

VU Modeling and Simulation

Cellular Automaton vs. Difference Equation Case Study: Predator Prey System

Autoren

Christoph Hämmerle

Philipp Ganiu

Jennifer Lumetzberger

Contents

1 Introduction

A forester wants to get information about the maximum number and the dynamics of the population of deer in its forest. In order to observe the correlation between the number of red-foxes and deer, a predator-prey approach is modelled. Different modelling concepts are compared: a cellular automata model and a formulation given by a damped version of the classic ODE model by Lotka and Volterra.

Damped Differential Equation Model

First of all a difference equation model is derived (The derivation is very similar to the derivation of the classic differential-equation model). The following is observed (e.g. by external datasets or by opinion of the forester):

- During each month in the absence of a natural predator the population of deer increases by a certain rate $\alpha > 0$ which includes natural birth and death rates. As the forest has a certain resource capacity M the growth is limited leading to $\alpha(M - x)$ (respectively $\alpha(M - x - y)$ if there is a fox-species too)
- During each month in the absence of any deer the population of foxes decreases by a certain rate $\gamma > 0$ as the population is starving.
- In case of cohabitation of the two species we could observe that the number of deer each month is reduced by $xy\beta$ with a positive β animals.
- Benefiting from fresh meat, the decreasing rate γ is balanced by a positive rate δx . In case $\delta x - \gamma > 0$ there is an increase of the number of foxes. Otherwise the number decreases.

Putting all those observations together leads to the following dynamics:

$$x_0(t) = x(t)\alpha \times (M - x(t) - y(t)) - x(t)y(t)\beta \quad (1)$$

$$y'_0(t) = y(t)(\delta x(t)\gamma) \quad (2)$$

Cellular Automata Model

Given the same information a cellular-automata is derived.

- We assume our forest to be approximately rectangular and apply a rectangular grid.
- At the beginning of the simulation each cell of the grid is either placed a prey, a predator or left empty. We furthermore sloppily say that a cell is either "prey", "predator" or "empty".
- The whole matrix is updated simultaneously each month by the following rules:
 - In case cell i,j is empty, we chose one neighboured cell at random and check if the cell is prey. In this case, the neighboured cell reproduces and cell i,j becomes prey as well with probability p_1 .
 - In case cell i,j is prey, we chose one neighboured cell at random and check if the cell is predator. In this case, the neighboured cell eats up the prey and reproduces. Therefore cell i,j becomes predator as well with probability p_2 .
 - In case cell i,j is predator, the predator unfortunately dies with a certain probability. Therefore cell i,j becomes empty with probability p_3 .

By a neighboured cell we understand any cell which is not father away as k cells from the original cell, whereas k is a natural number.

2 Task description

Task 1

Implement the damped differential equation model and perform some trial runs. Try several self chosen parameters $0 < \alpha, \beta, \gamma, \delta$ for the simulation and fix $x_0 = 4000$ and $y_0 = 1000$ as initial populations. Notice that $M \gg x_0 + y_0$ and fix a self chosen natural number for it.

Task 2

Implement the cellular automaton (CA) model and perform some trial runs. Try some self chosen parameters $0 < p_1, p_2, p_3 < 1$ and fix neighbourhood parameter $k = 1$ at the first place. Use the same initial conditions for the CA as for the difference model and initially mark 4000 cells as prey and 1000 cells as predator - note that you need to use a rectangle with more than $4000 + 1000$ cells (furthermore denoted as N) in order to perform simulations.

Task 3

Try to find out how to present the results of the CA in animated form and how you can aggregate the results of the CA in order to achieve comparable curves to the differential

equation.

Task 4

Fix all other parameters and experiment with different rectangles with same number of cells. Experiment whether the shape of the rectangle influences the results.

Task 5

Try to find parameter-settings $(\alpha, \beta, \gamma, \delta, M)$ and (p_1, p_2, p_3, N, k) so that the corresponding resulting curves of both models look alike.

Task 6

Summarize your project work as a protocol.

3 Methods

The predator-prey approach is modelled with Matlab.

4 Execution

4.1 Task 1

For the first task, the damped differential equation model is implemented in Matlab.

As initial values for the deer (x) and fox (y) population, we take $x_0 = 4000$ and $y_0 = 1000$, as given in the task. For the resource capacity of the wood M, we consider $M \gg x_0 + y_0$ and fix it with $M = 10000$.

The parameters α , β , γ and δ are also fixed with:

$$\alpha = 0.001$$

$$\beta = 0.003$$

$$\gamma = 2$$

$$\delta = 0.001$$

First of all, a matrix is created, which stores the month and the corresponding amount of deer and foxes, for every month from 0 (starting point) to the desired maximum time point:

month	0	1	2	...	m
deer	4000	?	?	?	?
foxes	1000	?	?	?	?

Table 1: Matrix scheme with information on amount of deer/foxes for each month

Via the given difference equations (??) and (??) the population of each time point can be calculated. After each iteration, the data is stored into the matrix.

4.2 Task 2

For implementing the cellular automata (CA) model, some self chosen parameters $0 < p_1, p_2, p_3 < 1$ are used. The neighbourhood parameter k is fixed with $k = 1$ at the first place. As with the difference model, the initial values are the same: $x_0 = 4000$ and $y_0 = 1000$.

So, the initial parameter setting is as following:

$$rows = 70$$

$$col = 200$$

$$p1 = 0.01$$

$$p2 = 0.8$$

$$p3 = 0.01$$

Now the first step is to set up a matrix with the 4000 deers and 1000 foxes randomly distributed in our rectangle (forest). The rectangle is built up by the number of rows and columns. Therefore in this case we have 14000 (70x200, universally valid: N) cells, that's why we create a row-vector *initvalues* with N entries. With the function **randperm** we pick 5000 ($x_0 + y_0$) cells with a random permutation between 0 and N , so each value appears only once. These random numbers we use as indices of *initvalues* and assign the first 4000 (x_0) cells to 1 (which stands for deer). The next 1000 (y_0) random numbers we assign to 2 (which means foxes). Since we initialized *initvalues* with zeros, the step of creating the initial population is done.

In order to create the rectangle, we take our *initvalues* vector and fill a new matrix *population* with its values, by using the **reshape** function.

For creating the states for each timestep, a for-loop is used, regarding each cell and applying the rules of the CA described at the beginning.

After each iteration we sum up all our foxes, deer and empty spaces, to have results which can be compared later with the values of the differential equation model.

4.3 Task 3

For achieving comparable results with the CA as with the differential equation, after each loop iteration, all cells belonging to the fox/deer/no population are summed up and stored in a matrix.

For animating the CA, a plot is created with the x-axis representing the length of our rectangle and the y-axis illustrating the width of the rectangle. Using the Matlab function **spy**, each cell of the rectangle/matrix can be marked in a different color. When updating the plot after each loop iteration (each time step), we have created a simple animation presenting the population development of our species.

4.4 Task 4

In Task 4 all parameters are fixed, except of the shape of the rectangle (number of cells stays the same). The results are shown in the results section.

4.5 Task 5

In Task 5 the parameters $\alpha, \beta, \gamma, \delta, M$ as well as p_1, p_2, p_3, N, k shall be changed so that the corresponding resulting curves of both models look alike. The results are shown in the results section.

4.6 Task 6

This task is obviously done.

5 Results

5.1 Task 1

When establishing the difference equation model with the parameters fixed in the execution section of Task1, we gain the following result:

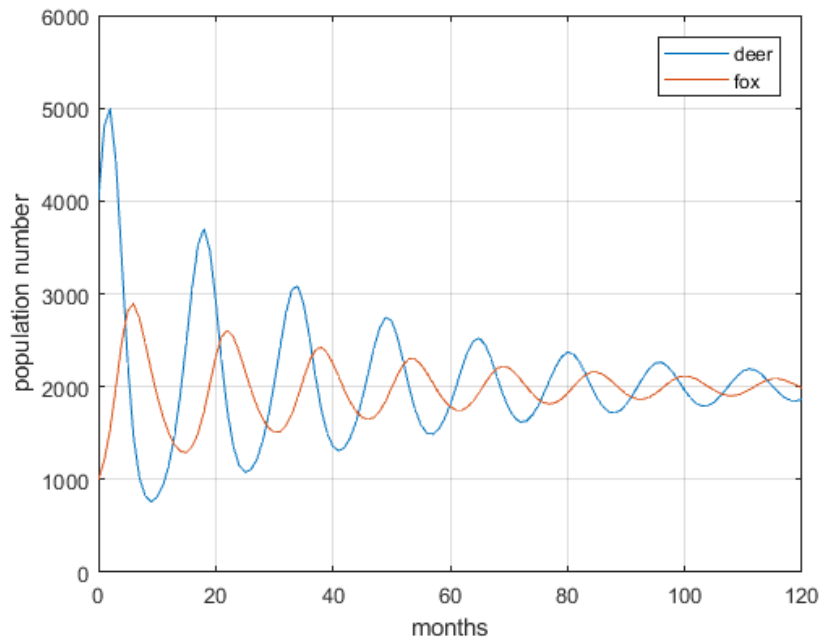


Figure 1: Population development of foxes and deer over 10 years

At the beginning, the deer population is 4 times the fox population. Due to the high amount of food for the foxes, the fox population rises, which leads to a fall in the deer population. After several years, an equilibrium is found, when each species comprises about 2000 individuals. In table ?? the population numbers of certain time points can be seen.

month	0	1	5	24	60	80	100	120
deer	4000	4800	2258	1154	1806	2370	1959	1864
foxes	1000	1200	2817	1375	1783	1927	2116	1984

Table 2: Difference Equation Model: Population development at certain time points

5.2 Task 2 and 3

In Task 2 and 3, the cellular automata model was created. We can compare the rectangle in the initial state with the results after 10 years. We also look at two other time steps, namely at $m_1 = 5$ months and $m_2 = 24$ months.

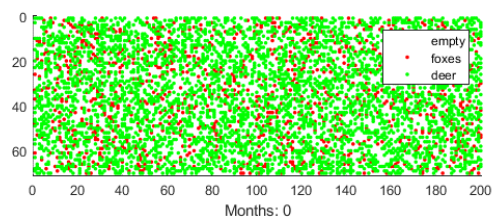


Figure 2: CA at initialization time

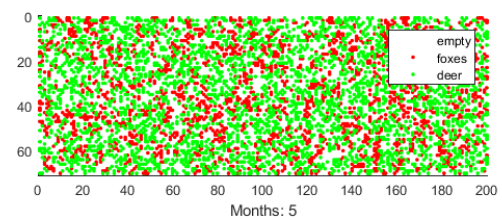


Figure 3: CA after 5 months

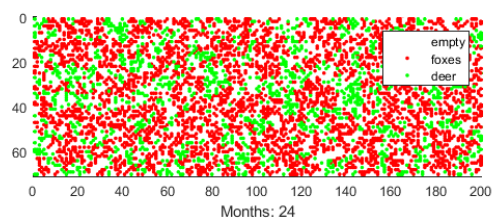


Figure 4: CA after 2 years

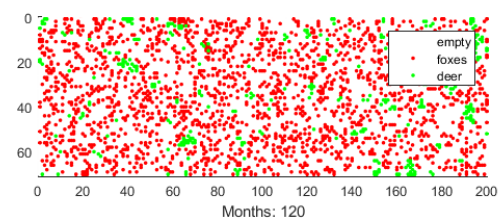


Figure 5: CA after 10 years

As can be seen in Figure ?? - ??, the population of deer decreases drastically with time, whereas the population of foxes increases until a certain time and then decreases again. In table?? some values of the population development can be seen.

month	0	1	4	24	60	80	100	120
deer	4000	3821	3082	1190	540	440	405	403
foxes	1000	1191	1952	3529	3146	2707	2275	1882

Table 3: Cellular Automata results at certain time points

The whole course of events can be seen in Figure ??.

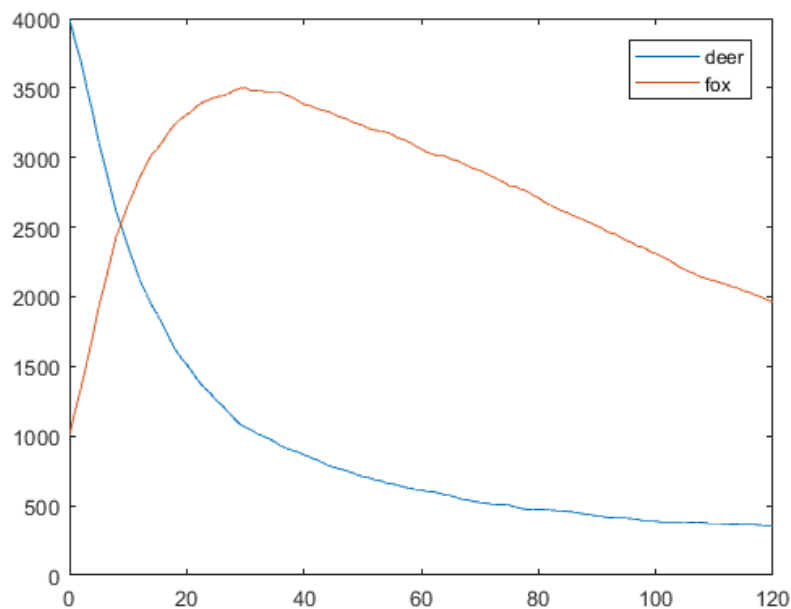


Figure 6: Population behaviour over 10 years with CA model

5.3 Task 4

The number of cells is fixed with $N = 14000$. We are only changing the shape of the rectangle and observe the results. When we do not change the shape of the rectangle dramatically, the behaviour of the model also does not change a lot. Due to the randomly created matrix and the probability influences, each run of the code is different and cannot be repeated. When we change the shape of the rectangle in a way in which we set one parameter (row,column) very small (to 1 or 2), the results change a lot (see Figures ?? and ??).

Since the algorithm of the implemented cellular automata does not consider all neighbours, but only 1 randomly chosen neighbour, there should not be a large difference by changing the shape of the rectangle. If for example, we have a very narrow and long trail of

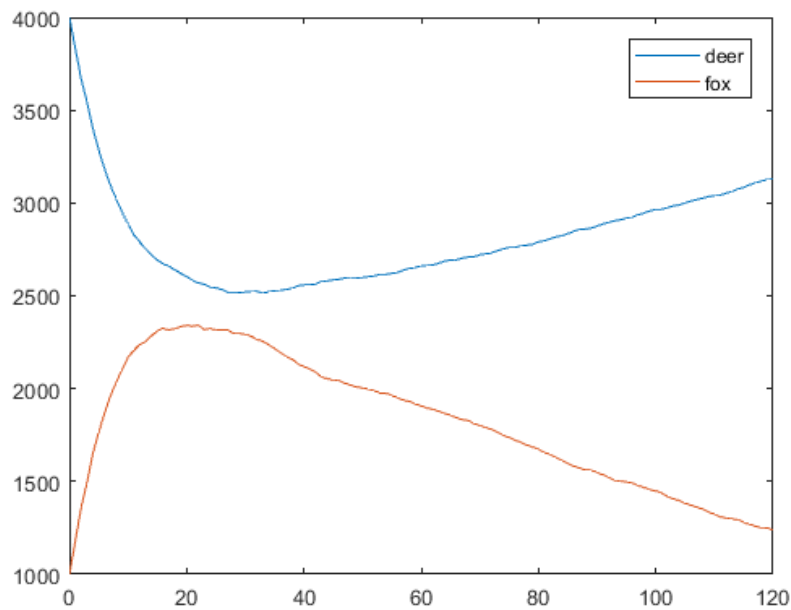


Figure 7: CA model with only 2 rows

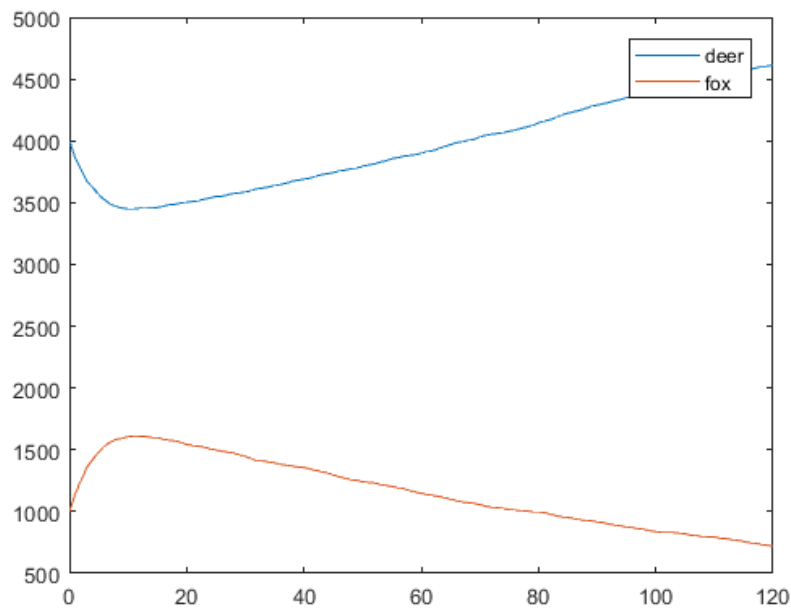


Figure 8: CA model with only 1 row

wood (only 2 neighbours possible, or 1 at each end of the trail, when we use $k = 1$), the probability for a deer to meet a fox is smaller than with a large-area wood where 4 neighbours are possible). Therefore, the large difference does not make sense to me.

5.4 Task 5

Now the parameters $\alpha, \beta, \gamma, \delta, M$ as well as p_1, p_2, p_3, N, k are changed so that the corresponding resulting curves of both models look alike.

The parameters were changed to:

$$p1 = 0.01$$

$$p2 = 0.8$$

$$p3 = 0.01$$

$$N =$$

$$k =$$

$$\alpha =$$

$$\beta =$$

$$\gamma =$$

$$\delta =$$

The resulting plot can be seen in Figure.

6 Discussion