Delft University of Technology

Numerical Analysis For PDE's
WI4014TU

Assignment:6

Ilambharathi Govindasamy

Numerical solution of a non-linear coupled reaction-diffusion problem

Schnakenberg Model

The Schnakenberg model suggests a physical mechanisms behind the emergence of Turing patters in nature. The model shows that a chemical reaction between two substances, the 'slow' activator u and the 'fast' inhibitor v , leads to the emergence of regular patters from noise. Such reactions happen, for instance, in animal skins and lead to the characteristic appearance of cheetah's and zebra's. The process is described by the following system of non-linear coupled reaction-diffusion PDE's

$$\frac{\partial u}{\partial t} = D_u \Delta u + k(a - u + u^2 v) \tag{1}$$

$$\frac{\partial v}{\partial t} = D_v \Delta v + k(b - u^2 v), \qquad (x, y) \in \Omega, \qquad t \in (0, T] \tag{2}$$

Boundary condition:

$$-D_u \nabla u \cdot \mathbf{n} = 0, -D_n \nabla n \cdot \mathbf{n} = 0, \qquad (x, y) \in \partial\Omega \tag{3}$$

Initial condition:

$$u(x, y, 0) = u_0(x, y), \qquad v(x, y, 0) = v_0(x, y) \qquad (x, y) \in \Omega \tag{4}$$

The rates of diffusion are determined by the corresponding diffusivity constants D_u=0.05, D_v=1.0, the reaction constants are k=5, a=0.1305, b=0.7695

$$u_0(x, y) = a + b + r(x, y)v_0(x, y) = \frac{b}{(a + b)^2} \tag{5}$$

r(x,y)) is a small nonuniform perturbation in the concentration of the activator.The computational domain Ω= (0,4) $\times$ (0,4). The pattern should be almost completely formed at T=20

Theory

1. (a) For a doubly-uniform grid on a rectangular domain and lexicographic ordering of the unknowns, derive the symmetric matrix A of the negative 2D FD Laplacian with zero Neumann Boundary. The negative second derivative operator is given by

$$-u''_{i,j} = \frac{-u_{i+1,j} + 2u_{i,j} - u_{i-1,j}}{h_x^2} + \frac{-u_{i,j-1} + 2u_{i,j} - u_{i,j+1}}{h_y^2}$$

which for $h_x = h_y$ is reduced to,

$$-u''_{i,j} = \frac{4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j-1} - u_{i,j+1}}{h_x^2}$$

Now homogeneous Neumann boundary is applied at the boundary, based on forward difference method which provides an approximation of first order. The right and top boundary nodes are included as unknowns in the system matrix, whereas at i+1 or j+1 for the Neumann boundary,

$$\frac{u_{n+1} - u_n}{\Delta x} = 0$$

,

$$u_{n+1} = u_n$$

Applying this for $N_x = N_y = 4$ (Intervals), we get the following

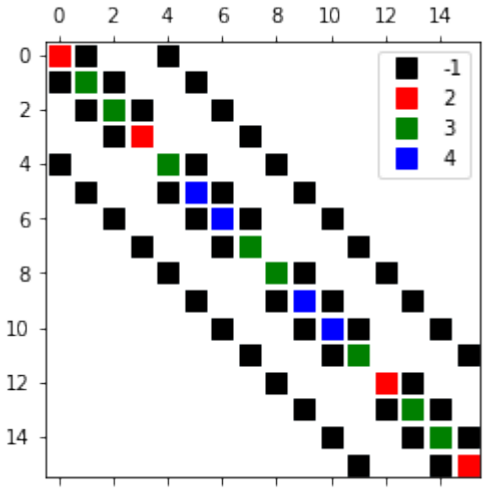

Fig 4: System matrix

$$\text{A}=1/h^2 \begin{vmatrix} 2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 3 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 2 \end{vmatrix}$$

(b) A can also be created by ,

$$A = I_y \bigotimes A_{xx} + A_{yy} \bigotimes I_x \tag{6}$$

$$A_{xx} = D_x^T D_x \quad ; \quad A_{yy} = D_y^T D_y \tag{7}$$

To find D_x and D_y, A_{xx} and A_{yy} are to be found. They are derived from the system matrix by recognising diagonal pattern,

$$A_{xx}{=}A_{yy}{=}1/h \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

$$D_x{=}D_y{=}1/h \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) The explicit function of vector functions are,

$$u' = -D_u \mathbf{A}u + \mathbf{k(a\text{-}u} + u^2\mathbf{v)}$$
$$v' = -D_v \mathbf{A}v + \mathbf{k(b\text{-}}u^2\mathbf{v)}$$

Let w = $\begin{bmatrix} u \\ v \end{bmatrix}$, R(w) = $\begin{bmatrix} a - u + u^2 v \\ b - u^2 v \end{bmatrix}$, DA= $\begin{bmatrix} D_u & 0 \\ 0 & D_v \end{bmatrix} \bigotimes A$

Therefore, single system equation can be written as,

$$\mathbf{w'} = \mathbf{\text{-}DAw} + \mathbf{kR(w)}$$

Numerical time integration

(a) **Forward Euler method for coupled ODE:**

$$w^{k+1} = w^k + dt(-DAw^k + kR(w^k)) \tag{8}$$

Expressing this with u and v,

$$u^{k+1} = u^k + dt(-D_u A u^k + K(a - u^k + u_k^2 v^k)) \tag{9}$$

$$v^{k+1} = v^k + dt(-D_v A v^k + K(b - u_k^2 v^k)) \tag{10}$$

(b) For a linearized equation, $\frac{\partial u}{\partial t} = \Delta u + ku$, the Forward Euler stability in case of zero Dirichlet conditions is found to be

$$0 < dt < \frac{2}{k + \lambda} \tag{11}$$

$$0 < \frac{T}{N_t} < \frac{2}{k + \lambda} \tag{12}$$

$$0 < \frac{(k + \lambda)T}{2} < N_t \tag{13}$$

$$N_t > \frac{(k + \lambda)T}{2} > 0 \tag{14}$$

where λ is the eigen value which should also follow the condition of $\lambda > 0$. This is derived based on method of lines.

(c) **Backward Euler iteration formula:**

$$w^{k+1} = w^k + dt[-DAw^{k+1} + kR(w^{k+1})] \tag{15}$$

Expressing this with u and v,

$$u^{k+1} = u^k + dt(-D_u A u^{k+1} + K(a - u^{k+1} + u_{k+1}^2 v^{k+1})) \tag{16}$$

$$v^{k+1} = v^k + dt(-D_v A v^{k+1} + K(b - u_{k+1}^2 v^{k+1})) \tag{17}$$

Newton-Raphson algorithm

Backward Euler iteration formula:

$$w^{k+1} = w^k + dt[-DAw^{k+1} + kR(w^{k+1})] \tag{18}$$

The residual that will be minimized by Newton-Raphson method is

$$\left\| (w^k + dt[-DAw^{k+1} + kR(w^{k+1})] - w_i^{k+1} \right\| \tag{19}$$

The inner point equation for Newton-Raphson algorithm is written as follows,

$$w_{i+1}^{k+1} = w_i^{k+1} + [I - h_t J(w_i^{k+1})]^{-1}[w^k + dt(-DAw^{k+1} + kR(w^{k+1})) - w_i^{k+1}] \tag{20}$$

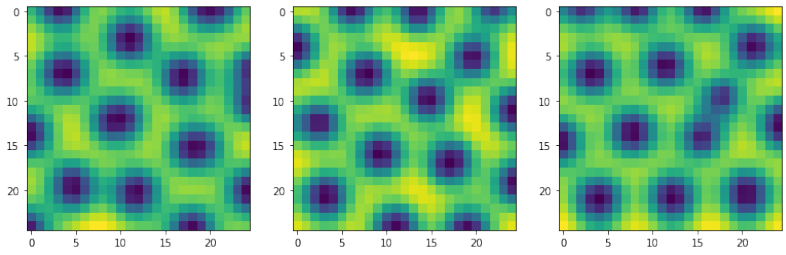

Figure 1: $\mathrm{N}_t = 30k, 50k, 70k(Test, t = T)$

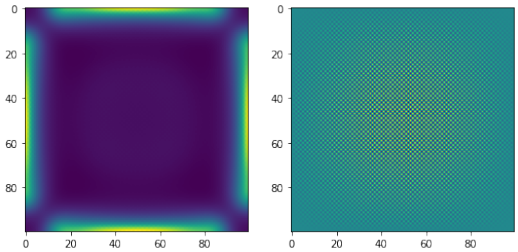

Figure 2: Activator(left) and Inhibitor(Right) at t=T

Jacobian matrix

$$J(u,v) = \begin{bmatrix} \frac{\partial f_u}{\partial u} & \frac{\partial f_u}{\partial v} \\ \frac{\partial f_v}{\partial u} & \frac{\partial f_v}{\partial v} \end{bmatrix} \tag{21}$$

The Jacobian matrix is constructed as follows,

$$J = \begin{bmatrix} -D_u A - I + 2diag(u_i)^{k+1} v^{k+1} & Kdiag(u_i^2)^{k+1} \\ -2Kdiag(u_i)^{k+1} v^{k+1} & -D_v A - I + 2diag(u_i)^{k+1} v^{k+1} \end{bmatrix} \tag{22}$$

Implementation

4.Forward Euler

(a) The empirical N_t is chosen based on time-lapse images for inhibitor v with $N_x = N_y = 25$, furthermore varying the values of K. It is found that the system does not get solved till N_t=5000, but after that the stability increases with increase in number of N_t to a point where it is less significantly visible. The results are clearer, and consistent from the N_t of 50000, therefore also considering the computational costs involved, N_t is chosen as low as possible , i.e.: 50000. The sample iterations are shown in Figure.1

(b) Elapsed time during the whole program in seconds: 3814.685 at $N_x = N_y = 100$ and N_t of 50000

(c) The code has been developed and therefore the activator, inhibitor results are plotted as reported with Figure.2 representing at time t=T, while Figure.3 represents t=0
The Backward Euler Newton Rapshon code is given as a comment in the attached code,, as it has bugs to be fixed

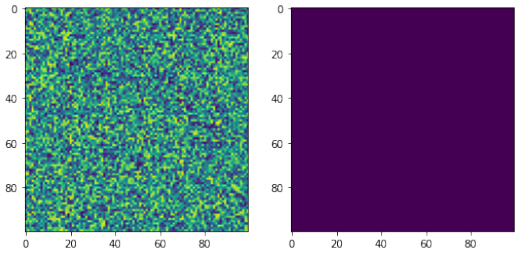

Figure 3: Activator(left) and Inhibitor(Right) at t=0