



# SYNC or Swim

## A Particle Model of the Interaction within Fish Schools

David Ebert and Mikaela Jordan

Tarleton State University

March 31, 2017



The coordination of events to operate a system in unison.  
Some natural physical examples:

- Circadian Rhythms
- Round of Applause (WHAT?!?! - Let's try it!)

# Example - Human Grouping



One object influencing another by providing feedback.  
Real life examples

- Animal Swarming



One object influencing another by providing feedback.  
Real life examples

- Human Imitation (Memes/Trends)





- The coordinated behavior of animals of the same species and the emergent properties that arise.



- The coordinated behavior of animals of the same species and the emergent properties that arise.
- For mathematical purposes, consider a swarm as an emergent behavior with no central coordination that arises due to several simple instinctual rule that animals of a given species follow.



- The coordinated behavior of animals of the same species and the emergent properties that arise.
- For mathematical purposes, consider a swarm as an emergent behavior with no central coordination that arises due to several simple instinctual rule that animals of a given species follow.
- Other terms we will be using interchangeably with “collective behavior”: swarm, school(specific to fish), aggregate



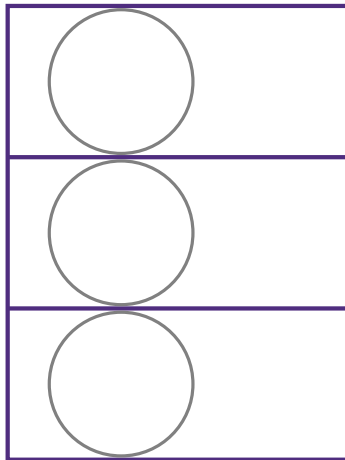


- Learning C/CUDA
- Applying mathematical models to real life phenomenon
- How will environmental factors affect the animal aggregate
- How animal aggregates will affect the environment



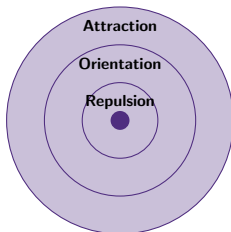
Our model represents each fish adhering to the following three rules:

- **Alignment**
- **Cohesion**
- **Separation**





- Lagrangian Algorithm
- Metric distance model



# Calculating Forces, Velocities, and Positions



At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

# Calculating Forces, Velocities, and Positions



At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

- Calculate distance between particle  $i$  and every other particle.

# Calculating Forces, Velocities, and Positions



At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

- Calculate distance between particle  $i$  and every other particle.
- If the distance between some particle  $j$  and particle  $i$  is smaller than predefined search distance, then use Hooke's law to determine the forces between particle  $j$  and particle  $i$ .

# Calculating Forces, Velocities, and Positions



At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

- Calculate distance between particle  $i$  and every other particle.
- If the distance between some particle  $j$  and particle  $i$  is smaller than predefined search distance, then use Hooke's law to determine the forces between particle  $j$  and particle  $i$ .
- Use force calculated above to update particle  $i$ 's velocity as follows:

# Calculating Forces, Velocities, and Positions



At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

- Calculate distance between particle  $i$  and every other particle.
- If the distance between some particle  $j$  and particle  $i$  is smaller than predefined search distance, then use Hooke's law to determine the forces between particle  $j$  and particle  $i$ .
- Use force calculated above to update particle  $i$ 's velocity as follows:

$$v_i = v_i + a_i dt$$





At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

- Calculate distance between particle  $i$  and every other particle.
- If the distance between some particle  $j$  and particle  $i$  is smaller than predefined search distance, then use Hooke's law to determine the forces between particle  $j$  and particle  $i$ .
- Use force calculated above to update particle  $i$ 's velocity as follows:

$$v_i = v_i + a_i dt$$

- And update particle  $i$ 's position using:

# Calculating Forces, Velocities, and Positions



At every timestep, the following calculations occur for each particle (let's call it particle  $i$ ):

- Calculate distance between particle  $i$  and every other particle.
- If the distance between some particle  $j$  and particle  $i$  is smaller than predefined search distance, then use Hooke's law to determine the forces between particle  $j$  and particle  $i$ .
- Use force calculated above to update particle  $i$ 's velocity as follows:

$$v_i = v_i + a_i dt$$

- And update particle  $i$ 's position using:

$$p_i = p_i + v_i dt$$



# Where Do We Go From Here?



- Add initial conditions for species-specific parameters
  - Density of swarms, how they behave towards targets and obstacles, etc.
- Move calculations from CPU to GPU to speed up calculation time



- Barbaro, Alethea, Bjorn Birnir, and Kirk Taylor. *Simulating the Collective Behavior of Schooling Fish With a Discrete Stochastic Model*. University of Iceland. 2006. Web.
- Bernoff, Andrew J. "Synchronization and Swarming: Clocks and Flocks." Harvey Mudd College.
- Morale, Daniela, Vincenzo Capasso, and Karl Oelschläger. "An Interacting Particle System Modelling Aggregation Behavior: From Individuals to Populations". *Journal of Mathematical Biology*. 2004. Web.
- Parrish, Julia K., Steven V. Viscido, and Daniel Grunbaum. "Self-Organized Fish Schools: An Examination of Emergent Properties". *The Biological Bulletin* 202. 2002:296-305. Web.
- Schellinck, Jen, and Tony White. "A Review of Attraction and Repulsion Models of Aggregation: Methods, Findings, and a Discussion of Model Validation". *Ecological Modelling* 222. 2011: 1897-1911. Web.



Thank you to Dr. Wyatt and the Particle Modelling Lab for their time and resources.

## QUESTIONS?