Instructions for running the 2D Turing pattern code APP (MATLAB)

Files can be accessed on GitHub at:

https://github.com/frm3-st-andrews/EASTBIO_2022.git

The simulation app is used to solve the (Turing) system of PDEs:

$$\frac{\partial u}{\partial t} = D_u\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\alpha_u - u + u^2 v \quad (1)$$

$$\frac{\partial v}{\partial t} = D_v\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+\alpha_v - u^2 v \quad (2)$$

where

- Production rate of u (α_u) - the source term for u per time-step
- Production rate of v (α_v)- the source term for v per time-step
- Diffusion rate of u (D_u)- how fast u can diffuse
- Diffusion rate of v (D_v)- how fast v can diffuse

We use reflective (zero-flux) boundary conditions - that is the concentrations u and v are conserved within the domain.

We set the initial conditions to be a random small perturbation (0.001) of the steady state (u,v)=(1,0.9):

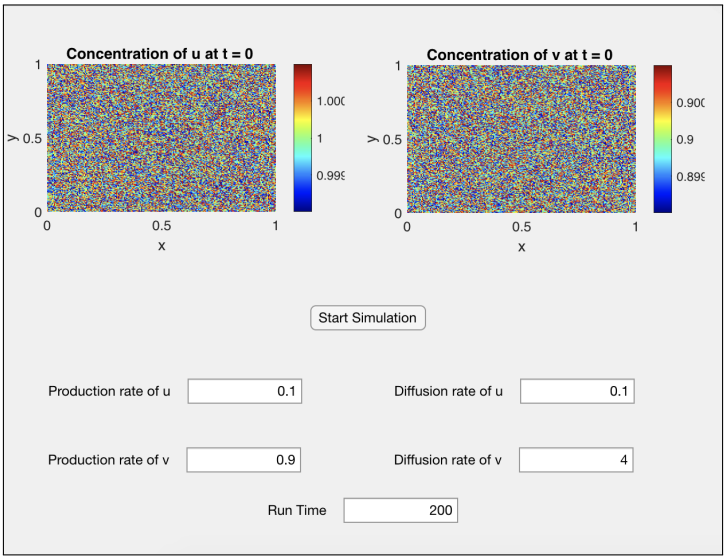
$$u(t=0,x,y) = 1 \pm 0.001\ rand(Nx,Ny) \quad (3)$$

$$v(t=0,x,y) = 0.9 \pm 0.001\ rand(Nx,Ny) \quad (4)$$

where $rand(Nx,Ny)$ is a matrix of random numbers between 0 and 1, the size of our domain ($Nx \times Ny$).

Step 1:

Download the app and install into MATLAB. Click on the app to open. You should see the following:



Note, this window is scalable, so you can stretch it to make images larger.

Step 2:
Pick your parameter values and 'Run Time' (how long you wish to run the simulation).

Step 2:
Run the simulation until finished by clicking the 'Start Simulation' button. The app will start running displaying the update concentrations of u and v until the final time-step is reached.

Note, if you click start simulation again before the simulation is finished the app resets the simulation.

Once finished the app will save two figures (one for u and one for v at the final time-step). The app also saves all data from the run into an '.m' file. Note, if you are re-running with different values then manually change the names of these files as they will be overwritten. The data stored in the '.m' file can be accessed through MATLAB. This data includes stored concentrations of u and v over time as well as data about the parameters, domain and time-steps used.

Things to investigate:

- The original ratio D_u and D_v is 1:40. What happens if you reduce this ratio? (e.g. reduce D_v or increase D_u.)
- What happens if you switch off the production rate of u or v?

- What happens if you increase/decrease the production rate of u or v? (choose values between 0 and 1)

Alternative: Running the full code

Download the **Turing_full_code.m** file. This code has the full code used to produce the app above. You can see the details of the code and modify more parameters/settings. Note, the code saves the data (lines 170/171) so if re-running then choose new filenames to avoid overwritten files.

Note - you can edit all lines of code, but some are chosen to ensure the numerics run (smoothly) (e.g a low time-step) so some changes may result in error.

Further things to investigate:

- What happens if you change to a rectangular domain (increase 'xmax/ymax' in Section 1 of code?
- What happens if we decrease the effects of our interaction terms (decrease 'K' in Section 4.1 of code, e.g K=0.5, 0.1)?
- What happens if we increase the effects of our interaction terms (increase 'K' in Section 4.1 of code, e.g K=5, 20)?

Note, we could change the form of f and g (Section 4.1 of code). The interaction terms we use are 'Schnakenberg' dynamics where:

$$\begin{aligned} f(u,v) &= \alpha_u - u + u^2 v \\ g(u,v) &= \alpha_v - u^2 v \end{aligned}$$

However other interactions between u and v could be considered. However, the steady state of the system is where

$$f(u,v) = 0, \quad g(u,v) = 0$$

so if we change f and g the steady state will change. So we may not get patterns unless we edit the initial conditions/parameter values. Feel free to explore this further.