

Modelling Complex Systems

Self-propelled particles

This lecture is adapted from Vicsek, T. & Zafeiris, A. (2012)
Collective Motion. And slides of David Sumpter

See: [arXiv:1010.5017v2](https://arxiv.org/abs/1010.5017v2)





Why do animals move together?

- Increased accuracy (many estimates)
- Increased awareness (many eyes)
- Confuse predators and reduce encounters

How do animals move together?

- Group formation usually seems to be *spontaneous*.
- Based on local interactions
- Phenomenological models
- Can ignore 'first principles' physics!
e.g. Conservation of momentum
- Use biological principles and limits instead.

Random walk in one dimension

- Run 'RandomWalk1D'

Diagram illustrating the equations for a 1D random walk, with labels pointing to the variables:

$$x_i(t+1) = x_i(t) + v_0 u_i(t)$$
$$u_i(t+1) = a u_i(t) + e_i(t)$$

Labels and their corresponding variables:

- future position: $x_i(t+1)$
- current position: $x_i(t)$
- current velocity: $u_i(t)$
- future velocity: $u_i(t+1)$
- current velocity: $u_i(t)$
- stochastic effect: $e_i(t)$

$e_i(t)$ is a random number selected uniformly at random from a range $[-\eta/2, \eta/2]$

Attraction in one dimension

- Run 'Aggregate1D'

Diagram illustrating the variables in the equations:

- $x_i(t+1)$ is labeled "future position".
- $x_i(t)$ is labeled "current position".
- v_0 is labeled "current velocity".
- $u_i(t)$ is labeled "current velocity".
- $u_i(t+1)$ is labeled "future velocity".
- $au_i(t)$ is labeled "current velocity".
- $(1-a)s_i(t)$ is labeled "Direction to most neighbours".
- $e_i(t)$ is labeled "stochastic effect".

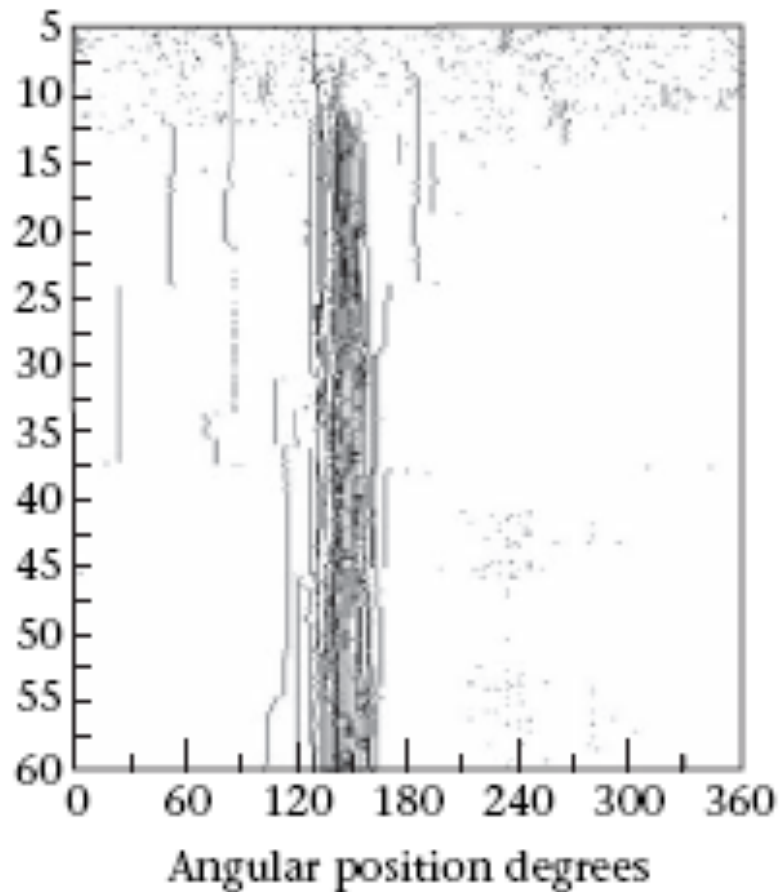
$$x_i(t+1) = x_i(t) + v_0 u_i(t)$$
$$u_i(t+1) = au_i(t) + (1-a)s_i(t) + e_i(t)$$

$$s_i(t) = \frac{1}{|R_i|} \sum_{j \in R_i} \text{sign}(x_i(t) - x_j(t))$$

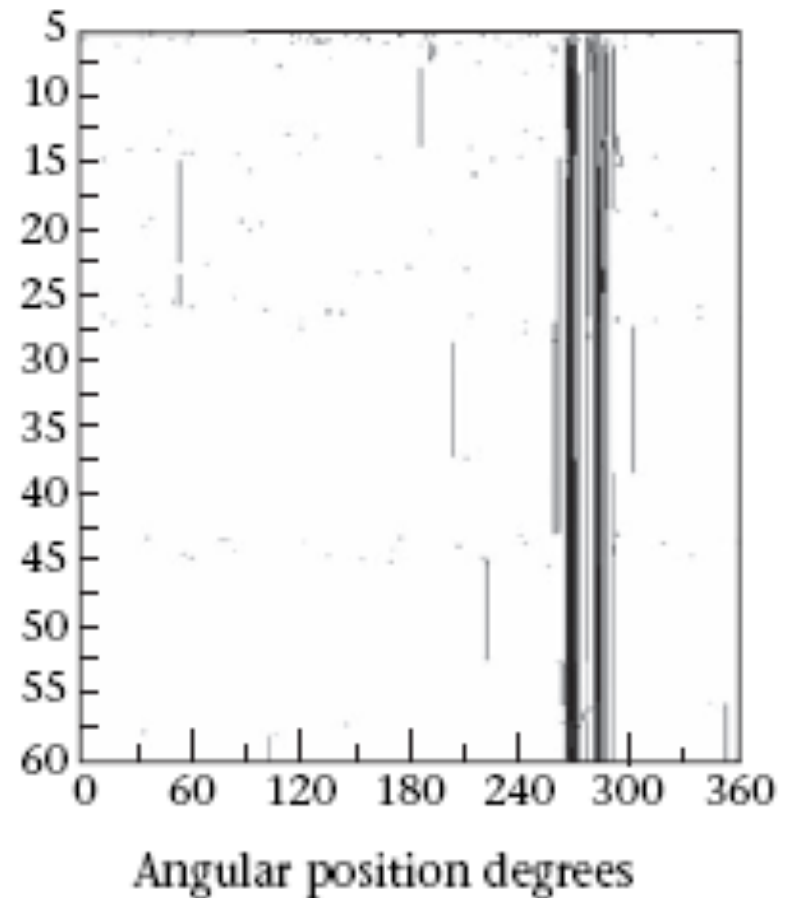
$e_i(t)$ is a random number selected uniformly at random from a range $[-\eta/2, \eta/2]$

Cockroach aggregation

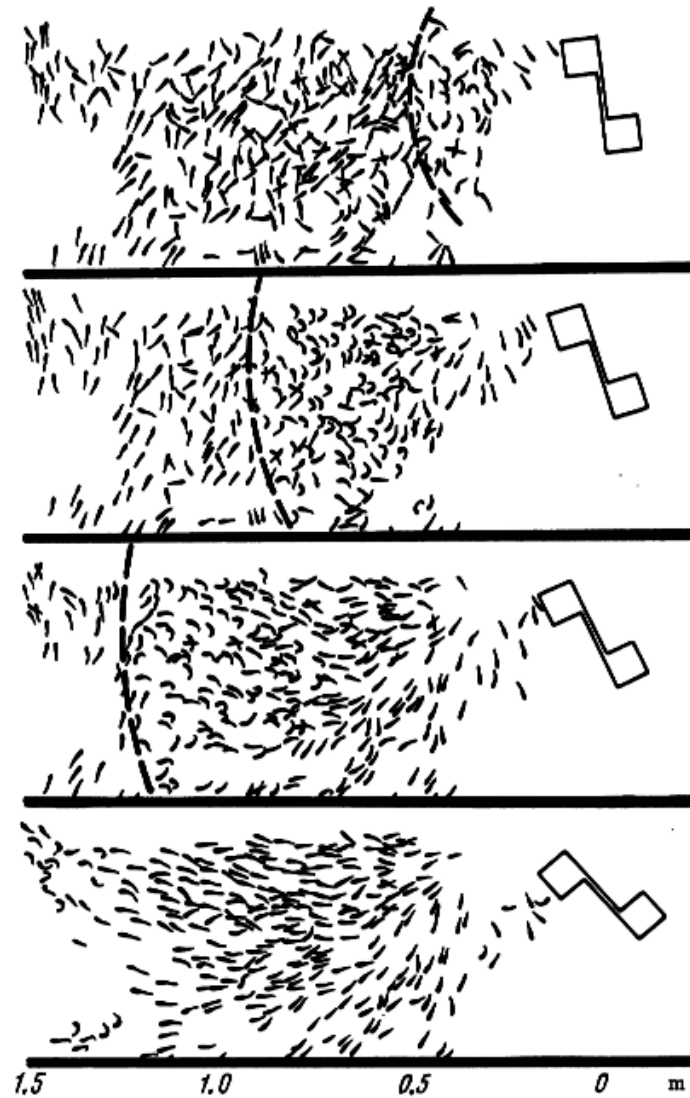
Cockroaches



Model



Radakov's fish



Alignment model in one dimension

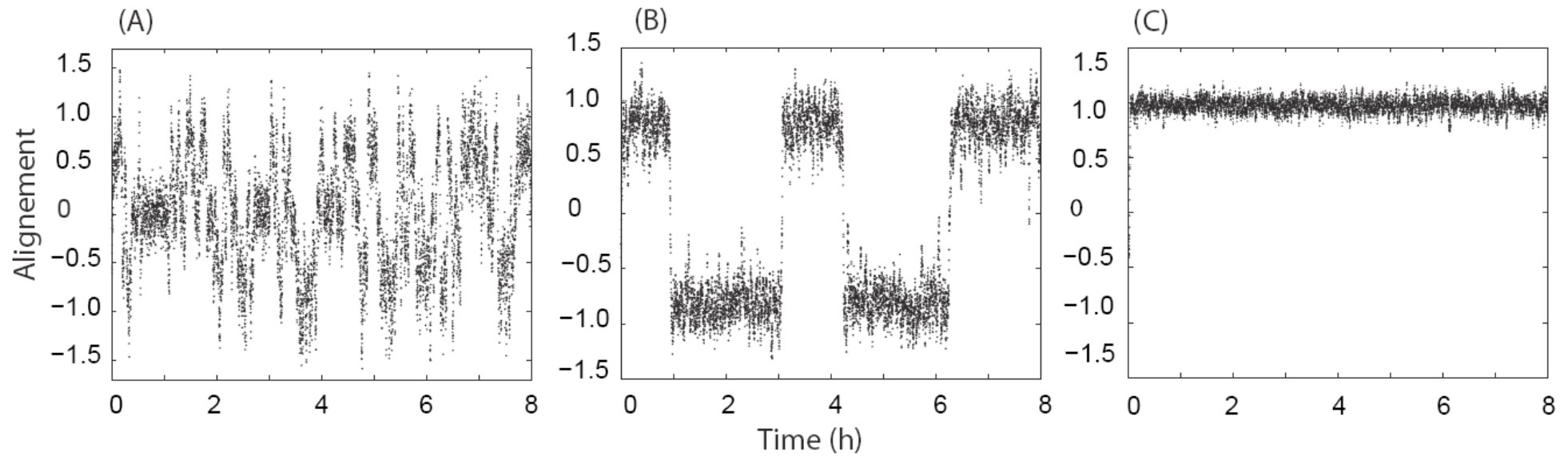
- Run 'Align1D'

$$\begin{aligned}
 & \text{future position} \quad \swarrow \quad x_i(t+1) = x_i(t) + v_0 u_i(t) \quad \swarrow \quad \text{current position} \quad \text{current velocity} \\
 & u_i(t+1) = a u_i(t) + (1-a) s_i(t) + e_i(t) \\
 & \quad \quad \quad \nearrow \quad \quad \quad \nearrow \quad \quad \quad \nearrow \quad \quad \quad \nearrow \\
 & \text{future velocity} \quad \text{current velocity} \quad \text{velocity of neighbours} \quad \text{stochastic effect}
 \end{aligned}$$

$$s_i = G \left(\frac{1}{|R_i|} \sum_{j \in R_i} u_j(t) \right) \quad G(u) = \begin{cases} (u+1)/2 & \text{for } u > 0 \\ (u-1)/2 & \text{for } u < 0 \end{cases}$$

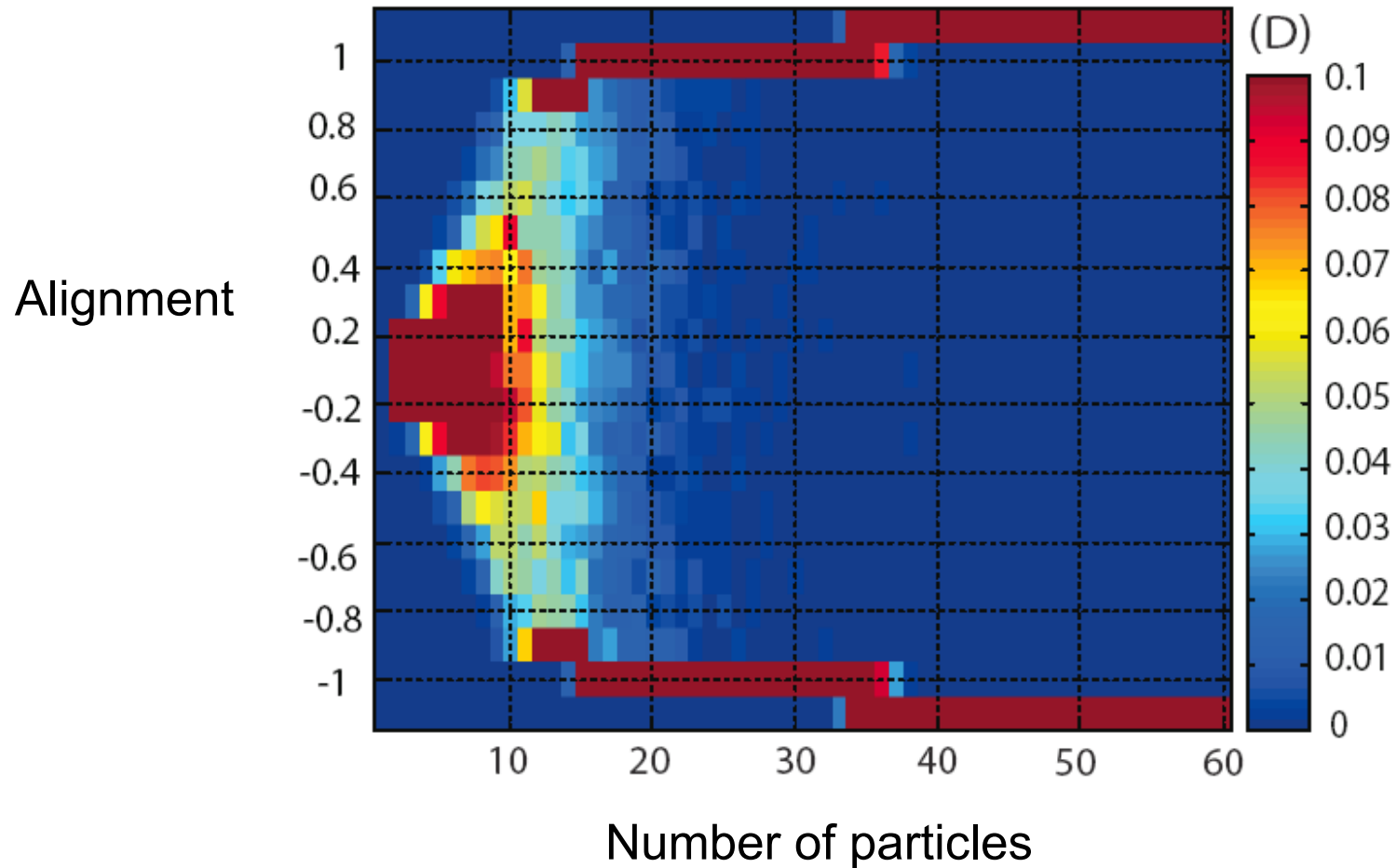
e is a random number selected uniformly at random from a range $[-\eta/2, \eta/2]$

Alignment



$$\phi = \frac{1}{n} \sum_{i=1}^n u_i(t) \quad \text{measures order in the system.}$$

1D self-propelled particles



$$\phi = \frac{1}{n} \sum_{i=1}^n \underline{u}_i(t) \text{ measures order in the system (alignment).}$$

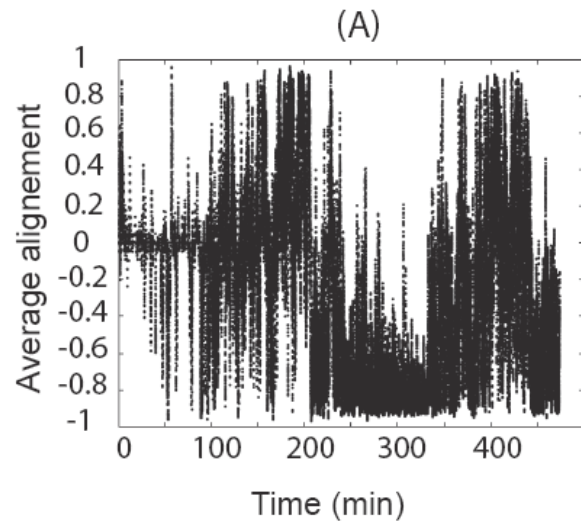




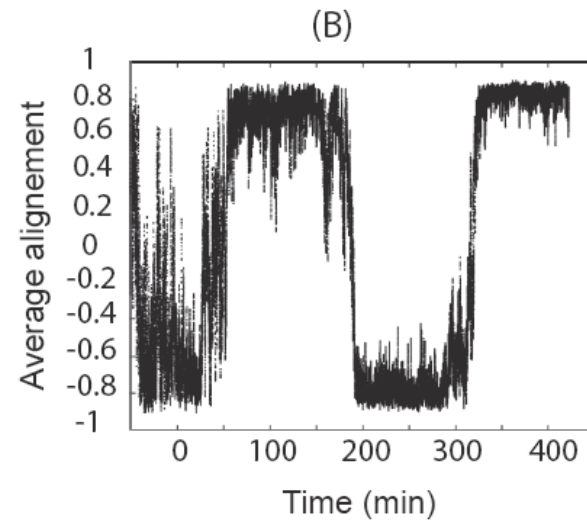
Buhl et al. (2006), *Science*
Yates et al. (2009), *PNAS*

Buhl et al. (2006), *Science*
Yates et al. (2009), *PNAS*

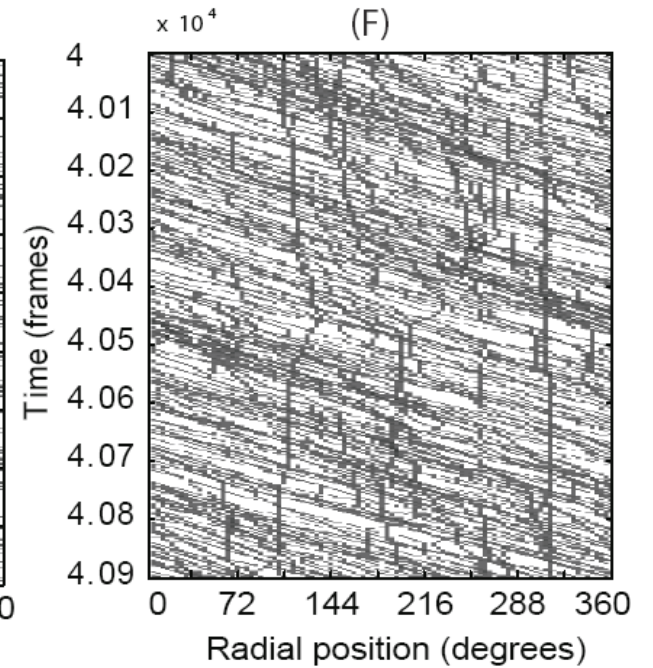
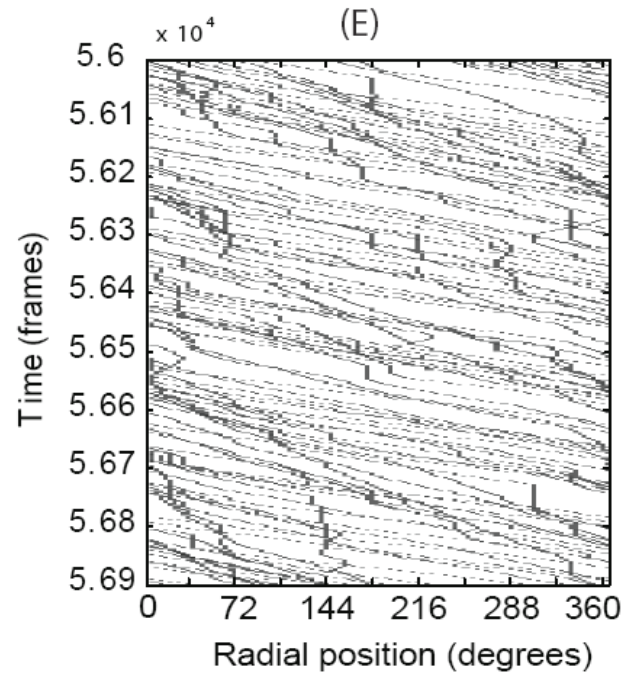
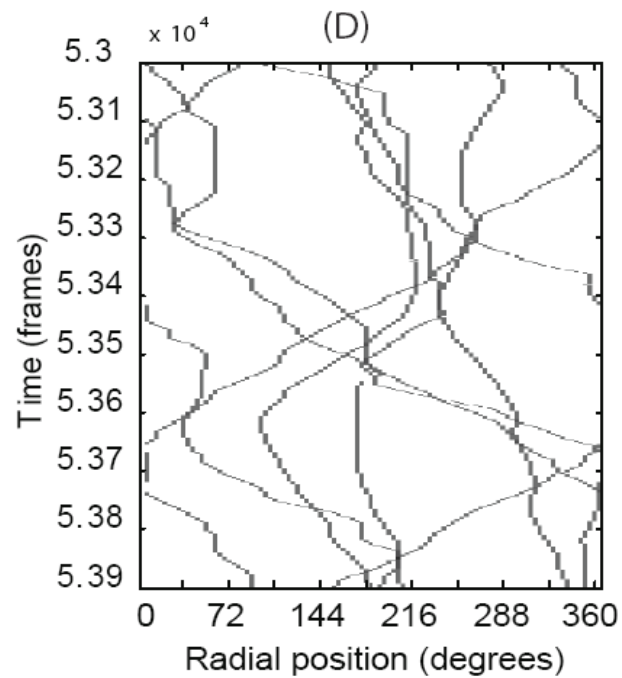
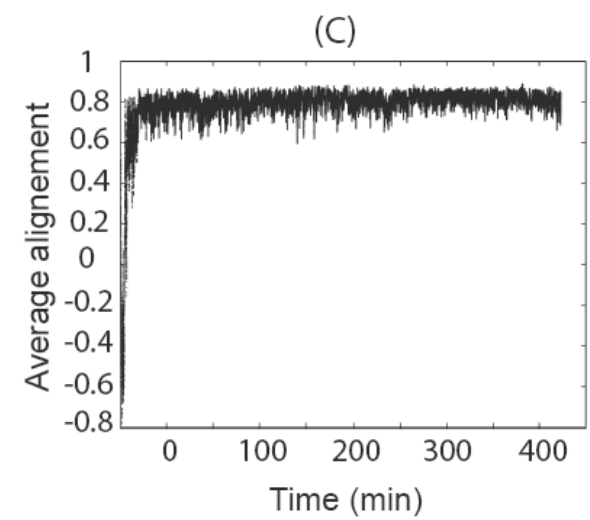
7 locusts

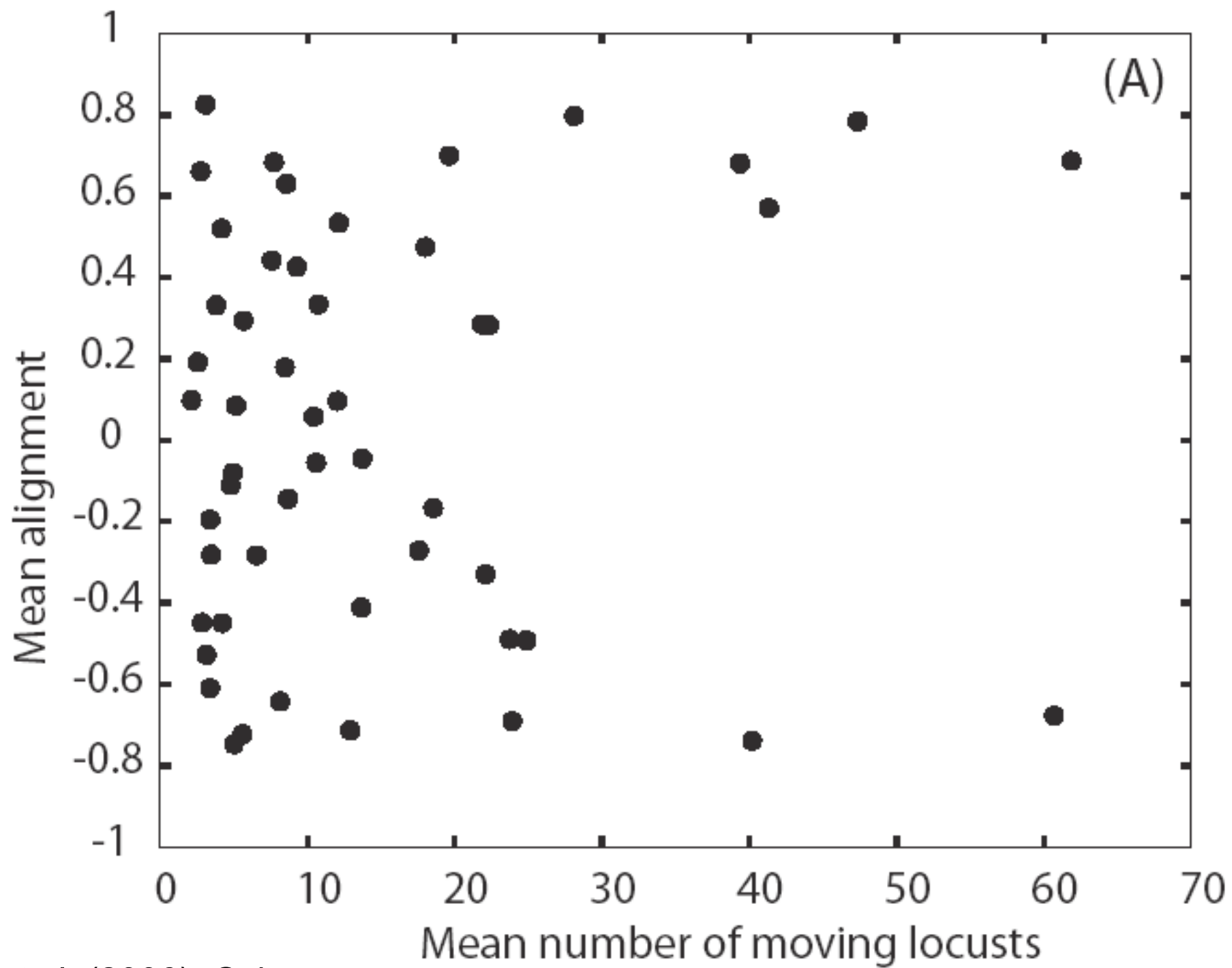


25 locusts

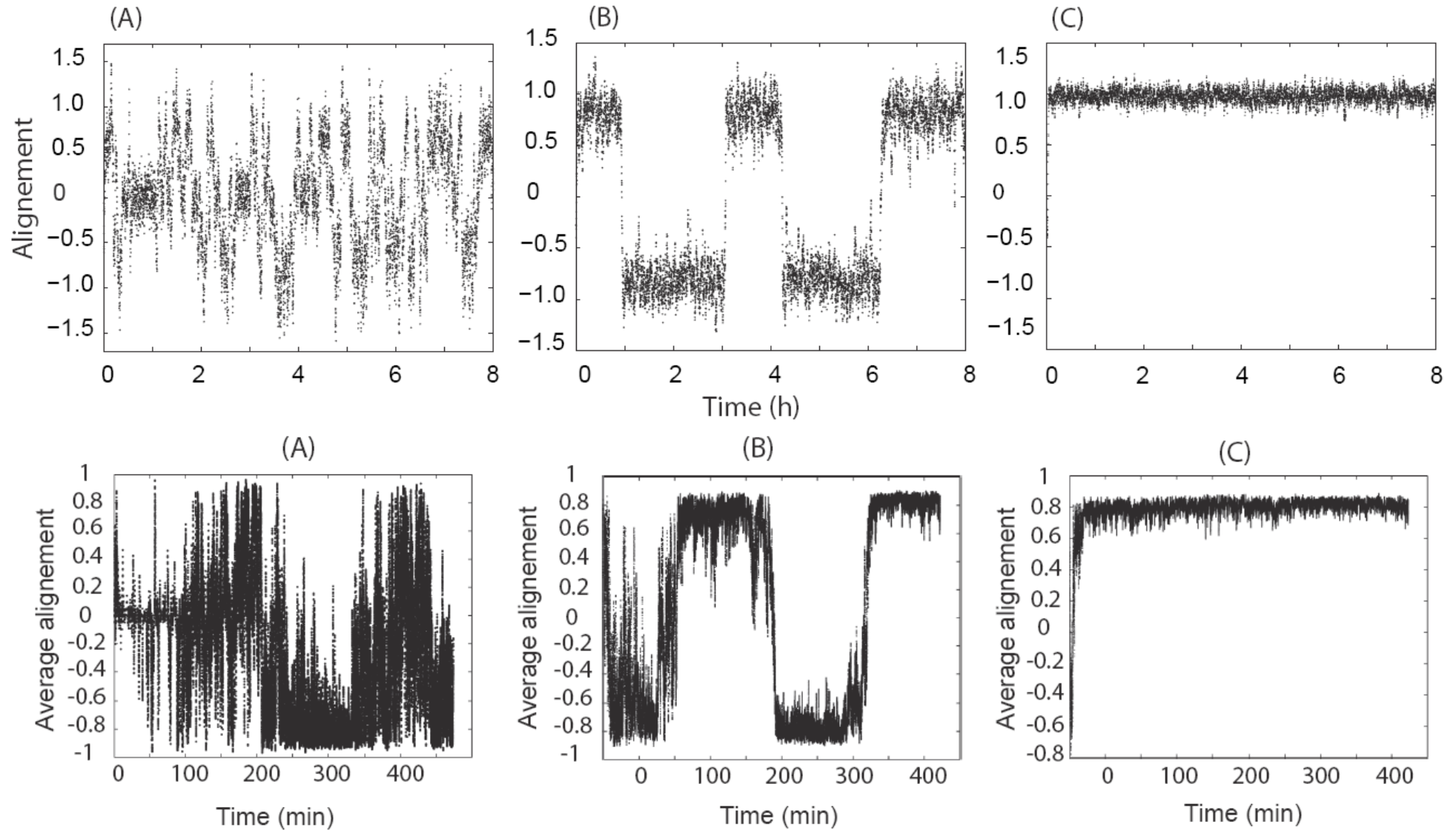


50 locusts

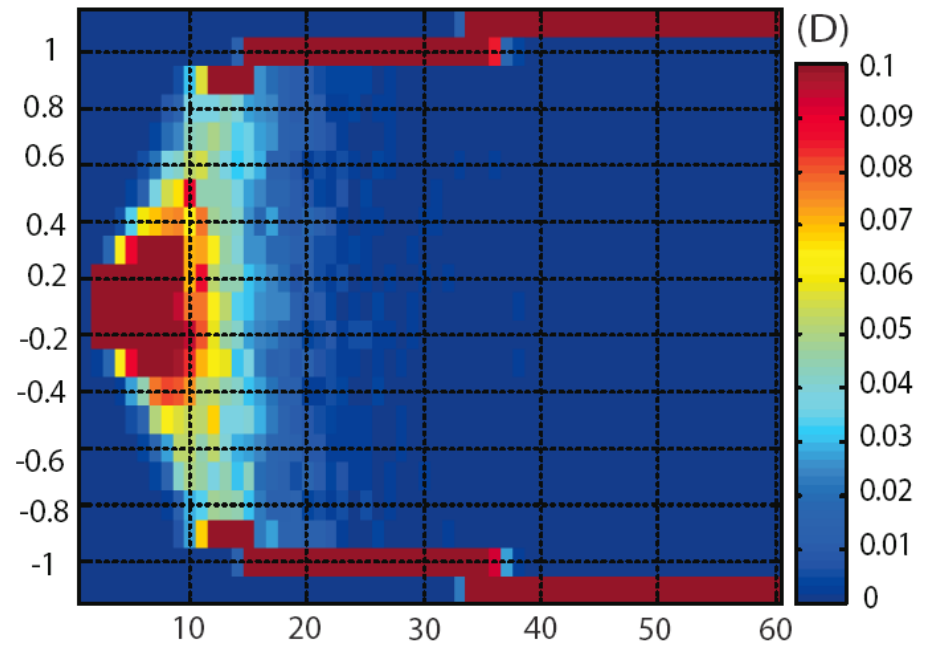
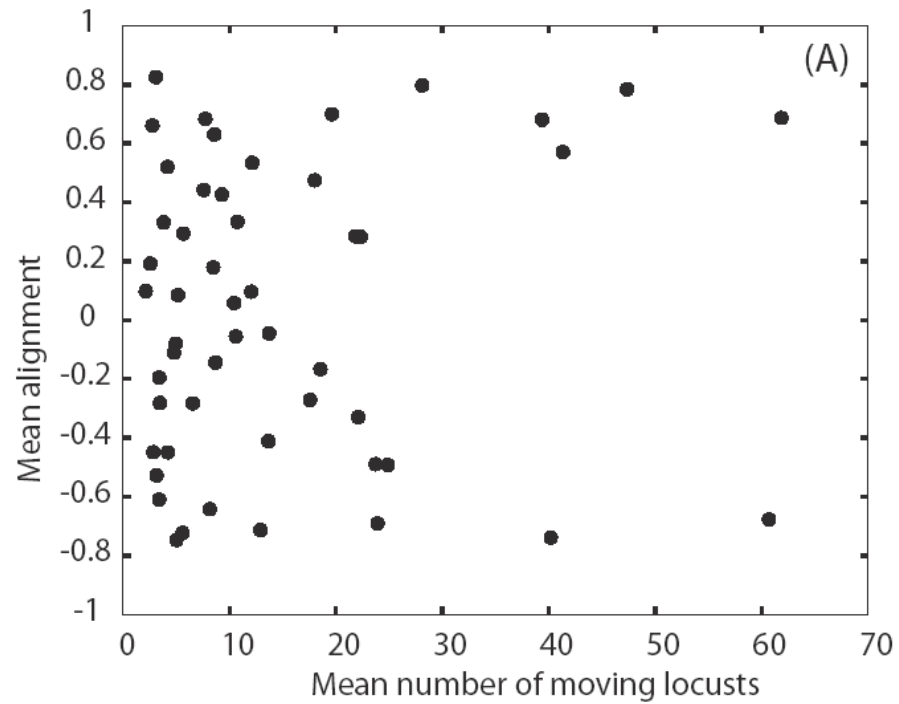




Model vs Experiment



Model vs Experiment



Vicsek Model

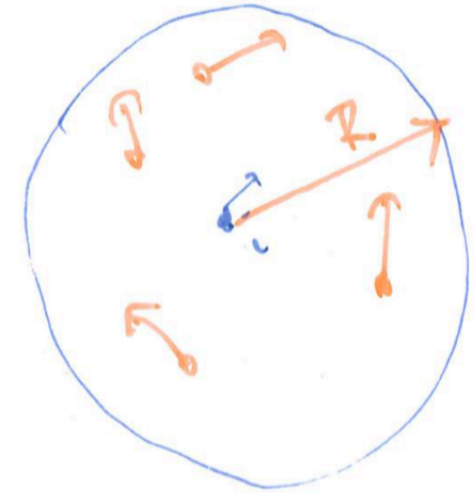
N : number of particles

η : noise parameter

L : size of domain

R : radius of interaction

v : speed



Angular update rule:

$$\theta_i(t+1) = \tan^{-1} \left(\frac{\sum_{j \in R_i} \sin(\theta_j(t))}{\sum_{j \in R_i} \cos(\theta_j(t))} \right) + e(t)$$

$e(t)$ is a random number selected uniformly at random from a range $[-\eta/2, \eta/2]$

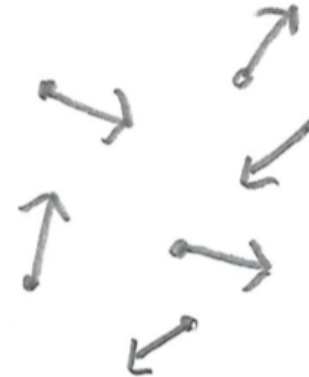
2D Alignment

- Run 'Align2D'

Measure of Alignment: Polarisation



High polarisation

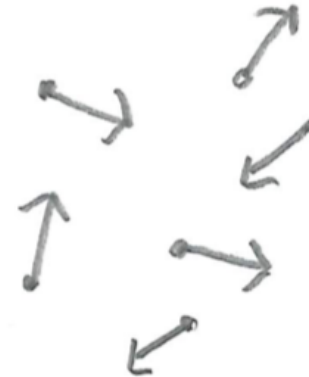


Low Polarisation

Measure of Alignment: Polarisation



High polarisation



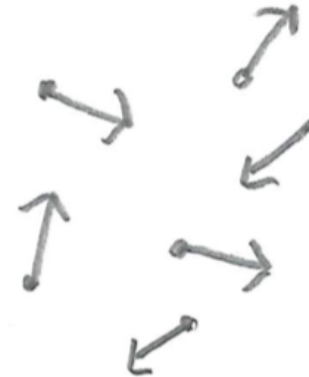
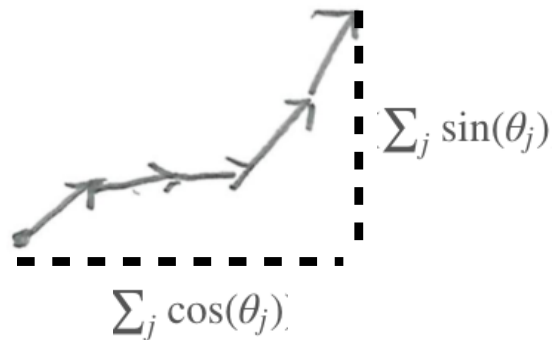
Low Polarisation



Measure of Alignment: Polarisation



High polarisation



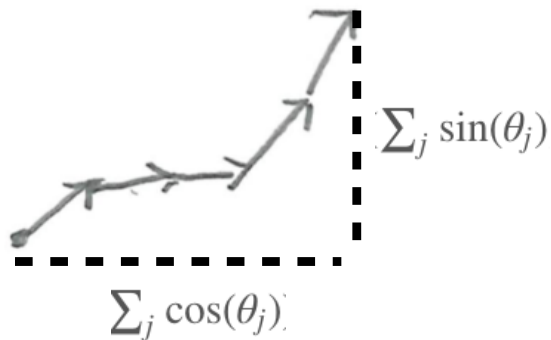
Low Polarisation



Measure of Alignment: Polarisation



High polarisation



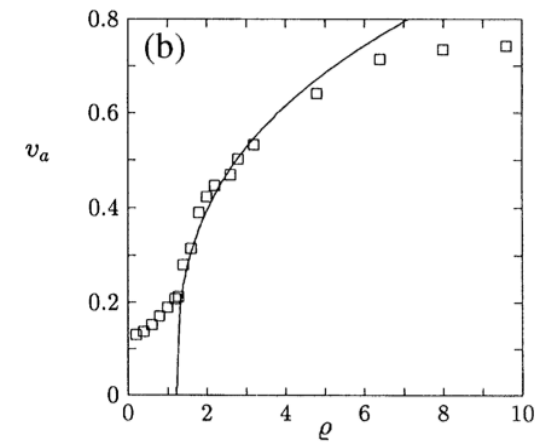
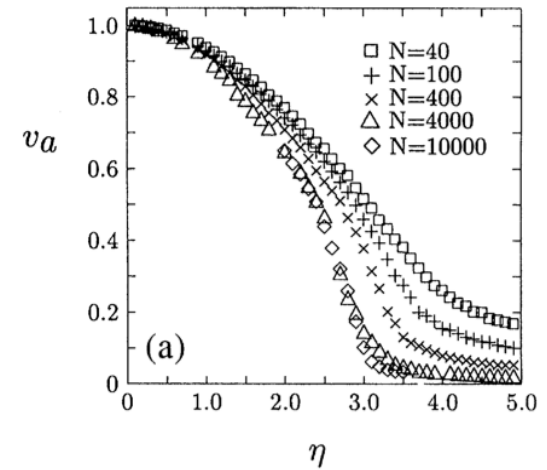
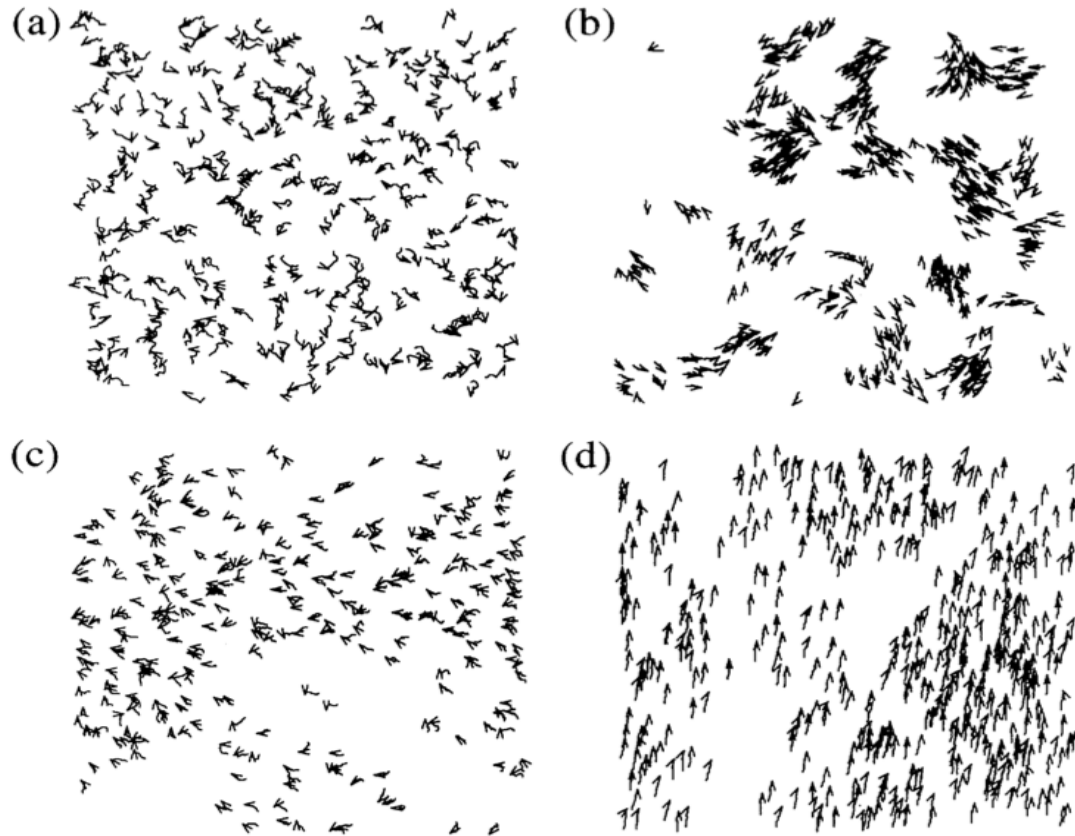
Low Polarisation



Polarisation of:
 $\theta_1, \theta_2, \dots, \theta_N$

$$= \frac{1}{N} \sqrt{(\sum_j \sin(\theta_j))^2 + (\sum_j \cos(\theta_j))^2}$$

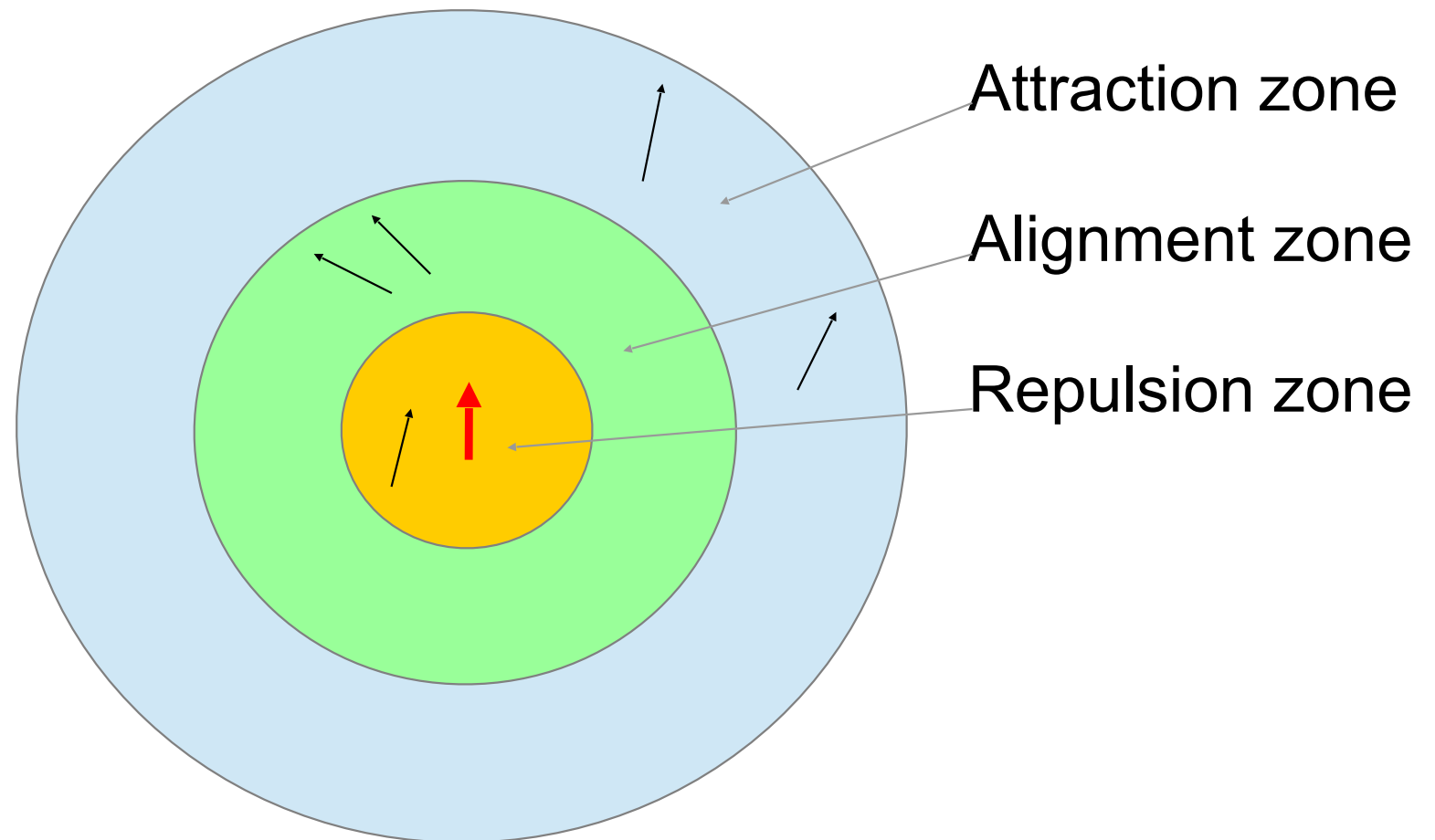
Vicsek Model



Vicsek et al., PRL 75 (1995)

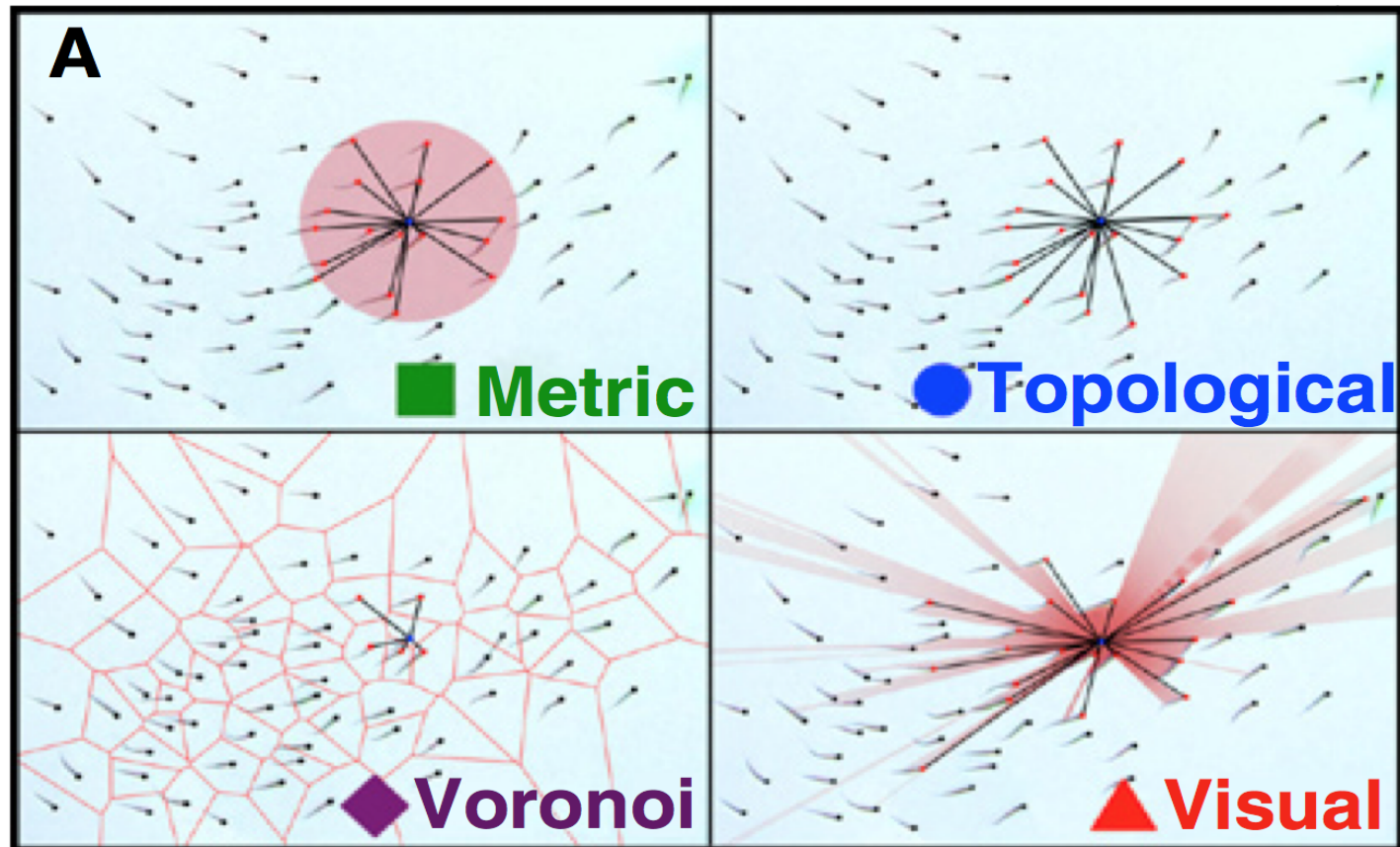
Attraction/Repulsion

“Boids” model

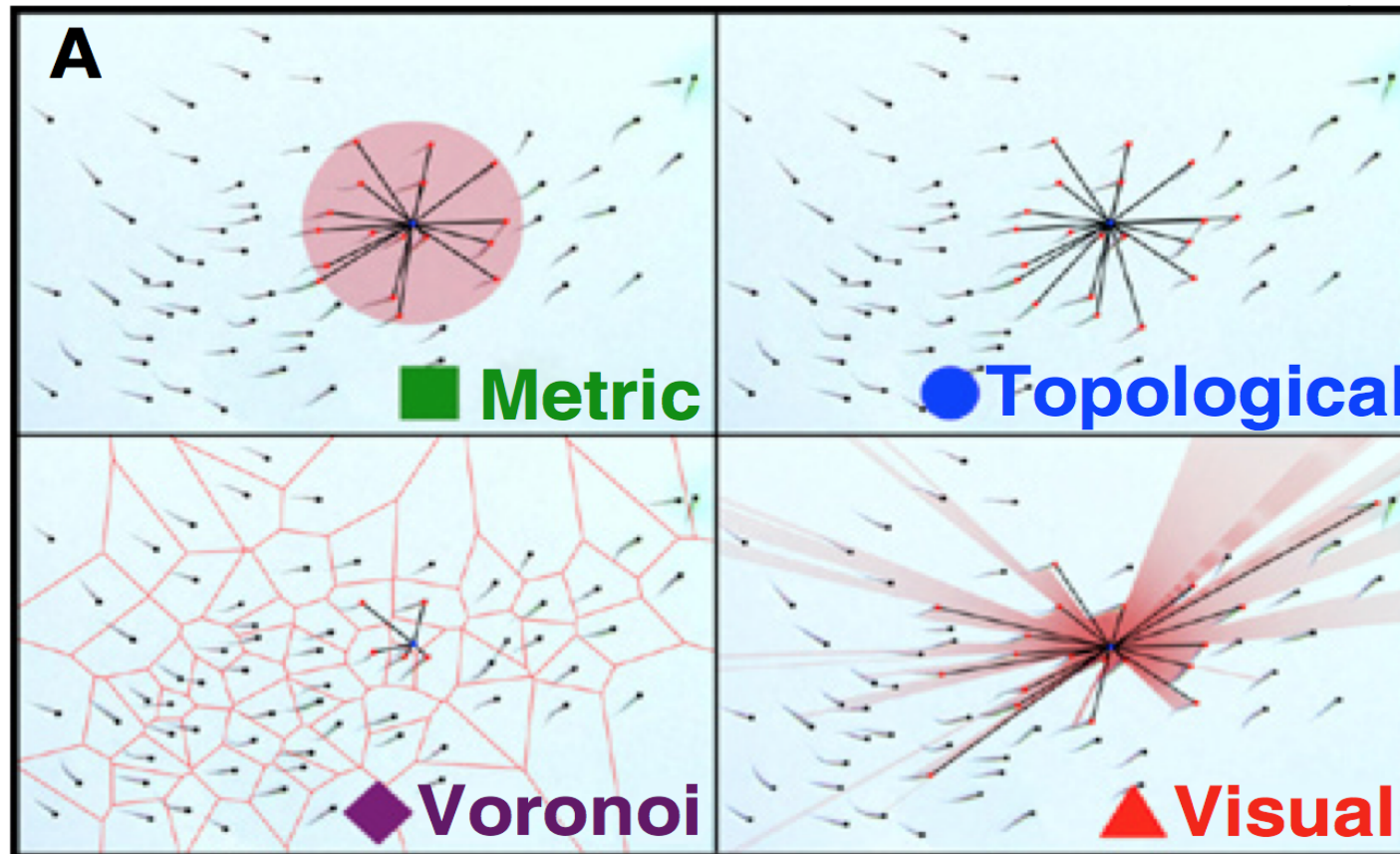


See: Couzin et al., J. Theor. Bio. (2002)

Alternative distance measures



Alternative distance measures



Metric: all individuals within a certain distance.

Topological: a fixed number of nearest neighbors.

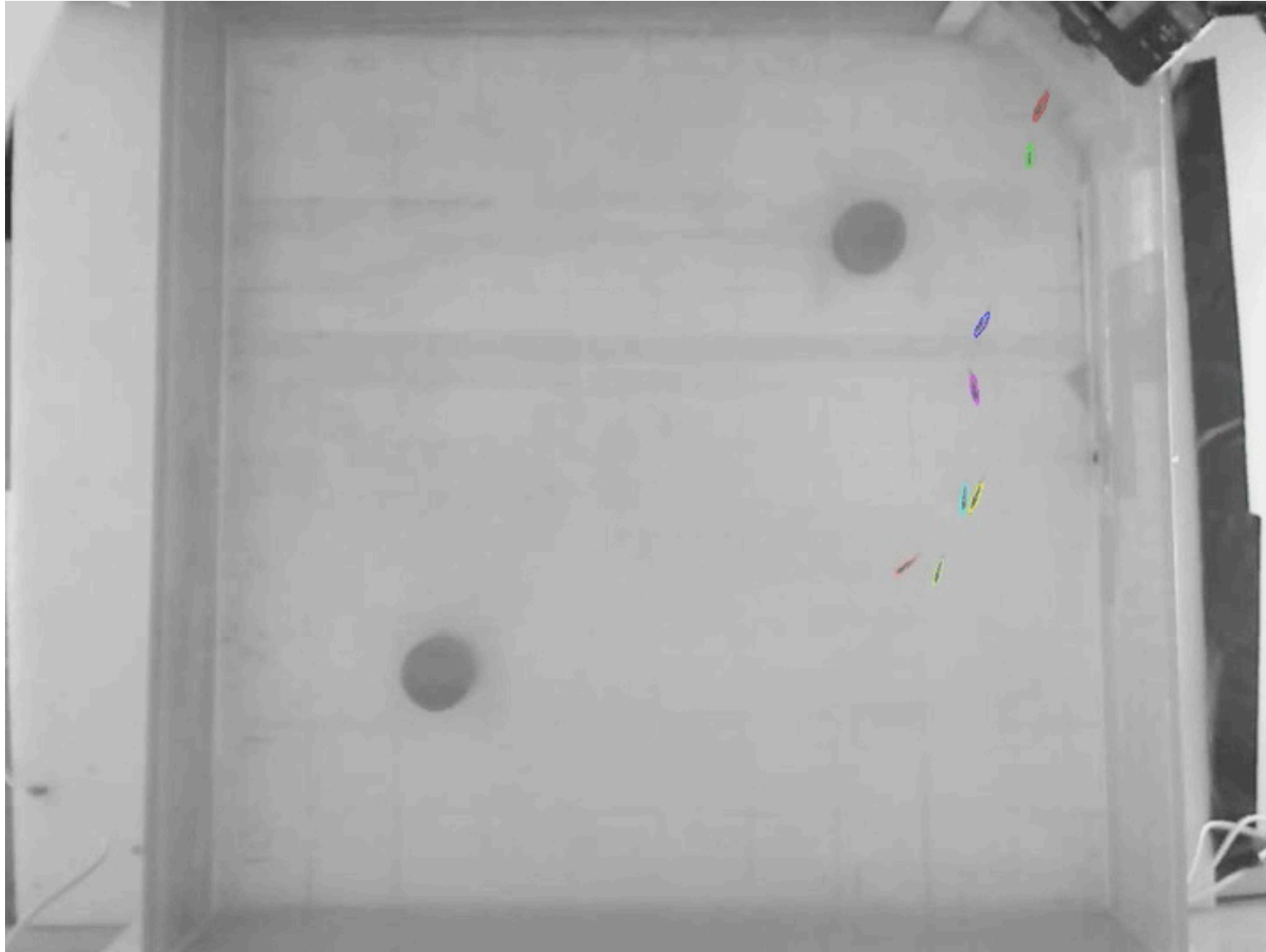
Voronoi: those individuals sharing a boundary in a Voronoi tessellation of the group.

Visual: all individuals that occupy an angular area on the retina of the focal fish that is greater than a threshold value.

Even more options

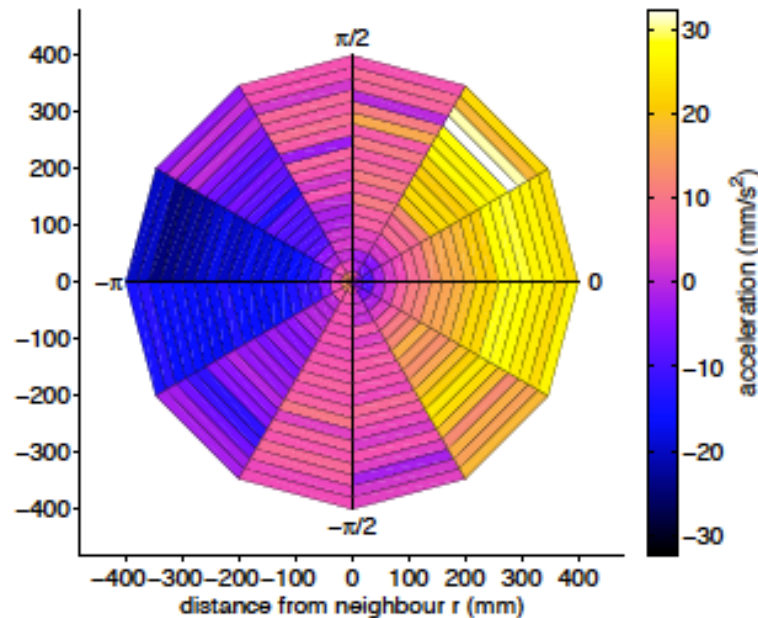
- Maximum turning angles
- Blind angles
- Attraction/repulsion potentials
- Reaction times
- Wall interactions
- Variable speed
- Variation in individuals
- Pheromone trails
- Etc....

Rules of motion

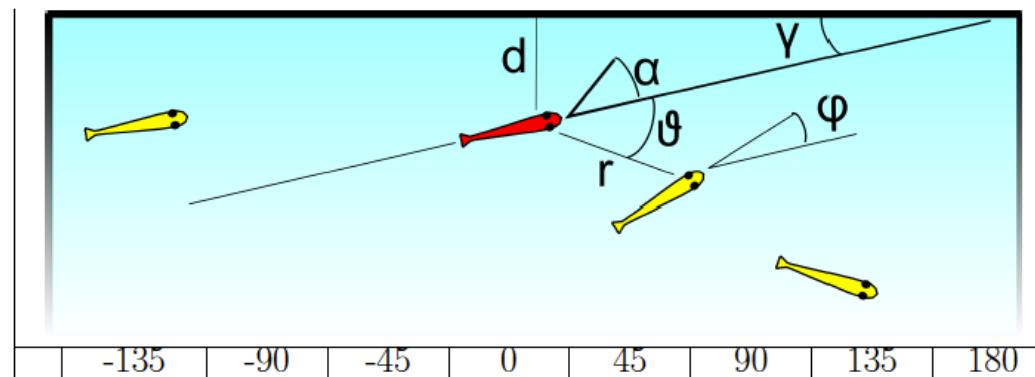
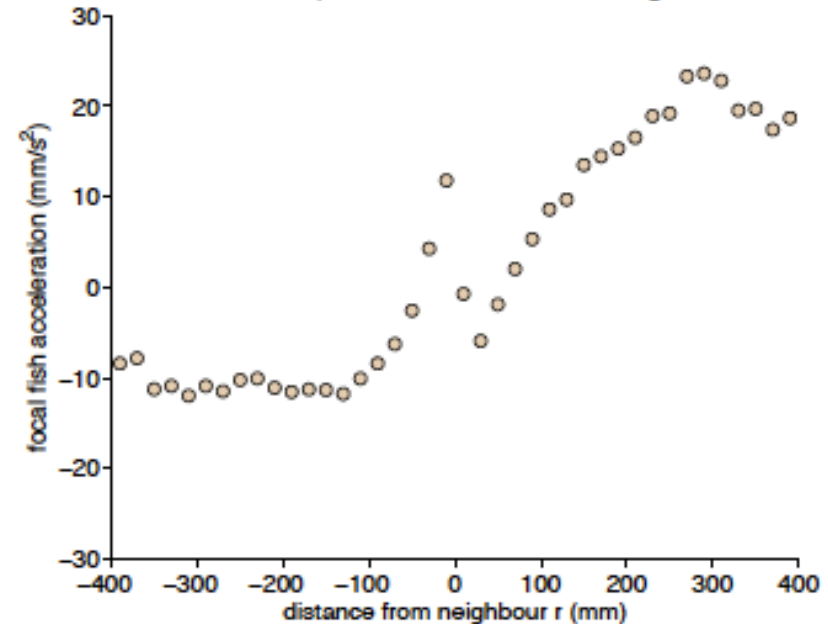


Using data to fit models

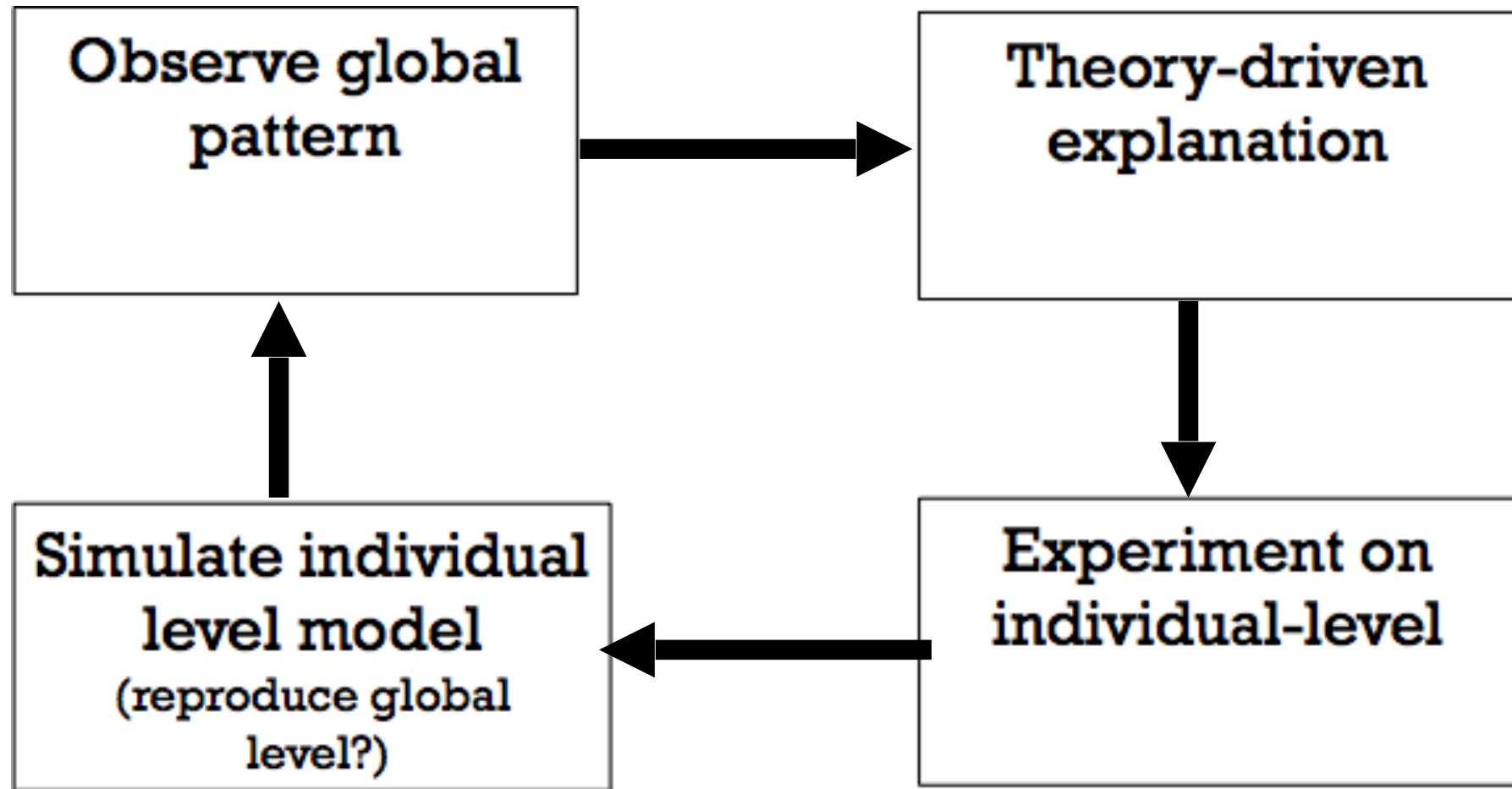
mean acceleration as a function of distance and angle of all neighbours

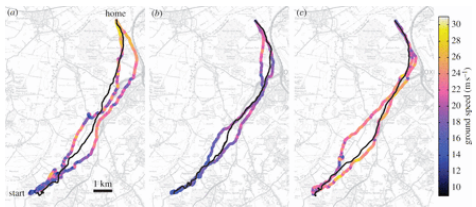
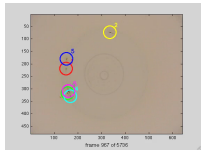
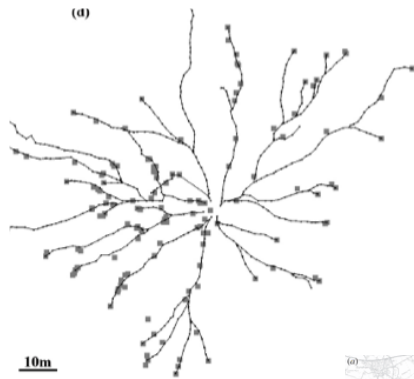
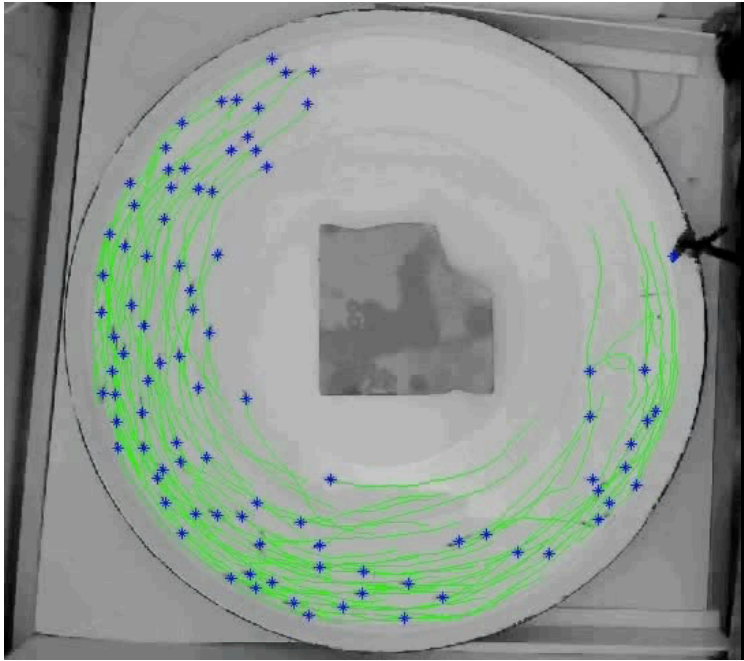


Acceleration profile vs. distance from all neighbours



The modelling cycle





Can you tell the difference between real and simulated fish?



The collage consists of four overlapping screenshots from a game interface:

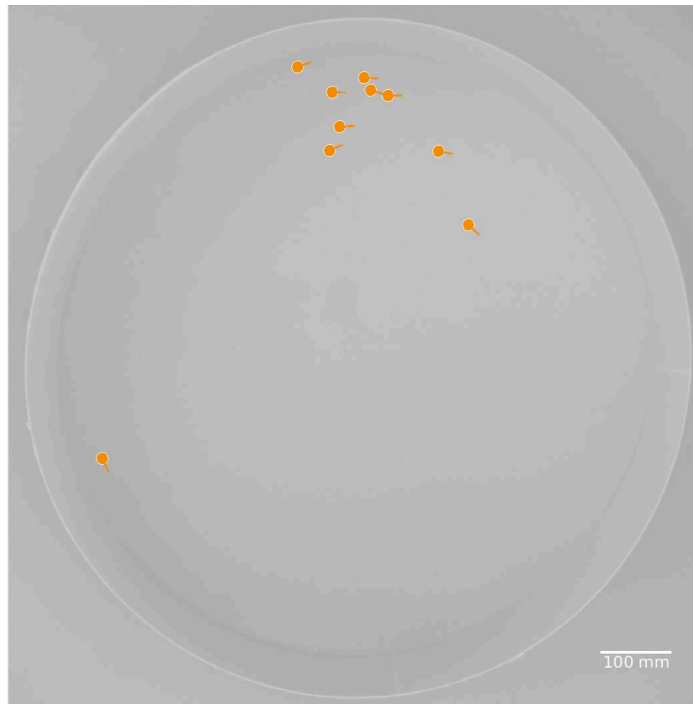
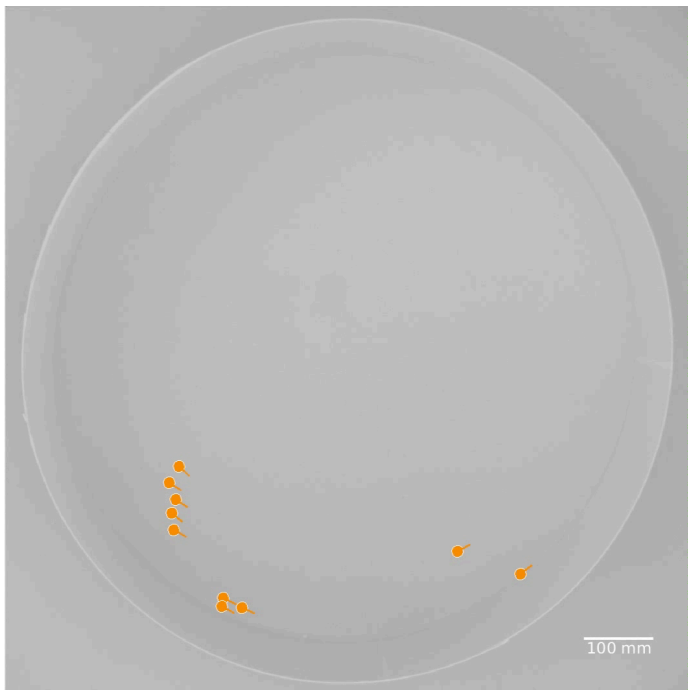
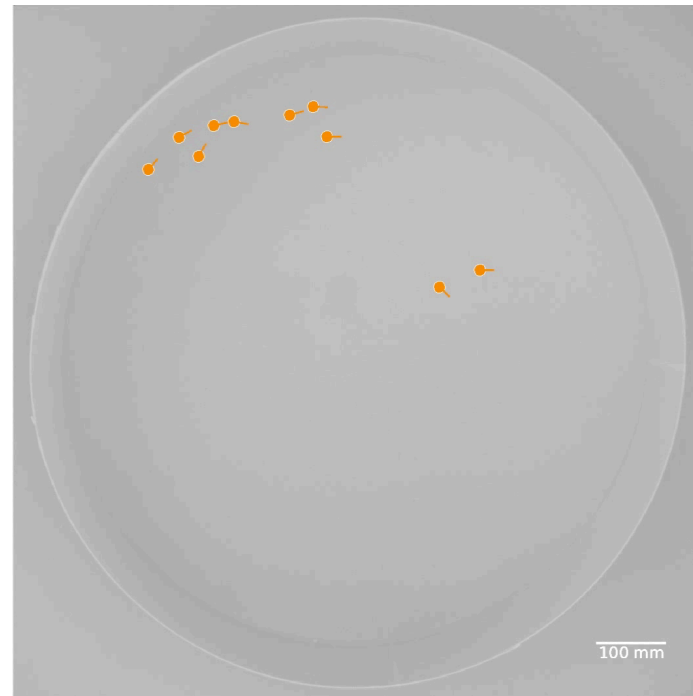
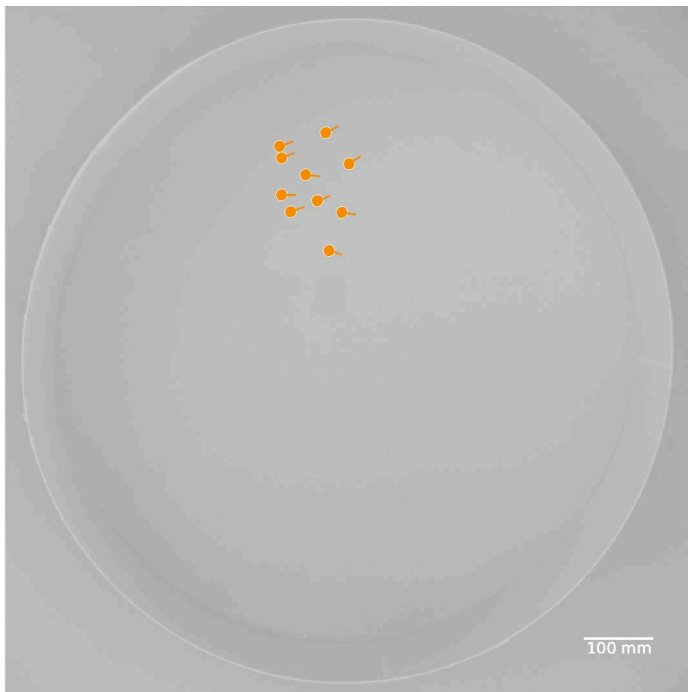
- Top Left:** A black and white video of a dense school of fish. A black rectangular button with the word "Play" in white is overlaid on the bottom right of the video.
- Top Right:** A game screen showing two circular arenas. The left arena is white and contains a small group of green dots. The right arena is grey and contains a larger group of green dots. The text "Make your choice" is written in green between the two arenas. A black rectangular button with the word "Next" in white is overlaid on the bottom right of the grey arena.
- Bottom Left:** A video of a white bowl containing a small number of fish. A black rectangular button with the word "Skip" in white is overlaid on the bottom right of the video.
- Bottom Right:** A white background with the text "you will see two videos. Try to identify movements of real fish and not simulated data." in a small font. A large black rectangular button with the word "Begin" in white is overlaid on the bottom right.

In the center of the collage is a white rectangular box with a black border. It contains the following text and graphics:

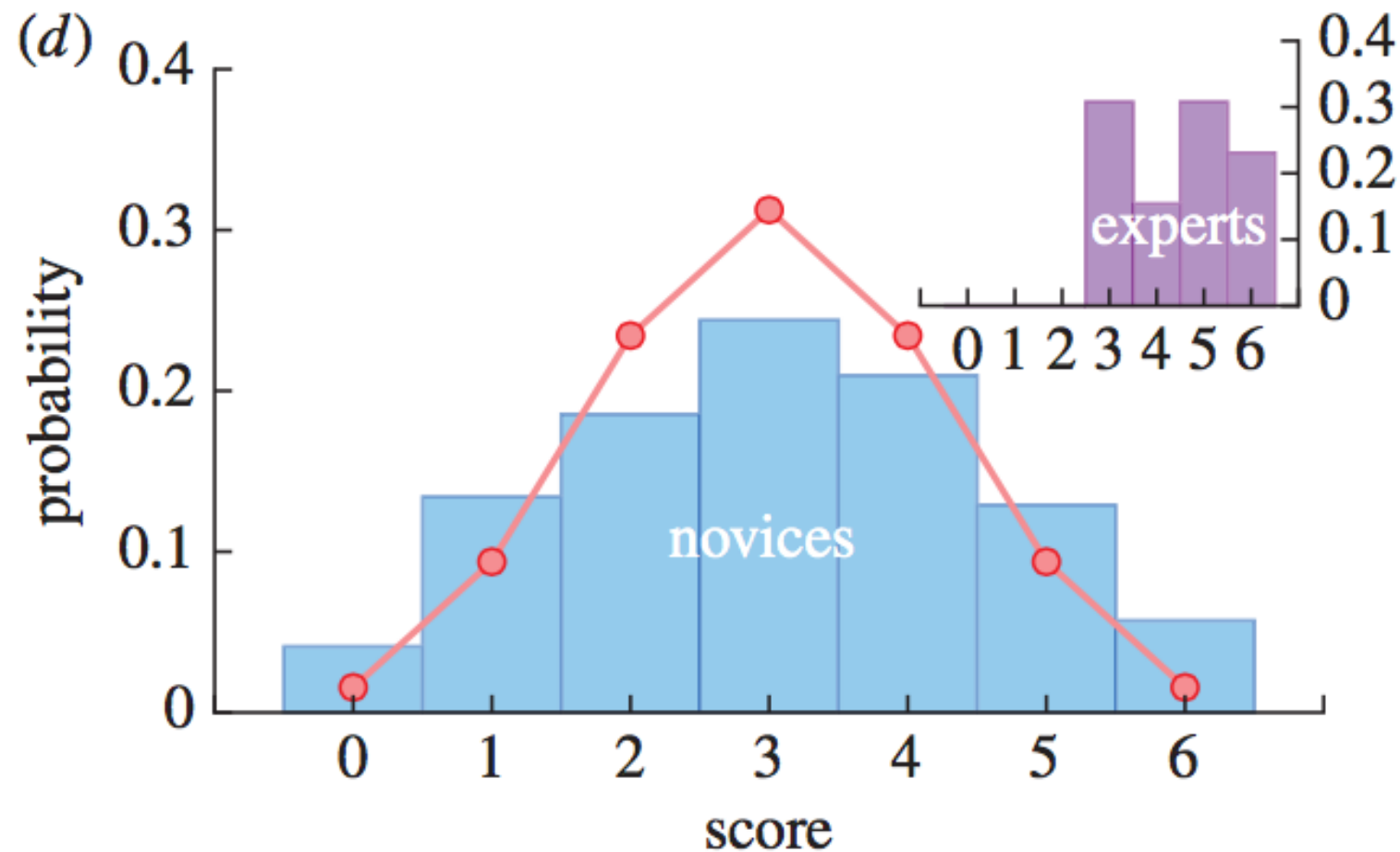
- At the top, the text "Congratulations!" in black.
- Below that, the text "You have answered 5 out of 6 questions correctly." in black.
- In the center, a black line drawing of a fish facing left.
- At the bottom, the text "Click refresh button to play again" in black.

Get playing!

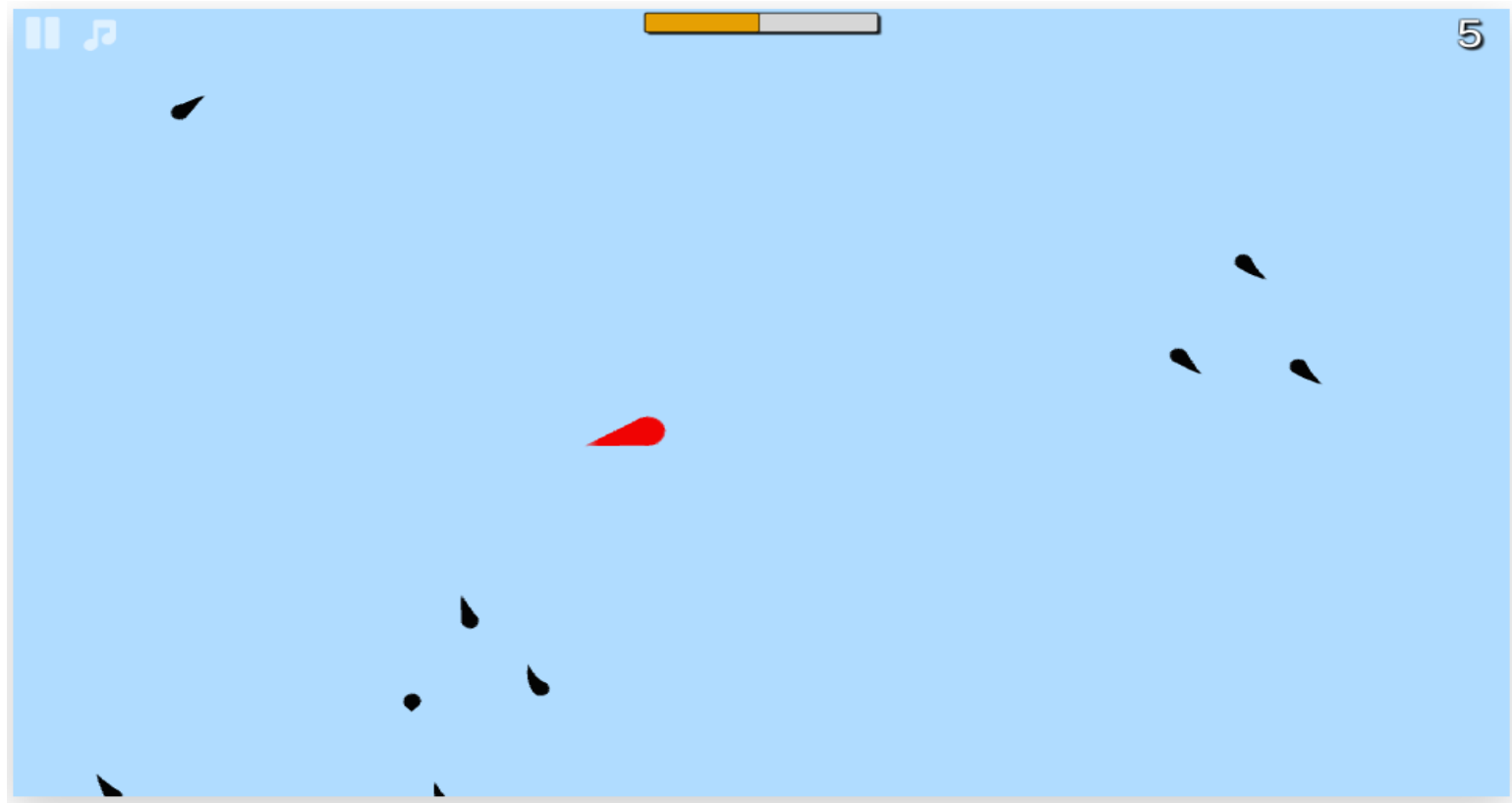
<http://www.collective-behavior.com/apps/>



Can people tell the difference between real and simulated fish?



Evolving prey

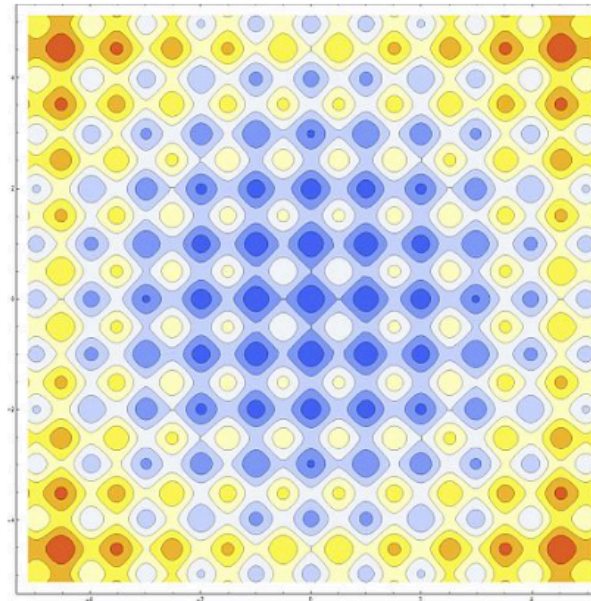
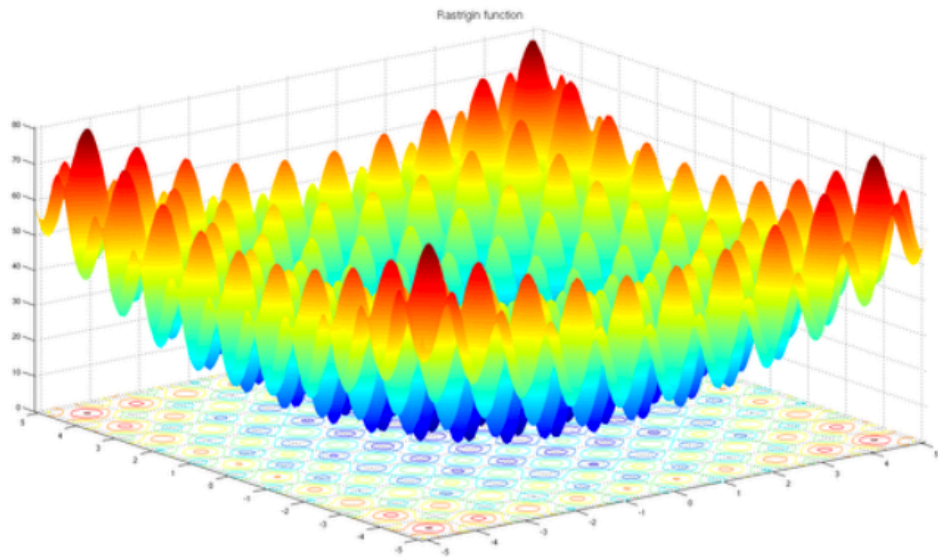


<http://collective-behavior.com/apps/fishindanger/webgl>

Project - Particle Swarm Optimisation

Optimisation problem.

Find global minimum/local minima.



Benchmark: Rastrigin function

$$F(x_1, x_2) = 10n + \sum_{i=1}^2 x_i^2 - 10 \cos(2\pi x_i) \quad x_i \in [-5.12, 5.12]$$

Recall: attraction in one dimension

Diagram illustrating the variables in the equations:

- $x_i(t+1)$ is labeled "future position".
- $x_i(t)$ is labeled "current position".
- v_0 is labeled "current velocity".
- $u_i(t)$ is labeled "current velocity".
- $u_i(t+1)$ is labeled "future velocity".
- a is labeled "current velocity".
- $s_i(t)$ is labeled "Direction to most neighbours".
- $e_i(t)$ is labeled "stochastic effect".

$$x_i(t+1) = x_i(t) + v_0 u_i(t)$$
$$u_i(t+1) = a u_i(t) + (1-a) s_i(t) + e_i(t)$$

$$s_i(t) = \frac{1}{|R_i|} \sum_{j \in R_i} \text{sign}(x_i(t) - x_j(t))$$

$e_i(t)$ is a random number selected uniformly at random from a range $[-\eta/2, \eta/2]$

Extension: Particle swarm optimisation

Diagram illustrating the equations for particle position and velocity updates in Particle Swarm Optimization (PSO).

The first equation shows the update for the future position:

$$x_i^{(t)} = x_i^{(t-1)} + v_i^{(t)}$$

Labels and arrows for the first equation:

- future position (points to $x_i^{(t)}$)
- current position (points to $x_i^{(t-1)}$)
- current velocity (points to $v_i^{(t)}$)

The second equation shows the update for the future velocity:

$$v_i^{(t)} = v_i^{(t-1)} + c_1 U(0, 1) \odot (p_i - x_i^{(t-1)}) + c_2 U(0, 1) \odot (p_g - x_i^{(t-1)})$$

Labels and arrows for the second equation:

- future velocity (points to $v_i^{(t)}$)
- current velocity (points to $v_i^{(t-1)}$)
- stochastic effect (points to the $U(0, 1)$ terms)

N particles. p_1, \dots, p_N best positions of each particle.

p_g - best position of particles in neighbourhood

Extension: Particle swarm optimisation

Diagram illustrating the equations for future position and future velocity in Particle Swarm Optimization, with arrows indicating the variables used in each equation.

future position

current position

current velocity

$$x_i^{(t)} = x_i^{(t-1)} + v_i^{(t)}$$
$$v_i^{(t)} = v_i^{(t-1)} + c_1 U(0, 1) \odot (p_i - x_i^{(t-1)}) + c_2 U(0, 1) \odot (p_g - x_i^{(t-1)})$$

future velocity

current velocity

cognitive

social

N particles. p_1, \dots, p_N best positions of each particle.

p_g - best position of particles in neighbourhood

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

```

x = rand(N, d)      # positions
v = rand(N, d)      # velocities
p = rand(N, d)      # previous best position
pbest = infinity(N) # best function value
g = 0               # index of best in neighborhood

# Run some amount of iterations
for t in range(iter):

    # Update all particles
    for i in range(N):

        # Check if F at current x is better than previous and update pbest, p.
        if F(x[i]) < pbest[i]:
            pbest[i] = F(x[i])
            p[i] = x[i]

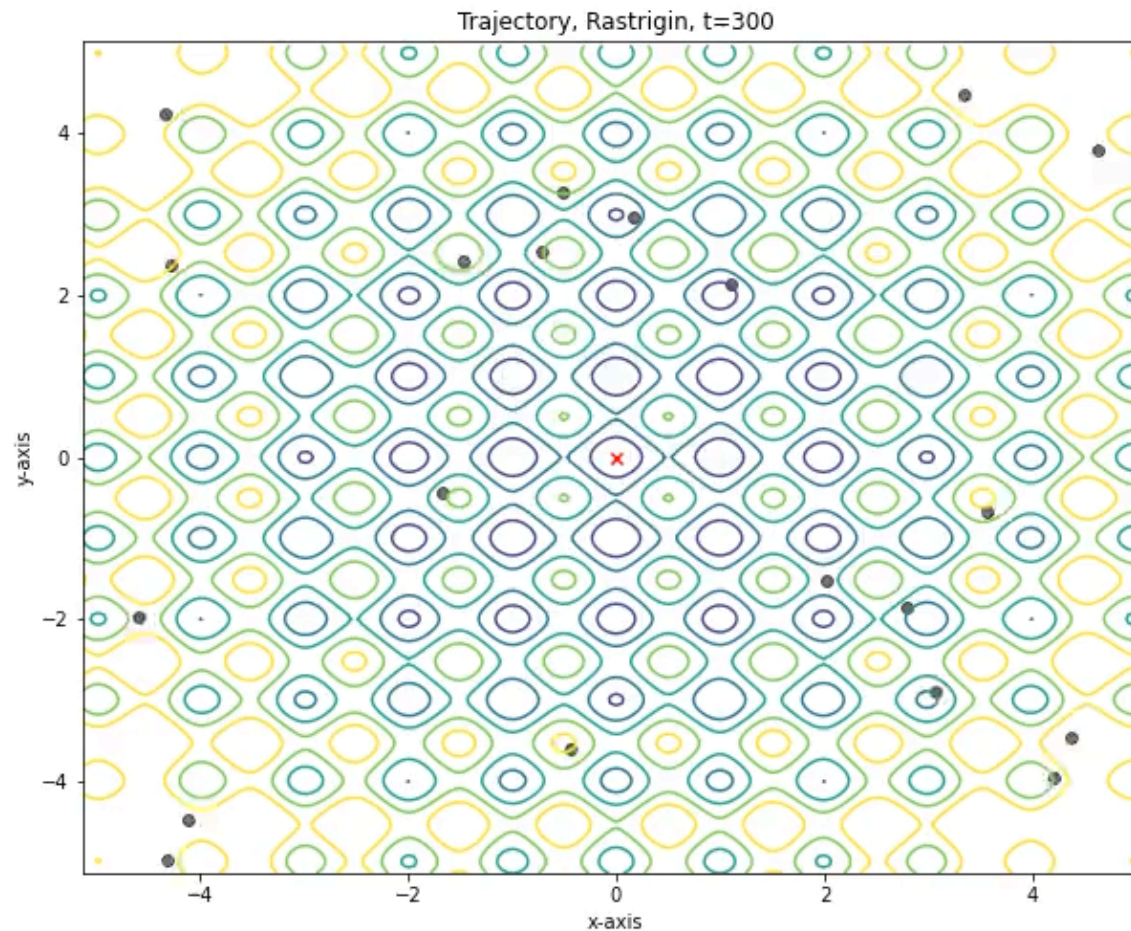
        # Get neighbors and get index of best performing particle
        neighbors = get_neighbors(i)
        g = best_performer(neighbors)

        # Update velocity and position
        v[i] += mult_elem(c1*rand(d), (p[i] - x[i])) +
                  mult_elem(c2*rand(d), (p[g] - x[i]))
        x[i] += v[i]

```


Extension: Particle swarm optimisation

Jonas Olsson



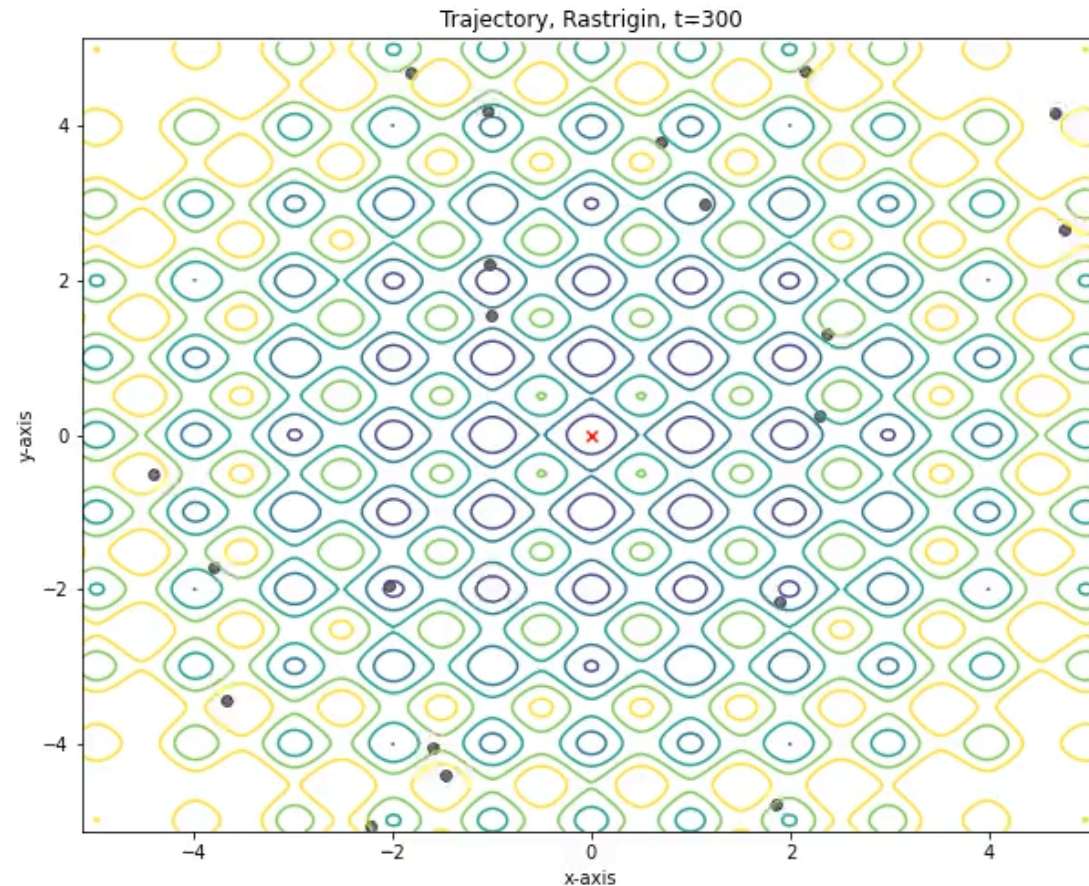
$$c_1 = 1.49618, c_2 = 1.49618$$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

Extension: Particle swarm optimisation

Jonas Olsson



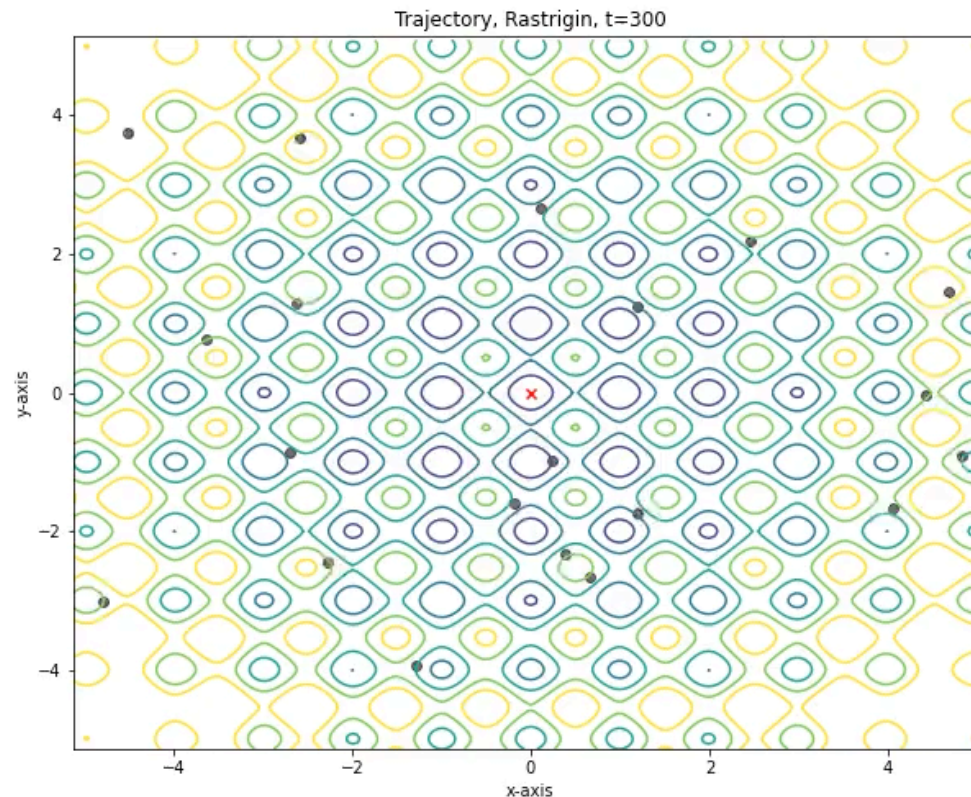
$$c_1 = 0, c_2 = 1.49618$$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

Extension: Particle swarm optimisation

Jonas Olsson



$$c_1 = 1.49618, c_2 = 1.49618, w = 0.7298$$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

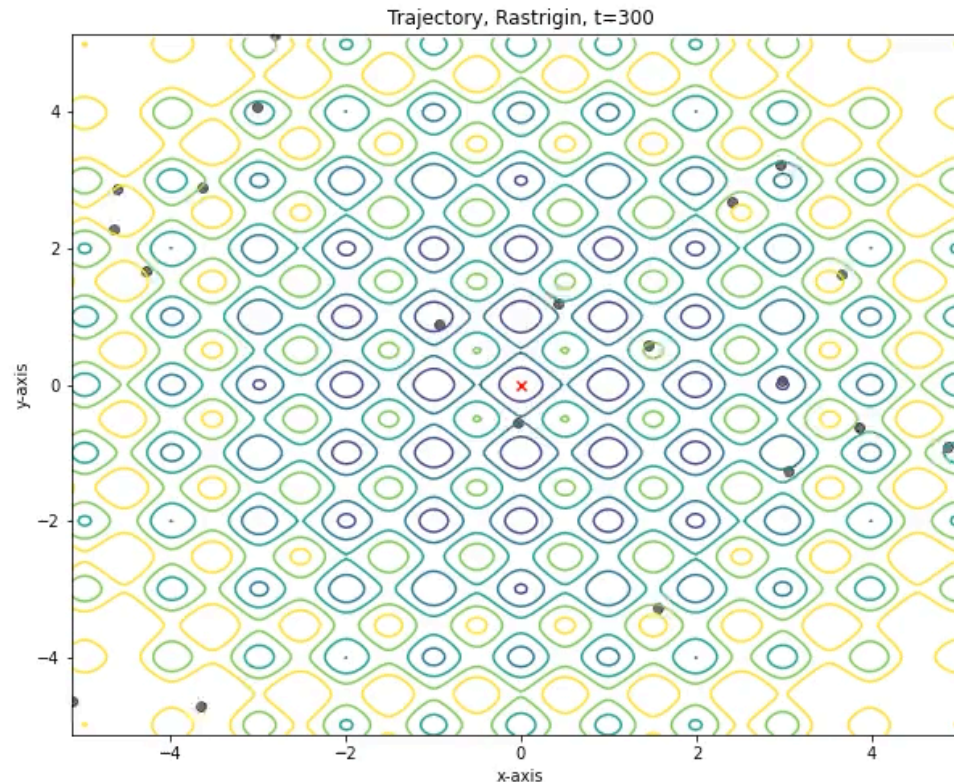
w inertia/constriction/viscosity,

large w - viscosity low, particles move easily - favours global min

Small w - viscosity high particles move slower - favours local min

Extension: Particle swarm optimisation

Jonas Olsson



$$c_1 = 0, c_2 = 1.49618, w = 0.7298$$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

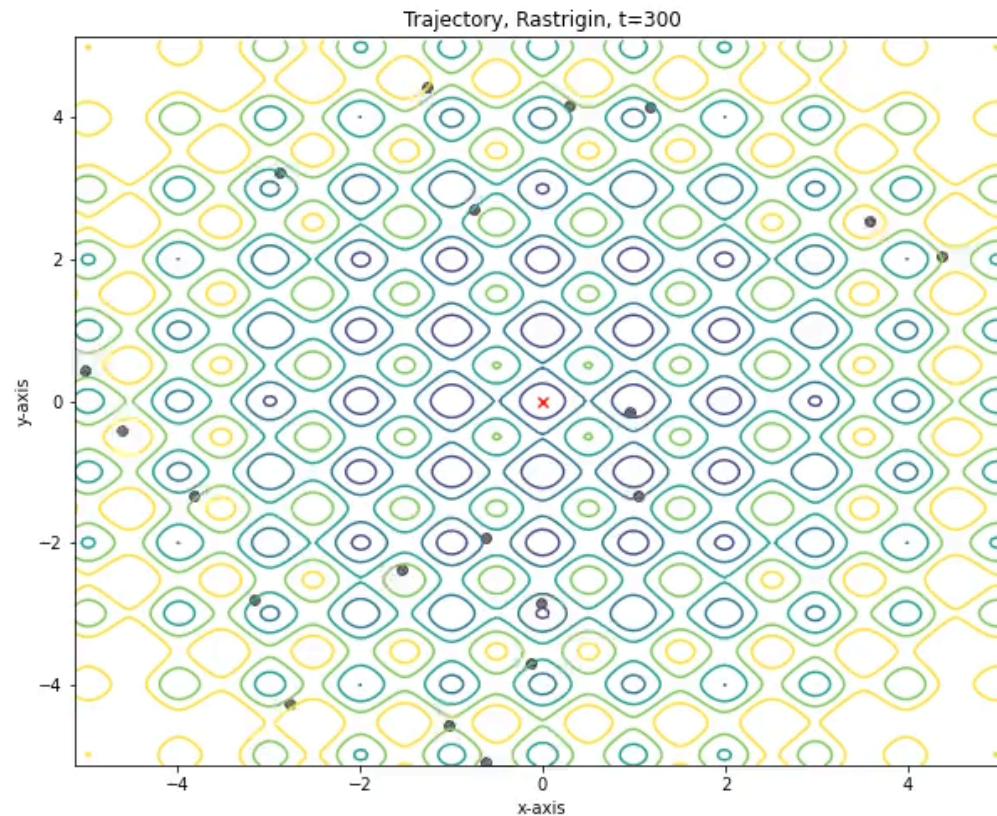
w inertia/constriction/viscosity,

large w - viscosity low, particles move easily - favours global min

Small w - viscosity high particles move slower - favours local min

Extension: Particle swarm optimisation

Jonas Olsson



$$c_1 = 1.49618, c_2 = 0, w = 0.7298$$

c_1 Cognitive - pulls particle towards best position it has had so far

c_2 Social - pulls particle towards best position so far of those in its neighbourhood

w inertia/constriction/viscosity,

large w - viscosity low, particles move easily - favours global min

Small w - viscosity high particles move slower - favours local min