Simulating the 2D Brusselator system in reproducing kernel Hilbert space

Maryam Mohammadi[a], Reza Mokhtari[a], Robert Schaback[b,*]

[a]*Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran*
[b]*Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestraße 16-18, D-37073 Göttingen, Germany*

Abstract

This paper simulates the two-dimensional Brusselator reaction-diffusion system by the Method of Lines. It is implemented as a meshless method based on spatial trial spaces chosen as reproducing kernel Hilbert spaces. For efficiency and stability reasons, we use the Newton basis introduced by Müller and Schaback [21]. The method is shown to work in all interesting situations described by Hopf bifurcations and Turing patterns.

Keywords: Reproducing kernel Hilbert space, Newton basis functions, Method of lines, 2D Brusselator system

1. Introduction

Reaction-diffusion equations frequently arise in the study of chemical and biological systems. The so-called Brussels school [7, 13, 15, 16, 22, 24] developed and analysed the behaviour of a non-linear oscillator [15, 24] associated with the chemical system

$$\begin{aligned}&\delta \to U,\\ &\rho + U \to V + D,\\ &2U + V \to 3U,\\ &U \to E\end{aligned}$$

where δ and ρ are input chemicals, D and E are output chemicals and U and V are intermediates. Let $u(x,t)$ and $v(x,t)$ be the concentrations of U and V, and assume that the concentrations of the input compounds δ and ρ are held constant during the reaction process. Then one obtains the following system of reaction-diffusion equations, known as the Brusselator system:

$$\begin{aligned}u_t(x,t) &= \delta + u^2 v - (\rho+1)u + \mu_1 \Delta u,\\ v_t(x,t) &= \rho u - u^2 v + \mu_2 \Delta v,\end{aligned}$$

where δ, ρ, and diffusion coefficients μ_1 and μ_2 are positive constants. The parameter ρ is often chosen as a parameter for studying bifurcation. The Brusselator system occurs in a large number of physical problems such as the formation of ozone from atomic oxygen, in enzymatic reactions, and arises in laser and plasma physics from multiple coupling between modes. No analytical solution of the system is known so far, and therefore numerical solutions have to be used. Moreover, there is very little literature on the numerical solution of the system. Known techniques are the Adomian decomposition method [1], second order finite difference method [27], modified Adomian decomposition method [29], dual-reciprocity boundary element method [2], differential quadrature method [17], and radial basis functions collocation method [28].

*Corresponding author
Email addresses: m_mohammadi@math.iut.ac.ir (Maryam Mohammadi), mokhtari@cc.iut.ac.ir (Reza Mokhtari), schaback@math.uni-goettingen.de (Robert Schaback)

Unlike traditional numerical methods in solving partial differential equations, *meshless methods* need no mesh generation. Collocation methods are truly meshless and simple enough to allow modelling of rather high dimensional problems [12, 11, 5, 14, 6, 10, 19].

The theory of reproducing kernels [3, 26] was used for the first time at the beginning of the 20th century by S. Zaremba in his work on boundary value problems for harmonic and biharmonic functions. There are plenty of papers in which kernels are successfully used for solving partial differential equations (see [18, 8, 20] and the references therein).

It is well known that representations of kernel-based approximants in terms of the standard basis of translated kernels are notoriously unstable. The Newton basis [21] with a recursively computable set of basis functions and vanishing at increasingly many data points turns out to be more stable. It is orthonormal in the native Hilbert space and complete, if infinitely many data locations are reasonably chosen. Recently, an adaptive calculation of Newton basis arising from a pivoted Cholesky factorization which is computationally cheap, has been introduced [23].

For time-dependent partial differential equations, meshless kernel-based methods are based on a fixed spatial interpolation, but since the coefficients are time-dependent, one obtains a system of ordinary differential equations. This is the well-known *method of lines*, and it turned out to be approximately useful in various cases [25, 9, 4].

In this study, a method of lines, implemented as a meshless method based on spatial trial spaces spanned by the Newton basis functions in the "native" Hilbert space of the reproducing kernel is developed for the numerical simulation of the two-dimensional Brusselator reaction-diffusion system.

The rest of the paper is organized as follows. In section 2 we describe the governing equations and the behaviour of the Brusselator system. Kernel-based trial functions, and particularly the Newton basis functions, are summarized in section 3. In section 4, we turn to Newton basis functions satisfying the Brusselator system and provide a Method of Lines that avoids time integration at all. The implementation of the method is given in section 5. Some numerical examples are presented in section 6. The last section is devoted to a brief conclusion.

2. Governing equations

We consider the 2D Brusselator system with the initial and Dirichlet or Neumann boundary conditions:

$$\begin{cases} u_t(x,t) = \delta + u^2 v - (\rho+1)u + \mu_1 \Delta u \\ \\ v_t(x,t) = \rho u - u^2 v + \mu_2 \Delta v \end{cases} \qquad x \in \Omega \subset \mathbb{R}^2, \quad t \in (0,T], \tag{1}$$

$$\begin{cases} (u(x,t), v(x,t)) = \big(f^D(x,t), g^D(x,t)\big) & x \in D \subseteq \partial\Omega, \quad t \in [0,T], \\ \\ \big(\frac{\partial u}{\partial n}(x,t), \frac{\partial v}{\partial n}(x,t)\big) = \big(f^{\mathcal{N}}(x,t), g^{\mathcal{N}}(x,t)\big) & x \in \mathcal{N} \subseteq \partial\Omega, \quad t \in [0,T], \end{cases} \tag{2}$$

$$\begin{cases} u(x,0) = u_0(x), \\ \\ v(x,0) = v_0(x), \end{cases} \qquad x \in \overline{\Omega}. \tag{3}$$

where u_0, v_0, f^D, g^D, $f^{\mathcal{N}}$ and $g^{\mathcal{N}}$ are known functions, $\Omega \subset \mathbb{R}^2$ is the domain set, $\partial\Omega$ is the boundary of the domain set Ω, and Δ is the Laplace operator. It can be shown that the only critical point of the Brusselator system is $\big(\delta, \frac{\rho}{\delta}\big)$. The Jacobian at the critical point is given by

$$J = \begin{bmatrix} \rho - 1 & \delta^2 \\ -\rho & -\delta^2 \end{bmatrix}$$

and its eigenvalues satisfy the characteristic equation

$$\lambda^2 + (1 - \rho + \delta^2)\lambda + \delta^2 = 0.$$

So the eigenvalues of J clearly depend on $1 - \rho + \delta^2$ and the quantity $\Delta \equiv (1 - \rho + \delta^2)^2 - 4\delta^2$. Therefore the homogeneous steady state $\big(\delta, \frac{\rho}{\delta}\big)$ undergoes a Hopf instability if $\rho > \rho^H = 1 + \delta^2$, evolving then into

a homogeneous limit cycle characterized by a critical frequency $\omega = \delta$. The stability properties and the existence of a limit cycle are summarized in Table 1 in relation to the four regions of Figure 1. A Hopf bifurcation occurs as the curve $1 - \rho + \delta^2$ is crossed, where a stable equilibrium point becomes unstable and a stable limit cycle exists for ρ and δ in regions 1 and 2. The homogeneous steady state of system may also go through a Turing instability induced by the presence of diffusion, when

$$\rho > \rho^T = \left(1 + \delta\sqrt{\frac{\mu_1}{\mu_2}}\right)^2.$$

A stationary spatial pattern then emerges, characterized by an intrinsic critical wave vector $k^2 = \delta/\sqrt{\frac{\mu_1}{\mu_2}}$.

3. Kernel-based trial functions

We take a smooth symmetric positive definite kernel $K : \Omega \times \Omega \to \mathbb{R}$ on the spatial domain Ω. Behind each such kernel there is a reproducing "native" Hilbert space

$$\mathcal{N}_K = \overline{\text{span}\{K(x,\cdot)\ |x \in \Omega\}},$$

of functions on Ω in the sense

$$\langle f, K(x,\cdot)\rangle_{\mathcal{K}} = f(x) \quad \text{for all } x \in \Omega, f \in \mathcal{K},$$

and whose inner product is linked to the kernel itself via

$$\langle K(x,\cdot), K(y,\cdot)\rangle_{\mathcal{K}} = K(x,y) \quad \text{for all } x \in \Omega.$$

The most important examples are the Whittle-Matern kernels

$$r^{m-d/2}K_{m-d/2}(r),\ r = \|x-y\|,\ x, y \in \mathbb{R}^d,$$

reproducing in the Sobolev space $W_2^m(\mathbb{R}^d)$ for $m > d/2$, where K_ν is the Bessel function of third kind. The following will be independent of the kernel chosen, but users should be aware that the kernel should be smooth enough to allow sufficiently many derivatives for the PDE and additional smoothness for fast convergence [30]. For scattered nodes $x_1, \ldots, x_n \in \overline{\Omega}$, the *translates* $K_j(x) = K(x_j, x)$ are the trial functions we want to start with. Since the kernel K is smooth and explicitly available, we can take derivatives with respect to both arguments cheaply, and this implies that we get cheap derivatives of the K_j. But the standard basis of translates leads to an ill-conditioned kernel matrix

$$A = (K(x_j, x_k))_{1 \le j,k \le n},$$

and hence the translates are notoriously unstable. The Newton basis with a recursively computable set of basis functions and vanishing at increasingly many data points turns out to be more stable. It is orthonormal in the native Hilbert space and complete, if infinitely many data locations are reasonably chosen. The Newton basis functions $\{N_k(x)\}_{k=1}^n$ can be expressed by

$$N_k(x) = \sum_{j=1}^{n} K(x, x_j)c_{jk}, \quad 1 \le k \le n. \tag{4}$$

If

$$N(x) = (N_1(x), \ldots, N_n(x)),$$

and

$$T(x) = (K(x, x_1), \ldots, K(x, x_n)),$$

from (4) we have

$$N(x) = T(x) \cdot C,$$

where $C = (c_{jk})_{1\le j,k\le n}$ is the coefficient matrix. Hence the value matrix $V = (N_j(x_i))_{1\le i,j\le n}$ is of the form $V = A \cdot C$. It has been proved that the Cholesky decomposition $A = L \cdot L^T$ with a nonsingular lower triangular matrix L leads to the Newton basis N with $N(x) = T(x) \cdot \left(L^T\right)^{-1}$, and $V = L$. It can be recursively calculated and has the property $N_j(x_k) = 0,\ 1 \le k \le j \le n$. If the values of the Newton basis and linear maps $\mathcal{L}$ like derivatives are needed to be calculated at other points, we get the linear systems

$$V \cdot N^T(x) = T(x)^T,$$

and

$$V \cdot \mathcal{L}(N^T(\cdot)) = \mathcal{L}(T(\cdot)^T),$$

respectively.

4. Method of Lines

We aim at the Method of Lines (MOL), which leads to a system of ordinary differential equations, and this implies that there will be neither time discretization at all nor artificial linearization of the differential equation. The problem of correct time-stepping will be automatically solved by the ODE solver we invoke. The discretization is at points $x_i, 1 \le i \le n$ for the PDE, $y_j,\ 1 \le j \le m$ for the Dirichlet and $z_k,\ 1 \le k \le l$ for the Neumann boundary conditions. We reorder these sequentially into points $w_i,\ 1 \le i \le m+n+l$, the y_j first and the x_i second, and form the Newton basis $N_1, \ldots, N_{m+n+l}$ for these points. Then $N_{m+1}, \ldots, N_{m+n+l}$ vanish on the Dirichlet points, and $N_{m+n+1}, \ldots, N_{m+n+l}$ also vanish on the PDE points. We write our trial space functions as

$$\begin{aligned} \tilde{u}(x,t) &= \sum_{j=1}^{m+n+l} \alpha_j(t) N_i(x) \\ \tilde{v}(x,t) &= \sum_{j=1}^{m+n+l} \beta_j(t) N_i(x) \end{aligned} \tag{5}$$

and care for the Dirichlet boundary conditions by solving

$$\begin{aligned} \tilde{u}(w_i,t) = f^D(w_i,t) &= \sum_{j=1}^{m} \alpha_j(t) N_j(w_i),\ 1 \le i \le m, \\ \tilde{v}(w_i,t) = g^D(w_i,t) &= \sum_{j=1}^{m} \beta_j(t) N_j(w_i),\ 1 \le i \le m, \end{aligned}$$

for the unknown vectors $a_1(t) = (\alpha_1(t), \ldots, \alpha_m(t))^T$, and $b_1(t) = (\beta_1(t), \ldots, \beta_m(t))^T$. This is just the Newton interpolant to the data f_i^D and g_i^D on the Dirichlet points. We will also need

$$\begin{aligned} f^{D\prime}(w_i,t) &= \sum_{j=1}^{m} \alpha_j'(t) N_j(w_i),\ 1 \le i \le m, \\ g^{D\prime}(w_i,t) &= \sum_{j=1}^{m} \beta_j'(t) N_j(w_i),\ 1 \le i \le m, \end{aligned}$$

for the formulation of the MOL, where the prime denotes the derivative with respect to t. Our unknowns in the trial space are only the vectors

$$\begin{aligned} a_2(t) &= (\alpha_{m+1}(t), \ldots, \alpha_{m+n}(t))^T, \\ b_2(t) &= (\beta_{m+1}(t), \ldots, \beta_{m+n}(t))^T, \\ a_3(t) &= (\alpha_{m+n+1}(t), \ldots, \alpha_{m+n+l}(t))^T, \\ b_3(t) &= (\beta_{m+n+1}(t), \ldots, \beta_{m+n+l}(t))^T. \end{aligned}$$

Now we implement the Neumann boundary conditions at a point w_i, $m+n+1 \leq i \leq m+n+l$ as follows:

$$f^{\mathcal{N}}(w_i,t) = \sum_{j=1}^{m} \alpha_j(t) \frac{\partial N_j}{\partial n}(w_i,t) + \sum_{j=m+1}^{m+n} \alpha_j(t) \frac{\partial N_j}{\partial n}(w_i,t) + \sum_{j=m+n+1}^{m+n+l} \alpha_j(t) \frac{\partial N_j}{\partial n}(w_i,t),$$
$$g^{\mathcal{N}}(w_i,t) = \sum_{j=1}^{m} \beta_j(t) \frac{\partial N_j}{\partial n}(w_i,t) + \sum_{j=m+1}^{m+n} \beta_j(t) \frac{\partial N_j}{\partial n}(w_i,t) + \sum_{j=m+n+1}^{m+n+l} \beta_j(t) \frac{\partial N_j}{\partial n}(w_i,t).$$

Thus the unknown vectors $a_3(t)$ and $b_3(t)$ can be written in terms of the unknown vectors $a_2(t)$ and $b_2(t)$ by solving the following equations:

$$\sum_{j=m+n+1}^{m+n+l} \alpha_j(t) \frac{\partial N_j}{\partial n}(w_i,t) = f^{\mathcal{N}}(w_i,t) - \sum_{j=1}^{m} \alpha_j(t) \frac{\partial N_j}{\partial n}(w_i,t) - \sum_{j=m+1}^{m+n} \alpha_j(t) \frac{\partial N_j}{\partial n}(w_i,t),$$
$$\sum_{j=m+n+1}^{m+n+l} \beta_j(t) \frac{\partial N_j}{\partial n}(w_i,t) = g^{\mathcal{N}}(w_i,t) - \sum_{j=1}^{m} \beta_j(t) \frac{\partial N_j}{\partial n}(w_i,t) - \sum_{j=m+1}^{m+n} \beta_j(t) \frac{\partial N_j}{\partial n}(w_i,t),$$

for $m+n+1 \leq i \leq m+n+l$. We now write the PDE at a point w_i, $m+1 \leq i \leq m+n$ as follows:

$$\begin{aligned}
&\sum_{j=1}^{m} \alpha_j'(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j'(t) N_j(w_i) \\
= \quad & \delta + \left(\sum_{j=1}^{m} \alpha_j(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t) N_j(w_i) \right)^2 \left(\sum_{j=1}^{m} \beta_j(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(t) N_j(w_i) \right) \\
& -(\rho+1) \left(\sum_{j=1}^{m} \alpha_j(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t) N_j(w_i) \right) \\
& +\mu_1 \left(\sum_{j=1}^{m} \alpha_j(t) \Delta N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t) \Delta N_j(w_i) + \sum_{j=m+n+1}^{m+n+l} \alpha_j(t) \Delta N_j(w_i) \right),
\end{aligned} \tag{6}$$

$$\begin{aligned}
&\sum_{j=1}^{m} \beta_j'(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j'(t) N_j(w_i) \\
= \quad & \rho \left(\sum_{j=1}^{m} \alpha_j(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t) N_j(w_i) \right) \\
& - \left(\sum_{j=1}^{m} \alpha_j(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t) N_j(w_i) \right)^2 \left(\sum_{j=1}^{m} \beta_j(t) N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(t) N_j(w_i) \right) \\
& +\mu_2 \left(\sum_{j=1}^{m} \beta_j(t) \Delta N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(t) \Delta N_j(w_i) + \sum_{j=m+n+1}^{m+n+l} \beta_j(t) \Delta N_j(w_i) \right).
\end{aligned}$$

Thus we get an implicit system of first-order ordinary differential equations. The initial conditions also provide

$$\tilde{u}_0(w_i) = \sum_{j=1}^{m} \alpha_j(0) N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(0) N_j(w_i),\ m+1 \leq i \leq m+n,$$
$$\tilde{v}_0(w_i) = \sum_{j=1}^{m} \beta_j(0) N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(0) N_j(w_i),\ m+1 \leq i \leq m+n.$$

5. Implementation

If we introduce suitable column vectors and matrices into the system (6), we have to satisfy

$$\begin{bmatrix} N_3 & \mathbf{0} \\ \mathbf{0} & N_3 \end{bmatrix} \begin{bmatrix} a_2'(t) \\ b_2'(t) \end{bmatrix} = \begin{bmatrix} R_1(a_2, b_2) \\ R_2(a_2, b_2) \end{bmatrix}, \tag{7}$$

with the initial conditions

$$\begin{cases} a_2(0) = (N_3)^{-1} \left((u_0(w_i),\ m+1 \le i \le m+n)^T - N_2 a_1(0) \right), \\ \\ b_2(0) = (N_3)^{-1} \left((v_0(w_i),\ m+1 \le i \le m+n)^T - N_2 b_1(0) \right), \end{cases}$$

where

$$\begin{aligned} R_1(a_2, b_2) = \quad & \delta * \mathbf{1} + ((N_2 * a_1 + N_3 * a_2) \,.\wedge 2) \,.* (N_2 * b_1 + N_3 * b_2) \\ & -(\rho + 1)(N_2 * a_1 + N_3 * a_2) + \mu_1 (D_1 * a_1 + D_2 * a_2 + D_3 * a_3) - N_2 a_1{}'(t), \end{aligned}$$

$$\begin{aligned} R_2(a_2, b_2) = \quad & \rho (N_2 * a_1 + N_3 * a_2) - ((N_2 * a_1 + N_3 * a_2) \,.\wedge 2) \,.* (N_2 * b_1 + N_3 * b_2) \\ & +\mu_2 (D_1 * b_1 + D_2 * b_2 + D_3 * b_3) - N_2 b_1{}'(t), \end{aligned}$$

in MATLAB notation for the pointwise product $.*$ and power $.\wedge$ between two matrices or vectors of the same shape. The necessary matrices and vectors are

$$\begin{aligned} & N_1 = (N_j(w_i))_{1 \le i \le m, 1 \le j \le m}, \ N_2 = (N_j(w_i))_{m+1 \le i \le m+n, 1 \le j \le m}, \\ & N_3 = (N_j(w_i))_{m+1 \le i \le m+n, m+1 \le j \le m+n}, \\ & a_1(t) = (N_1)^{-1} F^D(t), \qquad b_1(t) = (N_1)^{-1} G^D(t), \\ & a_1{}'(t) = (N_1)^{-1} F^{D'}(t), \qquad b_1{}'(t) = (N_1)^{-1} G^{D'} N(t), \\ & F^D(t) = (f^D(w_i, t),\ 1 \le i \le m)^T,\ G^D(t) := (g^D(w_i, t),\ 1 \le i \le m)^T, \\ & F^{D'}(t) = (f^{D'}(w_i, t),\ 1 \le i \le m)^T,\ G^{D'}(t) := (g^{D'}(w_i, t),\ 1 \le i \le m)^T, \\ & a_3(t) = \left(\frac{\partial N_3}{\partial n} \right)^{-1} \left(F^{\mathcal{N}}(t) - \frac{\partial N_1}{\partial n} a_1(t) - \frac{\partial N_2}{\partial n} a_2(t) \right) \\ & b_3(t) = \left(\frac{\partial N_3}{\partial n} \right)^{-1} \left(G^{\mathcal{N}}(t) - \frac{\partial N_1}{\partial n} b_1(t) - \frac{\partial N_2}{\partial n} b_2(t) \right) \\ & F^{\mathcal{N}}(t) = (f^{\mathcal{N}}(w_i, t),\ m+n+1 \le i \le m+n+l)^T, \\ & G^{\mathcal{N}}(t) := (g^{\mathcal{N}}(w_i, t),\ m+n+1 \le i \le m+n+l)^T, \\ & \frac{\partial N_1}{\partial n} = \left(\frac{\partial N_j(w_i)}{\partial n} \right)_{m+n+1 \le i \le m+n+l, 1 \le j \le m}, \\ & \frac{\partial N_2}{\partial n} = \left(\frac{\partial N_j(w_i)}{\partial n} \right)_{m+n+1 \le i \le m+n+l, m+1 \le j \le m+n}, \\ & \frac{\partial N_3}{\partial n} = \left(\frac{\partial N_j(w_i)}{\partial n} \right)_{m+n+1 \le i \le m+n+l, m+n+1 \le j \le m+n+l}, \\ & D_1 = (\Delta N_j(w_i))_{m+1 \le i \le m+n, 1 \le j \le m}, \\ & D_2 = (\Delta N_j(w_i))_{m+1 \le i \le m+n, m+1 \le j \le m+n}, \\ & D_3 = (\Delta N_j(w_i))_{m+1 \le i \le m+n, m+n+1 \le j \le m+n+l}, \end{aligned}$$

where j is the column index and i is the row index. The system (7) is the ODE system generated by the MOL and one can invoke any ODE integrator to solve it. The matrix of the left-hand side is time-independent, and in the case of the invertibility of it, the approximate solutions $u(x,t)$ and $v(x,t)$ will satisfy the differential equations at all points $w_1, ..., w_{m+n+l}$ and all times, the latter within the accuracy limit of the ODE integrator. Note that the nonlinearity of the PDE is preserved, and a good ODE solver will automatically use a reasonable time-stepping and detect stiffness of the ODE system.

6. Numerical results

In this section we present the results of our scheme for the numerical solution of the Brusselator reaction-diffusion system (1)-(3). In all test problems, we take the Matern kernel with RBF parameter $\nu = 2 = m - d/2$ and RBF scale $c = 10$, i.e. we work with the kernel

$$K(x,y) = \left(\frac{\|x-y\|_2^2}{10^2}\right) K_2\left(\frac{\|x-y\|_2^2}{10^2}\right).$$

We also assume that $\overline{\Omega} = [0,1] \times [0,1]$, such that we work in the Hilbert space $W_2^3(\mathbb{R}^2)$. We take 121 discretization points in the region, and 51 grid points along each axis for plotting of figures.

6.1. Test problem 1

Consider the Brusselator system together with the Dirichlet boundary conditions with $\rho = 1$, $\delta = 0$, and $\mu_1 = \mu_2 = 0.25$. The initial and boundary conditions are extracted from the exact solutions

$$\begin{cases} u(x,y,t) = \exp(-x - y - 0.5t), \\ v(x,y,t) = \exp(x + y + 0.5t). \end{cases}$$

Absolute and relative error distributions at time $T = 2$ are shown in Figures 2 and 3, respectively.

6.2. Test problem 2

In the second experiment, we choose parameters $\rho = 2$, $\delta = 1$, and $\mu_1 = \mu_2 = 0.25$, and start from zero initial conditions and fixed boundary conditions taken as the homogeneous steady state $(u, v) = (\delta, \frac{\rho}{\delta})$. Since the exact solutions are not known, we plot the error between the left and right hand sides of the 2 equations of system (1) for the grid points of the region in Figure 4. The plots of the values of u and v at the collocation point $(0.3, 0.3)$ versus time shown in Figure 5, indicate that the solutions converge toward the stationary ones $(\delta, \frac{\rho}{\delta})$.

In the next two test problems, we investigate the behaviour of the system when the sign of $1 - \rho + \delta^2$ changes and the Hopf bifurcation occurs.

6.3. Test problem 3

Consider the Brusselator system with the following initial and Neumann boundary conditions:

$$\begin{cases} u(x,y,0) = 2 + 0.25y, \\ v(x,y,0) = 1 + 0.8x, \end{cases}$$
$$\begin{cases} \frac{\partial u(x,y,t)}{\partial x}|_{x=0}, \frac{\partial u(x,y,t)}{\partial x}|_{x=1} = \frac{\partial u(x,y,t)}{\partial y}|_{y=0} = \frac{\partial u(x,y,t)}{\partial y}|_{y=1} = 0, \\ \frac{\partial v(x,y,t)}{\partial x}|_{x=0}, \frac{\partial v(x,y,t)}{\partial x}|_{x=1} = \frac{\partial v(x,y,t)}{\partial y}|_{y=0} = \frac{\partial v(x,y,t)}{\partial y}|_{y=1} = 0. \end{cases}$$

Computations are carried out with the parameters $\rho = 1$, $\delta = 2$, and $\mu_1 = \mu_2 = 0.002$. The algorithm is tested up to time $T = 5$. The concentration profiles of u and v at $T = 0$ and 5 are shown in Figures 6 and 7. From Figure 7, it can be seen that the numerical values of u and v at each collocation point approach to 2 and 0.5, respectively. These results show an agreement that $(u, v) \to (\delta, \frac{\rho}{\delta})$ as t increases, whenever $1 - \rho + \delta^2 > 0$ (region 4 of Figure 1). The plots of the values of u and v at the collocation point $(0.3, 0.3)$ versus time are shown in Figure 8. It can be noted from Figure 8, that $(u(0.3, 0.3), v(0.3, 0.3)) \to (2, 0.5)$ as $t \to \infty$. The results show an agreement with the results of [27] and [28].

6.4. Test problem 4

The algorithm is repeated with $\rho = 3.4$, $\delta = 1$ up to time $T = 40$. The concentrations profiles of u and v at $T = 40$ are shown in Figure 9. The plots of the values of u and v at the collocation point $(0.3, 0.3)$ versus time are shown in Figure 10. It can be noted from Figures 9 and 10 that the solutions are stable but oscillatory and the numerical method is seen not to converge to any fixed concentration (region 2 of Figure 1). The results show an agreement with the results of [27] and [28].

6.5. Test problem 5

Consider the diffusion–free Brusselator system corresponding to $\mu = 0$ with the following Neumann boundary conditions:

$$\begin{cases} \frac{\partial u(x,y,t)}{\partial x}|_{x=0}, \frac{\partial u(x,y,t)}{\partial x}|_{x=1} = \frac{\partial u(x,y,t)}{\partial y}|_{y=0} = \frac{\partial u(x,y,t)}{\partial y}|_{y=1} = 0, \\ \frac{\partial v(x,y,t)}{\partial x}|_{x=0}, \frac{\partial v(x,y,t)}{\partial x}|_{x=1} = \frac{\partial v(x,y,t)}{\partial y}|_{y=0} = \frac{\partial v(x,y,t)}{\partial y}|_{y=1} = 0. \end{cases}$$

Extensive numerical experiments, taking different values of ρ and δ in the four regions of Figure 1 with $0 < u_0, v_0 \leq 8$ were carried out. It was discovered that the scheme converged to the fixed point $(\delta, \frac{\rho}{\delta})$ whenever $1-\rho+\delta^2 > 0$ (regions 3 and 4 of Figure 1). Phase portraits for $\delta = 0.5$ with $\rho = 2.5, 2.0, 1.2$ and 0.2 (regions 1, 2, 3, 4 of Figure 1) are depicted in Figures 11, 12, 13 and 14, respectively. An enlargement of the area around the fixed point $(0.5, 2.4)$ of Figure 13 is depicted in Figure 15 and shows clearly that the solution sequence spirals into the fixed point. The limit cycles are visible in Figures 11 and 12. The results show an agreement with the results of [27] and [28]. Figure 13 was found in three stages with 3 different starting values. A set of 7 starting values was used for Figure 14.

In the next test problem, we show two different Turing patterns occuring in the Brusselator system by our scheme.

6.6. Test problem 6

Consider the Brusselator system with the parameters $\rho = 16.6$, $\delta = 3$, $\mu_1 = 0.06$, and $\mu_2 = 0.006$. Figure 16 shows the Turing pattern obtained at time $T = 15$ starting from the random initial conditions with no flux boundary conditions.

7. Conclusion

In this paper, the Newton basis functions were successfully used as spatial trial functions in the Method of Lines for the numerical solution of the 2D Brusselator reaction–diffusion system. The method is shown to work in all interesting situations described by Hopf bifurcations and Turing patterns.

References

[1] G. Adomian. The diffusion Brusselator equation. *Comput. Math. Appl.*, 29:1–3, 1995.

[2] W.T. Ang. The two-dimensional reaction-diffusion Brusselator system: a dual-reciprocity boundary element solution. *Eng. Anal. Bound. Elem.*, 27:897–903, 2003.

[3] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math.*, 68:337–404, 1950.

[4] Y. Dereli and R. Schaback. The meshless kernel-based method of lines for solving the equal width equation. Preprint Göttingen, 2010.

[5] C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. in Comp. Math.*, 8:381–399, 1998.

[6] C. Franke and R. Schaback. Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comp.*, 93:73–82, 1998.

[7] M. Herschkowitz-Kaufman and N. Nicolis. Localized spatial structures and non-linear chemical waves in dissipative systems. *J. Chem. Phys.*, 56:1890–1895, 1972.

[8] Y.C. Hon and R. Schaback. Solvability of partial differential equations by meshless kernel methods. *Adv. in Comp. Math.*, 28:283–299, 2008.

[9] Y.C. Hon and R. Schaback. The kernel-based method of lines for the heat equation. Preprint Göttingen, 2010.

[10] Y.C. Hon and R. Schaback. Solving the 3D Laplace equation by meshless collocation via harmonic kernels. to appear in Adv. in Comp. Math., 2010.

[11] Y.C. Hon, R. Schaback, and Zhou X. An adaptive greedy algorithm for solving large radial basis function collocation problem. *Numer. Algorithms.*, 32:13–25, 2003.

[12] E.J. Kansa. scattered data approximation scheme with applications to computational fluid-dynamics, I: Solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.*, 19:147–161, 1990.

[13] B. Lavenda, G. Nicolis, and M. Herschkowitz-Kaufman. Chemical instabilities and relaxation oscillations. *J. Theor. Biol.*, 32:283–292, 1971.

[14] C.F. Lee, L. Ling, and R. Schaback. On convergent numerical algorithms for unsymmetric collocation. *Adv. in Comp. Math.*, 30:339–354, 2009.

[15] R. Lefever. Dissipative structures in chemical systems. *J. Chem. Phys.*, 49:4977–4978, 1968.

[16] R. Lefever and G. Nicolis. Chemical instabilities and sustained oscillations. *J. Theor. Biol.*, 30:267–284, 1971.

[17] R.C. Mittal and R. Jiwari. Numerical study of two-dimensional reaction-diffusion Brusselator system by differential quadrature method. *Int. J. Comput. Methods Eng. Sci. Mech.*, 12:14–25, 2011.

[18] M. Mohammadi and R. Mokhtari. Solving the generalized regularized long wave equation on the basis of a reproducing kernel space. *J. Comput. Appl. Math.*, 14:4003–4014, 2011.

[19] R. Mokhtari and M. Mohammadi. Numerical solution of GRLW equation using Sinc-collocation method. *Comput. Phys. Comm.*, 181:1266–1274, 2010.

[20] R. Mokhtari, F. Toutian Isfahani, and M. Mohammadi. Solving a class of nonlinear differential-difference equations in the reproducing kernel space. to appear in Abstr. Appl. Anal., 2012.

[21] St. Müller and R. Schaback. A Newton basis for kernel spaces. *Journal of Approximation Theory*, 161:645–655, 2009. doi:10.1016/j.jat.2008.10.014.

[22] G. Nicolis and I. Prigogine. *Self-organization in non-equilibrium systems.* Wiley Interscience, New York, 1977.

[23] M. Pazouki and R. Schaback. Bases for kernel-based spaces. *J. Comput. Appl. Math.*, 236:575–588, 2011.

[24] I. Prigogine and R. Lefever. Symmetries breaking instabilities in dissipative systems ii. *J. Phys. Chem.*, 48:1695–1700, 1968.

[25] R. Schaback. The meshless kernel-based method of lines for solving nonlinear evolution equations. Preprint Göttingen, 2011.

[26] R. Schaback and H. Wendland. Kernel techniques: From machine learning to meshless methods. *Acta Numerica.*, 15:543–639, 2006.

[27] E.H. Twizell, A.B. Gumel, and Q. Cao. A second-order scheme for the "Brusselator" reaction-diffusion system. *J. Math. Chem.*, 26:297–316, 1999.

[28] Siraj ul Islam., S. Haq, and A. Ali. A computational modeling of the behavior of the two-dimensional reaction-diffusion Brusselator system. *Appl. Math. Model.*, 34:3896–3909, 2010.

[29] A.M. Wazwaz. The decomposition method applied to systems of partial differential equations and to the reaction-diffusion Brusselator model. *Appl. Math. Comput.*, 110:251–264, 2000.

[30] H. Wendland. *Scattered Data Approximation.* Cambridge University Press, 2005.

Table 1: Nature of the critical point and existence of the limit cycle.

Region	$1-\rho+\delta^2$	Δ	Eigenvalues	Type of critical point	Limit cycle exists
1	< 0	≥ 0	Positive real	Unstable node	Yes
2	< 0	< 0	Positive real parts	Unstable focus	Yes
	$= 0$	< 0	Imaginary	Stable fine focus	No
3	> 0	< 0	Negative real parts	Stable focus	No
4	> 0	≥ 0	Negative real	Stable node	No

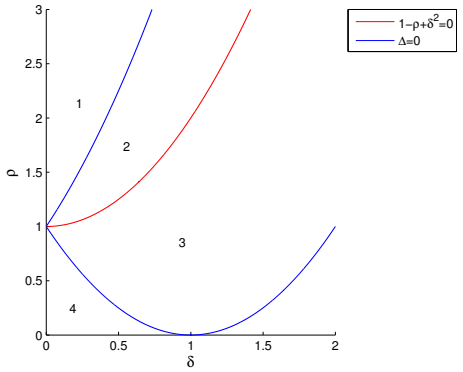


Figure 1: Stability regions of the Brusselator system.

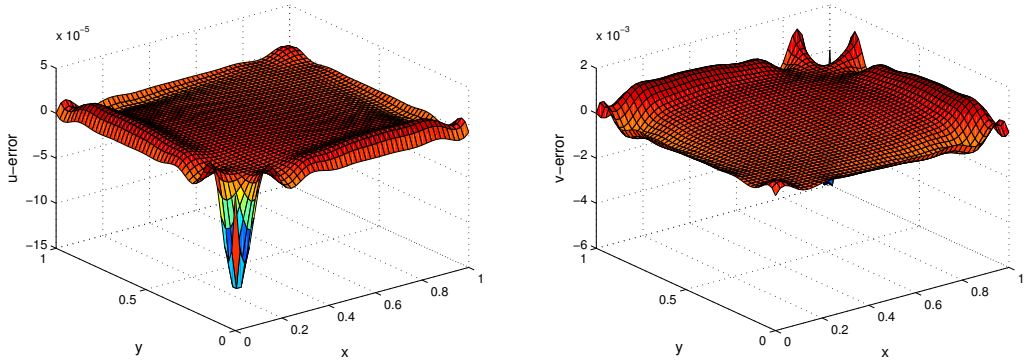


Figure 2: Absolute error graph at time $T = 2$.

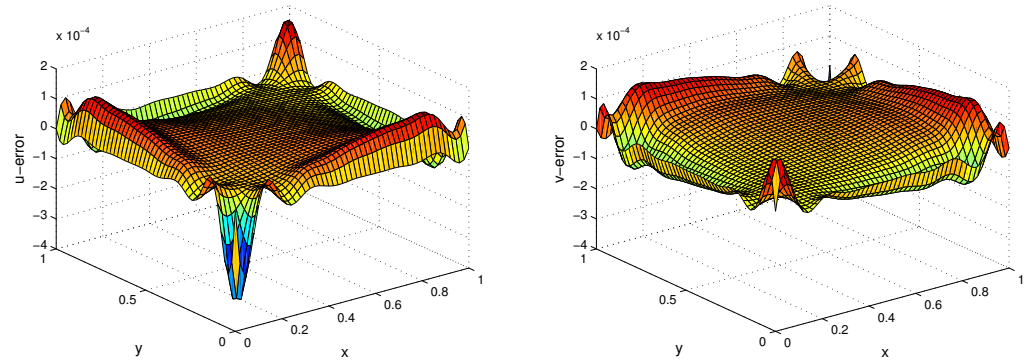
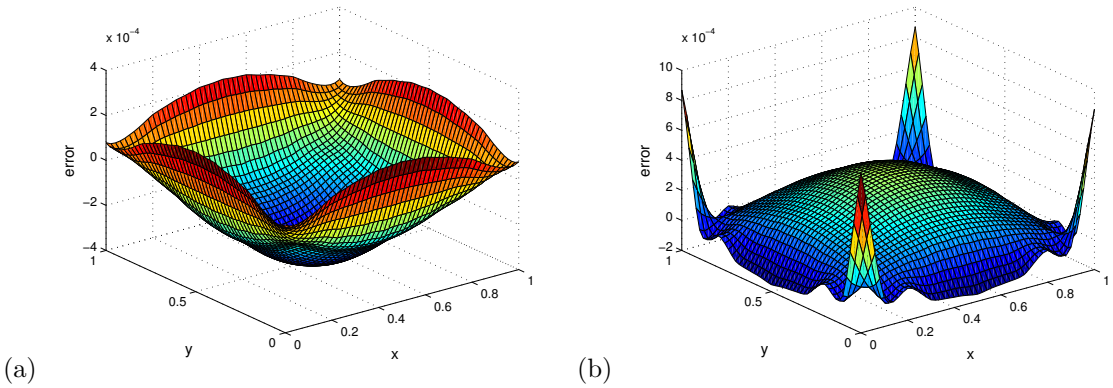
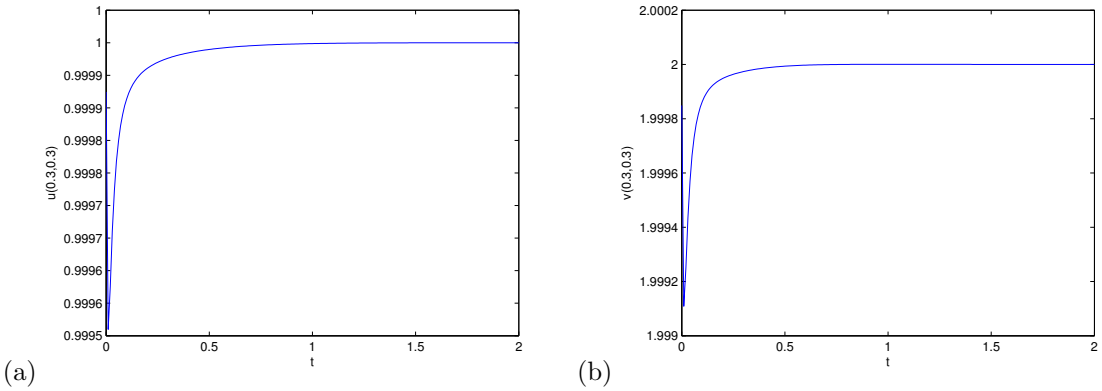


Figure 3: Relative error graph at time $T = 2$.



(a) (b)

Figure 4: Error graph of PDE at time $T = 2$. (a) Equation 1; (b) Equation 2.



(a) (b)

Figure 5: Plots of $u(0.3, 0.3)$ and $v(0.3, 0.3)$ versus time.

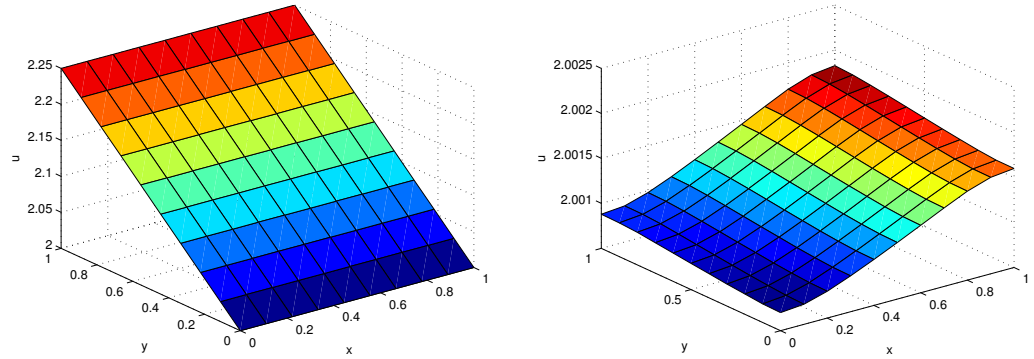


Figure 6: Initial concentration profiles of u and v.

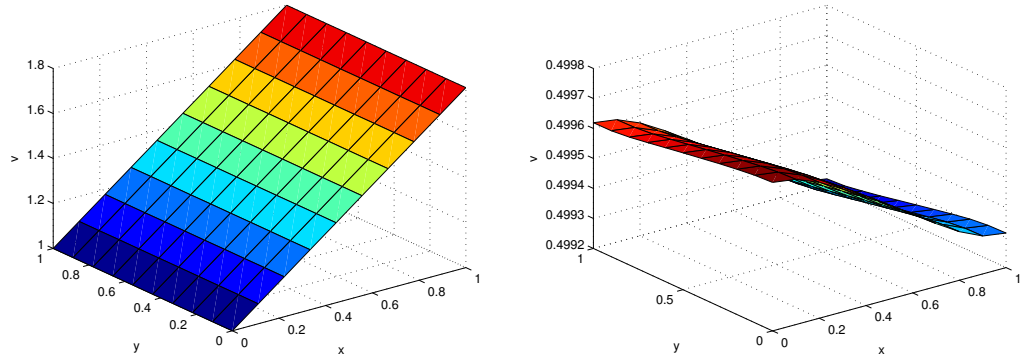


Figure 7: Plots of u and v at $T = 5$, $\rho = 1$, $\delta = 2$.

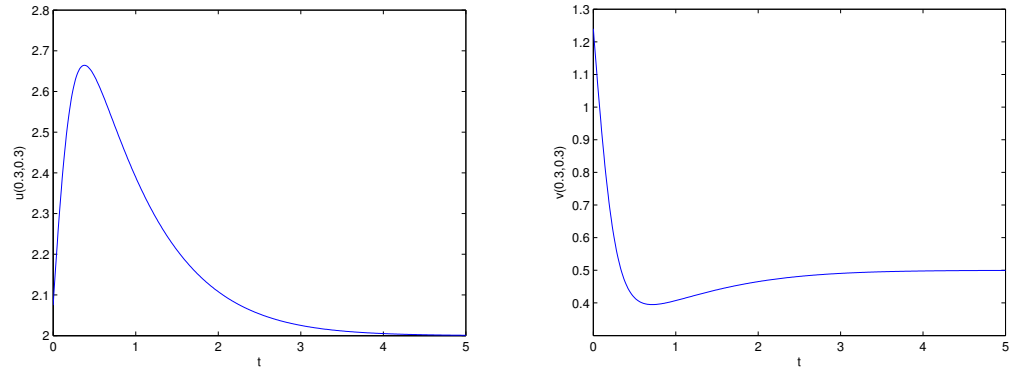


Figure 8: Plots of $u(0.3, 0.3)$ and $v(0.3, 0.3)$ versus time.

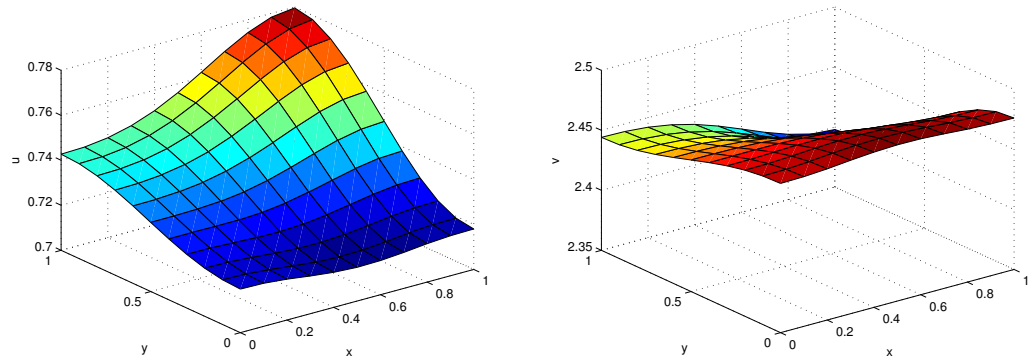


Figure 9: Plots of u and v at $T = 40$, $\rho = 3.4$, $\delta = 1$.

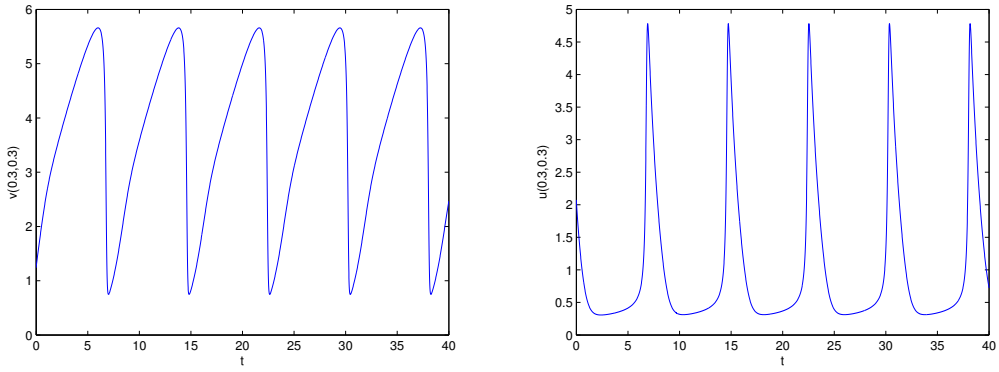


Figure 10: Plots of $u(0.3, 0.3)$ and $v(0.3, 0.3)$ versus time.

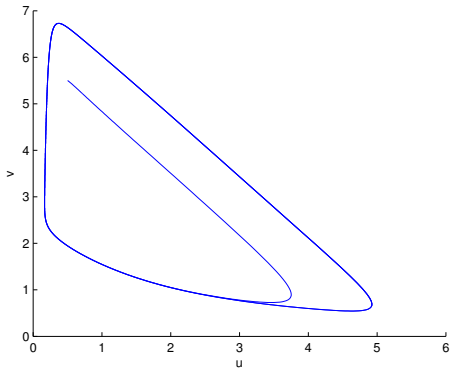


Figure 11: Solution of diffusion–free system with $\rho = 2.5$ and $\delta = 0.5$.

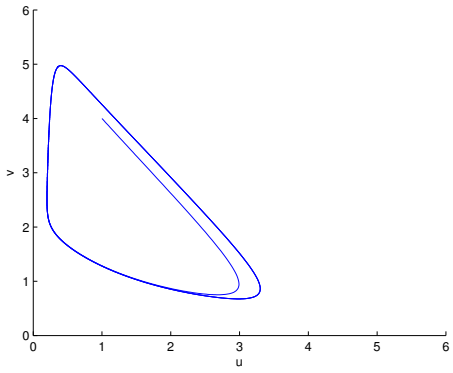


Figure 12: Solution of diffusion–free system with $\rho = 2$ and $\delta = 0.5$.

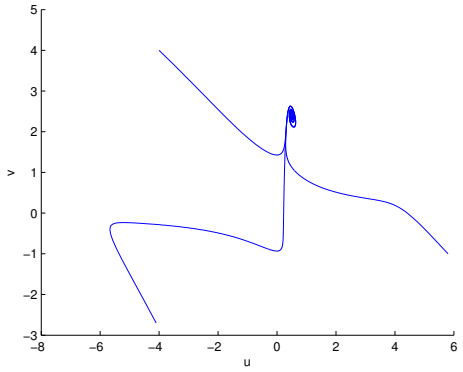


Figure 13: Solution of diffusion–free system with $\rho = 1.2$ and $\delta = 0.5$.

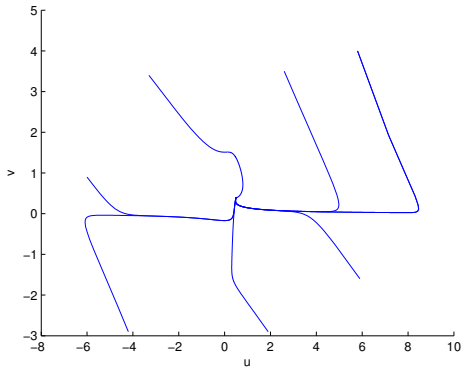


Figure 14: Solution of diffusion–free system with $\rho = 0.2$ and $\delta = 0.5$.

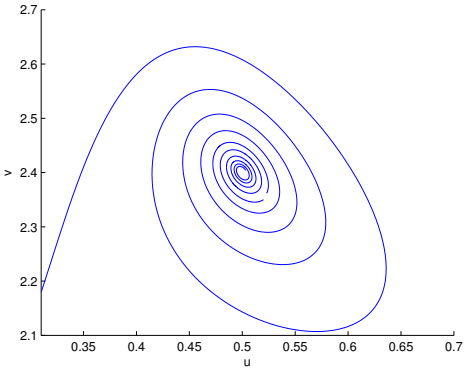


Figure 15: Enlargement of Fig. 13 in the vicinity of the fixed point $u = 0.5$ and $v = 2.4$.

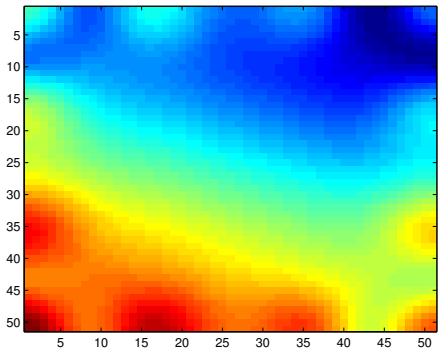


Figure 16: Turing pattern with the parameters $\rho = 16.6$, $\delta = 3$, $\mu_1 = 0.06$, $\mu_2 = 0.006$, and no flux boundary conditions.