***Part 1: Spatial Modelling***

***Task 1a.***

Open the file diffusion template.py

This contains code that simulates a linear (one-dimensional) grid system as discussed in Monday’s session. To begin with we will not model any reactions but just see how diffusion processes affect the system.

The grid is set up to contain a system of 20 cells. We are going to look how the distribution of a chemical species X evolves over time, under diffusion.

We will assume that all cells have a concentration of [X] = 1.0, except for cell 5 which has [X] = 3.0

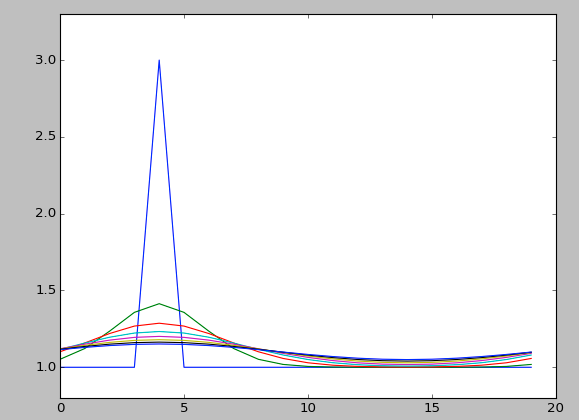
The code is set up to run the odeint solver to simulate the system over time, but at present the code is imcomplete, with the function to model diffusion missing. Follow the steps below to complete the code .

*(line 77)* Set up the initial conditions as detailed above.

*(line 37)* Insert code into the d\_diffusion function to calculate the diffusion rates by finding the net flow of X into and out of each cell.

Assume that the end cells are joined, so the system forms a loop with periodic boundary conditions. Refer to the session slides if you need guidance.

At the end of the file there is code that loops to plot the concentration profile of the cells at a series of time observations. If you add code to model the diffusion process correctly, this should produce the following figure:



**Task 1b.**

Open the file animation\_helper.py and copy the two functions into your code including the top import line.

These can be used to animate the stored data timeseries in s\_obs. Add the following code at the end of your file to create the animation.

1. fig2 = plt.figure()
2. ax2 = fig2.add\_subplot(1,1,1)
4. line,=ax2.plot(x\_coords, s\_obs[0,:])
5. ax2.set\_ylim([0.8,3.2])
7. anim=plot\_time\_series(fig2, line, s\_obs, frame\_interval=5)
8. fig2.show()

You can play with the animation by adjusting the number of observations you make with the t\_obs array, or changing the frame\_interval that controls the time between showing each animation frame.

**Task 1c.**

Create two additional functions: d\_diffusion\_closed and d\_diffusion\_open. These should allow you to simulate diffusion with closed or open boundaries at the ends of the grid.

*Note - change the code that sets the y-axis limits so they start from 0 rather than 0.8 and extend the time of the simulation until the system reaches a steady state.*

Examine the behaviour of the system under the three conditions:

* periodic boundaries
* open boundaries
* closed boundaries

Does the system behaved as you expected?

**Task 1d.**

Each value in the s array represents the amount of X in each cell. Suppose in addition to the diffusion process we wish to model a reaction that occurs in each grid cell.

Suppose that X is actually a bacteria that reproduces, such that in an isolated cell (referenced by s[i] in our code) the population [X]i increases in a process with a rate that can be modelled in accordance with logistic growth:

where we have additional parameters r (maximum rate of production), and [X]max the maximum steady state concentration.

First adjust the initial conditions so that all cells have zero concentration except cell 10 that has initial concentration 0.05.

Then add code to insert this reaction into our code. To do this you will need to delete the pass command and calculate ds[i] in accordance with the above expression.

Use values r = 0.1 and [X]max = 2.5 and set the diffusion rate kD = 0.01, using periodic boundary conditions.

*Remember to adjust the y-axis limits so you can see the full behaviour.*

Describe the behaviour you see.

How could this relate to the growth of a bacterial colony?

**Task 1e.**

Adjust the diffusion rate kD to 0.5.

What do you find?

How could we simplify our model of this system in this case?

***Part 2: Goodwin Oscillators***

In this exercise we will examine the behaviour of the following system.

In this system X, Y and Z are proteins that interact and regulate each others production rates.

This system is an example of a *Goodwin Oscillator* which can exhibit limit cycle oscillations as a result of non-linearity, negative feedback, and delay. This type of model is based on the work of the mathematical biologist Brian Goodwin who first discussed them in a 1965 paper *Oscillatory behavior in enzymatic control processes* (Adv Enzyme Regul 3: 425–438).

The aim of the model was an attempt to propose a potential mechanism by which the observation of periodic/rhythmic behaviour in cellular processes could be explained.

**Task 2a.**

Examine the model equations above.

1. Which terms are associated with production?
2. Which terms are associated with loss?
3. The proteins in the system have interactions which act either enzymatically or as transcription factors to activate production of the other proteins. The general model for such processes where protein S activates production of protein P is a production rate term like:

Under what conditions can this model be simplified and reduced to the model of activation used in the model equations above, i.e.

1. Describe the relationships between
2. X and Y
3. Y and Z
4. X and Z
5. Draw a diagram of this system

use an arrow like this ----| to identify a inhibiting relationship

use an arrow like this ----> to identify an activating relationship

vi) Explain which features of the model provide the necessary conditions for limit cycle oscillations (non-linearity and delay).

1. Suppose the system is initially in a state where the no proteins are present (i.e. the concentrations of X,Y and Z are all zero).

Try to use the model to predict how the system will behave from this point, and describe in words what you might expect to find.

**Task 2b.**

i) Code the system using the template code file oscillator\_template.py to get you started.

Run the system with parameters:

k1=1.0=k2=k4=k6=KI=1.0

k3=k5=10.0

n=1.0

and initial conditions:

X=Y=Z=0.0

ii) Describe the behaviour observed in the timeseries.

iii) Add code to draw the phase plot showing X against Z.

**Task 2c.**

If you increase **n** while holding the other parameters constant you can make the system produce limit cycle behavior. Find the minimum integer value of **n** required for the system to limit cycle behaviour and analyse the oscillations you observe.

i) How can you demonstrate that a limit cycle has been reached?

ii) Use the plots generated to determine the amplitude and period of the cycles.

*Think about how could you make more accurate measurements of these parameters using the values in the t\_obs and s\_obs arrays?*

iii) How does the limit cycle change as n is increased, e.g. to 20?

**Task 2d.**

The model we use assumes a chain in which:

X activates Y

Y activates Z

Z inhibits X.

Adjust your model to add an additional species into the chain such that:

X activates Y1

Y1 activates Y2

Y2 activates Z

Z inhibits X.

i) Investigate how this changes the minimum value of n required for oscillations.

ii) Explain this change in reference for the requirements for a Goodwin oscillator.