

The Effect of Fish Schooling on Predation

Louis Dijkstra* Jeroen Hofman[†] Daniel Karavolos[‡]

University of Amsterdam (UvA)

December 26, 2011

Abstract

We extended the model for the formation of fish schools introduced by Hemelrijk et. al. (2008) by adding a predator to the environment. Fish are thought to form schools in order to protect themselves from predators. We tested whether the forming of a school in this model might indeed be beneficial by testing how long it takes for the predator to eat the fish. We found that it takes the predator a longer time to catch all the fish when the fish swim in a random fashion rather than in a school. This result is consistent for different parameter settings. Therefore we conclude that, in this model, the fish do not benefit from swimming in a school. This is most likely caused by unrealistic modeling of the predator.

Keywords: • Fish school • Predation • 3 Dimensional Model

*Student no. VU: 2513226 | E-mail: Louis.Dijkstra@student.uva.nl

[†]Student no. VU: 2513225 | E-mail: Jeroen.Hofman@student.uva.nl

[‡]Student no. VU: 2510702 | E-mail: Daniel.Karavolos@student.uva.nl

Contents

1	Introduction	3
2	Model	3
2.1	Perception Zones	3
2.2	Forces	4
2.2.1	Interaction Forces	5
2.2.2	Individual Forces	6
2.3	Predator	6
3	Implementation	7
3.1	Perception zones	7
3.2	Framework	10
4	Experimental set-up	10
5	Results	12
6	Conclusions and Discussion	13

1 Introduction

There have been many models on flocking behavior of animals. Arguably the most famous of these is Craig Reynolds' boids model [6]. Most of the models focus on flocking patterns in two dimensions, but since the current generation of computers made it possible to handle the computations for these models in 3D, there has been a shift of focus towards these more realistic 3D models.

In 2008 Hemelrijk et al. [1] presented a 3D extension of their 2D model of fish schools. They showed that the individual rules that facilitate the emergent pattern of schooling in fish in 2D can be extended to 3D rules. Furthermore, they showed that these rules not only account for the emergence of schooling, but also for the oblong shape that is found in real fish schools. There is evidence that swimming in schools has certain hydrodynamic advantages [4]. We are, however, more interested whether swimming in a school results in a higher chance of survival [5].

In this report we present a very general model of predator/prey behavior, based on the 3D fish school model of Hemelrijk et al. [1]. There has been some research on the evolution of prey strategies in a similar setting [2]. This differs from our research in the sense that the strategies of our individuals are fixed in each situation and that our prey are able to sense the predator and flee away from it.

The main goal of our research is to investigate whether schooling among fish helps to protect them from predators in this model. We hypothesize that this is the case and we will test this for varying population sizes and predator speeds. In our model there is only one predator, so that we do not have to take cooperative hunting behavior into account.

In the next section we will present the model and in section 3 we will describe our implementation. Section 4 will describe our experimental setup, which is followed by our results in section 5. Finally, we will discuss the results and present our conclusions in section 6.

2 Model

2.1 Perception Zones

Fish in a school react differently to each other depending on the distance between them, hence there is the need for a certain model which describes these different types of perception. In the model we use a perception model of the fish consisting of three different behavioral zones which can also be found in figure 1:

- **Separation zone:** This is a small zone around an individual where it tries to avoid any other individuals in that zone. Individuals use both their vision and their lateral line system so they have a fairly large viewing angle (much broader than humans). Only a small area behind the fish is usually excluded from this zone.
- **Alignment zone:** This is a larger zone than the separation zone (excluding the separation zone itself), in which the individual tries to align with other individuals in that zone, using its lateral line system [1]. Since it depends on the lateral line system of the fish, parts at the back and the front of the fish are excluded from the zone.
- **Cohesion zone:** This is the largest zone, excluding both the separation zone and the alignment zone. Every individual is attracted to the other fish in this zone. The perception in this zone is mere visual, so a portion at the back of the fish is excluded.

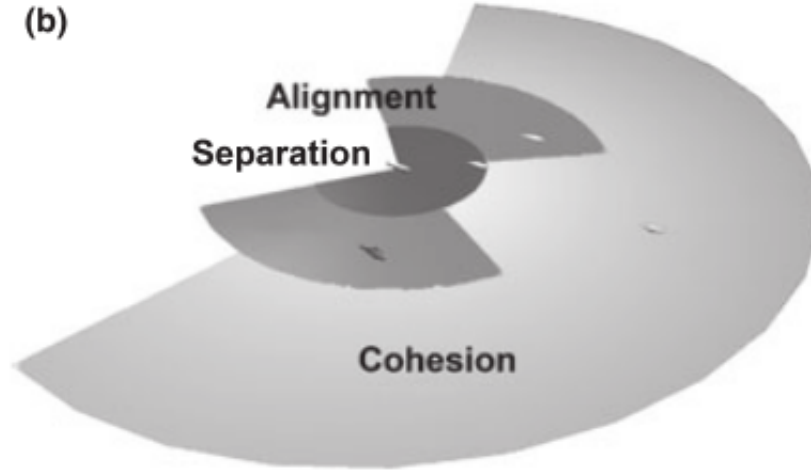


Figure 1: Two dimensional view of the fish perception zones. The figure is taken from [1]. We use another definition for the cohesion zone (see text).

In order to simulate local perception, Hemelrijk et al. proposed to adapt the radii of the alignment and cohesion zone every time step according to the total number of perceivable fish in the associated zones (the radius of the separation zone is fixed and set to R_{\min}). The rationale behind this flexible range of perception is that when the number of neighboring fish in- or decreases, fish, that are further away, are more/less visible.

The adaption of the fish range of perception, R_i , is performed as follows:

$$R_i(t + \Delta t) = \max \{ R_{\min}, R_i(t)(1 - n_i \Delta t (R_{\max} - n_w n(t) + 1)) \} \quad (1)$$

where $R_i(t + \Delta t)$ is the radius of perception at time step $(t + \Delta t)$, $n(t)$ is the number of neighbors in the cohesion zone at time t , n_w indicates the influence of one single neighbor and $0 \leq n_i \leq 1$ controls the smoothness of the radius adaptation, i.e., if n_i is equal to 1, the radius is adapted instantly to the current number of neighbors in the respective perception zone of the fish. When n_i is 0, the radius stays constant.

2.2 Forces

The movement of every fish is governed by seven different forces that can be grouped into two categories: interaction and individual forces. Under interaction forces we understand the separation, alignment, cohesion and flee force, since all four of them take the position and/or orientation of the neighboring fish or predator into consideration:

1. **Separation force, f_s :** in order to avoid collision, fish move away from neighbors in their separation zone.
2. **Alignment force, f_a :** fish align with the neighbors in their alignment zone.
3. **Cohesion force, f_c :** fish are drawn to the center of gravity of the school of fish.
4. **Flee force, f_{flee} :** fish flee from the position of the predator.

The three individual forces are:

5. **Correctional speed force, f_{speed} :** every fish tries to swim at its desired velocity, v_0 .

6. **Correctional pitch force, \mathbf{f}_p :** fish normally do not show large pitch angles (they tend not to swim downward or upward [1]). This force reduces the pitch angle fish might have.
7. **Random force, $\mathbf{f}_{\text{random}}$:** a random force is added to account for imperfect decision-making and noise in the environment.

All these forces together form the net force:

$$\mathbf{f}_{\text{net}} = \mathbf{f}_s + \mathbf{f}_a + \mathbf{f}_c + \mathbf{f}_{\text{flee}} + \mathbf{f}_{\text{speed}} + \mathbf{f}_p + \mathbf{f}_{\text{random}} \quad (2)$$

which is rescaled to 3, if the net force is larger than 3.

We will describe each of the forces in the sections 2.2.1 (interaction forces) and 2.2.2 (individual forces).

2.2.1 Interaction Forces

In order to avoid collision with the fish in the separation zone, every fish experiences a separation force pointed in the opposite direction of the weighted average of the directions of the neighboring fish. In order to compute this force, we compute $\mathbf{r}_j - \mathbf{r}_i$ for every fish j in the separation zone of the fish i (the fish currently under consideration).

The weighted average direction of the neighbors in the separation zone is defined as

$$\mathbf{d}_{s,i} = \frac{1}{n_s} \sum_{j=1}^{n_s} \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|^2}$$

where n_s is the number of fish in the separation zone of fish i and $|\cdot|$ denotes the norm. The separation force is then defined as follows:

$$\mathbf{f}_{s,i} = -w_s \frac{\mathbf{d}_{s,i}}{|\mathbf{d}_{s,i}|} \quad (3)$$

where w_s is a parameter, referred to as the separation weight.

The fish also perceives an alignment force, that steers its orientation towards the average directions of the surrounding fish in its alignment zone. The average orientation is defined as

$$\mathbf{d}_{a,i} = \frac{1}{n_a} \sum_{j=1}^{n_a} \mathbf{e}_{f,j}$$

where n_a is the number of neighbors in the alignment zone of fish i and $\mathbf{e}_{f,j}$ is the forward orientation of fish j . The alignment force then becomes

$$\mathbf{f}_{a,i} = -w_a \frac{\mathbf{d}_{a,i} - \mathbf{e}_{f,i}}{|\mathbf{d}_{a,i} - \mathbf{e}_{f,i}|} \quad (4)$$

where w_a is a parameter, referred to as the alignment weight.

Fish are attracted to the center of mass of the school. We compute

$$\mathbf{d}_{c,i} = \frac{1}{N} \sum_{j=1}^N \frac{\mathbf{r}_j - \mathbf{r}_i}{|\mathbf{r}_j - \mathbf{r}_i|}$$

where N is the total number of fish in the environment. We define the cohesion force, \mathbf{f}_c , as

$$\mathbf{f}_c = w_c \frac{\mathbf{d}_{c,i}}{|\mathbf{d}_{c,i}|} \quad (5)$$

where w_c is a parameter, referred to as the cohesion weight.

The cohesion force is, in contrast to the original model introduced by Hemelrijk et al. (2008), not locally defined, in the sense that in our implementation of this force, the positions of all fish in the environment are taken into account. This is of course not so realistic, but makes sure that, when under attack by a predator, the school of fish does not suddenly fall apart into smaller groups, which was the case when we used the same definition as Hemelrijk et al. (2008).

The fourth and last interaction force, is the result of interaction between the fish and the predator. As expected, the fish try to get away as far as possible from the predators position, $\mathbf{r}_{\text{predator}}$, at all cost, while the predator tries to devour as much fish as it possibly can.

The flee force for fish i is defined as

$$\mathbf{f}_{\text{flee},i} = -w_{\text{flee}} \frac{\mathbf{r}_{\text{predator}} - \mathbf{r}_i}{|\mathbf{r}_{\text{predator}} - \mathbf{r}_i|} \quad (6)$$

where w_{flee} is a parameter, $\mathbf{r}_{\text{predator}}$ is the current position of the predator and \mathbf{r}_i is the position of fish i . In words, the closer the fish get to the predator, the larger the flee force becomes.

Note that the magnitude of all these interaction forces are governed by their associated parameters (w_s , w_a , w_c and w_{flee}) and, hence, these parameters denote the relative influence on the net force (see equation 2).

2.2.2 Individual Forces

In addition to interaction forces, every fish is influenced by three other individual forces: a correctional speed force, $\mathbf{f}_{\text{speed},i}$, a correctional pitch force, $\mathbf{f}_{p,i}$, and a random force, $\mathbf{f}_{\text{random}}$.

The correctional speed force, $\mathbf{f}_{\text{speed},i}$, forces the fish to return to its desired/default velocity, v_0 . The force is defined as

$$\mathbf{f}_{\text{speed},i} = \frac{v_0 - |\mathbf{v}_i|}{\tau} \mathbf{e}_{f,i} \quad (7)$$

where v_0 is the desired velocity, $|\mathbf{v}_i|$ is the magnitude of the velocity vector, τ is the 'relaxation time' introduced to make sure the speed of a fish is not updated instantly and $\mathbf{e}_{f,i}$ is the forward direction of fish i .

The correctional pitch force returns fish to their natural horizontal position. The force is defined as

$$\mathbf{f}_{p,i} = \begin{pmatrix} 0 \\ 0 \\ -w_{pc} \cdot e_z \end{pmatrix} \quad (8)$$

where e_z is the z -element of the forward direction vector $\mathbf{e}_{f,i}$ and w_{pc} is a parameter, referred to as the correctional pitch weight.

We also introduce a random force in order to account for imperfect decision-making and a noisy environment. The random force, $\mathbf{f}_{\text{random},i}$, is a randomly generated vector with a maximum predefined magnitude.

2.3 Predator

The predator is defined by its position in the three dimensional environment, $\mathbf{r}_{\text{predator}}$, the speed of the predator, v_{predator} , and, what we refer to as, the relaxation time, t_{relax} . This relaxation time denotes the interval the predator needs after it caught a fish, before it can catch another one.

The behavior of the predator is simple; it moves with a constant speed to the nearest fish in its vicinity. Which fish is the closest is determined every single iteration step. This might cause the predator to be 'confused' (i.e., it might have to change its direction often). Which fish is closest to the predators position is determined by

$$\mathbf{r}_{\text{closest}} = \min_i |\mathbf{r}_{\text{predator}} - \mathbf{r}_i|$$

where \mathbf{r}_i is the current position of fish i . The position of the predator is adapted as follows

$$\mathbf{r}_{\text{predator}}(t + \Delta t) = \mathbf{r}_{\text{predator}}(t) + v_{\text{predator}} \Delta t \frac{\mathbf{r}_{\text{closest}} - \mathbf{r}_{\text{predator}}(t)}{|\mathbf{r}_{\text{closest}} - \mathbf{r}_{\text{predator}}(t)|} \quad (9)$$

When a fish is in the circle with a certain predefined radius, R_{eat} , around the predator, it is considered eaten. We remove the fish from the environment. The predator cannot devour any more fish until the end of its predefined relaxation time, t_{relax} .

3 Implementation

3.1 Perception zones

For the implementation of the perception zones we have to create an algorithm with which we can decide whether a fish is in one of the perception zones of some individual, described in the section 2. As explained before there are three perception zones of the fish: the cohesion zone, the alignment zone and the separation zone. The separation and alignment zone are implemented as in [1]. As mentioned before, we use an adaptation of the cohesion zone, i.e., the cohesion zone is the entire environment. In the model we have to check for every fish how many neighbors there are in these zones in order to compute the forces in equations 3, 4 and 5. The three different zones for a fish are shown in figure 1. The figure shows three circles where parts are cut out to accommodate the fact that fish cannot see certain parts behind them (in the case of the separation zone) and that alignment only takes place with fish which are in the region perpendicular to the forward direction of the individual considered. While this figure looks relatively simple, there are two properties which make this perception model complicated:

- Figure 1 shown below is two-dimensional, while our model is in three dimensions, so one has to imagine the forms to be spherical instead of circular, which complicates the part of cutting out regions of the spheres where the fish cannot perceive.
- The whole structure rotates in accordance to the forward direction of the fish.

We developed an algorithm to determine whether a fish is in the three-dimensional perception zones of an individual, by only taking into account the current forward direction of the individual. Figure 2 below shows all the computed quantities graphically. The algorithm consists of the following steps:

- Consider a fish i at position \mathbf{x}_i together with a forward direction \mathbf{e}_f of the fish. First we check for every fish not equal to fish i in the school whether or not this fish is contained in a sphere with radius R_{min} or R_a for the separation and alignment perception zones, respectively. If the fish is in one of the perception radii, it might be in the corresponding perception zone of fish i if it is not in an excluded part of the sphere, shown in figure 1. The next steps will be about determining this which regions are excluded.
- Start with the forward direction \mathbf{e}_f of fish i and determine a vector \mathbf{e}_s perpendicular to \mathbf{e}_f with the restriction that \mathbf{e}_s should have a zero z -component. \mathbf{e}_s is the side-ward direction of the fish. Note that by setting the z -component to zero it is implied there is no roll force and \mathbf{e}_s is unique up to a sign.

- Compute the cross product of \mathbf{e}_f with \mathbf{e}_s to compute a vector \mathbf{e}_z , perpendicular to both vectors. This vector is the normal vector of the plane of movement in which fish i is located. The equation of this plane is defined by all points (x, y, z) in \mathbb{R}^3 for which $\mathbf{e}_z \cdot (x, y, z) = 0$.
- Next we rotate the forward vector \mathbf{e}_f around \mathbf{e}_z with some angle θ , dependent on which part of sphere we want to cut out. For instance for the separation zone we want to rotate \mathbf{e}_f with $\theta = 2\pi/3$ and $\theta = 4\pi/3$ to obtain two vectors which mark the boundary of the part of the sphere with the separation radius which we want to exclude (these angles are given in table 1). We do this rotation by using the standard rotation matrix obtained from [7]. In the case of the alignment zone we need four of these vectors, but the two vectors on the front side of the fish are the opposite vectors of the two vectors on the backside.
- Next we take the cross product of these two or four boundary vectors with \mathbf{e}_z , to get the normal vector for two or four planes defined by one of the boundary vectors and \mathbf{e}_z . Call these normal vectors \mathbf{y}_k for $k = 1, 2$ for separation or $k = 1, 2, 3, 4$ for alignment.
- Finally we look at the position of some fish j with position \mathbf{x}_j which is in one of the perception radii (checked in the first step) and check whether its position relative to the position of fish i is in one of the excluded regions. We compute:

$$(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{y}_1 < 0 \text{ and } (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{y}_2 < 0 \quad (10)$$

If both conditions are met fish j is in the excluded region and so should not be included in the force calculations. In the case of alignment the same condition with \mathbf{y}_3 and \mathbf{y}_4 also needs to be checked.

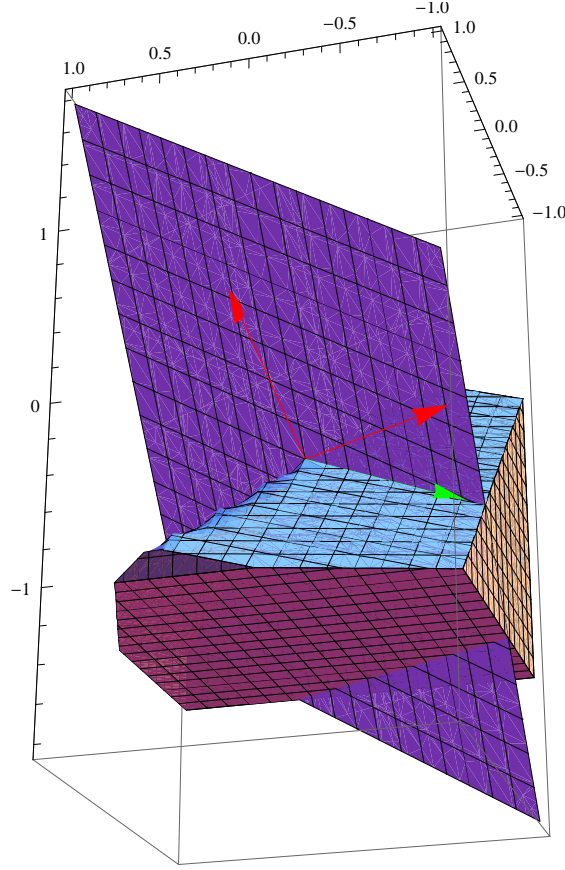


Figure 2: Graphical representation of the computed quantities in the steps above. The two red vectors represent \mathbf{e}_f and \mathbf{e}_s . The black vector (not clearly visible here) is the normal vector \mathbf{e}_z of the plane going through these two vectors. The green vectors are rotations of \mathbf{e}_f around this normal vector. The region beneath the light blue planes, which are the planes defined by \mathbf{e}_z and a green vector, gives the final excluded region. Note that the normals of the planes going through \mathbf{e}_z and a green vector are not drawn.

Fish are assigned random positions in a sphere with a radius of 50 and the origin as its center. The initial orientations are randomly generated. We start each time step by applying the previous algorithm to all fish in the school. This will give us the forces on each fish using the force equations from the previous chapter. Then we combine them to compute a net force on the fish according to equation 2. This net force is then converted to a movement by first calculating the acceleration of the fish:

$$\mathbf{a} = \mathbf{f}_{\text{net}}/m \quad (11)$$

and then calculating the new velocity of the fish at the next time step:

$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \mathbf{a}\Delta t \quad (12)$$

We can then use this velocity to update the position of fish i :

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \frac{\Delta t}{2} (\mathbf{v}(t) + \mathbf{v}(t + \Delta t)) \quad (13)$$

where we used a simple quadrature rule for estimating the new position. Every time step we set the forward direction \mathbf{e}_f equal to $\mathbf{v}(t + \Delta t)$ and hence we assume that the fish always head in the direction of movement.

The initial position of the predator is chosen randomly. The predator always starts within a 200 point distance of the origin. Every time step we also update the predator position, by calculating which of the fish is the closest one and updating the predators position in the direction of that fish. If the fish comes within the predators radius (R_{eat}) the fish will be eaten and removed from the environment. After eating the predator has a relaxation time during which it can hunt the next closest fish but it cannot eat it. After the relaxation time the predator will be able to eat the fish again.

3.2 Framework

We have implemented the above mentioned model in Java, using the MASON Toolkit (v.16) [3] for visualization. Generally it is good practice to separate the model from the visualization and the structure of MASON facilitates that, but it still requires the user to work exactly according to the rules of the framework. Therefore, we choose to implement our model separately from MASON and integrate them afterwards. This means that we can run simulations without having to use MASON.

Without MASON our main creates a `Model`, which contains one predator, called `predator`, and a `School` of `Fish`. The `Model` keeps track of time and calls the `update()` of `School`. The school then updates the positions of all the fish in the school by computing the forces and calling the `update()` of each individual fish.

The visualization of the model (as defined in `FishSchoolWithUI`) is quite simplistic. The environment is an infinite continuous blue space, in which each fish is represented as a green cone (top up) and the predator is a red cone (top sideways), see figure 3. We have added a wire frame as a landmark for the viewer. To update the visualization, MASON needs a separate model of the system. The main class should extend the built-in Mason-class `SimState`, which adds individuals to the environment. And each individual implements the built-in Mason-class `Steppable`. The individuals are then updated (in terms of behavior) through MASONs `Scheduler` by calling the overridden `Step()`-function.

To visualize our simulation, we must synchronize the update-loop of our model with the built-in update-loop (the scheduler) of MASON. Since we shield our implementation from MASON, we do not need the scheduler to update any behavior. We only need it to update the positions of our individuals in the environment. Therefore, the `step()` of our fish individuals (`Fish3D`) only consists of checking whether the fish is still alive and retrieving the updated position from the corresponding fish in the model or, when the fish is dead, remove itself from the environment. We have created a `Steppable SchoolUpdater`, to update the whole school of fish. This class has a `Model`, of which the `performOneTimeStep()` is called in each `step()` of the `SchoolUpdater`. After that, it calls the `step()` of each `Fish3D` in the environment.

Similar to the `Fish3D`, the `predator3D` only retrieves the position of the predator in the model. It is updated separately from the fish school, but it is always scheduled after the `SchoolUpdater` so that it always retrieves an updated position from the same time step.

4 Experimental set-up

This model depends on many parameters. We partly use parameter values from [1] and partly tweaked the parameters ourselves. Here is a short overview of the parameters we used in our model:

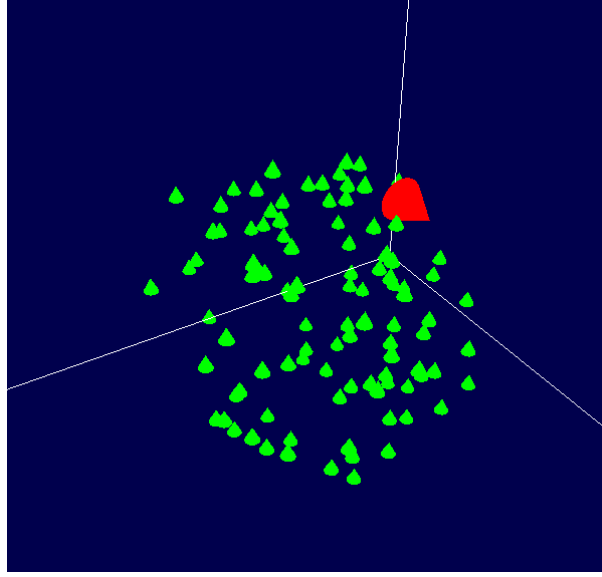


Figure 3: Visualization of the model. The red cone represents the predator, the green cones are the fish.

Table 1: Model parameters.

Parameter	Symbol	Values
Number of individuals	N	$\{2, 7, \dots, 72, 77\}$
Body mass	-	1
Time step	Δt	0.05
Zone of separation		
Radius	R_{\min}	2
Blind angle back	-	60
Zone of alignment		
Maximum radius	-	5
Blind angle back	-	60
Blind angle front	-	60
Zone of cohesion		
Maximum radius	-	15
Blind angle back	-	90
Cruise speed	v_0	2
Weights		
Separation	w_s	10
Alignment	w_a	5
Cohesion	w_c	9
Flee force	w_{flee}	12
Relaxation time	τ	0.2
Pitch control	w_{pc}	2
Random force	$ f_{\text{random}} $	0.5
Max. force	f_{max}	3
Neighbor influence	n_w	1
Smoothness radius adaption	n_i	0.01
Predator		
Eating radius	R_{eat}	2.0
Relaxation time	t_{relax}	1.0, 5.0
Speed	v_{predator}	10, 25

We allow certain parameters to change in order to test our hypothesis (see table 1). We vary the number of fish from 2 up to 77, the relaxation time of the predator (1.0 and 5.0) and the speed of the predator (10 and 25). We look at two different cases: one, where the fish swim in a school and one, where the fish wander around randomly. The latter is achieved by setting both the alignment and the cohesion force to zero (the separation force is still taking into account, because the fish should not collide). We will look in both cases how long it takes the predator to devour all the fish. We repeat each experiment 10 times for each fish size to make an error estimate with a 95% confidence interval defined in the usual way:

$$\text{error} = \frac{1.96\sigma}{\sqrt{10}} \quad (14)$$

where σ is the sample standard deviation.

5 Results

See figure 4 for the results of our simulations. As you can see, the time the predator needs for catching all the fish is overall larger when the fish swim in a random fashion rather than in schools. This result is consistent for all the parameter settings used.

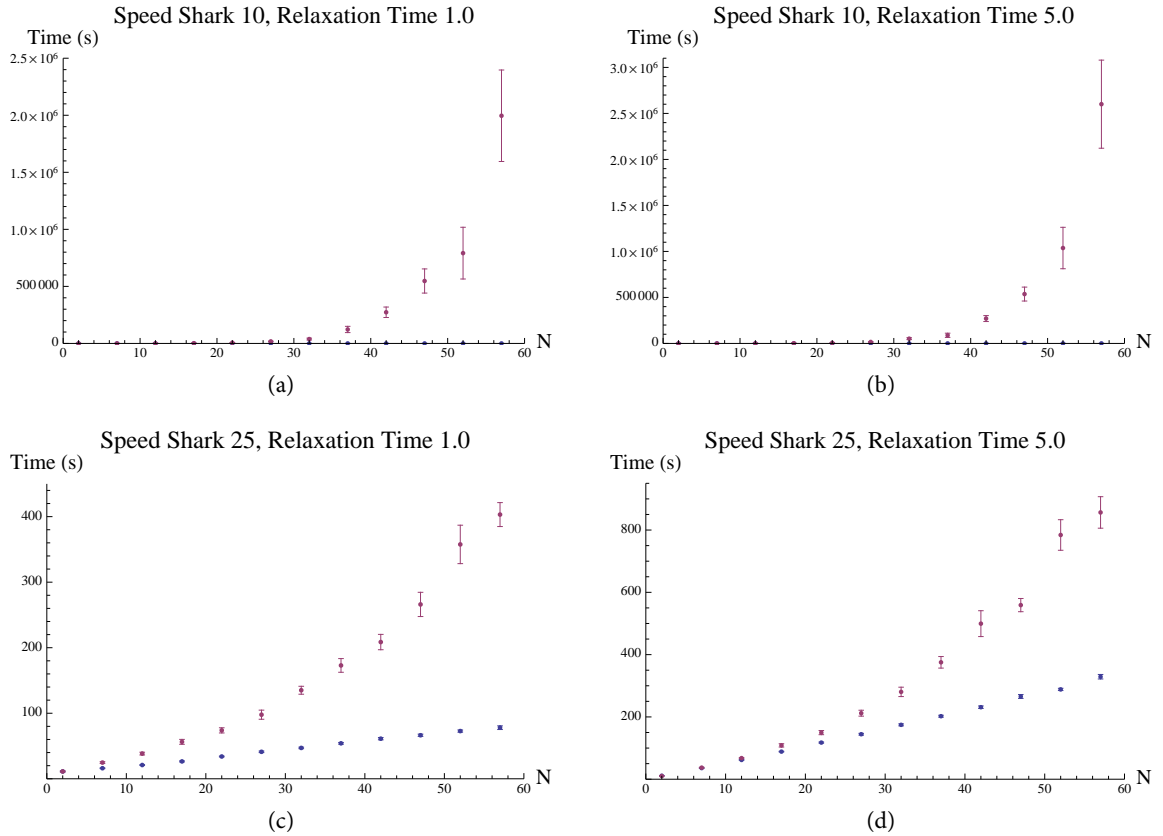


Figure 4: The experimental results. The x -axis represent the number of fish in the environment (N). The y -axis denotes the time the predator required to catch all the fish. The dots represent the mean value and the bars represent the 95% confidence interval. Blue denotes the original model, where fish swim in schools, while purple denotes the situation where fish swim randomly.

6 Conclusions and Discussion

We extended the model for the formation of fish schools introduced by Hemelrijk et. al. (2008) by adding a predator to the environment. Fish are thought to form schools in order to protect themselves from predators [1]. We tested whether the forming of a school in this model might indeed be beneficial. We found that it takes the predator a longer time to catch all the fish, when the fish swim in a random fashion rather than in a school. This result is consistent for different parameter settings (see section 5). Therefore, we conclude that, in this model, the fish do not benefit from swimming in a school.

There are some remarks to be made about the way the model is set up in this report. First of all the perception radius of the fish in the case of coherence, is set to infinity in our implementation. Hemelrijk et al. [1] used a finite perception radius to account for the limited range of vision of the fish in the school. As discussed previously we implemented it in this way because otherwise the fish school would fall apart easily when the predator is present, which would make our hypothesis untestable, since we need something that keeps resembling a stable school, even when a predator is nearby. Moreover, our predator is not very realistic. The predator turns instantly, because the forward direction of the predator is directly updated as being the same as the direction to the nearest fish. The same essentially happens for the update of the fish position, but because the fish position is governed by forces the change is very gradually, because the forces at time $t + \Delta t$ are highly dependent on the forces at time t . Also the predator is assumed to have a spherical area around it, with a fish being eaten when it is in that area. This is not very realistic. An improvement would for instance be a small cone in front of the predator, indicating the eating zone. Also the predator does not rest when it is eating a fish, but rather instantaneously starts hunting the next one, which is not very realistic as well. In short, there is a lot of room for improvement on the predator. There are also some comments to be made about the interaction between the predator and the fish. Our hypothesis was based upon the idea that the predator would get confused by all the fish in its neighborhood and hence constantly change directions, since the nearest fish changes constantly. This is however not the observed behavior in the model. An explanation is that possibly the ratio between the average distance between fish in the school and the eating radius of the predator is too small, which makes this change in direction occur less or not at all because this switching between nearest fish only occurs when the predator is close to the fish school. Also the fish school might be too static, which has the effect that the fish closest to the predator is always the same fish. Also, when the predator comes close to the fish school in our model, the dominant force becomes the fleeing force and this causes the school to become disorganized, which makes this single target chasing also more profound. The amount of parameters used in this model is very large and the model is very sensitive to these parameters. We partly used parameters from Hemelrijk et al. [1] and partly tweaked the parameters ourselves. However it is highly likely we did not find the optimal parameter set and that there exist some parameter set which would make the school more stable and both the school and the predator behave more realistically. Lastly, also the fact that we worked with an unbounded box instead of a tank might make a big difference, because in the case of random fish movement some fish are moving away quickly from their spawning point, and when the predator has for instance killed all the fish moving to the east of the origin, it has to swim back all the way to the west in order to chase the other fish (and which are by now very far away from the initial position). This causes our simulations for the random fish movement to run very slowly, since it takes a very long time for the predator to cover all these distances.

References

- [1] C.K. Hemelrijk, H. Hildenbrandt, Self-organized shape and frontal density of fish schools, 2008, *Ethology*, Vol. 114, 245-254
- [2] H. Kunz, T. Züblin & C.K. Hemelrijk, On prey grouping and predator confusion in artificial fish schools, 2006, *Proceedings of Tenth International Conference on Artificial Life*, 365-371

- [3] S. Luke et al., MASON: A Multi-Agent Simulation Environment. 2005, Simulation: Transactions of the society for Modeling and Simulation International, Vol. 82(7), 517-527
- [4] J.C. Svendsen et al., Intra-school positional preference and reduced tail beat frequency in trailing positions in schooling roach under experimental conditions. , 2003, Journal Of Fish Biology, Vol. 62, 834-846
- [5] S. R. Neill and J. M. Cullen, Experiments on whether schooling by their prey affects the hunting behavior of cephalopods and fish predators, 1974, Journal of Zoology, Vol 172, 549-569
- [6] C.W. Reynolds, 1987, SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, Vol. 21(4), 25-34
- [7] http://en.wikipedia.org/wiki/Rotation_matrix#Rotation_matrix_from_axis_and_angle