DOI 10.1007/s11253-018-1541-9
Ukrainian Mathematical Journal, Vol. 70, No. 6, November, 2018 (Ukrainian Original Vol. 70, No. 6, June, 2018)

EXPONENTIAL TWICE CONTINUOUSLY DIFFERENTIABLE B-SPLINE ALGORITHM FOR BURGERS' EQUATION

O. Ersoy, I. Dag, and N. Adar

UDC 517.5

Exponential twice continuously differentiable B-spline functions (known from the literature as exponential) are used to set up the collocation method for finding solutions of Burgers' equation. The effect of exponential cubic B-splines in the collocation method is analyzed by studying the test problems.

1. Introduction

In the present paper, we adapt exponential cubic B-spline functions to the collocation method with an aim to develop a numerical method for finding numerical solutions of Burgers' equation of the form

$$U_t + UU_x - \lambda U_{xx} = 0, \qquad a \le x \le b, \quad t \ge 0, \tag{1}$$

with the following initial condition and the boundary conditions:

$$U(x,0) = f(x), \quad a \le x \le b, \tag{2}$$

$$U(a,t) = \sigma_1, \qquad U(b,t) = \sigma_2, \tag{3}$$

where the subscripts x and t denote differentiation,

$$\lambda \doteq \frac{1}{\mathrm{Re}} > 0,$$

Re is the Reynolds number characterizing the intensity of viscosity, σ_1 and σ_2 are constants, $u = u(x,t)$ is an unknown function differentiable sufficiently many tines, and $f(x)$ is a bounded function. The initial and boundary conditions are specified in what follows depending on the test problems.

Burgers' equation was first introduced by [2]. The solutions of Burgers' equation were presented by using some numerical methods with splines. A cubic spline collocation procedure was developed for the numerical solution of Burgers' equation in [3, 4]. A B-spline Galerkin method was proposed to solve Burgers' equation for both fixed and variable distributions of knots used to define the B-splines in the studies by Davies [5, 6]. A numerical method was developed in [7–9] for the solution of Burgers' equation by using the splitting method and the cubic spline approximation method. In [14, 17, 18, 23], the numerical solutions of one-dimensional Burgers' equation were obtained by the methods based on the collocation of quadratic, cubic, and quintic B-splines over finite elements. In these methods, the approximate functions of the collocation method for Burgers' equation were constructed by using B-splines of various degrees. Galerkin's methods based on B-splines of various degrees

Eskişehir Osmangazi University, Eskişehir, Turkey.

Published in Ukrains'kyi Matematychnyi Zhurnal, Vol. 70, No. 6, pp. 788–800, June, 2018. Original article submitted October 20, 2016.

 0041-5995/18/7006–0906

were used to find approximate solutions of the Burgers' equation in [6, 11, 13]. The least-squares method was combined with B-splines to get numerical methods for the solution of Burgers' equation in [10, 22]. The numerical solutions of Burgers' equation were presented on the basis of cubic B-spline quasiinterpolation and a compact finite difference method in [20]. The Taylor-collocation and Taylor–Galerkin methods for the numerical solutions of Burgers' equation were formed by using both cubic and quadratic B-splines in [19]. The differential quadrature methods based on cubic and quartic B-splines were used to solve Burgers' equation in [21, 25, 26]. A hybrid spline difference method was developed to solve Burgers' equation by Wang, et al. [24].

The exponential cubic B-spline function and some of its properties were described in detail [27]. Since all used exponential bases are twice continuously differentiable, we can find twice continuously differentiable approximate solutions of differential equations. There are few articles used to construct numerical methods for the solution of differential equations. The exponential cubic B-splines are used with the collocation method in order to find the numerical solution of the singular perturbation problem posed by M. Sakai, et al. [28]. Another application of the collocation method based on the cardinal exponential cubic B-splines was proposed for finding numerical solutions of a singularly perturbed boundary-value problem by Radunovic [29]. An exponential cubic B-spline collocation method was designed to obtain numerical solutions of self-adjoint singularly perturbed boundary-value problems in [30]. The only linear partial differential equation known as the convection-diffusion equation was solved with the help of the exponential cubic B-spline collocation method in [31]. Moreover, the exponential cubic B-spline collocation method has been recently applied to obtain numerical solutions of the equal-width equation, Korteweg–de-Vries equation, Fisher equation, and Kuramoto–Sivashinsky equation [32–35].

In the present paper, we compare the results obtained for Burgers' equation with the results obtained by using both the cubic B-spline collocation method and the cubic B-spline Galerkin finite-element method [12, 13] because the B-spline and exponential cubic B-spline functions have almost the same properties. In Section 2, we describe the exponential cubic B-spline collocation method. In Section 3, three classical test examples are analyzed to show the versatility of the proposed algorithm and, finally, the conclusions are made to discuss the outcomes of the proposed algorithm.

2. Collocation Method via Exponential Cubic B-Spline

The analyzed domain $[a, b]$ is equally partitioned at the nodes

$$\pi: \ a = x_0 < x_1 < \ldots < x_N = b$$

with a distance $h = (b-a)/N$ between the consecutive nodes. The exponential cubic B-splines $B_i(x)$ can be defined at the points of π as follows:

$$B_i(x) = \begin{cases} b_2\Big((x_{i-2} - x) - \dfrac{1}{p}\left(\sinh(p(x_{i-2} - x))\right)\Big), & [x_{i-2}, x_{i-1}], \\ a_1 + b_1(x_i - x) + c_1 \exp\big(p(x_i - x)\big) + d_1 \exp\big(-p(x_i - x)\big), & [x_{i-1}, x_i], \\ a_1 + b_1(x - x_i) + c_1 \exp\big(p(x - x_i)\big) + d_1 \exp\left(-p(x - x_i)\right), & [x_i, x_{i+1}], \\ b_2\Big((x - x_{i+2}) - \dfrac{1}{p}(\sinh\left(p\left(x - x_{i+2}\right)\right))\Big), & [x_{i+1}, x_{i+2}], \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

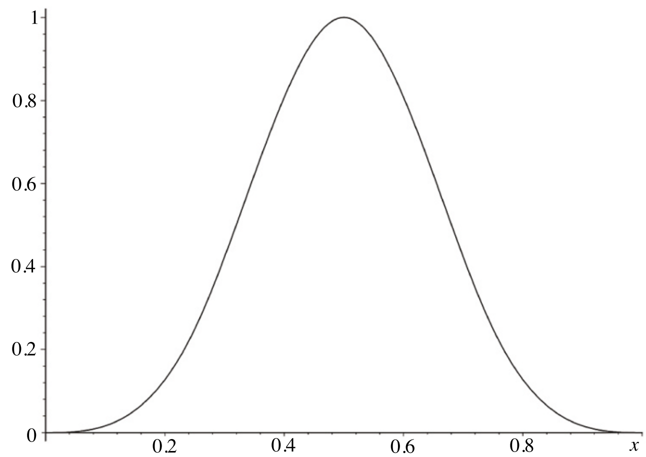

Fig. 1. Exponential cubic B-splines over the interval $[0,1]$.

Table 1

Values of $B_i(x)$ and Its Two Principal Derivatives at the Nodal Points

x	x_{i-2}	x_{i-1}	x_i	x_{i+1}	x_{i+2}
B_i	0	$\dfrac{s-ph}{2(phc-s)}$	1	$\dfrac{s-ph}{2(phc-s)}$	0
B_i'	0	$\dfrac{p(1-c)}{2(phc-s)}$	0	$\dfrac{p(c-1)}{2(phc-s)}$	0
B_i''	0	$\dfrac{p^2 s}{2(phc-s)}$	$-\dfrac{p^2 s}{phc-s}$	$\dfrac{p^2 s}{2(phc-s)}$	0

where

$$a_1 = \frac{phc}{phc-s}, \qquad b_1 = \frac{p}{2}\left[\frac{c(c-1)+s^2}{(phc-s)(1-c)}\right], \qquad b_2 = \frac{p}{2(phc-s)},$$

$$c_1 = \frac{1}{4}\left[\frac{\exp(-ph)(1-c)+s(\exp(-ph)-1)}{(phc-s)(1-c)}\right],$$

$$d_1 = \frac{1}{4}\left[\frac{\exp(ph)(c-1)+s(\exp(ph)-1)}{(phc-s)(1-c)}\right],$$

$c = \cosh(ph)$, $s = \sinh(ph)$, and p is a free parameter. For a specific interval $[0,1]$, the exponential cubic B-spline function is depicted for $p = 1$ in Fig. 1.

The functions $\{B_{-1}(x), B_0(x), \ldots, B_{N+1}(x)\}$ form a basis and, hence, any function defined on the interval $[a,b]$ can be expressed as a linear combination of elements of the basis. Every basis function $B_i(x)$ has the second derivative. The values of $B_i(x)$, $B_i'(x)$ and $B_i''(x)$ at the nodes x_i can be found from Eq. (4). They are shown in Table 1.

Let U_N be an approximate solution for U :

$$U_N(x,t) = \sum_{i=-1}^{N+1} \delta_i B_i(x), \tag{5}$$

where δ_i are time-dependent parameters. The nodal values of $U(x,t)$ and the values of its first and second derivatives at the nodes can be found from Eq. (5) depending on the following parameters:

$$\begin{aligned}
U_i = U(x_i,t) &= \frac{s-ph}{2(phc-s)}\delta_{i-1} + \delta_i + \frac{s-ph}{2(phc-s)}\delta_{i+1},\\
U_i' = U'(x_i,t) &= \frac{p(1-c)}{2(phc-s)}\delta_{i-1} + \frac{p(c-1)}{2(phc-s)}\delta_{i+1},\\
U_i'' = U''(x_i,t) &= \frac{p^2 s}{2(phc-s)}\delta_{i-1} - \frac{p^2 s}{phc-s}\delta_i + \frac{p^2 s}{2(phc-s)}\delta_{i+1}.
\end{aligned} \tag{6}$$

The time discretization of the unknown function U is realized by applying the Grank–Nicolson scheme to Burgers' equation. As a result, we obtain the following equation:

$$\frac{U^{n+1}-U^n}{\Delta t} + \frac{(UU_x)^{n+1} + (UU_x)^n}{2} - \lambda \frac{U_{xx}^{n+1} + U_{xx}^n}{2} = 0, \tag{7}$$

where $U^{n+1} = U(x,t)$ is the solution of the equation on the $(n+1)$th time level. Here, $t^{n+1} = t^n + \Delta t$, Δt is the time step, and the superscripts denote nth time level, $t^n = n\Delta t$.

The nonlinear term $(UU_x)^{n+1}$ in Eq. (7) is linearized by using the following form proposed by Rubin and Graves [3]:

$$(UUx)^{n+1} = U^{n+1}U_x^n + U^n U_x^{n+1} - U^n U_x^n. \tag{8}$$

It is applied to Eq. (7) to obtain time discretized Burgers' equation:

$$U^{n+1} - U^n + \frac{\Delta t}{2}\left(U^{n+1}U_x^n + U^n U_x^{n+1}\right) - \lambda\frac{\Delta t}{2}\left(U_{xx}^{n+1} - U_{xx}^n\right) = 0. \tag{9}$$

Substituting Eq. (5) in (9), we arrive at a fully discretized system of equations

$$\begin{aligned}
&\left(\alpha_1 + \frac{\Delta t}{2}\left(\alpha_1 L_2 + \beta_1 L_1 - \lambda\gamma_1\right)\right)\delta_{m-1}^{n+1} + \left(\alpha_2 + \frac{\Delta t}{2}\left(\alpha_2 L_2 - \lambda\gamma_2\right)\right)\delta_m^{n+1}\\
&\qquad + \left(\alpha_3 + \frac{\Delta t}{2}\left(\alpha_3 L_2 + \beta_2 L_1 - \lambda\gamma_3\right)\right)\delta_{m+1}^{n+1}\\
&= \left(\alpha_1 + \lambda\frac{\Delta t}{2}\gamma_1\right)\delta_{m-1}^n\\
&\qquad + \left(\alpha_2 + \lambda\frac{\Delta t}{2}\gamma_2\right)\delta_m^n + \left(\alpha_3 + \lambda\frac{\Delta t}{2}\gamma_3\right)\delta_{m+1}^n,
\end{aligned} \tag{10}$$

where

$$L_1 = \alpha_1\delta_{i-1} + \alpha_2\delta_i + \alpha_3\delta_{i+1},$$

$$L_2 = \beta_1\delta_{i-1} + \beta_2\delta_{i+1},$$

$$\alpha_1 = \frac{s-ph}{2(phc-s)}, \qquad \alpha_2 = 1, \qquad \alpha_3 = \frac{s-ph}{2(phc-s)},$$

$$\beta_1 = \frac{p(1-c)}{2(phc-s)}, \qquad \beta_2 = \frac{p(c-1)}{2(phc-s)},$$

$$\gamma_1 = \frac{p^2 s}{2(phc-s)}, \qquad \gamma_2 = -\frac{p^2 s}{phc-s}, \qquad \gamma_3 = \frac{p^2 s}{2(phc-s)}.$$

The system consists of $N+1$ linear equations with $N+3$ unknown parameters

$$\mathbf{d}^{n+1} = \left(\delta_{-1}^{n+1}, \delta_0^{n+1}, \ldots, \delta_{N+1}^{n+1}\right).$$

The boundary conditions

$$\sigma_1 = U_0, \qquad \sigma_2 = U_N$$

give two additional linear equations

$$\begin{aligned} \delta_{-1} &= \frac{1}{\alpha_1}\left(U_0 - \alpha_2\delta_0 - \alpha_3\delta_1\right), \\ \delta_{N+1} &= \frac{1}{\alpha_3}\left(U_N - \alpha_1\delta_{N-1} - \alpha_2\delta_N\right). \end{aligned} \tag{11}$$

Equations (11) can be used to eliminate δ_{-1} and δ_{N+1} from system (10). As a result, it turns into a solvable matrix equation for the unknown $\delta_0^{n+1}, \ldots, \delta_N^{n+1}$. A version of the Thomas algorithm is used to solve the system.

By using the initial condition and the first space derivative of the initial conditions on the boundaries, we get the following system:

$$U_N(x_i, 0) = U(x_i, 0), \quad i = 0, \ldots, N,$$

$$(U_x)_N(x_0, 0) = U'(x_0),$$

$$(U_x)_N(x_N, 0) = U'(x_N).$$

3. Computational Examples

The solution of the system produces the initial parameters $\delta_{-1}^0, \delta_0^0, \ldots, \delta_{N+1}^0$ and, hence, we can start solving the recursive system at the requested times. The numerical method described in the previous section will be tested for three test problems of finding the solutions of Burgers' equation. Three kinds of examples are presented in order

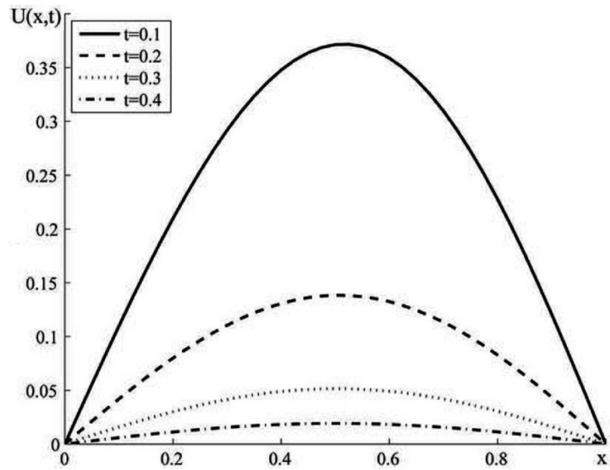

Fig. 2. Solutions for $\lambda = 1,\ N = 40,$ and $\Delta t = 0.0001$.

to demonstrate the versatility and accuracy of the proposed method. The discrete L_2 and L_∞ error norms, namely,

$$L_2 = \sqrt{h\sum_{j=0}^{N}\left|\left(U_j^n - (U_N)_j^n\right)^2\right|},$$

$$L_\infty = \max_j \left|U_j^n - (U_N)_j^n\right|$$

are used to measure the errors between the analytic and numerical solutions.

(a) Burger's equation with a sine-wave initial condition $U(x,0)=\sin(\pi x)$ and boundary conditions $U(0,t)= U(1,t) = 0$, possesses an analytic solution in the form of an infinite series defined by [15] as follows:

$$U(x,t) = \frac{4\pi\lambda \sum_{j=1}^{\infty} jI_j\left(\frac{1}{2\pi\lambda}\right)\sin(j\pi x)\exp(-j^2\pi^2\lambda t)}{I_0\left(\frac{1}{2\pi\lambda}\right) + 2\sum_{j=1}^{\infty} I_j\left(\frac{1}{2\pi\lambda}\right)\cos(j\pi x)\exp(-j^2\pi^2\lambda t)}, \tag{12}$$

where $\mathbf{I}_j$ are the modified Bessel functions. This problem describes the decay of a sinusoidal disturbance. Its numerical solutions at different times are depicted in Figs. 2–5 for the parameters $N = 40$ and $N = 80$, $\Delta t = 0.0001$, and $\lambda = 1,\ 0.1,\ 0.01,\ 0.001$. From these figures, we see that lower viscosities λ cause the development of a sharp front through the right boundary. The amplitude of the sharp front starts to decay with time. These properties of the solutions are in very good agreement with the results obtained by Saka and Dağ [16, 17].

The two-dimensional solutions are depicted in Figs. 6–9 from time $t = 0$ to $t = 1$ with time increments $\Delta t = 0.0001$ and space increments $h = 0.25$ for various λ. If a smaller value of $\lambda = 0.001$ is taken, then the solutions start to decay after about $t = 0.6$ for $N = 40$. Hence, in order to get acceptable solution with $\lambda = 0.001$, we decrease the space step down to $h = 0.125$. The plot of this solution is shown in Figs. 8, 9.

A comparison is made between the proposed collocation method and the alternative approaches, including the cubic B-spline collocation method and cubic B-spline Galerkin method for the parameters $\Delta t = 0.0001,\ N = 80$, and $\lambda = 0.01$. The exact solutions for $\lambda > 10^{-2}$ are not practical due to the weak convergence of the infinite

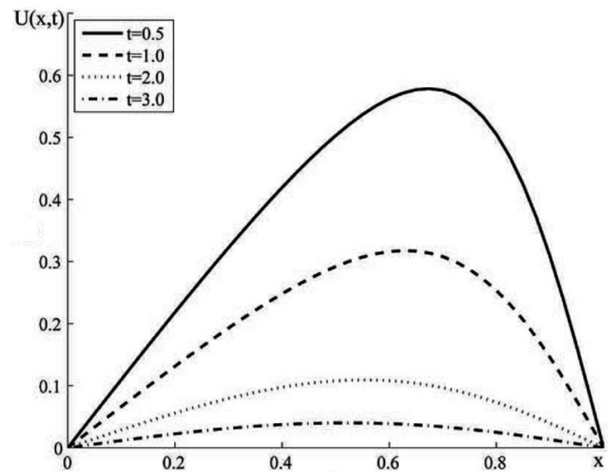

Fig. 3. Solutions for $\lambda = 0.1,\ N = 40$, and $\Delta t = 0.0001$.

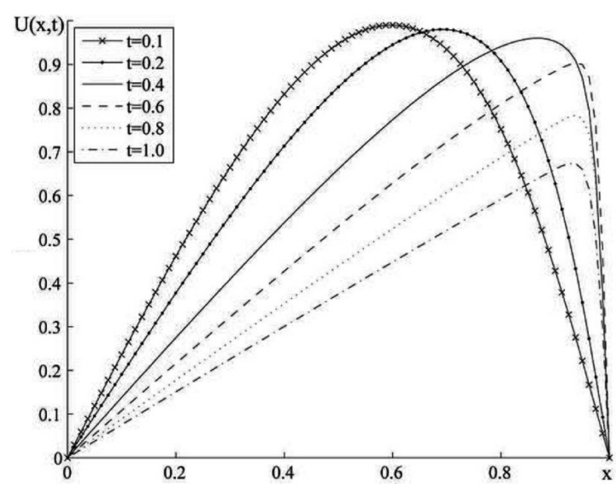

Fig. 4. Solutions for $\lambda = 0.01,\ N = 80$, and $\Delta t = 0.0001$.

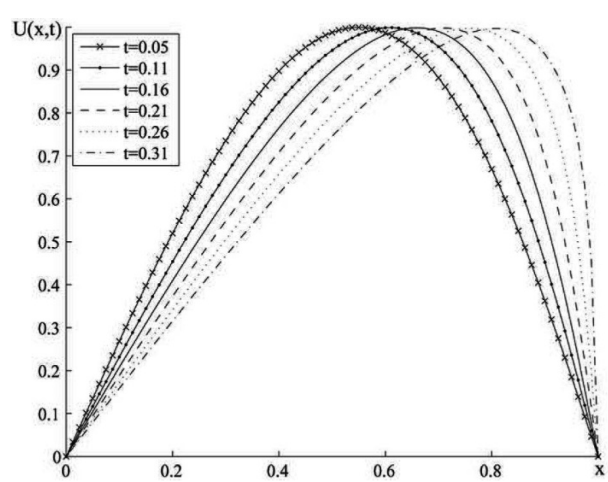

Fig. 5. Solutions for $\lambda = 0.001,\ N = 80$, and $\Delta t = 0.0001$.

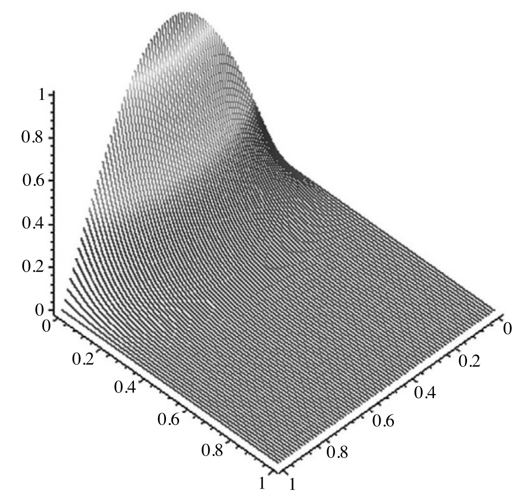

Fig. 6. Solutions for $\lambda = 1,\ N = 40$, and $\Delta t = 0.01$.

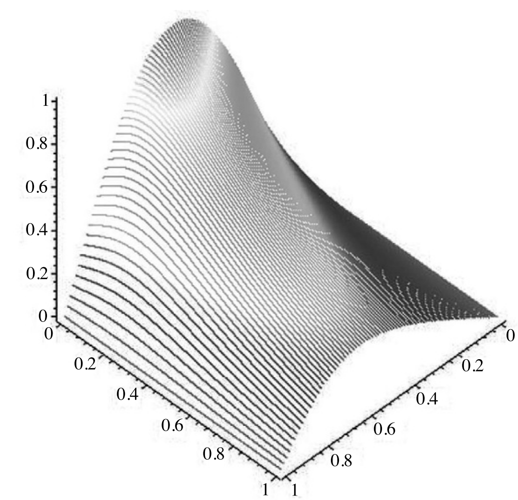

Fig. 7. Solutions for $\lambda = 0.1,\ N = 40$, and $\Delta t = 0.01$.

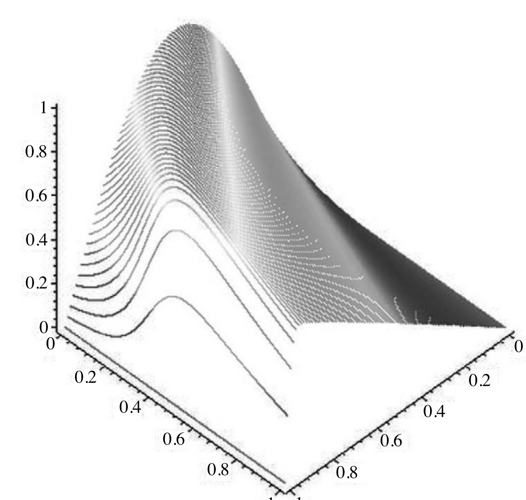

Fig. 8. Solutions for $\lambda = 0.01,\ N = 80$, and $\Delta t = 0.01$.

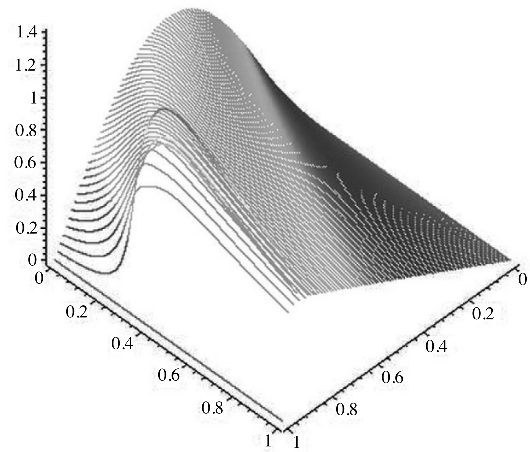

Fig. 9. Solutions for $\lambda = 0.001$, $N = 80$, and $\Delta t = 0.001$.

Table 2

Numerical Results for $p = 1$, $\lambda = 0.01$, $N = 40$, $\Delta t = 0.0001$, and Different Times

x	Time	Present	Ref. [12]	Ref. [13]	Exact
0.25	0.4	0.34192	0.34192	0.34192	0.34191
	0.6	0.26897	0.26897	0.26897	0.22896
	0.8	0.22148	0.22148	0.22148	0.22148
	1.0	0.18819	0.18819	0.18819	0.18819
	3.0	0.07511	0.07511	0.07511	0.07511
0.50	0.4	0.66071	0.66071	0.66071	0.66071
	0.6	0.52942	0.52942	0.52942	0.52942
	0.8	0.43914	0.43914	0.43914	0.43914
	1.0	0.37442	0.37442	0.37442	0.37442
	3.0	0.15018	0.15018	0.15018	0.15018
0.75	0.4	0.91027	0.91027	0.91027	0.91026
	0.6	0.76725	0.76725	0.76724	0.76724
	0.8	0.64740	0.64740	0.64740	0.64740
	1.0	0.55605	0.55605	0.55605	0.55605
	3.0	0.22483	0.22483	0.22481	0.22481

series as a result of which these results cannot be compared with the exact solutions. It follows from Tables 2 and 3 that the accuracy of the presented solutions is almost the same as the accuracy of both cubic B-spline collocation method and cubic B-spline Galerkin method. As the value of the space variable decreases, the error becomes lower than for the cubic B-spline collocation methods and is quite close to the error of the cubic B-spline Galerkin method. The values of the solution are presented in Table 3 at time $t = 0.1$.

Table 3

Numerical Results for $p = 1,\ t = 0.1,\ \lambda = 1,\ \Delta t = 0.0001$, and Different Sizes

x	h	Present	Ref. [12]	Ref. [13]	Exact
0.1	0.0125	0.10953	0.10952	0.10954	0.10954
0.2		0.20977	0.20975	0.20979	0.20979
0.3		0.29186	0.29184	0.29189	0.29190
0.4		0.34788	0.34788	0.34792	0.34792
0.5		0.37153	0.37153	0.37158	0.37158
0.6		0.35899	0.35896	0.35904	0.35905
0.7		0.30986	0.30983	0.30990	0.30991
0.8		0.22778	0.22776	0.22782	0.22782
0.9		0.12067	0.12065	0.12069	0.12069
0.1	$h = 0.0625$	0.10954	0.10953	0.10954	0.10954
0.2		0.20979	0.20977	0.20979	0.20979
0.3		0.29189	0.29186	0.29190	0.29190
0.4		0.34792	0.34788	0.34792	0.34792
0.5		0.37156	0.37153	0.37158	0.37158
0.6		0.35903	0.35900	0.35904	0.35905
0.7		0.30989	0.30986	0.30990	0.30991
0.8		0.22781	0.22778	0.22782	0.22782
0.9		0.12068	0.12067	0.12069	0.12069

(b) As the second example, we consider a particular solution of Burgers' equation with the initial condition

$$U(x,1) = \exp\left(\frac{1}{8\lambda}\right), \quad 0 \le x \le 1,$$

and boundary conditions $U(0,t) = 0$ and $U(1,t) = 0$.

This problem has the following analytic solution:

$$U(x,t) = \frac{\frac{x}{t}}{1 + \sqrt{\frac{t}{t_0}} \exp\left(\frac{x^2}{4\lambda t}\right)}, \qquad t \ge 1, \quad 0 \le x \le 1. \tag{13}$$

This solution reflects the propagation of a shock, and the choice of smaller λ gives a steep shock solution. Hence, the success of the numerical method depends on the possibility of efficient analysis of steep shocks.

Table 4

Numerical Results for $p = 1, \lambda = 0.0005,\ h = 0.005,\ \Delta t = 0.01$, and Different Times

x	Times	Present	Ref. [12]	Exact
0.1	1.7	0.05882	0.05883	0.05882
0.2		0.11765	0.11765	0.11765
0.3		0.17647	0.17648	0.17647
0.4		0.23529	0.23531	0.23529
0.5		0.29412	0.29414	0.29412
0.6		0.35294	0.35296	0.35294
0.7		0.00000	0.00000	0.00000
0.8		0.00000	0.00000	0.00000
0.9		0.00000	0.00000	0.00000
0.1	2.5	0.04000	0.04000	0.04000
0.2		0.08000	0.08000	0.08000
0.3		0.12000	0.12001	0.12000
0.4		0.16000	0.16001	0.16000
0.5		0.20000	0.20001	0.20000
0.6		0.24000	0.24001	0.24000
0.7		0.28000	0.28001	0.28000
0.8		0.00828	0.00811	0.00977
0.9		0.00000	0.00000	0.00000
0.1	3.25	0.03077	0.03077	0.03077
0.2		0.06154	0.06154	0.06154
0.3		0.09231	0.09231	0.09231
0.4		0.12308	0.12308	0.12308
0.5		0.15385	0.15385	0.15385
0.6		0.18462	0.18462	0.18462
0.7		0.21538	0.21539	0.21538
0.8		0.24615	0.24616	0.24615
0.9		0.12394	0.12358	0.12435

The process of propagation of shocks is studied for $\lambda = 0.005,\ 0.0005$. The numerical solutions obtained by the exponential collocation method can be favorably compared with the results reported in [12, 13] for certain times, as indicated in Table 4. In Figs. 10 and 11, we show the process of shock propagation for $\lambda = 0.005$, $h = 0.02,\ \Delta t = 0.1$ and $\lambda = 0.0005,\ h = 0.005,\ \Delta t = 0.01$ respectively. As time increases, the initial steep shock becomes smoother if the higher viscosity is used. However, for the low viscosity, it is steeper. These observations are in complete agreement with the results reported in [9].

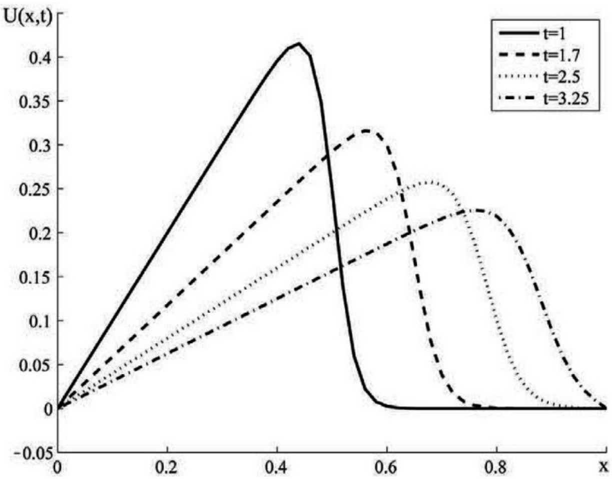

Fig. 10. Shock propagation; $\lambda = 0.005$.

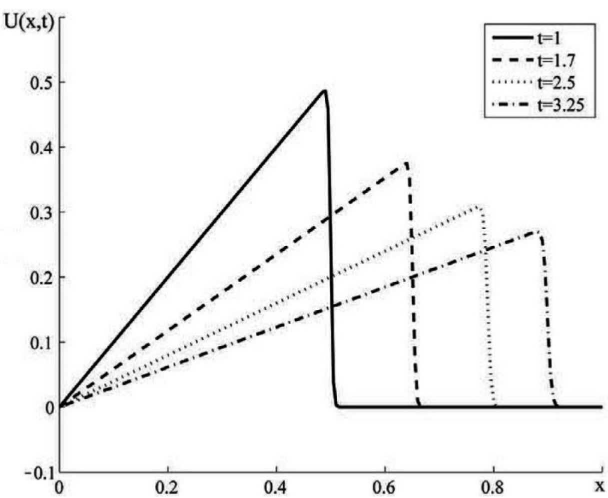

Fig. 11. Shock propagation; $\lambda = 0.0005$.

(c) Traveling-wave solution of Burgers' equation has the form:

$$U(x,t) = \frac{\alpha + \mu + (\mu - \alpha)\exp\eta}{1 + \exp\eta}, \qquad 0 \le x \le 1, \quad t \ge 0, \tag{14}$$

where

$$\eta = \frac{\alpha(x - \mu t - \gamma)}{\lambda},$$

and $\alpha,\ \mu$, and γ are arbitrary constants. The boundary conditions are

$$U(0,t) = 1, \qquad U(1,t) = 0.2$$

or

$$U_x(0,t) = 0, \qquad U_x(1,t) = 0 \qquad \text{for} \quad t \ge 0$$

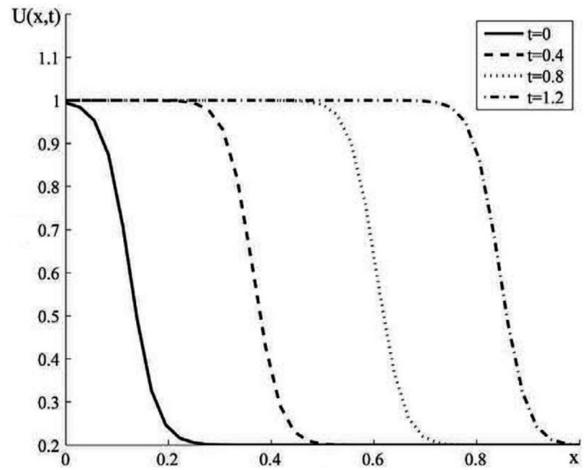

Fig. 12. Solutions for $\lambda = 0.01$.

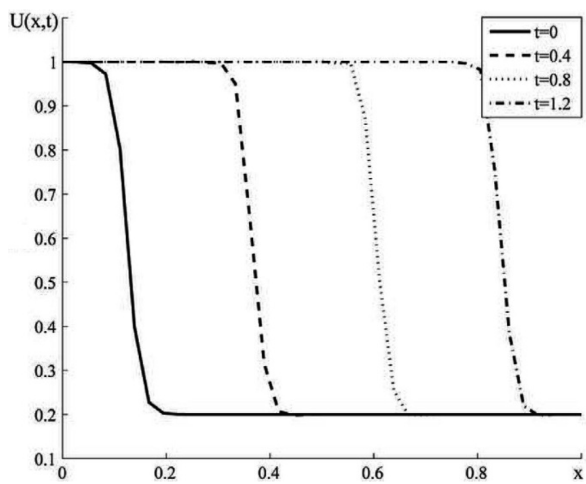

Fig. 13. Solutions for $\lambda = 0.005$.

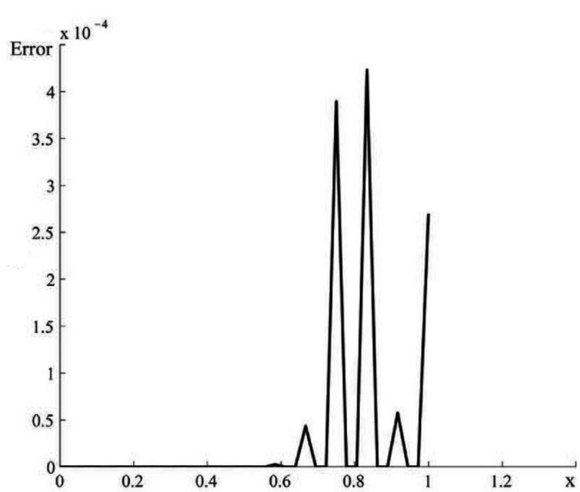

Fig. 14. L_2 error norm for $\lambda = 0.01$ the $t = 1.2$.

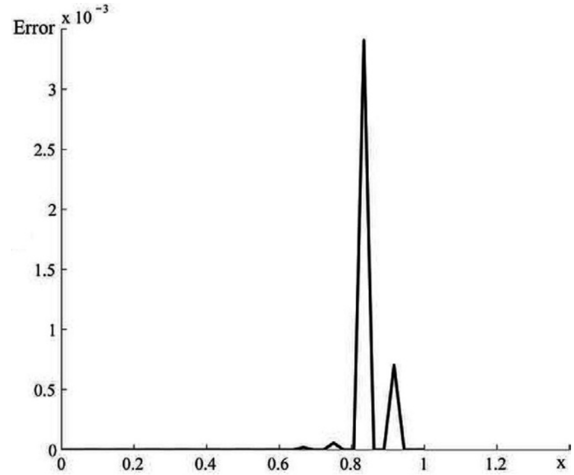

Fig. 15. L_2 error norm for $\lambda = 0.005$ the $t = 1.2$.

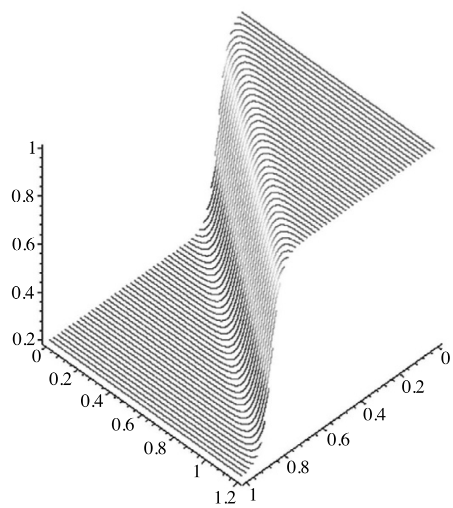

Fig. 16. Shock propagation; $\lambda = 0.01$.

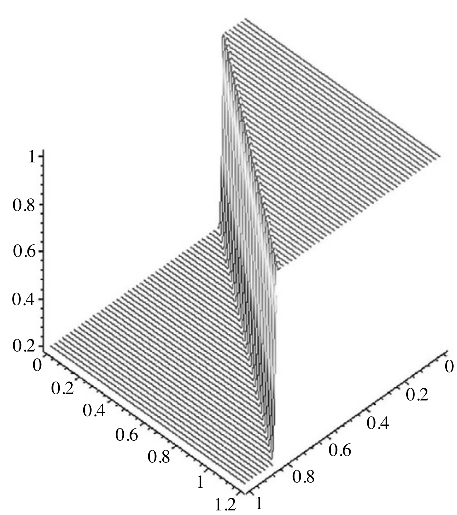

Fig. 17. Shock propagation; $\lambda = 0.005$.

and the initial condition is obtained from the analytic solution (14) for $t = 0$. The analytic solution takes values between 1 and 0.2, and the process of propagation of the wave front through the right is observed for variable λ. The lower the value of λ in Burgers' equation, the steeper the propagating wave front. The robustness of the algorithm can be shown by monitoring the motion of the wave front for lower λ. The algorithm was realized for the values $\alpha = 0.4$, $\mu = 0.6$, $\gamma = 0.125$ and $\lambda = 0.01$, $h = 1/36$, $\Delta t = 0.001$, and $p = 1$. The visual motion of the wave front is depicted in Figs. 12 and 13 for $\lambda = 0.01,\ 0.05$. The numerical results demonstrate the formation of a steep front and a very steep front. The plots of errors of the numerical solutions are also shown in Figs. 14 and 15. From the figures, it is clear that the maximum error is attained in the middle of the analyzed domain. The solutions obtained from the time $t = 0$ to $t = 1.2$ with certain time intervals are visualized in 3D plots to detect the propagation of sharp behaviors in Figs. 16 and 17 for $h = 1/80$ and $\Delta t = 0.0001$.

4. Conclusions

The exponential cubic B-spline collocation method for the numerical solutions of Burgers' equation is presented via the finite elements and, hence, the continuity of the dependent variable and its first two derivatives is satisfied for the approximate solution throughout the analyzed range. The equation is integrated into a system of the linearized iterative algebraic equations. The iterative system is solved by using the Thomas algorithm for each time step in which a three-banded matrix of coefficients is constructed. In general, the comparative results demonstrate that the data of our investigations are better than for the cubic B-spline collocation method and almost identical to the results obtained by the cubic B-spline Galerkin method. Since the cost of the cubic B-spline Galerkin method is higher than the cost of the suggested method, this is an advantage of the exponential cubic B-spline collocation method over the cubic B-spline Galerkin method. In all runs of the algorithm, the best results were obtained for the free parameter $p = 1$ and the exponential cubic B-spline functions.

REFERENCES

1. E. Hopf, "The partial differential equation $U_t + UU_x - \nu U_{xx} = 0$," *Comm. Pure Appl. Math.*, **3**, 201–230 (1950).
2. H. Bateman, "Some recent researches on the motion of fluids," *Monthly Weather Rev.*, **43**, 163–170 (1915).
3. S. G. Rubin and R. A. Graves, "Cubic spline approximation for problems in fluid mechanics, *Nasa Tech. Rep.*, R-436, Washington (1975).
4. S. G. Rubin and P. K. Khosla, "High-order numerical solutions using cubic splines," *AIAA J.*, **14**, No. 7, 851–858 (1976).
5. A. M. Davies, "A numerical investigation of errors arising in applying the Galerkin method of the solution of nonlinear partial differential equations," *Comput. Methods Appl. Mech. Eng.*, **11**, 341–350 (1977).
6. A. M. Davies, "Application of the Galerkin method to the solution of Burgers equation," *Comput. Methods Appl. Mech. Eng.*, **14**, 305–321 (1978).
7. P. C. Jain and D. H. Holla, "Numerical solutions of coupled Burgers' equation," *Int. J. Nonlin. Mech.*, **13**, Issue 4, 213–222 (1978).
8. P. C. Jain and B. L. Lohar, "Cubic spline technique for coupled nonlinear parabolic equations," *Comput. Math. Appl.*, **5**, 179–185 (1979).
9. B. L. Lohar and P. C. Jain, "Variable mesh cubic spline technique for N-wave solution of Burgers' equation," *J. Comput. Phys.*, **39**, 433–442 (1981).
10. S. Kutluay, A. Esen, and I. Dağ, "Numerical solutions of the Burgers' equation by the least squares quadratic B-spline finite element method," *J. Comput. Appl. Math.*, **167**, 21–33 (2004).
11. T. Öziş, A. Esen, and S. Kutluay, "Numerical solution of Burgers equation by quadratic B-spline finite elements," *Appl. Math. Comput.*, **165**, 237–249 (2005).
12. İ. Dağ, D. Irk, and B. Saka, "A numerical solution of the Burgers' equation using cubic B-splines," *Appl. Math. Comput.*, **163**, 199–211 (2005).
13. İ. Dağ, B. Saka, and A. Boz, "B-spline Galerkin methods for numerical solutions of the Burgers' equation," *Appl. Math. Comput.*, **166**, 506–522 (2005).
14. E. N. Aksan, "Quadratic B-spline finite element method for numerical solution of the Burgers equation," *Appl. Math. Comput.*, **174**, 884–896 (2006).
15. D. Cole, "On a quasilinear parabolic equation occurring in aerodynamics," *Quart. Appl. Math.*, **9**, 225–236 (1951).

16. B. Saka and İ. Dağ, "Quartic B-spline collocation methods to the numerical solutions of the Burgers' equation," *Chaos Solitons Fractals*, **32**, 1125–1137 (2007).
17. B. Saka and İ. Dağ, "A numerical study of Burgers' equation," *J. Franklin Inst.*, **345**, 328–348 (2008).
18. B. Sepehrian and M. Lashani, "A numerical solution of the Burgers equation using quintic B-splines," in: *Proc. of the World Congr. on Engineering, 2008*, **3**, Newswood Limited, London (2008).
19. İ. Dağ, A. Canıvar, and A. Şahin, "Taylor–Galerkin and Taylor-collocation methods for the numerical solutions of Burgers' equation using B-splines," *Comm. Nonlin. Sci. Numer. Simul.*, **16**, 2696–2708 (2011).
20. R. H. Wang, M. Xi, J. H. Zhang, and Q. Fang, "A novel numerical scheme for solving Burgers' equation," *Appl. Math. Comput.*, **217**, 4473–4482 (2011).
21. A. Korkmaz, A. M. Aksoy, and I. Dag, "Quartic B-spline differential quadrature method," *Int. J. Nonlin. Sci.*, **11**, No. 4, 403–411 (2011).
22. H. V. Chapani, V. H. Pradhan, and M. N. Mehta, "Numerical simulation of Burger's equation using quadratic B-splines," *Int. J. Appl. Math. Mech.*, **8**, No. 11, 18–32 (2012).
23. R. C. Mittal and R. K. Jain, "Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method," *Appl. Math. Comput.*, **218**, 7839–7855 (2012).
24. C.-C. Wang, W.-J. Liao, and Y.-S. Hsu, "Hybrid spline difference method for the Burgers' equation," *Appl. Math. Comput.*, **219**, 1031–1039 (2012).
25. A. Korkmaz and I. Dag, "Cubic B-spline differential quadrature methods and stability for Burgers' equation," *Eng. Comput.*, **30**, No. 3, 320–344 (2013).
26. G. Arora and B. K. Singh, "Numerical solution of Burgers' equation with modified cubic B-spline differential quadrature method," *Appl. Math. Comput.*, **224**, 166–177 (2013).
27. B. J. McCartin, "Theory of exponential splines," *J. Approx. Theory*, **66**, 1–23 (1991).
28. M. Sakai and R. A. Usmani, "A class of simple exponential B-splines and their application to numerical solution to singular perturbation problems," *Numer. Math.*, **55**, 493–500 (1989).
29. D. Radunovic, "Multiresolution exponential B-splines and singularly perturbed boundary problem," *Numer. Algorithms*, **47**, 191–210 (2008).
30. S. Chandra Sekhara Rao and M. Kumar, "Exponential B-spline collocation method for self-adjoint singularly perturbed boundary value problems," *Appl. Numer. Math.*, **58**, 1572–1581 (2008).
31. R. Mohammadi, "Exponential B-spline solution of convection-diffusion equations," *Appl. Math.*, **4**, 933–944 (2013).
32. I. Dag and O. Ersoy, "The exponential cubic B-Spline algorithm for equal width equation," *Adv. Stud. Contemp. Math.*, **25**, No. 34, 525–535 (2015).
33. O. Ersoy and I. Dag, "The exponential cubic B-Spline algorithm for Korteweg–de Vries (KdV) equation," *Adv. Numer. Anal.* (2015).
34. I. Dag and O. Ersoy, "The exponential cubic B-spline algorithm for Fisher equation," *Chaos Solitons Fractals*, **86**, 101–106 (2016).
35. O. Ersoy and I. Dag, "The exponential cubic B-spline collocation method for the Kuramoto–Sivashinsky(KS) equation," *Filomat*, **30**, No. 3, 853–861 (2016).