

A Meshless Simulations for 2D Nonlinear Reaction-diffusion Brusselator System

Ahmad Shirzadi¹, Vladimir Sladek², Jan Sladek³

Abstract: This paper is concerned with the development of a numerical approach based on the Meshless Local Petrov-Galerkin (MLPG) method for the approximate solutions of the two dimensional nonlinear reaction-diffusion Brusselator systems. The method uses finite differences for discretizing the time variable and the moving least squares (MLS) approximation for field variables. The application of the weak formulation with the Heaviside type test functions supported on local subdomains (around the nodes used in MLS approximation) to semi-discretized partial differential equations yields the finite-volume local weak formulation. A predictor-corrector scheme is used to handle the nonlinearity of the problem within each time step. Numerical test problems are given to verify the accuracy of the proposed method. Under particular conditions, this system exhibits Turing instability which results in a pattern forming instability. This concept is studied and a test problem is given.

Keywords: Meshless Local Petrov-Galerkin method; Moving least squares; Finite differences; Nonlinear problems; Brusselator equations; Turing instability; Pattern formation.

1 Introduction

In 1952 Turing showed [Turing (1952)] that a simple mathematical model describing self-diffusing and reacting chemicals could give rise to stationary spatial concentration patterns of fixed characteristic length from a random initial configuration and proposed that *reaction-diffusion models* might have relevance in describing *morphogenesis*, the growth of biological form. In the late 1960s, Ilya

¹ Department of Mathematics, Persian Gulf University, Bushehr, Iran. Email: shirzadi@pgu.ac.ir, shirzadi.a@gmail.com

² Institute of Construction and Architecture, Slovak Academy of Sciences, 84503 Bratislava, Slovakia. Email: vladimir.sladek@savba.sk

³ Institute of Construction and Architecture, Slovak Academy of Sciences, 84503 Bratislava, Slovakia. Email: jan.sladek@savba.sk

Prigogine among others at the University of Brussels developed the Brusselator model [Prigogine and Lefever (1968); Nicolis and Prigogine (1997)] which is one of the simplest reaction-diffusion models exhibiting Turing instability. From the mathematical point of view, the Brusselator is a system of two coupled nonlinear partial differential equations. The linear stability analysis can be used for predicting the parameter values that result in the Turing instability in a particular reaction-diffusion system. Based on the linear approximation one can also effectively predict the characteristic length of the resulting pattern. Linear analysis cannot, however, predict the spatial characteristics of the resulting patterns, since the pattern selection is governed by complex nonlinear dynamics. Thus some nonlinear analysis is needed. Since analytical solutions are unavailable, finding efficient numerical methods for solving these kind of problems has been an active area of research. Meshless methods as a tool for numerical simulations of linear or nonlinear problems, especially partial differential equations have become popular in recent years. This paper aims to present a new meshless method for the numerical simulations of the Brusselator equations. The proposed method is based on the MLPG formulation [Atluri and Zhu (1998a); Atluri and Zhu (1998b); Atluri, Kim, and Cho (1999); Atluri (2004); Atluri and Shen (2002)] which uses the local weak form equations and the MLS approximation to transform the model equations to the final system of algebraic equations. An application of the MLPG method for numerically solving coupled pair of nonlinear diffusion equations can be found in [Abbasbandy, Sladek, Shirzadi, and Sladek (2011)]. For a review of applications of the MLPG method in engineering and sciences the readers are referred to [Sladek, Stanak, Han, Sladek, and Atluri (2013)]. Analysis of the convergence of the MLPG method is well studied in [Shirzadi and Ling (2013)]. The non-linear reaction-diffusion Brusselator system is given by:

$$\begin{aligned}\frac{\partial u}{\partial t} &= A + u^2v - (B + 1)u + D_u \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} &= Bu - u^2v + D_v \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right),\end{aligned}\quad (1)$$

in the two-dimensional region $\Omega = [a, b]^2$, with initial condition

$$\begin{aligned}u(x, y, 0) &= f(x, y), \\ v(x, y, 0) &= g(x, y),\end{aligned}\quad (2)$$

and Dirichlet boundary condition:

$$\begin{aligned}u(\mathbf{x}, t) &= q_1(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega \times t, \quad t > 0, \\ v(\mathbf{x}, t) &= q_2(\mathbf{x}, t),\end{aligned}$$

or subject to Neumann's boundary conditions on the boundary $\partial\Omega$. Recall that A denotes the value of the source term and B is the bifurcation parameter setting the distance to the onset of instability. The chemical u is the activator concentration and v is the inhibitor. D_u and D_v are diffusion coefficients, f , g , q_1 and q_2 are known functions. Some existing papers on the study of this problem were concerned with theoretical concepts such as symmetry-breaking and bifurcations [Nicolis and Prigogine (1997);Walgraef, Dewel, and Borckmans (1980)]. The theory by Turing [Turing (1952)] predicts that a spatially uniform stationary state that is stable against perturbations in the absence of diffusion might become unstable against perturbations in the presence of diffusion resulting in a pattern forming instability. Some papers on numerical studies of this model employed the computational approach and addressed the problem of pattern selection as a function of system parameters in two-dimensional and three-dimensional systems [Verdasca, Wit, Dewel, and Borckmans (1992);Borckmans, Wit, and Dewel (1992);Leppänen (2004)]. For papers which deal with the numerical solution of the reaction-diffusion Brusselator system see Twizell et al. [Twizell, Gumel, and Cao (1999)] for a second order finite difference method, Adomian [Adomian (1995)] for decomposition method, Whye-Teong [Ang (2003)] for the dual-reciprocity boundary element method, S-u. Islam et al. [Siraj-ul, Arshed, and Sirajul (2010)] for a numerical scheme based on RBF collocation method, Mittal and Jiwari [Mittal and Jiwari (2011)] for polynomial based differential quadrature method, Kamranian et al. [Kamranian, Tatari, and Dehghan (2011)] for the finite point method and so on. The remaining content of the paper is organized as follows. Section 2 describes the basics of the MLS approximation used in this study. The time discretization approach is presented in Section 3. Construction of local weak equations are illustrated in Section 4 and the obtained local weak equations are discretized in Section 5. The predictor-corrector scheme which is proposed to handle the nonlinear terms is also illustrated in Section 5. To simplify the evaluation of local integrals, certain techniques are described in Section 6. Two test problems are given in Section 7 to verify the accuracy of the method. To address the problem of pattern selection as a function of system parameters, another test problem is given at the end of Section 7. Our conclusions are summarized in the last section.

2 The MLS approximation

Consider a sub-domain $\Omega_{\mathbf{x}}$, the neighborhood of a point \mathbf{x} which is located in the problem domain Ω . To approximate the distribution of function u in $\Omega_{\mathbf{x}}$ over a number of randomly located nodes $\mathbf{x}^i, i = 1, 2, \dots, n$, the MLS approximant $u^h(\mathbf{x})$ of u , $\forall \mathbf{x} \in \Omega_{\mathbf{x}}$, is defined by

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_{\mathbf{x}}, \quad (3)$$

where $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_m(\mathbf{x})]$ is a complete monomial basis of order m , and $\mathbf{a}(\mathbf{x})$ is a vector containing coefficients $a_j(\mathbf{x}), j = 1, 2, \dots, m$ which are functions of the space coordinates \mathbf{x} . For example, for a 2-D problem, $\mathbf{p}^T(\mathbf{x}) = [1, x, y]$ and $\mathbf{p}^T(\mathbf{x}) = [1, x, y, x^2, xy, y^2]$, for linear basis ($m = 3$) and quadratic basis ($m = 6$), respectively. The coefficient vector $\mathbf{a}(\mathbf{x})$ is determined by minimizing a weighted discrete L_2 norm, defined as

$$\begin{aligned}
 J(x) &= \sum_{i=1}^n w_i(\mathbf{x}) [p^T(\mathbf{x}_i)\mathbf{a}(\mathbf{x}) - \hat{u}_i]^2 \\
 &= [\mathbf{P}\cdot\mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}]^T \cdot W \cdot [\mathbf{P}\cdot\mathbf{a}(\mathbf{x}) - \hat{\mathbf{u}}],
 \end{aligned}
 \tag{4}$$

where $w_i(\mathbf{x})$ is the weight function associated with the node i , with $w_i(\mathbf{x}) > 0$ for all \mathbf{x} in the support of $w_i(\mathbf{x})$, \mathbf{x}^i denotes the value of \mathbf{x} at node i , n is the number of nodes in $\Omega_{\mathbf{x}}$ for which the weight functions $w_i(\mathbf{x}) > 0$, the matrices \mathbf{P} and \mathbf{W} are defined as

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}^T(\mathbf{x}^1) \\ \mathbf{p}^T(\mathbf{x}^2) \\ \dots \\ \mathbf{p}^T(\mathbf{x}^n) \end{pmatrix}_{n \times m}, \quad \mathbf{W} = \begin{pmatrix} w_1(\mathbf{x}) & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & w_n(\mathbf{x}) \end{pmatrix},$$

and $\hat{\mathbf{u}}^T = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n]$ where $\hat{u}_i, i = 1, 2, \dots, n$ in (4) are the fictitious nodal values, and not the nodal values of the unknown trial function $u^h(\mathbf{x})$ in general. In this work the Gaussian weight function is used:

$$w_i(\mathbf{x}) = \begin{cases} \frac{\exp[-(\frac{d_i}{c_i})^2] - \exp[-(\frac{r_i}{c_i})^2]}{1 - \exp[-(\frac{r_i}{c_i})^2]}, & 0 \leq d_i \leq r_i, \\ 0, & d_i \geq r_i, \end{cases}$$

where $d_i = \|\mathbf{x} - \mathbf{x}^i\|$, c_i is a constant controlling the shape of the weight function w_i and r_i is the size of the support domain. The stationarity of J in (4) with respect to $\mathbf{a}(\mathbf{x})$ leads to the following linear relation between $\mathbf{a}(\mathbf{x})$ and $\hat{\mathbf{u}}$

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\hat{\mathbf{u}}, \tag{5}$$

where the matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ are defined by

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T \mathbf{W} \mathbf{P} = \mathbf{B}(\mathbf{x}) \mathbf{P} = \sum_{i=1}^n w_i(\mathbf{x}) \mathbf{p}(\mathbf{x}^i) \mathbf{p}^T(\mathbf{x}^i),$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^T \mathbf{W} = [w_1(\mathbf{x})\mathbf{p}(\mathbf{x}^1), w_2(\mathbf{x})\mathbf{p}(\mathbf{x}^2), \dots, w_n(\mathbf{x})\mathbf{p}(\mathbf{x}^n)].$$

The MLS approximation is well defined if and only if the rank of \mathbf{P} equals m . A necessary condition for a well-defined MLS approximation is that at least m weight functions are non-zero (i.e. $n > m$) for each sample point $\mathbf{x} \in \Omega$ and that the nodes in $\Omega_{\mathbf{x}}$ will not be arranged in a special pattern such as on a straight line. Solving for $\mathbf{a}(\mathbf{x})$ from (5) and substituting it into (3) gives a relation which may be written as the form of an interpolation function similar to that used in FEM, as

$$u^h(\mathbf{x}) = \Phi^T(\mathbf{x}) \cdot \hat{\mathbf{u}} = \sum_{i=1}^n \phi_i(\mathbf{x}) \hat{u}_i; \quad u^h(\mathbf{x}^i) \equiv u_i; \quad \mathbf{x} \in \Omega_{\mathbf{x}}, \quad (6)$$

and essentially $u_i \neq \hat{u}_i$, where

$$\Phi^T(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}),$$

or

$$\phi_i(\mathbf{x}) = \sum_{j=1}^m p_j(\mathbf{x}) [\mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x})]_{ji}.$$

$\phi_i(\mathbf{x})$ is called the shape function of the MLS approximation corresponding to nodal point \mathbf{y}^i . The support of the nodal point \mathbf{y}^i is usually taken to be a circle of radius r_i , centered at \mathbf{y}^i . The fact that $\phi_i(\mathbf{x}) = 0$, for \mathbf{x} not in the support of nodal point \mathbf{y}^i preserves the local character of the Moving Least Squares approximation. The partial derivatives of $\phi_i(\mathbf{x})$ are obtained as

$$\phi_{i,k} = \sum_{j=1}^m [p_{j,k}(\mathbf{A}^{-1} \mathbf{B})_{ji} + p_j(\mathbf{A}^{-1}_{,k} \mathbf{B}_{,k} + \mathbf{A}_{,k}^{-1} \mathbf{B})_{ji}], \quad (7)$$

in which $\mathbf{A}_{,k}^{-1} = (\mathbf{A}^{-1})_{,k}$ represents the derivative of the inverse of A with respect to x_k , k th coordinate of \mathbf{x} , which is given by $\mathbf{A}_{,k}^{-1} = -\mathbf{A}^{-1} \mathbf{A}_{,k} \mathbf{A}^{-1}$, where $(\)_{,i}$ denotes $\partial(\) / \partial x_i$.

3 The finite difference approximation

The finite-difference approximation of the time derivatives in the θ method is given as follows

$$\theta \dot{u}_{k+1} + (1 - \theta) \dot{u}_k = \frac{u_{k+1} - u_k}{\Delta t}, \quad 0 \leq \theta \leq 1. \quad (8)$$

Considering Eq. (1) at the time instants $k\Delta t$ and $(k+1)\Delta t$, one obtains, respectively

$$\begin{aligned} \theta \dot{u}_{k+1} &= \theta D_u \nabla^2 u_{k+1} + \theta A + \theta (u_{k+1})^2 v_{k+1} - \theta (B+1) u_{k+1} \\ (1 - \theta) \dot{u}_k &= (1 - \theta) D_u \nabla^2 u_k + (1 - \theta) A + (1 - \theta) (u_k)^2 v_k - (1 - \theta) (B+1) u_k \end{aligned}$$

Hence and from (8), we have

$$\begin{aligned} \frac{u_{k+1} - u_k}{\Delta t} &= D_u \nabla^2 u_k + \theta D_u (\nabla^2 u_{k+1} - \nabla^2 u_k) + A + \theta (u_{k+1})^2 v_{k+1} \\ &+ (1 - \theta) (u_k)^2 v_k - (B + 1) (\theta u_{k+1} + (1 - \theta) u_k) \end{aligned} \quad (9)$$

Using Crank-Nicholson scheme ($\theta = \frac{1}{2}$) Eq. (9) becomes:

$$\begin{aligned} \frac{u_{k+1} - u_k}{\Delta t} &= D_u \frac{1}{2} \nabla^2 (u_k + u_{k+1}) + A + \frac{1}{2} ((u_{k+1})^2 v_{k+1} + (u_k)^2 v_k) \\ &- \frac{1}{2} (B + 1) (u_{k+1} + u_k) \end{aligned}$$

or

$$\begin{aligned} &\left[1 + \frac{\Delta t}{2} (B + 1) - D_u \frac{\Delta t}{2} \nabla^2 \right] u_{k+1} - \frac{\Delta t}{2} (u_{k+1})^2 v_{k+1} \\ &= \left[1 - \frac{\Delta t}{2} (B + 1) + D_u \frac{\Delta t}{2} \nabla^2 \right] u_k + \frac{\Delta t}{2} (u_k)^2 v_k + \Delta t A. \end{aligned} \quad (10)$$

Similarly, we have

$$\begin{aligned} &\left[1 - D_v \frac{\Delta t}{2} \nabla^2 \right] v_{k+1} - \frac{\Delta t}{2} B u_{k+1} + \frac{\Delta t}{2} (u_{k+1})^2 v_{k+1} \\ &= \left[1 + D_v \frac{\Delta t}{2} \nabla^2 \right] v_k + \frac{\Delta t}{2} B u_k - \frac{\Delta t}{2} (u_k)^2 v_k \end{aligned} \quad (11)$$

Therefore, the time dependent PDEs are transformed to the semi-discrete PDEs of the elliptic type for the field variables u_{k+1} and v_{k+1} , assuming the fields u_k and v_k being known from the computation in the previous time step. To treat the nonlinearity, a predictor-corrector scheme is proposed in this paper.

4 The local weak form

In this paper, we construct the weak form over local sub-domains such as Ω_{s^i} , located entirely inside Ω which is a circle of radius r_0 and centered at node \mathbf{x}^i . The local weak form of the equations (10) and (11) for $\mathbf{x}^i = (x^i, y^i) \in \Omega_{s^i}$ can be written as

$$\begin{aligned} &\int_{\Omega_{s^i}} \left[\left[1 + \frac{\Delta t}{2} (B + 1) - D_u \frac{\Delta t}{2} \nabla^2 \right] u_{k+1} - \frac{\Delta t}{2} (u_{k+1})^2 v_{k+1} \right] u^* d\mathbf{x} = \\ &\int_{\Omega_{s^i}} \left[\left[1 - \frac{\Delta t}{2} (B + 1) + D_u \frac{\Delta t}{2} \nabla^2 \right] u_k + \frac{\Delta t}{2} (u_k)^2 v_k + \Delta t A \right] u^* d\mathbf{x}, \end{aligned} \quad (12)$$

$$\int_{\Omega_{s^i}} \left[\left[1 - D_v \frac{\Delta t}{2} \nabla^2 \right] v_{k+1} - \frac{\Delta t}{2} B u_{k+1} + \frac{\Delta t}{2} (u_{k+1})^2 v_{k+1} \right] u^* d\mathbf{x} =$$

$$\int_{\Omega_{s^i}} \left[\left[1 + D_v \frac{\Delta t}{2} \nabla^2 \right] v_k + \frac{\Delta t}{2} B u_k - \frac{\Delta t}{2} (u_k)^2 v_k \right] u^* d\mathbf{x}$$
(13)

where u and v are trial functions and u^* is a test function. If the Heaviside step function

$$u^*(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_{s^i}, \\ 0, & \mathbf{x} \notin \Omega_{s^i}, \end{cases}$$

is chosen as the test function in each sub-domain, then $\nabla u^* = 0$ and the local weak forms (12) and (13) are transformed into the following equations

$$\left(1 + \Delta t \frac{(B+1)}{2} \right) \int_{\Omega_{s^i}} u_{k+1} d\mathbf{x} - \frac{D_u \Delta t}{2} \int_{\partial \Omega_{s^i}} \frac{\partial u_{k+1}}{\partial n} ds$$

$$- \frac{\Delta t}{2} \int_{\Omega_{s^i}} (u_{k+1})^2 v_{k+1} d\mathbf{x} = \left(1 - \Delta t \frac{(B+1)}{2} \right) \int_{\Omega_{s^i}} u_k d\mathbf{x}$$

$$+ \frac{D_u \Delta t}{2} \int_{\partial \Omega_{s^i}} \frac{\partial u_k}{\partial n} ds + \Delta t \int_{\Omega_{s^i}} \left(\frac{1}{2} (u_k)^2 v_k + A \right) d\mathbf{x},$$
(14)

$$\int_{\Omega_{s^i}} v_{k+1} d\mathbf{x} - \frac{D_v \Delta t}{2} \int_{\partial \Omega_{s^i}} \frac{\partial v_{k+1}}{\partial n} ds$$

$$+ \frac{\Delta t}{2} \int_{\Omega_{s^i}} \left((u_{k+1})^2 v_{k+1} - B u_{k+1} \right) d\mathbf{x} = \int_{\Omega_{s^i}} v_k d\mathbf{x} + \frac{D_v \Delta t}{2} \int_{\partial \Omega_{s^i}} \frac{\partial v_k}{\partial n} ds$$

$$- \frac{\Delta t}{2} \int_{\Omega_{s^i}} \left((u_k)^2 v_k - B u_k \right) d\mathbf{x}$$
(15)

5 Discretized equations and the predictor-corrector scheme

We consider N nodal points, some of them located on the boundaries and some in the domain. Let $u_k(\mathbf{x}) = u(\mathbf{x}, k\Delta t)$ be approximated according to (6) with denoting the nodal unknowns as $\hat{u}_{k,i}$. In each time step, our aim is to compute $\hat{u}_{k+1,i}$, $i = 1, 2, \dots, N$ assuming $\hat{u}_{k,i}$, $i = 1, 2, \dots, N$ being known from the previous time step. Since there are nonlinear terms in the equations, we propose a predictor-corrector scheme in each time step to obtain a desired accuracy. So we assume that $\hat{u}_{k+1}^{(l)}$ is an approximation of \hat{u}_{k+1} after l correction step and \hat{u}_{k+1} is identified with the value $\hat{u}_{k+1}^{(f)}$ obtained in the final correction. We will begin from $\hat{u}_{k+1}^{(0)} = \hat{u}_k$ in nonlinear term. For internal nodes, from (14), (15) and using the MLS approximation (6), we

have the following linear equations:

$$\begin{aligned} & \left(1 + \Delta t \frac{(B+1)}{2}\right) \sum_{j=1}^N \left(\int_{\Omega_{s_i}} \phi_j d\mathbf{x}\right) \hat{u}_{k+1,j}^{(l+1)} - \frac{D_u \Delta t}{2} \sum_{j=1}^N \left(\int_{\partial\Omega_{s_i}} \frac{\partial \phi_j}{\partial n} ds\right) \hat{u}_{k+1,j}^{(l+1)} \\ & - \frac{\Delta t}{2} \int_{\Omega_{s_i}} (\tilde{u}_{k+1}^{(l)}(\mathbf{x}))^2 \tilde{v}_{k+1}^{(l)}(\mathbf{x}) d\mathbf{x} = \left(1 - \Delta t \frac{(B+1)}{2}\right) \sum_{j=1}^N \left(\int_{\Omega_{s_i}} \phi_j d\mathbf{x}\right) \hat{u}_{k,j} \quad (16) \\ & + \frac{D_u \Delta t}{2} \sum_{j=1}^N \left(\int_{\partial\Omega_{s_i}} \frac{\partial \phi_j}{\partial n} ds\right) \hat{u}_{k,j} + \Delta t \int_{\Omega_{s_i}} \left(\frac{1}{2}(\tilde{u}_k(\mathbf{x}))^2 \tilde{v}_k(\mathbf{x}) + A\right) d\mathbf{x}, \end{aligned}$$

$$\begin{aligned} & \sum_{j=1}^N \left(\int_{\Omega_{s_i}} \phi_j d\mathbf{x}\right) \hat{v}_{k+1,j}^{(l+1)} - \frac{D_v \Delta t}{2} \sum_{j=1}^N \left(\int_{\partial\Omega_{s_i}} \frac{\partial \phi_j}{\partial n} ds\right) \hat{v}_{k+1,j}^{(l+1)} \\ & + \frac{\Delta t}{2} \int_{\Omega_{s_i}} \left((\tilde{u}_{k+1}^{(l)}(\mathbf{x}))^2 \tilde{v}_{k+1}^{(l)}(\mathbf{x})\right) d\mathbf{x} - B \frac{\Delta t}{2} \sum_{j=1}^N \left(\int_{\Omega_{s_i}} \phi_j(\mathbf{x}) d\mathbf{x}\right) \hat{u}_{k+1,j}^{(l+1)} \quad (17) \\ & = \sum_{j=1}^N \left(\int_{\Omega_{s_i}} \phi_j d\mathbf{x}\right) \hat{v}_{k,j} + \frac{D_v \Delta t}{2} \sum_{j=1}^N \left(\int_{\partial\Omega_{s_i}} \frac{\partial \phi_j}{\partial n} ds\right) \hat{v}_{k,j} \\ & - \frac{\Delta t}{2} \int_{\Omega_{s_i}} \left((\tilde{u}_k(\mathbf{x}))^2 \tilde{v}_k(\mathbf{x}) - B \tilde{u}_k(\mathbf{x})\right) d\mathbf{x}, \end{aligned}$$

where

$$\begin{aligned} \tilde{u}_k(\mathbf{x}) &= \sum_{j=1}^N \phi_j(\mathbf{x}) \hat{u}_{k,j} = \sum_{j=1}^N \phi_j(\mathbf{x}) \hat{u}_{k,j}^{(f)}, \quad \tilde{v}_k(\mathbf{x}) = \sum_{j=1}^N \phi_j(\mathbf{x}) \hat{v}_{k,j} = \sum_{j=1}^N \phi_j(\mathbf{x}) \hat{v}_{k,j}^{(f)}, \\ \tilde{u}_{k+1}^{(l)} &= \sum_{j=1}^N \phi_j(\mathbf{x}) \hat{u}_{k+1,j}^{(l)}, \quad \tilde{v}_{k+1}^{(l)} = \sum_{j=1}^N \phi_j(\mathbf{x}) \hat{v}_{k+1,j}^{(l)}. \end{aligned}$$

The Dirichlet boundary conditions are imposed directly using the equations of the boundary conditions and the MLS approximation (6). To impose the Neumann’s boundary conditions, we adopt a method proposed in Abbasbandy and Shirzadi (2011).

6 Computational techniques

The MLS approximation with Gaussian weight function and quadratic basis ($m = 6$) are used in all test problems. We have to deal with two kinds of integrals in the discretized local integral equations (16) and (17). Basically, both the integrations over the circular sub-domain Ω_{s_i} and/or its boundary $\partial\Omega_{s_i}$ can be performed analytically Sladek and Sladek (2010); Sladek, Sladek, and Zhang (2010) with saving the

computational time substantially because of eliminating the evaluation of the shape functions and/or its derivatives at the integration points. Then, we could write

$$\begin{aligned} \int_{\Omega_{s^i}} g(\mathbf{x})d\mathbf{x} &= \pi r_0^2 g(\mathbf{x}^i) + \frac{\pi}{8} r_0^4 g_{,kk}(\mathbf{x}^i) + O(r_0^6), \\ \int_{\partial\Omega_{s^i}} g(\mathbf{x})ds &= 2\pi r_0 g(\mathbf{x}^i) + \frac{\pi}{2} r_0^3 g_{,kk}(\mathbf{x}^i) + O(r_0^5), \\ \int_{\partial\Omega_{s^i}} \frac{\partial g}{\partial n}(\mathbf{x})ds &= \pi r_0^2 g_{,kk}(\mathbf{x}^i) + O(r_0^4). \end{aligned}$$

In order to avoid the inaccuracy of the second and higher-order derivatives of the shape functions in meshless approximations, we prefer a numerical integration over $\partial\Omega_{s^i}$. Since the number of the integration points in the domain integrals is much higher than in the contour integrals, the analytical integration in case of domain integrals yields more substantial savings in the computational time. Therefore we apply the analytical integrations over Ω_{s^i} with omitting the terms $O(r_0^4)$ involving the derivatives of the shape functions. Thus summarizing, the required integrals over $\partial\Omega_{s^i}$ are computed by using the regular Gauss-Legendre quadrature rule with seven integration points as

$$\begin{aligned} &\int_{\partial\Omega_{s^i}} \phi_j(\mathbf{x})ds \\ &= \int_0^{2\pi} \phi_j(x^i + r_0 \cos(\theta), y^i + r_0 \sin(\theta)) r_0 d\theta \\ &= \pi r_0 \int_{-1}^1 \phi_j(x^i + r_0 \cos(\pi\theta + \pi), y^i + r_0 \sin(\pi\theta + \pi)) d\theta \\ &\approx \pi r_0 \sum_{p=1}^7 w_p \phi_j(x^i + r_0 \cos(\pi\theta_p + \pi), y^i + r_0 \sin(\pi\theta_p + \pi)) \end{aligned}$$

where w_p and θ_p are the Gauss quadrature integration rule weights and points on $[-1, 1]$. The same technique can be used for evaluation of contour integrals of the normal derivatives of the shape functions $n_k(\mathbf{x})\phi_{i,k}(\mathbf{x})$ with utilizing Eq. (7) for the gradients of the shape functions. Finally, the required integrals over sub-domains Ω_{s^i} are approximated as follows:

$$\int_{\Omega_{s^i}} \phi_j(\mathbf{x})d\mathbf{x} \approx \pi \phi_j(\mathbf{x}^i) r_0^2.$$

and similarly, the domain integrals involving the nonlinear terms are approximated as:

$$\begin{aligned} \int_{\Omega_{s,i}} (\tilde{u}_{k+1}^{(l)})^2 \tilde{v}_{k+1}^{(l)} d\mathbf{x} &\approx \pi (\tilde{u}_{k+1}^{(l)}(\mathbf{x}^i))^2 \tilde{v}_{k+1}^{(l)}(\mathbf{x}^i) r_0^2, \\ \int_{\Omega_{s,i}} \left(\frac{1}{2} (\tilde{u}_k)^2 \tilde{v}_k + A \right) d\mathbf{x} &\approx \pi \left(\frac{1}{2} (\tilde{u}_k(\mathbf{x}^i))^2 \tilde{v}_k(\mathbf{x}^i) + A \right) r_0^2, \\ \int_{\Omega_{s,i}} \left((\tilde{u}_{k+1}^{(l)}(\mathbf{x}))^2 \tilde{v}_{k+1}^{(l)}(\mathbf{x}) \right) d\mathbf{x} - B \sum_{j=1}^N \left(\int_{\Omega_{s,i}} \phi_j(\mathbf{x}) d\mathbf{x} \right) \hat{u}_{k+1,j}^{(l+1)} \\ &\approx \pi \left((\tilde{u}_{k+1}^{(l)}(\mathbf{x}^i))^2 \tilde{v}_{k+1}^{(l)}(\mathbf{x}^i) - B \sum_{j=1}^N \phi_j(\mathbf{x}^i) \hat{u}_{k+1,j}^{(l+1)} \right) r_0^2 \\ \int_{\Omega_{s,i}} \left((\tilde{u}_k)^2 \tilde{v}_k - B \tilde{u}_k \right) d\mathbf{x} &\approx \pi \left((\tilde{u}_k(\mathbf{x}^i))^2 \tilde{v}_k(\mathbf{x}^i) - B \tilde{u}_k(\mathbf{x}^i) \right) r_0^2. \end{aligned}$$

Recall that the assumed approximations for the domain integrals are meaningful provided that the radius of the sub-domains is sufficiently small.

7 Test problems

The domain and boundary integrals are evaluated as demonstrated in the previous section. Because of the computational techniques described in the previous section, the radius of each local sub-domains, r_0 , should be small enough. A very small r_0 also causes much cancelation error and therefore, it is chosen as $0.05h \leq r_0 \leq 0.1h$. The shape parameter of the Gaussian weight function is chosen as $c_i \approx 0.6h$, and the radius of the support of the weight function corresponding to node i , r_i , is chosen as $r_i \approx 7h$ where h is the minimum distance between nodes. For the predictor corrector scheme, 3 or 4 corrections are calculated in each time step.

7.1 Example 1.

For the first test problem consider the reaction diffusion Brusselator System (1) with $D_u = D_v = \frac{1}{4}$, $B = 1$ and $A = 0$ in the region $\Omega = \{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$ and the boundary condition extracted from the exact solution given by:

$$\begin{aligned} u(x, y, t) &= \exp\left(-\frac{t}{2} - x - y\right), \\ v(x, y, t) &= \exp\left(\frac{t}{2} + x + y\right). \end{aligned}$$

In this example, the relative error which will be reported is defined as:

$$\| e_u \|_R = \sqrt{\frac{\sum_{i=1}^N (u_i - \bar{u}_i)^2}{\sum_{i=1}^N (u_i)^2}},$$

and the infinity-norm are defined as:

$$\| e_u \|_\infty = \text{Max}\{| u_i - \bar{u}_i |, \quad i = 1, 2, \dots, N\},$$

u_i and \bar{u}_i are the exact and approximate value of u at point \mathbf{x}_i , respectively, and N is the number of nodes. In order to show the convergence of the proposed method with respect to the spatial variables, Table 1 presents the results obtained with different number of nodal points and fixed time step $\Delta t = 0.01$ at time instant $t = 2$. By going through each column of Table 1, one can see increasing accuracy with increasing the number of nodal points.

Table 1: The results obtained with different number of nodal points, $\Delta t = 0.01$ at $t = 2$.

N	$\ e_u \ _\infty$	$\ e_v \ _\infty$	$\ e_u \ _R$	$\ e_v \ _R$
64	1.276869×10^{-4}	1.285399×10^{-3}	3.895784×10^{-4}	4.415529×10^{-5}
100	1.235532×10^{-4}	7.079024×10^{-4}	3.870789×10^{-4}	2.481163×10^{-5}
256	1.206197×10^{-4}	1.950160×10^{-4}	3.838014×10^{-4}	6.842202×10^{-6}
400	1.20025×10^{-4}	1.265732×10^{-4}	3.784170×10^{-4}	3.863627×10^{-6}

The results obtained at time instant $t = 4$ with using $N = 441$ nodal points and different size of time step Δt are presented in Table 2. By going through each column of Table 2, one can see increasing accuracy with decreasing the size of the time step Δt . Table 3 presents the results obtained at different time instants with

Table 2: Results obtained at $t = 4$ with using $N = 441$ nodal points and various time steps Δt

Δt	$\ e_u \ _\infty$	$\ e_v \ _\infty$	$\ e_u \ _R$	$\ e_v \ _R$
0.1	4.445481×10^{-4}	5.516135×10^{-4}	3.901414×10^{-3}	1.2192290×10^{-5}
0.05	2.212092×10^{-4}	3.358346×10^{-4}	1.94276×10^{-3}	4.8399237×10^{-6}
0.01	4.414410×10^{-5}	3.127980×10^{-4}	3.884246×10^{-4}	3.9160857×10^{-6}
0.005	2.211150×10^{-5}	3.122602×10^{-4}	1.949392×10^{-4}	3.897547×10^{-6}
0.001	4.548992×10^{-6}	3.122046×10^{-4}	4.0414576×10^{-5}	3.887129×10^{-6}

using $N = 441$ nodal points and the time step $\Delta t = 0.01$. By going through each column of Table 3, we can see increasing accuracy of u and decreasing accuracy of v . By the finite differences applied to discretize the time variable, the behavior of v is natural. The reason of increasing the accuracy of u by increasing time variable t is cancelation error; by increasing t , $u \rightarrow 0$ and very small values of u causes cancelation error.

Table 3: Results obtained with using time step $\Delta t = 0.01$, $N = 441$ nodal points at different time instants

t	$\ e_u\ _\infty$	$\ e_v\ _\infty$	$\ e_u\ _R$	$\ e_v\ _R$
2	1.198174×10^{-4}	1.139365×10^{-4}	3.879027×10^{-4}	4.265328×10^{-6}
4	4.414410×10^{-5}	3.12798×10^{-4}	3.884246×10^{-4}	3.916085×10^{-6}
6	1.624269×10^{-5}	8.51417×10^{-4}	3.884891×10^{-4}	3.902435×10^{-6}
8	5.975504×10^{-6}	2.314813×10^{-3}	3.884978×10^{-4}	3.901254×10^{-6}
10	2.198272×10^{-6}	6.292470×10^{-3}	3.884990×10^{-4}	3.901106×10^{-6}

Error profiles of u and v at $t=4$ obtained by using $N = 441$ number of nodal points and $\Delta t = 0.01$ are presented in Figures 1 and 2.

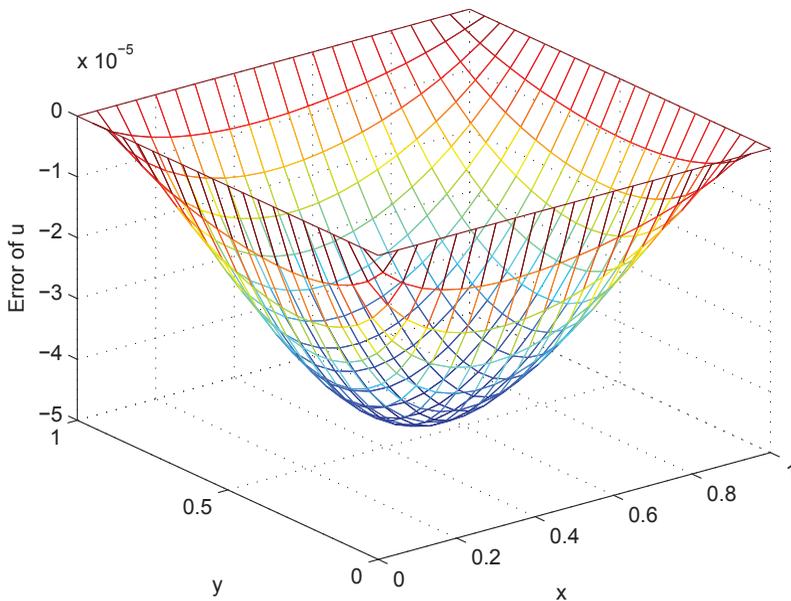


Figure 1: Error profiles of u with $N = 441$ and $\Delta t = 0.01$ at $t = 4$.

7.2 Example 2.

For the second test problem, we consider the reaction diffusion Brusselator System (1) with $D_u = D_v = 0.002$, $B = \frac{1}{2}$ and $A = 1$ in the region $\Omega = \{(x,y) : 0 \leq x \leq$

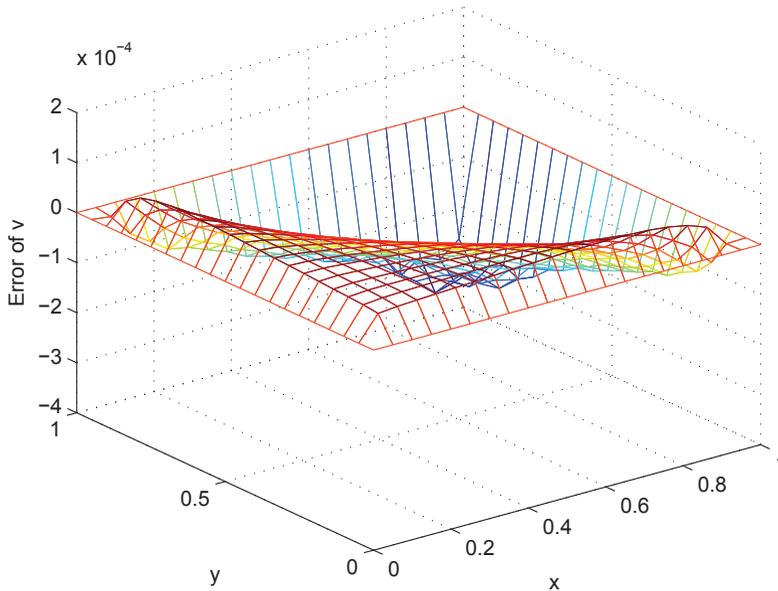


Figure 2: Error profiles of v with $N = 441$ and $\Delta t = 0.01$ at $t = 4$

1, $0 \leq y \leq 1$ } with the initial and Neumann’s boundary conditions given by:

$$u(x, y, 0) = \frac{1}{2}x^2 - \frac{1}{3}x^3,$$

$$v(x, y, 0) = \frac{1}{2}y^2 - \frac{1}{3}y^3,$$

$$\frac{\partial u(x, y, t)}{\partial x} \Big|_{x=0} = \frac{\partial u(x, y, t)}{\partial x} \Big|_{x=1} = \frac{\partial u(x, y, t)}{\partial y} \Big|_{y=0} = \frac{\partial u(x, y, t)}{\partial y} \Big|_{y=1} = 0,$$

$$\frac{\partial v(x, y, t)}{\partial x} \Big|_{x=0} = \frac{\partial v(x, y, t)}{\partial x} \Big|_{x=1} = \frac{\partial v(x, y, t)}{\partial y} \Big|_{y=0} = \frac{\partial v(x, y, t)}{\partial y} \Big|_{y=1} = 0,$$

for which the exact solution is unknown. For small values of the diffusion coefficients D_u and D_v , if $1 - B + A^2 > 0$ then the steady state solution of the Brusselator system (2) converges to equilibrium point $(A, B/A)$. This is valid for Example 2 and is confirmed by the numerical solutions. Here, the relative error which will be reported is defined as:

$$\| e_u \|_R = \sqrt{\frac{\sum_{i=1}^{N_1} (\bar{u}_i^{N_2} - \bar{u}_i^{N_1})^2}{\sum_{i=1}^{N_1} (\bar{u}_i^{N_2})^2}},$$

and the infinity-norm are defined as:

$$\|e_u\|_\infty = \text{Max}\{|\bar{u}_i^{N_2} - \bar{u}_i^{N_1}|, \quad i = 1, 2, \dots, N_1\},$$

where \bar{u}_i^N is the approximation of u at node \mathbf{x}^i obtained with using N nodal points and $\{\mathbf{x}^i, i = 1, 2, \dots, N_1\} \subset \{\mathbf{x}^i, i = 1, 2, \dots, N_2\}$. The numerical solution obtained at some points with using $N = 441$ nodal points and $\Delta t = 0.01$ at different time instants are presented in Table 4. This table confirms that $(u, v) \rightarrow (A, B/A)$ when $t \rightarrow \infty$. The results obtained at time instant $t = 4$ with using $N_2 = 441$ and $N_1 = 121$

Table 4: The solution obtained at some points.

	(0.2,0.2)		(0.4,0.6)		(0.5,0.5)		(0.8,0.9)	
t	u	v	u	v	u	v	u	v
1	0.5532	0.2341	0.5549	0.2415	0.55529	0.24012	0.55747	0.24694
2	0.7258	0.4193	0.7260	0.4197	0.72603	0.41968	0.72623	0.42006
3	0.8419	0.5171	0.8420	0.5171	0.84200	0.51712	0.84202	0.51712
4	0.9302	0.5399	0.9302	0.5399	0.93023	0.53993	0.93023	0.53992
5	0.9830	0.5271	0.9830	0.5271	0.98309	0.52719	0.98308	0.52719
6	1.0037	0.5109	1.0037	0.5109	1.00373	0.51096	1.00372	0.51095
7	1.0066	0.5019	1.0066	0.5019	1.00665	0.50198	1.00665	0.50197
8	1.0040	0.4990	1.0040	0.4990	1.00409	0.49903	1.00409	0.49903
9	1.0015	0.4988	1.0015	0.4988	1.00156	0.49884	1.00155	0.49884
10	1.0002	0.4993	1.0002	0.4993	1.00025	0.49935	1.00024	0.49935
↓	↓	↓	↓	↓	↓	↓	↓	↓
∞	1	0.5	1	0.5	1	0.5	1	0.5

nodal points and different time step Δt are presented in Table 5. From this table, it can be seen that by decreasing the size of the time step Δt , the accuracy of the numerical solution increases.

Table 5: Results obtained at $t = 4$, with different time step Δt , $N_2 = 441$ and $N_1 = 121$ points.

Δt	$\ e_u\ _\infty$	$\ e_v\ _\infty$	$\ e_u\ _R$	$\ e_v\ _R$
0.1	2.903307×10^{-6}	9.11515×10^{-7}	1.788346×10^{-6}	6.068615×10^{-7}
0.05	2.7798×10^{-6}	8.73329×10^{-7}	1.756504×10^{-6}	5.5350×10^{-7}
0.01	2.772280×10^{-6}	8.433490×10^{-7}	1.733112×10^{-6}	5.227214×10^{-7}
0.005	2.771383×10^{-6}	8.396910×10^{-7}	1.730165×10^{-6}	5.190052×10^{-7}
0.001	2.770672×10^{-6}	8.367800×10^{-7}	1.727803×10^{-6}	5.160543×10^{-7}

Initial profiles of u and v are presented in Figures 3 and 4. Plots of the numerical solutions obtained for u and v at $t = 10$ by using $N = 441$ nodal points and $\Delta t =$

0.01 are presented in Figures 5 and 6. Figures 7 and 8 presents the numerical solution at point $(0.5,0.5)$ versus time t . From these figures we can see that at every point of the domain the solution (u,v) tends to $(A,B/A)$.

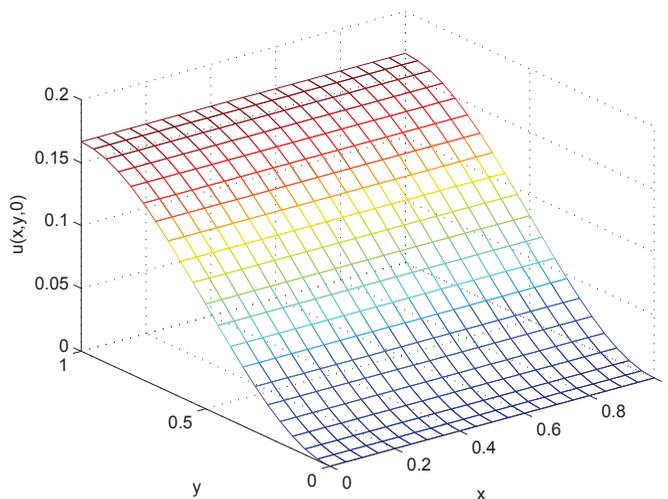


Figure 3: Initial concentration profile of u .

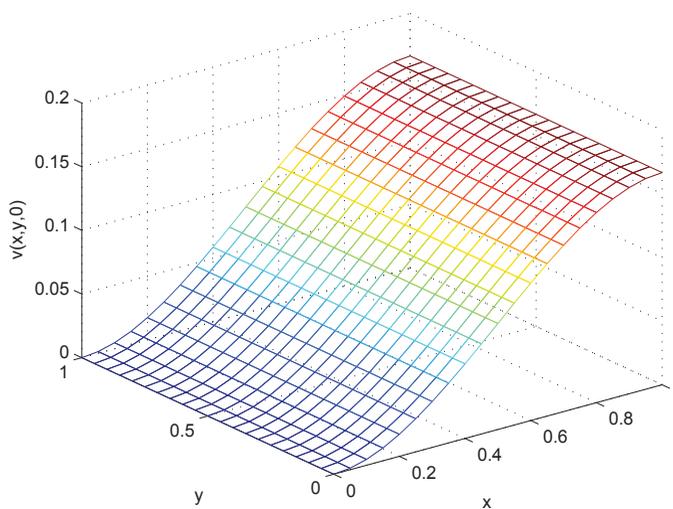


Figure 4: Initial concentration profile of v .

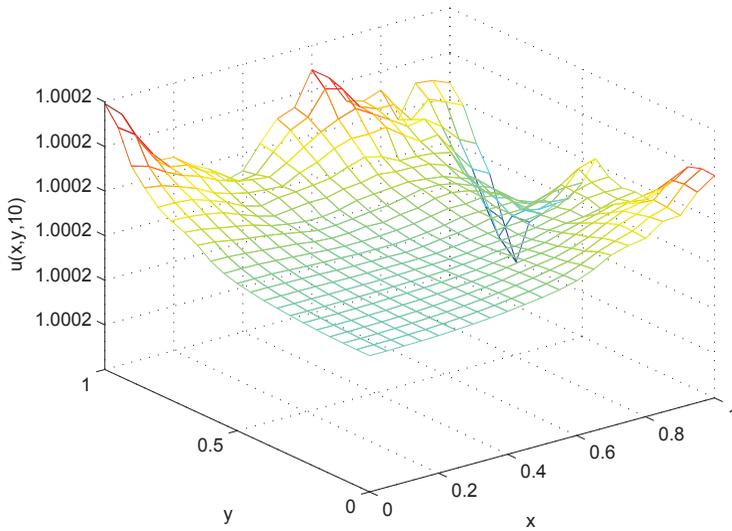


Figure 5: The solution u at $t = 10$, $N = 441$, $\Delta t = 0.01$.

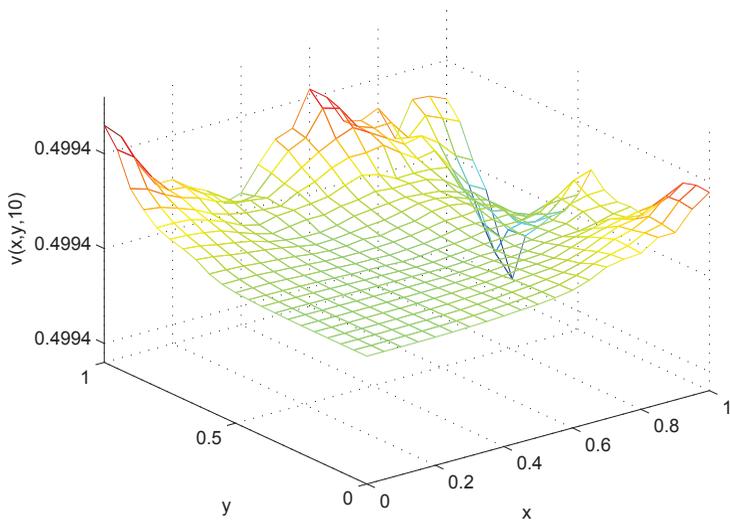


Figure 6: The solution v solutions at $t = 10$, $N = 441$, $\Delta t = 0.01$.

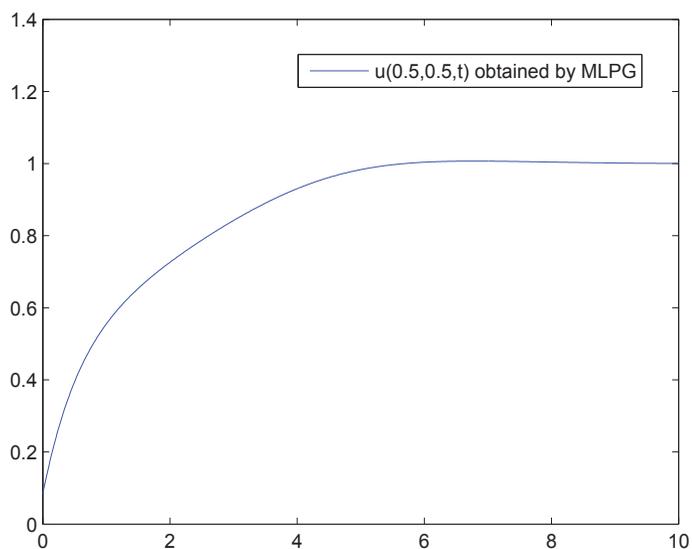


Figure 7: Plot of $u(0.5, 0.5, t)$ versus time t with $N = 441$, $\Delta t = 0.01$.

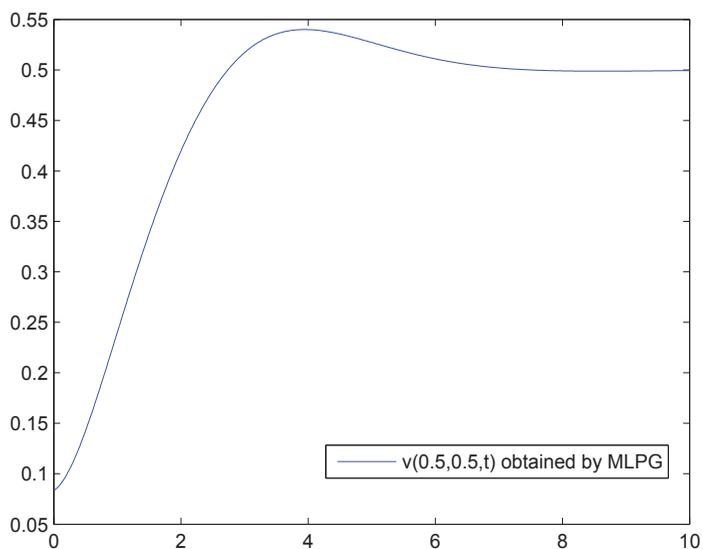
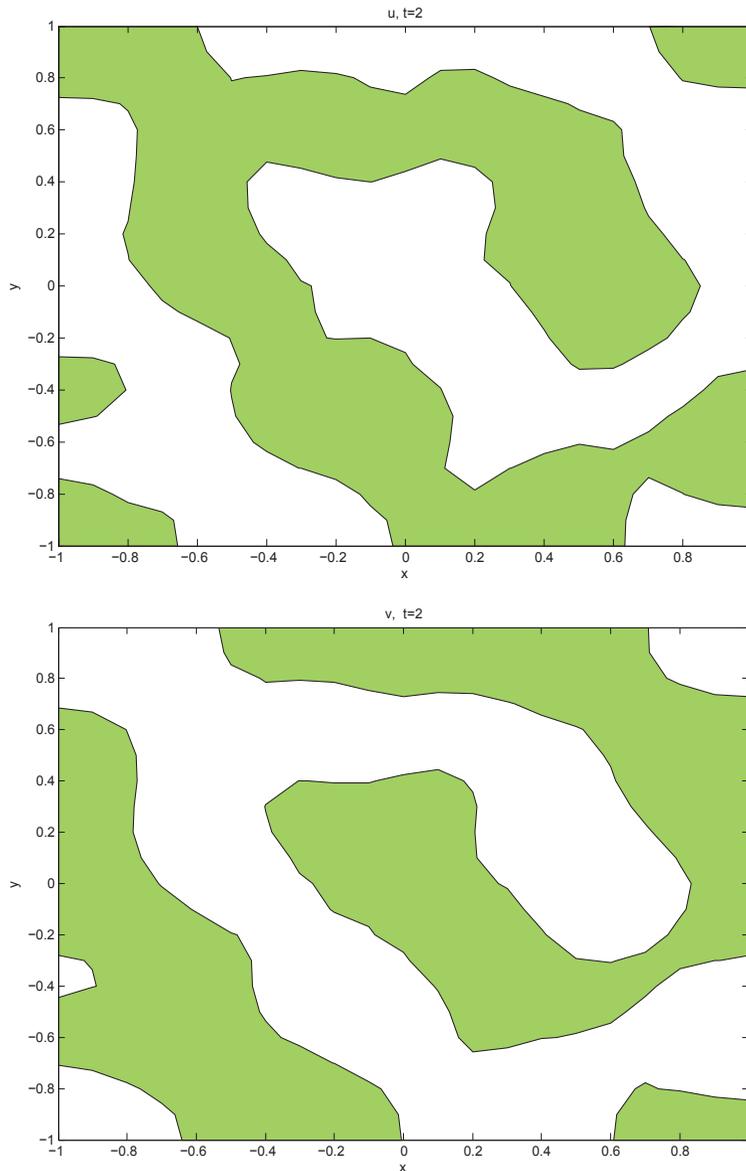


Figure 8: Plot of $v(0.5, 0.5, t)$ versus time t with $N = 441$, $\Delta t = 0.01$.

7.3 Example 3. Pattern formation in Brusselator system

Consider the Brusselator equation (1) with no flux boundary conditions. The threshold for the Turing instability is $B_c^T = [1 + A\sqrt{D_u/D_v}]^2$. The wave number cor-

Figure 9: u and v at $t=2$

responding to the most unstable Turing mode is given by $k_c^2 = A/\sqrt{D_u D_v}$. The corresponding characteristic length of the pattern is defined by $\lambda_c = \frac{2\pi}{k_c}$. This test problem considers (1) with $A = 4.5$ and $\frac{D_v}{D_u} = 8$ resulting in $B_c^T \approx 6.71$. The pa-

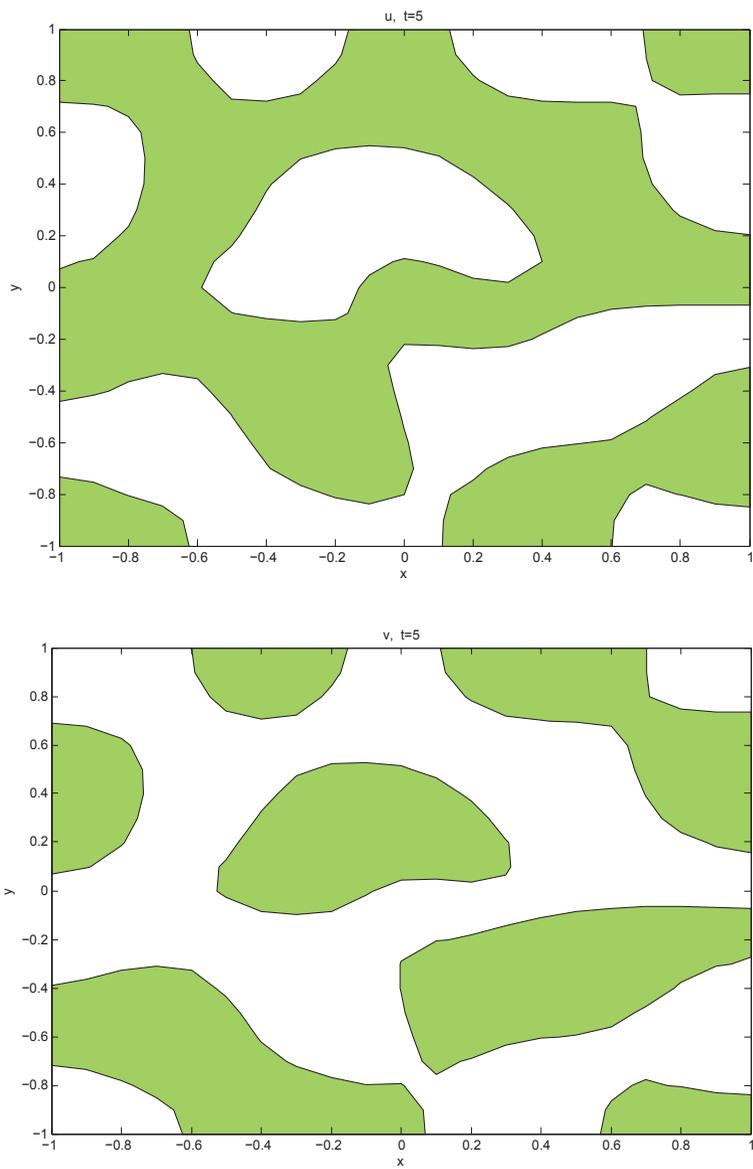
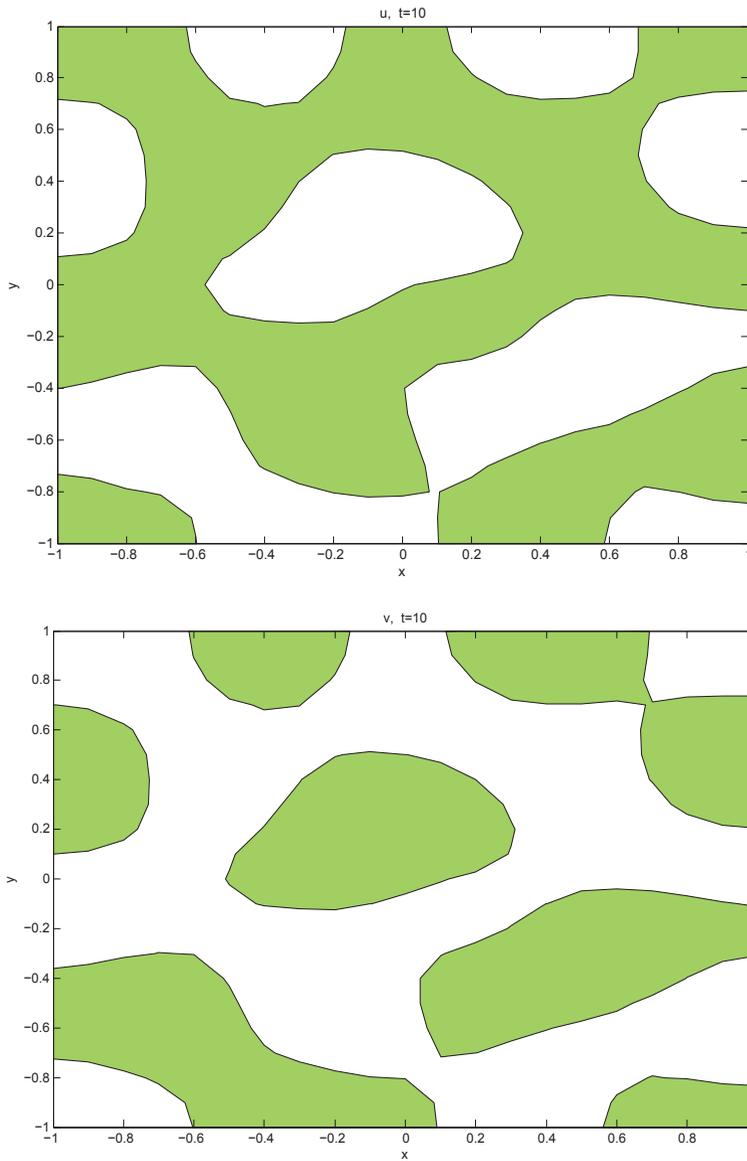


Figure 10: u and v at $t=5$

parameter B is chosen as $B = 6.75$ resulting in spot patterns as shown in Leppänen (2004). The domain of the problem is chosen as $[-1, 1]^2$ and the diffusion coefficients are fixed with $D_v = 0.2$ $D_u = 0.025$ resulting in $\lambda_c \approx 0.79$. The initial

Figure 11: u and v at $t=10$

configuration was random perturbations around the uniform stationary state. Figures 9-12 presents the process of the pattern formation for the considered model. In all contour graphs, coloration is determined by a constant threshold value, u_s for

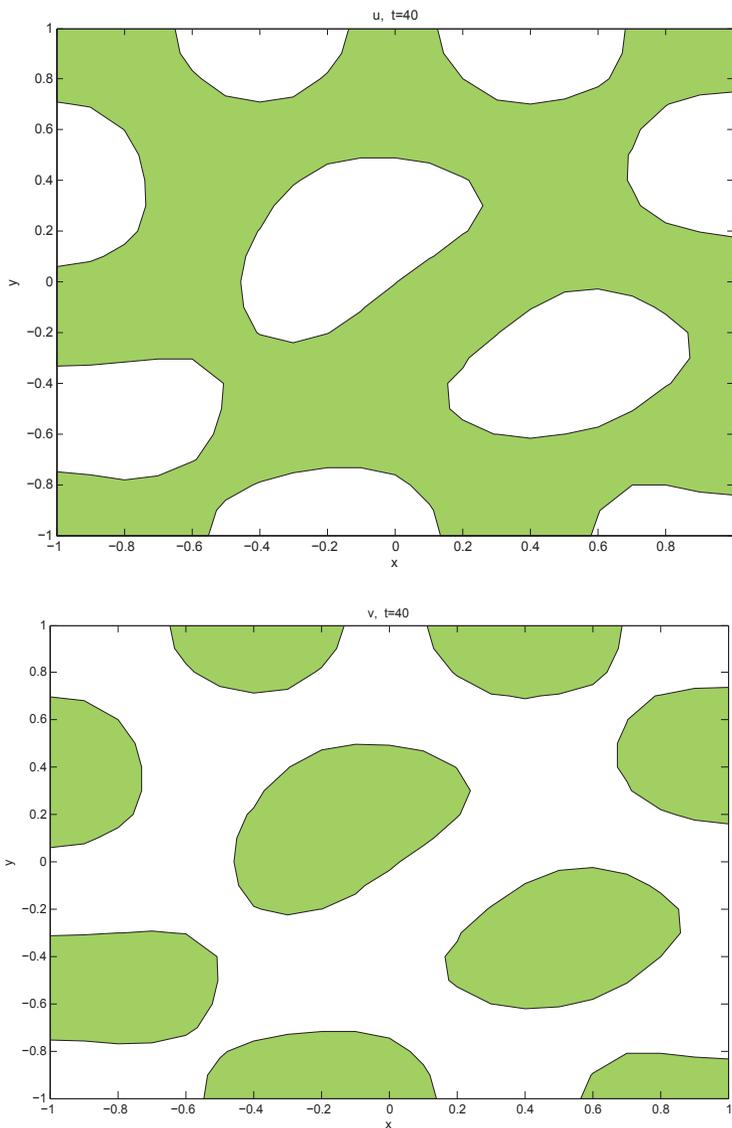


Figure 12: u and v at $t=40$

u and v_s for v , such that in the regions with white color, $u < u_s$ and $v < v_s$ while the regions which experience the concentration $u > u_s$ and $v > v_s$ are colored with green. The results confirm that the profiles of the function v are always 180° out of phase to those of u .

8 Conclusions

A numerical method for solving the two dimensional nonlinear reaction-diffusion Brusselator system was presented. The time variable was discretized by using one-step θ method. Then the resultant elliptic type PDEs were considered in weak forms on the local subdomains and the MLS approximation was used to approximate the field variables. The Heaviside step function was used as test function in each local subdomain. The nonlinear terms were treated by using the predictor corrector scheme. Test problems reveal that the method is of high accuracy and stability. A model system exhibiting Turing instability was given as a test problem and the resulting patterns were presented.

Acknowledgements

The financial supports by APVV-0032-10 (Slovak Research and Development Agency) is greatly acknowledged.

References

- Abbasbandy, S.; Shirzadi, A.** (2011): MLPG method for two-dimensional diffusion equation with Neumann's and non-classical boundary conditions. *Appl. Numer. Math.*, vol. 61, pp. 170–180.
- Abbasbandy, S.; Sladek, V.; Shirzadi, A.; Sladek, J.** (2011): Numerical simulations for coupled pair of diffusion equations by MLPG method. *CMES Compt. Model. Eng. Sci.*, vol. 71, no. 1, pp. 15–37.
- Adomian, G.** (1995): The diffusion-brusselator equation. *Computers and Mathematics with Applications*, vol. 29, no. 5, 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, no. 9, pp. 897–903.
- Atluri, S.** (2004): *The meshless method (MLPG) for domain and BIE discretizations*. Tech Science Press.
- Atluri, S.; Kim, H.-G.; Cho, J. Y.** (1999): A critical assessment of the truly meshless local Petrov-Galerkin (MLPG) and local boundary integral equation (LBIE) methods. *Comput. Mech.*, vol. 24, pp. 348–372.
- Atluri, S.; Shen, S.** (2002): *The Meshless Local Petrov-Galerkin (MLPG) Method*. Tech Science Press.
- Atluri, S.; Zhu, T.** (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comput. Mech.*, vol. 22, no. 2, pp. 117–127.

- Atluri, S.; Zhu, T.** (1998): A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. *Comput. Modeling Simulation in Engrg.*, vol. 3, pp. 187–196.
- Borckmans, P.; Wit, A. D.; Dewel, G.** (1992): Competition in ramped turing structures. *Physica A Statistical Mechanics and its Applications*, vol. 188, no. 1-3, pp. 137–157.
- Kamranian, M.; Tatari, M.; Dehghan, M.** (2011): The finite point method for reaction-diffusion systems in developmental biology. *CMES Compt. Model. Eng. Sci.*, vol. 82, no. 1, pp. 1–27.
- Leppänen, T.** (2004): *Computational Studies of Pattern Formation in Turing Systems*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 2004.
- Mittal, R.; Jiwari, R.** (2011): Numerical solution of two-dimensional reaction-diffusion brusselator system. *Appl. Math. Comp.*, vol. 217, no. 12, pp. 5404–5415.
- Nicolis, G.; Prigogine, I.** (1997): *Self-organization in nonequilibrium systems: from dissipative structures to order through fluctuations*. John Wiley & Sons.
- Prigogine, I.; Lefever, R.** (1968): Symmetry breaking instabilities in dissipative systems.II. *J. Chem. Phys.*, vol. 48, pp. 1695–1700.
- Shirzadi, A.; Ling, L.** (2013): Convergent overdetermined-RBF-MLPG for solving second order elliptic PDEs. *Adv. Appl. Math. Mech.*, vol. 5, pp. 78–89.
- Siraj-ul, I.; Arshed, A.; Sirajul, H.** (2010): A computational modeling of the behavior of the two- dimensional reaction-diffusion brusselator system. *Appl. Math. Model.*, vol. 34, no. 12, pp. 3896–3909.
- Sladek, J.; Stanak, P.; Han, Z. D.; Sladek, V.; Atluri, S. N.** (2013): Applications of the MLPG method in engineering & sciences: A review. *CMES Compt. Model. Eng. Sci.*, vol. 92, no. 5, pp. 423–475.
- Sladek, V.; Sladek, J.** (2010): Local integral equations implemented by MLS-approximation and analytical integrations. *Eng. Anal. Bound. Elem.*, vol. 34, no. 11, pp. 904–913.
- Sladek, V.; Sladek, J.; Zhang, C.** (2010): On increasing computational efficiency of local integral equation method combined with meshless implementations. *CMES Compt. Model. Eng. Sci.*, vol. 63, no. 3, pp. 243–263.
- Turing, A.** (1952): The chemical basis of morphogenesis. *Phil. Trans. R. Soc. Lond. B*, vol. 237, no. 641, pp. 37–72.
- Twizell, E. H.; Gumel, A. B.; Cao, Q.** (1999): A second-order scheme for the “Brusselator” reaction-diffusion system. *J. Math. Chem.*, vol. 26, no. 4, pp. 297–316 (2000).

Verdasca, J.; Wit, A. D.; Dewel, G.; Borckmans, P. (1992): Reentrant hexagonal turing structures. *Physics Letters A*, vol. 168, no. 3, pp. 194–198.

Walgraef, D.; Dewel, G.; Borckmans, P. (1980): Fluctuations near nonequilibrium phase transitions to nonuniform states. *Phys. Rev. A*, vol. 21, pp. 397–404.