Proceedings of the Institute of Mathematics and Mechanics,
National Academy of Sciences of Azerbaijan
Volume 45, Number 2, 2019, Pages 295–310
https://doi.org/10.29228/proc.10

A HIGHER ORDER COMPACT SCHEME FOR THE NONLINEAR ADVECTION DIFFUSION PROCESSES

MURAT SARI, SUFII H. MUSSA, AND HUSEYIN TUNC

Abstract. This paper presents a highly accurate finite difference based scheme through the Padé approximation in analyzing the behaviour of the nonlinear advection-diffusion processes governed by the unsteady Burgers equation. It has then been proved that the present method is unconditionally stable based on the von Neumann stability analysis. The proposed approach has been shown to be capable of solving the model equation effectively. Two challenging examples have been taken to illustrate the physical behaviour of the model in detail. The computed results have been seen to be highly accurate and be oscillation free even if advection dominated cases are considered.

1. Introduction

The numerical simulation of nonlinear partial differential equations is of great importance scientifically in various topics for so many decades, especially in the fluid mechanics and heat transfer [1]. As differential equations generally describe behavior of events that occur in space and time, many physical problems are represented by partial differential equations that mostly can only be solved numerically [2]. One of the most frequent and popular partial differential equations in applied mathematics is the Burgers equation. Many analytical and numerical methods for the analysis of the Burgers equation have been encountered in literature [4–10]. In this work we pay our attention to accurately capture the behaviour of nonlinear advection-diffusion processes represented by the Burgers equation

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}=v\frac{\partial^2 u}{\partial x^2},\quad x\in(a,b)\,,\quad t\in(0,T] \tag{1.1}$$

with initial condition

$$u(x,0)=f(x),\quad x\in(a,b)$$

and boundary conditions

$$u(a,t)=0,\quad \text{and}\quad u(b,t)=0$$

2010 *Mathematics Subject Classification.* 42B25, 31C15.

Key words and phrases. Nonlinear advection diffusion process, Compact finite difference method, Nonlinear modelling, Padé approximation, Burgers equation.

where $v > 0$ is the kinematic viscosity constant of the fluid. Bateman [3] illustrated the steady state solution of (1.1) and for some time later, modeling of turbulence were described using this equation by Burgers [7]. Separately, Hopf [8] and Cole [9] explored a transformation technique by which equation (1.1) can be converted into linear heat conduction equation that can be easily solved.

Many numerical methods for solving the Burgers equation have been intensively produced for so many years. In doing that, several approximation schemes like finite difference methods (FDM), finite element methods (FEM) have been studied in details. To obtain numerical solution of equation (1.1), Gulsu [10] used restrictive Padé approximation of classical implicit finite difference schemes. Xie et al [11] solved the Burgers equation by reproducing kernel function using the Hopf-Cole transformation. Hassanien et al [12] introduced a fourth-order finite difference method for solving the Burgers equation with the use of local approximation with two-level three-point finite difference methods of order four. Gulsu and Ozis [13] developed a restrictive Taylor approximation scheme for the classical explicit finite difference method to solve the Burgers equation numerically. Zhang and Wang [14] applied two approaches, a predictor-corrector procedure known as MacCormack method in time derivative and fourth-order compact finite difference scheme in space derivative to compute the solutions of the model equation (1.1). Kutluay et al [15] illustrated explicit finite difference techniques to capture numerical behaviour of the model equation.

A fourth order compact finite difference scheme was proposed by Liao [16] using the Padé reconstruction of difference operators. Sari and Gurarslan [17] presented a sixth-order compact finite difference scheme for the spatial derivative and strong stability third preserving Runge-Kutta (SSPRK3) method for the time integration. Dogan [18] solved the Burgers equation by the Galerkin FEM in space and the Crank-Nicolson approach in time. Aksan [19] applied a variant of quadratic B-spline FEM to solve the equation. Zhu and Wang [20] solved the equation with the use of the cubic B-spline quasi-interpolation in space and a low order forward difference in time. A modified local discontinuous Galerkin method was introduced by Rong-Pei et al [21] to analyse behaviour of the Burgers equation with less amount of oscillations. Sari and Tunc [22] described an optimization approach in analyzing the Burgers equation based on a quadratic B-spline FEM and α-family of the time approximation.

To effectively capture the response of the nonlinear advection–diffusion processes, a new form of a high–order compact difference has been presented. We have combined fourth order central difference formulation and Padé approximation of the second order difference operator in discretization of spatial derivative operator. Fourth order form of the second order difference operator is constructed for the boundary points using the Padé approximation. Fourth order forward and backward difference formulae have been applied to handle the Neumann boundary conditions come out after the Hopf-Cole transformation. Thus a completely fourth order scheme has been produced in dealing with the model equation. The approach here makes the method of solution to the model equation entirely different approach what Liao [16] proposed. The scheme has then been tested for stability conditions under the von Neumann stability analysis and thus it has been shown to be unconditionally stable.

2. Method of Solution

Lets us consider the Burgers equation and apply the Hopf-Cole transformation [8, 9]

$$u(x,t) = -2v\frac{w_x(x,t)}{w(x,t)}. \tag{2.1}$$

With the use of transformation (2.1), the Burgers equation is converted to the following diffusion equation

$$\frac{\partial w}{\partial t} = v\frac{\partial^2 w}{\partial x^2}, \quad x \in (a,b), \quad t \in (0,T]. \tag{2.2}$$

In a similar manner, the initial condition is to be also converted to the form of

$$w(x,0) = exp(\left\{-\int_0^x \frac{f(s)}{2v}ds\right\} \tag{2.3}$$

and the given Dirichlet boundary conditions changed to the following homogeneous Neumann boundary conditions

$$w_x(a,t) = 0 \quad \text{and} \quad w_x(b,t) = 0. \tag{2.4}$$

Using the Taylor series expansion of $w(x,t)$, the Crank-Nicolson scheme leads to

$$\frac{w^{n+1} - w^n}{dt} = \frac{v}{24(\Delta x)^2}\left(\delta^4 w_i^{n+1} + \delta^4 w_i^n\right) \tag{2.5}$$

where $dt = T/N$ and $\Delta x = (b-a)/m = h$ are the temporal and spatial increments respectively. In Equation (2.5), $\delta^4 w_i$ is the fourth-order central finite difference operator given by $\delta^4 w_i = -w_{i-2} + 16w_{i-1} - 30w_i + 16w_{i+1} - w_{i+2}$.
Equation (2.5) is used to obtain a discrete system of linear equations for all $i = 2, \cdots, m-2$ that can also be rewritten as

$$(1 - r\delta^4)w_i^{n+1} = (1 + r\delta^4)w_i^n, \quad i = 2, 3, \ldots, m-2 \tag{2.6}$$

where $r = (vdt)/24h^2$. Boundary elements, $i = 1$ and $i = m-1$, were not included in the algorithm, so to incorporate all interior points, the second-order central finite difference operator defined by $\delta_x^2 w_i = w_{i+1} - 2w_i + w_{i-1}$ is implemented to Equation (2.5). Since this is only of a second order accuracy, and to make full use of fourth-order approximation, we prefer to use the Padê approximation [10] of the operator δ_x^2 defined as

$$\delta_x^2 = \frac{\delta_x^2}{1 + \frac{1}{12}\delta_x^2}. \tag{2.7}$$

(2.7) takes the place of the second order central difference operator and the whole approximation scheme for the boundary elements has been changed to the fourth order [16] which is given as follows

$$(1 - r\delta^4)w_i^{n+1} = (1 + r\delta^4)w_i^n, \quad i = 2, 3, \ldots, m-2 \tag{2.8}$$

$$(1 + \frac{1}{12}\delta_x^2 - s\delta_x^2)w_i^{n+1} = (1 + \frac{1}{12}\delta_x^2 + s\delta_x^2)w_i^n, \quad i = 1, m-1 \tag{2.9}$$

where $s = vdt/2h^2$. Since we have homogeneous Neumann boundary conditions (2.4), the first-order derivative is discretized using the fourth-order forward and backward schemes. After imposing the boundary conditions, the expressions (2.6), (2.8) and (2.9) can be stated as follows,

$$Aw_i^{n+1} = Bw_i^n \tag{2.10}$$

where A and B are square matrices of $(m-1) \times (m-1)$. Then system (2.10) can be solved by any appropriate linear solver to get the approximate values of $w(x,t)$. The initial approximation vector w^0 can be obtained from initial condition (2.3).

In attempting to solve the Burgers equation (1.1), the solution $u(x,t)$ now depends on two other functions $w(x,t)$ and $w_x(x,t)$ after the transformation. Solution for $w_x(x,t)$ will be obtained through the fourth-order approximation scheme by applying the second-order central finite difference operator to the first derivative Δ_x on $w(x_i,t)$ defined as $\Delta_x w_i = w_{i+1} - w_{i-1}$. To make use of the same order of accuracy, the Padê approximation has been used by replacing Δ_x with $\dfrac{\Delta_x}{1+\frac{1}{6}\delta_x^2}$ to achieve a fourth-order scheme. Thus a linear system has been obtained to get approximate results for $w_x(x_i,t)$ for all $i = 1,2,3,\ldots,m-1$, as follows

$$\left(1 + \frac{1}{6}\delta_x^2\right) w_x(x_i,t) = \frac{\Delta_x}{2h} w(x_i,t). \tag{2.11}$$

Constructed solution of $w(x,t)$ and the solution of $w_x(x,t)$ in (2.11) lead to the evaluation of the term $u(x,t)$.

3. Stability Analysis

Now we will see how well errors are being handled by this approximation scheme as the computation proceeds. We analyse the stability of the obtained matrix equation (2.10) by considering the present compact finite difference formulation for the nonlinear Burgers equation. Consider the von Neumann theory with the Fourier growth factor defined by

$$w_s^n = \hat{w}^n e^{isk\Delta x} \tag{3.1}$$

where k and Δx are mode number and the spatial increment, respectively. We aim to determine how much w^{n+1} grows under the consideration of of w_s^n by using (2.10). For typical value of r, equation (2.8) can be expressed as follows

$$\begin{aligned} rw_{i+2}^{n+1} - 16rw_{i+1}^{n+1} + (1+30r)w_i^{n+1} - 16rw_{i-1}^{n+1} + rw_{i+2}^{n+1} = \\ -rw_{i-2}^n + 16rw_{i-1}^n + (1-30r)w_i^n + 16rw_{i+1}^n - rw_{i+2}^n \end{aligned} \tag{3.2}$$

Substitution of (3.1) into (3.2) and applying Euler expansion for the exponential term give rise to

$$\frac{\hat{w}^n}{\hat{w}^{n+1}} = \frac{1 - 30r - 2r\cos 2k\Delta x + 32r\cos k\Delta x}{1 + 30r + 2r\cos 2k\Delta x - 32r\cos k\Delta x} = \beta. \tag{3.3}$$

The stability condition is hold by $\|\beta\| \leq 1$. Using considerable values of Δx and r, $0 \leq \cos k\Delta x \leq 1$ and $0 \leq \cos 2k\Delta x \leq 1$ hold. Thus it is concluded that $\|\beta\| \leq 1$ and the present algorithm is unconditionally stable.

4. Numerical Illustrations

Numerical solutions of the Burgers equation (1.1) for two test problems are considered in this section to display the accuracy and performance of the present method along the specified domain. The computed results are presented for comparison with exacts solution and the literature [15, 17],[23]–[31]. All results have been computed using the presently produced codes in MATLAB 2019a.
Problem 1 [22]: Let us consider the model equation (1.1) with initial condition

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

and the boundary conditions

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

Exact solution of Problem 1 was given by Cole [9] in a Fourier series form

$$u(x,t) = 2\pi v \frac{\sum_{n=1}^{\infty} a_n exp(-n^2\pi^2 vt) n sin(n\pi x)}{a_0 + \sum_{n=1}^{\infty} a_n exp(-n^2\pi^2 vt) n cos(n\pi x)} \tag{4.1}$$

where the coefficients a_0 and a_n of (4.1) can be calculated as

$$a_0 = \int_0^1 exp\{-(2\pi v)^{-1}[1 - cos(\pi x)]\} dx$$

$$a_n = \int_0^1 exp\{-(2\pi v)^{-1}[1 - cos(\pi x)]\} cos(n\pi x) dx, \quad \text{n} = 1, 2, 3, \ldots.$$

The produced results for the current problem are exhibited in Tables 1–5 for various selections of the kinematic viscosity constant as $v = 1, 0.1, 0.01$ and 0.004. The derived method for various values of the parameters h, dt, v at different levels of time t have been analysed and compared with the literature [15, 17],[23]–[29]. The computed results revealed that the present method offer far more accurate results comparison to the literature. This paper produce that have sharp gradients for the advection-dominated cases, $v << 1$, of the Burgers equation. The literature has shown that many numerical methods produce unwanted oscillations in dealing with such type of problems. However, as can be realized in Figures 1–2, the present method can overcome the advection dominated problem without any significant oscillatory results. Evaluation of the absolute errors has been displayed in Figures 3 and 4 and they have demonstrated that the derived scheme converges accurately to the exact solution. One more important advantage of the current method is that, to produce these results there is no restriction on time and space increments as proved in the stability analysis.

Table 1. Comparison of the computed results with the literature and exact solution for Problem 1 with $v = 1$, $h = 0.0125$, $dt = 0.00001$ and $t = 0.1$

x	Present	Explicit[15]	Exact-Explicit[15]	Exact
0.1	0.109538	0.10952	0.10955	0.109538
0.2	0.209792	0.20975	0.20981	0.209792
0.3	0.291896	0.29184	0.29192	0.291896
0.4	0.347923	0.34786	0.34795	0.347923
0.5	0.371577	0.37151	0.37161	0.371577
0.6	0.359045	0.35898	0.35907	0.359045
0.7	0.309905	0.30985	0.30993	0.309905
0.8	0.227817	0.22778	0.22783	0.227817
0.9	0.120686	0.12067	0.12070	0.120686

Table 2. Comparison of the computed results with the literature and exact solution for Problem 1 with $v = 0.1$, $h = 0.1$, $dt = 0.01$ at $t = 2.3$

x	Present	CFEM[25]	GFEM[25]	Exact
0.1	0.0221392	0.0221378	0.0221389	0.0221396
0.2	0.0427948	0.0427917	0.0427942	0.0427956
0.3	0.0604301	0.0604248	0.0604292	0.0604313
0.4	0.0734417	0.0734337	0.0734405	0.0734431
0.5	0.0802294	0.0802187	0.0802280	0.0802310
0.6	0.0793973	0.0793843	0.0793957	0.0793988
0.7	0.0701054	0.0700918	0.0701039	0.0701068
0.8	0.0525187	0.0525071	0.0525175	0.0525198
0.9	0.0281735	0.0281666	0.0281728	0.0281740

Table 3. Comparison of the computed results with the literature and exact solution at various times for Problem 1 with $v = 0.1$, $h = 0.0125$ and $dt = 0.0001$

x	t	Present	Ref [27]	Ref [15]	Exact
0.25	0.4	0.308894	0.30415	0.31215	0.30889
	0.6	0.240739	0.23629	0.24360	0.24074
	0.8	0.195675	0.19150	0.19815	0.19568
	1.0	0.162564	0.15861	0.16473	0.16256
0.50	0.4	0.569632	0.56711	0.57293	0.56963
	0.6	0.447205	0.44360	0.45088	0.44721
	0.8	0.359236	0.35486	0.36286	0.35924
	1.0	0.291916	0.28710	0.29532	0.29192
0.75	0.4	0.625437	0.61874	0.63038	0.62544
	0.6	0.487214	0.47855	0.49268	0.48721
	0.8	0.373921	0.36467	0.37912	0.37392
	1.0	0.291916	0.27860	0.29204	0.28747

Table 4. Comparison of the computed results with the literature and exact solution at various time for Problem 1 with $v = 0.01$ and $h = 0.01$

x	t	$dt = 0.004$			$dt = 0.001$		Exact
		Present	CFEM[25]	GFEM[25]	Ref[26]	Ref[17]	
	0.5	0.1211461	0.1211626	0.1211416	0.12079	0.12114	0.1211435
0.10	2.0	0.0429641	0.0429683	0.0429634	0.04300	0.04295	0.0429637
	4.0	0.0231043	0.0231057	0.0231041	0.02324	0.02310	0.0231042
	0.5	0.3602728	0.3602699	0.3602668	0.36113	0.36027	0.3602710
0.30	2.0	0.1288405	0.1288446	0.1288389	0.12887	0.12882	0.1288398
	4.0	0.0693085	0.0693104	0.0693080	0.06935	0.06930	0.0693082
	0.5	0.5886785	0.5886937	0.5886975	0.59559	0.58870	0.5886957
0.50	2.0	0.2145588	0.2145611	0.2145565	0.21468	0.21455	0.2145580
	4.0	0.1154950	0.1154973	0.1154943	0.11550	0.11549	0.1154947
	0.5	0.7933729	0.7935009	0.7935022	0.81257	0.79354	0.7934934
0.70	2.0	0.2999979	0.2999991	0.2999958	0.30075	0.29999	0.2999977
	4.0	0.1612150	0.1612261	0.1612140	0.16125	0.16121	0.1612146
	0.5	0.9378161	0.9382118	0.9381339	0.97184	0.93822	0.9381057
0.90	2.0	0.3732779	0.3734554	0.3732749	0.37452	0.37328	0.3732776
	4.0	0.1660592	0.1659905	0.1660577	0.16515	0.16605	0.1660587

Table 5. Comparison of the computed results with the literature and exact solution at various times for Problem 1 with $v = 0.004$ and $h = 0.01$

x	t	$dt = 0.0125$			$dt = 0.001$	Exact
		Present	CFEM[25]	GFEM[25]	Ref[29]	
0.25	1	0.18891	0.18889	0.18893	0.18891	0.18889
	5	0.04697	0.04698	0.04698	0.04698	0.04697
	10	0.02421	0.02422	0.02422	0.02422	0.02421
	15	0.01631	0.01632	0.01632	0.01632	0.01631
0.50	1	0.37567	0.37594	0.37603	0.37598	0.37596
	5	0.09393	0.09394	0.09395	0.09394	0.09393
	10	0.04843	0.04844	0.04844	0.04843	0.04843
	15	0.03259	0.03260	0.03260	0.03259	0.03259
0.75	1	0.55884	0.55880	0.55892	0.55883	0.55881
	5	0.14089	0.14089	0.14090	0.14089	0.14089
	10	0.07220	0.07221	0.07221	0.07221	0.07220
	15	0.04677	0.04678	0.04678	0.04678	0.04677

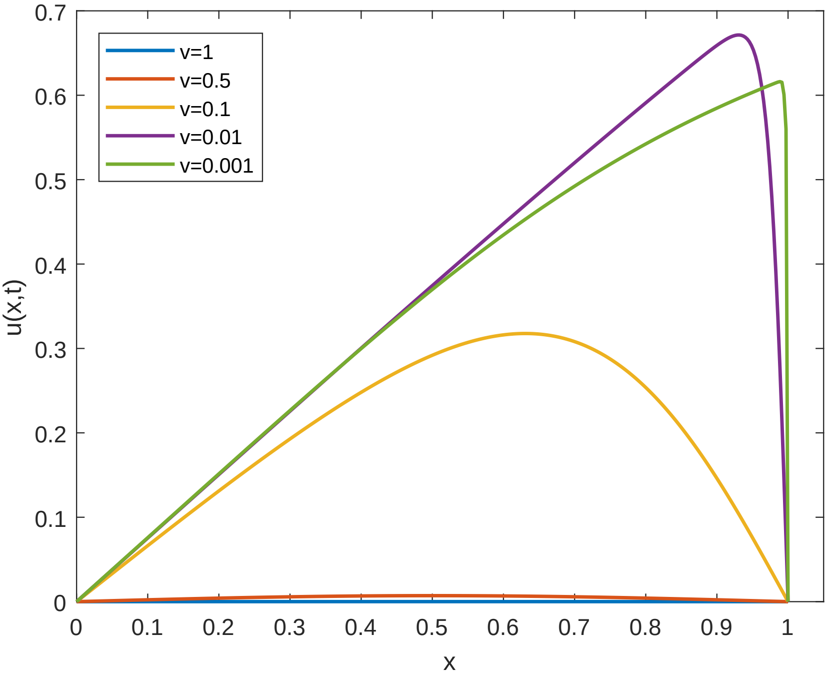

FIGURE 1. Solutions for various physical cases of Problem 1 with $t = 1$, $dt = 0.01$ and $h = 0.0025$

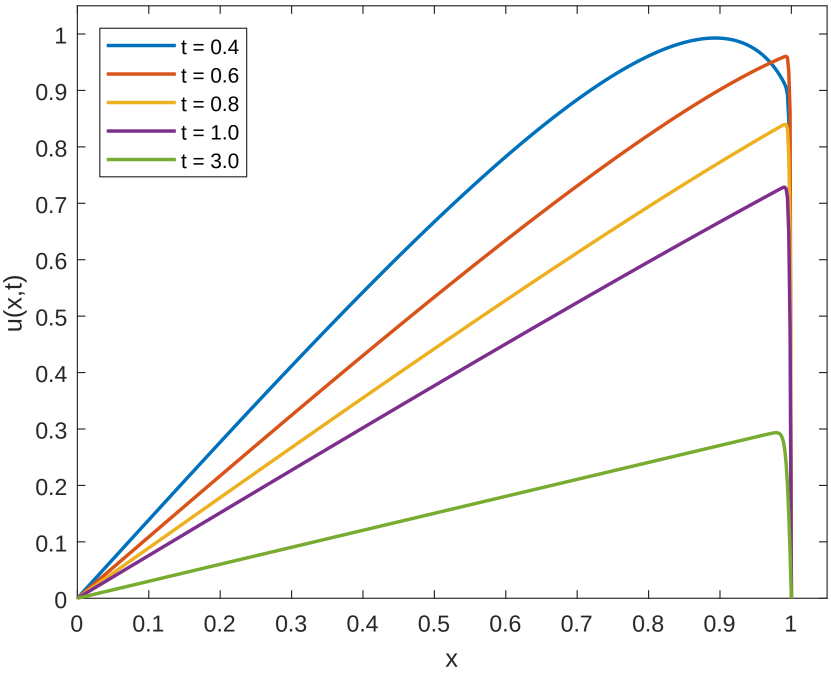

FIGURE 2. Advection dominated results of Problem 1 with $h = 0.0015$, $dt = 0.001$ and $v = 0.001$

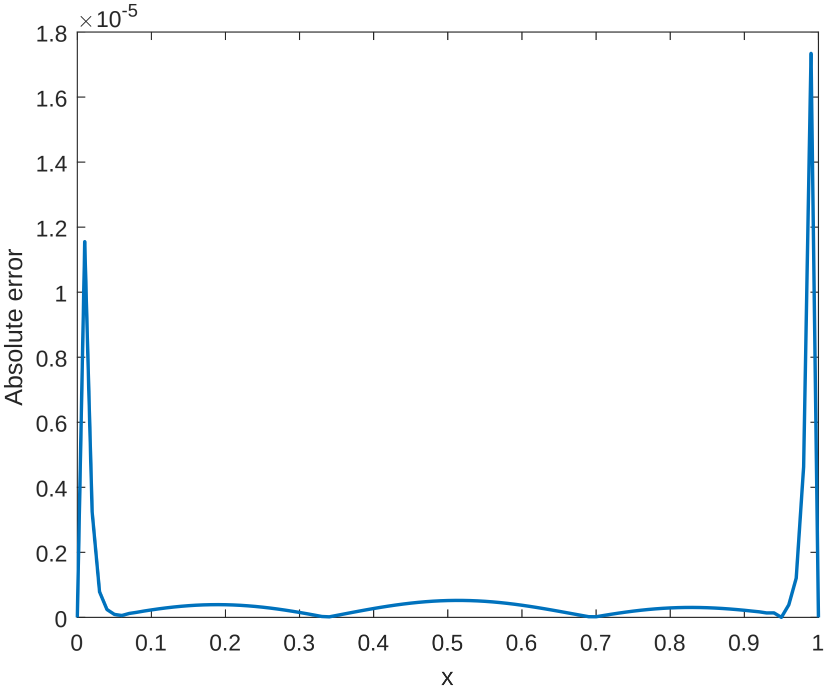

FIGURE 3. Absolute errors produced with $v = 0.1$, $dt = 0.001$, $t = 0.1$ and $h = 0.01$ for Problem 1

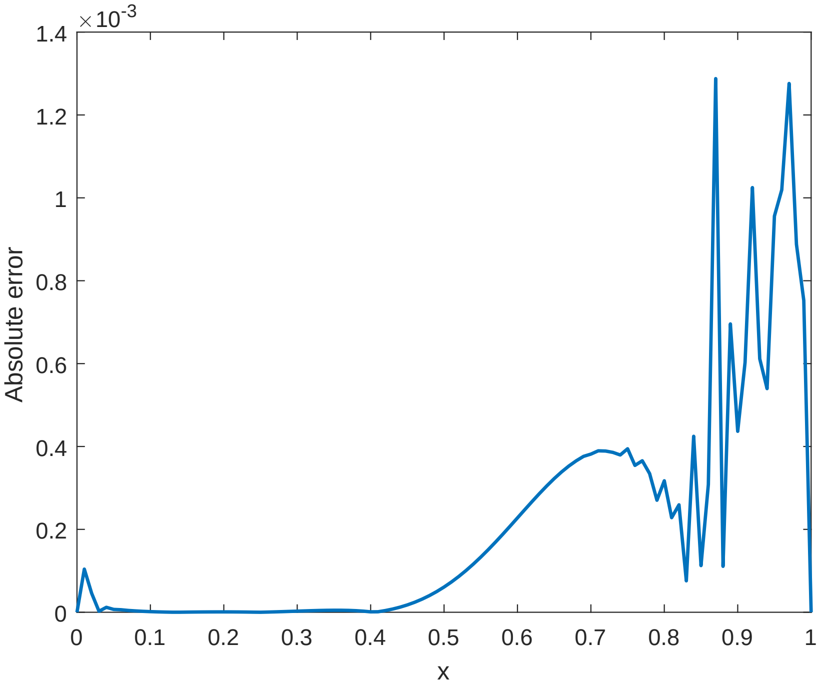

FIGURE 4. Absolute errors produced with $v = 0.01$, $dt = 0.001$, $t = 0.1$ and $h = 0.01$ for Problem 1

Problem 2 [22]: Let us consider the Burgers equation (1.1) with the initial condition

$$u(x,0) = 4x(1-x), \quad 0 \leq x \leq 1$$

and homogeneous boundary conditions,

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

The coefficients of the Fourier series solution (4.1) of this problem are

$$a_0 = \int_0^1 exp\{-x^2(3v)^{-1}[3-2x]\}dx$$

$$a_n = 2\int_0^1 exp\{-x^2(3v)^{-1}[3-2x]\}cos(n\pi x)dx, \quad \text{n} = 1,2,3,\ldots$$

In this problem, the numerical results have also been displayed and compared for the various parameters. The accuracy of the present method has been examined by comparing with literature [15, 17, 25, 26, 27, 30, 31] at $v = 1, 0.1, 0.01$ and $v = 0.004$. The numerical solutions revealed that our method offer highly significant accuracy comparison to the literature even less number of elements are considered as seen in Tables 6–9

Note that less elements are considered in both space and time for the present numerical algorithm than the rival methods. Advection-dominated cases of Problem 2 are illustrated in Figures 6 and 5. The current algorithm has high ability to produce physically accepted results as seen in the figures. Error distribution through the given domain has been shown in Figures 7 and 8 for the kinematic viscosity constant values of $v = 0.1$ and $v = 0.01$. Then for the same parameters values of m, dt and t the numerical results have shown to give better approximation to the exact solution with v of 0.1 compared to slightly less kinematic viscosity constants.

Table 6. Comparison of the computed results with the literature and exact solution at different times for Problem 2 with $v = 1$ and $h = 0.01$

x	t	$dt = 0.0002$			$dt = 0.0001$		Exact
		$h = 0.01$			$h = 0.0125$	$h = 0.25$	
		Present	CFEM[25]	GFEM[25]	Ref[15]	Ref[30]	
0.25	0.05	0.4262854	0.4262723	0.4262854	0.42629	0.4262864	0.4262856
	0.10	0.2614797	0.2614633	0.2614797	0.26149	0.2614801	0.2614798
	0.15	0.1614775	0.1614605	0.1614775	0.16148	0.1614777	0.1614776
	0.25	0.0610875	0.0610757	0.0610875	0.06109	0.0610875	0.0610875
0.50	0.05	0.6280837	0.6280605	0.6280837	0.62809	0.6280846	0.6280837
	0.10	0.3834223	0.3833919	0.3834223	0.38343	0.3834228	0.3834224
	0.15	0.2340552	0.2340271	0.2340552	0.23406	0.2340554	0.2340553
	0.25	0.0872326	0.0872150	0.0872326	0.08724	0.0872327	0.0872327
0.75	0.05	0.4652527	0.4652268	0.4652526	0.46526	0.4652528	0.4652526
	0.10	0.2815726	0.2815447	0.2815726	0.28158	0.2815727	0.2815726
	0.15	0.1697382	0.1697150	0.1697382	0.16974	0.1697383	0.1697382
	0.25	0.0622898	0.0622768	0.0622898	0.06229	0.0622898	0.0622898

Table 7. Comparison of the computed results with the literature and exact solution at various times for Problem 2 with $v = 0.1$ and $h = 0.0125$

x	t	$dt = 0.001$			$dt = 0.0001$		Exact
		Present	CFEM[25]	GFEM[25]	Ref[31]	Ref[27]	
0.25	0.4	0.3175230	0.3175299	0.3175226	0.32091	0.31749	0.3175229
	0.6	0.2461385	0.2461491	0.2461383	0.24910	0.24612	0.2461385
	0.8	0.1995553	0.1995677	0.1995552	0.20211	0.19954	0.1995553
	1.0	0.1655987	0.1656106	0.1655986	0.16782	0.16559	0.1655986
	3.0	0.0277587	0.0277529	0.0277587	0.02828	0.02776	0.0277587
0.50	0.4	0.5845376	0.5845555	0.5845372	0.58788	0.58448	0.5845373
	0.6	0.4579766	0.4579982	0.4579763	0.46174	0.45793	0.4579764
	0.8	0.3673983	0.3674150	0.3673981	0.37111	0.36736	0.3673982
	1.0	0.2983432	0.2983514	0.2983430	0.30183	0.29831	0.2983431
	3.0	0.0410650	0.0410543	0.0410650	0.04185	0.04106	0.0410650
0.75	0.4	0.6456160	0.6455490	0.6456160	0.65054	0.64547	0.6456155
	0.6	0.5026761	0.5025960	0.5026760	0.50825	0.50255	0.5026758
	0.8	0.3853358	0.3852565	0.3853356	0.39068	0.38523	0.3853355
	1.0	0.2958569	0.2957873	0.2958567	0.30057	0.29578	0.2958567
	3.0	0.0304396	0.0304300	0.0304396	0.03106	0.03044	0.0304396

Table 8. Comparison of the computed results with the literature and exact solution at different times for Problem 2 with $v = 0.01$ and $h = 0.01$

x	t	$dt = 0.002$			$dt = 0.001$		Exact
		Present	CFEM[25]	GFEM[25]	Ref[26]	Ref[17]	
0.10	0.5	0.128467	0.128486	0.128462	0.12808	0.12846	0.128462
	2.0	0.043814	0.043819	0.043814	0.04388	0.04379	0.043814
	4.0	0.023345	0.023347	0.023345	0.02351	0.02334	0.023345
0.30	0.5	0.378489	0.378492	0.378488	0.37956	0.37849	0.378489
	2.0	0.131346	0.131351	0,131345	0.13129	0.13131	0.131345
	4.0	0.070027	0.070030	0.070027	0.07009	0.07002	0.070027
0.50	0.5	0.609905	0.609886	0.609886	0.61768	0.60991	0.609886
	2.0	0.218589	0.218593	0.218588	0.21873	0.21858	0.218588
	4.0	0.116682	0.116685	0.116682	0.11671	0.11667	0.116682
0.70	0.5	0.809878	0.809785	0.809784	0.83022	0.80986	0.809782
	2.0	0.305349	0.305352	0.305348	0.30614	0.30534	0.305348
	4.0	0.162878	0.162890	0.162878	0.16293	0.16287	0.162878
0.90	0.5	0.946085	0.946090	0.946019	0.98068	0.94615	0.946014
	2.0	0.380282	0.380461	0.380273	0.38163	0.38027	0.380274
	4.0	0.168578	0.168511	0.168577	0.16766	0.16857	0.168577

TABLE 9. Comparison of the computed results with the literature and exact solution at various times for Problem 2 with $v = 0.004$ and $h = 0.01$

x	t	$dt = 0.01$			$dt = 0.001$	Exact
		Present	CFEM[25]	GFEM[25]	Ref[29]	
0.25	1	0.19640	0.19639	0.19638	0.19636	0.19641
	5	0.04744	0.04744	0.04744	0.04744	0.04744
	10	0.02434	0.02434	0.02434	0.02434	0.02434
	15	0.01637	0.01637	0.01637	0.01637	0.01637
0.50	1	0.38835	0.38847	0.38847	0.38842	0.38846
	5	0.09486	0.09486	0.09486	0.09491	0.09493
	10	0.04868	0.04869	0.04868	0.04868	0.04869
	15	0.03270	0.03271	0.03271	0.03270	0.03270
0.75	1	0.57209	0.57318	0.57319	0.57312	0.57315
	5	0.14225	0.14225	0.14224	0.14224	0.14225
	10	0.07258	0.07259	0.07258	0.07258	0.07257
	15	0.04696	0.04697	0.04696	0.04696	0.04695

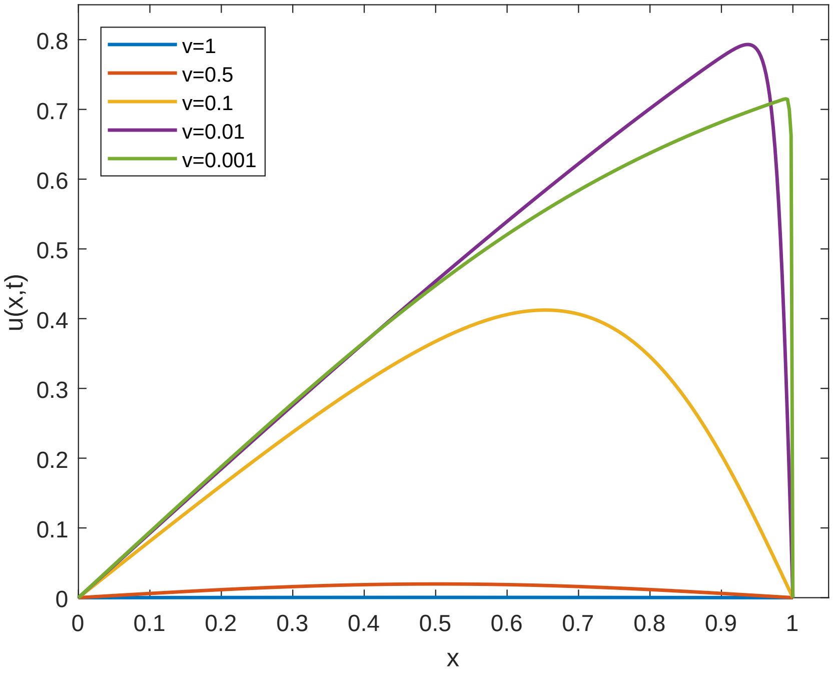

FIGURE 5. Solutions for various physical cases of Problem 2 with $t = 0.8$, $dt = 0.008$ and $h = 0.0025$

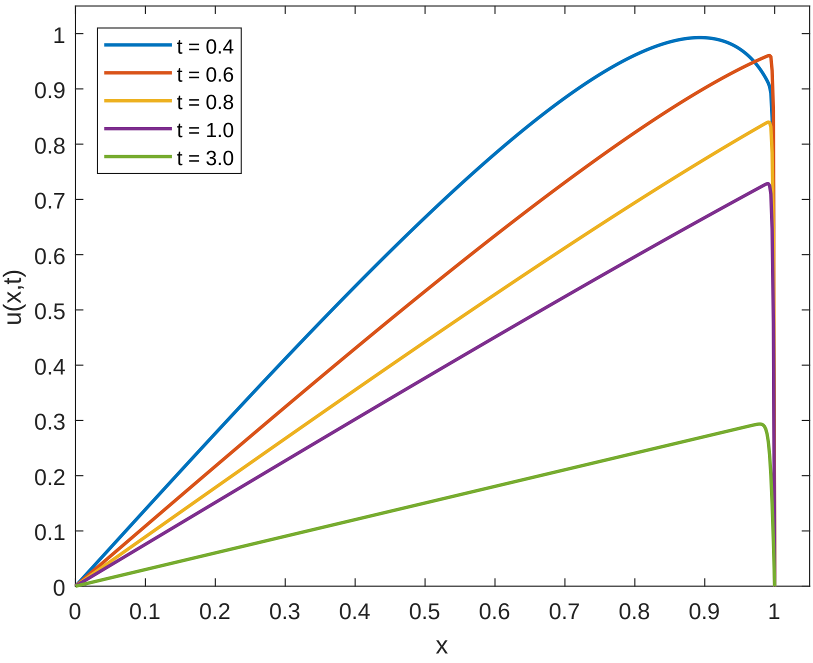

FIGURE 6. Advection dominated results of Problem 2 with $h = 0.0015$, $dt = 0.001$ and $v = 0.001$

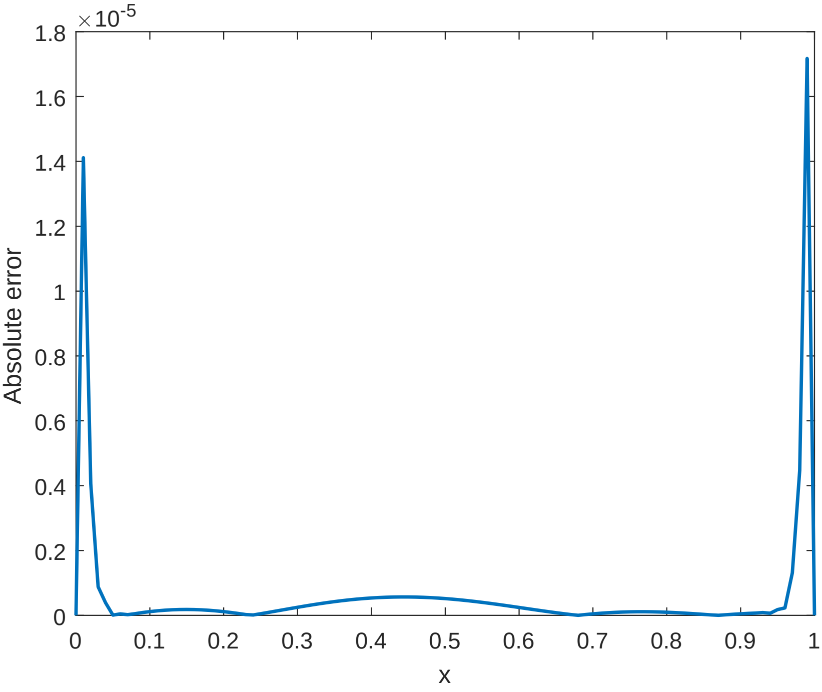

FIGURE 7. Absolute errors produced with $v = 0.1$, $dt = 0.001$, $t = 0.1$ and $h = 0.01$ for Problem 2

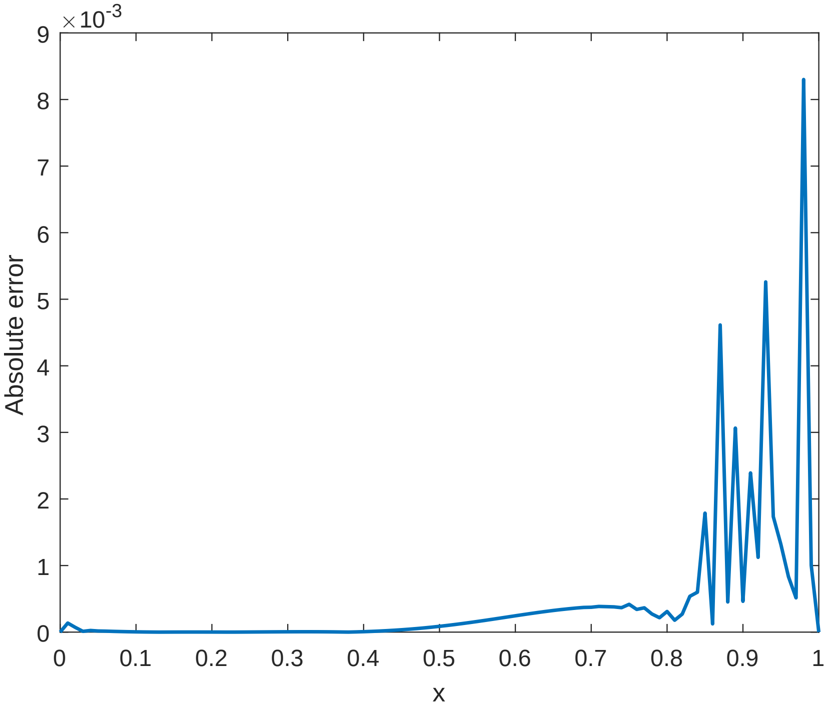

FIGURE 8. Absolute errors produced with $v = 0.01$, $dt = 0.001$, $t = 0.1$ and $h = 0.01$ for Problem 2

5. Conclusions

In this paper, to capture the behaviour of the nonlinear advection–diffusion process, a fourth-order finite difference scheme has been presented. In the process of producing a compact scheme, the Padé approximation has been applied to maintain the fourth-order accuracy in space. Numerical treatments revealed that the currently developed scheme is of oscillation free for advection dominated problems. The present approach has been shown to be unconditionally stable based on the von Neumann stability analysis. The current compact scheme has proven to yield precise and more accurate results in comparison with the literature.

References

[1] A. Thom and C. J. Apelt, Field Computations in Engineering and Physics, *D. Van Nostrand.* (1961).

[2] S. C. Chapra and P. C. Raymond, Numerical Methods for Engineers, *McGraw-Hill Higher Education.* (2006),Boston.

[3] H. Bateman, Some recent researches on the motion of fluids, *Monthy Weather Review.* **43**(1915), 163–170.

[4] B. Inan and A.R. Bahadir, Numerical solution of the one-dimensional Burgers equation: Implicit and fully Implicit exponential finite difference methods, *PRAMANA Journal of Physics.* **81**(2013), 547–556.

[5] M. Vijitha and A. Ashish, Linearized Implicit Numerical Method for Burgers Equation, *Nonlinear Engineering.* **5**(2012), 219–234.

[6] A. Cordero, A. Franques and J. R. Torregrosa, Numerical Solution of Turbulence Problems by Solving Burgers Equation, *Algorithms* **8**(2015), 224–233.

[7] J. M. Burgers, A mathematical model illustrating the theory of turbulence, *Advances in Applied Mechanics.* **1**(1948), 171-199.

[8] E. Hopf, The partial differential equation $u_t + uu_x = \mu u_{xx}$, *Communication on Pure and Applied Mathematics.* **3**(1950), 201–230.

[9] J.D. Cole, On a quasi-linear parabolic equation occurring in aerodynamics, *Quarterly of Applied Mathematics.* **9**(1951), 225-236.

[10] M. Gulsu, A finite difference approach for solution of Burgers equation, *Applied Mathematics and Computation.* **175**(2006), 1245–1255.

[11] S. Xie, S. Heo, S. Kim, G. Woo and S. Yi, Numerical solution of one-dimensional Burgers equation using reproducing kernel function, *Journal of Computational and Applied Mathematics.* **214**(2008), 417–434.

[12] T. A. Hassanien, A.A. Salama and H.A. Hosham, Fourth-order finite difference method for solving Burgers equation, *Applied Mathematics and Computation.* **170**(2005), 781–800.

[13] M. Guslu and T. Ozis, Numerical solution of Burgers equation with restrictive Taylor approximation, *Applied Mathematics and Computation.* **171**(2005), 1192–1200.

[14] P. G. Zhang and J. P. Wang, A predictorcorrector compact finite difference scheme for Burgers equation, *Applied Mathematics and Computation.* **219** (2012), 892–898.

[15] S. Kutluay, A.R. Bahadir and A. Ozdes, Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods, *Journal of Computational and Applied Mathematics.* **103**(1999), 251–261.

[16] W. Liao, An implicit fourth-order compact finite difference scheme for one-dimensional Burgers equation, *Applied Mathematics and Computation.* **206**(2008), 755–764.

[17] M. Sari and G. Gurarslan, A sixth-order compact finite difference scheme to the numerical solutions of Burgers equation, *Applied Mathematics and Computation.* **208**(2009), 475–483.

[18] A. Dogan, A Galerkin finite element approach to Burgers equation, *Applied Mathematics and Computation.* **157**(2004), 331–346.

[19] E. N. Aksan, Quadratic B-spline finite element method for numerical solution of the Burgers equation, *Applied Mathematics and Computation.* **174**(2006), 884-896.

[20] C. Zhu and R. Wang, Numerical solution of Burgers equation by cubic B-spline quasi-interpolation, *Applied Mathematics and Computation.* **208**(2009), 260–272.

[21] Z. Rong-Pei, Y. Xi-Jun and Z. Guo-Zhong Local discontinuous Galerkin method for solving Burgers and coupled Burgers equations, *Chinese Physics B.* **20**(2011), 110–205.

[22] M. Sari and H. Tunc, An Optimization Technique in Analyzing the Burgers Equation, *Sigma Journal of Engineering and Natural Sciences.* **35**(2017), 369-386.

[23] K. Sakai and I. Kimura, A numerical scheme based on a solution of nonlinear advectiondiffusion equations, *Journal of Computational and Applied Mathematics.* **173**(2005), 39–55.

[24] S. Kutluay and A. Esen, A linearized numerical scheme for Burgers-like equations, *Applied Mathematics and Computation.* **156**(2004), 295–305.

[25] H. Tunc, Various Finite Element Techniques for Advection Difffusion Reaction Processes Master Thesis, *Yildiz Technical University.* (2017), Istanbul.

[26] A. R. Bahadır and M. Sağlam, A mixed finite difference and boundary element approach to one-dimensional Burgers equation, *Applied Mathematics and Computation.* **160**(2005), 663–673.

[27] S. Kutluay and A. Esen, A lumped Galerkin method for solving the Burgers equation, *International Journal of Computational Mathematics.* **81**(2004), 1433–1444.

[28] D.K. Salkuyeh and F. S. Sharafeh, On the numerical solution of the Burgers equation, *International Journal of Computer Mathematics.* **86**(2009), 1334–1344.

[29] R. Jiwari, A hybrid numerical scheme for the numerical solution of the Burgers equation, *Computer Physics Communications.* **188**(2015) 59–67.

[30] L. Shao, X. Feng and Y. He, The local discontinuous Galerkin finite element method for Burgers equation, *Mathematical and Computer Modelling.* **54**(2011), 2943–2954.

[31] S. Kutluay, A. Esen and I. Dag, Numerical solutions of the Burgers equation by the least-squares quadratic B-spline finite element method, *Journal of Computational and Applied Mathematics.* **167**(2004), 21–33.

Murat Sari
Department of Mathematics, Yildiz Technical University, Istanbul, Turkey.
E-mail address: `sarim@yildiz.edu.tr`

Sufii H. Mussa
Department of Mathematics, Yildiz Technical University, Istanbul, Turkey.
E-mail address: `f6516002@std.yildiz.edu.tr`

Huseyin Tunc
Department of Mathematics, Yildiz Technical University, Istanbul, Turkey.
E-mail address: `huseyin.tunc@std.yildiz.edu.tr`

Received: June 2, 2019; Accepted: September 23, 2019