

Numerical Solution of Burger's Type Equation Using Finite Element Collocation Method with Strang Splitting

Yusuf Uçar, Nuri Murat Yağmurlu* and İhsan Çelikkaya

Abstract

The nonlinear Burgers equation, which has a convection term, a viscosity term and a time dependent term in its structure, has been split according to the time term and then has been solved by finite element collocation method using cubic B-spline bases. By splitting the equation $U_t + UU_x = vU_{xx}$ into two simpler sub problems $U_t + UU_x = 0$ and $U_t - vU_{xx} = 0$ have been obtained. A discretization process has been performed for each of these sub-problems and the stability analyzes have been carried out by Fourier (von Neumann) series method. Then, both sub-problems have been solved using the Strang splitting technique to obtain numerical results. To see the effectiveness of the present method, which is a combination of finite element method and Strang splitting technique, we have calculated the frequently used error norms $\|e\|_1$, L_2 and L_∞ in the literature and have made a comparison between exact and a numerical solution.

Keywords: Strang Splitting; Burgers Equation; Collocation method; Finite Element method; Cubic B-Spline Basis.

AMS Subject Classification (2020): Primary: 35Q51 ; Secondary: 74J35; 33F10.

*Corresponding author

1. Introduction

Nonlinear Burgers equation is in the form of

$$U_t + UU_x = vU_{xx}, \quad a < x < b, \quad t \geq t_0, \quad (1.1)$$

where $U = U(x, t)$ is a differentiable smooth function of space variable x and time variable t and $v > 0$ kinematics viscosity coefficient. The Burgers equation, which derives two basic steady solutions, was first introduced by Bateman [1] while studying fluid mechanics. This equation is known as the simplest mathematical model expressing the balance between convection and diffusion. Then, the equation (1.1) has been studied by Burgers [10, 11] and is referred to as Burgers equation after this study. This equation is an equation involving both nonlinearity and dissipation in the simplest possible way and can be thought of as a nonlinear type of heat equation [9]. If $v = 0$ is taken in Equation (1.1), the inviscid Burgers equation expressing the shock waves and having many important applications in physics is obtained [16]. The reasons for the Burgers equation to attract the attention of many researchers are that it includes the nonlinear advection term UU_x in the simplest form, the dissipation term vU_{xx} that models the physical wave phenomena, the shock wave property for the small values of the kinematic viscosity coefficient v , the comparison with the exact solution [12] obtained by [9]. The Burgers equation is an important partial derivative differential equation with a widespread usage in mathematical physics. The increase in interest in nonlinear science makes this equation an observation of scientists because it describes very well a lot of gas dynamics, heat conduction, traffic flow and shock waves models [34]. Numerous mathematical techniques and

methods have been developed in recent years to obtain numerical solutions of Burgers equation and to make comparisons with both analytical and numerical solutions.

Among others, Jawad et al. [2] have dedicated one section of their study on the perturbed Burgers equation. Nerameh and Eslami [3] have obtained some new analytical solutions, such as trigonometric function, rational function, and hyperbolic function solutions by using new extension of the (G'/G) -expansion method to the coupled (2+1)-dimensional Painleve integrable Burgers equation with the aid of the computer software Maple. Haq et al. [4] have used a meshless method of lines (MMOL) by some radial basis functions (RBFs), Dağ et al. [13] have applied cubic B-splines bases using a linearization technique and collocation finite element method, AHA Ali et al. [5] again have used finite element cubic collocation method, Kutluay et al. [23] have utilized the least-squares quadratic B-spline finite element method with three test problems, R.C. Mittal and R.K. Jain [35] have used a method based on collocation-modified cubic B-splines using SSP-RK43 or SSP-RK54 scheme techniques. G. Arora and B.K. Singh [6] have utilized modified cubic-B-spline differential quadrature method (MCB-DQM). C.G. Zhu and R.H. Wang [20] solved the problem using cubic B-spline quasi-interpolation method. Dağ et al. [14] have used both time- and space-split Burgers' equations with quadratic B-spline collocation method. İ Dag and A. Korkmaz [15] have utilized cubic B-spline differential quadrature methods with Runge-Kutta method of order four in time, M.A. Ramadan et al. [37] have obtained the numerical solutions of the nonlinear Burgers equation by a method based on collocation of septic B-splines over finite elements. R. Jiwari [21] has used uniform Haar wavelets and the quasilinearization process. I.A. Ganaie and V.K. Kukreja [17] solved the non-linear Burgers equation by numerically using cubic Hermite collocation method (CHCM). S. Kutluay and A. Esen [24] have solved the problem using a lumped Galerkin method with quadratic B-spline finite elements. Moreover, one can find various studies on numerical solutions of the Burgers' equation in Refs ([25]-[32]).

In this study, we are going to obtain the numerical solution of the Burgers equation by combining the splitting techniques, which are easy and effective to apply, with the finite element collocation method. For this, we will split the equation (1.1) by time and convert into two subproblems with a simpler structure, one containing the convection term and the other the diffusion term. Next, approximations to each of the sub-problem are obtained by cubic B-spline bases and ordinary differential equation systems are obtained and solved by finite element collocation method.

2. Splitting Techniques

Many physical phenomena in the natural world can be expressed by means of differential equations. In recent years, with the development of computers, almost every physical and engineering problem can be expressed with differential equations. In many cases the equation that models the physical phenomenon is complex and it is difficult to find an effective method to solve this equation. Operator splitting techniques, which have recently been used to overcome such complex problems, have been a very successful approach. The basic idea of this approach is to formally write all the evolution operators in the mathematical model as the sum of several evolution operators. We will concentrate on the situation of two linear operators. Now, let us consider the following Cauchy problem

$$\frac{du(t)}{dt} = Cu(t), \quad t \in [0, T], \quad u(0) = u_0 \quad (2.1)$$

and assume that it is split as $C = A + B$. Eq. (2.1) can be seen as a semi-discretization of a linear PDE with a homogeneous periodic boundary condition. Here, an initial function $u_0 \in X$ is assumed to be a finite linear operator in the Banach space X , with $C = A + B$, $A, B : X \rightarrow X$. We also have a norm associated with the X -space, and if it is a matrix, then Euclid norm [18]. $u(t_{n+1}) = e^{tC}u(t_n)$ is the formal solution of the equation (2.1). Then, $\Delta t = t_{n+1} - t_n$ is being the most basic type splitting technique, we obtain

$$u(t_{n+1}) \simeq e^{\Delta t B} e^{\Delta t A} u(t_n) \quad (2.2)$$

Here if A and B are commutative operators, then the method gives the exact solution. (2.2) is the most basic splitting technique and presents the solutions of the following two sub-problems

$$\begin{aligned} \frac{du(t)}{dt} &= Au^*(t), \quad u^*(0) = u_0 \text{ on } [0, \Delta t], \\ \frac{du(t)}{dt} &= Bu^{**}(t), \quad u^{**}(0) = u^*(\Delta t) \text{ on } [0, \Delta t] \end{aligned}$$

Thus, the solutions at the desired time step are calculated by $u^{**}(\Delta t)$. This technique is called $(A - B)$ splitting scheme. It can easily be obtained the splitting scheme $(B - A)$ by exchanging the locations of the operators A and B in the splitting scheme $(A - B)$ [39].

2.1 Second order symmetric Strang splitting

In order to obtain a better accuracy, first of all Strang[40] scheme

$$u(\Delta t) = \frac{1}{2} [u_{AB}(\Delta t) + u_{BA}(\Delta t)] \quad (2.3)$$

is considered. Here u_{AB} and u_{BA} are the solutions obtained using $(A - B)$ and $(B - A)$ splitting schemes, respectively. Since each operator needs to be calculated twice in this scheme, the calculation cost is high. In place of (2.3), a more cost-effective symmetric scheme is preferred $u(t_{n+1}) \simeq \left(e^{\frac{\Delta t}{2}A} e^{\frac{\Delta t}{2}B} \right) \left(e^{\frac{\Delta t}{2}B} e^{\frac{\Delta t}{2}A} \right) u(t_n) = e^{\frac{\Delta t}{2}A} e^{\Delta t B} e^{\frac{\Delta t}{2}A} u(t_n)$. This scheme is described explicitly as follows

$$\begin{aligned} \frac{du^*(t)}{dt} &= Au^*(t), \quad u^*(0) = u_0 \text{ on } [0, \Delta t/2], \\ \frac{du^{**}(t)}{dt} &= Bu^{**}(t), \quad u^{**}(0) = u^*(\Delta t/2) \text{ on } [0, \Delta t], \\ \frac{du^{***}(t)}{dt} &= Au^{***}(t), \quad u^{***}(0) = u^{**}(\Delta t) \text{ on } [0, \Delta t/2]. \end{aligned} \quad (2.4)$$

Finally, the desired numerical solutions are calculated by $u^{***}(\Delta t/2)$. As it is seen from the condition (2.4), while the term $u^*(0)$ is calculated from the original initial condition of the problem, the other two initial conditions use the previously calculated solutions as initial conditions. If (2.4) is called as the scheme $(A - B - A)$, then in a similar way, the scheme $(B - A - B)$ can also be constructed.

3. Cubic B-spline collocation method

Let us assume that the space solution domain is $[a, b]$ and a uniform discretization of this domain by the nodal points $x_m, m = 0, 1, \dots, N$, is given by $a = x_0 < x_1 < \dots < x_N = b$. If we define the distance between two successive points by $h = x_{m+1} - x_m$, and $\Phi_m(x), m = -1(1)N + 1$, then cubic B-spline functions on the domain $[a, b]$ can be presented as follows in terms of nodal points x_m

$$\Phi_m(x) = \frac{1}{h^3} \begin{cases} (x - x_{m-2})^3, & x \in [x_{m-2}, x_{m-1}] \\ h^3 + 3h^2(x - x_{m-1}) + 3h(x - x_{m-1})^2 - 3(x - x_{m-1})^3, & x \in [x_{m-1}, x_m] \\ h^3 + 3h^2(x_{m+1} - x) + 3h(x_{m+1} - x)^2 - 3(x_{m+1} - x)^3, & x \in [x_m, x_{m+1}] \\ (x_{m+2} - x)^3, & x \in [x_{m+1}, x_{m+2}] \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

as stated by Prenter [36]. It is obvious that the set $\{\Phi_{-1}(x), \Phi_0(x), \dots, \Phi_{N+1}(x)\}$ constitutes a base on the domain $[a, b]$. If we assume that the function $u(x, t)$ is defined on the domain $[a, b]$, then the function $u(x, t)$ can be approximated as follows in terms of cubic B-spline functions and time dependent parameters $\delta_m(t)$ as follows

$$U(x, t) \cong \sum_{m=-1}^{N+1} \delta_m(t) \Phi_m(x). \quad (3.2)$$

Here time-dependent parameters $\delta_m(t)$ are going to be determined using the Eq. (1.1) and its auxiliary conditions. Since the Eq.(1.1) contains the terms u , the first and second derivatives of u , we need the values of u , its first and second order derivatives in terms of cubic B-spline functions using the approximations (3.1) and (3.2) using the time-dependent parameters $\delta_m(t)$ Thus the nodal values are obtained as follows

$$\begin{aligned} U_m &= U(x_m) = \delta_{m-1} + 4\delta_m + \delta_{m+1} \\ U'_m &= U'(x_m) = \frac{3}{h} (\delta_{m+1} - \delta_{m-1}) \\ U''_m &= U''(x_m) = \frac{6}{h^2} (\delta_{m-1} - 2\delta_m + \delta_{m+1}) \end{aligned} \quad (3.3)$$

Here ' and '' describe the first and second order derivatives with respect to space variable x .

4. Application of the method

The time split form of the Burgers (1.1) equation is as follows

$$u_t - vu_{xx} = 0, \quad (4.1)$$

$$u_t + uu_x = 0. \quad (4.2)$$

If we use the values of U, U' and U'' given by (3.3) in equations given by (4.1) and (4.2), we obtain the following first order system of ordinary differential equations

$$\dot{\delta}_{m-1} + 4\dot{\delta}_m + \dot{\delta}_{m+1} - \frac{6v}{h^2} (\delta_{m-1} - 2\delta_m + \delta_{m+1}) = 0 \quad (4.3)$$

$$\dot{\delta}_{m-1} + 4\dot{\delta}_m + \dot{\delta}_{m+1} + \frac{3z_m}{h} (\delta_{m+1} - \delta_{m-1}) = 0. \quad (4.4)$$

Here the symbol \circ denotes the first order derivative with respect to time variable t and linearized as follows

$$z_m = \delta_{m-1} + 4\delta_m + \delta_{m+1}.$$

If we write Crank-Nicolson approximation $\frac{\delta_m^{n+1} + \delta_m^n}{2}$ in place of the parameters δ_m in Eqs. (4.3) and (4.4), forward difference equations $\frac{\delta_m^{n+1} - \delta_m^n}{\Delta t}$ in place of the parameters $\dot{\delta}_m$, we obtain the following equations, respectively

$$k_1 \delta_{m-1}^{n+1} + k_2 \delta_m^{n+1} + k_3 \delta_{m+1}^{n+1} = k_3 \delta_{m-1}^n + k_4 \delta_m^n + k_5 \delta_{m+1}^n \quad (4.5)$$

$$k_5 \delta_{m-1}^{n+1} + k_6 \delta_m^{n+1} + k_7 \delta_{m+1}^{n+1} = k_7 \delta_{m-1}^n + k_6 \delta_m^n + k_5 \delta_{m+1}^n \quad (4.6)$$

$$k_1 = 1 - \frac{3v\Delta t}{h^2}, k_2 = 4 + \frac{6v\Delta t}{h^2}, k_3 = 1 + \frac{3v\Delta t}{h^2}, k_4 = 4 - \frac{6v\Delta t}{h^2},$$

$$k_5 = 1 - \frac{3z_m\Delta t}{2h}, k_6 = 4, k_7 = 1 + \frac{3z_m\Delta t}{2h}.$$

The Eqs. (4.5) and (4.6) consist of $(N + 1)$ equations and $(N + 3)$ unknown time dependent parameters $\delta_m, m = 0, 1, \dots, N + 1$. In order to obtain a solvable system, we need to eliminate the time dependent parameters δ_{-1} and δ_{N+1} . For this purpose, we are going to use the boundary conditions $u(a, t) = u(b, t) = 0$ for the system (4.5), and the boundary conditions $u_{xx}(a, t) = u_{xx}(b, t) = 0$ for the system (4.6) and obtain the following equations

$$\delta_{-1} = -4\delta_0 - \delta_1, \delta_{N+1} = -4\delta_N - \delta_{N-1} \quad (4.7)$$

$$\delta_{-1} = 2\delta_0 - \delta_1, \delta_{N+1} = 2\delta_N - \delta_{N-1}. \quad (4.8)$$

If we use the equations (4.7) and (4.8) in the system of equations given by (4.5) and (4.6), respectively; we obtain a three diagonal $(N + 1) \times (N + 1)$ band matrix. Thus a unique solution of these systems is easily obtained using Thomas algorithm. To be able to solve this $(N + 1) \times (N + 1)$ system, we need the initial vector δ_m^0 to start the iterative process. This initial vector is constructed as follows using the initial condition $u(x, 0) = f(x)$ and the approximation given in (3.2) as follows

$$u(x_m, 0) = f(x_m) = U(x_m, 0), \quad m = 0(1)N$$

$$u_m = \delta_{m-1}^0 + 4\delta_m^0 + \delta_{m+1}^0$$

$$u_0 = \delta_{-1}^0 + 4\delta_0^0 + \delta_1^0$$

$$u_1 = \delta_0^0 + 4\delta_1^0 + \delta_2^0$$

$$\vdots$$

$$u_N = \delta_{N-1}^0 + 4\delta_N^0 + \delta_{N+1}^0.$$

For this system to be solvable, the parameters δ_{-1} and δ_{N+1} are eliminated using the boundary conditions $U''(a, 0) = U''(b, 0) = 0$. Thus a three dimensional $(N + 1) \times (N + 1)$ band matrix which can be solved using Thomas algorithms is obtained as follows.

$$\begin{bmatrix} 6 & 0 & 0 & & \\ 1 & 4 & 1 & & \\ & & \ddots & & \\ & & & 1 & 4 & 1 \\ & & & 0 & 0 & 6 \end{bmatrix} \begin{bmatrix} \delta_0^0 \\ \delta_1^0 \\ \vdots \\ \delta_{N-1}^0 \\ \delta_N^0 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \\ u_N \end{bmatrix}$$

4.1 The stability analysis

Firstly to investigate the stability of the scheme given in Eq. (4.5) via von Neumann method [43] the expression $\delta_m^n = e^{i\beta m h} \xi^n$ is written in Eq. (4.5) for halved time step. If the necessary operations are taken, one get, ρ_A and ρ_B are being the notations for the stability condition of Eqs. (4.5) and (4.6), respectively ,

$$(k_1 e^{-i\beta h} + k_2 + k_1 e^{i\beta h}) \xi^{n+1/2} = \xi^n (k_3 e^{-i\beta h} + k_4 + k_3 e^{i\beta h})$$

$$\rho_A \left(\frac{\xi^{n+1/2}}{\xi^n} \right) = \frac{2k_3 \cos \beta h + k_4}{2k_1 \cos \beta h + k_2}$$

$$k_1 = 1 - \frac{3\nu\Delta t}{h^2}, k_2 = 4 + \frac{6\nu\Delta t}{h^2}, k_3 = 1 + \frac{3\nu\Delta t}{h^2}, k_4 = 4 - \frac{6\nu\Delta t}{h^2}.$$

If the avlues of k_1, k_2, k_3 and k_4 are written in their places, the following equations are obtained

$$\rho_A \left(\frac{\xi^{n+1/2}}{\xi^n} \right) = \frac{P - Q}{P + Q}$$

$$P = 2 \cos \beta h + 4, Q = \frac{6\nu\Delta t}{h^2} (1 - \cos \beta h).$$

Here, for the condition $\left| \rho_A \left(\frac{\xi^{n+1/2}}{\xi^n} \right) \right| \leq 1$ to be satisfied, the denominator should be greater than the nominator , that is

$$P + Q \geq P - Q \Rightarrow Q \geq 0$$

is required. Since $Q = \frac{6\nu\Delta t}{h^2} (1 - \cos \beta h) \geq 0$ is always true, the condition $\left| \rho_A \left(\frac{\xi^{n+1/2}}{\xi^n} \right) \right| \leq 1$ is naturally satisfied. Secondly, for the stability of the scheme (4.6), once the U in the term UU_x has been linearized, z_m will then behave as a local constant and the von Neumann method will become feasible to investigate the stability of the difference scheme given in Eq.(4.6). If the expression $\delta_m^n = e^{i\beta m h} \xi^n$ is written in Eq.(4.6) and the necessary operations are performed, the following equation

$$\rho_B \left(\frac{\xi^{n+1}}{\xi^n} \right) = \frac{k_7 e^{-i\beta h} + k_6 + k_5 e^{i\beta h}}{k_5 e^{-i\beta h} + k_6 + k_7 e^{i\beta h}}$$

$$k_5 = 1 - \frac{3z_m\Delta t}{2h}, k_6 = 4, k_7 = 1 + \frac{3z_m\Delta t}{2h}$$

is obtained. After some arrangements, the following equalities

$$\rho_B \left(\frac{\xi^{n+1}}{\xi^n} \right) = \frac{X - iY}{X + iY}$$

$$X = k_6 + (k_5 + k_7) \cos \beta h, Y = (k_7 - k_5) \sin \beta h$$

are obtained. Thus, the following equalities follow

$$\left| \rho_B \left(\frac{\xi^{n+1}}{\xi^n} \right) \right| = \sqrt{\frac{X^2 + Y^2}{X^2 + Y^2}} \leq 1.$$

Therefore, since the scheme obtained by Strang splitting method is as follows

$$\rho(\xi) = \rho_A^{n+1/2} \rho_B^{n+1} \rho_A^{n+1/2}$$

$$|\rho(\xi)| \leq \left| \rho_A \left(\frac{\xi^{n+1/2}}{\xi^n} \right) \right| \left| \rho_B \left(\frac{\xi^{n+1}}{\xi^n} \right) \right| \left| \rho_A \left(\frac{\xi^{n+1/2}}{\xi^n} \right) \right| \leq 1$$

the scheme is unconditionally stable.

5. Numerical results and examples

The numerical solutions of the Burgers equation are going to be sought for three standard test problems frequently found in the literature. The efficiency of the numerical method has been tested using the following error methods between exact and approximate solutions for all test problems. All computations have been carried out using the splitting scheme ($A - B - A$) on a pentium 4 pc using the programming language Fortran with double arithmetic precision

$$L_2 = \sqrt{h \sum_{j=0}^N |(u_j - U_j)^2|}, \quad L_\infty = \max_j |u_j - U_j|$$

$$\|e\|_1 = \frac{1}{N} \sum_{j=1}^{N-1} \frac{|u_j - U_j|}{|u_j|}.$$

Problem 1: As a first test problem, Burgers equation is going to be considered together with the following initial

$$u(x, 0) = \sin \pi x, \quad 0 \leq x \leq 1$$

and boundary conditions

$$u(0, t) = u(1, t) = 0, \quad t \geq 0.$$

The exact solution of this problem has been obtained as a summation of an infinite series by D. Cole [12] as follows

$$u(x, t) = 2\pi v \frac{\sum_j^\infty j a_j \sin(j\pi x) \exp(-j^2 \pi^2 vt)}{a_0 + \sum_j^\infty a_j \cos(j\pi x) \exp(-j^2 \pi^2 vt)}. \quad (5.1)$$

Here a_0 and a_j are Fourier coefficients given as follows

$$a_0 = \int_0^1 e^{-(2\pi v)^{-1}(1-\cos(\pi x))} dx,$$

$$a_n = 2 \int_0^1 e^{-(2\pi v)^{-1}(1-\cos(\pi x))} \cos(n\pi x) dx, \quad n = 1, 2, \dots$$

Table 1. A comparison of the results of Problem 1 for $\Delta t = 0.001$, $v = 1$ and different mesh sizes at time $t = 0.1$.

x	$h = 0.1$	$h = 0.05$	$h = 0.025$	$h = 0.0125$	$h = 0.00625$	Exact
0.1	0.10889	0.10938	0.10950	0.10953	0.10954	0.10954
0.2	0.20849	0.20947	0.20971	0.20977	0.20979	0.20979
0.3	0.28995	0.29141	0.29178	0.29187	0.29189	0.29190
0.4	0.34540	0.34729	0.34777	0.34788	0.34791	0.34792
0.5	0.36862	0.37084	0.37139	0.37153	0.37157	0.37158
0.6	0.35593	0.35826	0.35885	0.35900	0.35903	0.35905
0.7	0.30699	0.30917	0.30972	0.30986	0.30989	0.30991
0.8	0.22554	0.22724	0.22767	0.22778	0.22781	0.22782
0.9	0.11943	0.12037	0.12061	0.12067	0.12068	0.12069
$L_2 \times 10^3$	2.134684	0.537097	0.134897	0.034241	0.009076	
$L_\infty \times 10^3$	3.120196	0.782520	0.196568	0.049987	0.013366	
$ e _1$ (present) $\Delta t = 0.001$	0.007253	0.001920	0.000495	0.000127	0.000034	
$\Delta t = 0.00001$						
$ e _1$ ([13]cub-col.)	0.00734	0.00095	0.00014	0.00003	0.00001	
$ e _1$ ([20]-BSQI)	0.01025	0.00383	0.00082	0.00027	—	
$ e _1$ ([23]-FEM)	0.012165	0.006941	0.003651	0.001858	0.000928	
$ e _1$ ([33]-exp.)	0.007571	0.002025	0.000555	0.000177	—	
$ e _1$ ([33]exact-exp.)	0.007278	0.001885	0.000448	0.000077	—	

Table 2. A comparison of the results of Problem 1 for $\Delta t = 0.001$, $v = 1, 0.1$ and 0.01 at various values of time t .

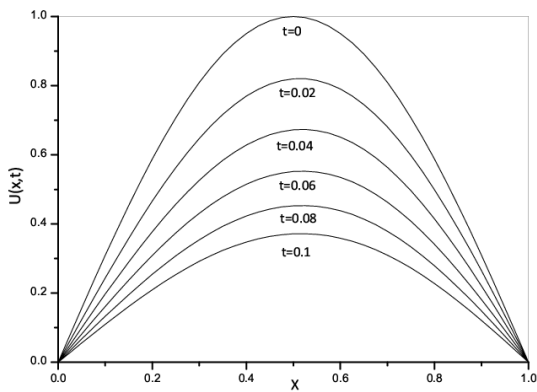
x	t	$v = 1$		$v = 0.1$		$v = 0.01$	
		Numerical	Exact	Numerical	Exact	Numerical	Exact
0.25	0.4	0.01357	0.01357	0.30890	0.30889	0.34192	0.34191
	0.6	0.00189	0.00189	0.24075	0.24074	0.26896	0.26896
	0.8	0.00026	0.00026	0.19569	0.19568	0.22148	0.22148
	1.0	0.00004	0.00004	0.16258	0.16256	0.18819	0.18819
	3.0	0.00000	0.00000	0.02720	0.02720	0.07511	0.07511
0.5	0.4	0.01923	0.01924	0.56965	0.56963	0.66071	0.66071
	0.6	0.00267	0.00267	0.44723	0.44721	0.52942	0.52942
	0.8	0.00037	0.00037	0.35925	0.35924	0.43914	0.43914
	1.0	0.00005	0.00005	0.29192	0.29192	0.37442	0.37442
	3.0	0.00000	0.00000	0.04019	0.04020	0.15018	0.15018
0.75	0.4	0.01362	0.01363	0.62538	0.62544	0.91027	0.91026
	0.6	0.00189	0.00189	0.48715	0.48721	0.76725	0.76724
	0.8	0.00026	0.00026	0.37385	0.37392	0.64740	0.64740
	1.0	0.00004	0.00004	0.28741	0.28747	0.55605	0.55605
	3.0	0.00000	0.00000	0.02976	0.02977	0.22483	0.22481

Table 3. A comparison of numerical solutions of Problem 1 for $h = 0.0125$, $v = 1$ and $t = 0.1$ with various studies.

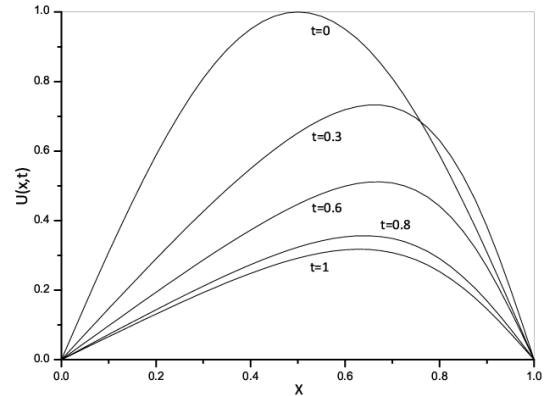
x	Exact	Present	Ref. [13]	Ref. [23]	Ref. [20]	Ref. [6]	Ref. [7]
		$\Delta t = 10^{-3}$	$\Delta t = 10^{-5}$	$\Delta t = 10^{-5}$	$\Delta t = 10^{-5}$	$\Delta t = 10^{-5}$	$\Delta t = 10^{-4}$
		Cub. col.	Least S.	Cub. QI.	Cub.DQ.	Automatic D.	
0.1	0.10954	0.10953	0.10952	0.10978	0.10951	0.109526	0.10955
0.2	0.20979	0.20977	0.20975	0.21019	0.20974	0.209766	0.20981
0.3	0.29190	0.29187	0.29184	0.29238	0.29182	0.291855	0.29193
0.4	0.34792	0.34788	0.34785	0.34845	0.34783	0.347869	0.34796
0.5	0.37158	0.37153	0.37149	0.37212	0.37147	0.371512	0.37163
0.6	0.35905	0.35900	0.35896	0.35960	0.35894	0.358975	0.35910
0.7	0.30991	0.30986	0.30983	0.31044	0.30981	0.309839	0.30995
0.8	0.22782	0.22778	0.22776	0.22827	0.22775	0.227766	0.22786
0.9	0.12069	0.12067	0.12065	0.12097	0.12065	0.120659	0.12071

Table 4. A comparison of the results for values of $v = 1, \Delta t = 0.001, h = 0.0125$.

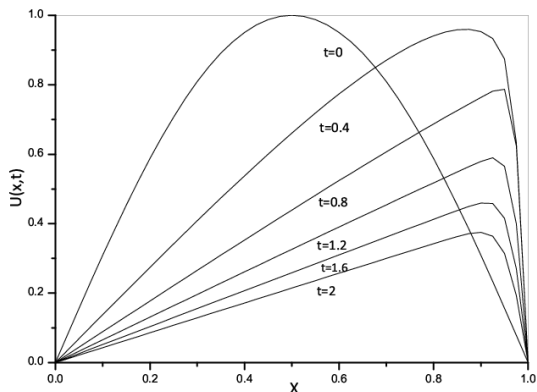
x	t	Exact	Present $\Delta t = 10^{-3}$	Ref. [19] $\Delta t = 10^{-5}$	Ref. [41]-DE $\Delta t = 10^{-5}$	Ref. [41]-HC $\Delta t = 10^{-5}$	Ref. [41]-RHC $\Delta t = 10^{-5}$
0.25	0.1	0.25364	0.25361	0.252875	0.263624	0.252942	0.264126
	0.15	0.15660	0.15657	0.155447	0.161000	0.156059	0.165683
	0.20	0.09644	0.09642	0.094289	0.098311	0.095889	0.101617
	0.25	0.05922	0.05920	0.055674	0.060052	0.056174	0.059113
0.5	0.1	0.37158	0.37158	0.373085	0.372799	0.376474	0.393354
	0.15	0.22682	0.22678	0.228940	0.227657	0.235875	0.251788
	0.20	0.13847	0.13844	0.142127	0.138972	0.153645	0.163931
	0.25	0.08454	0.08451	0.091944	0.084942	0.112810	0.120967
0.75	0.1	0.27258	0.27258	0.272368	0.263624	0.271517	0.285579
	0.15	0.16437	0.16433	0.163628	0.160973	0.162739	0.176957
	0.20	0.09944	0.09941	0.098656	0.098305	0.098431	0.111020
	0.25	0.06035	0.06033	0.059343	0.060051	0.057394	0.068569



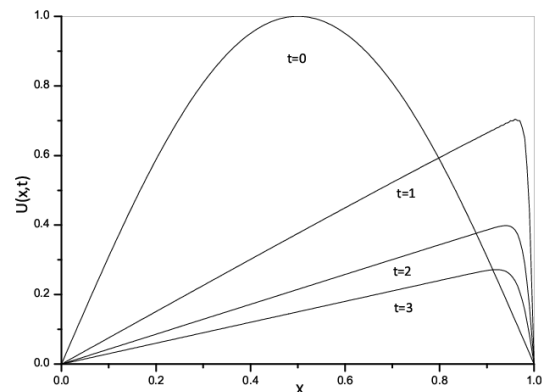
(a)



(b)

Figure 1. The behaviour of numerical solutions of Problem 1 at various times a) $h = 0.025, \Delta t = 0.01, v = 1$. b) $h = 0.025, \Delta t = 0.01, v = 0.1$.

(a)



(b)

Figure 2. The behaviour of numerical solutions of Problem 1 at various times a) $h = 0.025, \Delta t = 0.01$ for $v = 0.01$. b) $h = 0.005, \Delta t = 0.01, v = 0.005$.

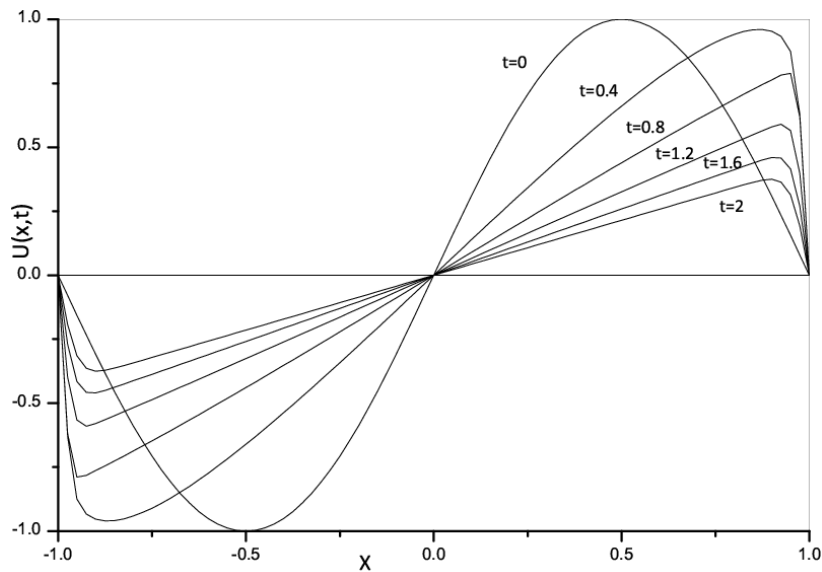


Figure 3. The physical behaviour of numerical solutions of Problem 1 for $h = 0.025$, $\Delta t = 0.01$, $v = 0.01$, $a = -1$, $b = 1$ at times $t \leq 2$.

In Table 1, a comparison of exact and approximate solutions of Problem 1 for $t = 0.001$, $v = 1$ at various mesh sizes and time $t = 0.1$ has been presented. In addition to this, the error norms L_2 and L_∞ have been computed and the error norm $\|e\|_1$ has been compared with some of the studies. As it can be seen from Table 1, as the value of h decreases, the numerical solutions approach to exact solutions. Additionally, when we look at the error norm $\|e\|_1$, although the results in Ref. [13, 20, 23, 33] are computed for $\Delta t = 10^{-5}$ our results computed using $\Delta t = 10^{-3}$ are in good agreement with their results. In Table 2, the values $\Delta t = 0.001$, $h = 0.0125$, $v = 1, 0.1, 0.01$ have been used and a comparison of numerical and analytical results has been presented at $x = 0.25, 0.5, 0.75$. As it can be seen from this table, our numerical results are very approximate to the exact ones. In Tables 3 and 4, for values of $v = 1, h = 0.0125$ a comparison of numerical and analytical results has been given with those in Ref. [6, 7, 13, 20, 23] and Ref. [19, 41], respectively. As it can be seen from those tables, although we have used larger time step values Δt , our results are very satisfactory. Since the Fourier series solution of Problem 1 converges very slowly for small values of v , a comparison of exact and approximate solutions has not been calculated. But in Figures 1 and 2, the graphics of numerical solutions corresponding to the values of $v = 1, 0.1, 0.01$ and 0.005 have been presented and it is seen from these graphics that as the value of v decreases, the numerical solution goes to the right in a sharpening manner. Moreover, the numerical solutions of Problem 1 for values of $v = 0.01, h = 0.025, a = -1$ and $b = 1$ are computed. It is seen from Figure 3 that the numerical solutions of Problem one exhibit the right physical behaviours for different values of time variable t .

Problem 2: As a second test problem, the Burgers equation with the following initial and boundary conditions as given in Refs. [7, 24, 35] is taken into consideration

$$\begin{aligned} u(x, 0) &= 4x(1 - x), & 0 \leq x \leq 1, \\ u(0, t) &= u(1, t) = 0, & t \geq 0. \end{aligned}$$

The exact solution for this problem is obtained by taking the coefficients in (5.1) as follows

$$\begin{aligned} a_0 &= \int_0^1 e^{-x^2(3v)^{-1}(3-2x)} dx, \\ a_n &= 2 \int_0^1 e^{-x^2(3v)^{-1}(3-2x)} \cos(n\pi x) dx, \quad n = 1, 2, \dots \end{aligned}$$

Table 5. A comparison of the solutions of Problem 2 for $\Delta t = 0.001, h = 0.0125, v = 1, 0.1$ and 0.01 at various values of time t .

x	t	$v = 1$		$v = 0.1$		$v = 0.01$	
		Numerical	Exact	Numerical	Exact	Numerical	Exact
0.25	0.4	0.01400	0.01400	0.31752	0.31752	0.36226	0.36226
	0.6	0.00195	0.00195	0.24615	0.24614	0.28203	0.28204
	0.8	0.00027	0.00027	0.19957	0.19956	0.23045	0.23045
	1.0	0.00004	0.00004	0.16561	0.16560	0.19469	0.19469
	3.0	0.00000	0.00000	0.02775	0.02776	0.07613	0.07613
0.5	0.4	0.01984	0.01985	0.58455	0.58454	0.68368	0.68368
	0.6	0.00276	0.00276	0.45800	0.45798	0.54832	0.54832
	0.8	0.00038	0.00038	0.36742	0.36740	0.45371	0.45371
	1.0	0.00005	0.00005	0.29835	0.29834	0.38567	0.38568
	3.0	0.00000	0.00000	0.04105	0.04106	0.15218	0.15218
0.75	0.4	0.01406	0.01407	0.64556	0.64562	0.92051	0.92050
	0.6	0.00195	0.00195	0.50261	0.50268	0.78300	0.78299
	0.8	0.00027	0.00027	0.38527	0.38534	0.66272	0.66272
	1.0	0.00004	0.00004	0.29580	0.29586	0.56932	0.56932
	3.0	0.00000	0.00000	0.03043	0.03044	0.22776	0.22774

Table 6. A comparison of numerical solutions of Problem 2 for $v = 0.01$ at various time levels.

x	t	Xu at al. [42]	Mittal & Jain [35]	Kutluay & Esen [24]	Present	Exact
		$\Delta t = 0.0001$	$\Delta t = 0.001$	$\Delta t = 0.0001$	$\Delta t = 0.01$	-
		$h = 0.0125$	$h = 0.025$	$h = 0.0125$	$h = 0.0125$	-
0.25	0.4	0.3622	0.36225	0.36218	0.36225	0.36226
	0.6	0.2820	0.28202	0.28197	0.28202	0.28204
	0.8	0.2304	0.23044	0.23040	0.23044	0.23045
	1.0	0.1947	0.19468	0.19465	0.19468	0.19469
	3.0	0.0761	0.07613	0.07613	0.07613	0.07613
0.5	0.4	0.6836	0.68368	0.68364	0.68370	0.68368
	0.6	0.5483	0.54832	0.54829	0.54832	0.54832
	0.8	0.4537	0.45371	0.45368	0.45371	0.45371
	1.0	0.3856	0.38567	0.38564	0.38567	0.38568
	3.0	0.1522	0.15218	0.15217	0.15218	0.15218
0.75	0.4	0.9205	0.92052	0.92047	0.92059	0.92050
	0.6	0.7830	0.78300	0.78297	0.78304	0.78299
	0.8	0.6627	0.66272	0.66270	0.66274	0.66272
	1.0	0.5693	0.56932	0.56930	0.56932	0.56932
	3.0	0.2277	0.22782	0.22773	0.22776	0.22774

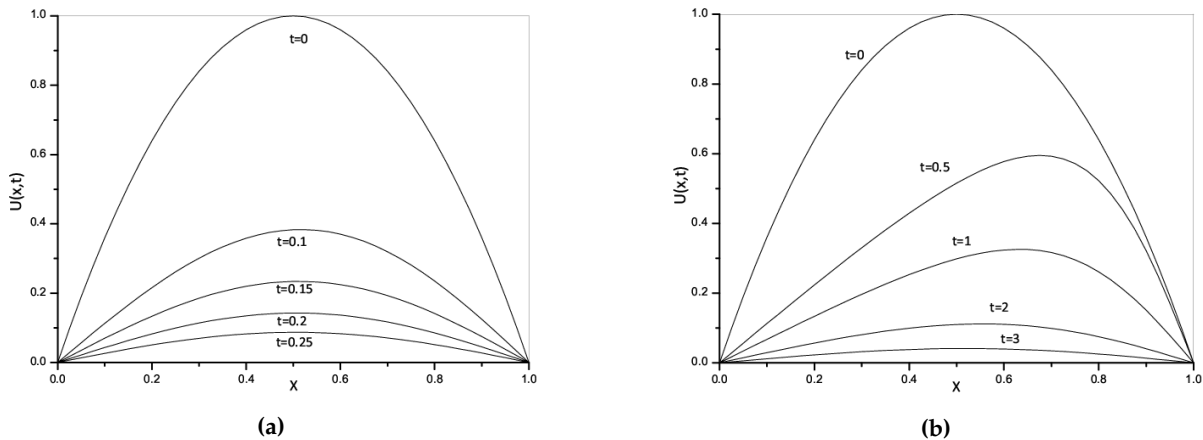


Figure 4. The behaviour of numerical solutions of Problem 2 at various times a) $h = 0.025, \Delta t = 0.01, v = 1$. b) $h = 0.025, \Delta t = 0.01, v = 0.1$.

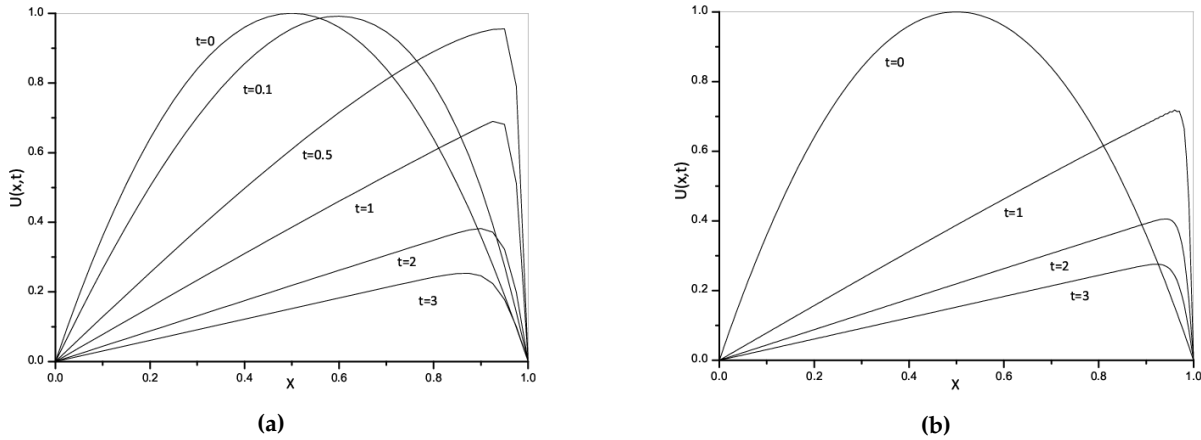


Figure 5. The behaviour of numerical solutions of Problem 2 at various times a) $h = 0.025, \Delta t = 0.001, v = 0.01$. b) $h = 0.005, \Delta t = 0.01, v = 0.005$.

In Table 5, a comparison of numerical and exact solutions of Problem 2 for $\Delta t = 0.001, h = 0.0125, v = 1, 0.1, 0.01$ at various time and space sizes has been presented. As it is seen from this table, the newly obtained numerical results are in good agreement with exact ones. In Table 6, the results in Ref. [24, 35, 42] are compared with ours for Problem 2 for $v = 0.01$. Our results are better than those found in Ref. [24, 42], although their time step value Δt is smaller than ours. Mittal and Jain [35] have calculated their results for smaller values of Δt but bigger values of h . From Table 6, it is seen that our results are in good agreement with theirs. The physical behaviors of the numerical solutions are illustrated in Figures 4 and 5.

Problem 3: As a final problem, the Burgers equation is taken with the following initial condition given at time $t = 1$ as given in Ref. [24, 35]

$$u(x, t) = \frac{x}{1 + \exp\left[\frac{1}{4v}\left(x^2 - \frac{1}{4}\right)\right]},$$

and the boundary conditions $u(a, t) = u(b, t) = 0$. The exact solution of this problem is

$$u(x, t) = \frac{x/t}{1 + \sqrt{t/t_0} \exp\left[\frac{1}{4v}\left(x^2 - \frac{1}{4}\right)\right]}, \quad t \geq 1$$

where $t_0 = \exp\left(\frac{1}{8v}\right)$.

Table 7. A comparison of numerical results and error norms of Problem 3 for values of $\Delta t = 0.01, h = 0.005, v = 0.005, a = 0, b = 1$

x	Present $t = 1.7$	Exact $t = 1.7$	Present $t = 2.5$	Exact $t = 2.5$	Present $t = 3.25$	Exact $t = 3.25$
0.1	0.05882	0.05882	0.04000	0.04000	0.03077	0.03077
0.2	0.11764	0.11765	0.08000	0.08000	0.06154	0.06154
0.3	0.17646	0.17646	0.12000	0.12000	0.09230	0.09231
0.4	0.23517	0.23517	0.15998	0.15998	0.12307	0.12307
0.5	0.29193	0.29190	0.19983	0.19983	0.15380	0.15380
0.6	0.29585	0.29591	0.23814	0.23812	0.18430	0.18430
0.7	0.04197	0.04193	0.25312	0.25310	0.21271	0.21270
0.8	0.00064	0.00065	0.10212	0.10210	0.21844	0.21844
0.9	0.00000	0.00000	0.00553	0.00554	0.10093	0.10126
	$t = 1.7$	$t = 2.4$	$t = 2.5$	$t = 3.1$	$t = 3.25$	
$L_2 \times 10^3$ (Present)	0.03565	0.02167	0.02436	0.55501	1.10263	
$L_\infty \times 10^3$ (Present)	0.16110	0.08346	0.11560	4.10425	7.99529	
$L_2 \times 10^3$ Ref. [14](QBCM)	0.07215	—	0.05103	—	1.24901	
$L_\infty \times 10^3$ Ref. [14](QBCM)	0.31153	—	0.18902	—	8.98390	
$L_2 \times 10^3$ Ref. [5]	0.857	0.423	—	0.235	—	
$L_\infty \times 10^3$ Ref. [5]	2.576	1.242	—	0.688	—	
$L_2 \times 10^3$ Ref. [38]	0.02681	—	0.03135	—	1.11149	
$L_\infty \times 10^3$ Ref. [38]	0.09174	—	0.11515	—	8.00069	

Table 8. A comparison of numerical results and error norms of Problem 3 for values of $\Delta t = 0.01$, $h = 0.005$, $v = 0.005$, $a = 0$, $b = 1.2$.

x	t	Shu et al. [38] $h = 10^{-4}, \beta = 1$ $\Delta t = 0.01$	Shu et al. [38] $h = 10^{-4}, \beta = 0.5$ $\Delta t = 0.01$	Mittal & Jain [35] $h = 0.005$ $\Delta t = 0.001$	Present $h = 0.005$ $\Delta t = 0.001$	Exact
0.2	1.7	0.1176565	0.1174841	0.1176452	0.1176452	0.1176452
	2.5	0.0800527	0.0798389	0.0799990	0.0799990	0.0799989
	3.0	0.0667147	0.0665176	0.0666658	0.0666658	0.0666658
	3.5	0.0571820	0.0570060	0.0571422	0.0571422	0.0571422
0.4	1.7	0.2332111	0.2348504	0.2351690	0.2351690	0.2351677
	2.5	0.1591735	0.1596608	0.1599771	0.1599771	0.1599769
	3.0	0.1328314	0.1330273	0.1333211	0.1333211	0.1333209
	3.5	0.1139606	0.1140077	0.1142780	0.1142780	0.1142779
0.6	1.7	0.2940048	0.2961269	0.2958570	0.2958570	0.2959097
	2.5	0.2347876	0.2376699	0.2381299	0.2381300	0.2381207
	3.0	0.1973222	0.1990478	0.1994839	0.1994839	0.1994805
	3.5	0.1697753	0.1708231	0.1712257	0.1712257	0.1712242
0.8	1.7	0.0008917	0.0006640	0.0006381	0.0006381	0.0006465
	2.5	0.1103866	0.1036067	0.1021325	0.1021324	0.1020957
	3.0	0.2088346	0.2093735	0.2088032	0.2088032	0.2088359
	3.5	0.2119293	0.2143409	0.2145938	0.2145939	0.2145869

Table 9. A comparison of error norms of Problem 3 for values of $v = 0.005$, $a = 0$, $b = 1.2$.

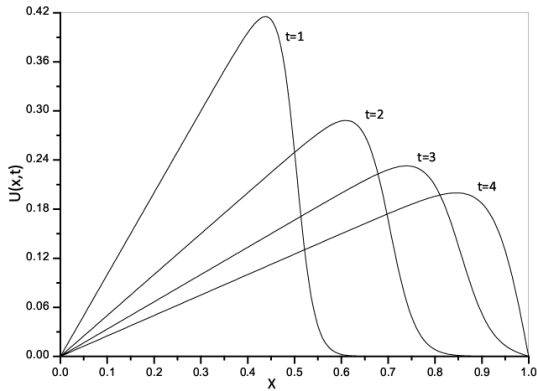
t		[38] ($\beta = 1$) $h = 10^{-4}$ $\Delta t = 0.01$	[38] ($\beta = 0.5$) $h = 10^{-4}$ $\Delta t = 0.01$	[35] $h = 0.005$ $\Delta t = 0.001$	Present $h = 0.005$ $\Delta t = 0.01$	Present $h = 0.005$ $\Delta t = 0.001$
$t = 1.7$	$L_2 \times 10^3$	3.08966	0.384209	0.0252	0.03565	0.02531
	$L_\infty \times 10^3$	10.40404	1.347279	0.0994	0.16110	0.10003
$t = 2.5$	$L_2 \times 10^3$	2.72048	0.491345	0.0151	0.01927	0.01508
	$L_\infty \times 10^3$	8.29747	1.554700	0.0549	0.07719	0.05517
$t = 3.0$	$L_2 \times 10^3$	2.39922	0.515077	0.0118	0.01451	0.01185
	$L_\infty \times 10^3$	6.98801	1.552891	0.0414	0.05464	0.04150
$t = 3.5$	$L_2 \times 10^3$	2.12110	0.525855	0.0117	0.01328	0.01175
	$L_\infty \times 10^3$	5.94321	1.521961	0.0486	0.04858	0.04858

Table 10. A comparison of error norms of Problem 3 for values of $v = 0.0015$, $a = 0$, $b = 1$ with those in Ref. [37].

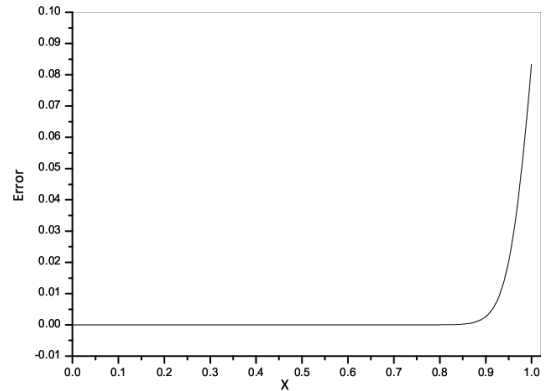
t	Septic B-spline[37] $h = 0.005, \Delta t = 0.01$		Present $h = 0.005, \Delta t = 0.01$	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1.2	0.3853879498	3.2368476652	0.3718509991	3.1646275222
1.4	0.4644348073	3.3488926879	0.2863329944	2.4415417573
1.6	0.4964749406	3.1923238319	0.2288596826	1.8783862136
1.8	0.5092790346	2.9867895721	0.1887021734	1.4497813951
2.0	0.5125147492	2.7831501328	0.1593751139	1.1782635194
2.2	0.5105961277	2.5951212085	0.1371827934	1.0357077227
2.4	0.5058011505	2.4255446546	0.1199015172	0.8279599830
2.8	0.4921096936	2.1380460738	0.0949225040	0.6403753054
3.0	0.4843911639	2.0164737965	0.0856621454	0.5858051083
3.4	0.4687308313	1.8086943377	0.0717301538	0.4764481427
3.7	0.6446318395	4.7081914051	0.2791695008	3.0371207632

Table 11. A comparison of error norms of Problem 3 for values of $v = 0.005$, $a = 0$, $b = 1$ with those in Ref. [37].

t	Septic B-spline[37]		Present	
	$h = 0.02, \Delta t = 0.01$		$h = 0.02, \Delta t = 0.01$	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1.2	0.5814317166	3.1354062730	0.5537663018	2.1699139717
1.4	0.6467617678	2.9313488434	0.4921119039	1.8710420482
1.6	0.6760951564	2.6922645525	0.4184750799	1.6238728732
1.8	0.6876183734	2.4718915438	0.3592643498	1.4099229281
2.0	0.6895350821	2.3766574001	0.3129510874	1.1485717697
2.2	0.6860166616	2.2731921896	0.2762460981	1.0081759068
2.4	0.6794328066	2.1678477506	0.2467279685	0.8919346792
2.8	0.6830706255	1.9689522786	0.2228476846	0.6805470080
3.0	0.8295132819	2.9572472407	0.3106757860	1.5961621658
3.2	1.4855981917	7.4914648713	0.6551177234	4.0139480396

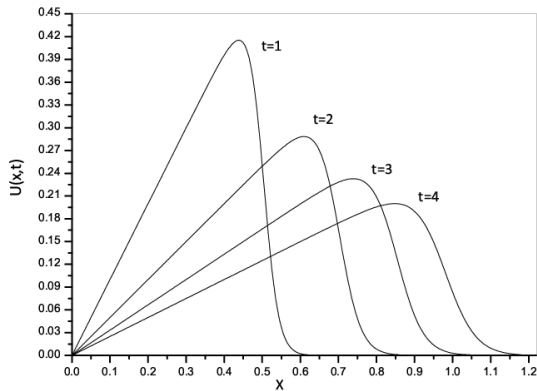


(a)

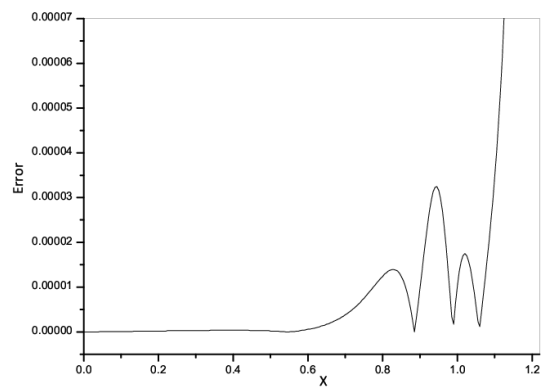


(b)

Figure 6. The physical behaviour of numerical solution of Problem 3 a) $h = 0.005$, $\Delta t = 0.01$, $v = 0.005$. b) and the error for the same values at time $t = 4$.



(a)



(b)

Figure 7. The physical behaviour of numerical solution of Problem 3 a) $h = 0.005$, $\Delta t = 0.01$, $v = 0.005$, $a = 0$, $b = 1.2$. b) and the error for the same values at time $t = 4$.

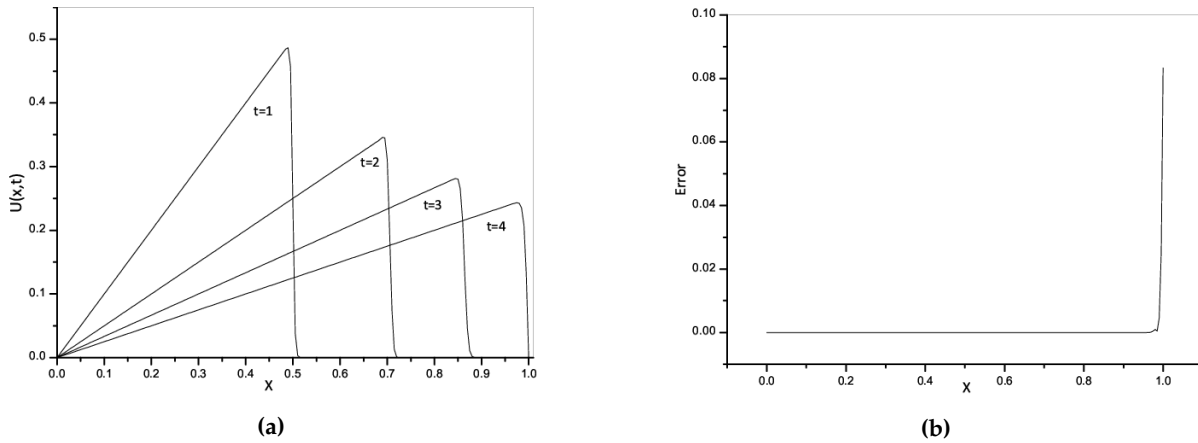


Figure 8. The physical behaviour of numerical solution of Problem 3 a) $h = 0.005$, $\Delta t = 0.01$, $v = 0.0005$. b) and the error for the same values at time $t = 4$.

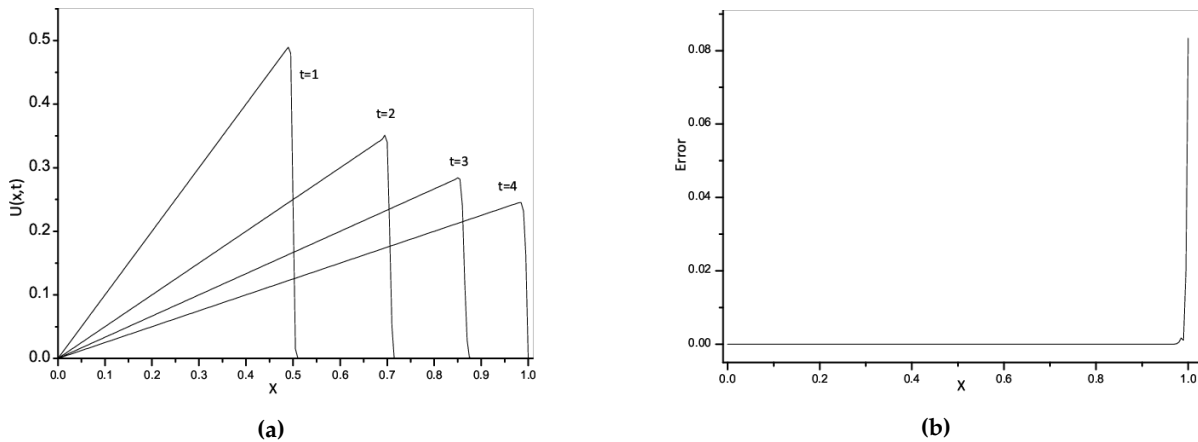


Figure 9. The physical behaviour of numerical solution of Problem 3 a) $h = 0.005$, $\Delta t = 0.01$, $v = 0.00036$. b) and the error for the same values at time $t = 4$.

The solution of Problem 3 corresponds to propagation of the shocks, which becomes smoother as time progresses. In Table 7, the pointwise values of the problem for values of $\Delta t = 0.01$, $h = 0.005$, $v = 0.005$, $a = 0$, $b = 1$ at various times have been given and a comparison of the error norms L_2 and L_∞ with those in Ref. [5, 14, 38] has been presented. As it is seen from this table, the numerical solutions obtained using the present method are in good agreement with exact ones. Moreover the error norms L_2 and L_∞ in our study are better than those in Ref. [5, 14] and in good agreement with those in Ref. [38]. In Table 8, for values of $\Delta t = 0.01$, $h = 0.005$, $v = 0.005$, $a = 0$, $b = 1.2$ the results are compared with those of Shu et al. [38] and Mittal & Jain [35] for various space and time values. As it is seen from the table, our results are much better than those of Ref. [38] and are in good agreement with those in Ref. [35]. In Table 9, a comparison of the error norms L_2 and L_∞ with those given in Ref. [38] and Ref. [35] has been presented. As it is seen from the table, even for bigger values of Δt our error norms are much better than those given in Ref. [38] and are nearly the same those given in. In Tables 10 and 11, a comparison of the error norms L_2 and L_∞ with those given in Ref. [37] for various values of time t . As it is seen from these two tables, our error norms are very low when compared with those found in Ref. [37]. Moreover, the three figures plotted in Figures 6-9 shows the behaviour of the propagation of the shocks very well. As it is seen from these graphics, as the value of viscosity v decreases, the solution curves become steeper and steeper.

6. Conclusions

In this work, numerical solutions of nonlinear Burgers equation given by Dirichlet boundary conditions and cubic B-spline basis functions are calculated by collocation method. The main problem is transformed into two sub-problems according to time, and each sub-problem is discretized by position with the help of cubic B-spline

bases and two ordinary differential equation systems are obtained. The stability analyzes of these two systems were examined by the Von-Neumann method and found to be unconditionally stable. These two systems were then solved using the Strang time splitting technique. This method has been applied to three test problems, and the results obtained are compared to those in the literature. As a result, we can see that the method we use is an easy and effective way to apply nonlinear equations.

References

- [1] Bateman, H.: *Some recent researches on the motion of fluids*. Monthly Weather Rev. 43, 163-170 (1915).
- [2] Jawad, A.J.M., Kumar, S., Biswas, A.: *Soliton solutions of a few nonlinear wave equations in engineering sciences*. Sharif University of Technology Scientia Iranica, Transactions D 21(3), 861-869 (2014).
- [3] Neirameh, A., Eslami, M.: *An analytical method for finding exact solitary wave solutions of the coupled (2+1)-dimensional Painleve Burgers equation*. Scientia Iranica B 24(2), 715-726 (2017) .
- [4] Haq, S., Hussain, A., Uddin, M.: *On the numerical solution of nonlinear Burgers'-type equations using meshless method of lines*. Applied Mathematics and Computation 218, 6280-6290 (2012) . doi:10.1016/j.amc.2011.11.106
- [5] Ali, A.H.A., Gardner, G.A., Gardner, L.R.T.: *A collocation solution for Burgers' equation using cubic B-spline finite elements*. Comput. Methods Appl. Mech.Eng. 100, 325-337 (1992).
- [6] Arora, G., Singh, B. K.: *Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method*. Appl. Math. Comput. 224, 166-177 (2013). <http://dx.doi.org/10.1016/j.amc.2013.08.071>
- [7] Asaithambi, A.: *Numerical solution of the Burgers' equation by automatic differentiation*. Appl. Math. Comput. 216, 2700-2708 (2010). doi:10.1016/j.amc.2010.03.115
- [8] Benton, E., Platzman, G.W.: *A table of solutions of the one-dimensional Burges equations*. Quart. Appl. Math. 30, 195-212 (1972).
- [9] Bratsos, A.G.: *A fourth-order numerical scheme for solving the modified Burgers equation*. Computers and Mathematics with Applications 60, 1393-1400 (2010). doi:10.1016/j.camwa.2010.06.021
- [10] Burgers, J.M.: *A mathematical model illustrating the theory of turbulence*. Adv. Appl. Mech. 1, 171-199 (1948).
- [11] Burgers, J.M.: *Mathematical examples illustrating relations occurring in the theory of turbulent fluid motion*. Trans. R. Neth. Acad. Sci. Amst. 17, 1-53 (1939) .
- [12] Cole, J.D.: *On a quasilinear parabolic equation occurring in aerodynamics*. Quart. Appl. Math. 9, 225-236 (1951) .
- [13] Dağ, İ., Irk, D., Saka, B.: *A numerical solution of the Burgers' equation using cubic B-splines*. Applied Mathematics and Computation 163, 199-211 (2005). doi:10.1016/j.amc.2004.01.028
- [14] Dağ, İ., Irk, D., Şahin, A.: *B-spline collocation methods for numerical solitions of the Burgers' equation*. Mathematical Problems in Engineering 2005:5, 521-538 (2005). DOI: 10.1155/MPE.2005.521
- [15] Korkmaz, A., Dağ, İ.: *Cubic B-spline differential quadrature methods and stability for Burgers' equation*. International Journal for Computer-Aided Engineering and Software 30, 320-344 (2013). DOI: 10.1108/02644401311314312
- [16] Brezis, H., Browder, F.: *Partial Differential Equations in the 20th Century*. 135, 76-144 (1998).
- [17] Ganaie, I.A., Kukreja, V.K.: *Numerical solution of Burgers' equation by cubic Hermite collocation method*. Applied Mathematics and Computation 237, 571-581 (2014). <http://dx.doi.org/10.1016/j.amc.2014.03.102>.
- [18] Geiser, J.: *Iterative Splitting Methods for Differential Equations*. CRC Press. London (2011).
- [19] Gulsu, M.: *A finite difference approach for solution of Burgers equation*. Applied Mathematics and Computation 175, 1245-1255 (2006). doi:10.1016/j.amc.2005.08.042
- [20] Zhu, C.G., Wang, R.H.: *Numerical solution of Burgers' equation by cubic B-spline quasi-interpolation*. Appl. Math. Comput. 208, 260-272 (2009). DOI: 10.1016/j.amc.2008.11.045.

- [21] Jiware, R.: *A Haar wavelet quasilinearization approach for numerical simulation of Burgers' equation*. Computer Physics Communications 183, 2413-2423 (2012) . doi:10.1016/j.cpc.2012.06.009.
- [22] Raslan, K.R.: *A Collocation solution for burgers equation using quadratic B-spline finite elements*. Intern. J. Computer Math. 80 931-938 (2003). DOI: 10.1080=0020716031000079554
- [23] Kutluay, S., Esen, A., Dağ, İ.: *Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method*. Journal of Computational and Applied Mathematics 167, 21-33 (2004). doi:10.1016/j.cam.2003.09.043
- [24] Kutluay, S., Esen, A.: *A Lumped galerkin method for solving the Burgers equation*. International Journal of Computer Mathematics 81, 1433-1444 (2004). DOI: 10.1080/00207160412331286833
- [25] Jiware, R.: *A hybrid numerical scheme for the numerical solution of the Burgers' equation*. Computer Physics Communications 188 59-67 (2015). <http://dx.doi.org/10.1016/j.cpc.2014.11.004>
- [26] Seydaoğlu, M., Erdoğan, U., Öziş, T.: *Numerical solution of Burgers' equation with high order splitting methods*. Journal of Computational and Applied Mathematics 291, 410-421 (2016). <http://dx.doi.org/10.1016/j.cam.2015.04.021>
- [27] Sarboland, M., Aminataei, A.: *On the Numerical Solution of One-Dimensional Nonlinear Nonhomogeneous Burgers' Equation*. Journal of Applied Mathematics. 2014, Article ID 598432, 15 pages(2014), <http://dx.doi.org/10.1155/2014/598432>
- [28] Ganaie, I.A., Kukreja, V.K.: *Numerical solution of Burgers' equation by cubic Hermite collocation method*. Applied Mathematics and Computation 237, 571-581 (2014). <http://dx.doi.org/10.1016/j.amc.2014.03.102>
- [29] Ashpazzadeh, E., Han, B., Lakestani, M.: *Biorthogonal multiwavelets on the interval for numerical solutions of Burgers' equation*. Journal of Computational and Applied Mathematics 317, 510-534 (2017). <http://dx.doi.org/10.1016/j.cam.2016.11.045>
- [30] Mukundan, V., Awasthi, A.: *Efficient numerical techniques for Burgers' equation*. Applied Mathematics and Computation 262, 282-297 (2015). <http://dx.doi.org/10.1016/j.amc.2015.03.122>
- [31] Shesha, S.R., Nargund, A.L., Bujurke, N.M.: *Numerical solution of non-planar Burgers equation by Haar wavelet method*. Journal of Mathematical Modeling 5(2), 89-118 2017.
- [32] Tamsir, M., Srivastava, V.K., Jiware, R.: *An algorithm based on exponential modified cubic B-spline differential quadrature method for nonlinear Burgers' equation*. Applied Mathematics and Computation 290, 111-124 (2016). <http://dx.doi.org/10.1016/j.amc.2016.05.048>
- [33] Kutluay, S., Bahadir, A.R., Özdeş, A.: *Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods*. Journal of Computational and Applied Mathematics 103, 251-261 (1999).
- [34] Gao, Y., Le, L., Shi, B.: *Numerical solution of Burgers' equation by lattice Boltzmann method*. Applied Mathematics and Computation 219, 7685-7692 (2013). <http://dx.doi.org/10.1016/j.amc.2013.01.056>
- [35] Mittal, R.C., Jain, R.K.: *Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method*. Applied Mathematics and Computation 218, 7839-7855 (2012). doi:10.1016/j.amc.2012.01.059
- [36] Prenter, P.M.: Splines and Variational Methods. Wiley Publications. New York (1975).
- [37] Ramadan, M. A., El-Danaf, T. S., Abd Alaal, F.E.I.: *A numerical solution of the Burgers equation using septic B-splines*. Chaos, Solitons and Fractals 26, 795-804 (2005). doi:10.1016/j.chaos.2005.01.054
- [38] Xiea, S.S., Heob, S., Kimc, S., Wooc, G., Yi, S.: *Numerical solution of one-dimensional Burgers' equation using reproducing kernel function*. Journal of Computational and Applied Mathematics 214, 417-434 (2008). doi:10.1016/j.cam.2007.03.010
- [39] Sportisse, B.: *An analysis of operator splitting techniques in the stiff case*. Journal of Computational Physics 161, 140-168 (2000). doi:10.1006/jcph.2000.6495

- [40] Strang, G.: *On the construction and comparison of difference schemes*. SIAM J. Numer. Anal. 5, 506-517 (1968).
- [41] Gulsu, M., Öziş, T.: *Numerical solution of Burgers equation with restrictive Taylor approximation*. Applied Mathematics and Computation 171, 1192-1200 (2005). doi:10.1016/j.amc.2005.01.106
- [42] Xu, M., Wang, R.H., Zhang, J.H., Fang, Q.: *A novel numerical scheme for solving Burgers' equation*. Applied Mathematics and Computation 217, 4473-4482 (2011). doi:10.1016/j.amc.2010.10.050
- [43] VonNeumann, J., Richtmyer, R. D.: *A Method for the Numerical Calculation of Hydrodynamic Shocks*. J. Appl. Phys. 21, 232-237 (1950).

Affiliations

YUSUF UÇAR

ADDRESS: Inonu University, Dept. of Mathematics, 44280, Malatya-TURKEY.

E-MAIL: yusuf.ucar@inonu.edu.tr

ORCID ID: 0000-0003-1469-5002

NURI MURAT YAĞMURLU

ADDRESS: Inonu University, Dept. of Mathematics, 44280, Malatya-TURKEY.

E-MAIL: murat.yagmurlu@inonu.edu.tr

ORCID ID: 0000-0003-1593-0254

İHSAN ÇELİKKAYA

ADDRESS: Batman University, Dept. of Mathematics, 72070, Batman-TURKEY.

E-MAIL: ihsancelikkaya85@gmail.com

ORCID ID: 0000-0002-8684-5922