Assignment 1: Cellular Motility

Integrated Workshop

Due Friday, Oct. 6, 2023 at 11:59 PM EST

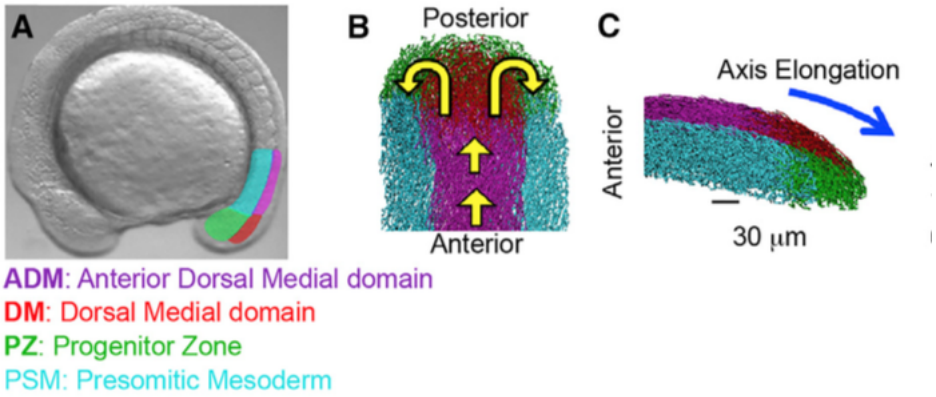

Fig. 1: A. Snapshot of the zebrafish embryo, with the four tailbud domains labeled. B. Schematic of tailbud elongation, with cells moving from the ADM into the DM and and then turning into the PSM. C. Same snapshot in B seen from the side of the embryo.

Repulsive Vicsek Simulation

In Prof. Holley's talk, we encountered the behavior of migrating spinal tissue cells in developing zebrafish embryos (see Refs. [2, 3] for the literature referenced in his talk, and Fig.1 for a description of the experimental system). In particular, we saw how cells in the tailbud (posterior end of the embryo, where the spinal column and tail will eventually be) undergo a epithelial-to-mesenchymal transition (EMT) and qualitatively change their migration strategy. To understand this transition more quantitatively, Prof Holley and his lab employed the Vicsek model, which has been used extensively to model active, collective cell migration. We will implement a simpler version of this model in MATLAB to interrogate how the model can capture some of the physics behind the EMT.

In this assignment you will modify pre-written code to implement the repulsive Vicsek-like model from Chaté *et al.*[1]. In this model, cells move in 2D with a constant speed, but have a variable velocity. The direction of the velocity of a given cell is determined by alignment

of velocity vectors, a force due to the repulsion from overlapping cells, and random noise. You will analyze the polarization as a function of the strength of the noise and repulsive interactions. For this section, you will need to download `vicsek.m`, `vicsekvelocities.m` and `assignment1_template.m`, which are provided in the Assignments folder on Canvas.

Our system will contain a set of N particles with positions $\vec{r}_i$ and velocities $\vec{v}_i$ for $i = 1, ..., N$. Each particle will be endowed with a repulsive radius r_c and an "alignment zone" of radius r_0, which influences the distance at which a particle will align its velocities with its other neighbors. Given a set of particle positions at time t, we will update the particle positions using a straightforward integration scheme:

$$\vec{r}_i(t + \delta t) = \vec{r}_i(t) + \vec{v}_i(t + \delta t)\delta t, \tag{1}$$

where δt is the time step in the simulation, and $\vec{v}_i(t + \delta t)$ is the velocity at the next time step. The velocity update is given by:

$$\vec{v}_i\left(t + \delta t\right) = v_0 \vartheta \left(\sum_{j \in S_i} \vec{v}_j\left(t\right) + \beta \sum_{j \in S_i} \vec{f}_{ij} + \eta \left|S_i\right| \vec{\xi}_i \right), \tag{2}$$

where $\vartheta\left(\vec{x}\right) = \hat{x} = \frac{\vec{x}}{|x|}$ is a normalizing function, S_i is the set of neighbors within a distance r_0 of cell i, $|S_i|$ is the number of cells in S_i, $\vec{\xi}$ is a randomly oriented unit vector, L is the size of the simulation box, v_0 is the speed of the cells, $\vec{f}_{ij}$ is the repulsive force on cell i due to cell j, and η is the magnitude of the noise. We define a spring-like repulsive force between cells i and j as follows:

$$\vec{f}_{ij} = \begin{cases} \left(1 - \frac{r_{ij}}{r_c}\right) \hat{r}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} > r_c \end{cases}.$$

Here, r_c is the diameter of a cell, r_{ij} is the center-to-center distance between cell i and cell j, and $\hat{r}_{ij} = \frac{\vec{r}_i - \vec{r}_j}{|\vec{r}_i - \vec{r}_j|}$ is the unit vector that points from cell j to cell i.
So the general rule for creating a simulation is

1. Initialize the positions $\vec{r}_i(0)$.
2. Initialize the velocities $\vec{v}_i(0)$.
3. For a given number of time steps $n = 1, ..., N_T$, do the following:
   (a) Update velocities $\vec{v}_i(t + \delta t)$ based on Eq. (2).
   (b) Update positions $\vec{r}_i(t + \delta t)$ based on Eq. (1).
   (c) Repeat until n reaches N_T.
4. End here.

Here, we provide some guidance in setting up the simulation using the provided code.

1. Open `vicsekvelocities.m`. The inputs and outputs have been defined at the top of the script and described in the comment block.

2. For each cell, i, we will need to calculate the sum of the velocities of cells within a circle of radius r_0, the sum of the repulsive forces ($\sum_{j=1}^{N} \vec{f}_{ij}$), and a noise term. For the noise term, we will need to calculate the number of cells within r_0, which we call $|S_i|$. Initialize `sum_vs` and `Fi` to an $N \times 2$ matrix of zeros. Initialize `Si_norm` to an N-length vector of zeros. `Si_norm` will keep track of $|S_i|$.

3. Use the for loop to calculate `sum_vs`, `Fi`, and `Si_norm`.

   (a) Code has been included to calculate `dists`, a vector of the distances of each cell j ($j = 1, ..., N$) to cell i, accounting for periodic boundary conditions. The distance between cell i and cell j is callable as `dists(j)`. The matrix of distance vectors $\vec{r}_{ij}$ is assigned to `rijs`. Once again this accounts for periodic boundary conditions.

   (b) Calculate the set of cells within a radius `r0` of cell i and assign them to `Si`. **Hint:** you can use a boolean comparison to create this set.

   (c) Assign the number of cells within `r0` to `Si_norm(i)`.

   (d) Calculate `sum_vs(i,:)` using `Si` and `vs`. **Hint:** as in Eq. (2) the sum is only over cells in `Si`

   (e) Calculate the $N \times 2$ matrix of forces $\vec{f}_{ij}$ from `dists`, `rc`, and `rijs`. **Hint:** `rijs./dists` is a matrix of unit vectors. You can use a boolean comparison to assign the piece-wise nature of this function. Make sure the contribution to the force of the cell on itself is zero (i.e. $\vec{f}_{ii} = 0$).

   (f) Assign the sum along the first axis to `Fi(i,:)`.

4. After the loop, a matrix of unit vectors has been assigned to `noise`. This utilizes the function `normer` which performs a row-wise normalization to a matrix.

5. Finally, `vs` is calculated using `normer` and the values calculated above.

Setting up the simulation

Also provided is the function `vicsek.m`, which will run the Vicsek model simulation for a given number of time points N_T, with a given set of input parameters. To run the simulation, you can specify the relevant parameters in a separate script, and call the function `vicsek` with those input parameters (see the example script `assignment1_template.m` provided for a demonstration of how to call the `vicsek` function). The function will output `xtotal` and `ytotal`, which are all x and y values of all particles throughout the duration of the simulation, as well as the global polarization $\Phi(t)$ values (see below) during the simulation. Throughout the assignment, we will use the following parameters:

Parameter	Variable Name	Simulation Value	Description
N	`N`	100	particle number
L	`L`	1.0	box length
r_0	`r0`	$2r_c$	attractive zone
v_0	`v0`	0.05	speed
δt	`dt`	0.005	time step
β	`beta`	10000	repulsion strength

We will change the packing fraction ϕ and noise level η. Since the packing fraction is defined as the total particle area divided by the box area, i.e.

$$\phi = \sum_{i=1}^{N} \frac{\pi r_c^2}{4L^2} = \frac{N\pi r_c^2}{4L^2} \tag{3}$$

and thus the particle size is determined by the packing fraction, i.e. through

$$r_c = 2L\sqrt{\frac{\phi}{N\pi}}. \tag{4}$$

Once the `vicsekvelocities.m` file is set up, you will first calculate the global polarization Φ, which is defined by

$$\Phi = \frac{1}{v_0}|\langle \vec{v}_i(t)\rangle_{i,t}| = \langle \Phi(t)\rangle. \tag{5}$$

which is just the magnitude of the average velocity vector over all N particles and over all times in the simulations, normalized by the cell speed v_0. Here $\Phi(t)$ is the polarization as a function of time during the simulation; Φ is therefore the *time-averaged* polarization.

The assignment

1. In a new script, calculate the time-averaged polarization Φ as a function of noise strength η and packing fraction ϕ; use $\phi = 0.25, 0.375$, and 0.5. For the noise strength, use values between $\eta = 0.5$ and $\eta = 0.7$, with points separated by 0.01. **Make a plot** of Φ vs η for the three different values of ϕ from simulations run for $N_T = 10000$ time steps. For your calculation of Φ, only use velocities from the second half of the simulation to minimize transient effects that occur at thebeginning of the simulations. Name the script used to calculate the polarization plot `LASTNAME-FIRSTNAME-polarization.m`, and the plot itself `LASTNAME-FIRSTNAME-polarization_plot.png`.

2. In a new script, calculate the mean squared displacement (MSD) averaged over all particles in the system for different values of the noise; use $\eta = 0.4, 0.5, 0.6, 0.7$ and 0.8, and *at fixed packing fraction* $\phi = 0.5$. Use a simulation with at least $N_T = 5 \times 10^4$ steps, and neglect data from the first 2500 steps. **Make a plot** of MSD vs. the time window size Δt for these different values for the noise. As in assignment 0, calculate the diffusion coefficient D and power law α from the plotted data; **make a plot** of D and α as a function of noise η. Name the script used to calculate these two plots `LASTNAME-FIRSTNAME-msd.m`, and the plots `LASTNAME-FIRSTNAME-msd_plot.png` and `LASTNAME-FIRSTNAME-coefficients_plot.png`

3. **In a separate document**, write a paragraph or two that addresses the following questions:

   (a) What do you observe when you increase the noise strength of the particles at fixed packing fraction ϕ? What about when you increase ϕ for fixed η? Why?

   (b) In the zebrafish embryo, there is an important role played by an epithelia-to-mesenchymal transition (EMT) in the elongating tailbud, where directed, ordered cells suddenly behave less collectively and more disordered; why would the Vicsek model be a useful model in studying the EMT?

   Name the document `LASTNAME-FIRSTNAME-writeup.x`, where the file can be a .pdf, .doc or .txt file.

Submission

Upload all three files (two scripts and write-up document) as well as the three required plots to the Assignments section of Canvas.

References

[1] H. Chaté, F. Ginelli, G. Grégoire, and F. Raynaud. Collective motion of self-propelled particles interacting without cohesion. *Physical Review E*, 77(4):046113, 2008.

[2] D. Das, V. Chatti, T. Emonet, and S. A. Holley. Patterned Disordered Cell Motion Ensures Vertebral Column Symmetry. *Developmental Cell*, 42(2):170+, JUL 24 2017.

[3] D. Das, D. Julich, J. Schwendinger-Schreck, E. Guillon, A. K. Lawton, N. Dray, T. Emonet, C. S. O'Hern, M. D. Shattuck, and S. A. Holley. Organization of embryonic morphogenesis via mechanical information. *Developmental Cell*, 49(6):829 – 839.e5, 2019.