Journal of Mathematical Modeling
Vol. 7, No. 4, 2019, pp. 445-467

JMM

Galerkin finite element method for forced Burgers' equation

Sarita Thakar and Sunil S Kumbhar*

Department of Mathematics, Shivaji University, Kolhapur (Maharashtra), India
email: sunilmath25@gmail.com, saritakolhapur@gmail.com

Abstract. In this paper second order explicit Galerkin finite element method based on cubic B-splines is constructed to compute numerical solutions of one dimensional nonlinear forced Burgers' equation. Taylor series expansion is used to obtain time discretization. Galerkin finite element method is set up for the constructed time discretized form. Stability of the corresponding linearized scheme is studied by using von Neumann analysis. The accuracy, efficiency, applicability and reliability of the present method is demonstrated by comparing numerical solutions of some test examples obtained by the proposed method with the exact and numerical solutions available in literature.

Keywords: Forced Burgers' equation, cubic B-splines, Galerkin Finite Element Method, Taylor series, von Neumann analysis.
AMS Subject Classification: 65M60, 65N30.

1 Introduction

The Burgers' equation is a one dimensional form of Navier-Stokes equation. It was firstly introduced by Harry Batman and was taken later by J.M. Burger as a model of turbulent fluid motion. This equation arises in various fields such as Fluid Dynamics, Nonlinear Acoustics, Gas Dynamics, Traffic Flow, etc. In this paper, we consider the one dimensional nonlinear forced Burgers' equation,

$$u_t + uu_x - \nu u_{xx} = F(x,t), \quad a \le x \le b, \quad t > 0, \tag{1}$$

*Corresponding author.
Received: 13 May 2019 / Revised: 26 July 2019 / Accepted: 24 October 2019.
DOI: 10.22124/jmm.2019.13259.1265

 http://jmm.guilan.ac.ir

with the initial condition

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

and the boundary conditions,

$$u(a,t) = g_1(t), \quad u(b,t) = g_2(t), \quad t > 0, \tag{3}$$

where $\nu > 0$ is the coefficient of kinematic viscosity, $f(x)$, $g_1(t)$, $g_2(t)$ and the forcing term $F(x,t)$ are known functions. The control function $F(x,t)$ is assumed to be differentiable with respect to time. In the literature one can see that, several numerical methods are available for Burgers' equation with $F(x,t) = 0$. The Galerkin finite element method based on cubic B-splines is constructed by [2] to obtain numerical solutions of Eq. (1) with $F(x,t) = 0$. This method is implicit and unconditionally stable. Time-splitting of homogeneous form of Eq. (1) is done by [6] to obtain system of partial differential equations. Galerkin finite element methods based on quadratic and cubic B-splines are constructed to obtain numerical solution of the splitting system. Cubic B-spline and modified cubic B-spline collocation methods are discussed by [7] and [12] respectively. Refs. [15] and [16] set up least squares algorithms with cubic and quadratic B-splines. Ref. [5] converted (1) with $F(x,t) = 0$ to a system of nonlinear ordinary differential equations by method of discretization in time and space. Quadratic B-spline Galerkin finite element method is employed on the resulting system. Weighted avarage differential quadrature method is developed by [11]. The Crank-Nicolson type finite difference method for Eq. (1) with $F(x,t) = 0$ is discussed by [14]. In constructing all the methods discussed above, the homogeneous form of Eq. (1) is converted into system of ordinary differential equations. The solution to this system of ordinary differential equations is obtained by constructing first or second order finite difference schemes.

The numerical solutions of Eq. (1) based on multiquadratic quasi-interpolation operator and radial basis function network schemes are obtained by [9]. In these methods the solution or its space derivative is quasi interpolated by using Hardy basis functions. Both the methods are conditionally stable. Stability of both the methods depends upon the shape parameters and the number of collocation points.

In the present paper we propose unconditionally stable cubic B-spline Galerkin finite element method for Eqs. (1)-(3). The time discretization of Eq. (1) is considered at the beginning. The recurrence relation is obtained by using forward difference approximations. Galerkin finite element method is then applied to construct a solution. The cubic B-splines has small support and therefore many elements of the matrices in the final assembled system of Galerkin method are zero. In fact matrices in the final assembled system are septadiagonal and computations with these matrices requires less computational cost. On the other hand Lagrange polynomials with the Gauss-Legendre points are defined on the whole domain.

The paper is organized as follows. In Section 2 second order finite difference scheme is constructed and Galerkin finite element method is applied to this second order finite difference scheme. Stability analysis of the corresponding linearized method is discussed in Section 3. Numerical solution of some test examples obtained by proposed method are reported in Section 4. These solutions are compared with exact solutions and numerical solutions available in the literature.

2 Method of solution

The domain $[a, b]$ is partitioned uniformly as $a = x_0 < x_1 < x_2 < \ldots < x_N = b$ into N number of finite elements with equal length $h = (b-a)/N$ and $x_j = x_0 + jh$, $j = 0, 1, 2, \ldots, N$. The time discretization of Eq. (1) is obtained by using following forward second order Taylor series formula.

$$u_t^n = \frac{u^{n+1} - u^n}{\Delta t} - \frac{\Delta t}{2} u_{tt}^n, \tag{4}$$

where $t_n = t_0 + n\Delta t$, $u^n = u(x, t_n), u_t^n = u_t(x, t_n)$ and $u_{tt}^n = u_{tt}(x, t_n)$. The discretization of Eq. (1) with $F(x, t) = 0$ is obtained earlier by [3]. Differentiating Eq. (1) w.r.t. t we get,

$$u_{tt} = -\partial_x(uu_t) + \nu\partial_x^2 u_t + F_t(x, t), \tag{5}$$

where ∂_x and ∂_x^2 denote the first and second order partial derivatives with respect to x. Substitution of u_t and u_{tt} from Eqs. (1) and (5) respectively into Eq. (4) gives

$$\begin{aligned} &-u^n\partial_x u^n + \nu\partial_x^2 u^n + F(x, t_n) \\ &\qquad = \frac{(u^{n+1} - u^n)}{\Delta t} - \frac{\Delta t}{2}\left[-u_t^n\partial_x u^n - u^n\partial_x u_t^n + \nu\partial_x^2 u_t^n + F_t(x, t_n)\right]. \end{aligned}$$

Using forward difference approximation to u_t^n in above equation and after the simplification we obtain

$$\begin{aligned} &\left[1 - \frac{\Delta t}{2}\left(-\partial_x u^n - u^n\partial_x + \nu\partial_x^2\right)\right]\frac{(u^{n+1} - u^n)}{\Delta t} \\ &\qquad = -u^n\partial_x u^n + \nu\partial_x^2 u^n + F(x, t_n) + \frac{\Delta t}{2}F_t(x, t_n). \end{aligned} \tag{6}$$

Eq. (6) gives the recurrence relation

$$\left[1+\frac{\Delta t}{2}\left(\partial_x u^n + u^n\partial_x - \nu\partial_x^2\right)\right]u^{n+1} = \left[1+\frac{\nu\Delta t}{2}\partial_x^2\right]u^n + \Delta t F(x,t_n) + \frac{(\Delta t)^2}{2}F_t(x,t_n). \quad (7)$$

The truncation error is given by $(T.E.) = PDE - FDE$ [8]. From Eqs. (1) and (6) we have

$$\begin{aligned} T.E. = &\left\{u_t^n + u^n u_x^n - \nu u_{xx}^n - F(x,t_n)\right\} - \\ &\left\{\left[1-\frac{\Delta t}{2}\left(-\partial_x u^n - u^n\partial_x + \nu\partial_x^2\right)\right]\frac{(u^{n+1}-u^n)}{\Delta t} + u^n\partial_x u^n \right.\\ &\left. -\nu\partial_x^2 u^n - F(x,t_n) - \frac{\Delta t}{2}F_t(x,t_n)\right\} \\ = &-\frac{(\Delta t)^2}{12}\left(2u_{ttt}^n + 3u_x^n u_{tt}^n + 3u^n u_{xtt}^n - 3\nu u_{xxtt}^n\right) - \cdots . \end{aligned}$$

Therefore Eq. (7) is a second order explicit scheme in the variable t.

Assume that the solution $u(x,t)$ of the Burgers' equation (1) is of the form

$$u(x,t) = \sum_{m=-1}^{N+1} \delta_m(t)\phi_m(x), \quad (8)$$

where $\delta_m(t)$, $m=-1,0,1,\ldots,N+1$ are the time dependent functions to be determined and $\phi_m(x)$, $m=-1,0,1,\ldots,N+1$ are cubic B-splines given by [10]

$$\phi_m(x) = \frac{1}{h^3}\begin{cases} (x-x_{m-2})^3, & [x_{m-2},x_{m-1}], \\ h^3 + 3h^2(x-x_{m-1}) + 3h(x-x_{m-1})^2 \\ \quad -3(x-x_{m-1})^3, & [x_{m-1},x_m], \\ h^3 + 3h^2(x_{m+1}-x) + 3h(x_{m+1}-x)^2 \\ \quad -3(x_{m+1}-x)^3, & [x_m,x_{m+1}], \\ (x_{m+2}-x)^3, & [x_{m+1},x_{m+2}], \\ 0, & o.w., \end{cases} \quad (9)$$

Using boundary conditions (3) we obtain

$$\delta_{-1}(t) = g_1(t) - 4\delta_0(t) - \delta_1(t), \quad (10)$$

$$\delta_{N+1}(t) = g_2(t) - \delta_{N-1}(t) - 4\delta_N(t). \quad (11)$$

The solution given by Eq. (8) now becomes

$$u(x,t) = g_1(t)\phi_{-1}(x) + g_2(t)\phi_{N+1}(x) + \sum_{i=0}^{N} \delta_i(t) B_i(x), \tag{12}$$

where

$$\begin{aligned} B_0(x) &= \phi_0(x) - 4\phi_{-1}(x), \quad B_1(x) = \phi_1(x) - \phi_{-1}(x), \\ B_j(x) &= \phi_j(x), \quad \text{for } j = 2, 3, \ldots, N-2, \\ B_{N-1}(x) &= \phi_{N-1}(x) - \phi_{N+1}(x), \quad B_N(x) = \phi_N(x) - 4\phi_{N+1}(x). \end{aligned}$$

From Eq. (12) we have

$$u_x(x,t) = g_1(t)\phi'_{-1}(x) + g_2(t)\phi'_{N+1}(x) + \sum_{i=0}^{N} \delta_i(t) B'_i(x), \tag{13}$$

$$u_{xx}(x,t) = g_1(t)\phi''_{-1}(x) + g_2(t)\phi''_{N+1}(x) + \sum_{i=0}^{N} \delta_i(t) B''_i(x). \tag{14}$$

Define

$$h_i(x,t) = \begin{cases} g_1(t)\big[\phi_{-1}(x)B_i(x)\big]', & i = 0, 1, 2, \\ g_2(t)\big[\phi_{N+1}(x)B_i(x)\big]', & i = N-2, N-1, N, \\ 0, & \textit{otherwise}, \end{cases} \tag{15}$$

$$\begin{aligned} &R_1(x, t_n, t_{n+1}) = \big[g_1(t_n) - g_1(t_{n+1})\big]\phi_{-1}(x) \\ &+\frac{\Delta t}{2}\big[\nu(g_1(t_n) + g_1(t_{n+1}))\phi''_{-1}(x) - 2g_1(t_n)g_1(t_{n+1})\phi_{-1}(x)\phi'_{-1}(x)\big], \end{aligned} \tag{16}$$

$$\begin{aligned} &S_N(x, t_n, t_{n+1}) = \big[g_2(t_n) - g_2(t_{n+1})\big]\phi_{N+1}(x) \\ &+\frac{\Delta t}{2}\big[\nu(g_2(t_n) + g_2(t_{n+1}))\phi''_{N+1}(x) - 2g_2(t_n)g_2(t_{n+1})\phi_{N+1}(x)\phi'_{N+1}(x)\big]. \end{aligned} \tag{17}$$

Using Eqs. (12)-(16) in Eq. (7), on element $[x_0, x_1]$ we have

$$\begin{aligned} &\sum_{i=0}^{2}\Big[B_i(x) + \frac{\Delta t}{2}\big(h_i(x,t_n) + \sum_{j=0}^{2}\delta_j(t_n)(B_iB_j)' - \nu B''_i\big)\Big]\cdot\delta_i(t_{n+1}) \\ &\quad = \sum_{i=0}^{2}\Big[B_i(x) - \frac{\Delta t}{2}\big(h_i(x,t_{n+1}) - \nu B''_i(x)\big)\Big]\cdot\delta_i(t_n) + R_1 + \Delta t F(x,t_n) \\ &\qquad + \frac{(\Delta t)^2}{2}F_t(x,t_n). \end{aligned} \tag{18}$$

The solution $u(x,t)$ and its derivatives on the typical element $[x_l, x_{l+1}]$ for $l = 1, 2, \dots, N-2$ are

$$u(x,t) = \sum_{i=l-1}^{l+2} \delta_i(t) B_i(x), \tag{19}$$

$$u_x(x,t) = \sum_{i=l-1}^{l+2} \delta_i(t) B_i'(x), \tag{20}$$

$$u_{xx}(x,t) = \sum_{i=l-1}^{l+2} \delta_i(t) B_i''(x). \tag{21}$$

On $[x_l, x_{l+1}]$ for $l = 1, 2, \dots, N-2$, Eq. (7) becomes

$$\sum_{i=l-1}^{l+2} \Big[B_i(x) + \frac{\Delta t}{2} \big(\sum_{j=l-1}^{l+2} \delta_j(t_n)(B_i B_j)' - \nu B_i^{''} \big) \Big] \cdot \delta_i(t_{n+1})$$
$$= \sum_{i=l-1}^{l+2} \Big[B_i(x) + \frac{\nu \Delta t}{2} B_i^{''}(x) \Big] \cdot \delta_i(t_n) + \Delta t F(x, t_n) + \frac{(\Delta t)^2}{2} F_t(x, t_n). \tag{22}$$

On $[x_{N-1}, x_N]$ Eq. (17) is used in Eq. (7) to obtain

$$\sum_{i=N-2}^{N} \Big[B_i(x) + \frac{\Delta t}{2} \big(h_i(x, t_n) + \sum_{j=N-2}^{N} \delta_j(t_n)(B_i B_j)' - \nu B_i^{''} \big) \Big] \cdot \delta_i(t_{n+1})$$
$$= \sum_{i=N-2}^{N} \Big[B_i(x) - \frac{\Delta t}{2} \big(h_i(x, t_{n+1}) - \nu B_i^{''}(x) \big) \Big] \cdot \delta_i(t_n) + S_N + \Delta t F(x, t_n)$$
$$+ \frac{(\Delta t)^2}{2} F_t(x, t_n). \tag{23}$$

The element wise Galerkin weak formulation is obtained by the following procedure. On multiplying Eq. (18) by the weight function $B_k(x)$, $k = 0, 1, 2$ and integrating by parts on the interval $[x_0, x_1]$ we get

$$\Big[\boldsymbol{A_1} + \frac{\Delta t}{2} \big(\boldsymbol{h_1^n} - \boldsymbol{B_1} + \nu \boldsymbol{C_1} \big) \Big] \cdot \boldsymbol{\delta_1^{n+1}}$$
$$= \Big[\boldsymbol{A_1} - \frac{\Delta t}{2} \big(\boldsymbol{h_1^{n+1}} + \nu \boldsymbol{C_1} \big) \Big] \cdot \boldsymbol{\delta_1^n} + \boldsymbol{R_1^n} + \boldsymbol{F_1^n}, \tag{24}$$

where $\boldsymbol{\delta_1^n} = (\delta_0^n, \delta_1^n, \delta_2^n)^T$,

$$\boldsymbol{R_1^n} = \frac{140(g_1(n\Delta t) - g_1((n+1)\Delta t))}{h} \begin{bmatrix} 49 \\ 40 \\ 1 \end{bmatrix}$$

$$+\frac{\Delta t}{2}\left\{\frac{\nu(g_1(n\Delta t)+g_1((n+1)\Delta t))}{10h}\begin{bmatrix}51\\54\\3\end{bmatrix}+\frac{2g_1(n\Delta t)g_1((n+1)\Delta t))}{168}\begin{bmatrix}97\\70\\1\end{bmatrix}\right\},$$

$$\boldsymbol{F_1^n}=\Delta t\begin{bmatrix}\int_{x_0}^{x_1}F(x,n\Delta t)B_0(x)dx\\ \int_{x_0}^{x_1}F(x,n\Delta t)B_1(x)dx\\ \int_{x_0}^{x_1}F(x,n\Delta t)B_2(x)dx\end{bmatrix}+\frac{(\Delta t)^2}{2}\begin{bmatrix}\int_{x_0}^{x_1}F_t(x,n\Delta t)B_0(x)dx\\ \int_{x_0}^{x_1}F_t(x,n\Delta t)B_1(x)dx\\ \int_{x_0}^{x_1}F_t(x,n\Delta t)B_2(x)dx\end{bmatrix}.$$

In order to compute $\boldsymbol{F_1^n}$, $F(x,t)$ and $F_t(x,t)$ are evaluated at $t=n\Delta t$ and then $\int_{x_0}^{x_1}F(x,n\Delta t)B_idx$, $\int_{x_0}^{x_1}F_t(x,n\Delta t)B_idx$, $i=0,1,2$ are computed. We have also

$$\boldsymbol{A_1}=\frac{h}{140}\begin{bmatrix}476&644&56\\644&1088&128\\56&128&20\end{bmatrix},\quad \boldsymbol{C_1}=\frac{1}{10h}\begin{bmatrix}222&108&-24\\108&192&24\\-24&24&18\end{bmatrix},$$

$$\boldsymbol{h_1^n}=\frac{-g_1(n\Delta t)}{840}\begin{bmatrix}1235&1586&89\\758&1244&98\\-1&26&5\end{bmatrix},$$

and for $i,j=1,2,3$, $(i,j)^{th}$ element of matrix $\boldsymbol{B_1}$, $(\boldsymbol{B_1})_{ij}$ is computed by the formula

$$(\boldsymbol{B_1})_{ij}=\left(\int_{x_0}^{x_1}B_{j-1}B_0B'_{i-1}dx,\int_{x_0}^{x_1}B_{j-1}B_1B'_{i-1}dx,\int_{x_0}^{x_1}B_{j-1}B_2B'_{i-1}dx\right)\boldsymbol{\delta_1^n}.$$

Thus the elements of matrix $\boldsymbol{B_1}$ are

$$\begin{aligned}(\boldsymbol{B_1})_{11}&=\frac{1}{840}(280,-4292,-878)\boldsymbol{\delta_1^n},\quad (\boldsymbol{B_1})_{12}=\frac{-1}{840}(4292,13616,2264)\boldsymbol{\delta_1^n},\\ (\boldsymbol{B_1})_{13}&=\frac{-1}{840}(878,2264,380)\boldsymbol{\delta_1^n},\quad (\boldsymbol{B_1})_{21}=\frac{1}{840}(11944,13528,796)\boldsymbol{\delta_1^n},\\ (\boldsymbol{B_1})_{22}&=\frac{1}{840}(13528,17920,1408)\boldsymbol{\delta_1^n},\quad (\boldsymbol{B_1})_{23}=\frac{1}{840}(796,1408,160)\boldsymbol{\delta_1^n},\\ (\boldsymbol{B_1})_{31}&=\frac{1}{840}(2596,4828,610)\boldsymbol{\delta_1^n},\quad (\boldsymbol{B_1})_{32}=\frac{1}{840}(4828,10624,1600)\boldsymbol{\delta_1^n},\\ (\boldsymbol{B_1})_{33}&=\frac{1}{840}(610,1600,280)\boldsymbol{\delta_1^n}.\end{aligned}$$

Multiply Eq. (22) by the weight function $B_k(x)$, $k=l-1,l,l+1,l+2$ and integrate by parts on the interval $[x_l,x_{l+1}]$. Because the overall contribution of the terms $B_i(x)B_j(x)B_k(x)|_{x_l}^{x_{l+1}}$ and $B'_i(x)B_k(x)|_{x_l}^{x_{l+1}}$ vanishes in the assembled system, we exclude them from the final expression. Thus on the element $[x_l, x_{l+1}]$ we have

$$\begin{aligned}\Big[\boldsymbol{A_{l+1}} + \frac{\Delta t}{2}\big(\boldsymbol{B_{l+1}} + \nu \boldsymbol{C_{l+1}}\big)\Big] \cdot \boldsymbol{\delta_{l+1}^{n+1}} \\ = \Big[\boldsymbol{A_{l+1}} - \frac{\nu \Delta t}{2}\boldsymbol{C_{l+1}}\big)\Big] \cdot \boldsymbol{\delta_{l+1}^{n}} + \boldsymbol{F_{l+1}^{n}},\end{aligned} \tag{25}$$

where for $l = 1, 2, \ldots, N-2$, $\boldsymbol{\delta_{l+1}^{n}} = (\delta_{l-1}^n, \delta_l^n, \delta_{l+1}^n, \delta_{l+2}^n)^T$,

$$\boldsymbol{F_{l+1}^{n}} = \Delta t \begin{bmatrix} \int_{x_l}^{x_{l+1}} F(x, n\Delta t) B_{l-1}(x) dx \\ \int_{x_l}^{x_{l+1}} F(x, n\Delta t) B_{l}(x) dx \\ \int_{x_l}^{x_{l+1}} F(x, n\Delta t) B_{l+1}(x) dx \\ \int_{x_l}^{x_{l+1}} F(x, n\Delta t) B_{l+2}(x) dx \end{bmatrix} + \frac{(\Delta t)^2}{2} \begin{bmatrix} \int_{x_l}^{x_{l+1}} F_t(x, n\Delta t) B_{l-1}(x) dx \\ \int_{x_l}^{x_{l+1}} F_t(x, n\Delta t) B_{l}(x) dx \\ \int_{x_l}^{x_{l+1}} F_t(x, n\Delta t) B_{l+1}(x) dx \\ \int_{x_l}^{x_{l+1}} F_t(x, n\Delta t) B_{l+2}(x) dx \end{bmatrix},$$

$$\boldsymbol{A_{l+1}} = \frac{h}{140}\begin{bmatrix} 20 & 129 & 60 & 1 \\ 129 & 1188 & 933 & 60 \\ 60 & 933 & 1188 & 129 \\ 1 & 60 & 129 & 20 \end{bmatrix}, \quad \boldsymbol{C_{l+1}} = \frac{1}{10h}\begin{bmatrix} 18 & 21 & -36 & -3 \\ 21 & 102 & -87 & -36 \\ -36 & -87 & 102 & 21 \\ -3 & -36 & 21 & 18 \end{bmatrix}$$

and for $i, j = 1, 2, 3, 4; (i, j)^{th}$ element of matrix $\boldsymbol{B_{l+1}}$, $(\boldsymbol{B_{l+1}})_{ij}$ is computed by the formula

$$\begin{aligned}(\boldsymbol{B_{l+1}})_{ij} = \Bigg(\int_{x_l}^{x_{l+1}} B_{j+l-2} B_{l-1} B'_{i+l-2} dx, \int_{x_l}^{x_{l+1}} B_{j+l-2} B_{l} B'_{i+l-2} dx, \\ \int_{x_l}^{x_{l+1}} B_{j+l-2} B_{l+1} B'_{i+l-2} dx, \int_{x_l}^{x_{l+1}} B_{j+l-2} B_{l+2} B'_{i+l-2} dx \Bigg) \boldsymbol{\delta_{l+1}^{n}}.\end{aligned}$$

Thus the elements of matrix $\boldsymbol{B_{l+1}}$ are

$(\boldsymbol{B_{l+1}})_{11} = \frac{-1}{840}(280, 1605, 630, 5)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{12} = \frac{-1}{840}(1605, 10830, 5349, 108)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{13} = \frac{-1}{840}(630, 5349, 3468, 129)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{14} = \frac{-1}{840}(5, 108, 129, 10)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{21} = \frac{-1}{840}(150, 1305, 792, 21)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{22} = \frac{-1}{840}(1305, 17640, 17541, 1314)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{23} = \frac{-1}{840}(792, 17541, 25002, 2781)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{24} = \frac{-1}{840}(21, 1314, 2781, 420)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{31} = \frac{1}{840}(420, 2781, 1314, 21)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{32} = \frac{1}{840}(2781, 25002, 17541, 792)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{33} = \frac{1}{840}(1314, 17541, 17640, 1305)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{34} = \frac{1}{840}(21, 792, 1305, 150)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{41} = \frac{1}{840}(10, 129, 108, 5)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{42} = \frac{1}{840}(129, 3468, 5349, 630)\boldsymbol{\delta_{l+1}^{n}}$,

$(\boldsymbol{B_{l+1}})_{43} = \frac{1}{840}(108, 5349, 10830, 1605)\boldsymbol{\delta_{l+1}^{n}}$, $(\boldsymbol{B_{l+1}})_{44} = \frac{1}{840}(5, 630, 1605, 280)\boldsymbol{\delta_{l+1}^{n}}$.

On multiplying (23) by the weight function $B_k(x)$, $k = N-2, N-1, N$

and integrating by parts on the interval $[x_{N-1}, x_N]$ we get

$$\Big[\boldsymbol{A_N} + \frac{\Delta t}{2}\big(\boldsymbol{h_N^n} - \boldsymbol{B_N} + \nu \boldsymbol{C_N}\big)\Big] \cdot \boldsymbol{\delta_N^{n+1}} = \Big[\boldsymbol{A_N} - \frac{\Delta t}{2}\big(\boldsymbol{h_N^{n+1}} + \nu \boldsymbol{C_N}\big)\Big] \cdot \boldsymbol{\delta_N^n} + \boldsymbol{S_N^n} + \boldsymbol{F_N^n}, \tag{26}$$

where $\boldsymbol{\delta_N^n} = (\delta_{N-2}^n, \delta_{N-1}^n, \delta_N^n)^T$,

$$\boldsymbol{S_N^n} = \frac{140(g_2(n\Delta t) - g_2((n+1)\Delta t))}{h}\begin{bmatrix}1\\40\\49\end{bmatrix}$$

$$+\frac{\Delta t}{2}\left\{\frac{\nu(g_2(n\Delta t) + g_2((n+1)\Delta t))}{10h}\begin{bmatrix}3\\54\\51\end{bmatrix} - \frac{2g_2(n\Delta t)g_2((n+1)\Delta t))}{168}\begin{bmatrix}1\\70\\97\end{bmatrix}\right\},$$

$$\boldsymbol{F_N^n} = \Delta t\begin{bmatrix}\int_{x_{N-1}}^{x_N} F(x, n\Delta t)B_{N-2}(x)dx\\ \int_{x_{N-1}}^{x_N} F(x, n\Delta t)B_1(x)dx\\ \int_{x_{N-1}}^{x_N} F(x, n\Delta t)B_2(x)dx\end{bmatrix} + \frac{(\Delta t)^2}{2}\begin{bmatrix}\int_{x_{N-1}}^{x_N} F_t(x, n\Delta t)B_{N-2}(x)dx\\ \int_{x_{N-1}}^{x_N} F_t(x, n\Delta t)B_1(x)dx\\ \int_{x_{N-1}}^{x_N} F_t(x, n\Delta t)B_2(x)dx\end{bmatrix},$$

$$\boldsymbol{A_N} = \frac{h}{140}\begin{bmatrix}20 & 128 & 56\\128 & 1088 & 644\\56 & 644 & 476\end{bmatrix}, \quad \boldsymbol{C_N} = \frac{1}{10h}\begin{bmatrix}18 & 24 & -24\\24 & 192 & 108\\-24 & 108 & 222\end{bmatrix},$$

$$\boldsymbol{h_N^n} = \frac{g_2(n\Delta t)}{840}\begin{bmatrix}5 & 26 & -1\\98 & 1244 & 758\\89 & 1586 & 1235\end{bmatrix},$$

and for $i, j = 1, 2, 3$, $(i, j)^{th}$ element of matrix $\boldsymbol{B_N}$, $(\boldsymbol{B_N})_{ij}$ is computed by the formula

$$(\boldsymbol{B_N})_{ij} = \left(\int_{x_{N-1}}^{x_N} B_{j+N-3}B_{N-2}B'_{i+N-3}dx, \int_{x_{N-1}}^{x_N} B_{j+N-3}B_{N-1}B'_{i+N-3}dx, \int_{x_{N-1}}^{x_N} B_{j+N-3}B_N B'_{i+N-3}dx\right)\boldsymbol{\delta_N^n}.$$

Thus the elements of matrix $\boldsymbol{B_N}$ are

$$\begin{aligned}
(\boldsymbol{B_N})_{11} &= \frac{-1}{840}(280, 1600, 610)\boldsymbol{\delta^n_N}, \quad (\boldsymbol{B_N})_{12} = \frac{-1}{840}(1600, 10624, 4828)\boldsymbol{\delta^n_N},\\
(\boldsymbol{B_N})_{13} &= \frac{-1}{840}(610, 4828, 2596)\boldsymbol{\delta^n_N}, \quad (\boldsymbol{B_N})_{21} = \frac{-1}{840}(160, 1408, 796)\boldsymbol{\delta^n_N},\\
(\boldsymbol{B_N})_{22} &= \frac{-1}{840}(1408, 17920, 13528)\boldsymbol{\delta^n_N}, \quad (\boldsymbol{B_N})_{23} = \frac{-1}{840}(796, 13528, 11944)\boldsymbol{\delta^n_N},\\
(\boldsymbol{B_N})_{31} &= \frac{1}{840}(380, 2264, 878)\boldsymbol{\delta^n_N}, \quad (\boldsymbol{B_N})_{32} = \frac{1}{840}(2264, 13616, 4292)\boldsymbol{\delta^n_N},\\
(\boldsymbol{B_N})_{33} &= \frac{1}{840}(878, 4292, -280)\boldsymbol{\delta^n_N}.
\end{aligned}$$

Since $\phi_{-1}(x)$ is zero on $[x_l, x_{l+1}]$, $l = 1, 2, \cdots, N-1$, $\boldsymbol{R^n_k}$, $k = 2, 3, \ldots, N$ are zero vectors. Similarly $\phi_{N+1}(x)$ is zero on $[x_l, x_{l+1}]$, $l = 0, 1, \ldots, N-2$, and therefore $\boldsymbol{S^n_k}$, $k = 1, 2, \ldots, N-1$ are zero vectors. Also for $k = 2, 3, \ldots, N-1$, $\boldsymbol{h^n_k}$ are zero matrices. Combining the contributions from Eqs. (24), (25) and (26) in usual way we obtain the system of $(N+1) \times (N+1)$ algebraic equations

$$\begin{aligned}
&\Big[\boldsymbol{A} + \frac{\Delta t}{2}\big(\boldsymbol{h}^n - \boldsymbol{B} + \nu\boldsymbol{C}\big)\Big] \cdot \boldsymbol{\delta}^{n+1}\\
&\quad = \Big[\boldsymbol{A} - \frac{\Delta t}{2}\big(\boldsymbol{h}^{n+1} + \nu\boldsymbol{C}\big)\Big] \cdot \boldsymbol{\delta}^n + \boldsymbol{R}^n + \boldsymbol{S}^n + \boldsymbol{F}^n, \qquad (27)
\end{aligned}$$

where $\boldsymbol{\delta}^n = (\delta^n_0, \delta^n_1, \ldots, \delta^n_N)^T$. The initial solution $\boldsymbol{\delta}^0$ is obtained from the initial condition (2). Since $\boldsymbol{\delta}^0$ has $(N+1)$ components, the system of $(N+1)$ equations is obtained by evaluating (12) at distinct points $x = y_j \in (a, b)$, $j = 0, 1, \ldots, N$ and $t = 0$. Thus we have

$$u(y_j, 0) = f(y_j) = g_1(0)\phi_{-1}(y_j) + g_2(0)\phi_{N+1}(y_j) + \sum_{i=0}^{N} \delta^0_i B_i(y_j). \qquad (28)$$

The solution of this system of equations is the initial solution $\boldsymbol{\delta}^0$. The recurrence relation (27) generates $\boldsymbol{\delta}^{n+1}$ and the solution of Eqs. (1)-(3) at $t = t_{n+1}$ is given by (12) as

$$u(x, t_{n+1}) = g_1(t_{n+1})\phi_{-1}(x) + g_2(t_{n+1})\phi_{N+1}(x) + \sum_{i=0}^{N} \delta^{n+1}_i B_i(x). \qquad (29)$$

3 Stability analysis

Since problem (1)-(3) is nonlinear, its discretization (7) and the algebraic scheme (27) are nonlinear. To study the stability of scheme (27), the corresponding linearized scheme is considered for von Neumann analysis. The linearized form of (27) is obtained by assuming that the solution u in the nonlinear term uu_x is locally constant and is equal to U. Thus the linear system corresponding to scheme (27) is

$$\left[\boldsymbol{A} + \frac{U\Delta t}{2}\mathbf{B}^* + \frac{\nu\Delta t}{2}\boldsymbol{C}\right] \cdot \boldsymbol{\delta}^{n+1} = \left[\boldsymbol{A} - \frac{\nu\Delta t}{2}\boldsymbol{C}\right] \cdot \boldsymbol{\delta}^{n} + \boldsymbol{R}^n + \boldsymbol{S}^n + \boldsymbol{F}^n, \quad (30)$$

where $\mathbf{B}^*$ is obtained by combining contributions from $\int_{x_l}^{x_{l+1}} B_i'(x)B_j(x)dx$ in usual way. The error equation corresponding to above equation is

$$\left[\boldsymbol{A} + \frac{U\Delta t}{2}\mathbf{B}^* + \frac{\nu\Delta t}{2}\boldsymbol{C}\right] \cdot \boldsymbol{\epsilon}^{n+1} = \left[\boldsymbol{A} - \frac{\nu\Delta t}{2}\boldsymbol{C}\right] \cdot \boldsymbol{\epsilon}^{n}, \quad (31)$$

where, ϵ^n is error in the solution at $t = t_n$. Matrices $\boldsymbol{A}, \boldsymbol{B}^*$ and $\boldsymbol{C}$ are septadiagonal matrices and general row of these matrices are

$$\begin{aligned}
\boldsymbol{A} &: \frac{h}{140}(1, 120, 1191, 2416, 1191, 120, 1)\\
\boldsymbol{B^*} &: \frac{1}{20}(1, 56, 245, 0, -245, -56, -1)\\
\boldsymbol{C} &: \frac{-1}{10h}(3, 72, 45, -240, 45, 72, 3)
\end{aligned}$$

The l^{th} error equation in (31) is

$$\begin{aligned}
&\alpha_1\epsilon_{l-3}^{n+1} + \alpha_2\epsilon_{l-2}^{n+1} + \alpha_3\epsilon_{l-1}^{n+1} + \alpha_4\epsilon_{l}^{n+1} + \alpha_5\epsilon_{l+1}^{n+1} + \alpha_6\epsilon_{l+2}^{n+1} + \alpha_7\epsilon_{l+3}^{n+1}\\
&\quad = \alpha_8\epsilon_{l-3}^{n} + \alpha_9\epsilon_{l-2}^{n} + \alpha_{10}\epsilon_{l-1}^{n} + \alpha_{11}\epsilon_{l}^{n} + \alpha_{10}\epsilon_{l+1}^{n} + \alpha_9\epsilon_{l+2}^{n} + \alpha_8\epsilon_{l+3}^{n} \quad (32)
\end{aligned}$$

where ϵ_j^n is the error in δ_j at $t = n\Delta t$, $0 \le j \le N, 4 \le l \le N-3$,

$$\begin{aligned}
\alpha_1 &= r_1 + r_2 - 3r_3, \quad \alpha_2 = 120r_1 + 56r_2 - 72r_3,\\
\alpha_3 &= 1191r_1 + 245r_2 - 45r_3, \quad \alpha_4 = 2416r_1 + 240r_3,\\
\alpha_5 &= 1191r_1 - 245r_2 - 45r_3, \quad \alpha_6 = 120r_1 - 56r_2 - 72r_3,\\
\alpha_7 &= r_1 - r_2 - 3r_3, \quad \alpha_8 = r_1 + 3r_3, \quad \alpha_9 = 120r_1 + 72r_3,\\
\alpha_{10} &= 1191r_1 + 45r_3, \quad \alpha_{11} = 2416r_1 - 240r_3,\\
r_1 &= \frac{h}{140}, \quad r_2 = \frac{U\Delta t}{40}, \quad r_3 = \frac{\nu\Delta t}{20h}.
\end{aligned}$$

The Fourier growth factor is defined as

$$\epsilon_l^n = \xi^n e^{ilkh} \quad (33)$$

where k is mode number and h is the length of finite element. Using (33) Eq. (32) gives

$$[(a-c)-ib]\xi^{n+1}=[a+c]\xi^{n}, \tag{34}$$

where

$$\begin{aligned}
a &= [2\cos(3kh)+240\cos(2kh)+2382\cos(kh)+2416]\, r_1,\\
b &= [2\sin(3kh)+112\sin(2kh)+490\sin(kh)]\, r_2,\\
c &= [6\cos(3kh)+144\cos(2kh)+90\cos(kh)-240]\, r_3.
\end{aligned}$$

From Eq. (34) the amplification factor

$$\xi=\frac{a+c}{(a-c)-ib}.$$

Since $r_3>0$, $c\leq 0$ and hence $|\xi|\leq 1$ and therefore the linearized scheme (30) is unconditionally stable.

4 Numerical experiments

In this section we illustrate seven test examples to support the method. Mathematica 10.0 software is used to compute numerical solutions and errors in it. The L_2 and L_∞ errors are defined as,

$$L_2=\sqrt{h\sum_{j=0}^{N}|U_j^{exact}-u_j^{numer}|^2},\qquad L_\infty=\max_{1\leq j\leq N}|U_j^{exact}-u_j^{numer}|$$

where, U_j^{exact} and u_j^{numer} are exact and numerical solutions at $x=x_j$ respectively.

Example 1. In this test example we consider Eq. (1) with the initial condition $u(x,0)=\sin(\pi x)$, boundary conditions $u(0,t)=0, u(1,t)=0$ and $F(x,t)=0$. The exact solution of this problem is

$$u(x,t)=2\pi\nu\frac{\sum_{n=1}^{\infty}a_n e^{-n^2\pi^2\nu t}n\sin(n\pi x)}{a_0+\sum_{n=1}^{\infty}a_n e^{-n^2\pi^2\nu t}n\cos(n\pi x)}$$

where the Fourier coefficients a_n, $n=0,1,2,\ldots$, are given by

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

Table 1: Comparison of numerical and exact solution for $\nu = 0.01$ of Example 1.

x	t	[6] (CBGM) $\Delta t = 0.0001$ $h = 0.0125$	[7] $\Delta t = 0.0001$ $h = 0.0125$	Present $\Delta t = 0.0001$ $h = 0.0125$	Exact
0.25	0.4	0.34191	0.34192	0.34192	0.34191
	0.6	0.26896	0.26897	0.26897	0.26896
	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	0.76724	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	0.22481	0.22483	0.22481	0.22481

The numerical solutions obtained by the proposed method, solutions obtained in [6, 7, 12, 15] and the exact solution for $\nu = 0.01$ and 0.1 are shown in Table 1 and Table 2 respectively. From Table 1 and Table 2, it is observed that the numerical solutions obtained by the proposed method are compatible with numerical solutions available in the literature for $\nu = 0.01$, whereas for $\nu = 0.1$ numerical solutions obtained by proposed method are better than the solutions obtained in [6, 12, 15] even for large value of Δt and h.

Example 2. Consider Eq. (1) with initial condition $u(x, 0) = 4x(1 - x)$ and boundary conditions $u(0, t) = 0$; $u(1, t) = 0$ with $F(x, t) = 0$. The numerical solution obtained by the proposed method, solution obtained in [5, 12, 16] and exact solution for $\nu = 0.01$ are listed in Table 3. It is observed that the numerical solutions obtained by the proposed method are better than the solutions obtained in [5, 16] even for large values of Δt and h. The obtained numerical solutions are slightly less accurate than solutions listed in [12] in the neighborhood of shock.

Table 2: Comparison of numerical and exact solution for $\nu = 0.1$ of Example 1.

x	t	[12] $\Delta t = 0.0025$ $h = 0.025$	[15] $\Delta t = 0.0001$ $h = 0.0125$	[6] (CBGM) $\Delta t = 0.0001$ $h = 0.0125$	Present $\Delta t = 0.0025$ $h = 0.025$	Exact
0.25	0.4	0.30892	0.30752	0.30890	0.30889	0.30889
	0.6	0.24077	0.24042	0.24074	0.24074	0.24074
	0.8	0.19572	0.19555	0.19568	0.19568	0.19568
	1.0	0.16261	0.16234	0.16257	0.16257	0.16256
	3.0	0.02718	-	0.02720	0.02720	0.02720
0.50	0.4	0.56970	0.55953	0.56964	0.56963	0.56963
	0.6	0.44729	0.44797	0.44721	0.44721	0.44721
	0.8	0.35930	0.35739	0.35924	0.35924	0.35924
	1.0	0.29195	0.29134	0.29191	0.29192	0.29192
	3.0	0.04016	-	0.04020	0.04020	0.04021
0.75	0.4	0.62520	0.64561	0.62541	0.62544	0.62544
	0.6	0.48694	0.48267	0.48719	0.48722	0.48721
	0.8	0.37365	0.37533	0.37390	0.37392	0.37392
	1.0	0.28724	0.28585	0.28746	0.28747	0.28747
	3.0	0.02974	-	0.02977	0.02977	0.02977

Example 3. In this example we consider Eq. (1) with the initial condition $u(x,0) = \frac{2\nu\pi\sin(\pi x)}{\alpha+\cos(\pi x)}$; $\alpha > 1$, boundary conditions $u(0,t) = 0$; $u(1,t) = 0$ and $F(x,t) = 0$. The exact solution of this problem is

$$u(x,t) = \frac{2\nu\pi e^{-\pi^2\nu t}\sin(\pi x)}{\alpha + e^{-\pi^2\nu t}\cos(\pi x)}.$$

The numerical solutions and L_2 and L_∞ errors for $\alpha = 2, h = 0.025, \nu = 1.0, 0.5, 0.2, 0.1$ and $\Delta t = 0.0001$ at $t = 0.001$ are shown in Table 4 and Table 5. From Table 4 and Table 5, it is seen that for $\nu = 0.5, 1.0$ the numerical solutions obtained by the present method are much better than solutions obtained in [12] whereas for $\nu = 0.1, 0.2$ they are better than solutions in [12].

Example 4. Consider Eq. (1) with the initial condition

$$u(x,1) = \frac{x}{1 + e^{\frac{1}{4\nu}}(x^2 - \frac{1}{4})},$$

and boundary conditions $u(0,t) = u(1.2,t) = 0$ and $F(x,t) = 0$. The exact

Table 3: Comparison of numerical and exact solution for $\nu = 0.01$ of Example 2.

x	t	[12] $\Delta t = 0.001$ $h = 0.025$	[5] $\Delta t = 0.0001$ $h = 0.025$	[16] $\Delta t = 0.0001$ $h = 0.0125$	Present $\Delta t = 0.001$ $h = 0.025$	Exact
0.25	0.4	0.36225	0.36225	0.36218	0.36226	0.36226
	0.6	0.28202	0.28199	0.28197	0.28204	0.28204
	0.8	0.23044	0.23039	0.23040	0.23045	0.23045
	1.0	0.19468	0.19463	0.19465	0.19469	0.19469
	3.0	0.07613	0.07611	0.07613	0.07613	0.07613
0.50	0.4	0.68368	0.68371	0.68364	0.68368	0.68368
	0.6	0.54832	0.54835	0.54829	0.54832	0.54832
	0.8	0.45371	0.45374	0.45368	0.45371	0.45371
	1.0	0.38567	0.38568	0.38564	0.38568	0.38568
	3.0	0.15218	0.15216	0.15217	0.15218	0.15218
0.75	0.4	0.92052	0.92047	0.92047	0.92044	0.92050
	0.6	0.78300	0.78302	0.78297	0.78288	0.78299
	0.8	0.66272	0.66276	0.66270	0.66267	0.66272
	1.0	0.56932	0.56936	0.56930	0.56931	0.56933
	3.0	0.22782	0.22773	0.22773	0.22774	0.22774

solution of this example is

$$u(x,t) = \frac{(\frac{x}{t})}{1 + (\frac{t}{t_0})^{\frac{1}{2}} e^{\frac{x^2}{4\nu t}}},$$

where $t_0 = e^{\frac{1}{8\nu}}$. The comparison of numerical solutions with exact solutions for $h = 0.005, \nu = 0.005$ and $\Delta t = 0.001$ is given in the Table 6. The L_2 and L_∞ errors are computed for $\nu = 0.005, h = 0.005$ and $\Delta t = 0.001$ at different time levels and their comparison with [11, 12] is shown in Table 7. In this example, Table 6 shows that solutions produced by the present method are better than solutions in [12] and are much close to exact solutions even for small value of ν. It is also observed from Table 7 that, the L_2 and L_∞ errors by the proposed method are less than the errors obtained in [11, 12]. We have computed the numerical solutions for $\nu = 0.01, 0.001, N = 100, 300$ and $\Delta t = 0.001$ at different time levels. These solutions have been depicted in Figure 1 and Figure 2. It is noted that steepness in the solution curves increases as value of ν decreases and for sufficiently small value of ν solutions becomes discontinuous.

Table 4: Comparison of numerical and exact solutions for $\alpha = 2, h = 0.025$ and $\Delta t = 0.0001$ at $t = 0.001$ of Example 3.

x	$\nu = 1$			$\nu = 0.5$		
	[12]	Present	Exact	[12]	Present	Exact
0.1	0.653547	0.653544	0.653544	0.327870	0.327870	0.327870
0.2	1.305540	1.305533	1.305534	0.655071	0.655069	0.655069
0.3	1.949376	1.949363	1.949364	0.978416	0.978412	0.978413
0.4	2.565949	2.565924	2.565925	1.288469	1.288464	1.288463
0.5	3.110778	3.110738	3.110739	1.563074	1.563063	1.563064
0.6	3.492910	3.492665	3.492866	1.756654	1.756642	1.756642
0.7	3.549585	3.549595	3.549595	1.787204	1.787206	1.787206
0.8	3.049957	3.050138	3.050134	1.537649	1.537696	1.537696
0.9	1.816379	1.816666	1.816660	0.916786	0.916863	0.916860
$L_\infty \times 10^4$	2.85	0.056	-	0.744	0.030	-
$L_2 \times 10^4$	1.07	0.021	-	0.279	0.011	-

Table 5: Comparison of numerical and exact solutions for $\alpha = 2, h = 0.025$ and $\Delta t = 0.0001$ at $t = 0.001$ of Example 3.

x	$\nu = 0.2$			$\nu = 0.1$		
	[12]	Present	Exact	[12]	Present	Exact
0.1	0.131412	0.131412	0.131412	0.065750	0.065750	0.065750
0.2	0.262581	0.262581	0.262581	0.131383	0.131383	0.131383
0.3	0.392263	0.392262	0.392262	0.196281	0.196281	0.196281
0.4	0.516710	0.516709	0.516710	0.258576	0.258576	0.258576
0.5	0.627081	0.627079	0.627079	0.313850	0.313849	0.313849
0.6	0.705122	0.705120	0.705120	0.352972	0.352972	0.352972
0.7	0.717882	0.717882	0.717882	0.359443	0.359443	0.359443
0.8	0.618129	0.618137	0.618136	0.309579	0.309581	0.309580
0.9	0.368802	0.368815	0.368814	0.184751	0.184754	0.184754
$L_\infty \times 10^5$	1.22	0.123	-	0.308	0.063	-
$L_2 \times 10^6$	4.57	0.454	-	1.15	0.229	-

Example 5. Consider Burgers' equation (1) with $F(x,t) = \frac{kx}{(2\beta t+1)^2}$, $k > 0$, $\beta \geq 0$ and the initial condition $u(x,0) = kx$, $k > \beta$. The exact solution of this problem for $\nu = 1$ is obtained by Rao and Yadav [4] as follows

$$u(x,t) = \frac{A_0 x}{(2\beta t + 1)}, \qquad A_0 = \beta + \sqrt{\beta^2 + k}.$$

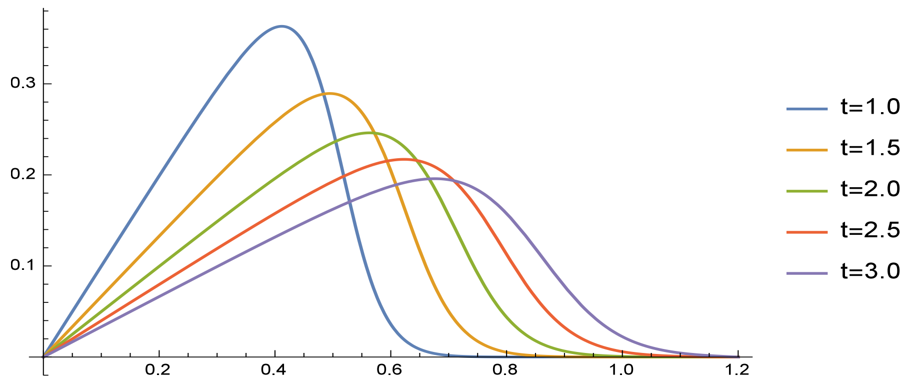

Figure 1: Numerical solutions of Example 4 for $\nu = 0.01, \Delta t = 0.001$ and $N = 100$.

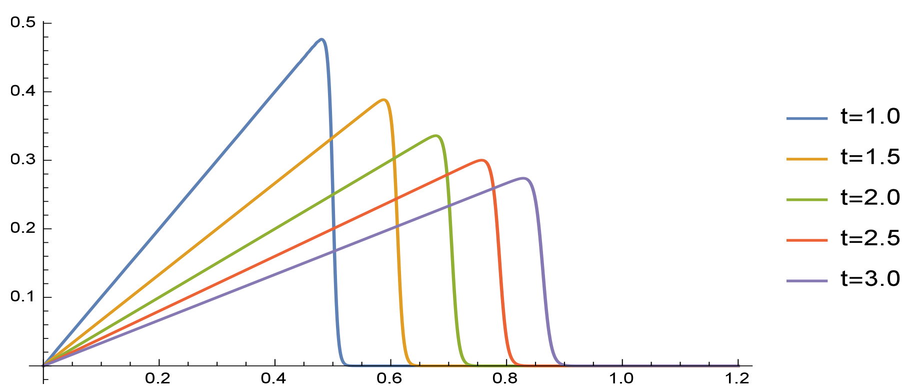

Figure 2: Numerical solutions of Example 4 for $\nu = 0.001, \Delta t = 0.001$ and $N = 300$.

The boundary conditions are given from the exact solution. The numerical solutions of this example obtained for $k = 5$, $\beta = 2$ in $x \in [-1, 1]$ is used to compute L_2 and L_∞ errors. The comparison of L_2 and L_∞ errors obtained in [9] is given in Table 8. From Table 8 it is seen that, the numerical solutions obtained by the present method are compatible with solutions in [9]. We have computed L_2 and L_∞ errors in the solution for $N = 20, k = 100$ and $\beta = 1$ at $t = 1$ for different values of Δt. These errors are listed in Table 9. It is observed that L_2 and L_∞ errors decreases with decreasing value of Δt. This shows that proposed scheme do not amplify errors with increase in number of iterations and the scheme is numerically

Table 6: Comparison of numerical and exact solutions for $\nu = 0.005, \Delta t = 0.001$ and $h = 0.005$ of Example 4.

x	t	[12]	Present	Exact
0.2	1.7	0.1176452	0.1176452	0.1176452
	2.5	0.0799990	0.0799990	0.0799990
	3.0	0.0666658	0.0666658	0.0666658
	3.5	0.0571422	0.0571422	0.0571422
0.4	1.7	0.2351690	0.2351677	0.2351677
	2.5	0.1599771	0.1599769	0.1599769
	3.0	0.1333211	0.1333209	0.1333209
	3.5	0.1142780	0.1142779	0.1142779
0.6	1.7	0.2958570	0.2959101	0.2959097
	2.5	0.2381299	0.2381207	0.2381207
	3.0	0.1994839	0.1994806	0.1994805
	3.5	0.1712257	0.1712242	0.1712242
0.8	1.7	0.0006381	0.0006465	0.0006465
	2.5	0.1021325	0.1020955	0.1020957
	3.0	0.2088032	0.2088360	0.2088359
	3.5	0.2145938	0.2145869	0.2145869

Table 7: Comparison of L_2 and L_∞ errors for $\nu = 0.005, \Delta t = 0.001$ and $h = 0.005$ of Example 4.

t	[11] $\beta = 0.5$ $\Delta t = 0.01, h = 0.001$		[12] $\Delta t = 0.001, h = 0.005$		Present $\Delta t = 0.001, h = 0.005$	
	$L_\infty \times 10^4$	$L_2 \times 10^4$	$L_\infty \times 10^4$	$L_2 \times 10^4$	$L_\infty \times 10^4$	$L_2 \times 10^4$
1.7	13.47279	3.84209	0.994	0.252	0.006	0.0017
2.5	15.54700	4.91345	0.549	0.151	0.002	0.0008
3.0	15.52891	5.15077	0.414	0.118	0.023	0.0029
3.5	15.21961	5.25855	0.486	0.117	0.572	0.0754

stable.

Example 6. In this example, we consider the initial condition $u(x, 0) = 0$,

Table 8: Comparison of L_2 and L_∞ errors for $k = 5, \beta = 2, \Delta t = 0.01$ and $N = 10$ of Example 5.

	L_∞		L_2	
	$t = 5$	$t = 10$	$t = 5$	$t = 10$
[9] IMQQI	2.816×10^{-9}	1.876×10^{-10}	2.020×10^{-9}	1.345×10^{-10}
Present	2.811×10^{-9}	1.872×10^{-10}	2.854×10^{-9}	1.901×10^{-10}

Table 9: Comparison of L_2 and L_∞ errors for $k = 100, \beta = 1$ and $N = 20$ at $t = 1$ of Example 5.

Δt	0.01	0.005	0.001
L_2	2.88×10^{-5}	5.90×10^{-6}	1.41×10^{-6}
L_∞	2.85×10^{-5}	5.89×10^{-6}	1.39×10^{-6}

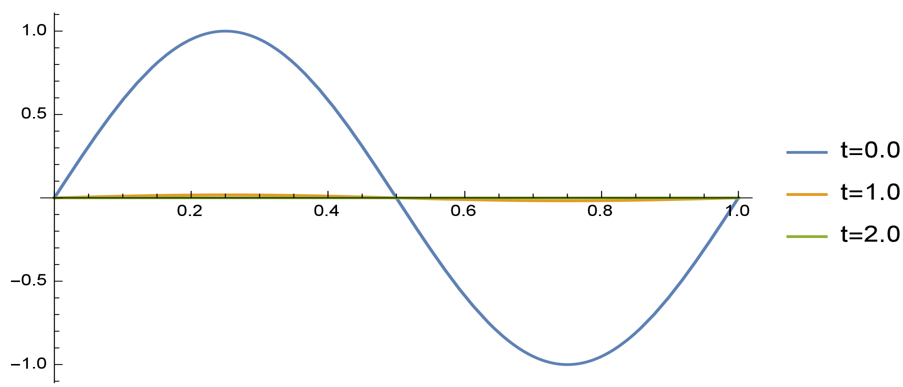

Figure 3: Numerical solutions of Example 7 for $\nu = 0.1, \Delta t = 0.01$ and $N = 50$.

boundary conditions $u(0,t) = u(\pi,t) = 0$ and $F(x,t) = A\sin(x)$; $A > 0$. The exact solution of this example is obtained by [1]. The numerical solutions obtained by the proposed method, exact solutions and solutions given in [9] for $\nu = 1, A = 20$ and $\nu = 0.1, A = 1$ are listed in Table 10 and Table 11, respectively. The proposed method produces better solutions than solutions obtained by DMQQI method and are compatible with the solutions obtained by IMQQI method for $\nu = 1$. For $\nu = 0.1$ the solutions computed by the proposed method are compatible with the solutions obtained by DMQQI and IMQQI methods.

Table 10: Comparison of numerical results for $\nu = 1, A = 20, \Delta t = 0.001$ at $t = 3.0$ of Example 6.

x	[9] IMQQI	[9] DMQQI	Present		Exact
	$N = 20$	$N = 30$	$N = 20$	$N = 30$	
0.5	2.1481	2.1474	2.1481	2.1481	2.1481
1.0	4.1562	4.1558	4.1563	4.1563	4.1562
1.5	5.8928	5.8924	5.8928	5.8928	5.8928
2.0	7.2404	7.2400	7.2403	7.2404	7.2404
2.5	8.0358	8.0298	8.0307	8.0303	8.0302
3.0	4.5143	4.4965	4.5155	4.5125	4.5140

Table 11: Comparison of numerical results for $\nu = 0.1, A = 1, \Delta t = 0.001$ at $t = 3.0$ of Example 6.

x	[9] IMQQI	[9] DMQQI	Present		Exact
	$N = 30$	$N = 60$	$N = 30$	$N = 60$	
0.5	0.4851	0.4851	0.4853	0.4853	0.4824
1.0	0.9392	0.9391	0.9392	0.9392	0.9331
1.5	1.3300	1.3320	1.3320	1.3320	1.3221
2.0	1.6271	1.6371	1.6372	1.6372	1.6222
2.5	1.8122	1.8322	1.8315	1.8323	1.8102
3.0	1.6207	1.6476	1.6633	1.6551	1.6155

Example 7. In this test problem we consider Eq. (1) with $F(x,t) = 0$, initial condition $u(x,0) = \sin 2\pi x, 0 \leq x \leq 1$ and boundary conditions $u(0,t) = u(1,t) = 0$. We have computed the numerical solutions for $\nu = 0.1, 0.01$ and 0.001 at $t = 0, 1, 2$. These solutions are depicted in Figure 3, Figure 4, and Figure 5. Similar solutions are reported in [13]. It is observed that when value of ν decreases, steepness of the curves increases and eventually solution becomes discontinuous.

5 Conclusion

The cubic B-spline Galerkin finite element method is successfully implemented to the one dimensional nonlinear forced Burgers' equation. Cubic

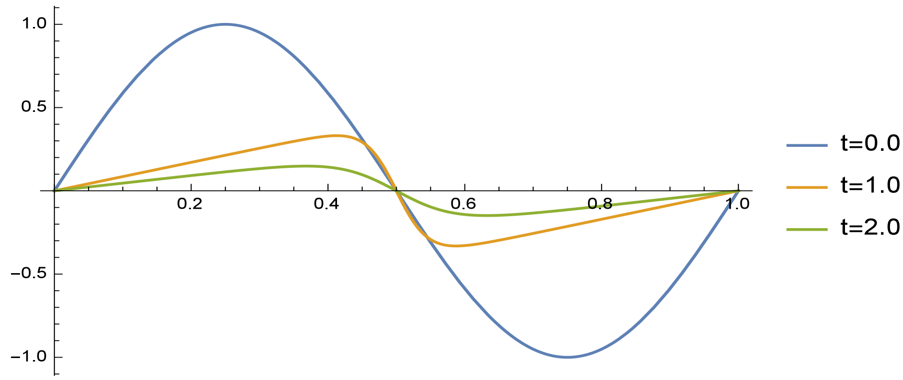

Figure 4: Numerical solutions of Example 7 for $\nu = 0.01, \Delta t = 0.01$ and $N = 100$.

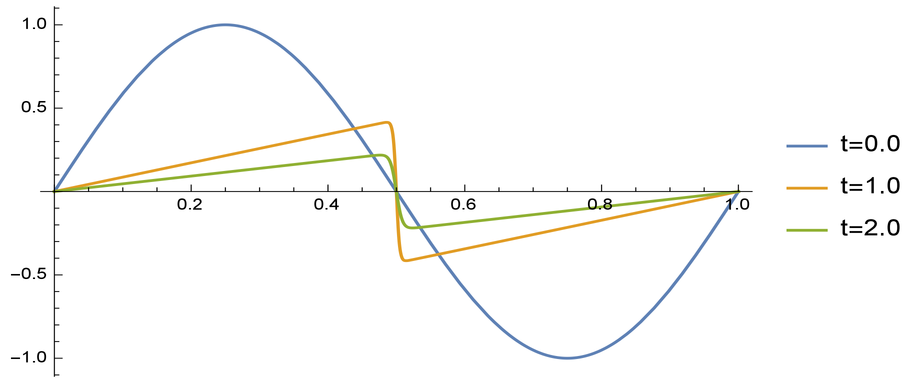

Figure 5: Numerical solutions of Example 7 for $\nu = 0.001, \Delta t = 0.0001$ and $N = 300$.

B-splines are redefined to accommodate the boundary conditions. Burgers' equation is discretized in time by using Taylors series expansion and then Galerkin finite element method is constructed for discretized equation. The Von Neumann stability analysis shows that the corresponding linearized scheme is unconditionally stable. some numerical test examples are solved to support the proposed method. It is seen that the method is efficient and reliable for solving the one dimensional forced Burgers' equation.

Acknowledgements

The authors expresses sincere thanks to the anonymous reviewers for their careful reading of the manuscript and many valuable comments and suggestions that improves quality of the paper.

References

[1] A.A. Karabutov, E.A. Lapshin and O.V. Rudenko, *Interaction between light waves and sound under acoustic nonlinearity conditions*, J. Exp. Theor. Phys. **44** (1976) 58-63.

[2] A.A. Soliman, *A Galerkin Solution for Burgers' Equation Using Cubic B-Spline Finite Elements*, Abstr. Appl. Anal. (2012) 1-15.

[3] B.V. Rathish Kumar and M. Mehra, *Wavelet-Taylor Galerkin Method For The Burgers Equation*, BIT Numerical Mathematics. **45** (2005) 543-560.

[4] C.S. Rao and M.K. Yadav *On the solution of a nonhomogeneous Burgers equation*, Int. J. Nonlinear Sci. **10** (2010) 141-145.

[5] E.N. Aksan *Quadratic B-spline finite element method for numerical solution of the Burgers' equation*, Appl. Math. Comput. **174** (2006) 884-896.

[6] I. Dag, B. Saka and A. Boz, *B-spline Galerkin methods for numerical solutions of the Burgers' equation*, Appl. Math. Comput. **166** (2005) 506-522.

[7] I. Dag, D. Irk and B. Saka *A numerical solution of the Burgers' equation using cubic B-splines*, Appl. Math. Comput. **163** (2005) 199-211.

[8] J.C. Tannehill, D.A. Anderson and R.H. Pletcher, *Computational fluid mechanics and heat transfer*, Taylor and Francis, Philadelphia, 2010.

[9] M. Sarboland and A. Aminataei *On the numerical solution of one-dimensional nonlinear nonhomogeneous Burgers' equation*, J. Appl. Math. (2014) 1-15.

[10] P.M. Prenter, *Splines and variational Mmethods*, Wiley, New York, 1975.

[11] R. Jiwari, R.C. Mittal and K.K. Sharma *A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers' equation*, Appl. Math. Comput. **219** (2013) 6680-6691.

[12] R.C. Mittal and R. Jain *Numerical solutions of nonlinear Burgers' equation with modified cubic B-splines collocation method*, Appl. Math. Comput. **218** (2012) 7839-7855.

[13] R.C. Mittal and R. Rohila *Traveling and shock wave simulations in a viscous Burgers' equation with periodic boundary conditions*, Int. J. Appl. Comput. Math. **4** (2018) 1-15.

[14] S.S. Wani and S.H. Thakar *Crank-Nicolson type method for Burgers equation*, Int. J. Appl. Phys. Math. **3** (2013) 324-328.

[15] S. Dhawan and S. Kumar *Finite element method for Burgers equation using cubic B-spline approximation*, Proc. Math. Meth. Comp. Tech. Ele. Eng. Con. Appl. Comp. (2011) 40-43.

[16] S. Kutluay, A. Esen and I. Dag *Numerical solutions of the Burgers' equation by thleast squares quadratic B-spline finite element method*, J. Comput. Appl. Math. **167** (2004) 21-33.