Analysis of an interacting particle model with applications to capelin (*Mallotus villosus*)

Baldvin Einarsson *et al.*

Center for Complex and Nonlinear Sciences
University of California, Santa Barbara

San Diego, November 14, 2011
Collaborators:

Björn Birnir, UCSB
Alethea Barbaro, UCLA
Sven Sigurdsson, University of Iceland
Outline

Interacting particle model

Spawning migrations of the capelin

DEB model

Future work
Interacting particle models

Interacting particle models have been used extensively to model movements of

- fish
- locust
- birds

Also, Reynolds (1987), used an IBM to generate computer graphics. This technique is widely used today, e.g. in the movie “Lion King”.

Interacting particle models

The model we use here is similar to that Vicsek et al. (1995). However, we include three zones around each particle which determine how a particle reacts to its neighbors (Based on Aoki (1982), and Huth and Wissel (1992)).
Zones of interaction

We look at three zones around each particle, determining how neighboring particles affect the particle.

\( R_k \), Repulsion,  
\( O_k \), Orientation and  
\( A_k \), Attraction.

(Sets of indices of neighboring particles)
Zones of interaction

Particle $k$ responds to another particle within...

- the zone of repulsion by heading away from that particle
- the zone of orientation by adjusting its directional heading to the other particle’s directional heading
- the zone of attraction by heading towards the particle

When many particles are within these zones, these factors have to be weighed together.
Reaction to neighbors

The reaction to the neighbors is as follows:

\[ d_k(t + \Delta t) := \frac{1}{|I_k(t)|} \left( \sum_{r \in R_k} \frac{q_k(t) - q_r(t)}{\| q_k(t) - q_r(t) \|} \right) \]

\[ + \sum_{o \in O_k} \left( \begin{array}{c} \cos(\phi_o(t)) \\ \sin(\phi_o(t)) \end{array} \right) \]

\[ + \sum_{a \in A_k} \frac{q_a(t) - q_k(t)}{\| q_a(t) - q_k(t) \|}, \]

where \( |I_k(t)| \) denotes the number of neighbors of particle k.
Underlying particle model

We look at the system on a torus of size $L^2$. Particle $k$ therefore updates its position as follows:

$$\begin{pmatrix} x_k(t + \Delta t) \\ y_k(t + \Delta t) \end{pmatrix} = \begin{pmatrix} x_k(t) \\ y_k(t) \end{pmatrix} + \Delta t \cdot \nu \cdot \frac{d_k(t + \Delta t)}{\|d_k(t + \Delta t)\|}$$

- Here $d_k$ is the directional heading of particle $k$ according to the interaction zones.
- Here, the speed $\nu$ is fixed.
Scaling of parameters

Number of capelin is several billions. For computational reasons we simulate with fewer particles.

Want to establish how parameters should scale when number of particles is changed, such that:

- Number of particles within the sensory zones to be constant
- Behavior of a school unchanged
Scaling arguments

In Barbaro et al. (2009) we present the following scaling arguments:

\[ \frac{1}{\sqrt{N}} \sim r_r \sim r_o \sim r_a \sim \Delta q \sim \Delta t \]

where \( \Delta q \) is the typical distance a particle travels during each time step.
Measures - order parameters

We now look at several order parameters:

\[ R := \| \frac{1}{N} \sum_{i=1}^{N} \begin{pmatrix} \cos(\theta_i(t)) \\ \sin(\theta_i(t)) \end{pmatrix} \| \]

\[ r_k := \| \frac{1}{|I_k|} \sum_{j \in I_k} \begin{pmatrix} \cos(\theta_j(t)) \\ \sin(\theta_j(t)) \end{pmatrix} \| \]

\[ \bar{r} := \frac{1}{N} \sum_{i=1}^{N} r_i \]

We call \( R \) the global order parameter. Similarly, \( r_i \) is a local order parameter, and \( \bar{r} \) is the average local order parameter.
Measures - number of neighbors

We let \( n_k \) denote the number of neighbors of particle \( k \) at each time step. Let

\[
\bar{n} := \frac{1}{N} \sum_{i=1}^{N} n_i
\]

denote the average number of neighbors. We see that

\[
\bar{n}_U = \frac{N}{L^2 \pi r_a^2} - 1
\]

is the average number of neighbors at uniform density, to which we compare \( \bar{n} \).
Behavior

Both $R$ and $\bar{r}$ reach a value of 1, as expected in the absence of noise. The average number of neighbors reaches a limit $\bar{n}_E$. 
Average number of neighbors

In each simulation, the average number of neighbors reaches an equilibrium value, $\bar{n}_E$.

In all simulations we see that the average number of neighbors is higher than expected from $\bar{n}_U$.

Additionally, the ratio $\bar{n}_E/\bar{n}_U$ is around 30% higher in all simulations.
In each simulation, the average number of neighbors reaches an equilibrium value, $\bar{n}_E$.

In all simulations we see that the average number of neighbors is higher than expected from $\bar{n}_U$.

Additionally, the ratio $\bar{n}_E/\bar{n}_U$ is around 30% higher in all simulations.
Relaxation times

Both $R$ and $\bar{r}$ reach 1. Define $t_R$ and $t_r$ as relaxation times to consensus.

We see that $t_r \ll t_R$, meaning that the system reaches a local consensus much faster than it reaches a global consensus.
Order parameters

![Graph showing order parameters over time](graph.png)
Introduction

We now want to apply the above model on the spawning migrations of the Icelandic capelin (*Mallotus villosus*) which is a small pelagic fish:

We are simulating the route of the spawning migration.
Migration routes

Undertake a feeding migration to the feeding grounds near Jan Mayen. [Not simulated]
Return in October and November and undertake a spawning migration to spawning grounds in the south of Iceland.

The capelin spawn in February-March and then all die.
Migration routes

Undertake a feeding migration to the feeding grounds near Jan Mayen. [Not simulated]
Return in October and November and undertake a spawning migration to spawning grounds in the south of Iceland.

The capelin spawn in February-March and then all die.
The capelin are known to be quite sensitive to oceanic temperature. We model this reaction with a temperature reaction function, described below.

Partridge (1982) pointed out that particles vary their speeds according to neighboring fish. This was first modeled by Hubbard et al. (2004).
The capelin

The capelin are known to be quite sensitive to oceanic temperature. We model this reaction with a temperature reaction function, described below.

Partridge (1982) pointed out that particles vary their speeds according to neighboring fish. This was first modeled by Hubbard et al. (2004).
Speeds

Particles now average their speeds to that of their neighbors within the zone of orientation, $O_k(t)$:

$$v_k(t + \Delta t) = \frac{1}{|O_k|} \sum_{j \in O_k} v_j(t)$$
Temperature function

The function $r$ describes the reaction to the temperature, $T$. The interval $[T_1, T_2]$ is the *preferred temperature range* which a particle tends to head into.

$$r(T) := \begin{cases} 
-(T - T_1)^4 & \text{if } T \leq T_1 \\
0 & \text{if } T_1 \leq T \leq T_2 \\
-(T - T_2)^2 & \text{if } T_2 \leq T 
\end{cases}$$
Migration model

The directional heading of each particle is now updated as follows:

$$D_k(t + \Delta t) := \left( 1 - \beta \right) \frac{d_k(t + \Delta t)}{\|d_k(t + \Delta t)\|} + \beta \frac{\nabla r(T(q_k(t)))}{\|\nabla r(T(q_k(t)))\|}$$

- Interactions
- Influence of temperature
Migration model

Particle $k$ now updates its position $\mathbf{q}_k = (x_k, y_k)^T$ by

$$
\mathbf{q}_k(t + \Delta t) = \mathbf{q}_k(t) + \Delta t \cdot \mathbf{v}_k(t + \Delta t) \cdot \left( \begin{array}{c} \cos(\phi_k(t + \Delta t)) \\ \sin(\phi_k(t + \Delta t)) \end{array} \right) + \Delta t \cdot \mathbf{C}(\mathbf{q}_k(t))
$$

- Here $\phi_k$ is the angle of $\mathbf{D}_k$
- $\mathbf{C}$ denotes the currents
- The speed is the average speed of particles within the zone of orientation
Temperature around Iceland
The temperature from February 2008:
Currents around Iceland

The simulated currents go clockwise around Iceland. The particles do not sense the current, i.e. it only translates them. The maximum speed of the current is 15 km/day, similar to the particles’ swimming speed.
Simulations

Using the above model, we simulated several years with good results. Shown are snapshots from the 2008 spawning migration:
We know that several other factors, e.g. fat and roe content, affect the behavior of migrating capelin, such as their:

- speed
- temperature preference

We have tailored a Dynamic Energy Budget (DEB) model of the inner dynamics of the capelin to the Icelandic capelin stock (Einarsson et al. 2011).
Dynamics of DEB

\[ \frac{de}{dt} = \frac{\nu}{L_m l} (f - e) \]

\[ \frac{dl}{dt} = \left\{ \begin{array}{ll} \frac{\nu}{3L_m} \frac{e-l}{e+g}, & l < e \\ 0, & \text{else} \end{array} \right. \]

\[ \frac{d\mu_H}{dt} = \left\{ \begin{array}{ll} \frac{\nu}{L_m} (1 - \kappa) e l^2 \frac{l+g}{e+g} - k_J \mu_H, & \mu_H < \mu_H^p \\ 0, & \text{else} \end{array} \right. \]

\[ \frac{d\mu_R}{dt} = \left\{ \begin{array}{ll} 0, & \mu_H < \mu_H^p \\ \frac{\nu}{L_m} (1 - \kappa) e l^2 \frac{l+g}{e+g} - k_J \mu_H^p, & \text{else} \end{array} \right. \]
Data

Use data from the Marine Research Institute and Matis.

- Compare to data from the 1999-2000 season.
- Species-specific parameters found
- Measurable quantities can be obtained from the DEB model.
Results of DEB model

The DEB model captures the weight, length, fat and roe percentage:
DEB and IBM

Gives us the timing of the onset of increased roe production. Can use this to model the capelin’s dependence of roe percentage, which plays a large role in the spawning migration of the capelin.

- Introduce a preferred speed
- Trigger a change in the preferred temperature range
Scaling behavior

Want to run further simulations to investigate the scaling behavior of the system.

Want to understand the ripple formation (next slide) which could explain the high number of neighbors.
Pattern formation
ODEs with noise

Want to add noise to the model in Birnir (2007):

\[
\begin{align*}
\dot{v}_k &= \alpha \bar{v} r \cos(\psi - \phi_k) - \alpha v_k + \xi_k \\
\mathbf{v}_k \cdot \mathbf{\phi}_k &= \alpha \bar{v} r \sin(\psi - \phi_k) \\
\dot{r}_k &= v_k \cos(\phi_k - \theta_k) \\
r_k \dot{\theta}_k &= v_k \sin(\phi_k - \theta_k)
\end{align*}
\]

where \( r_k (\cos(\theta_k), \sin(\theta_k)) \) denotes the position of fish \( k \).
ODEs with noise

Birnir (2007) showed the existence of swarming solutions which we want to simulate with noise added.
Thank you!