it is (there *will* be someone who has had a similar programming problem and posted it on an Internet forum somewhere).
- Don't expect things to work first time. Be patient, remember your objectives and keep going!

SUMMARY

As with all types of model, simulation and reduced complexity models are simplified representations of reality that can be useful in different ways depending on the objectives of the modeller. As simplified representations they share some features of their targets but not all, the choice of which can be likened to writing an essay. Much like modelling, when writing an essay (or this book chapter!) decisions must be made about the importance of different aspects of a subject in deciding the emphasis and time spend discussing them. In both essay writing and modelling, aspects of the target/subject need to be weighted by their importance based on previous understanding (literature), data, resources (computing power/word limit) and the objectives of the modeller/writer and what they want to explore. In simulation modelling, decisions are needed about what objects and processes will be included within the model boundary and how they are coupled, with the choice and weighting determined by the aim (e.g. explain or predict). For the simulation models discussed here, the focus has been on explanation and the exploration of system dynamics and feedbacks rather prediction of (static) states at particular points in time or space. My emphasis has been on models for explanation rather than prediction and I have stressed the utility of simulation for dynamically representing feedback loops. Developing your first simulation model, deciding what to include and leave out and working out how represent the real world in computer code, will likely be challenging. But understanding the need and utility of the nested loops discussed here – iterated execution of computer code to represent feedbacks in the geographical world through a continually reflexive modelling process – will be helpful, as should (re-)reading book chapters like Millington (2016).

Acknowledgements

I'm grateful to Nick Clifford, Andreas Baas (who provided Figure 24.1), Jacob White and Brynmor Saunders for comments and assistance with this particular manuscript, and to numerous other colleagues for their support during the loops of my own modelling career.

Further Reading

An excellent textbook covering a wide range of modelling approaches for physical geographers, including the process of modelling, is provided by Wainwright and Mulligan (2013). For more specific treatment of modelling geomorphological systems, see Brasington and Richards (2007) and Malamud and Baas (2013). Wainwright and Millington (2010) provide examples of how agent-based modelling might be combined with geomorphological models, whereas the textbook by Grimm and Railsback (2012) provide a comprehensive introduction to agent-based and individual-based modelling. Finally, as discussed throughout the chapter, O'Sullivan and Perry (2013) explore the use of simulation models for investigating pattern and process in geographical and ecological systems, offering multiple examples for use in NetLogo (Willensky 1999) and proposing a series of fundamental 'building block' models on which many geographical systems might be examined.

Note: Full details of the above can be found in the references list below.

References

Anderson, K., Reuter, G., and Flannigan, M.D. (2007) 'Fire-growth modelling using meteorological data with random and systematic perturbations', *International Journal of Wildland Fire* 16: 174–82.

Baas, A.C.W. (2002) 'Chaos, fractals and self-organization in coastal geomorphology: Simulating dune landscapes in vegetated environments', *Geomorphology* 48: 309–28.

Bithell, M., Brasington, J. and Richards, K. (2008) 'Discrete-element, individual-based and agent-based models: Tools for interdisciplinary enquiry in geography?', Geoforum 39: 625–42.

Box, G.E.P. (1979) 'Robustness in the strategy of scientific model building', in: R.L., Launer and G.N. Wilkinson (eds), *Robustness in Statistics*. New York, NY: Academic Press. pp. 201–36.

Brasington, J. and Richards, K. (2007) 'Reduced-complexity, physically-based geomorphological modelling for catchment and river management', Geomorphology 90: 171–7.

Brown, J.D. (2004) 'Knowledge, uncertainty and physical geography: Towards the development of methodologies for questioning belief', *Transactions of the Institute of British Geographers* 29: 367–81.

Clifford, N.J. (2008) 'Models in geography revisited', *Geoforum* 39: 675–86.

Coulthard, T.J., Hicks, D.M. and Van De Wiel, M.J. (2007) 'Cellular modelling of river catchments and reaches: Advantages, limitations and prospects', *Geomorphology* 90: 192–207.

Coulthard, T.J., Macklin, M.G. and Kirkby, M.J. (2002) 'A cellular model of Holocene upland river basin and alluvial fan evolution', *Earth Surface Processes and Landforms* 27: 269–88.

Goering, J. (2006) 'Shelling Redux: How sociology fails to make progress in building and empirically testing complex causal models regarding race and residence', *Journal of Mathematical Sociology* 30: 299–317.

Grimm, V., Berger, U., DeAngelis, D.L., Polhill, J.G., Giske, J. and Railsback, S.F. (2010) 'The ODD protocol: A review and first update', *Ecological Modelling* 221: 2760–8.

Grimm, V. and Railsback, S.F. (2012) 'Pattern-oriented modelling: A "multi-scope" for predictive systems ecology', *Philosophical Transactions of the Royal Society B: Biological Sciences* 367: 298–310.

Grimm, V. and Railsback, S.F. (2013a) *Individual-based modeling and ecology*. Princeton: Princeton University Press.

Grimm, V. and Railsback, S.F. (2013b) 'Introduction',in V. Grimm and S.F. Railsback (eds) *Individual-based modeling and ecology*. Princeton: Princeton University Press. pp. 3–21.

Grimm, V., Revilla, E., Berger, U., Jeltsch, F., Mooij, W.M., Railsback, S.F., Thulke, H.H., Weiner, J., Wiegand, T. and DeAngelis, D.L. (2005) 'Pattern-oriented modeling of agent-based complex systems: Lessons from ecology', *Science* 310, 987–91.

Hacking, I. (1995) 'The looping effects of human kinds', in: D. Sperber, D. Premack, and A.J. Premack (eds) *Causal Cognition: A Multidisciplinary Approach.* Oxford: Oxford University Press. pp. 351–83.

Harrison, S. (2001) 'On reductionism and emergence in geomorphology', *Transactions of the Institute of British Geographers* 26: 327–39.

Heppenstall, A.J., Crooks, A.T., See, L.M. and Batty, M. (2012) *Agent-Based Models of Geographical Systems*. London: Springer.

HilleRisLambers, R., Rietkerk, M., van den Bosch, F., Prins, H.H. and de Kroon, H. (2001) 'Vegetation pattern formation in semi-arid grazing systems', *Ecology* 82: 50–61.

Keeley, J.E. and Zedler, P.H. (2009) 'Large, high-intensity fire events in southern California shrublands: Debunking the fine-grain age patch model', *Ecological Applications* 19: 69–94.

Kleindorfer, G.B., O'Neill, L. and Ganeshan, R. (1998) 'Validation in simulation: Various positions in the philosophy of science', *Management Sciience* 44: 1087–99.

Lane, S.N. (2001) 'Constructive comments on D Massey – "Space-time, 'science' and the relationship between physical geography and human geography"', *Transactions of the Institute of British Geographers* 26: 243–56.

Lane, S.N., Bradbrook, K.F., Richards, K.S., Biron, P.A. and Roy, A.G. (1999) 'The application of computational fluid dynamics to natural river channels: Three-dimensional versus two-dimensional approaches', *Geomorphology* 29: 1–20.

Lane, S.N., Odoni, N.A., Landström, C., Whatmore, S.J., Ward, N. and Bradley, S. (2011) 'Doing flood risk science differently: An experiment in radical scientific method', *Transactions of the Institute of British Geographers* 36: 15–36.

Liu, J. and Ashton, P.S. (1995) 'Individual-based simulation models for forest succession and management', Forest Ecology and Management 73: 157–75.

Loepfe, L., Martinez-Vilalta, J., Oliveres, J., Piñol, J. and Lloret, F. (2010) 'Feedbacks between fuel reduction and landscape homogenisation determine fire regimes in three Mediterranean areas', *Forest Ecology and Management* 259: 2366–74.

Malamud, B.D. and Baas, A.C.W. (2013) 'Nine considerations for constructing and running geomorphological models', in: A.C.W. Baas (ed.) *Quantitative Modeling of Geomorphology* (Treatise on Geomorphology, vol. 2). San Diego: Academic Press. pp. 6–28.

Millington, J.D.A. (2016) 'Simulation and Reduced Complexity Models', in N. Clifford et al. (2016) *Key Methods in Geography*. London: Sage pp. 119–59.
Millington, J.D.A, Walters, M.B., Matonis, M.S. and Liu, J. (2013) 'Filling the gap: A compositional gap regeneration model for managed northern hardwood forests', *Ecological Modelling* 253: 17–27.
Millington, J.D.A., Romero Calcerrada, R. and Demeritt, D. (2011) 'Participatory evaluation of agent-based land-use models', *Journal of Land Use Science* 6: 195–210.
Millington, J.D.A., Romero-Calcerrada, R., Wainwright, J. and Perry, G.L.W., (2008)' An agent-based model of Mediterranean agricultural land-use/cover change for examining wildfire risk', *Journal of Artificial Societies and Social Simulation* 11 (4): 4. http://jasss.soc.surrey.ac.uk/11/4/4.html.
Millington, J.D.A., Wainwright, J., Perry, G.L.W., Romero Calcerrada, R. and Malamud, B.D. (2009) 'Modelling Mediterranean landscape succession-disturbance dynamics: A landscape fire-succession model', *Environmental Modelling & Software* 24: 1196–1208.
Millington, J.D.A., O'Sullivan, D. and Perry, G.L.W. (2012) 'Model histories: Narrative explanation in generative simulation modelling', *Geoforum* 43: 1025–34.
Mulligan, M. and Wainwright, J. (2013) 'Modelling and model building', in: J. Wainwright, and M. Mulligan (eds), *Environmental Modelling: Finding Simplicity in Complexity*. Chichester: Wiley. pp. 7–26.
Murray, A.B. (2007) 'Reducing model complexity for explanation and prediction', *Geomorphology* 90: 178–91.
Murray, A.B. and Paola, C. (1994) 'A cellular model of braided rivers', *Nature* 371: 54–7.
Nicholas, A.P., Sandbach, S.D., Ashworth, P.J., Amsler, M.L., Best, J.L., Hardy, R.J., Lane, S.N., Orfeo, O., Parsons, D.R. and Reesink, A.J. (2012) 'Modelling hydrodynamics in the Rio Paraná, Argentina: An evaluation and intercomparison of reduced-complexity and physics based models applied to a large sand-bed river', *Geomorphology* 169: 192–211.
Nield, J.M. and Baas, A.C. (2008) 'Investigating parabolic and nebkha dune formation using a cellular automaton modelling approach', *Earth Surface Processes and Landforms* 33, 724–40.
O'Sullivan, D. and Perry, G.L.W. (2013a) *Spatial Simulation: Exploring Pattern and Process.* Chichester: Wiley-Blackwell..
O'Sullivan, D. and Perry, G.L.W. (2013b) 'Weaving it all together', in D. O'Sullivan and G.LW. Perry, *Spatial Simulation: Exploring Pattern and Process.* Chichester: Wiley-Blackwell. 229–64.
Odoni, N.A. and Lane, S.N. (2011) 'The significance of models in geomorphology: From concepts to experiments', in K.J. Gregory and A.S. Goudie (eds) *The SAGE Handbook of Geomorphology*. London: Sage. p. 154–173
Oreskes, N., Shrader-Frechette, K. and Belitz, K. (1994) 'Verification, validation, and confirmation of numeric models in the earth sciences', *Science* 263: 641–6.
Pacala, S.W., Canham, C.D., Saponara, J., Silander, J.A., Kobe, R.K. and Ribbens, E. (1996) 'Forest models defined by field measurements: Estimation, error analysis and dynamics' Ecological Monographs 66: 1–43.
Peck, S.L. (2004) 'Simulation as experiment: A philosophical reassessment for biological modelling', *TRENDS in Ecology and Evolution* 19: 530–4.
Peck, S.L. (2008) 'The hermeneutics of ecological simulation' *Biology & Philosophy* 23: 383–402.
Perry, G.L.W. and Millington, J.D.A. (2008) 'Spatial modelling of succession-disturbance dynamics in forest ecosystems: Concepts and examples', *Perspectives in Plant Ecology, Evolution and Systematics* 9: 191–210.
Peterson, G.D. (2002) 'Contagious disturbance, ecological memory, and the emergence of landscape pattern', *Ecosystems* 5: 329–38.
Piñol, J., Beven, K. and Viegas, D.X. (2005) 'Modelling the effect of fire-exclusion and prescribed fire on wildfire size in Mediterranean ecosystems' *Ecological Modelling* 183: 397–409.
Railsback, S.F. and Grimm, V. (2012) *Agent-Based and Individual-Based Modeling: A Practical Introduction*, Princeton University Press, Princeton, NJ.
Reddy (2011) http://dx.doi.org/10.1007/978-90-481-8702-7_37.
Richards, K.S. (1990) '"Real" geomorphology', *Earth Surface Processes and Landforms* 15: 195–7.
Rykiel, E.J. (1996) 'Testing ecological models: The meaning of validation', Ecological Modeling 90, 229–44.
Scheller, R.M. and Mladenoff, D.J. (2007) 'An ecological classification of forest landscape simulation models: tools and strategies for understanding broad-scale forested ecosystems', *Landscape Ecology* 22: 491–505.
Simandan, D. (2010) 'Beware of contingency', *Environment and Planning D-Society & Space* 28: 388–96.

Souchère, V., Millair, L., Echeverria, J., Bousquet, F., Le Page, C. and Etienne, M. (2010) 'Co-constructing with stakeholders a role-playing game to initiate collective management of erosive runoff risks at the watershed scale', *Environmental Modelling & Software* 25: 1359–70.
Stott, T. (2010) 'Fluvial geomorphology', *Progress in Physical Geography* 34 (2): 221–45.
Torrens, P.M. (2012) 'Moving agent pedestrians through space and time', Annals of the Association of American Geographers 102: 35–66.
Tucker, G.E. and Hancock, G.R. (2010) 'Modelling landscape evolution', *Earth Surface Processes and Landforms* 35: 28–50.
Van De Wiel, M.J., Coulthard, T.J., Macklin, M.G. and Lewin, J. (2007) 'Embedding reach-scale fluvial dynamics within the CAESAR cellular automaton landscape evolution model', *Geomorphology* 90: 283–301.
Wainwright, J. (2008) 'Can modelling enable us to understand the role of humans in landscape evolution?' *Geoforum* 39: 659–74.
Wainwright, J. and Millington, J.D.A. (2010) 'Mind, the gap in landscape-evolution modelling', Earth Surface Processes and Landforms 35: 842–55.
Wainwright, J. and Mulligan, M.(eds) (2013) *Environmental Modelling: Finding Simplicity in Complexity*. Chichester: Wiley.
Wilensky, U. (1999) 'NetLogo' http://ccl.northwestern.edu/netlogo (accessed 25 November 2015).
Winsberg, E. (2010) *Science in the Age of Computer Simulation*. Chicago: Chicago University Press.

ON THE COMPANION WEBSITE…

Visit **https://study.sagepub.com/keymethods** for author videos, chapter exercises, resources and links, plus **free** access to the following recommended articles:

1. **O'Sullivan, D. (2008) Geographical information science: agent-based models. *Progress in Human Geography*, 32 (4): 783–91.**

This article reviews the range of approaches that agent-based simulation models embody, considering their implications for representation of geographical processes and patterns.

2. **Stott, T. (2010) Fluvial geomorphology, *Progress in Physical Geography*, 34 (2): 221–45.**

This paper provides an overview of reduced-complexity geomorphological modelling for river and catchment management and how this approach relates to others.