

A computational modeling of two dimensional reaction–diffusion Brusselator system arising in chemical processes

Ram Jiwari · Jinyun Yuan

Received: 30 November 2013 / Accepted: 31 January 2014 / Published online: 17 February 2014
© Springer International Publishing Switzerland 2014

Abstract In this article, the authors proposed a modified cubic B-spline differential quadrature method (MCB-DQM) to show computational modeling of two-dimensional reaction–diffusion Brusselator system with Neumann boundary conditions arising in chemical processes. The system arises in the mathematical modeling of chemical systems such as in enzymatic reactions, and in plasma and laser physics in multiple coupling between modes. The MCB-DQM reduced the Brusselator system into a system of nonlinear ordinary differential equations. The obtained system of nonlinear ordinary differential equations is then solved by a four-stage RK4 scheme. Accuracy and efficiency of the proposed method successfully tested on four numerical examples and obtained results satisfy the well known result that for small values of diffusion coefficient, the steady state solution converges to equilibrium point $(B, A/B)$ if $1 - A + B^2 > 0$.

Keywords Two-dimensional reaction–diffusion Brusselator system · Cubic B-spline functions · Modified cubic B-spline differential quadrature method · System of ordinary differential equations · Runge–Kutta 4th order method

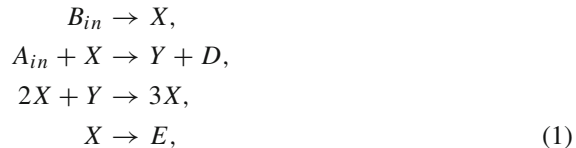
The work was partially supported by the CAPES and CNPq, Brazil.

R. Jiwari · J. Yuan
Departamento de Matemática, Universidade Federal do Paraná, Centro Politécnico,
Curitiba CP 19.081, 81531-980, Brazil

R. Jiwari (✉)
School of Mathematics & Computer Applications, Thapar University, Patiala, India
e-mail: ram1maths@gmail.com

1 Introduction

Reaction-diffusion models frequently arise in the study of chemical and biological systems which are reaction–diffusion equations. The importance of oscillations in biochemical systems has been emphasized by a number of authors' e.g. Turing [1] showed that when certain reactions are coupled with the process of diffusion, it is possible to obtain a stable spatial pattern (which leads to the theory of morphogenesis). The so-called Brussels school [2–5] developed and analysed the behaviour of a non-linear oscillator associated with the chemical system



where B_{in} and A_{in} are input chemicals, D and E are output chemicals and X and Y are intermediates. The chemical system (1), known as the “Brusselator” system, is important in that it admits limit-cycle oscillations and yet contains only two dependent variables (X and Y) thus enabling the use of two-dimensional mathematical systems [2]. It is known [7] that the trimolecular reaction step [third term in Eq. (1)] arises in the formation of ozone by atomic oxygen via a triple collision, in enzymatic reactions, and in plasma and laser physics in multiple couplings between modes. For more detail see [6].

Finally, the non-linear system of partial differential equations associated with two-dimensional reaction–diffusion Brusselator system is given by [7]

$$\frac{\partial u}{\partial t} = B + u^2v - (A + 1)u + \alpha \nabla^2 u, \quad (2a)$$

$$\frac{\partial v}{\partial t} = Au - u^2v + \alpha \nabla^2 v \quad (x, y, t) \in \Omega \times (0, T] \quad (2b)$$

together with initial conditions

$$(u(x, y, 0), v(x, y, 0)) = (f_1(x, y), f_2(x, y)) \quad (3)$$

and Neumann boundary conditions

$$\frac{\partial u}{\partial n} = 0, \quad \frac{\partial v}{\partial n} = 0, \quad (x, y, t) \in \partial\Omega \times (0, T] \quad (4)$$

where $u(x, y, t)$, $v(x, y, t)$ represent dimensionless concentrations of two reactants, A and B are constants concentrations of the two reactants, α is diffusion coefficient, ∇^2 is Laplace operator and $f_1(x, y)$, $f_2(x, y)$ are suitably prescribed functions. It is well known that for small values of the diffusion coefficient α , the steady state solution of the Brusselator system (2) converges to equilibrium point $(B, A/B)$ if $1 - A + B^2 > 0$ (see [6]).

The reaction–diffusion Brusselator system contains a pair of variables intermediates with reactant and product chemicals whose concentrations are controlled. The system (2a) and (2b) represents a useful model for study of co-operative processes in chemical kinetics. Such Brusselator system arises in the formation of ozone by atomic oxygen via a triple collision. It also arises in the modeling of certain chemicals reaction–diffusion processes such as in enzymatic reactions, and in plasma and laser physics in multiple coupling between modes. The analytical solution of the system is not yet known and therefore it is interested from numerical point of view.

In recent years, much attention has been paid in literature in the development of numerical schemes for the numerical solutions of reaction–diffusion Brusselator system such as second order finite-difference scheme [6], decomposition method [7,8], dual-reciprocity boundary element method [9], Runge–Kutta–Chebyshev method [10], collocation method using the radial basis functions [11], differential quadrature method [12], homotopy perturbation method [13] etc.

In this article, a different technique based on modified cubic-B-spline functions is proposed to find the weighting coefficients of differential quadrature method than the traditional technique of Lagrange interpolation [14]. Then, the modified cubic-B-spline differential quadrature method (MCB-DQM) is applied to solve the two-dimensional reaction–diffusion Brusselator system. The MCB-DQM reduced the system into a system of nonlinear ordinary differential equations. Finally, the obtained system of nonlinear ordinary differential equations is then solved by a four-stage RK4 scheme given by Pike and Roe [15]. In order to demonstrate the accuracy and efficiency of the proposed method, some test examples have chosen from literature.

2 Description of modified cubic B-spline differential quadrature method (MCB-DQM)

Differential quadrature method [16] is a numerical technique for solving differential equations. By this method, we approximate the spatial derivatives of unknown function at any grid points using weighted sum of all the functional values at certain points in whole computational domain. In two dimensional DQM, the first step is to discretize the domain $D = \{(x, y) : a \leq x \leq b; c \leq y \leq d\}$ as $D^1 = \{(x_i, y_j), i = 1, 2, \dots, N; j = 1, 2, \dots, M\}$ by taking step length $\Delta x = x_i - x_{i-1}$ in x -axis direction and $\Delta y = y_j - y_{j-1}$ in y -axis direction. According to this method, the first order partial derivative with respect to x of the dependent function $u(x, y, t)$ (by keeping the point y_j fixed) is approximated at point x_i as follows

$$u_x(x_i, y_j, t) = \sum_{k=1}^N w_{ik}^{(1)} u(x_k, y_j, t), \quad i = 1, 2, \dots, N \quad (5)$$

Similarly, the first order partial derivative with respect to y of the dependent function $u(x, y, t)$ (by keeping the point x_i fixed) can be approximated at point y_j as follows

$$u_y(x_i, y_j, t) = \sum_{k=1}^N \bar{w}_{jk}^{(1)} u(x_i, y_k, t), \quad j = 1, 2, \dots, N \quad (6)$$

where $w_{ij}^{(1)}$ and $\bar{w}_{jk}^{(1)}$ are unknown and represent the weighting coefficients of the first order partial derivatives w. r. t. x and y . There are many approaches to find these weighting coefficients such as Bellman's approaches [16], Quan and Chang's approach [17, 18], and Shu's approach [14]. Shu's approach is very general approach and in recent years most of the differential quadrature methods using various test functions such as Legendre polynomials, Lagrange interpolation polynomials, spline functions, Lagrange interpolated cosine functions, etc. are based on this approach. These days in literature, most frequently used differential quadrature methods are based on Lagrange interpolation polynomials and sine-cosine expansion. Korkmaz and Dağ [19, 20] proposed sinc differential quadrature method and cosine expansion based differential quadrature method for many nonlinear partial differential equations while Mittal et al. [12, 21–23] have used polynomial based differential quadrature method for numerical solutions of some nonlinear partial differential equations. Here, the authors proposed one more approach based on modified cubic B-spline functions to find the weighting coefficients except the above approaches.

2.1 Modified cubic B-spline functions

In this method, modified cubic B-spline functions are used to find the weighting coefficients $w_{ik}^{(1)}$ and $\bar{w}_{jk}^{(1)}$. The cubic B-spline basis functions at the knots are defined as follows

$$B_m(x) = \frac{1}{h^3} \begin{cases} (x - x_{m-2})^3 & x \in [x_{m-2}, x_{m-1}) \\ (x - x_{m-2})^3 - 4(x - x_{m-1})^3 & x \in [x_{m-1}, x_m) \\ (x_{m+2} - x)^3 - 4(x_{m+1} - x)^3 & x \in [x_m, x_{m+1}) \\ (x_{m+2} - x)^3 & x \in [x_{m+1}, x_{m+2}) \\ 0 & m = 0, 1, \dots, N + 1 \end{cases} \quad (7)$$

where $\{B_0(x), B_1(x), \dots, B_N(x), B_{N+1}(x)\}$ forms a basis over the domain Ω . The values of cubic B-splines and its derivatives at the nodal points are tabulated in Table 1. The cubic B-spline basis functions are modified in such way that the resulting matrix system of equations is diagonally dominant. Modified cubic B-spline basis functions at the knots are defined as follows [24]

$$\begin{aligned} \Phi_1(x) &= B_1(x) + 2B_0(x) \\ \Phi_2(x) &= B_2(x) - B_0(x) \\ \Phi_l(x) &= B_j(x), \quad l = 3, 4, \dots, N - 2 \\ \Phi_{N-1}(x) &= B_{N-1}(x) - 2B_{N+1}(x) \\ \Phi_N(x) &= B_N(x) + 2B_{N+1}(x) \end{aligned} \quad (8)$$

In the same way, the functions $\Phi_l(x)$, $l = 1, 2, \dots, N$ forms a basis over the domain Ω .

Table 1 Value of $B_k(x)$ and its first derivatives at the nodal points

	x_{m-2}	x_{m-1}	x_m	x_{m+1}	x_{m+2}
$B_m(x)$	0	1	4	1	0
$B'_m(x)$	0	$\frac{3}{h}$	0	$-\frac{3}{h}$	0
$B''_m(x)$	0	$\frac{6}{h^2}$	$-\frac{12}{h^2}$	$\frac{6}{h^2}$	0

2.2 To determine the weighting coefficients

Since y -axis is fixed in Eq. (5), so to find the weighting coefficients $w_{ik}^{(1)}$, put the functions $\Phi_m(x)$, $m = 1, 2, \dots, N$ in Eq. (5), we have

$$\Phi'_m(x_i) = \sum_{k=1}^N w_{ik}^{(1)} \Phi_m(x_k), \quad i = 1, 2, \dots, N \tag{9}$$

For any arbitrary choice of m , we have the following algebraic system of equations

$$\begin{bmatrix} \Phi_{1,1} & \Phi_{1,2} & & & & & & & & \\ \Phi_{2,1} & \Phi_{2,2} & \Phi_{2,3} & & & & & & & \\ & \Phi_{3,2} & \Phi_{3,3} & \Phi_{3,4} & & & & & & \\ & \dots & \dots & \dots & & & & & & \\ & \dots & & & \Phi_{N-1,N-2} & \Phi_{N-1,N-1} & \Phi_{N-1,N} & & & \\ & & & \dots & & \Phi_{N,N-1} & \Phi_{N,N} & & & \end{bmatrix} \begin{bmatrix} w_{i1}^{(1)} \\ w_{i2}^{(1)} \\ \vdots \\ \vdots \\ w_{iN-1}^{(1)} \\ w_{iN}^{(1)} \end{bmatrix} = \begin{bmatrix} \Phi'_{1,i} \\ \Phi'_{1,i} \\ \vdots \\ \vdots \\ \Phi'_{N-1,i} \\ \Phi'_{N,i} \end{bmatrix} \tag{10}$$

The Eq. (10) are systems of tridiagonal algebraic system of equations for each i , which can solved by well known “Thomas algorithm” and whose solution provides the coefficients weighting coefficients of first order derivative $w_{ik}^{(1)}$. For example, let $i = 1$, we have following tridiagonal system of equations

$$\begin{bmatrix} 6 & 1 & & & & \\ 0 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & & & 1 & 4 & 0 \\ & & & & & 1 & 6 \end{bmatrix} \begin{bmatrix} w_{11}^{(1)} \\ w_{12}^{(1)} \\ \vdots \\ \vdots \\ w_{1N-1}^{(1)} \\ w_{1N}^{(1)} \end{bmatrix} = \begin{bmatrix} -6/h \\ 6/h \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

After solving the system (11) by using “Thomas algorithm”, we get the weighting coefficients $w_{11}^{(1)}, w_{12}^{(1)}, \dots, w_{1N}^{(1)}$. In similar way, we can find the weighting coefficients for $i = 2, 3, \dots, N$. The second order and higher order derivatives can be calculated by following recurrence formulas [14]

$$w_{ij}^{(r)} = r \left[w_{ij}^{(1)} w_{ii}^{(r-1)} - \frac{w_{ij}^{(r-1)}}{x_i - x_j} \right], \text{ for } i \neq j \quad (12)$$

$$i, j = 1, 2, \dots, N; \quad r = 2, 3, \dots, N - 1$$

$$w_{ii}^{(r)} = - \sum_{j=1, j \neq i}^N w_{ij}^{(r)}, \text{ for } i = j \quad (13)$$

where $w_{ij}^{(r-1)}$ and $w_{ij}^{(r)}$ are weighting coefficients of $(r - 1)$ th and (r) th order partial derivatives with respect to x .

In the similar way, we can find out the weighting coefficients $\bar{w}_{jk}^{(1)}$ of first order partial derivatives w. r. t. y by putting modified cubic B-Spline functions in Eq. (6) and second order and higher order derivatives can be calculated by the recurrence formulas

$$\bar{w}_{ij}^{(r)} = r \left[\bar{w}_{ij}^{(1)} \bar{w}_{ii}^{(r-1)} - \frac{\bar{w}_{ij}^{(r-1)}}{x_i - x_j} \right], \text{ for } i \neq j \quad (14)$$

$$i, j = 1, 2, \dots, N; \quad r = 2, 3, \dots, N - 1$$

$$\bar{w}_{ii}^{(r)} = - \sum_{j=1, j \neq i}^N \bar{w}_{ij}^{(r)}, \text{ for } i = j \quad (15)$$

where $\bar{w}_{ij}^{(r-1)}$ and $\bar{w}_{ij}^{(r)}$ are weighting coefficients of $(r - 1)$ th and (r) th order partial derivatives with respect to y .

3 Numerical scheme based on MCB-DQM for reaction–diffusion Brusselator system

Discretize the spatial derivatives by applying the modified cubic B-spline differential quadrature method to the system of Eqs. (2)–(4), we get following system of non linear ordinary differential equations

$$\frac{du_{i,j}}{dt} = B + u_{i,j}^2 v_{i,j} - (A + 1)u_{i,j} + \alpha \left(\sum_{k=1}^N w_{i,k}^{(2)} u_{k,j} + \sum_{k=1}^M \bar{w}_{j,k}^{(2)} u_{i,k} \right) \quad (16)$$

$$\frac{dv_{i,j}}{dt} = Au_{i,j} - u_{i,j}^2 v_{i,j} + \alpha \left(\sum_{k=1}^N w_{i,k}^{(2)} v_{k,j} + \sum_{k=1}^M \bar{w}_{j,k}^{(2)} v_{i,k} \right) \quad (17)$$

with initial conditions

$$u(x_i, y_j, 0) = f_1(x_i, y_j), \quad (x_i, y_j) \in \Omega$$

$$v(x_i, y_j, 0) = f_2(x_i, y_j), \quad (x_i, y_j) \in \Omega \quad (18)$$

where $u_{i,j} = u(x_i, y_j, t)$ and $w_{i,k}^{(2)}, \bar{w}_{j,k}^{(2)}$ are weighting coefficients of second order partial derivatives of $u(x, y, t), v(x, y, t)$ with respect to x and y respectively.

3.1 Implementation of boundary conditions

The Neumann boundary conditions on boundary $\partial\Omega$ given in Eq. (4) can be approximated as

$$\sum_{k=1}^N w_{1,k}^{(1)} u_{k,j} = 0, \tag{19}$$

$$\sum_{k=1}^N w_{N,k}^{(1)} u_{k,j} = 0, \quad j = 1, 2, \dots, M \tag{20}$$

The Eqs. (19) and (20) can be written as follows

$$w_{1,1}^{(1)} u_{1,j} + w_{1,N}^{(1)} u_{N,j} = - \sum_{k=2}^{N-1} w_{1,k}^{(1)} u_{k,j} \tag{21}$$

$$w_{N,1}^{(1)} u_{1,j} + w_{N,N}^{(1)} u_{N,j} = - \sum_{k=2}^{N-1} w_{N,k}^{(1)} u_{k,j} \tag{22}$$

Solving the Eqs. (21)–(22) for $u_{1,j}$ and $u_{N,j}$, we get

$$u_{1,j} = \frac{w_{N,N}^{(1)}(S_1) - w_{1,N}^{(1)}(S_2)}{\left(w_{1,N}^{(1)} w_{N,1}^{(1)} - w_{1,1}^{(1)} w_{N,N}^{(1)} \right)}, \quad j = 1, 2, \dots, M \tag{23}$$

$$u_{N,j} = \frac{w_{1,1}^{(1)}(S_2) - w_{N,1}^{(1)}(S_1)}{\left(w_{1,N}^{(1)} w_{N,1}^{(1)} - w_{1,1}^{(1)} w_{N,N}^{(1)} \right)}, \quad j = 1, 2, \dots, M \tag{24}$$

where $S_1 = \sum_{k=2}^{N-1} w_{1,k}^{(1)} u_{k,j}$, $S_2 = \sum_{k=2}^{N-1} w_{N,k}^{(1)} u_{k,j}$.

Similarly, other Neumann boundary conditions can be approximated as

$$\sum_{k=1}^M \bar{w}_{1,k}^{(1)} u_{i,k} = 0, \tag{25}$$

$$\sum_{k=1}^M \bar{w}_{M,k}^{(1)} u_{i,k} = 0, \quad i = 1, 2, \dots, N \tag{26}$$

Solving the Eqs. (25)–(26) for $u_{i,1}$ and $u_{i,M}$, we get

$$u_{i,1} = \frac{\bar{w}_{M,M}^{(1)}(S_3) - \bar{w}_{1,M}^{(1)}(S_4)}{\left(\bar{w}_{1,M}^{(1)} \bar{w}_{M,1}^{(1)} - \bar{w}_{1,1}^{(1)} \bar{w}_{M,M}^{(1)} \right)}, \quad j = 1, 2, \dots, M \tag{27}$$

$$u_{i,M} = \frac{\bar{w}_{1,1}^{(1)}(S_4) - \bar{w}_{M,1}^{(1)}(S_3)}{\left(\bar{w}_{1,M}^{(1)} \bar{w}_{M,1}^{(1)} - \bar{w}_{1,1}^{(1)} \bar{w}_{M,M}^{(1)} \right)}, \quad j = 1, 2, \dots, M \tag{28}$$

Table 2 Approximated solutions of Example 1 with $A = 0.5, B = 1.0, \alpha = 0.002$ at different time and mesh points

t	(0.2, 0.2)		(0.4, 0.6)		(0.8, 0.9)	
	u	v	u	v	u	v
1.0	1.8100	0.5873	2.3893	0.1968	3.1524	0.1489
2.0	1.4064	0.3414	1.4478	0.2971	1.7250	0.2538
3.0	1.1133	0.3998	1.1094	0.3877	1.2047	0.3536
5.0	0.9846	0.4851	0.9799	0.4858	0.9831	0.4748
7.0	0.9929	0.5028	0.9925	0.5035	0.9894	0.5032
8.0	0.9978	0.5022	0.9979	0.5026	0.9963	0.5031
9.0	0.9995	0.5010	1.0001	0.5011	0.9996	0.5016
10.0	1.0005	0.5002	1.0001	0.5003	1.0005	0.5005
↓	↓	↓	↓	↓	↓	↓
∞	1.0	0.5	1.0	0.5	1.0	0.5

where $S_3 = \sum_{k=2}^{M-1} \bar{w}_{1,k}^{(1)} u_{i,k}, S_4 = \sum_{k=2}^{M-1} \bar{w}_{M,k}^{(1)} u_{i,k}$.

In similar way, we can find out the boundary conditions for the variable $v(x, y, t)$.

The system of nonlinear ordinary differential equations (16)–(17) with the initial conditions (18) and boundary conditions (23), (24), (27) (28) cannot be solved directly by Pike and Roe’s fourth-stage RK4 [15]. So, first apply boundary conditions on the system (16)–(17) then we have a system of nonlinear ordinary differential equations of the form

$$\begin{aligned} \frac{dU}{dt} &= F(U, V) \\ \frac{dV}{dt} &= G(U, V) \end{aligned} \tag{29}$$

with initial conditions

$$U(0) = f_1, \quad V(0) = f_2 \tag{30}$$

where $U = [u_{2,2}, u_{2,3} \dots, u_{2,M}, \dots, u_{N-1,2}, u_{N-1,3}, \dots, u_{N-1,M-1}]^T, V = [v_{2,2}, v_{2,3} \dots, v_{2,M}, \dots, v_{N-1,2}, v_{N-1,3}, \dots, v_{N-1,M-1}]^T$ are $(N - 2)(M - 2) \times 1$ dimensional vectors. The following Pike and Roe’s fourth-stage RK4 [15] is used to solve the above system

$$\begin{cases} U = U^n \\ V = V^n \\ H = F(U, V) \\ I = G(U, V) \end{cases} \rightarrow \rightarrow \rightarrow \rightarrow \begin{cases} U = U + \frac{\Delta t}{4} \cdot F \\ V = V + \frac{\Delta t}{4} \cdot G \\ H = F(U, V) \\ I = G(U, V) \end{cases} \tag{31}$$

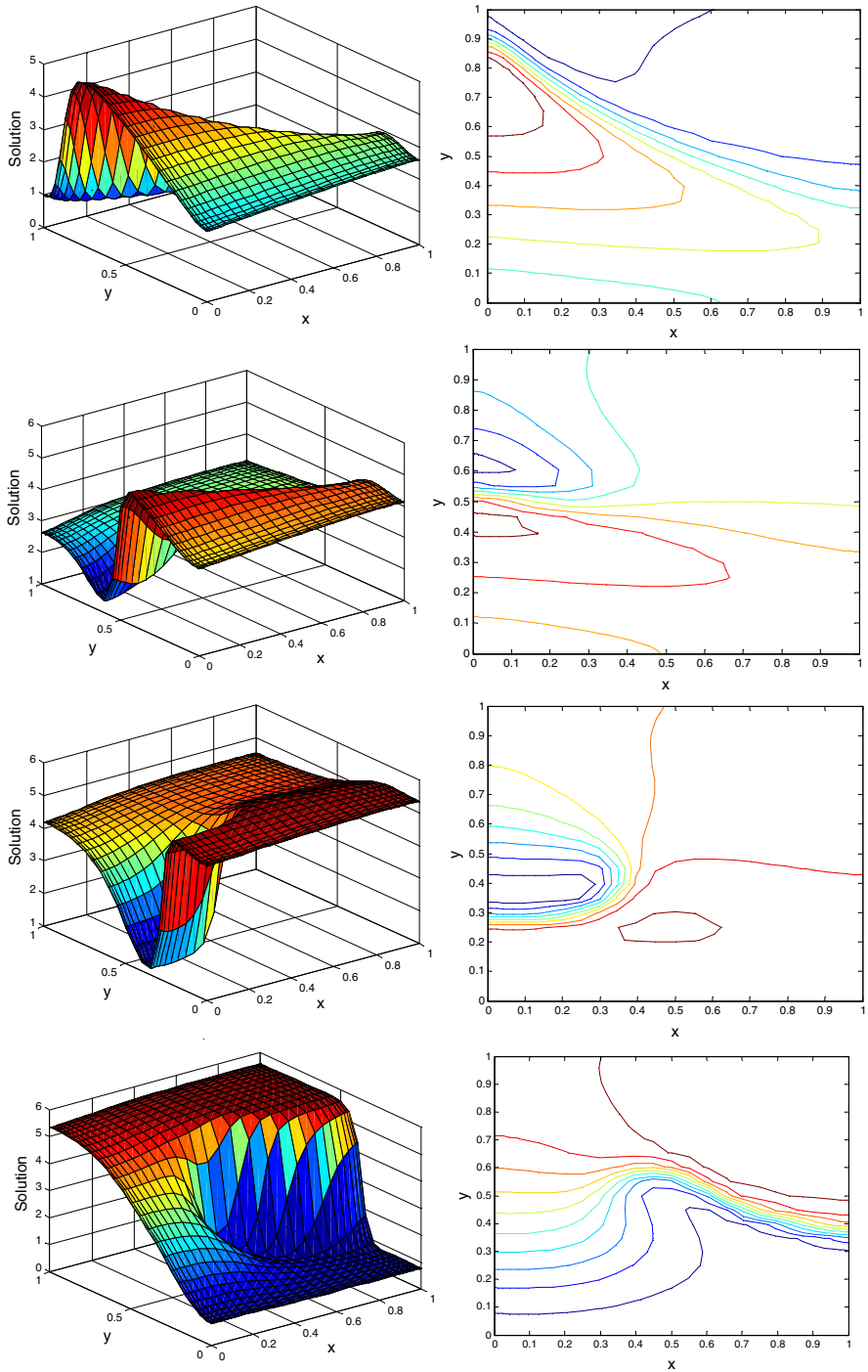


Fig. 1 Approximated solutions in 3D (left) and contour form (right) of Example 1 at times $t = 1, 3, 5, 7, 10, 15$ s respectively with $\Delta t = 0.001, N = 31$

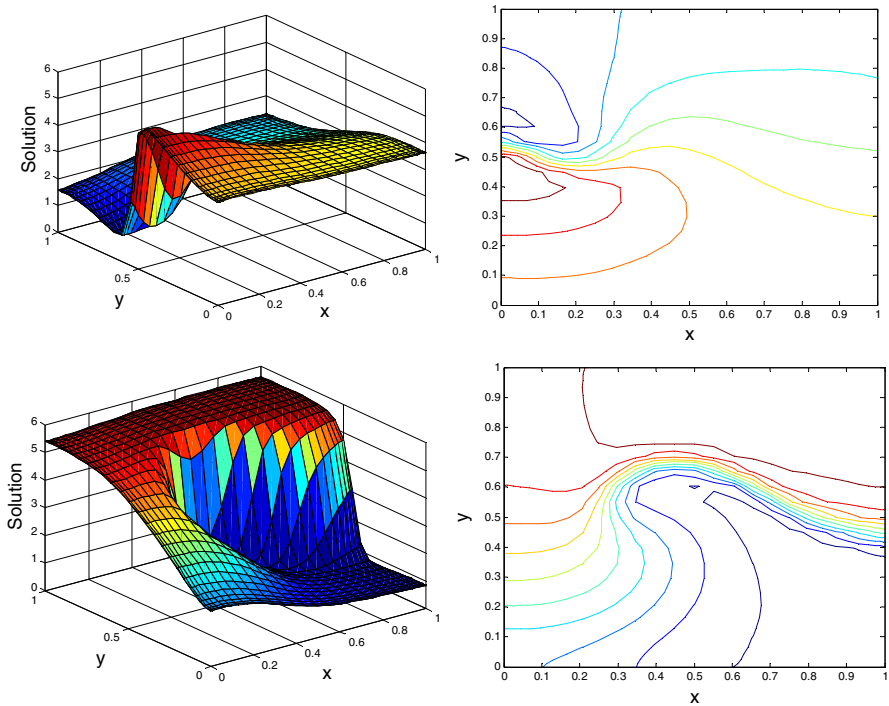


Fig. 1 continued

$$\begin{cases} U = U + \frac{\Delta t}{3} \cdot F \\ V = V + \frac{\Delta t}{3} \cdot G \\ H = F(U, V) \\ I = G(U, V) \end{cases} \rightarrow \rightarrow \rightarrow \begin{cases} U = U + \frac{\Delta t}{2} \cdot F \\ V = V + \frac{\Delta t}{2} \cdot G \\ H = F(U, V) \\ I = G(U, V) \end{cases} \quad (32)$$

$$\begin{cases} U^{n+1} = U + \Delta t \cdot F \\ V^{n+1} = V + \Delta t \cdot G \end{cases} \quad (33)$$

where F, G should be independent of variable t .

4 Numerical experiments and discussions

In this section, some test examples have considered to check the accuracy and efficiency of the proposed method. The whole computation work is done with the help of MATLAB 7.0, DEV C++ and time step $\Delta t = 0.001$. L_∞ , root mean square (RMS) and L_2 errors are calculated by using following formulas

$$L_\infty = \max_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M}} |u_{ij}|, \quad RMS = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^M |(u_{ij} - \bar{u}_{ij})|^2}{N \times M}}$$

Table 3 Approximated solutions of Example 2 with $A = 0.5, B = 1.0, \alpha = 0.002$ at different time and mesh points

t	(0.2, 0.2)		(0.4, 0.6)		(0.8, 0.9)	
	u	v	u	v	u	v
1.0	0.5327	0.1708	0.5468	0.2150	0.5804	0.3078
2.0	0.7044	0.3729	0.7183	0.4028	0.7509	0.4622
3.0	0.8191	0.4954	0.8339	0.5092	0.8669	0.5333
5.0	0.9724	0.5304	0.9793	0.5282	0.9921	0.5228
7.0	1.0062	0.5037	1.0064	0.5026	1.0064	0.5007
8.0	1.0047	0.4996	1.0043	0.4993	1.0034	0.4988
9.0	1.0020	0.4989	1.0017	0.4989	1.0011	0.4990
10.0	1.0005	0.4993	1.0003	0.4993	1.0001	0.4995
↓	↓	↓	↓	↓	↓	↓
∞	1.0	0.5	1.0	0.5	1.0	0.5

$$L_2 = \sqrt{\sum_{i=1}^N \sum_{j=1}^M |(u_{ij} - \bar{u}_{ij})|^2}$$

where u_{ij}, \bar{u}_{ij} are approximate and exact solutions respectively.

Example 1 [9, 11] The system (2a) and (2b) is considered subject to Neumann boundary conditions (4) and with initial conditions

$$\begin{cases} f_1(x, y, 0) = 0.5 + y \\ f_2(x, y, 0) = 1.0 + 5x \end{cases}$$

Table 2 shows approximated solutions at different times and points with constants $A = 0.5, B = 1.0$ and $\alpha = 0.002$. It is clear from the table that solution converges to equilibrium point $(B, A/B)$ since $1 - A + B^2 > 0$ and satisfies the result discussed in [6]. The concentration profiles of u in 3D and contour form are depicted in Fig. 1 for time from $t = 1.0$ to $t = 15.0$ with $A = 3.4, B = 1.0$ and $\alpha = 0.002$. These figures are quite similar to those obtained in [10, 12].

Example 2 [9, 12] In this example, the two-dimensional reaction–diffusion Brusselator system (2a) and (2b) is considered subject to Neumann boundary conditions (4) and initial conditions

$$\begin{cases} f_1(x, y, 0) = 0.5x^2 - \frac{1}{3}x^3 \\ f_2(x, y, 0) = 0.5y^2 - \frac{1}{3}y^3 \end{cases}$$

In this example, the constants A, B and α are taken as 0.5, 1 and 0.002 respectively. Table 3 shows approximated solutions at different times and points and solutions satisfy the well know result discussed in [6]. Figure 2 shows concentration profiles of

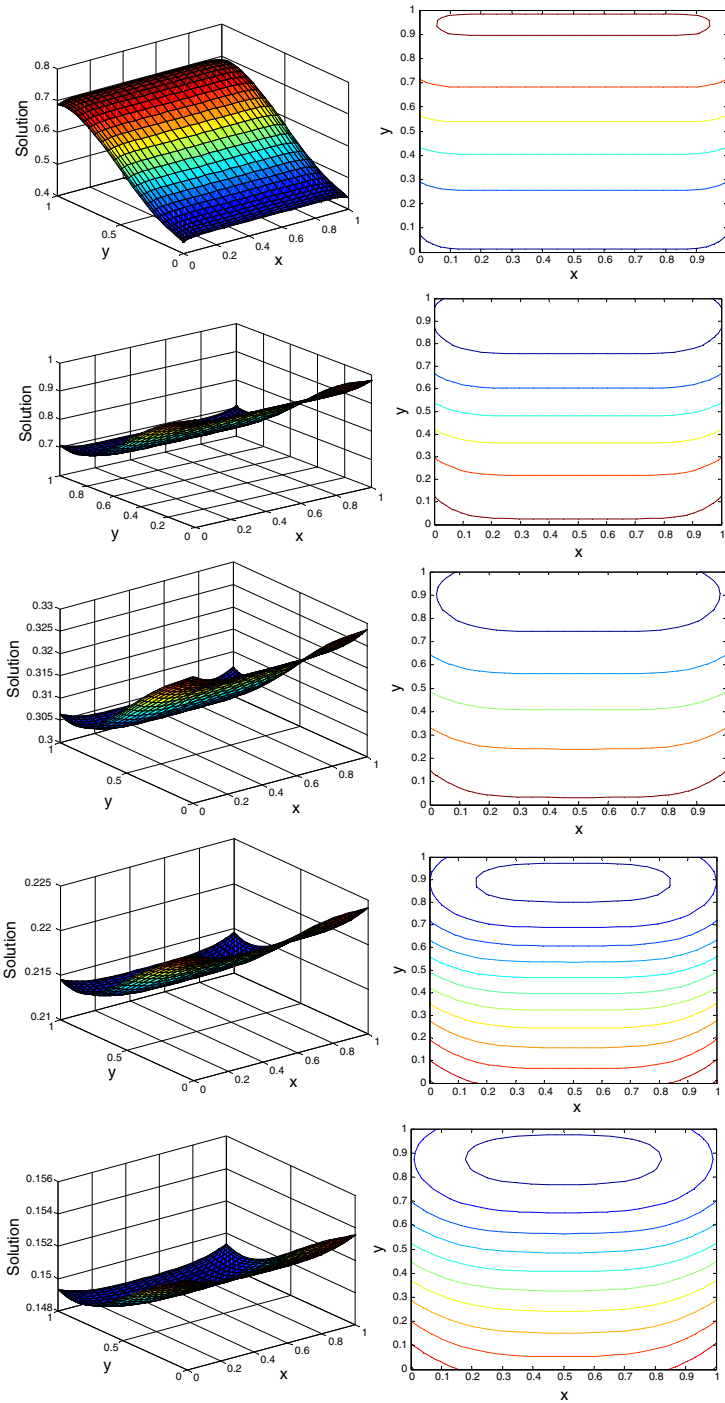


Fig. 2 Approximated solutions in 3D (*left*) and contour form (*right*) of Example 2 at times $t = 1, 3, 5, 7, 10$ s respectively with $\Delta t = 0.001, N = 31$

Table 4 Approximated solutions of Example 3 with $A = 1.0, B = 2.0, \alpha = 0.002$ at different time and mesh points

t	(0.2, 0.2)		(0.4, 0.6)		(0.8, 0.9)	
	u	v	u	v	u	v
1.0	2.3454	0.4163	2.4479	0.3956	2.6069	0.3693
2.0	2.0952	0.4689	2.1256	0.4605	2.1757	0.4478
3.0	2.0219	0.4916	2.0298	0.4889	2.0433	0.4844
5.0	2.0008	0.4996	2.0011	0.4995	2.0017	0.4993
7.0	2.0001	0.4999	2.0003	0.4998	2.0001	0.4999
8.0	2.0	0.5	2.0	0.5	2.0	0.5
9.0	2.0	0.5	2.0	0.5	2.0	0.5
↓	↓	↓	↓	↓	↓	↓
∞	2.0	0.5	2.0	0.5	2.0	0.5

u in 3D and contour form at time from $t = 1.0$ to $t = 15.0$. The Figure is very similar to those obtained in [12].

Example 3 ([6,12]) Consider the Brusselator system (2a) and (2b) with Neumann boundary conditions (4) and initial conditions

$$\begin{cases} f_1(x, y, 0) = 2.0 + 0.25y \\ f_2(x, y, 0) = 1.0 + 0.8x \end{cases}$$

The results of the example are shown in Table 4 and Fig. 3 with constants $A = 1.0, B = 2.0$ and $\alpha = 0.002$. Table 4 shows approximated solutions at different times and points which converge to the equilibrium point $(B, A/B)$ i.e. $(2.0, 0.5)$ while Fig. 3 depicts concentration profiles of u in 3D and contour form at time from $t = 1.0$ to $t = 8.0$. It is clear from the figure that the computed results are quite agree with results discussed in [12].

Example 4 ([9,11]) In this example, the two-dimensional reaction–diffusion Brusselator system is considered with the following exact solution

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

The initial conditions are taken from the exact solution. Table 6 reports the L_∞ , RMS and L_2 errors with CPU time for the concentrations u and v are computed using with the constants $A = 1, B = 0, \alpha = 0.25$ at different values of time t . The errors are computed with mesh points $N = 21, M = 21$ and similar to the error in [11]. Figure 4 depicts the concentration profiles of u in 3D and contour form at time from $t = 1.0$ to $t = 5.0$.

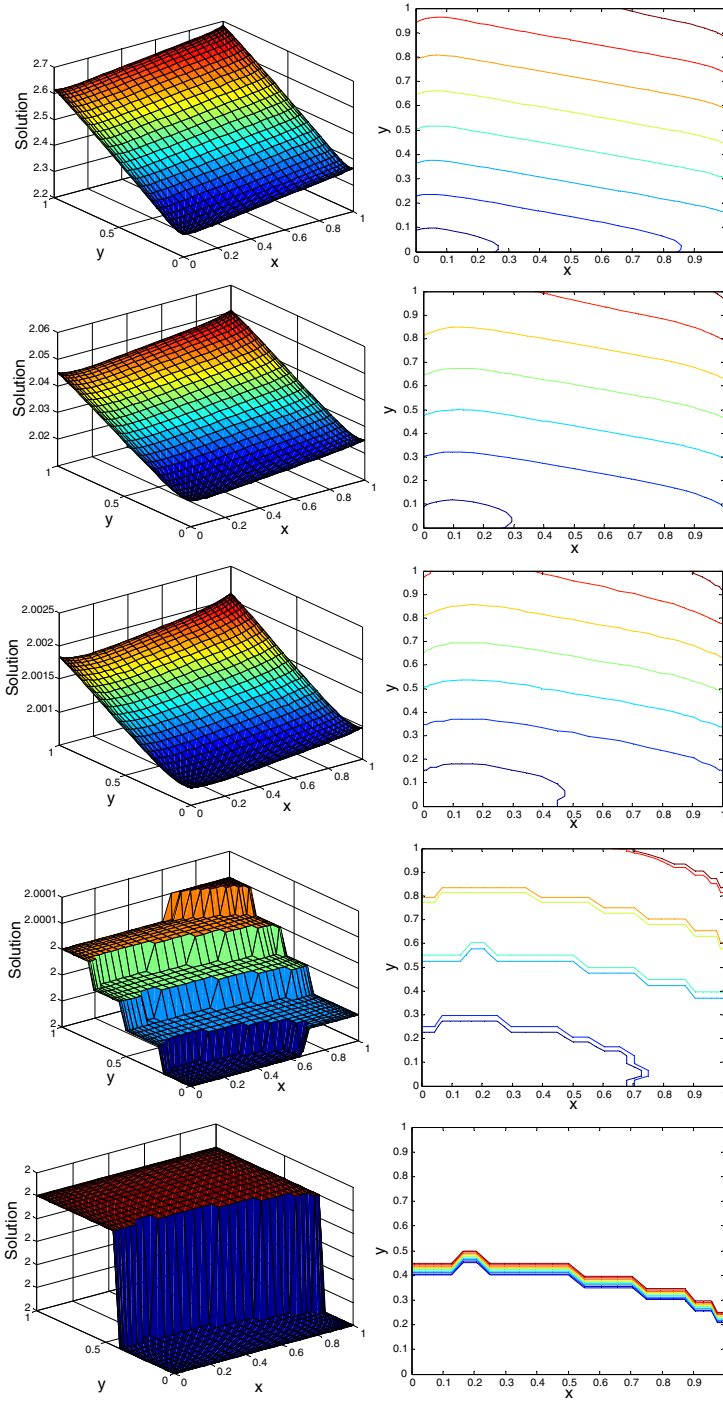


Fig. 3 Approximated solutions in 3D (left) and contour form (right) of Example 3 at times $t = 1, 3, 5, 7, 8$ s respectively with $\Delta t = 0.001, N = 31$

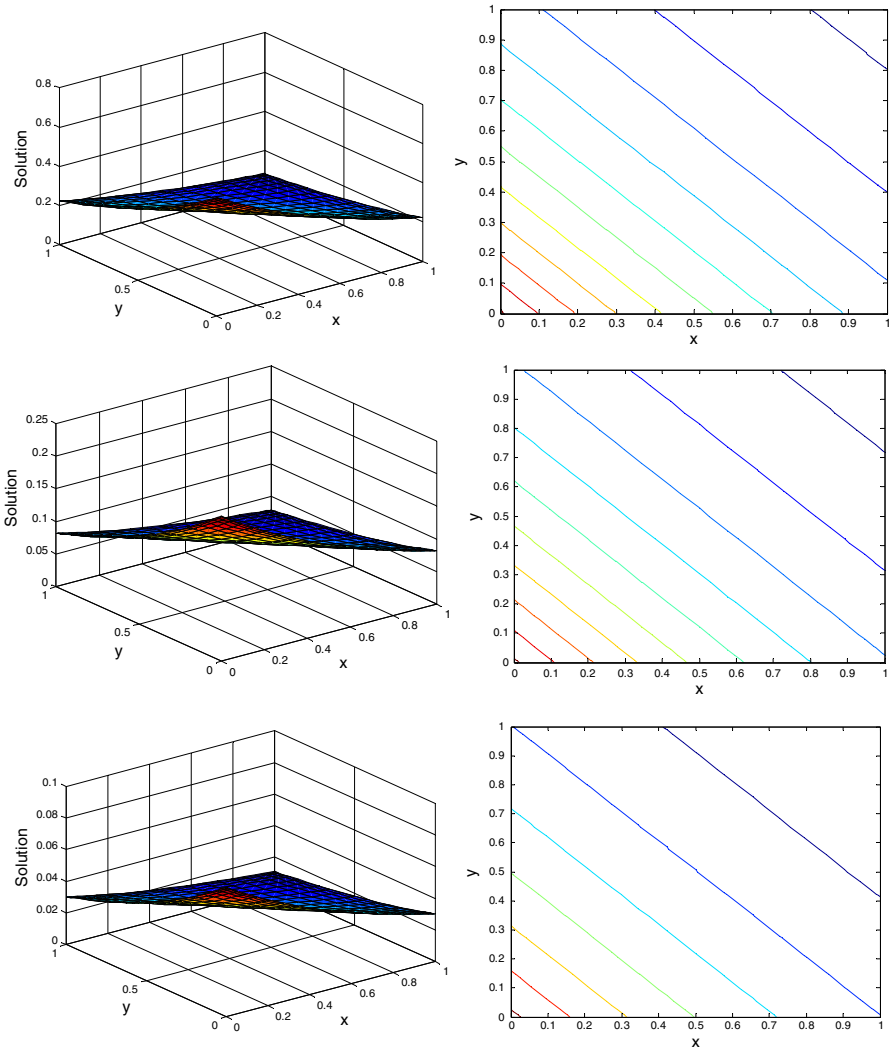


Fig. 4 Approximated solutions in 3D (*left*) and contour form (*right*) of Example 4 at times $t = 1, 3, 5$ s respectively with $\Delta t = 0.001, N = 21$

5 Conclusion

In this article, the authors proposed a modified cubic B-spline differential quadrature method (MCB-DQM) to show the computational modeling of two-dimensional reaction–diffusion Brusselator system with Neumann boundary conditions. The proposed method tested on four examples available in literature. The main outcomes of the work are summarize as follows

Table 5 Approximated solutions of Examples with $A = 3.4$, $B = 1.0$, $\alpha = 0.002$ at different time and mesh point (0.4, 0.6)

t	Example 1		Example 2		Example 3	
	u	v	u	v	u	v
1.0	0.8368	3.2762	0.2331	0.8850	0.6461	2.5522
2.0	0.6709	3.5366	0.2501	1.8699	0.3245	3.4457
3.0	0.4298	4.1448	0.2692	2.8515	0.3169	4.1375
5.0	0.4515	5.2417	0.3392	4.7821	0.4133	5.3282
7.0	3.6545	0.9481	1.4307	5.6919	4.6299	0.7520
8.0	1.4573	1.7857	7.5380	0.58519	1.7955	1.5454
9.0	0.4913	2.9030	0.6120	0.5117	0.5496	2.7299
10.0	0.3234	3.6883	9.6697	0.4555	0.3186	3.5706

Table 6 Maximum absolute L_∞ , RMS, L_2 errors and CPU time of Example 4 at different time t

t	u			v			CPU Time
	L_∞	RMS	L_2	L_∞	RMS	L_2	
0.5	1.337E-06	7.796E-06	1.637E-06	3.654E-06	9.482E-06	1.991E-06	02
1.0	3.241E-06	6.174E-06	1.296E-06	4.546E-06	1.239E-06	2.603E-06	06
1.5	1.526E-06	4.813E-06	1.010E-06	6.324E-06	1.593E-06	3.347E-06	07
2.0	1.354E-06	3.748E-06	7.872E-06	5.245E-06	2.046E-06	4.298E-06	10
3.0	1.134E-06	2.273E-06	4.475E-06	1.230E-06	3.374E-06	7.686E-06	15

- (i) A different technique based on modified cubic-B-spline functions is proposed to find the weighting coefficients of differential quadrature method than the traditional technique of Lagrange interpolation [14].
- (ii) The proposed method gives very similar results to those discussed in [6,9–12] and good accuracy for small number of grid points $N = 31$, $M = 31$ with small computational cost i.e. CPU time.
- (iii) Tables 2, 3, 4 show that the obtained results satisfy the well known result that for small values of diffusion coefficient, the steady state solution converges to equilibrium point $(B, A/B)$ if $1 - A + B^2 > 0$ i.e. the approximated solutions converge. But, Table 5 shows that for $1 - A + B^2 < 0$ the approximated solutions not converge. The Figures in 3D and contour form show the real computational modeling of two-dimensional reaction–diffusion Brusselator system.
- (iv) In the proposed method, Neumann boundary conditions are very easy to handle and easy to make the Matlab code of the method.
- (v) The present method with some modifications can be easily extended to solve model equations in two or higher dimensional problems including mechanical, physical or biophysical effects, such as nonlinear convection, reaction, linear diffusion and dispersion.

Acknowledgments The authors are very thankful to the reviewers for their valuable suggestions to improve the quality of the paper.

References

1. A.M. Turing, The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond. Ser.* **237**, 37 (1952)
2. R. Lefever, G. Nicolis, Chemical instabilities and sustained oscillations. *J. Theor. Biol.* **30**, 267 (1971)
3. G. Nicolis, I. Prigogine, *Self-Organization in Non-Equilibrium Systems* (Wiley/Interscience, New York, 1977)
4. I. Prigogine, R. Lefever, Symmetries breaking instabilities in dissipative systems II. *J. Phys. Chem.* **48**, 1695–1700 (1968)
5. J. Tyson, Some further studies of nonlinear oscillations in chemical systems. *J. Chem. Phys.* **58**, 3919 (1973)
6. E.H. Twizell, A.B. Gumel, Q. Cao, A second-order scheme for the Brusselator' reaction–diffusion system. *J. Math. Chem.* **26**, 297–316 (1999)
7. G. Adomian, The diffusion-Brusselator equation. *Comput. Math. Appl.* **29**, 1–3 (1995)
8. 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)
9. W.T. Ang, The two-dimensional reaction-diffusion Brusselator system: a dual-reciprocity boundary element solution. *Eng. Anal. Bound. Elem.* **27**, 897–903 (2003)
10. J.G. Verwer, W.H. Hundsdorfer, B.P. Sommeijer, Convergence properties of the Runge–Kutta–Chebyshev method. *Numer. Math.* **57**, 157–178 (1990)
11. A.A. Siraj-ul-Islam, S. Haq, A computational modeling of the behavior of the two-dimensional reaction-diffusion Brusselator system. *Appl. Math. Model.* **34**, 3896–3909 (2010)
12. R.C. Mittal, R. Jiwari, Numerical study of two-dimensional reaction–diffusion Brusselator system. *Appl. Math. Comput.* **217**(12), 5404–5415 (2011)
13. S. Kumar, Y. Khan, A. Yildirim, A mathematical modeling arising in the chemical systems and its approximate numerical solution. *Asia Pac. J. Chem. Eng.* **7**, 835–840 (2012)
14. C. Shu, *Differential Quadrature and its Application in Engineering* (Athenaeum Press Ltd., Great Britain, 2000)
15. J. Pike, P.L. Roe, Accelerated convergence of Jameson's finite volume Euler scheme using Van Der Houwen integrators. *Comput. Fluids* **13**, 223–236 (1985)
16. R. Bellman, B.G. Kashef, J. Casti, Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations. *J. Comput. Phys.* **10**, 40–52 (1972)
17. J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature methods-I. *Comput. Chem. Eng.* **13**, 779–788 (1989)
18. J.R. Quan, C.T. Chang, New insights in solving distributed system equations by the quadrature methods-II. *Comput. Chem. Eng.* **13**, 1017–1024 (1989)
19. A. Korkmaz, İ. Dağ, Shock wave simulations using Sinc differential quadrature method. *Eng. Comput.* **28**(6), 654–674 (2011)
20. A. Korkmaz, İ. Dağ, A differential quadrature algorithm for nonlinear Schrödinger equation. *Nonlinear Dyn.* **56**(1–2), 69–83 (2009)
21. R.C. Mittal, R. Jiwari, A differential quadrature method for solving Burgers'-type equation. *Int. J. Numer. Methods Heat Fluid Flow* **22**(7), 880–895 (2012)
22. R. Jiwari, S. Pandit, R.C. Mittal, Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method. *Comput. Phys. Commun.* **183**(3), 600–616 (2012)
23. R. Jiwari, S. Pandit, R.C. Mittal, A differential quadrature algorithm to solve the two dimensional linear hyperbolic telegraph equation with Dirichlet and Neumann boundary conditions. *Appl. Math. Comput.* **218**(13), 7279–7294 (2012)
24. R.C. Mittal, R.K. Jain, Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method. *Appl. Math. Comput.* **218**, 7839–7855 (2012)