Hindawi
Discrete Dynamics in Nature and Society
Volume 2023, Article ID 5102374, 8 pages
https://doi.org/10.1155/2023/5102374

Research Article

Numerical Simulation and Dynamics of Burgers' Equation Using the Modified Cubic B-Spline Differential Quadrature Method

Geeta Arora,[1] **Shubham Mishra**,[1] **Homan Emaifar**,[2] **and Masoumeh Khademi**[2]

[1]*Department of Mathematics, School of Chemical Engineering and Physical Sciences, Lovely Professional University, Phagwara, Punjab, India*
[2]*Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran*

Correspondence should be addressed to Homan Emaifar; homan_emadi@yahoo.com

Received 14 September 2022; Revised 20 November 2022; Accepted 17 December 2022; Published 29 March 2023

Academic Editor: Mehmet Emir Koksal

In the present work, a numerical approach using the Crank–Nicolson scheme along with the modified cubic B-spline differential quadrature (CN-MCDQ) method is proposed to find the numerical approximations to Burgers' equation. After applying the well-known Crank–Nicolson technique, Burgers' equation is solved in this study by using the differential quadrature approach to approximate the derivatives that lead to a system of equations to be solved. When compared to other methods for obtaining numerical solutions, the proposed method is shown to be efficient and easy to implement while still providing accurate results. The obtained results are in agreement with the earlier available approaches and are even better in comparison in terms of less domain partition. Three test problems were used to evaluate the methodology, and the results are tabulated and graphically shown below.

1. Introduction

Differential equations play a significant part in applied mathematics and are involved in various computations in wave motion, wave distribution, electromagnetics, network design, telecommunications, electronic dynamics, fluid dynamics, and many related branches of engineering and sciences [1]. Researchers are looking for new methods for solving the modelled differential equation as a result of mathematical modelling utilising various analytical and numerical methodologies. However, it is not always possible that the differential equation can be solved by the current analytical techniques to yield the exact solution. There comes the role of numerical techniques that can help in solving differential equations with the requisite accuracy.

With the growth of technology, there are numerous available choices for finding the numerical solution of differential equations using software like MATLAB, Maple, and Mathematica. Researchers are continuously working on the modification of the numerical methods to deliver the solution efficiently and precisely. One such attempt to adapt the computational numerical technique to solve the nonlinear differential equations is provided in this work.

Burgers' equation is a basic partial differential equation that appears in many applications, including classical dynamics, nonlinear harmonics, plasma physics, and traffic flow [1]. Bateman [2] was the first researcher to present this equation, and Burgers [3] later examined it for its possible solution. There are various well-known numerical methods reported in the literature to solve Burgers' equation. Some of the important numerical methods in the last few years include but are not limited to the differential quadrature method [4–8], the finite element method [9, 10], the collocation approach [11–13], and the finite difference approach [14–18].

Burgers' equation in its general form is given as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = v\frac{\partial^2 u}{\partial x^2}, \tag{1}$$

where v depicts the kinetic viscosity or the diffusion coefficient. This form of the equation is called the viscous

Burgers' equation and is known as the inviscid Burgers' equation in its absence.

The differential quadrature method is a numerical approach that was initially proposed to find the solution to the partial differential equations by Bellman et al. [19]. Quan and Chang [20] improved it further to compute the weighted coefficient. To determine weighing coefficients, a variety of test functions are available, including the sine function, spline function, Lagrange's polynomial, and radial basis functions. In the current study, the differential quadrature approach uses cubic B-spline basis functions. B-splines are one of the unique spline functions that can be used to compute the required linear combination to generate a piece-wise polynomial. Due to the fact that each B-spline basis function of order m is normally nonzero over at most m consecutive intervals and zero everywhere, each of them offers its own unique computational benefit. Cubic B-spline basis functions, which have been used in the past to solve various physical models, have specific advantages over other basis functions due to their smoothness and capacity to withstand local occurrences.

In the present work, the Crank–Nicolson technique is employed to solve Burgers' equation, followed by DQM with a modified form of the cubic B-spline basis function to remove extra knot points. The equation is thus reduced to a system of equations by utilising the differential quadrature technique to approximate the spatial derivatives. The next step consists of computing the solution of the produced system of equations using MATLAB programming in an iterative way. Since the present method uses the concept of differential quadrature after discretizing the domain, it acts as a tool to reduce the computational complexity of the collocation approach. The collocation method usually requires complex algebraic manual calculation.

This is how the research is presented: in the first section, an introduction to the scheme and its significant contributions in several areas are presented. Following this, section two includes an explanation of the differential quadrature method (DQM) and its involvement with the modified cubic B-spline with the discussion of the Crank–Nicolson scheme in Section 3. The numerical test problems of the equation with different parameters and time levels are presented in Section 4, followed by observations on the utility of the proposed scheme and its future implementation.

2. The Differential Quadrature Method (DQM)

DQM has been extensively applied for computation in science and engineering. In DQM, the values of function u at grid points are combined to approximate the value of the derivative of the smooth function at grid points x_i. The grid distribution of the finite interval $[a, b]$ using the formula $a = x_1 < x_2 < \ldots < x_N = b$ is considered. DQM can be provided in a simple form for derivatives as follows, where p_{ij} and q_{ij} are the weighting coefficients that are used to derive the first two derivatives with respect to x, respectively:

$$u_x(x_i, t) = \sum_{j=1}^{N} p_{ij} u(x_j, t), u_{xx}(x_i, t) = \sum_{j=1}^{N} q_{ij} u(x_j, t). \tag{2}$$

The preceding relationships define the cubic B-splines $\delta_j(x)$ in the following way:

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

The equation above defines the cubic B-spline basis functions at the knots, where $\{\delta_0, \delta_1, \ldots\ldots, \delta_N, \delta_{N+1}\}$ creates a basis over the considered domain. To compute the values of cubic B-splines and its derivatives at the nodal locations, following formulas can be used:

$$\delta_j(x_m) = \begin{cases} 1 & m = j-1, j+1 \\ 4 & k = j \\ 0 & \text{otherwise} \end{cases} ; \quad \delta_j'(x_m) = \begin{cases} 3h^{-1} & m = j-1 \\ -3h^{-1} & m = j+1 \\ 0 & \text{otherwise,} \end{cases}$$
$$\delta_j''(x_m) = \begin{cases} 6h^{-2} & m = j-1 \\ -6h^{-2} & m = j+1 \\ -12h^{-2} & m = j \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Further adjustments are made in the basis functions to produce a diagonally dominating matrix system of equations. Additionally, it removes any excess knots that appeared in the image as ghost points.

$$\begin{cases} \psi_1(x) = \delta_1(x) + 2\delta_0(x) \\ \psi_2(x) = \delta_2(x) - \delta_0(x) \\ \psi_j(x) = \delta_j(x)\, for\, j = 3, \ldots, N-2 \\ \psi_{N-1}(x) = \delta_{N-1}(x) - \delta_{N+1}(x) \\ \psi_N(x) = \delta_N(x) + 2\delta_{N+1}(x). \end{cases} \quad (5)$$

Using the DQM approach and modified cubic basis functions, equation (3) is utilised to estimate the first-order derivative. Consequently, the first derivative at the grid points can be expressed as

$$\psi_k'(x_i) = \sum_{j=1}^{N} p_{ij}\psi_k\left(x_j\right), k = 1, 2, \ldots, N, \quad (6)$$

where p_{ij} are the unknown weighting coefficients to obtain and $\psi_k(x_j)$ are the modified cubic basis functions. Substituting the values in equation (6) for $i = 1$ leads to the following system of equations:

$$X\vec{p}[i] = \vec{Y}[i], i = 1, 2, \ldots, N, \quad (7)$$

where X depicts the coefficient matrix presented as follows:

$$\begin{bmatrix} 6 & 1 & & & & & \\ 0 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & 4 & 1 & \\ & & & & 1 & 4 & 0 \\ & & & & & 1 & 6 \end{bmatrix}, \quad (8)$$

with the weighting coefficients in a column form as $p[i] = [p_{i1}, p_{i2}, \ldots, p_{iN}]^T$ and the coefficient vector $\vec{Y}[i] = [y_{i1}, y_{i2}, \ldots, y_{iN}]^T$ corresponds to grid point x_i, $i = 1, 2, \ldots, N$ that can be evaluated as follows:

$$Y[1] = \begin{bmatrix} \frac{-6}{h} \\ \frac{6}{h} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, Y[2] = \begin{bmatrix} \frac{-3}{h} \\ 0 \\ \frac{3}{h} \\ \vdots \\ \vdots \\ 0 \end{bmatrix}, \ldots, Y[N-1] = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ \frac{-3}{h} \\ 0 \\ \frac{3}{h} \end{bmatrix}, Y[N] = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{-6}{h} \\ \frac{6}{h} \end{bmatrix}. \quad (9)$$

The resulted tridiagonal system of equations provides the weighting coefficients $p_{i1}, p_{i2}, \ldots, p_{iN}$ for $i = 1, 2, \ldots, N$. Using the obtained coefficients p_{ij}, the coefficients q_{ij} for $i = j = 1$ to N can be evaluated as

$$q_{ij} = 2p_{ij}\left(p_{ii} - \frac{1}{x_i - x_j}\right) for\, i \neq j, q_{ii} = -\sum_{i=1, i\neq j}^{N} q_{ij}. \quad (10)$$

3. Crank–Nicolson Scheme

The Crank–Nicolson scheme is a finite difference approach for numerical simulation of the differential equations. From last several years, it is implemented to solve numerous well-known equations using the collocation approach. The purpose of this scheme is the discretization of the derivative with the average of the values taken at the two successive time levels. In this work, Burgers' equation is solved by applying DQM for approximating the derivatives after implementation of the well-known Crank–Nicolson scheme on the time derivative as $u_t = (u^{n+1} + u^n)/(\Delta t)$.

4. Scheme Implementation

Consider Burgers' equation (1) in a simplified form as

$$u_t = \nu u_{xx} - u u_x. \quad (11)$$

On discretizing the derivative of u with respect to t using the finite forward difference approach and applying the Crank–Nicolson scheme on the right-hand side terms generate the system as follows:

$$\frac{u^{n+1} - u^n}{\Delta t} = \nu\left[\frac{(u_{xx})^n + u_{xx}^{n+1}}{2}\right] - \frac{(uu_x)^n + (uu_x)^{n+1}}{2},$$

$$u^{n+1} - u^n = \frac{\nu\Delta t}{2}\left[(u_{xx})^n + (u_{xx})^{n+1}\right] - \frac{\Delta t}{2}\left[(uu_x)^n + (uu_x)^{n+1}\right]. \quad (12)$$

Applying the well-known quasi-linearization to linearize the nonlinear part of the equation produces $(uu_x)^{n+1}$ as $(uu_x)^{n+1} = u^n u_x^{n+1} + u^{n+1}u_x^n - (uu_x)^n$.

Thus, the system can be simplified, considering $\alpha = \nu\Delta t/2$; $\beta = \Delta t/2$, and hence reduces to the following form:

$$u^{n+1} = u^n + \alpha\left[u_{xx}^n + u_{xx}^{n+1}\right] - \beta\left[u^n u_x^{n+1} + u^{n+1}u_x^n\right]. \tag{13}$$

On separating the values of u at different time levels,

$$u^{n+1}\left(1 + \beta u_x^n\right) - \alpha u_{xx}^{n+1} + \beta u_x^{n+1}u^n = u^n + \alpha u_{xx}^n. \tag{14}$$

The values of the derivatives are obtained using DQM in the above system for the $(n+1)^{\text{th}}$ time level, followed by the leads to the following system of equation for $i = 1$ to N:

$$\left(1 + \beta A_i^n\right)u_i^{n+1} - \alpha\sum_{j=1}^{N} q_{ij}u_j^{n+1} + \beta u_i^n\sum_{j=1}^{N} p_{ij}u_j^{n+1} = R.H.S, \tag{15}$$

where $A_i^n = \sum_{j=1}^{N} p_{ij}u(x_j,t)^n$ is the approximation of the derivative at the n^{th} level. The obtained system of equation can be visualized as follows: $JU^{n+1} = R^n$.

Here, the matrix J can be discussed as follows:

$$J_{ij} = \begin{cases} 1 + \beta A_i + \beta u_i P_{ij} - \alpha q_{ij}, & \text{for } i = j \\ \beta u_i P_{ij} - \alpha q_{ij}, & \text{for } i \neq j, \end{cases} \tag{16}$$

and $R^n = u^n + \alpha\sum_{j=1}^{N} q_{ij}u(x_j,t)$ can be then solved by any known method for solution of system of linear equations. Here, the Gauss-elimination method is implemented in an iterative way.

5. Convergence Analysis

To analyse the convergence of the scheme, let $\overline{U}$ be the modified cubic B-spline interpolant of $u \in C^6[x_L, x_R]$. It satisfies the following interpolation condition [21]:

$$\overline{U}(x_i) = u(x_i), i = 1, 2, \ldots\ldots, n, \tag{17}$$

and end conditions given as

$$\overline{U}''(x_L) = u''(x_L) - \frac{h^2}{12}u^4(x_L), \overline{U}''(x_R) = u''(x_R) - \frac{h^2}{12}u^4(x_R). \tag{18}$$

Thus,

$$\overline{U}'(x_i) = u'(x_i) + o\left(h^4\right), 0 \le i \le n, \tag{19}$$

$$\overline{U}''(x_i) = u''(x_i) - \frac{h^2}{12}u^4(x_i) + 0\left(h^2\right), 0 \le i \le n. \tag{20}$$

Theorem 1. *Let $u \in C^6[x_L, x_R]$ be the approximations of r^{th} derivatives of u ($r = 1, 2$), as reported in (2) have the following error bounds:*

$$u'\left(x_j\right) = U'\left(x_j\right) + o\left(h^4\right), \tag{21}$$

and

$$u''\left(x_j\right) = U''\left(x_j\right) + o\left(h^2\right). \tag{22}$$

Proof. Let $\overline{U}$ be the modified cubic B-spline interpolant of $u \in C^6[x_L, x_R]$ that satisfies the interpolation and end conditions as discussed above. The triangle inequality yields

$$\left|u_i' - U_i'\right| \le \left|u_i' - \overline{U}_i'\right| + \left|\overline{U}_i' - U_i'\right|. \tag{23}$$

On utilizing (19) and (20), we get

$$\left|u_i' - U_i'\right| \le \left|u_i' - \overline{U}_i'\right| + O\left({h_y}^4\right). \tag{24}$$

Now,

$$\begin{aligned} \overline{U}_i' - U_i' &= \sum_{k=1}^{n}\delta_k\left(\varphi_k'\right)_i - \sum_{j=1}^{n}P_{ij}^{(1)}u\left(x_j,t\right) \\ &= \sum_{k=1}^{n}\delta_k\sum_{j=1}^{n}P_{ij}^{(1)}\left(\varphi_k'\right)_i - \sum_{j=1}^{n}P_{ij}^{(1)}u\left(x_j,\tau\right) \\ &= \sum_{j=1}^{n}P_{ij}^{(1)}\left(\sum_{k=1}^{n}\delta_k\left(\varphi_k'\right)_i - u\left(x_j,\tau\right)\right) \\ &= \sum_{j=1}^{n}P_{ij}^{(1)}\left(U\left(x_j,t\right) - u\left(x_j,\tau\right)\right). \end{aligned} \tag{25}$$

Utilizing the interpolations conditions leads to

$$\overline{U}_i' - U_i' = 0. \tag{26}$$

After utilizing (26), equation (24) transforms into

$$\left|u_i' - U_i'\right| \le O\left({h_y}^4\right). \tag{27}$$

In the similar manner, the result of (22) can be proved. □

6. Numerical Experiments

In this subsection, three numerical examples are presented using the proposed blended method to solve Burgers' equation numerically with varying sets of parameters. Standard errors are used to evaluate the performance of various approaches.

6.1. 1st Test Problem. Consider equation (1) to be solved in the domain [0, 8] with boundary condition taken from the exact solution [6], given as

Table 1: Solution of the equation at different time levels for the first test problem.

x	$T = 1.5$		$T = 3.0$		$T = 4.5$	
	Exact	Present	Exact	Present	Exact	Present
1.0	0.265771	0.265604	0.118804	0.118783	0.071869	0.071864
2.0	0.261421	0.261363	0.167623	0.167591	0.113387	0.113376
3.0	0.088070	0.088151	0.127382	0.127381	0.109491	0.109486
4.0	0.011859	0.011889	0.057975	0.057996	0.073605	0.073612
5.0	0.000741	0.000744	0.016735	0.016750	0.035717	0.035726
6.0	0.000023	0.000023	0.003238	0.003243	0.012919	0.012908
7.0	0.000000	0.000000	0.000433	0.000432	0.003582	0.003452

Table 2: Errors of Burgers' equation at different time levels for the first test problem.

	$T = 1.5$		$T = 3.0$		$T = 4.5$	
	Present	[8]	Present	[8]	Present	[8]
$\mathbf{L_2}$	$1.9281e-04$	$2.11e-03$	$4.9057e-05$	$1.64e-03$	$3.4738e-04$	$1.38e-03$
$\mathbf{L_\infty}$	$1.7771e-04$	$3.18e-03$	$3.4300e-05$	$1.43e-03$	$5.5472e-04$	$0.89e-03$

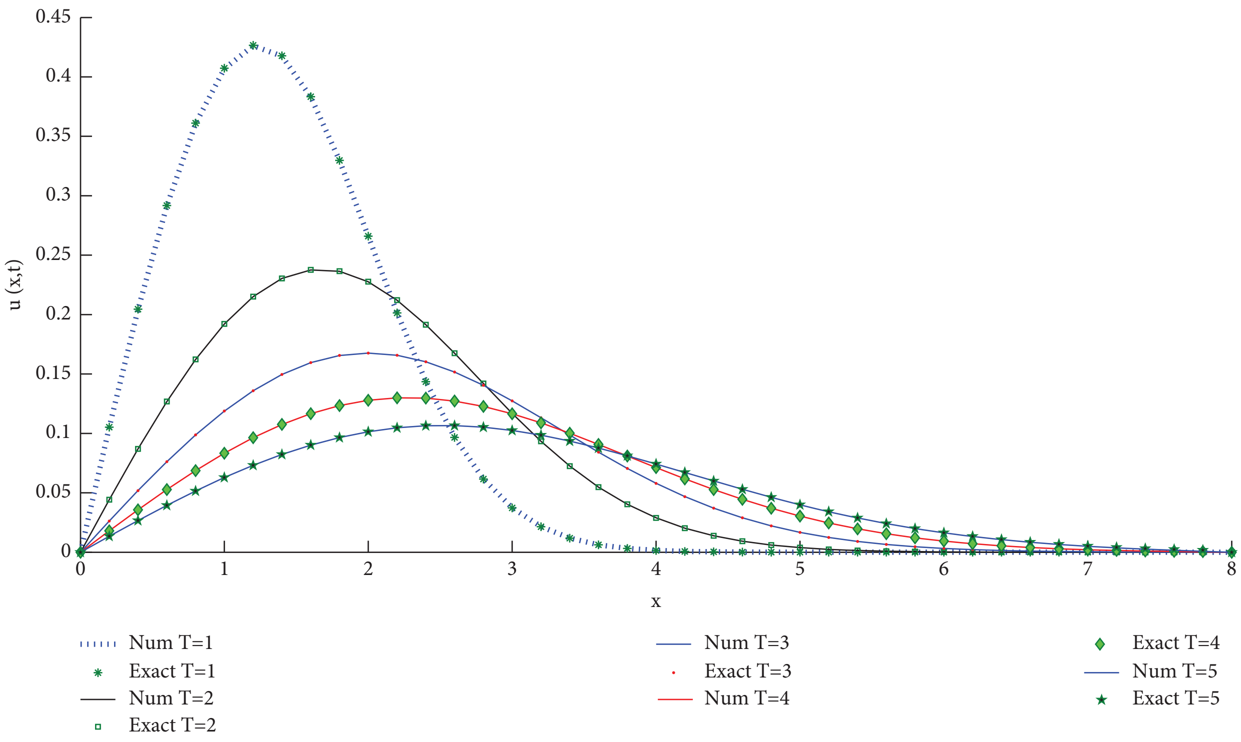

Figure 1: Numerical solution of the equation at different time levels.

$$u(x,t) = \frac{x/t}{1 + \sqrt{(t/g)}\exp\left(x^2/4\nu t\right)},\ g = \frac{0.125}{\nu} \text{ for } t \geq 1$$

$$\text{and the initial solution as } u(x,0) = \frac{x}{1 + \exp\left(x^2 - 0.25/4\nu\right)}. \tag{28}$$

The numerical solution has been obtained for the high values of $\nu = 0.5$ at different time levels for $t \leq 5$. The domain is taken as per the literature review. There is no limitation of changing the domain but the physical behaviour can be depicted in the considered domain. Table 1 presents the comparison of the numerical and exact solutions at different points with respect to time. In Table 2, the errors are estimated at different time levels for time step, $k = 0.01$, and the number of domain partition as 41, thus $h = 0.2$, which is very less as compared to the domain. Figure 1 exhibits the

Table 3: Solution of the equation at time $t = 0.1$ for the second test problem.

			$\nu = 10^{-2}$		$\nu = 10^{-4}$		$\nu = 10^{-6}$	
$t = 0.1$	h	k	L_2	L_∞	L_2	L_∞	L_2	L_∞
Present	0.5	10^{-2}	$2.23e-04$	$2.23e-04$	$2.38e-08$	$2.38e-08$	$2.39e-12$	$2.39e-12$
[13]	0.025	10^{-3}	$3.55e-03$	$4.41e-03$	$3.74e-07$	$4.62e-07$	$3.74e-11$	$4.62e-11$
[4]	0.1	10^{-2}	$3.41e-03$	$3.89e-03$	$3.56e-07$	$4.11e-07$	$3.56e-11$	$4.11e-11$
[8]	0.025	10^{-3}	$3.47e-03$	$4.26e-03$	—	—	—	—

Table 4: Solution of the equation at time $t = 1$ for the second test problem.

			$\nu = \mathbf{10^{-2}}$		$\nu = \mathbf{10^{-4}}$		$\nu = \mathbf{10^{-6}}$	
$t = 1$	h	k	L_2	L_∞	L_2	L_∞	L_2	L_∞
Present	0.5	10^{-2}	$2.16e-03$	$2.16e-03$	$2.39e-07$	$2.39e-07$	$2.39e-11$	$2.39e-11$
[13]	0.025	10^{-3}	$2.66e-02$	$3.13e-02$	$3.72e-06$	$4.61e-06$	$3.74e-10$	$4.62e-10$
[4]	0.1	10^{-2}	$2.63e-02$	$2.92e-02$	$3.55e-06$	$4.09e-06$	$3.56e-10$	$4.11e-10$
[8]	0.025	10^{-3}	$2.63e-02$	$3.08e-02$	—	—	—	—

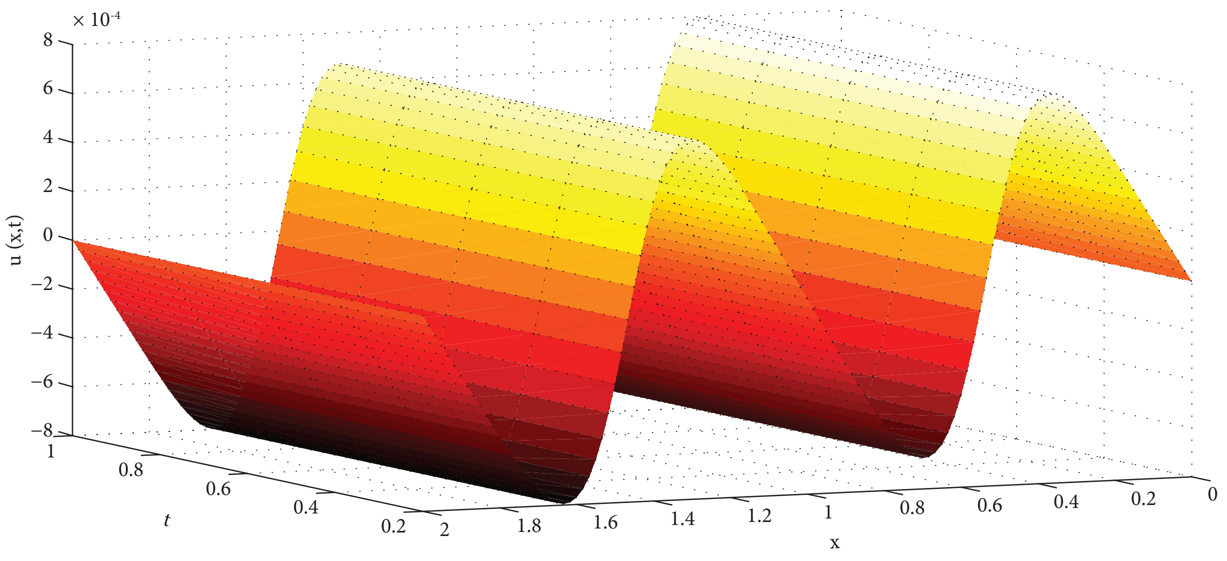

Figure 2: Physical behaviour of the numerical solutions of the second test problem for $\nu = 10^{-4}$ at different time levels.

solution behaviour to highlight the physical behaviour of the equation together with time. As can be depicted from the figure that with the advancement of time, the solution values are decreasing but satisfying the same characteristics at the boundary points.

6.2. 2nd Test Problem. Consider Burgers' equation (1) with domain [0, 2], zero boundary condition, and the initial condition taken from the exact solution. The solution of the equation is verified from the exact solution obtained analytically as

$$u(x,t) = 2\pi\nu \frac{\sin(\gamma)e^{\beta} + 4\sin(2\gamma)e^{-4\beta}}{4 + \cos(\gamma)e^{\beta} + 2\cos(2\gamma)e^{-4\beta}}. \quad (29)$$

Here, $\gamma = \pi x, \beta = -\pi^2\nu^2 t$.

The numerical solution of the equation is obtained for different time levels for $k = 0.1$ at different values of ν. In Tables 3 and 4, the obtained numerical results are compared with the exact solutions and are presented in the form of errors for $t = 0.1$ and $t = 1$, as was available in the literature. As shown by the obtained results in the form of L_2 and L_∞ errors, the numerical solutions are in good agreement with the exact solution and is much better as compared to the result available in the literature at relatively small value of domain partition. The surface plot of the solutions with 121 nodes is presented in Figure 2 to discuss the physical behaviour of the equation along with time, while the physical behaviour of the solution obtained at $t = 1$ for different values of ν is presented in Figure 3. It can be seen that the solution is showing the same behaviour with the changing of time, but the amplitude of the wave is decreasing with the decreasing value of ν.

6.3. 3rd Test Problem. Consider Burgers' equation (1) with domain [0, 1] and the beginning conditions as

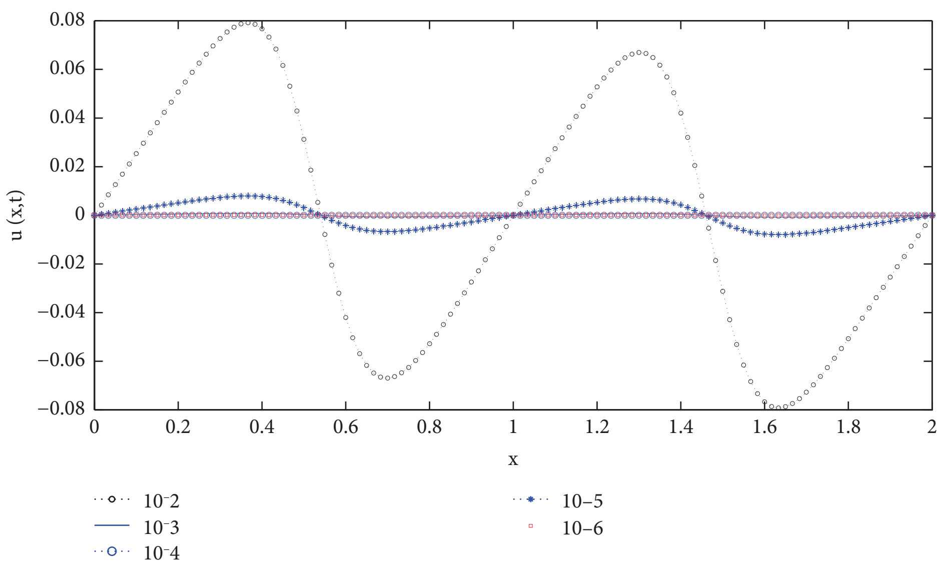

Figure 3: Physical behaviour of the solution of the second test problem obtained at $t = 1$ for different values of ν.

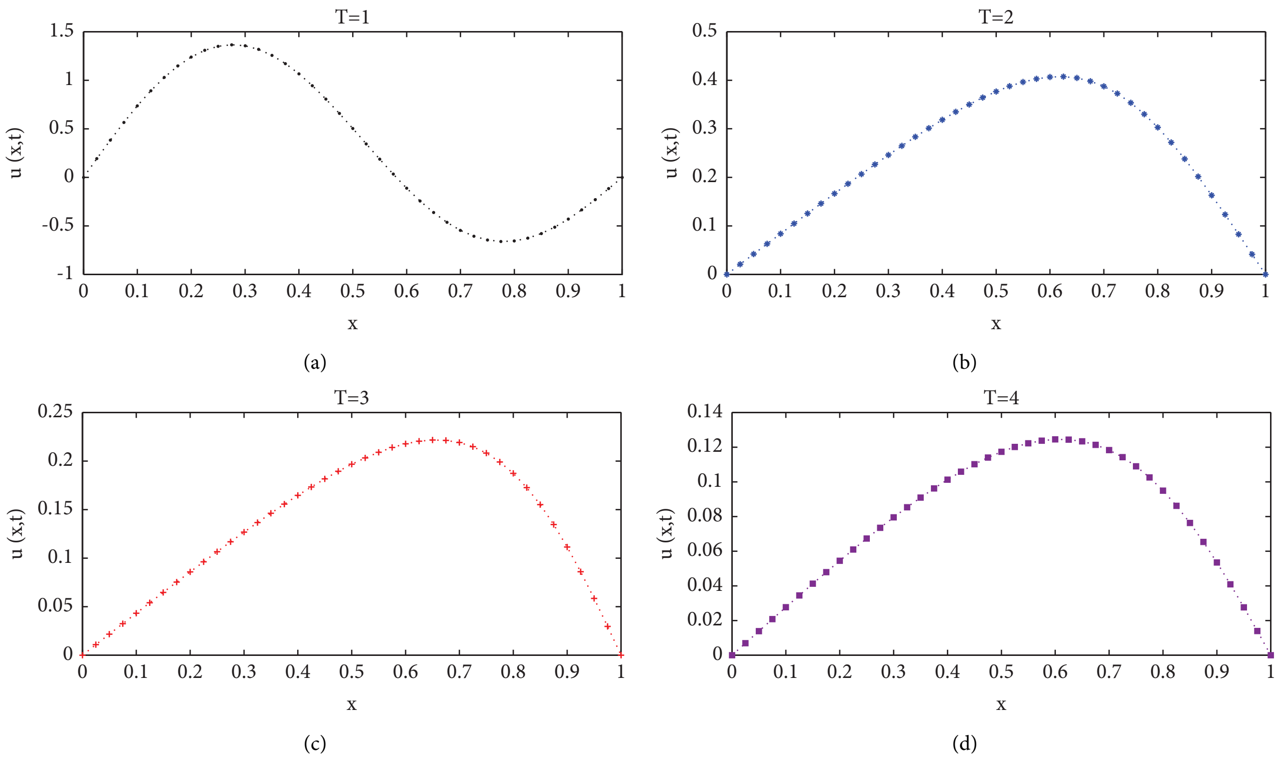

Figure 4: Physical behaviour of the test problem 3 for different values of time. (a) $T = 1$. (b) $T = 2$. (c) $T = 3$. (d) $T = 4$.

$$u(x, 0) = \sin(2\pi x) + 0.5 \quad \sin \quad (\pi x). \tag{30}$$

The equation's solution is calculated and validated with the results already presented in the form of graphs [8]. The solution of the equation is calculated for the different values of time for the viscosity parameter ν with value set as 0.05 for time step $k = 0.01$ and the number of partitions as 41. The results of the numerical simulation are presented for the different time levels in Figure 4. From Figure 4, the physical behaviour of the solution can be seen changing the form along with the passage of time. The solution is of wave form can be seen changing from the two phases to single phase and then decreasing in amplitude with time.

7. Conclusion

The present study demonstrates how effective the proposed hybrid approach of DQM with modified cubic B-splines and the finite difference scheme for discretization can be implemented to solve the partial differential equations. This work is an effort to develop a numerical approach that can provide the approximate solution, which is close to the exact solution of Burgers' equation. The present work is exemplified with the assistance of three examples that demonstrate the benefit of the method in terms of the accuracy obtained with less domain partition and for the adequate required time steps. The presented algorithm is capable of being modified that can be applied to higher-order nonlinear differential equations in either linear or nonlinear form.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] B. Saka and İ. Dağ, "Quartic B-spline collocation method to the numerical solutions of the Burgers' equation," *Chaos, Solitons and Fractals*, vol. 32, no. 3, pp. 1125–1137, 2007.

[2] H. Bateman, "Some recent researches on the motion of fluids," *Monthly Weather Review*, vol. 43, no. 4, pp. 163–170, 1915.

[3] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.

[4] G. Arora and B. K. Singh, "Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method," *Applied Mathematics and Computation*, vol. 224, pp. 166–177, 2013.

[5] A. Korkmaz and İ. Dağ, "Cubic B-spline differential quadrature methods and stability for Burgers' equation," *Engineering Computations*, vol. 30, pp. 320–344, 2013.

[6] G. Arora and V. Joshi, "A computational approach using modified trigonometric cubic B-spline for numerical solution of Burgers' equation in one and two dimensions," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 1087–1098, 2018.

[7] M. Tamsir, V. K. Srivastava, and R. Jiwari, "An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation," *Applied Mathematics and Computation*, vol. 290, pp. 111–124, 2016.

[8] A. Başhan, "Nonlinear dynamics of the Burgers' equation and numerical experiments," *Mathematical Sciences*, vol. 16, no. 2, pp. 183–205, 2022.

[9] T. Öziş, E. N. Aksan, and A. Özdeş, "A finite element approach for solution of Burgers' equation," *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 417–428, 2003.

[10] A. Dogan, "A Galerkin finite element approach to Burgers' equation," *Applied Mathematics and Computation*, vol. 157, no. 2, pp. 331–346, 2004.

[11] I. Dag, D. Irk, and A. Sahin, "B-spline collocation methods for numerical solutions of the Burgers' equation," *Mathematical Problems in Engineering*, vol. 2005, no. 5, Article ID 928423, 18 pages, 2005.

[12] R. C. Mittal and G. Arora, "Numerical solution of the coupled viscous Burgers' equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 3, pp. 1304–1313, 2011.

[13] R. C. Mittal and R. K. Jain, "Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method," *Applied Mathematics and Computation*, vol. 218, no. 15, pp. 7839–7855, 2012.

[14] S. Kutluay, A. R. Bahadir, and A. Özdeş, "Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods," *Journal of Computational and Applied Mathematics*, vol. 103, no. 2, pp. 251–261, 1999.

[15] M. K. Kadalbajoo, K. K. Sharma, and A. Awasthi, "A parameter-uniform implicit difference scheme for solving time-dependent Burgers' equations," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1365–1393, 2005.

[16] I. A. Hassanien, A. A. Salama, and H. A. Hosham, "Fourth-order finite difference method for solving Burgers' equation," *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 781–800, 2005.

[17] W. Liao, "An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation," *Applied Mathematics and Computation*, vol. 206, no. 2, pp. 755–764, 2008.

[18] B. Inan and A. R. Bahadir, "Numerical solution of the one-dimensional Burgers' equation: implicit and fully implicit exponential finite difference methods," *Pramana*, vol. 81, no. 4, pp. 547–556, 2013.

[19] R. Bellman, B. G. Kashef, and J. Casti, "Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations," *Journal of Computational Physics*, vol. 10, no. 1, pp. 40–52, 1972.

[20] J. R. Quan and C. T. Chang, "New insights in solving distributed system equations by the quadrature method—I. Analysis," *Computers and Chemical Engineering*, vol. 13, no. 7, pp. 779–788, 1989.

[21] M. Ghasemi, "High order approximations using spline-based differential quadrature method: implementation to the multi-dimensional PDEs," *Applied Mathematical Modelling*, vol. 46, pp. 63–80, 2017.