Reaction-diffusion equations

Marc R. Roussel

November 29, 2005

1 Introduction

1.1 The rate of diffusion

In chemical systems which are not well mixed, we have to take into account the fact that chemicals move through space due to diffusion. We begin these notes with a short account of the laws of diffusion.

The fundamental quantity in the study of diffusion is the **flux**, i.e. the amount of a chemical which passes near a point in space per unit area per unit time. The SI unit of flux would therefore be $\mathrm{mol\,m^{-2}s^{-1}}$. Flux is a vector: We can talk about fluxes in each of the Cartesian directions in space. The best way to think about flux is to imagine a small surface of area A oriented perpendicularly to one of the coordinate axes, say the x axis. Now examine figure 1. The number of moles of the chemical whose flux along the x axis is J_x entering the volume between the two surfaces per unit time is $J_x(x)A$, while the number of moles leaving this volume is $J_x(x+\Delta x)A$. The change in the number of moles per unit time is therefore $A\left[J_x(x+\Delta x) - J_x(x)\right]$. To get the change in the concentration, we divide the change in the number of moles by the volume, which is just $A\Delta x$:

$$\frac{\partial c}{\partial t} = \frac{J_x(x+\Delta x) - J_x(x)}{\Delta x} = -\frac{J_x(x) - J_x(x+\Delta x)}{\Delta x}.$$

If we now let $\Delta x \to 0$, we get

$$\frac{\partial c}{\partial t} = -\frac{\partial J_x}{\partial x}, \tag{1a}$$

or, in three dimensions,

$$\frac{\partial c}{\partial t} = -\nabla \mathbf{J}, \tag{1b}$$

where

$$\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$$

is the usual vector differential operator.

Equations 1 are general equations for the rate of transport of a chemical through space which hold *regardless of the nature of the flux*. We will use these equations for diffusive fluxes, but we could equally well use them to describe systems in which there is a flux due to fluid motion (e.g. a current in a river). We just need an appropriate expression for the flux.
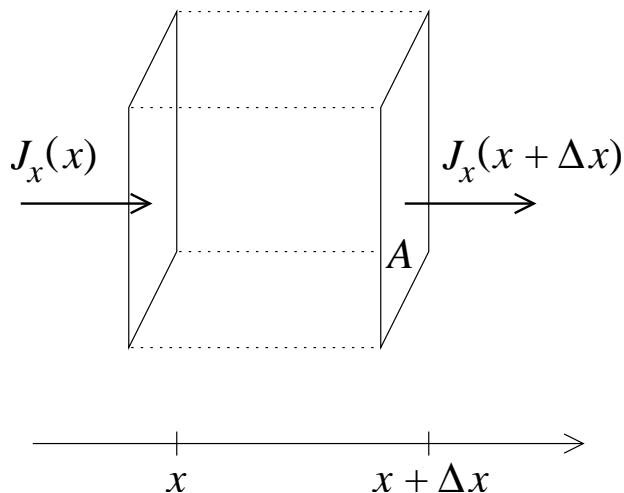
Figure 1: A small volume of space bounded by two surfaces of area A each perpendicular to the x axis and separated by a distance Δx.

What can we take to be an appropriate expression for the diffusive flux? This is actually a surprisingly complicated question. However, we can give a simple answer which is good enough for most purposes. Diffusion is, at its core, a simple effect: Suppose that there is more of our chemical on the left-hand side of our imaginary surface than there is on the right. Then, random thermal motion of the particles in the system should result in a net left-to-right flux through the surface. Intermolecular forces *may* prevent this simple picture from being exactly correct, but in a dilute solution where these forces are unimportant, we expect the flux to oppose the concentration gradient. In other words,

$$J_x = -D_x(c)\frac{\partial c}{\partial x},$$

where $D_x(c)$ is a nonnegative functional[1] of its argument known as a **diffusion coefficient**. This equation is a form of **Fick's first law of diffusion**. If D_x depends only weakly on c near the conditions of interest, we can treat D_x as a constant,[2] say D_x^0. Furthermore, ordinary solutions and gases are isotropic (every direction in space is the same) so that there is just one diffusion coefficient $D = D_x^0 = D_y^0 = D_z^0$. When these approximations are valid, we get

$$J_x = -D\frac{\partial c}{\partial x},$$

or

$$\mathbf{J} = -D\nabla c.$$

[1] A functional is a mapping of functions onto numbers (e.g. the real numbers. By saying that D_x is a functional of c, we are saying that this quantity could in principle depend on c, its derivatives, etc.

[2] If D_x were an ordinary function of c, we would take a Taylor expansion and neglect the higher-order terms. However, since there isn't a straightforward way to define the derivative of a functional with respect to a function, this hand-waving argument will have to do.

If we combine these equations with equations 1, we get **Fick's second law**:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}$$

in one spatial dimension, or

$$\frac{\partial c}{\partial t} = D\nabla^2 c$$

in higher-dimensional spaces. Surprisingly given the number of assumptions and approximations made, Fick's second law does a very reasonable job of representing diffusion in many situations. This will be the starting point for today's discussion, although we should be aware that everything after equations 1 is a hypothesis to be validated for the system under study.

1.2 Reaction-diffusion equations

What happens if we have reactions *and* diffusion? It turns out that the net effect of the two processes is just the sum of the individual rates of change. Thus, if a reaction or set of reactions leads to reaction rate terms R, then

$$\frac{\partial c}{\partial t} = D\nabla^2 c + R.$$

For obvious reasons, this is called a reaction-diffusion equation. Reaction-diffusion equations are members of a more general class known as **partial differential equations** (PDEs), so called because they involve the partial derivatives of functions of many variables. In the case of a reaction-diffusion equation, c depends on t and on the spatial variables.

To fully specify a reaction-diffusion problem, we need the differential equations, some initial conditions, and **boundary conditions**. The initial conditions will be initial values of the concentrations over the domain of the problem. The boundary conditions are a new element: In general, there will be special conditions at the boundary of the domain such that the differential equation doesn't apply there. For instance, in chemical systems, the boundary usually represents the walls of a container. These walls are impermeable to the chemicals, so we need a condition which says that the chemicals can't leak through these walls. Such boundary conditions are called **no-flux boundary conditions**. Mathematically, the no-flux condition is

$$\mathbf{J}\cdot\mathbf{n} = 0,$$

where $\mathbf{n}$ is a vector normal to the boundary.

We will shortly be doing some analysis with a reaction-diffusion system, but first it's useful to learn something about the simulation of these systems. We will limit our discussion to one-dimensional systems, although it's not all that hard to move up to higher-dimensional systems. The simplest way to integrate reaction-diffusion equations is to use the **finite-difference** method. In this method, we store concentrations at (say) $N+1$ **mesh points** spaced by Δx and numbered 0 to N, and estimate the second derivative of the concentration at every point using these values. The result of carrying out this procedure is a **discretization** of the equations. The discretization is a set of ODEs which reduce to the original PDE provided the mesh points are sufficiently close together.

For a general point i ($i \neq 0$ or N), we have

$$D\frac{\partial^2 c}{\partial x^2} = D\frac{\partial}{\partial x}\left(\frac{\partial c}{\partial x}\right).$$

The partial derivative to the left of point i is approximated by

$$\left.\frac{\partial c}{\partial x}\right|_{i-1/2} \approx \frac{c_i - c_{i-1}}{\Delta x}, \tag{2a}$$

while the partial derivative to the right is

$$\left.\frac{\partial c}{\partial x}\right|_{i+1/2} \approx \frac{c_{i+1} - c_i}{\Delta x}. \tag{2b}$$

The best way to think about these two quantities is that they are estimates of the spatial derivatives of c at points midway between two of the mesh points. These are therefore estimates of the derivatives at two points which are themselves spaced by Δx. The rate of diffusion at point i is therefore approximately equal to

$$D\frac{\partial^2 c}{\partial x^2} \approx \frac{D}{\Delta x}\left(\left.\frac{\partial c}{\partial x}\right|_{i+1/2} - \left.\frac{\partial c}{\partial x}\right|_{i-1/2}\right) = \frac{D}{(\Delta x)^2}\left(c_{i+1} - 2c_i + c_{i-1}\right).$$

Multiplying this expression by the diffusion coefficient D gives the rate of diffusion.

What about the boundaries? Suppose that we have no-flux boundaries. Then, we can use equation 2b to estimate the flux on the right of point 0 (by multiplying this expression by $-D$), but the flux to the left is zero. Using the same reasoning as above, we get

$$-\left.\frac{\partial J}{\partial x}\right|_0 \approx -\frac{1}{\Delta x}\left(J_{1/2} - J_{-1/2}\right) = -\frac{1}{\Delta x}\left(-D\frac{c_1 - c_0}{\Delta x} - 0\right) = \frac{D}{(\Delta x)^2}\left(c_1 - c_0\right).$$

Similarly, at point N,

$$-\left.\frac{\partial J}{\partial x}\right|_N \approx -\frac{1}{\Delta x}\left(J_{N+1/2} - J_{N-1/2}\right) = -\frac{1}{\Delta x}\left(0 - (-D)\frac{c_N - c_{N-1}}{\Delta x}\right) = \frac{D}{(\Delta x)^2}\left(c_{N-1} - c_N\right).$$

Example 1.1 As mentioned earlier, the logistic differential equation is used extensively in population biology to model populations whose growth is limited by environmental factors. In fully dimensional form, this equation is

$$\frac{dA}{dT} = rA(1 - A/K),$$

where A is the population density, r is the specific growth rate (analogous to a first-order rate constant) at low population densities, and K is the carrying capacity of the environment, i.e. the maximum population density which can be supported. Of course, living organisms do

migrate, and this process is often modeled by diffusion. Thus we get the logistic reaction-diffusion equation

$$\frac{\partial A}{\partial T} = D\frac{\partial^2 A}{\partial X^2} + rA(1 - A/K).$$

In this equation X represents the spatial coordinate. Obviously, in a realistic model, we would probably consider a two-dimensional domain.

To facilitate our analysis, we will put this equation in dimensionless form. Start with A and T:

$$
\begin{aligned}
a &= A/K, \\
\text{and} \quad t &= rT. \\
\therefore \frac{\partial a}{\partial t} &= \frac{D}{r}\frac{\partial^2 a}{\partial X^2} + a(1 - a).
\end{aligned}
$$

From the definition of the diffusion coefficient, it's not too hard to see that it has units of m^2/s. D/r therefore has units of m^2, and the square root of this quantity has units of m. This provides us with a transformation for the spatial variable X:

$$x = X\sqrt{r/D}. \tag{3}$$

$$\therefore \frac{\partial a}{\partial t} = \frac{\partial^2 a}{\partial x^2} + a(1 - a).$$

The diffusion coefficient just disappears from our equation. It turns out that we can always do this: By choosing an appropriate scaling for the spatial variables, one of the diffusion coefficients (in a system with multiple chemical species) can be set to 1.

Suppose that we want to use this equation to model the spread of a population on an island to which it has been introduced. A small colony is established at one end of the island at $t = 0$. We could represent that by a simple square pulse:

$$a(x,0) = \begin{cases} 1 & \text{for} \quad x < \rho, \\ 0 & \text{for} \quad x \geq \rho. \end{cases}$$

If our creatures can't swim, then no-flux boundaries are appropriate.

Let's say, for the sake of argument, that the island is 10 km long, that $r = 5\,\text{y}^{-1}$, and that $D = 5\,\text{km}^2/y$, values which are reasonable for small mammals, then the length of the island in our rescaled variables is, using equation 3,

$$L = (10\,\text{km})\sqrt{(5\,\text{y}^{-1})/(5\,\text{km}^2/y)} = 10.$$

If we use $N = 100$ mesh points, then $\Delta x = 10/100 = 0.1$.

Xpp can be used to integrate the discretized ODEs, using a few facilities which have been built into it for the purpose. The following xpp input file illustrates the relevant techniques:

```
# Logistic reaction-diffusion equation with no-flux boundaries
da0/dt = (a1-a0)/deltax2 + a0*(1-a0)
da[1..99]/dt = (a[j+1]-2*a[j]+a[j-1])/deltax2 + a[j]*(1-a[j])
da100/dt = (a99-a100)/deltax2 + a100*(1-a100)

param deltax=0.1
!deltax2=deltax^2

a[0..10](0) = 1
a[11..100](0) = 0

@ MAXSTOR=1000000, DT=0.005

done
```

The differential equations are written using pseudo-arrays. These are much more limited than the arrays in real programming languages, but they are sufficient to do what we need to do here. To use pseudo-arrays, you need to define all the variables in the array (in this case, the a_i's for which we have a set of differential equations) in numerical order. In other words, it would *not* have done to write the differential equation for a_0 at the end, nor would it be OK to insert other differential equations between those for the a_i's. Note that the right-hand side of the general differential equation uses the index j to designate members of the pseudo-array. You *must* use this letter for your array indices in differential equations.

The rest of the file is most fairly obvious, including the specification of the initial condition which also uses the pseudo-array notation. In order to avoid computing $(\Delta x)^2$ every time it is used, we define a derived parameter deltax2 which has this value. Note the syntax here. This tells xpp that if we change deltax, we want deltax2 to be recomputed according to this formula.

If we now run xpp, we can of course plot each of the a_i's vs t, but what we really want to do is to see how the whole distribution evolves as a function of time. To do this, click on Viewaxes→Array, and fill in the values in the dialog box which appears as follows:
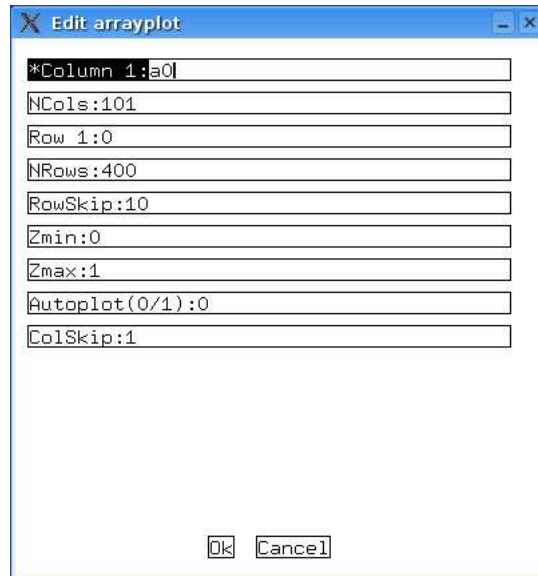
`Column 1` is the first of your variables to plot. `Ncols` is the total number of variables. `Row 1` is the row of data (from the data browser) at which to start. Setting this to zero starts at $t = 0$. `Nrows` and `RowSkip` are set together to determine the number of time points which will be used to generate the graph. It doesn't make any sense to use thousands of time points since most output devices at normal resolutions and graphic sizes won't use this many points. A few hundred points will do. In our case, we integrated for the default 20 time units at a time step of 0.005, so we should have generated $20/0.005 = 4000$ data points. (Actually, it's 4001 if you think about it for a minute, but the difference really doesn't matter here.) We'll therefore set `Nrows` to 400 and `RowSkip` to 10, which will plot every tenth row, from zero to 4000. `Zmin` and `Zmax` are the minimum and maximum values of our variables. I'm not sure what `Autoplot` does, and the `xpp` documentation seems to be mum on this point. Finally, `Colskip` plays a similar role to `RowSkip`, but for the columns of data, i.e. for the a_i's. The result is a space-time plot of the solution, with colors indicating the variable values. Figure 2 shows the result.[3] Note that you may have to hit the `Redraw` button in the window which pops up to see the result.

It is important to learn how to read space-time plots since they are one of the main tools used to visualize solutions of reaction-diffusion equations. Let us therefore look at the figure carefully. At $t = 0$ (the top of the graphic), we placed a high population near $x = 0$ (to the left). We therefore see a dark area in the top left corner, with white areas to the right where there are initially no critters. The population at the point of invasion actually decreases at first, because of diffusion into the unoccupied interior of the island. We soon have some population right across the island. This population thereafter continues to grow toward its carrying capacity.

---

[3] I printed the figure out in grey scale, using the value -1 for the `Render` option in the `Print` dialog box. The boxes you have to fill in to print out one of these graphs have reasonably obvious meanings, except for `Render`. In addition to grey scale, you can choose a red-blue encoding (0) or the red-yellow-green-blue encoding which you see on your computer screen (1), or another encoding which is similar to 1, except that the top and bottom values map to the same color. This is useful for plotting periodic quantities.

Figure 2: Space-time plot of the solution of the logistic reaction-diffusion equation. The details of the simulation are given in the text. The values on the x axis are actually indices into our pseudo-array rather than actual values of x. The latter are recovered by multiplying the indices by Δx.

2 Stability analysis of reaction-diffusion equations

As you can probably guess by now, the next step is to develop methods for the stability analysis of reaction-diffusion equations. It's much easier to do this by example than to lay out a general theory, so we'll go back to our old acquaintance the Brusselator. Let us first recall the analysis carried out in a previous lecture: In dimensionless form, the Brusselator ODEs are

$$
\begin{aligned}
\dot{x} &= a - bx + x^2 y - x, \\
\dot{y} &= bx - x^2 y.
\end{aligned}
$$

This model has a single equilibrium point at $(x^*, y^*) = (a, b/a)$. This equilibrium point undergoes an Andronov-Hopf bifurcation at $b = b_{\mathrm{AH}} = a^2 + 1$, with oscillations being observed for $b > b_{\mathrm{AH}}$.

If we want to study the reaction-diffusion version of this model, we just add diffusion terms to the two equations. We can immediately obtain a dimensionless form by setting one of the diffusion coefficients to unity:

$$
\begin{aligned}
\frac{\partial x}{\partial t} &= \frac{\partial^2 x}{\partial z^2} + a - bx + x^2 y - x, \\
\frac{\partial y}{\partial t} &= D \frac{\partial^2 y}{\partial z^2} + bx - x^2 y.
\end{aligned}
$$

Here, z is our spatial variable, while D is the relative diffusion coefficient of Y, i.e. the ratio of D_Y to D_X. Note that the steady state of the well-mixed model is also a steady state of the reaction-diffusion model: If $(x, y) = (x^*, y^*)$, then the spatial derivatives of this constant function are zero, as are the reaction terms, so the time derivatives must be zero too. This solution is called the **homogeneous steady state**.[4]

To linearize these equations, let

$$
\begin{aligned}
x &= \delta x + x^* = \delta x + a, \\
\text{and} \qquad y &= \delta y + y^* = \delta y + b/a.
\end{aligned}
$$

Here, δx and δy are displacements from equilibrium, which now depend *both* on time and space. As usual, we will ask what happens when δx and δy are small. Because the above definitions are linear in δx and δy, we have

$$
\frac{\partial x}{\partial t} = \frac{\partial(\delta x)}{\partial t}, \qquad \frac{\partial^2 x}{\partial z^2} = \frac{\partial^2(\delta x)}{\partial z^2}, \qquad \text{etc.}
$$

The linearization of the reaction terms is just $\mathbf{J}^* [\delta x \quad \delta y]^T$, where $\mathbf{J}^*$ is the Jacobian evaluated at the equilibrium point, and the superscripted T represents the transpose operation. Thus we have the linearized equation

$$
\frac{\partial}{\partial t} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = \mathbf{D} \frac{\partial^2}{\partial z^2} \begin{bmatrix} \delta x \\ \delta y \end{bmatrix} + \mathbf{J}^* \begin{bmatrix} \delta x \\ \delta y \end{bmatrix},
$$

---

[4]Analyzing the stability of the homogeneous steady state assumes that the boundary conditions are consistent with this steady state. Otherwise, this is a hollow exercise.

where $\mathbf{D}$ is the matrix which has the diffusion coefficients on its diagonal, and zeros everywhere else:

$$\mathbf{D} = \begin{bmatrix} 1 & 0 \\ 0 & D \end{bmatrix}.$$

Also, the Jacobian for the Brusselator is

$$\mathbf{J}^* = \begin{bmatrix} b-1 & a^2 \\ -b & -a^2 \end{bmatrix}.$$

We again want to determine if the steady state is stable against small perturbations, but this time we want to introduce a spatial aspect. Suppose therefore that our perturbations are inhomogeneous in space. One convenient form is

$$[\delta x \quad \delta y] = [\delta x_0 \quad \delta y_0] e^{\lambda t} e^{ikz}.$$

The $e^{\lambda t}$ part is easily recognizable from our previous work. The term $e^{ikz} = \cos(kz) + i\sin(kz)$ is a convenient way to represent a spatial wave. Our question now will be whether or not conditions can be found under which the steady state is unstable ($\Re(\lambda) > 0$) when a wiggly disturbance is introduced. Since any disturbance over a finite domain can be synthesized by adding up sine waves, this will answer the question of whether the steady state is stable against small, but otherwise arbitrary perturbations.

If we substitute the perturbation into the linearized equation, we get, after cancelling off the common factors of $e^{\lambda t} e^{ikz}$,

$$\lambda \begin{bmatrix} \delta x_0 \\ \delta y_0 \end{bmatrix} = -k^2 \mathbf{D} \begin{bmatrix} \delta x_0 \\ \delta y_0 \end{bmatrix} + \mathbf{J}^* \begin{bmatrix} \delta x_0 \\ \delta y_0 \end{bmatrix},$$

or, after rearranging,

$$\left\{ \lambda \mathbf{I} + k^2 \mathbf{D} - \mathbf{J}^* \right\} \begin{bmatrix} \delta x_0 \\ \delta y_0 \end{bmatrix} = 0,$$

where $\mathbf{I}$ is the identity matrix. This is a homogeneous equation in $[\delta x_0 \quad \delta y_0]$ which only has nontrivial solutions if

$$\left| \lambda \mathbf{I} + k^2 \mathbf{D} - \mathbf{J}^* \right| = 0,$$

where again the bars represent the determinant operation. Note that we haven't done anything so far that is specific to the Brusselator, so that this is a *general characteristic equation for reaction-diffusion equations in one spatial dimension*. Our task will be to determine whether this characteristic equation has solutions for which the real part of λ is positive, and if so, under what conditions. Because of the appearance of the unknown wavenumber k in this equation, this equation has a different solution for every k, so again we are faced with equations where we can't simply find a finite number of roots and stop there.

Let us plug in our $\mathbf{J}^*$ and complete the analysis for the Brusselator:

$$\begin{vmatrix} \lambda + k^2 + 1 - b & -a^2 \\ b & \lambda + Dk^2 + a^2 \end{vmatrix} = 0.$$

$$\therefore \lambda^2 + \lambda \left[k^2(1+D) + 1 - b + a^2 \right] + Dk^4 + k^2 \left[D(1-b) + a^2 \right] + a^2 = 0.$$

10

This quadratic equation is of the form

$$\lambda^2 + q\lambda + p = 0.$$

The solutions are

$$\lambda = \frac{1}{2}\left\{-q \pm \sqrt{q^2 - 4p}\right\}.$$

There will be at least one root with a positive real part provided one of the following two conditions are met:

1. $q < 0$. If this is the case, then $-q$ is positive, and the root of the quadratic equation which uses the plus sign certainly has a positive real part, regardless of whether the discriminant of the equation is positive or negative.

   For our characteristic equation, this condition reduces to

   $$b > 1 + a^2 + k^2(1 + D) > 1 + a^2$$

   since $k^2(1 + D) > 0$. To be clear: If $b > 1 + a^2$, then there exist values of k such that the steady state is unstable. This is just the Andronov-Hopf bifurcation condition, which isn't very interesting since we get this instability without the diffusion terms.

2. $q > 0$ and $p < 0$. If $p < 0$, then the discriminant is both positive and greater in magnitude than $|q|$. Again, this makes the root with the positive sign positive overall.

   For the Brusselator, the condition $q > 0$ is realized when $b < 1 + a^2$. Note that this will take us out of the regime where we expect to see an Andronov-Hopf bifurcation. The condition $p < 0$ is

   $$Dk^4 + k^2[D(1 - b) + a^2] + a^2 < 0.$$

   Note that the left-hand side of this inequality is a concave parabola in the variable $\gamma = k^2$:

   $$f(\gamma) = D\gamma^2 + \gamma[D(1 - b) + a^2] + a^2 < 0.$$

   There will be a range of γ (i.e. k^2) where this is negative provided the minimum of the parabola is itself negative. That's not quite enough though. Since $\gamma = k^2$, only positive values of γ are physically significant. Thus we want to know under what conditions $f(\gamma)$ is negative for some positive values of γ. Note that $f(0) = a^2 > 0$. This means that, if the minimum lies in the region $\gamma < 0$, the entire range containing negative values also lies in this range. We must therefore insist that the minimum lie to the right of the axis. Let us find this minimum:

   $$\frac{df}{d\gamma} = 2D\gamma + D(1 - b) + a^2 = 0.$$
   $$\therefore \gamma = \frac{1}{2}\left(b - 1 - a^2/D\right).$$

11

Since we want γ at the minimum to be positive, we must insist that $b > 1 + a^2/D$.

$$
\begin{aligned}
\therefore f_{\min} &= \frac{D}{4}\left(b - 1 - a^2/D\right)^2 + \frac{1}{2}\left(b - 1 - a^2/D\right)\left[D(1 - b) + a^2\right] + a^2 \\
&= \frac{D}{4}\left(b - 1 - a^2/D\right)^2 - \frac{D}{2}\left(b - 1 - a^2/D\right)^2 + a^2 \\
&= a^2 - \frac{D}{4}\left(b - 1 - a^2/D\right)^2 < 0. \\
\therefore 4a^2/D &< \left(b - 1 - a^2/D\right)^2. \\
\therefore 2a/\sqrt{D} &< b - 1 - a^2/D.
\end{aligned}
$$

Note that this last step is only legitimate because we are restricting ourselves to values of b such that $b > 1 + a^2/D$. Continuing, we have

$$
\begin{aligned}
2a/\sqrt{D} &< b - 1 - a^2/D. \\
\therefore b &> 1 + \frac{2a}{\sqrt{D}} + \frac{a^2}{D}. \\
\therefore b &> \left(1 + \frac{a}{\sqrt{D}}\right)^2.
\end{aligned}
$$

Since we must also have $b < 1 + a^2$, this gives the following bifurcation condition:

$$
\left(1 + \frac{a}{\sqrt{D}}\right)^2 < b < 1 + a^2. \tag{4}
$$

Note that this bifurcation occurs *only* when the steady state would be stable in the absence of diffusion. Thus, this is a purely diffusive instability. Moreover, it occurs for a finite range of wavenumbers k. This instability will therefore form a spatial pattern of some sort since adding up a bunch of sine waves within a finite range of wavelengths should produce a nontrivial wave pattern. This is a **Turing bifurcation**. Technically, a Turing bifurcation is the destabilization of an other stable steady state by diffusive terms, leading to pattern formation.

Suppose that D is small. Then $\left(1 + \frac{a}{\sqrt{D}}\right)^2 = 1 + \frac{2a}{\sqrt{D}} + \frac{a^2}{D} > 1 + a^2$, and it becomes impossible to satisfy both parts of condition 4 at the same time. We conclude that D must be sufficiently large in order to make a Turing bifurcation possible. In other words, Y must diffuse faster than X in the Brusselator model. How much faster? We want

$$
1 + \frac{2a}{\sqrt{D}} + \frac{a^2}{D} < 1 + a^2.
$$

$$
\therefore \quad \sqrt{D} - \frac{1}{\sqrt{D}} > \frac{2}{a}.
$$

In two-variable systems, and in the absence of other complications, Turing bifurcations always require that one of the chemicals diffuse faster than the other. Which chemical has to

diffuse faster depends on the signs of the terms in the Jacobian in a way which we won't go into here.

Condition 4 isn't quite enough to guarantee a Turing bifurcation. Waves with wavenumbers in the appropriate range have to actually fit within the reaction domain. Moreover, the boundary conditions may suppress waves with certain wavenumbers. The methods required to explore these issues aren't too much more complex than we have seen here, but after a while algebra gets a little tedious, and we omit this topic. Suffice it to say that it's easier to observe Turing bifurcations in large systems where many waves can fit.