A numerical approximation for the one-dimensional Burger–Fisher equation

Abstract

In this paper, an implicit finite difference method based on the Crank–Nicolson method is proposed for the numerical solution of the one-dimensional Burger–Fisher equation. The Crank–Nicolson scheme provides a system of nonlinear difference equations, which is solved by an integration of the Jacobian-Free-Newton-Krylov (JFNK) and GMRES methods. Various numerical examples are given to demonstrate the efficiency of the proposed scheme. Comparison of the computed solutions with the analytical ones demonstrates the accuracy of this proposed method.

Keywords: Crank–Nicolson scheme, Burger–Fisher equation, Jacobian-Free-Newton-Krylov method, GMRES method.

1. Introduction

Being a combination of convection, diffusion, and reaction mechanisms, the Burger–Fisher equation is highly nonlinear. This equation has many applications in many scientific fields such as gas dynamics, number theory, elasticity, heat conduction, etc. [1]. Recently, several methods have been proposed to solve it. Here, we briefly discuss the methods of some researchers.

Pirdawood and Sabawi [2] proposed an accurate scheme using the compact finite difference method and the Runge Kutta method. Gürbüz and Sezer [3] applied a modified Laguerre matrix-collocation method. Singh et al. [4] constructed a fourth-order B-spline collocation method; in this method, the Crank–Nicolson scheme is used for the discretization, and the Quasi-linearization is used for obtained nonlinear equations. Mohanty and Sharma [5] presented a numerical method based on off-step non-polynomial spline approximations. Wasim et al. [6] proposed a hybrid B-spline collocation method; in this method, the usual finite difference scheme and the Crank-Nicolson are used. Yadav and Jiwari [7] presented the finite element approximate, which has less complicacy. The trial equation method is applied by Triki and Wazwaz [8]. In [9], two implicit finite difference schemes were designed to solve the one-dimensional nonlinear Burger–Fisher equation. The exact finite difference scheme and nonstandard finite difference schemes were used by Zhang et al. [10]. Zhang et al. [11] studied the local discontinuous Galerkin. Kocacoban et al. [12] solved the Burger–Fisher equation using the reduced differential transformation method. Zhang and Yan [13] investigated a lattice Boltzmann model. Sari et al. [14] introduced a compact finite difference method. Rashidi et al. [15] proposed an analytical solution using the homotopy perturbation method. Khattak [16] presented a computational meshless method. Golbabai and Javidi [17] and Ismail et al. [18] studied Adomian decomposition methods. Javidi et al. [19] introduced the spectral collocation method. The tanh method is proposed by Wazwaz [20]. Ismail and Rabboh [21] developed a restrictive Padé approximation. Kaya and Sayed [22] found explicit solutions to the generalized Burger–Fisher equation. Zhu and Kang [1] studied the Burger–Fisher equation by applying the B-spline quasi-interpolation method. Mickens and Gumel [23] developed a nonstandard finite difference scheme.

A generalized form of the one-dimensional Burger–Fisher equation is as follows.

$$\frac{\partial U}{\partial t} - \frac{\partial^2 U}{\partial x^2} + \alpha(t)\, U^{\delta}\, \frac{\partial U}{\partial x} = \beta(t)\, U \left(1 - U^{\delta}\right), \qquad x \in \Omega, 0 \le t \le T, \tag{1}$$

where $\Omega = \{x | a \le x \le b \; and \; a, b \in \mathbb{R}\}$, $\alpha(t)$ and $\beta(t)$ are continuous functions, and $U(x,t)$ is an unknown continuous function. The initial and boundary conditions of this problem are as follows.

$$U(x,0) = \Phi(x), \tag{2}$$

$$U(a,t) = \Psi 1(x,t), U(b,t) = \Psi 2(x,t), \qquad , t > 0. \tag{3}$$

The remainder of this paper is structured as follows. In Section 2, the numerical method is explained. Some numerical experiments are reported in Section 3. Section 4 is dedicated to the conclusion.

2. The method of solution

In this section, the discretization of equation (1) is explained, and the Jacobian-Free-Newton-Krylov (JFNK) method is explained to solve the obtained system of nonlinear equations.

2.1. The discretization method

Consider Δx as a grid size in the space such that $\{x_i | x_i = i\,\Delta x,\ i = 0,1,\dots,I\}$ covers Ω. Consider a positive integer N. The grid size in time (Δt) for the finite difference scheme is $\frac{T}{N}$. Consider U_i^n as the value of $U(x_i, t_n)$. The Crank–Nicolson approximation of (1) can be written as follows.

$$\frac{U_i^n - U_i^{n-1}}{\Delta t} - \frac{1}{2}\left[\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{(\Delta x)^2} + \frac{U_{i+1}^{n-1} - 2U_i^{n-1} + U_{i-1}^{n-1}}{(\Delta x)^2}\right]$$
$$+\frac{1}{2}\left[\alpha(t^n)(U^\delta)_i^n\frac{U_{i+1}^n - U_i^n}{(\Delta x)} + \alpha(t^{n-1})(U^\delta)_i^{n-1}\frac{U_{i+1}^{n-1} - U_i^{n-1}}{(\Delta x)}\right]$$
$$=\frac{1}{2}\left[\beta(t^n)U_i^n\left(1 - (U^\delta)_i^n\right) + \beta(t^{n-1})U_i^{n-1}\left(1 - (U^\delta)_i^{n-1}\right)\right], \tag{4}$$
$$i = 1,2,\dots,I, \quad n = 1,2,\ \dots,N$$

To solve the system (4) of nonlinear equations, the JFNK method is used together with the GMRES method. The main advantage of the JFNK method [24] is that using it, no cost is incurred for the creation and storage of the Jacobian matrix.

2.2. The JFNK method

The JFNK method integrates the Newton method to solve a system of nonlinear equations and a Krylov subspace method to solve the system of linear equations resulted from the Newton method. In this method, the creation and storage of the Jacobian matrix are not required. The integration of the JFNK with GMRES methods is described in [25]. Assume a system of nonlinear equations as follows

$$F(u) = 0,$$

where u is the unknown vector. In each iteration r of the Newton method, the system of linear equations

$$J^r\delta u^r = -F(u^r), \tag{5}$$

where J^r is the Jacobian matrix in iteration r, is solved by the GMRES method. Then, the approximate solution is updated as

$$u^{r+1} = u^r + \delta u^r, \qquad r = 0,1,2,\cdots,$$

where u^0 is an initial approximate solution for the Newton method. The Newton method is stopped by the following criterion

$$\|F(u^{r+1})\| \le \varepsilon_{\text{Newton}},$$

where $\varepsilon_{\text{Newton}}$ is a tolerance for stopping the Newton method. In the JFNK method, the product $J^r.x_j$ is approximated as

$$J^r.x_j \approx \frac{F(u^r + \varepsilon x_j) - F(u^r)}{\varepsilon}.$$

The value of ε can be determined by various approaches. See [26], [27], and [28]. In this work, as in [25], the following value is used.

$$\varepsilon = 10^{-8}\frac{\|u^r\|_2}{\|x_j\|_2}.$$

Algorithm 1 gives one iteration of this integrated method [29].

Algorithm 1[29]: $u^{r+1} \coloneqq JFNK(F, u^r)$

/* This Algorithm solves $A\delta u^r = b$ by the GMRES method. where $A = J^r$, $b = -F(u^r)$, u^r is the approximate solution for $F(u) = 0$. Then it updates u^r */
/* e_1 is the unit vector: $e_1 = [1\ 0\ \dots 0]^T$ */
/* H is Hessenberg matrix: $H = \{h_{ij}\}$ */
/* V is unitary matrix: $V = [v_1, v_2, \dots]$ */

1. Choose ε_G /*A tolerance to stop the GMRES method */
2. Choose δu^r /* An initial approximate solution */
   (In this experience $\delta u^r = 0$)
3. $\varepsilon \coloneqq 10^{-8}$
4. $r0 \coloneqq -F(u^r) - \frac{F(u^r+\varepsilon\delta u^r)-F(u^r)}{\varepsilon}$
5. $\gamma := \|r0\|_2$
6. $v_1 = \frac{r0}{\gamma}$
7. For $j = 1, 2, \cdots$, until convergence, Do
   - $\varepsilon := (10^{-8})\frac{\|u^r\|_2}{\|v_j\|_2}$
   - $w_j \coloneqq \frac{F(u^r+\varepsilon v_j)-F(u^r)}{\varepsilon}$
   - For i=1, $\cdots, j$, Do
     - $h_{ij} \coloneqq (w_j, v_i)$
     - $w_j \coloneqq w_j - h_{ij}v_i$
   - End For
   - $h_{j+1,j} \coloneqq \|w_j\|_2$
   - Compute y_j, as the minimizer of $\|\gamma e_1 - Hy\|_2$
   - $v_{j+1} \coloneqq \frac{w_j}{h_{j+1,j}}$
   - If $\|\gamma e_1 - Hy_j\|_2 < \varepsilon_G$
     - $\delta u^r := \delta u^r + Vy_{j,}$ Go to the step 8, End If

   End For
8. $u^{r+1} \coloneqq u^r + \delta u^r$, **End.**

3. Numerical experiments

In this section, numerical solutions of problem (1) with initial and boundary conditions (2)-(3) are computed using the approximation scheme (4). Then, these are compared with the analytical solutions.

The accuracy of this proposed, numerical method is measured in terms of the relative error

$$L_{r_\infty}(t) = \max_{1\le i\le I-1}\left|\frac{\widetilde{U}(x_i,t) - U(x_i,t)}{U(x_i,t)}\right|,$$

where $\widetilde{U}$ and U are the numerical approximation and the exact solution, respectively.

Example 1. We know from [11] that, if $\alpha, \beta, \delta \in \mathbb{R}$, then the exact solution of equation (1) is

$$U(x,t) = (\frac{1}{2} + \frac{1}{2}\tanh(\theta_1(x - \theta_2 t)))^{\frac{1}{\delta}}, \tag{6}$$

where $\theta_1 = \frac{-\alpha\delta}{2(1+\delta)}$ and $\theta_2 = \frac{\alpha}{1+\delta} + \frac{\beta(1+\delta)}{\alpha}$. Consider equation (1) for $x \in [0,1]$, with initial and boundary conditions as in (6).

In Table 1, the maximum relative errors at $t = 0.5$ and $t = 1$ with $\Delta x = \Delta t = .01$ for $\delta = 1$ and various values of α and β are presented. In Figure 1, the function U introduced in relation (6) is plotted with $\beta = 8$, $\alpha = 10^{-4}$, and $\delta = 1$.

Table 1. The maximum relative errors at $t = 0.5$ and $t = 1$ with $\delta = 1$, and various values of α and β in Example 1.

		$\alpha = 10^{-2}$	$\alpha = 10^{-4}$
t=0.5	$\beta = 1$	L_{r_∞} = 1.6618e-05	L_{r_∞} = 1.6440e-05
	$\beta = 8$	L_{r_∞} =1.1716e-06	L_{r_∞} =1.2030e-06
t=1	$\beta = 1$	L_{r_∞} =1.6308e-05	$L_{r_\infty} = 1.6319e-05$
	$\beta = 8$	$L_{r_\infty} = 4.9791e-06$	$L_{r_\infty} = 5.2182e-06$

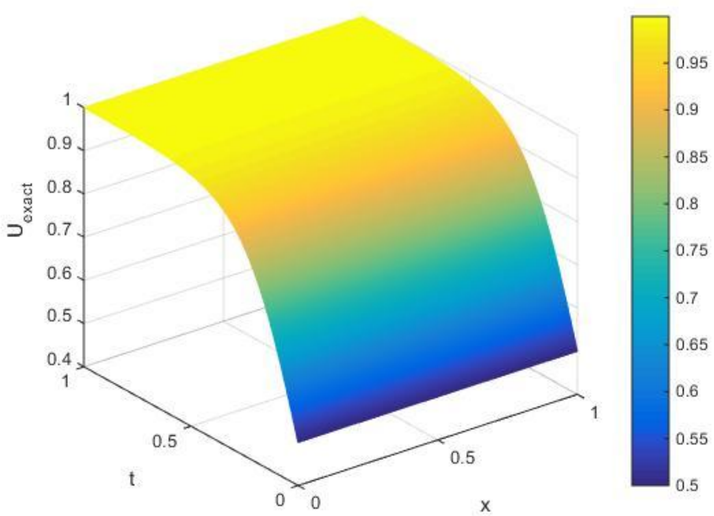

Figure 1. The function U with $\beta = 8, \alpha = 10^{-4}$, and $\delta = 1$, in Example 1.

Example 2. We know from [30] that the exact solution of equation (1) is

$$U(\mathrm{x},\mathrm{t}) = \frac{\exp(\mathrm{kx} - \mathrm{k}^2\mathrm{t}) + 1.5 - \exp(-\mathrm{kx} + \mathrm{k}^2\mathrm{t})}{\exp(\mathrm{kx} - \mathrm{k}^2\mathrm{t}) + 2}, \tag{7}$$

where $\delta = 1$, k and β are arbitrary, and $\alpha = \frac{\beta}{k}$. Consider equation (1) for $x \in [0, .5]$, with initial and boundary conditions as in (7). Table 2 shows the maximum relative errors at $t = 0.5$ and $t = 1$ with $\beta = 100 + \cos(t)$ and various values of $k, \Delta t,$ and $\Delta \mathrm{x}$. In Figures 2 and 3, the function U introduced in relation (7) is plotted with $k = 3, 4$, respectively.

Table 2. The maximum relative errors at $t = 0.5$ and $t = 1$ with $\beta = 100 + \cos(t)$, various values of $k, \Delta t,$ and $\Delta \mathrm{x}$, and $\alpha = \frac{\beta}{k}$ in Example 2.

		$\boldsymbol{\Delta t = \Delta \mathrm{x} = 0.01}$	$\boldsymbol{\Delta t = \Delta \mathrm{x} = 0.004}$	$\boldsymbol{\Delta t = \Delta \mathrm{x} = 0.002}$
t=0.5	***K=3***	L_{r_∞} =3.2864e-5	L_{r_∞} =5.5480e-6	L_{r_∞} =2.0943e-6
	K=4	L_{r_∞} =1.4554e-4	L_{r_∞} =1.8431e-5	L_{r_∞} =4.5961e-6
t=1	***K=3***	$L_{r_\infty} = 8.9482e-5$	L_{r_∞} =1.0789e-5	L_{r_∞} =2.2481e-6
	K=4	L_{r_∞} =3.0119e-4	L_{r_∞} =4.1730e-5	L_{r_∞} =1.0138e-5

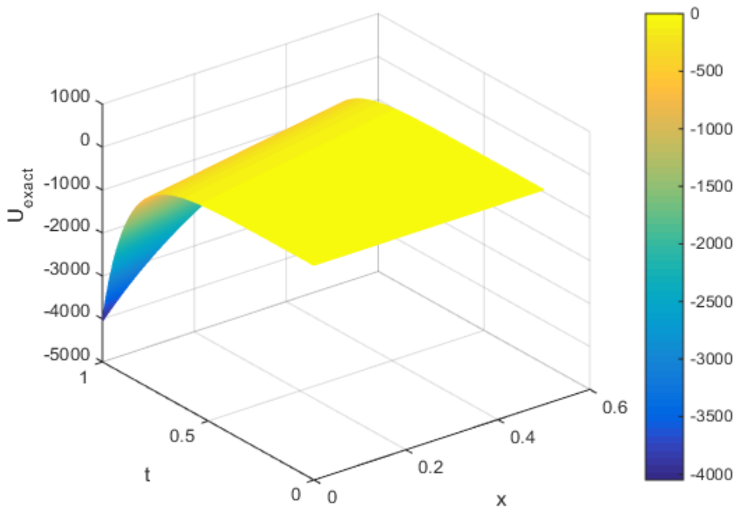

Figure 2. Function U with $k = 3$ in Example 2.

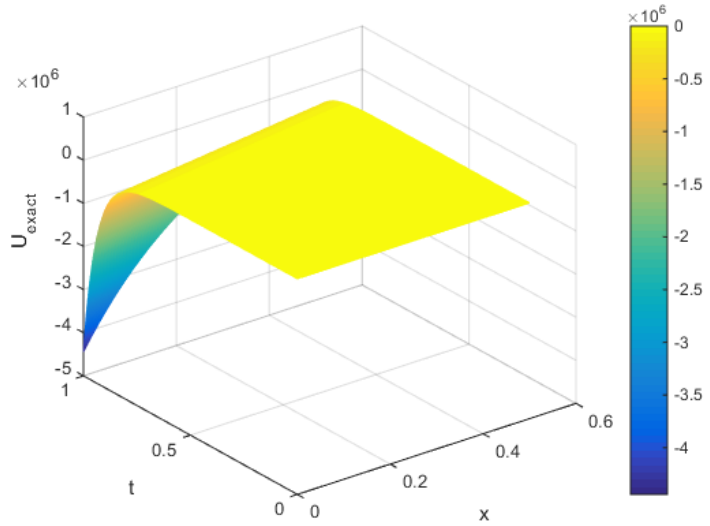

Figure 3. Function U with $k = 4$ in Example 2.

Example 3. We know from [30] that the exact solution of equation (1) is

$$U(x,t) = a_0 \exp\left(-kx + \int k^2 + \beta \, dt\right) \tag{8}$$

where $\delta = 1$, a_0, k, and β are arbitrary, and $\alpha = \frac{\beta}{k}$. Set $a_0 = 2$. Consider equation (1) for $x \in [0,1]$, with initial and boundary conditions as in (8). Table 3 shows the maximum relative errors at $t = 0.5$ and $t = 1$ with $\beta = 6t^5$ and various values of $k, \Delta t,$ and Δx. In Figure 4 and Figure 5, the function U introduced in relation (8) is plotted with $a_0 = 2$, $\beta = 6t^5$, and $k = 2, 3$, respectively.

Table 3. The maximum relative errors at $t = 0.5$ and $t = 1$ with $\beta = 6t^5$ and various values of $k, \Delta t,$ and Δx, and $\alpha = \frac{\beta}{k}$ in Example 3.

		$\boldsymbol{\Delta t = \Delta x = 0.01}$	$\boldsymbol{\Delta t = \Delta x = 0.004}$	$\boldsymbol{\Delta t = \Delta x = 0.002}$
t=0.5	**K=2**	L_{r_∞} =9.2685e-5	L_{r_∞} =1.3769e-5	L_{r_∞} =3.0017e-6
	K=3	L_{r_∞} =6.2437e-4	L_{r_∞} =9.9817e-5	L_{r_∞} =2.4877e-5
t=1	**K=2**	L_{r_∞} =1.8225e-4	$L_{r_\infty} = 2.9006e-5$	L_{r_∞} =7.2472e-6
	K=3	L_{r_∞} =8.3802e-4	L_{r_∞} =1.2849e-4	L_{r_∞} =2.9344e-5

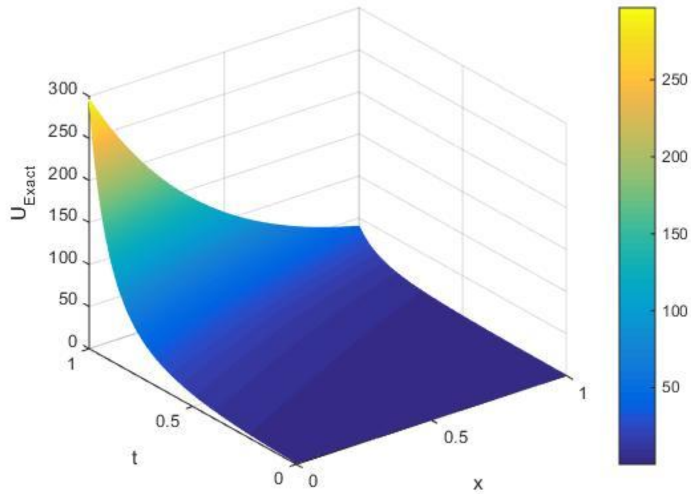

Figure 4. Function U with $a_0 = 2,\ \beta = 6t^5$ and $k = 2$ in Example 3.

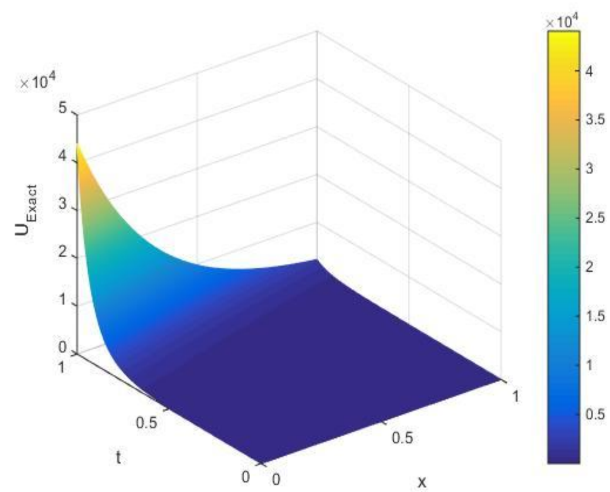

Figure 5. Function U with $a_0 = 2,\ \beta = 6t^5$ and $k = 3$ in Example 3.

Figure 1 shows that the changes of the function U are not large for $x \in [0,1]$ and $t \in [0,1]$. Figures 2, 3 show that the changes of the function U are large for $x \in [0,.5]$ and $t \in [0,1]$. Figures 4 and 5 show that the changes of the function U are large for $x \in [0,1]$ and $t \in [0,1]$. The numerical errors in Tables 1, 2, and 3 confirm that the proposed method is accurate enough. As far as the author is aware, many researchers have not tested their proposed methods on the Burger–Fisher equation that the changes of its solution function are large. The Crank-Nicholson method with the JFNK method seems to be a powerful tool for approximating the Burger–Fisher equation that the changes of its solution function are large.

Conclusion

In this paper, an implicit finite difference scheme based on the Crank–Nicolson method was used to discretize the Burger–Fisher equation with initial and boundary conditions. The JFNK method was applied for solving the system of nonlinear equations. Despite the big changes in the solution functions, numerical tests confirmed the accuracy of the proposed numerical method.

COMPETING INTERESTS

'Not applicable'

References

1. Zhu, C.-G. and W.-S. Kang, *Numerical solution of Burgers–Fisher equation by cubic B-spline quasi-interpolation.* Applied Mathematics and Computation, 2010. **216**(9): p. 2679-2686.
2. Pirdawood, M.A. and Y.A. Sabawi. *High-order solution of Generalized Burgers–Fisher Equation using compact finite difference and DIRK methods*. in *Journal of Physics: Conference Series*. 2021. IOP Publishing.
3. Gürbüz, B. and M. Sezer, *A Modified Laguerre Matrix Approach for Burgers–Fisher Type Nonlinear Equations*, in *Numerical Solutions of Realistic Nonlinear Phenomena*. 2020, Springer. p. 107-123.
4. Singh, A., S. Dahiya, and S. Singh, *A fourth-order B-spline collocation method for nonlinear Burgers–Fisher equation.* Mathematical Sciences, 2020. **14**(1): p. 75-85.
5. Mohanty, R.K. and S. Sharma, *A high-resolution method based on off-step non-polynomial spline approximations for the solution of Burgers-Fisher and coupled nonlinear Burgers' equations.* Engineering Computations, 2020.
6. Wasim, I., M. Abbas, and M. Amin, *Hybrid B-spline collocation method for solving the generalized Burgers-Fisher and Burgers-Huxley equations.* Mathematical Problems in Engineering, 2018. **2018**.
7. Yadav, O.P. and R. Jiwari, *Finite element analysis and approximation of Burgers'-Fisher equation.* Numerical Methods for Partial Differential Equations, 2017. **33**(5): p. 1652-1677.
8. Triki, H. and A.-M. Wazwaz, *Trial equation method for solving the generalized Fisher equation with variable coefficients.* Physics Letters A, 2016. **380**(13): p. 1260-1262.
9. Chandraker, V., A. Awasthi, and S. Jayaraj, *Numerical treatment of Burger-Fisher equation.* Procedia Technology, 2016. **25**: p. 1217-1225.
10. Zhang, L., L. Wang, and X. Ding, *Exact finite difference scheme and nonstandard finite difference scheme for Burgers and Burgers-Fisher equations.* Journal of Applied Mathematics, 2014. **2014**.
11. Zhang, R., X. Yu, and G. Zhao, *The local discontinuous Galerkin method for Burger's–Huxley and Burger's–Fisher equations.* Applied Mathematics and Computation, 2012. **218**(17): p. 8773-8778.
12. Kocacoban, D., et al. *A better approximation to the solution of Burger-Fisher equation*. in *Proceedings of the World Congress on Engineering*. 2011.
13. Zhang, J. and G. Yan, *A lattice Boltzmann model for the Burgers–Fisher equation.* Chaos: An Interdisciplinary Journal of Nonlinear Science, 2010. **20**(2): p. 023129.
14. Sari, M., G. Gürarslan, and İ. Dağ, *A compact finite difference method for the solution of the generalized Burgers–Fisher equation.* Numerical Methods for Partial Differential Equations: An International Journal, 2010. **26**(1): p. 125-134.
15. Rashidi, M., D. Ganji, and S. Dinarvand, *Explicit analytical solutions of the generalized Burger and Burger–Fisher equations by homotopy perturbation method.* Numerical Methods for Partial Differential Equations: An International Journal, 2009. **25**(2): p. 409-417.
16. Khattak, A.J., *A computational meshless method for the generalized Burger's–Huxley equation.* Applied Mathematical Modelling, 2009. **33**(9): p. 3718-3729.

17. Golbabai, A. and M. Javidi, *A spectral domain decomposition approach for the generalized Burger's–Fisher equation.* Chaos, Solitons & Fractals, 2009. **39**(1): p. 385-392.
18. Ismail, H.N., K. Raslan, and A.A. Abd Rabboh, *Adomian decomposition method for Burger's–Huxley and Burger's–Fisher equations.* Applied mathematics and computation, 2004. **159**(1): p. 291-301.
19. Javidi, M., *Spectral collocation method for the solution of the generalized Burger–Fisher equation.* Applied Mathematics and Computation, 2006. **174**(1): p. 345-352.
20. Wazwaz, A.-M., *The tanh method for generalized forms of nonlinear heat conduction and Burgers–Fisher equations.* Applied Mathematics and Computation, 2005. **169**(1): p. 321-338.
21. Ismail, H.N. and A.A. Abd Rabboh, *A restrictive Padé approximation for the solution of the generalized Fisher and Burger–Fisher equations.* Applied Mathematics and Computation, 2004. **154**(1): p. 203-210.
22. Kaya, D. and S.M. El-Sayed, *A numerical simulation and explicit solutions of the generalized Burgers–Fisher equation.* Applied Mathematics and computation, 2004. **152**(2): p. 403-413.
23. Mickens, R. and A. Gumel, *Construction and analysis of a non-standard finite difference scheme for the Burgers-Fisher equation.* Journal of sound and vibration, 2002. **257**(4): p. 791-797.
24. Knoll, D.A. and D.E. Keyes, *Jacobian-free Newton–Krylov methods: a survey of approaches and applications.* Journal of Computational Physics, 2004. **193**(2): p. 357-397.
25. Heyouni, M., *Newton Generalized Hessenberg method for solving nonlinear systems of equations.* Numerical Algorithms, 1999. **21**(1-4): p. 225-246.
26. Mousseau, V.A., D.A. Knoll, and W.J. Rider, *A Multigrid Newton-Krylov Solver for Non-linear Systems.* Lecture Notes in Computational Science and Engineering, 2000. **14**: p. 200-206.
27. Brown, P.N. and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations.* SIAM Journal on Scientific and Statistical Computing, 1990. **11**(3): p. 450-481.
28. Chan, T.F. and K.R. Jackson, *Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms.* SIAM Journal on Scientific and Statistical Computing, 1984. **5**(3): p. 533-542.
29. Hajinezhad, H., et al., *Numerical solution and convergence analysis of steam injection in heavy oil reservoirs.* Computational Geosciences, 2018. **22**(6): p. 1433-1444.
30. Chen, B.-K., et al., *Exp-function method for solving the Burgers-Fisher equation with variable coefficients.* arXiv preprint arXiv:1004.1815, 2010.