Name: Sign:

Exercise 3
III: Pattern Formation in Reaction-Diffusion Systems

The goal of this exercise is to develop an intuition about simple reaction-diffusion systems and explore how they behave.

We will study the following reaction diffusion system:

$$\frac{da}{dt} = c_1 a + c_2 b - r_a a^3 + D_a \nabla^2 a \tag{1}$$

$$\frac{db}{dt} = c_3 a + c_4 b - r_b b^3 + D_b \nabla^2 b \tag{2}$$

a and b are the deviation from the equilibrium concentration of two chemicals A and B in the tissue. We will assume that A is an autocatalytic activator and B an inhibitor. The constants c_i represent their interaction; we will use $c_1 = c_3 = 1$, $c_2 = c_4 = -1$ as a starting point: A catalyses its own production and is inhibited by B, B is produced when A is present but inhibited by itself. r_a and r_b represent nonlinear breakdown of A and B. D_a and D_b are their respective diffusion coefficients, which are assumed to be different.

1 Part 1: simulation in one dimension

Simulate the above system for a one-dimensional closed chain of cells. Set $c_1 = c_3 = 1$, $c_2 = c_4 = -1$, $r_a = r_b = 0.1$, $D_a = 1$ and $D_b = 3$.

The reaction-diffusion equations can be approximately solved using Euler's method: $u'(t,x) = f(x)$ is replaced with $u(t+h,x) = u(t,x) + hf(t,x)$, with a small h. Diffusion can be handled with the discretization

$$\nabla^2 u(t,x) \approx u(t,x-h) + u(t,x+h) - 2u(t,x)$$

in 1D and

$$\nabla^2 u(t,x,y) \approx u(t,x-h,y) + u(t,x+h,y) + u(t,x,y-h) + u(t,x,y+h) - 4u(t,x,y)$$

in 2D. This way the differential equation is turned into a difference equation that can be solved iteratively. People with a background in numerical analysis might want to do it properly using a more exact or stable method; for the purposes of this exercise it is enough to study the rough qualitative behavior of the system.

1.1 Matlab Implementation

This exercise is best done in Matlab, although you can implement it in any software you like as long as you can visualise the results.

A simple matlab framework to solve a similar kind of differential equation $\frac{du}{dt} = k\sin(u) + D_u\nabla^2 u$ for a random initial state might look like:

```
N=50;                                       % Number of cells
Du=1;                                       % Diffusion coefficient
k=2.3;                                      % Constant
u=zeros(N,1)+.01*rand(N,1);                 % Random initial state
ff=[];                                      % History
for t=0:3000
    ud1=[u(2:N)' u(1)]';                    % u(x+1) with wraparound
    ud2=[u(N) u(1:(N-1))']';                % u(x-1) with wraparound
    deltau=k*sin(u) + Du*(ud1+ud2-2*u);     % Calculate du/dt
    u=u+deltau*.1;                          % Update value of u
    ff=[ff; u'];                            % Store new value
    imagesc(ff);                            % Plot the values
    drawnow
end
```

This creates 50 cells, whose levels `u` are set on the third line to random values in the interval $0 - 0.01$. The fourth line introduces a variable `ff` to hold previous values of `u`, making it possible to plot the evolution over time. Inside the loop a bit of Matlab trickery creates two helper variables `ud1` and `ud2`, which are the values of `u` translated one cell to the left or right. This is then used to calculate the time derivative, which in turn updates the values `u` with a timestep of 0.1. Afterwards, the resulting values are stored and plotted as a colored field with the time direction vertically and the space direction horizontally.

It is worth noting that `imagesc(u)` plots values with colors, selected so that the maximum value is red (assuming you have the default colormap) and the minimum blue. At the start, most values are near zero so pixels that were once red or blue will during the run turn green as the scale changes. If you find this plot confusing, try `mesh(ff)` or `plot(u)` instead. The speed of the program is *vastly* increased if you do not plot every timestep.

Adapting the above code for the reaction-diffusion system should be fairly simple.

1.2 Questions

Implement the 1D system described in equations (1) and (2), and then answer the following questions:

Question 1: Start the simulation with $a = b = 0$ for all cells, except for one where you set $a = 0.14$. What happens over time?

Question 2: What happens if you add some noise to all cells in the initial state? Is the period of the pattern altered?

Question 3: Change D_a and D_b. What happens with the period of the pattern? Under what circumstances do you not get any patterns at all? (Note that too large values will make the simulation numerically unstable)

Question 4: Try changing the value of c_2 from -1 to -2. What happens? Why? Try intermediate values and watch the behavior - does there exist any intermediate state?

Question 5: What happens when you increase c_1?

Question 6: Sketch a "phase-portrait" of the behavior of the system in the $c_1 - c_2$ plane with values between 0 and 2. There are a number of qualitatively different behaviors. What are they, and are they sharply delimited from each other? It might be useful to extend the program to walk through different combinations and plot them.

2 2D Simulation

Set up the same simulation in 2D. What you mainly need to change is the initialisation of the variables, the calculation of ∇^2 to the 2D version and the way of plotting. Start with a square region, perhaps 50x50.

Question 7: Start from a random state. What pattern emerges?

Question 8: Why does not the pattern behave in the same regular way as in the 1D case? Try using a fairly narrow rectangular region (such as 50x20 or 50x10) instead of a square region.

Question 8: What happens if you start with a single cell with high A? With two cells some distance from each other?

3 Own experiments (optional)

Try what happens when the various parameters are different for different cells. What happens if you make them activity dependent?

More complex patterns can be created when one pair of reaction-diffusion equations control another. For example, the value of one activator might determine the values of the constants in the second activator-inhibitor pair (for example c_1, c_2 or the diffusion constants) of the other, creating regions with different dynamics or driving oscillations in the second activator. For even more complexity, the second pair can be allowed to influence the first. The possibilities with just two pairs of activator-inhibitors are nearly endless.

What happens if the region is not rectangular but has some other shape?

What happens if the domain grows over time? Can this make previously stable patterns unstable?

Instead of using a system of an activator and inhibitor, what happens if you use an activator and a depleted substrate? The activator is autocatalytic and depletes the substrate, which is replenished at a slow rate. Does this change anything? Start again with a 1D system and then try out what happens in 2D as well. What happens if the interaction is nonlinear (e.g. has ab terms)?