

Teaching of Biology Supported by Mathematical Models

Nauczanie biologii wspomagane modelami matematycznymi

Joanna Maciejewska¹, Janusz Uchmański²

¹ Center for Ecology and Ecophilosophy, Cardinal Stefan Wyszyński University in Warsaw

² Institute of Biological Sciences, Cardinal Stefan Wyszyński University in Warsaw

ORCID: JU <http://ORCID.org/0000-0001-8087-8371> • j.uchmanski@uksw.edu.pl

Received: 21 Jan, 2023; Revised: 28 Mar, 2023; Accepted: 31 Mar, 2023

Abstract: In the life sciences, the use of mathematical modelling and computer simulation methods has brought many benefits. This is the case, for example, in physics. Biology makes very little use of these methods. One of the reasons for this may be that the biology teaching process is conducted in such a way that it discourages young people with an interest in mathematics and computer science. We propose to support the process of teaching biology by introducing simple elements of mathematical modelling of ecological processes. In this work, we showed what benefits would be the application of the *NetLogo* program in teaching biology in secondary schools. *NetLogo* is a multi-agent programming language designed to simulate complex phenomena. It is intended for both research and education and is used in a wide variety of scientific disciplines. We used the *NetLogo* program to create a model of searching for food by ants. We have shown that teaching biology by building a computer model forces a deeper understanding of the problem we are working on and uses and develops the creative abilities of young people.

Keywords: teaching biology, computer simulations, *NetLogo*

Streszczenie: W naukach przyrodniczych zastosowanie metod modelowania matematycznego i symulacji komputerowych przyniosło wiele korzyści. Tak jest na przykład w fizyce. Biologia w bardzo niewielkim stopniu wykorzystuje te metody. Jedną z przyczyn takiego stanu rzeczy może być fakt, że proces nauczania biologii jest prowadzony tak, że zniechęca młodzież z zainteresowaniami matematycznymi i informatycznymi. Proponujemy, aby proces nauczania biologii wspomóc poprzez wprowadzenie prostych elementów modelowania matematycznego procesów ekologicznych. W niniejszej pracy pokazaliśmy, z jakimi korzyściami wiązałoby się zastosowanie programu napisanego w języku *NetLogo* w nauczaniu biologii w szkołach ponadpodstawowych. *NetLogo* to wieloagentowy język programowania stworzony do symulacji złożonych zjawisk. Jest przeznaczony zarówno do badań, jak i edukacji. Jest używany w różnych dyscyplinach naukowych. Wykorzystaliśmy program *NetLogo* do stworzenia modelu poszukiwania pokarmu przez mrówki. Wykazaliśmy, że uczenie biologii przez budowę modelu komputerowego, zmusza do głębszego zrozumienia problemu, nad którym pracujemy, wykorzystuje i rozwija twórcze zdolności młodzieży.

Słowa kluczowe: nauczanie biologii, symulacje komputerowe, *NetLogo*

Introduction- what is mathematics and how is it used in physics and biology?

Nature can be described in different languages. Nature can also be studied in various ways. It can be studied by observing it or by experimenting with it. Then we use natural language to describe it, as biology has been doing for many centuries, which is mainly an empirical and descriptive science (Uchmański 1980). One can also study nature by building its mathematical models and analysing the properties of these mathematical structures (Heller 2014). Then we use mathematical language to describe nature. This is what physicists do, which have been using the language of mathematics for centuries (Tegmark 2014). This has brought great benefits to physics. Physics owes its incredible development, especially in the 20-th century, to the use of the language of mathematics to describe inanimate nature. However, it should be remembered that both ways of studying nature contributed equally to this success of physics – by constructing mathematical models and conducting experiments. A reflection of this is the division of physics into experimental and theoretical one.

A mathematical description or model of physical reality allows us to formulate hypotheses, predict future states of the system, and also allows us to discover the existence of “things” that, without looking at the mathematical structure, it would be difficult for us to find and understand among the enormous complexity of nature. For example, the concept of mass is needed to formulate Newton’s equations of motion, so we can describe them through mathematics. Thanks to it, we know what mass is and why it must appear in the equations of dynamics.

Much, if not most, of mathematics was created for physics. Mathematics and physics benefit from each other’s existence. In the case of biology, another important natural science, the situation is completely different. As we have already mentioned, it is mainly a descriptive science, using natural

language. Mathematical language is used in biology only to a very limited extent (Uchmański 2015). It is also difficult to indicate any new branches of mathematics, that emerged to meet the needs of biology.

1. Computer-assisted teaching of natural sciences

The fact that biology is still a descriptive science using natural language means that the process of teaching biology from primary school to university studies is also conducted using these methods. It relies heavily on paying attention to details, emphasizing their memorization, and neglects the development of the ability to see general patterns, generalize and look at biological systems holistically. As we have already mentioned, physics owes its very rapid development to the use of mathematical methods, and biology has never taken advantage of this opportunity. To some extent, this is related to the traditional way of teaching biology, which discourages young people interested in mathematics and computer science from biology.

The current, widespread interest in nature conservation, especially among young people, which ultimately brings various benefits, is to a large extent conditioned by social media and a very general feeling that civilization poses threats to the environment. However, this is a rather superficial view, and it is doubtful whether it will contribute to anything more than technological advances in the future. It can be assumed, that further development of biology as a science, in addition to the development of experimental methods, should involve the use of mathematical methods. The resulting increased interest of young people with mathematical talents in science and biological studies could in future result in the fact that both teachers at various levels and scientists dealing with biology will be people who adopt the way of thinking about natural sciences that prevails in physics.

In this work, we would like to propose adopting a new perspective on teaching

biology in various-level schools and on introducing methods based on the use of computers. Computers have been used in teaching physics for a long time (Mos-towski, Natorf and Tomaszewska 2002). However, teaching of physics, unlike biology teaching, is at a greater disadvantage when it comes to the use of computers. Well, in physics, to put it very briefly, almost everything is already known. Mathematical language has been present there for a long time. Most physical phenomena have been mathematically described. Therefore, in teaching of physics computers are mainly used to conduct virtual experiments. The role of students is then quite passive here. They observe the course of a certain physical process visualized on the screen of a computer monitor. Of course, they can modify its course by changing the parameters of the mathematical model, but there is little left to discover.

In biology, and especially in ecology, the situation is different. Very few biological phenomena and processes have been described in mathematical language. Thus, the study of biology can consist in students' active participation in building a mathematical model of a biological phenomenon or process. It may even be a mathematical model that no one has ever tried to build. Building a model from scratch requires asking questions, thinking about the best solution and looking for alternative solutions. It will be a process consisting in decomposing the described process or phenomenon into its component parts, but at the same time in holistic thinking about its various aspects. Thus, it will be a creative activity, which makes it even more attractive.

This way of teaching is subject to several preconditions. Availability of computers – both at home and in educational institutions – is not currently a problem. Of course, it is necessary to provide appropriate software for these computers. It must be easily accessible and easy to use so as not to discourage users who are not very familiar with computer technologies. It is also necessary

to choose appropriate branches of biology, where the use of mathematical language and computer applications is particularly justified and easy.

2. What mathematical methods are used in ecology?

It is more or less known, what falls within the scope of biology and what becomes the object of interest in physics. Traditionally, in the part of animate nature, we distinguish molecular and cellular levels, physiology, individuals and populations, communities, ecosystems and the entire biosphere. In many places mathematical descriptions known from physics can be applied in biology. Many phenomena and processes known from physics also take place in biology. However, it is usually the case that the same physical phenomena that we know from physics are much more difficult to describe mathematically in biology. For example, the flow of blood through blood vessels is a problem in hydrodynamics. However, neither the blood vessel nor the blood resembles the simple flow of an ideal liquid through a rigid pipe.

Since the beginning of the last century, mathematical modelling methods have been introduced to ecology, understood as the science of biodiversity, and today ecology is the most mathematized branch of biology. Since mathematical methods in ecology were mainly introduced by physicists (Kingsland 1995), they were taken from physics. Traditionally in physics, differential and difference equations are used to describe the dynamics of systems. The same method was also used to describe the dynamics of populations and multi-species communities. The consequence of this was the assumption that the state variable whose changes were described by the above differential and difference equations was population density, i.e., the number of individuals per unit of space. And further, this led to the assumption that all processes that affect population dynamics depend on the density of this population and

the populations associated with it by ecological interactions. The belief in the universality of dependence on density is the main paradigm of ecology (Uchmański 2015).

One may ask, however, whether density is a sufficient characteristic of the population, or perhaps it is an oversimplification. The density relationships so favoured by ecologists are true, but only in terms of mere statistical relationships, not causal relationships. Let us also note that with such assumptions as to the type of mathematical methods used, we assume a kind of continuity of the structure of the system, which together with the continuity of the time course of processes may give a false picture of reality (Uchmański 2022).

This issue has raised doubts in recent decades (Ovadia et al. 2007). Perhaps the traditional mathematical methods used so far in ecology are not suitable for describing the dynamics of ecological systems. Let us look at populations. Their basic component are individuals of species forming an ecological system. Biological individuals are the product of natural selection, which boils down to the fact that they have a certain task to perform, a life “purpose” that they must fulfil. The goal of an individual from the moment of its emergence is growth and development as well as reproduction, i.e., production of copies of itself. Each individual, developing, goes through successive developmental stages, i.e., its properties change over time. To develop and produce offspring, individuals need resources, for which most of them have to compete with other individuals. That is, each individual from the population during its life is guided by certain strategies and makes decisions to survive and reproduce. Biology is largely algorithmic (Uchmański 2020). From this description it is clear that a biological population is certainly not like a physical population, for example a “population” of identical gas particles. In a closed vessel, the gas molecules are the same, and in a population, each individual is different.

Therefore, according to many ecologists, a better way to apply mathematics in ecology is the so-called individual-based modelling (Grimm and Railsback 2005). It seems that when we take into account the fate of each individual in the population, and not the density, which by definition is an average value, we seem to create a correct picture of systems dynamics. Each individual has its vicissitudes of life, and all individuals, to varying degrees, interact with others. The fate of individuals, their ability to survive and reproduce affect interactions among individuals, their number in a given population and the dynamics of the entire ecological system.

Looking at the above arguments, we can see that the use of mathematical language can help us understand the functioning of ecological systems, and thus give us great practical benefits, only we need to modify the repertoire of mathematical methods used in ecology and not directly apply the methods used in physics.

In physics, the use of computers for research, description and teaching is already common. Ready-made programs, models of physical problems and virtual experiments have been used for years to teach successive generations of physicists. In physics, we have recognized mathematical models showing the functioning of physical systems. Looking at this, we can certainly say that this branch of science is already somewhat less attractive to researchers.

Contrarily, adaptation of mathematical methods to descriptions of ecological processes and phenomena may pose a challenge. There are still many mysteries in ecology that are yet to be solved. For example, just creating models explaining the co-occurrence of all species of animals and plants that we know about, starting from our small garden and ending with the still unexplored Amazon forests, could be a task for next generations.

Nature, especially when viewed from the perspective of an ecologist, is extremely diverse. Individuals are diverse. They differ in age, sex and stages of development,

and even when they do not differ in these characteristics, they will still be different. The number of species inhabiting the Earth is estimated at tens of millions, although we do not know the exact number. Environments in which these species live are also characterized by great diversity. Even if we are interested in a specific problem, for example interspecific competition, this huge variety of living nature gives us the opportunity to build many unique models that no one has ever tried to build. This makes ecology a very attractive field for “modellers”.

3. What are computer simulations?

The above considerations encourage us to use individual-based models in ecology, which by their nature will be simulation models created on computers. Computer simulation is one of the forms of solving a problem that can be presented in the form of a mathematical model, but it is not possible to solve it using analytical methods. Computer simulation is an approximate and numerical reproduction of the phenomenon or behaviour of a given object using its computer model, very often additionally illustrated with computer graphics. Such a model, using appropriate algorithms, imitates the reality (Cormen, Leiserson and Rivest 1994).

Simulation is considered a specific method of investigation. Similarly to classical mathematical modelling methods, computer simulation can be treated as an alternative to experimental and observational techniques when phenomena are difficult to observe, for example because of the time scale in which they occur, or too difficult and expensive when it comes to experimental research. Computers have been used in technical and natural sciences other than biology for several decades. Their use in physics is particularly interesting. Here, however, their use is different than in ecology. There are problems in physics that can be well formulated in the form of mathematical equations, but they are either too difficult to solve analytically and we do not know the solution yet,

or by the very nature of the problem such an analytical solution does not exist and we have strict evidence for it. One such problem in physics is the three-body problem.

Simulation is one of the most promising methods of mathematical modelling of natural phenomena today. As has been mentioned above, nature can be studied in general in two ways: by observing it and experimenting with it, or by building mathematical models of the observed phenomena and processes, and then examining the properties of these mathematical models. Computer simulations are one of the methods used in the construction of mathematical models in natural sciences and it must be said that they are very effective where other methods of mathematical modelling fail.

In practice, the use of computer simulation methods is as follows: researchers studying, for example, the temporal evolution of a system, figuratively speaking, stop time, calculate all the interactions taking place in each modelled object at a given moment, let time flow again and repeat all these operations many times, using computer programs. This iteration process continues until we obtain the information we need about the evolution of the system under study. The only problem with this approach is that the error of such a result is difficult to estimate because computer simulations are approximate solutions. However, here experiments and observations of nature come in handy as an old and very effective method of verifying scientific hypotheses.

Methods of mathematical modelling, as we already know, are also used in ecology. Often, due to the temporal or spatial scale as well as the complexity of the problem, it is the only comprehensive method of understanding ecological systems. Let us look at this problem on the example of the lake ecosystem, which we often study in terms of eutrophication of its waters. We can distinguish two approaches here: the study of the circulation of elements in the lake and the study of the demographic processes

taking place in the lake by tracking changes in the population of species inhabiting the lake.

In the first approach, when we study the circulation of elements, the differential equations well known from physics are sufficient for the mathematical description of the processes taking place. This is because, in fact, we are dealing with physical processes, and the observed phenomena are the result of the collective action of a very large number of organisms inhabiting the lake. However, when we treat the same lake as a habitat for fish and plankton and we are interested in the numbers of these groups of organisms, then we have to use computer simulations. At this point, the algorithmic nature of many ecological processes is revealed. Populations are composed of individuals, and the life of an individual, as we mentioned earlier, is from a mathematical point of view, an algorithm implemented by that individual. The complication is that this algorithm must be implemented by an individual taking into account interactions with other individuals of the same and other species. Such a system cannot be described by traditional mathematical modelling methods. We have to use computer simulations.

4. The process of creating a model and its didactic significance

The main thesis of this work that we would like to present is as follows: we learn biology by building computer models of biological systems. This forces us to think about the modelled object and try to understand its functioning. Then, the results of the model verify our ideas about the modelled object with the information we know from experiments and observations of nature. Although the computer program itself may be simple, solving a specific biological problem often requires a great deal of biological knowledge and understanding of the biological problem. We suggest that the process of creating a model should start with asking some questions. Thanks to them,

we will understand whether we have enough information to create a model, or whether we need to learn more to find answers to the questions that bother us.

Asking questions and looking for answers, get bring many benefits. We learn by solving an ecological problem, by trying to understand it. Such use of computer programs can be helpful in education. When a student is given the task of creating a certain model, even a small one, he will start thinking about solving this problem. While looking for solutions, reading biological literature, he learns the necessary concepts and consolidates his knowledge. Purely theoretical knowledge that students gain, can be quickly forgotten, while practical knowledge acquired during the process of solving a problem may teach them thinking and allow to remember the essence of the solved problem for longer.

5. NetLogo

As we already know, due to the fact that the basic units that create ecological systems are individuals, in the last three decades, computer simulations in ecology have taken the form of so-called individual-based modelling. Sometimes, especially in the social sciences, we speak of agents instead of individuals, but in ecology we will use the term individuals. Each individual executes the assigned algorithm separately, but in interactions with other individuals, and it contributes to the final result of the simulation.

One of the most commonly used programs today for creating computer simulation models describing the behaviour of systems composed of many lower-order units, each of which executes its own program to achieve a certain task, is *NetLogo* (Wilensky 1999). This program uses an object-oriented language, i.e., a language that provides the programmer with ready-made objects that, in order to work properly, must have the features needed in a given model. This language, along with the entire environment containing tools for easy writing

of programs, allows us to perform a simulation combined with the visualization of results.

Unlike other object-oriented languages, *NetLogo* does not require extensive IT knowledge (Uchmański and Zielińska 2010). That is why, *NetLogo* is one of the most popular programs in scientific and didactic applications (Railsback and Grimm 2012; Murphy, Ciuti and Kane 2020). In addition to being easy to use, it offers the user many ready-made programming solutions, and at the same time creates great opportunities to write models of the functioning of systems characterized by a diverse internal structure.

Due to its specificity, *NetLogo* is an excellent tool for building simulation models of ecological phenomena and processes. It primarily enables the creation of individual-based models. Using *NetLogo*, we will be able to create such a model faster than in traditional languages, and the program code will be simpler and more understandable for an observer.

The basic objects of *NetLogo* are: individuals (the so-called “turtles”) and the space in which they “live”. A model built using *NetLogo* can describe phenomena occurring in two- or three-dimensional space. The results obtained from the model can be presented both in graphic form (charts, histograms) and in numerical form. The *NetLogo* interface allows us to use ready-made elements (buttons, sliders, switches) that can be placed on the screen by “drag and drop”. This allows us to change the parameters of the model during the simulation using these sliders and switches. The program also offers a code validation function, which allows users to avoid the most common syntax errors. Due to the fact that the program was created in its first version at the end of the nineties and is widely developed, it has very extensive and reliable documentation that helps to familiarize with its capabilities and suggests many ready-made solutions.

The first steps in using it do not require any programming background, because the program is intuitive and all elements of the *NetLogo* language are described in the “HELP” tab, therefore the only problem may be knowledge of English. Although, even this is only required in a basic dimension.

6. Model - biological basis

In order to show the benefits of building mathematical models in biology, we focused on how ants search for food, which will illustrate one of the hypotheses formulated by ecologists studying these issues. It seems that the most effective way of searching for an unknown place with food is random movement in the area, which is why this way of exploration was adopted in the model.

First, the ant goes in search of food and does so in a random way. This means that it randomly changes the length of the road segment as it moves forward, and at the end of such a segment randomly changes the direction of further travel. The moment it smells food from a distance, it tends towards the maximum intensity of its smell and takes a bit of food with it. When describing the return mechanism, it was assumed that ants have a directional compass. At the stage of searching for food, at each turn at the end of a straight section of the road, the ant “turns” its head, “looks” at the sun and orients itself in relation to the characteristic points of the terrain and thanks to this it knows where and in which direction the nest is. Perhaps it works so that the ant remembers in which direction the nest is in the previous step, and in the next step it orients itself in relation to this previous position, knowing the position of the sun and remembering the characteristic points, it knows the direction to the nest. After obtaining food, it returns along this direction, leaving a trail in the form of a scented substance.

7. Model algorithm

Ants created with *NetLogo* will live in a flat, two-dimensional space, made up

of many square cells arranged side by side. Such a spatial structure is called a cellular automaton. In *NetLogo*, cells are called *patches*. Each cell has its own x and y coordinates. The origin of the coordinate system is located in the lower left corner of the cellular automaton. There will be 90 cells along each axis of the coordinate system, so the cellular automaton will have a size of $90 \times 90 = 8100$ cells. All this is set in the *NetLogo* interface. The ants will move from cell to cell. This will be the size of the spatial step in the model. The simulation of ants' movement will take place in discrete time steps called *ticks*. We will use also singular form for above words – *patch* and *tick*. This distinction between the singular and the plural is important because one can refer to a single cell or a single time step while commanding all cells and all time steps at other times.

We start writing a program by determining the names of the variables used in the program. The instruction *globals* is used for this. It has been written in the following form:

```
globals [ p d f ]
```

where p , d , f are the names of the variables used in the model. Their meaning will be explained below. What lives in space is traditionally called *turtles* in *NetLogo*, but these objects, i.e., individuals from a biological point of view, can be given another name. We will call them *mrowki* (ants in Polish are *mrówki*) and use the *breed* instruction, which we will write in the following form:

```
breed [ mrowki mrowka ]
```

The singular and plural forms are also given here.

Each patch will have a certain characteristic which we have called *zapach* (smell in Polish). Its numerical value will mean the intensity of the smell of the food the ants are looking for. We inform the computer of this fact using the following instructions:

```
patches-own [ zapach ]
```

The first part of the program ends with an instruction whose meaning will become clear in a moment. It is related to the assumption that was made earlier and concerns the way in which ants remember the direction of returning to the nest. It looks like this:

```
directed-link-breed [ streets street ]
```

After these introductory instructions, the body of the program needs to be filled with content. It always consists of two fixed parts. From the part called *setup*, where we arrange the model space and set the initial conditions for the simulation. It starts with an instruction:

```
to setup
```

This means that everything the computer encounters now will refer to the *setup* part. It will then end with the *end* statement. Let us remind that the computer reads and executes the instructions line by line from top to bottom. The first statement we will fill in *setup* will be:

```
ca
```

It is an abbreviation of the phrase clear all. It will clear the space where the ants are moving and prepare it for the simulation. Next, we will ask the *patches* that make up the space to be coloured the same colour. All colours in *NetLogo* have their numbers, but some of them can be described by names. We will choose light green, which is colour number 68:

```
ask patches [set pcolour 68]
```

where *pcolour* is the colour of the *patch*. In the *patch* with coordinates (70,70) we place the food sought by the ants. We will mark it with a different colour (it will be a slightly darker shade of green with the number 65) and a small intensity of 100:


```
ask patch 70 70
[ set pcolour 65 set zapach 100 ]
```

The smell spreads throughout the space, but its intensity decreases as one moves away from the place where it was placed. We assumed that the smell decreases exponentially (hence the *exp* instruction) with the distance from the cell with coordinates (70,70), and the speed of smell decay is determined by the *alfa* parameter. In the next instruction, we ask the computer to calculate for each cell its distance *d* from the cell at coordinates (70,70) (*distancexy* instruction) and the intensity of the smell in that cell, taking into account the distance to the source of the smell (the auxiliary variable *f* is used for this purpose). If this intensity is less than 1, the computer will assume that the smell in that cell is equal to 0. If the smell intensity is greater than 1, then the cell will have an smell of *f* and the corresponding colour (*ifelse* instruction allows us to do this). We will do all this thanks to the instructions: 0

```
ask patches
[ set d distancexy 70 70
if d > 0 [set f 100 * exp (- alfa * d)
ifelse f < 1 [set zapach 0] [ set zapach f set
pcolour 65 * exp ( - 0.005 * alfa * d) ] ] ]
```

In this instruction, in addition to giving the cells a smell as a certain numerical value depending on the distance from its source, the cells for which this value is greater than or equal to 1, have been coloured with different shades of green that change depending on the distance from the source of the smell.

Next, we will mark the place where the ant nest is located. We will put them in a patch at coordinates (25,25), which we will colour blue, but we will also colour all the nearest neighbours of the nest in the same way to make this place more visible. We will use the following instructions for this:

```
ask patch 25 25 [set pcolour blue]
ask patch 26 25 [set pcolour blue]
```

```
ask patch 24 25 [set pcolour blue]
ask patch 25 26 [set pcolour blue]
ask patch 25 24 [set pcolour blue]
ask patch 26 26 [set pcolour blue]
ask patch 24 24 [set pcolour blue]
ask patch 24 26 [set pcolour blue]
ask patch 26 24 [set pcolour blue]
```

We will now bring to life two ants (*create-ants*). We need to determine their screen size (*size*), colour (*colour*) and position (*xcor* and *ycor*) at the initial moment. We will do this using the following instructions:

```
create-mrowki 2 [ set size 1 set colour yellow set xcor 25 set ycor 25]
```

One can see that the ants will be yellow and at the beginning both will be sitting in the nest, i.e. in the cell with the coordinates (25,25). Since we did not specify the direction, the ants are facing, the computer will automatically cause them to “look” north, i.e. vertically up the monitor screen. The ants are automatically numbered by the computer: one will be numbered 0 and the other 1. The value of the variable *p* will tell you whether the ants have found food or not. First, we assume that the ants have not found food:

```
set p 0
```

which means that $p = 0$. We finish the setup part with the *reset-ticks* instruction, which resets the simulation time step counter, and with the end instruction.

We now turn to the main part of the program called *go*. We start it with the following instruction:

```
to go
```

We will leave the ant number 0 in the nest. It will be the queen. The ant number 1 is a forager. We will ask her to go in search of food (*ask ant 1*). First, we will consider the situation when the ant has not found food, i.e., $p = 0$. If this is the case (*if p = 0*)

and if the forager is in a patch for which the *zapach* > 0, it will move in the direction of increasing smell intensity (*uphill zapach*), and if it reaches the patch with the maximum smell, then the value of the parameter *p* changes to 1, which means that the ant has found food. However, if a food-seeking ant stands on a patch where *zapach* = 0, it advances forward (*fd*) a random number of steps (*random odleglosc + 1*, *odległość* is distance in Polish), then rotates (*rt*) a random angle to the right (*random kat*, *kat* is angle in Polish) and then a random angle to the left (*rt random (- kat)*). The patch on which she performed these rotations is marked in red (*ask patch-here [set pcolour red]*). These two different ant behaviour choices are implemented using the instruction *ifelse zapach > 0*. In total, it looks like this:

```
ifelse zapach > 0 [uphill zapach set p 1] [fd
  random odleglosc + 1 rt random kat
  rt random (- kat) ask patch-here
  [set pcolour red] ]].
```

The values of the *odleglosc* and *kat* parameters, as well as the previously used *alfa* parameter, are set using sliders available in the *NetLogo* interface, and their activation is very simple. The instruction *random odleglosc + 1* was used. This is because if you wrote *random odleglosc*, the ant would march forward a random number of steps from 0 to the *odleglosc*. Since we do not want the ant to stand still, we add 1 to the *odleglosc*. Then the ant moves at least to the adjacent patch. Another trick we used was asking the ant to turn twice, once to the left and then to the right at a random angle each time. In this way, we provided the ant with the possibility of random rotations in both directions from the current direction of movement. At each point where the forage ant changes direction, we ask it to “look” in the direction of the nest at the location of the queen and set the direction to this place. The *street* instruction is used for this,

and the full command we give to the forager is as follows:

```
ask mrowka 1 [create-street-to mrowka 0].
```

To make the above actions visible on the screen, we will use the following command:

```
ask mrowka 1 [show one-of my-in-links].
```

This will draw a line from the place where forager was turning to where the queen is. If the ant finds food (*if p = 1*), we ask ant number 1 (*ask ant 1*) to turn towards where the queen is (*facexy 25 25*), move forward in small steps (*fd 1*) and to left a pheromone trace of the way it travelled. This last action is mimicked by the line drawn by the moving ant on its way to the nest (*pen-down*). If the returning ant hits a patch in blue (*if [pcolour] of patch-here = blue*), *p* is set to 3. The command looks like this:

```
if p = 1 [ask mrowka 1
  [facexy 25 25 pen-down fd 1
  if [pcolour] of patch-here = blue [set p 3] ]].
```

If *p* = 3 then the program terminates:

```
if p = 3 [stop].
```

The part of the program called *go* ends with the *tick* and, of course, the *end* instructions. The entire program looks like this:

```
globals [ p d f ]
breed [ mrowki mrowka ]
patches-own [ zapach ]
directed-link-breed [streets street]
to setup
  ca
  ask patches [set pcolour 68]
  ask patch 70 70 [set pcolour 65 set zapach
    100]
  ask patches [ set d distancexy 70 70 if d >
    0 [set f 100 * exp (- alfa * d) ifelse f < 1 [set
    zapach 0] [set zapach f set pcolour 65 * exp
    (- 0.005 * alfa * d) ] ]]
```

```

ask patch 25 25 [set pcolour blue]
ask patch 26 25 [set pcolour blue]
ask patch 24 25 [set pcolour blue]
ask patch 25 26 [set pcolour blue]
ask patch 25 24 [set pcolour blue]
ask patch 26 26 [set pcolour blue]
ask patch 24 24 [set pcolour blue]
ask patch 24 26 [set pcolour blue]
ask patch 26 24 [set pcolour blue]
create-mrowki 2 [ set size 1 set colour yellow
  set xcor 25 set ycor 25]
  set p 0
  reset-ticks
  end
  to go
    ask mrowka 1 [
if p = 0 [ ifelse zapach > 0 [uphill zapach set
  p 1]
[fd random odleglosc + 1 rt random kat rt
random (- kat) ask patch-here [set pcolour
red] ]]]
ask mrowka 1 [create-street-to mrowka 0]
ask mrowka 1 [show one-of-my-in-links ]
if p = 1 [ask mrowka 1 [facexy 25 25 pen-
down fd 1
if [pcolour] of patch-here = blue [set p 3]]]
if p = 3 [stop]
tick
end

```

Figure 1 demonstrates an illustration of how *NetLego* shows the user the modelled object and graphically marks the results of the simulation.

8. Model properties analysis

Once we have a model, we will analyse its properties. This can be used to understand its functioning, to check its correctness, but it can also be useful in discovering such properties of the modelled object that we did not expect before, or that were difficult to notice while observing the real object.

A simple example of the analysis of the properties of the model presented in this paper may be the study of how the time of searching for food depends on the intensity of the signal about its presence. The *zapach* variable, the value

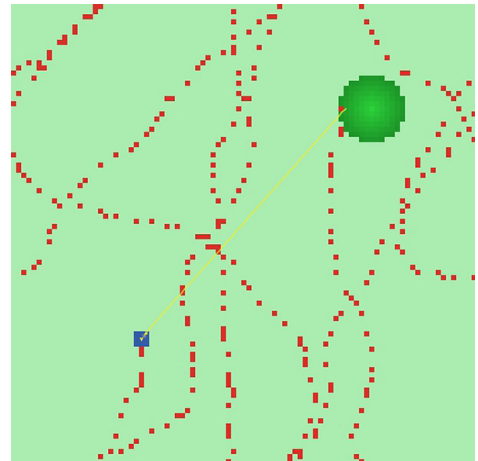


Figure 1. Illustration of the space in which an ant searches for food. The oval, darker patch in the upper right corner is a place where food is located. In this area *zapach* > 0. The remaining lighter area is the area where *zapach* = 0. The larger darker square in the lower left corner is the ant nest. The queen is there. The small darker squares are patches where the forager changed direction. A clear straight line is the return path of the forager to the nest. Simulation for *alfa* = 0.4, *odleglosc* = 4, *kat* = 30

of which is assigned to each cell of the space in which the ants move, is just such a signal. The place where the food is smells strongest. The intensity of the smell decreases as we move away from this place. As we remember, the size of the area in which the *zapach* > 0 is regulated by the *alfa* parameter. The higher the value of this parameter, the faster the smell disappears as you move away from the place where the food is.

A cycle of simulations was carried out for two different *alfa* values: 0.4 and 0.7. The remaining parameters of the model were the same in all simulations: *odleglosc* = 4 and *kat* = 30. For each *alfa* value, 70 simulations were performed, and the results were presented in the form of distributions of food search times (Fig. 2).

It can be seen that increasing the range of the food signal clearly reduces the time of searching for food. This result is seemingly obvious, but only the model allows us to numerically assess how large this

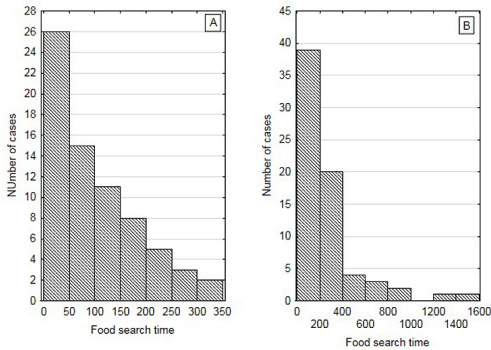


Figure 2. Distribution of food search times.

A – $\alpha = 0.4$, $odleglosc = 4$ and $kat = 30$. Average foraging time – 101.2-time steps, minimum time – 20 time steps, maximum time – 336 time steps, standard deviation – 81.8, skewness coefficient – 1.09. **B** – $\alpha = 0.7$, $odleglosc = 4$ and $kat = 30$. Average foraging time – 241.2 time steps, minimum time – 21 time steps, maximum time – 1401 time steps, standard deviation – 280.3, skewness coefficient – 2.39

reduction in search time is. It is also worth noting that the results of the model indicate significance of random events. This is the result of the method of searching for food adopted in the model, namely random penetration of space. This is reflected in quite large values of standard deviations. In both cases, however, the shortest foraging times are almost identical (20- and 21-time steps). However, the longest times differ significantly (336- and 1401-time steps). This draws attention to the so-called individual variability, which arises among individuals as a result of their interactions with the environment and other individuals. It is also worth emphasizing the effectiveness of the method of random penetration of the environment in order to search for food. Both distributions are clearly positively skewed. They are dominated by the shortest times of searching for food, and in every simulation an ant looking for food found it.

In the example above, the impact of environmental arrangement (more precisely, the distribution of information about food in the environment) on the effectiveness

of foraging for food was analysed. Other features of the model can be analysed in a similar way. For example, the way of searching for food. By maintaining its random character, the influence of the *odleglosc* and the *kat* parameters can be studied. Increasing the value of the first one means extending the rectilinear jump in space, but at the same time reducing the number of turns, while increasing the value of the second one means an increase in the dynamics of the forager movements. Are there optimal values for these parameters that lead to the shortest foraging times? The answer to this question could be attempted by analysing the properties of the model presented in this work.

Conclusions

Creating the program using the *NetLogo* tool forced us to search for information needed to build the model, which at the same time allowed the author of the model to enrich his knowledge about the modelled object. At the same time, it was not a chaotic search for information, but one focused on solving a specific problem. The program allows one to systematize information about the tested object. If we followed the path of further expansion of the model, this process would deepen systematically, but it would still be specific and systematized information. At the same time, it allowed us to realize how many facts about the modelled object we still do not know. In the course of this search, an additional effect appeared, which, in our opinion, was of considerable didactic importance. We will call it arousing interest in the modelled object. Well, we created a model of random ants foraging for food. Inspired by this topic, we started to look at all kinds of information and discovered other facts unrelated to the modelled object, which confirmed our belief that ants are really unusual insects.

Our example shows that the use of a tool such as *NetLogo* in teaching biology could be very developing for students who would be given the task to create their own model.

Just as we discovered how much interesting and biology can be, so students could discover their interests and passions.

Children at school age should develop curiosity about the world. In school, we mostly accept knowledge without asking why it is the way the teacher tells us. Because of this, we do not remember much information. In modern education, teachers should use various teaching methods in teaching children and young people. The use of *NetLogo* can be attached to the method of a curious child described in Radek Kotarski's book *Włam się do mózgu* [Break into the brain] (Kotarski 2017). This method consists in turning a student into a curious and creative child who creates a context for the situation, asks questions and looks for answers to these questions. A child who takes an active part in finding information by asking "Why...?", "How is it possible ...?", discovers the meaning of this information and due to the fact that the information becomes a logical consequence of the questions asked, it is easier for him to remember it.

We believe that such a process is best carried out only in a secondary school or in the last grades of primary school, because at an earlier age children might not be able to cope with the IT task of building a model. In addition, the Internet, which is now the first place to look for information, contains a lot of contradictory or incorrect information that younger children would find difficult to assess adequately. On the other hand, in high school, young people should discover their life path, passions and interests that they could develop in the future. In addition, older children already have sufficient knowledge of mathematics and foreign languages to undertake the task of building a computer model. They are able to adopt a more critical approach to evaluating the information from various sources. *NetLogo*, due to its simple structure and ease of use, could arouse children's passion for learning biology, and at the same time be an Introduction to teaching

programming, which may prove useful when teaching other subjects, such as physics.

The example of the ecological model described above concerned the behaviour of an individual in the environment. A huge number of such models can be built. As we wrote earlier in the exact natural sciences, for example in physics, we have general laws, formulated in mathematical language. Most often these are different kinds of equations. Description of a specific physical situation then boils down to applying these equations and solving them using an analytical or numerical method. In ecology it is different as general problems are formulated in natural language. For one of the authors of this paper, this would be intra- and interspecific competition. Considering the huge number of environments and the huge number of different life cycles and strategies of individuals, we can realize that the number of models describing competition will be huge. Competition models can be included in the category of models describing the behaviour of individuals in the environment. Other categories of models would be models of the dynamics of populations, multi-species systems and entire ecosystems formulated as demographic models, i.e., mimicking the number dynamics, or in terms of the circulation of elements or energy flows. Another interesting topic may be ecological models of organism growth. Models simulating a complex, two- or three-dimensional geometric structure, for example, of land plants, where fractal modelling techniques are used, seem particularly attractive here.

Computer simulation methods can be used primarily to teach how nature works. However, we cannot forget about traditional biology and traditional ways of teaching it. Taxonomy is often considered by young people with interests in mathematics and computer science as not very intellectual and, above all, not very interesting science. There is no denying that this is a descriptive science whose primary function is classification. However, one of the fundamental tasks

of biology is to understand biodiversity and its dynamics (Agnarsson and Kuntner 2007). Nowadays, when species are disappearing at an accelerating rate (Pimm and Raven 2000), we have less and less time to discover and describe species.

Thus, the traditional form of teaching biology is still important, but the use of *NetLogo* in a few biology lessons could help attract to this subject young people who do not want to learn species names but would like to learn about the functioning of nature and ecological systems. Therefore, we believe that a good solution would be to combine the traditional teaching method with several classes on modelling ecological processes in the *NetLogo* program.

The proposed method of teaching, of course, also has its disadvantages. For example, modelling the behaviour of ants, we discovered that we were somewhat constrained by the *NetLogo* program, where many of the commands did not have a simple biological interpretation. Therefore, we must use what is available in *NetLogo* and try to give biological interpretations to the commands that *NetLogo* creators gave non-biological names and interpretations. This can be a kind of difficulty in the didactic process. Students creating a biological model may give up searching of appropriate commands, because it will be difficult for them to guess that they can use a certain command. For example, in the model presented in this paper, we used the command “*street*” to describe the ant’s biological directional compass and check the position of the sun, which seems hardly intuitive. This example shows that a program with many ready-made commands can create some limitations that would not appear if we did everything ourselves using other programming languages, for example C++. But this, in turn, requires much deeper knowledge of programming techniques.

Also, one of *NetLogo*’s more annoying drawbacks is its relatively small speed of work. This is especially noticeable when a lot of “*turtles*” have to appear in the model.

Then the waiting time for simulation results can be significantly longer. On the other hand, the use of *NetLogo* to conduct computer simulations of natural phenomena should remind young people what a computer is for. The primary use of a computer is to do calculations, not to communicate on social media. Not without significance is also the fact that *NetLogo* is a free program.

Author Contributions: Conceptualization, J.U.; Methodology, J.M. and J.U.; Validation, J.M. and J.U.; Writing – Original Draft Preparation, J.M.; Writing – Review & Editing, J.M. and J.U. Both authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Agnarsson, Ingi, Matjaž Kuntner. 2007. “Taxonomy in changing world: seeking solutions for a science in crisis.” *Systematic Biology* 56(3): 531-539.
- Cormen, Thomas H., Charles E. Leiserson, and Ronald L. Rivest. 1994. *Introduction to algorithms*. Cambridge MA: Massachusetts Institute of Technology.
- Grimm, Volker, and Steven F. Railsback. 2005. *Individual-based modeling and ecology*. Princeton: Princeton University Press.
- Heller, Michał. 2014. *Elementy mechaniki kwantowej dla filozofów* [Elements of quantum mechanics for philosophers]. Warszawa: Copernicus Center Press.
- Kotarski, Radosław. 2017. *Włam się do mózgu* [Break into the brain]. Warszawa: Altenberg.
- Kingsland, Sharon. E. 1995. *Modeling nature. Episodes in the history of population ecology*. Chicago: The University of Chicago Press.
- Mostowski, Jan, Włodzimierz Natorf, and Nina Tomaszewska. 2002. *Fizyka i astronomia Część 1. Poradnik dla nauczyciela i program nauczania w liceum ogólnokształcącym, liceum profilowanym i technikum* [Physics and astronomy Part 1. Teacher’s guide and curriculum in high school and technical school]. Warszawa: Wydawnictwa Szkolne i Pedagogiczne.

- Murphy, Kilian J., Simone Ciuti, and Adam Kane. 2020. "An Introduction to agent-based models as an accessible surrogate to field-based research and teaching." *Ecology and Evolution* 10: 12486-12498. <https://doi.org/10.1002/ece3.6848>.
- Ovadia, Ofer, Heinrich zu Dohna, Ginger Boothy, and Oswald J. Schmitz. 2007. "Consequences of body size variation among herbivores on the strength of plant-herbivore interactions in a seasonal environment." *Ecological Modelling* 206: 119-130.
- Pimm, Stuart L., and Peter Raven. 2000. "Biodiversity: extinction by numbers." *Nature* 403: 843-845. <https://doi.org/10.1038/35002708>.
- Railsback, Steven F. and Volker Grimm. 2012. *Agent-based and individual-based modeling. A practical introduction*. Princeton: Princeton University Press.
- Tegmark, Max. 2014. *Our mathematical universe. My quest for the ultimate nature of reality*. New York: Alfred A. Knopf.
- Uchmański, Janusz. 1980. "Czy matematyka jest językiem ekologii?" [Is math the language of ecology?]. *Wiadomości Ekologiczne* 26: 221-231.
- Uchmański, Janusz. 2015. "Matematyczność biologii." [Mathematics of biology]. *Filozofia i Nauka* 3: 345-352.
- Uchmański, Janusz. 2020. "On algorithmic essence of biology." *Computer Research and Modeling* 12(3): 641-652. <https://doi.org/10.20537/2076-7633-2020-12-3-641-652>.
- Uchmański, Janusz. 2022. "Density or variability: is it time for a paradigm shift in ecology?" *Ecological Questions* 33(4): 7-20. <https://doi.org/10.12775/EQ.2022.029>.
- Uchmański, Janusz, and Julia Zielińska. 2010. "NetLogo – wygodne narzędzie do modelowania procesów ekologicznych." [NetLogo – a convenient tool for modeling ecological processes]. *Wiadomości Ekologiczne* 56(3): 95-110.
- Wilensky, Uri. 1999. *NetLogo*. <http://ccl.northwestern.edu/netlogo/>. Evanston, IL. Center for Connected Learning and Computer-Based Modelling. Northwestern University.