Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

Swarm behaviour of Antarctic krill

Jules Bachmann, Anna Jaeggi, Felix Sarnthein & Laura Wülfroth

Zurich
December 2017

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Jules Bachmann

Anna Jaeggi

Felix Sarnthein

Laura Wülfroth

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

1 Abstract

The goal of this project was to find out what percentage of a swarm in necessary to move an entire swarm of antarctic krill. The krill behaviour was simulated using the boids model. First, the parameters were adjusted until realistic-looking behaviour was produced by the simulation, then simulations were run with this model with different amounts of krill escaping at different angles. The result is not a simple threshold value, as usually a certain portion of the swarm follows, while the other part doesn't. A non-linear correlation between the percentage of escaping krill and following krill is observed. After 30% of the swarm escapes, the entire swarm follows.

2 Individual contributions

1. Concept idea: *all*
2. Simulation implementation
    (a) Basic structure: *all*
    (b) Force model: *Felix Sarnthein*
    (c) Implementation of escapists and result computation: *Jules Bachmann*
    (d) Plots etc.: *Laura Wülfroth*
3. Thesis:
    (a) Abstract: *Anna Jaeggi*
    (b) Introduction: *Laura Wülfroth*
    (c) Description of the Model: *Anna Jaeggi*
    (d) Implementation: *Felix Sarnthein, Jules Bachmann*
    (e) Simulation results and discussion: *Laura Wülfroth*
    (f) Summary and Outlook: *Jules Bachmann, Anna Jaeggi, Laura Wülfroth*

3 Introduction and Motivations

Swarming behaviour is a very common phenomenon in nature. Animals like birds, fish or ants are the most obvious ones. But also humans often group themselves and act collectively. A swarm is a group of individuals of a similar size which cooperate in a coordinated manner. Most species swarm to get an evolutionary benefit like fish escaping from predators. It is remarkable that swarms have a collective mind without direct communication. Their reaction is only based on a balance of their own experience and their desire to stay within the group. For example if a shark swims into a swarm of fish a few already have experienced this and try to get away, so the rest follows because they try not to loose the group. Swarming has already been observed and simulated a lot but still it is interesting to decipher the rules that keeps a swarm together.

This project is focusing on Antarctic krill and their swarming behavior. Antarctic krill (Euphausia superba) are small crustaceans who live in the Southern Ocean. The length of an individual krill is around 6 centimeters and they can weight up to two grams. They feed on phytoplankton and algae which grow on the underside of sea ice. Antarctic krill live in huge swarms which can reach a density of 10'000 to 30'000 individuals per square meter and can even be seen from space. Thus it is a widespread and abundant species in the Southern Ocean. Krill is the main food source of a lot of animals like whales, seals and different sea birds all depend on their existence. However, because of various reasons like fishery or melting sea ice the amount of krill is decreasing which has an effect on the whole ecosystem of the Southern Ocean. To understand krill and maybe even predict their future fate, depending on climate change and human impact gets more and more important. But to come to this point we have to understand how krill swarms work and what causes them to move. The goal of this project is to understand how individual krill changes the behavior of the whole swarm. In particular we want to know which percentage of "escaping" krill causes a swarm to change direction and to follow them.

4 Description of the Model (1)

The boids model was developed in 1986 to simulate the flocking behaviour of birds and fish by Craig Reynolds. It describes behaviour of a swarm emerging from identical individuals following a set of simple rules. The boids model is a "social force" model, meaning the individuals experience forces from their surrounding individuals, and in some cases nearby objects, for example walls, food sources or predators. In the basic boids model the force acting on an individual is a combination of three components: Separation, Cohesion and Alignment. All these forces require the definition of a neighbourhood from which the average position or average speed is calculated. The neighbourhood is characterized by a radius, which can be different for each component. In some implementations the neighbourhood isn't circular but is missing a "slice" of a certain angle, simulating the limited field of vision of certain animals. For our model this is not necessary, since krill sense their swarm mates through vibrations of the surrounding water, not through an asymmetric field of vision. The three components can be weighed differently to produce realistic behaviour.

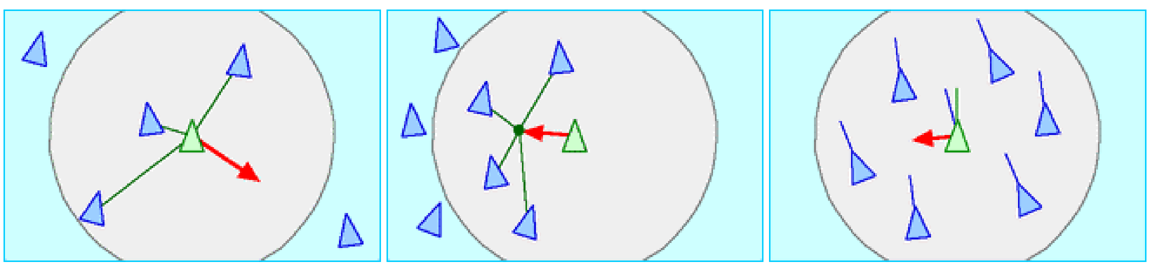


Figure 1: Schematic representations of the three components of the force acting on an individual krill: Separation, cohesion and alignment. The circle represents the neighbourhood in which krill are taken into account. (1)

4.1 Separation

Separation leads individuals to steer away from nearby swarm members. This force is basically the sum of individual repulsive forces from individual krill in the neighbourhood. The force acting from one krill on the other is proportional to their distance, but negative, so the closer they are, the larger the repulsive force is. This is a description of behaviour to avoid collisions between individuals.

4.2 Cohesion

Cohesion simply makes and individual move towards the center of mass in the neighbourhood. The force is proportional to the distance to that center of mass, so the further away the krill are from their swarm, the stronger the force draws them to the swarm. This is likely a behaviour that evolved to ensure that the krill would stay in the swarm, where it is safer.

4.3 Alignment

Alignment doesn't take into account the positions of surrounding krill but their velocity, and adjusts the velocity of the individual to match the average velocity of other krill in the neighbourhood. The forces acting on krill in the neighbourhood is averaged and a fraction of that is the alignment component force. This behaviour quickly communicates movement through the swarm, which is useful in case of a sudden attack.

4.4 Escapists

For our project, we have incorporated so called escapists, which start moving towards a designated target vector after the simulation reached a certain time stamp. While doing so they are still influenced by the other krill, but are mainly following their target vector.

5 Implementation

5.1 Autonomous Character Model

In his paper 'Steering Behaviors for Autonomous Characters', Craig Reynolds proposed a solution for models and animations of autonomous individuals. Autonomous meaning in this context that the characters follow a set of rules individually without superior coordination. The paper describes many rules and properties. For our project it was enough to implement a simplified version. In our implementation each individual is defined by its position and its current velocity. It has knowledge of its environment and can adjust the velocity according to different steering rules. This creates the impression of the individual reproducing a specified behavior. Simple rules are independent from other individuals, we call them non-social steering behaviors, while complex rules can reproduce social behaviors. We implemented the following steering behaviors: seek, fee, wander, cohesion, separation, alignment

5.2 The Steering Force

The concept of the steering force is what makes the animations feel natural. To explain the concept we use the steering force 'seek' as an example. The individual seeks the target and thus its desired velocity is towards the target. It would not be natural for an individual to just turn around immediately and change the current velocity towards the desired velocity.

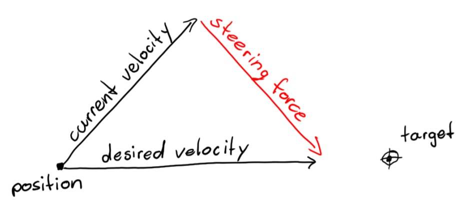


Figure 2: The steering force is the difference between the current and the desired velocity

Instead Reynolds proposed to compute the steering force for each specific behavior. Then we can process and combine multiple forces. The final steering force is then added to the current velocity to obtain the subsequent velocity.

steering_force = desired_velocity − current_velocity
steering_force = truncate(steering_force, max_force)

current_velocity = current_velocity + steering_force
current_velocity = truncate (current_velocity, max_speed)

position = position + velocity

This concept is an important abstraction of the whole implementation because, it enabled us a modular approach where we only had to compute the desired velocity for each steering behavior.

5.3 Non-Social Behaviours

5.3.1 Seeking

As shown in the example above, the seeking behavior consists of seeking the a target. Thus the desired velocity is just towards the target.

desired_velocity = target – position

5.3.2 Fleeing

Similar to the seeking behavior the flee behavior consist of fleeing a target. Thus the desired velocity is away form the target.

desired_velocity = - (target – position)

5.3.3 Wandering

Reynolds proposed a smart way of achieving random wandering. The main idea is that the individual seeks a target on a circle in front of it. After each step, the target can move randomly on the circle.

desired_velocity = target – position

5.4 Social Behaviours

5.4.1 Cohesion

As described in the chapter above, the individuals want to steer towards the center of mass of their neighborhood. To detect the neighborhood, we used the rangesearch()-method of matlab which uses a kd-tree algorithm to compute for each point its neighbors within a specified radius r. For every individual we then compute the desired velocities towards every neighbor. So every neighbor is represents a target, when thinking about the seeking behavior. The mean of all those desired velocities is equal to the desired velocity towards the center of mass.

```
neighbors = rangesearch(positions,radius)
for all neighbors i
    desires[i] = neighbor[i] – position
desired_velocity = mean(desires)
```

5.4.2 Separation

To reproduce separation, Reynolds proposed to normalize the vector and then compute the separation desired velocity inverse-linear to the distance. This results in dividing each desire by the square of its length and then taking the negative.

```
neighbors = rangesearch(positions,radius)
for all neighbors i
    desires[i] = neighbor[i] – position
    length[i] = norm(desires[i])
    desires[i] = - desires[i] / length[i]^2
desired_velocity = mean(desires)
```

5.4.3 Alignment

As expected, the desired velocity of resulting from the alignment behavior is the mean of all neighbors.

```
neighbors = rangesearch(positions,radius)
for all neighbors i
    desires[i] = neighbor_velocity[i]
desired_velocity = mean(desires)
```

5.5 Escapists

The escapists are implemented to take action once the simulation has been running for *lactiv*, which we set to 50, turns at line 38 of the main function. The function *leading* then computes the new vector for the escapists which try to reach the *targetV* vector by building a new direction vector built by applying the maximal steering force to their current vector in the direction of $targetV$. The 0.75 factor was an attempt to slow down the escapists, which were too fast, about which we forgot later on. The final vector for the escapists is then computed by combining the new vector calculated in the first part of the function with the F vector which was previously calculated in main with the different influences on the krill. This combination is weighted by the *leadRatio* variable which we set to 0.5, weighing both vectors the same. The $leaders$ variable in the initialization is the percentage of escaping krill while in main the variable holds the exact amount of leaders for computation purpose, whoch are going to always be the $leaders$ first krill int the Position and Velocity vectors. $targetV$ is the vector which the escapists want to achieve. *lactiv* and *leadRatio* have already been explained above.

5.6 Result computation

Our simulation gives out the matrices R and C as result. R holds the number of non-escapist krill which had a vector close to $targetV$ at every moment in time. C gives out the number of non-escaping krill that are in sensing range of an escapist. At every simulation step of every simulation these two computed values are saved into the matrix indices corresponding to the simulation number as row number and turn of the actual simulation as the column number. This computation takes place in the *compResult* function where first every non-escaping krill is tested whether his vector is close to the target vector $targetV$, the variable $errMargin$ defining how close it has to be. Afterward *compResult* testes for every non-escaping krill if there is an escaping inside it's sensing distance. The variable *closeL* is used as value of the sensing distance, but put into a separate variable in case we wanted to change the distance for the result measurement.

5.7 Technical error

Well, Matlab is column-major, and we knew that. Nonetheless we decided to build the vectors P and V in a form that is more appropriate to row-major systems because we were comfortable and found this was intuitively better looking. We realized how bad of an error this was only too late, when we finally had completed the program and went on running the simulations. We really ought to have built these vectors in a more appropriate way and apologize here for the additional time you'll be using if you happen to use our program.

6 Simulation Results and Discussion

The escaping krill in this simulations swims to the left by 90°. Following krill was defined as krill that swims in a similar direction as the escapists. The distance between the individuals of the swarm and the escapists was also considered. A krill is close to an escapist, if it has an escapist in it's sensing distance. As result after the simulation, the plot in figure 3 was created. In the beginning there is a small peak in the number of similar direction. In that moment the krill form themselves to a swarm and swim chaotically but somehow in a similar direction. When the escapists decide to leave the swarm (after 50 calculation steps) some of the non-escapists follow. But because of the standard deviation we got in the number of krill with similar swimming direction, it's not possible to give an exact number. However, it is possible to see that the escapists are a lot faster. The swarm reacts slowly, so the escapists swim away. This explains the fast loss of close non-escapists after 100 calculation steps.

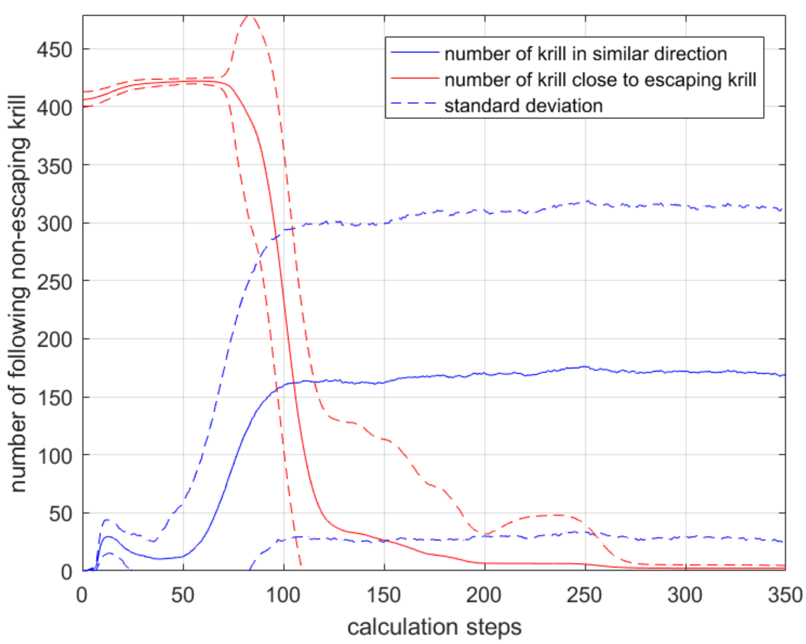


Figure 3: Result of simulation with direction [0,1] and 15%

For each percentage from 5 to 50% in steps of 5 the krill swarm was simulated and plotted like in figure 3. The number of following krill was calculated in percentage to the non-escaping krill, in order to compare them. This was done because if there are

50% escapists, there are less non-escapists then if there were 10% escapists. Now the point after 125 calculation steps was taken and compared to the others. The result is a bar diagram that shows the percentage of not escaping krill that has a similar direction as the escaping ones at the point with 125 calculation steps.

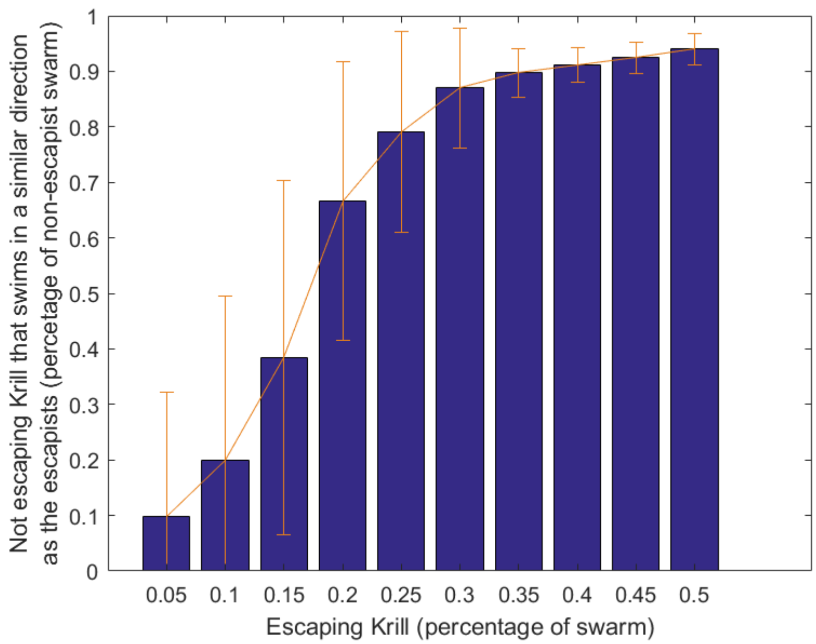


Figure 4: Followers of the krill swarm dependent on escaping krill percentage after 125 calculation steps

After around 30% of escapists the whole swarm completely follows the escapists. This can be seen in Figure 5. The escapists swim ahead and the whole swarm follows. But in the diagram the mean is not 100%. This is due to the variation of swimming directions in a swarm, so the direction can be not similar to the escapists direction although it is still in the swarm following the escaping krill. Also it is possible that a few krill were swimming around alone and have not been in the swarm before. At around 20% escaping krill the escapists influence the swimming direction of the swarm but they are not following them, as can be seen in figure 5. The swimming direction of the escapists and the swarm is now really similar, so it is possible that krill in the swarm is counted to the followers because of the direction variation, although they are not. So the percentage at 20% escaping krill in figure 4 is a little bit higher then it actually is.

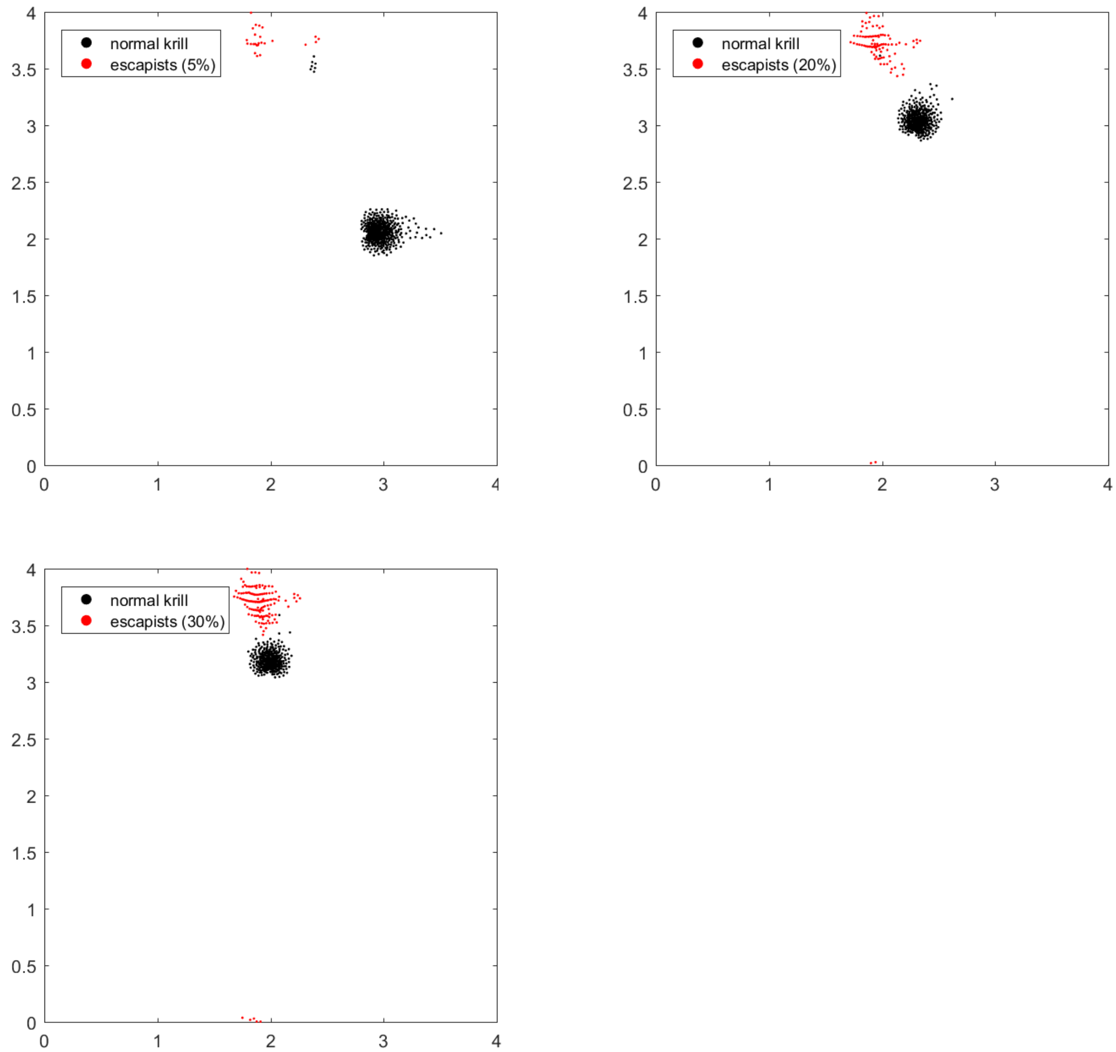
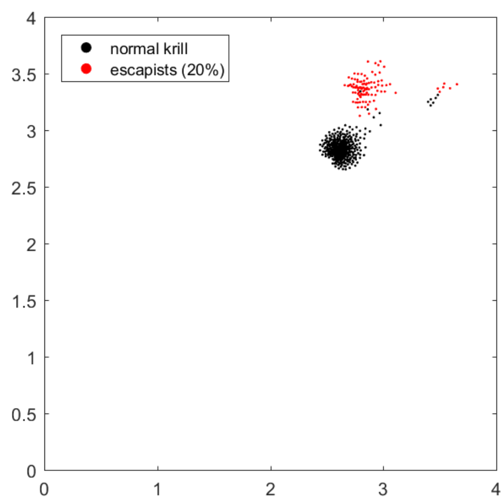


Figure 5: Krill swarm with 500 individuals after 125 calculation steps

Up to 20% of escapists it is hard to say how much krill exactly become followers because there is a huge variation in the simulation. However, it is possible to say that it is a lot less then 100%. Beyond 30%, however, it is clear that the whole swarm follows the escapists. So the point where the entire swarm changes its direction by 90 degrees is 30%.

The same simulation with the krill swimming away left by 45 degrees has slightly different results. Already with 10% swimming away the swarm adjusts its direction as in the last simulation with 20%. So in the simulation only a few really follow but the whole swarm has a kind of similar direction. This explains the to high percentage with 10% escapists in figure 6 (left). However, the second simulation clearly shows that the swarm follows with less percentage of escapists. Already with 20% escapists the entire swarm follows, as can be seen in figure 6 (right). This makes sense because the swarm can adjust small direction changes faster then larger ones. In the first simulation with 90 degrees the escapists were already out of the sensing distance of the individuals of the swarm when the swarm started to change the direction.



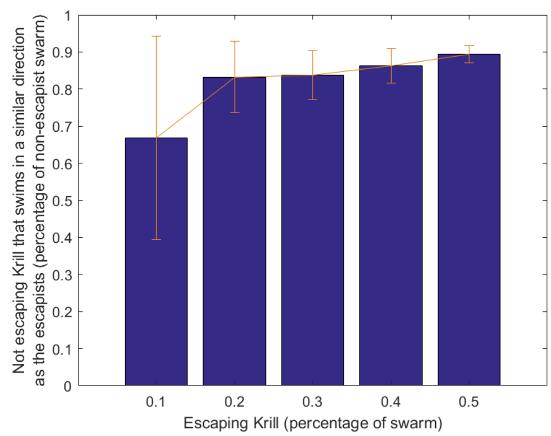


Figure 6: (left) Krill swarm with 500 individuals after 125 calculation steps with direction [1,1] and (right) the bar diagram: followers of the krill swarm dependent on escaping krill percentage after 125 calculation steps

The problem with our model is, as described before, that we have a large variation of swimming directions, so we defined a range. This range sometimes causes individuals to be described as followers although they were not. In reverse a lot of krill were not counted to the following ones because their direction was out of the range although they were followers of the escapists.

7 Summary and Outlook

To summarize, the idea of this project was to get the percentage of "escaping" krill that causes an entire swarm to move in the same direction as the escapists. We used the so called boids model and implemented some away-swimming krill after 50 calculation steps. To analyze the amount of following krill we considered two variables: (1) amount of krill that has a similar direction to the escapists and (2) amount of krill that is close to the escapists. We simulated various percentages and two different escape directions. We discovered that with a direction change of 90 degrees the whole swarm follows after 30% of the swarm escaped. With a direction change of 45 degrees only 20% is needed. So we got the result we anticipated but for further simulations another approach must be used because the direction varies too much. For example as described in the results at 20% the swarm changes its direction but does not follow the escapists. However, we couldn't measure this direction change. In a further analysis the direction of the swarm could be considered as well, to see not only if the swarm follows, but also if not, if the swarm adjusted the direction. For future simulations, other variables could be considered as well. Since it is known that unique krill can swim against ocean currents but that matter being unknown for whole swarms (2), we could take a look at a situation where escapists are swimming against the current of an ocean stream and see whether the swarm is able to follow.

In future simulations, one could examine if the position of the escapists within the swarm influences the percentage that follows. Instead of just choosing escapists randomly from the swarm, a small group could be positioned at the center or edge of the swarm and the results observed.

8 References

[1] *https://www.red3d.com/cwr/boids/*,Craig Reynolds, Last update: September 6, 2001

[2] Siegel, Volker: Biology and Ecology of Antarctic Krill, ISBN 978-3-319-29277-9, 2016