Practical 5: Hand in your solution to MyUni before N/A at 1pm.

You should read the practical, and prepare before the actual session.

give them code get them to use different input functions compare the outputs

comparing models

functions as input arguments to functions

errors abs distributional chi-squared test

The goals of this practical are

1. To learn about how you might compare two models
2. To use a function as an input argument to another function.

Notes: in many programming problems we want to write code to be as general and reusable as possible. That might mean it should work for many cases, each of which has a different form, *e.g.,* it should work for a sine function, and for a quadratic, and for a log function. Thus, we need to write our function in such a way that this choice can be input.

In MATLAB we do this by passing a function *handle*. That is, the function that describes the curve of interest (be it sine, quadratic or log) can be passed into another function using it's handle. In MATLAB the `@` symbol is used to create a handle.

In this practical, we will experiment with using a function as an input argument in reaction-diffusion equations:

$$\begin{aligned}\frac{\partial u_1}{\partial t} &= r_1(u_1,u_2) + \alpha_1 \nabla^2 u_1 \\ \frac{\partial u_2}{\partial t} &= r_2(u_1,u_2) + \alpha_2 \nabla^2 u_2\end{aligned}$$

which can be written in vector form as

$$\frac{\partial \mathbf{u}}{\partial t} = R(\mathbf{u}) + D\nabla^2 \mathbf{u},$$

where $R(\cdot)$ is a vector-valued function governing the reactions, and D is a diagonal matrix with the α_i along its diagonals. We will put in different forms of the function $R(\cdot)$.

Tasks:

1. (a) Read
   https://au.mathworks.com/help/matlab/matlab_prog/pass-a-function-to-another-function.html
   https://au.mathworks.com/help/matlab/matlab_prog/creating-a-function-handle.html
   https://au.mathworks.com/help/matlab/matlab_prog/anonymous-functions.html
   https://stackoverflow.com/questions/8615206/passing-a-function-to-a-function-in-matlab
   You might not understand all of this in the first pass – the exercise will hopefully help.

   (b) Download the MATLAB code `reaction_diffusion.m` from the web page, and read the code. It will simulate a reaction-diffusion equation. You can input the type of reaction by specifying the input function. For example the code `reaction_diffusion_4.m` will simulate the Schnakenberg reaction-diffusion system. The critical part of the code is

```
D = diag([1, 4.8]);
gamma = 1000; % or 100, 1600, 6400
a = -0.55;
b  = 1.9;
R = @(u) gamma*[ a - u(1) + u(1)^2*u(2);
                 b        - u(1)^2*u(2)];
u0(1,:) = (a+b) +     0.25*rand(1,num_x);
u0(2,:) = b/(a+b)^2 + 0.25*rand(1,num_x);
u = reaction_diffusion( R, D, u0, num_t, dt);
```

   Note here that the function `R` takes as input a vector `u` with two elements $(u(1), u(2))$ and outputs a similar vector with two elements giving the two reaction rates (compare it to your lecture notes).

   (c) Experiment with different reaction terms. Several examples to test are

      i. The *identity* reaction, or a reaction where the chemicals don't actually interact, and hence their concentrations are subject only to diffusion.

      ii. The Schnakenberg case, given in lectures and in the example above.

      iii. The linear case, *i.e.,* the case where the reaction rates are linear in the concentrations. We will use here a linearised version of the Schnakenberg case, rewritten in variables $x_i = u_i - u_i^*$ where u_i^* is the equilibrium solution (see lectures). The result is the equations

$$
\begin{aligned}
\frac{\partial x_1}{\partial t} &= \left(\frac{b-a}{a+b}\right) x_1 + (a+b)^2 x_2 + \alpha_1 \nabla^2 x_1 \\
\frac{\partial x_2}{\partial t} &= -\left(\frac{2b}{a+b}\right) x_1 - (a+b)^2 x_2 + \alpha_2 \nabla^2 x_2
\end{aligned}
$$

      and now the initial values of x_i should be near to zero (indicating a small perturbation from the equilibrium).

   Note that if you have M reagents, your function should take a vector of M concentrations as input, and output a vector of M reaction rates.

2. Analyse the signal produced from the Schnakenberg system.
   (a) Use a FTT to see if $u_1(x)$ it is a simple sine curve, or if it has more components.
   (b) Examine the *marginal* distribution of the values of u_1 by plotting a histogram of $u_1(x)$. Here the marginal means the distribution of values ignoring the x axis. You can do this in MATLAB using the command

   ```
   histogram( u(1,:), 0:0.25:3 )
   ```

   Have a look at the help documents for `histogram` to understand this command and play with it.

   You might need to increase the number of x points to 100 or more to see the marginal distribution more clearly.
   (c) Imagine that the signal/pattern we are trying to create was the function was a simple sine wave with frequency 4 in the units given, amplitude 1, and mean given by the equilibrium value of u_1.
   - Calculate the mean and variance of the sine function and our model.
   - Plot the two along side each other.
   - Plot their marginal distributions.

   Now compare the two.