Numerical Simulations of Shock Wave Propagating by a Hybrid Approximation Based on High-Order Finite Difference Schemes

A. Zeytinoglu[a], M. Sari[b,*] and B.P. Allahverdiev[a]

[a]Department of Mathematics, Suleyman Demirel University, Isparta, Turkey

[b]Department of Mathematics, Yildiz Technical University, Istanbul, Turkey

(Received May 8, 2017; in final form October 19, 2017)

In this paper, we attempt to display effective numerical simulations of shock wave propagating represented by the Burgers equations known as a significant mathematical model for turbulence. A high order hybrid approximation based on seventh order weighted essentially non-oscillatory finite difference together with the sixth order finite difference scheme implemented for spatial discretization is presented and applied without any transformation or linearization to the Burgers equation and its modified form. Then, the produced system of first order ordinary differential equations is solved by the MacCormack method. The efficiency, accuracy and applicability of the proposed technique are analyzed by considering three test problems for several values of viscosity that can be caused by the steep shock behavior. The performance of the method is measured by some error norms. The results are in good agreement with the results reported previously, and moreover, the suggested approximation relatively comes to the forefront in terms of its low cost and easy implementation.

DOI: 10.12693/APhysPolA.133.140

PACS/topics: 02.60.Cb, 02.70.Bf, 47.11.Bc, 47.40.Nm

1. Introduction

The real world problems in many scientific areas such as plasma physics, acoustics, fluid mechanics, electricity, hydraulics, elasticity, structural analysis, magnetism, optics etc. are represented by partial differential equations or systems, generally, in nonlinear form. One of the significant model equations is viscous Burgers equation introduced first by Bateman [1] and then treated as a mathematical model for turbulence by Burgers [2, 3]. Since it includes three important features of the Navier–Stokes equations: diffusion, nonlinear convection and unsteadiness, it is known as a simple nonlinear PDE comprising diffusion and convection in fluid mechanics. This equation is introduced to describe the shock wave behaviors, characteristics of turbulent flow caused by the interaction of the opposite effects of diffusion and convection, mass transport, continuous stochastic processes, gas dynamics, longitudinal elastic waves in an isotropic solid, sound waves in a viscous medium, wave processes in thermoelastic media, transport and dispersion of pollutants in rivers, traffic flow, etc. The equation is given in general form by

$$u_t + u^{\mu} u_x - \upsilon u_{xx} = 0, \quad x \in R, t > 0 \qquad (1)$$

where υ is kinematic viscosity checking the balance between diffusion and convection, and μ is a positive constant. The above equation is called as the Burgers equation with $\mu = 1$, while it is known as the modified Burgers equation that has the strong nonlinearity for $\mu \geq 2$. In both case, shock behavior occurs when the value of viscosity υ is taken smaller. These equations have some analytical solutions involving infinite series, but they are not practical enough due to the slow convergence of them for small viscosity values. Hence, the derivation of reliable, accurate, applicable, and efficient methods for simulation of these problems is both necessary and important for technological and scientific developments in many disciplines. In order to help engineers and physicists for the above equation and their applications, a lot of numerical methods have been derived and developed to understand correctly the process of the physical model for many years. Especially, researchers have given their attention for solving the Burgers or modified Burgers equations with small viscosity parameters. Various numerical studies including such as the Galerkin finite element method [4], least-squares quadratic B-splines finite element method [5], cubic [6] and quartic [7] B-spline collocation method, modified cubic B-splines collocation method [8], a reproducing kernel function technique [9], Sinc differential quadrature method [10], fourth order finite difference method [11], implicit fourth order compact finite difference scheme [12], a sixth order compact finite difference method [13] etc. have been presented for numerical solutions of the Burgers equation. On the other hand, several numerical techniques suggested for solving the modified Burgers equation can be summarized as follows.

Septic B-splines collocation method is presented by Ramadan et al. [14]. A collocation method based on quantic splines is used by Ramadan and El-Danaf [15]. Saka and Dag [16] apply time and space splitting technique and then employ quintic B-spline collocation proce-

[*] corresponding author; e-mail: sarim@yildiz.edu.tr

dure. Duan et al. [17] develop a special lattice Boltzmann method. A Chebyshev spectral collocation method, known as El-Gendi method, is elaborated by Temsah [18]. Irk [19] proposes a collocation method based on sextic B-spline. An implicit finite difference scheme based on fourth order rational approximations and an explicit finite difference scheme based on second order rational approximations to the matrix-exponential term are proposed by Bratsos [20, 21], Bratsos and Petrakis [22], respectively. Gupta and Kadalbajoo [23] construct a numerical scheme that comprises of implicit Euler method for time discretization and a monotone hybrid finite difference operator for spatial discretization, and use the quasi-linearization process to tackle the nonlinearity. A Petrov–Galerkin method is applied by Roshan and Bhamra [24]. Also, Kutlay et al. [25] prefer to linearize the nonlinear term and implement a cubic B-spline collocation method.

The effort of finding a more accurate numerical approach is still in progress. Investigating an effective and accurate numerical method encourages us to produce a new hybrid approach based on some high order finite difference (FD) schemes for analyzing the shock wave propagating. One of these FD schemes is a seventh-order weighted essentially non-oscillatory (WENO7) [26–28] method. It is well-known that the WENO method is based on ENO schemes and these are very good alternatives for numerical solutions of PDEs in conservative form. High order accuracy can be achieved in the smooth regions and discontinuities can be computed without spurious oscillations [29]. Some studies in recent years have introduced various versions of the WENO scheme derived for improving ENO properties [26–32], while the WENO schemes have been combined with a high order method to overcome some drawbacks [33–35]. Inspired by these drawbacks in the corresponding studies, we prefer to combine the WENO7 scheme with the sixth-order finite difference (FD6) scheme [36, 37] because the FD6 gives convergence approximations as well as being effective, reliable, and easy to implement. Because of calculating an original nonlinear equation without any transformation or linearization provides preserving the actual physical properties of solutions, we directly implement the constructed scheme to the relevant terms in the equations. Some error norms are presented to demonstrate the applicability, validity, efficiency, and accuracy of the proposed method.

The organization of this paper is summarized as follows. In Sect. 2, the constructed hybrid scheme is introduced in both space and time. In Sect. 3, three test problems that the first one is an example for the Burgers equation with $\mu = 1$ and the others are examples for modified Burgers equation with $\mu = 2$ and $\mu = 3$, respectively, are solved. The results are presented by way of figures and tables compared with some previous studies in literature. In the last section, the findings in the paper are summarized.

2. Construction of the hybrid scheme

One can rewrite Eq. (1) with the following form:

$$u_t = -f(u)_x + \upsilon u_{xx}, \quad f(u)_x = (\frac{u^{\mu+1}}{\mu+1})_x. \tag{2}$$

The strength of nonlinearity of the equation changes for the different values of parameter μ. The proposed approach is involved the FD6 and WENO7 finite difference formulations to the spatial derivatives, and the MacCormack discretization is taken into account for the time derivative. Details of the implementation of the present method are introduced as follows.

First of all, we divide the domain of problem $[a, b]$ into N subintervals such as $a = x_1 < x_2 < \ldots < x_n < x_{N+1} = b$ with the spatial step size $h = \Delta x = x_{i+1} - x_i$ for $i = 1, 2 \ldots, N$. Also, $(n+1)$-th time level is defined by $t^{n+1} = t^n + \Delta t$ where t^n is the initial time for $n = 0$. Thus, the numerical solution of u is represented by u_i^n at grid point (x_i, t^n). To discretize the diffusion term u_{xx} in Eq. (2), we use the FD6 scheme derived for the second order derivatives. The FD6 scheme can briefly be introduced as follows.

u' and u'' in space, can be approximated by the following FD6 formulae using 7-point stencil

$$u_i' = \frac{1}{h}\sum_{j=-L}^{R} a_{j+L}u_{i+j}, u_i'' = \frac{1}{h^2}\sum_{j=-L}^{R} \tilde{a}_{j+L}u_{i+j},$$
$$1 \le i \le N+1.$$

In the above equations, $(N+1)$ denotes the number of grid points, a_k and $\tilde{a}_k (k = 0, \ldots, R+L)$ are unknown constants, R and L denote the number of grid points in the right and left hand side for the taken stencil, respectively. At internal points, R and L is equal while they are different for the boundary nodes.

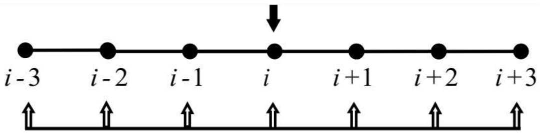


Fig. 1. Stencils for FD6 schemes.

For instance, in order to find the values of a_k for internal nodes, the above 7-point stencil is used (see Fig. 1), and the approximate values of each function at related points is calculated by the Taylor series expansions as written in Eq. (3):

$$u_{i-3} = u_i - 3hu_i' + \frac{(3h)^2}{2!}u_i'' - \frac{(3h)^3}{3!}u_i''' + \frac{(3h)^4}{4!}u_i^{(4)}$$
$$-\frac{(3h)^5}{5!}u_i^{(5)} + \frac{(3h)^6}{6!}u_i^{(6)} - \frac{(3h)^7}{7!}u_i^{(7)},$$
$$u_{i-2} = u_i - 2hu_i' + \frac{(2h)^2}{2!}u_i'' - \frac{(2h)^3}{3!}u_i''' + \frac{(2h)^4}{4!}u_i^{(4)}$$
$$-\frac{(2h)^5}{5!}u_i^{(5)} + \frac{(2h)^6}{6!}u_i^{(6)} - \frac{(2h)^7}{7!}u_i^{(7)},$$
$$u_{i-1} = u_i - hu_i' + \frac{h^2}{2!}u_i'' - \frac{h^3}{3!}u_i''' + \frac{h^4}{4!}u_i^{(4)} - \frac{h^5}{5!}u_i^{(5)}$$

$$+\frac{h^6}{6!}u_i^{(6)}-\frac{h^7}{7!}u_i^{(7)},$$

$$u_{i+1}=u_i+hu_i'+\frac{h^2}{2!}u_i''+\frac{h^3}{3!}u_i'''+\frac{h^4}{4!}u_i^{(4)}+\frac{h^5}{5!}u_i^{(5)}$$

$$+\frac{h^6}{6!}u_i^{(6)}+\frac{h^7}{7!}u_i^{(7)},$$

$$u_{i+2}=u_i+2hu_i'+\frac{(2h)^2}{2!}u_i''+\frac{(2h)^3}{3!}u_i'''+\frac{(2h)^4}{4!}u_i^{(4)}$$

$$+\frac{(2h)^5}{5!}u_i^{(5)}+\frac{(2h)^6}{6!}u_i^{(6)}+\frac{(2h)^7}{7!}u_i^{(7)},$$

$$u_{i+3}=u_i+3hu_i'+\frac{(3h)^2}{2!}u_i''+\frac{(3h)^3}{3!}u_i'''+\frac{(3h)^4}{4!}u_i^{(4)}$$

$$+\frac{(3h)^5}{5!}u_i^{(5)}+\frac{(3h)^6}{6!}u_i^{(6)}+\frac{(3h)^7}{7!}u_i^{(7)}. \qquad (3)$$

After implementing some algebraic operations to Eq. (3), the expression of each function belonging to internal nodes is found as

$$u_i'=\frac{-u_{i-3}+9u_{i-2}-45u_{i-1}+45u_{i+1}-9u_{i+2}+u_{i+3}}{60h}$$

$$-0.007142857h^6u_i^{(7)},$$

where the term $0.007142857h^6u_i^{(7)}$ defines the truncation error. In a similar way, the coefficients a_k and $\tilde{a}_k$ for all nodes can be determined with the Taylor series expansions and found as seen in Table I.

For the term $f(u)_x$ in Eq. (2), the WENO7 scheme is implemented together with the FD6 scheme. The WENO schemes are based on ENO schemes and it was first suggested by Liu et al. [38]. They use a convex combination of all candidate stencils against the ENO schemes. Hence, high order accurate solutions in smooth regions and a good convergence are provided. In the literature, many researchers have focused on the WENO schemes in order to improve them. Taking inspiration from those studies, the present work discusses a combination of the WENO7 finite difference scheme with the FD6 scheme in computing highly accurate results. The mentioned WENO scheme is applied to internal nodes and the FD6 formulae given in above are implemented for near the boundaries. We can then introduce the WENO7 procedure with its main points herein below [26–28].

The WENO schemes are successful in terms of the numerical approximation for discretization of the spatial derivatives in the following hyperbolic conservation law

$$u_t+f(u)_x=0. \qquad (4)$$

To obtain high-order accurate solutions, a reconstruction procedure based on the local smoothness of numerical solution is used as the main point of the WENO finite difference scheme. The term $f(u)_x$ called the numerical flux is approximated by

$$f(u)_x|_{x=x_j}\approx\frac{1}{\Delta x}(\hat{f}_{j+\frac{1}{2}}-\hat{f}_{j-\frac{1}{2}}). \qquad (5)$$

The WENO7 scheme uses 7 candidate stencils written as a set $S=\{x_{j-3},\dots,x_{j+3}\}$ for these numerical fluxes. It is divided into four subset as $S^m=\{x_{j-3+m},\dots,x_{j+m}\}$, $m=0,1,2,3$ as seen in Fig. 2.

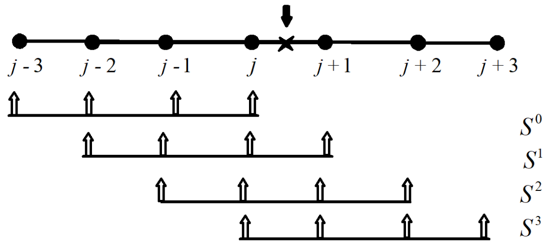


Fig. 2. Candidate stencils for the WENO7 scheme.

The coefficients a_k and $\tilde{a}_k$. TABLE I

i		k						
		0	1	2	3	4	5	6
1	a_k	–147	360	–450	400	–225	72	–10
	$\tilde{a}_k$	812	–3132	5265	–5080	2970	–972	137
2	a_k	–10	–77	150	–100	50	–15	2
	$\tilde{a}_k$	137	–147	–255	470	–285	93	–13
3	a_k	2	–24	–35	80	–30	8	–1
	$\tilde{a}_k$	–13	228	–420	200	15	–12	2
internal nodes	a_k	–1	9	–45	0	45	–9	1
	$\tilde{a}_k$	2	–27	270	–490	270	–27	2
$N-1$	a_k	1	–8	30	–80	35	24	–2
	$\tilde{a}_k$	2	–12	15	200	–420	228	–13
N	a_k	–2	15	–50	100	–150	77	10
	$\tilde{a}_k$	–13	93	–285	470	–255	–147	137
$N+1$	a_k	10	–72	225	–400	450	–360	147
	$\tilde{a}_k$	137	–972	2970	–5080	5265	–3132	812

*Each given value of a_k and $\tilde{a}_k$ in the table must be divided by 60 and 180, respectively

Using these stencil sets S^m, the numerical flux $\hat{f}_{j+\frac{1}{2}}$ is written as

$$\hat{f}_{j+\frac{1}{2}}=\sum_{m=0}^{3}\omega_m\hat{f}^{(m)}_{j+\frac{1}{2}}, \qquad (6)$$

where

$$\hat{f}^{(m)}_{j+\frac{1}{2}}=\sum_{i=0}^{3}b_{mi}f_{j+m-i}, \quad \omega_m=\frac{\alpha_m}{\sum_{k=0}^{3}\alpha_k},$$

$$\alpha_m=d_m(1+(\frac{\tau_7}{(\varepsilon+\beta_m)})^q). \qquad (7)$$

In Eq. (7), $\tau_7=|\beta_0-\beta_3|$, the linear weights are $d_0=1/35$, $d_1=12/35$, $d_2=18/35$, $d_3=4/35$, and ω_m are called non-linear weights. The coefficients b_{mi} can be calculated with the approach inspired by Xie [39] using a fourth order polynomial

$$h(x)=A+B(x-x_{j+1/2})+C(x-x_{j+1/2})^2$$

$$+D(x-x_{j+1/2})^3+E(x-x_{j+1/2})^4,$$

with the four candidate stencils in Fig. 2 and are listed in Table II. The coefficients required for $\hat{f}_{j-\frac{1}{2}}$ can be found using the same stencils in a similar way. In calculations, ε is used to avoid the division by zero and it is selected to be quite small, $\varepsilon=10^{-10}$ and $q=2$.

TABLE II
The coefficients b_{mi} for the WENO7 scheme.

b_{mi}	$i=0$	$i=1$	$i=2$	$i=3$
$m=0$	25/12	−23/12	13/12	−1/4
$m=1$	1/4	13/12	−5/12	1/12
$m=2$	−1/12	7/12	7/12	−1/12
$m=3$	1/12	−5/12	13/12	1/4

The smoothness indicators, β_m, are given by [28]:

$$\beta_0 = f(u_{j-3})$$
$$\times\left[547f(u_{j-3})-3882f(u_{j-2})+4642f(u_{j-1})-1854f(u_j)\right]$$
$$+f(u_{j-2})\left[7043f(u_{j-2})-17246f(u_{j-1})+7042f(u_j)\right]$$
$$+f(u_{j-1})\left[11003f(u_{j-1})-9402f(u_j)\right]+f(u_j)\left[2107f(u_j)\right],$$

$$\beta_1 = f(u_{j-2})$$
$$\times\left[267f(u_{j-2})-1642f(u_{j-1})+1602f(u_j)-494f(u_{j+1})\right]$$
$$+f(u_{j-1})\left[2843f(u_{j-1})-5966f(u_j)+1922f(u_{j+1})\right]$$
$$+f(u_j)\left[3443f(u_j)-2522f(u_{j+1})\right]+f(u_{j+1})\left[547f(u_{j+1})\right],$$

$$\beta_2 = f(u_{j-1})$$
$$\times\left[547f(u_{j-1})-2522f(u_j)+1922f(u_{j+1})-494f(u_{j+2})\right]$$
$$+f(u_j)\left[3443f(u_j)-5966f(u_{j+1})+1602f(u_{j+2})\right]$$
$$+f(u_{j+1})\left[2843f(u_{j+1})-1642f(u_{j+2})\right]+f(u_{j+2})\left[267f(u_{j+2})\right],$$

$$\beta_3 = f(u_j)$$
$$\times\left[2107f(u_j)-9402f(u_{j+1})+7042f(u_{j+2})-1854f(u_{j+3})\right]$$
$$+f(u_{j+1})\left[11003f(u_{j+1})-17246f(u_{j+2})+4642f(u_{j+3})\right]$$
$$+f(u_{j+2})\left[7043f(u_{j+2})-3882f(u_{j+3})\right]+f(u_{j+3})\left[547f(u_{j+3})\right].$$

For more details of the WENO finite difference scheme, interested readers are referred to literature such as [26, 28, 30].

After the implementation of the aforementioned schemes to Eq. (2), the MacCormack method is applied to calculate new values of u at the next time level. This method is widely used for solving nonlinear PDEs representing fluid flows and provides accurate results [40]. Let us consider the following general form of governing equation:

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = Pu_i.$$

In this form, P represents a spatial differential operator, and each value on the right hand side of the above equation are already known through the method described in above. The MacCormack approach is then implemented to solve this semi-discrete equation via the following process:

pre. step:

$$\overline{u_i^{n+1}} = u_i^n + \Delta t P u_i^n,$$

cor. step:

$$u_i^{n+1} = u_i^{n+1/2} + \frac{\Delta t}{2} P\overline{u_i^{n+1}},$$

$$u_i^{n+1/2} = \frac{u_i^n + \overline{u_i^{n+1}}}{2}.$$

3. Numerical treatments

In this section, the behaviors of abovementioned models for different values of viscosity are investigated and the performance of the suggested method is presented in terms of the accuracy. There are three test problems including one Burgers equation with $\mu = 1$ and two modified Burgers equation with $\mu = 2$ and $\mu = 3$. Some figures are drawn to reveal the behaviors of per models with some values of v.

To demonstrate the accuracy and validity of the method, L_2 and L_∞ error norms, defined as

$$L_2 = \sqrt{h\sum_{j=1}^{N}\left|u_j^{\text{analytical}} - u_j^{\text{numerical}}\right|^2},$$

$$L_\infty = \max\left|u_j^{\text{analytical}} - u_j^{\text{numerical}}\right|,$$

are computed for various values of viscosity v at different times t for chosen convenient space and time steps. The obtained results are summarized in tables with some known results in the literature. Also, the changes of absolute errors in the relevant domains, and the changes of L_2 and L_∞ errors in time are plotted.

Example 1. Firstly, we solve the Burgers equation which is of the form with $\mu = 1$:

$$u_t + uu_x - vu_{xx} = 0,$$

which has the following analytical solution representing shock-like solutions of the above equation

$$u(x,t) = \frac{x}{t}\frac{t}{1+\sqrt{\frac{t}{t_0}}\exp(\frac{x^2}{4vt})},\ 0 \le x \le 1,\ t \ge 1,$$

where $t_0 = \exp(1/8v)$. The initial condition is written from the exact solution as follows:

$$u(x,1) = x\frac{1}{1+\sqrt{\frac{1}{t_0}}\exp(\frac{x^2}{4v})},$$

and the boundary conditions are taken as $u(0,t) = 0, u(1,t) = 0$. This solution represents the propagation of the shock [41]. Numerical results of this problem are obtained by taking the viscosity v as 0.01, 0.005, 0.0015 and 0.0005 for various values of space and time steps. L_2 and L_∞ error norms are documented in Tables III–VI. The obtained results can be compared with some previous works given in the same tables.

We have seen from the corresponding tables that the error norms of the proposed algorithm for all values of viscosity v used here are quite small, and furthermore, better than most of the compared results. The compared

Comparison of the error norms at various times for $v = 0.0005$ for example 1. TABLE III

t	Present method ($h = 0.001$, $\Delta t = 10^{-5}$)		[10] ($h = 0.005$, $\Delta t = 0.001$)	
	$L_2 \times 10^4$	$L_\infty \times 10^3$	$L_2 \times 10^4$	$L_\infty \times 10^3$
1.1	0.034057	0.048405	–	–
1.7	0.113256	0.136531	2.01	2.374
2.4	0.117941	0.129910	0.87	0.662
3.1	0.112151	0.115540	0.52	0.377

Comparison of the error norms at various times for $v = 0.0015$ for example 1. TABLE IV

t	Present method ($h = 0.005$, $\Delta t = 0.001$)		[14] ($h = 0.005$, $\Delta t = 0.01$)	
	$L_2 \times 10^4$	$L_\infty \times 10^3$	$L_2 \times 10^4$	$L_\infty \times 10^3$
1.2	0.35996	0.24217	3.85388	3.23685
1.4	0.36445	0.27047	4.64435	3.34889
1.6	0.39823	0.29614	4.96475	3.19232
1.8	0.42992	0.29917	5.09279	2.98679
2	0.45435	0.30556	5.12515	2.78315
2.2	0.47144	0.30725	5.10596	2.59512
2.4	0.48316	0.31193	5.05801	2.42555
2.8	0.49596	0.30442	4.92110	2.13805
3	0.49886	0.30092	4.84391	2.01647
3.4	0.50396	0.29144	4.68731	1.80869
3.7	2.91257	3.04635	6.44632	4.70819

results have generally been calculated by using some finite element based methods known as one of the strong alternative approaches to produce accurate numerical solutions. It is worthy note that the produced results here have relatively higher accuracy than the literature results computed by some of these powerful alternatives. In addition to its efficiency, the proposed hybrid scheme relatively comes to the forefront in terms of its low cost and easy implementation to the problems. It is also noted from the tables that the errors have tendency to fluctuating decrease as v is decreased. The propagation of wave at various times for corresponding viscosity values are visualized in Fig. 3.

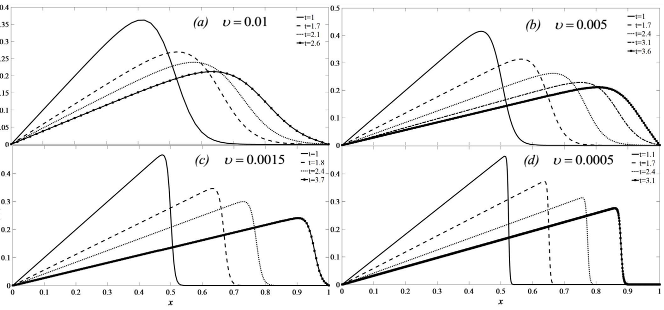


Fig. 3. The numerical solutions of example 1 at various times using several v parameters: (a) $v = 0.01$, (b) $v = 0.005$, (c) $v = 0.0015$, (d) $v = 0.0005$.

Comparison of the error norms at various times using $v = 0.01$, $\Delta t = 0.01$ and $h = 0.02$ for example 1. TABLE V

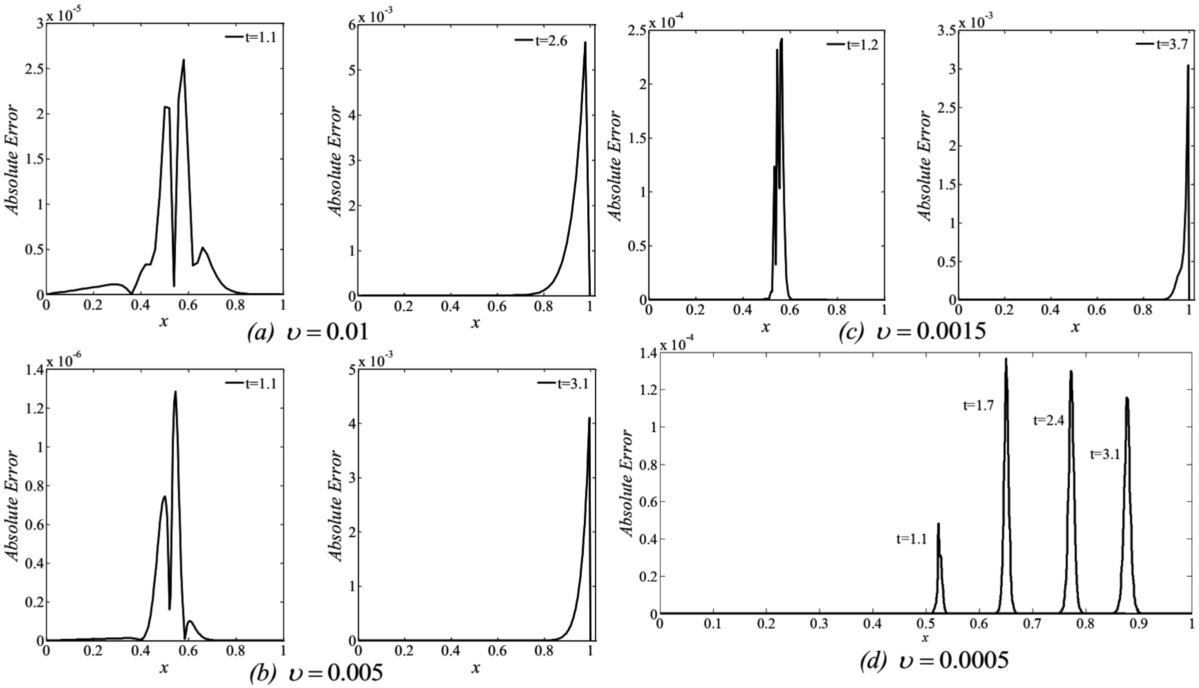
t	Present method		[14]	
	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1.1	0.00705	0.0260	0.4665	2.8683
1.3	0.00784	0.0245	0.6175	3.4716
1.5	0.00698	0.0190	0.6775	3.3548
1.7	0.01103	0.0543	0.6991	3.1348
1.9	0.04064	0.2292	0.7061	2.8880
2.1	0.13089	0.7245	0.7298	2.6699
2.3	0.34307	1.8493	0.8731	2.8077
2.6	1.0818	5.6084	1.7457	8.0680
	[7]		[18]	
1.1	–	–	–	–
1.3	–	–	–	–
1.5	–	–	–	1.41
1.7	0.1701	0.4043	–	1.01
1.9	–	–	–	0.930
2.1	0.2048	0.8636	–	0.834
2.3	–	–	–	0.717
2.6	1.2995	6.6943	–	0.576
	[16]		[19]	
1.1	–	–	–	–
1.3	–	–	–	–
1.5	–	–	–	–
1.7	0.0175	0.0959	0.2083	0.4524
1.9	–	–	–	–
2.1	0.2117	1.1476	–	–
2.3	–	–	–	–
2.6	1.5924	8.0680	1.5776	8.0680

It is well-known that the wave will be broken at time $t_b = \frac{-1}{\min_x u_0'(x)}$ and discontinuous solutions will occur if the Burgers equation, in the case of $v = 0$, is solved with the smooth initial condition $u_0(x)$ for which $u_0'(x)$ is somewhere negative [42]. Adding the viscosity-dispersion term vu_{xx} with $v \neq 0$ viscosity values to the inviscid form, as given in Eq. (1), provides to suppress the wave-breaking and, therefore, prevent to occur discontinuous solutions. This is reasonable because of that the dispersion gives rise to the wave to spread and this behaves against the steep gradient of the nonlinear term. Therefore, the smooth solutions of viscous Burgers equation approaches a shock wave as $v \to 0$ [43]. It can be said that the wave behaviors shown in Fig. 3 are in accord with this physical interpretation. It is seen from the figure that the wave at initial time for $v = 0.01$ is smooth and the steep behavior decreases with the effect of dispersion as the time goes on. For smaller viscosity value, $v = 0.0005$, it develops into a sharp shock and this sharpness continues as time progresses. It can also be observed from Fig. 3 that the shock wave propagates faster when viscosity v is taken to be smaller.

Absolute errors are plotted in Fig. 4 for all viscosity used here at different times t. It is clear for all v values that absolute errors reach their maximum values at the midpoints of the problem domain near the initial time $t = 1$ while they become maximum near the right hand boundary at corresponding latest times t since the right end of the shock crashes the right hand boundary. We also show the variations of L_2, L_∞ error norms with respect to time for all values of viscosity in Fig. 5.

Comparisons of the error norms at various times for $\upsilon = 0.005$ for example 1. TABLE VI

Present method $\Delta t = 0.001$, $h = 0.005$			[14] $\Delta t = 0.01$, $h = 0.02$		[16] $\Delta t = 0.01$, $h = 0.005$		[19] $\Delta t = 0.01$, $h = 0.005$	
t	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$L_2 \times 10^3$	$L_\infty \times 10^3$
1.2	0.00030	0.0014	0.58143	3.1354	–	–	–	–
1.4	0.00028	0.0012	0.64676	2.9314	–	–	–	–
1.5	0.00026	0.0010	–	–	–	–	–	–
1.6	0.00023	0.0009	0.67610	2.6923	–	–	–	–
1.7	0.00021	0.0008	–	–	0.01053	0.03771	0.02210	0.07702
1.8	0.00019	0.0007	0.68762	2.4719	–	–	–	–
2.0	0.00019	0.0010	0.68954	2.3767	–	–	–	–
2.2	0.00105	0.0089	0.68602	2.2733	–	–	–	–
2.4	0.00641	0.0527	0.67943	2.1676	0.00952	0.06464	–	–
2.5	0.01421	0.1150	–	–	–	–	–	–
2.8	0.10925	0.8435	0.68307	1.9690	–	–	–	–
3	0.33600	2.5178	0.82951	2.9573	–	–	–	–
3.1	0.55560	4.1038	–	–	0.65307	4.7906	0.6312	4.7906
3.2	0.88640	6.4567	1.4856	7.4915	–	–	–	–
	[4]		[7] $\Delta t = 0.01$, $h = 0.005$		[9] $\Delta t = 0.01$, $h = 0.005$		[10] $\Delta t = 0.001$, $h = 0.05$	
1.7	2.107	8.099	0.01705	0.0619	0.02681	0.09174	0.007	0.034
2.4	3.345	11.641	0.01252	0.0588	–	–	0.007	0.025
2.5	–	–	–	–	0.03135	0.11515	–	–
3.1	4.820	15.867	0.60199	4.4347	–	–	0.009	0.077



(a) $\upsilon = 0.01$

(c) $\upsilon = 0.0015$

(b) $\upsilon = 0.005$

(d) $\upsilon = 0.0005$

Fig. 4. Absolute errors for example 1 for various υ parameters.

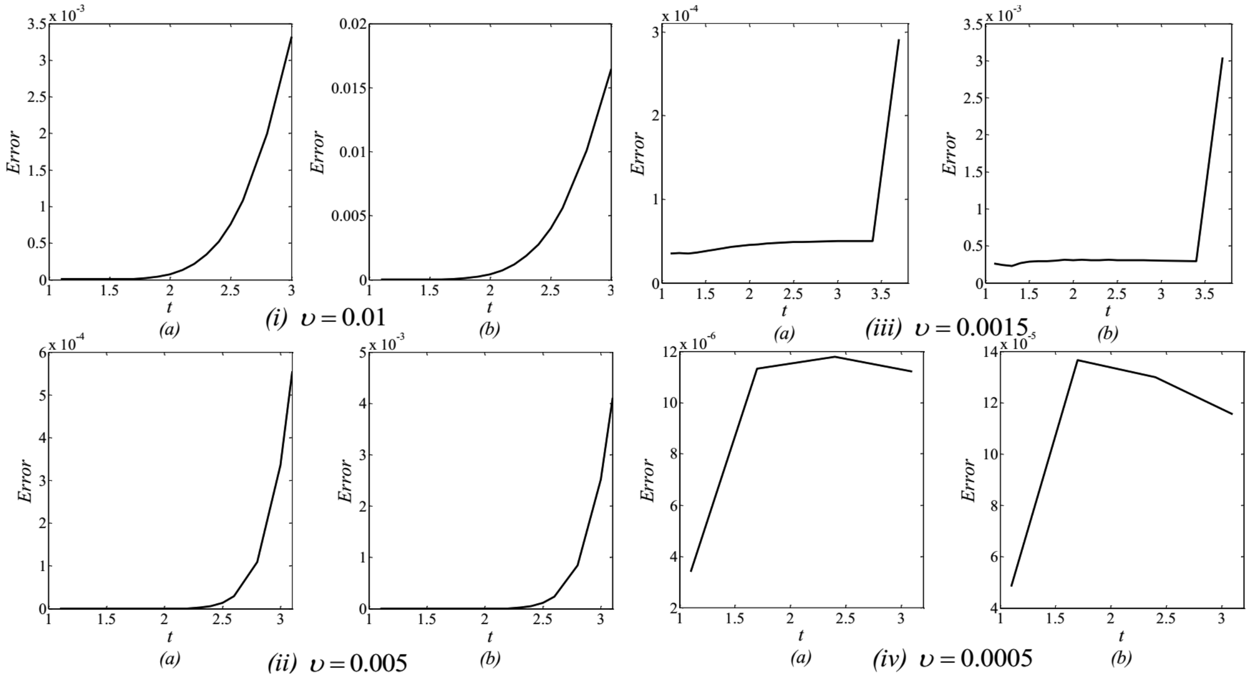


Fig. 5. (a) L_2 and (b) L_∞ errors for example 1 using various υ parameters.

Example 2. We consider the modified Burgers equation which is of the form with $\mu = 2$:

$$u_t + u^2 u_x - \upsilon u_{xx} = 0.$$

The initial and boundary conditions are taken as

$$u(x,1) = x\frac{1}{1+\frac{1}{t_0}\exp(\frac{x^2}{4\upsilon})},\ u(0,t) = 0,$$

and

$$u(1,t) = 0,$$

written from the following analytical solution [44]:

$$u(x,t) = \frac{x}{t}\frac{1}{1+\frac{\sqrt{t}}{t_0}\exp(\frac{x^2}{4\upsilon t})},$$

for $0 \le x \le 1,\ t \ge 1,\ t_0 \in (0,1)$.

TABLE VII

Some error norms at different times using $h = 0.0125$, $\Delta t = 0.001$ and various υ parameters for example 2

t	$L_2 \times 10^4$	$L_\infty \times 10^3$	$L_2 \times 10^4$	$L_\infty \times 10^3$
	$\upsilon = 0.01$		$\upsilon = 0.001$	
2	3.79363	0.81575	0.681639	0.258274
3	3.44789	0.70956	0.617946	0.226454
4	3.17070	0.60496	0.564976	0.192488
5	3.06678	0.52647	0.525541	0.167045
6	3.22355	0.49063	0.493984	0.147930
7	3.63238	0.71291	0.467485	0.132945
8	4.18502	0.91727	0.444596	0.120741
9	4.77503	1.09189	0.424471	0.110673
10	5.33834	1.23368	0.406555	0.102599
11	5.84513	1.34419	0.390459	0.095445
12	6.28531	1.42703	0.375888	0.089435
	$\upsilon = 0.005$		$\upsilon = 0.0001$	
2	2.26539	0.579128	0.111533	0.0772467
3	2.05702	0.503712	0.116994	0.0755334
4	1.88170	0.429411	0.116509	0.0640031
5	1.75080	0.372495	0.111137	0.0580037
6	1.64604	0.329594	0.104917	0.0532005
7	1.55893	0.296190	0.099020	0.0479225
8	1.48704	0.269349	0.093711	0.0429854
9	1.43197	0.247128	0.088995	0.0396847
10	1.39755	0.228859	0.084805	0.0371205
11	1.38760	0.213391	0.081069	0.0346197
12	1.40390	0.200569	0.077717	0.0322778

The case with $t_0 = 0.5$ is considered in the numerical treatments. During the calculation process, we take various values of viscosity υ from 0.01 to 0.0001 in order to compare the obtained results with the literature. The accuracy of the suggested scheme is examined by computing L_2 and L_∞ error norms, and the obtained results up to time $t = 12$ are presented in Table VII. It can be seen from the table that the errors are small enough and acceptable for all values of viscosity used. It is also noted that the accuracy depends on the value of viscosity, and we get better accuracy when the viscosity takes smaller as is seen from Table VII. The obtained L_2 and L_∞ errors are compared with some relevant works in the literature for various values of viscosity υ at times $t = 2, 4, 6, 8, 10$ in Tables VIII and IX. It is deduced from the comparisons of the results listed in these tables that the proposed scheme produces the equivalent accuracies to the literature results for all time levels. The suggested approximation is relatively more advantageous than the others because of producing the equivalent accurate results by using fewer grids. The behavior of the wave throughout the given domain is depicted in Fig. 6 for some values of viscosity $\upsilon = 0.01$, 0.005, 0.001 and 0.0001 from initial time $t = 1$ to $t = 10$. We can say from the figure that the curve of initial wave decays as the time progresses, and the decay gets fast as the viscosity υ decreases. Also note that amplitude of the wave decreases as the viscosity υ gets smaller. For above viscosity values, the absolute errors throughout the domain at $t = 2$ and $t = 10$, and L_2, L_∞ errors versus the time are drawn in Figs. 7 and 8, respectively.

It is observed from Fig. 7 that the absolute error for $\upsilon = 0.01$ reaches its maximum value on the first half of the domain at time $t = 2$ and close by the right hand boundary at time $t = 10$. The propagation of the wave to the rightward, as the time progresses, confirms that the absolute error increases towards the right hand boundary of the domain. For the smaller viscosity value, such as $\upsilon = 0.0001$, the maximum error occurs near the left hand boundary where the corresponding wave arises at both $t = 2$ and $t = 10$. Furthermore, it is seen in Fig. 8 that as the time advances, L_2 and L_∞ errors for $\upsilon = 0.01$ increase while they decrease for the smaller υ values.

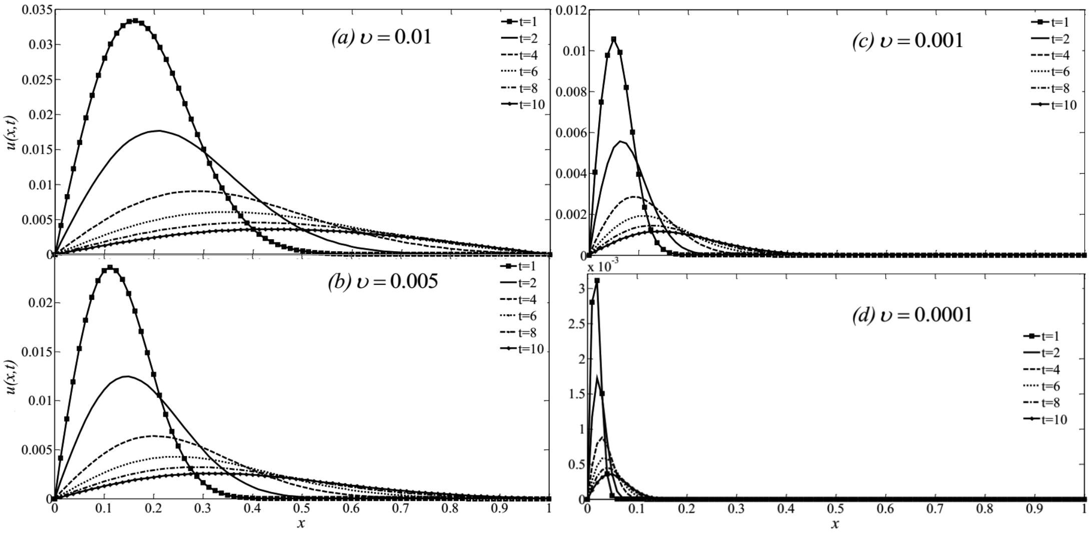
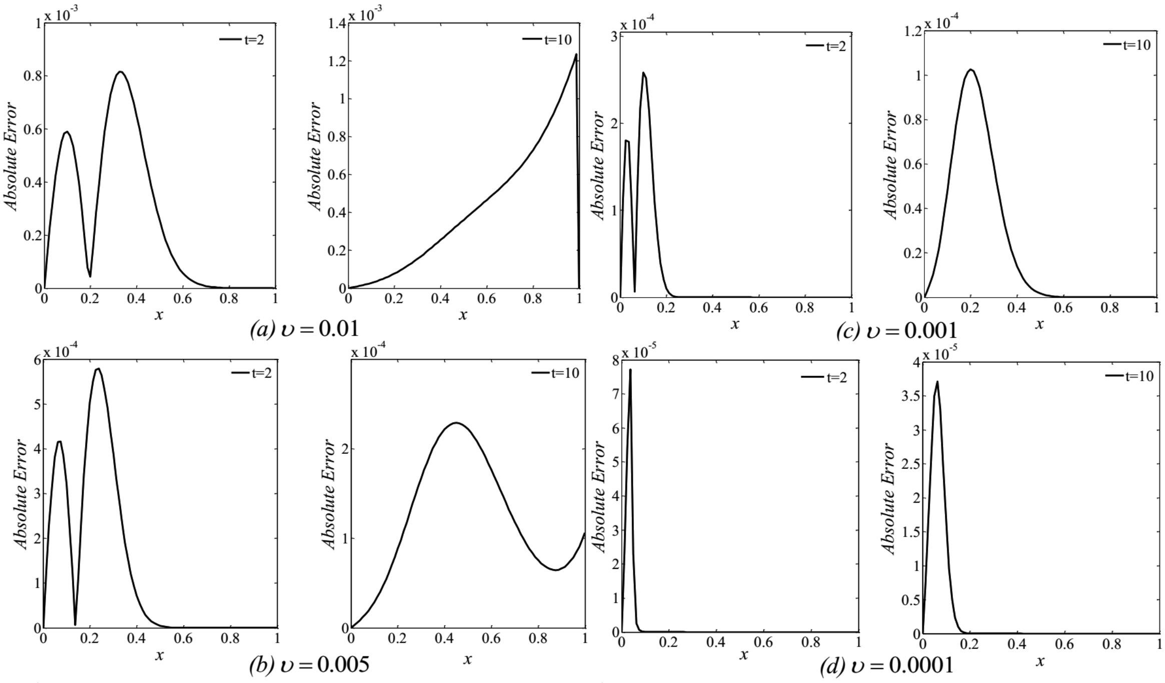


Fig. 6. The numerical solutions of example 2 at various times using the parameters $h = 0.0125$ and $\Delta t = 0.001$.



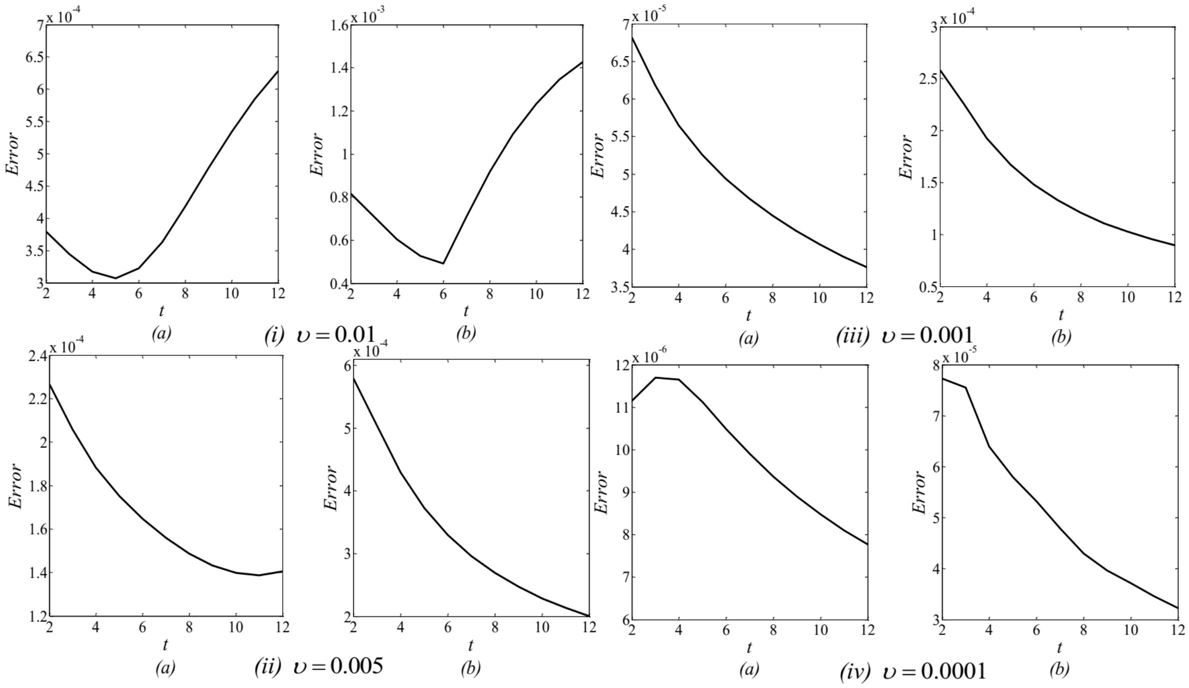
(a) $\upsilon = 0.01$

(c) $\upsilon = 0.001$

(b) $\upsilon = 0.005$

(d) $\upsilon = 0.0001$

Fig. 7. Absolute errors for example 2 using $h = 0.0125$ and $\Delta t = 0.001$.



(i) $\upsilon = 0.01$ (iii) $\upsilon = 0.001$

(ii) $\upsilon = 0.005$ (iv) $\upsilon = 0.0001$

Fig. 8. (a) L_2 and (b) L_∞ errors for example 2 using $h = 0.0125$ and $\Delta t = 0.001$.

Comparisons of the error norms at different times for various values of υ for example 2. TABLE VIII

t	Method	$\upsilon = 0.01$		$\upsilon = 0.005$		$\upsilon = 0.001$	
		$L_2 \times 10^4$	$L_\infty \times 10^3$	$L_2 \times 10^4$	$L_\infty \times 10^4$	$L_2 \times 10^4$	$L_\infty \times 10^4$
2	present ($\Delta t = 0.001$, $h = 0.0125$)	3.794	0.8158	2.265	5.791	0.6816	2.583
	[14]* ($\Delta t = 0.01$, $h = 0.02$)	7.904	1.703	–	–	1.836	8.185
	[15]** ($\Delta t = 0.01$, $h = 0.005$)	5.231	1.217	2.579	7.227	0.6704	2.797
	[16] ($\Delta t = 0.01$, $h = 0.005$)	3.793	0.8168	2.265	5.800	0.6811	2.609
	[18] ($\Delta t = 0.01$)	–	0.758	–	–	–	2.73
	[19]** ($\Delta t = 10^{-5}$, $h = 0.005$)	3.849	0.8293	2.289	5.862	0.6843	2.623
	[20] ($\Delta t = 10^{-5}$, $h = 0.005$)	–	–	2.271	5.809	–	–
	[21] ($\Delta t = 10^{-5}$, $h = 0.001$)	3.792	0.8167	2.265	5.803	0.6817	2.611
	[22] ($\Delta t = 0.0001$, $h = 0.005$)	3.832	0.8164	2.285	5.804	0.6184	2.628
	[24] ($\Delta t = 0.01$, $h = 0.005$)	3.755	0.8177	2.233	5.808	0.6607	2.619
	[25] ($\Delta t = 0.01$, $h = 0.005$)	3.789	0.8163	2.260	5.792	0.6729	2.591
4	present ($\Delta t = 0.001$, $h = 0.0125$)	3.171	0.6050	1.882	4.294	0.5650	1.925
	[14] ($\Delta t = 0.01$, $h = 0.02$)	5.577	0.9965	–	–	1.144	3.564
	[15] ($\Delta t = 0.01$, $h = 0.005$)	5.163	0.9314	2.528	5.545	0.6670	2.186
	[16] ($\Delta t = 0.01$, $h = 0.005$)	3.172	0.6054	1.882	4.294	0.5652	1.929
	[18] ($\Delta t = 0.01$)	–	0.564	–	–	–	1.57
	[20] ($\Delta t = 10^{-5}$, $h = 0.005$)	–	–	1.883	4.297	–	–
	[21] ($\Delta t = 10^{-5}$, $h = 0.001$)	3.155	0.6056	1.882	4.295	0.5652	1.929
	[22] ($\Delta t = 0.0001$, $h = 0.005$)	3.144	0.6030	1.878	4.285	0.5662	1.933
	[24] ($\Delta t = 0.01$, $h = 0.005$)	3.168	0.6081	1.893	4.321	0.5740	1.958
	[25] ($\Delta t = 0.01$, $h = 0.005$)	3.172	0.6052	1.881	4.291	0.5635	1.922
6	present ($\Delta t = 0.001$, $h = 0.0125$)	3.224	0.4906	1.646	3.296	0.4940	1.479
	[14] ($\Delta t = 0.01$, $h = 0.02$)	5.167	0.7611	–	–	0.8142	2.135
	[15] ($\Delta t = 0.01$, $h = 0.005$)	4.902	0.7225	2.257	4.308	0.6046	1.718
	[16] ($\Delta t = 0.01$, $h = 0.005$)	3.260	0.5258	1.646	3.299	0.4942	1.481
	[18] ($\Delta t = 0.01$)	–	0.459	–	–	–	1.39
	[20] ($\Delta t = 10^{-5}$, $h = 0.005$)	–	–	1.646	3.300	–	–
	[21] ($\Delta t = 10^{-5}$, $h = 0.001$)	2.731	0.4650	1.646	3.299	0.4942	1.481
	[22] ($\Delta t = 0.0001$, $h = 0.005$)	2.710	0.4622	1.638	3.287	0.4932	1.481
	[24] ($\Delta t = 0.01$, $h = 0.005$)	2.749	0.4675	1.664	3.326	0.5063	1.509
	[25] ($\Delta t = 0.01$, $h = 0.005$)	3.260	0.5258	1.646	3.298	0.4939	1.478

Authors used: $^*h = 0.005$ for $\upsilon = 0.001$; $^{**}\Delta t = 0.001$ for $\upsilon = 0.005$.

TABLE VIII (Cont.)

t	Method	$\upsilon = 0.01$		$\upsilon = 0.005$		$\upsilon = 0.001$	
		$L_2 \times 10^4$	$L_\infty \times 10^3$	$L_2 \times 10^4$	$L_\infty \times 10^4$	$L_2 \times 10^4$	$L_\infty \times 10^4$
8	present ($\Delta t = 0.001$, $h = 0.0125$)	4.185	0.9173	1.487	2.694	0.4446	1.207
	[14] ($\Delta t = 0.01$, $h = 0.02$)	6.428	1.358	–	–	0.6484	1.683
	[15] ($\Delta t = 0.01$, $h = 0.005$)	5.478	0.9634	2.030	3.530	0.5480	1.419
	[18] ($\Delta t = 0.01$)	–	0.366	–	–	–	1.10
	[24] ($\Delta t = 0.01$, $h = 0.005$)	2.347	0.3762	1.498	2.717	0.4557	1.233
	[25] ($\Delta t = 0.01$, $h = 0.005$)	4.280	0.9634	1.487	2.693	0.4449	1.208
10	present ($\Delta t = 0.001$, $h = 0.0125$)	5.338	1.234	1.398	2.289	0.4066	1.026
	[14] ($\Delta t = 0.01$, $h = 0.02$)	8.003	1.802	–	–	0.5512	1.394
	[15] ($\Delta t = 0.01$, $h = 0.005$)	6.401	1.281	1.874	3.001	0.5011	1.213
	[16] ($\Delta t = 0.01$, $h = 0.005$)	5.470	1.281	1.396	2.289	0.4067	1.026
	[18] ($\Delta t = 0.01$)	–	0.300	–	–	–	0.936
	[19] ($h = 0.005\Delta t = 0.01$)	5.483	1.281	1.404	2.302	0.4080	1.030
	[20] ($\Delta t = 10^{-5}$, $h = 0.005$)	–	–	1.352	2.287	–	–
	[21] ($\Delta t = 10^{-5}$, $h = 0.001$)	1.934	0.3018	1.352	2.287	0.4067	1.026
	[22] ($\Delta t = 0.0001$, $h = 0.005$)	1.909	0.2991	1.344	2.276	0.4053	1.025
	[24] ($\Delta t = 0.01$, $h = 0.005$)	1.939	0.3029	1.366	2.307	0.4160	1.047
	[25] ($\Delta t = 0.01$, $h = 0.005$)	5.470	1.281	1.396	2.288	0.4069	1.026

TABLE IX

Comparisons of the error norms at various times for several υ values for example 2.

t	$L_2 \times 10^4$	$L_\infty \times 10^4$	$L_2 \times 10^4$	$L_\infty \times 10^4$
	Present method ($h = 0.0125$, $\Delta t = 0.001$)		[20] ($h = 0.005$, $\Delta t = 10^{-5}$)	
	$\upsilon = 0.002$			
2	1.143813	3.678070	1.151	3.6923
4	0.949068	2.722907	0.9507	2.7279
6	0.829942	2.089229	0.8302	2.0931
10	0.683114	1.449166	0.6830	1.4501
	Present method ($h = 0.01$, $\Delta t = 0.001$)		[21] ($h = 0.001$, $\Delta t = 10^{-5}$)	
	$\upsilon = 0.0004$			
2	0.343409	1.619166	0.3437	1.6550
4	0.284483	1.219831	0.2847	1.2217
6	0.248707	0.931724	0.2488	0.9377
10	0.204669	0.648530	0.2048	0.6496
	Present method ($h = 0.0125$, $\Delta t = 0.001$)		[24] ($h = 0.005$, $\Delta t = 0.01$)	
	$\upsilon = 0.0001$			
2	0.111533	0.772467	0.112962	0.818695
4	0.116509	0.640031	0.107709	0.640708
6	0.104917	0.532005	0.095058	0.498194
10	0.084805	0.371205	0.077383	0.344082

Example 3. As the third problem, we take the following modified Burgers equation with $\mu = 3$:

$$u_t + u^3 u_x - \upsilon u_{xx} = 0.$$

The initial and boundary conditions are used as

$$u(x,0) = A \sin\left(\frac{\pi x}{l}\right), \quad u(0,t) = 0, \quad u(l,t) = 0,$$

where $A = 1$ and $l = \pi$. The problem has an asymptotic solution as follows [45]:

$$u(x,t) = f_0(x,t)\exp(-kt) + f_1(x,t)\exp(-4kt)$$
$$+ f_2(x,t)\exp(-7kt) + \ldots,$$

where

$$k = \upsilon\pi^2/l^2, \quad f_0(x,t) = A_1 \sin\left(\frac{\pi x}{l}\right),$$
$$f_1(x,t) = B_1 t \sin\left(\frac{2\pi x}{l}\right) + B_2 \sin\left(\frac{4\pi x}{l}\right),$$
$$B_1 = -A_1^4 \pi/4l, \quad B_2 = A_1^4 l/96\upsilon\pi, \quad A_1 = 0.365366,$$
$$f_2(x,t) = g_3(t)\sin\left(\frac{\pi x}{l}\right) + g_4(t)\sin\left(\frac{3\pi x}{l}\right)$$
$$+ g_5(t)\sin\left(\frac{5\pi x}{l}\right) + g_6(t)\sin\left(\frac{7\pi x}{l}\right),$$
$$g_3(t) = -l^2\left(D_1 t + E_1 + \frac{l^2 D_1}{6\upsilon\pi^2}\right)\Big/6\upsilon\pi^2$$
$$g_4(t) = l^2\left(D_2 t + E_2 - \frac{l^2 D_2}{2\upsilon\pi^2}\right)\Big/2\upsilon\pi^2,$$
$$g_5(t) = l^2\left(D_3 t + E_3 - \frac{l^2 D_3}{18\upsilon\pi^2}\right)\Big/18\upsilon\pi^2,$$
$$g_6(t) = l^2 E_4/42\upsilon\pi^2,$$
$$D_1 = A_1^3 B_1 \pi/4l, \quad E_1 = -A_1^3 B_2 \pi/8l,$$
$$D_2 = -9A_1^3 B_1 \pi/8l, \quad E_2 = 9A_1^3 B_2 \pi/8l,$$
$$D_3 = 5A_1^3 B_1 \pi/8l, \quad E_3 = -15A_1^3 B_2 \pi/8l,$$
$$E_4 = 7A_1^3 B_2 \pi/8l.$$

In order to compare the results with the previous works,

TABLE X

Comparisons of the error norms at various times for $v = 0.005$ for example 3.

t	Error norm $\times 10^3$	Present $h = 0.02$, $\Delta t = 0.01$	[17] $h = 0.01$, $\Delta t = 0.01$	[21] $h = 0.005$, $\Delta t = 10^{-5}$	[22] $h = 0.005$, $\Delta t = 0.0001$	[24] $h = 0.005$, $\Delta t = 0.01$
150	L_2	6.127935	3.227	6.1258	6.1237	6.128090
	L_∞	6.845355	5.172	6.8400	6.8252	6.845380
200	L_2	2.229242	0.991	2.2273	2.2181	2.229470
	L_∞	2.045451	1.671	2.0416	2.0231	2.045900
250	L_2	0.913681	0.5031	0.91238	0.90306	0.914486
	L_∞	0.836173	1.400	0.83351	0.81642	0.839436
300	L_2	0.4145668	0.5939	0.41341	0.40256	0.416609
	L_∞	0.3974457	1.452	0.39559	0.38132	0.399913
350	L_2	0.2319149	0.6940	0.23070	0.21750	0.234546
	L_∞	0.2199448	1.488	0.21860	0.20661	0.221936
400	L_2	0.1628484	0.7567	0.16168	0.14767	0.165391
	L_∞	0.1426408	1.513	0.14160	0.13131	0.144269
450	L_2	0.1293832	0.7990	0.12836	0.11516	0.131525
	L_∞	0.1044738	1.531	0.10361	0.09461	0.105834

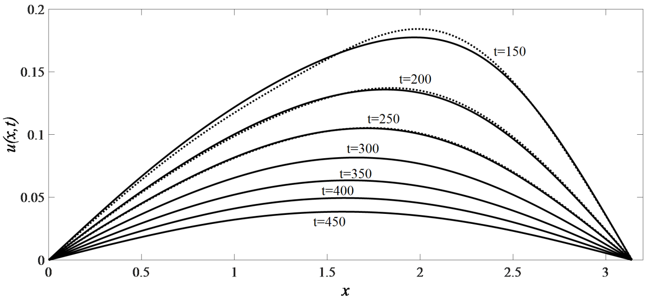


Fig. 9. The numerical and asymptotic solutions of example 3 at various times using the parameters $h = 0.02$, $\Delta t = 0.01$ and $v = 0.005$ (dots — asymptotic, solid line — numerical).

the value of viscosity v is taken to be 0.005, and the space and time steps are preferred as 0.02 and 0.01, respectively, for the experiments. The process started from the time $t = 150$ then runs up to the time $t = 450$. The obtained results in terms of L_2, L_∞ error norms are listed in Table X with the results of some previous studies.

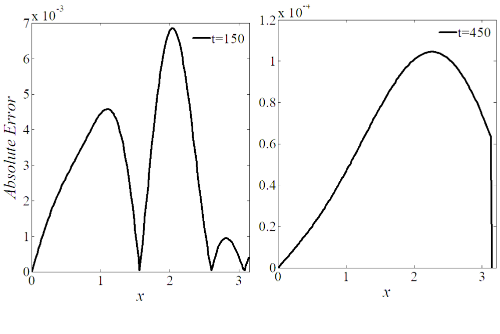


Fig. 10. Absolute errors for example 3 using $v = 0.005$, $h = 0.02$ and $\Delta t = 0.01$.

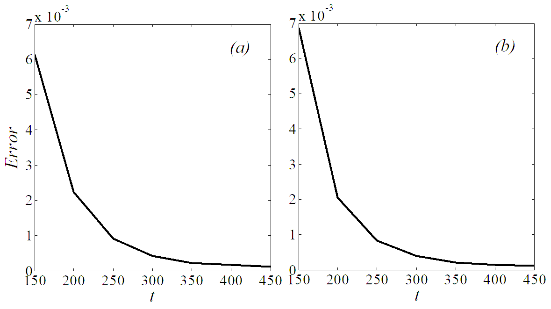


Fig. 11. (a) L_2 and (b) L_∞ errors for example 3 using $v = 0.005$, $h = 0.02$ and $\Delta t = 0.01$.

It is seen from the table that the errors decrease as the time progresses, and produced results are in good agreement with the literature. The produced results are especially close to the results given by the studies [21, 22] using some finite difference based methods, and it is noted that our results are produced by using fewer number of grids as seen in Table X.

Behavior of the wave in terms of the numerical and asymptotic solutions are shown in Fig. 9 for different times. It can be observed from the figure that the numerical behavior of wave appears a little different from the asymptotic behavior at time $t = 150$ while there are no visible differences between them as the time goes on, especially for $t \geq 250$. Also, the changes of absolute, L_2 and L_∞ errors are plotted in Figs. 10 and 11, respectively.

It is reasonable that the absolute error at time $t = 150$ changes frequently throughout the domain and gets the maximum value around the midpoints since the numerical and asymptotic behaviors are exactly not overlapped. Similarly, the maximum value of absolute error is around the midpoints for $t = 450$ but it increases smoothly up to midpoints in contrast to the changes of absolute error at time $t = 150$.

4. Conclusion

Burgers equation is a simple form of the Navier–Stokes equations and describes several physical phenomena such as the shock wave behaviors, continuous stochastic processes, sound waves in a viscous medium etc. Therefore, the derivation of accurate, applicable and efficient methods for simulation of these problems is both necessary and important for technological and scientific developments in many disciplines. In this study, a hybrid scheme including sixth order finite difference and seventh order weighted essentially non-oscillatory finite difference schemes is introduced and applied to carry out effective numerical simulations of shock wave propagating represented by the Burgers equation and its modified form. The efficiency, accuracy, and applicability of the proposed technique are examined by some test problems for several values of viscosity, and the performance of the method is measured by some error norms. The obtained results, even for small viscosity values caused to occur shock behavior, are in good agreement with the literature. Moreover, some produced results are considerably better than some of the given results in literature in terms of accuracy. Therefore, the suggested scheme is a good alternative in terms of both accurate results produced by it and relatively is able to come to the forefront due to its low cost and easy implement for scientists.

Acknowledgments

This research was supported by Suleyman Demirel University through Scientific Research Project Program (3539-D1-13).

References

[1] H. Bateman, *Mon. Weather Rev.* **43**, 163 (1915).
[2] J.M. Burgers, *Verh. Kon. Ned. Akad. Wet.* **17**, 1 (1939).
[3] J.M. Burgers, *Adv. Appl. Mech.* **1**, 171 (1948).
[4] A. Dogan, *Appl. Math. Comput.* **157**, 331 (2004).
[5] S. Kutluay, A. Esen, I. Dag, *J. Comput. Appl. Math.* **167**, 21 (2004).
[6] I. Dag, D. Irk, B. Saka, *Appl. Math. Comput.* **163**, 199 (2005).
[7] B. Saka, I. Dag, *Chaos Soliton Fract.* **32**, 1125 (2007).
[8] R.C. Mittal, R.K. Jain, *Appl. Math. Comput.* **218**, 7839 (2012).
[9] S.S. Xie, S. Heo, S. Kim, G. Woo, S. Yi, *J. Comput. Appl. Math.* **214**, 417 (2008).
[10] A. Korkmaz, I. Dag, *Eng. Comput.* **28**, 654 (2011).
[11] I.A. Hassanien, A.A. Salama, H.A. Hosham, *Appl. Math. Comput.* **170**, 781 (2005).
[12] W. Liao, *Appl. Math. Comput.* **206**, 755 (2008).
[13] M. Sari, G. Gürarslan, *Appl. Math. Comput.* **208**, 475 (2009).
[14] M.A. Ramadan, T.S. El-Danaf, F.E.I. Abd Alaal, *Chaos Soliton Fract.* **26**, 795 (2005).
[15] M.A. Ramadan, T.S. El-Danaf, *Math. Comput. Simulat.* **70**, 90 (2005).
[16] B. Saka, I. Dag, *J. Franklin Inst.* **345**, 328 (2008).
[17] Y. Duan, R. Liu, Y. Jiang, *Appl. Math. Comput.* **202**, 489 (2008).
[18] R.S. Temsah, *Commun. Nonlinear Sci.* **14**, 760 (2009).
[19] D. Irk, *Kybernetes* **38**, 1599 (2009).
[20] A.G. Bratsos, in: *Proc. HERCMA 2009, 9th Hellenic-European Conf. on Computer Mathematics and Its Applications*, Athens 2009.
[21] A.G. Bratsos, *Comput. Math Appl.* **60**, 1393 (2010).
[22] A.G. Bratsos, L.A. Petrakis, *Int. J. Numer. Meth. Biomed. Eng.* **27**, 232 (2011).
[23] V. Gupta, M.K. Kadalbajoo, *Neural Parallel Sci. Comput.* **18**, 167 (2010).
[24] T. Roshan, K.S. Bhamra, *Appl. Math. Comput.* **218**, 3673 (2011).
[25] S. Kutluay, Y. Ucar, N.M. Yagmurlu, *Bull. Malaysian Math. Soc.* **39**, 1603 (2016).
[26] Y. Shen, G. Zha, in: *46th AIAA Aerospace Sciences Meeting and Exhibit*, Reno (Nevada, USA), AIAA 2008-0757, 2008.
[27] Y. Shen, G. Zha, in: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, Orlando (Florida, USA), AIAA 2010-1451, 2010.
[28] D.S. Balsara, C.W. Shu, *J. Comput. Phys.* **160**, 405 (2000).
[29] Y.H. Zahran, M.M. Babatin, *Appl. Math. Comput.* **219**, 8198 (2013).
[30] G.S. Jiang, C.W. Shu, *J. Comput. Phys.* **126**, 202 (1996).
[31] Z.J. Wang, R.F. Chen, *J. Comput. Phys.* **174**, 381 (2001).
[32] D. Ponziani, S. Prizzoli, F. Grasso, *Int. J. Numer. Meth. Fl.* **42**, 953 (2003).
[33] S. Pirozzoli, *J. Comput. Phys.* **178**, 81 (2002).
[34] D. Kim, J.H. Kwon, *J. Comput. Phys.* **210**, 554 (2005).
[35] Y.Q. Shen, G.W. Yang, *Int. J. Numer. Meth. Fl.* **53**, 531 (2007).
[36] M. Sari, G. Gürarslan, A. Zeytinoglu, *Numer. Meth. Part. D E* **27**, 1313 (2011).
[37] A. Zeytinoglu, M.Sc. Thesis, Suleyman Demirel University, Turkey 2010.
[38] X.D. Liu, S. Osher, T. Chan, *J. Comput. Phys.* **115**, 200 (1994).
[39] P. Xie, Ph.D. Thesis, The University of Texas, 2007.
[40] R.H. Pletcher, J.C. Tannehill, D.A. Anderson, *Computational Fluid Mechanics and Fluid Transfer*, Taylor and Francis, 2013.
[41] H. Nguyen, J. Reynen, in: *Numerical Methods for Nonlinear Problems*, Eds. C. Taylor, E. Hinton, D.R.J. Owen, Pineridge, Swansea 1984, p. 718.
[42] R.J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhauser Verlag, 1992.
[43] M. Landajuela, *Burgers Equation*, BCAM Internship, 2011.
[44] S.E. Harris, *Eur. J. Appl. Math.* **7**, 201 (1996).
[45] P.L. Sachdev, C.S. Rao, B.O. Enflo, *Stud. Appl. Math.* **114**, 307 (2005).