MATLAB-based Modelling and Simulation of Forest Development: Growth – Harvesting – Draught – Bark Beetle – Biomass Plant Usage

Constantin Königswieser*, Anicia Jahn

Inst. of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10,
1040 Vienna Austria; *Constantin.Königswieser@student.tuwien.ac.at*

SNE 30(2), 2020, 77 - 84, DOI: 10.11128/sne.30.stn.10517
Received: April 2, 2020; Revised May 5, 2020;
Accepted: May 10, 2020
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The present study addresses modelling and effective MATLAB simulation of forest of growth of a forest. The modeling of the forest growth was conducted using certain assumptions on growth. Besides the simulation the aim of the study was to simulate as many individual trees as possible with the limited processing power of a common notebook. The developed simulation algorithm calculates many individual objects – trees - with the main goal to receive a fast and efficient program. This was achieved via the ideas from cellular automata.

Introduction

Global warming has led to many environmental problems, such as increased periods of drought and less precipitation. The climate change will affect various types of animals and plants and change the growth dynamics of forest trees. This will change Austrian landscapes permanently. Especially spruce, which is very common in Austria's forests, will cease in higher temperatures.

According to this, certain steps have to be made to ensure the health of our forests [1]:

- developing tools for the assessment of ecosystem sustainability
- studying the quantitative and qualitative patterns of ecosystem dynamics under the impact of global warming
- establishing new paradigms of sustainable forest management

This prospect requires effective predictions of forest growth and the dynamics of its basic characteristics as well as the implementation of disturbances, which can be considered as the key drivers of forest ecosystem dynamics because of their significant influence on its structure and functioning. To ascertain an effective prediction, a simulation model, namely an imitation of a real life system, can be developed [1].

This study addresses the application of a forest simulation using MATLAB. Besides the actual simulation of a spruce forest, the aim of this study was to simulate as many individual trees as possible with the limited processing power of a common notebook. The boundaries of MATLAB were tested by developing an algorithm calculating many individual objects with the main goal to develop a fast and efficient program. This was achieved via ideas from cellular automata, which will be discussed in Chapter 2.1. We used MATLAB versions R2016a and R2019b to build and visualize our model.

As mentioned before we model a naturally growing forest with as many trees as possible to test the boundaries of MATLAB. The goal is to find out if such a program would be feasible on a common notebook. Different forest sizes are tested on a notebook with the following properties: Intel Core i7-4710HQ, 2,5 GHz, 16GB RAM.

Furthermore, to assess the influence of different scenarios, four general simulation scenarios were designed, which are the following:

- Contamination
- Harvest
- Drought
- Bark beetles
- Biomass plant usage

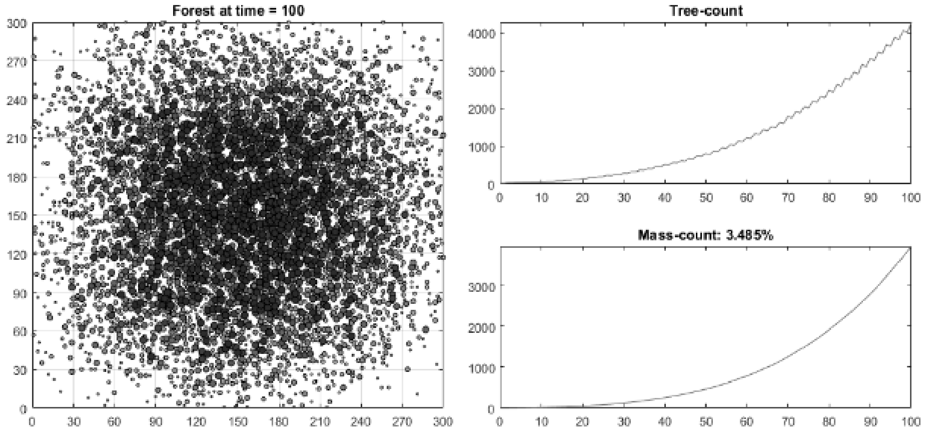

Figure 1: Standard simulation diagrams, which are updated each time step. Left: 2D-presentation of the forest. Right: number of trees and wood mass of the forest.

1 Modelling

1.1 Forest modelling

The first thing that comes to mind, when modelling a forest with individual trees, would be object orientated programming, which MATLAB offers but is not suited for in large scales as the program is an analysis software. So instead of creating an object structure we use MATLAB's matrix-based data structure to store the different properties of every single tree in corresponding cells.

This model was inspired by the cellular automata, a discrete, abstract computational system [2]. In these automata an occupied space is represented by one, otherwise it is zero. Similar to cellular automata our model is discrete; however, as a cellular automaton is deterministic and we introduce probabilities, our model cannot be classified as such.

We generate a 2D discrete space (see Figure 1) where each cells future state is dependent on its current state and neighbors, namely depending on position.

One of the major questions arising, when developing a model, is the level of detail and processing resolution. Building any simulation model, a number of assumptions and simplifications have to be made.

We assume a flat and confined area with unchanging weather conditions. For this model, two main assumptions are made. The first is that the quantity of neighbors and the tree's age is relevant for spreading. The second assumption is that trees can randomly die depending on age. In this random number, influences such as light, water, nutrition and air are included.

For this project, a variable square forest area is populated with 20 trees, age 10, as initial condition. The planting locations are randomly chosen within a radius of 20 m around the central point of the area. Different simulation scenarios are conducted, taking into account both the regime of forest management, namely harvest, as well as natural disturbances, such as droughts. Certain parameters, such as spreading, death rate etc. of a spruce tree, are necessary for model initialization.

Structure of the forest. For this kind of object-orientated programing it would be obvious to populate a forest object with tree objects in which all properties, such as age, mass, number of neighbors etc., are stored. As MATLAB is not suited for this kind of programming, we chose a different approach.

We generate only one object (class as it is called in MATLAB), namely the forest, in which all properties are stored in separate matrices. Inspired by the aforementioned cellular automata, the size of the forest determines the size of the matrix so that each entry corresponds to a location in the area. Furthermore we assume that one entry matches one square meter. For example, a forest of size 100 m × 100 m generates a 100 × 100 matrix.

Since the matrices provide us with a grid, there is no need to compare each tree with every other tree. Instead we only need to check the cells within a given radius for neighboring tree, which reduces the number of operations needed dramatically.

Methods	Descriptions
Forest()	Construction-method of the forest. Initializes all matrices for storing properties.
findTrees()	Searches for all trees in the given area and stores their x- and y-location in a 2xn-matrix, where n is the number of found trees (namely *foundTrees*).
findNeighbours()	Loops through all found trees and counts all neighbors in a given radius.
plantTree()	Plants a tree, with a given age, by changing the corresponding entry of the tree-matrix to 1.
markTrees()	Loops through all found trees and applies the rules for dying for each tree and marks those which will be removed in the next time step.
removeTree()	Removes all marked trees and resets the properties of the given location.
spread()	Loops through all found trees, generates a random location for the sapling and checks afterwards, depending on neighbor count, if this sapling can be planted.
aging()	Increases the age of the tree by one as well as the mass. Additionally it increases a color counter every 5 years by one; up to a maximum of 10.
die()	Finds all marked trees and calls the *removeTree()*-function.
harvest()	Loops over the current harvest area and marks all trees with in that area over a given age and records the harvested mass.
infest()	Finds all trees within a given (small) area and choses randomly a tree and increases the beetle count by 20 on the chosen trees (100 times).
beetleSpread()	Finds every infested tree and loops over them. It multiplies the number of beetles by a given factor and spreads the number of beetles evenly on neighboring trees.
boundaries()	A helpful function which generates the boundaries of the search area with a given tree-location and radius, including edge cases.
draw()	Generates a scatter-plot of all trees with full circles of different sizes and shades of green depending on age (see Figure 1).
drawMarked()	Generates a scatter-plot of all marked trees with red crosses.

Table 1: Summary of the forest's methods.

To update the forest each year, a number of different functions (methods as they are called in MATLAB classes) are used, given in Table 1.

To retrieve the indices, the find function is heavily used. The find function returns the indices and values of nonzero elements and thus allows us to loop over occupied spaces only.

Similar to cellular automata we appoint a few simple rules:

- Spreading: depending on the number of neighbors the tree plants a sapling in a randomly generated unoccupied space in a certain radius; however, not too close to other trees
- Aging: every year the tree grows older and accumulates mass depending on age
- Dying: each tree has a certain probability to die, depending on its age

Spreading: Every other year, as it is the case in nature [3], the trees spread and plant new saplings. This is achieved by appointing different probabilities according to different numbers of neighbors in a certain distance; for this study a radius of 10 m is chosen.

- zero neighbors: no spreading: 0%
- less than or equal to three neighbors: 50% chance to spread
- between three and ten neighbors: 100% chance to spread
- greater than or equal to ten neighbors: 50% chance to spread
- more than fifty neighbors: no spreading: 0%

These probabilities are conducted according to the following considerations: if the tree has no neighbors, there is no chance to pollinate the tree. If it has a certain amount of neighbors there are enough pollen to fertilize the seeds.

However, if the tree has too many neighbors there is not enough space for the seedlings to grow and no spreading occurs.

Aging: every year, the tree grows older and accumulates more mass depending on age. Till the age of 70, we assumed a linear growth rate of 28 kg/year. From 70 to 140 the tree has a growth rate of 14 kg/year until it reaches its final mass of approximately 3.000 kg. When calculating the mass, we assume that the harvested wood has already sufficiently dried with a water content of 20%. [4]

Dying: Each tree has a certain probability to die, according to its age.

- Less than or equal to three years: 5% chance to die
- Greater than three years: 1% chance to die
- Greater than 150 years: 20% chance to die

With the probability to die each year, general stress-factors such as diseases, damage by wildlife, or storm losses are considered. The effects of drought and bark beetles are dealt with separately. Younger trees and very old trees are naturally more susceptible to these general damages.

As a forest is a very complex ecosystem, data on these specific parameters and probabilities is missing. As a result, the authors decided to conduct the procedure of model initialization by using the following parameters:

- trees grow on an area of one hectare [5]
- The forest has a growth rate of 2,5 % cubic meter each year [6]

To arrive at these targets, we adjusted the spreading and neighbor radii by means of computational experimentation.

1.2 Contamination modelling

The first issue is the implementation of a contaminated area in the forest, where trees can only survive for a short amount of time as there are toxic chemicals in the ground, which prevent the trees to grow further.

This is achieved by implementing a given area, which is marked as “contaminated” where trees, which grow on this ground, will cease after two years. This implementation could be used for rivers or roads as well.

We preset a matrix, namely contaminated, with the elements one and zero for contaminated and non-contaminated areas respectively. In the function markTrees() every tree is checked if the location is considered contaminated and if so, it will die if it is older than three years.

1.3 Harvest modelling

The second issue is harvesting specific areas, which is conducted by dividing the whole area in ten equal strips. At a given time, all trees over a given age in one strip (0,9 ha) are harvested. In the function harvest(), each entry of the strip is checked for trees. Then, each tree reaching the requested age is marked and subsequently removed. Furthermore, the mass count is recorded. Each iteration (namely one year), the harvesting strip shifts to the next starting point, which is conducted each year till it reaches the beginning and starts once again.

1.4 Drought modelling

Less precipitation causes increasing stress for the trees which results in a higher mortality rate and reduced mass growth as well as fewer offspring [7]. We assume an increased mortality rate of 20% and a growth reduction of 50% and a reduced chance of generating offspring by 50%.

To implement this natural phenomenon, these rates are changed in the functions markTrees(), spread() and aging() depending on a drought flag, which is set for a given time period.

1.5 Bark beetle modelling

The last issue is the bark beetle infestation which will be a great threat in the future, since global warming causes warmer winters and therefore increases their chances to survive.

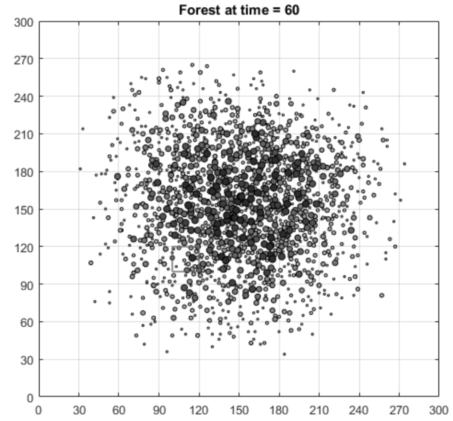

Figure 2: Red box: Initial infestation area.

The biggest problem of these beetles is the explosive growth rate and the fact that the trees succumb to these immense numbers [8].

In our model the beetles spread three times a year and infest each tree without exception (such as age etc.). The number of beetles on each infested tree multiplies tenfold and the same amount is equally distributed to neighboring trees within the radius of 10 m. Furthermore, each tree with 200 beetles dies and the beetles with it.

To initiate the infestation the function infest() injects a given number of beetles into a certain area of the forest once (see Figure 2). To spread the infestation the function beetleSpread() is called three times per time step. The markTrees()-function checks the number of beetles on every tree and marks them, if the given threshold is met.

2 Simulation Results

At first to verify the model and to meet the set targets a 9 ha forest (300x300) is simulated five times and the mean values are calculated. The results are shown in figures 3 and 4. The figures show that the target tree population of 1000 trees per ha is met in its equilibrium state. This state is reached after 250 years.

The forest of this model follows a logistical growth and thus has a changing growth rate. For the second target, we estimate the growth rate by taking the mean value between the years 50 and 200, which results in a mean growth rate of 2,67% m³ per year, which corresponds well with the set target of 2,5% m³ per year.

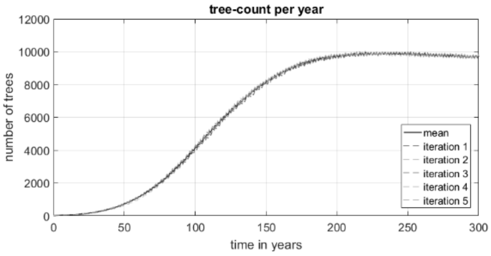

Figure 3: 5 iterations and mean value (blue) of the forest's tree count each year over time span of 300 years.

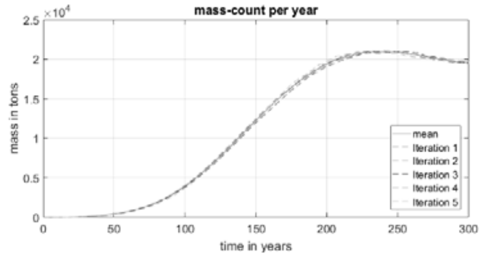

Figure 4: 5 iterations and mean value (blue) of the forest's wood mass each year over time span of 300 years.

As pictured the deviation from the mean value is small enough to justify the low number of iterations. Thus every following scenario (contamination, harvest, drought and bark beetle) is conducted in the same way, by iterating five times and taking the mean. Additionally other scenarios are deactivated, to examine each scenario by itself.

2.1 Contamination

As Figure 5 shows, the contaminated area in the forest remains blank, except for a few trees of age 1-3 years.

The contaminated area acts as a barrier and the forest has to grow around it. Therefore the forest spreads more slowly behind the contaminated area.

As expected the growth is delayed and terminates in a smaller total amount of trees and mass compared to the forest without a contaminated area (see Figures 6 and 7).

2.2 Harvesting in strips

In this simulation harvesting starts after 100 years and lasts for 150 years and only trees older than 90 years are felled, since the ideal age for harvesting of spruce is between 80 and 120 years. [9]

The mean value of the harvested wood over this time span is 116,1 tons per year. The first 50 years of harvesting have considerably lower yields, as in the beginning the forest is still growing in size and only few trees are over 90 years old.

However, after 95 years of harvesting the forest is already at its fullest with 9.000 trees and therefore a mean harvest of 188,32 tons per year is guaranteed.

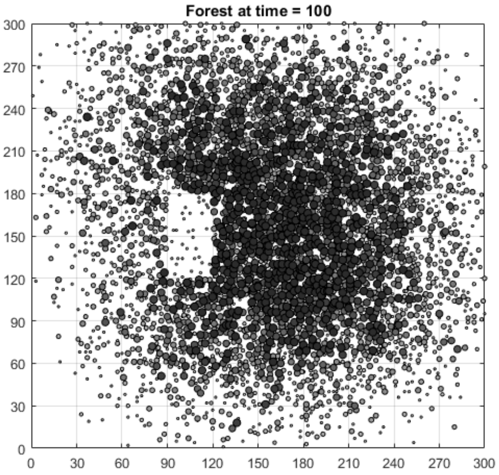

Figure 5: Forest at the age of 100 years with a contaminated area between (90,120) and (120,180).

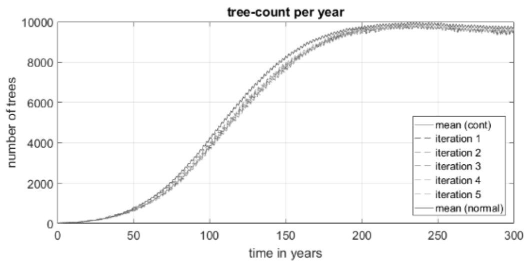

Figure 6: Tree count of the whole forest with contaminated area.

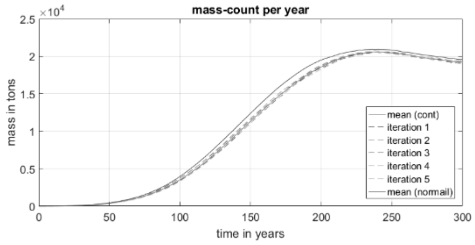

Figure 7: Mass of whole forest with contaminated area.

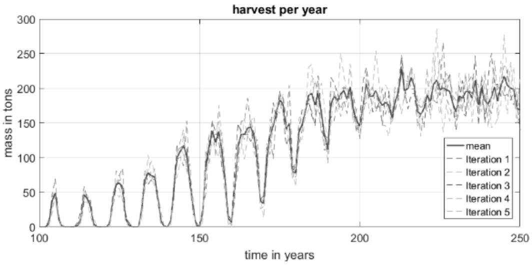

Figure 8: Five iterations and the mean value (blue) of the harvested wood over a time span of 150 years.

As shown in Figure 9 and Figure 10 the forest can maintain a steady growth rate and can sustain itself without problems.

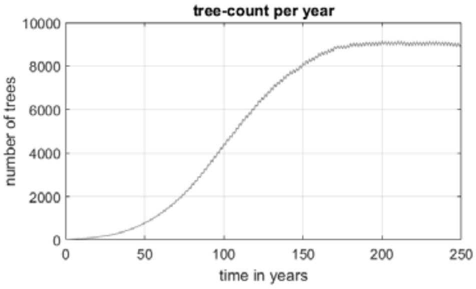

Figure 9: Tree count of the whole forest while harvesting (mean value over 5 iterations).

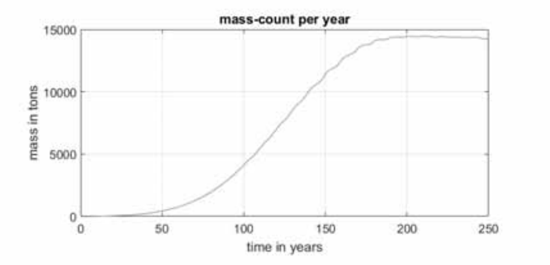

Figure 10: Mass count of the whole forest while harvesting (mean value over 5 iterations).

2.3 Household energy supply by wood

'How many households can be supplied with electric energy using the felled wood?'

This question arises, how much energy can be gained using 116,1 tons of felled wood. An average household in Austria consumes 4.415 kWh electric energy each year [10]. Assuming, that the wood is already sufficiently dry with a water content of 20%, the wood has a fuel value of 4 kWh per kg [11]. Consequently, transferring these 116,1 tons of dry wood to a biomass power plant, it would amount to 464.400 kWh which could provide approximately 105 households with electric energy in Austria each year.

2.4 Drought

The drought is implemented twice, after 50 and 250 years, each period lasts for ten years.

The model shows that a drought at an early stage has no significant impact, the growth rate merely stagnates. The drought at the latter period, with a full grown forest of 9.000 trees, causes an apparent reduction of growth rate, as well as a higher mortality. The forest seems to recover after a certain amount of time, however it takes more than 100 years to regain the same amount of trees as well as wood mass.

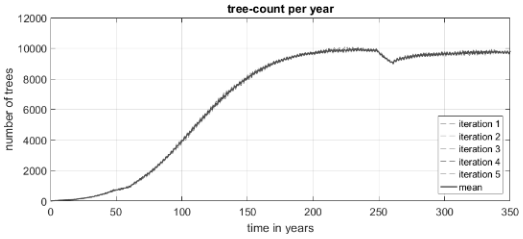

Figure 11: Mass count per year of drought periods at 50 and 250 years over a time span of 350 years.

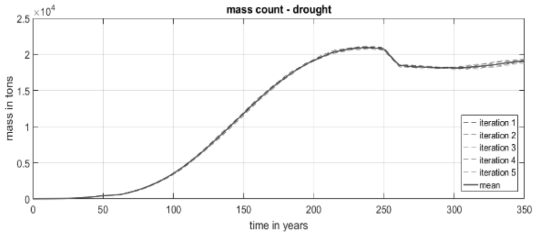

Figure 12: Mass count per year of drought periods at 50 and 250 years over a time span of 350 years.

2.5 Bark beetle

To investigate under what conditions a bark beetle outbreak will occur, we conduct nine scenarios in which we differ the year of infestation. 2.000 bark beetles are inserted in a specified (de-central) area (see Figure 2) starting in the year 20, each time incrementing by 5 up to 60 years.

As Figures Figure 13 and Figure 14 show, the first bark beetle outbreak happens in the year 40. Before that, the bark beetles couldn’t spread, as the number of trees in the specified area is too small (see Figure 15). The outbreak lasts approximately five years and eradicates nearly all of the tree population, as a consequence the beetles go extinct since there are too few trees left and the distance between these trees is too high.

Even though the bark beetles cause the forest population to decreases immensely and delay its growth by ten years, it is always able to recover within approximately 150 years. The simulation-data shows an overshoot in mass as well as tree count 200 years after the outbreak, which may be explained by the following reasoning:

- At the time of the outbreak the trees occupy a far spread area
- After the outbreak only saplings remain

These factors cause a very homogenous forest of widespread, same aged trees. Since in this model the mortality rate is increased significantly (by 19%) at the age of 150 an over proportional amount of trees die and therefore mass and tree count drop. This phenomenon can be especially well observed in the mass count.

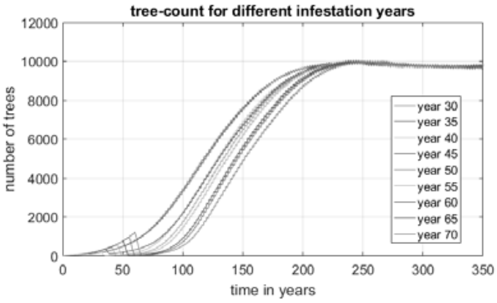

Figure 13: Tree count per year during the bark beetle infestation over a time span of 350 years.

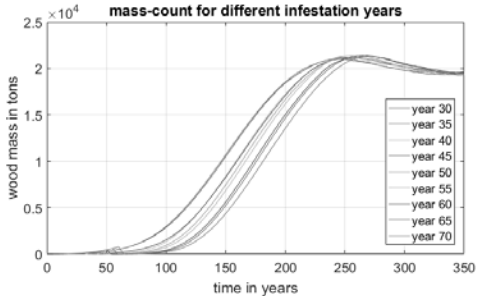

Figure 14: Tree count per year during the bark beetle infestation over a time span of 250 years.

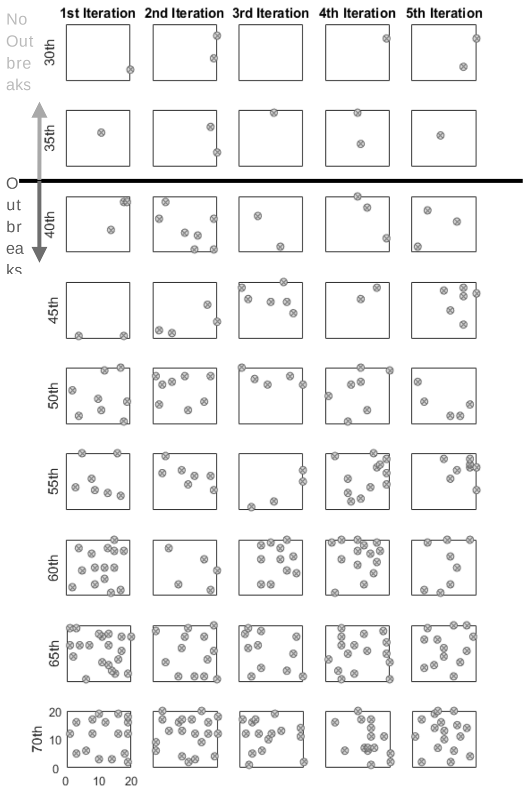

Figure 15: Close-Up of the infestation area (as seen in Figure 1) for every simulation. The rows depict the different years in which the forest is infested.

We assume that each tree killed by bark beetles is removed and as a consequence the beetles from this tree cannot spread further. By doing so, the bark beetles eventually go extinct. Transferring this circumstance to forestry, infested and dead trees have to be removed in order to defeat the bark beetle.

As Figure 15 shows the outbreak only occurs if the trees have enough neighbors, namely more than two. This would suggest that a forest with less planting density is less prone to bark beetles.

2.6 Computational power

To identify the computational power on a notebook, different forest sizes were simulated, starting from 1 ha to 64 ha. The results are shown in Figure 16, and it can be clearly seen that the simulation time grows linearly to the side length of the forest area.

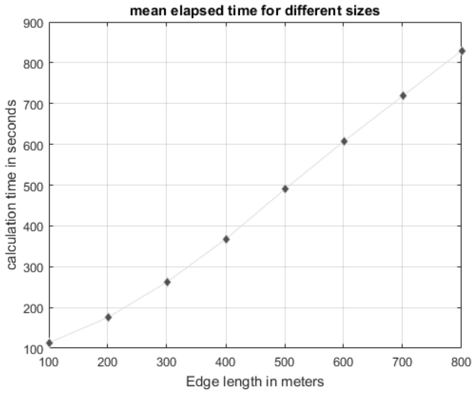

Figure 16: Mean time elapsed for different forest sizes.

3 Conclusion

This study shows, that complex systems with a considerable amount of data and computational power are able to run on a simple notebook using MATLAB, if the program is implemented with certain computational effort saving techniques, such as avoiding nested loops – in this regard the find()-function provided by MATLAB was profoundly useful.

According to this model it is possible to harvest wood without constraining the forest's natural growth, if certain rules are followed. Here the harvest took place in strips and only trees older than 90 years are felled. Other harvesting techniques may be examined in future studies. Finally it should be mentioned that a yearly reliable yield can only be achieved, if the forest has reached nearly its maximum size.

Likewise, the forest size respectively the planting density plays a crucial role in the bark beetle outbreak. According to this model, the outbreak can only occur, if there are enough neighboring trees around the infestation center. Though if the density is high enough, the forest will be nearly annihilated within less than 5 years.

However the forest always recovers and returns to its pervious size in less than 50 years. It has to be considered that the forest would only survive a bark beetle invasion, if the infested and dead trees are removed and thus eradicate the bark beetles.

The ideas of the cellular automata proved to be very viable to depict a very complex system in a comparably simple way. Over all this model manages to generally resemble a naturally growing forest with two very simple rules.

To receive more realistic results, nonlinear interactions between the individual trees and their abiotic environment, such as light, air and soil, have to be considered, which could be implemented by adding more complex rules.

Acknowledgement

This study is a student's work for preparation of a master thesis (TU Wien, Institute of Analysis and Scientific Computing; supervisor: Prof. Dr. Felix Breitenecker *Felix.Breitenecker@tuwien.ac.at*). This is the first *SNE Student Note*, a new type of SNE notes introduced in 2020.

References

[1] Shanin VN, Komarov AS, Bykhovets SS. *Simulation modelling for sustainable forest management: a case study*. Procedia Environmental Sciences; 2012, Vol. 13.

[2] *Stanford Encyclopedia of Philosophy*. [Online]. Available: https://plato.stanford.edu/entries/cellular-automata/. [Accessed 8 March 2020].

[3] *tierfreund*. [Online]. Available: https://www.tierfreund.de/waldkiefer/. [Accessed 8 March 2020].

[4] *Handwerkerratgeber*. [Online]. Available: https://handwerkerratgeber.info/spezifisches-gewicht-von-holz-bestimmen/#. [Accessed 8 March 2020].

[5] *Pro Holz* [Online]. Available: https://www.proholz.at/zuschnitt/51/der-oesterreichische-wald. [Accessed 8 March 2020].

[6] *Holz ist genial*. [Online]. Available: https://www.holzistgenial.at/blog/10-fakten-zum-wald-in-oesterreich/. [Accessed 8 March 2020].

[7] *Waldwissen*. [Online]. Available: https://www.waldwissen.net/waldwirtschaft/schaden/trockenheit/bfw_trockenstress/index_DE. [Accessed 8 March 2020].

[8] *Waldverband*. [Online]. Available: https://www.waldverband.at/wp-content/uploads/2016/06/Borkenk%C3%A4fer-2016-web-1.pdf. [Accessed 12 March 2020].

[9] *wald-prinz*. [Online]. Available: http://www.wald-prinz.de/umtriebszeit-wie-lange-benotigt-ein-baum-bis-zur-hiebsreife/3697. [Accessed 14 March 2020].

[10] *Österreichs Energie*. [Online]. Available: https://oesterreichsenergie.at/daten-fakten-zum-stromverbrauch.html. [Accessed 8 March 2020].

[11] *Kesselheld*. [Online]. Available: https://www.kesselheld.de/heizwert-hackschnitzel/. [Accessed 8 March 2020].