 CMES, vol.101, no.2, pp.113-138, 2014

# A Meshless Method for Solving the 2D Brusselator Reaction-Diffusion System

**M. Mohammadi[1], R. Mokhtari[2,3] and R. Schaback[4]**

**Abstract:** In this paper, the two-dimensional (2D) Brusselator reaction-diffusion system is simulated numerically by the method of lines. The proposed method is implemented as a meshless method based on spatial trial functions in the reproducing kernel Hilbert spaces. For efficiency and stability reasons, we use the Newton basis introduced recently by Müller and Schaback. The method is shown to work in all interesting situations described by Hopf bifurcations and Turing patterns.

**Keywords:** 2D Brusselator reaction-diffusion system, method of lines, meshless method, Newton basis functions, reproducing kernel Hilbert space.

## 1 Introduction

Reaction-diffusion equations frequently arise in the study of chemical and biological systems. The so-called Brussels school [Herschkowitz-Kaufman and Nicolis (1972); Lavenda, Nicolis, and Herschkowitz-Kaufman (1971); Lefever (1968); Lefever and Nicolis (1971); Nicolis and Prigogine (1977); Prigogine and Lefever (1968)] developed and analyzed the behaviour of a non-linear oscillator [Lefever (1968); Prigogine and Lefever (1968)] associated with the chemical system

$$\begin{aligned} &\delta \to U, \\ &\rho + U \to V + D, \\ &2U + V \to 3U, \\ &U \to E \end{aligned} \tag{1}$$

[1] Faculty of Mathematical Sciences and Computer, Kharazmi University, 50 Taleghani Ave., Tehran 1561836314, Iran.
[2] Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 84156-83111, Iran.
[3] Corresponding author.
[4] Institut für Numerische und Angewandte Mathematik, Universität Göttingen, Lotzestraße 16-18, D-37073 Göttingen, Germany.

where $\delta$ and $\rho$ 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 $\delta$ and $\rho$ are held constant during the reaction process. Then using the law of mass action, the kinetic equations associated with (1) are given by the following system of reaction-diffusion equations, known as the Brusselator system [Prigogine and Lefever (1968)]:

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

where $u$ and $v$ represent dimensionless concentrations of two reactants, $\delta$, $\rho$, and diffusion coefficients $\mu_1$ and $\mu_2$ are positive constants. The parameter $\rho$ 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 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 [Adomian (1995); Lin, Liu, and Li (2012)], second order finite difference method [Gumel, Langford, Twizel, and Wu (2000); Twizell, Gumel, and Cao (1999)], modified Adomian decomposition method [Wazwaz (2000)], dual-reciprocity boundary element method [Ang (2003)], differential quadrature method [Mittal and Jiwari (2011)], modified cubic B-spline differential quadrature method [Jiwari and Yuan (2014)], Multistage homotopy perturbation [Lee, Park, and Jang (2013)], and radial basis functions collocation method [ul Islam, Haq, and Ali (2010)].

Unlike traditional numerical methods in solving partial differential equations (PDEs), *meshless methods* [Dong, Alotaibi, Mohiuddine, and Atluri (2014); Han and Atluri (2014)] need no mesh generation. Collocation methods are truly meshless and simple to program, and they allow various approaches for solving PDEs. Taking translates of kernels as trial functions, meshless collocation in unsymmetric and symmetric form dates back to [Franke and Schaback (1998b); Franke and Schaback (1998a); Kansa (1986)] and has proven to be highly successful, because the arising linear systems are easy to generate and allow good accuracy at low computational cost. In addition, it was proven recently [Schaback (2013)] that symmetric collocation [Fasshauer (1997); Franke and Schaback (1998b); Franke and Schaback (1998a)] using kernels is optimal along all linear PDE solvers using the same input data. This motivates the use of kernels for solving PDEs. An overview of kernel methods before 2006 is given in [Schaback and Wendland (2006)], while recent variations of the theme are in [Hon and Schaback (2008);

Lee, Ling, and Schaback (2009); Mohammadi and Mokhtari (2011); Mohammadi and Mokhtari (2013); Mohammadi and Mokhtari (2014); Mohammadi, Mokhtari, and Panahipour (2013), Mohammadi, Mokhtari, and Panahipour (2014), Mohammadi, Mokhtari, and Toutian Isfahani (2014); Mokhtari and Mohammadi (2010); Mokhtari and Mohseni (2012); Mokhtari, Toutian Isfahani, and Mohammadi (2012)] 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 [Müller and Schaback (2009)] with a recursively computable set of functions which vanish 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 [Pazouki and Schaback (2011)].

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 accurate in several problems [Dereli and Schaback (2010)].

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 behaviour of the Brusselator system. In Section 3, we give the governing equations. Kernel-based trial functions, and particularly the Newton basis functions, are summarized in Section 4. In Section 5, we turn to Newton basis functions satisfying the Brusselator system and provide a method of lines which leads to an ODE system. The implementation of the method is given in Section 6. Some numerical examples are presented in Section 7. The last section is devoted to a brief conclusion.

## 2 Analysis of the Brusselator system

We first describe the behaviour of the Brusselator system [Ma and Wang (2011); Roussel (2005)]. As one is interested in the stability analysis of a nonlinear reaction-diffusion system, one typically first determines the stationary state of the model in the absence of diffusion. This is done by solving the system (2) with conditions $u_t = v_t = 0$ and $\mu_1 = \mu_2 = 0$. So the only equilibrium point of the ordinary differential equation (ODE) system is $(u^*, v^*) = \left(\delta, \frac{\rho}{\delta}\right)$. The Jacobian at the equilibrium

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 $\Delta \equiv (1 - \rho + \delta^2)^2 - 4\delta^2$. When the real parts of the eigenvalues are negative , the equilibrium point is a stable focus and when they cross zero and become positive the equilibrium point becomes an unstable focus, with orbits spiralling out. The stability properties and the existence of a limit cycle in the diffusion-free Brusselator system are summarized in Tab. 1 in relation to the four regions of Fig. 1.

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

| Region | $1 - \rho + \delta^2$ | $\Delta$ | Eigenvalues | Type of critical point | Limit cycle exists |
|---|---|---|---|---|---|
| 1 | $< 0$ | $\geq 0$ | 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 | $> 0$ | $\geq 0$ | Negative real | Stable node | No |

The appearance or the disappearance of a periodic orbit through a local change in the stability properties of an equilibrium point is known as the Hopf bifurcation. In a differential equation a Hopf bifurcation typically occurs when a complex conjugate pair of eigenvalues of the linearized flow at an equilibrium point becomes purely imaginary. So the equilibrium point $\left(\delta, \frac{\rho}{\delta}\right)$ undergoes a Hopf bifurcation at $\rho = \rho^H = 1 + \delta^2$, with oscillations being observed for $\rho > \rho^H$.

Now we want to study the stability analysis of the Brusselator reaction-diffusion system (2). Note that the steady state of the diffusion-free system is also a steady state of the reaction-diffusion one. If $(u, v) = (u^*, v^*)$, then the spatial derivatives of this constant function are zero, as are the reaction terms, so the time derivatives must be zero too. This solution is called the homogeneous steady state. To study

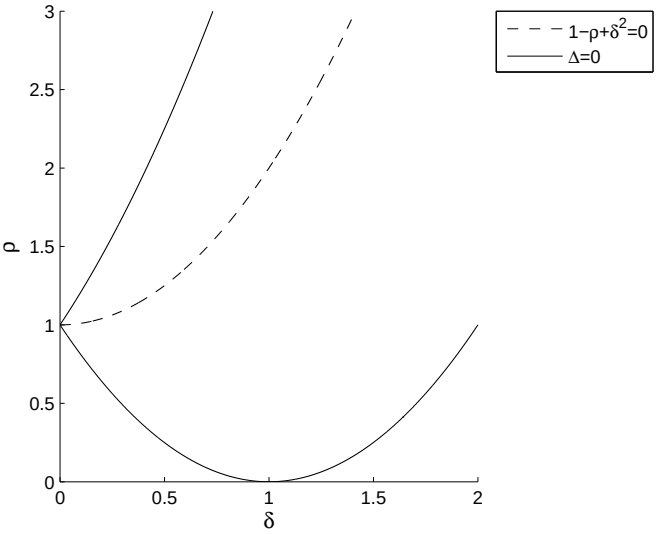



Figure 1: Stability regions of the diffusion-free Brusselator system.

the behaviour of the system, we will linearize (2) around the equilibrium point, i.e., we substitute

$$u = U + \delta,$$

$$v = V + \frac{\rho}{\delta},$$

where $U$ and $V$ are displacements from equilibrium, which now depend both on time and space, and neglect high order terms in $u$ and $v$. Then

$$U_t = (\rho - 1)U + \delta^2 V + \mu_1 \Delta U,$$

$$V_t = -\rho U - \delta^2 V + \mu_2 \Delta V,$$

and so

$$\begin{bmatrix} U \\ V \end{bmatrix}_t = J \begin{bmatrix} U \\ V \end{bmatrix} + D \begin{bmatrix} \Delta U \\ \Delta V \end{bmatrix},$$

where

$$J = \begin{bmatrix} \rho - 1 & \delta^2 \\ -\rho & -\delta^2 \end{bmatrix}, \qquad D = \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix}.$$

We again want to determine if the steady state is stable against small perturbations. But this time we want to introduce a spatial aspect. Suppose that perturbations are inhomogeneous in space. The convenient form is

$$\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} e^{\lambda t} e^{i\mathbf{k}\cdot x},$$

where $x = (x_1, x_2)$, and $\mathbf{k}$ represents a vector of two wavenumbers $(\mathbf{k}_1, \mathbf{k}_2)$. The question now will be whether or not conditions can be found under which the steady state is unstable $(\mathrm{Re}(\lambda) > 0)$ when a wiggly disturbance is introduced. Since any disturbance over a finite domain can be synthesized by adding up sine waves, this will answer the question of whether the steady state is stable against small, but otherwise arbitrary perturbations. If we substitute the perturbation into the linearized equation, after cancelling off the common factors of $e^{\lambda t} e^{i\mathbf{k}\cdot x}$, we get

$$\lambda \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} = J \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} - \|\mathbf{k}\|^2 D \begin{bmatrix} U_0 \\ V_0 \end{bmatrix},$$

or

$$(\lambda I + \|\mathbf{k}\|^2 D - J) \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where $I$ is the identity matrix. This is a homogeneous equation in $[U_0, V_0]$ which only has nontrivial solutions if

$$\begin{aligned} &\left|\lambda \mathbf{I} + 2k^2 \mathbf{D} - \mathbf{J}\right| \\ &= \lambda^2 + \lambda \left(\|\mathbf{k}\|^2(\mu_1 + \mu_2) + 1 - \rho + \delta^2\right) + \|\mathbf{k}\|^4 \mu_1 \mu_2 + \|\mathbf{k}\|^2 \left(\mu_2(1-\rho) + \mu_1 \delta^2\right) + \delta^2 \\ &= 0. \end{aligned}$$

Our task will be to determine whether this characteristic equation has solutions for which the real part of $\lambda$ is positive, and if so, under what conditions. Because of the appearance of the unknown wavenumber $\mathbf{k}$ in this equation, this equation has a different solution for every $\mathbf{k}$. This quadratic equation is of the form

$$\lambda^2 + q\lambda + p = 0.$$

The solutions are

$$\lambda = \frac{1}{2}\left\{-q \pm \sqrt{q^2 - 4p}\right\}.$$

There will be at least one root with a positive real part provided one of the following two conditions are met:

1. $q < 0$, which leads to

   $$\rho > 1 + \delta^2 + \|\mathbf{k}\|^2(\mu_1 + \mu_2) > 1 + \delta^2.$$

   So the steady state may also go through a Hopf instability if $\rho > 1 + \delta^2$ evolving then into a homogeneous limit cycle characterized by a critical frequency $\omega = \delta$.

2. $q > 0$ and $p < 0$, which leads to the following bifurcation conditions:

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

   Note that this bifurcation occurs only when the steady state would be stable in the absence of diffusion. Thus this is a purely diffusive instability. Moreover, it occurs for a finite range of wavenumbers $\mathbf{k}$. Therefore, this instability will form a spatial pattern of some sort since adding up a bunch of sine waves within a finite range of wavelengths should produce a nontrivial wave pattern. This is a Turing bifurcation. Technically, a Turing bifurcation is the destabilization of an other stable steady state by diffusive terms, leading to pattern formation.

## 3 Governing equations

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

$$\begin{cases} u_t(x,t) = \delta + u^2 v - (\rho+1)u + \mu_1 \Delta u \\ \\ v_t(x,t) = \rho u - u^2 v + \mu_2 \Delta v \end{cases} \qquad x \in \Omega \subset \mathbb{R}^2, \quad t \in (0,T], \tag{3}$$

$$\begin{cases} (u(x,t), v(x,t)) = \left(f^{\mathscr{D}}(x,t), g^{\mathscr{D}}(x,t)\right) & x \in \mathscr{D} \subseteq \partial\Omega, \quad t \in [0,T], \\ \\ \left(\frac{\partial u}{\partial n}(x,t), \frac{\partial v}{\partial n}(x,t)\right) = \left(f^{\mathscr{N}}(x,t), g^{\mathscr{N}}(x,t)\right) & x \in \mathscr{N} \subseteq \partial\Omega, \quad t \in [0,T], \end{cases} \tag{4}$$

$$\begin{cases} u(x,0) = u_0(x), \\ \\ v(x,0) = v_0(x), \end{cases} \qquad x \in \overline{\Omega}. \tag{5}$$

where $u_0$, $v_0$, $f^{\mathscr{D}}$, $g^{\mathscr{D}}$, $f^{\mathscr{N}}$ and $g^{\mathscr{N}}$ are known functions, $\Omega \subset \mathbb{R}^2$ is the domain set, $\partial\Omega = \mathscr{N} \cup \mathscr{D}$ is the boundary of the domain set $\Omega$, and $\Delta$ is the Laplace operator.

## 4 Kernel-based trial functions

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

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

of functions on $\Omega$ in the sense

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

and whose inner product is linked to the kernel itself via

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

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 $K_\nu$ is the modified Bessel function of the second kind [Schaback (2011)]. 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 [Wendland (2005)]. 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

$$N_k(x) = \sum_{j=1}^{n} K(x,x_j)c_{jk}, \quad 1 \le k \le n. \tag{6}$$

If $N(x) = (N_1(x),\ldots,N_n(x))$, and $T(x) = (K(x,x_1),\ldots,K(x,x_n))$, from (6) we have $N(x) = T(x)\cdot C$, where $C = \left(c_{jk}\right)_{1\le j,k\le n}$ 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 $V = L$. Hence the condition number of the collocation matrix corresponding to Newton basis functions is smaller than the one corresponding to translated RBFs. Consequently, using Newton basis functions for collocation will lead to more stable methods than using the basis of translates. The Newton basis functions can be recursively calculated and have the property $N_j(x_k) = 0,\ 1 \le k \le j \le n$. If the values of the Newton basis and linear maps $\mathscr{L}$ like derivatives are needed to be calculated at other points, we get the linear systems $V \cdot N^T(x) = T(x)^T$, and $V \cdot \mathscr{L}(N^T(\cdot)) = \mathscr{L}(T(\cdot)^T)$, respectively.

## 5 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

$$\begin{aligned} \tilde{u}(x,t) &= \sum_{j=1}^{m+n+l} \alpha_j(t) N_j(x) \\ \tilde{v}(x,t) &= \sum_{j=1}^{m+n+l} \beta_j(t) N_j(x) \end{aligned} \tag{7}$$

and care for the Dirichlet boundary conditions by solving

$$\tilde{u}(w_i,t) = f^{\mathscr{D}}(w_i,t) = \sum_{j=1}^{m} \alpha_j(t) N_j(w_i),\ 1 \le i \le m,$$

$$\tilde{v}(w_i,t) = g^{\mathscr{D}}(w_i,t) = \sum_{j=1}^{m} \beta_j(t) N_j(w_i),\ 1 \le i \le m,$$

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 $f_i^{\mathscr{D}}$ and $g_i^{\mathscr{D}}$ on the Dirichlet points. We will also need

$$f^{\mathscr{D}'}(w_i,t) = \sum_{j=1}^{m} \alpha_j'(t) N_j(w_i),\ 1 \le i \le m,$$

$$g^{\mathscr{D}'}(w_i,t) = \sum_{j=1}^{m} \beta_j'(t) N_j(w_i),\ 1 \le i \le m,$$

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

$$a_2(t) = (\alpha_{m+1}(t), \ldots, \alpha_{m+n}(t))^T,$$

$$b_2(t) = (\beta_{m+1}(t), \ldots, \beta_{m+n}(t))^T,$$

$$a_3(t) = (\alpha_{m+n+1}(t), \ldots, \alpha_{m+n+l}(t))^T,$$

$$b_3(t) = (\beta_{m+n+1}(t), \ldots, \beta_{m+n+l}(t))^T.$$

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

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

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

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:

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

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

for $m+n+1 \le i \le m+n+l$. We now write the PDE (3) at a point $w_i$, $m+1 \le i \le m+n$ as follows:

$$\begin{aligned}
&\sum_{j=1}^{m} \alpha_j'(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j'(t)N_j(w_i) \\
=&\delta + \left(\sum_{j=1}^{m} \alpha_j(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t)N_j(w_i)\right)^2 \left(\sum_{j=1}^{m} \beta_j(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(t)N_j(w_i)\right) \\
&-(\rho+1)\left(\sum_{j=1}^{m} \alpha_j(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t)N_j(w_i)\right) \\
&+\mu_1\left(\sum_{j=1}^{m} \alpha_j(t)\Delta N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t)\Delta N_j(w_i) + \sum_{j=m+n+1}^{m+n+l} \alpha_j(t)\Delta N_j(w_i)\right), \\
&\sum_{j=1}^{m} \beta_j'(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j'(t)N_j(w_i) \\
=&\rho\left(\sum_{j=1}^{m} \alpha_j(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t)N_j(w_i)\right) \\
&-\left(\sum_{j=1}^{m} \alpha_j(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \alpha_j(t)N_j(w_i)\right)^2 \left(\sum_{j=1}^{m} \beta_j(t)N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(t)N_j(w_i)\right) \\
&+\mu_2\left(\sum_{j=1}^{m} \beta_j(t)\Delta N_j(w_i) + \sum_{j=m+1}^{m+n} \beta_j(t)\Delta N_j(w_i) + \sum_{j=m+n+1}^{m+n+l} \beta_j(t)\Delta N_j(w_i)\right).
\end{aligned} \tag{8}$$

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

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

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

## 6 Implementation

If we introduce suitable column vectors and matrices into the system (8), we have to satisfy

$$\begin{bmatrix} N_3 & \mathbf{0} \\ \mathbf{0} & N_3 \end{bmatrix} \begin{bmatrix} a_2'(t) \\ b_2'(t) \end{bmatrix} = \begin{bmatrix} R_1(a_2,b_2) \\ R_2(a_2,b_2) \end{bmatrix}, \tag{9}$$

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

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

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

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

$$N_1 = (N_j(w_i))_{1\le i\le m, 1\le j\le m},\ N_2 = (N_j(w_i))_{m+1\le i\le m+n, 1\le j\le m},$$

$$N_3 = (N_j(w_i))_{m+1\le i\le m+n, m+1\le j\le m+n},$$

$$a_1(t) = (N_1)^{-1} F^{\mathscr{D}}(t), \qquad b_1(t) = (N_1)^{-1} G^{\mathscr{D}}(t),$$

$$a_1{}'(t) = (N_1)^{-1} F^{\mathscr{D}'}(t), \qquad b_1{}'(t) = (N_1)^{-1} G^{\mathscr{D}'} N(t),$$

$$F^{\mathscr{D}}(t) = (f^{\mathscr{D}}(w_i,t),\ 1 \le i \le m)^T,\ G^{\mathscr{D}}(t) := (g^{\mathscr{D}}(w_i,t),\ 1 \le i \le m)^T,$$

$$F^{\mathscr{D}'}(t) = (f^{\mathscr{D}'}(w_i,t),\ 1 \le i \le m)^T,\ G^{\mathscr{D}'}(t) := (g^{\mathscr{D}'}(w_i,t),\ 1 \le i \le m)^T,$$

$$a_3(t) = \left(\frac{\partial N_3}{\partial n}\right)^{-1} \left(F^{\mathscr{N}}(t) - \frac{\partial N_1}{\partial n} a_1(t) - \frac{\partial N_2}{\partial n} a_2(t)\right)$$

$$b_3(t) = \left(\frac{\partial N_3}{\partial n}\right)^{-1} \left(G^{\mathscr{N}}(t) - \frac{\partial N_1}{\partial n} b_1(t) - \frac{\partial N_2}{\partial n} b_2(t)\right)$$

$$F^{\mathscr{N}}(t) = (f^{\mathscr{N}}(w_i,t),\ m+n+1 \le i \le m+n+l)^T,$$

$$G^{\mathscr{N}}(t) := (g^{\mathscr{N}}(w_i,t),\ m+n+1 \le i \le m+n+l)^T,$$

$$\frac{\partial N_1}{\partial n} = \left(\frac{\partial N_j(w_i)}{\partial n}\right)_{m+n+1\le i\le m+n+l,1\le j\le m},$$
$$\frac{\partial N_2}{\partial n} = \left(\frac{\partial N_j(w_i)}{\partial n}\right)_{m+n+1\le i\le m+n+l,m+1\le j\le m+n},$$
$$\frac{\partial N_3}{\partial n} = \left(\frac{\partial N_j(w_i)}{\partial n}\right)_{m+n+1\le i\le m+n+l,m+n+1\le j\le m+n+l},$$
$$D_1 = (\Delta N_j(w_i))_{m+1\le i\le m+n,1\le j\le m},$$
$$D_2 = (\Delta N_j(w_i))_{m+1\le i\le m+n,m+1\le j\le m+n},$$
$$D_3 = (\Delta N_j(w_i))_{m+1\le i\le m+n,m+n+1\le j\le m+n+l},$$

where $j$ is the column index and $i$ is the row index.

The system (9) is the ODE system generated by the MOL and one can invoke any ODE integrator to solve it. The method of lines thus reduces the PDE problem to a system of ODEs and relies numerically on the ODE solvers invoked to solve these ODEs. The discretization of the PDE does not involve any time–stepping at all, and therefore the time integration plays only a role within the ODEs. But since ODEs are not in the focus of this paper, we ignore time–stepping here.

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. The matrices $N_1$, $N_2$, and $N_3$ are the nonzero blocks of the value matrix of the Newton basis, which is upper triangular and nonsingular. Thus the square matrices $N_1$ and $N_3$ are upper triangular and nonsingular. However, there is no guarantee that the matrices of normal derivatives are nonsingular.

## 7 Numerical results

In this section we present the results of our scheme for the numerical solution of the Brusselator reaction-diffusion system (3)-(5). 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

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

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 uniformly distributed discretization points in the region $\overline{\Omega}$ as shown in Fig. 2. We also take 51 grid points along each axis for plotting of figures.

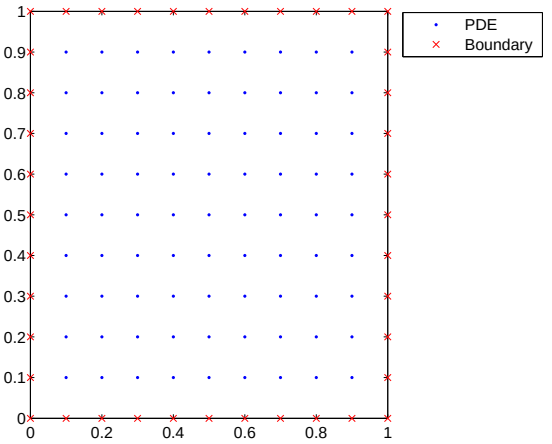



Figure 2: Points distribution in the region $\overline{\Omega}$.

### *7.1 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

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

The concentrations profiles of $u$ and $v$ at different time levels $T = 1$, $T = 3$, $T = 5$, and $T = 20$ are shown in Figs. 3-6, respectively. Absolute and relative error distributions at time $T = 2$ are shown in Figs. 7 and 8, respectively. The results are in agreement with the results of [Jiwari and Yuan (2014)].

### *7.2 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 (3) for the grid points of the region in Fig. 9. The plots of the values of $u$ and $v$ at the collocation point $(0.3, 0.3)$ versus time shown in Fig. 10, indicate that the solutions converge toward the stationary ones $(\delta, \frac{\rho}{\delta})$ as $t$ increases, whenever $1-\rho+\delta^2 > 0$. 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.

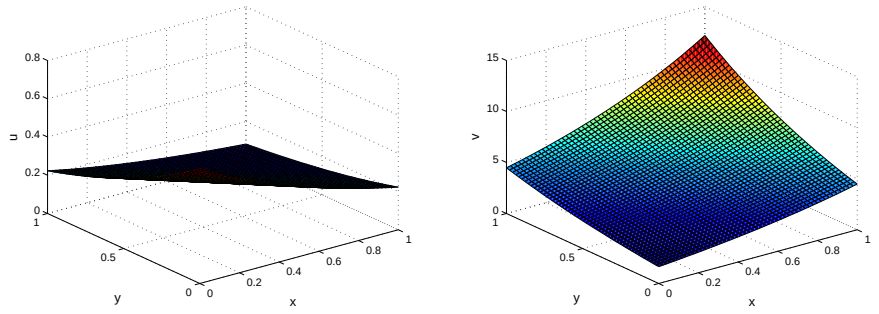



Figure 3: Plots of $u$ and $v$ at $T = 1$, $\rho = 1$, $\delta = 0$. (Test problem 1)

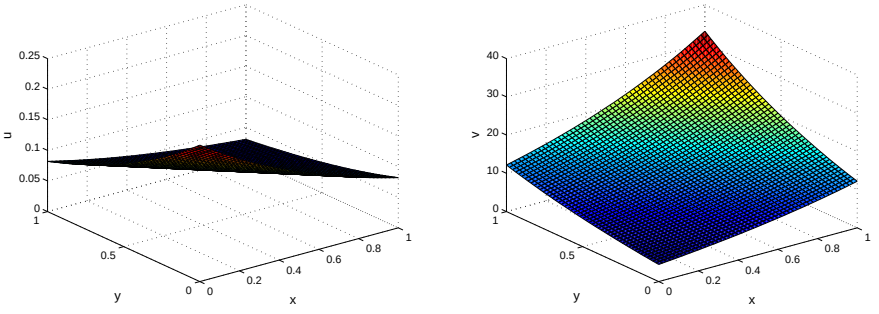



Figure 4: Plots of $u$ and $v$ at $T = 3$, $\rho = 1$, $\delta = 0$. (Test problem 1)

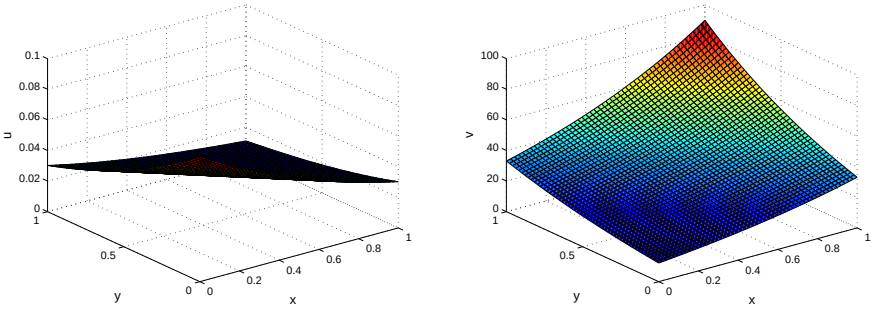



Figure 5: Plots of $u$ and $v$ at $T = 5$, $\rho = 1$, $\delta = 0$. (Test problem 1)

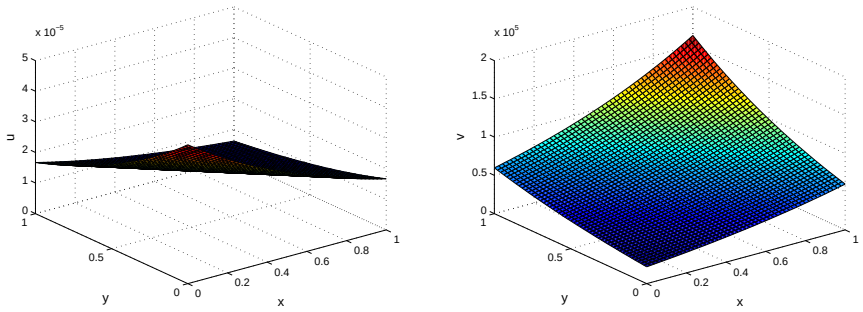



Figure 6: Plots of $u$ and $v$ at $T = 20$, $\rho = 1$, $\delta = 0$. (Test problem 1)

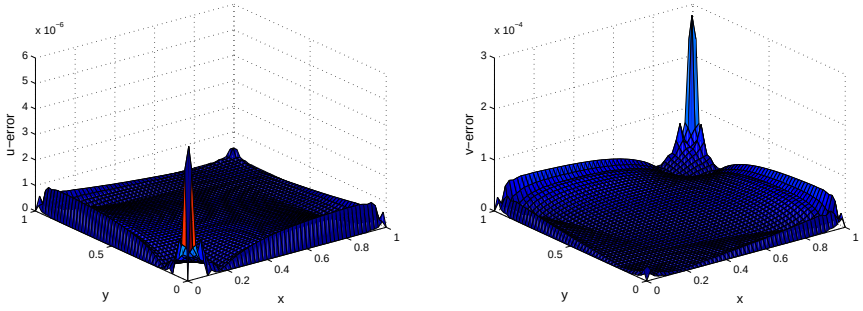



Figure 7: Absolute error graph at time $T = 2$. (Test problem 1)

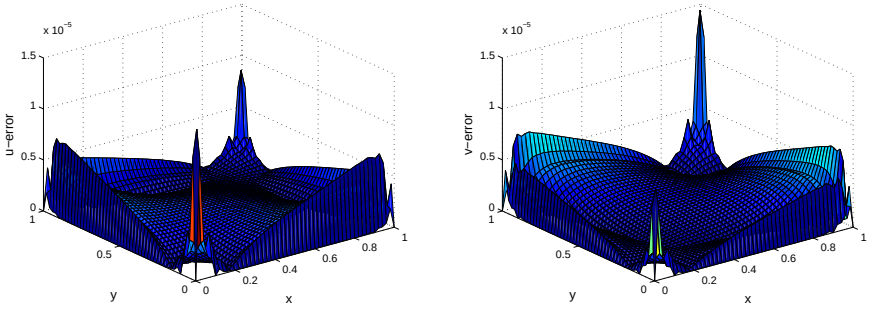
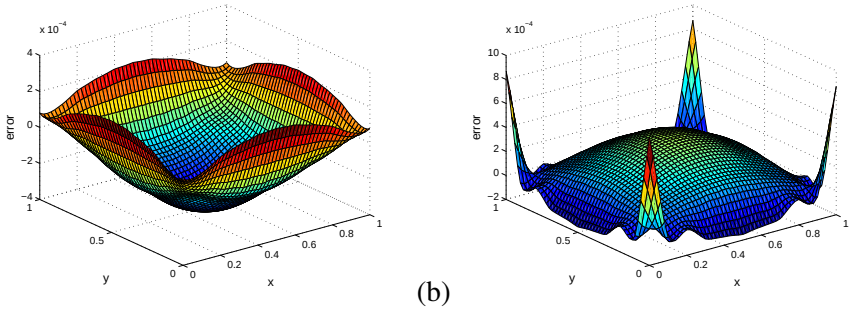
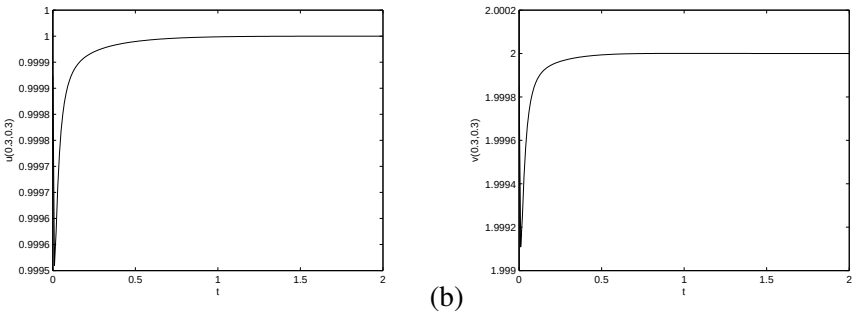



Figure 8: Relative error graph at time $T = 2$. (Test problem 1)




(a) (b)

Figure 9: Error graph of PDE at time $T = 2$. (a) Equation 1; (b) Equation 2. (Test problem 2)




(a) (b)

Figure 10: Plots of $u(0.3,0.3)$ and $v(0.3,0.3)$ versus time. (Test problem 2)

### *7.3 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 time from $T = 1$ to $T = 8$ are shown in Fig. 11. It can be noted from Fig. 11 that $(u,v) \to (\delta, \frac{\rho}{\delta})$ as $t$ increases, whenever $1 - \rho + \delta^2 > 0$. The results show an agreement with the results of [Jiwari and Yuan (2014)]. The plots of the values

of $u$ and $v$ at the collocation point $(0.3, 0.3)$ versus time are shown in Fig. 12. It can be noted from Fig. 12, 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 [Twizell, Gumel, and Cao (1999)] and [ul Islam, Haq, and Ali (2010)].

### *7.4 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 Fig. 13. The plots of the values of $u$ and $v$ at the collocation point $(0.3, 0.3)$ versus time are shown in Fig. 14. It can be noted from Figs. 13 and 14 that the solutions are stable but oscillatory and the numerical method is seen not to converge to any fixed concentration. The results show an agreement with the results of [Twizell, Gumel, and Cao (1999)] and [ul Islam, Haq, and Ali (2010)].

### *7.5 Test problem 5*

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

$$\begin{cases} u(x,y,0) = 0.5 + y, \\ v(x,y,0) = 1 + 5x, \end{cases}$$

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

The plots of the values of $u$ and $v$ at the collocation point $(0.2, 0.2)$ versus time are shown in Fig. 15 with the parameters $\rho = 0.5$, $\delta = 1$, and $\mu_1 = \mu_2 = 0.002$. It can be noted from Fig. 15, that $(u(0.2,0.2), v(0.2,0.2)) \to (1, 0.5)$ as $t \to \infty$. The concentration profiles of $u$ and $v$ at time from $T = 1$ to $T = 15$ are shown in Fig. 16 with the parameters $\rho = 3.4$, $\delta = 1$, and $\mu_1 = \mu_2 = 0.002$. The results are similar to those obtained in [Jiwari and Yuan (2014)]. In the next test problem, we show the Turing pattern occuring in the Brusselator system by our scheme.

### *7.6 Test problem 6*

Consider the Brusselator system with the parameters $\rho = 8.72$, $\delta = 4.5$, $\mu_1 = 1$, and $\mu_2 = 8$. Fig. 17 shows the Turing pattern obtained at time $T = 20$ starting from the initial conditions which are random perturbations around the stationary state $(\delta, \frac{\rho}{\delta})$, with no flux boundary conditions. This spotty pattern verifies the fact that in two dimensions, reaction-diffusion systems typically exhibit either stripes or spots [Leppänen (2004)].

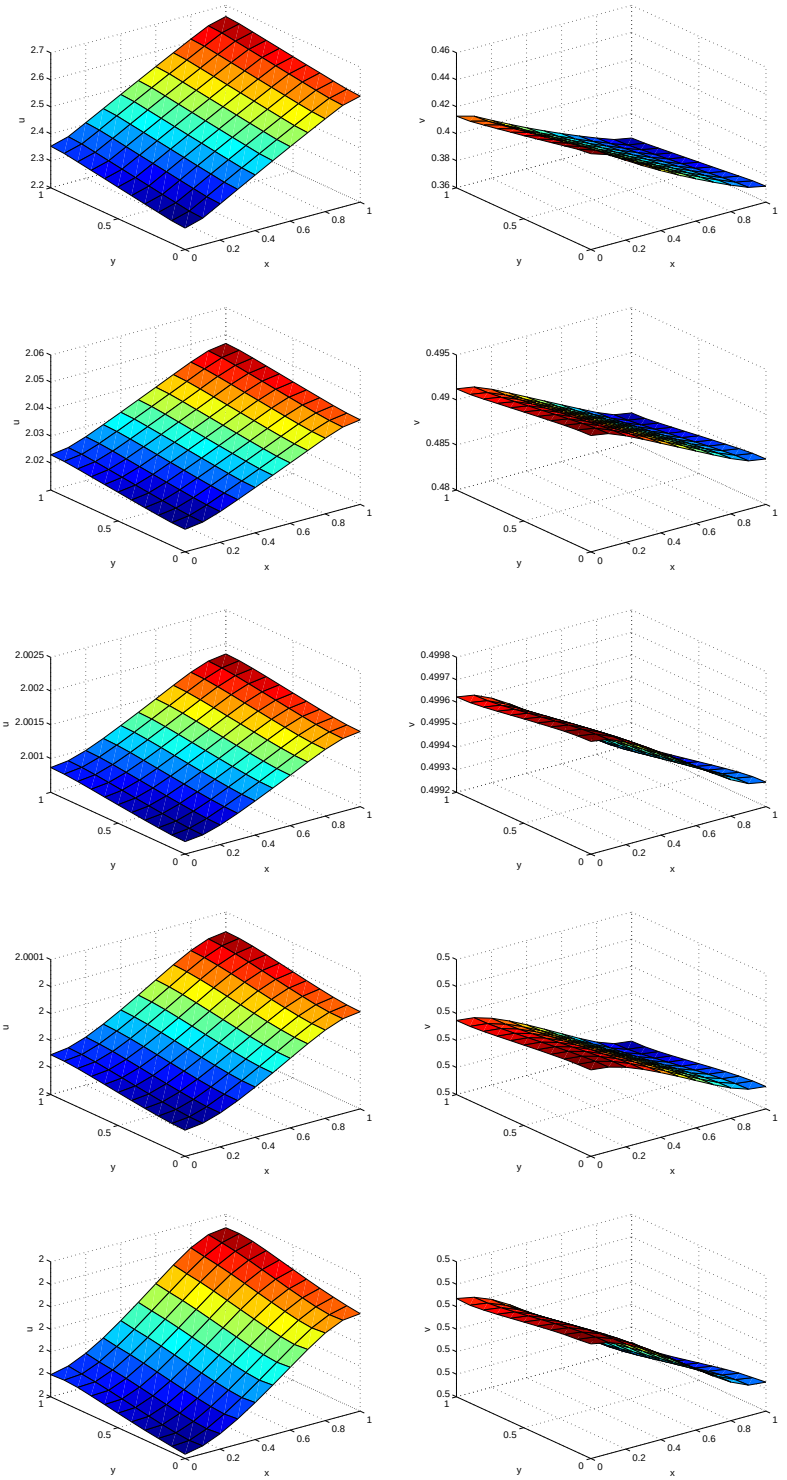



Figure 11: Plots of $u$ and $v$ at times $T = 1,3,5,7,8$, $\rho = 1$, $\delta = 2$. (Test problem 3)

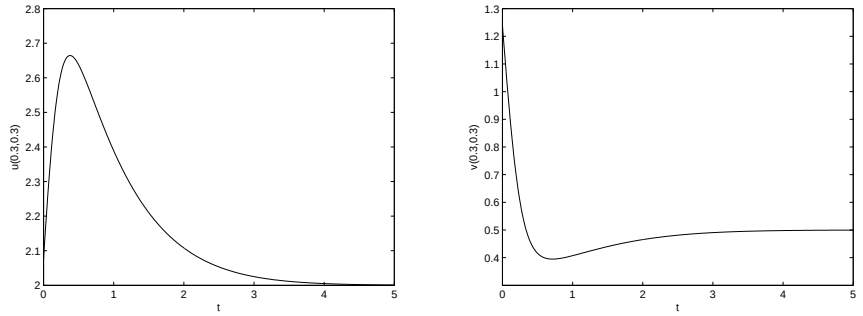



Figure 12: Plots of $u(0.3,0.3)$ and $v(0.3,0.3)$ versus time. (Test problem 3)

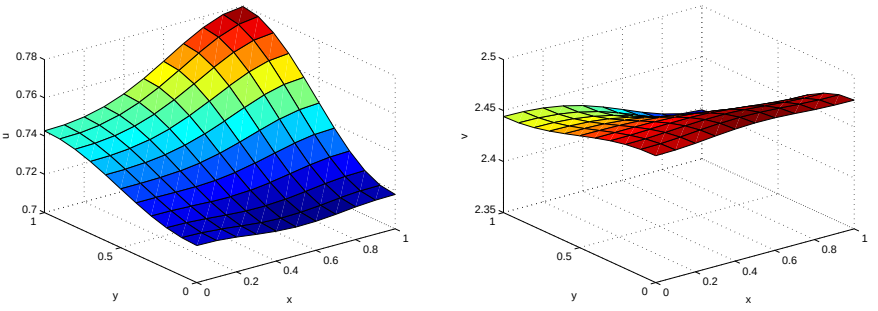



Figure 13: Plots of $u$ and $v$ at $T = 40$, $\rho = 3.4$, $\delta = 1$. (Test problem 4)

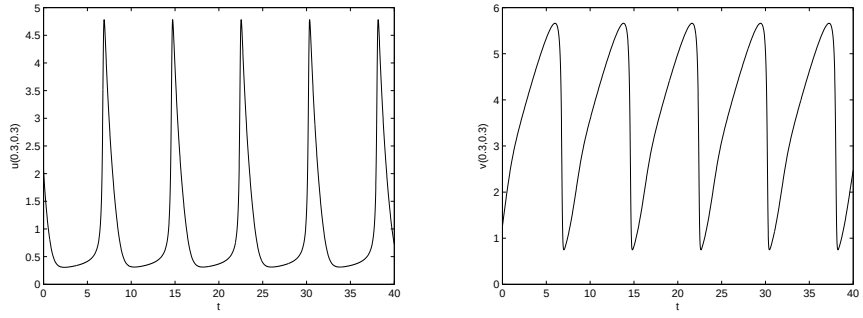



Figure 14: Plots of $u(0.3,0.3)$ and $v(0.3,0.3)$ versus time. (Test problem 4)

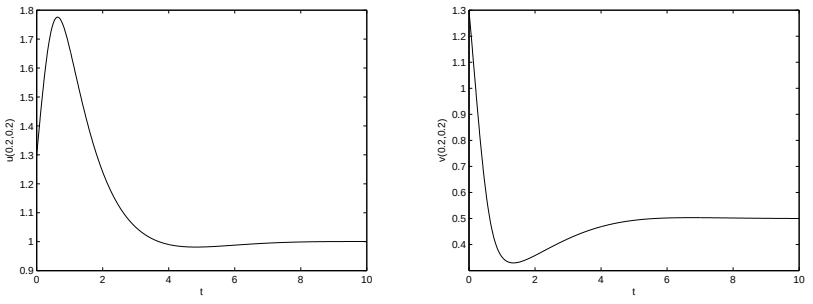
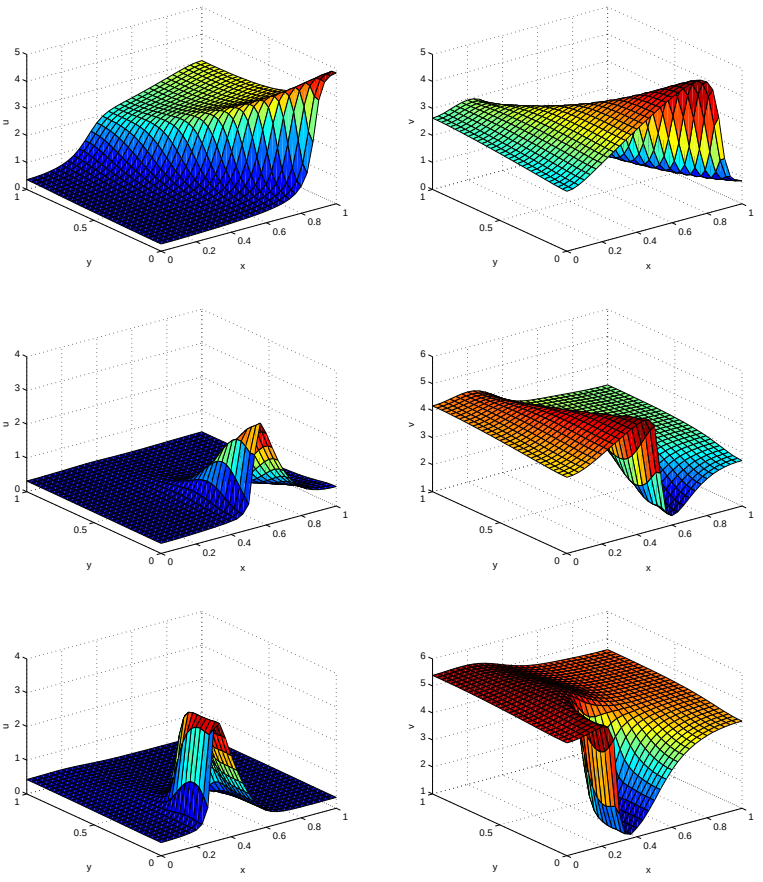



Figure 15: Plots of $u(0.2, 0.2)$ and $v(0.2, 0.2)$ versus time. (Test problem 5)



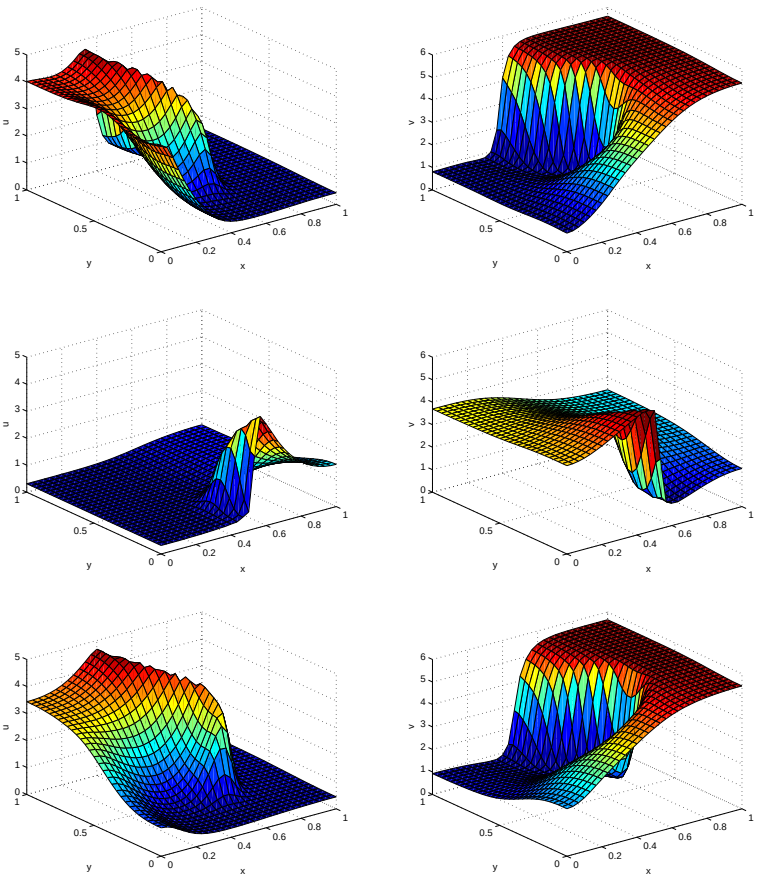


Figure 16: Plots of $u$ and $v$ at times $T = 1,3,5,7,10,15$, $\rho = 3.4$, $\delta = 1$. (Test problem 5)

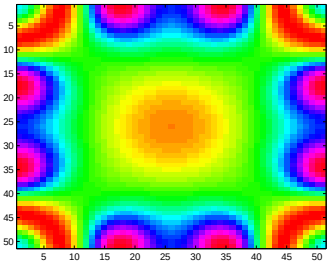


Figure 17: Turing pattern with the parameters $\rho = 8.72$, $\delta = 4.5$, $\mu_1 = 1$, $\mu_2 = 8$, and no flux boundary conditions. (Test problem 6 )

## 8 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 Hopf bifurcations and Turing patterns.

## References

**Adomian, G.** (1995): The diffusion Brusselator equation. *Comput. Math. Appl.*, vol. 29, pp. 1–3.

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

**Dereli, Y.; Schaback, R.** (2010): The meshless kernel-based method of lines for solving the equal width equation. Preprint Göttingen, 2010.

**Dong, L.; Alotaibi, A.; Mohiuddine, S.; Atluri, S. N.** (2014): Computational methods in engineering: A variety of primal & mixed methods, with global & local interpolations, for well-posed or ill-posed bcs. *CMES: Comput. Modeling Eng. Sci.*, vol. 99, pp. 1–85.

**Fasshauer, G.** (1997): Solving partial differential equations by collocation with radial basis functions. pp. 131–138. Vanderbilt University Press, Nashville, TN.

**Franke, C.; Schaback, R.** (1998): Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. Comput. Math.*, vol. 8, pp. 381–399.

**Franke, C.; Schaback, R.** (1998): Solving partial differential equations by collocation using radial basis functions. *Appl. Math. Comp.*, vol. 93, pp. 73–82.

**Gumel, A.; Langford, W.; Twizel, E.; Wu, J.** (2000): Numerical solutions for a coupled non-linear oscillator. *J. Math. Chem.*, vol. 28, pp. 325–340.

**Han, Z. D.; Atluri, S. N.** (2014): On the (Meshless local Petrov-Galerkin) MLPG-Eshelby method in computational finite deformation solid mechanics - part II. . *CMES: Comput. Modeling Eng. Sci.*, vol. 97, pp. 199–237.

**Herschkowitz-Kaufman, M.; Nicolis, N.** (1972): Localized spatial structures and non-linear chemical waves in dissipative systems. *J. Chem. Phys.*, vol. 56, pp. 1890–1895.

**Hon, Y. C.; Schaback, R.** (2008): Solvability of partial differential equations by meshless kernel methods. *Adv. Comput. Math.*, vol. 28, pp. 283–299.

**Jiwari, R.; Yuan, J.** (2014): A computational modeling of the behavior of the two-dimensional reaction-diffusion Brusselator system arising in chemical processes. *J. Math. Chem.*

**Kansa, E. J.** (1986): Application of Hardy's multiquadric interpolation to hydrodynamics. In *Proc. 1986 Simul. Conf., Vol. 4*, pp. 111–117.

**Lavenda, B.; Nicolis, G.; Herschkowitz-Kaufman, M.** (1971): Chemical instabilities and relaxation oscillations. *J. Theor. Biol.*, vol. 32, pp. 283–292.

**Lee, C.-F.; Ling, L.; Schaback, R.** (2009): On convergent numerical algorithms for unsymmetric collocation. *Adv. Comput. Math.*, vol. 30, pp. 339–354.

**Lee, C. H.; Park, K.; Jang, B.** (2013): Multistage homotopy perturbation method for nonlinear reaction networks. *J. Math. Chem.*, vol. 51, pp. 1945–1960.

**Lefever, R.** (1968): Dissipative structures in chemical systems. *J. Chem. Phys.*, vol. 49, pp. 4977–4978.

**Lefever, R.; Nicolis, G.** (1971): Chemical instabilities and sustained oscillations. *J. Theor. Biol.*, vol. 30, pp. 267–284.

**Leppänen, T.** (2004): The theory of Turing pattern formation. available online at http://www.lce.hut.fi/research/polymer/turing_review.pdf, 2004.

**Lin, Y.; Liu, Y.; Li, Z.** (2012): Symbolic computation of analytic approximate solutions for nonlinear differential equations with initial conditions. *Comput. Physics Commun.*, vol. 183, pp. 106–117.

**Ma, T.; Wang, S.** (2011): Phase transitions for the Brusselator model. *J. Math. Phys.*, vol. 52, pp. 1–23.

**Mittal, R.; Jiwari, R.** (2011): Numerical study of two-dimensional reaction-diffusion Brusselator system by differential quadrature method. *Int. J. Comput. Methods Eng. Sci. Mech.*, vol. 12, pp. 14–25.

**Mohammadi, M.; Mokhtari, R.** (2011): Solving the generalized regularized long wave equation on the basis of a reproducing kernel space. *J. Comput. Appl. Math.*, vol. 235, pp. 4003–4014.

**Mohammadi, M.; Mokhtari, R.** (2013): A new algorithm for solving nonlinear Schrödinger equation in the reproducing kernel space. *Iranian J. Sci. Tech. Section A. Sci.*, vol. 37, pp. 523–546.

**Mohammadi, M.; Mokhtari, R.** (2014): A reproducing kernel method for solving a class of nonlinear systems of PDEs. *Math. Model. Anal.*, vol. 19, pp. 180–198.

**Mohammadi, M.; Mokhtari, R.; Panahipour, H.** (2013): A Galerkin-reproducing kernel method: Application to the 2d nonlinear coupled Burgers' equations. *Eng. Anal. Bound. Elem.*, vol. 37, pp. 1642–1652.

**Mohammadi, M.; Mokhtari, R.; Panahipour, H.** (2014): Solving two parabolic inverse problems with a nonlocal boundary condition in the reproducing kernel space. *Appl. Comput. Math.*, vol. 13, pp. 91–106.

**Mohammadi, M.; Mokhtari, R.; Toutian Isfahani, F.** (2014): Solving an inverse problem for a parabolic equation with a nonlocal boundary condition in the reproducing kernel space. *Iranian J. Numer. Anal. Optimization*, vol. 5, pp. 57–76.

**Mokhtari, R.; Mohammadi, M.** (2010): Numerical solution of GRLW equation using sinc-collocation method. *Comput. Physics Commun.*, vol. 181, pp. 1266–1274.

**Mokhtari, R.; Mohseni, M.** (2012): A meshless method for solving mKdv equation. *Comput. Physics Commun.*, vol. 183, pp. 1259–1268.

**Mokhtari, R.; Toutian Isfahani, F.; Mohammadi, M.** (2012): Solving a class of nonlinear differential-difference equations in the reproducing kernel space. *Abstr. Appl. Anal.* Article ID 514103.

**Müller, S.; Schaback, R.** (2009): A Newton basis for kernel spaces. *J. Approx. Theory*, vol. 161, pp. 645–655.

**Nicolis, G.; Prigogine, I.** (1977): *Self-organization in non-equilibrium systems*. Wiley Interscience, New York.

**Pazouki, M.; Schaback, R.** (2011): Bases for kernel-based spaces. *J. Comput. Appl. Math.*, vol. 236, pp. 575–588.

**Prigogine, I.; Lefever, R.** (1968): Symmetries breaking instabilities in dissipative systems ii. *J. Phys. Chem.*, vol. 48, pp. 1695–1700.

**Roussel, M.** (2005): Reaction-diffusion equations. available online at http://www.people.uleth.ca/ roussel/nld/Turing.pdf, 2005.

**Schaback, R.** (2011): Kernel–based meshless methods. Lecture Note, Göttingen, http://num.math.uni-goettingen.de/schaback/teaching/AV2.pdf, 2011.

**Schaback, R.** (2013): A computational tool for comparing all linear PDE solvers. submitted, http://www.num.math.uni-goettingen.de/schaback/research/group.html, 2013.

**Schaback, R.; Wendland, H.** (2006): Kernel techniques: from machine learning to meshless methods. *Acta Numerica*, vol. 15, pp. 543–639.

**Twizell, E.; Gumel, A.; Cao, Q.** (1999): A second-order scheme for the "Brusselator" reaction-diffusion system. *J. Math. Chem.*, vol. 26, pp. 297–316.

**ul Islam, S.; Haq, S.; Ali, A.** (2010): A computational modeling of the behavior of the two-dimensional reaction-diffusion Brusselator system. *Appl. Math. Model.*, vol. 34, pp. 3896–3909.

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

**Wendland, H.** (2005): *Scattered Data Approximation*. Cambridge University Press.