Studienleistung für Komplexe Systeme

Axelrod's Dissemination of Culture and it's implementation into a python application

Lenny Samuel Müller[†]

[†] *Universität Bremen, Physik VF Master, Matrikelnr: 6033508*

Abstract

The following report is part of the study achievement for the course *Komplexe Systeme* (01-PHY-MA-TheoPhys10-V). Axelrod's model for the dissemination of culture will be explained as well as the history of the development of this and similar models. Also the python implementation of the model which was also part of the study achievement will be discussed.

1. Introduction

In 1997 the paper *The Dissemination of Culture: A Model with Local Convergence and Global Polarization* by Robert Axelrod was published in the Journal of Conflict Resolution[1]. This publication is preceded by numerous works dealing with the same or similar phenomena.

Despite all the exchange and a general tendency towards convergence, why are there still social and cultural differences existing? Previous works have given useful explanations:

People identifying with a certain group often times actively emphasize their difference to other groups[2]. For ethnic groups this can reinforce cultural boundaries[3][4].

There are certain dynamics like fads and fashion.

Preference for extreme views[5][6][7].

Drift.

Geographic isolation[8].

Specialization[9].

Change in environment.

A key principle these other mechanisms are not employing is that exchange is most successful or mostly occurring between individuals that are similar in culture.

There are dozens of different ways to define the word *culture* and most of them remain very vague and therefore hard to formalize scientifically. In his work Axelrod required culture to satisfy two conditions:

The more cultural attributes two individuals share, the more likely their interaction will be.

An interaction between two individuals tends to increase their similarity respectively the amount of shared attributes.

It intuitively makes sense that if only these conditions were implemented in a simulation, there would be a drive towards a higher similarity globally. But interestingly this does not necessarily lead to a global convergence where all individuals share all the same cultural attributes, as he shows in his paper.

The model he introduced is based on three principles:

Agent-based: The changes taking place are determined by interaction between agents and are only local.

No central authority: In this model each agent is independent of any authoritary unit, which also poses an abstraction of the real world, where powerful leaders like politicians, cult leaders heavily influence interactions.

Adaptive agents: In the model there is only an adaptive interaction taking place which is not directly following any optimization process to maximize fitness or any other rational principle.

2. The model

As already mentioned, Axelrod used an agent-based model. These are acting on a grid where the local changes given by interactions between two agents can be determined. Each agent is represented by a cultural vector with M features where each of these features could take values of the N possible traits.

2.1. Initialization

The initialization of this model is by choosing a length L so that the grid is of size LxL. The initial cultural vectors a chosen randomly for each agent, where each of the N traits a feature can be is represented by a number from $[1, N]$. So for the example of an agent with a cultural vector of 4 features with 3 possible traits could then be $(1, 3, 2, 2)^T$. An example how an initial grid could look lke is illustrated in Fig. 1.

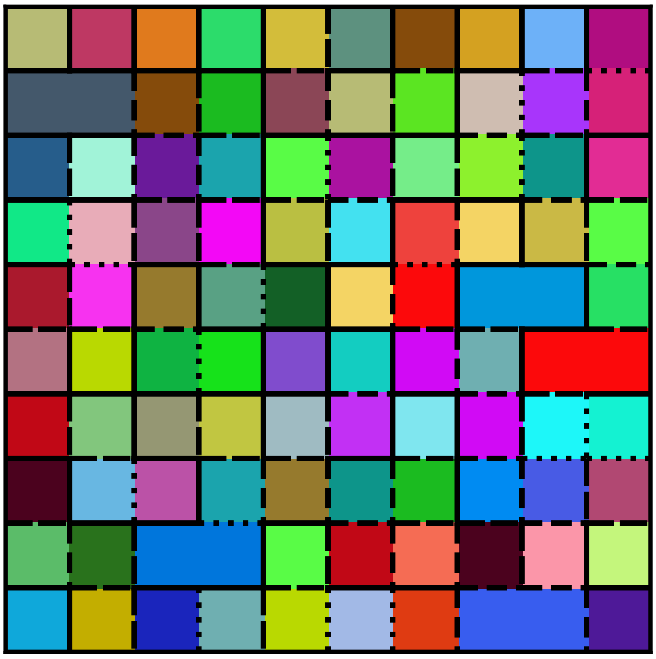

Figure 1. Example of an initial grid of size 10x10 with cultural vectors of 5 features with 3 possible traits. Every color denotes a different vector, the lines between the agents denote the similarity depicted by the dot density.

As can be seen, in the initial configuration there is a lot of fragmentation. In the following time steps this fragments will perish due to the pairwise interaction of the local agents reaching a globally more similar culture.

2.2. Interaction

This is an iterative process where every iteration step looks like the following. One random agent across the grid is chosen. Then a random neighbor of this agent (the definition of neighbors is variable and leads to different results as discussed later) is chosen and their cultural vectors are compared element wise. Their interaction probability p is given by the amount of features they share divided by the total amount of features, therefore giving a value between 0 - for completely different vectors - to 1 - for identical vectors. If the interaction takes place the chosen agent adopts a trait of a feature from its neighbor which they do not share. Then the iteration is repeated. Again for the example of 4 features with 3 possible trais the interaction probability p of one iteration between the agent a at position 2x3 and its neighbor at 2x4 could look like this:

$$\vec{a}_{23} = \begin{pmatrix} 1 \\ 3 \\ 2 \\ 2 \end{pmatrix} \text{ and } \vec{a}_{24} = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 1 \end{pmatrix} \rightarrow p = 50\%.$$

When p is equal to 1 the interaction between the two agents is guaranteed but redundant since for $p = 1$ they must already share all the same features. When $p = 0$ none of the features of the two agents is alike and no interaction takes place. For both of these cases nothing happens and the algorithm proceeds with the next iteration. So the stepwise procedure goes:

1. Choose random agent and a random neighbor of this agent.
2. Compare their features and determine the probability of interaction p.
3. If interaction takes place set one feature of the agent to the same trait of the its chosen neighbor which was different until now.
4. Repeat.

3. Dynamics

3.1. Zones & Regions

To explore the dynamics of the model, two properties are of interest: The *zones* and the *regions*. A region is defined as an area on the grid which has borders of zero similarity to its neighbor. A zone on the other hand is defined as an area where the border agent are just not fully similar to their neighboring agents. As illustrated by Fig. 1 in the beginning there is quite a lot of fragmentation, therefore a lot of cultural zones, usually close to the amount of total agents on the grid since the initial traits of their features is random. The initial amount of regions is usually much smaller but also strongly dependent on the amount of features and possible traits.

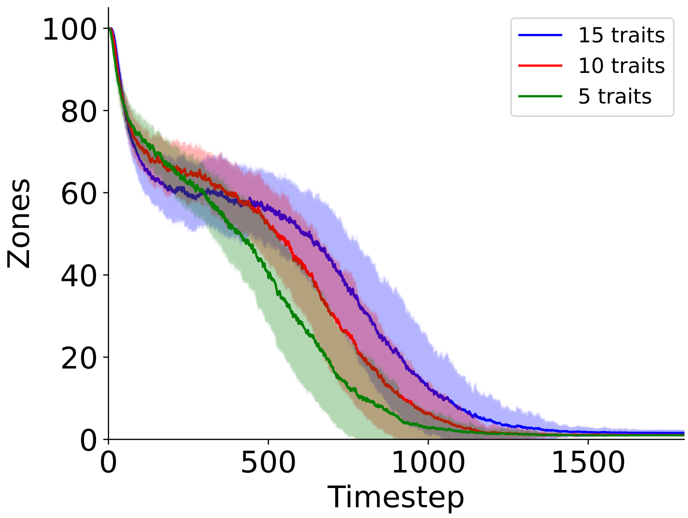

Figure 2. Example of the dependence of the dynamic of the zones on the amount of possible traits. The runs were generated on a 10x10 grid with a cultural vector with ten features. Results were averaged over 100 simulations.

As depicted by Fig. 2 the initial amount of zones roughly stays the same when varying the number of possible traits. This is due to the fact that when initializing the grid with random traits the chance of sharing traits in all the features is small for all numbers of traits shown. The dynamics of the system approaching their fixed point is different though. With more possible traits the initial drop in zones is stronger but overall it takes a longer time until it converges to its final state. Note that one time step is defined as the number of interactions so that each of the agents (on average) was selected one time. For a 10x10 grid one time step would correspond to 100 time repetition of choosing a random agent and performing the interaction with one of its neighbors with the probability based on how similar their cultural vectors are.

For the same conditions as in Fig. 2 the regions were counted. The area around the colored lines marks the respective standard deviations.

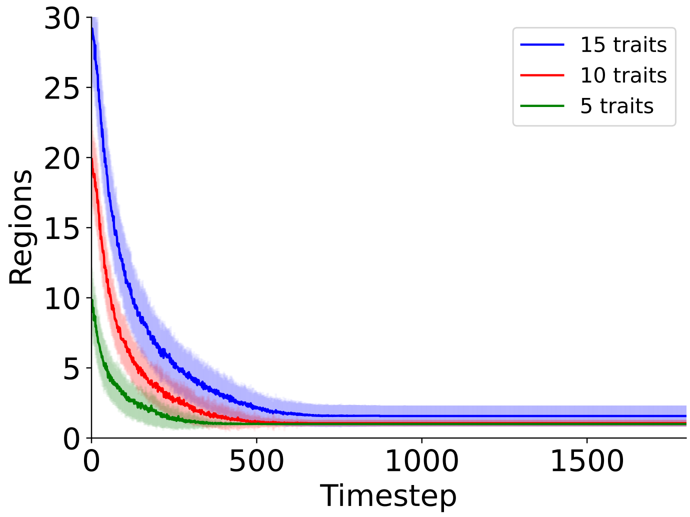

Figure 3. Example of the dependence of the dynamic of the regions on the amount of possible traits. The runs were generated on a 10x10 grid with a cultural vector with ten features. Results were averaged over 100 simulations.

As already stated, the initial amount of regions is much smaller than for the zones. Here also the initial count of regions depends on the number of traits. The dynamics with which the number of regions evolve is quite similar though and only differs slightly in their final number. This slight difference in the final count also applies to the zones in Fig. 2.

Of course the dynamics are not only affected by the number of possible traits but also by the length of the cultural vector, the amount of features it contains. In Fig. 4 the final amount of zones on a 10x10 grid for the respective numbers of traits and features are shown. Note that since the fixed point is reached when each agent has only neighbors of 100 percent or 0 percent similarity, the final number of zones also represents the final region count.

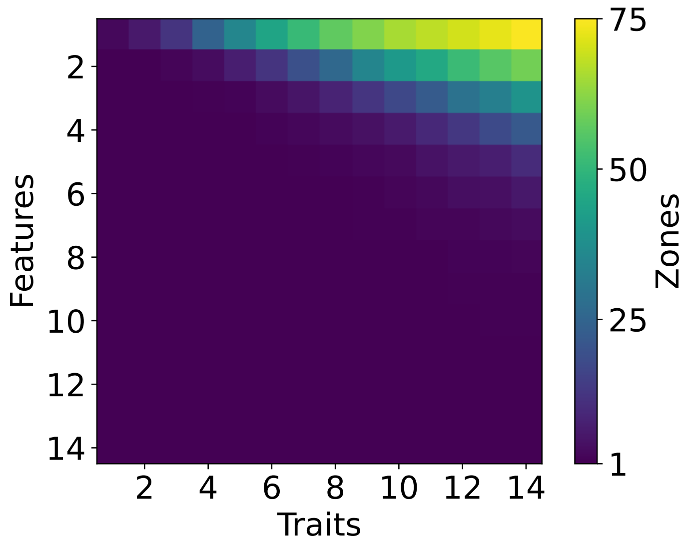

Figure 4. Final amount of zones on a 10x10 grid for a range of traits and features. Results were averaged over 100 simulations.

For most combinations there is a single cultural vector that will prevail. With an increasing number of traits the number of zones/regions also increases while an increase of features has the opposite result. At first this might seem counter intuitive as with more features the total number of possible cultural vectors that can be established increases, but this contradiction can easily be resolved. For more features the chance of at least one having the same trait, so to say enabling an interaction, is simply higher. For this reason with a higher number of features in the cultural vector the final zone/region count decreases.

3.2. Other factors that affect the dynamics

Besides the amount of features and traits there are other parameters that influence on the dynamics. One is the grid size L. The initial number of zones which in Fig. 2-4, for the 10x10 grid was around the number of total agents (≈ 100). This also applies to larger grids (see Fig. 5) but the way the number of zones decays differently for the different grid sizes. Note that in Fig. 5 the amount of zones was divided by the initial number of zones (so roughly divided by the total grid size respectively) in order to visualize all the graphs in one figure and compare their relative decay of zones.

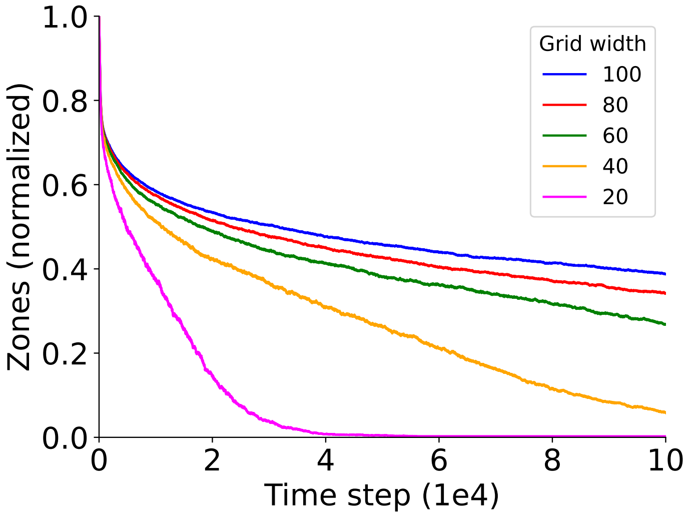

Figure 5. Zones over time steps for different grid sizes for agents with a cultural vector of 7 features with 5 possible traits. Note that the amount of zones was normalized. Results were averaged over 100 simulations.

The simulations for Fig. 5 were only performed up to 10000 time steps. For most of the evaluated grid sizes, namely the larger ones, this was not enough for the system to reach its equilibrium state. But it was already enough roughly guess where that state will be. It can be seen that for a larger grid size the final amount of zones will be much higher for a fixed number of features and traits. If the grid is large enough it allows for a higher amount of fragmentation so that more there will be more zones in the end.

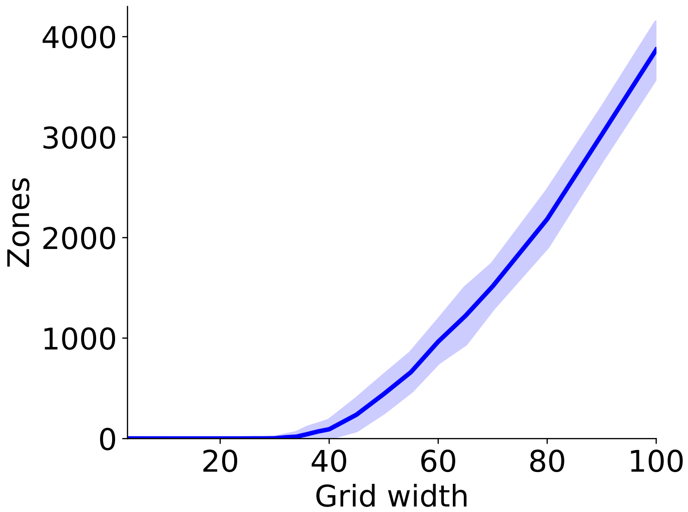

Figure 6. Zones over grid size after 10000 time steps. Agents with a cultural vector of 7 features with 5 possible traits. Results were averaged over 100 simulations.

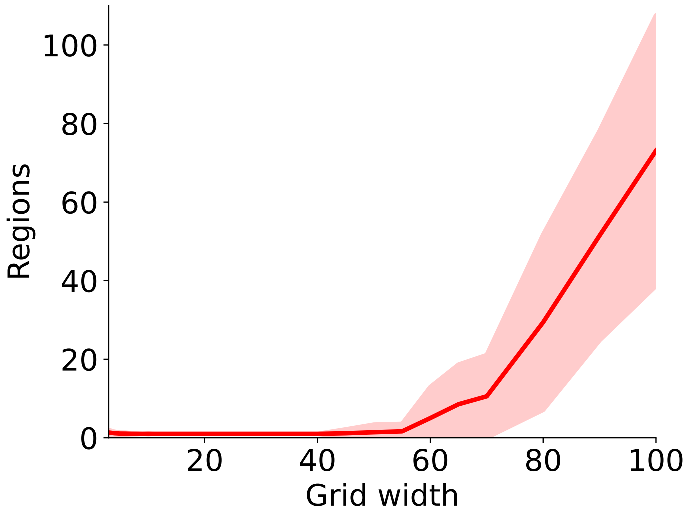

Figure 7. Zones over grid size after 10000 time steps. Agents with a cultural vector of 7 features with 5 possible traits. Results were averaged over 100 simulations.

Fig. 6 & 7 demonstrate this more clearly. Note that this shows the zones and regions after 10000 time steps which, especially for the larger grid sizes, not necessarily represents the final state of the system. As already stated the grid size affects the amount of time steps needed to reach a stable state. Fig. 8 & 9 the convergence times for different grid sizes, which managed to converge during the 10000 time steps, are shown.

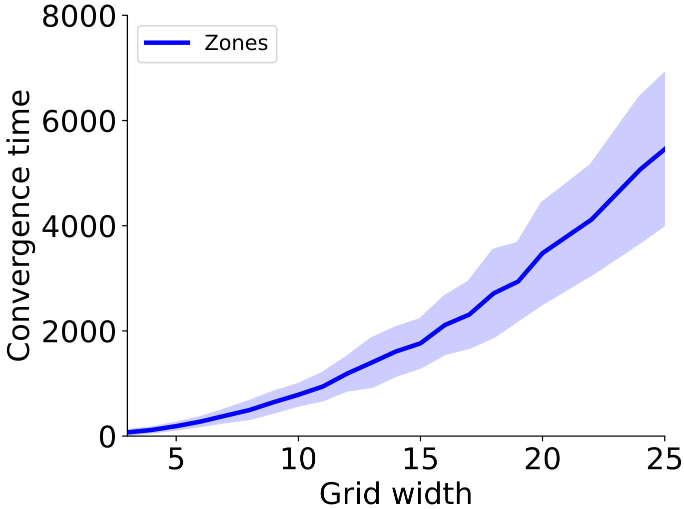

Figure 8. Convergence time over grid size for zones. Agents with a cultural vector of 7 features and 5 possible traits. Results were averaged over 100 simulations.

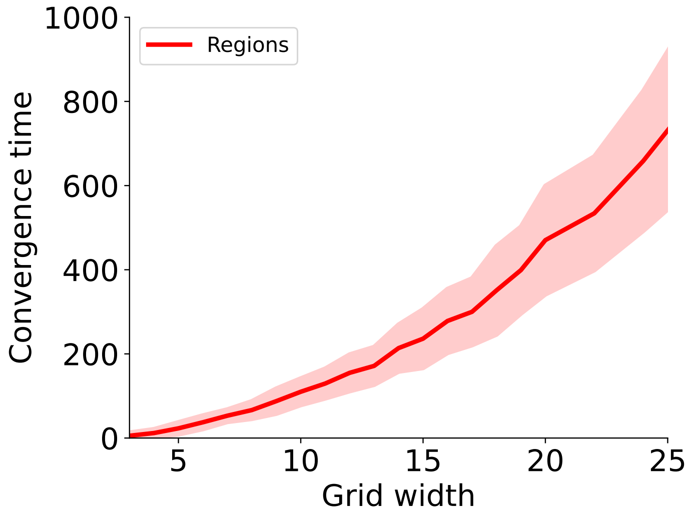

Figure 9. Convergence time over grid size for regions. Agents with a cultural vector of 7 features and 5 possible traits. Results were averaged over 100 simulations.

The graphs look very much alike except for the different scale in the convergence time. The result found is consistent with [1] where it is stated that the amount of events (time steps) it takes until the stable state is reached is proportional to the amount of agents. In Fig. 8 & 9 the grid size is on the x-axis which is why a quadratic relation was found.

Also the number of neighbors which are allowed for interaction of one agent can be adjusted. Up to now all the results were generated by using only the four nearest neighbors (see Fig. 10a). But this does not necessarily have to be the case. Fig. 10 shows four possible neighborhood configurations. The type of neighborhood affects the dynamics heavily because previous borders (for the case of 4 neighbors) can now simply be overcome for a different configuration like Fig. 10c for example.

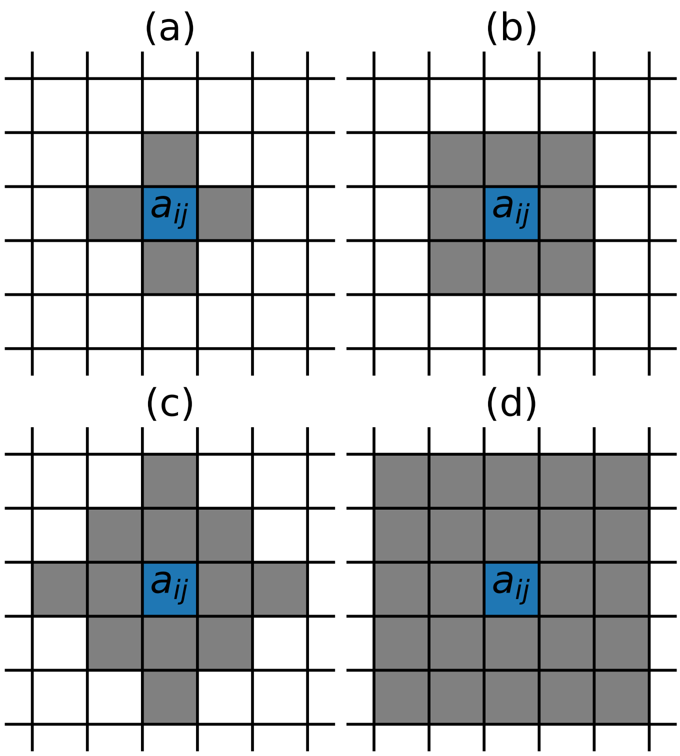

Figure 10. Possible neighborhood configurations for an agent a_{ij}. (a) Only the four nearest neighbors. (b) Eight nearest neighbors. (c) Twelve nearest neighbors. (d) All the agents on the grid.

With increasing neighborhood size there is much less possibility for keeping up the initial fragmentation which can only remain due to the spatial structure on the grid. When an agent is allowed to interact with all the other agents on the grid like in Fig. 10d, in the final state there will always be only one cultural region (as long as in the initial state there is not a single agent who shares not a single feature with any other agent on the grid). Therefore the spatial limitation of the interaction between agents is crucial for cultural zones/regions to remain. This also seems realistic when looking at the actual world.

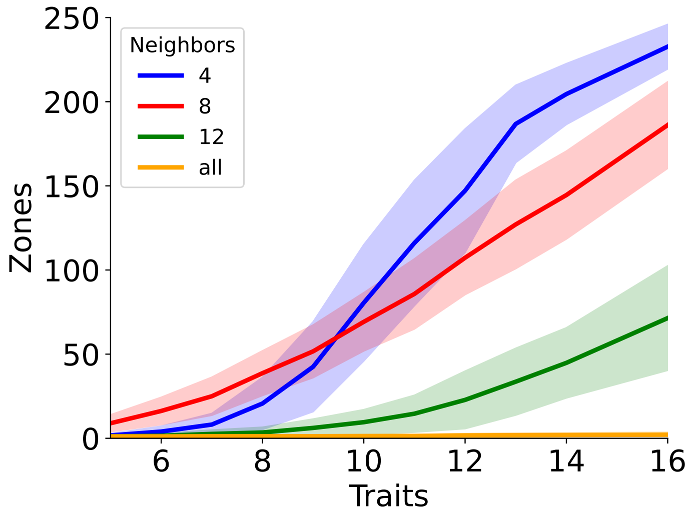

Figure 11. Final amount of zones over range of traits for different types of neighborhood configuration. Agents with 3 features on a 20x20 grid. Results averaged over 100 simulations.

Fig. 11 depicts this impact of the neighborhood size. The previously studied effect of more zones with more possible traits is again observable. With increasing neighborhood size the fragmentation gets lost leading to a smaller amount of zones in the final state.

A last aspect that affects the dynamics is the implementation of periodic boundaries. With this the agents on the border have more possible interaction partners which elevates the chance for having a neighbor with one or more shared features and therefore increasing the interaction probability. So the stable state will generally end up with less zones compared to the case of fixed boundaries but the overall properties, like the increasing number of zones for a higher number of possible traits, will remain the same.

4. Implementation in to a python application

The implementation of the model into python was done using mainly three packages; NumPy for the numerical realization of the model, PyQt for creating the GUI and matplotlib for the visualization of the grid. These three parts will be discussed in the following sections.

4.1. Realization of the model

The easiest part was the implementation of the actual model. Since the agents are on a grid the straightforward way was to save the grid in a matrix where each entry represents an agent. The entry for the agent was then a vector, the cultural vector, with its entries, the features, being one of the possible traits represented by a number, as already demonstrated in section 2.

Using NumPy's random module, a random agent on the grid was selected and given the neighborhood size an appropriate neighbor using the same module was chosen. The rest took place as already described in the prior sections, the similarity was computed and the possible interaction was performed.

4.2. Visualization

The visualization at first seemed like a fairly easy task too. Using matplotlibs matshow function, the grid could be displayed like in Fig.1. The problem with this was that each cultural vector should have a unique color. At best this color should be randomly chosen to not confuse anybody, since there are usually a lot of vectors with the same similarity to a given cultural vector and this similarity is difficult to express in color. For example the vector $\vec{v}_1 = [0, 1, 1, 0]$ and $\vec{v}_2 = [0, 1, 2, 0]$ are very similar so one could come up with the idea to give them similar colors like orange and red. But then $\vec{v}_3 = [0, 2, 2, 0]$ shares a smaller similarity with $\vec{v}_1$ but the same with $\vec{v}_2$ therefore there is a problem to attach a color gradient to the similarity. This is why random colors were chosen.

I then tried to pre-generate the set of random colors for all possible cultural vectors which worked out for the small initial numbers of traits and features. But when these numbers are already slightly increased a combinatoric explosion occurs. The amount of possible cultural vectors for M traits and N features is given by M^N. For 4 features and 3 traits this is feasible with 81 possible combinations but already for 10 and 10 features and traits respectively this scales up to 10^{10} possibilities which takes way to long to compute and therefore would very much limit the application.

For this reason I chose to check for new cultural vectors in each iteration. Every new vector gets assigned with a number which is assigned to a random color. With this every vector keeps the same color throughout the simulation even if a certain vector disappears from the grid for some time due to the dynamics.

Since the color representing the cultural vector of the agent on the grid gives no clue about how similar it is to for example its neighbors I also added the possibility of showing separation lines on the grid where the dot density of the separation lines represents the similarity in steps of 20 percent, so less than 20 percent similarity is depicted by a solid line, for 80 percent or more no line is shown and a gradual increase of dot density for less than 80, 60 and 40 percent. The similarity for the separation lines has to be calculated for each neighbor of each individual agent in every time step. Therefore this takes a lot of computational effort and runs smoothly only on small grid sizes.

4.3. Embedding into a GUI

Most of the time of the project went into this part. This was the first time for me creating a GUI so I had to look into the PyQt documentation many times. I decided to realize the GUI by using the PyQt grid so that every container of the GUI covers one or several squares of the grid. This allowed for an easy assembling of the different features of the GUI and also made a stepwise addition of features possible.

The key element of the application is the aforementioned visualization of the grid which is simply realised by showing the matplotlib figure. matplotlib comes with a good compatibility with PyQt. The sliders to adjust the parameters, namely the grid size, the number of features and traits respectively and the number of neighbors up for interaction, were implemented with the PyQt QSlider class. Each of the respective values of the sliders is shown next to them and also passed on to the model.

Every time after adjusting one of the parameters the simulation is reset to lead to the right results. The pause and reset buttons are established using the QPushButton class and the checkboxes using the QCheckBox class.

The explanation of the model and its properties is embedded into a single scrollable QLabel in which the text elements and images in form of PDF files are placed.

5. Final remarks on the project

When offered to do this project instead of regular weekly exercises for the study achievement, I gladly took it. I generally preferred this option as it offers more freedom and allowed me to organise my working hours freely, for example during lecture-free periods. Also the models which were presented to examine in the project were interesting and I liked working on it. Although I have to say that I underestimated the amount of work that went into this. As already mentioned before in the report, most of the time I spent on the project went into the making of the GUI. Since I had no prior experience with JavaScript the level of the provided GitHub repository from the ComplexityExplorables was too high for me. It was very helpful that later on the option was given to also create the GUI in python. With this option it was possible to make the GUI in a reasonable amount of time even without having worked with PyQt before.

In my opinion it also would have been sufficent enough to just write the report and instead of a GUI just require a simple application of the model in a language of choice with a proper visualization where the adjustable parameters can just be passed from command line or something similar.

The model itself was not too challenging. Exploring it in the creation of the report has definitely helped understanding the model better. For my specific model it was also very helpful to have the Master thesis of Alexander Jochim at hand.

References

[1] R. Axelrod, "The dissemination of culture: A model with local convergence and global polarization," *Journal of Conflict Resolution*, vol. 41, no. 2, pp. 203–226, 1997. [Online]. Available: https://doi.org/10.1177/0022002797041002001.

[2] G. Simmel, *The Web of Group-affiliations: Conflict*. Free Press, 1955.

[3] F. Barth, *Ethnic groups and boundaries: The social organization of culture difference*. Waveland Press, 1998.

[4] M. T. Hannan, "The dynamics of ethnic boundaries in modern states," *National development and the world system*, pp. 253–75, 1979.

[5] R. P. Abelson and A. Bernstein, "A computer simulation model of community referendum controversies," *Public Opinion Quarterly*, vol. 27, no. 1, pp. 93–122, 1963.

[6] A. Nowak, J. Szamrej, and B. Latané, "From private attitude to public opinion: A dynamic theory of social impact.," *Psychological review*, vol. 97, no. 3, p. 362, 1990.

[7] B. Latané, A. Nowak, and J. H. Liu, "Measuring emergent social phenomena: Dynamism, polarization, and clustering as order parameters of social systems," *Behavioral science*, vol. 39, no. 1, pp. 1–24, 1994.

[8] T. C. Schelling, *Micromotives and macrobehavior*. WW Norton & Company, 2006.

[9] N. E. Friedkin and E. C. Johnsen, "Social positions in influence networks," *Social networks*, vol. 19, no. 3, pp. 209–222, 1997.