Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Project Report

Predator-Prey Swarming Model

Jonas Purtschert & Luca Tondelli

Zürich
Date 11.05.2014

ETH

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):

Predater-Prey Swarming Model

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

Name(s):	**First name(s):**
~~Jonas~~ Purtschert	Jonas
Tondelli	Luca

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)**
Zürich 13.05.2014	J. Purtschert
	Luca Tondelli

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

Agreement for free-download

We hereby agree to make our source code of 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.

Jonas Purtschert Luca Tondelli

Table of Content

1. Abstract

In this paper we examine the predator-prey swarming model published by Vladimir Zhdakin and J. C. Sprott in 2010 [3].
First we explain the model and mention all of its variations. Then we compare some results that got published in the paper mentioned above to the new created results to check if the new implementation fits the one made by V. Zhdakin and J. C. Sprott. Next we discuss how each parameter of the model influences the results and how the parameters should be interpreted.
Another important point is the question which parameters deliver the most realistic results. We determine what range the parameters should be chosen to create a simulation as close to the real world as the model allows.
At last we discuss which aspects of swarming and hunting behaviours of fishes are fulfilled and which are not.

2. Individual Contributions

Special thanks go to Vladimir Zhdakin and J. C. Sprott for giving us the guideline for this paper and providing us an insight into their model.

3. Introduction and Motivation

A fish swarm that is under attack by sharks are spectacular to watch. It moves as a unit to minimize every fish's risk of getting killed. There are many models that try to describe and examine the movement of the swarm and the predators. One of it is an agent-based model by Vladimir Zhdakin and J. C. Sprott that tries to reproduce the movement by using forces between the agents and friction between every agent and the environment. The model is kept simple at the expense of making it as realistic as possible. There are only a few parameters to make it possible to explore most of the variations in reasonable expenditure of time. Another reason for this simple model is to keep the simulation of swarming independent of specific organisms, which would need to include also organism specific parameters and properties.

4. Description of the Model

The model consists of two agents, predators and preys. They are simulated on a two dimensional Cartesian plane. Because the model will be kept as simple as possible the agents do only interact with the environment via friction.
All forces between the agents are directed radially from the agents. Prey pairs interact with long-range attractive and short-range repulsive forces with each other. This is used to model the behaviour in a swarm as the single members try to stay at a certain distance from each other. Predator-prey pairs both interact with an anti-Newtonian force.
This model will use three different forces between predator pairs which all occur in nature. First there is no interaction between predators, this will simulate attackers, which do not hunt or interact with each other at all. Second is an attractive force between the predators so they can form a group of predators to chase the swarm. Thirdly this paper will explore a repulsive force

between the predators. This could be a model of predators trying to attack a swarm from different sides to confuse them. Because the anti-Newtonian force is non conservative the system is able to stay indefinitely in motion.
The long-range and short-range force fij, gij between agent i due to agent j are given by:

$$f_{ij} = r_{ij}^{\gamma-1} * (r_j - r_i) \quad (1)$$

$$g_{ij} = r_{ij}^{\alpha-1} * (r_j - r_i) \quad (2)$$

$$z_{ij} = f_{ij} - g_{ij} \quad (3)$$

Where r_{ij} is the distance between agent i and agent j, and γ, α are the force parameters for long-range attractive and short-range repulsive force respectively. The model uses $\alpha < \gamma$, the reason for this will be explained in the next section. Long-range force will only be used for attraction and short-range force for repulsion.
The resulting force for the motion of a prey is:

$$\ddot{r}_i * m_0 = \sum_{prey}(f_{ij} - g_{ij}) - \sum_{pred} f_{ij} - b_0 * \dot{r}_i \quad (4)$$

Where m_0 is the mass of a prey and b_0 the coefficient of the friction. The subscript 0 denotes parameters of prey agents whereas x denotes parameters of predators.
An additional restriction to the above formula is $(i \neq j)$. The first sum adds all forces acting on prey i due to all other preys, except itself. It accounts the attractive and repulsive forces between the preys that form the swarm. The second summation adds all forces acting on prey i due to all predators, which is responsible for the anti-Newtonian. The last term is the friction of the prey and the environment, which is proportional to the velocity of the prey.
Next the three forces between predators are considered.

$$\ddot{r}_i * m_x = \sum_{prey} f_{ij} - b_x * \dot{r}_i \quad (5)$$

In this formula there is no interaction between the predators. The summation over all preys j is responsible for the force directed in direction of the preys. The last term is the friction, which is the product of the predators' coefficient b_x and it's velocity.

$$\ddot{r}_i * m_x = \sum_{pred}(\pm f_{ij} - g_{ij}) + \sum_{prey} f_{ij} - b_x * \dot{r}_i \quad (6)$$

By choosing a minus or a plus in the first summation considers a repulsion or attraction between the predators respectively. Note that in the repulsive case of predators the short-range repulsive force is not essential.
At the beginning of a simulation the predators and preys are distributed randomly in a certain interval on the on the plane, which should approximate a swarm, that is unorganised at the beginning.
The choice of the parameters in the model is crucial for the behaviour of the swarm. Some of the choices as seen below will give a more realistic behaviour whereas others result in odd, not swarming typical behaviour.

5. Implementation

For the implementation of the model we used MATLAB. The starting point of the predators and preys are randomly chosen within a defined interval. For generating random numbers the MATLAB build-in function rand is used. In general MATLAB build-in functions are used as often as possible due to optimization reasons.
For the computation of the iteration steps the Euler method is used. Runge-Kutta could also be used to get more accurate results. For the purpose within this computation we considered the Euler method as accurate enough.
Our first implementation used nested for loops to do the computation, which is pretty slow. Due to the fact that MATLAB is optimized for matrix operations we changed the program such that the most computations are made with matrices, which increased the speed of the execution of the program. We updated all agents in one step, which we think is the best way for our simulation.
Further the computations are well commented and should be easy to understand.

6. Interpretation of the parameters

Here we discuss the different parameters that appear in the model. We figure out how to interpret them and how to choose them to get a realistic simulation.

6.1 The motion parameters γ, α

These two parameters define two of the forces that apply to the agents. γ defines the attractive force and α defines the repulsive force. In this section we always look at the force between two preys.

The first thing to notice is that they have to be negative. To understand why, we have a look at two agents. The two forces between these two preys, attractive and repulsive, have to be strong when they are close together, i.e. r_{ij} is small. Additionally they have to be weak when they are far apart, i.e. r_{ij} is big. To achieve this, we can simply choose negative motion parameters.

The next observation is that α has to be smaller than γ. This follows from the fact that the agents should swarm. The attractive force has to be stronger than the repulsive force when agents are far apart. But when the agents get too close the opposite situation has to appear, i.e. the repulsive force is stronger than the attractive force. This prevents agents from swimming into each other. Choosing α to be smaller than γ and using them as exponents on the distance between the agents creates this situation. In this case, when the distance between the agents gets smaller than 1, the repulsive force becomes stronger than the attractive force.

To find out how to choose the motion parameters to get a realistic result we examine two properties between them:

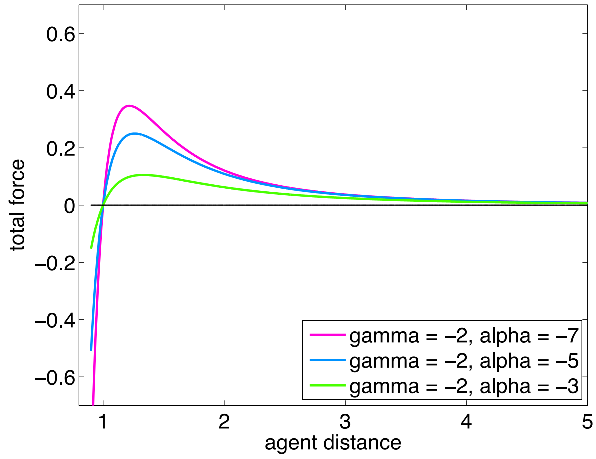

Figure 1: Shows the total force on an agent due to another agent in relation to their distance

The first property is the difference between α and γ. The effect of this difference can be seen on figure 1. It shows the total force on an agent due to another agent in relation to their distance. When the difference becomes larger, the peak of the force curve increases and the curve becomes less flat. This leads to a faster stabilization of the system. For realistic results we found a difference of 1 to be best because it provides the most continuous movement.

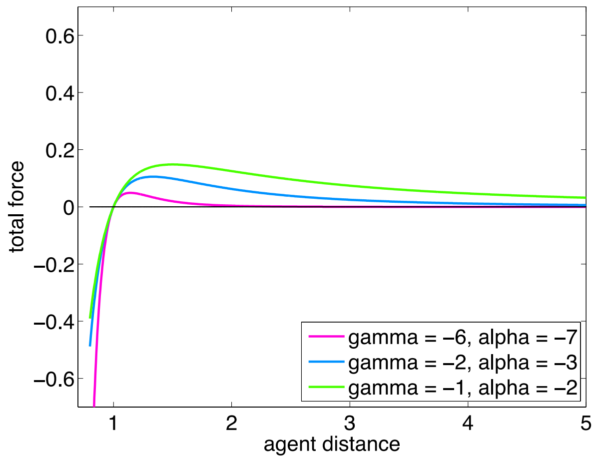

Figure 2: Shows the total force on an agent due to another agent in relation to their distance

Now that we know the difference between α and γ we investigate the second property, which is the value of α. The effect of changing the value of α can be seen on figure 2. It shows the total force on an agent due to another agent in relation to their distance, like figure 1. The most noticeable effect of changing the value of α is that for distances greater 1 the total force becomes stronger when α becomes greater. For most realistic results we choose $\alpha = -2$ and $\gamma = -1$, because in our simulation a distance of 5 between the agents appears quite often, so we need enough force applying on the prey for a natural movement. This is not given for $\alpha < -2$, as we can see on figure 2.

If we choose $\alpha > -2$ the total force becomes too strong, especially for distances greater than 5.

6.2 The Mass m

In this model the mass has just one function. It defines the inertia of an agent. The total force on an agent divided by the mass gives the acceleration of this agent. If the mass becomes too big, the acceleration becomes too slow and the agent more inert. This leads to a slow simulation and faster stabilization, which is unrealistic. On the other side if the mass becomes very small, the opposite happens, i.e. the simulation becomes too fast and it stabilizes very slowly. This is also unrealistic, because it leads to agents jumping around.
For most realistic results we found $0.1 \leq m \leq 1$ to be best.

6.3 The Friction b

The main function of friction is, that agents do not swim away too far. The friction stops agents that swim away from the swarm with high speed. It holds the swarm together and is crucial for a swarming behaviour. The value of the friction is dependent on how many agents there are. If there are more agents, friction needs to be stronger, because it has to neutralize stronger forces. This is because more agents apply a stronger total force on one agent, which implies a greater acceleration. This leads to greater speeds and therefore the friction has to be stronger.
The best value is hard to determine, but it has shown that
$0.1 \leq b \leq 1$ is good for a simulation with $5 - 10$ agents

7. Simulation Results and Discussion

In all our simulations $\propto = -2$ and $\gamma = -1$. The reason for this is explained in the section above.

7.1 Trivial Case: No Predator

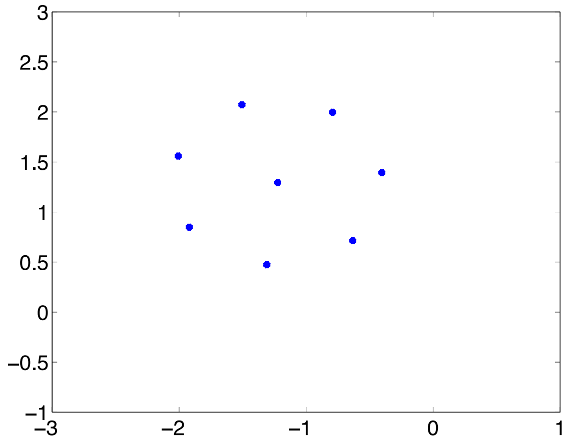

Figure 3: Shows an equilibrium that can appear when using only preys

In this case the equation for motion of the preys is simplified to:

$$\ddot{r}_i * m_0 = \sum_{prey}(f_{ij} - g_{ij}) - b_0 * \dot{r}_i \quad (7)$$

Since there are no predators there is no need to take them into account.
In this trivial case with 8 preys, $m_0 = 0.1$ and $b_0 = 0.1$, the preys eventually form a stationary equilibrium depicted in the figure 3. Further it is to note that there exist different stationary equilibriums for only preys. The only case where they do not form a stationary equilibrium is when the friction is removed in the above equation i.e. the term $b_0 * \dot{r}_i$. That follows from the fact that the energy is conserved, which implies that there do not exist stable equilibriums anymore.
The result obtained in this case equal the result from the paper proposed the model.

7.2. Single Predator

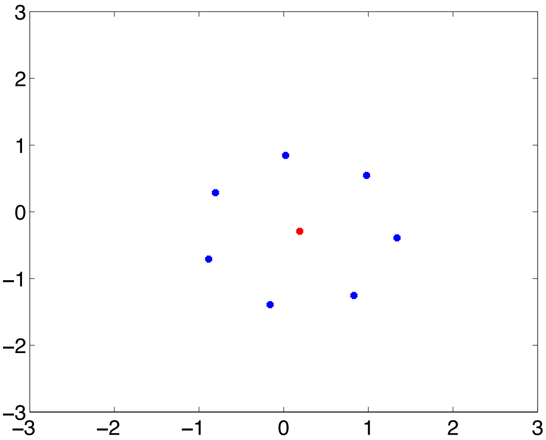

Figure 4: Show an equilibrium that can appear when using one predator and seven preys

This case uses those two equations for preys and predator respectively:

$$\ddot{r}_i * m_0 = \sum_{prey}(f_{ij} - g_{ij}) - \sum_{pred} f_{ij} - b_0 * \dot{r}_i$$

$$\ddot{r}_i * m_x = \sum_{prey} f_{ij} - b_x * \dot{r}_i$$

In general there is no stable equilibrium anymore like in the trivial case with no predators. However, with the configuration $m_0 = 0.1$, $b_0 = 0.5$, $m_x = 0.5$ and $b_x = 1$ the predator gets trapped in the middle of the preys. The situation is depicted in figure 4. This is an unrealistic result and therefore is not discussed further in this paper. Nevertheless it is the same result as in paper [3]. A lot of configurations in the single predator case result in an unrealistic behaviour of the swarm.
By analysis of the parameters in the section above and several tests with those parameters we found out that the range of the parameters described there gives the most realistic in case of a single predator. In most cases the predator follows the swarm and split it into two parts. As soon as the predator is some distance away from the preys, they meet each other again and build a swarm. This increases the chance of survival for the preys as they try to stay close together. Sometimes the predator is able to chase a prey away from the

swarm but instead trying to chase the prey further away, as it would appear in nature, the predator turns into the direction of the swarm. The multiple preys that apply a stronger attractive force on the predator than a single prey cause this phenomenon. It is not possible to change this behaviour of the predator without changing the model, since the predator always gets more attracted by the bigger group of preys. This could also be interpreted as a situation where the predator is able to eat or kill a prey and then turns around to chase more preys in the swarm. It is important to notice that not all configurations in the mentioned ranges result in realistic simulations.

7.3. Multiple predators

This scenario is the most complicated one, because the different interactions between predators need to be considered as well. Therefore this part is divided into three scenarios.

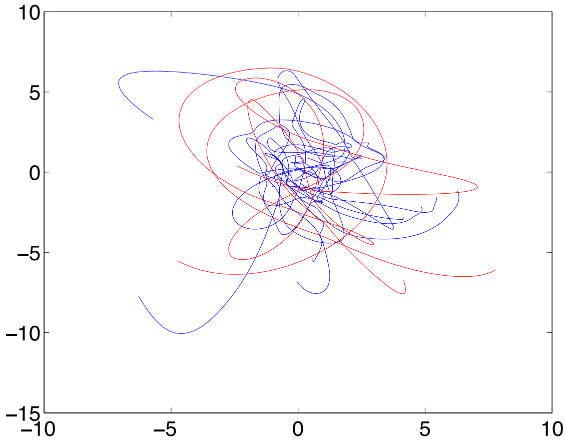

Figure 5: Shows the trajectories of the two predators (red) and the seven preys (blue)

The first scenario shows the case where there is no interaction between predators. The following equations are used for preys and predators respectively:

$\ddot{r}_i * m_0 = \sum_{prey}(f_{ij} - g_{ij}) - \sum_{pred} f_{ij} - b_0 * \dot{r}_i$

$\ddot{r}_i * m_x = \sum_{prey} f_{ij} - b_x * \dot{r}_i$

This scenario corresponds to the case where the predators do not interact with each other. In the most of the configurations the predators have a chaotic trajectory. There are configurations, where the predators circle around the group of preys from time to time. But it is too unstable and chaotic. Figure 5 shows the trajectories with the following parameters: of $b_x = 0.3$, $m_x = 0.1$, $b_0 = 0.2$, $m_0 = 0.1$, 2 predators and 7 preys.

When two predators come close to each other they often move along the same trajectory. The reason for this is that once they come close enough together the acting forces on them will be nearly the same. Nevertheless they normally split after a while.

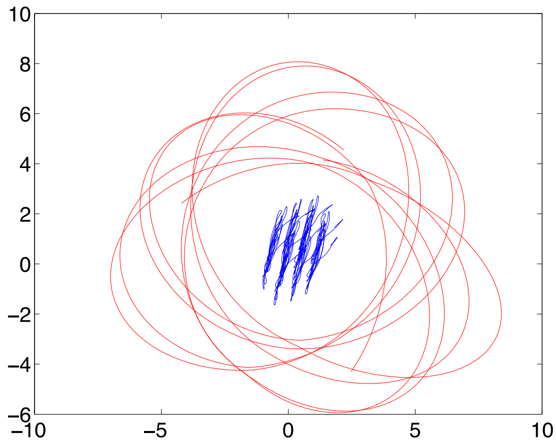

Figure 6: Shows two predators (red) circling around the preys (blue) as a group.

The next scenario shows the case where the predators interact the same with each other as the preys i.e. they try to build a swarm. The following equations are used:

$\ddot{r}_i * m_0 = \sum_{prey}(f_{ij} - g_{ij}) - \sum_{pred} f_{ij} - b_0 * \dot{r}_i$

$\ddot{r}_i * m_x = \sum_{pred}(f_{ij} - g_{ij}) + \sum_{prey} f_{ij} - b_x * \dot{r}_i$

In contrary to the above case where the predators sometimes behave exactly the same, they always hold a certain distance between each other due to the short-range repulsive force.

With a configuration of $b_x = 0.05$, $m_x = 1$, $b_0 = 0.5$, $m_0 = 0.5$, 2 predators and 10 preys, it first looks that the predators do some random walk with the preys in the middle. But after some time, this can take seconds but also minutes, the predators circulate around the prey swarm in their middle, this is shown in figure 6.

Another interesting behaviour is given by the configuration $m_x = 0.2$, $b_x = 0.2$, $b_0 = 0.2$, $m_0 = 0.1$, 2 predators and 7 preys. This is more realistic scenario in which the predators try to stay together, but after a while mostly chase the preys alone. They do often separate the swarm and go after some preys that lost the connection to the swarm. Anyway after a while the swarm finds together again and the scenario starts over.

The last scenario shows the case where the predators repulse each other. These are the corresponding equations:

$\ddot{r}_i * m_0 = \sum_{prey}(f_{ij} - g_{ij}) - \sum_{pred} f_{ij} - b_0 * \dot{r}_i$

$\ddot{r}_i * m_x = \sum_{pred}(-f_{ij} - g_{ij}) + \sum_{prey} f_{ij} - b_x * \dot{r}_i$

In this scenario the predators avoid each other. With this kind of interaction they often attack the swarm from different sides what is close to realistic behaviour, where the predators try to surround the preys. The predators often rush through the swarm and divide them for a short time into multiple smaller swarms until the swarm eventually finds together again. This outcome can be observed with the following configuration: $m_x = 0.3$, $b_x = 0.2$, $m_0 = 0.1$ and $b_0 = 0.2$. Also remarkable is that the swarm finds together relatively quick after a predator divided them into two parts. In contrast to the configuration where $m_x = 0.2$, $b_x = 0.2$, $m_0 = 0.1$ and $b_0 = 0.2$, where the preys need a long time to find together when they got separated. In some cases the latter configuration leads to a separation of the swarm where one part gets chased by the predators.

8. Discussion of the Results

The results obtained in the above section where there is no predator end up in a stable equilibrium except the case where the friction is removed. This is clearly not a realistic result of swarming behaviour, which makes the model useless for swarming behaviour without predators.
In the case of a single predator there are several configurations, which lead to a more realistic behaviour of the swarm. Nevertheless the parameters have to be chosen carefully. The change of just one parameter can result in an unrealistic behaviour of the swarm, for example too fast moving agents. Further there exist some configurations that end up in a periodic movement of all agents, which is also not realistic behaviour. If the right parameters are chosen, the simulation leads to a realistic and contiguous movement of the agents. There exist several of such configurations that approximately simulate a realistic swarm attacked by a predator.
With multiple predators we need to consider three subcases that differ in the forces between predators. The difference between the simulations with no force and with attractive force is small. The most remarkable difference is that the predators in the attractive force model do not come as close as in the model with no force. This follows form the repulsive force in the attractive case, when predators are too close by each other.
Predators with a repulsive force between each other result in a simulation, which differs the most from the other cases with multiple predators. Here the predators normally surround the swarm of preys and rush into several times. The repulsive force between the predators causes this behaviour. It forces them to keep distance between each other and on the other hand, the preys attract them.

9. Summary and Outlook

One of the advantages of the model used in this paper is its simplicity. There are only few parameters to choose and the swarming is modelled by only using friction and radial forces. The anti-Newtonian force keeps the system in movement despite loss of energy due to friction.
The computation is also not that complex and therefore does not cost a lot of computation time per step, so that it can be done in a reasonable time. However, too much agents added to the simulation will slow down the computation.
The swarming simulation of the model can reproduce the basic behaviour of a swarm when the right parameters are chosen. Some of the simulations approximate the behaviour of swarming observed in nature like a fish swarm

attacked by a predator, although the simulations in the paper are not in the three dimensional Cartesian plane. To do the computations in the 3-dimensional plane could possibly improve the behaviour of the swarm in respect to realistic movement.
The interaction of the preys as a group but also as an autonomous individual is a remarkable result of this model. When the swarm of preys gets split into several parts, all those little swarms try to stay together and eventually connect to another swarm. This can be interpreted as an optimal behaviour of a swarm since it also occurs in nature.
When a prey gets scatted away from the swarm it acts autonomous and tries to go back to the swarm.
Clearly the model is too simple to capture the exact behaviour of swarming observed in nature. To adjust this problem it would help to add more parameters to the model. For example a prey could consider a predator at a certain distance not as a danger or simply add a range of influence to every agent. Randomness or noise, which certainly occurs in nature, could also be added to the model to allow agents making errors in their behaviour. The angle of vision or the direction of the current also occurs in nature, so the predator can sneak up undetected to the swarm from a certain direction or take advantage of the current.
Another disadvantage is that the predators do not have a specific hunt tactic as for example lions do have in nature. The model distinguishes between three forces between the predators, which approximates such behaviour but does not follow an explicit strategy. To generate such behaviour more organism specific computations have to be done. In such a case it would be necessary to first study the behaviour of a certain species from a biological point of view and then develop a specific model.

In spite of everything, the model used in this paper gives a good approximation of basic swarming behaviour. The described simulations and results do reflect roughly the natural behaviour of swarming.
It is hard to analyse what behaviour of preys and what kind of shape is the best for preys attacked by a predator. Certainly the found configurations are close to the optimum since the behaviour described by the simulation is approximately the one observed in nature. We assume that not optimal behaviour would be wiped out by evolution.

Accurate swarming models can also be used in a lot of technical applications. For example for controlling unmanned vehicles for military purposes or controlling robots within the body for killing viruses or cancer. Certainly the models need to be adapted to a certain application but the model described in this paper gives a flavour of how this can be achieved.

10. References

[1] The Euler method, Wikipedia
http://en.wikipedia.org/wiki/Euler_method
[2] The Runge-Kutta method, Wikipedia
http://en.wikipedia.org/wiki/Runge_kutta
[3] Original paper by Vladimir Zhdakin and J. C. Sprott from 2010:
http://sprott.physics.wisc.edu/pubs/paper347.pdf
[4]Further articles considering predator-prey models
http://scholar.google.ch/scholar?q=predator+prey+model&hl=de&as_sdt=0&as_vis=1&oi=scholart&sa=X&ei=ko9vU8CVKqrV0QXy6IGQDA&ved=0CC4QgQMwAA