ADI Time-Stepping for a 2D Reaction–Diffusion System on a Fixed Domain

Josh Dalal
ENM 5220 — Numerical Solution of PDEs
University of Pennsylvania

Spring 2026

Contents

1 Introduction

In 1952, Turing [5] showed that diffusion can sometimes destabilize a spatially uniform steady state when two reacting species diffuse at different rates. Instead of smoothing out perturbations, diffusion can cause spatial patterns such as stripes, spots, and labyrinth-like structures to form from nearly uniform initial conditions. Reaction–diffusion systems with this type of instability have been used to model biological pattern formation, chemical reactions, and morphogenesis.

The Schnakenberg model [4] is a well-known two-species reaction–diffusion system commonly used to study Turing patterns. On a fixed two-dimensional domain, the system transitions from stripe-like patterns at lower values of the scaling parameter γ to spot-like patterns at larger γ. As γ increases, the characteristic wavelength of the patterns also decreases.

Madzvamuse & Maini [1] studied how mesh structure affects numerical solutions of the Schnakenberg system on both fixed and growing domains. In Section 4 of their paper, they computed steady-state solutions using a Peaceman–Rachford ADI finite difference scheme on a uniform 150×150 grid for $\gamma = 114$, 1000, and 5000. One of their main observations was that the ADI scheme on a square mesh tends to produce more symmetric patterns than a finite element method on an unstructured mesh.

The main goal of this project was to reproduce the fixed-domain results from Section 4 of [1] using my own MATLAB implementation of the ADI scheme. I first verified the solver on a pure diffusion problem with a known exact solution before moving to the full Schnakenberg system. One thing that showed up pretty quickly was that the explicit reaction treatment used in the paper became unstable for the larger γ cases, especially at $\gamma = 1000$ and 5000. After trying a few alternatives, I ended up using a predictor–corrector style update for the reaction terms, which produced stable patterns while keeping the ADI diffusion sweeps unchanged.

2 Mathematical Formulation

The Schnakenberg reaction–diffusion system on the fixed unit square $[0,1]^2$ is

$$u_t = \nabla^2 u + \gamma\big(a - u + u^2 v\big), \tag{1}$$

$$v_t = d\,\nabla^2 v + \gamma\big(b - u^2 v\big), \tag{2}$$

with zero-flux boundary conditions $\partial u/\partial n = \partial v/\partial n = 0$ on $\partial\Omega$. The reaction terms are $f(u,v) = a - u + u^2 v$ and $g(u,v) = b - u^2 v$, where a and b are positive constants. For Turing patterns to form, the inhibitor species v must diffuse faster than the activator u, so the diffusion ratio must satisfy $d > 1$.

The homogeneous steady state is found by setting the reaction terms equal to zero, giving

$$u^* = a + b, \qquad v^* = \frac{b}{(a+b)^2}.$$

The parameter values used in this project follow [1]: $a = 0.1$, $b = 0.9$, and $d = 10$, which gives the steady state $(u^*, v^*) = (1.0,\ 0.9)$. The scaling parameter γ is varied over $\{114,\ 1000,\ 5000\}$.

Initial conditions are taken as small random perturbations around the steady state:

$$u(x,y,0) = u^* + \epsilon\,\xi(x,y), \qquad v(x,y,0) = v^* + \epsilon\,\xi(x,y),$$

where $\xi(x,y) \in [-1,1]$ is uniformly distributed random noise and $\epsilon = 0.01$. The domain is discretized using a uniform grid with $\Delta x = \Delta y = 1/N$ and $N = 150$.

3 Numerical Method

3.1 The Peaceman–Rachford ADI Scheme for the Heat Equation

Consider the scalar heat equation on $[0,1]^2$:

$$u_t = u_{xx} + u_{yy}.$$

The domain is discretized on a uniform grid $x_i = i\Delta x$, $y_j = j\Delta y$, with $i, j = 0, \ldots, N$, and $u^n_{i,j} \approx u(x_i, y_j, t_n)$. The ADI method splits each timestep Δt into two half-steps, alternating which spatial direction is treated implicitly [2].

First half-step (implicit in x, explicit in y):

$$\frac{u^{n+1/2}_{i,j} - u^n_{i,j}}{\Delta t/2} = \delta_x^2 u^{n+1/2}_{i,j} + \delta_y^2 u^n_{i,j},$$

where $\delta_x^2 u_{i,j} = (u_{i-1,j} - 2u_{i,j} + u_{i+1,j})/\Delta x^2$ and $\delta_y^2 u_{i,j} = (u_{i,j-1} - 2u_{i,j} + u_{i,j+1})/\Delta y^2$. Defining $r_x = \Delta t/(2\Delta x^2)$ and $r_y = \Delta t/(2\Delta y^2)$, the equation becomes

$$-r_x\, u^{n+1/2}_{i-1,j} + (1 + 2r_x)\, u^{n+1/2}_{i,j} - r_x\, u^{n+1/2}_{i+1,j} = r_y\, u^n_{i,j-1} + (1 - 2r_y)\, u^n_{i,j} + r_y\, u^n_{i,j+1}. \tag{3}$$

For each fixed value of j, eq. (3) is a tridiagonal system in the x-direction, which can be solved efficiently using the Thomas algorithm (section 3.5).

Second half-step (explicit in x, implicit in y):

$$\frac{u^{n+1}_{i,j} - u^{n+1/2}_{i,j}}{\Delta t/2} = \delta_x^2 u^{n+1/2}_{i,j} + \delta_y^2 u^{n+1}_{i,j}.$$

Rearranging gives

$$-r_y\, u^{n+1}_{i,j-1} + (1 + 2r_y)\, u^{n+1}_{i,j} - r_y\, u^{n+1}_{i,j+1} = r_x\, u^{n+1/2}_{i-1,j} + (1 - 2r_x)\, u^{n+1/2}_{i,j} + r_x\, u^{n+1/2}_{i+1,j}. \tag{4}$$

For each fixed value of i, eq. (4) is a tridiagonal system in the y-direction.

For the linear diffusion equation, the ADI scheme is unconditionally stable and second-order accurate in both space and time. When nonlinear reaction terms are added, additional stability issues appear, which are discussed in section 3.4.

Analysis. One advantage of the ADI splitting is that each timestep only requires $2N$ tridiagonal solves of size N, giving an overall computational cost of $O(N^2)$ per timestep instead of the $O(N^3)$ cost of solving a fully implicit 2D system directly.

The method is also second-order accurate in time. Adding the two half-step equations gives a scheme equivalent to Crank–Nicolson plus a correction term of the form

$$\frac{\Delta t^2}{4} \delta_x^2 \delta_y^2 (u^{n+1} - u^n).$$

Since $u^{n+1} - u^n = O(\Delta t)$, this correction term is $O(\Delta t^3)$ locally and does not affect the overall second-order accuracy of the method.

The ADI scheme is unconditionally stable for the linear diffusion problem. Using a von Neumann analysis with $u_{i,j}^n = g^n e^{i(k_x x_i + k_y y_j)}$, and defining $\xi_x = 4r_x \sin^2(k_x \Delta x/2)$ and $\xi_y = 4r_y \sin^2(k_y \Delta y/2)$, the amplification factor becomes

$$g = \frac{(1-\xi_x)(1-\xi_y)}{(1+\xi_x)(1+\xi_y)}.$$

Since $\xi_x, \xi_y \geq 0$, it follows that $|g| \leq 1$ for all wavenumbers, so there is no CFL-type timestep restriction on Δt for the diffusion part of the problem. This is a major advantage over explicit methods, where stability requires a very small timestep as the grid is refined.

3.2 Neumann Boundary Condition Treatment

Zero-flux boundary conditions require $\partial u/\partial x = 0$ at $x = 0, 1$ and $\partial u/\partial y = 0$ at $y = 0, 1$. These boundary conditions are enforced using ghost points at each ADI half-step, which preserves second-order spatial accuracy.

At the left boundary ($i = 0$), applying a centered difference approximation to the Neumann condition gives

$$\frac{u_{1,j} - u_{-1,j}}{2\Delta x} = 0 \qquad \Rightarrow \qquad u_{-1,j} = u_{1,j}.$$

Substituting this into the second derivative stencil gives

$$\delta_x^2 u_{0,j} = \frac{u_{-1,j} - 2u_{0,j} + u_{1,j}}{\Delta x^2} = \frac{2u_{1,j} - 2u_{0,j}}{\Delta x^2}.$$

This effectively doubles the off-diagonal coefficient at the boundary row. By symmetry, the same idea applies at the right boundary, where $u_{N+1,j} = u_{N-1,j}$ for $i = N$. The identical procedure is used during the y-sweep at $j = 0$ and $j = N$, replacing r_x with r_y.

The resulting tridiagonal system for the x-sweep can be written as

$$\mathbf{A}_x \mathbf{u}_j^{n+1/2} = \mathbf{b}_j,$$

where

$$\mathbf{A}_x = \begin{pmatrix} 1+2r_x & -2r_x & & & \\ -r_x & 1+2r_x & -r_x & & \\ & \ddots & \ddots & \ddots & \\ & & -r_x & 1+2r_x & -r_x \\ & & & -2r_x & 1+2r_x \end{pmatrix}.$$

The interior rows use the standard centered-difference stencil, while the boundary rows contain the modified coefficients coming from the ghost-point treatment. The matrix for the y-sweep, $\mathbf{A}_y$, has the same form. The two sweeps are handled independently, with the boundary modification applied only in the direction currently being solved implicitly.

3.3 Extension to the Coupled Schnakenberg System

The diffusion terms in eqs. (1) and (2) act separately on each species, so the ADI discretization gives one tridiagonal system for u and another for v at each half-step. Define the diffusion numbers

$$r_x^u = \frac{\Delta t}{2\Delta x^2}, \qquad r_x^v = \frac{d\,\Delta t}{2\Delta x^2} = d\,r_x^u,$$

and similarly for r_y^u and r_y^v. The reaction terms evaluated at time level n are written as

$$F_{i,j}^n = f(u_{i,j}^n, v_{i,j}^n), \qquad G_{i,j}^n = g(u_{i,j}^n, v_{i,j}^n),$$

with the reaction stabilization method discussed later in section 3.4.

First half-step (implicit in x, explicit in y):

$$-r_x^u\,u_{i-1,j}^{n+1/2} + (1+2r_x^u)\,u_{i,j}^{n+1/2} - r_x^u\,u_{i+1,j}^{n+1/2} = r_y^u\,u_{i,j-1}^n + (1-2r_y^u)\,u_{i,j}^n + r_y^u\,u_{i,j+1}^n + \tfrac{\Delta t}{2}\gamma F_{i,j}^n, \quad (5)$$

$$-r_x^v\,v_{i-1,j}^{n+1/2} + (1+2r_x^v)\,v_{i,j}^{n+1/2} - r_x^v\,v_{i+1,j}^{n+1/2} = r_y^v\,v_{i,j-1}^n + (1-2r_y^v)\,v_{i,j}^n + r_y^v\,v_{i,j+1}^n + \tfrac{\Delta t}{2}\gamma G_{i,j}^n. \quad (6)$$

For each fixed value of j, these become two independent tridiagonal systems in the x-direction.

Second half-step (explicit in x, implicit in y):

$$-r_y^u\,u_{i,j-1}^{n+1} + (1+2r_y^u)\,u_{i,j}^{n+1} - r_y^u\,u_{i,j+1}^{n+1} = r_x^u\,u_{i-1,j}^{n+1/2} + (1-2r_x^u)\,u_{i,j}^{n+1/2} + r_x^u\,u_{i+1,j}^{n+1/2} + \tfrac{\Delta t}{2}\gamma F_{i,j}^n, \quad (7)$$

$$-r_y^v\,v_{i,j-1}^{n+1} + (1+2r_y^v)\,v_{i,j}^{n+1} - r_y^v\,v_{i,j+1}^{n+1} = r_x^v\,v_{i-1,j}^{n+1/2} + (1-2r_x^v)\,v_{i,j}^{n+1/2} + r_x^v\,v_{i+1,j}^{n+1/2} + \tfrac{\Delta t}{2}\gamma G_{i,j}^n. \quad (8)$$

For each fixed value of i, these become two independent tridiagonal systems in the y-direction.

The coefficient matrices $\mathbf{A}_x^u$, $\mathbf{A}_x^v$, $\mathbf{A}_y^u$, and $\mathbf{A}_y^v$ all have the same tridiagonal structure described in section 3.2, with coefficients scaled appropriately by r^u or r^v. Since these matrices do not change in time, they are assembled and factored once at initialization and then reused throughout the simulation.

3.4 Reaction Term Treatment

Why straightforward explicit treatment becomes unstable. At the parameter values used in the paper ($N = 150$, $\Delta t = 0.01$), the diffusion number is

$$\frac{\Delta t}{\Delta x^2} = \frac{0.01}{(1/150)^2} = 225.$$

Although the ADI diffusion sweeps are unconditionally stable, the reaction terms are still being treated explicitly, and this creates stability problems at large values of γ.

To understand why, the reaction terms can be linearized around the steady state $(u^*, v^*) = (1, 0.9)$. The Jacobian matrix is

$$J_F = \gamma \begin{pmatrix} -1 + 2u^*v^* & (u^*)^2 \\ -2u^*v^* & -(u^*)^2 \end{pmatrix} = \gamma \begin{pmatrix} 0.8 & 1 \\ -1.8 & -1 \end{pmatrix}.$$

The eigenvalues are

$$\gamma(-0.1 \pm 0.995i),$$

which are nearly purely imaginary and scale in magnitude with γ. For $\gamma = 5000$, a forward Euler treatment of the reaction terms requires roughly

$$\Delta t \lesssim \frac{1}{|\lambda_{\max}|} \approx 2 \times 10^{-4}$$

to remain stable, which is much smaller than the timestep used in [1].

Several different approaches were tested during development:

1. *Strang splitting with a separate reaction solve.*

   Reaction and diffusion were solved in separate substeps, with the reaction part handled using an implicit Newton iteration. While this approach is mathematically reasonable, the resulting solutions often decayed back toward the homogeneous steady state instead of forming patterns. Since Turing instability depends on the interaction between reaction and diffusion, separating the two too aggressively appeared to weaken the instability mechanism at the timesteps used here.

2. *Semi-implicit linearized reaction treatment.*

   Another attempt was to absorb the linearized reaction terms into the implicit tridiagonal solve. In practice this caused the spatially uniform mode to grow without bound when $f_u^* > 0$, leading to instability in the mean solution.

3. *Fully explicit reaction with adaptive timestep.*

   Reducing Δt enough to satisfy the explicit stability condition did improve stability, but the nearly imaginary eigenvalues of the Jacobian still caused noticeable oscillations in time when forward Euler was used directly.

Predictor–corrector reaction treatment. The approach that ultimately worked best was to keep diffusion implicit through the ADI sweeps while treating the reaction terms using a predictor–corrector scheme that approximates Crank–Nicolson time integration.

Each timestep proceeds as follows:

1. **Predictor step:** Evaluate the reaction terms at the current state,

   $$F^n = \gamma f(u^n, v^n), \qquad G^n = \gamma g(u^n, v^n),$$

   and perform one full ADI sweep to obtain a predicted state (u^p, v^p).

2. **Corrector step:** Recompute the reaction terms at the predicted state,

   $$F^p = \gamma f(u^p, v^p), \qquad G^p = \gamma g(u^p, v^p),$$

and perform a second ADI sweep using the averaged reaction terms

$$\tfrac{1}{2}(F^n + F^p), \qquad \tfrac{1}{2}(G^n + G^p).$$

This produces the updated solution (u^{n+1}, v^{n+1}).

In practice, this predictor–corrector approach removed the temporal oscillations seen with forward Euler and produced stable Turing patterns across all tested values of γ.

The timestep was chosen adaptively according to

$$\Delta t = \min\left(\Delta t_{\text{user}}, \frac{0.5}{\gamma\,\rho(J_F)}\right),$$

where $\rho(J_F) \approx 2$ is the approximate spectral radius of the Jacobian near the steady state. For the parameter values used here, this gives approximately

$$\Delta t \approx 2.2 \times 10^{-3} \quad \text{for } \gamma = 114,$$

$$\Delta t \approx 2.5 \times 10^{-4} \quad \text{for } \gamma = 1000,$$

and

$$\Delta t \approx 5 \times 10^{-5} \quad \text{for } \gamma = 5000.$$

Computational cost. The predictor–corrector method requires two complete ADI sweeps per timestep, one for the predictor and one for the corrector. This results in $4N$ tridiagonal solves of size $N + 1$ per species at each timestep. The overall complexity remains $O(N^2)$ per timestep, although with roughly twice the cost of the single-sweep ADI method described in section 3.3.

Difference from the original paper. Madzvamuse & Maini [1] describe using explicit reaction evaluation within a single ADI sweep while taking $\Delta t = 0.01$ on a 150×150 grid. In our implementation, this became unstable for $\gamma \geq 1000$, so the predictor–corrector approach with adaptive timestepping was used instead. The diffusion discretization and boundary condition treatment remain the same as in the original paper, and the final steady-state patterns are still qualitatively consistent with the published results.

3.5 Thomas Algorithm

Each ADI half-step requires solving a tridiagonal linear system of size $N + 1$:

$$\begin{pmatrix} b_0 & c_0 & & & \\ a_1 & b_1 & c_1 & & \\ & \ddots & \ddots & \ddots & \\ & & a_{N-1} & b_{N-1} & c_{N-1} \\ & & & a_N & b_N \end{pmatrix} \mathbf{x} = \mathbf{d}.$$

For the ADI discretization used here, the matrix coefficients are constant in time. The interior rows use

$$a_i = -r, \qquad b_i = 1 + 2r, \qquad c_i = -r,$$

while the boundary rows are modified according to the ghost-point treatment from section 3.2, giving $c_0 = -2r$ and $a_N = -2r$. The value of r depends on the current sweep and corresponds to one of

$$r_x^u, \quad r_x^v, \quad r_y^u, \quad r_y^v.$$

These systems are solved using the Thomas algorithm, which is a simplified Gaussian elimination method specialized for tridiagonal matrices.

Forward sweep. The modified diagonal and right-hand side are initialized as

$$\tilde{b}_0 = b_0, \qquad \tilde{d}_0 = d_0.$$

Then for $i = 1, \ldots, N$:

$$m_i = \frac{a_i}{\tilde{b}_{i-1}}, \qquad \tilde{b}_i = b_i - m_i c_{i-1}, \qquad \tilde{d}_i = d_i - m_i \tilde{d}_{i-1}.$$

Back substitution. After the forward sweep, the final unknown is computed from

$$x_N = \frac{\tilde{d}_N}{\tilde{b}_N}.$$

The remaining entries are then recovered recursively for $i = N-1, \ldots, 0$:

$$x_i = \frac{\tilde{d}_i - c_i x_{i+1}}{\tilde{b}_i}.$$

Each tridiagonal solve requires only $O(N)$ operations, so the total cost of all ADI sweeps in a timestep scales as $O(N^2)$.

Since the matrix coefficients do not change during the simulation, the modified diagonals $\tilde{b}_i$ and multipliers m_i can be precomputed once at initialization and reused throughout the computation. At each timestep, only the right-hand side update and back substitution need to be performed.

4 Verification: Pure Diffusion Convergence Test

Before adding the reaction terms, I first tested the ADI solver on a pure diffusion problem with a known exact solution. The test function

$$u(x, y, t) = e^{-2\pi^2 t} \cos(\pi x) \cos(\pi y)$$

satisfies

$$u_t = u_{xx} + u_{yy}$$

on $[0, 1]^2$ with the zero-flux Neumann boundary conditions exactly. This makes it a good verification problem because the boundary conditions themselves do not introduce any additional discretization error.

The solution was computed to $t = 0.5$ using a fixed timestep $\Delta t = 10^{-4}$. I chose this timestep small enough that the spatial error would dominate over the time discretization error. The solver was then run on a sequence of increasingly refined grids.

The discrete L_2 error was computed using

$$\|e\|_2 = \Delta x \left(\sum_{i,j} \big(u_{i,j} - u(x_i, y_j, t)\big)^2 \right)^{1/2}.$$

N	Δx	L_2 error	Rate
10	1.000×10^{-1}	2.6152×10^{-6}	—
20	5.000×10^{-2}	5.8270×10^{-7}	2.17
40	2.500×10^{-2}	1.3807×10^{-7}	2.08
80	1.250×10^{-2}	3.3620×10^{-8}	2.04
160	6.250×10^{-3}	8.2831×10^{-9}	2.02

Table 1: L_2 errors and convergence rates for the pure diffusion ADI solver. The observed convergence rates approach 2 as the grid is refined, confirming second-order spatial accuracy.

The convergence rates approach 2 as the grid is refined (table 1 and fig. 1), which matches the expected second-order accuracy of the centered finite difference discretization. The slightly larger rates on the coarser grids are not unusual and decrease as the mesh is refined.

This test also helped verify that the Thomas solver, ghost-point Neumann boundary conditions, and ADI splitting were all implemented correctly before moving to the full reaction–diffusion system.

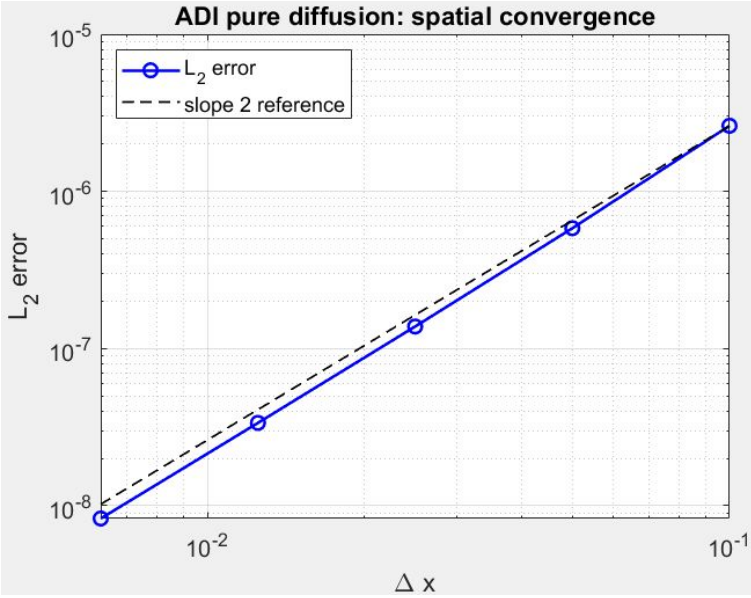

Figure 1: Log-log plot of L_2 error versus Δx for the pure diffusion ADI solver. The dashed line shows the expected slope-2 behavior.

5 Results

The ADI solver with the predictor–corrector reaction treatment was applied to the Schnakenberg system on a 150×150 uniform grid with zero-flux boundary conditions. Simulations were run to a

final time of $t_F = 5.0$ for $\gamma = 114$, 1000, and 5000, using the adaptive timestep strategy described in section 3.4.

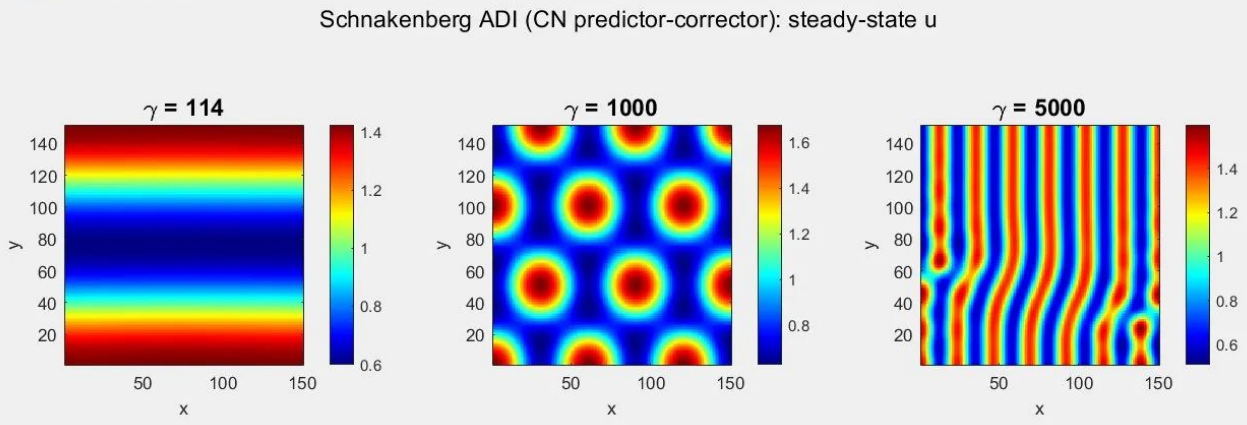

Figure 2: Steady-state u concentration for $\gamma = 114$ (left), $\gamma = 1000$ (center), and $\gamma = 5000$ (right). Compare with Figure 1 (left column) of [1].

Figure 2 shows the steady-state concentration field for u at the three tested values of γ. The patterns match the results from [1] pretty well overall.

- For $\gamma = 114$, the solution develops broad horizontal stripe patterns with values of u ranging from about 0.60 to 1.43. These are the largest-scale structures that appear in the domain.
- For $\gamma = 1000$, the solution transitions to a spot-like pattern. The spots end up arranged in a fairly symmetric grid, which is exactly the kind of behavior the paper observed on uniform square meshes.
- For $\gamma = 5000$, the solution develops much finer spatial structure, with a mixture of spots and stripe fragments and $u \in [0.51, 1.60]$. Increasing γ decreases the pattern wavelength, so the domain fills with many smaller features instead of a few large ones.

This also matches the linear stability picture for the Schnakenberg system. If eqs. (1) and (2) is linearized around the steady state and perturbations of the form

$$e^{\sigma t} \cos(k\pi x) \cos(l\pi y)$$

are substituted in, the resulting dispersion relation has the form

$$\sigma = \sigma(k^2 + l^2; \gamma, d).$$

As γ increases, the unstable modes shift toward larger wavenumbers, which means the dominant patterns have shorter wavelengths. That is consistent with what shows up in the simulations: a few broad stripes at $\gamma = 114$ and much finer spot-like structures by $\gamma = 5000$.

Table 2 summarizes the computational cost of each simulation. Because the adaptive timestep scales approximately as $O(1/\gamma)$, the $\gamma = 5000$ case requires substantially more timesteps than the $\gamma = 114$ case. Most of this additional cost comes from the reaction stability restriction in the predictor step. A fully implicit treatment of the reaction terms could potentially remove this timestep restriction, although it would require solving a nonlinear system at each timestep.

γ	Δt	Steps	$\max(u)$	$\min(u)$	Wall time (s)
114	2.19×10^{-3}	2,280	1.426	0.598	8
1000	2.50×10^{-4}	20,000	1.676	0.628	69
5000	5.00×10^{-5}	100,000	1.598	0.507	437

Table 2: Summary of timestep sizes, runtime statistics, and solution ranges for each value of γ.

5.1 Mesh Sensitivity Study

One of the main conclusions of [1] is that an ADI scheme on a uniform square mesh tends to favor symmetric, grid-aligned patterns. To investigate this effect, the $\gamma = 1000$ case was repeated on four different grid resolutions,

$$N \in \{50, 100, 150, 200\},$$

while keeping all other parameters fixed. The resulting steady-state solutions are shown in fig. 3.

N	$\max(u)$	$\min(u)$	Spot count	Pattern regularity
50	1.713	0.520	8 (irregular)	low
100	1.686	0.470	6–7 (diagonal)	medium
150	1.676	0.628	6 (2×3 grid)	high
200	1.676	0.627	6 (2×3 grid)	high

Table 3: Steady-state statistics for the mesh sensitivity study at $\gamma = 1000$.

Several trends show up in the results:

1. **Coarse grids give less regular patterns.**

   At $N = 50$, several of the spots are distorted or partially merged together, and the overall pattern is pretty asymmetric. The $N = 100$ case looks more organized, but the spots are still arranged diagonally and are not strongly aligned with the computational grid.

2. **Fine grids converge toward the same symmetric structure.**

   The $N = 150$ and $N = 200$ runs both produce almost the same 2×3 spot arrangement, with very similar values of $\max(u)$ and $\min(u)$. At these resolutions the spots become much more symmetric and clearly line up with the square mesh.

3. **The transition happens fairly suddenly.**

   The minimum value of u changes quite a bit between $N = 100$ and $N = 150$, but then barely changes between $N = 150$ and $N = 200$. The solution does not seem to converge smoothly toward the final pattern. Instead, it appears to lock into a particular symmetric configuration once the grid becomes fine enough.

Overall, the mesh study ended up matching the main observation from [1] pretty closely. As the grid gets finer, the ADI solutions become much more symmetric and strongly aligned with the square mesh. The paper reports that the finite element solutions on unstructured meshes do not show this same behavior nearly as strongly, which makes it seem likely that some of the symmetry in the ADI results is coming from the grid itself rather than only from the underlying PDE.

6 Fourier Spectral (IMEX) Solver

6.1 Setup

For the periodic boundary condition case, the parameter values from [3], as referenced in [1] Section 4.2, are used:

$$a = 0.126779, \qquad b = 0.792366, \qquad d = 10,$$

with

$$\gamma \in \{114,\ 1000,\ 5000\}.$$

The domain is again $[0,1]^2$, but now with periodic boundary conditions:

$$u(0,y,t) = u(1,y,t), \qquad u(x,0,t) = u(x,1,t),$$

and similarly for v.

The domain is discretized on a uniform grid

$$x_i = \frac{i}{N}, \qquad y_j = \frac{j}{N},$$

for $i,j = 0,\dots,N-1$. Unlike the Neumann case, only N grid points are needed in each direction because the periodic endpoints are identified with one another.

The spectral method is based on the 2D Discrete Fourier Transform (DFT):

$$\hat{U}_{k,l} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} U_{i,j}\, e^{-2\pi \mathrm{i}(ki+lj)/N},$$

with inverse transform

$$U_{i,j} = \frac{1}{N^2}\sum_{k,l} \hat{U}_{k,l}\, e^{2\pi \mathrm{i}(ki+lj)/N}.$$

The main reason the Fourier approach is useful here is that spatial derivatives become simple multiplications in Fourier space. For the second derivatives,

$$\widehat{\left(\frac{\partial^2 U}{\partial x^2}\right)}_{k,l} = -(2\pi k)^2\, \hat{U}_{k,l},$$

and

$$\widehat{\left(\frac{\partial^2 U}{\partial y^2}\right)}_{k,l} = -(2\pi l)^2\, \hat{U}_{k,l}.$$

Adding the two pieces together gives the Fourier-space Laplacian:

$$\mu_{k,l} = -(2\pi)^2(k^2+l^2).$$

So instead of solving a large coupled diffusion system in physical space, each Fourier mode can be updated independently. This is what makes the implicit diffusion step in the spectral solver so simple and fast for periodic domains.

6.2 IMEX Time Discretization

Applying Crank–Nicolson time discretization to the diffusion terms and forward Euler to the reaction terms gives the following IMEX update for u:

$$\frac{\hat{u}_{k,l}^{n+1} - \hat{u}_{k,l}^{n}}{\Delta t} = \mu_{k,l} \frac{\hat{u}_{k,l}^{n+1} + \hat{u}_{k,l}^{n}}{2} + \gamma \widehat{F^n}_{k,l},$$

where

$$\widehat{F^n}_{k,l} = \mathcal{F}\{f(u^n, v^n)\}_{k,l}$$

is the Fourier transform of the nonlinear reaction term evaluated in physical space.

Solving for $\hat{u}_{k,l}^{n+1}$ gives

$$\hat{u}_{k,l}^{n+1} = \underbrace{\frac{1 + \frac{\Delta t}{2}\mu_{k,l}}{1 - \frac{\Delta t}{2}\mu_{k,l}}}_{P_{k,l}} \hat{u}_{k,l}^{n} + \underbrace{\frac{\Delta t}{1 - \frac{\Delta t}{2}\mu_{k,l}}}_{Q_{k,l}} \gamma \widehat{F^n}_{k,l}.$$

Since

$$\mu_{k,l} \leq 0,$$

the denominator

$$1 - \frac{\Delta t}{2}\mu_{k,l}$$

is always positive, so the update is well-defined for all Fourier modes. The same update formula is used for v, replacing $\mu_{k,l}$ with $d\,\mu_{k,l}$.

Predictor–corrector reaction treatment. As with the ADI solver discussed in section 3.4, treating the reaction terms with straightforward forward Euler becomes unstable at large values of γ. The main issue is that the reaction Jacobian has eigenvalues that are close to purely imaginary, which leads to oscillatory growth in explicit time integration.

To improve stability, a predictor–corrector treatment of the reaction terms was used. This gives an effective trapezoidal-rule (Crank–Nicolson style) update for the reaction contribution.

The same stability issue from the ADI solver showed up here as well. The reaction Jacobian has eigenvalues that are close to purely imaginary, so a plain forward Euler treatment tends to introduce oscillations that grow over time. The trapezoidal rule behaves much better in this situation since its amplification factor satisfies

$$|g| = \left| \frac{1 + z/2}{1 - z/2} \right| = 1$$

for purely imaginary

$$z = \mathrm{i}\omega\Delta t.$$

Forward Euler instead gives

$$|g| = |1 + z| > 1,$$

so the oscillations get amplified rather than damped out.

To avoid that issue, the reaction terms are handled with the same predictor–corrector idea used in the ADI solver. Each timestep has two stages:

1. **Predictor step:**

   Apply the IMEX update using the reaction terms evaluated at the current solution to get a predicted state (u^p, v^p).

2. **Corrector step:**

   Recompute the reaction terms using the predicted solution to obtain $\widehat{F^p}$ and $\widehat{G^p}$. The final update then uses the averaged reaction contribution

   $$\frac{1}{2}\left(\widehat{F^n} + \widehat{F^p}\right),$$

   and similarly for G.

The coefficient arrays

$$P_u, \quad Q_u, \quad P_v, \quad Q_v$$

only depend on the Fourier wavenumbers and timestep size, so they can be precomputed once at initialization and reused throughout the simulation.

The final update can be written as

$$\hat{u}^{n+1} = P_u \odot \hat{u}^n + Q_u \odot \gamma\widehat{F^{\text{avg}}},$$

where $\odot$ denotes pointwise multiplication in Fourier space.

6.3 Algorithm and Cost

Each timestep of the spectral solver proceeds as follows:

1. Evaluate the reaction terms in physical space:

   $$F^n = f(U^n, V^n), \qquad G^n = g(U^n, V^n).$$

2. Compute the Fourier transforms of the solution and reaction terms:

   $$\hat{U}^n = \texttt{fft2}(U^n), \qquad \hat{V}^n = \texttt{fft2}(V^n),$$

   $$\widehat{F^n} = \texttt{fft2}(F^n), \qquad \widehat{G^n} = \texttt{fft2}(G^n).$$

3. Update each Fourier mode independently using the IMEX formula:

   $$\hat{U}^{n+1} = P_u \odot \hat{U}^n + Q_u \odot \gamma\widehat{F^n},$$

   with the same update applied to $\hat{V}^{n+1}$.

4. Transform the updated solution back to physical space:

   $$U^{n+1} = \text{Re}\left(\texttt{ifft2}(\hat{U}^{n+1})\right),$$

   $$V^{n+1} = \text{Re}\left(\texttt{ifft2}(\hat{V}^{n+1})\right).$$

The computational cost of each timestep is

$$O(N^2 \log N),$$

coming mainly from the FFT and inverse FFT operations. By comparison, the ADI method scales as $O(N^2)$ per timestep.

Even though the spectral method scales like $O(N^2 \log N)$ instead of $O(N^2)$, it still ran faster in practice on the larger grids. Most of the work is done through FFTs and vectorized matrix operations, so there are no tridiagonal sweeps or directional loops like in the ADI solver.

The Crank–Nicolson diffusion update is also unconditionally stable, so the spectral solver can use the timestep

$$\Delta t = 10^{-5}$$

from [1] without running into diffusion stability issues.

6.4 Results and Comparison

Figure 4 shows the steady-state solutions computed using the spectral IMEX solver with periodic boundary conditions for all three values of γ. The overall pattern structures are qualitatively similar to those shown in Figure 2 of [1].

- For $\gamma = 114$, the solution forms a smooth periodic structure with four high-value regions near the corners and a lower-value region near the center. This corresponds to one of the lowest unstable Fourier modes on the domain.
- For $\gamma = 1000$, the solution transitions to a more regular spot-like pattern arranged diagonally across the periodic domain, with
$$u \in [0.58,\ 1.59].$$
- For $\gamma = 5000$, the solution develops much finer spatial structure, producing a dense hexagonal-style arrangement of spots with approximately 25 spots visible in the computational domain and
$$u \in [0.59,\ 1.56].$$

γ	ADI Δt	ADI wall time	Spectral Δt	Spectral wall time
114	2.19×10^{-3}	8 s	1.00×10^{-3}	5 s
1000	2.50×10^{-4}	69 s	4.00×10^{-4}	13 s
5000	5.00×10^{-5}	437 s	8.00×10^{-5}	80 s

Table 4: Comparison of timestep sizes and wall-clock runtimes for the ADI and spectral solvers on a 150×150 grid. Both methods use a predictor–corrector treatment for the reaction terms.

Performance comparison. Table 4 compares the runtime of the ADI and spectral solvers. Although the spectral method has asymptotic complexity

$$O(N^2 \log N),$$

compared to

$$O(N^2)$$

for ADI, the spectral implementation was noticeably faster in practice, especially for the larger γ cases.

A big part of the speed difference is that the spectral solver is almost fully vectorized. Each timestep is basically a few FFT calls and pointwise matrix operations, while the ADI solver has to perform tridiagonal sweeps in both directions every step.

The spectral method also stayed stable at somewhat larger timesteps than the ADI solver for the same parameter values. At $\gamma = 1000$, for example, the spectral code was stable with a timestep about $1.6\times$ larger, which reduced the total number of timesteps quite a bit.

I also suspect a lot of the performance comes from MATLAB's FFT implementation being extremely optimized. Even though the spectral method scales as $O(N^2 \log N)$ instead of $O(N^2)$, it still ended up running much faster on the 150×150 grid used here.

Tradeoffs. The spectral solver ended up working really well on the periodic-domain cases, and it was noticeably faster than the ADI code in practice. The main limitation is that it depends on periodic boundary conditions. For the zero-flux problems, the ADI method was much easier to work with since the Neumann BCs can be built directly into the finite difference stencils using ghost points.

There are ways to extend spectral methods to Neumann boundary conditions using cosine transforms or Chebyshev methods, but that gets more complicated than the straightforward FFT-based setup used here.

7 Conclusion

The fixed-domain results from Sections 4.1 and 4.2 of Madzvamuse & Maini [1] were reproduced using two separate numerical solvers: a Peaceman–Rachford ADI finite difference method for the zero-flux boundary condition case, and a Fourier spectral IMEX method for the periodic case.

Both solvers produced the expected qualitative Turing patterns for all tested values of γ. At $\gamma = 114$, the solutions formed large-scale stripe or corner-type structures. At $\gamma = 1000$, the solutions transitioned to spot patterns, and at $\gamma = 5000$, the patterns developed much finer spatial structure. The ADI solver was also verified against an exact solution of the heat equation, with the measured convergence rates approaching second order as the grid was refined.

One of the main findings of this project involved the treatment of the reaction terms. The original paper describes evaluating the reaction terms explicitly within the ADI update, but in our implementation this became unstable for $\gamma \geq 1000$. The reaction Jacobian has eigenvalues that are close to purely imaginary and whose magnitude scales with γ, which makes forward Euler treatment prone to oscillatory instability at large timesteps.

To address this issue, a predictor–corrector approach was used for the reaction terms. This gives an effective Crank–Nicolson style treatment of the reaction contribution and significantly improves stability. In practice, this eliminated the temporal oscillations seen with forward Euler while still allowing the Turing instability to grow and form patterns. The same issue and same solution appeared in both the ADI and spectral solvers.

I also spent some time testing operator splitting approaches, especially Strang splitting, but they never worked very well for this problem. In a lot of the runs, the solution would either drift back

toward the homogeneous steady state or take much longer to develop clear patterns. My guess is that separating the reaction and diffusion steps too much weakens the Turing instability itself, since the instability depends on both effects interacting at the same time. For the parameter values used here, keeping the reaction terms coupled directly into the diffusion update gave much more reliable pattern formation.

The mesh sensitivity study also reproduced one of the main conclusions of [1]: the square computational mesh influences the final pattern selection. At $\gamma = 1000$, coarse grids produced irregular spot arrangements, while finer grids converged toward a highly symmetric 2×3 spot configuration aligned with the coordinate directions of the mesh. This suggests that the computational grid itself acts as a symmetry selection mechanism for the ADI solutions.

The ADI and spectral solvers each have different advantages. The ADI method naturally handles zero-flux boundary conditions and is more flexible for bounded biological domains, although it requires repeated tridiagonal solves at every timestep. The spectral method is restricted to periodic domains, but it is much faster in practice because the FFT operations are highly optimized and fully vectorized.

Both solvers were able to reproduce the main pattern behavior reported in the paper, but the project also showed how sensitive these simulations can be to details like timestep treatment, boundary conditions, and even the structure of the computational mesh.

References

[1] Anotida Madzvamuse and Philip K. Maini. Velocity-induced numerical solutions of reaction-diffusion systems on continuously growing domains. *Journal of Computational Physics*, 225(1): 100–119, 2007. doi: 10.1016/j.jcp.2006.11.022.

[2] Donald W. Peaceman and Henry H. Rachford. The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for Industrial and Applied Mathematics*, 3(1):28–41, 1955.

[3] Steven J. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 34(2):148–176, 1995.

[4] J. Schnakenberg. Simple chemical reaction systems with limit cycle behaviour. *Journal of Theoretical Biology*, 81:389–400, 1979.

[5] Alan M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B*, 237:37–72, 1952.

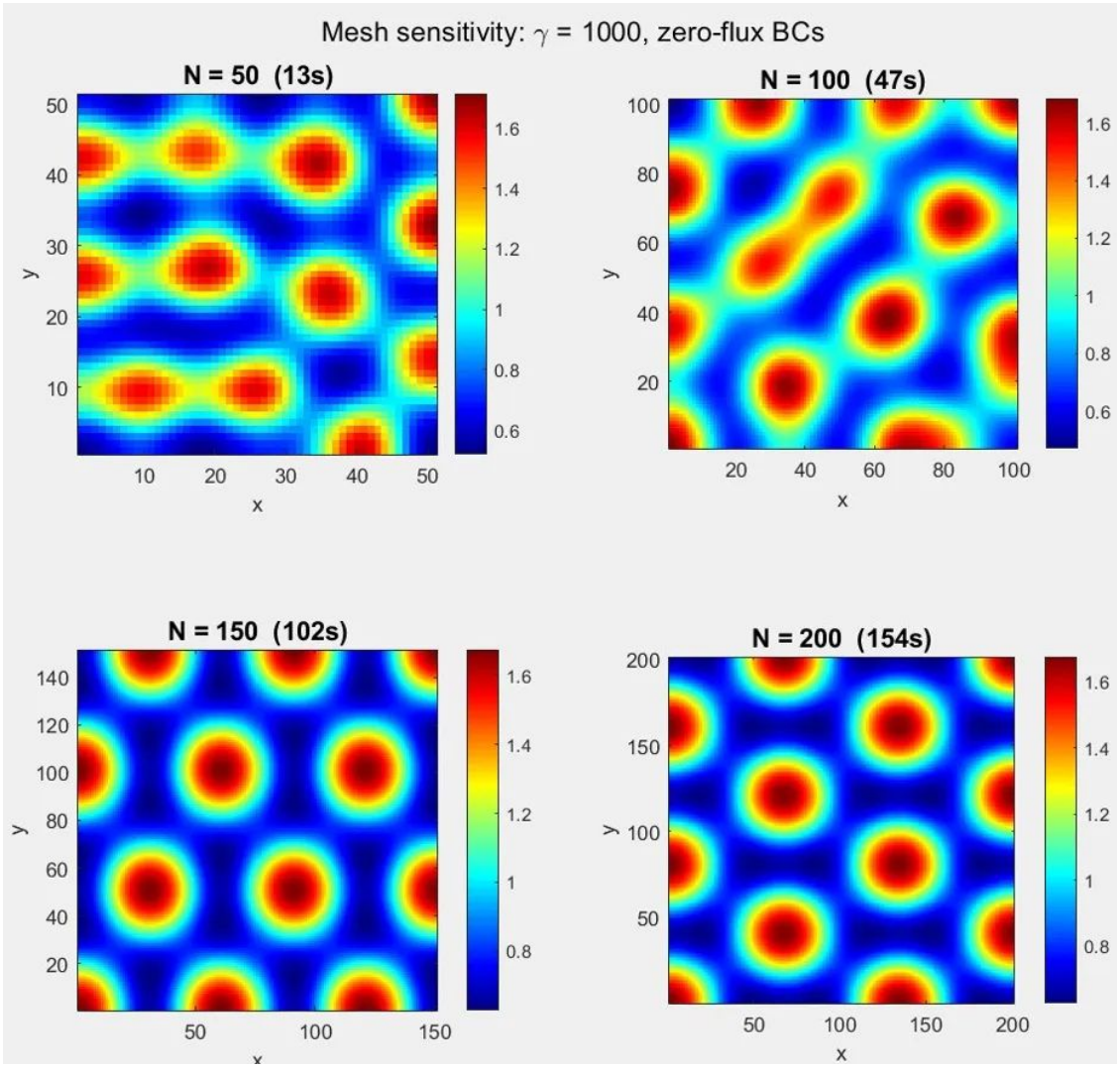

Figure 3: Steady-state u concentration for $\gamma = 1000$ on uniform grids of different resolutions. Coarser grids produce less regular spot arrangements, while finer grids converge toward a symmetric grid-aligned pattern.

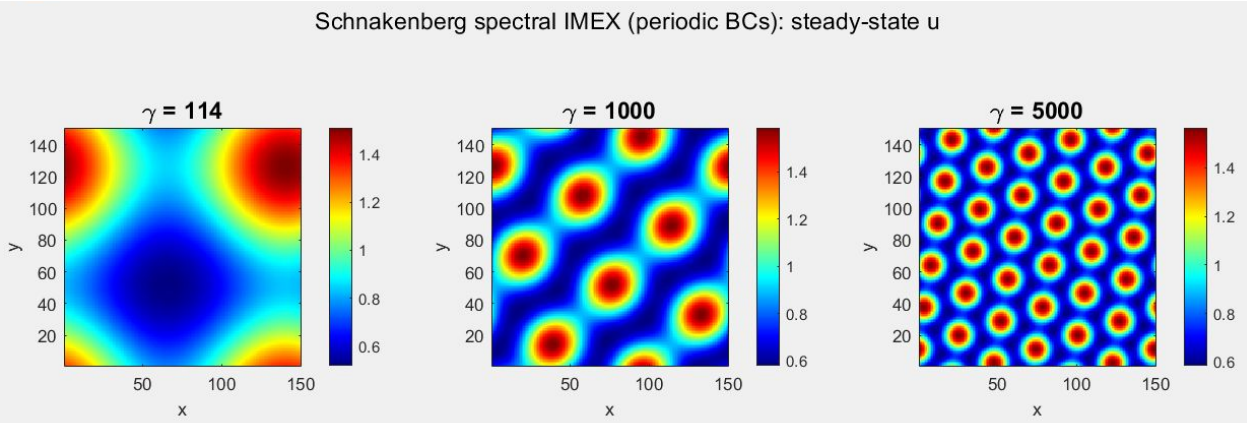

Figure 4: Steady-state u concentration computed with the spectral IMEX solver under periodic boundary conditions. Parameters $a = 0.126779$, $b = 0.792366$, and $d = 10$ follow [3]. Compare with Figure 2 (left column) of [1].