Numerical Analysis [WI4014TU]
Assignment 5

February 13, 2025

Chirag Bansal
Student number: 6137776
MS Aerospace Engineering

---

1 Schnakenberg Model

The Schnakenberg model suggests a physical mechanisms behind the emergence of Turing patters in nature. The model demonstrates how a chemical reaction between two substances, the 'slow' activator u and the 'fast' inhibitor v, leads to the emergence of regular patterns from noise. It is given by a system of non-linear coupled reaction-diffusion PDE's:

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

$$\frac{\partial v}{\partial t} = D_v \Delta v + \kappa \left(b - u^2 v \right) \tag{2}$$

$$(x, y) \in \Omega, \quad t \in (0, T]$$

The initial and boundary conditions are given as:

$$u(x, y, 0) = a + b + r(x, y), \quad v(x, y, 0) = \frac{b}{(a+b)^2}, \quad (x, y) \in \Omega \tag{3}$$

$$-D_u \nabla u \cdot \mathbf{n} = 0, \quad -D_v \nabla v \cdot \mathbf{n} = 0, \quad , \quad (x, y) \in \partial\Omega \tag{4}$$

where, $D_u = 0.05$ and $D_v = 1.0$ are diffusivity constants, $\kappa = 5$, $a = 0.1305$, $b = 0.7695$, and $r(x, y)$ is a small non-uniform perturbation in the concentration of the activator. The computational domain $\Omega = (0, 4) \times (0, 4)$ and max computation time $T = 20$.

2 Theory

2.1 Finite-Difference Discretisation

2.1.1 FD negative-Laplacian matrix with Neumann BC's

1. Let the vectors $\mathbf{u}$ and $\mathbf{v}$ contain the grid values $u(x_i, y_j, t)$ and $v(x_i, y_j, t)$, respectively. Write down (in matrix-vector notation) the two coupled systems of ODE's that will be obtained after discretizing the PDE's (1) and (2) in space with the FD Method.

Solution: The discretised PDE's in space using finite-difference method will take the following form:

$$\frac{d\mathbf{u}}{dt} = -D_u A\mathbf{u} + \kappa(aI - \mathbf{u} + \mathbf{u}^2\mathbf{v}) \tag{5}$$

$$\frac{d\mathbf{v}}{dt} = -D_v A\mathbf{v} + \kappa(bI - \mathbf{u}^2\mathbf{v}) \tag{6}$$

2. 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 BC's. Start by writing down the discrete FD equations for the problem (1)–(4). Symmetrization can be obtained by a suitable multiplier in the boundary FD equations. Derive the matrix A for the grid with $N_x = N_y = 2$.

Solution: The FD discretised equation for $\mathbf{u}$ (Equation 5) and $\mathbf{v}$ (Equation 6):

$$\frac{1}{h^2}(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) + \kappa(a - u_{i,j} + u_{i,j}^2 v_{i,j}) = 0 \tag{7}$$

$$\frac{1}{h^2}(v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j}) + \kappa(b - u_{i,j}^2 v_{i,j}) = 0 \tag{8}$$

The negative laplacian matrix A is going to be the same for both the discretised PDE's for $\mathbf{u}$ and $\mathbf{v}$. Therefore, solving either at all grid points will yield the required matrix.
For $N_x = N_y = 2$, $h_x = h_y = h = 2$.
Using backward-differencing for the boundary conditions gives:

$$\textbf{(4)} \implies \frac{u_{i,j} - u_{i-1,j}}{h} = 0, \quad \frac{u_{i,j} - u_{i,j-1}}{h} = 0 \tag{9}$$

Solving for $\mathbf{u}$, the FD equations at all the grid nodes after including the Neumann boundary conditions, therefore, are:

$$
\begin{aligned}
-0.25u_{1,0} - 0.25u_{0,1} + 0.5u_{0,0} &= f_{0,0}\\
-0.25u_{2,0} - 0.25u_{0,0} - 0.25u_{1,1} + 0.75u_{1,0} &= f_{1,0}\\
-0.25u_{1,0} - 0.25u_{2,1} + 0.5u_{2,0} &= f_{2,0}\\
-0.25u_{1,1} - 0.25u_{0,2} - 0.25u_{0,0} + 0.75u_{0,1} &= f_{0,1}\\
-0.25u_{2,1} - 0.25u_{0,1} - 0.25u_{1,2} - 0.25u_{1,0} + 1u_{1,1} &= f_{1,1}\\
-0.25u_{1,1} - 0.25u_{2,2} - 0.25u_{2,0} + 0.75u_{2,1} &= f_{2,1}\\
-0.25u_{1,2} - 0.25u_{0,1} + 0.5u_{0,2} &= f_{0,2}\\
-0.25u_{2,2} - 0.25u_{0,2} - 0.25u_{1,1} + 0.75u_{1,2} &= f_{1,2}\\
-0.25u_{1,2} - 0.25u_{2,1} + 0.5u_{2,2} &= f_{2,2}
\end{aligned}
$$

The negative laplacian matrix A is:

$$
A = \begin{bmatrix}
0.5 & -0.25 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0\\
-0.25 & 0.75 & -0.25 & 0 & -0.25 & 0 & 0 & 0 & 0\\
0 & -0.25 & 0.5 & 0 & 0 & -0.25 & 0 & 0 & 0\\
-0.25 & 0 & 0 & 0.75 & -0.25 & 0 & -0.25 & 0 & 0\\
0 & -0.25 & 0 & -0.25 & 1.0 & -0.25 & 0 & -0.25 & 0\\
0 & 0 & -0.25 & 0 & -0.25 & 0.75 & 0 & 0 & -0.25\\
0 & 0 & 0 & -0.25 & 0 & 0 & 0.5 & -0.25 & 0\\
0 & 0 & 0 & 0 & -0.25 & 0 & -0.25 & 0.75 & -0.25\\
0 & 0 & 0 & 0 & 0 & -0.25 & 0 & -0.25 & 0.5
\end{bmatrix} \tag{10}
$$

2.1.2 Constructing FD matrix via Kronecker products

1. Construct the sparse FD matrix of the negative 2D Laplacian (with Neumann's BC's) using the Kronecker products of sparse matrices.

Solution: Given $N_x = N_y = 2$. D_x and D_y are the backward difference matrix in the x- and y- direction. Keeping the Neumann boundary conditions under consideration, the required matrices are given as follows:

$$I_x = I_{N_x+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$I_y = I_{N_y+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since the number of sub-grids is same, matrices D_x and D_y will also be the same. However, due to the boundary conditions, backward-differencing scheme will result in the first and last row of these matrices to be 0.

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

Now, $L_{yy} = L_{xx} = D_x^T D_x$

$$\implies L_{yy} = L_{xx} = \frac{1}{h^2}\begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

Now, $A = I_y \bigotimes L_{xx} + L_{yy} \bigotimes I_x$

$$I_y \bigotimes L_{xx} = \begin{bmatrix} 0.25 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.25 & 0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & -0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.25 & 0.5 & -0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.25 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.25 & -0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0.5 & -0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0.25 \end{bmatrix}$$

$$L_{yy} \bigotimes I_x = \begin{bmatrix} 0.25 & 0 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & -0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & -0.25 & 0 & 0 & 0 \\ -0.25 & 0 & 0 & 0.5 & 0 & 0 & -0.25 & 0 & 0 \\ 0 & -0.25 & 0 & 0 & 0.5 & 0 & 0 & -0.25 & 0 \\ 0 & 0 & -0.25 & 0 & 0 & 0.5 & 0 & 0 & -0.25 \\ 0 & 0 & 0 & -0.25 & 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 0 & 0.25 \end{bmatrix}$$

Adding the two gives,

$$A = \begin{bmatrix} 0.5 & -0.25 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 \\ -0.25 & 0.75 & -0.25 & 0 & -0.25 & 0 & 0 & 0 & 0 \\ 0 & -0.25 & 0.5 & 0 & 0 & -0.25 & 0 & 0 & 0 \\ -0.25 & 0 & 0 & 0.75 & -0.25 & 0 & -0.25 & 0 & 0 \\ 0 & -0.25 & 0 & -0.25 & 1.0 & -0.25 & 0 & -0.25 & 0 \\ 0 & 0 & -0.25 & 0 & -0.25 & 0.75 & 0 & 0 & -0.25 \\ 0 & 0 & 0 & -0.25 & 0 & 0 & 0.5 & -0.25 & 0 \\ 0 & 0 & 0 & 0 & -0.25 & 0 & -0.25 & 0.75 & -0.25 \\ 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & -0.25 & 0.5 \end{bmatrix}$$

The obtained matrix is similar to the one obtained earlier from the discrete FD equations for the grid points.

2.1.3 Structure of the source vector function

1. Write down the explicit expressions for the vector functions $\mathbf{f}_u$ and $\mathbf{f}_v$ in terms of A, $\mathbf{u}$, $\mathbf{v}$, etc.

Solution: The discretised vector form of the equations of u and v are:

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= -D_u A\mathbf{u} + \kappa(aI - \mathbf{u} + \mathbf{u}^2\mathbf{v}) \\ \frac{d\mathbf{v}}{dt} &= -D_v A\mathbf{v} + \kappa(bI - \mathbf{u}^2\mathbf{v}) \end{aligned}$$

In the matrix-vector format;

$$\frac{d\mathbf{w}}{dt} = \begin{bmatrix} \mathbf{u}' \\ \mathbf{v}' \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u(\mathbf{u},\mathbf{v}) \\ \mathbf{f}_v(\mathbf{u},\mathbf{v}) \end{bmatrix} = \begin{bmatrix} -D_u A\mathbf{u} + \kappa(aI - \mathbf{u} + \mathbf{u}^2\mathbf{v}) \\ -D_v A\mathbf{v} + \kappa(bI - \mathbf{u}^2\mathbf{v}) \end{bmatrix} = \mathbf{f}_w(\mathbf{w}) \tag{11}$$

2.2 Numerical time-integration

2.2.1 Forward-Euler Method

1. Write down the iteration formula of the Forward-Euler (FE) Method in block-matrix form.

Solution: In the block-matrix form, the Forward-Euler time integration formula will be;

$$\mathbf{w}^{k+1} = \begin{bmatrix} \mathbf{u}^{k+1} \\ \mathbf{v}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^k + h_t[-D_u A\mathbf{u}^k + \kappa(aI - \mathbf{u}^k + (\mathbf{u}^k)^2\mathbf{v}^k)] \\ \mathbf{v}^k + h_t[-D_v A\mathbf{v}^k + \kappa(bI - (\mathbf{u}^k)^2\mathbf{v}^k)] \end{bmatrix} \tag{12}$$

2.2.2 Local stability of the FE Method

1. Derive the stability condition for the FE Method around the point $(\mathbf{0}, \mathbf{0})$.
   1. Write down the explicit form of the Jacobian matrix $J(\mathbf{u}, \mathbf{v})$ for your system of ODE's
   2. Write down $J_0 = J(\mathbf{0}, \mathbf{0})$ in the block-matrix form using the negative Laplacian FD matrix A
   3. Show that the eigenvalues of J_0 are negative, assuming that the eigenvalues of A are positive
   4. Find an upper estimate of $\max_j(-\lambda_j)$ by applying the Gerschgorin Lemma to the matrix $-J_0$
   5. Use this estimate to write down the stability condition on h_t and the number of time steps N_t in terms of the problem parameters D_u, D_v, κ, T, and the spatial grid-step h

Solution:

1. In general, the explicit form of the Jacobian matrix $J(\mathbf{u}, \mathbf{v})$;

$$J(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} \frac{\partial \mathbf{f}_u}{\partial \mathbf{u}} & \frac{\partial \mathbf{f}_u}{\partial \mathbf{v}} \\ \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}} & \frac{\partial \mathbf{f}_v}{\partial \mathbf{v}} \end{bmatrix}$$

calculating each term, we get:

$$\begin{aligned} \frac{\partial \mathbf{f}_u}{\partial \mathbf{u}} &= -D_u A + \kappa(2\mathbf{u}\mathbf{v} - 1) \\ \frac{\partial \mathbf{f}_u}{\partial \mathbf{v}} &= \kappa(\mathbf{u}^2) \\ \frac{\partial \mathbf{f}_v}{\partial \mathbf{u}} &= -\kappa(2\mathbf{u}\mathbf{v}) \\ \frac{\partial \mathbf{f}_v}{\partial \mathbf{v}} &= -D_v A - \kappa(\mathbf{u}^2) \end{aligned}$$

Therefore, the Jacobian for our case will be:

$$J(\mathbf{u},\mathbf{v}) = \begin{bmatrix} -D_u A + \kappa(2\mathbf{uv} - I) & \kappa(\mathbf{u}^2) \\ -\kappa(2\mathbf{uv}) & -D_v A - \kappa(\mathbf{u}^2) \end{bmatrix} \tag{13}$$

2. For $(\mathbf{u},\mathbf{v}) = (\mathbf{0},\mathbf{0})$, the Jacobian becomes:

$$J(\mathbf{u},\mathbf{v}) = \begin{bmatrix} -D_u A - \kappa I & \mathbf{0} \\ \mathbf{0} & -D_v A \end{bmatrix} \tag{14}$$

3. J_0 is a block-matrix with matrices on it's principal diagonal. Therefore, It's eigen values will be the eigen values of the block matrices.
For the first (or u) block, $\lambda_{J_0}^{(u)} = -D_u\lambda_A - \kappa$. Since $D_u, \kappa > 0$ and eigen values of A, $\lambda_A > 0$, the eigen values of the u-block are strictly negative, i.e., $\lambda_{J_0}^{(u)} < 0$.
For the second (or v) block, $\lambda_{J_0}^{(v)} = -D_v\lambda_A$. Since $D_v > 0$ and eigen values of A, $\lambda_A > 0$, the eigen values of the v-block are also strictly negative, i.e., $\lambda_{J_0}^{(v)} < 0$.
Therefore, the eigen values of the J_0 matrix are negative, i.e. $\lambda_{J_0} < 0$.

4. From the Gerschgorin Lemma, we know that:

$$|-\lambda_M - m_{ii}| \leq \sum_{j\neq i} |m_{ij}|$$

where, λ_M is the eigen value of matrix M, and m_{ii} are the diagonal elements of this matrix. For our case $M = -J_0$.
The eigen values of J_0 are $D_u\lambda_A + \kappa$ for the u-block, and $D_v\lambda_A$ for the v-block. Using the Gerschgorin Lemma;

$$\begin{aligned}
&\Longrightarrow \quad |-\lambda_j - (D_u A_{ii} + \kappa)| \leq \sum_{j\neq i} |D_u A_{ij}| \\
&\Longrightarrow \quad -\sum_{j\neq i} |D_u A_{ij}| \leq -\lambda_j - (D_u A_{ii} + \kappa) \leq \sum_{j\neq i} |D_u A_{ij}| \\
&\Longrightarrow \quad -\lambda_j \leq (D_u A_{ii} + \kappa) + \sum_{j\neq i} |D_u A_{ij}| \\
&\Longrightarrow \quad \max_j(-\lambda_j) \leq \max_i((D_u A_{ii} + \kappa) + \sum_{j\neq i} |D_u A_{ij}|) = M_u
\end{aligned}$$

Similarly for the v-block;

$$
\begin{aligned}
\implies \quad & |-\lambda_j - D_v A_{ii}| \leq \sum_{j\neq i} |D_v A_{ij}| \\
\implies \quad & -\sum_{j\neq i} |D_v A_{ij}| \leq -\lambda_j - D_v A_{ii} \leq \sum_{j\neq i} |D_v A_{ij}| \\
\implies \quad & -\lambda_j \leq (D_u A_{ii} + \kappa) + \sum_{j\neq i} |D_u A_{ij}| \\
\implies \quad & \max_j(-\lambda_j) \leq \max_i(D_v A_{ii} + \sum_{j\neq i} |D_v A_{ij}|) = M_v
\end{aligned}
$$

5. Stability condition on h_t for the u- and the v- block respectively, using the Gerschgorin inequality derived above, are:

$$h_t \leq \frac{2}{M_u}, \quad h_t \leq \frac{2}{M_v}$$

Combining the 2 inequalities yields,

$$
\begin{aligned}
h_t \quad &\leq \quad \frac{2}{\max(M_u, M_v)} \\
\textbf{where, } M_u \quad &= \quad \max_i((D_u A_{ii} + \kappa) + \sum_{j\neq i} |D_u A_{ij}|), \\
M_v \quad &= \quad \max_i(D_v A_{ii} + \sum_{j\neq i} |D_v A_{ij}|)
\end{aligned}
$$

2.2.3 Backward-Euler Method

1. Write down the iteration formula of the Backward-Euler Method in block-matrix form.

Solution: In the block-matrix form, the Backward-Euler time integration formula will be;

$$\mathbf{w}^{k+1} = \begin{bmatrix} \mathbf{u}^{k+1} \\ \mathbf{v}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}^k + h_t[-D_u A\mathbf{u}^{k+1} + \kappa(aI - \mathbf{u}^{k+1} + (\mathbf{u}^{k+1})^2\mathbf{v}^{k+1})] \\ \mathbf{v}^k + h_t[-D_v A\mathbf{v}^{k+1} + \kappa(bI - (\mathbf{u}^{k+1})^2\mathbf{v}^{k+1})] \end{bmatrix} \tag{15}$$

2.2.4 Newton-Raphson algorithm for the BE Method

1. Let $\mathbf{u}^k$ and $\mathbf{u}^k$ be given. Using the general block-notation, write down the residual block-vector.

Solution: The residual vector that must be checked at the i-th iteration of the Newton-Raphson algorithm will be:

$$\begin{bmatrix} \mathbf{r}_u(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) \\ \mathbf{r}_v(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) \end{bmatrix} = \begin{bmatrix} ||\mathbf{u}^k + h_t[-D_u A\mathbf{u}_i^{k+1} + \kappa(aI - \mathbf{u}_i^{k+1} + (\mathbf{u}_i^{k+1})^2\mathbf{v}_i^{k+1})] - \mathbf{u}_i^{k+1}|| \\ ||\mathbf{v}^k + h_t[-D_v A\mathbf{v}_i^{k+1} + \kappa(bI - (\mathbf{u}_i^{k+1})^2\mathbf{v}_i^{k+1})] - \mathbf{v}_i^{k+1}|| \end{bmatrix}$$

2. Using the block-matrix notation for the Jacobian matrix of the system and the residual, write down the formula for the $(i+1)$-th iteration the Newton-Raphson algorithm.

Solution: The $(i+1)$th iteration for the BENR method will be calculated as follows:

$$\mathbf{w}_{i+1}^{k+1} = \begin{bmatrix} \mathbf{u}_{i+1}^{k+1} \\ \mathbf{v}_{i+1}^{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_i^{k+1} + [I - h_t J(\mathbf{u}_i^{k+1})]^{-1}[\mathbf{r}_u(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1})] \\ \mathbf{v}_i^{k+1} + [I - h_t J(\mathbf{v}_i^{k+1})]^{-1}[\mathbf{r}_v(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1})] \end{bmatrix}$$

3 Implementation

3.1 Simulations with the FE Method

1. empirical choice of a stable h_t or N_t for the FE Method.

Solution: Consider the following PDE:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta u - ku, \quad k > 0 \\ u^{k+1} &= u^k - h_t A u^k \\ u^{k+1} &= (1 - h_t A - h_t k)^k u_0 \\ u^{k+1} &= \sum_{i=1}^{n} b_i(0)(1 - h_t\lambda_i - h_t k)^k u_i \end{aligned}$$

For Forward Euler to be stable, $1 - h_t\lambda_i - h_t k < 1$. It is also known that $h_t > 0$. We can rewrite this as:

$$\begin{aligned}
|1 - h_t\lambda_i - h_t k| &< 1 \\
0 < h_t &< \frac{2}{\lambda_{i,\max} + k} \\
\therefore h_t &< \frac{2}{\lambda_{i,\max} + k}
\end{aligned} \tag{16}$$

Using (16), and $h_t = T/N_t$, the following table is established for $N_x = N_y = 25$;

Table 1: Minimum N_t for $N_x = N_y = 25$.

κ	2	5	10
N_t	3134	3164	3214

For $N_x = N_y = 100$, and $\kappa = 5$, the minimum value of $N_t = 50034$. To ensure convergence, we choose $N_t = 51000$.

2. The final code should implement one test run of a stable FE Method and print out the value of N_t and the CPU time.

Solution: For the case $N_x = N_y = 100$, and $\kappa = 5$, $N_t = 51000$ and the CPU time is 10.48 seconds.

3. Plots of the activator $u(x, y, t)$ and inhibitor $v(x, y, t)$ at $t = 0$ and $t = T$, obtained with the FE Method.

Solution:

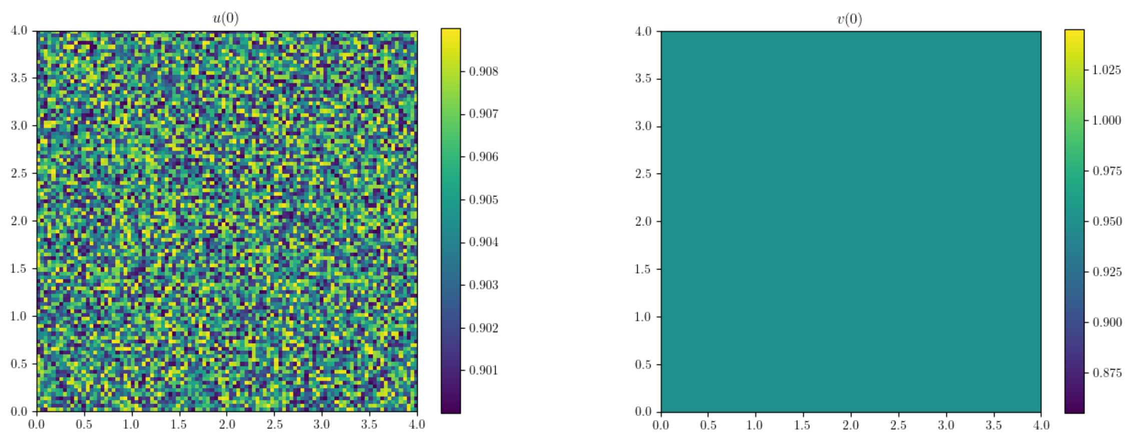

Figure 1: $u(x, y, 0)$ and $v(x, y, 0)$ for $N_x = N_y = 100$, $\kappa = 5$

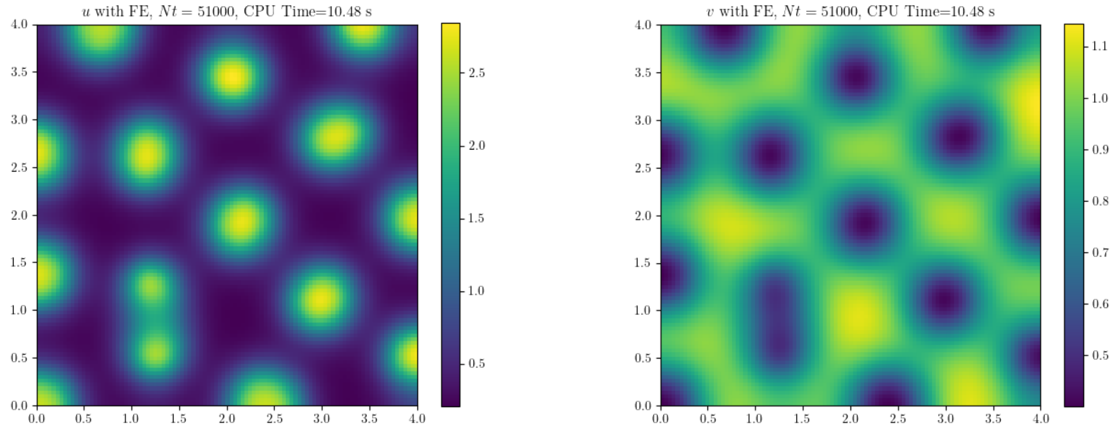

Figure 2: $u(x, y, T)$ and $v(x, y, T)$ for $N_x = N_y = 100$, $\kappa = 5$, using FE Method.

3.2 Simulations with the BE-NR Method

1. Empirical choice of N_t for the BE-NR Method such that:

   (a) The number of inner iterations of the NR algorithm required to achieve the tolerance is in the order of 3 iterations

   (b) The final result at T is similar to the one obtained by the FE Method

   (c) The execution time is minimized

Solution: The value of N_t is chosen after running the code at $N_t \in \{50, 100, 200, 400\}$, for $N_x = N_y = 25$, and $\kappa \in \{5, 10\}$ with a permitted tolerance of $\epsilon = 10^{-3}$. The mean of the norm of the residual was chosen to check the convergence of the method. The results along with the CPU time are tabulated below.

Table 2: Prescribed N_t for $N_x = N_y = 25$, with $\kappa \in \{5, 10\}$.

N_t	$\kappa = 5$		$\kappa = 10$	
	mean(residual)	CPU time(s)	mean(residual)	CPU time(s)
50	7.8432×10^{-5}	1.43	14715.5683	1.96
100	1.5847×10^{-4}	2.44	8.6676×10^{-5}	3.02
200	1.9207×10^{-4}	3.84	2.4723×10^{-4}	4.83
400	2.0501×10^{-4}	6.37	2.8032×10^{-4}	6.69

We can observe that κ plays a critical role in convergence. It sets a minimum value for N_t below which the solution begins to diverge– as is seen for the case $\kappa = 10$, $N_t = 50$. For our case, $\kappa = 5$, and thus an $N_t = 50$ should be sufficient to obtain the desired convergence for $N_x = N_y = 100$ case.

2. The final code should contain one test run of the BENR Method with an automatically chosen N_t and print out the value of N_t, the norm of the residual at each inner iteration, and the total CPU time.

Solution: For the case $N_x = N_y = 100$, and $\kappa = 5$, $N_t = 50$ and the CPU time is 185.01 seconds. The mean residual for this case is 1.2728×10^{-4} which is below the set tolerance of $\epsilon = 10^{-3}$.

3. Plots of the activator $u(x, y, t)$ and inhibitor $v(x, y, t)$ at $t = T$, obtained with the BE-NR Method for the same initial guess as the FE Method.

Solution:

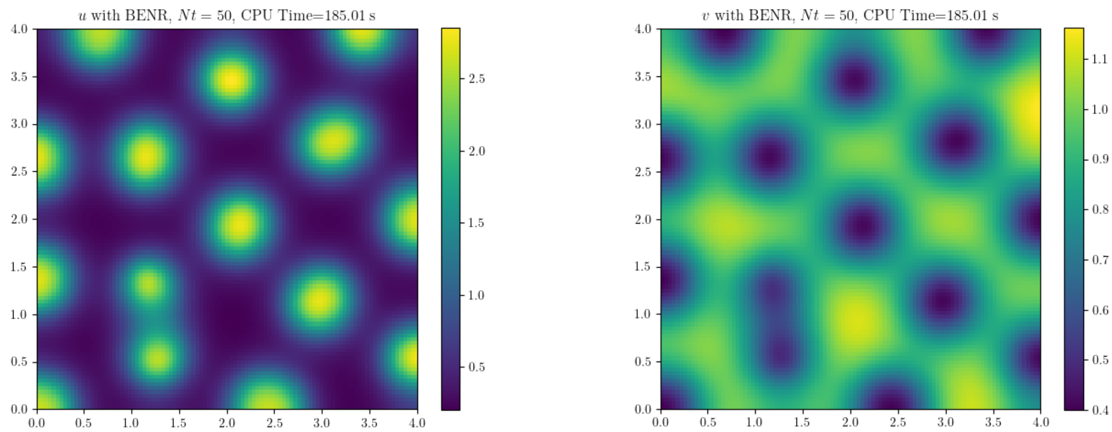

Figure 3: $u(x, y, T)$ and $v(x, y, T)$ for $N_x = N_y = 100$, $\kappa = 5$, using BENR Method.

The plots obtained for the Forward Euler method (Figure 2) concur with the results obtained for the Backward Euler Netwon-Rhapson Method (Figure 3).