Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

A Numerical Algorithm for the Computational Study of the Semi-Linear Parabolic Partial Differential Equations

Sameer Kumar, Rajni Rohila and Alka Chadha
The NorthCap University, IIT Raichur

Abstract:- We investigate solutions of the two-dimensional semi-linear parabolic equations by a numerical method that employs cubic B- spline functions. Such type of equations arise in chemical reaction theory, mathematical biology, population dynamics, material science, and many other areas of science and engineering. The spirit of the method lies in the evaluation of first and second-order weighting coefficients by differential quadrature approximations. The two-dimensional equation has been discretized by the cubic B-spline differential quadrature method to obtain a system of ordinary differential equations(ODEs). A highly stability-preserving SSPRK-43 method has been applied to solve the system of ODEs. This method uses less storage which reduces the accumulation of numerical errors. Solutions computed by implementing the modified differential quadrature method have been presented graphically and in tabular form.

***Keywords**:* Cubic B-spline Functions, Two dimensional parabolic equations, Differential quadrature technique, SSPRK-43 Method

1. Introduction

We consider over $(x, y) \in (a, b) \times (c, d)$, the two dimensional parabolic equation given by

$$u_t = p(t)\left(u_{xx} + u_{yy}\right) + f(u, x, y, t),\ t \in (0, T) \tag{1.1}$$

There are a number of nonlinear partial differential equations for which analytical solutions are not feasible. Such equations are investigated by numerical techniques. A number of numerical methods have been studied to solve the two dimensional semi-linear parabolic equation. The differential quadrature technique is proved to be a very strong alternative to the collocation, pseudo-spectral and Rayleigh Ritz methods. The theoretical analysis and numerical experiments prove that differential quadrature is very efficient for highly nonlinear problems. There are different kinds of basis functions which have been employed efficiently to solve the problems involving steep gradients. This striking feature of the differential quadrature method makes it very popular among the researchers. For problems with the regular geometries and problems of distributed-parameter systems the method is more efficient than the finite difference and finite element techniques.

Milstein and Tretyakov [1] developed a probabilistic approach for constructing special layer methods to obtain numerical solutions of the Cauchy problem for semi-linear parabolic equations. Hochbruck and Ostermann [2] studied semi-linear parabolic problems by the explicit exponential Runge Kutta method. Dehghan [3] proposed the finite difference scheme to solve such type of equations.

The differential quadrature method was developed by Bellman and his associated in 1970s. It is a numerical discretization technique. As a result of innovative work for the calculation of weighting coefficients, differential quadrature method was developed as a discretization technique for the numerical solutions of differential equations. Dag et al. [4] proposed the cosine differential quadrature method to study the regularized long wave equations. Korkmaz [5] et. al developed the quartic B-spline differential quadrature technique to study the sinusoidal disturbance and shock wave solutions of Burgers' equations. Mittal and Bhatia [6] solved the hyperbolic partial differential equations by using spline differential quadrature method. Recently, the modified B-spline [7] and the Bernstein differential quadrature [8] methods have been applied by Mittal. The exponential cubic B spline [9] and trigonometric B-spline [10] is applied by Dag and Zahra respectively to obtain numerical solutions of parabolic partial differential equations.

The article is written as follows:

1. In Section 2, the cubic B-splines have been introduced which is followed by the definition of the modified B-spline functions.
2. In Section 3, weighting coefficients have been calculated and implemented to the two dimensional problems.
3. A matrix method has been applied to check stability of the proposed method in Section 4.
4. In Section 5 , we check the applicability and efficiency of the method to solve parabolic problems by applying method on four important equations taken from the literature. Finally we sum up with a brief discussion concluding the findings of the method in section 6 .

2. Modified Cubic B-spline Differential Quadrature Method

Consider a partition of the domain $\Omega = (a,b) \times (c,d)$ by the nodes $x_i's$ and $y_j's$ by taking N and M points along the co-ordinate directions x and y respectively such that $x_i - x_{i-1} = h$ and $y_j - y_{j-1} = k$. Cubic B-spline functions have been used for approximation as they offer smooth interpolation.

The cubic B-splines at the nodes are defined by

$$S_j(x) = \frac{1}{h^3}\begin{cases} (x - x_{j-2})^3, & x \in [x_{j-2}, x_{j-1}] \\ (x - x_{j-2})^3 - 4(x - x_{j-1})^3, & x \in [x_{j-1}, x_j] \\ (x_{j+2} - x)^3 - 4(x_{j+1} - x)^3, & x \in [x_j, x_{j+1}] \\ (x_{j+2} - x)^3, & x \in [x_{j+1}, x_{j+2}] \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where $\{S_0(x), S_1(x), S_2(x), \ldots, S_N(x), S_{N+1}(x)\}$ is a basis over the interval (a,b). Cubic B-splines and derivatives have been evaluated by using above definition and have been presented in Table 1.

In this work cubic B- splines have been modified. Modified cubic B-splines at a boundary will provide full support and will give a system of equations in which the matrix is diagonally dominant.

TABLE 1 .

x	x_{j-2}	x_{j-1}	x_j	x_{j+1}	x_{j+2}
$S_i(x)$	0	1	4	1	0
$S_i'(x)$	0	$\frac{3}{h}$	0	$\frac{-3}{h}$	0
$S_i''(x)$	0	$\frac{6}{h^2}$	$\frac{-12}{h^2}$	$\frac{6}{h^2}$	0

Values at the knots for cubic B-splines and its derivatives

The modified cubic B-spline functions [11] at the nodes are defined by equations $(2.2) - (2.6)$:

$$\tilde{S}_1(x) = S_1(x) + 2S_0(x),\ j = 1 \quad (2.2)$$
$$\tilde{S}_2(x) = S_2(x) - S_0(x),\ j = 2 \quad (2.3)$$
$$\tilde{S}_j(x) = S_j(x),\ j = 3,4,5 \dots .N - 2 \quad (2.4)$$
$$\tilde{S}_{N-1}(x) = S_{N-1}(x) - S_{N+1}(x),\ j = N - 1 \quad (2.5)$$
$$\tilde{S_N}(x) = S_N(x) + 2S_{N+1}(x),\ j = N \quad (2.6)$$

where the set $\{\tilde{S}_1(x), \tilde{S}_2(x), \dots . \tilde{S_N}(x)\}$ constructs a basis. De Boor [12] has given many important properties of B-spline functions which reader may explore for detailed study of spline functions.

3. Development of the Numerical Scheme

To approximate the given partial differential equation, we first determine weighting coefficients as follows:

Determination of weighting coefficients Differential quadrature method is basically a discretization technique for solving partial differential equations. The n-th derivative of u at (x_i, y_j, t) with respect to x may be approximated as

$$u_x^{(n)}(x_i, y_j, t) = \sum_{k=1}^{N} a_{ik}^{(n)} u(x_k, y_j, t), i = 1,2 \dots N, j = 1,2 \dots M \quad (3.1)$$

In a similar way m-th derivative of u at (x_i, y_j, t) with respect to y may be approximated as

$$u_y^{(m)}(x_i, y_j, t) = \sum_{k=1}^{N} b_{jk}^{(n)} u(x_i, y_k, t), i = 1,2 \dots N, j = 1,2 \dots M \quad (3.2)$$

where $a_{ij}^{(n)}$ and $b_{ij}^{(n)}$ are weighting coefficients corresponding to derivatives with respect to x and y at the point (x_i, y_j). For the first order derivative, the differential quadrature approximation gives

$\tilde{S}_l'(x_i) = \sum_{j=1}^{N} a_{ij}^{(1)} \tilde{S}_l(x_j), l = 1,2,3 \dots ..N$ and $i = 1,2..N$

At point x_1, we obtain as follows

$\tilde{S}_l'(x_1) = \sum_{j=1}^{N} a_{1j}^{(1)} \tilde{S}_l(x_j)$ where $l = 1,2,3 \dots .N$

A system of equations is obtained as follows

$$\begin{pmatrix} 6 & 1 & & & & & \\ 0 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ .. & .. & .. & .. & .. & .. & .. \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 0 \\ & & & & & 1 & 6 \end{pmatrix} \times \begin{pmatrix} a_{11}^{(1)} \\ a_{12}^{(1)} \\ . \\ . \\ . \\ . \\ a_{1N}^{(1)} \end{pmatrix} = \begin{pmatrix} \frac{-6}{h} \\ \frac{6}{h} \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix}$$

This tridiagonal system contains N unknowns $a_{11}^{(1)}, a_{12}^{(1)}, .., a_{1N}^{(1)}$ which have been found by applying the Thomas algorithm. In a similar fashion, for the point x_2, we write

$\tilde{S}_l'(x_2) = \sum_{j=1}^{N} a_{2j}^{(1)} \tilde{S}_l(x_j), l = 1,2,3 \dots ..N.$

We obtain a system of equations as below

$$\begin{pmatrix} 6 & 1 & & & & & \\ 0 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ .. & .. & .. & .. & .. & .. & .. \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 0 \\ & & & & & 1 & 6 \end{pmatrix} \times \begin{pmatrix} a_{21}^{(1)} \\ a_{22}^{(1)} \\ . \\ . \\ . \\ . \\ a_{2N}^{(1)} \end{pmatrix} = \begin{pmatrix} \frac{-3}{h} \\ 0 \\ \frac{3}{h} \\ . \\ . \\ . \\ 0 \end{pmatrix}.$$

This system is solved by the Thomas algorithm to find $a_{21}^{(1)}, a_{22}^{(1)}, \ldots, a_{2N}^{(1)}$. Proceeding in this way, we calculate weighting coefficients corresponding to the knots $x_3, x_4 \ldots x_{N-1}$. At the knot x_N,

$\tilde{S}_l'(x_N) = \sum_{j=1}^{N} a_{Nj}^{(1)} S_l(x_j), l = 1,2,3 \ldots. N,$

we obtain a system as follows,

$$\begin{pmatrix} 6 & 1 & & & & & \\ 0 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ .. & .. & .. & .. & .. & .. & .. \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 0 \\ & & & & & 1 & 6 \end{pmatrix} \times \begin{pmatrix} a_{N1}^{(1)} \\ a_{N2}^{(1)} \\ . \\ . \\ . \\ . \\ a_{NN}^{(1)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ . \\ . \\ \frac{-6}{h} \\ \frac{6}{h} \end{pmatrix}$$

Unknowns $a_{N1}^{(1)}, a_{N2}^{(1)}, \ldots, a_{NN}^{(1)}$ are calculated by applying the Thomas algorithm again . Thus all the first order weighting coefficients have been determined. Weighting coefficients of order more than one have been determined by using a recursive formula given by Shu 13 as follows

$$a_{ij}^{(r)} = r\left[a_{ij}^{(1)} a_{ii}^{(r-1)} - \frac{a_{ij}^{(r-1)}}{(x_i - x_j)}\right], i \neq j \quad (3.3)$$

$$a_{ii}^{(r)} = -\sum_{j=1, j\neq i}^{N} a_{ij}^{(r)}, i = j \text{ and } i = 1,2 \ldots N \quad (3.4)$$

Weighting coefficients corresponding to the spatial discretization in the y direction may be evaluated following the same procedure as above.

3.1. Implementation. : We discretize (1.1) by differential quadrature method as follows:

$$\frac{du_{i,j}}{dt} = p(t)\left(\sum_{k=1}^{N} a_{ik}^{(2)} u_{k,j} + \sum_{k=1}^{M} b_{jk}^{(2)} u_{i,k}\right) + f(u_i, x_i, y_j, t),\ i = 1,2 \ldots N,\ j = 1,2 \ldots M \quad (3.5)$$

By applying boundary conditions, we obtain

$$\frac{du_{i,j}}{dt} = p(t)\left(\sum_{k=2}^{N-1} a_{ik}^{(2)} u_{k,j} + \sum_{k=2}^{M-1} b_{jk}^{(2)} u_{i,k}\right) + f\left(u_i, x_i, y_j, t\right) + a_{i1}^{(2)} u_{1,j} + a_{iN}^{(2)} u_{N,j} + b_{j1}^{(2)} u_{i,1} + b_{jM}^{(2)} u_{i,M} \quad (3.6)$$

$i = 2 \dots N-1, j = 2 \dots M-1.$

It gives $N + M$ ordinary differential equations. For solving this system of equations, we have applied stage four, order three Runge Kutta[SSPRK-43] which is a strong stability preserving time stepping method. If $\frac{dx}{dt} = L(x)$ is the system of ODEs. It is solved by SSPRK-43 time stepping scheme given by Spiteri and Ruuth [15] as follows:

$$x^{(1)} = x^{(m)} + \frac{\Delta t}{2} L\left(x^{(m)}\right) \quad (3.7)$$

$$x^{(2)} = x^{(1)} + \frac{\Delta t}{2} L\left(x^{(1)}\right) \quad (3.8)$$

$$x^{(3)} = \frac{2}{3} x^{(m)} + \frac{x^{(2)}}{3} + \frac{\Delta t}{6} L\left(x^{(2)}\right) \quad (3.9)$$

$$x^{(m+1)} = x^{(3)} + \frac{\Delta t}{2} L\left(x^{(3)}\right) \quad (3.10)$$

Initial values are calculated by using initial condition of the problem. Hence the approximate values of w at any time are completely obtained.

4. Numerical Stability

For any numerical method, it is necessary that the approximations generated by the scheme are close to the true solution. In this context, it will be governed by the stability of the scheme which is often termed as numerical stability of the method. For this we have to be very careful in choosing step size in the numerical solver used. The numerical stability means that the errors introduced during calculation remains bounded. This is very necessary, since in any numerical scheme errors are impossible to avoid. We have to deal with discretization errors. If errors amplify without limit, they change the approximate solution so much which makes it useless. For finite difference schemes, von Neumann method may be used to check stability. For the present method, we have applied matrix analysis method 17 18 19 for stability of the system derived in differential quadrature method. System (3.6) may be written as

$$\frac{d[u]}{dt} = A[u] + b \quad (4.1)$$

where $[u]$ is a vector of unknowns at the interior nodes of the domain and A is matrix of coefficients. Stability of numerical scheme is analyzed by finding the eigenvalues of matrix A. To obtain converged solutions, it is necessary that A must have negative real eigenvalues. Matrix A may be written as follows: $A = A_1 + A_2$

$$\text{where } A_1 = p(t)\begin{pmatrix} a_{22}^{(2)} I_{M-2} & a_{23}^{(2)} I_{M-2} & a_{24}^{(2)} I_{M-2} & & a_{2,N-1}^{(2)} I_{M-2} \\ a_{32}^{(2)} I_{M-2} & a_{33}^{(2)} I_{M-2} & a_{34}^{(2)} I_{M-2} & & a_{23,N-1}^{(2)} I_{M-2} \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \;\; \cdot\cdot & \cdot\cdot \\ a_{N-1,2}^{(2)} I_{M-2} & a_{N-1,3}^{(2)} I_{M-2} & a_{N-1,4}^{(2)} I_{M-2} & & a_{2N-1,N-1}^{(2)} I_{M-2} \end{pmatrix},$$

where I_{M-2} is an identity matrix of order $M-2$.

$$\text{Similarly } A_2 = p(t)\begin{pmatrix} P_{ij} & O & O & & & O \\ O & P_{ij} & O & & & O \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ O & O & O & & & P_{ij} \end{pmatrix},$$

where $P_{ij} = \begin{pmatrix} b_{22}^{(2)} & b_{23}^{(2)} & b_{24}^{(2)} & & b_{2,M-1}^{(2)} \\ b_{32}^{(2)} & b_{33}^{(2)} & b_{34}^{(2)} & & b_{3,M-1}^{(2)} \\ \vdots & \cdots & \cdots & \cdots & \vdots \\ b_{M-1,2}^{(2)} & b_{M-1,3}^{(2)} & b_{M-1,4}^{(2)} & & b_{M-1,M-1}^{(2)} \end{pmatrix}$.

Stability of DQM scheme depends on eigenvalues of A. We have determined eigenvalues of A and found them to be negative real(Figure 1) for different grid points which confirms that our scheme produces stable solutions.

5. Numerical simulations

In this section, numerical experiments have been performed by taking four important two dimensional equations to validate the efficiency and performance of the present method. To check the accuracy of the method, maximum absolute error norm is found as follows

$$L_\infty = \|u^{\text{exact}} - u^N\|_\infty = \max\left|u_{i,j}^{\text{exact}} - u_{i,j}^N\right| \tag{5.1}$$

here u^N stands for numerical solution. $u_{i,j}^{\text{exact}}$ and $u_{i,j}^N$ represent the exact and approximate solutions respectively at the point (x_i, y_j). Computed results have been presented in tables and also depicted in graphs.

Example 1: We consider equation (1.1) with $p(t) = \frac{1}{2\pi^2}, a = 1, b = -1$ and $f(u, x, y, t) = 0$. The exact solution of the problem is

$$u(x, y, t) = e^{-t}\sin(\pi x)\sin(\pi y) \tag{5.2}$$

By using equation (5.2), we can find initial and boundary conditions. The computed solutions at $t = 1.0$ with $N = 100$ have been presented in Table 2. We can observe from maximum absolute errors that computed solutions are very close to analytical values. Root mean square (rms) and maximum absolute errors have been given in Table 3. CPU time is reported in the same table. We can observe that CPU time is very small. Solutions at $t = 0.5$ and $t = 1.0$ have been depicted in Figures 2 and 3 respectively. A 3-D view of solutions is also depicted in Figure 4.

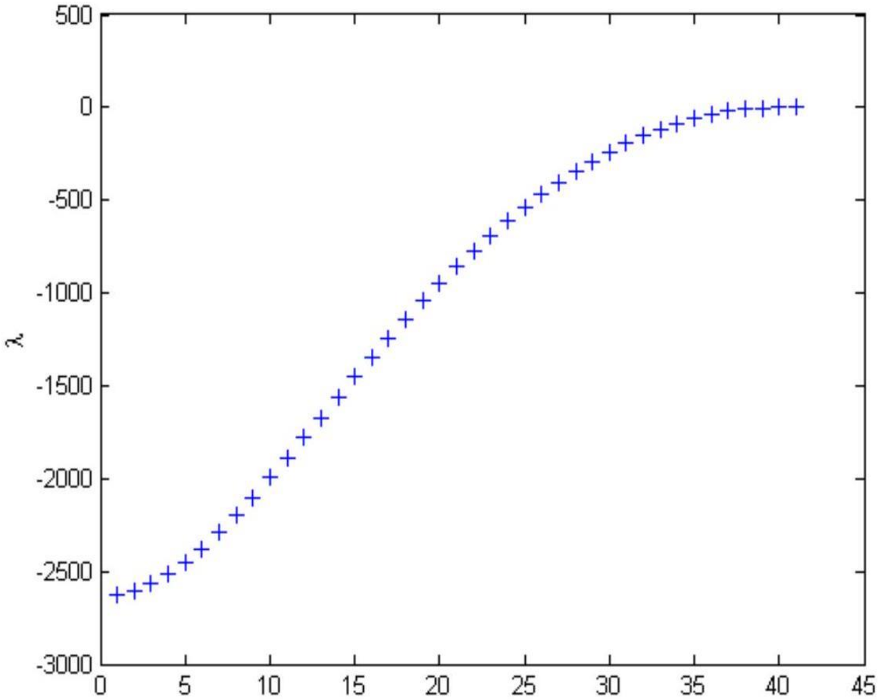

Figure 1. Plot of eigenvalues

TABLE 2 .

(x_i, y_j)	Numerical Solution	Analytical solution	Maximum absolute error (L_∞)
$(-0.9, -0.9)$	0.066822	0.066820	$2.19\text{E}-06$
$(-0.8, -0.8)$	0.205653	0.205651	$2.46\text{E}-06$
$(-0.7, -0.7)$	0.283056	0.283054	$1.57\text{E}-06$
$(-0.6, -0.6)$	0.205651	0.205651	$5.87\text{E}-07$
$(-0.5, -0.5)$	0.000000	0.000000	0.000000
$(-0.4, -0.4)$	0.205651	0.205651	$2.22\text{E}-07$
$(-0.3, -0.3)$	0.283054	0.283054	$3.55\text{E}-07$
$(-0.2, -0.2)$	0.205650	0.205650	$5.80\text{E}-07$
$(-0.1, -0.1)$	0.066820	0.066820	$6.41\text{E}-07$
$(0.0,0.1)$	0.000000	0.000000	0.000000
$(0.2,0.1)$	-0.127100	-0.127099	$1.23\text{E}-06$
$(0.3,0.1)$	-0.174939	-0.174937	$1.72\text{E}-06$
$(0.4,0.1)$	-0.205653	-0.205651	$2.10\text{E}-06$
$(0.5,0.1)$	-0.216236	-0.216234	$2.34\text{E}-06$
$(0.6,0.1)$	-0.205653	-0.205651	$2.46\text{E}-06$
$(0.7,0.1)$	-0.174940	-0.174940	$2.48\text{E}-06$
$(0.8,0.1)$	-0.127102	-0.127099	$2.39\text{E}-06$
$(0.9,0.1)$	-0.066822	-0.066820	$2.19\text{E}-06$

Numerical and exact results of Example 1 at $t = 1.0$ with $N = 100$

Example 2: Consider (1.1) with $a = -1, b = 1, p(t) = 1,$ and $f(u, x, y, t) = \frac{1}{1+u^2} + \varphi(x, y, t)$. The exact solution is given by

$$u(x, y, t) = e^{-t} \cos(\pi x) \cos(\pi y) \quad (5.3)$$

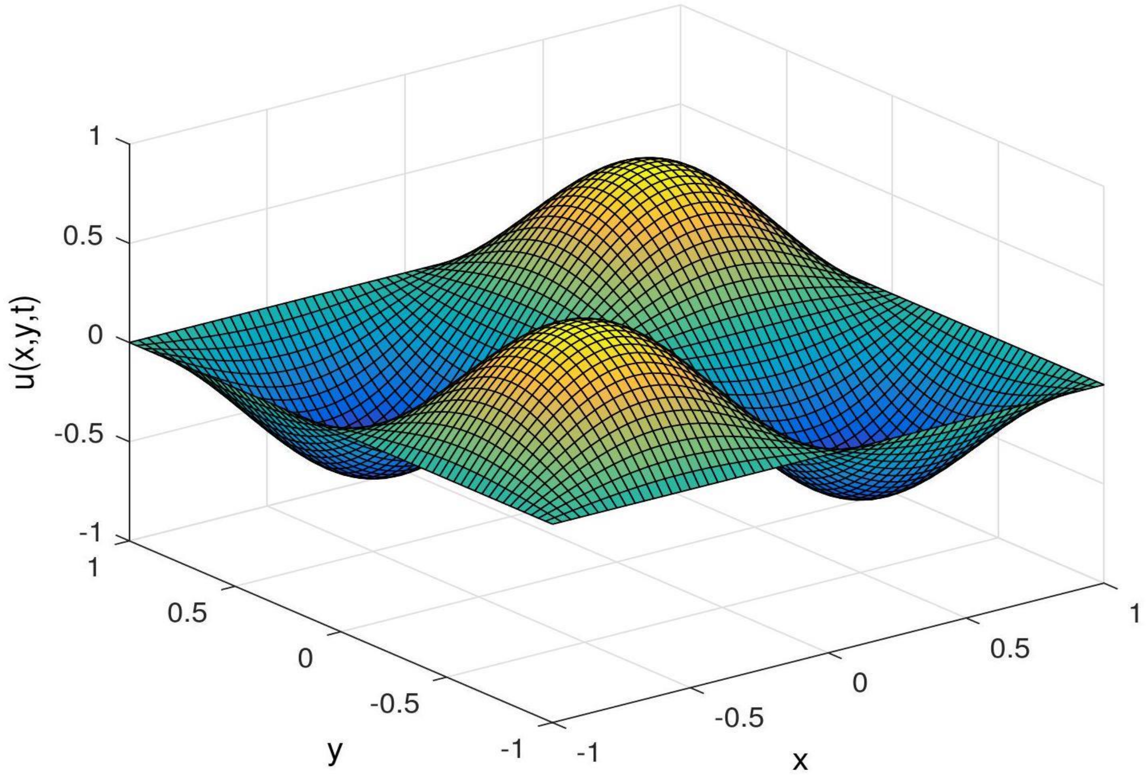

Figure 2. Numerical solutions of Example 1 at $t = 0.5$

TABLE 3 .

t	L_∞ error	L_2 error	CPU Time(s)
1	$3.01\text{E}-04$	$2.92\text{E}-03$	1
2	$1.92\text{E}-04$	$2.25\text{E}-03$	2
5	$1.64\text{E}-05$	$1.85\text{E}-04$	5
10	$3.13\text{E}-07$	$3.32\text{E}-06$	10
20	$2.93\text{E}-11$	$3.01\text{E}-10$	19

Numerical solutions of Example 1 with $N = 20$ and $\Delta t = 0.001$

Here boundary conditions are the Dirichlet conditions.

The initial condition is

$$u(x, y, 0) = \cos(\pi x)\cos(\pi y) \tag{5.4}$$

The function ψ has been found as follows

$$\psi = -e^{-t}\cos(\pi x)\cos(\pi y) + 2\pi^2 e^{-t}\cos(\pi x)\cos(\pi y) - \frac{1}{1 + e^{-2t}\cos^2(\pi x)\cos^2(\pi y)} \tag{5.5}$$

Table 4 presents solutions at $t = 1$ at different knots. Maximum absolute error norms at time levels from $t = 1$ to $t = 20$ have been listed in Table 5 .

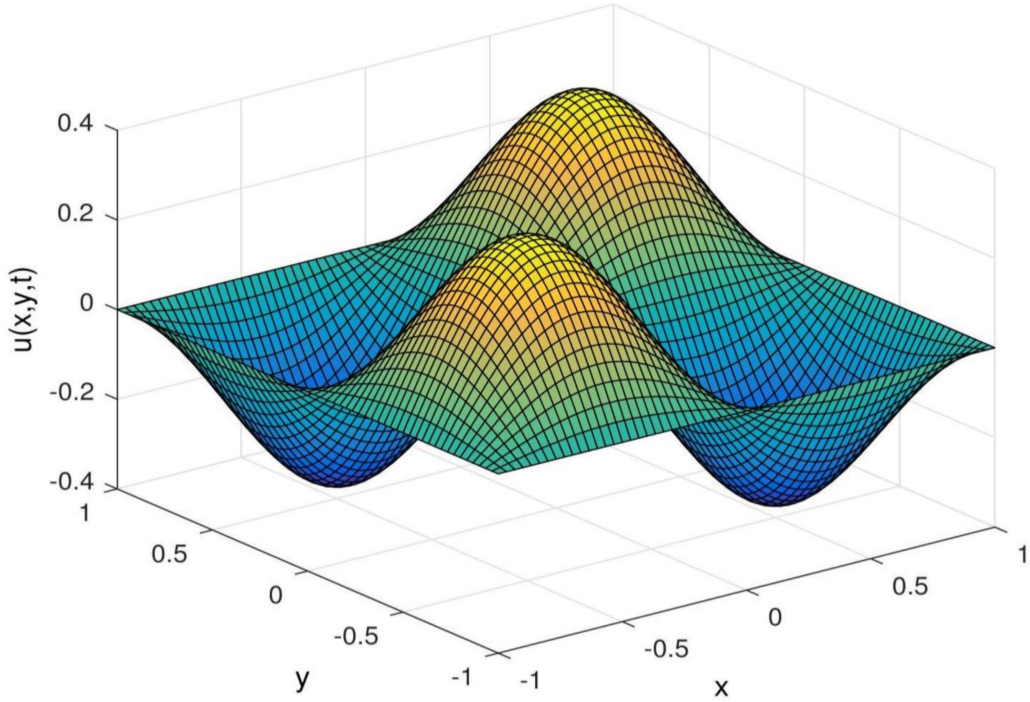

Figure 3. Numerical solutions of Example 1 at $t = 1$

We can observe that CPU time is very less and absolute errors are also very small. Physical behavior of numerical solutions has been depicted in Figures 5 and 6 at $t = 1$. Similar behavior of solution has been depicted by Liu et al. [16].

Example 3: Consider the following exact solution of (1.1)

$$u(x, y, t) = e^{-t} \sin(2\pi x) \sin(2\pi y) \tag{5.6}$$

and following initial condition

$$u(x, y, 0) = \sin(2\pi x) \sin(2\pi y) \tag{5.7}$$

Also $p(t) = \sin (t), a = -1, b = 1, f(u, x, y, t) = \frac{1}{1+u^2} + \psi(x, y, t)$. The boundary conditions have been taken from the analytical solution.

$$\psi = -e^{-t}\sin (2\pi x)\sin (2\pi y) - \frac{1}{1 + e^{-2t} \sin^2(2\pi x) \sin^2(2\pi y)} + 8\pi^2 \sin(t)\, e^{-t} \sin(2\pi x) \sin(2\pi y) \tag{5.8}$$

We performed numerical experiments by taking different values of parameters. Numerical solutions for $t = 1$ have been represented in Table 6. We can observe that absolute errors are very small. Computed solutions have been presented graphically in Figures 7 and 8. Our method is giving very good solutions with very small CPU time.

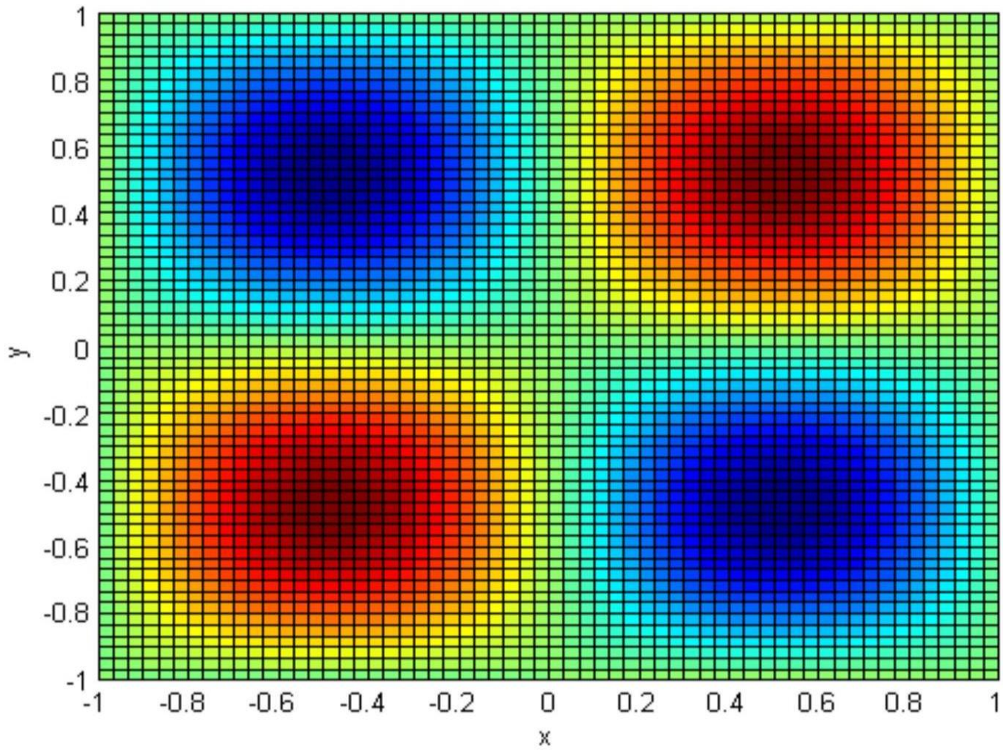

Figure 4. 3D view of solutions of Example 1 at $t = 1$

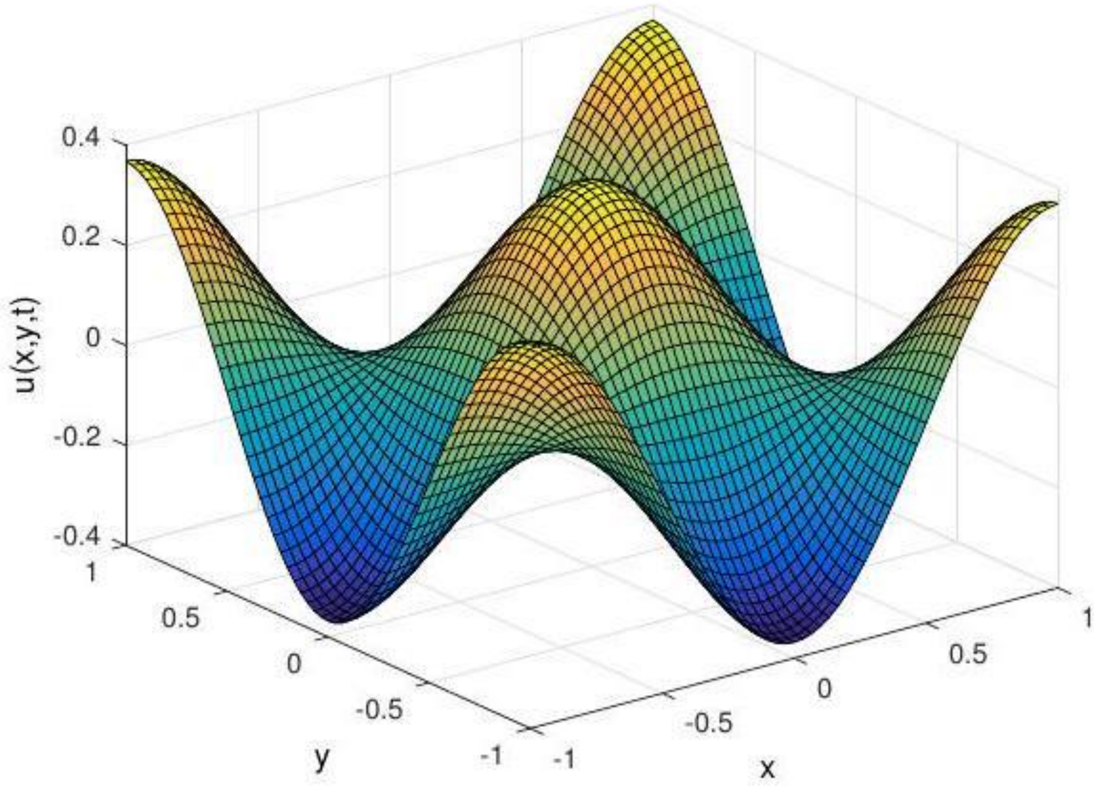

Figure 5. Numerical solutions of Example 2 at $t = 1$

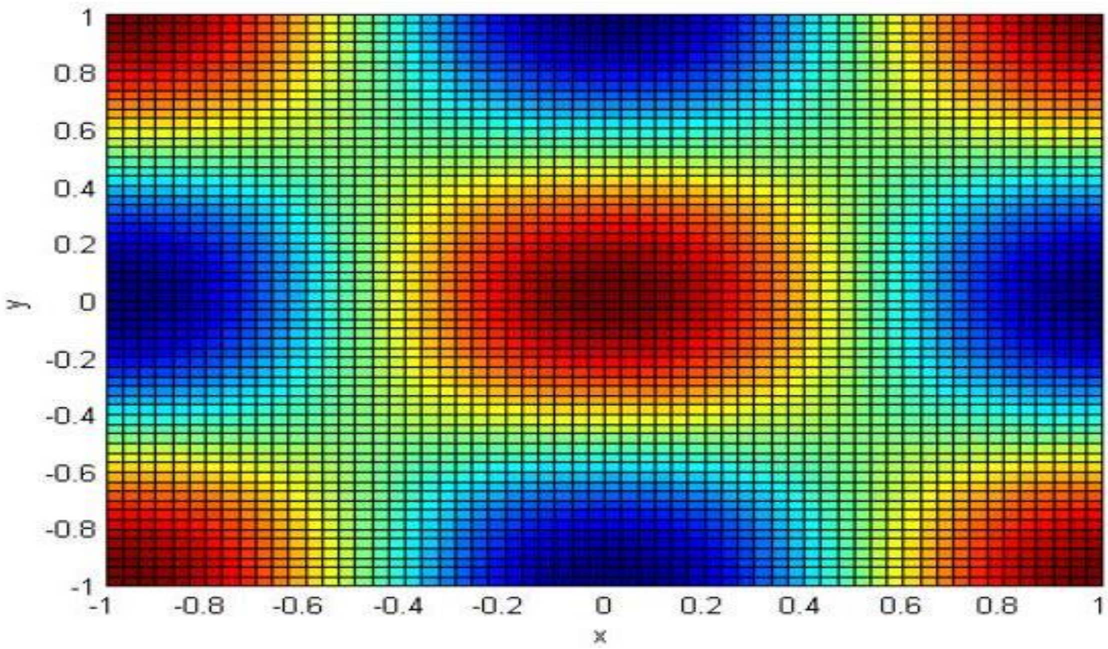

Figure 6. 3D view of solutions of Example 2 at $t = 1$

Example 4: Consider (1.1) with $p(t) = \frac{1}{1+t^2}, f(u, x, y, t) = \frac{1}{1+u^2} + \psi(x, y, t)$. Source function ψ has been taken so that the exact solution is

$$u(x, t) = e^{-t}\cos(2\pi x)\cos(2\pi y) \quad (5.9)$$

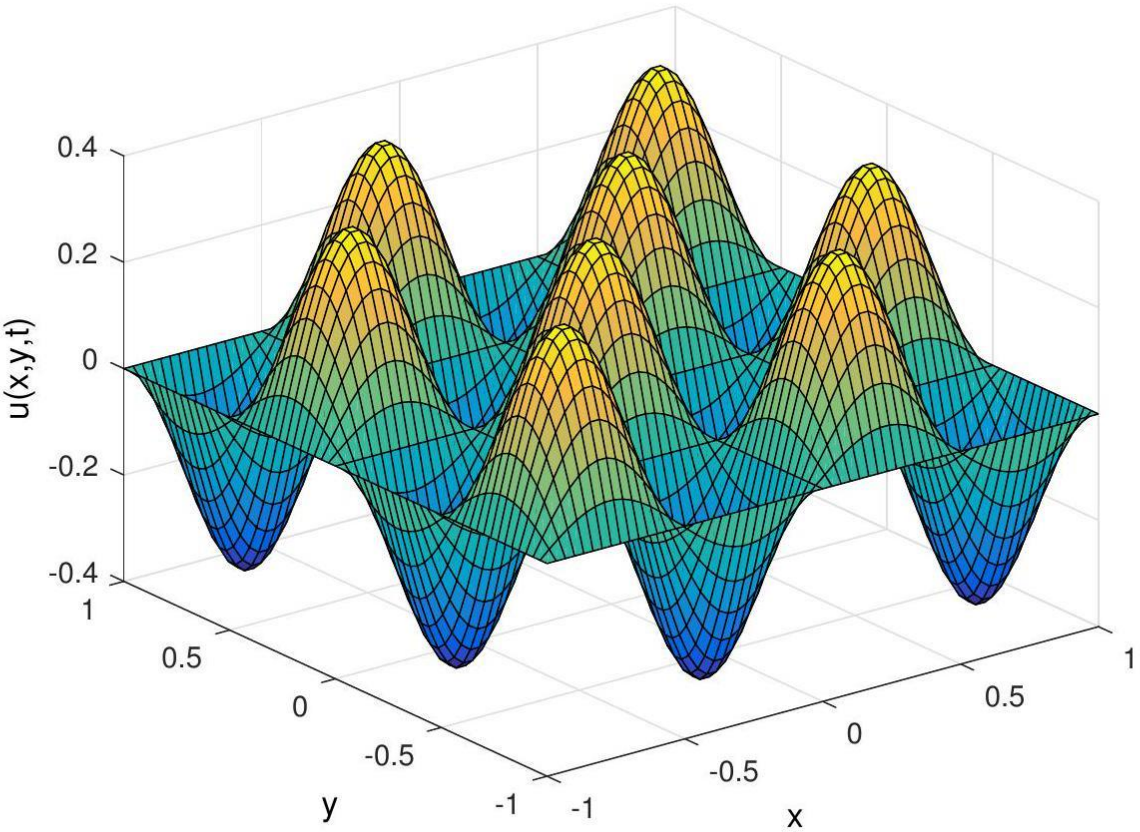

Figure 7. Numerical solutions of Example 3 at $t = 1$

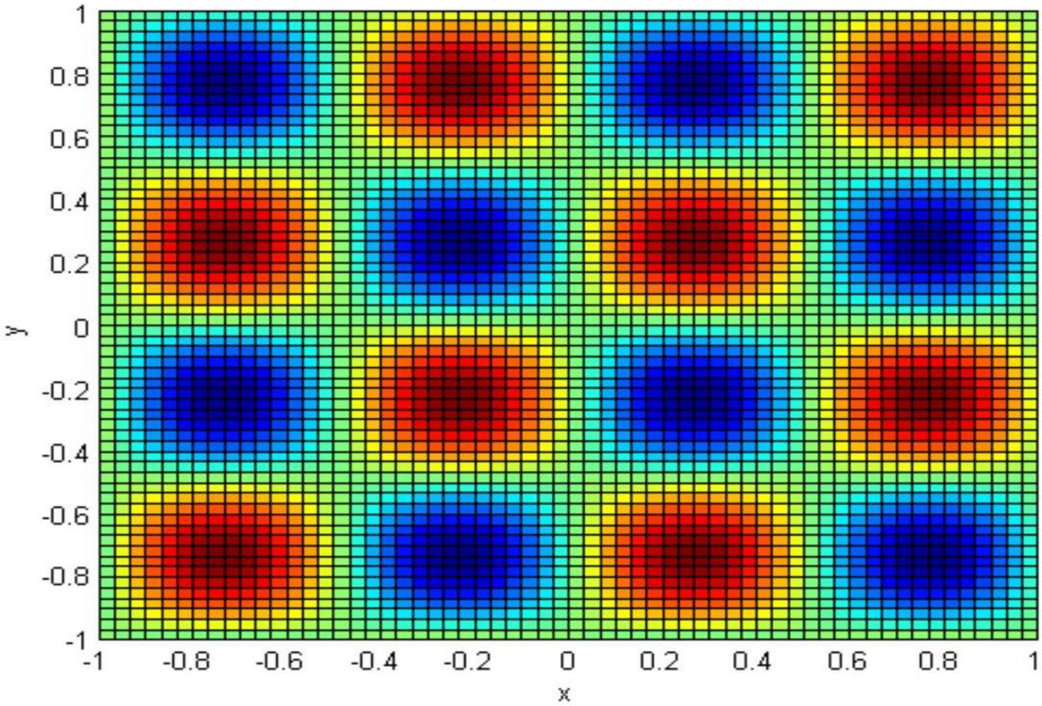

Figure 8. 3 D view of solutions of Example 3 at $t = 1$

TABLE 4 .

(x_i, y_j)	Numerical Solution	Analytical solution	Maximum absolute error (L_∞)
$(-0.9, -0.9)$	0.332729	0.332750	$2.14E-05$
$(-0.8, -0.8)$	0.240766	0.240780	$1.43E-05$
$(-0.7, -0.7)$	0.127092	0.127099	$6.84E-06$
$(-0.6, -0.6)$	0.035128	0.035129	$9.15E-07$
$(-0.5, -0.5)$	0.000000	0.000000	0.000000
$(-0.4, -0.4)$	0.035130	0.035129	$9.72E-07$
$(-0.3, -0.3)$	0.127096	0.127099	$2.81E-06$
$(-0.2, -0.2)$	0.240772	0.240780	$7.78E-06$
$(-0.1, -0.1)$	0.332738	0.332750	$1.19E-05$
$(0.1, 0.1)$	0.332738	0.332750	$1.19E-05$
$(.2, 0.1)$	0.283044	0.283054	$9.65E-06$

(.3,0.1)	-0.205645	-0.205651	$6.16\mathrm{E}-06$
(.4,0.1)	-0.108115	-0.108117	$1.73\mathrm{E}-06$
(.5,0.1)	0.000000	0.000000	0.000000
(.6,0.1)	-0.108109	-0.108117	$8.35\mathrm{E}-05$
(.7,0.1)	-0.205638	-0.205651	$1.32\mathrm{E}-05$
(.8,0.1)	-0.283037	-0.283054	$1.73\mathrm{E}-05$
(.9,0.1)	-0.332730	-0.332750	$2.05\mathrm{E}-05$

Numerical and exact results of Example 2 at $t = 1.0$ with $N = 60$

TABLE 5 .

t	L_∞ error	L_2 error	CPU Time(s)
1	$5.08\mathrm{E}-03$	$1.76\mathrm{E}-02$	3
2	$2.00\mathrm{E}-03$	$7.19\mathrm{E}-03$	5
5	$1.43\mathrm{E}-05$	$3.28\mathrm{E}-04$	12
10	$6.54\mathrm{E}-07$	$2.31\mathrm{E}-06$	23
20	$2.92\mathrm{E}-11$	$1.02\mathrm{E}-10$	46

Numerical solutions of Example 2 with $N = 20$ and $\Delta t 0.001$

so that the source function is

$$\psi = -e^{-t}\cos(2\pi x)\cos(2\pi y) + \frac{8\pi^2 e^{-t}\cos(2\pi x)\cos(2\pi y)}{1+t^2} - \frac{1}{1+e^{-2t}\cos^2(2\pi x)\cos^2(2\pi y)} \quad (5.10)$$

Boundary conditions have been taken from analytical solution. Initial condition is

$$u(x,0) = \cos(2\pi x)\cos(2\pi y) \quad (5.11)$$

Numerical simulations have been carried out for different parameters. It is observed that method is giving very good results (Table 7). Solution profiles have been presented in Figures 9 and 10.

6. Conclusion

Two dimensional semi-linear parabolic equation has been solved by the cubic B-spline differential quadrature method. Weighting coefficients have been evaluated by using cubic B-spline basis functions. Two dimensional parabolic equation is discretized by differential quadrature to get system of ordinary differential equations which is solved by SSPRK-43 method to get the final solution. This is a relatively easy method and uses very less storage. Numerical solutions have been depicted graphically and also presented graphically. It is seen that approximate solutions coincide with the analytical values. The

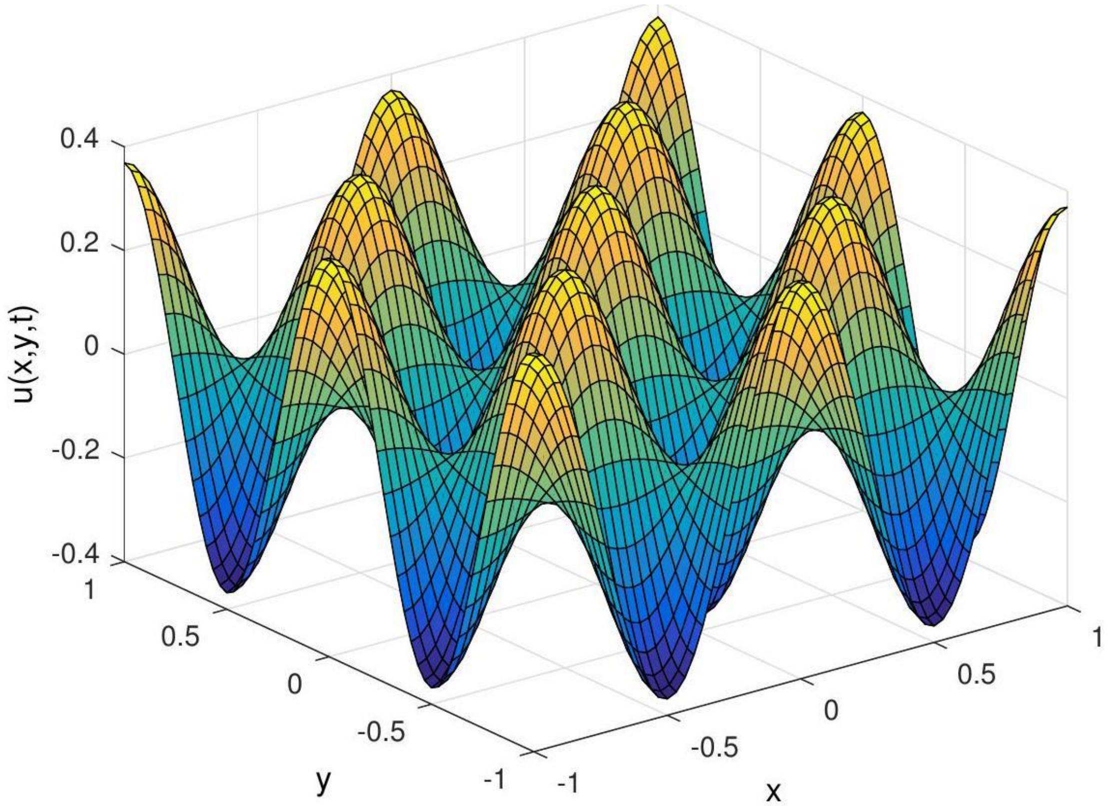

Figure 9. Numerical solutions of Example 4 at $t = 1$

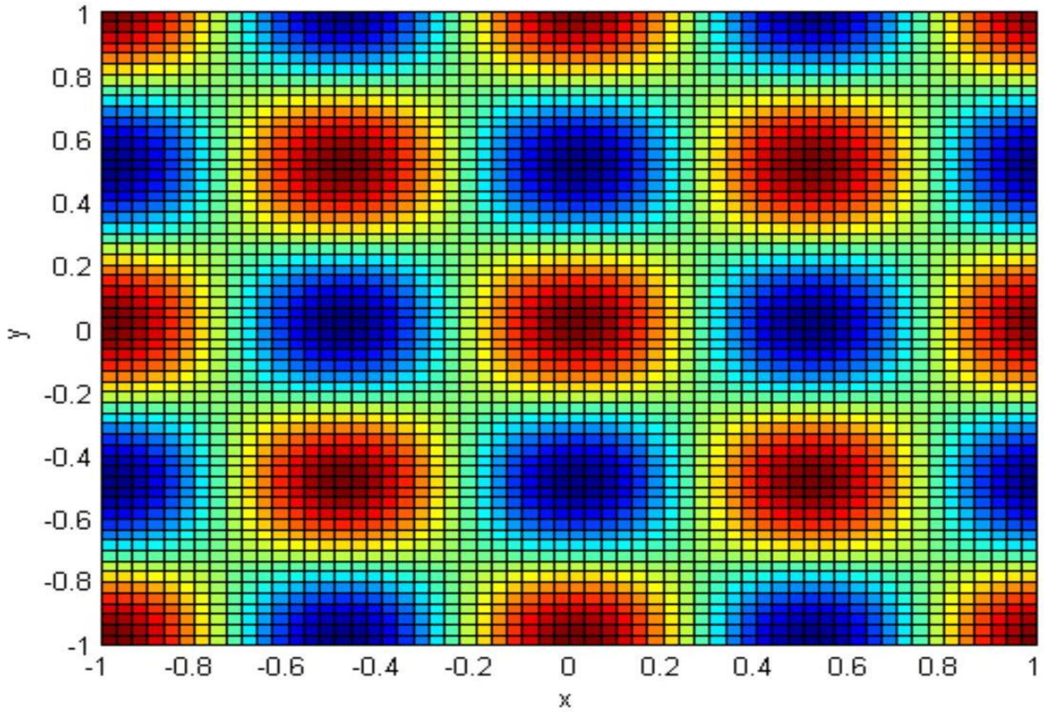

Figure 10. 3D view of solutions of Example 4 at $t = 1$

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

TABLE 6 .

(x_i, y_j)	Numerical Solution	Analytical solution	Maximum absolute error (L_∞)
$(-0.9, -0.9)$	0.127154	0.127099	$5.43\text{E}-05$
$(-0.8, -0.8)$	0.332786	0.332750	$3.58\text{E}-05$
$(-0.7, -0.7)$	0.332762	0.332750	$1.16\text{E}-05$
$(-0.6, -0.6)$	0.127102	0.127099	$2.43\text{E}-05$
$(-0.5, -0.5)$	0.000000	0.000000	0.000000
$(-0.4, -0.4)$	0.127089	0.127099	$9.92\text{E}-06$
$(-0.3, -0.3)$	0.332731	0.332750	$1.94\text{E}-05$
$(-0.2, -0.2)$	0.332732	0.332750	$1.83\text{E}-05$
$(-0.1, -0.1)$	0.127091	0.127099	$7.72\text{E}-06$
$(0.1,0.1)$	0.127123	0.127099	$2.37\text{E}-05$
$(.2,0.1)$	0.205689	0.205651	$3.84\text{E}-05$
$(.3,0.1)$	0.205689	0.205651	$3.84\text{E}-05$
$(.4,0.1)$	0.127122	0.127099	$2.25\text{E}-05$
$(.5,0.1)$	0.000000	0.000000	0.000000
$(.6,0.1)$	-0.127129	-0.127099	$2.94\text{E}-05$
$(.7,0.1)$	-0.205699	-0.205651	$4.84\text{E}-05$
$(.8,0.1)$	-0.205707	-0.205651	$5.58\text{E}-05$
$(.9,0.1)$	-0.127154	-0.127099	$5.48-05$

Numerical and exact results of Example 3 at $t = 1.0$ with $N = 60$

TABLE 7 .

(x_i, y_j)	Numerical Solution	Analytical solution	Maximum absolute error (L_∞)
$(-0.9, -0.9)$	0.240747	0.240780	$3.37\text{E}-05$
$(-0.8, -0.8)$	0.035126	0.035129	$3.34\text{E}-06$
$(-0.7, -0.7)$	0.035131	0.035129	$1.85\text{E}-06$
$(-0.6, -0.6)$	0.240770	0.240780	$1.03\text{E}-05$
$(-0.5, -0.5)$	0.367859	0.367879	$2.04\text{E}-05$
$(-0.4, -0.4)$	0.240763	0.240780	$1.74\text{E}-05$
$(-0.3, -0.3)$	0.035120	0.035129	$9.44\text{E}-06$
$(-0.2, -0.2)$	0.035119	0.035129	$1.05\text{E}-05$
$(-0.1, -0.1)$	0.127091	0.127099	$7.72\text{E}-06$
$(.1, 0.1)$	0.240749	0.240780	$3.16\text{E}-05$
$(.2, 0.1)$	0.091957	0.091970	$1.30\text{E}-05$
$(.3, 0.1)$	-0.091960	-0.091969	$1.01\text{E}-05$
$(.4, 0.1)$	-0.240751	-0.240780	$2.90\text{E}-05$
$(.5, 0.1)$	-0.297584	-0.297621	$3.64\text{E}-05$
$(.6, 0.1)$	-0.240750	-0.240780	$2.91\text{E}-05$
$(.7, 0.1)$	-0.091959	-0.091970	$1.14\text{E}-05$
$(.8, 0.1)$	-0.091958	-0.091970	$1.18\text{E}-05$
$(.9, 0.1)$	-0.240747	-0.240780	$3.37\text{E}-05$

Numerical and exact results of Example 4 at $t = 1.0$ with $N = 80$

accuracy of the method has been measured by evaluating maximum absolute error norm and it is found that method is performing very efficiently. Differential quadrature method is a potential alternative to finite element and finite difference methods. We hope for further development in the applications of differential quadrature method to a fairly wide range of problems.

Conflicts of interest statement - The authors declare that they have no conflict of interest.

References

[1] G. N. Milstein and M.V. Tretyakov(1999), Numerical alsorithms for semilinear parabolic equations with small parameter based on approximation of stochastic equations, Mathematics of Computation, 69, pp.237-267

[2] Marlis Hochbruckand Alexander Ostermann(2005), Explicit Exponential Runge Kutta Methods for Semilinear Parabolic Equations, Siam J. of Num. anal.,43, pp.1069-1090

[3] Mehdi Dehghan(1999),Numerical methods for two dimensional parabolic inverse problemswith energy overspecification,Int. J. Comput. Mathe.,77,pp.441-455

[4] Idris Dag, Bulent Saka and Alper Korkmaz (2009), Numerical Methods for Partial Differential Equations, 26, pp. $544 - 560$

[5] Alper Korkmaz, A. Murat Aksoy and Idris Dag (2011), Quartic B-spline differential quadrature method , International J. Nonlin. Sci.,11, pp. 403-411

[6] R.C. Mittal and Rachna Bhatia (2014), Numerical study of two dimensional hyperbolic telegraph equation by modified B-spline differential quadrature method, Appl. Mathe. Comput.,244, pp.976-997

[7] R. C. Mittal and Rajni Rohila (2016), Numerical Simulation of Reaction-Diffusion Systems by Modified Cubic B-Spline Differential Quadrature Method, Chaos, Solitons and Fractals, Vol-92, pp. 9-19

[8] R.C. Mittal and Rajni Rohila (2016), A study of one dimensional nonlinear diffusion equations by Bernstein polynomial based differential quadrature method, J. Mathe. Chemistry, 55,pp. 673-695

[9] Ozlem Ersoy, Idiris Dag (2016),The Exponential Cubic B-Spline Collocation Method for the Kuramoto-Sivashinsky Equation,Filomat 30:3, pp.853861

[10] W.K Zahra((2017), Trigonometric B-Spline Collocation Method for Solving PHI-Four and AllenCahn Equations,Mediterr. J. Math. DOI 10.1007/s00009-017-0916-8

[11] R.C. Mittal, R.K. Jain (2012), 'Numerical solutions of nonlinear Burgers equation with modified cubic B-splines collocation method', Appl. Math. Comput.,218,pp. 7839-7855.

[12] Boor C.de. A practical Guide to Splines. Springer-Verlag, 1978.

[13] Chang, Shu (2000), Differential Quadrature and its Application in Engineering, Athenaeum Press Ltd., Great Britain.

[14] G.Guraslan, M. Sari (2011), Numerical solutions of linear and nonlinear diffusion equations by a differential quadrature method (DQM), Int. J. Numer.Methods Biomed. Eng. , 27, pp.6977.

[15] R.J.Spiteri and S.J. Ruuth (2002), A new class of optimal high order stromg stability preserving time stepping schemes, SIAM Journal of Numerical Analysis,40,pp.469-491

[16] Wenjie Liu, Jiebao Sun and BoyingWu (2016), Space-time spectral method for two-dimensional semilinear parabolic equations, Mathematical Methods in Applied Science, 39 , pp. 16461661

[17] S. Tomasiello(2003), Stability and accuracy of the iterative differential quadrature method, Int. J. Numer. Methods Eng. 58 (6) 12771296 .

[18] S. Tomasiello (2010), Numerical solutions of the BurgersHuxley equation by the IDQ method, Int. J. Comput. Math. 87 (1) 129140 .

[19] S. Tomasiello (2011), Numerical stability of DQ solutions of wave problems, Numer. Algorithms 57 (3) 289312