Computer Practical: Reaction-Diffusion Modelling

Paul Connolly and David Topping, November 2021

1 Overview

In this practical we consider two coupled equations that describe two components that can react together and diffuse / move around in two-dimensions. The model is applicable for a wide variety of situations and very interesting patterns can arise in the solutions. We use the Gray-Scott model (http://www.karlsims.com/rd.html). This example is adopted from Benjamin F. Maier's (https://github.com/benmaier) personal Github repository, supplemented with descriptions taken from Karl Sims (http://www.karlsims.com/rd.html).

2 Model formulation

Figure 1 displays a simple schematic of a two component system with compounds A and B. In this system components A and B start with an initial concentration but as component A is introduced with a constant rate, two B's react with one A to produce another B, whilst B is also removed from our system with a constant rate.

Now let us start to think about this problem from a computational perspective. We model the reaction-diffusion of two chemicals A and B which are distributed across a grid of size N. In other words, we have a 2D matrix with $N \times N$ cells. As they diffuse and react, they will move across the 2D space according to their initial conditions and parameters that dictate these interactions. In the following we present equations that define this model. Do not worry too much about the format of these equations and the operator symbols we use, the point of this exercise is to practice modifying and running an existing model. The Gray-Scott model is described using two partial differential equations:

$$\frac{\partial A}{\partial t} = D_A(\nabla^2 A) - AB^2 + f(1-A) \tag{1}$$

$$\frac{\partial B}{\partial t} = D_B(\nabla^2 B) + AB^2 - (k+f)B \tag{2}$$

The first equation states that that rate of change of concentration of chemical A, at any given point in space, is a function of how quickly it diffuses, represented by a diffusion coefficient D_A, and the operator ∇^2. This operator is called the Laplacian, which is a differential operator that calculates the divergence of the gradient. This provides us with information about concentration gradients, which is needed for diffusion. This makes sense qualitatively since a concentration gradient can result in a mass flux from regions of high-to-low concentrations. Before we move on, the discrete version of these equations are:

$$A_{ij}(t+1) = A_{ij}(t) + \Big[D_A(\nabla^2 A)_{ij} - A_{ij}B_{ij}^2 + f(1-A_{ij})\Big] \times \Delta t \tag{3}$$

$$B_{ij}(t+1) = B_{ij}(t) + \Big[D_B(\nabla^2 B)_{ij} + A_{ij}B_{ij}^2 - (k+f)B_{ij}\Big] \times \Delta t \tag{4}$$

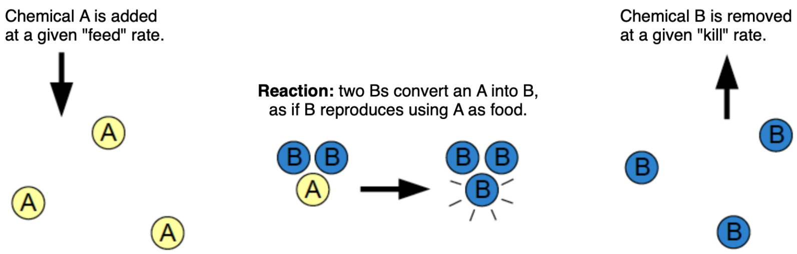
Chemical A is added at a given "feed" rate.
Reaction: two Bs convert an A into B, as if B reproduces using A as food.
Chemical B is removed at a given "kill" rate.
A
A
A
B
B
A
B
B
B
B
B
B

Figure 1: Basic schematic of how two compounds A and B interact in our hypothetical system

The symbol A_{ij} represents the concentration of chemical A at grid coordinates (i, j) (similar for B). We also have parameters f and k which represent a "feed" rate f and "death" rate k.

First, we need to take care of the discretised Laplacian terms $\nabla^2 A$ and $\nabla^2 B$. The discretised Laplacian of a grid cell (i, j) can be computed by summing over neighboring cells and subtract the value of the original cell with the total weight of neighbouring cells. One possible implementation is to only recognize direct neighbors of grid difference $\Delta = 1$, i.e. at $(i, j-1)$, $(i, j+1)$, $(i-1, j)$, and $(i+1, j)$. The formula, which we can implement as code, is

$$(\nabla^2 A)_{ij} = A_{i,j-1} + A_{i,j+1} + A_{i-1,j} + A_{i+1,j} - 4A_{ij} \tag{5}$$

which, along with Equations 3-4, enables us to calculate future values of A and B.

3 Downloading and compiling the code

Log into the server computer with an SSH window. Also, log into the server computer with an SFTP window. Space the windows out on your screen so that you can easily move between them.
If you have not done this already, download the code you will be using today from GitHub by typing the following:

```
git clone https://github.com/EnvModelling/Env_modelling
```

this should download the code to your working directory.
In order to access the files you will need to change the working directory to where the files are. Type the following:
`cd Env_modelling/Spatio-temporal-modelling`
followed by the `tab` key and the command should auto-complete. Then press the `enter` key.

Type `ls` followed by the `enter` key. The screen should list the files in this directory.

4 Viewing and Editing the code

We will use the `nano` text editor to view some files. In this practical there is one file of interest called `Reaction_diffusion_2D.py`. To look at the file type:

```
nano -l Reaction_diffusion_2D.py
```

the `-l` means to show line numbers. The code can be edited in the text editor and saved by typing `Ctrl-X` at the same time and pressing `Y` to save the file. The important lines are 117-135 and 295-299 of this file.

5 Running the model

The procedure for running the model is to:

1. Edit the file `Reaction_diffusion_2D.py` to configure the model.
2. Type `python3 Reaction_diffusion_2D.py` at the command line, followed by `enter` to run the model.

After running the model a plot of the model simulation will be shown in `/tmp/<username>/*.png`, which you can download using SFTP by typing:

```
get /tmp/<username>/<filename>.png
```

6 Experiments

FOR ALL EXPERIMENTS: I STRONGLY SUGGEST YOU OPEN UP A WORD OR POWERPOINT DOCUMENT AND INSERT THE FIGURES AND MAKE SOME NOTES AS YOU GO. YOU COULD BE ASKED QUESTIONS ABOUT THEM IN THE ASSESSMENT.

6.1 Standard *Branching* instability simulation

Run the standard model by typing:

```
python3 Reaction_diffusion_2D.py
```

Download the resulting output image files from the SFTP window

```
get /tmp/<username>/exp0.png
get /tmp/<username>/exp1.png
```

These two files show the initial values of A and B and the final values respectively. Notice how quite generic initial conditions (a square in the middle) gives rise to more natural conditions where a network of branches develop. This shows that complex structure can form from arbitrary initial conditions.

This standard model has the feed rate, $f = 0.060$, and the kill rate, $k = 0.062$.

6.2 Sensitivity to feed rate across the domain

Set `EXPERIMENT=2`.

This changes the feed-rate, f, to vary so that it is small ($f = 0.30$) in the middle of the domain and large on the outside of the domain ($f = 0.80$). The formula for this variation is $f = \exp\left(\frac{|x||y|}{2500}\right)$. Following this run the model by typing:

```
python3 Reaction_diffusion_2D.py
```

Download the resulting output image files from the SFTP window

```
get /tmp/<username>/exp0.png
get /tmp/<username>/exp2.png
```

These two files show the initial values of A and B and the final values respectively.

Question: How does the pattern change with distance from the centre? Write a sentence as the feed rate is increased the pattern changes from being ______ to ______)

6.3 Bubbles / skin

Set `EXPERIMENT=3`.

This increases the feed-rate, $f = 0.11$, and reduces the kill rate slightly. The result is that we form bubbles or structures that look like skin cells. Following this run the model by typing:

```
python3 Reaction_diffusion_2D.py
```

Download the resulting output image files from the SFTP window

```
get /tmp/<username>/exp0.png
get /tmp/<username>/exp3.png
```

These two files show the initial values of A and B and the final values respectively.

Description: You should notice the bubble structures that look like skin cells, or Giraffe skin. This demonstrates that seemingly complex structures in nature can arise from some very simple laws.

6.4 Mitosis / cell division

The last two simulations are animations of what happens when the initial condition is a circular cell at the centre.

Set `EXPERIMENT=4`.

This sets the feed rate, $f = 0.0367$, and the kill rate, $k = 0.0649$.

```
python3 Reaction_diffusion_2D.py
```

Download the resulting output image files from the SFTP window

```
get /tmp/<username>/exp4.png
get /tmp/<username>/animation.gif
```

These two files show the initial values of A and B and animation of how A varies with time respectively.

Question: We have a small feed rate here (c.f. Section 6.2). What happens over time to the single circle at the centre?

6.5 Coral Growth / Branching instability

With `EXPERIMENT=4`, scroll down to lines 298 and 299 to increase the feed rate to $f = 0.0545$ and $k = 0.062$.

Run the model and download the files.

Description: You should notice that branching takes place when the corners start to develop. This is a process known as rare-faction, which is when the chemical creating a new phase is directed to certain parts of the structure allow it to grow faster. The same process occurs during ice crystal growth.