Experimental Soft Matter Physics
Module 1: Turing Patterns Lab

Learning Objectives

1. To view the emergence of spatially organized patterns in biological systems through interdisciplinary and historical lenses, putting physics into conversation with other scientific disciplines.
2. To explore the basic properties of reaction-diffusion systems, emphasizing the role played by between instabilities and the formation of Turing patterns.
3. To gain additional experience with basic scientific programming.

Introduction

Throughout the biological world we find an unmistakable repetition of recognizable structures. When we look through a microscope, for example, we find cellular structures. In larger organisms we find specialized organs, such as the brain, the stomach, or the heart, that have specific physiological functions. Likewise, a variety of branched networks and symmetrical forms can be found in many different plants and animals. The presence of similar structures in so many different organisms can be investigated from both evolutionary and genetic perspectives. Both of these perspectives, which involve widely separated time scales, provide significant insight. Evolution and gene expression are not, however, the only processes can generate nontrivial spatial patterns in biological systems. In this lab, we will explore a surprising mechanism that is still under investigation today.

In 1917, D'Arcy Wentworth Thompson published a highly influential book, *On Growth and Form*, arguing against an over-reliance on evolutionary explanations for structures like those mentioned above. Thompson's emphasis on the importance mathematical and physical principles, which was highly unusual for its time, inspired many scientists in the years that followed. Among these was Alan Turing, a mathematician known for his pioneering work on cryptography and computer science. In 1952, Turing published a remarkable paper, "The Chemical Basis of Morphogenesis", exploring the properties of what are now known as *reaction-diffusion systems*. A generic reaction-diffusion model describing two chemical concentrations $U(x,t)$ and $V(x,t)$ has the form,

$$\frac{\partial}{\partial t}U = f(U,V) + D_U \frac{\partial^2}{\partial x^2}U, \tag{1a}$$

$$\frac{\partial}{\partial t}V = g(U,V) + D_V \frac{\partial^2}{\partial x^2}V. \tag{1b}$$

Here, $f(U,V)$ and $g(U,V)$ are reaction terms describing how U and V interact, while the diffusion terms on the right describe the tendency of molecules to move down concentration gradients, i.e., from regions of higher concentration into regions of lower concentration. Generally, we expect that diffusion tends to smooths out spatial variations in concentration, pushing the model towards states of featureless uniformity. Turing found, however, that diffusion can have a destabilizing effect, generating spatially periodic concentration patterns now known as *Turing patterns*. These patterns have been a controversial topic in biology and, a half century later, researchers are still working hard to explore their feasibility in real systems.

Creating Turing patterns on the computer

Our first task in this lab is to confirm that reaction-diffusion systems can indeed generate spatial patterns. We will do this using Matlab code that solves Eqns. (1) for a particular model of nonlinear interactions between U and V. The model we'll be working with,

$$f(U,V) = a - (b+1)U + U^2V, \tag{2a}$$

$$g(U,V) = bU - U^2V. \tag{2b}$$

has an interesting history, which we will discuss later in the course, and has itself been highly influential as well. For now, let's assume $a = 2$ and $b = 3$. (Later, you'll explore how other choices affect the results.) You can confirm, by setting $f(U^*, V^*) = 0$ and $g(U^*, V^*) = 0$ in Eqns. (4), that this model has just one spatially uniform steady state: $U^* = a$, $V^* = b/a$.

The numerical technique we'll be using is an example of the finite differences method. We don't need to examine the method in detail but, in case you're curious, the basic idea involves dividing space and time into discrete steps and replacing all derivatives in space and time with discrete approximations. This converts the whole model, which was a nonlinear partial differential equation, into a series of matrix equations that are solved numerically. The code I'm providing you with takes care of all of the details here, including a randomly chosen initial condition close but not equal to the spatially uniform steady state identified above, i.e., $U(x,0) \sim a$ and $V(x,0) \sim b/a$. All you will need to do is choose the model parameters a and b, as well as a number of time steps.

To get started, download the file "Turing.m" and place it wherever you are storing files related to this course. I like to create separate folders for Matlab code, but this isn't necessary. When you open up Matlab, you'll need to tell it where the code is. There are a couple of ways to do this:

1. You can navigate to the appropriate folder using the "Browse for folder" button, which looks like a yellow folder with a green arrow on it and is in the upper left part of the window.

2. If you've been to this folder recently, Matlab may already know its location. In this case, you select the appropriate folder from a drop-down list that appears when you click a down arrow on right side of the Matlab window.

To run the code for $a = 2$ and $b = 3$, simply type:

```
>>  [u,v,x,t] = Turing(2,3,1e2);
```

Matlab functions, like this one, have organized lists of input and output variables that occur in particular order. The second input variable, for example, always determines the value of the parameter b (set equal to 3, in this case). If you're wondering what these different input and output variables are, you can type

```
>>  help Turing
```

To look at how the U variable at a particular position evolves with time, type

```
>>  plot(t,u(:,40),'.');
```

Here, we chose the 40th position, but you could try others. This plot should look like a decaying oscillation, which suggests that the u variable at this position converges over time to some preferred value. Do recognize the value? Try this with v as well and, again, see if you recognize the value.

Now look at the spatial variation of U and V at the final time (after 100 time steps):

```
>>  plot(x,u(end,:),'.',x,v(end,:),'.');
```

These different plots should all say more or less the same thing. The results may change if we try a different value of b. As an example, try

```
>>  [u,v,x,t] = Turing(2,4,4e3);
```

Note that we increased the number of time steps significantly here. If you look at the time series for u or v at a particular position, you'll see why this was necessary. Bw sure to look at the spatial variation of U and V at the final time (after 4000 time steps). See anything interesting?! (You should find, seemingly out of nowhere, a periodic pattern emerging... This is your first Turing pattern!) We'll take closer look at the difference between these two choices for b in the next section.

Fingerprints of pattern-forming instabilities

As we discussed in class, Turing's mathematical analysis focused on what happens near spatially uniform steady states. In particular, for small departures from such states, $u = U - U^*$ and $v = V - V^*$, reaction-diffusion models takes the generic linear form,

$$\frac{\partial}{\partial t}\begin{bmatrix} u \\ v \end{bmatrix} = M\begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} D_U & 0 \\ 0 & D_V \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}. \tag{3}$$

In the absence of diffusion, the fate of small departures from the steady state is determined entirely by the matrix M. Whenever M has both negative trace and positive determinant,

$$\tau = M_{11} + M_{22} < 0, \tag{4a}$$

$$\Delta = M_{11}M_{22} - M_{12}M_{22} > 0, \tag{4b}$$

we describe the steady state as *asymptotically stable*: all nearby phase trajectories converge at the steady state. For the particular nonlinear model introduced in the previous section, one finds:

$$M = \begin{bmatrix} b-1 & a^2 \\ -b & -a^2 \end{bmatrix}. \tag{5}$$

Show, using this result, that the steady state is asymptotically stable whenever b is less than $1+a^2$. Is this condition satisfied for either of the two situations explored in the previous section ($a = 2$ and two different b values)?

At the heart of Turing's paper is the unexpected discovery that diffusion can destabilize the asymptotically stable steady state described above. This occurs whenever the following condition is satisfied:

$$(D_V M_{11} + D_U M_{22})^2 > 4D_U D_V \Delta. \tag{6}$$

Combining this with the information given above, we find that Turing instability occurs in the particular model we're using whenever

$$b < b_H = 1 + a^2, \tag{7a}$$

$$b > b_T = (1 + a\sqrt{D_U/D_V})^2. \tag{7b}$$

The first of these conditions is the one you derived above. The second is derived by manipulating Eqn. (6). (You can check this, if you're interested!)

To see how these conditions work together, try plotting both curves on the same axes and adjusting the plot range to make the curves easier to see:

```
>>  a = 0:0.01:3;
>>  Du = 1; Dv = 5;
>>  plot(a,1 + a.^2, a,(1 + a*sqrt(Du/Dv)).^2);
>>  axis([0,3,0,5]);
```

The spatially uniform steady state is asymptotically stable for all a and b combinations that fall below the blue curve. Turing instabilities are found for those combinations that fall below the blue curve but also above the orange curve. (Above the blue curve things get more complicated.) What does this tell you about the two combinations of a and b explored in the previous section. Is a Turing pattern predicted for either of these? To test your interpretation, try out some other combinations of a and b and keep track of where they fall relative to the two curves you just plotted or, equivalently, where they fall relative to the instability conditions listed above.

Additional opportunities

As we discussed in class, one of the more important aspects of the Turing instability mechanism is its spatial structure. That is, Turing patterns emerge from a selection process in which spatially periodic perturbations with preferred wavelengths grow, while all other decay exponentially. If you're wondering how to explain the wavelengths produced by your Matlab code, recall that the fastest growing mode has a wavenumber k that satisfies the following expression:

$$k^2 = \frac{1}{2}\left(\frac{M_{11}}{D_U} + \frac{M_{22}}{D_V}\right)^2. \tag{8}$$

At the critical value of b for Turing instability, b_T, this expression reduces to $k_c^2 = a/\sqrt{D_U D_V}$. See if you can derive this using the expression for b_T given in Eqn. (7) and, then, see if it helps you make sense of the patterns selected by your Matlab code. Remember that k is related to wavelength by a factor of 2π and consider also the size of your spatial domain. Other opportunities for further exploration are listed below:

- If you're interested in the history of overlap between biology and mathematical reasoning, you might enjoy learning more about D'Arcy Wentworth Thompson and the specific arguments presented in his *On Growth and Form*.
- Turing's original paper is a challenging read but you might find it interesting to explore his choice of words and see which parts of his paper are the most accessible.
- The nonlinear model we've used in this lab will be discussed in more detail later in the course. In the meantime, you might find it interesting to read about another highly influential nonlinear model, now known as the *Gierer-Meinhardt model*. If so, take a look at the classic 1972 paper by Gierer and Meinhardt. Note in particular his emphasis on short-range activation and long-range inhibition, which offers a physically intuitive reinterpretation of the need for $K^2 > 0$ in Eqn. (8).