



An optimal B-spline collocation technique for numerical simulation of viscous coupled Burgers' equation

Shallu and Vijay Kumar Kukreja *

Department of Mathematics, Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India.

Abstract

In this paper, an optimal cubic B-spline collocation method is applied to solve the viscous coupled Burgers' equation, which helps in modeling the polydisperse sedimentation. As it is not possible to obtain optimal order of convergence with the standard collocation method, so to overcome this, posteriori corrections are made in cubic B-spline interpolant and its higher-order derivatives. This optimal cubic B-spline collocation method is used for space integration and for time-domain integration, the Crank-Nicolson scheme is applied along with the quasilinearization process to deal with the nonlinear terms in the equations. Von-Neumann stability analysis is carried out to discuss the stability of the technique. Few test problems are solved numerically along with the calculation of L_2 , L_∞ error norms as well as the order of convergence. The obtained results are compared with those available in the literature, which shows the improvement in results over the standard collocation method and many other existing techniques also.

Keywords. Coupled Burgers' equation, Cubic B-splines, Optimal collocation method, Crank-Nicolson scheme, Quasilinearization, Von-Neumann stability analysis.

2010 Mathematics Subject Classification. 65M70.

1. INTRODUCTION

In this work, the numerical analysis of the nonlinear time-dependent viscous coupled Burgers' equation is performed. The general form of the equation is as follows:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \alpha u \frac{\partial u}{\partial x} - \eta \frac{\partial(uw)}{\partial x}, \quad x \in [x_L, x_R], \quad t \in [0, T], \quad (1.1)$$

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2} - \alpha w \frac{\partial w}{\partial x} - \rho \frac{\partial(uw)}{\partial x}, \quad x \in [x_L, x_R], \quad t \in [0, T], \quad (1.2)$$

with the initial conditions:

$$u(x, 0) = \Phi_1(x), \quad w(x, 0) = \Phi_2(x), \quad x \in [x_L, x_R], \quad (1.3)$$

and the boundary conditions:

$$\begin{aligned} u(x_L, t) &= \Psi_1(t), \quad u(x_R, t) = \Psi_2(t), \\ w(x_L, t) &= \Psi_3(t), \quad w(x_R, t) = \Psi_4(t), \end{aligned} \quad (1.4)$$

where α is a real constant, η and ρ are the arbitrary constants, depending on the value of different parameters such as Brownian diffusivity, Stokes velocity of particles due to gravity, and Peclet number [26]. $\Phi_1(x)$, $\Phi_2(x)$, $\Psi_1(t)$, $\Psi_2(t)$, $\Psi_3(t)$, and $\Psi_4(t)$ are sufficiently smooth functions. Here u and w are the components of velocity, u_t is the unsteady term, uu_x is the convection term, and u_{xx} is the diffusion term. The coupled Burgers' equation was proposed by

Received: 21 May 2021 ; Accepted: 14 January 2022.

* Corresponding Author. Email: vkukreja@gmail.com.

Esipov [12] for the study of polydisperse sedimentation, which models the accumulation or progression of the scaled volume assemblage of two types of particles in fluid adjournment or colloids, under gravity. Nee and Duan [26] represented the trajectories of coupled Burgers' equation starting with the not too large initial data, which converges to zero equilibrium as the time approaches to infinity.

In the past several decades, various techniques have been implemented to solve this coupled equation, due to its vast applications in various fields of science and engineering. To name a few, Kaya [17] implemented the Adomian decomposition method to solve the coupled Burgers' equation, in which no linearization or weak nonlinearity assumptions were required. Abdou and Soliman [2] solved this equation using the variational iteration method and showed that this technique is better than the Adomian decomposition method. Soliman [41] implemented the modified extended tanh-function, Khater et al. [18] opted spectral collocation method with Chebyshev polynomials (CCM) and solved the system of equations using the fourth-order Runge-Kutta method. Rashid and Ismail [28] used Fourier pseudo-spectral method (FPSM) and showed that this technique performs better than the Chebyshev collocation method. Abazari and Borhanifar [1] implemented the differential transformation method (DTM), Mittal and Arora [22] applied the classical cubic B-spline collocation method (CSCM) for the space discretization and Crank-Nicolson scheme for the time discretization. In this, it is shown that results with CSCM are better than the Chebyshev collocation method and the Fourier Pseudospectral method. Mittal and Jiwari [23] implemented the differential quadrature method (DQM) for space integration and used the Runge Kutta method of fourth-order to solve the system of ODEs. Srivastava et al. [42] proposed an implicit finite-difference method (IFDM) to discretize the coupled Burgers' equation, which resulted in a nonlinear system of equations. Then Newton's iterative method has been used to convert the nonlinear system to a linear one. Further, the Gauss elimination method has been implemented to solve the required system of equations.

Kutluay and Ucar [20] applied the quadratic B-spline Galerkin finite element method (GQFEM) and found that results are better than the cubic B-spline collocation method, differential transformation method, etc. Dehghan et al. [10] implemented the combination of finite difference formula and Galerkin method, using the interpolating scaling functions to solve the Burgers' equation. Mittal and Tripathi [25] implemented a modified cubic B-spline collocation method (MCSCM) for space integration and strong stability preserving scheme to solve a system of ODEs. Kumar and Pandit [19] proposed a combination of Haar wavelets (HWM) and a forward finite difference scheme to solve this equation. Results were found to be better than the cubic B-spline collocation method, differential quadrature method, implicit finite-difference scheme, etc. Ali et al. [3] implemented the non-polynomial spline method, Raslan et al. [29] used the cubic trigonometric B-spline collocation method, Bhatt and Khaliq [7] implemented a fourth-order compact finite difference scheme for space integration and opted fourth-order modified exponential Runge-Kutta scheme for the time integration. Ersoy and Dag [11] implemented the combination of trigonometric cubic B-spline collocation method (TCSCM) and Crank-Nicolson scheme. Ashpazzadeh et al. [4] proposed a method for constructing the wavelet bases, derived from the symmetric biorthogonal multiwavelets (Hermite cubic splines) to solve the Burgers' equation. Gadain [14] implemented the modified double Laplace decomposition method, Chuathong and Kaennakham [8] used meshfree Hermite collocation method with Gaussian radial basis function. Jafarabadi and Shivanian [16] implemented a combination of meshless radial point interpolation and spectral collocation method with thin-plate splines for spatial discretization, a predictor-corrector scheme to linearize the nonlinear terms, and a finite difference scheme for time-domain discretization. Shallal et al. [36] solved this equation using the septic B-spline collocation method (SSCM) and the Crank-Nicolson scheme and found the results to be better than the cubic trigonometric B-spline collocation method. Bashan [5] implemented a combination of the differential quadrature method and finite difference method for a very small value of kinematic viscosity, Fisher and Bialecki [13] used an extrapolated alternating direction implicit (ADI) Crank-Nicolson orthogonal r^{th} order spline collocation method. Zadvan and Rashidinia [43] developed a non-polynomial cubic spline functions based on trigonometric functions. Nemati Saray et al. [35] design and analyze the multiwavelets Galerkin method for solving the two-dimensional Burgers' equation.

In the present paper, cubic B-splines have been chosen as the basis function in the collocation method, due to their higher smoothness property. Also, the matrices corresponding to the spline functions are sparse in nature and



hence easy to handle and solve. But here optimal cubic B-splines have been taken instead of standard cubic B-splines, because they provide better accuracy in results. In this work, the scope of the optimal cubic B-spline collocation method is extended to solve the coupled equations. To date, it is applied to ordinary differential equations [33, 39, 40], and parabolic partial differential equations [15, 24, 30, 37, 38]. Roul [31, 32, 34] applied optimal spline collocation method to solve the singular boundary value problems, for pricing the Asian options with the fixed strike price, for solving a nonhomogeneous time-fractional diffusion equation, etc. For the formation of optimal spline solution, cubic B-splines are forced to satisfy some interpolatory and specific end conditions, which enhances the order of convergence. To solve the coupled Burgers' equation, firstly Crank-Nicolson scheme is implemented to discretize the time domain, the quasilinearization process is followed to tackle the nonlinear terms, and then the optimal spline collocation method is applied for space discretization.

The paper is summed up in a synchronized manner as follows: In section 2, the formation of the optimal cubic B-spline collocation method is presented. In section 3, the proposed combination of techniques is implemented. A pseudo-code is given for better understanding of the implementation technique. In section 4, stability analysis of the technique is carried out using von-Neumann analysis, and the technique is shown to be unconditionally stable. Section 5 contain numerical examples and the corresponding results. L_2, L_∞ error norms and order of convergence are calculated as well as compared with the existing results and the behavior of solutions is demonstrated via 3-D graphs.

2. OPTIMAL CUBIC B-SPLINE COLLOCATION METHOD

The properties of cubic B-spline interpolant and the formation of optimal cubic B-spline solution are discussed hereunder.

2.1. Properties of the Cubic B-Spline Interpolant: Consider the uniform partitioning of the spatial and temporal domain $([x_L, x_R] \times [0, T])$ as, $x_L = x_0 < x_1 < \dots < x_{N-1} < x_N = x_R$ and $0 = t^0 < t^1 < \dots < t^m < \dots < T$ respectively, where $h = x_{j+1} - x_j$ is the spatial step size and $\Delta t = t^{m+1} - t^m$ is the temporal step size. So, $x_{j+1} = x_L + jh, j = 0, 1, 2, \dots, N$ and $t^m = m\Delta t, m = 0, 1, \dots$, where N represents the number of nodal points on the spatial domain. The cubic B-spline basis functions were explained by Prenter [27] and are given below:

$$C_{j,3}(x) = \frac{1}{h^3} \begin{cases} (x - x_{j-2})^3, & [x_{j-2}, x_{j-1}), \\ h^3 + 3h^2(x - x_{j-1}) + 3h(x - x_{j-1})^2 - 3(x - x_{j-1})^3, & [x_{j-1}, x_j), \\ h^3 + 3h^2(x_{j+1} - x) + 3h(x_{j+1} - x)^2 - 3(x_{j+1} - x)^3, & [x_j, x_{j+1}), \\ (x_{j+2} - x)^3, & [x_{j+1}, x_{j+2}), \\ 0, & \text{Otherwise.} \end{cases} \tag{2.1}$$

The collection of cubic B-spline functions $C_{j,3} = \{C_{-1}(x), C_0(x), C_1(x), \dots, C_N(x), C_{N+1}(x)\}$, forms the basis for the $(N + 3)$ dimensional subspace X of $C^2[x_L, x_R]$. The cubic B-spline approximate solutions $V(x, t)$ and $Y(x, t)$ corresponding to the exact solutions $u(x, t)$ and $w(x, t)$ are given below:

$$V(x, t) = \sum_{j=-1}^{N+1} \delta_j(t)C_j(x), \tag{2.2}$$

$$Y(x, t) = \sum_{j=-1}^{N+1} \sigma_j(t)C_j(x), \tag{2.3}$$

with $\delta_j(t)$'s and $\sigma_j(t)$'s as the unknown time-dependent quantities to be determined from the collocation form of the differential equation and the boundary conditions.



2.2. Posteriori Corrections to the Cubic B-Spline Interpolant: Suppose that the cubic B-spline solution $V(x, t)$ satisfies the following interpolatory condition:

$$V(x_j, t) = u(x_j, t), \quad j = 0, 1, \dots, N, \quad (2.4)$$

and the end condition:

$$V_{xx}(x_j, t) = u_{xx}(x_j, t) - \frac{h^2}{12}u_{xxxx}(x_j, t), \quad j = 0, N. \quad (2.5)$$

Theorem 1. For sufficiently smooth function $u(x, t)$ in $[x_L, x_R]$ and the unique cubic B-spline interpolant (CSI) $V(x, t)$, satisfying Eqs. (2.4) -(2.5), the following relations hold:

$$V_{xx}(x_j, t) = u_{xx}(x_j, t) - \frac{h^2}{12}u_{xxxx}(x_j, t) + O(h^4), \quad j = 0, 1, \dots, N, \quad (2.6)$$

$$V(x_j, t) = u(x_j, t) + O(h^4), \quad j = 0, 1, \dots, N. \quad (2.7)$$

Proof. Reported in [9, 21].

Lemma 1. For $u(x, t) \in \mathbb{C}^6[x_L, x_R]$, the following relations hold:

$$\begin{aligned} u_{xxxx}(x_0, t) &= \frac{2V_{xx}(x_0, t) - 5V_{xx}(x_1, t) + 4V_{xx}(x_2, t) - V_{xx}(x_3, t)}{h^2} + O(h^2), \\ u_{xxxx}(x_j, t) &= \frac{V_{xx}(x_{j-1}, t) - 2V_{xx}(x_j, t) + V_{xx}(x_{j+1}, t)}{h^2} + O(h^2), \quad j = 1, 2, \dots, N-1, \\ u_{xxxx}(x_N, t) &= \frac{2V_{xx}(x_N, t) - 5V_{xx}(x_{N-1}, t) + 4V_{xx}(x_{N-2}, t) - V_{xx}(x_{N-3}, t)}{h^2} + O(h^2). \end{aligned}$$

Proof. This can be proved by simple application of finite differences and Taylor's series expansion [9].

Corollary 1. For $u(x, t) \in \mathbb{C}^6[x_L, x_R]$, the below mentioned relations hold:

$$\begin{aligned} u_x(x_j, t) &= V_x(x_j, t) + O(h^4), \quad j = 0, 1, \dots, N, \\ u_{xx}(x_0, t) &= \frac{14V_{xx}(x_0, t) - 5V_{xx}(x_1, t) + 4V_{xx}(x_2, t) - V_{xx}(x_3, t)}{12} + O(h^4), \\ u_{xx}(x_j, t) &= \frac{V_{xx}(x_{j-1}, t) + 10V_{xx}(x_j, t) + V_{xx}(x_{j+1}, t)}{12} + O(h^4), \quad j = 1, 2, \dots, N-1, \\ u_{xx}(x_N, t) &= \frac{14V_{xx}(x_N, t) - 5V_{xx}(x_{N-1}, t) + 4V_{xx}(x_{N-2}, t) - V_{xx}(x_{N-3}, t)}{12} + O(h^4). \end{aligned}$$

The value of $V_{xx}(x_j, t)$ is as follows, using Eqs. (2.1)-(2.2):

$$V_{xx}(x_j, t) = \frac{6}{h}(\delta_{j-1} - 2\delta_j + \delta_{j+1}), \quad j = 0, 1, 2, \dots, N.$$

Substituting the value of $V_{xx}(x_j, t)$ in Corollary 1, the below mentioned optimal second-order derivatives are obtained:

$$\begin{aligned} u_{xx}(x_0, t) &= \frac{14\delta_{-1} - 33\delta_0 + 28\delta_1 - 14\delta_2 + 6\delta_3 - \delta_4}{2h^2} + O(h^4), \\ u_{xx}(x_j, t) &= \frac{\delta_{j-2} + 8\delta_{j-1} - 18\delta_j + 8\delta_{j+1} + \delta_{j+2}}{2h^2} + O(h^4), \quad j = 1, 2, \dots, N-1, \\ u_{xx}(x_N, t) &= \frac{14\delta_{N+1} - 33\delta_N + 28\delta_{N-1} - 14\delta_{N-2} + 6\delta_{N-3} - \delta_{N-4}}{2h^2} + O(h^4). \end{aligned}$$



The values of $u(x, t)$ and $u_x(x, t)$ are as follows "at nodal points" using definition of cubic B-splines:

$$\begin{aligned} u(x_j, t) &= \delta_{j-1} + 4\delta_j + \delta_{j+1} + O(h^4), \\ u_x(x_j, t) &= -\frac{3}{h}(\delta_{j-1} - \delta_{j+1}) + O(h^4). \end{aligned} \tag{2.8}$$

The value of $w(x, t)$ and its higher-order derivatives can be obtained from the above relations, just by replacing δ with σ .

3. IMPLEMENTATION OF PROPOSED TECHNIQUE

Apply Crank-Nicolson scheme to discretize the temporal domain of coupled Burgers' equations (1.1) and (1.2) as follows:

$$\frac{u^{m+1} - u^m}{\Delta t} = \frac{u_x^{m+1} + u_x^m}{2} - \alpha \left[\frac{(uu_x)^{m+1} + (uu_x)^m}{2} \right] - \eta \left[\frac{(uw_x)^{m+1} + (uw_x)^m}{2} \right] - \eta \left[\frac{(wu_x)^{m+1} + (wu_x)^m}{2} \right]. \tag{3.1}$$

$$\frac{w^{m+1} - w^m}{\Delta t} = \frac{w_x^{m+1} + w_x^m}{2} - \alpha \left[\frac{(ww_x)^{m+1} + (ww_x)^m}{2} \right] - \rho \left[\frac{(uw_x)^{m+1} + (uw_x)^m}{2} \right] - \rho \left[\frac{(wu_x)^{m+1} + (wu_x)^m}{2} \right]. \tag{3.2}$$

Linearize the nonlinear terms using the quasilinearization process, reported by [6] as follows:

$$(uw_x)^{m+1} = (uw_x)^m + (u^{m+1} - u^m) \left(\frac{\partial(uw_x)}{\partial u} \right)^m + (w_x^{m+1} - w_x^m) \left(\frac{\partial(uw_x)}{\partial w_x} \right)^m + O(\Delta t^2).$$

Similar relation holds for the terms $(uu_x)^{m+1}$ and $(wu_x)^{m+1}$. Substitute the above values in Eqs. (3.1)-(3.2) and separate the terms of $(m + 1)^{th}$ and m^{th} time level:

$$\left[\frac{1}{\Delta t} + \frac{\alpha}{2} u_x^m + \frac{\eta}{2} w_x^m \right] u^{m+1} + \left[\frac{\alpha}{2} u^m + \frac{\eta}{2} w^m \right] u_x^{m+1} - \frac{u_x^{m+1}}{2} + \left(\frac{\eta}{2} u_x^m \right) w^{m+1} + \left(\frac{\eta}{2} u^m \right) w_x^{m+1} = \frac{u^m}{\Delta t} + \frac{u_x^m}{2}. \tag{3.3}$$

$$\left[\frac{1}{\Delta t} + \frac{\alpha}{2} w_x^m + \frac{\rho}{2} u_x^m \right] w^{m+1} + \left[\frac{\alpha}{2} w^m + \frac{\rho}{2} u^m \right] w_x^{m+1} - \frac{w_x^{m+1}}{2} + \left(\frac{\rho}{2} w_x^m \right) u^{m+1} + \left(\frac{\rho}{2} w^m \right) u_x^{m+1} = \frac{w^m}{\Delta t} + \frac{w_x^m}{2}. \tag{3.4}$$

At any j^{th} nodal point, Eqs. (3.3)-(3.4) can be written as follows:

$$P_1(j)u_j^{m+1} + Q_1(j)(u_x)_j^{m+1} - \frac{(u_{xx})_j^{m+1}}{2} + R_1(j)w_j^{m+1} + S_1(j)(w_x)_j^{m+1} = E_1(j). \tag{3.5}$$

$$R_2(j)w_j^{m+1} + S_2(j)(w_x)_j^{m+1} - \frac{(w_{xx})_j^{m+1}}{2} + P_2(j)u_j^{m+1} + Q_2(j)(u_x)_j^{m+1} = E_2(j). \tag{3.6}$$

Substitute the optimal values of $u(x, t)$, $w(x, t)$, and their higher-order derivatives and clubbing the coefficients of δ_j^{m+1} 's and σ_j^{m+1} 's for $j = 0, 1, \dots, N$:



For $j = 0$:

$$\left[P_1(0) - \frac{3Q_1(0)}{h} - \frac{7}{2h^2} \right] \delta_{-1}^{m+1} + \left[4P_1(0) + \frac{33}{4h^2} \right] \delta_0^{m+1} + \left[P_1(0) + \frac{3Q_1(0)}{h} - \frac{7}{h^2} \right] \delta_1^{m+1} + \frac{7}{2h^2} \delta_2^{m+1} - \frac{3}{2h^2} \delta_3^{m+1} + \frac{1}{4h^2} \delta_4^{m+1} + \left[R_1(0) - \frac{3S_1(0)}{h} \right] \sigma_{-1}^{m+1} + 4R_1(0) \sigma_0^{m+1} + \left[R_1(0) + \frac{3S_1(0)}{h} \right] \sigma_1^{m+1} = E_1(0) + O(h^4).$$

$$a_0 \delta_{-1}^{m+1} + b_0 \delta_0^{m+1} + c_0 \delta_1^{m+1} + d_0 \delta_2^{m+1} + e_0 \delta_3^{m+1} + f_0 \delta_4^{m+1} + g_0 \sigma_{-1}^{m+1} + l_0 \sigma_0^{m+1} + n_0 \sigma_1^{m+1} = E_1(0) + O(h^4). \tag{3.7}$$

$$\left[P_2(0) - \frac{3Q_2(0)}{h} \right] \delta_{-1}^{m+1} + 4P_2(0) \delta_0^{m+1} + \left[P_2(0) + \frac{3Q_2(0)}{h} \right] \delta_1^{m+1} + \left[R_2(0) - \frac{3S_2(0)}{h} - \frac{7}{2h^2} \right] \sigma_{-1}^{m+1} + \left[4R_2(0) + \frac{33}{4h^2} \right] \sigma_0^{m+1} + \left[R_2(0) + \frac{3S_2(0)}{h} - \frac{7}{h^2} \right] \sigma_1^{m+1} + \frac{7}{2h^2} \sigma_2^{m+1} - \frac{3}{2h^2} \sigma_3^{m+1} + \frac{1}{4h^2} \sigma_4^{m+1} = E_2(0) + O(h^4).$$

$$p_0 \delta_{-1}^{m+1} + q_0 \delta_0^{m+1} + r_0 \delta_1^{m+1} + s_0 \sigma_{-1}^{m+1} + v_0 \sigma_0^{m+1} + y_0 \sigma_1^{m+1} + d_0 \sigma_2^{m+1} + e_0 \sigma_3^{m+1} + f_0 \sigma_4^{m+1} = E_2(0) + O(h^4). \tag{3.8}$$

For $j = 1, 2, \dots, N - 1$:

$$-\frac{1}{4h^2} \delta_{j-2}^{m+1} + \left[P_1(j) - \frac{3Q_1(j)}{h} - \frac{2}{h^2} \right] \delta_{j-1}^{m+1} + \left[4P_1(j) + \frac{9}{2h^2} \right] \delta_j^{m+1} + \left[P_1(j) + \frac{3Q_1(j)}{h} - \frac{2}{h^2} \right] \delta_{j+1}^{m+1} - \frac{1}{4h^2} \delta_{j+2}^{m+1} + \left[R_1(j) - \frac{3S_1(j)}{h} \right] \sigma_{j-1}^{m+1} + 4R_1(j) \sigma_j^{m+1} + \left[R_1(j) + \frac{3S_1(j)}{h} \right] \sigma_{j+1}^{m+1} = E_1(j) + O(h^4).$$

$$a_j \delta_{j-2}^{m+1} + b_j \delta_{j-1}^{m+1} + c_j \delta_j^{m+1} + d_j \delta_{j+1}^{m+1} + a_j \delta_{j+2}^{m+1} + g_j \sigma_{j-1}^{m+1} + l_j \sigma_j^{m+1} + n_j \sigma_{j+1}^{m+1} = E_1(j) + O(h^4). \tag{3.9}$$

$$\left[P_2(j) - \frac{3Q_2(j)}{h} \right] \delta_{j-1}^{m+1} + 4P_2(j) \delta_j^{m+1} + \left[P_2(j) + \frac{3Q_2(j)}{h} \right] \delta_{j+1}^{m+1} - \frac{1}{4h^2} \sigma_{j-2}^{m+1} + \left[R_2(j) - \frac{3S_2(j)}{h} - \frac{2}{h^2} \right] \sigma_{j-1}^{m+1} + \left[4R_2(j) + \frac{9}{2h^2} \right] \sigma_j^{m+1} + \left[R_2(j) + \frac{3S_2(j)}{h} - \frac{2}{h^2} \right] \sigma_{j+1}^{m+1} - \frac{1}{4h^2} \sigma_{j+2}^{m+1} = E_2(j) + O(h^4).$$

$$p_j \delta_{j-1}^{m+1} + q_j \delta_j^{m+1} + r_j \delta_{j+1}^{m+1} + a_j \sigma_{j-2}^{m+1} + s_j \sigma_{j-1}^{m+1} + v_j \sigma_j^{m+1} + y_j \sigma_{j+1}^{m+1} + a_j \sigma_{j+2}^{m+1} = E_2(j) + O(h^4). \tag{3.10}$$

For $j = N$:

$$\frac{1}{4h^2} \delta_{N-4}^{m+1} - \frac{3}{2h^2} \delta_{N-3}^{m+1} + \frac{7}{2h^2} \delta_{N-2}^{m+1} + \left[P_1(N) - \frac{3Q_1(N)}{h} - \frac{7}{h^2} \right] \delta_{N-1}^{m+1} + \left[4P_1(N) + \frac{33}{4h^2} \right] \delta_N^{m+1} + \left[P_1(N) - \frac{3Q_1(N)}{h} - \frac{7}{2h^2} \right] \delta_{N+1}^{m+1} + \left[R_1(N) - \frac{3S_1(N)}{h} \right] \sigma_{N-1}^{m+1} + 4R_1(N) \sigma_N^{m+1} + \left[R_1(N) + \frac{3S_1(N)}{h} \right] \sigma_{N+1}^{m+1} = E_1(N) + O(h^4).$$

$$a_N \delta_{N-4}^{m+1} + b_N \delta_{N-3}^{m+1} + c_N \delta_{N-2}^{m+1} + d_N \delta_{N-1}^{m+1} + e_N \delta_N^{m+1} + f_N \delta_{N+1}^{m+1} + g_N \sigma_{N-1}^{m+1} + l_N \sigma_N^{m+1} + n_N \sigma_{N+1}^{m+1} = E_1(N) + O(h^4). \tag{3.11}$$

$$\left[P_2(N) - \frac{3Q_2(N)}{h} \right] \delta_{N-1}^{m+1} + 4P_2(N) \delta_N^{m+1} + \left[P_2(N) + \frac{3Q_2(N)}{h} \right] \delta_{N+1}^{m+1} + \frac{1}{4h^2} \sigma_{N-4}^{m+1} - \frac{3}{2h^2} \sigma_{N-3}^{m+1} + \frac{7}{2h^2} \sigma_{N-2}^{m+1} + \left[R_2(N) - \frac{3S_2(N)}{h} - \frac{7}{h^2} \right] \sigma_{N-1}^{m+1} + \left[4R_2(N) + \frac{33}{4h^2} \right] \sigma_N^{m+1} + \left[R_2(N) + \frac{3S_2(N)}{h} - \frac{7}{2h^2} \right] \sigma_{N+1}^{m+1} = E_2(N) + O(h^4).$$



3.2. Pseudocode:

- Input the values of $x_L, x_R, h, \Delta t, t, N, x_0, t_0$;
- Space integration
FOR ($j = 1; j < N + 1; j++$)
 $x_j = x_0 + jh$;
ENDFOR
- Time integration
FOR ($m = 1; m < \frac{t}{\Delta t} + 1; m++$)
 $t^m = t^0 + m\Delta t$;
ENDFOR
- Input matrix corresponding to initial condition, to calculate the value of δ and σ at initial time using Eqs. (3.15)-(3.16);
- Calculate the value of $u(x, t)$ and $u_x(x, t)$ using Eq. (2.8);
- Similarly calculate the value of $w(x, t)$ and $w_x(x, t)$;
- Input matrix \mathcal{A} and \mathcal{B} to solve the system (3.14);
- The value of δ and σ will be calculated at next time level.;
- Using Eq. (2.8), the value of $u(x, t)$, $u_x(x, t)$, $w(x, t)$, and $w_x(x, t)$ can be calculated at next time level;
- Continue until the value at required time level is obtained;
- Compute L_∞ and L_2 error norms using Eq. (5.1);
- Compute order of convergence using Eq. (5.2);

4. STABILITY ANALYSIS

The stability analysis of the proposed optimal spline collocation technique is carried out using the von-Neumann scheme. To linearize the nonlinear terms in the Eqs. (1.1)-(1.2), take u and w as local constants λ_1 and λ_2 respectively. With this, Eq. (1.1) becomes the following:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - \alpha\lambda_1 \frac{\partial u}{\partial x} - \eta\lambda_2 \frac{\partial u}{\partial x} - \eta\lambda_1 \frac{\partial w}{\partial x}. \quad (4.1)$$

Apply the Crank-Nicolson scheme to discretize the temporal domain at any nodal point 'j':

$$\frac{u_j^{m+1} - u_j^m}{\Delta t} = \frac{(u_{xx})_j^{m+1} + (u_{xx})_j^m}{2} - \alpha\lambda_1 \left[\frac{(u_x)_j^{m+1} + (u_x)_j^m}{2} \right] - \eta\lambda_2 \left[\frac{(u_x)_j^{m+1} + (u_x)_j^m}{2} \right] - \eta\lambda_1 \left[\frac{(w_x)_j^{m+1} + (w_x)_j^m}{2} \right]. \quad (4.2)$$

Separate the terms of $(m+1)^{th}$ and m^{th} time level:

$$\frac{u_j^{m+1}}{\Delta t} + (\alpha\lambda_1 + \eta\lambda_2) \frac{(u_x)_j^{m+1}}{2} - \frac{(u_{xx})_j^{m+1}}{2} + \eta\lambda_1 \frac{(w_x)_j^{m+1}}{2} = \frac{u_j^m}{\Delta t} - (\alpha\lambda_1 + \eta\lambda_2) \frac{(u_x)_j^m}{2} + \frac{(u_{xx})_j^m}{2} - \eta\lambda_1 \frac{(w_x)_j^m}{2}. \quad (4.3)$$

Substitute the optimal values of $u(x, t)$, $w(x, t)$, and their higher-order derivatives and clubbing the coefficients of δ_j^{m+1} 's and σ_j^{m+1} 's:

$$\begin{aligned} & -\frac{1}{4h^2} \delta_{j-2}^{m+1} + \left[\frac{1}{\Delta t} - \frac{3(\alpha\lambda_1 + \eta\lambda_2)}{2h} - \frac{2}{h^2} \right] \delta_{j-1}^{m+1} + \left[\frac{4}{\Delta t} + \frac{9}{2h^2} \right] \delta_j^{m+1} + \left[\frac{1}{\Delta t} + \frac{3(\alpha\lambda_1 + \eta\lambda_2)}{2h} - \frac{2}{h^2} \right] \delta_{j+1}^{m+1} - \\ & \frac{1}{4h^2} \delta_{j+2}^{m+1} - \frac{3\eta\lambda_1}{2} \sigma_{j-1}^{m+1} + \frac{3\eta\lambda_1}{2} \sigma_{j+1}^{m+1} = \frac{1}{4h^2} \delta_{j-2}^m + \left[\frac{1}{\Delta t} + \frac{3(\alpha\lambda_1 + \eta\lambda_2)}{2h} + \frac{2}{h^2} \right] \delta_{j-1}^m + \left[\frac{4}{\Delta t} - \frac{9}{2h^2} \right] \delta_j^m + \\ & \left[\frac{1}{\Delta t} - \frac{3(\alpha\lambda_1 + \eta\lambda_2)}{2h} + \frac{2}{h^2} \right] \delta_{j+1}^m + \frac{1}{4h^2} \delta_{j+2}^m + \frac{3\eta\lambda_1}{2} \sigma_{j-1}^m - \frac{3\eta\lambda_1}{2} \sigma_{j+1}^m. \end{aligned} \quad (4.4)$$



For simplification, write the above equation in the following form:

$$\begin{aligned} z_1\delta_{j-2}^{m+1} + z_2\delta_{j-1}^{m+1} + z_3\delta_j^{m+1} + z_4\delta_{j+1}^{m+1} + z_1\delta_{j+2}^{m+1} + z_5\sigma_{j-1}^{m+1} - z_5\sigma_{j+1}^{m+1} &= -z_1\delta_{j-2}^m + z_6\delta_{j-1}^m + \\ z_7\delta_j^m + z_8\delta_{j+1}^m - z_1\delta_{j+2}^m - z_5\sigma_{j-1}^m + z_5\sigma_{j+1}^m. \end{aligned} \tag{4.5}$$

Put $\delta_j^m = A\chi^m e^{ij\phi}$ and $\sigma_j^m = B\chi^m e^{ij\phi h}$, where χ is the amplification factor, A, B are the amplitudes, $i = \sqrt{-1}$, and $\phi = \kappa h$, where κ is the mode number and h is the spatial step length.

$$\begin{aligned} \chi &= \frac{-z_1 A e^{-2i\phi} + z_6 A e^{-i\phi} + z_7 A + z_8 A e^{i\phi} - z_1 A e^{2i\phi} - z_5 B e^{-i\phi} + z_5 B e^{i\phi}}{z_1 A e^{-2i\phi} + z_2 A e^{-i\phi} + z_3 A + z_4 A e^{i\phi} + z_1 A e^{2i\phi} + z_5 B e^{-i\phi} - z_5 B e^{i\phi}}, \\ &= \frac{-2z_1 A \cos(2\phi) + (z_6 + z_8) A \cos(\phi) + z_7 A + i(-z_6 + z_8) A \sin(\phi) + i2z_5 B \sin(\phi)}{2z_1 A \cos(2\phi) + (z_2 + z_4) A \cos(\phi) + z_3 A + i(-z_2 + z_4) A \sin(\phi) - i2z_5 B \sin(\phi)}, \\ &= \frac{F_1 + iG_1}{F_2 + iG_2}, \end{aligned} \tag{4.6}$$

where

$$\begin{aligned} F_1 &= A \left[\frac{1}{2h^2} \cos(2\phi) + \left(\frac{2}{\Delta t} + \frac{4}{h^2} \right) \cos(\phi) + \frac{4}{\Delta t} - \frac{9}{2h^2} \right], \\ G_1 &= -\frac{3A(\alpha\lambda_1 + \eta\lambda_2)}{h} \sin(\phi) - 3\eta\lambda_1 B \sin(\phi), \\ F_2 &= A \left[-\frac{1}{2h^2} \cos(2\phi) + \left(\frac{2}{\Delta t} - \frac{4}{h^2} \right) \cos(\phi) + \frac{4}{\Delta t} + \frac{9}{2h^2} \right], \\ G_2 &= \frac{3A(\alpha\lambda_1 + \eta\lambda_2)}{h} \sin(\phi) + 3\eta\lambda_1 B \sin(\phi). \end{aligned}$$

For the stability of the technique, one needs to prove that $|\chi| \leq 1$. Since $G_1 = -G_2$, so $G_1^2 = G_2^2$. For $|\chi| \leq 1$, one only needs to prove that $F_2 \geq F_1$ or $F_2 - F_1 \geq 0$.

$$F_2 - F_1 = A \left[-\frac{1}{h^2} \cos(2\phi) - \frac{8}{h^2} \cos(\phi) + \frac{9}{h^2} \right]. \tag{4.7}$$

Take $\cos(\phi) = 1$ for the minimum possible value of $F_2 - F_1$, which gives $F_2 - F_1 = 0$. Hence $F_2 - F_1 \geq 0$. Since $G_1^2 = G_2^2$ and $F_2 \geq F_1$, so $|\chi| \leq 1$. Hence the scheme is unconditionally stable. This means that there is no restriction on the spatial and temporal step size, i.e., on h and Δt . Due to symmetric nature of u and w , similar results can be obtained for Eq. (1.2).

5. NUMERICAL EXAMPLES

To illustrate the applicability of the proposed optimal collocation technique, few problems of coupled Burgers' equations are solved hereunder. The L_∞ and L_2 error norms are calculated using the following formulae:

$$L_\infty = \max_{0 \leq j \leq N} |u_j^{exact} - u_j^{num}|, \quad L_2 = \frac{\sqrt{\sum_{j=0}^N |u_j^{exact} - u_j^{num}|^2}}{\sqrt{\sum_{j=0}^N |u_j^{exact}|^2}}, \tag{5.1}$$



where u_j^{exact} and u_j^{num} represents the exact and optimal cubic B-spline solutions respectively at any nodal point ' x_j ' for some fixed time.

To show the improvement in the results over the standard CSCM, the following formula is used to compute the order of convergence:

$$Order = \frac{\log(Err(N_1)/Err(N_2))}{\log(N_2/N_1)}, \quad (5.2)$$

where $Err(N_1)$ and $Err(N_2)$ represents the error with N_1 and N_2 number of partitions of the spatial domain. Order of convergence is calculated corresponding to both L_∞ and L_2 error norms.

Example 1. Consider the coupled Burgers' equations (1.1) and (1.2) with $\alpha = -2$, $\eta = 1$, $\rho = 1$ in the spatial domain $[-\pi, \pi]$, with the following initial conditions:

$$u(x, 0) = w(x, 0) = \sin(x), \quad (5.3)$$

and the boundary conditions:

$$u(-\pi, t) = u(\pi, t) = w(-\pi, t) = w(\pi, t) = 0. \quad (5.4)$$

The exact solution of the problem is given in [17] as:

$$u(x, t) = w(x, t) = \exp(-t)\sin(x). \quad (5.5)$$

Numerical results are obtained for different spatial and temporal step sizes and are compared with existing results. Table 1 show the comparison of $L_\infty(u)$ and $L_2(u)$ error norms with $\alpha = -2$, $\eta = 1$, $\rho = 1$, $\Delta t = 0.001$, and $t = 0.1$. It is observed that the results with the proposed optimal cubic B-spline collocation technique are better than septic B-spline collocation method [36]. In Table 2 values of exact and numerical solutions is compared at different nodal points with $\alpha = -2$, $\eta = 1$, $\rho = 1$, $\Delta t = 0.001$, and $N = 50$ at different time levels. In Table 3, a comparison of order of convergence is given with $\Delta t = 0.001$. Comparison shows that proposed OCSCM provides fourth-order while the classical CSCM [22] gives second-order of convergence in spatial domain. In Table 4, the order of convergence is obtained numerically which agrees with the theoretical results. In Table 5, $L_\infty(u)$ error norm is calculated at different time levels $t = 0.5, 1.0, 2.0$, and 3.0 . The comparison shows that results are better than many existing techniques such as the differential quadrature method [23], Galerkin quadratic B-spline finite element method [20], modified cubic B-spline collocation method [25], and trigonometric cubic B-spline collocation method [11]. In Table 6, the comparison of $L_\infty(u)$ and $L_2(u)$ error norms is presented at different time levels. The result shows that the proposed technique is more accurate than the cubic B-spline collocation method [22], implicit finite difference scheme [42], Haar wavelets [19], etc even with less number of spatial nodal points. In Table 7, a comparison of $L_\infty(u)$ and $L_\infty(w)$ error norm is given at different time levels and it is shown that the technique is computationally efficient. In Table 8 comparison of $L_\infty(u)$ and $L_\infty(w)$ error norm is given with $\alpha = -20$, $\eta = 10$, and $\rho = 10$. Figure 1 shows the similarity between numerical and exact solutions at different time levels and Figure 2 represents the 3-D plot of the exact and numerical solutions with $N = 100$, $\Delta t = 0.005$, and $t = 1$.

Example 2. Consider the coupled Burgers' equations (1.1) and (1.2) with $\alpha = 2$, for different value of η and ρ in the spatial domain $[-10, 10]$, with the following initial conditions:

$$\begin{aligned} u(x, 0) &= \varrho[1 - \tanh(\vartheta x)], \\ w(x, 0) &= \varrho \left[\left(\frac{2\rho - 1}{2\eta - 1} \right) - \tanh(\vartheta x) \right]. \end{aligned} \quad (5.6)$$



The exact solution of the problem is given in [41] as:

$$\begin{aligned}
 u(x, t) &= \varrho[1 - \tanh(\vartheta(x - 2\vartheta t))], \\
 w(x, t) &= \varrho \left[\left(\frac{2\rho - 1}{2\eta - 1} \right) - \tanh(\vartheta(x - 2\vartheta t)) \right],
 \end{aligned}
 \tag{5.7}$$

where

$$\varrho = 0.05, \quad \vartheta = \frac{\varrho}{2} \left(\frac{4\eta\rho - 1}{2\eta - 1} \right).$$

In Table 9, values of exact and numerical solutions are compared at different nodal points with $\alpha = 2$, $\eta = 0.1$, $\rho = 0.3$, $\Delta t = 0.001$, and $N = 100$ at different time levels. Table 10-11 give the comparison of $L_\infty(u)$ and $L_\infty(w)$ error norms with $\alpha = 2$ and $\Delta t = 0.01$ for different values of η and ρ . The comparison shows that results with OCSCM are better than Chebyshev spectral collocation method [18], Fourier pseudospectral method [28], cubic B-spline collocation method [22], differential quadrature method [23], and Haar wavelets [19]. Also, better results are obtained even with fewer collocation points, which reduces computations also. In Table 12, a comparison of $L_\infty(u)$, $L_\infty(w)$ error norms, and CPU time are given. Figure 3 shows the comparison of numerical and exact solutions and Figure 4 gives the 3-D plot of exact and numerical solutions $N = 100$, $\Delta t = 0.005$, and $T = 1$.

Example 3. Consider the coupled Burgers' equations (1.1) and (1.2) for different value of α , η , and ρ in the spatial domain $[0, 1]$, with the following initial conditions:

$$\begin{aligned}
 u(x, 0) &= \begin{cases} \sin(2\pi x), & 0 \leq x \leq 0.5, \\ 0, & 0.5 < x \leq 1, \end{cases} \\
 w(x, 0) &= \begin{cases} 0, & 0 \leq x \leq 0.5, \\ -\sin(2\pi x), & 0.5 < x \leq 1, \end{cases}
 \end{aligned}
 \tag{5.8}$$

and boundary conditions:

$$u(0, t) = u(1, t) = 0, \quad w(0, t) = w(1, t) = 0.
 \tag{5.9}$$

Very limited work is available in the literature dealing with the coupled Burgers' equation with a periodic initial condition in contrast to coupled Burgers' equation with non-periodic initial condition. Also, the exact solution of this problem is not available in the literature. The solution of this problem is examined numerically at different time levels. In Table 13, a comparison of $L_\infty(u)$ error norm and order of convergence is reported with $\Delta t = 0.01$, $\alpha = 2$, and $\rho = \eta = 10$ at $t = 0.1$. Due to the non-availability of the exact solution, the solution with a different number of partitions is compared with the solution with 400 number of partitions, considering it as exact solution. Figure 5 represents the 2-D plots of numerical solution with $N = 100$, $\Delta t = 0.001$, $\eta = \rho = 10$, for different values of α at different time levels. In Figure 6, 3-D plots is given with $N = 100$, $\Delta t = 0.001$, $\eta = \rho = 10$ and $\alpha = 2$ at $t = 0.01$ and $t = 0.5$.

6. CONCLUSION

In this analysis, a simple numerical technique has been implemented to solve the coupled Burgers' equation. Just by making few posteriori corrections in cubic B-spline interpolant, the optimal cubic B-spline collocation method is providing better results in terms of accuracy than many other techniques such as Chebyshev spectral collocation method [18], Fourier pseudospectral method [28], implicit finite difference scheme [42], cubic B-spline [22], modified cubic B-spline [25], trigonometric cubic B-spline [11], septic B-spline collocation method [36], differential quadrature



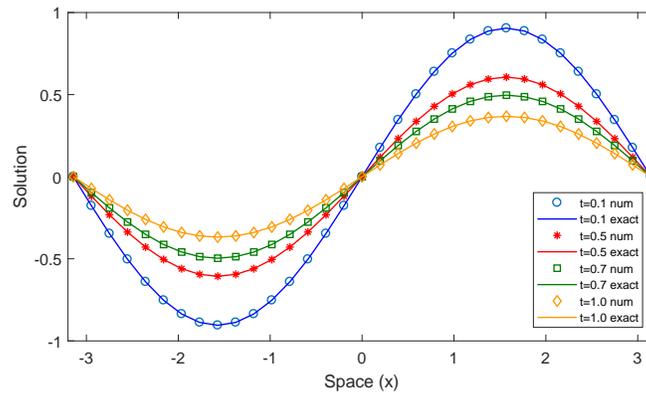


FIGURE 1. Comparison of numerical and exact solution of Example 1 at different time levels with $N = 32$ and $\Delta t = 0.01$.

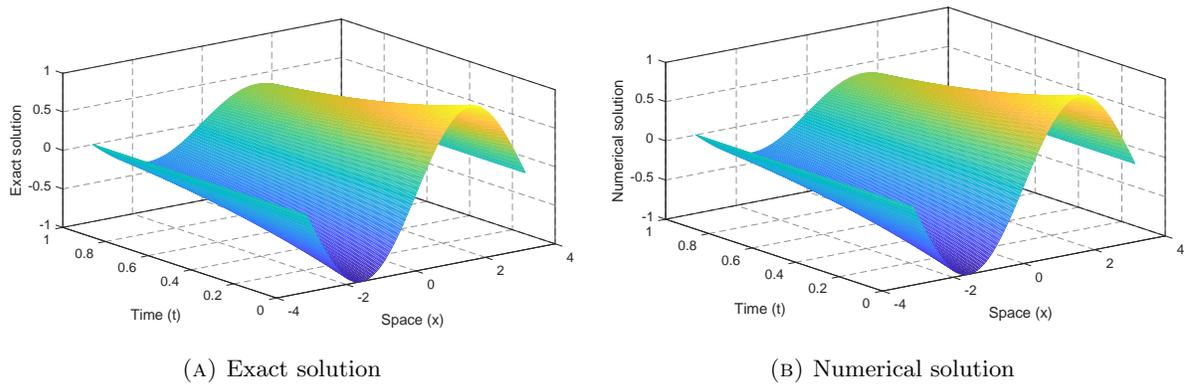


FIGURE 2. 3-D plots of Example 1 with $N = 100$, $\Delta t = 0.005$, and $t = 1$.

method [23], Galerkin quadratic B-spline FEM [20], etc. Also, it is computationally efficient, as better accuracy is achieved with a very small number of nodal points. This technique can be extended to solve higher-order problems arising in various fields of science and engineering.

7. ACKNOWLEDGMENT

Ms. Shallu is thankful to CSIR New Delhi for providing financial assistance in the form of SRF with File No. 09/797(0016)/2018-EMR-I.



TABLE 1. Comparison of $L_\infty(u)$ and $L_2(u)$ error norms of Example 1 with $\alpha = -2$, $\eta = 1$, $\rho = 1$, $\Delta t = 0.001$, and $t = 0.1$.

N	OCSCM		SSCM [36]		OCSCM		SSCM [36]	
	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$	$L_\infty(w)$	$L_2(w)$	$L_\infty(w)$	$L_2(w)$
50	7.0019E-8	7.0001E-8	7.4345E-7	6.7212E-7	7.0019E-8	7.0001E-8	7.4345E-7	6.7212E-7
100	1.1457E-8	1.1455E-8	7.9486E-7	7.3458E-7	1.1457E-8	1.1455E-8	7.9486E-7	7.3458E-7
128	8.9994E-9	8.9992E-9	8.1239E-7	7.8341E-7	8.9994E-9	8.9992E-9	8.1239E-7	7.8341E-7
200	7.7851E-9	7.7851E-9	9.6414E-7	9.0329E-7	7.7851E-9	7.7851E-9	9.6414E-7	9.0329E-7

TABLE 2. Absolute error of Example 1 with $\alpha = -2$, $\eta = 1$, $\rho = 1$, $\Delta t = 0.001$, and $N = 50$ at different time levels.

x	t = 0.1			t = 0.5		
	u^{ext}	u^{num}	Absolute Error	u^{ext}	u^{num}	Absolute Error
-3.1416	-1.108106248E-16	-1.548379480E-13	1.5495E-13	-7.427858310E-17	-1.483466128E-13	1.4827E-13
-2.5133	-0.531850090044	-0.531850049240	4.0804E-08	-0.356509776842	-0.356509639736	1.3710E-07
-1.8850	-0.860551522611	-0.860551455900	6.6711E-08	-0.576844936252	-0.576844713105	2.2315E-07
-1.2566	-0.860551522611	-0.860551455886	6.6724E-08	-0.576844936252	-0.576844712766	2.2349E-07
-0.6283	-0.531850090044	-0.531850048806	4.1238E-08	-0.356509776842	-0.356509638659	1.3818E-07
0	4.018285340E-16	-2.983724379E-16	7.0020E-16	2.693537214E-16	-2.777292285E-15	3.0466E-15
0.6283	0.531850090044	0.531850048806	4.1238E-08	0.356509776842	0.356509638659	1.3818E-07
1.2566	0.860551522611	0.860551455886	6.6724E-08	0.576844936252	0.576844712766	2.2349E-07
1.8850	0.860551522611	0.860551455900	6.6711E-08	0.576844936252	0.576844713105	2.2315E-07
2.5133	0.531850090044	0.531850049240	4.0805E-08	0.356509776842	0.356509639736	1.3710E-07
3.1416	-6.928464432E-16	-2.775557561E-17	6.6509E-16	-4.644288597E-16	1.734723476E-18	4.6616E-16

TABLE 3. Comparison of order of convergence with $\Delta t = 0.001$.

N	OCSCM (t = 0.1)		CSCM [22] (t = 0.1)		OCSCM (t = 0.5)		CSCM [22] (t = 0.5)	
	$L_\infty(u)$	Order	$L_\infty(u)$	Order	$L_\infty(u)$	Order	$L_\infty(u)$	Order
8	1.1310E-4	-	-	-	2.2378e-04	-	-	-
16	6.9261E-6	4.0294	-	-	1.3089e-05	4.0957	-	-
32	3.8023E-7	4.1871	2.9104E-4	-	8.0682E-7	4.0200	9.7478E-4	-
64	2.0876E-8	4.1870	7.2704E-5	2.001	5.0343E-8	4.0024	2.4361E-4	2.005
128	1.9994E-9	3.3842	1.8178E-5	1.999	5.0162E-9	3.3271	6.0896E-5	2.001

TABLE 4. Order of convergence with $\Delta t = 0.005$, and $t = 1$.

N	$L_\infty(u)$	Order	$L_2(u)$	Order
8	2.1807E-4	-	2.1402E-4	-
16	1.3446E-5	4.0195	1.3030E-5	4.0378
32	9.2700E-7	3.8584	9.2669E-7	3.8136
64	8.6120E-8	3.4282	8.6118E-8	3.4277



TABLE 5. Comparison of $L_\infty(u)$ error norm of Example 1 with $\alpha = -2$, $\eta = 1$, $\rho = 1$, $\Delta t = 0.01$, and $N = 50$ at different time levels.

t	OCSCM	DQM [23]	GQFEM [20]	MCSCM [25]	TCSCM [11]
0.5	2.7315E-6	1.5169E-4	2.2336E-5	1.1031E-4	3.7144E-4
1.0	3.3133E-6	1.8397E-4	1.4618E-5	1.3369E-4	4.5072E-4
2.0	2.4377E-6	1.3525E-4	0.7380E-5	9.8182E-5	3.3183E-4
3.0	1.3451E-6	7.4601E-5	0.4027E-5	1.02995E-5	1.8322E-4

TABLE 6. Comparison of $L_\infty(u)$ and $L_2(u)$ error norms of Example 1 with $\alpha = -2$, $\eta = 1$, $\rho = 1$, and $\Delta t = 0.001$ at different time levels.

t	0.1		0.5		1.0	
	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$	$L_\infty(u)$	$L_2(u)$
OCSCM ($N = 32$)	3.8023E-7	4.1669E-7	1.2682E-6	2.0812E-6	1.5348E-6	4.1620E-6
CPU time(s)	0.05921	0.05797	0.13794	0.13998	0.22829	0.23394
CSCM [22] ($N = 200$)	7.4500E-6	8.2100E-6	4.1000E-5	2.4900E-5	8.2100E-5	3.0000E-5
CSCM [22] ($N = 400$)	1.8600E-6	2.0500E-6	6.2200E-6	1.0200E-5	7.5600E-6	2.0400E-5
IFDM [42] ($N = 200$)	5.3000E-5	5.8600E-5	1.7900E-4	2.9004E-4	2.1700E-4	5.9100E-4
HWM [19] ($N = 32$)	3.7650E-5	3.2690E-8	7.2700E-6	1.0540E-8	2.3810E-5	5.0290E-7
CPU time(s)	0.145	0.153	0.506	0.539	0.961	0.992

TABLE 7. Comparison of $L_\infty(u)$ and $L_\infty(w)$ error norms of Example 1 with $\alpha = -20$, $\eta = 10$, $\rho = 10$, $N = 16$ and $\Delta t = 0.001$ at different time levels.

t	OCSCM			HWM [19]		
	$L_\infty(u)$	$L_\infty(w)$	CPU time(s)	$L_\infty(u)$	$L_\infty(w)$	CPU time(s)
0.1	3.8023E-7	3.8023E-7	0.071487	3.765E-5	3.765E-5	0.15
0.5	1.2682E-6	1.2682E-6	0.176013	7.270E-6	7.270E-6	0.52
1.0	1.5348E-6	1.5348E-6	0.294007	2.381E-4	2.381E-4	0.95
2.0	1.1279E-6	1.1279E-6	0.387541	8.125E-4	8.125E-4	1.84

TABLE 8. Comparison of $L_\infty(u)$ and $L_\infty(w)$ error norms of Example 1 with $\alpha = -20$, $\eta = 10$, $\rho = 10$, $N = 16$, and $\Delta t = 0.0001$ at different time levels.

t	OCSCM			HWM [19]		
	$L_\infty(u)$	$L_\infty(w)$	CPU time(s)	$L_\infty(u)$	$L_\infty(w)$	CPU time(s)
0.2	6.7385E-7	6.7399E-7	0.412670	2.645E-6	2.645E-6	2.28
0.4	1.1002E-6	1.1002E-6	0.706992	2.101E-5	2.101E-5	11.13
0.6	5.3535E-5	5.3535E-5	0.963793	8.753E-5	8.753E-5	22.44
0.5	1.4711E-6	1.4710E-6	1.260307	2.245E-5	2.245E-5	35.79



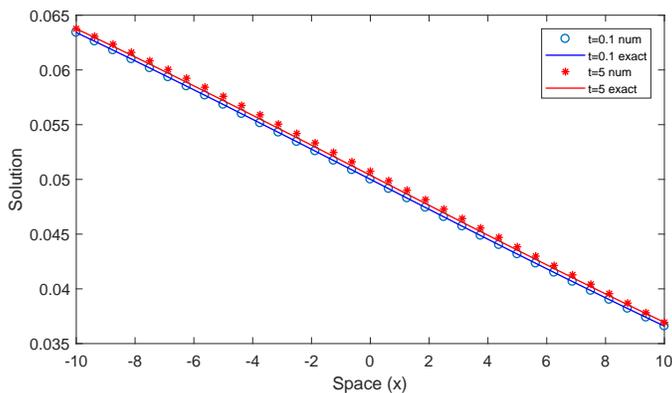


FIGURE 3. Comparison of numerical and exact solution of Example 2 at different time levels with $N = 32$ and $\Delta t = 0.01$.

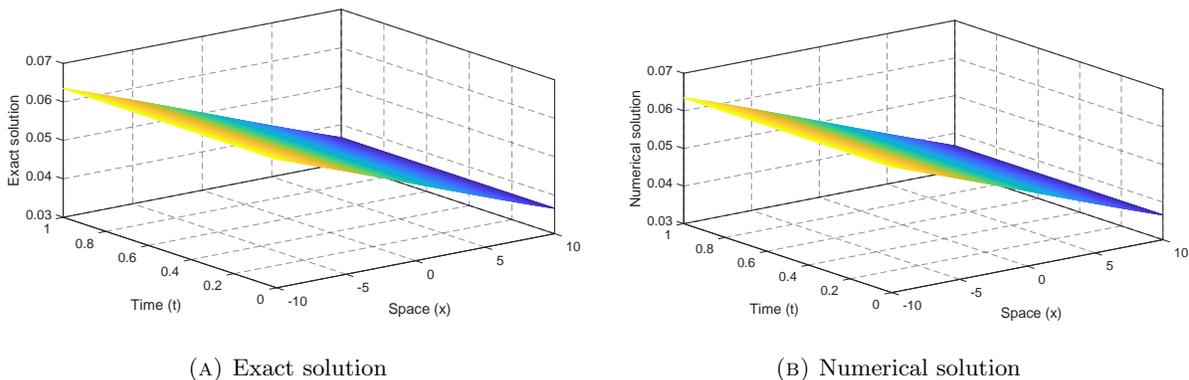


FIGURE 4. 3-D plots of Example 2 with $N = 100$, $\Delta t = 0.005$, and $t = 1$.

TABLE 9. Absolute error of Example 2 with $\alpha = 2$, $\eta = 0.1$, $\rho = 0.3$, $N = 100$ and $\Delta t = 0.001$ at different time levels.

x	$t = 0.1$			$t = 0.5$		
	u^{ext}	u^{num}	Absolute Error	u^{ext}	u^{num}	Absolute Error
-10	0.0634205770475	0.0634205770475	2.3176E-15	0.0634486431326	0.0634486431326	5.6482E-14
-8	0.0608331108080	0.0608415529515	8.4421E-06	0.0608619370148	0.0609036066266	4.1670E-05
-6	0.0581832977931	0.0581915285569	8.2308E-06	0.0582127345817	0.0582539176980	4.1183E-05
-4	0.0554853951137	0.0554933475837	7.9525E-06	0.0555152790429	0.0555550768408	3.9798E-05
-2	0.0527547700412	0.0527623832857	7.6132E-06	0.0527849272084	0.0528230338414	3.8107E-05
0	0.0500075624999	0.0500147834711	7.2210E-06	0.0500378124928	0.0500739617361	3.6149E-05
2	0.0472603092976	0.0472670943564	6.7851E-06	0.0472904694723	0.0473244420603	3.3972E-05
4	0.0445295483399	0.0445358643071	6.3160E-06	0.0445594382121	0.0445910671461	3.1629E-05
6	0.0418314227888	0.0418372474662	5.8247E-06	0.0418608683152	0.0418900413952	2.9173E-05
8	0.0391813051308	0.0391866272811	5.3222E-06	0.0392101426680	0.0392365453825	2.6403E-05
10	0.0365934594148	0.0365934594148	6.9389E-18	0.0366215391690	0.0366215391690	6.9389E-18



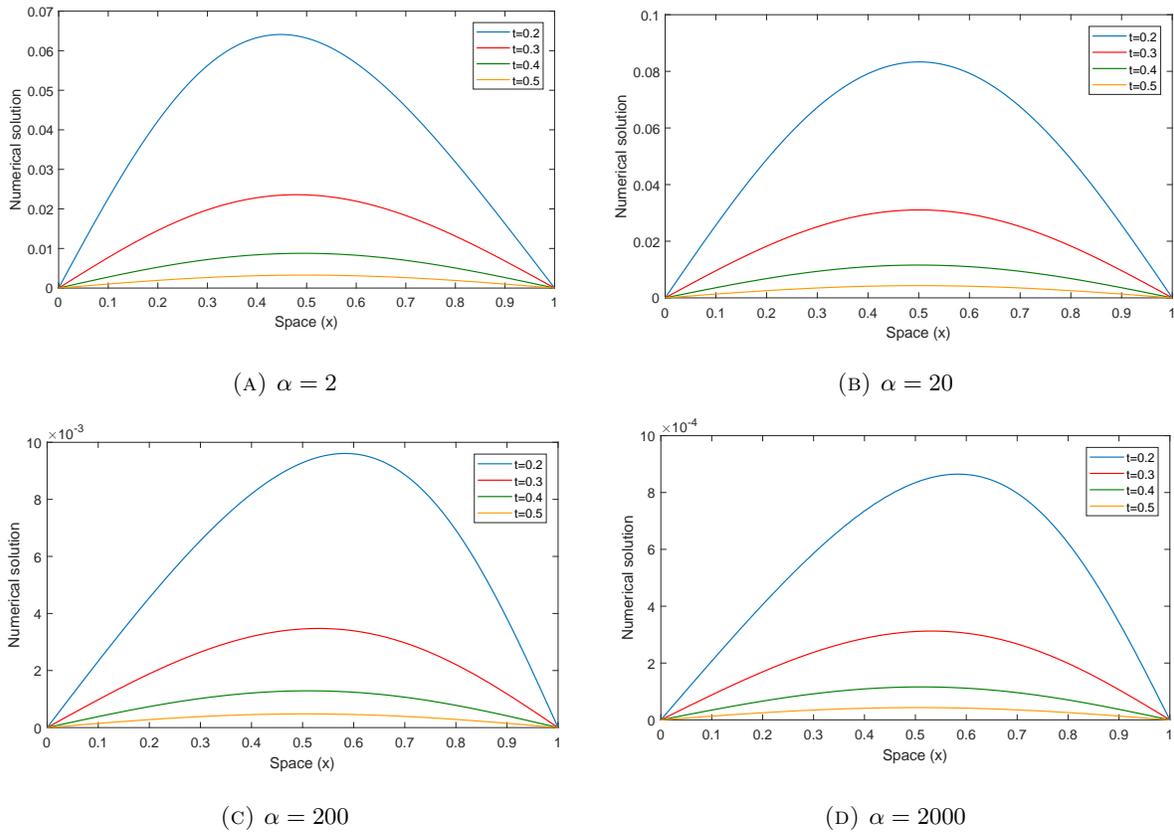


FIGURE 5. 2-D plots of Example 3 with $N = 100$, $\Delta t = 0.001$, and $\eta = \rho = 10$.

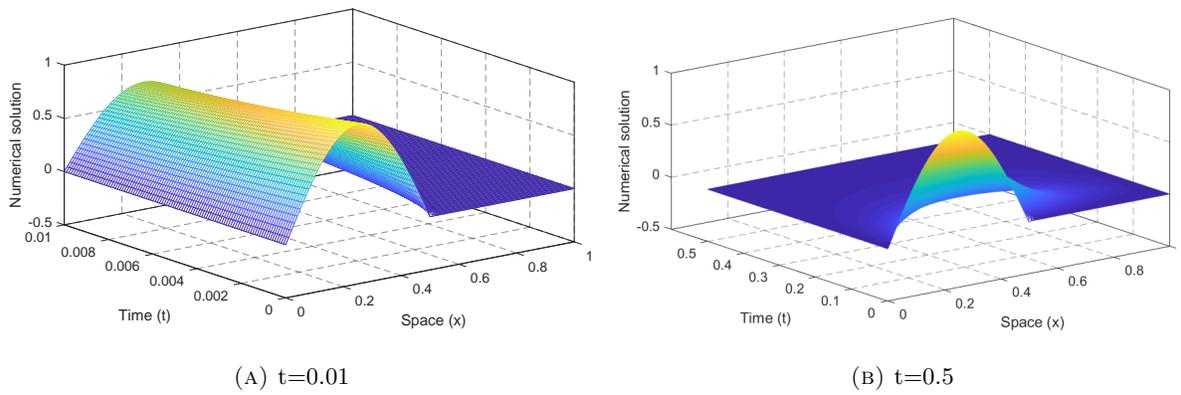


FIGURE 6. 3-D plots of Example 3 with $N = 100$, $\Delta t = 0.001$, $\alpha = 2$, and $\eta = \rho = 10$.



TABLE 10. Comparison of $L_\infty(u)$ error norm of Example 2 with $\alpha = 2$ and $\Delta t = 0.01$.

t	η	ρ	OCSCM ($N = 16$)	CCM [18] ($N = 20$)	FPSM [28] ($N = 20$)	CSCM [22] ($N = 100$)	DQM [23] ($N = 21$)	HWM [19] ($N = 16$)
0.5	0.1	0.3	4.1923E-5	4.43E-5	9.619E-4	4.167E-5	4.173E-5	5.709E-5
	0.3	0.03	4.5905E-5	4.48E-5	4.310E-4	4.590E-5	4.585E-5	1.682E-5
1.0	0.1	0.3	8.2569E-5	8.66E-5	1.153E-3	8.258E-5	8.275E-5	1.100E-5
	0.3	0.03	9.1816E-5	9.16E-5	1.268E-3	9.182E-5	9.167E-5	3.223E-5

TABLE 11. Comparison of $L_\infty(w)$ error norm of Example 2 with $\alpha = 2$ and $\Delta t = 0.01$ at different time levels.

t	η	ρ	OCSCM ($N = 16$)	CCM [18] ($N = 20$)	FPSM [28] ($N = 20$)	CSCM [22] ($N = 100$)	DQM [23] ($N = 21$)	HWM [19] ($N = 16$)
0.5	0.1	0.3	2.1581E-5	4.99E-5	3.332E-4	1.480E-4	5.418E-5	3.697E-5
	0.3	0.03	1.8093E-4	1.81E-4	1.148E-3	5.729E-4	2.826E-5	2.639E-5
1.0	0.1	0.3	4.2137E-5	9.92E-5	1.162E-3	4.770E-5	1.074E-4	6.940E-5
	0.3	0.03	3.6173E-4	3.62E-4	1.638E-3	3.617E-4	5.673E-5	5.219E-4

TABLE 12. Comparison of $L_\infty(u)$ and $L_\infty(w)$ error norms of Example 2 with $\alpha = 2$, $\eta = 0.1$, $\rho = 0.3$, and $\Delta t = 0.001$ at different time levels.

t	OCSCM ($N = 16$)			HWM [19] ($N = 64$)		
	$L_\infty(u)$	$L_\infty(w)$	CPU time(s)	$L_\infty(u)$	$L_\infty(w)$	CPU time(s)
0.5	4.1923E-5	2.1581E-5	0.053673	5.675E-5	3.679E-5	3.064
1	8.2569E-5	4.2137E-5	0.065548	2.085E-5	1.359E-5	6.131
2	1.6216E-4	7.9579E-5	0.070561	2.085E-4	1.359E-4	11.95
3	2.4079E-4	1.1535E-4	0.095301	3.006E-4	2.049E-4	18.33

TABLE 13. Comparison of $L_\infty(u)$ error norm and order of convergence of Example 3 with $\alpha = 2$ at $t = 0.1$.

N	OCSCM		CSCM [22]	
	$L_\infty(u)$	Order	$L_\infty(u)$	Order
50	9.9540E-3	-	0.018812	-
100	1.0123E-3	3.2976	0.005508	1.772
200	1.0541E-4	3.2637	0.001649	1.740



REFERENCES

- [1] R. Abazari and A. Borhanifar, *Numerical study of the solution of the Burgers and coupled Burgers equations by a differential transformation method*, *Comput. Math. Appl.*, 59(8) (2010), 2711–2722.
- [2] M. A. Abdou and A. A. Soliman, *Variational iteration method for solving Burger's and coupled Burger's equations*, *J. Comput. Appl. Math.*, 181(2) (2005), 245–251.
- [3] K. K. Ali, K. R. Raslan, and T. S. El-Danaf, *Non-polynomial spline method for solving coupled Burgers equations*, *Comput. Methods Differ. Equ.*, 3(3) (2015), 218–230.
- [4] E. Ashpazzadeh, B. Han, and M. Lakestani, *Biorthogonal multiwavelets on the interval for numerical solutions of Burgers' equation*, *J. Comput. Appl. Math.*, 317 (2017), 510–534.
- [5] A. Bashan, *A numerical treatment of the coupled viscous Burgers' equation in the presence of very large Reynolds number*, *Physica A*, 545 (2020), 123755.
- [6] R. E. Bellman and R. E. Kalaba, *Quasilinearization and nonlinear boundary-value problems*, New York : Elsevier, 1965.
- [7] H. P. Bhatt and A. Q. M. Khaliq, *Fourth-order compact schemes for the numerical simulation of coupled Burgers' equation*, *Comput. Phys. Commun.*, 200 (2016), 117–138.
- [8] N. Chuathong and S. Kaennakham, *Numerical solution to coupled Burgers' equations by Gaussian-based Hermite collocation scheme*, *J. Appl. Math.*, 2018.
- [9] J. W. Daniel and B. K. Swartz, *Extrapolated collocation for two-point boundary-value problems using cubic splines*, *IMA J. Appl. Math.*, 16(2) (1975), 161–174.
- [10] M. Dehghan, B. N. Saray, and M. Lakestani, *Mixed finite difference and Galerkin methods for solving Burgers equations using interpolating scaling functions*, *Math. Methods Appl. Sci.*, 37(6) (2014), 894–912.
- [11] O. Ersoy and I. Dag, *A trigonometric cubic B-spline finite element method for solving the nonlinear coupled Burger equation*, arXiv preprint arXiv:1604.04419, 2016.
- [12] S. E. Esipov, *Coupled Burgers equations: a model of polydisperse sedimentation*, *Phys. Rev. E*, 52(4) (1995), 3711.
- [13] N. Fisher and B. Bialecki, *Extrapolated ADI Crank–Nicolson orthogonal spline collocation for coupled Burgers' equations*, *J. Differ. Equ. Appl.*, 26(1) (2020), 45–73.
- [14] H. E. Gadain, *Solving coupled pseudo-parabolic equation using a modified double Laplace decomposition method*, *Acta Math. Sci.*, 38(1) (2018), 333–346.
- [15] M. Ghasemi, *An efficient algorithm based on extrapolation for the solution of nonlinear parabolic equations*, *Int. J. Nonlin. Sci. Num.*, 19(1) (2018), 37–51.
- [16] A. Jafarabadi and E. Shivanian, *Numerical simulation of nonlinear coupled Burgers' equation through meshless radial point interpolation method*, *Eng. Anal. Bound. Elem.*, 95 (2018), 187–199.
- [17] D. Kaya, *An explicit solution of coupled viscous Burgers' equation by the decomposition method*, *Int. J. Math. Math. Sci.*, 27(11) (2001), 675–680.
- [18] A. H. Khater, R. S. Temsah, and M. Hassan, *A Chebyshev spectral collocation method for solving Burgers'-type equations*, *J. Comput. Appl. Math.*, 222(2) (2008), 333–350.
- [19] M. Kumar and S. Pandit, *A composite numerical scheme for the numerical simulation of coupled Burgers' equation*, *Comput. Phys. Commun.*, 185(3) (2014), 809–817.
- [20] S. Kutluay and Y. Ucar, *Numerical solutions of the coupled Burgers' equation by the Galerkin quadratic B-spline finite element method*, *Math. Methods Appl. Sci.*, 36(17) (2013), 2403–2415.
- [21] T. R. Lucas, *Error bounds for interpolating cubic splines under various end conditions*, *SIAM J. Numer. Anal.*, 11(3) (1974), 569–584.
- [22] R. C. Mittal and G. Arora, *Numerical solution of the coupled viscous Burgers' equation*, *Commun. Nonlinear. Sci.*, 16(3) (2011), 1304–1313.
- [23] R. C. Mittal and R. Jiwari, *Differential quadrature method for numerical solution of coupled viscous Burgers' equations*, *Int. J. Comput. Methods Eng. Sci. Mech.*, 13(2) (2012), 88–92.
- [24] R. C. Mittal and R. Rohila, *A fourth order cubic B-spline collocation method for the numerical study of the RLW and MRLW equations*, *Wave motion*, 80 (2018), 47–68.



- [25] R. C. Mittal and A. Tripathi, *A collocation method for numerical solutions of coupled Burgers' equations*, Int. J. Comput. Methods Eng. Sci. Mech., 15(5) (2014), 457–471.
- [26] J. Nee and J. Duan, *Limit set of trajectories of the coupled viscous Burgers' equations*, Appl. Math. Lett., 1(1) (1998), 57–61.
- [27] P. M. Prenter, *Splines and Variational Methods*, New York : Wiley-interscience publication, 1975.
- [28] A. Rashid and A. I. B. M. Ismail, *A Fourier pseudospectral method for solving coupled viscous Burgers equations*, Comput. Methods. Appl. Math., 9(4) (2009), 412–420.
- [29] K. R. Raslan, T. S. El-Danaf, and K. K. Ali, *Collocation method with cubic trigonometric B-spline algorithm for solving coupled Burgers' equation*, Far East J. Appl. Math., 95(2) (2016), 109.
- [30] R. Rohila and R. C. Mittal, *Numerical study of reaction diffusion Fisher's equation by fourth order cubic B-spline collocation method*, Math. Sci., 12(2) (2018), 79–89.
- [31] P. Roul, *A fourth-order non-uniform mesh optimal B-spline collocation method for solving a strongly nonlinear singular boundary value problem describing electrohydrodynamic flow of a fluid*, Appl. Numer. Math., 153 (2020), 558–574.
- [32] P. Roul, *A fourth order numerical method based on B-spline functions for pricing Asian options*, Comput. Math. Appl., 80(3) (2020), 504–521.
- [33] P. Roul and V. P. Goura, *B-spline collocation methods and their convergence for a class of nonlinear derivative dependent singular boundary value problems*, Appl. Math. Comput., 341 (2019), 428–450.
- [34] P. Roul and V. P. Goura, *A high-order B-spline collocation scheme for solving a nonhomogeneous time-fractional diffusion equation*, Math. Methods Appl. Sci., 44(1) (2021), 546–567.
- [35] B. N. Saray, M. Lakestani, and M. Dehghan, *On the sparse multiscale representation of 2-D Burgers equations by an efficient algorithm based on multiwavelets*, Numer. Methods Partial Differ. Equ., (2021).
- [36] M. A. Shallal, K. K. Ali, K. R. Raslan, and A. H. Taqi, *Septic B-spline collocation method for numerical solution of the coupled Burgers' equations*, Arab. J. Basic Appl. Sci., 26(1) (2019), 331–341.
- [37] Shallu and V. K. Kukreja, *Analysis of RLW and MRLW equation using an improvised collocation technique with SSP-RK43 scheme*, Wave Motion, (2021), p.102761.
- [38] Shallu and V. K. Kukreja, *An improvised collocation algorithm with specific end conditions for solving modified Burgers equation*, Numer. Methods Partial Differ. Equ., 37(1) (2021), 874–896.
- [39] Shallu, A. Kumari, and V. K. Kukreja, *An efficient superconvergent spline collocation algorithm for solving fourth order singularly perturbed problems*, Int. J. Appl. Comput. Math., 6(5) (2020), 1–23.
- [40] Shallu, A. Kumari, and V. K. Kukreja, *An improved extrapolated collocation technique for singularly perturbed problems using cubic B-spline Functions*, Mediterr. J. Math., 18(4) (2021), 1–29.
- [41] A. A. Soliman, *The modified extended tanh-function method for solving Burgers-type equations*, Physica A, 361(2) (2006), 394–404.
- [42] V. K. Srivastava, M. K. Awasthi, and M. Tamsir, *A fully implicit finite-difference solution to one dimensional coupled nonlinear Burgers' equations*, Int. J. Math. Math. Sci., 7(4) (2013), 23.
- [43] H. Zadvan and J. Rashidinia, *Development of non polynomial spline and new B-spline with application to solution of Klein-Gordon equation*, Comput. Methods Differ. Equ., 8(4) (2020), 794–814.

