Mathematical Sciences (2020) 14:75–85
https://doi.org/10.1007/s40096-019-00317-5

ORIGINAL RESEARCH

A fourth-order B-spline collocation method for nonlinear Burgers–Fisher equation

Aditi Singh[1] · Sumita Dahiya[1] · S. P. Singh[1]

Received: 29 April 2019 / Accepted: 9 December 2019 / Published online: 9 January 2020

Abstract

A fourth-order B-spline collocation method has been applied for numerical study of Burgers–Fisher equation, which illustrates many situations occurring in various fields of science and engineering including nonlinear optics, gas dynamics, chemical physics, heat conduction, and so on. The present method is successfully applied to solve the Burgers–Fisher equation taking into consideration various parametric values. The scheme is found to be convergent. Crank–Nicolson scheme has been employed for the discretization. Quasi-linearization technique has been employed to deal with the nonlinearity of equations. The stability of the method has been discussed using Fourier series analysis (von Neumann method), and it has been observed that the method is unconditionally stable. In order to demonstrate the effectiveness of the scheme, numerical experiments have been performed on various examples. The solutions obtained are compared with results available in the literature, which shows that the proposed scheme is satisfactorily accurate and suitable for solving such problems with minimal computational efforts.

Keywords Burgers–Fisher equation · Cubic B-spline · Collocation method · Crank–Nicolson method · Gauss elimination method

Introduction

We consider the following Burgers–Fisher equation of the form

$$\frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial x^2} + \alpha v \frac{\partial v}{\partial x} + \beta v(1-v) = 0, \quad a \le x \le b \quad \text{and} \quad t \ge 0 \tag{1}$$

where α and β are advection and source/sink constants.

Initial and boundary conditions are as follows:

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

$$v(a,t) = g_0(t) \tag{3}$$

$$v(b,t) = g_1(t) \tag{4}$$

This manuscript deals with the numerical solution of Burgers–Fisher equation, which is nonlinear and parabolic in nature. It describes the mathematical model of many physical situations occurring in various fields of science and engineering such as heat conduction, gas dynamics, chemical physics and nonlinear optics. For example, it models velocity profile of viscous fluid in fluid dynamics [1], gas dynamics in an exhaust pipe [2], etc. It represents a prototypical model for relating the interaction between the convection effect, reaction mechanism and diffusion transport. It plays a significant role in nonlinear physics and thus has a great practical importance. Proposed by Fisher [3], it models population dynamics explaining the spatial spread of an advantageous allele and discussing its traveling wave solutions, and the equation originated as

$$\frac{\partial v}{\partial t} - D\frac{\partial^2 v}{\partial x^2} = kv(1-v) \tag{5}$$

Fisher's equation in its initial stages is extensively worked upon, and its solutions are given by various analytical and numerical methods [4–10]. The Burgers' equation, which was proposed by Burgers [11] modeling various physical

✉ Sumita Dahiya
sumita.iitr@gmail.com

Aditi Singh
aditi.dei2014@gmail.com

S. P. Singh
shail.dei@gmail.com

[1] Department of Mathematics, DEI Dayalbagh, Agra, UP 282005, India

phenomena such as gas dynamics, fluid mechanics, traffic flow and nonlinear acoustics, is given as

$$\frac{\partial v}{\partial t} - D\frac{\partial^2 v}{\partial x^2} + v\frac{\partial v}{\partial x} = 0 \tag{6}$$

Various numerical and analytical solutions of this equation are available in the literature [12–19]. Various noble methods were developed to numerically solve Fisher's reaction–diffusion equation shown in the papers [20–22]. The combination of these two equations is commonly known as the Burgers–Fisher equation given by (6).

Recently, various numerical and analytical methods have been used by various researchers to deal with the Burgers–Fisher equation. In 2004, Kaya and El-Sayed numerically simulated the generalized Burgers–Fisher equation [23] and came up with its explicit solutions. Ismail et al. [24] applied Adomian decomposition method (ADM), Javidi [25] employed modified pseudospectral method, Rashidi et al. [26] used homotopy perturbation method (HPM), Khattak [27] employed collocation-based radial basis functions method (CBRBF) and Xu and Xian [28] applied Exp-function method to find the analytic as well as numerical solutions of the generalized Burgers–Fisher equation. Also many other authors used different methods to obtain the analytical and numerical solution of the generalized Burgers–Fisher equation; for example, Zhu and Kang [29] used the B-spline quasi-interpolation method, Zhang and Yan [30] used a lattice Boltzmann model, Sari et al. [31] used the compact finite difference method, Sari et al. [32] developed the polynomial-based differential quadrature method, Zhang et al. [33] used the local discontinuous Galerkin (LDG) methods and Nawaz et al. [34] employed optimal homotopy asymptotic method (OHAM).

Very recently, Yadav and Jiwari [35] employed Galerkin's finite element method to analyze and approximate the Burgers–Fisher equation. S Malik, Qureshi, Amir, A Malik and Haq [36] used the Exp-function method hybridized with heuristic computation for the numerical simulation of the Burgers–Fisher equation. In 2015, Mittal and Tripathi developed a collocation method using cubic B-splines to numerically solve generalized Burgers–Fisher and generalized Burgers–Huxley equations [37].

Recently, B-spline functions have gained popularity as a powerful tool in the field of image processing, approximation theory and numerical simulation of boundary and initial value problems. B-splines as basis functions have been used in various numerical methods such as B-spline differential quadrature method and B-spline collocation method to deal with the partial differential equations. Cubic B-spline collocation method is used by Goh et al. [38] to solve heat and advection diffusion equations in one dimension. Dag and Saka [39] used the B-spline collocation method for equal-width equation. B-spline collocation method has been also used by Kadalbajoo and Arora [40] to deal with the singular perturbation problems and by Zahra [41] to study PHI-four and Allen–Cahn equations. Ersoy and Dag [42] applied this method to solve Kuramoto–Sivashinsky equation. Khater et al. [43] obtained numerical solution of the Burgers-type equations by using cubic spline collocation method.

In the proposed work, the fourth-order cubic B-spline collocation method is adopted to solve Burgers–Fisher equation. Fourth-order approximation for both single and double derivatives is employed. It has been done by using different end conditions and taking one more term in the Taylor series expansion, thus resulting in very accurate and efficient numerical solutions. Moreover, the present method does not require any involvement of integrals to get the final set of equations, thus reducing the computational efforts to a great extent.

The aim of this work is to investigate the numerical solutions of the Burgers–Fisher equation for different parametric values using collocation method with cubic B-splines as basis functions.

To the best of our knowledge, nobody has yet dealt with the Burgers–Fisher equation with the scheme considered in this work. The present scheme gives the approximate solution at any point of the solution domain. Our work is compared with the previous literature, and results are found to be better in terms of accuracy and efficiency. The proposed method is quite simple and produces highly accurate results for considerably lesser grid size, hence reducing complexity and computational cost.

The organization of this paper is as follows. "Mathematical formulation" section gives a description of the cubic B-spline collocation method. In "Implementation of the method" section, the method is applied to the Burgers–Fisher equation with the treatment of boundary conditions. In "Stability of the scheme" section, stability analysis of the method is carried out. "Numerical experiments and discussions" section presents some test examples of the Burgers–Fisher equation. A summary is given at the end of the paper in "Conclusion" section.

Mathematical formulation

Let us consider an equal partition of the domain Ω by the knots $x_j, j = 0, 1, 2, \ldots, N$, such that $h = x_j - x_{j-1}$ is the length of each interval. The third-degree B-splines termed as cubic B-splines are given as:

$$B_{i,3}(x) = \frac{1}{h^3}\begin{cases} (x - x_{i-2})^3 & x \in [x_{i-2}, x_{i-1}) \\ (x - x_{i-2})^3 - 4(x - x_{i-1})^3 & x \in [x_{i-1}, x_i) \\ (x_{i+2} - x)^3 - 4(x_{i+1} - x)^3 & x \in [x_i, x_{i+1}) \\ (x_{i+2} - x)^3 & x \in [x_{i+1}, x_{i+2}) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $[B_{-1}(x), B_0(x), B_1(x), \dots, B_N(x), B_{N+1}(x)]$ forms a basis over the interval.

In cubic B-spline collocation method, exact solution $v(x, t)$ is approximated by $K(x, t)$ in the form:

$$K(x,t) = \sum_{j=-1}^{N+1} a_j(t)B_j(x), \tag{8}$$

where $a_j(t)$'s are unknown quantities that are time dependent which we find using boundary conditions and collocation method. It is considered that $K(x, t)$ satisfies the following interpolatory and end conditions

$$K(x_j,t) = v(x_j,t), \qquad 0 \le j \le N \tag{9}$$

$$K''(x_j,t) = v''(x_j,t) - \frac{1}{12}h^2 v^{(4)}(x_j,t), \quad j = 0, N \tag{10}$$

If $v(x, t)$ is a smooth function and $K(x, t)$ is a unique cubic spline interpolant which satisfies the above boundary conditions, then from [44], we have:

$$K'(x_j,t) = v'(x_j,t) + O(h^4), \qquad 0 \le j \le N \tag{11}$$

$$K''(x_j,t) = v''(x_j,t) - \frac{1}{12}h^2 v^{(4)}(x_j,t) + O(h^4), \quad 0 \le j \le N \tag{12}$$

The approximate values $K(x, t)$ and their first-order derivatives at the knots are defined using Taylor expansions and finite difference as follows:

For $j = 0$,

$$v^{(4)}(x_j,t) = \frac{2K''(x_0,t) - 5K''(x_1,t) + 4K''(x_2,t) - K''(x_3,t)}{h^2} + O(h^2) \tag{13}$$

For $1 \le j \le N-1$,

$$v^{(4)}(x_j,t) = \frac{K''(x_{j-1},t) - 2K''(x_j,t) + K''(x_{j+1},t)}{h^2} + O(h^2) \tag{14}$$

For $j = N$,

$$v^{(4)}(x_j,t) = \frac{2K''(x_N,t) - 5K''(x_{N-1},t) + 4K''(x_{N-2},t) - K''(x_{N-3},t)}{h^2} + O(h^2) \tag{15}$$

Using Eqs. (13), (14) and (15) in (11), (12), we get

$$v'(x_j,t) = K'(x_j,t) + O(h^4), \quad 0 \le j \le N \tag{16}$$

For $j = 0$,

$$v''(x_0,t) = \frac{14K''(x_0,t) - 5K''(x_1,t) + 4K''(x_2,t) - K''(x_3,t)}{12} + O(h^4) \tag{17}$$

For $1 \le j \le N-1$,

$$v''(x_j,t) = \frac{K''(x_{j-1},t) + 10K''(x_j,t) + K''(x_{j+1},t)}{12} + O(h^4) \tag{18}$$

For $j = N$,

$$v''(x_N,t) = \frac{14K''(x_N,t) - 5K''(x_{N-1},t) + 4K''(x_{N-2},t) - K''(x_{N-3},t)}{12} + O(h^4) \tag{19}$$

Using Eqs. (7) and (8), Eqs. (17), (18), (19) can be simplified to be written as

For $j = 0$,

$$v''(x_0,t) = \frac{14a_{-1} - 33a_0 + 28a_1 - 14a_2 + 6a_3 - a_4}{2h^2} \tag{20}$$

For $1 \le j \le N-1$,

$$v''(x_j,t) = \frac{a_{j-2} + 8a_{j-1} - 18a_j + 8a_{j+1} + a_{j+2}}{2h^2} \tag{21}$$

For $j = N$,

$$v''(x_N,t) = \frac{14a_{N+1} - 33a_N + 28a_{N-1} - 14a_{N-2} + 6a_{N-3} - a_{N-4}}{2h^2} \tag{22}$$

Implementation of the method

We discretize Burgers–Fisher equation (1) by Crank–Nicolson scheme to get

$$\frac{v^{(n+1)} - v^{(n)}}{\Delta t} - \frac{v_{xx}^{(n+1)} + v_{xx}^{(n)}}{2} + \alpha\frac{(vv_x)^{(n+1)} + (vv_x)^{(n)}}{2} + \beta\frac{(v(1-v))^{(n+1)} + (v(1-v))^{(n)}}{2} = 0 \tag{23}$$

Separating the terms of nth and $(n+1)$th time levels, we get

$$v^{(n+1)}\left[1 + \frac{\alpha\Delta t}{2}v_x^{(n)} + \frac{\beta\Delta t}{2} - \beta\Delta t v^{(n)}\right] + v_x^{(n+1)}\left[\frac{\alpha\Delta t}{2}v^{(n)}\right] - \frac{\Delta t}{2}v_{xx}^{(n+1)} = v^{(n)}\left[1 - \frac{\beta\Delta t}{2}\right] + \frac{\Delta t}{2}v_{xx}^{(n)} \tag{24}$$

For $j = 0$,

$$\left[a_{-1}^{(n+1)}+4a_0^{(n+1)}+a_1^{(n+1)}\right]\left(1+\frac{\alpha\Delta t}{2}u_x^{(n)}+\frac{\beta\Delta t}{2}-\beta\Delta t u^{(n)}\right)+\left[a_1^{(n+1)}-a_{-1}^{(n+1)}\right]\left(\frac{3\alpha\Delta t}{2h}u^{(n)}\right)-\frac{\Delta t}{4h^2}\left[14a_{-1}^{(n+1)}-33a_0^{(n+1)}+28a_1^{(n+1)}-14a_2^{(n+1)}+6a_3^{(n+1)}-a_4^{(n+1)}\right]=\left[a_{-1}^{(n)}+4a_0^{(n)}+a_1^{(n)}\right]\left(1-\frac{\beta\Delta t}{2}\right)+\frac{\Delta t}{4h^2}\left[14a_{-1}^{(n)}-33a_0^{(n)}+28a_1^{(n)}-14a_2^{(n)}+6a_3^{(n)}-a_4^{(n)}\right] \tag{25}$$

We may write it as

$$s_1a_{-1}^{(n+1)}+s_2a_0^{(n+1)}+s_3a_1^{(n+1)}+s_4a_2^{(n+1)}+s_5a_3^{(n+1)}+s_6a_4^{(n+1)}=b_1a_{-1}^{(n)}+b_2a_0^{(n)}+b_3a_1^{(n)}+b_4a_2^{(n)}+b_5a_3^{(n)}+b_6a_4^{(n)} \tag{26}$$

For $1\le j\le N-1$,

$$\left[a_{j-1}^{(n+1)}+4a_j^{(n+1)}+a_{j+1}^{(n+1)}\right]\left(1+\frac{\alpha\Delta t}{2}u_x^{(n)}+\frac{\beta\Delta t}{2}-\beta\Delta t u^{(n)}\right)+\left[a_{j+1}^{(n+1)}-a_{j-1}^{(n+1)}\right]\left(\frac{3\alpha\Delta t}{2h}u^{(n)}\right)-\frac{\Delta t}{4h^2}\left[a_{j-2}^{(n+1)}+8a_{j-1}^{(n+1)}-18a_j^{(n+1)}+8a_{j+1}^{(n+1)}+a_{j+2}^{(n+1)}\right]=\left[a_{j-1}^{(n)}+4a_j^{(n)}+a_{j+1}^{(n)}\right]\left(1-\frac{\beta\Delta t}{2}\right)+\frac{\Delta t}{4h^2}\left[a_{j-2}^{(n)}+8a_{j-1}^{(n)}-18a_j^{(n)}+8a_{j+1}^{(n)}+a_{j+2}^{(n)}\right] \tag{27}$$

We may write it as

$$t_1a_{j-2}^{(n+1)}+t_2a_{j-1}^{(n+1)}+t_3a_j^{(n+1)}+t_4a_{j+1}^{(n+1)}+t_1a_{j+2}^{(n+1)}=p_1a_{j-2}^{(n)}+p_2a_{j-1}^{(n)}+p_3a_j^{(n)}+p_2a_{j+1}^{(n)}+p_1a_{j+2}^{(n)} \tag{28}$$

For $j=N$,

$$\left[a_{N-1}^{(n+1)}+4a_N^{(n+1)}+a_{N+1}^{(n+1)}\right]\left(1+\frac{\alpha\Delta t}{2}u_x^{(n)}+\frac{\beta\Delta t}{2}-\beta\Delta t u^{(n)}\right)+\left[a_N^{(n+1)}-a_{N-2}^{(n+1)}\right]\left(\frac{3\alpha\Delta t}{2h}u^{(n)}\right)-\frac{\Delta t}{4h^2}\left[14a_{-1}^{(n+1)}-33a_N^{(n+1)}+28a_{N-1}^{(n+1)}-14a_{N-2}^{(n+1)}+6a_{N-3}^{(n+1)}-a_{N-4}^{(n+1)}\right]=\left[a_{N-1}^{(n)}+4a_N^{(n)}+a_{N+1}^{(n)}\right]\left(1-\frac{\beta\Delta t}{2}\right)+\frac{\Delta t}{4h^2}\left[14a_{N+1}^{(n)}-33a_N^{(n)}+28a_{N-1}^{(n)}-14a_{N-2}^{(n)}+6a_{N-3}^{(n)}-a_{N-4}^{(n)}\right] \tag{29}$$

We may write it as

$$v_1a_{N-4}^{(n+1)}+v_2a_{N-3}^{(n+1)}+v_3a_{N-2}^{(n+1)}+v_4a_{N-1}^{(n+1)}+v_5a_N^{(n+1)}+v_6a_{N+1}^{(n+1)}=d_1a_{N-4}^{(n)}+d_2a_{N-3}^{(n)}+d_3a_{N-2}^{(n)}+d_4a_{N-1}^{(n)}+d_5a_N^{(n)}+d_6a_{N+1}^{(n)} \tag{30}$$

Hence, we get the following system of linear equations:

$$AC^{(n+1)}=BC^{(n)} \tag{31}$$

where

$$C=[a_{-1},a_0,a_1,\dots,a_{N+1}]^{\mathrm{T}} \tag{32}$$

$$A=\begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & \dots & \dots \\ t_1 & t_2 & t_3 & t_4 & t_1 & \dots & \dots & \dots \\ & t_1 & t_2 & t_3 & t_4 & t_1 & \dots & \dots \\ & \dots & \dots & \dots & \dots & \dots & & \\ & \dots & \dots & \dots & \dots & \dots & & \\ & \dots & \dots & t_1 & t_2 & t_3 & t_4 & t_1 \\ & \dots & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{bmatrix} \tag{33}$$

$$B=\begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & \dots & \dots \\ p_1 & p_2 & p_3 & p_2 & p_1 & \dots & \dots & \dots \\ & p_1 & p_2 & p_3 & p_2 & p_1 & \dots & \dots \\ & \dots & \dots & \dots & \dots & \dots & & \\ & \dots & \dots & \dots & \dots & \dots & & \\ & \dots & \dots & p_1 & p_2 & p_3 & p_2 & p_1 \\ & \dots & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \end{bmatrix} \tag{34}$$

Here, we can see that there are $N+1$ equations in $N+3$ unknowns. Dirichlet or Neumann boundary conditions a_{-1} and a_{N+1} can be eliminated to get $N+1$ equations in $N+1$ unknowns. After eliminating a_{-1} and a_{N+1}, system of equations can be solved with the initial vector $[a_0^{(0)},a_1^{(0)},a_2^{(0)},\dots,a_N^{(0)}]^{\mathrm{T}}$ at any desired time level. B-spline approximation of initial condition helps to get the initial vector.

Stability of the scheme

In Eq. (24), let us assume

$$v^n=k,\quad p_1=1+\frac{\alpha\Delta t}{2}v_x^{(n)}+\frac{\beta\Delta t}{2}-\beta\Delta t v^{(n)}, \tag{35}$$

$$p_2=\frac{3\alpha\Delta t}{2h}v^{(n)},\quad p_3=1-\frac{\beta\Delta t}{2} \tag{36}$$

Then,

$$\left[a_{j-1}^{(n+1)}+4a_j^{(n+1)}+a_{j+1}^{(n+1)}\right]p_1+\left[a_{j+1}^{(n+1)}-a_{j-1}^{(n+1)}\right]p_2 -\frac{\Delta t}{4h^2}\left[a_{j-2}^{(n+1)}+8a_{j-1}^{(n+1)}-18a_j^{(n+1)}+8a_{j+1}^{(n+1)}+a_{j+2}^{(n+1)}\right] = \left[a_{j-1}^{(n)}+4a_j^{(n)}+a_{j+1}^{(n)}\right]p_3 +\frac{\Delta t}{4h^2}\left[a_{j-2}^{(n)}+8a_{j-1}^{(n)}-18a_j^{(n)}+8a_{j+1}^{(n)}+a_{j+2}^{(n)}\right] \tag{37}$$

Assume $\frac{\Delta t}{h^2}=L$

$$\frac{-L}{4}a_{j-2}^{(n+1)}+[p_1-p_2-2L]a_{j-1}^{(n+1)}+\left[4p_1+\frac{9L}{2}\right]a_j^{(n+1)} +[p_1+p_2-2L]a_{j+1}^{(n+1)}-\frac{L}{4}a_{j+2}^{(n+1)}=\frac{L}{4}a_{j-2}^{(n)} +[p_3+2L]a_{j-1}^{(n)}+\left[4p_3-\frac{9L}{2}\right]a_j^{(n)} +[p_3+2L]a_{j+1}^{(n)}+\frac{L}{4}a_{j+2}^{(n)} \tag{38}$$

Substituting $a_j^{(n)}=D\xi^{(n)}\exp(ij\psi h)$, where $i=\sqrt{-1}$, h is step length, D is amplitude and ψ is mode number, we have

$$D\xi^{(n+1)}\left(\frac{-L}{4}e^{-2i\psi h}+[p_1-p_2-2L]e^{-i\psi h} +\left[4p_1+\frac{9L}{2}\right]+[p_1+p_2-2L]e^{i\psi h}-\frac{L}{4}e^{2i\psi h}\right) =D\xi^{(n)}\left(\frac{L}{4}e^{-2i\psi h}+[p_3+2L]e^{-i\psi h}+\left[4p_3-\frac{9L}{2}\right] +[p_3+2L]e^{i\psi h}+\frac{L}{4}e^{2i\psi h}\right) \tag{39}$$

$$\xi=\frac{\frac{L}{4}e^{-2i\psi h}+[p_3+2L]e^{-i\psi h}+\left[4p_3-\frac{9L}{2}\right]+[p_3+2L]e^{i\psi h}+\frac{L}{4}e^{2i\psi h}}{\frac{-L}{4}e^{-2i\psi h}+[p_1-p_2-2L]e^{-i\psi h}+\left[4p_1+\frac{9L}{2}\right]+[p_1+p_2-2L]e^{i\psi h}-\frac{L}{4}e^{2i\psi h}} \tag{40}$$

$$\xi=\frac{\frac{L}{2}\cos(2\psi h)+2[p_3+2L]\cos(\psi h)+\left[4p_3-\frac{9L}{2}\right]}{\frac{-L}{2}\cos(2\psi h)+2[p_1-2L]\cos(\psi h)+2ip_2\sin(\psi h)+\left[4p_1+\frac{9L}{2}\right]} \tag{41}$$

or

$$\xi=\frac{A}{B+Ci} \tag{42}$$

where

$$A=\frac{L}{2}\cos(2\psi h)+2[p_3+2L]\cos(\psi h) +\left[4p_3-\frac{9L}{2}\right]$$
$$B=\frac{-L}{2}\cos(2\psi h)+2[p_1-2L]\cos(\psi h) +\left[4p_1+\frac{9L}{2}\right] \tag{43}$$
$$C=2p_2\sin(\psi h)$$

For stability of the present scheme, we should have

$$|\ \xi\ |^2<1 \tag{44}$$

$$\implies \left|\frac{A}{B+Ci}\right|^2<1 \tag{45}$$

$$\implies \left|\frac{B+Ci}{A}\right|^2>1 \tag{46}$$

$$\implies \frac{B^2+C^2}{A^2}>1 \tag{47}$$

We need to show

$$B^2+C^2-A^2>0 \tag{48}$$

For minimum value of $B^2+C^2-A^2$, $\cos(\psi h)=1$. Thus, on putting values of A, B, C from Eq. (43) in $B^2+C^2-A^2$, we get $36\beta\Delta t\,(1-k)(2-\beta\Delta t\,k)$ which is obviously positive.

Hence, the proposed collocation method using B-splines as basis function is unconditionally stable.

Numerical experiments and discussions

The exact solution of the Burgers–Fisher equation (1) over the domain $[0,1]\times[0,T]$ is given by [24, 26, 34, 45]

$$v(x,t)=\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{-\alpha}{4}\left[x-\left(\frac{\alpha}{2}-\frac{2\beta}{\alpha}\right)t\right]\right) \tag{49}$$

Initial and boundary conditions are as follows:

$$v(x,0)=\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{-\alpha x}{4}\right) \tag{50}$$

$$v(0,t)=\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{\alpha}{4}\left[\frac{\alpha}{2}-\frac{2\beta}{\alpha}\right]t\right) \tag{51}$$

$$v(1,t)=\frac{1}{2}+\frac{1}{2}\tanh\left(\frac{-\alpha}{4}\left[1-\left[\frac{\alpha}{2}-\frac{2\beta}{\alpha}\right]t\right]\right) \tag{52}$$

In order to test the accuracy and efficiency of the proposed scheme, comparisons of the obtained results are made with the above exact solution and traditional methods such as [26, 31, 32, 35, 36]. MATLAB 8.1 has been utilized in this work for simulations.

Example 1 Here, results are computed for $\alpha = -1$ and $\beta = -1$ for different times. Table 1 makes comparison of absolute errors of the present scheme with FEM [35] at different times T. The absolute errors at grid points at times $T = 0.001$, 0.005 and 0.01 are shown in Table 2. Error decreases as time reduces. Method is highly accurate at middle grid points. Figure 1 shows computed solutions in 3D form for $T = 0.01$. Figure 2 depicts error profiles.

Example 2 Taking $\alpha = 0.001$ and $\beta = -0.001$, the obtained results of present scheme are compared with FEM [35], EFM [36], OHAM [34] and CFDM [31] at different times in Table 3. The absolute errors at grid points at times $T = 0.001$, 0.005 and 0.01 are shown in Table 4. CPU-time (s) has been calculated for all time levels and is shown in Table 3. Accuracy and low computational cost are the advantages of the method. In Table 5, the absolute errors at grid points for $T = 0.1$ are compared with EFM [36]. Error profiles are depicted in Fig. 3.

Example 3 The comparison of the present scheme results is made with the results of analytic solutions given by HPM [26] at a different set of values of α and β: Firstly, we take $\alpha = 0.1$ and $\beta = -0.1$ and then we take $\alpha = 0.5$ and $\beta = -0.5$ for times $T = 0.1, 0.4, 0.8$ in Table 6. It can be noticed that accuracy of the present method is better than the method used by HPM [26] for the former case and the accuracy gets better as time increases for the latter case. Evolution of computed solutions with space and time variables for $\alpha = 0.5$ and $\beta = -0.5$ at $T = 1$ is shown in Fig. 4. Comparison of the results obtained by the present method with the analytical solutions reveals the accuracy and ease of implementation of the new method.

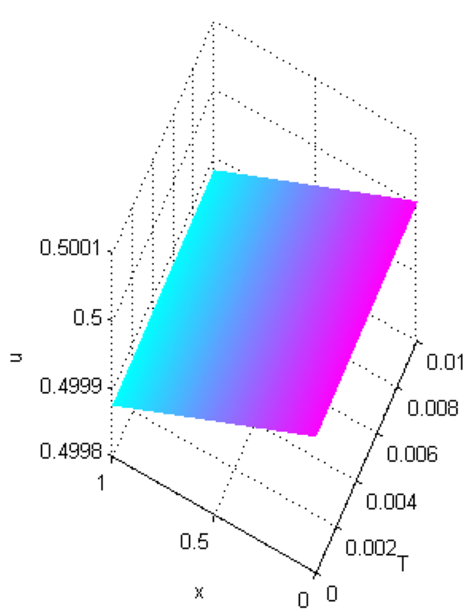

Fig. 1 Evolution of computed solutions of Example 1 with space and time variables for $T = 0.01$

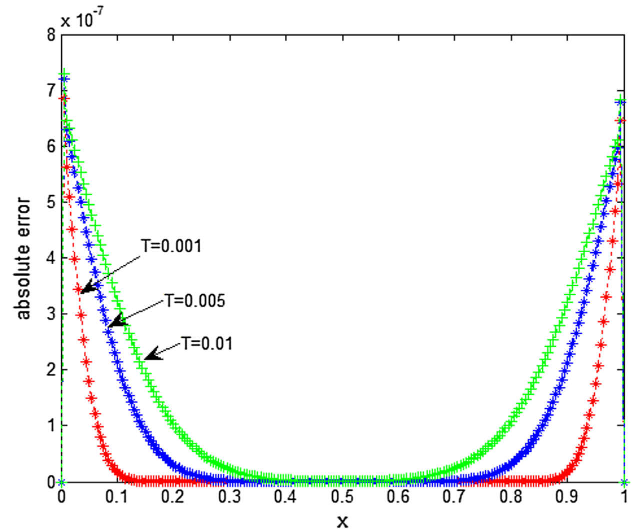

Fig. 2 Absolute errors of Example 1 for $\alpha = -1$ and $\beta = -1$ at different times T

Example 4 Numerical results are obtained for $\alpha = 0.01$, $\beta = -0.01$ and $\alpha = 0.0001, \beta = -0.0001$ at different times. Table 7 makes comparison of absolute errors E_A and relative errors E_R of the present scheme with PDQM [32] at different times T. It is noticed that the order of errors is the same for small as well as large times and thus maintains accuracy to a far greater extent. For $\alpha = 0.0001$ and $\beta = -0.0001$, it can be seen from Table 8 that the same order is maintained for times $T = 1$, 10 and 50 and the results maintain excellent accuracy by just taking $n = 16$. For the same α and β, CPU-time (s) of the present method is calculated and shown in Table 8. Thus, it can be inferred that the present scheme gives an easily computable numerical solution, which needs low storage, minimal computational effort and cost. Hence, it can be resolved that present method is easy and simpler to apply in comparison with other existing methods, e.g., finite volume, finite element, spectral collocation methods, etc. Figure 5 shows error profiles for $\alpha = 0.0001$ and $\beta = -0.0001$ at time $T = 50$. Computed solutions of the present method for $\alpha = 0.0001$ and $\beta = -0.0001$ at different times T are depicted in Fig. 6.

Example 5 Now, we take a different set of values of α and β as $\alpha = -0.1, \beta = -0.1$ and $\alpha = -0.01, \beta = -0.01$. The absolute and relative errors of the obtained results have been presented for $T = 0.01$, 0.1 and 1 in Table 9 and compared with

Table 1 Absolute error comparison of the present method with FEM [35] for $\alpha = -1$ and $\beta = -1$

x	$T = 0.001$		$T = 0.005$		$T = 0.01$	
	Present method	FEM [35]	Present method	FEM [35]	Present method	FEM [35]
0.1	1.70E−008	1.91E−006	2.14E−007	7.08E−007	3.23E−007	4.99E−007
0.5	2.53E−014	4.72E−007	9.03E−013	1.69E−006	5.53E−010	1.08E−006
0.9	1.70E−008	1.72E−006	2.13E−007	5.11E−007	3.21E−007	3.54E−007

Table 2 Absolute errors at grid points for $\alpha = -1, \beta = -1$ at different times T

x	$T = 0.001$	$T = 0.005$	$T = 0.01$
0.1	1.7078E−008	2.1434E−007	3.2389E−007
0.2	5.0626E−012	2.9936E−008	1.0350E−007
0.3	1.1102E−014	1.7287E−009	2.1711E−008
0.4	1.1657E−014	3.9512E−011	2.9320E−009
0.5	2.5313E−014	9.0339E−013	5.5391E−010
0.6	3.4306E−014	4.6625E−011	3.4449E−009
0.7	3.4528E−014	1.9248E−009	2.4153E−008
0.8	5.3543E−012	3.1502E−008	1.0880E−007
0.9	1.7003E−008	2.1318E−007	3.2173E−007

Table 4 Absolute errors at grid points for $\alpha = 0.001, \beta = -0.001$ at different times T

x	$T = 0.001$	$T = 0.005$	$T = 0.01$
0.1	4.6470E−011	5.8512E−010	8.8446E−010
0.2	7.0222E−014	8.4073E−011	2.9048E−010
0.3	5.5789E−014	5.2426E−012	6.3005E−011
0.4	5.6677E−014	3.9008E−013	9.1993E−012
0.5	5.3957E−014	2.7317E−013	2.0401E−012
0.6	5.4845E−014	3.8886E−013	9.1954E−012
0.7	5.4456E−014	5.2395E−012	6.3002E−011
0.8	7.0832E−014	8.4079E−011	2.9048E−010
0.9	4.6478E−011	5.8515E−010	8.8450E−010
CPU-time (s)	12.7352	63.0485	125.7683

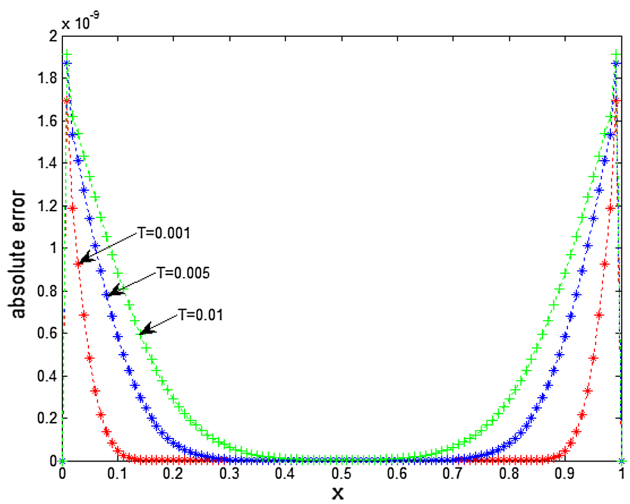

Fig. 3 Absolute errors of Example 2 for $\alpha = 0.001$ and $\beta = -0.001$ at different times T

Table 5 Comparison of absolute errors with EFM [36] for time $T = 0.1$ and $\alpha = 0.001, \beta = -0.001$

x	Present method	EFM [36]
0.1	1.575E−009	1.988E−008
0.2	1.332E−009	1.706E−008
0.3	1.139E−009	1.390E−008
0.4	1.015E−009	1.040E−008
0.5	9.731E−010	6.547E−009
0.6	1.015E−009	2.354E−009
0.7	1.139E−009	2.182E−009
0.8	1.332E−009	7.062E−009
0.9	1.575E−009	1.228E−008

Table 3 Absolute error comparison of the present scheme with different schemes for $\alpha = 0.001$ and $\beta = -0.001$

x	T	Present method	FEM [35]	EFM [36]	OHAM [34]	CFDM [31]
0.1	0.001	4.64E−011	1.21E−009	1.97E−008	2.25E−008	1.01E−007
	0.005	5.85E−010	1.69E−009	1.97E−008	1.12E−007	4.38E−007
	0.01	8.84E−010	1.28E−009	1.97E−008	2.25E−007	7.53E−007
0.5	0.001	5.39E−014	2.28E−012	3.58E−009	4.58E−008	1.04E−007
	0.005	2.73E−013	2.49E−009	3.71E−009	2.29E−007	5.21E−007
	0.01	2.04E−012	2.50E−009	3.88E−009	4.58E−007	1.04E−006
0.9	0.001	4.64E−011	1.20E−010	1.80E−008	4.58E−008	1.01E−007
	0.005	5.85E−010	1.69E−009	1.77E−008	2.29E−007	4.38E−007
	0.01	8.84E−010	1.28E−009	1.74E−008	4.58E−007	7.53E−007

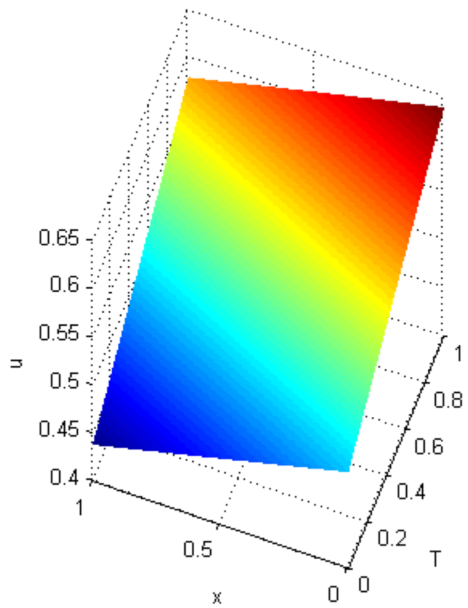

Fig. 4 Evolution of computed solutions of Example 3 with space and time variables for $\alpha = 0.5$ and $\beta = -0.5$ at $T = 1$

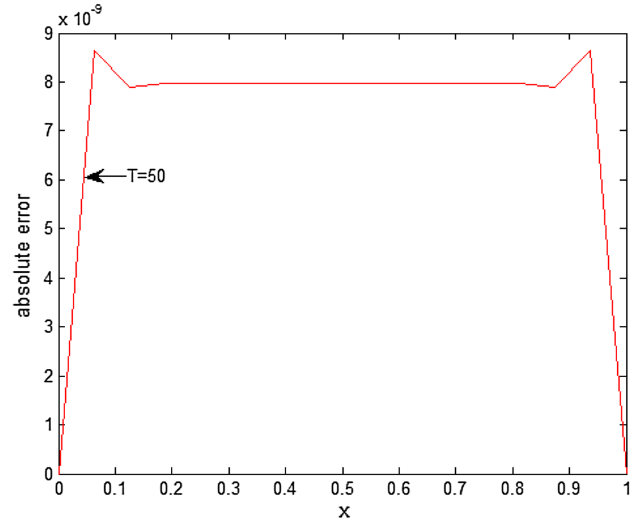

Fig. 5 Absolute errors of Example 4 for $\alpha = 0.0001$ and $\beta = -0.0001$ at time $T = 50$

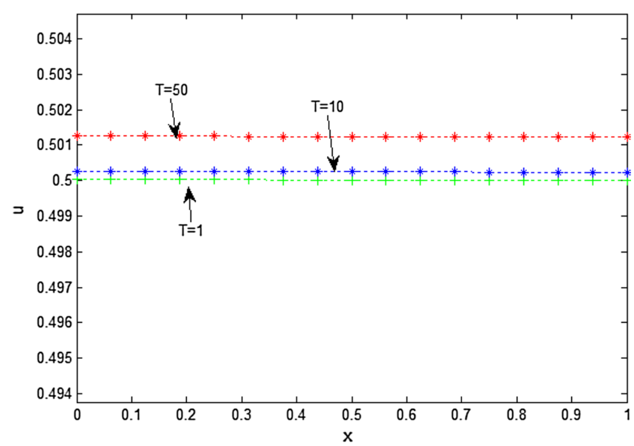

Fig. 6 Computed solutions of the present method in Example 4 for $\alpha = 0.0001$ and $\beta = -0.0001$ at different times T

PDQM [32]. Figure 7 depicts approximate solutions of the present method for $\alpha = -0.1$ and $\beta = -0.1$ at different times T. We also calculated CPU-time (s) of the present method. When $\alpha = -0.1$ and $\beta = -0.1$, CPU-time (s) for $T = 0.01$ is 0.1546, $T = 0.1$ is 0.6453 and $T = 1$ is 5.4911. This shows that computational cost of the present method is low. Hence, it can be clearly seen that the present method is more efficient, accurate and reliable.

Table 6 Comparison of numerical solutions of the present method with analytic solutions of HPM [26] at a different set of values of α and β

T	x	$\alpha = 0.5, \beta = -0.5$		$\alpha = 0.1, \beta = -0.1$	
		Present method	HPM [26]	Present method	HPM [26]
0.1	0.2	1.2205E−006	6.1768E−006	2.1998E−007	4.3262E−008
	0.4	9.2477E−007	1.6029E−005	1.6726E−007	1.0883E−007
	0.6	9.0491E−007	2.5802E−005	1.666E−007	1.7457E−007
	0.8	1.1797E−006	3.5447E−005	2.1871E−007	2.4012E−007
0.4	0.2	1.6342E−006	7.8774E−005	2.9985E−007	3.8516E−007
	0.4	1.6248E−006	7.8951E−005	2.9715E−007	6.6533E−007
	0.6	1.6277E−006	2.3628E−004	2.9713E−007	1.7158E−006
	0.8	1.6438E−006	3.9244E−004	2.9982E−007	2.7658E−006
0.8	0.2	1.6046E−006	1.2446E−003	3.0381E−007	7.2803E−006
	0.4	1.6189E−006	6.2245E−004	3.0384E−007	3.0801E−006
	0.6	1.6316E−006	2.8091E−006	3.0392E−007	1.1209E−006
	0.8	1.6427E−006	6.2804E−004	3.0403E−007	5.3215E−006

Table 7 Comparison of E_A and E_R of the present method with PDQM [32] at different times T for different parametric values

x	T	$\alpha = 0.01, \beta = -0.01$				$\alpha = 0.0001, \beta = -0.0001$			
		Present method		PDQM [32]		Present method		PDQM [32]	
		E_A	E_R	E_A	E_R	E_A	E_R	E_A	E_R
x_3	1	7.91E−007	1.57E−006	2.14E−005	4.27E−005	7.97E−009	1.57E−008	2.15E−007	4.31E−007
	10	7.89E−007	1.50E−006	2.04E−005	3.89E−005	7.97E−009	1.57E−008	2.15E−007	4.30E−007
	50	7.43E−007	1.19E−006	1.53E−005	2.46E−005	7.96E−009	1.57E−008	2.15E−007	4.29E−007
x_8	1	7.98E−007	1.59E−006	1.28E−004	2.56E−004	7.96E−009	1.59E−008	1.29E−006	2.58E−006
	10	7.96E−007	1.51E−006	1.22E−004	2.33E−004	7.95E−009	1.59E−008	1.29E−006	2.57E−006
	50	7.50E−007	1.20E−006	9.16E−005	1.47E−004	7.94E−009	1.58E−008	1.28E−006	2.56E−006
x_{13}	1	7.98E−007	1.59E−006	4.49E−005	8.95E−005	7.97E−009	1.59E−008	4.50E−007	9.00E−007
	10	7.96E−007	1.51E−006	4.28E−005	8.16E−005	7.98E−009	1.59E−008	4.50E−007	9.00E−007
	50	7.50E−007	1.20E−006	3.20E−005	5.15E−005	7.98E−009	1.58E−008	4.49E−007	8.95E−007

Table 8 Absolute errors at grid points for $\alpha = 0.0001, \beta = -0.0001$ at different times T

x	$T = 1$	$T = 10$	$T = 50$
x_1	8.6414E−009	8.6415E−009	8.6415E−009
x_2	7.8946E−009	7.8948E−009	7.8948E−009
x_3	7.9700E−009	7.9703E−009	7.9702E−009
x_4	7.9623E−009	7.9626E−009	7.9626E−009
x_5	7.9630E−009	7.9634E−009	7.9634E−009
x_6	7.9628E−009	7.9633E−009	7.9633E−009
x_7	7.9628E−009	7.9633E−009	7.9633E−009
x_8	7.9628E−009	7.9633E−009	7.9633E−009
x_9	7.9628E−009	7.9633E−009	7.9633E−009
x_{10}	7.9628E−009	7.9633E−009	7.9633E−009
x_{11}	7.9630E−009	7.9634E−009	7.9634E−009
x_{12}	7.9623E−009	7.9626E−009	7.9626E−009
x_{13}	7.9700E−009	7.9703E−009	7.9702E−009
x_{14}	7.8946E−009	7.8948E−009	7.8948E−009
x_{15}	8.6414E−009	8.6415E−009	8.6415E−009
CPU-time(s)	5.2598	50.4549	250.6945

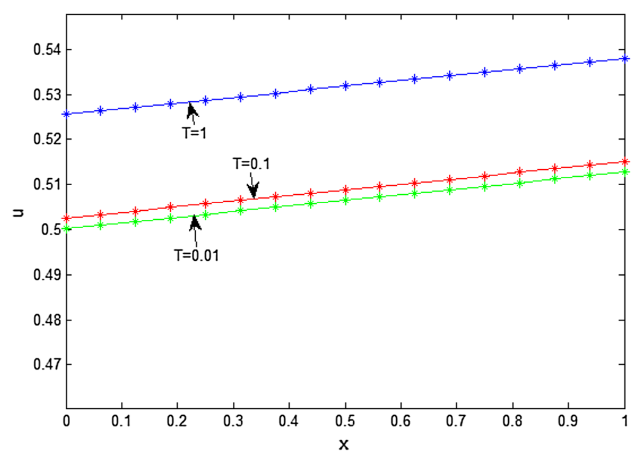

Fig. 7 Computed solutions of the present method in Example 5 for $\alpha = -0.1$ and $\beta = -0.1$ at different times T

Conclusion

- The fourth-order cubic B-spline method has been adopted to numerically solve nonlinear Burgers–Fisher equation.
- Crank–Nicholson for discretization and quasi-linearization to deal with the nonlinear nature of the equation are used.
- Five examples with varying parameters have been taken to elaborate the efficacy of the method.
- The numerical results obtained comply with the nature of solution of Burgers–Fisher equation and are better than results available in the literature.
- Method is very efficient, less complex and can be extended to higher dimensional partial differential equations.

Table 9 Comparison of E_A and E_R of the present method with PDQM [32]

x	T	$\alpha = -0.1, \beta = -0.1$				$\alpha = -0.01, \beta = -0.01$			
		Present method		PDQM [29]		Present method		PDQM [29]	
		E_A	E_R	E_A	E_R	E_A	E_R	E_A	E_R
x_3	0.01	2.97E−006	5.92E−006	4.18E−005	8.35E−005	2.91E−007	5.83E−007	6.89E−006	1.38E−005
	0.1	6.59E−006	1.30E−005	1.47E−004	2.92E−004	6.46E−007	1.29E−006	2.84E−005	5.68E−005
	1	8.04E−006	1.52E−005	2.05E−004	3.89E−004	7.91E−007	1.57E−006	7.97E−005	1.59E−004
x_8	0.01	1.55E−008	3.07E−008	1.03E−004	2.04E−004	1.53E−009	3.06E−009	1.04E−005	2.07E−005
	0.1	4.35E−006	8.56E−006	7.83E−004	1.54E−003	4.26E−007	8.51E−007	1.02E−004	2.04E−004
	1	8.13E−006	1.53E−005	1.21E−003	2.28E−003	7.98E−007	1.58E−006	4.68E−004	9.29E−004
x_{13}	0.01	6.18E−007	1.21E−006	7.14E−005	1.40E−004	6.00E−008	1.19E−007	9.68E−006	1.93E−005
	0.1	5.43E−006	1.06E−005	2.94E−004	5.72E−004	5.30E−007	1.05E−006	5.39E−005	1.07E−004
	1	8.12E−006	1.51E−005	4.18E−004	7.78E−004	7.98E−007	1.58E−006	1.65E−004	3.27E−004

Acknowledgements The authors thank the anonymous referees for their useful and timely suggestions. The corresponding author would like to thank Prof R.C. Mittal for being a great mentor and guiding towards the right path. His support, patience and time is truly acknowledged.

References

1. Debnath, L.: Nonlinear Partial Differential Equations for Scientists and Engineers, 3rd edn. Birkhäuser, Basel (2012)
2. Gasser, I., Rybicki, M.: Modelling and simulation of gas dynamics in an exhaust pipe. Appl. Math. Model. **37**, 2747–2764 (2013)
3. Fisher, R.A.: The wave of advance of advantageous genes. Ann. Eugen. **7**, 355–369 (1937)
4. Mavoungou, T., Cherruault, Y.: Numerical study of Fisher's equation by Adomian's method. Math. Comput. Model. **19**, 89–95 (1994)
5. Luogeng, H.: Acta mathematicae applicatae sinica. Celest. Mech. Dyn. Astron. **35**, 8–8 (1985)
6. Petrovskii, S., Shigesada, N.: Some exact solutions of a generalized Fisher equation related to the problem of biological invasion. Math. Biosci. **172**, 73–94 (2001)
7. Tang, S., Weber, R.: Numerical study of Fisher's equation by a Petrov–Galerkin finite element method. J. Aust. Math. Soc. Ser. B Appl. Math. **33**, 27–38 (1991)
8. Roessler, J., Hüssner, H.: Numerical solution of the 1 + 2 dimensional Fisher's equation by finite elements and the Galerkin method. Math. Comput. Model. **25**, 57–67 (1997)
9. Mickens, R.E.: A best finite-difference scheme for the Fisher equation. Numer. Methods Partial Differ. Equ. **10**, 581–585 (1994)
10. Dag, I., Sahin, A., Korkmaz, A.: Numerical investigation of the solution of Fisher's equation via the B-spline Galerkin method. Numer. Methods Partial Differ. Equ. **26**, 1483–1503 (2010)
11. Burgers, J.M.: A mathematical model illustrating the theory of turbulence. In: von Mises, R., von Kármán, T. (eds.) Advances in Applied Mechanics, pp. 171–199. Academic Press, New York (1948)
12. Kutluay, S., Bahadir, A.R., Özdes, A.: Numerical solution of one-dimensional Burgers equation: explicit and exact–explicit finite difference methods. J. Comput. Appl. Math. **103**, 251–261 (1999)
13. Cole, J.D.: On a quasi-linear parabolic equation occurring in aerodynamics. Q. Appl. Math. **9**, 225–236 (1951)
14. Hopf, E.: The partial differential equation $u_t + uu_x = \mu_{xx}$. Commun. Pure Appl. Math. **3**, 201–230 (1950)
15. Jiwari, R.: A Haar wavelet quasilinearization approach for numerical simulation of Burgers equation. Comput. Phys. Commun. **183**, 2413–2423 (2012)
16. Jiwari, R., Mittal, R.C., Sharma, K.K.: A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burgers equation. Appl. Math. Comput. **219**, 6680–6691 (2013)
17. Jiwari, R.: A hybrid numerical scheme for the numerical solution of the Burgers equation. Comput. Phys. Commun. **188**, 59–67 (2015)
18. Gorgulu, M.Z., Dag, I., Irk, D.: Wave propagation by way of exponential B-spline Galerkin method. J. Phys. Conf. Ser. **766**, 1 (2016)
19. Biazar, J., Ghazvini, H.: Exact solutions for nonlinear Burgers' equation by homotopy perturbation method. Numer. Methods Partial Differ. Equ. **25**, 833–842 (2009)
20. Rohila, R., Mittal, R.C.: Numerical study of reaction diffusion Fisher's equation by fourth order cubic B-spline collocation method. Math. Sci. **12**, 79–89 (2018)
21. Mittal, R.C., Jain, R.K.: Numerical solutions of nonlinear Fisher's reaction–diffusion equation with modified cubic B-spline collocation method. Math. Sci. **7**, 12 (2013)
22. Mittal, R.C., Arora, G.: Efficient numerical solution of Fisher's equation by using B-spline method. Int. J. Comput. Math. **87**, 3039–3051 (2010)
23. Kaya, D., El-Sayed, S.M.: A numerical simulation and explicit solutions of the generalized Burgers–Fisher equation. Appl. Math. Comput. **152**, 403–13 (2004)

24. Ismail, H.N.A., Raslan, K., Rabboh, A.A.A.: Adomian decomposition method for Burger's–Huxley and Burger's–Fisher equations. Appl. Math. Comput. **159**(1), 291–301 (2004)
25. Javidi, M.: Modified pseudospectral method for generalized Burger's–Fisher equation. Int. Math. Forum **1**(32), 1555–1564 (2006)
26. Rashidi, M.M., Ganji, D.D., Dinarvand, S.: Explicit analytical solutions of the generalized Burger and Burger–Fisher equations by homotopy perturbation method. Numer. Methods Partial Differ. Equ. **25**, 409–417 (2009)
27. Khattak, A.J.: A computational meshless method for the generalized Burger's–Huxley equation. Appl. Math. Model. **33**, 3718–3729 (2009)
28. Xu, Z.H., Xian, D.Q.: Application of Exp-function method to generalized Burgers–Fisher equation. Acta Math. Appl. Sin. Engl. Ser. **26**(4), 669–676 (2010)
29. Zhu, C.G., Kang, W.S.: Numerical solution of Burgers–Fisher equation by cubic B-spline quasi-interpolation. Appl. Math. Comput. **216**(9), 2679–2686 (2010)
30. Zhang, J., Yan, G.: A lattice Boltzmann model for the Burgers–Fisher equation. Chaos **20**, 023129 (2010)
31. Sari, M., Gürarslan, G., Dag, I.: A compact finite difference method for the solution of the generalized Burgers–Fisher equation. Numer. Method Partial Differ. Equ. **26**, 125–134 (2010)
32. Sari, M.: Differential quadrature solutions of the generalized Burgers–Fisher equation with a strong stability preserving high-order time integration. Math. Comput. Appl. **16**(2), 477–486 (2011)
33. Zhang, R., Yu, X., Zhao, G.: The local discontinuous Galerkin method for Burger's–Huxley and Burger's–Fisher equations. Appl. Math. Comput. **218**, 8773–8778 (2012)
34. Nawaz, R., Ullah, H., Islam, S., Idrees, M.: Application of optimal homotopy asymptotic method to Burger equations. J. Appl. Math. **2013**, 1–8 (2013)
35. Yadav, O.P., Jiwari, R.: Finite element analysis and approximation of Burgers'–Fisher equation. Numer. Methods Partial Differ. Equ. **33**(5), 1652–1677 (2017)
36. Malik, S.A., Qureshi, I.M., Amir, M., Malik, A.N., Haq, I.: Numerical solution to generalized Burgers–Fisher equation using exp-function method hybridized with heuristic computation. PLoS ONE **10**, e0121728 (2015)
37. Mittal, R.C., Tripathi, A.: Numerical solutions of generalized Burgers–Fisher and generalized Burgers–Huxley equations using collocation of cubic B-splines. Int. J. Comput. Math. **92**, 1053–1077 (2015)
38. Goh, J., Majid, A.A., Ismail, A.I.M.: Cubic B-spline collocation method for one-dimensional heat and advection–diffusion equations. J. Appl. Math. **2012**, 1–8 (2012)
39. Dag, I., Saka, B.: A cubic B-spline collocation method for the EW equation. Math. Comput. Appl. **9**, 81–92 (2004)
40. Kadalbajoo, M.K., Arora, P.: B-spline collocation method for the singular-perturbation problem using artificial viscosity. Comput. Math. Appl. **57**, 650–663 (2009)
41. Zahra, W.K.: Trigonometric B-spline collocation method for solving PHI-four and Allen–Cahn equations. J. Math. Mediterr. **14**, 122 (2017)
42. Ersoy, O., Dag, I.: The exponential cubic B-spline collocation method for the Kuramoto–Sivashinsky equation. Filomat **30**(3), 853–861 (2016)
43. Khater, A.H., Temsah, R.S., Hassan, M.M.: A Chebyshev spectral collocation method for solving Burgers-type equations. J. Comput. Appl. Math. **222**(2), 333–50 (2008)
44. Lucas, T.R.: Error bounds for interpolating cubic splines under various end conditions. SIAM J. Numer. Anal. **11**, 569–584 (1975)
45. Wang, X., Lu, Y.: Exact solutions of the extended Burgers–Fisher equation. Chin. Phys. Lett. **7**, 145–147 (1990)