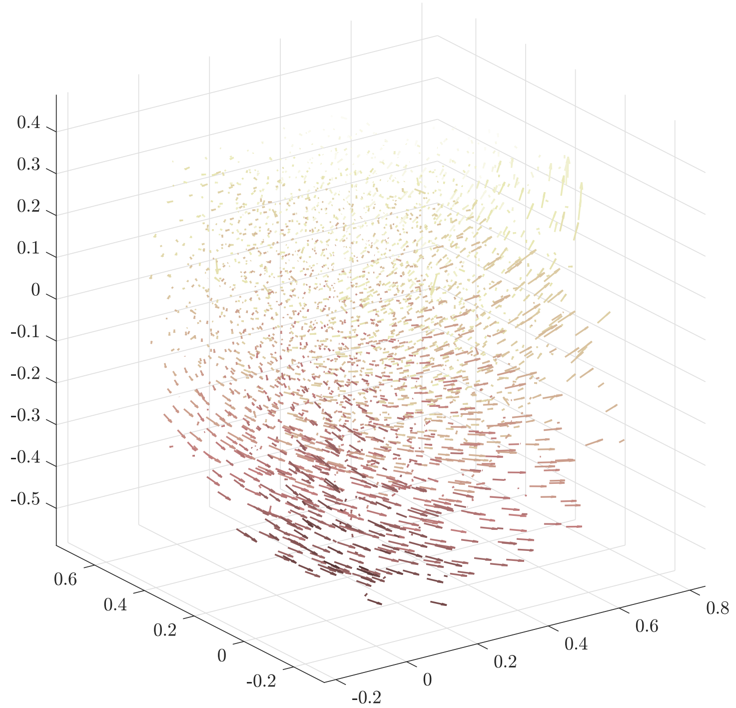

Figure 1: Snapshot of a 3D flocking simulation

Boids Swarm Simulation

Motivation

Starlings or mackerels are only two examples for species that classically spend their whole lives in the protection of a so-called swarm. This formation is very well known to give shelter from possible predators as they have difficulties focusing their attack on single, weaker individuals. It is also beneficial from a psychological point of view.

For large swarms ($\gg 1000$) fantastic lava-lamp-like "murmurings" (in case of birds also called flocking) can be observed when investigating the swarm as a whole. Corresponding videos can quickly be found on the internet. Though the swarm as a whole is defined by its parts, this so-called "swarm" behaviour cannot be attributed to any single individuals in the swarm, but results from their interactions.

Model

In 1986 American artificial-life expert Craig Reynolds published a model to demonstrate that though the individuals – he called them "Boids" – follow very simple rules the swarm as a whole might behave very complex.

Reynold's model is best described as an agent-based model, wherein each agent depicts one Boid. We define this model via its initialisation phase and its time-update.

Initialisation:

- Let N be a sufficiently large natural number. Create that many Boids $i \in \{1, \ldots, N\}$ and assign them a uniformly distributed initial position $\vec{x}_i(0)$ on $[-1, 1]^3$ (a symmetric cube around $[0, 0, 0]$ with length 2).
- Moreover, every Boid $i \in \{1, \ldots, N\}$ is assigned the initial velocity $\vec{v}_i(0) := [0, 0, 0]^T$.

Moreover the model is updated in time-steps.

Dynamics:

- Each time-step t, each Boid is addressed once. Hereby it will update its velocity and based on that, finally, its position.

- The velocity will be updated based on three (four) rules. Each of them will result in a contribution $\vec{w}_i, i = 1 \ldots, 4$ to its new velocity:

  1. The Boid regards all Boids in its so-called observation-radius d_o and calculates the average position of its observed neighbours. The Boid will furthermore tend towards this point

  $$\vec{w}_1 := \frac{1}{|\{j : |\vec{x}_i(t) - \vec{x}_j(t)| \leq d_o\}|} \left(\sum_{j:|\vec{x}_i(t)-\vec{x}_j(t)|\leq d_o} \vec{x}_j(t) \right) - \vec{x}_i(t). \tag{1}$$

  2. The Boid regards all Boids in its observation radius d_o and calculates the average velocity of its neighbours. The Boid will furthermore tend to swim into the same direction as its neighbours.

  $$\vec{w}_2 := \frac{1}{|\{j : |\vec{x}_i(t) - \vec{x}_j(t)| \leq d_o\}|} \left(\sum_{j:|\vec{x}_i(t)-\vec{x}_j(t)|\leq d_o} \vec{v}_j(t) \right). \tag{2}$$

  3. The Boid regards all Boids in its so-called collision radius $d_c < d_o$ and calculates the average distance vector to them. The Boid will furthermore tend away from these neighboured Boids as they are too close.

  $$\vec{w}_3 := -\frac{1}{|\{j : |\vec{x}_i(t) - \vec{x}_j(t)| \leq d_c\}|} \left(\sum_{j:|\vec{x}_i(t)-\vec{x}_j(t)|\leq d_c} \vec{x}_j(t) - \vec{x}_i(t) \right). \tag{3}$$

  4. The Boids additionally feel comfortable in the cube. Therefore, an additional velocity component is defined to retrieve them, when outside this area.

  $$\vec{w}_4(t) := \begin{cases} 0, & \max(abs(\vec{x})) \leq 1 \\ -x(t), & \text{else.} \end{cases} \tag{4}$$

  Note that the this contribution to the new velocity has nothing to do with the interaction of the fish, but is necessary to keep the swarm in sight.

  Finally all components form the new velocity via

  $$\vec{v}(t+1) = \vec{v}(t)l_0 + \vec{w}_1(t)l_1 + \vec{w}_2(t)l_2 + \vec{w}_3(t)l_3 + \vec{w}_4(t)l_4, \tag{5}$$

  with positive weights l_i. If the total velocity $|\vec{v}_i(t)|$ is larger than a defined maximum v_{max}, use

  $$\vec{v}(t+1) := v_{max} \frac{\vec{v}(t+1)}{|\vec{v}(t+1)|} \tag{6}$$

  to cut it to the maximum value.

- After all new velocities of all Boids have been calculated, update all Boids' position via

  $$\vec{x}_i(t+1) := \vec{x}_i(t) + \vec{v}_i(t+1). \tag{7}$$

ToDo's

Task 1

Implement the model. Use

$$N = 500,\ v_{max} := 0.03,\ d_o := 0.02,\ d_c := 0.01,$$

$$l_0 := 0.31,\ l_1 := 0.001,\ l_2 := 1.2,\ l_3 := 2,\ l_4 := 0.01.$$

for an initial parametrisation of the model. Adapt these parameters, if better results can be obtained.

Task 2

Look for a performant way to 3D-visualise the results of the model in run-time. The Boids should change their appearance (colour) according to their position and direction in the environment (e.g. by shading, reflection,...). Always take care that the visualisation does not slow down the simulation too much.

Task 3

Alter the parameters (apart from N) and describe their influence on the flocking behaviour.

Task 4

Improve your coding! Try to find the maximum number of fish N that can be simulated and visualised on run-time. Look for performance-tricks on the internet.

Task 5

Document your findings in a written protocol.

Programming Language

Any fundamental programming language that supports 3D rendering (e.g. Python, MATLAB, Java, ...).