12/2 (2014), 229–240
tmcs@math.klte.hu
http://tmcs.math.klte.hu
Teaching Mathematics and Computer Science

Modelling and simulation in education and the NetLogo simulation environment

Péter Bernát

Abstract. Just like real experimentation, computer simulation is a method for understanding the world. In the present paper I will demonstrate its possible didactic advantages and application potentials. The displayed simulations, which will be analyzed in a separate section, were all made in the NetLogo environment, one of them by the author himself.

Key words and phrases: modelling, simulation, natural sciences, NetLogo, programming.

ZDM Subject Classification: K90, M10, R30, U60.

1. Experimentation and simulation

As a scientific method used for understanding reality, experimenting was first documented as early as the 17th century. With its help, both deductive and inductive conclusions can be drawn; that is, with the appropriate experiment a theory can be justified, and results gained and generalized from experiments can lead to new theories and knowledge.

Experimenting is a generally applied method in Hungarian public education. Not only is it illustrative, but in given circumstances (if the students do the experiments themselves) it can also be task-oriented, which, for one, facilitates the deeper understanding of a given phenomenon, and, for two, raises students' interest for the task.

In numerous cases, however, it occurs that the experiment cannot be performed within the given circumstances (of the school or of the scholar). For instance, it is too dangerous or too expensive, which are the chief reasons listed why only the teacher performs the experiment, but it can also be an obstacle if the phenomenon cannot be observed or manipulated adequately due to its speed, size, and so on.

For these cases the computer, a universal experimenting tool, offers a solution: a simulation environment for the experiments, with which we can overcome the above listed limitations.

Just like real experimentation, computer simulation is a method suitable for understanding the world, and its application in different disciplines is growing. As an educational tool, it offers the same didactic advantages described above regarding real experimentation.

In Hungarian education, however, simulation rarely appears. My goal, therefore, is to argue for its place in education, while also demonstrating its application potentials. The simulation programs, one of which is my own, were all made in the simulation environment of NetLogo, which will be presented in a separate section.

2. Modelling and simulation

The basic purpose of simulation is to help understand a real-world system, that is, to uncover the behavioral rules, characteristic to all elements of the system, which are responsible for the tangible features of the system.

Let me illustrate this with an example. An obvious, directly perceptible feature of birds of the same breed is that they flock and fly together. Theoretically, it is a possible presumption that there is a leader that determines the direction, but it is similarly valid to assume that the birds flying together does not require a lead. To explain the phenomenon, we have to look for behavioral rules that hold for birds.

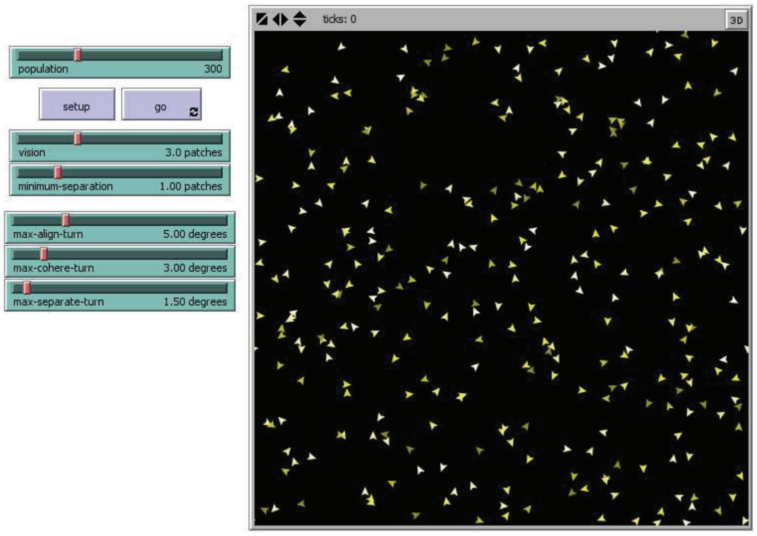
The simulation-based method [1] begins with *observing* the real-world phenomenon. As a tendency, we will be able to observe general-level rules about the system, and only occasionally specific-level rules about its elements. Regarding the birds, we can notice not only that they fly in the same direction, but also that their specific position is not fixed. As a consequence, we can assume that it is not a leader that defines the path but that they all behave according to the same principles. In the case of a specific bird, we can set the rules accordingly [2]:

1. it flies in the direction of the birds nearby,

2. it adjusts to the direction of the birds nearby,
3. it keeps distance from birds which are too close.

According to this *hypothesis*, we design the *model* of the real-world system. Typically, the model determines the objects and the rules that define the states of these objects, which can often be parameterized. In this specific case, the objects of the model are the birds of specific location and direction, while the rules can be set based on the principles of the hypothesis. The size of the flock, just like the data appearing in the rules, such as distance (what is "close" and "too close") and alignment willingness (to how much degree a bird is willing to align to neighbors), are parameterized.

Based on the model, a *simulation program* can be made, which will execute the rules on the objects, bringing the model to life. Both general and specific – simulation – programs can be used for such a purpose. NetLogo, for example, contains several simulation programs ready to use, one of which is modelling the very same flocking phenomenon I just presented (Figure 1).



population 300
vision 3.0 patches
minimum-separation 1.00 patches
max-align-turn 5.00 degrees
max-cohere-turn 3.00 degrees
max-separate-turn 1.50 degrees

Figure 1. Flocking simulation (from NetLogo's library)

During the *simulation* process, different parameter settings can be used for the model. We do this, on the one hand, to *validate* the model: if the processes taking place in our simulation resemble the real-world phenomena adequately, the model is valid (with the given parameters). In the NetLogo simulation, anyone can see that the birds flock together, and, after a while, fly in one direction, without any leader (Figure 2).

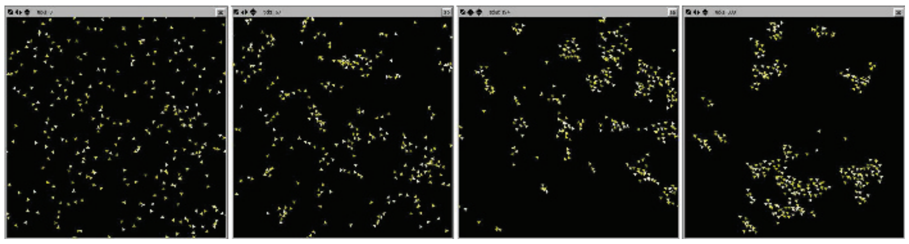


Figure 2. The birds flock together and fly in one direction (Flocking simulation)

On the other hand, if our model is valid, the simulation program can be used for experimenting about the real-world system as well. In this case, we *evaluate* the real-world system with the help of the experimental results. With the NetLogo simulation, we can test how the behavior of the birds influence, for instance, how big a flock is, how fast the direction is formed, and how durable it proves to be.

In short, we can gather information about the real-world system through, first, the validation of the simulation model and, second, the experiments performed in the simulation environment. The whole process of understanding real-world phenomena is illustrated by figure 3 below.

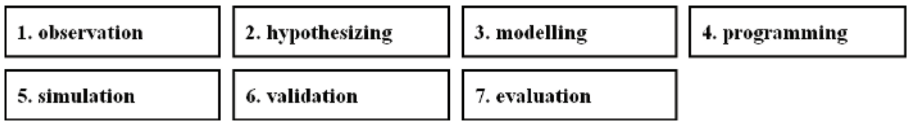


Figure 3. The steps of modelling and simulation

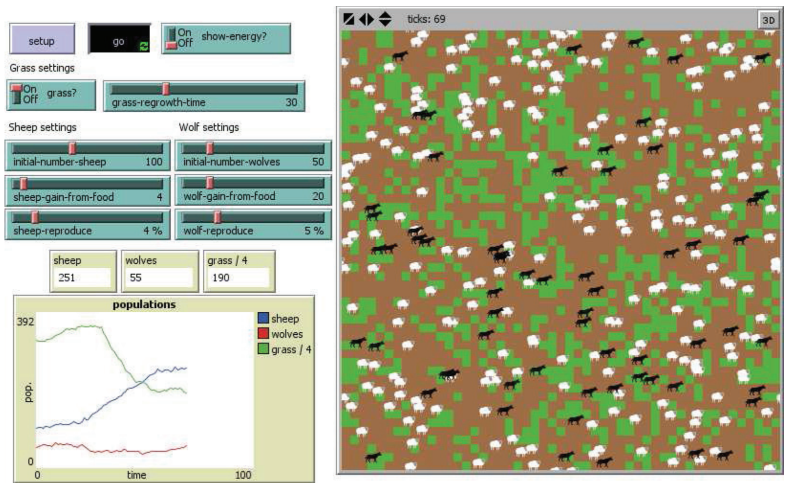
3. NetLogo

NetLogo [3] is a free, open-source environment and programming language, used both for education and research. Thanks to a high-quality instructions kit compiled specifically for simulations, one can design simulation programs without professional programming experiences. NetLogo facilitates individual learning and work with sample programs, paired up with lengthy documentations and concise presentations of programming options. Next to this, it contains a fair amount of ready-to-use simulation programs in the fields of physics, biology, geography, informatics, mathematics, and social sciences, with added questions and explanations (in English) [4].

With NetLogo's programming language, you can control the square-shaped patches, which form the simulation space, and the mobile objects, which, like in all Logo-like languages, are called turtles. The patches and the turtles have their own variables, which can be managed with the appropriate basic instructions. New breeds can be created, to accompany turtles, which will also have own variables.

In NetLogo you can add to your program various input items, like buttons, sliders, switches, choosers, and input boxes, and output items, like monitors and plots (Figure 4).

The user surface contains an additional slider to control simulation speed and an option to temporarily switch off image refreshment, thus, to accelerate the simulation.



ticks: 69

grass-regrowth-time 30

initial-number-sheep 100
sheep-gain-from-food 4
sheep-reproduce 4 %

initial-number-wolves 50
wolf-gain-from-food 20
wolf-reproduce 5 %

sheep 251
wolves 55
grass / 4 190

Figure 4. Input-output items, and agents on the simulation field

4. Simulation in education

In education, simulation can be applied in two ways, depending on whether the given simulation program exists already or the students have to design it themselves. First, let us take a look at the former case.

It is predominantly the disciplines of science that have available simulation programs; in fact, NetLogo's library abounds in these. Nevertheless, one can imagine cases in which even humanities could benefit from the usage of simulation programs: experiments that cannot be performed in real life for some reason could be taken to a simulation environment. Although in such cases observation is limited to vision and hearing, we can make advantage of the virtual environment: by setting the conditions more precisely, or controlling the display of the process, the results can be more closely observed.

Simulation is not without the didactic advantages of real experimentation. Illustrative and task-oriented, simulation forces students to draw their own conclusions, instead of providing them with dry facts. It is also suitable for differentiation: depending on the level of interest and expertise, simpler or more complex tasks can be designed. In addition, it makes room for varied types of classroom activities, from individual work to pair and group-work.

For the reasonable application of a simulation program, the students need to understand the model on which the program is based. This also requires them to be familiar with the features of the real-world system, which the model was designed to offer some explanation for. Effectively, they need to perform each step of the previously described method of understanding real-world phenomena.

As an example, let us elaborate NetLogo's simulation about *Changing Pressure* [5]. With the help of the program, we can observe the behavior of the gas particles and the correlation of the number of particles and pressure, with constant volume (Figure 5).

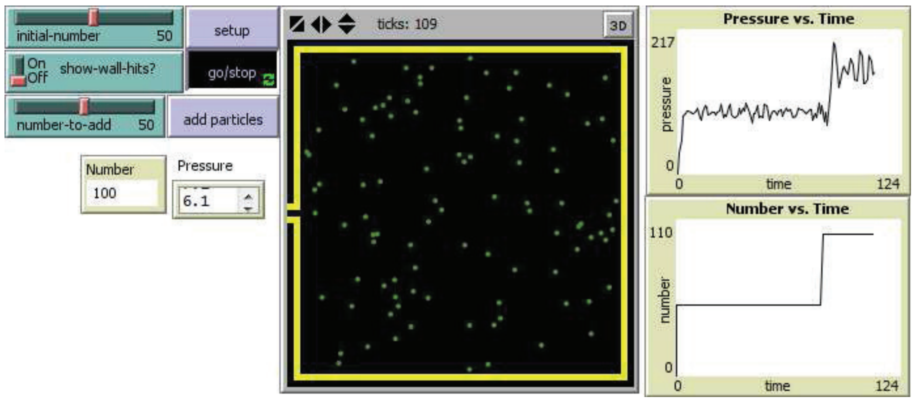


Figure 5. Changing Pressure (from NetLogo's library)

During the *observation* process, we can collect not only the perceptible features of the given real-world system but also the qualities of related everyday occurrences. Questions and tasks can motivate simple experiments or reflections on everyday experiences.

If, for example, the students examine the air within a football as a system, they can easily identify some of its features. On the one hand, the air automatically and evenly fills the space, without any need for forceful dispersing. On the other hand, the inflated ball is hard, because the air within can tense up the wall, applying pressure. It can also be concluded, using a pressure gauge or simply our hands, that an increase in the amount of trapped air increases the pressure as well.

Based on the observations, the students can formulate their own *hypotheses* about the operation of the system, in a verbal, graphic, or written form. While doing so, they may become more sensitive about the specific phenomenon and even find the hypothesis of the applied model. In the end, of course, it is necessary to reveal the valid hypothesis.

The features of the air trapped in the football can lead us to understand the behavior of the gas particles. The students can make “microscopic” drawings about the different states of the football. As a final conclusion, we can say that air is composed of quickly moving particles, which fill up the given space and, by bouncing off the wall, bring about the phenomenon of pressure.

Modelling and programming must be replaced by the *understanding* of *the model and the program*, respectively. These two activities coincide, because the simulation program is the illustration of the model itself. In this phase we need to get familiar with the objects and the rules of the model, its correspondences with the real-world system, along with the applied simplifications and possible parameters. It is useful to urge students to discover as much of the world of simulation as possible.

We can encourage students to connect the objects of the model with the elements of the inflated football. This will help them notice simplifications, such as the wall of the simulation space being square and rigid, as opposed to that of the football. The teacher can also add, as part of the verbal or written instructions, that the real-world system comprises of different gas particles, not just one kind, like in the simulation. The particles of the model are governed by the ideal gas law, which students will be able to define based on the simulation. Both the initial number of the particles and the numbers to add can be parameterized.

In the *validation* phase, we need to compare the events of the simulation with our real-life observations. It is a crucial step, because the users of the program cannot stay indifferent about the validity of the model, with which they aim to understand the operations of the original system. Occasionally, it can happen that a thorough validation would require initial observations for which school circumstances are not suitable. In such cases it is useful to call students' attention to the fact that the systemic phenomena spotted in the simulation take place in real life too.

If we introduce gas particles in an empty simulation space, after a short while they will fill the space evenly. In addition, they will create constant pressure by bouncing off the wall of the space. We will also notice that the increase in their numbers will affect the amount of pressure. All these experiences meet our expectations, which confirms the validity of the model.

After understanding the operation of the model, we can turn to *evaluating* it. We can set different initial states to test the results, but the settings for different final states can also be checked. It is worth controlling the process with instructions and questions.

We can check, for instance, how much time it takes to stabilize pressure, how to minimize fluctuations, and what kind of mathematical correlation holds between the number of particles and pressure.

5. Modelling and simulation in education

Theoretically, it is possible for students to perform each step of the process of understanding reality; that is, it is feasible to integrate modelling and designing a simulation program in the curriculum. It is especially useful for students with an interest in science, for whom simulation programming languages can prove to be practical tools later on.

Modelling calls for a thorough research about the real-world system, as a preparation, and serious planning, for the execution. It is highly probable that students will need to get acquainted with the sources (course books, articles, etc.) related to the phenomenon in focus, which can help bridge the observational gaps and give ideas about the hypothesis and the model. From these, we need to separate the assessments available about the system, because those pieces of information cannot be used as starting points, only as results to verify with the simulation.

It is not enough that our model is valid; it also needs to be simple and to-the-point so it can clearly illustrate the correlations. The secret of a good model is simplicity (even though in the beginning the simplifications may seem excessive).

For students, and generally for amateurs, modelling equals programming: it is the program that embodies and materializes the model, which means that the model will always conform to and be shaped by what the programming environment makes possible.

Modelling, and programming, can be an individual, a pair, or a group task, supported by some assistance from the instructor, but it can also be the fruit of a teacher-student cooperation. The available sample models and algorithms connected to the disciplines of physics, chemistry, or biology [6], [7], [8] can be used in the classes.

To illustrate the entire process of understanding reality from the beginning to the end, I am going to demonstrate a simulation program about stalactites, designed by myself.

It all began with a thorough *observation* of stalactites, which involved the photo analysis of different types of stalactites (of course, going to a cave would have been more real). Stalactites typically have an oblong shape and a pointy edge, but their dimensions vary.

It is a well-known fact – and this is already part of my *hypothesis* – that it is the calcareous water dripping from the ceiling that is responsible for the formation of stalactites, because a part of the lime sticks to the surfaces. Less people know – I was not familiar with it before either – that stalactites develop from straws, which are thin-walled, elongated, hollow formations. The real stalactite formation begins when the water flow moves from the inside to the sides of the straws.

Modelling proved to be a challenge. Simpler and simpler versions followed one another, until I reached the final one, which is as simplified as possible. Instead of the entire ceiling of the cave, I decided to focus only on one stalactite, giving up on straw formation as well. The complicated motion of the drops was replaced with a simple one, and even the more complex issues of chemical dissolution and precipitation were ignored.

Let us see what was left. The objects of the model are the moving drops, for one, and the grids, functioning as empty spaces and lime particles (depending on the color), for two. Notice the influence of NetLogo’s features on the model.

Each of the drops have a specific lime content as an own variable and a propensity to precipitate a lime particle (changing the color of the respective grid and decreasing its lime content with one) if the neighboring grid has some lime to stick to. The drops can appear in the uppermost line of the grid plane on either side of the stalactite in a random moment of time, then in every second it jumps to the next line. Nevertheless, if that specific line already contains a lime particle, it will pick the empty grid closest to the middle vertical axis (Figure 6). At the end (or edge) of the stalactite, the drop joins the axis, adding to its length (in case of precipitation).

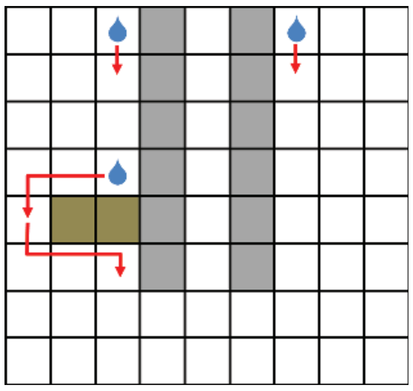


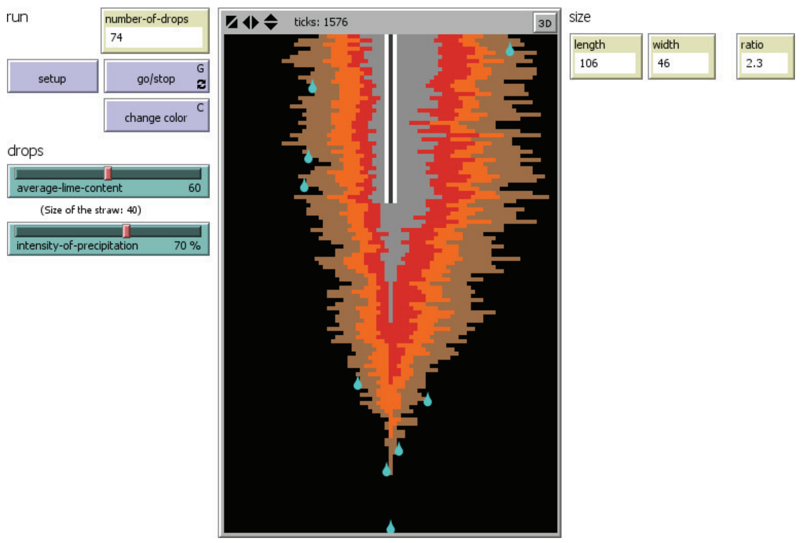
Figure 6. The motion of drops in the model

The average lime content of the drops and the intensity of precipitation can be parameterized. Also, some monitors were added to the *program*: such data are displayed on the screen as the number of drops entering the simulation space or the current length and width of the stalactite, along with their quotient. In addition, there is a button with which we can change the color of the lime particles, so to illustrate the layers of the stalactite and the levels of precipitation in different heights (Figure 7). (The program can be downloaded from `http://bernatp.hu/bernatp-stalactite-v1.0-EN.zip`)

When using the *simulation*, we need to be aware that in reality the intensity of lime precipitation is influenced by the amount of water and temperature. The increase in water accelerates the flow and decreases the intensity of precipitation, whereas the increase in temperature has the opposite impact on precipitation.

With non-extreme parameters, the program can create stalactites that resemble the real-world phenomena. Therefore, we can consider the model as *valid*, suitable for explaining stalactite formation.

With the help of the simulation, we can discover how the lime content of the drops or the intensity of lime precipitation affects the shape of the stalactites



number-of-drops 74

ticks: 1576

average-lime-content 60

(Size of the straw: 40)

intensity-of-precipitation 70 %

length 106

width 46

ratio 2.3

Figure 7. The development of the stalactite (own simulation)

(evaluation). For example, it can be noticed that in the presence of little water and intensive precipitation, the top of the stalactite will be heavy. On the contrary, an increase in water and a decrease in precipitation result in long, thin shapes. The initial lime content of the drops also influences the shape of the stalactite, which can be examined with the help of the simulation as well.

6. Recognition and understanding

Simulation is an educational method, which, just like experimenting, has a number of didactic advantages and offers various application forms, from the usage of a simulation program to the independent design of a model and a program. With its help students can not only familiarize themselves with a scientific phenomenon but they can also get acquainted with a method to understand reality. While gathering the information, they can also learn the path, of observing, hypothesizing, and proving or disproving. Eventually, they can not only understand but also experience that science is based on proven but constantly evolving theories. Such an experience would be priceless for anyone who is part of the educational process.

References

[1] L. Horváth, P. Szlávi and L. Zsakó, *Modellezés és szimuláció (Modelling and simulation)*, Mikrológia 1, ELTE IK, 2006.

[2] U. Wilensky, NetLogo flocking model, 1998, `http://ccl.northwestern.edu/netlogo/models/Flocking`.

[3] U. Wilensky, NetLogo, 1999, `http://ccl.northwestern.edu/netlogo`.

[4] S. Tisue and U. Wilensky, NetLogo: Design and implementation of a multi-agent modeling environment, 2004, `http://ccl.northwestern.edu/papers/agent2004.pdf`.

[5] U. Wilensky, NetLogo connected chemistry 2 changing pressure model, 2004, `http://ccl.northwestern.edu/netlogo/models/ConnectedChemistry2ChangingPressure`.

[6] Cs. Szabadhegyi, P. Szlávi and L. Zsakó, *Szimulációs modellek a fizikában (Simulation models in physics)*, Mikrológia 17, ELTE IK, 2005.

[7] G. Siegler, I. Pintácsi and L. Zsakó, *Szimulációs modellek a kémiában (Simulation models in chemistry)*, Mikrológia 15, ELTE IK, 2004.

[8] P. Szlávi and L. Zsakó, Szimulációs modellek a populációbiológiában (Simulation models in population biology), Mikrológia 9, ELTE IK, 2006.

PÉTER BERNÁT
DOCTORAL SCHOOL OF INFORMATICS
EÖTVÖS LORÁND UNIVERSITY
HUNGARY

E-mail: `bernatp@inf.elte.hu`

(Received March, 2014)