Journal of Applied Analysis and Computation
Volume 13, Number 1, February 2023, 313–328
Website:http://www.jaac-online.com
DOI:10.11948/20220095

FOUR NUMERICAL SCHEMES FOR SOLUTION OF BURGERS' EQUATION VIA OPERATOR SPLITTING TRIGONOMETRIC CUBIC B-SPLINE COLLOCATION METHOD

İhsan Çelikkaya[1,†] and Ahmet Güzel[1]

Abstract In this study, we have used operator splitting methods for numerical solutions of the Burgers' equation by given four different numerical schemes. To set these schemes, we divide the Burgers equation into two sub-problems according to the time term, as linear $U_t = \mathcal{L}(U)$ and nonlinear $U_t = \mathcal{N}(U)$. Then, numerical schemes have been obtained by the finite element method using trigonometric cubic B-spline basis for each sub-problem. Splitting $\mathcal{L} \circ \mathcal{N}$, $\mathcal{N} \circ \mathcal{L}$ Lie-Trotter and $\mathcal{L} \circ \mathcal{N} \circ \mathcal{L}$, $\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$ Strang splitting solution schemes have been used to obtain the solution of the main equation. Numerical results calculated with these schemes have been compared among themselves in terms of L_2, L_∞ error norms and CPU time. Furthermore, the numerical results have been compared with some studies that solved the equation directly with the same method. It has been observed that the numerical results obtained with the proposed schemes are in agreement with the exact solution and other studies in the literature. All calculations are obtained using Matlab Version R2015a.

Keywords Burgers equation, strang splitting, lie-trotter splitting, collocation method, trigonometric cubic b-spline basis.

MSC(2010) 65M70, 35G31, 65Z05, 65D05, 65D07, 65L20.

1. Introduction

The form of the nonlinear Burgers' equation is

$$U_t + UU_x = \nu U_{xx},\ a < x < b,\ t \geq t_0 \tag{1.1}$$

where $U = U(x,t)$ is a differentiable smooth function of space variable x and time variable t and $\nu > 0$ kinematics viscosity coefficient. The Burgers' equation, which derives two basic steady solutions, was first introduced by Bateman [2] 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' [3,4] and is referred to as Burgers' equation after this study. This equation involves nonlinearity and dissipation in the simplest possible way and can be thought of as a nonlinear type of heat equation [5]. If $\nu = 0$ is taken

†The corresponding author. Email: ihsan.celikkaya@batman.edu.tr (İ. Çelikkaya)
[1]Department of Mathematics, Batman University, West Raman Campus, 72100, Turkey

in Equation (1.1), the inviscid Burgers' equation expressing the shock waves and having many important applications in physics is obtained [6]. The reasons for the Burgers' equation attracting the attention of many researchers are that it includes the nonlinear advection term UU_x in its simplest form, the dissipation term νU_{xx} that models the physical wave phenomena, the shock wave property for the small values of the kinematic viscosity coefficient ν, the comparison with the exact solution [8] obtained by [5]. The Burgers' equation is an important partial derivative differential equation with widespread usage in mathematical physics. The increase in interest in nonlinear science makes this equation an observation of scientists because it describes very well many gas dynamics, heat conduction, traffic flow, and shock wave models [16]. Numerous mathematical techniques and methods have been developed in recent years to obtain numerical solutions of Burgers' equation and compare analytical and numerical solutions.

Among others, Dağ et al. [9] used the trigonometric cubic B-spline (CTB) functions to set up the collocation method for finding solutions of the Burgers' equation. Ersoy et al. [13] found the solutions of Burgers' equation using exponential cubic B-splines in the collocation method. Ö. Ersoy [14] adapted the trigonometric cubic B-splines to the collocation method for solutions of the Kuramoto-Sivashinsky(KS) equation. Uçar et al. [29] solved Burgers' equation by using the operator splitting cubic B-spline collocation method. B. Saka and İ. Dağ [25] applied time and space splitting techniques to get the approximate solutions of the Burgers' equation via a quintic B-spline collocation procedure. Dağ et al. [10] applied cubic B-splines bases using a linearization technique and collocation finite element method. Kutluay et al. [18] utilized the least-squares quadratic B-spline finite element method with three test problems. R.C. Mittal and R.K. Jain [21] calculated the numerical solutions of the Burgers' equation via collocation-modified cubic B-splines using SSP-RK43 and SSP-RK54. Dağ et al. [11] solved the Burgers' equation using both time and space splitting with the quadratic B-spline collocation method. Ö. Ersoy and G. Yiğit [12] investigated the numerical solutions of advection-diffusion equation via quartic-trigonometric tension (QTT) B-spline. Ay et al. [1] found the numerical solutions of Burgers' equation using the subdomain Galerkin method based on the trigonometric B-splines as approximate functions. Ö. Ersoy [15] solved the generalized Kuramoto–Sivashinsky (gKS) equation with the quartic trigonometric tension (QTT) B-spline Galerkin method. S. Kutluay and A. Esen [19] solved the Burgers' equation using a lumped Galerkin method with quadratic B-spline finite elements. B. Saka and İ. Dağ [24] used the quartic B-splines in the collocation method for the numerical solution of the time split Burgers' equation. A. T. Onarcan and Ö. Ersoy [23] obtained numerical solutions of the Coupled Burgers' equation using trigonometric B-spline functions with the collocation method.

This study will obtain the numerical solution of the Burgers' equation by combining the operator splitting, which is easy and practical to apply with the finite element collocation method. Some advantages of operator splitting methods are easy to apply and explicit methods, their algorithms are sequential, and mid-stage solutions are stored in the solution vector, they retain the structural features of the solution (volume preservation, time symmetricity, and simple implementation can be given) [7]. For this purpose, we will split the equation (1.1) by time and convert it into two subproblems with a more straightforward structure, one containing the convection term (UU_x) and the other the diffusion term (U_{xx}). Next, ordinary differential equation systems are obtained using trigonometric cubic B-

spline bases for each of the sub-problem. Then, with the help of finite difference approximations, sub-problems were discretized and linear equation systems in the form $A\delta^{n+1} = B\delta^n$ were obtained. Numerical results were calculated using these systems in $\mathcal{L}\circ\mathcal{N}$, $\mathcal{N}\circ\mathcal{L}$, $\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$ and $\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$ splitting schemes.

2. Operator Splitting Methods

Generally, in splitting methods, the original equation is split into several equations, and each of the sub equation is solved independently over the time interval $[t_n, t_{n+1}]$. These methods, in general, are called time splitting or fractional step methods. In the case of splitting the equation in such a way that it includes different physical phenomena, the method is known as "*operator splitting*" [17].

Let us consider a Cauchy problem given as follows

$$\frac{dU(t)}{dt} = \Pi U(t), \quad U(0) = U^0, \quad t \in [0, T]. \tag{2.1}$$

It is assumed in (2.1) that the function $U(x,t)$ is semi-discretized along spatial direction. We will concentrate on cases where the operator $\Pi = \hat{\mathcal{L}} + \hat{\mathcal{N}}$ can be written as the summation of two linear (and/or nonlinear) operators. That is, it can be written as follows

$$\frac{dU(t)}{dt} = \hat{\mathcal{L}}U(t) + \hat{\mathcal{N}}U(t), \quad U(0) = U^0, \quad t \in [0, T]. \tag{2.2}$$

Where, the vector $U(x,t)$ is the solution vector obtained from the $U^0 \in X$ initial condition, and the operators $\Pi, \hat{\mathcal{L}}, \hat{\mathcal{N}}$ are bounded or unbounded in a finite or infinite X Banach space. With the aid of the Lie operator formulation, the expression (2.2) in general (may be nonlinear) can be written as follows

$$\frac{dU(t)}{dt} = \mathcal{L}U(t) + \mathcal{N}U(t) \tag{2.3}$$

where $\mathcal{L}$ and $\mathcal{N}$ are Lie operators applied to the function $U(t)$ as follows

$$\mathcal{L} = \hat{\mathcal{L}}\left(U(t)\right)\frac{\partial}{\partial U}, \quad \mathcal{N} = \hat{\mathcal{N}}\left(U(t)\right)\frac{\partial}{\partial U}.$$

$U(t_{n+1}) = e^{\Delta t(\mathcal{L}+\mathcal{N})}U(t_n)$ is the formal solution (2.3). This solution can also be written as follows using the Taylor series expansion of the exponential function

$$U(t_{n+1}) = e^{\Delta t(\mathcal{L}+\mathcal{N})}U(t_n) = \sum_{k=0}^{\infty}\frac{t^k}{k!}\left(\hat{\mathcal{L}}\left(U(t)\right)\frac{\partial}{\partial U} + \hat{\mathcal{N}}\left(U(t)\right)\frac{\partial}{\partial U}\right)^k U(t_n).$$

In order to solve Eq. (2.3) numerically, the splitting technique splits the problem into two sub problems as follows

$$\frac{dU(t)}{dt} = \mathcal{L}U(t) \quad \text{and} \quad \frac{dU(t)}{dt} = \mathcal{N}U(t) \tag{2.4}$$

and tries to find the solution numerically or analitically [26]. Let us assume that $\varphi_{\Delta t}^{[\mathcal{L}]}$ and $\varphi_{\Delta t}^{[\mathcal{N}]}$ are the exact or numerical solutions of the equations in (2.4) involving the

operators $\mathcal{L}$ and $\mathcal{N}$, respectively, then the most basic first order splitting technique is defined as follows

$$L_{\Delta t} = \varphi_{\Delta t}^{[\mathcal{L}]} \circ \varphi_{\Delta t}^{[\mathcal{N}]} \equiv e^{\Delta t \mathcal{L}} e^{\Delta t \mathcal{N}} \quad \text{or} \quad L_{\Delta t}^{*} = \varphi_{\Delta t}^{[\mathcal{N}]} \circ \varphi_{\Delta t}^{[\mathcal{L}]} \equiv e^{\Delta t \mathcal{N}} e^{\Delta t \mathcal{L}}$$

and is known as Lie-Trotter splitting [28]. The algorithm for Lie-Trotter splitting $\mathcal{N} \circ \mathcal{L}$ as follows

$$\begin{aligned} &\frac{dU^{*}(t)}{dt} = \mathcal{L}U^{*}(t),\ U^{*}(t_n) = U^{0}(t_n), \qquad t \in [t_n, t_{n+1}], \\ &\frac{dU^{**}(t)}{dt} = \mathcal{N}U^{**}(t),\ U^{**}(t_n) = U^{*}(t_{n+1}),\ t \in [t_n, t_{n+1}], \end{aligned}$$

the final solutions are taken from $U(t_{n+1}) = U^{**}(t_{n+1})$ to complete the splitting step. If we swap the positions of operators $\mathcal{L}$ and $\mathcal{N}$, we get the following half-time step combination

$$\begin{aligned} S_{\Delta t} &= \varphi_{\frac{\Delta t}{2}}^{[\mathcal{L}]} \circ \varphi_{\Delta t}^{[\mathcal{N}]} \circ \varphi_{\frac{\Delta t}{2}}^{[\mathcal{L}]} \equiv e^{\frac{\Delta t}{2}\mathcal{L}} e^{\Delta t \mathcal{N}} e^{\frac{\Delta t}{2}\mathcal{L}} \\ \text{or} \quad S_{\Delta t}^{*} &= \varphi_{\frac{\Delta t}{2}}^{[\mathcal{N}]} \circ \varphi_{\Delta t}^{[\mathcal{L}]} \circ \varphi_{\frac{\Delta t}{2}}^{[\mathcal{N}]} \equiv e^{\frac{\Delta t}{2}\mathcal{N}} e^{\Delta t \mathcal{L}} e^{\frac{\Delta t}{2}\mathcal{N}}. \end{aligned}$$

We obtain so-called symmetric Marchuk [22] or more widely known as Strang splitting [27] having the schemes "$\mathcal{L} \circ \mathcal{N} \circ \mathcal{L}$" and "$\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$". The algorithm for Strang splitting as follows

$$\begin{aligned} &\frac{dU^{*}(t)}{dt} = \mathcal{L}U^{*}(t), \quad U^{*}(t_n) = U^{0}(t_n), && t \in \left[t_n, t_{n+\frac{1}{2}}\right], \\ &\frac{dU^{**}(t)}{dt} = \mathcal{N}U^{**}(t), \quad U^{**}(t_n) = U^{*}(t_{n+\frac{1}{2}}), && t \in [t_n, t_{n+1}], \\ &\frac{dU^{***}(t)}{dt} = \mathcal{L}U^{***}(t), \quad U^{***}(t_{n+\frac{1}{2}}) = U^{**}(t_{n+1}), && t \in \left[t_{n+\frac{1}{2}}, t_{n+1}\right]. \end{aligned}$$

Where $t_{n+\frac{1}{2}} = t_n + \frac{\Delta t}{2}$ and the desired solutions are obtained from equation $U(t_{n+1}) = U^{***}(t_{n+1})$ to complete the splitting time step.

3. Trigonometric Cubic B-spline Functions

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

$$T_m(x) = \frac{1}{r}\begin{cases} p^3(x_{m-2}), & x \in [x_{m-2}, x_{m-1}] \\ p(x_{m-2})(p(x_{m-2})q(x_m) + q(x_{m+1})p(x_{m-1})) + q(x_{m+2})p^2(x_{m-1}), & x \in [x_{m-1}, x_m] \\ p(x_{m-2})q^2(x_{m+1}) + q(x_{m+2})(p(x_{m-1})q(x_{m+1}) + q(x_{m+2})p(x_m)), & x \in [x_m, x_{m+1}] \\ q^3(x_{m+2}), & x \in [x_{m+1}, x_{m+2}] \\ 0, & \text{othwerwise} \end{cases} \tag{3.1}$$

as stated by [20, 30]. Where $p(x_m) = \sin(\frac{x-x_m}{2})$, $q(x_m) = \sin(\frac{x_m-x}{2})$ and $r = \sin(\frac{h}{2})\sin(h)\sin(\frac{3h}{2})$.

It is obvious that the set $\{T_{-1}(x), T_0(x), ..., T_{N+1}(x)\}$ 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 trigonometric cubic B-spline functions and time dependent parameters $\delta_m(t)$ as follows

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

Where 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, U' and U'', we need the values of U, its first and second order derivatives in terms of trigonometric cubic B-spline functions. Using the approximations (3.1) and (3.2), the nodal values of U, U' and U'' are obtained in terms of the time-dependent parameters $\delta_m(t)$ as follows

$$\begin{aligned} U_m &= U(x_m) = \alpha_1\delta_{m-1} + \alpha_2\delta_m + \alpha_1\delta_{m+1}, \\ U'_m &= U'(x_m) = \beta_1\delta_{m-1} + \beta_2\delta_{m+1}, \\ U''_m &= U''(x_m) = \gamma_1\delta_{m-1} + \gamma_2\delta_m + \gamma_1\delta_{m+1}, \end{aligned} \tag{3.3}$$

with

$$\alpha_1 = \frac{\sin^2(h/2)}{\sin(h)\sin(3h/2)},\ \alpha_2 = \frac{2}{1+\cos(2h)},\ \beta_1 = -\frac{3}{4\sin(3h/2)},\ \beta_2 = \frac{3}{4\sin(3h/2)},$$
$$\gamma_1 = \frac{3(1+3\cos(h))}{16\sin^2(h/2)(2\cos(h/2)+\cos(3h/2))},\ \gamma_2 = \frac{3\cos^2(h/2)}{\sin^2(h/2)(2+4\cos(h))}.$$

Where $'$ and $''$ denote the first and second order derivatives with respect to the space variable x, respectively.

4. Application of the method

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

$$U_t - \nu U_{xx} = 0, \tag{4.1}$$

$$U_t + UU_{xx} = 0. \tag{4.2}$$

We obtain the following first-order system of ordinary differential equations by using the values of U, $U^{'}$ and $U^{''}$ given by (3.3) in equations given by (4.1) and (4.2),

$$\alpha_1\overset{\circ}{\delta}_{m-1} + \alpha_2\overset{\circ}{\delta}_m + \alpha_1\overset{\circ}{\delta}_{m+1} - \nu\left(\gamma_1\delta_{m-1} + \gamma_2\delta_m + \gamma_1\delta_{m+1}\right) = 0 \tag{4.3}$$

$$\alpha_1\overset{\circ}{\delta}_{m-1} + \alpha_2\overset{\circ}{\delta}_m + \alpha_1\overset{\circ}{\delta}_{m+1} + z_m\left(\beta_1\delta_{m-1} + \beta_2\delta_{m+1}\right) = 0. \tag{4.4}$$

Where the symbol $\circ$ denotes the first order derivative with respect to the time variable t and

$$z_m = \alpha_1\delta_{m-1} + \alpha_2\delta_m + \alpha_1\delta_{m+1}.$$

While solving the problems, the $\delta^{n+1} = \frac{\delta^{n+1}+\delta^n}{2}$ approach was used to improve the z_m term with inner iteration. If we use the 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 $\overset{\circ}{\delta}_m$,we obtain the following equations, respectively

$$\kappa_1\delta_{m-1}^{n+1} + \kappa_2\delta_m^{n+1} + \kappa_1\delta_{m+1}^{n+1} = \kappa_3\delta_{m-1}^n + \kappa_4\delta_m^n + \kappa_3\delta_{m+1}^n, \tag{4.5}$$

$$\kappa_5\delta_{m-1}^{n+1} + \kappa_6\delta_m^{n+1} + \kappa_7\delta_{m+1}^{n+1} = \kappa_8\delta_{m-1}^n + \kappa_6\delta_m^n + \kappa_9\delta_{m+1}^n, \tag{4.6}$$

$$\kappa_1 = \alpha_1 - \frac{\nu\Delta t\gamma_1}{2}, \kappa_2 = \alpha_2 - \frac{\nu\Delta t\gamma_2}{2}, \kappa_3 = \alpha_1 + \frac{\nu\Delta t\gamma_1}{2}, \kappa_4 = \alpha_2 + \frac{\nu\Delta t\gamma_2}{2},$$

$$\kappa_5 = \alpha_1 + \frac{z_m\Delta t\beta_1}{2}, \kappa_6 = \alpha_2,\ \kappa_7 = \alpha_1 + \frac{z_m\Delta t\beta_2}{2},\ \kappa_8 = \alpha_1 - \frac{z_m\Delta t\beta_1}{2},$$

$$\kappa_9 = \alpha_1 - \frac{z_m\Delta t\beta_2}{2}.$$

The Eqs. (4.5) and (4.6) consist of $(N+1)$ equations and $(N+3)$ unknown time dependent parameters $\delta_m, m = 0, 1, ..., N+1$. In order to obtain a solvable system, we need to eliminate the time dependent parameters δ_{-1} and δ_{N+1}. To accomplish this, we will use the boundary conditions $U(a,t) = f(t)$, $U(b,t) = g(t)$ for the system (4.5) and $U_x(a,t) = h(t)$, $U_x(b,t) = i(t)$ for the system (4.6) to obtain the following equations

$$\delta_{-1} = -\frac{\alpha_2}{\alpha_1}\delta_0 - \delta_1 + \frac{f(t)}{\alpha_1},\ \delta_{N+1} = -\delta_{N-1} - \frac{\alpha_2}{\alpha_1}\delta_N + \frac{g(t)}{\alpha_1}, \tag{4.7}$$

$$\delta_{-1} = -\frac{\beta_2}{\beta_1}\delta_1 + \frac{h(t)}{\beta_1},\ \delta_{N+1} = -\frac{\beta_1}{\beta_2}\delta_{N-1} + \frac{i(t)}{\beta_2}. \tag{4.8}$$

We get a three-diagonal $(N+1)\times(N+1)$ band matrix if we use the equations (4.7) and (4.8) in the system of equations given by (4.5) and (4.6). A unique solution of these systems is easily obtained using the Thomas algoritm. 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. The initial vector is built as follows, using the initial condition $U(x,0) = s(x)$ and the approximation given in (3.2)

$$\begin{aligned}
&U(x_m,0) = s(x_m),\ m = 0(1)N, \\
&U_m = \alpha_1\delta_{m-1}^0 + \alpha_2\delta_m^0 + \alpha_1\delta_{m+1}^0, \\
&U_0 = \alpha_1\delta_{-1}^0 + \alpha_2\delta_0^0 + \alpha_1\delta_1^0, \\
&U_1 = \alpha_1\delta_0^0 + \alpha_2\delta_1^0 + \alpha_1\delta_2^0, \\
&\vdots \\
&U_N = \alpha_1\delta_{N-1}^0 + \alpha_2\delta_N^0 + \alpha_1\delta_{N+1}^0.
\end{aligned} \tag{4.9}$$

To solve this system, the parameters δ_{-1} and δ_{N+1} are eliminated using the boundary conditions $U_{xx}(a,0) = f_1(t)$, $U_{xx}(b,0) = f_2(t)$. As a result, the following three-dimensional $(N+1)\times(N+1)$ band matrix can be solved using the Thomas

algorithm

$$\begin{bmatrix} -\frac{\alpha_1\gamma_2}{\gamma_1}+\alpha_2 & 0 & 0 & & \\ \alpha_1 & \alpha_2 & \alpha_1 & & \\ & \ddots & & & \\ & \alpha_1 & \alpha_2 & \alpha_1 & \\ & 0 & 0 & -\frac{\alpha_1\gamma_2}{\gamma_1}+\alpha_2 & \end{bmatrix} \begin{bmatrix} \delta_0^0 \\ \delta_1^0 \\ \vdots \\ \delta_{N-1}^0 \\ \delta_N^0 \end{bmatrix} = \begin{bmatrix} U_0 - \frac{\alpha_1 f_1(t)}{\gamma_1} \\ U_1 \\ \vdots \\ U_{N-1} \\ U_N - \frac{\alpha_1 f_2(t)}{\gamma_1}. \end{bmatrix}. \tag{4.10}$$

5. Numerical results and discussion

The numerical solutions of the Burgers' equation are going to be sought for two 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 schemes $\mathcal{L}\circ\mathcal{N}$, $\mathcal{N}\circ\mathcal{L}$, $\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$ and $\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$ on Maltab Version R2015a.

$$L_2 = \sqrt{h\sum_{j=0}^{N}\left|\left(U_j^{\text{Exact}} - U_j^{\text{App}}\right)^2\right|},\ L_\infty = \max_j\left|U_j^{\text{Exact}} - U_j^{\text{App}}\right|.$$

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

$$U(x,0) = s(x) = \sin\pi x, \qquad 0 \le x \le 1$$

and boundary conditions

$$f(t) = g(t) = h(t) = i(t) = f_1(t) = f_2(t) = 0.$$

The exact solution of this problem has been obtained as a summation of an infinite series by D. Cole [8] as follows

$$U(x,t) = 2\pi\nu\frac{\sum_j^\infty j a_j \sin(j\pi x)\exp(-j^2\pi^2\nu t)}{a_0 + \sum_{j=1}^\infty a_j\cos(j\pi x)\exp(-j^2\pi^2\nu t)}. \tag{5.1}$$

Where a_0 and a_j are Fourier coefficients given as follows

$$a_0 = \int_0^1 e^{-(2\pi\nu)^{-1}(1-\cos(\pi x))}dx,$$
$$a_n = 2\int_0^1 e^{-(2\pi\nu)^{-1}(1-\cos(\pi x))}\cos(n\pi x)dx, \qquad n = 1,2,....$$

When all calculations are done in this problem, the inner iteration is taken as 3. L_2 and L_∞ error norms, as well as total CPU times calculated by $\mathcal{L}\circ\mathcal{N}$, $\mathcal{N}\circ\mathcal{L}$, $\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$, $\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$ methods, are compared in the Tables 1, 2. When we look at the L_2 and L_∞ error norms and CPU times, it has been seen that the error norms are close to each other, but the $\mathcal{L}\circ\mathcal{N}$ and $\mathcal{N}\circ\mathcal{L}$ methods have given faster results.

Table 1. Comparison of problem 1's error norms L_2, L_∞ at different times t and ν for $h = 0.025$, $\Delta t = 0.0001$.

		$\nu = 1$		$\nu = 0.1$		$\nu = 0.01$	
Method	t	L_2	L_∞	L_2	L_∞	L_2	L_∞
	0.4	0.2071E-4	0.2928E-4	0.2625E-3	0.6906E-3	0.2465E-2	0.1403E-1
	0.6	0.4317E-5	0.6105E-5	0.2320E-3	0.5862E-3	0.2543E-2	0.1554E-1
$\mathcal{L} \circ \mathcal{N}$	0.8	0.7996E-6	0.1131E-5	0.1759E-3	0.4153E-3	0.2599E-2	0.1309E-1
	1.0	0.1388E-6	0.1963E-6	0.1298E-3	0.2894E-3	0.2443E-2	0.1187E-1
	3.0	0.0000E-6	0.0000E-6	0.2058E-4	0.3046E-4	0.4332E-3	0.1846E-2
	0.4	0.2062E-4	0.2916E-4	0.2228E-3	0.5907E-3	0.2355E-2	0.1316E-1
	0.6	0.4305E-5	0.6088E-5	0.2012E-3	0.5116E-3	0.2506E-2	0.1557E-1
$\mathcal{N} \circ \mathcal{L}$	0.8	0.7980E-6	0.1128E-5	0.1545E-3	0.3649E-3	0.2526E-2	0.1323E-1
	1.0	0.1386E-6	0.1960E-6	0.1151E-3	0.2567E-3	0.2373E-2	0.1113E-1
	3.0	0.0000E-6	0.0000E-6	0.1942E-4	0.2873E-4	0.4245E-3	0.1806E-2
	0.4	0.2066E-4	0.2922E-4	0.2426E-3	0.6407E-3	0.2410E-2	0.1360E-1
	0.6	0.4310E-5	0.6096E-5	0.2165E-3	0.5489E-3	0.2522E-2	0.1556E-1
$\mathcal{L} \circ \mathcal{N} \circ \mathcal{L}$	0.8	0.7987E-6	0.1129E-5	0.1651E-3	0.3901E-3	0.2561E-2	0.1316E-1
	1.0	0.1387E-6	0.1961E-6	0.1224E-3	0.2731E-3	0.2408E-2	0.1150E-1
	0.3	0.0000E-6	0.0000E-6	0.2000E-3	0.2960E-3	0.4289E-3	0.1826E-2
	0.4	0.2066E-4	0.2922E-4	0.2426E-3	0.6407E-3	0.2410E-2	0.1360E-1
	0.6	0.4311E-5	0.6097E-5	0.2165E-3	0.5489E-3	0.2522E-2	0.1556E-1
$\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$	0.8	0.7988E-6	0.1130E-5	0.1651E-3	0.3901E-3	0.2561E-2	0.1316E-1
	1.0	0.1387E-6	0.1962E-6	0.1224E-3	0.2731E-3	0.2408E-2	0.1150E-1
	0.3	0.0000E-6	0.0000E-6	0.2000E-4	0.2960E-4	0.4289E-3	0.1826E-2

Table 2. Comparison of CPU times at different t and ν of problem 1 for $h = 0.025$, $\Delta t = 0.0001$.

	Total CPU Time				
ν	t	$\mathcal{L} \circ \mathcal{N}$	$\mathcal{N} \circ \mathcal{L}$	$\mathcal{L} \circ \mathcal{N} \circ \mathcal{L}$	$\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$
	0.4	1.646	1.659	1.891	2.413
	0.6	2.169	2.134	2.437	3.249
1	0.8	2.558	2.598	2.998	4.108
	1.0	3.052	3.092	3.581	4.790
	3.0	7.501	7.764	9.127	12.737
	0.4	1.639	1.670	1.855	2.371
	0.6	2.108	2.194	2.431	3.107
0.1	0.8	2.566	2.544	2.967	3.921
	1.0	3.010	2.962	3.518	4.699
	3.0	7.637	7.514	9.177	12.937
	0.4	1.545	1.514	1.731	2.284
	0.6	1.988	2.028	2.302	3.069
0.01	0.8	2.380	2.395	2.839	3.878
	1.0	2.851	2.897	3.415	4.745
	3.0	7.386	7.599	9.113	12.567

Some nodal values are given at different positions and times for $\nu = 1$, 0.1, 0.01, $h = 0.025$, $\Delta t = 0.0001$ in the Tables 3, 4, 5 and compared with Ref. [9, 21, 29]. The results calculated with the proposed schemes are closer to the exact solution at some points according to the trigonometric cubic B-spline collocation method

Table 3. Comparison of some nodal values of problem 1 with exact solution and Ref. [9, 21, 29] at the different t and x for $h = 0.025$, $\Delta t = 0.0001$, $\nu = 1$.

x	t	$\mathcal{L}\circ\mathcal{N}$	$\mathcal{N}\circ\mathcal{L}$	$\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$	$\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$	Ref. [9]	Ref. [29]	Ref. [21]	Exact
0.25	0.4	0.01355	0.01355	0.01355	0.01355	0.01355	0.01357	0.01354	0.01357
	0.6	0.00188	0.00188	0.00188	0.00188	0.00188	0.00189	0.00188	0.00189
	0.8	0.00026	0.00026	0.00026	0.00026	0.00026	0.00026	0.00026	0.00026
	1.0	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004
	3.0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.5	0.4	0.01921	0.01921	0.01921	0.01921	0.01920	0.01923	0.01920	0.01924
	0.6	0.00267	0.00267	0.00267	0.00267	0.00266	0.00267	0.00266	0.00267
	0.8	0.00037	0.00037	0.00037	0.00037	0.00037	0.00037	0.00037	0.00037
	1.0	0.00005	0.00005	0.00005	0.00005	0.00005	0.00005	0.00005	0.00005
	3.0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.75	0.4	0.01361	0.01361	0.01361	0.01361	0.01361	0.01362	0.01360	0.01363
	0.6	0.00189	0.00189	0.00189	0.00189	0.00188	0.00189	0.00188	0.00189
	0.8	0.00026	0.00026	0.00026	0.00026	0.00026	0.00026	0.00026	0.00026
	1.0	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004	0.00004
	3.0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 4. Comparison of some nodal values of problem 1 with exact solution and Ref. [9, 21, 29] at the different t and x for $h = 0.025$, $\Delta t = 0.0001$, $\nu = 0.1$.

x	t	$\mathcal{L}\circ\mathcal{N}$	$\mathcal{N}\circ\mathcal{L}$	$\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$	$\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$	Ref. [9]	Ref. [29]	Ref. [21]	Exact
0.25	0.4	0.30893	0.30892	0.30892	0.30892	0.30892	0.30890	0.30892	0.30889
	0.6	0.24078	0.24078	0.24078	0.24078	0.24078	0.24075	0.24077	0.24075
	0.8	0.19573	0.19572	0.19573	0.19573	0.19572	0.19569	0.19572	0.19569
	1.0	0.16262	0.16261	0.16262	0.16262	0.16261	0.16258	0.16261	0.16258
	3.0	0.02719	0.02719	0.02719	0.02719	0.02718	0.02720	0.02718	0.02720
0.5	0.4	0.56972	0.56971	0.56972	0.56972	0.56971	0.56965	0.56970	0.56963
	0.6	0.44731	0.44730	0.44731	0.44731	0.44730	0.44723	0.44729	0.44723
	0.8	0.35932	0.35932	0.35932	0.35932	0.35932	0.35925	0.35930	0.35925
	1.0	0.29197	0.29197	0.29197	0.29197	0.29197	0.29192	0.29195	0.29192
	3.0	0.04018	0.04018	0.04018	0.04018	0.04017	0.04019	0.04016	0.04019
0.75	0.4	0.62521	0.62527	0.62524	0.62524	0.62524	0.62538	0.62520	0.62544
	0.6	0.48696	0.48701	0.48699	0.48699	0.48798	0.48715	0.48694	0.48721
	0.8	0.37368	0.37372	0.37370	0.37370	0.37369	0.37385	0.37365	0.37392
	1.0	0.00004	0.28729	0.28728	0.28728	0.28727	0.28741	0.28724	0.28747
	3.0	0.02975	0.02975	0.02975	0.02975	0.02974	0.02976	0.02974	0.02977

without using splitting the Ref. [9] and modified cubic B-splines collocation method the Ref. [21]. Furthermore, our results appear to be in good agreement with those of Ref. [29], which uses the cubic B-spline collocation method with splitting. The physical behaviors of the exact and approximate solutions for $\nu = 1$, 0.1, 0.01 at different times are obtained with the $\mathcal{N}\circ\mathcal{L}$ scheme displayed in Figure 1. It has been seen from the figures that the approximate solution is in a good harmony with the exact solution. Also, since the exact solution is distorted for small values of ν, only the approximate solution is plotted for $\nu = 0.001$.

Problem 2. As the second problem, the Burgers' equation is taken with the following exact solution

$$U(x,t) = \frac{\alpha + \mu + (\mu - \alpha)\exp\eta}{1 + \exp\eta},\ 0 \le x \le 1,\ t \ge 0, \text{ where } \eta = \frac{\alpha(x - \mu t - \gamma)}{\nu}.$$

Table 5. Comparison of some nodal values of problem 1 with exact solution and Ref. [9, 21, 29] at the different t and x for $h = 0.025$, $\Delta t = 0.0001$, $\nu = 0.01$.

x	t	$\mathcal{L}\circ\mathcal{N}$	$\mathcal{N}\circ\mathcal{L}$	$\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$	$\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$	Ref. [9]	Ref. [29]	Ref. [21]	Exact
0.25	0.4	0.34192	0.34192	0.34192	0.34192	0.34191	0.34192	0.34192	0.34192
	0.6	0.26897	0.26897	0.26897	0.26897	0.26896	0.26896	0.26896	0.22896
	0.8	0.22148	0.22148	0.22148	0.22148	0.22148	0.22148	0.22148	0.22148
	1.0	0.18819	0.18819	0.18819	0.18819	0.18819	0.18819	0.18819	0.18819
	3.0	0.07511	0.07511	0.07511	0.07511	0.07511	0.07511	0.07511	0.07511
0.5	0.4	0.66072	0.66071	0.66072	0.66072	0.66071	0.66071	0.66071	0.66071
	0.6	0.52942	0.52942	0.52942	0.52942	0.52942	0.52942	0.52942	0.52942
	0.8	0.43914	0.43914	0.43914	0.43914	0.43914	0.43914	0.43914	0.43914
	1.0	0.37442	0.37442	0.37442	0.37442	0.37442	0.37442	0.37442	0.37442
	3.0	0.15018	0.15018	0.15018	0.15018	0.15017	0.15018	0.15018	0.15018
0.75	0.4	0.91029	0.91029	0.91029	0.91029	0.91029	0.91027	0.91027	0.91026
	0.6	0.76726	0.76725	0.76725	0.76725	0.76725	0.76725	0.76724	0.76724
	0.8	0.64740	0.64740	0.64740	0.64740	0.64740	0.64740	0.64740	0.64740
	1.0	0.55605	0.55605	0.55605	0.55605	0.55605	0.55605	0.55605	0.55605
	3.0	0.22490	0.22490	0.22490	0.22490	0.22489	0.22483	0.22483	0.22481

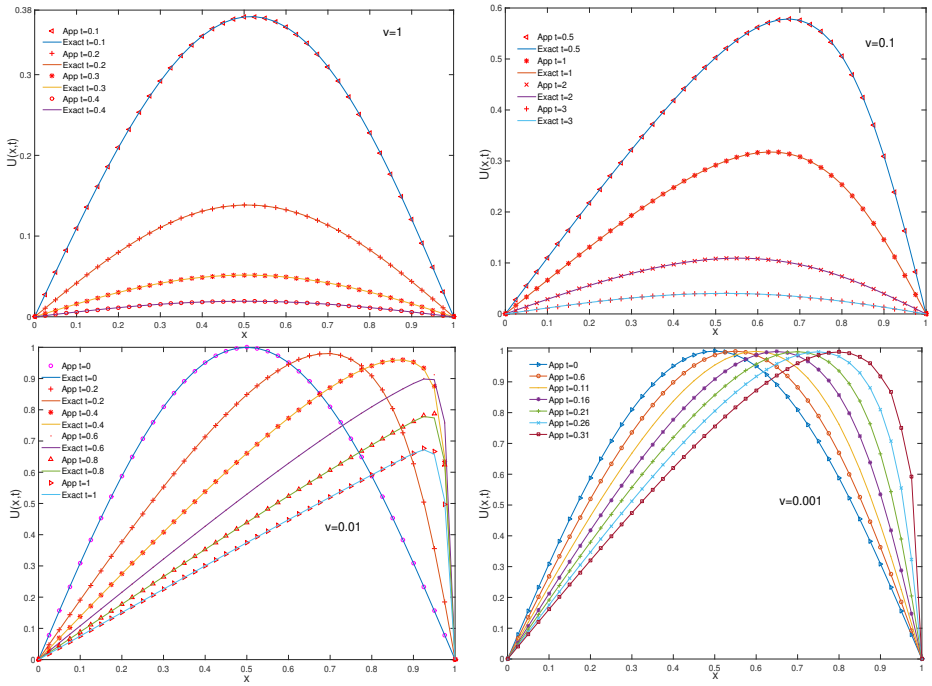

Figure 1. Exact and numerical solutions of Problem 1 obtained by $\mathcal{N}\circ\mathcal{L}$ method for different ν and t values for $h = 0.025$, $\Delta t = 0.0001$.

The boundary conditions for the system (4.5) and (4.6) are $f(t) = 1$, $g(t) = 0.2$, $h(t) = i(t) = 0$, the initial solutions are taken from the exact solution while $t = 0$ and the boundary conditions for the initial solutions are obtained from $U_{xx}(a, 0) = f_1(t)$, $U_{xx}(b, 0) = f_2(t)$.

In this problem, when all calculations are made, the inner iteration is taken as 2. The L_2, L_∞ error norms and CPU times at $t = 0.4, 0.5, 0.8, 1.2$ are calculated

Table 6. Comparison of the L_2, L_∞ error norms of problem 2 at the different t for $h = 1/36$, $\Delta t = 0.01$, $\nu = 0.01$ and compared with the Ref. [1, 11, 21, 25] for $t = 0.5$.

t		$\mathcal{L}\circ\mathcal{N}$	$\mathcal{N}\circ\mathcal{L}$	$\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$	$\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$
	L_2	0.311461E-2	0.137427E-2	0.211717E-2	0.161583E-2
0.4	L_∞	0.127109E-1	0.437095E-2	0.575369E-2	0.560973E-2
	Total CPU T.	$0.056s$	$0.049s$	$0.063s$	$0.069s$
	L_2	0.320078E-2	0.142152E-2	0.218591E-2	0.161787E-2
0.5	L_∞	0.125215E-1	0.466784E-2	0.558301E-2	0.509848E-2
	Total CPU T.	$0.060s$	$0.057s$	$0.064s$	$0.070s$
	L_2	0.343313E-2	0.152890E-2	0.241270E-2	0.161441E-2
0.8	L_∞	0.151762E-1	$0.466922E$-2	0.678776E-2	0.542833E-2
	Total CPU T.	$0.065s$	$0.060s$	$0.068s$	$0.082s$
	L_2	0.382040E-2	0.172349E-2	0.276081E-2	0.164411E-2
1.2	L_∞	0.149692E-1	0.454242E-2	0.748851E-2	0.440989E-2
	Total CPU T.	$0.068s$	$0.063s$	$0.076s$	$0.091s$
		Ref. [25]	Ref. [21]	Ref. [11]	Ref. [1]
	L_2	1.72434E-3	1.45E-3	5.86664E-3	1.48311E-3
0.5	L_∞	5.78454E-3	5.97E-3	22.2345E-3	5.51404E-3

Table 7. Comparison of some nodal values of problem 2 with exact solution and Ref. [9, 11, 21] at the $t = 0.5$ for $h = 1/36$, $\Delta t = 0.01$, $\nu = 0.01$.

x	$\mathcal{L}\circ\mathcal{N}$	$\mathcal{N}\circ\mathcal{L}$	$\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$	$\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$	Ref. [9]	Ref. [21]	Ref. [11]	Exact
0.000	1.	1.	1.	1.	1.	1.	1.	1.
0.056	1.	1.	1.	1.	1.	1.	1.	1.
0.111	1.	1.	1.	1.	1.	1.	1.	1.
0.167	1.	1.	1.	1.	1.	1.	1.	1.
0.222	1.	1.	1.	1.	1.	1.	1.	1.
0.278	0.999	0.999	1.001	0.999	0.999	0.998	1.	0.998
0.333	0.981	0.984	0.985	0.982	0.983	0.982	0.983	0.980
0.389	0.835	0.848	0.845	0.843	0.845	0.844	0.825	0.847
0.444	0.458	0.451	0.457	0.457	0.456	0.458	0.465	0.452
0.500	0.239	0.235	0.237	0.237	0.237	0.238	0.244	0.238
0.556	0.203	0.203	0.203	0.203	0.203	0.203	0.204	0.204
0.611	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
0.667	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
0.722	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
0.778	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
0.833	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
0.889	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
0.944	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200
1.000	0.200	0.200	0.200	0.200	0.200	0.200	0.200	0.200

with $\mathcal{L}\circ\mathcal{N}$, $\mathcal{N}\circ\mathcal{L}$, $\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$, $\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$ schemes for the values $\alpha = 0.4$, $\gamma = 0.125$, $\mu = 0.6$, $\nu = 0.01$ and $h = 1/36$, $\Delta t = 0.01$ are given in the Table 6. Also, the L_2, L_∞ error norms are compared with Ref. [1, 11, 21, 25] at $t = 0.5$ and it has been seen that our results are smaller than those given in other studies. The calculated results with all the schemes in terms of CPU times are in good agreement and the $\mathcal{L}\circ\mathcal{N}$, $\mathcal{N}\circ\mathcal{L}$ schemes are faster than $\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$, $\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$. In the Table 7, some nodal values calculated with four splitting schemes are given in the range $0 \le x \le 1$ for $\nu = 0.01$, $h = 1/36$, $\Delta t = 0.01$ at $t = 0.5$ and compared with the Ref. [9, 11, 21].

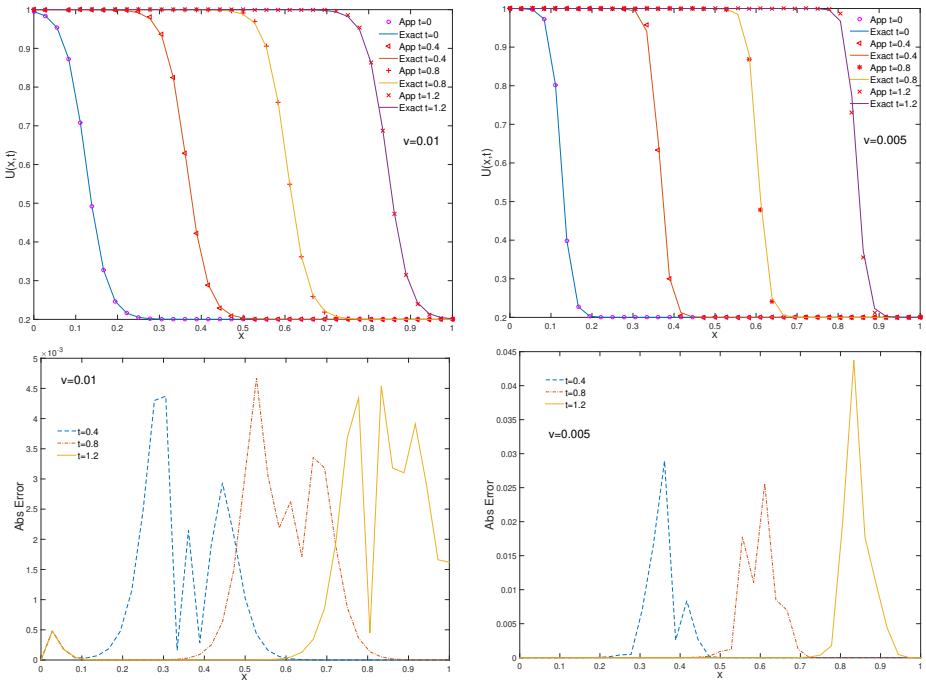

Figure 2. Physical behaviors of the solution and absolute error of problem 2 obtained by $\mathcal{N} \circ \mathcal{L}$ method for different ν and t values for $h = 1/36$, $\Delta t = 0.01$.

It is seen that the approximate results obtained by splitting schemes are in good harmony with other studies and even better converge to the exact solution at some points. The physical behavior of the problem and the absolute error for the values $\nu = 0.01,\ 0.005$ and $h = 1/36$, $\Delta t = 0.01$ are plotted using the $\mathcal{N} \circ \mathcal{L}$ scheme in Figure 2. As seen from the figure, as ν becomes smaller, the waves become steeper and move to the right over time.

Problem 3. As a last problem, the solutions of Burgers' equation modeling the shock wave problem, which becomes sharper as the viscosity parameter ν decreases, are examined. The initial condition and exact solution of Burgers' equation for $t = 1$ for this problem are as follows

$$U(x,1) = \frac{x}{1 + \sqrt{1/t_0}\exp(x^2/(4\nu))} \text{ and } U(x,t) = \frac{x/t}{1 + \sqrt{t/t_0}\exp(x^2/(4\nu t))}.$$

Where $t_0 = \exp(1/(8\nu))$, $t \geq 1$, $0 \leq x \leq 1$ and boundary conditions $f(t) = g(t) = h(t) = i(t) = 0$.

For this problem, the inner iteration is taken as 9 in all calculations. The total CPU times of the calculations obtained with the four numerical schemes for $\Delta t = 0.01$ are given in the Table 8. The L_2, L_∞ error norms calculated with the $\mathcal{L} \circ \mathcal{N}$, $\mathcal{N} \circ \mathcal{L}$, $\mathcal{L} \circ \mathcal{N} \circ \mathcal{L}$ and $\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$ schemes are given in the Table 9 and compared to those in Ref. [1, 24, 25]. As can be seen from the Table 9, the results calculated with the four schemes are in agreement with other studies, and the error norms obtained with the $\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$ scheme are smaller. Furthermore, while the Ref. [24](QBCM2) study used a split methodology, the L_2, L_∞ error norms

Table 8. Comparison of CPU times for different h, t and ν of problem 3 for $\Delta t = 0.01$.

			Total CPU Time			
ν	h	t	$\mathcal{L}\circ\mathcal{N}$	$\mathcal{N}\circ\mathcal{L}$	$\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$	$\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$
		1.7	0.304	0.331	0.336	0.545
0.005	0.005	2.4	0.633	0.558	0.646	1.145
		3.1	0.867	0.877	0.876	1.568
		1.8	0.081	0.080	0.115	0.164
0.005	0.02	2.4	0.131	0.125	0.164	0.251
		3.2	0.185	0.180	0.203	0.337
		1.7	0.075	0.077	0.078	0.120
0.01	0.02	2.1	0.099	0.110	0.111	0.174
		2.6	0.144	0.138	0.153	0.252

Table 9. Comparison of problem 3's error norms L_2, L_∞ for different h, t, ν and $\Delta t = 0.01$.

			$L\circ N$		$N\circ L$		$L\circ N\circ L$		$N\circ L\circ N$	
h	ν	t	$L_2\times10^3$	$L_\infty\times10^3$	$L_2\times10^3$	$L_\infty\times10^3$	$L_2\times10^3$	$L_\infty\times10^3$	$L_2\times10^3$	$L_\infty\times10^3$
		1.7	0.37076	1.29496	0.33200	1.15368	0.03558	0.16065	0.02926	0.12473
0.005	0.005	2.4	0.25631	0.85606	0.23821	0.78146	0.02209	0.08303	0.01943	0.06770
		3.1	0.63477	4.79061	0.71135	4.79061	0.65023	4.79061	0.65010	4.79061
		1.8	0.64544	2.35585	0.24122	0.65249	0.35798	1.40339	0.35404	1.37448
0.005	0.02	2.4	0.45170	1.54547	0.19201	0.49144	0.24567	0.88582	0.24372	0.87075
		3.2	1.23589	7.49146	1.27728	7.49146	1.24571	7.49146	1.24545	7.49147
		1.7	0.31859	0.85415	0.08616	0.22512	0.15368	0.43257	0.15294	0.42712
0.01	0.02	2.1	0.32795	1.14759	0.23627	1.14759	0.24322	1.14759	0.24304	1.14760
		2.6	1.53580	8.06798	1.58845	8.06798	1.55803	8.06798	1.55795	8.06798
			[25]		[1]		[24]QBCM1		[24]QBCM2	
			$L_2\times10^3$	$L_\infty\times10^3$	$L_2\times10^3$	$L_\infty\times10^3$	$L_2\times10^3$	$L_\infty\times10^3$	$L_2\times10^3$	$L_\infty\times10^3$
		1.7	0.35008	1.21175	0.01699	0.06511	0.01705	0.06192	0.35891	1.21170
0.005	0.005	2.4	0.24439	0.80771	0.01463	0.04322	0.01252	0.05882	0.25132	0.80777
		3.1	0.63309	4.79061	0.65079	4.79061	0.60199	4.43469	0.63052	4.79061
		1.8	0.33376	1.15263	0.36608	1.42677	0.19127	0.54058	0.49130	1.16930
0.005	0.02	2.4	0.24522	0.80008	0.25011	0.89905	0.14246	0.39241	0.41864	0.93664
		3.2	1.22981	7.49147	0.12456	7.49147	0.93617	5.54899	1.28863	7.49147
		1.7	0.17792	0.47456	0.15117	0.42203	0.17014	0.40431	0.24003	0.48800
0.01	0.02	2.1	0.24875	1.14759	0.24363	1.14760	0.20476	0.86363	0.30849	1.14760
		2.6	1.54108	8.06798	1.55800	8.06799	1.29951	6.69425	1.57548	8.06799

obtained by the operator splitting method we used are less. In Figure 3, absolute error graphs are shown as well as exact and approximate solutions. As can be seen from Figure 3, the exact solution and the approximate solution overlap. Also, for $\nu = 0.005$, the shock wave is soft and travels to the right with decreasing amplitude. For $\nu = 0.0005$, it is clearly seen that the shock wave becomes sharper. Since we take the boundary conditions as zero, the absolute error at $t = 3.1$ for $v = 0.005$ is greatest in the right boundary. When the solution region is $[0, 1.2]$ instead of $[0, 1]$ for $v = 0.005$ at $t = 3.1$, the L_2 error norm changes from 0.65010×10^3 to 0.01223×10^3. In addition, the L_∞ error norm decreases from 4.79061×10^3 to 0.04382×10^3.

6. Conclusions

In this study, Burgers' equation is divided into two sub-problems such that linear $U_t = \mathcal{L}(U)$ and nonlinear $U_t = \mathcal{N}(U)$. $\mathcal{L}\circ\mathcal{N}$, $\mathcal{N}\circ\mathcal{L}$, $\mathcal{L}\circ\mathcal{N}\circ\mathcal{L}$ and $\mathcal{N}\circ\mathcal{L}\circ\mathcal{N}$ numerical solution schemes based on operator splitting methods are used to obtain

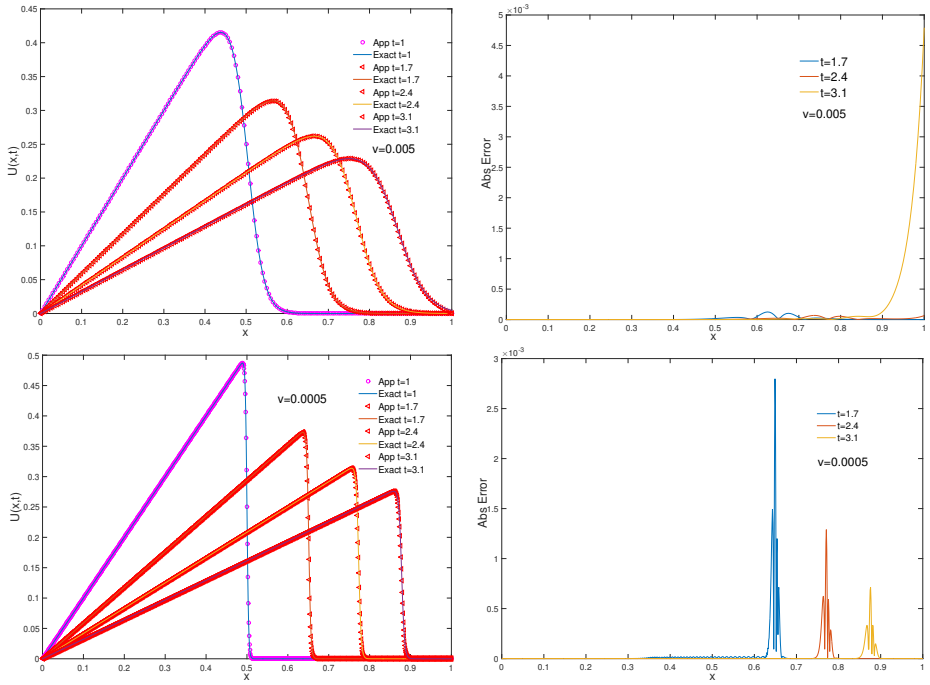

Figure 3. Physical behaviors of the solution and absolute error of problem 3 obtained by $\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$ method for $\nu = 0.005$, $h = 0.005$ and $\nu = 0.0005$, $h = 0.001$, $\Delta t = 0.01$.

the solution of the main equation by using sub-problems. Each sub-problem has been solved by the finite element trigonometric cubic B-spline collocation method. The numerical results calculated with these schemes are compared among themselves and with some studies in the literature. According to the numerical data, the results calculated with the $\mathcal{L} \circ \mathcal{N}$, $\mathcal{N} \circ \mathcal{L}$ schemes are found to be reasonably good and faster in terms of CPU time than the $\mathcal{L} \circ \mathcal{N} \circ \mathcal{L}$, $\mathcal{N} \circ \mathcal{L} \circ \mathcal{N}$ schemes. In addition, in comparison with Ref. [9], which solves Burgers' equation directly with trigonometric cubic B-spline collocation, it has been seen that solving the equation using operator splitting has more convergent at some points. It is seen from the results that the proposed methods are easy to apply and economical. The operator splitting trigonometric cubic B-spline collocation method will be effective in solving many nonlinear equations as well as Burgers' equation.

References

[1] B. Ay, İ. Dağ and M. Z. Görgülü, *Trigonometric quadratic B-spline subdomain Galerkin algorithm for the Burgers' equation*, Open Phys., 2015, 13, 400–406. DOI 10.1515/phys-2015-0059

[2] H. Bateman, *Some recent researches on the motion of fluids*, Monthly Weather Rev., 1915, 43, 163–170.

[3] J. M. Burgers, *A mathematical model illustrating the theory of turbulence*, Adv. Appl. Mech., 1948, 1, 171–199.

[4] J. M. Burgers, *Mathematical examples illustrating relations occurring in the*

theory of turbulent fluid motion, Trans. R. Neth. Acad. Sci. Amst., 1939, 17, 1–53.

[5] A. G. Bratsos, *A fourth-order numerical scheme for solving the modified Burgers' equation*, Comput. Math. with Appl., 2010, 60, 1393–1400. doi:10.1016/j.camwa.2010.06.021

[6] H. Brezis and F. Browder, *Partial Differential Equations in the 20th Century*, 1998, 135, 76–144. Article No. AI971713.

[7] S. Blanes, F. Casas and A. Murua, *Splitting and composition methods in the numerical integration of differential equations*, Bol. Soc. Esp. Mat. Apl., 2008, 45, 89–145.

[8] J. D. Cole, *On a quasilinear parabolic equation occurring in aerodynamics*, Quart. Appl. Math., 1951, 9, 225–236.

[9] İ. Dağ, Ö. E. Hepson and Ö. Kaçmaz, *The trigonometric cubic B-spline Algorithm for Burgers' Equation*, Int. J. Nonlinear Sci., 2017, 24(2), 120–128.

[10] İ. Dağ, D. Irk and B. Saka, *A numerical solution of the Burgers' equation using cubic B-splines*, Appl. Math. Comput., 2005, 163, 199–211. doi:10.1016/j.amc.2004.01.028

[11] İ. Dağ, D. Irk and A. Şahin, *B-spline collocation methods for numerical solitions of the Burgers' equation*, Math. Probl. Eng., 2005, 5, 521–538. DOI: 10.1155/MPE.2005.521

[12] Ö. Ersoy and G. Yiğit, *Quartic Trigonometric Tension B-Spline Method for the Solution of the Advection-Diffusion Equation*, Comput. Appl. Math., 2021, 40(141), 1–15. doi.org/10.1007/s40314-021-01526-2

[13] O. Ersoy, I. Dag and N. Adar, *The exponential cubic B-spline algorithm for Burgers' equation*, Ukr. Math. J., 2018, 70(6), 906–921. doi.org/10.1007/s11253-018-1541-9

[14] Ö. Ersoy, *Generation Of The Trigonometric Cubic B-spline Collocation Solutions for the KuramotoSivashinsky(KS) Equation*, AIP Conf. Proc., 2018, 470099. doi.org/10.1063/1.5044169

[15] Ö. Ersoy, *A Numerical Simulation of Kuramoto-Sivashinsky Equation in Reaction-Diffusion via Galerkin Method*, Mathematical Science, 2021, 15, 199–206. doi.org/10.1007/s40096-021-00402-8

[16] Y. Gao, L. Le and B. Shi, *Numerical solution of Burgers' equation by lattice Boltzmann method*, Appl. Math. Comput., 2013, 219, 7685–7692. doi.org/10.1016/j.amc.2013.01.056

[17] W. Hundsdorfer, *Numerical Solution of Advection-Diffüsion-Reaction Equations*, Lecture notes for PH. D. course, Thomas Stieltjes İnstitute, Amsterdam, 2000.

[18] S. Kutluay, A. Esen and İ. Dağ, *Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method*, J. Comput. Appl. Math., 2004, 167, 21–33. doi:10.1016/j.cam.2003.09.043

[19] S. Kutluay and A. Esen, *A Lumped galerkin method for solving the Burgers' equation*, Int. J. Comput. Math., 2004, 81, 1433–1444. DOI: 10.1080/00207160412331286833

[20] P. E. Koch, *Multivariate trigonometric B-splines*, J. Approx. Theory, 1988, 54, 162–168.

[21] R. C. Mittal and R. K. Jain, *Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method*, Appl. Math. Comput., 2012, 218, 7839–7855. doi:10.1016/j.amc.2012.01.059

[22] G. I. Marchuk, *Some application of splitting-up methods to the solution of mathematical physics problems*, Aplikace matematiky, 1968, 13, 103–132.

[23] A. T. Onarcan and Ö. Ersoy, *Higher order Trigonometric B-spline Algorithms to the solution of Coupled Burgers' Equation*, AIP Conf. Proc., 2018, 1926, 020044. doi.org/10.1063/1.5020493

[24] B. Saka and İ. Dağ, *Quartic B-spline collocation method to the numerical solutions of the Burgers' equation*, Chaos Solitons Fractals, 2007, 32, 1125–1137. doi:10.1016/j.chaos.2005.11.037

[25] B. Saka and İ. Dağ, *A numerical study of the Burgers' equation*, J. Franklin Inst., 2008, 345, 328–348. doi:10.1016/j.jfranklin.2007.10.004

[26] M. Seydaoğlu, U. Erdoğan and T. Öziş, *Numerical solution of Burgers' equation with high order splitting methods*, J. Comput. Appl. Math., 2016, 291, 410–421.

[27] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Numer. Anal., 1968, 5, 506–517.

[28] H. F. Trotter, *On the product of semi-groups of operators*, Proc. American Math. Society, 1959, 10, 545–551.

[29] Y. Uçar, N. M. Yağmurlu and İ. Çelikkaya, *Numerical Solution of Burger's Type Equation Using Finite Element Collocation Method with Strang Splitting*, M. S. A. E. N., 2020, 8(1), 29–45. DOI.ORG/10.36753/MATHENOT.598635

[30] G. Walz, *Identities for trigonometric B-splines with an application to curve design*, BIT, 1997, 37, 189–201.