

RESEARCH

Open Access



An efficient numerical scheme based on Lucas polynomials for the study of multidimensional Burgers-type equations

Ihteram Ali¹, Sirajul Haq¹, Kottakkaran Sooppy Nisar^{2*} and Dumitru Baleanu^{3,4,5}

*Correspondence:

n.sooppy@psau.edu.sa

²Department of Mathematics,
College of Arts and Sciences, Prince
Sattam bin Abdulaziz University,
Wadi Aldawaser, 11991, Saudi
Arabia

Full list of author information is
available at the end of the article

Abstract

We propose a polynomial-based numerical scheme for solving some important nonlinear partial differential equations (PDEs). In the proposed technique, the temporal part is discretized by finite difference method together with θ -weighted scheme. Then, for the approximation of spatial part of unknown function and its spatial derivatives, we use a mixed approach based on Lucas and Fibonacci polynomials. With the help of these approximations, we transform the nonlinear partial differential equation to a system of algebraic equations, which can be easily handled. We test the performance of the method on the generalized Burgers–Huxley and Burgers–Fisher equations, and one- and two-dimensional coupled Burgers equations. To compare the efficiency and accuracy of the proposed scheme, we computed L_∞ , L_2 , and root mean square (RMS) error norms. Computations validate that the proposed method produces better results than other numerical methods. We also discussed and confirmed the stability of the technique.

MSC: 65Mxx

Keywords: Lucas polynomials; Fibonacci polynomials; Finite differences; Stability analysis

1 Introduction

Nonlinear partial differential equations (PDEs) are used to model many physical phenomena arising in sciences and engineering. As a result of their considerable applications and popularity, much attention has been devoted to develop an accurate and efficient numerical method for solving PDEs. Consider a one-dimensional nonlinear parabolic partial differential equation of the form

$$Y_t + \alpha Y^m Y_\xi - \beta Y_{\xi\xi} = \gamma f(Y), \quad (1)$$

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

where α, β, γ are real parameters, m is a positive integer, and $f(Y)$ is a nonlinear function. The initial and boundary conditions are

$$Y(\xi, 0) = Y_0(\xi), \quad a \leq \xi \leq b, t > 0, \tag{2}$$

$$Y(a, t) = g_1(t), \quad Y(b, t) = g_2(t). \tag{3}$$

When $f(Y) = Y(1 - Y^m)(Y^m - \epsilon)$, Eq. (1) defines the generalized Burger–Huxley (GBH) equation

$$Y_t + \alpha Y^m Y_\xi - \beta Y_{\xi\xi} = \gamma Y(1 - Y^m)(Y^m - \epsilon), \quad 0 \leq \epsilon \leq 1, t > 0. \tag{4}$$

Equation (4) describes the interaction between convection, reaction, and diffusion processes. When $\alpha = 0, \beta = 1$, and $m = 1$, Eq. (4) reduces to the Huxley equation investigating wall motion in liquid crystallography and propagation of pulse in nerve fibers. For $\gamma = 0, m = 1$, and $\beta = 1$, Eq. (4) reduces to the Burger equation used for analysis of nonlinear wave propagation, aspect of turbulence, traffic flows, and shock waves [1]. Similarly, when $f(Y) = Y(1 - Y^m)$, Eq. (1) becomes the generalized Burger Fisher (GBF) equation

$$Y_t + \alpha Y^m Y_\xi - \beta Y_{\xi\xi} = \gamma Y(1 - Y^m), \quad t > 0. \tag{5}$$

The generalized Burger–Fisher equation has a wide application in fluid mechanics, gas dynamics, plasma physics, number theory, elasticity, and heat conduction problems [2]. Equation (5) is a highly nonlinear model, which includes a combination of reaction, convection, and diffusion mechanisms. When $\gamma = 0$ and $m = 1$, Eq. (5) reduces to the Fisher equation having applications in population biology, chemistry, and biological sciences such as spreading of bacterial colonies, spread of reaction fronts in chemically bistable systems, and switching in nonlinear optics [1].

Next, we consider the following two-dimensional coupled viscous Burger equations:

$$Y_t + \mu\{Y_{\xi\xi} + Y_{\eta\eta}\} + \nu(Y Y_\xi) + \alpha(Y Z)_\xi + \gamma\{Y Y_\xi + Z Y_\eta\} = 0, \tag{6}$$

$$Z_t + \mu\{Z_{\xi\xi} + Z_{\eta\eta}\} + \nu(Z Z_\xi) + \beta(Y Z)_\xi + \gamma\{Y Z_\xi + Z Z_\eta\} = 0, \quad (\xi, \eta) \in \Gamma, \tag{7}$$

with initial and boundary conditions

$$\begin{aligned} Y(\xi, \eta, t) &= Y_0(\xi, \eta), & Z(\xi, \eta, t) &= Z_0(\xi, \eta), & (\xi, \eta) \in \Gamma, t = 0, \\ Y(\xi, \eta, t) &= h(t), & Z(\xi, \eta, t) &= g(t), & (\xi, \eta) \in \partial\Gamma, t \geq 0, \end{aligned} \tag{8}$$

where Γ and $\partial\Gamma$ represent the domain and its boundary, respectively, and μ, η, α , and β are arbitrary constants. The system was introduced by Esipov [3] to study a model of polydisperse sedimentation. This system has numerous applications in science and engineering such as gas dynamics, viscous flow of turbulence, shock waves, sedimentation of particles in fluid suspension, elasticity, and heat conduction problems [4, 5].

Many numerical methods have been applied to approximate solutions of these equations. For example, the finite difference method [6], spectral method [7], differential quadrature method [8, 9] and Adomian decomposition method [10]. Khattak [11] used a

meshfree collocation method, whereas Zhu and Kang [12] applied B-spline interpolation for solution the Burger–Fisher equation. Celik [13] studied the Haar wavelet method for solving GBH equation. Dehghan [14] worked on a mixed collocation and finite difference method. Haq et al. [15] numerically solved the Burger–Huxley equation using a meshless method of line. Zhang et al. [16] used the local discontinuous Galerkin method, whereas in [17] the author proposed a pseudospectral method for approximation of GBF equation. Wasim et al. [18] used a hybrid B-spline collocation technique for approximation of GBH and GBF equations. Mittal and Tripathi [2] proposed a cubic B-spline technique.

The modified Burger equation has been investigated in [19, 20] using a meshless method and hybrid Haar wavelet finite difference method. The author of [21] obtained approximate solution of coupled Burger equations using Adomian–Pade technique. Khater et al. [5] explained the cubic-spline collocation method for solving the coupled Burger equations. Recently, Mittal and Jiwari [22] obtained an approximate solution of one-dimensional coupled Burger equation with the help of the differential quadrature method. Dehghan et al. [23, 24] proposed a mixed finite difference and Galerkin method and multisymplectic box method for numerical study of Burgers equations. Oruc et al. [25, 26] applied a unified finite difference Chebyshev wavelet approach for time fractional Burger equations. The same authors studied the Chebyshev wavelet method for approximation of coupled Burgers equations [27]. Ali et al. [4] applied a meshfree collocation method based on the Crank–Nicolson method for time discretization and radial basis function for space discretization to solve two-dimensional coupled Burger equations, whereas the meshless method of radial basis functions (RBFs) and local RBFs were described in [28, 29] for approximate solutions of Burger-type equations. In [30] a multiscale variational algorithm was combined with the Kriging element-free Galerkin method to produce the discontinuous solutions of Burgers type equations. Srivastava et al. [31] studied a fully implicit finite difference scheme for solving two-dimensional coupled viscous Burger equations.

In this work, we compute numerical solutions of the generalized Burger–Huxley, Burger–Fisher, and coupled Burger equations using mixed Lucas and Fibonacci polynomials combined with finite differences. The main advantage of the proposed scheme is that the higher-order derivatives can be easily computed using relation of Lucas and Fibonacci polynomials. Moreover, the proposed scheme produces better accuracy for small number of collocation points, which reduces the computational cost. These polynomials have considerable applications in the area of ordinary differential equations. For example, Elhameed and Youssri [32, 33] described connection between Chebyshev and Lucas polynomials and obtained accurate solutions of boundary value problems. In [34–36] the author implemented the Lucas polynomials for solutions of fractional and coupled fractional differential equations in the Caputo sense. Mostefa [37] proposed the Lucas sequence for approximation of integro-differential equations. Cetin [38] obtained numerical solution of higher-order differential equations using the Lucas polynomial approach. Farshid et al. [39] proposed a Fibonacci polynomial approach for numerical solution of Volterra–Fredholm integral differential equations. Bayku [40] presented a hybrid Taylor–Lucas polynomial method and obtained numerical solution of delay difference equations. Oruc [41, 42] for the first time applied these polynomials for solution of time-dependent partial differential equations called a mixed Lucas and Fibonacci polynomial technique.

The rest of the paper is organized as follows. In Sect. 2, we discuss description of solution methodology. In Sect. 3, describe the stability of the method. In Sects. 4 and 5, we present numerical experiments followed by conclusion of the paper.

2 Methodology

In this section, we give a description of the proposed method for two different cases. The suggested technique will be tested by some examples.

Case.1 Nonlinear PDEs

For solution of Eq. (1) using the proposed technique, we discretize the time derivative of the equation with finite differences and apply the θ -weighted scheme to its spatial part to get

$$\begin{aligned} \frac{1}{\delta t} [Y^{n+1} - Y^n] + \theta \{ \alpha (Y^m Y_\xi)^{n+1} - \beta Y_{\xi\xi}^{n+1} - \gamma (f(Y))^{n+1} \} \\ + (1 - \theta) \{ \alpha Y^m Y_\xi - \beta Y_{\xi\xi} - \gamma f(Y) \}^n = 0, \quad 0 \leq \theta \leq 1, \end{aligned} \tag{9}$$

where $Y^n = Y(\xi, t^n)$, $t^n = n\delta t$, $n = 1, 2, \dots, N$, δt is the time step. The nonlinear term in Eq. (9) is linearized using the lagging method given as

$$(Y^m Y_\xi)^{n+1} = (Y^n)^m Y_\xi^{n+1}. \tag{10}$$

Using Eq. (10) in Eq. (9), we get

$$\begin{aligned} Y^{n+1} + \theta \delta t \{ \alpha (Y^n)^m Y_\xi^{n+1} - \beta Y_{\xi\xi}^{n+1} - \gamma (f(Y))^{n+1} \} \\ = Y^n + (\theta - 1) \delta t \{ \alpha (Y^m Y_\xi)^n - \beta Y_{\xi\xi}^n - \gamma (f(Y))^n \}. \end{aligned} \tag{11}$$

Approximating $Y^n(\xi)$ by Lucas polynomials is as follows:

$$Y^n(\xi) = \sum_{k=1}^N C_k^n L_k(\xi) = WC^n, \tag{12}$$

where C_k^n are unknown coefficients to be computed, and $L_k(\xi)$ are the Lucas polynomials defined by [41]

$$L_k(\xi) = \xi L_{k-1}(\xi) + L_{k-2}(\xi), \quad k \geq 2, \quad \text{with } L_0(\xi) = 2, \quad L_1(\xi) = \xi.$$

To determine C_k^n , we use the collocation method. At collocation points $\xi_i = a + i\delta\xi$, Eq. (12) can be written as

$$Y^n(\xi_i) = \sum_{k=0}^N C_k^n L_k(\xi_i), \quad i = 1, \dots, N. \tag{13}$$

Putting the values from Eq. (13) into Eq. (11), we obtain the following system of N equations:

$$\begin{aligned} & \sum_{k=1}^N C_k^{n+1} L_k(\xi_i) \\ & + \theta \delta t \left[\alpha \left\{ \sum_{k=1}^N C_k^n L_k(\xi_i) \right\}^m \sum_{k=1}^N C_k^{n+1} L'_k(\xi_i) - \beta \sum_{k=1}^N C_k^{n+1} L''_k(\xi_i) - \gamma (f(Y))^{n+1} \right] \\ & = \sum_{k=1}^N C_k^n L_k(\xi_i) \\ & + (\theta - 1) \delta t \left[\alpha \left\{ \sum_{k=1}^N C_k^n L_k(\xi_i) \right\}^m \sum_{k=1}^N C_k^n L'_k(\xi_i) - \beta \sum_{k=1}^N C_k^n L''_k(\xi_i) - \gamma (f(Y))^n \right]. \end{aligned} \tag{14}$$

The primes in Eq. (14) represent differentiation with respect to ξ , which allows us to replace L_k by Fibonacci polynomials [41]:

$$L'_k(\xi) = kF_k(\xi), \quad L''_k(\xi) = kF_k(\xi)D, \tag{15}$$

where $F_k(\xi)$ are the Fibonacci polynomials defined as [41].

$$F_k(\xi) = \xi F_{k-1}(\xi) + F_{k-2}(\xi) \quad \text{for } k \geq 2, \quad \text{with } F_0(\xi) = 0, \quad F_1(\xi) = 1,$$

and D is differentiation matrix given by [41]

$$D = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & d & \\ 0 & & & \end{bmatrix},$$

where d is the square matrix of order N defined by

$$d_{m,n} = \begin{cases} m(-1)^{\frac{(n-m-1)}{2}} & \text{if } n > m, n - m \text{ odd,} \\ 0 & \text{otherwise.} \end{cases}$$

Substituting the values from Eq. (15) into Eq. (14), we can write

$$\begin{aligned} & \sum_{k=1}^N C_k^{n+1} L_k(\xi_i) \\ & + \theta \delta t \left[\alpha \left\{ \sum_{k=1}^N C_k^n L_k(\xi_i) \right\}^m \sum_{k=1}^N C_k^{n+1} kF_k(\xi_i) - \beta \sum_{k=1}^N C_k^{n+1} kF_k(\xi_i)D - \gamma (f(Y))^{n+1} \right] \\ & = \sum_{k=1}^N C_k^n L_k(\xi_i) + (\theta - 1) \delta t \left[\alpha \left\{ \sum_{k=1}^N C_k^n L_k(\xi_i) \right\}^m \sum_{k=1}^N C_k^n kF_k(\xi_i) \right. \\ & \quad \left. - \beta \sum_{k=1}^N C_k^n kF_k(\xi_i)D - \gamma (f(Y))^n \right]. \end{aligned} \tag{16}$$

Similarly, Eqs. (2) and (3) can be transformed to

$$\begin{aligned} \sum_{k=1}^N C_k^1 L_k(\xi_i) &= Y_0(\xi_i), & \sum_{k=1}^N C_k^{n+1} L_k(\xi_1) &= g_1(t), \\ \sum_{k=1}^N C_k^{n+1} L_k(\xi_N) &= g_2(t) \quad i = 1, \dots, N. \end{aligned} \tag{17}$$

In matrix form, Eqs. (16)–(17) can be written as

$$HC^{n+1} = GC^n + A^{n+1}, \tag{18}$$

where $H, G,$ and A are square matrices of order N with components given by

$$H_{ik} = \begin{cases} L_k(\xi_i) + \delta t \theta \{ \alpha L_k(\xi_i) k F_k(\xi_i) - \beta k F_k(\xi_i) D - \gamma f(Y) \}, & i = 2, \dots, N-1, k = 1, 2, \dots, N, \\ L_k(\xi_i), & i = 1, N, k = 1, 2, \dots, N, \end{cases} \tag{19}$$

$$G_{ik} = \begin{cases} L_k(\xi_i) + \delta t (\theta - 1) \{ \alpha L_k(\xi_i) k F_k(\xi_i) - \beta k F_k(\xi_i) D - \gamma f(Y) \}, & i = 2, \dots, N-1, k = 1, 2, \dots, N, \\ 0, & i = 1, N, k = 1, 2, \dots, N, \end{cases} \tag{20}$$

$$A_{ik} = \begin{cases} 0, & i = 2, \dots, N-1, k = 1, 2, \dots, N, \\ g^{n+1}(\xi_i), & i = 1, N, k = 1, 2, \dots, N. \end{cases} \tag{21}$$

Solution of Eq. (18) give required unknowns $C,$ and hence a solution of problem (1) can be obtained with the help of Eq. (13).

Case 2: Coupled PDEs

To construct a scheme for coupled Burger Eqs. (6)–(7), discretizing the temporal and spatial parts in a similar way as discussed in case 1, we have

$$\begin{aligned} & \frac{Y^{n+1} - Y^n}{\delta t} + \theta [\mu (Y_{\xi\xi}^{n+1} + Y_{\eta\eta}^{n+1}) + \nu (Y Y_{\xi})^{n+1} + \alpha ((Y Z_{\xi})^{n+1} + (Z Y_{\xi})^{n+1}) \\ & \quad + \gamma [(Y Y_{\xi})^{n+1} + (Z Y_{\eta})^{n+1}]] \\ & = (\theta - 1) [\mu (Y_{\xi\xi}^n + Y_{\eta\eta}^n) + \nu (Y Y_{\xi})^n + \alpha ((Y Z_{\xi})^n + (Z Y_{\xi})^n) \\ & \quad + \gamma [(Y Y_{\xi})^n + (Z Y_{\eta})^n]], \end{aligned} \tag{22}$$

$$\begin{aligned} & \frac{Z^{n+1} - Z^n}{\delta t} + \theta [\mu (Z_{\xi\xi}^{n+1} + Z_{\eta\eta}^{n+1}) + \nu (Z Z_{\xi})^{n+1} + \beta ((Y Z_{\xi})^{n+1} + (Z Y_{\xi})^{n+1}) \\ & \quad + \gamma [(Y Z_{\xi})^{n+1} + (Z Z_{\eta})^{n+1}]] \\ & = (\theta - 1) [\mu (Z_{\xi\xi}^n + Z_{\eta\eta}^n) + \nu (Z Z_{\xi})^n + \beta ((Y Z_{\xi})^n + (Z Y_{\xi})^n) \\ & \quad + \gamma [(Y Z_{\xi})^n + (Z Z_{\eta})^n]]. \end{aligned} \tag{23}$$

For $\theta = 1/2$, these equations become the well-known Cran–Nicolson scheme with accuracy $O(\delta t^2)$ [43]. For nonlinear terms, we use the lagging method

$$(YY_\xi)^{n+1} = Y^{n+1}Y_\xi^n. \tag{24}$$

Using Eq. (24) in Eqs. (22)–(23) and waiving the error terms, we get

$$\begin{aligned} &Y^{n+1} + \delta t\theta [\mu(Y_{\xi\xi}^{n+1} + Y_{\eta\eta}^{n+1}) + \nu Y^{n+1}Y_\xi^n + \alpha(Y^{n+1}Z_\xi^n + Z^n Y_\xi^{n+1}) \\ &\quad + \gamma(Y^{n+1}Y_\xi^n + Z^n Y_\eta^{n+1})] \\ &= Y^n + (\theta - 1)\delta t [\mu(Y_{\xi\xi}^n + Y_{\eta\eta}^n) + \nu(YY_\xi)^n + \alpha((YZ_\xi)^n + (ZY_\xi)^n) \\ &\quad + \gamma(YY_\xi + ZY_\eta)^n], \end{aligned} \tag{25}$$

$$\begin{aligned} &Z^{n+1} + \delta t\theta [\mu(Z_{\xi\xi}^{n+1} + Z_{\eta\eta}^{n+1}) + \nu Z^{n+1}Z_\xi^n + \beta(Y^n Z_\xi^{n+1} + Z^{n+1}Y_\xi^n) \\ &\quad + \gamma(Y^n Z_\xi^{n+1} + Z^{n+1}Z_\eta^n)] \\ &= Z^n + (\theta - 1)\delta t [\mu(Z_{\xi\xi}^n + Z_{\eta\eta}^n) + \nu(ZZ_\xi)^n + \beta((YZ_\xi)^n + (ZY_\xi)^n) \\ &\quad + \gamma((YZ_\xi)^n + (ZZ_\eta)^n)]. \end{aligned} \tag{26}$$

Now we approximate Y^n and Z^n by Lucas polynomials as follows:

$$Y^n(\xi, \eta) = \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j), \quad Z^n(\xi, \eta) = \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j), \tag{27}$$

where λ_{km}^n and C_{km}^n are unknown coefficients, and $\xi_i = \eta_i = a + (i - 1)d\xi_i$, $d\xi = d\eta$, are the regular collocation points or the Chebyshev–Gauss–Lobatto (CGL) collocation points

$$\xi_i = \eta_i = a + \frac{b - a}{2} (1 - \cos((i - 1)\pi/M))$$

with $a = \xi_1 = \eta_1$ and $b = \xi_M = \eta_M$; $d\xi$ is spacial step size. Plugging Eq. (27) into Eqs. (25)–(26), we get

$$\begin{aligned} &\sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \\ &\quad + \delta t\theta \left\{ \mu \left(\sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k''(\xi_i) L_m(\eta_j) + \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m''(\eta_j) \right) \right. \\ &\quad + \nu \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k'(\xi_i) L_m(\eta_j) \\ &\quad + \alpha \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \\ &\quad \left. * \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k'(\xi_i) L_m(\eta_j) + \alpha \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k'(\xi_i) L_m(\eta_j) \right\} \end{aligned}$$

$$\begin{aligned}
 & + \gamma \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L'_k(\xi_i) L_m(\eta_j) \\
 & + \gamma \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L'_m(\eta_j) \Big\} \\
 = & \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \tag{28}
 \end{aligned}$$

$$\begin{aligned}
 & + \delta t(\theta - 1) \left\{ \mu \left(\sum_{k=1}^N \sum_{m=1}^N C_{km}^n L''_k(\xi_i) L_m(\eta_j) + \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L''_m(\eta_j) \right) \right. \\
 & + \nu \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L'_k(\xi_i) L_m(\eta_j) + \alpha \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \\
 & * \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L'_k(\xi_i) L_m(\eta_j) + \alpha \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L'_k(\xi_i) L_m(\eta_j) \\
 & + \gamma \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L'_k(\xi_i) L_m(\eta_j) \\
 & \left. + \gamma \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L'_m(\eta_j) \right\}, \\
 & \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \\
 & + \delta t \theta \left\{ \mu \left(\sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L''_k(\xi_i) L_m(\eta_j) + \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L''_m(\eta_j) \right) \right. \\
 & + \nu \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L'_k(\xi_i) L_m(\eta_j) + \beta \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \\
 & * \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L'_k(\xi_i) L_m(\eta_j) + \beta \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L'_k(\xi_i) L_m(\eta_j) \\
 & + \gamma \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L'_k(\xi_i) L_m(\eta_j) \\
 & \left. + \gamma \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L'_m(\eta_j) \right\} \tag{29} \\
 = & \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) + \delta t(\theta - 1) \\
 & * \left\{ \mu \left(\sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L''_k(\xi_i) L_m(\eta_j) + \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L''_m(\eta_j) \right) \right. \\
 & + \nu \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L'_k(\xi_i) L_m(\eta_j) + \beta \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j)
 \end{aligned}$$

$$\begin{aligned}
 & * \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L'_k(\xi_i) L_m(\eta_j) + \beta \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L'_k(\xi_i) L_m(\eta_j) \\
 & + \gamma \sum_{k=1}^N \sum_{m=1}^N C_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L'_k(\xi_i) L_m(\eta_j) \\
 & + \gamma \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L_m(\eta_j) \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^n L_k(\xi_i) L'_m(\eta_j) \Big\}.
 \end{aligned}$$

In these equations, “*” represents the usual product. Similarly, boundary conditions given in Eq. (8) take the form

$$\sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) = h_1(t^{n+1}), \quad \sum_{k=1}^N \sum_{m=1}^N C_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) = h_2(t^{n+1}), \tag{30}$$

$$\sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) = g_1(t^{n+1}), \quad \sum_{k=1}^N \sum_{m=1}^N \lambda_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) = g_2(t^{n+1}), \tag{31}$$

considering the relation between the Lucas and Fibonacci polynomials

$$L'_k(\xi) = kF_k(\xi), \quad L''_k(\xi) = kF_k(\xi)D, \tag{32}$$

where $F(\xi)$ and D have the same meaning as before. Putting values from Eq. (32) into Eqs. (28)–(29), the matrix form of Eqs. (28)–(31) can be written as

$$\begin{aligned}
 & A + \delta t \theta \{ \mu(Q_3 + Q_4) + \nu A * Y_\xi^n + \alpha(A * Z_\xi^n + Q_1 * Z^n) \\
 & \quad + \gamma(A * Y_\xi^n + Q_2 * Z^n) \} C^{n+1} \tag{33}
 \end{aligned}$$

$$\begin{aligned}
 & = A + \delta t(\theta - 1) \{ \mu(Q_3 + Q_4) + \nu A * Y_\xi^n + \alpha(A * Z_\xi^n + Q_1 * Z^n) \\
 & \quad + \gamma(A * Y_\xi^n + Q_2 * Z^n) \} C^n + H^{n+1},
 \end{aligned}$$

$$\begin{aligned}
 & A + \delta t \theta \{ \mu(Q_3 + Q_4) + \nu A * Z_\xi^n + \beta(A * Y_\xi^n + Q_1 * Y^n) \\
 & \quad + \gamma(Q_1 * Y^n + A * Z_\eta^n) \} \lambda^{n+1} \tag{34}
 \end{aligned}$$

$$\begin{aligned}
 & = A + \delta t(\theta - 1) \{ \mu(Q_3 + Q_4) + \nu A * Z_\xi^n + \beta(A * Y_\xi^n + Q_1 * Y^n) \\
 & \quad + \gamma(Q_1 * Y^n + A * Z_\eta^n) \} \lambda^n + G^{n+1},
 \end{aligned}$$

where

$$\begin{aligned}
 & A = L_k(\xi_i) L_m(\eta_j) \Big|_{k,m=1}^N, \quad Q_1 = kF_k(\xi_i) L_m(\eta_j) \Big|_{k,m=1}^N, \quad Q_2 = L_k(\xi_i) mF_m(\eta_j) \Big|_{k,m=1}^N, \\
 & Q_3 = kF_k(\xi_i) DL_m(\eta_j) \Big|_{k,m=1}^N, \quad Q_4 = L_k(\xi_i) mF_m D(\eta_j) \Big|_{k,m=1}^N,
 \end{aligned}$$

and

$$\begin{aligned}
 & H^{n+1} = \{ h_1^{n+1}, 0, 0, \dots, h_2^{n+1} \}^T, \\
 & G^{n+1} = \{ g_1^{n+1}, 0, 0, \dots, g_2^{n+1} \}^T.
 \end{aligned}$$

Equations (33)–(34) can be written as

$$C^{n+1} = M_1^{-1}N_1C_1^n + M_1^{-1}H^{n+1}, \tag{35}$$

$$\lambda^{n+1} = M_2^{-1}N_2C_2^n + M_2^{-1}G^{n+1}, \tag{36}$$

where

$$M_1 = A + \delta t \theta \{ \mu(Q_3 + Q_4) + \nu A * Y_\xi^n + \alpha(A * Z_\xi^n + Q_1 * Z^n) + \gamma(A * Y_\xi^n + Q_2 * Z^n) \},$$

$$N_1 = A + \delta t(\theta - 1) \{ \mu(Q_3 + Q_4) + \nu A * Y_\xi^n + \alpha(A * Z_\xi^n + Q_1 * Z^n) + \gamma(A * Y_\xi^n + Q_2 * Z^n) \},$$

$$M_2 = A + \delta t \theta \{ \mu(Q_3 + Q_4) + \nu A * Z_\xi^n + \beta(A * Y_\xi^n + Q_1 * Y^n) + \gamma(Q_1 * Y^n + A * Z_\eta^n) \},$$

$$N_2 = A + \delta t(\theta - 1) \{ \mu(Q_3 + Q_4) + \nu A * Z_\xi^n + \beta(A * Y_\xi^n + Q_1 * Y^n) + \gamma(Q_1 * Y^n + A * Z_\eta^n) \}.$$

Since $Y^n = AC^n$ and $Z^n = A\lambda^n$, we get

$$Y^{n+1} = AM_1^{-1}N_1A^{-1}Y^n + AM_1^{-1}H^{n+1}, \tag{37}$$

$$Z^{n+1} = AM_2^{-1}N_2A^{-1}Z^n + AM_2^{-1}G^{n+1}. \tag{38}$$

If M_1, M_2 are fully ranked, then M_1^{-1} and M_2^{-1} exist [44, 45]. In our computation, these matrices are fully ranked, which is checked using Matlab command $\text{rank}(M_i)$. Therefore system (35)–(36) can be solved for unknown C^n and λ^n , and a solution of original coupled equations can be obtained from Eq. (27).

3 Numerical stability analysis

To check the stability of the proposed technique, we use the matrix method. For this purpose, first rewrite Eq. (18) as follows:

$$C^{n+1} = H^{-1}GC^n + H^{-1}A^{n+1} \quad \text{for } n \geq 0. \tag{39}$$

If Y denotes the approximate solution, and u denotes the exact one, then the error is defined as

$$E^n = u^{n+1} - Y^{n+1} \quad \text{with } Y^{n+1} = WC^{n+1}. \tag{40}$$

Substituting the values from Eq. (39) into Eq. (40), we get

$$E^{n+1} = WH^{-1}GW^{-1}E^n = ME^n. \tag{41}$$

Here $M = WH^{-1}GW^{-1}$ is known as the implication matrix. Scheme (39) is stable if the matrix M satisfies the Lax–Richtmyer stability condition

$$\|M\| \leq 1. \tag{42}$$

If $\rho(\mathbf{M})$ denotes the spectral radius of a matrix \mathbf{M} , then $\rho(\mathbf{M}) \leq \|\mathbf{M}\|$. The validation of this condition is described with the help of different numerical examples in the next section.

4 Numerical examples

In this section, we apply the proposed technique to some problems. Efficiency of the scheme is tested by comparing the obtained results with exact and approximate solutions available in the literature. We study the accuracy of the method in the form of the root mean square (RMS), L_2 and L_∞ norms, and the time convergence rate given by

$$L_\infty = \max_{i=1}^N |E(\xi_i, t) - Y(\xi_i, t)|, \quad RMS = \sqrt{\frac{\sum_{i=1}^N |E(\xi_i, t) - Y(\xi_i, t)|^2}{N}},$$

$$L_2 = \sqrt{\frac{\sum_{i=1}^N |E(\xi_i, t) - Y(\xi_i, t)|^2}{d\xi}}, \quad Rate = \frac{\log(\frac{L_\infty(2\delta t, d\xi)}{L_\infty(\delta t, d\xi)})}{\log(\frac{2\delta t}{\delta t})}.$$

Example 1 Consider Eq. (4) with $\beta = 1$ and exact solution [2]

$$Y(\xi, t) = \left[\frac{\epsilon}{2} + \frac{\epsilon}{2} \tanh(a_1(\xi - a_2 t)) \right]^{1/m}, \quad t \geq 0, \tag{43}$$

where

$$a_1 = \frac{-\alpha m + m\sqrt{\alpha^2 + 4\gamma(1+m)}}{4(1+m)} \epsilon,$$

$$a_2 = \frac{\alpha \epsilon}{(1+m)} - \frac{(1+m-\epsilon)(-\alpha + \sqrt{\alpha^2 + 4\gamma(1+m)})}{2(1+m)}.$$

We obtain the approximate solution the proposed method taking special domains $[0, 1]$ and $[-10, 20]$. Initial and boundary conditions are taken from the exact solution. In this example, we discuss various cases for different values of the parameters α , γ , ϵ , and m appearing in Eq. (4). We compute the solution for both regular and CGL collocation points. We compare the computed results in the form of error norms with those available in the literature [2, 7]. From comparison it is clear that the present method gives better accuracy or comparable results with those available in the literature. We can observe from the comparison that the proposed method produces slightly more accurate results for CGL collocations points than for regular points. We also report and show in tables the comparison carried out for stability of the scheme in the form of spectral radius.

Case 1.1: In this case, we take $\alpha = \gamma = 1$, $m = 2$, and $\epsilon = 0.5$ with nodal points $N = 10$. The solution is computed over the domain $[0, 1]$ for different values of $T = 15, 30, 60$, and 120 with step size 0.01 and is shown in Tables 1 and 3. From Table 1 it is clear that the accuracy of the solution increases with time where the system remains stable, that is, $\rho(\mathbf{M}) < 1$. The computed results are compared with those of the collocation cubic B-spline method [2]. From the comparison it is obvious that the present technique gives better accuracy than the collocation cubic B-spline method.

Case 1.2: In this case, we take the parameters are $\alpha = \gamma = m = 1$ and $\epsilon = 0.5$ with nodal points $N = 10$. The solution has been computed at different time levels $T = 15, 30, 60$, and 120 with step size $\delta t = 0.01$. For accuracy of the scheme, different error norms were calculated and compared with the error norms of cubic B-spline [2] and tabulated in Tables 2

Table 1 Error norms and spectral radius for the solution of Example 1 in Case 1.1

T	Present method using CGL points					[2]	
	L_∞	L_2	RMS	CPU time	$\rho(\mathbf{M})$	L_∞	L_2
15	1.28718E-04	7.86412E-05	7.49814E-05	0.66911	0.90538	5.06827E-07	6.49722E-07
30	3.23867E-07	1.97190E-07	1.88013E-07	1.33715	0.90538	5.02830E-07	6.43856E-07
60	1.31743E-08	7.88550E-09	7.51850E-09	2.65801	0.90538	5.00811E-07	6.40890E-07
120	5.45880E-09	3.87940E-09	3.69890E-09	5.26212	0.90538	5.00144E-07	6.30883E-07

Table 2 Error norms and spectral radius for the solution of Example 1 in Case 1.2

T	Present method using CGL points					[2]	
	L_∞	L_2	RMS	CPU time	$\rho(\mathbf{M})$	L_∞	L_2
15	3.81435E-03	2.33487E-03	2.22621E-03	0.67051	0.90525	2.83295E-07	4.07965E-07
30	7.27212E-04	4.45113E-04	4.24399E-04	1.25709	0.90536	1.63593E-07	3.21668E-07
60	1.78124E-05	1.09024E-05	1.03950E-05	2.47020	0.90538	1.58949E-07	2.95373E-07
120	6.28300E-09	3.71490E-09	3.54210E-09	4.90538	0.90538	6.94339E-07	1.15759E-07

Table 3 Error norms of Example 1 using regular points

T	Case 1.1			Case 1.2		
	L_∞	L_2	RMS	L_∞	L_2	RMS
15	1.28637E-04	9.37455E-05	8.93828E-05	3.81426E-03	2.78445E-03	2.65487E-03
30	3.33900E-07	2.30756E-07	2.20017E-07	7.27194E-04	5.30825E-04	5.06122E-04
60	7.44355E-08	5.89236E-08	5.61815E-08	1.77841E-05	1.29857E-05	1.23814E-05
120	1.62962E-08	1.04962E-08	1.00077E-08	3.14269E-08	1.55306E-08	1.48079E-08

Table 4 Error norms and spectral radius for the solution of Example 1 in Case 1.3

m	T	Present method using CGL points					[7]	[2]
		L_∞	L_2	RMS	CPU time	$\rho(\mathbf{M})$	L_∞	L_∞
1	0.2	1.33150E-05	8.21325E-06	7.83102E-06	0.87604	0.99950	4.01380E-08	3.74874E-08
		3.41652E-05	2.10921E-05	2.01105E-05	0.99275	0.99901	1.31390E-05	1.22706E-05
		4.32643E-05	2.67392E-05	2.54948E-05	0.94157	0.99901	3.55400E-05	3.31915E-05
1	1	1.54346E-05	9.45976E-06	9.01953E-06	4.56786	0.99505	4.68490E-08	4.29397E-08
		4.24238E-05	2.60076E-05	2.47973E-05	4.88423	0.99505	1.53250E-05	1.40455E-05
		5.07108E-05	3.10875E-05	2.96408E-05	4.57482	0.99505	4.14070E-05	3.79493E-05

and 3. From Table 2 we observe that for small times, the results of cubic B-spline are better than those of the present method, but as time increases, the accuracy of the proposed technique increases. We can also observe from the table that the spectral radius $\rho(\mathbf{M}) < 1$ for all time levels, which shows the stability of the scheme.

Case 1.3: In this case, the solution was computed using CGL grid points for $\alpha = \gamma = 1$ and $\epsilon = 0.001$, and various values of $m = 1, 4, 8$. The spectral radius and error norms were computed at different values of $T = 0.2, 1$ and compared with the results available in the literature, which are shown in Table 4. From the table it is clear that the present method gives an excellent solution in comparison to available techniques. We can observe from the table that the present method gives the same accuracy for all values of the parameter m , whereas the accuracy of spectral and cubic B-spline methods [2, 7] suddenly decreases, which reflects the feasibility of the proposed scheme. The solution is also been computed using regular grid point and shown in Table 6.

Case 1.4: In this case, we take $\alpha = \gamma = m = 1$ and $\epsilon = 0.01$. The solution is computed over the domain $[-10, 20]$ with step size $\delta t = 0.003$. The error norms and spectral radius

Table 5 Error norms and spectral radius for the solution of Example 1 in Case 1.4

T	Present method using CGL points					[46]
	L_∞	L_2	RMS	CPU time	$\rho(\mathbf{M})$	L_∞
0.01	5.52020E-06	1.61179E-05	2.80576E-06	0.04790	0.99996	6.42780E-05
1	6.44885E-04	1.88153E-03	3.27533E-04	0.23661	0.99996	9.45430E-04
5	2.84188E-03	8.78479E-03	1.52924E-03	1.02861	0.99996	3.42880E-03
10	5.10071E-03	1.48716E-02	2.58882E-03	1.97531	0.99998	6.51530E-03

Table 6 Error norms of Example 1 using regular points

m	T	Case 1.3			T	Case 1.4		
		L_∞	L_2	RMS		L_∞	L_2	RMS
1	0.2	1.52588E-05	1.10913E-05	1.05751E-05	1	5.37444E-04	1.78976E-03	3.11557E-04
4		3.32664E-05	2.44805E-05	2.33412E-05	5	2.53004E-03	8.25125E-03	1.43636E-03
8		4.38909E-05	3.23169E-05	3.08129E-05	10	4.36676E-03	1.37152E-02	2.38750E-03

are computed for different time levels $T = 0.1, 1, 5,$ and $10,$ which are presented in Tables 5 and 6. From the table we can observe that the present method gives better results than those available in the literature even for large space and time domains. We can easily understand from these tables that as the domain increases, the spectral radius remains $\rho(\mathbf{M}) < 1,$ which shows the stability of the proposed scheme. Overall, it is obvious that the present method gives better results and is flexible to implement.

Example 2 Consider Eq. (5) with $\beta = 1.$ In this case, the exact solution [2] is given by

$$Y(\xi, t) = \left[\frac{1}{2} + \frac{1}{2} \tanh(w_1(\xi - w_2t)) \right]^{1/m}, \quad t \geq 0, \tag{44}$$

where

$$w_1 = \frac{-\alpha m}{2(1 + m)}, \quad w_2 = \frac{\alpha}{(1 + m)} + \frac{\gamma(1 + m)}{\alpha}.$$

The initial and boundary conditions are taken from the exact solution, and numerical solution is computed using the suggested technique with domain $[-1, 1].$ The error norms were computed for comparison of the proposed scheme with exact and available solutions in the literature. Two different cases were discussed for different values of the real parameter $m,$ whereas $\alpha = \gamma = \beta = 1.$

Case 2.1: In this case, the parameter $m = 1$ with step size $\delta t = 0.0001.$ The solution was obtained for $T = 0.5, 1, 2,$ and $4.$ The computed results are compared with the exact and available numerical solution in the form of error norms. The stability of the scheme was studied in the form of spectral radius shown in Tables 7 and 9. From Table 7 we can notice that the proposed scheme is stable in the given domain and gives a better accuracy than the collocation cubic B-spline method [2].

Case 2.2: Now we take $m = 2$ and compute the solution for different time levels $T = 5, 10, 15,$ and $20.$ The error norms along with spectral radius were obtained at each time level, and the results were compared with existing numerical solutions described in Tables 8 and 9. It is obvious from the table that the accuracy increases with time and converges toward the true solution. It is also clear from the table that in all the three error

Table 7 Error norms and spectral radius for the solution of Example 2 in Case 2.1

T	Present method using CGL points					[2]	
	L_∞	L_2	RMS	CPU time	$\rho(\mathbf{M})$	L_∞	L_2
0.5	1.30003E-06	1.19098E-06	8.02962E-07	2.47751	0.99977	9.17284E-06	6.95296E-06
1	2.71589E-06	2.40019E-06	1.61820E-06	4.96976	0.99975	6.85294E-06	6.13104E-06
2	1.83876E-06	1.59027E-06	1.07216E-06	9.90485	0.99973	7.30254E-06	7.22859E-06
4	2.15134E-07	1.84222E-07	1.24203E-07	19.74891	0.99972	6.44377E-07	6.70231E-07

Table 8 Error norms and spectral radius for the solution of Example 2 in Case 2.2

T	Present method using CGL points					[2]	
	L_∞	L_2	RMS	CPU time	$\rho(\mathbf{M})$	L_∞	L_2
5	8.82513E-09	7.44850E-09	5.02178E-09	2.62885	0.99728	2.13998E-06	1.18957E-06
10	2.15147E-10	2.02252E-10	1.36358E-10	5.18462	0.99728	2.13948E-06	1.18924E-06
15	1.28184E-10	9.95507E-11	6.71170E-11	7.69960	0.99728	2.13899E-06	1.18892E-06
20	9.16529E-11	8.23733E-11	5.55361E-11	10.28641	0.99728	2.13869E-06	1.18744E-06

Table 9 Error norms of Example 2 using regular points

T	Case 2.1			T	Case 2.2		
	L_∞	L_2	CPU time		L_∞	L_2	CPU time
0.5	1.3015E-06	1.4090E-06	2.54465	5	7.9591E-09	7.9238E-09	2.63473
1	2.7606E-06	2.8584E-06	5.21997	10	3.7959E-10	3.8231E-10	5.36142
2	1.8656E-06	1.8930E-06	10.30594	15	3.1634E-10	3.5744E-10	8.05531
4	2.1323E-07	2.1460E-07	20.12716	20	3.6716E-10	3.2228E-10	9.75643

norms the proposed method gives excellent results at each time level as compared to available results in the literature. The table also shows that the value of spectral radius remains less than 1, which clarifies the stability of the scheme.

Example 3 Putting $f(Y) = Y(1 - Y)(Y - 0.5)$ and $\alpha = \gamma = m = 1$ in Eq. (1), we have

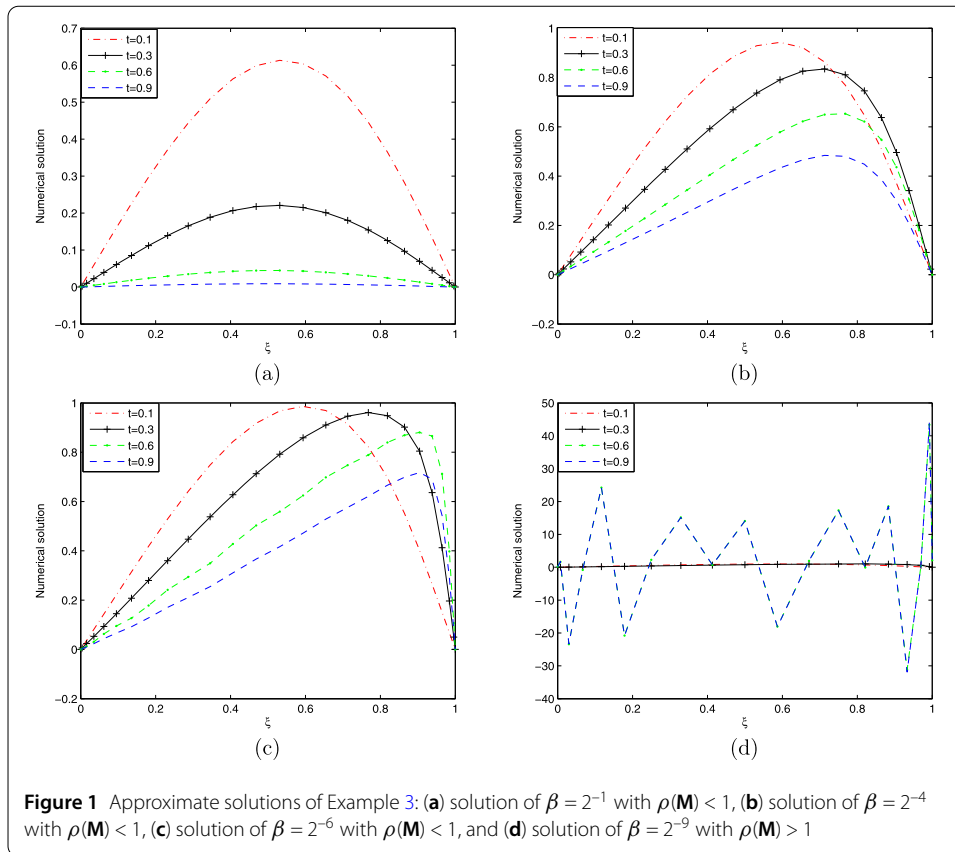
$$Y_t + YY_\xi - \beta Y_{\xi\xi} = Y(1 - Y)(Y - 0.5) \tag{45}$$

with initial and boundary conditions [2]

$$Y(\xi, 0) = \sin(\pi\xi), \quad 0 \leq x \leq 1, \tag{46}$$

$$Y(0, t) = Y(1, t) = 0, \quad t \geq 0. \tag{47}$$

The solution is computed for different time levels $T = 0.1, 0.3, 0.6,$ and 0.9 and various values of $\beta = 2^{-1}, 2^{-4}, 2^{-6},$ and 2^{-9} , respectively. The solution profile for different time levels is plotted in Fig. 1, which shows the same pattern reported in [2]. Due to unavailability of exact solution, we studied the accuracy via the stability of the method and noticed that as the value of β approaches zero, RMS solution diverges, and the scheme becomes unstable. From Figs. 1(a)–1(c) it is clear that the solution profile shows a proper behavior and the scheme is stable, that is, $\rho(\mathbf{M}) < 1$, and as the spectral radius increases, the scheme goes to an unstable region, and the solution diverges, as shown in Fig. 1(d). We can also observe from the figure that for a fixed value of β , the graph tends to zero as time increases, and for various values of β , the curve propagates to the right and follows a sharp decay. Thus the present method shows the proper behavior of Eq. (45) for various values of β and t .



The same behavior of this equation was illustrated by Mahanty and Sharma [1] and Mittal [2].

Example 4 Consider Eq. (1) with $\gamma = 0$ and $\alpha = m = 1$ to obtain

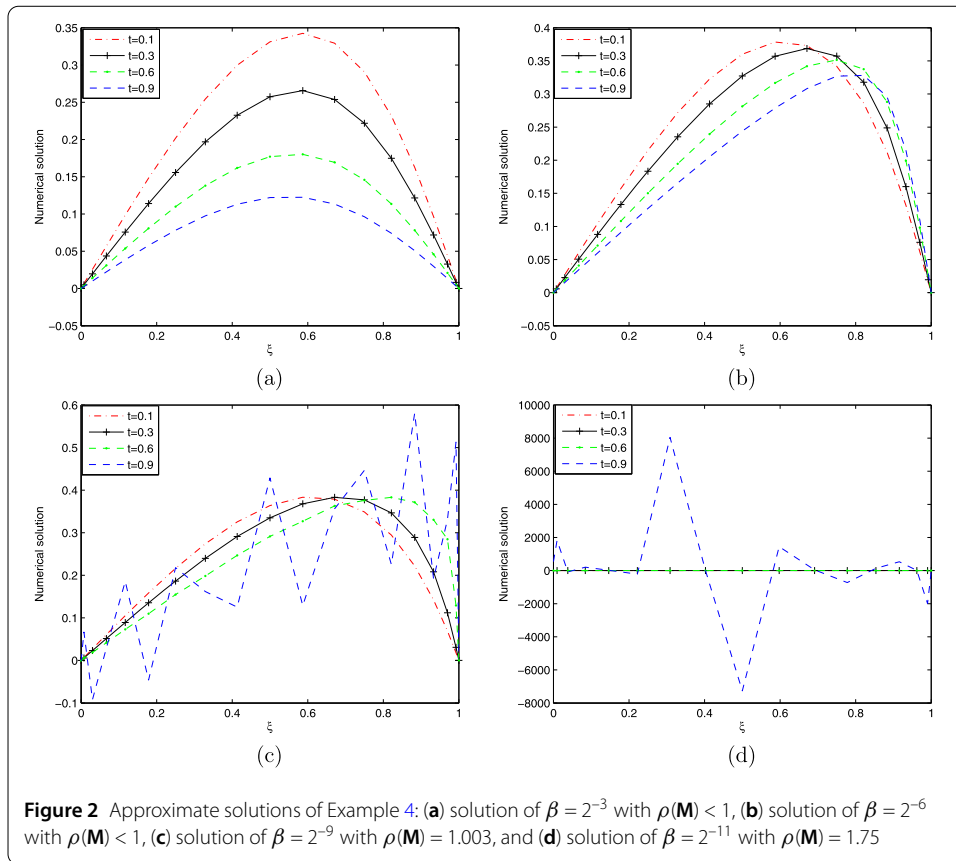
$$Y_t + YY_\xi - \beta Y_{\xi\xi} = 0 \tag{48}$$

with initial and boundary conditions [1]

$$Y(\xi, 0) = \xi(1 - \xi^2), \quad 0 \leq x \leq 1, \tag{49}$$

$$Y(0, t) = Y(1, t) = 0, \quad t \geq 0. \tag{50}$$

We computed the solution of the problem for various values of $\beta = 2^{-3}, 2^{-6}, 2^{-9}, 2^{-11}$ and time $T = 0.1, 0.3, 0.6, 0.9$ with step size $\delta t = 0.001$. The obtained results are presented in Fig. 2, which clarifies the behavior of the problem. Similarly to Example 3, We can notice that as the value of β decreases, the solution diverges, and the scheme become unstable. The same pattern of the problem has been reported by [8] and [1]. The accuracy of solution was discussed by means of the stability of the scheme.



Example 5 In this case, we consider the coupled one-dimensional Burger Eq. (7) by taking $\mu = -1$, $\nu = 2$, and $\gamma = 0$, which leads to

$$Y_t - Y_{\xi\xi} + 2YY_{\xi} + \alpha(YZ)_{\xi} = 0, \tag{51}$$

$$Z_t - Z_{\xi\xi} + 2ZZ_{\xi} + \beta(ZY)_{\xi} = 0. \tag{52}$$

The exact solution is [47, 48]

$$Y(\xi, t) = a_0(1 - \tanh(A(\xi - 2At))), \tag{53}$$

$$Z(\xi, t) = a_0 \left(\left(\frac{2\beta - 1}{2\alpha - 1} \right) - \tanh(A(\xi - 2At)) \right), \tag{54}$$

where a_0 is an arbitrary constant, and $A = \frac{1}{2}a_0 \left(\frac{4\alpha\beta - 1}{2\alpha - 1} \right)$. The initial and boundary conditions are extracted from the exact solution. The numerical solution was computed for different values of α , β , and $T = 0.5, 1$ in the domain $[-10, 10]$ with $a_0 = 0.05$ and $dt = 0.001$. For comparison, the error norms were computed and shown in Table 10. The stability of the scheme was discussed in terms of the spectral radius shown in the table. From the table it is clear that the proposed scheme is stable and produces better results even in large domains. The solution profile of Y and Z for $T = 1$ is plotted in Fig. 3. From the figure we can easily notice the betterment of the proposed technique.

Table 10 Error norms and spectral radius for $Y(\xi, t)$ and $Z(\xi, t)$ of Example 5 for $dt = 0.001$

T	Present method using CGL points						[47]		[48]	
	α	β	$\rho(M)$	L_∞	L_2	RMS	L_∞	L_2	L_∞	L_2
solution of Y										
0.5	0.1	0.3	1.00030	2.04E-06	1.80E-06	1.74E-06	4.17E-05	6.74E-04	9.62E-04	3.25E-05
	0.3	0.03	1.00003	6.45E-06	3.66E-06	3.54E-06	4.59E-05	7.33E-04	4.31E-04	2.73E-05
1	0.1	0.3	1.00077	4.03E-06	3.53E-06	3.41E-06	8.26E-05	1.33E-03	1.15E-03	2.41E-05
	0.3	0.03	1.00024	1.29E-05	7.18E-06	6.96E-06	9.18E-05	1.45E-03	1.27E-03	5.83E-05
solution of Z										
0.5	0.1	0.3	1.00003	1.75E-06	8.68E-07	8.40E-07	1.48E-04	9.06E-04	3.33E-04	2.75E-05
	0.3	0.03	1.00004	1.05E-05	8.36E-06	8.09E-06	5.73E-04	1.59E-03	1.15E-03	2.45E-04
1	0.1	0.3	1.00007	3.51E-06	1.71E-06	1.66E-06	4.77E-05	1.25E-03	1.16E-03	3.75E-05
	0.3	0.03	1.00002	2.10E-05	1.64E-05	1.59E-05	3.62E-04	2.25E-03	1.64E-03	4.53E-04

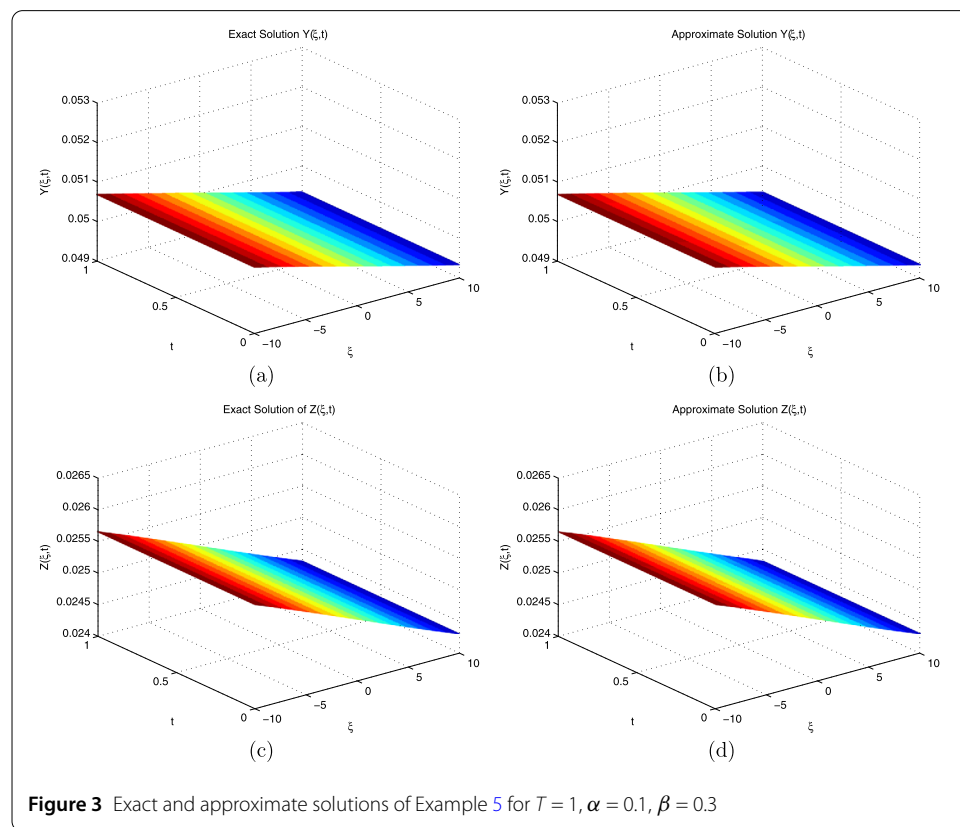


Figure 3 Exact and approximate solutions of Example 5 for $T = 1, \alpha = 0.1, \beta = 0.3$

Example 6 In this example, we consider one-dimensional Eq. (7) with $\mu = -1, \nu = -2, \gamma = 0,$ and $\alpha = \beta = 1$ which leads to

$$Y_t - Y_{\xi\xi} - 2YY_\xi + (YZ)_\xi = 0, \tag{55}$$

$$Z_t - Z_{\xi\xi} - 2ZZ_\xi + (ZY)_\xi = 0, \tag{56}$$

with exact solution [22]

$$Y(\xi, t) = e^{-t} \sin(\xi), \tag{57}$$

$$Z(\xi, t) = e^{-t} \sin(\xi). \tag{58}$$

Table 11 Error norms and spectral radius for $Y(\xi, t)$ and $Z(\xi, t)$ of Example 6 for $dt = 0.001$

T	Present method using CGL points					[22]
	L_∞	L_2	RMS	$\rho(\mathbf{M})$	CPU time	L_∞
solution of Y						
0.5	2.52724E-06	3.95542E-06	1.53590E-06	1.00003	2.55860	1.51688E-04
1	3.06570E-06	1.91419E-06	1.86313E-06	1.00047	2.66805	1.83970E-04
2	2.25560E-06	3.53027E-06	1.37081E-06	0.99855	2.92445	1.35250E-04
3	1.24468E-06	1.94806E-06	7.56436E-07	0.99788	3.16637	7.46014E-05
solution of Z						
0.5	2.52724E-06	3.95542E-06	1.53590E-06	1.00003	2.55860	1.51688E-04
1	3.06570E-06	1.91419E-06	1.86313E-06	1.00047	2.66805	1.83970E-04
2	2.25560E-06	3.53027E-06	1.37081E-06	0.99855	2.92445	1.35250E-04
3	1.24468E-06	1.94806E-06	7.56436E-07	0.99788	3.16637	7.46014E-05

Table 12 Convergence rate of maximum error of Example 6 at $T = 1$

dt	Solution of u			Solution of v		
	L_∞	Rate	CPU time	L_∞	Rate	CPU time
1/10	3.06899E-04	-	2.46295	3.06899E-04	-	2.46295
1/20	7.66623E-05	2.00117	2.44621	7.66623E-05	2.00117	2.44621
1/40	1.91617E-05	2.00029	2.49829	1.91617E-05	2.00029	2.49829
1/80	4.79018E-06	2.00007	2.57567	4.79018E-06	2.00007	2.57567
1/160	1.19753E-06	2.00002	2.83212	1.19753E-06	2.00002	2.83212

The initial and boundary conditions are extracted from the exact solution. The numerical solution was obtained for time levels $T = 0.5, 1$ in the domain $[-\pi, \pi]$. The obtained results were compared in the form of error norms with those of the differential quadrature method [22] and are shown in Table 11. The rate of convergence using CGL collocation points is shown in Table 12. From Table 11 we can observe that the present method is stable and produces a better solution than the available techniques. In Fig. 4 the numerical and exact solutions of Y and Z are plotted for $T = 1$. The figures reflect a good agreement of the obtained numerical result with exact solution.

Example 7 In this case, we take Eq. (7) with $\nu = 0, \alpha = 0, \beta = 0, \gamma = 1$, and $\mu = -1/Re$, which leads to the following two-dimensional coupled Burger equation

$$Y_t + YY_\xi + ZY_\eta - \frac{1}{Re}(Y_{\xi\xi} + Y_{\eta\eta}) = 0, \tag{59}$$

$$Z_t + YZ_\xi + ZZ_\eta - \frac{1}{Re}(Z_{\xi\xi} + Z_{\eta\eta}) = 0. \tag{60}$$

The exact solution is

$$Y(\xi, \chi, t) = 0.75 - 0.25 \left[1 + \exp \left((-4\xi + 4\eta - t) \frac{Re}{32} \right) \right]^{-1}, \tag{61}$$

$$Z(\xi, \chi, t) = 0.75 + 0.25 \left[1 + \exp \left((-4\xi + 4\eta - t) \frac{Re}{32} \right) \right]^{-1}. \tag{62}$$

The initial and boundary conditions are taken from the exact solution. The numerical solution was computed in the domain $[0, 1] \times [0, 1]$ for different values of nodal points M and Reynolds number Re when $T = 0.01$. The spectral radius and error norms were

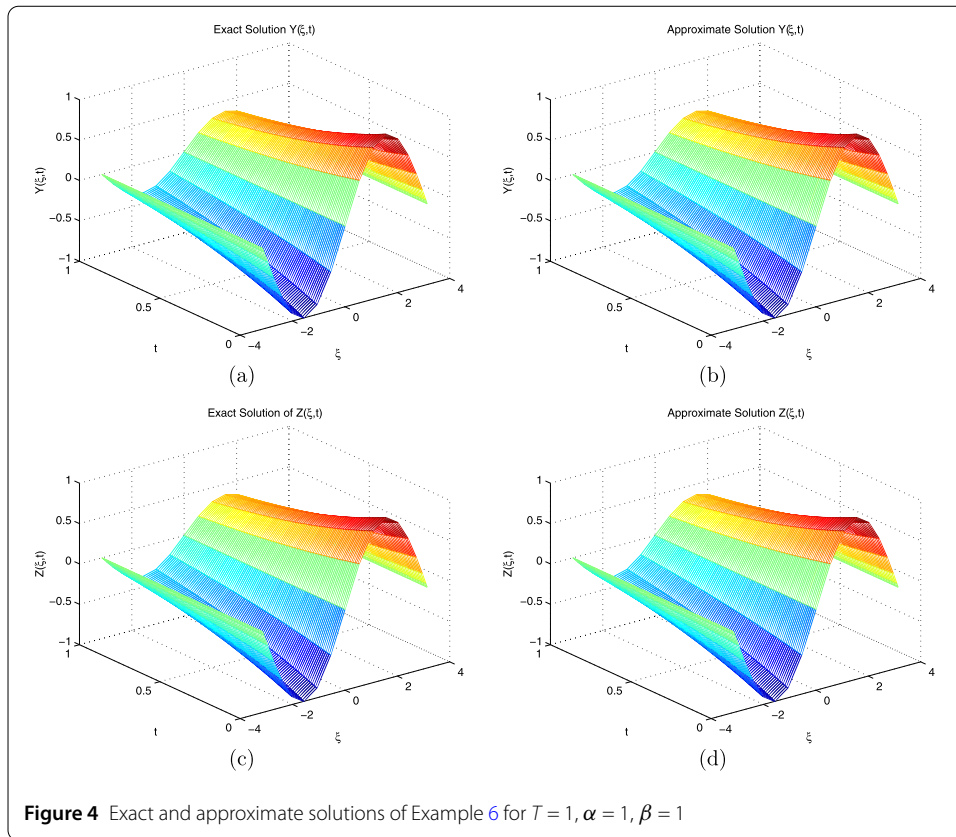


Table 13 Error norms and spectral radius for $Y(\xi, \eta, t)$ and $Z(\xi, \eta, t)$ of Example 7 for $T = 0.01$ using regular points

N	Present method							[4]
	dt	Re	L_∞	L_2	RMS	$\rho(\mathbf{M})$	CPU time	L_∞
solution of Y								
100	0.0001	1	4.85E-05	3.20E-05	1.28E-05	1.00000	1.28383	2.29E-04
100	0.005	10	2.33E-04	1.77E-04	3.01E-05	1.00003	0.07259	7.23E-05
400	0.001	100	6.43E-03	2.06E-03	1.03E-04	1.00030	9.37317	8.80E-05
solution of Z								
100	0.0001	1	3.93E-05	2.91E-05	1.22E-05	1.00000	-	3.23E-04
100	0.005	10	4.64E-04	3.50E-04	4.23E-05	1.00005	-	1.03E-04
400	0.001	100	6.83E-03	2.22E-03	1.07E-04	1.00034	-	1.00E-04

computed for $Re = 1, 10,$ and 100 and nodal points $N = 100, 400,$ that is, $N = (10 \times 10)$ and $(20 \times 20),$ and compared with the error norms obtained by Arshad [4] using the meshfree technique presented in Table 13. From the table we notice that the present results are more accurate when $Re = 1,$ whereas the accuracy decreases as Re increases with increasing spectral radius. The solution and error plots are shown in Fig. 5, which shows a kink-like behavior for $Re = 50.$

Example 8 Finally, we study two-dimensional coupled Burger equations (59)–(60) in the domain $[0, 1] \times [0, 1]$ with the following initial and boundary conditions taken from [4]:

$$Y(\xi, \eta, 0) = \sin(\pi \xi) \sin(\pi \eta),$$

$$Z(\xi, \eta, 0) = [\sin(\pi \xi) + \sin(2\pi \xi)][\sin(\pi \eta) + \sin(2\pi \eta)],$$

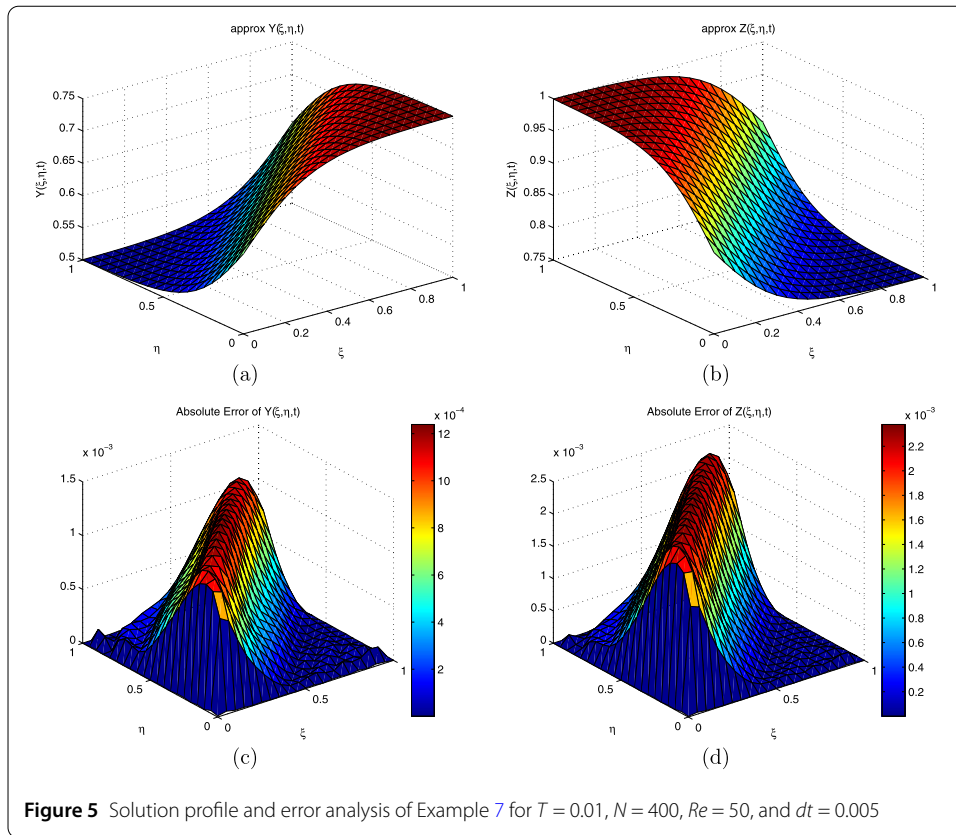


Figure 5 Solution profile and error analysis of Example 7 for $T = 0.01$, $N = 400$, $Re = 50$, and $dt = 0.005$

$$Y(0, \eta, t) = Y(1, \eta, t) = Y(\xi, 0, t) = Y(\xi, 1, t) = 0, \quad t > 0,$$

$$Z(0, \eta, t) = Z(1, \eta, t) = Z(\xi, 0, t) = Z(\xi, 1, t) = 0, \quad t > 0.$$

The problem was solved using the proposed technique for nodal points $N = 100, 400$, that is, $N = (10 \times 10)$ and (20×20) at time $t = 0.01$, and $Re = 1$. Due to the nonavailability of the exact solution, the obtained results were compared at different collocation points with the numerical solution by the meshfree method [4] and finite element technique [49] shown in Table 14. From the table it is clear that the proposed method produces almost the same results as those of existing methods. The solution profile for different time levels $t = 0, 0.01, 0.05$ at fixed values of η are plotted in Fig. 6. From the figure we can observe that as the time increases, $Z(\xi, \eta, t)$ moves from the negative part to the positive one, and the graphs tend to zero. Similarly, a 3D plot of the solution is shown in Fig. 7.

5 Conclusion

In this paper, we studied a numerical method based on the Lucas polynomials and computed solutions of three different models, including the generalized Burger–Huxley equation, generalized Burger–Fisher equation, and one- and two-dimensional nonlinear coupled Burger equations. The dependent variable is approximated by the Lucas polynomials, whereas the Fibonacci polynomials are used for its derivatives. We discussed the stability of the proposed scheme in the form of spectral radius. For comparison of the proposed method, we computed the error norms in different domains and compared the results with

Table 14 Comparison of the values of $Y(\xi, \eta, t)$ and $Z(\xi, \eta, t)$ of Example 8 for $T = 0.01$ using regular points

Points	Present method		[49]		[4]	
	$N = 100$ $dt = 0.0002$	$N = 400$ 0.0001	$N = 100$ $dt = 0.0002$	$N = 400$ 0.0001	$N = 100$ $dt = 0.00125$	$N = 400$ -
solution of Y						
(0.1,0.1)	0.07299	0.07260	0.07279	0.07257	0.07254	0.07251
(0.2,0.8)	0.28886	0.28861	0.28867	0.28842	0.27778	0.27778
(0.4,0.4)	0.72340	0.72256	0.72375	0.7221	0.72174	0.72174
(0.7,0.1)	0.20148	0.20125	0.20157	0.20117	0.20481	0.20484
(0.9,0.9)	0.07939	0.07945	0.07951	0.07947	0.07942	0.07944
solution of Z						
(0.1,0.1)	0.43027	0.43239	0.44132	0.44336	0.43159	0.43087
(0.2,0.8)	-0.1242	-0.1225	-0.13172	-0.12366	-0.12428	-0.1241
(0.4,0.4)	1.65850	1.65514	1.66212	1.65499	1.65245	1.65244
(0.7,0.1)	0.06571	0.06662	0.06306	0.06621	0.06716	0.06705
(0.9,0.9)	0.01212	0.01364	0.01459	0.01367	0.01358	0.01335

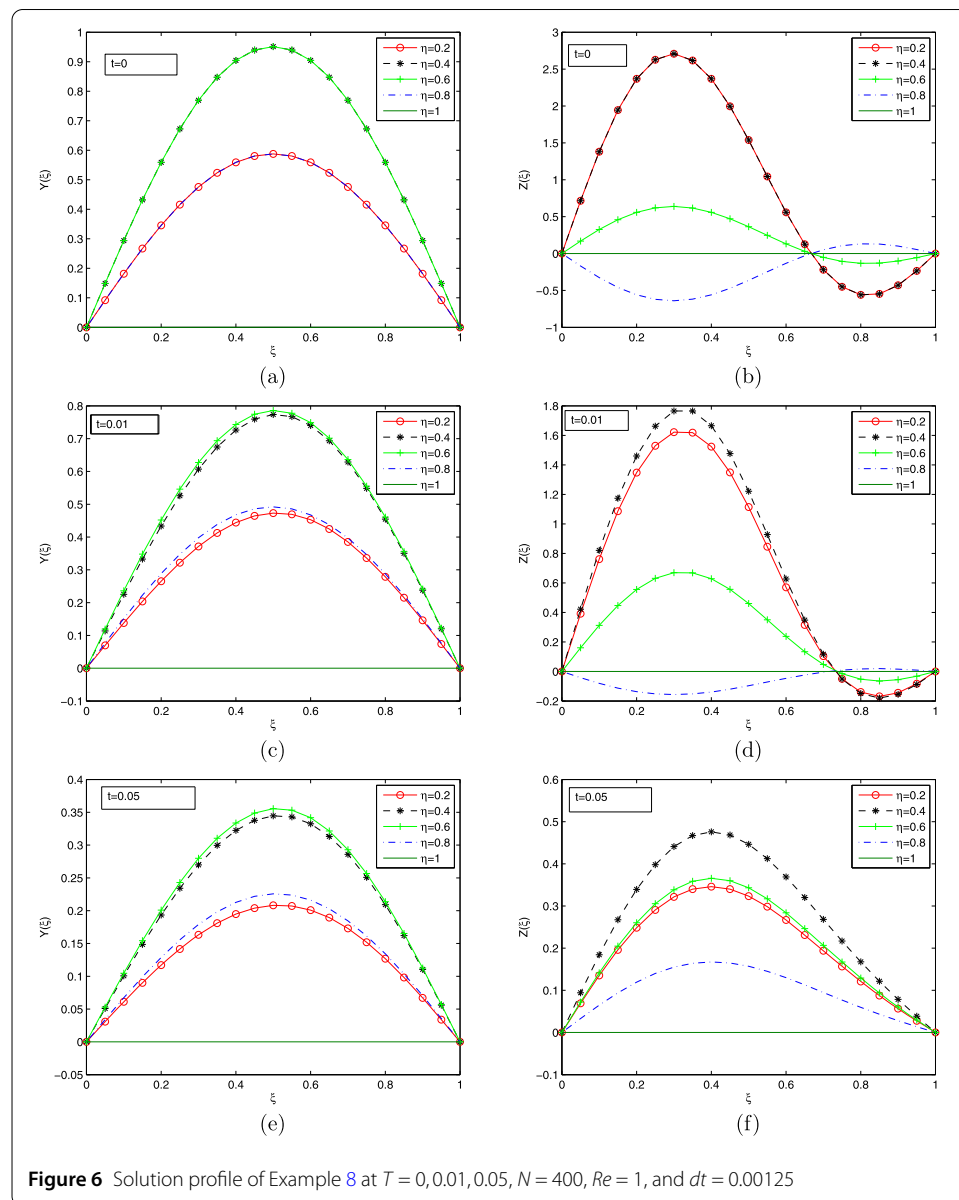
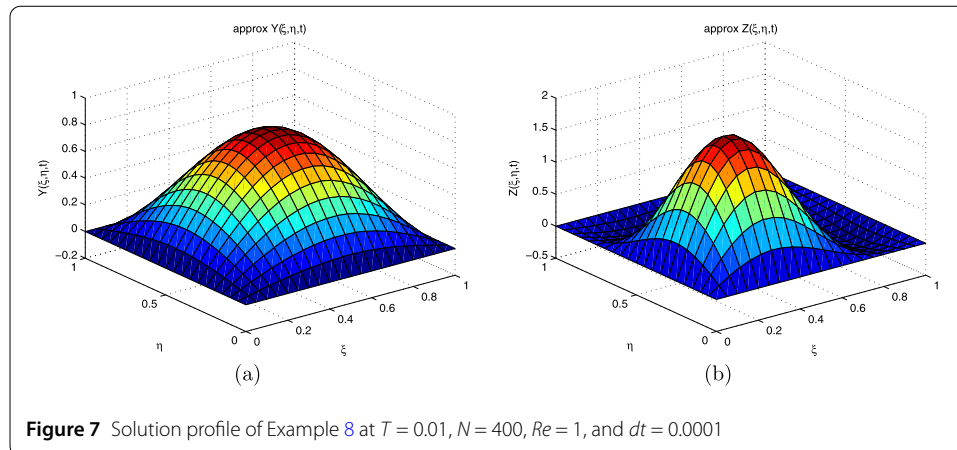


Figure 6 Solution profile of Example 8 at $T = 0, 0.01, 0.05$, $N = 400$, $Re = 1$, and $dt = 0.00125$



exact and available results in the literature. From comparison it is clear that the proposed technique gives a better accuracy and is easy to implement.

Acknowledgements

The authors would like to express their sincere thanks for the financial support of Department of Mathematics, Prince Sattam bin Abdulaziz University, Saudi Arabia, and to the Institute of Space Sciences, 077125, Magurele-Bucharest, Romania.

Funding

Not applicable.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All the authors have read and approved the final manuscript.

Author details

¹Faculty of Engineering Sciences, GIK Institute, Topi 23640, KPK, Pakistan. ²Department of Mathematics, College of Arts and Sciences, Prince Sattam bin Abdulaziz University, Wadi Aldawaser, 11991, Saudi Arabia. ³Department of Mathematics, Cankaya University, Ankara 06790, Turkey. ⁴Institute of Space Sciences, 077125, Magurele-Bucharest, Romania. ⁵Department of Medical Research, China Medical University Hospital, China Medical University, Taichung, 40447, Taiwan.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 28 July 2020 Accepted: 3 December 2020 Published online: 11 January 2021

References

- Mahonty, R.K., Sharma, S.: High-accuracy quasi-variable mesh method for the system of 1D quasi-linear parabolic partial differential equations based on off-step spline in compression approximations. *Adv. Differ. Equ.* **2017**, 212 (2017)
- Mittal, R.C., Tripathia, A.: Numerical solutions of generalized Burgers–Fisher and generalized Burgers–Huxley equations using collocation of cubic B-splines. *Int. J. Comput. Math.* (2014)
- Esipov, S.E.: Coupled Burgers equations: a model of polydisperse sedimentation. *Phys. Rev. E* **52**, 3711–3718 (1995)
- Ali, A., Islam, S., Haq, S.: A computational meshfree technique for the numerical solution of the two-dimensional coupled Burgers' equations. *Comput. Methods Appl. Mech. Eng.* **10**, 406–422 (2009)
- Khater, A.H., Tamsah, R.S., Hassan, M.M.: A Chebyshev spectral collocation method for solving Burger's-type equations. *J. Comput. Appl. Math.* **222**, 333–350 (2008)
- Sari, M., Gürarslan, G., Dag, I.: A compact finite difference method for the solution of the generalized Burgers–Fisher equation. *Numer. Methods Partial Differ. Equ.* **26**(1), 125–134 (2010)
- Javidi, M.: A numerical solution of the generalized Burgers–Huxley equation by spectral collocation method. *Appl. Math. Comput.* **178**(2), 338–344 (2006)
- Mittal, R.C., Jiwari, R.: Numerical study of Burger–Huxley equation by differential quadrature method. *Int. J. Appl. Math. Mech.* **5**, 1–9 (2009)

9. Tomasiello, S.: Numerical solutions of the Burgers–Huxley equation by the IDQ method. *Int. J. Comput. Math.* **87**(1), 129–140 (2010)
10. Ismail, H.N.A., Raslan, K., Abd Rabboh, A.A.: Adomian decomposition method for Burger’s–Huxley and Burger’s–Fisher equations. *Appl. Math. Comput.* **159**(1), 291–301 (2004)
11. Khattak, A.J.: A computational meshless method for the generalized Burger’s–Huxley equation. *Appl. Math. Model.* **33**(9), 3718–3729 (2009)
12. 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)
13. Celik, I.: Haar wavelet method for solving generalized Burgers–Huxley equation. *Arab J. Math. Sci.* **18**, 25–37 (2011)
14. Dehghan, M., Saray, B.N., Lakestani, M.: Three methods based on the interpolation scaling functions and the mixed collocation finite difference schemes for the numerical solution of the nonlinear generalized Burgers–Huxley equation. *Math. Comput. Model.* **55**, 1129–1142 (2011)
15. Haq, S., Hussain, A., Uddin, M.: On the numerical solution of nonlinear Burgers’-type equations using meshless method of lines. *Appl. Math. Comput.* **218**, 6280–6290 (2011)
16. 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)
17. Zhao, T., Li, C., Zang, Z., Wu, Y.: Chebyshev–Legendre pseudo-spectral method for the generalised Burgers–Fisher equation. *Appl. Math. Model.* **36**(3), 1046–1056 (2012)
18. Wasim, I., Abbas, M., Amin, M.: Hybrid B-spline collocation method for solving the generalized Burgers–Fisher and Burgers–Huxley equations. In: *Mathematical Problems in Engineering* (2018). <https://doi.org/10.1155/2018/6143934>.
19. Oruc, O.: Two meshless methods based on pseudo spectral delta-shaped basis functions and barycentric rational interpolation for numerical solution of modified Burgers equation. *Int. J. Comput. Math.*, 1–19 (2020)
20. Oruc, O., Bulut, F., Esen, A.: A Haar wavelet-finite difference hybrid method for the numerical solution of the modified Burgers’ equation. *J. Math. Chem.* **53**(7), 1592–1607 (2015)
21. Dehghan, M., Hamidi, A., Shakourifar, M.: The solution of coupled Burger’s equations using Adomian–Pade technique. *Appl. Math. Comput.* **189**, 1034–1047 (2007)
22. Mittal, R.C., Jiwari, R.: Differential quadrature method for numerical solution of coupled viscous Burgers’ equations. *Int. J. Comput. Methods Eng. Sci. Mech.* **13**, 88–92 (2012)
23. Dehghan, M., Saray, B.N., Lakestani, M.: Mixed finite difference and Galerkin methods for solving Burgers equations using interpolating scaling functions. *Math. Methods Appl. Sci.* **37**(6), 894–912 (2014)
24. Tabatabaei, A.H.A., Shakour, E., Dehghan, M.: Some implicit methods for the numerical solution of Burgers equation. *Appl. Math. Comput.* **191**(2), 560–570 (2007)
25. Esen, A., Bulut, F., Oruc, O.: A unified approach for the numerical solution of time fractional Burgers’ type equations. *Eur. Phys. J. Plus* **131**(4), 116 (2016)
26. Oruc, O., Esen, A., Bulut, F.: A unified finite difference Chebyshev wavelet method for numerically solving time fractional Burgers’ equation. *Discrete Contin. Dyn. Syst., Ser. S* **12**(3), 533–542 (2019)
27. Oruc, O., Bulut, F., Esen, A.: Chebyshev wavelet method for numerical solutions of coupled Burgers equation. *Hacet. J. Math. Stat.* **48**(1), 1–16 (2019)
28. Dehghan, M., Abbaszadeha, M., Mohebbib, A.: The numerical solution of nonlinear high dimensional generalized Benjamin–Bona–Mahony–Burgers equation via the meshless method of radial basis functions. *Comput. Math. Appl.* **68**(3), 212–237 (2014)
29. Dehghan, M., Abbaszadeha, M.: The space-splitting idea combined with local radial basis function meshless approach to simulate conservation laws equations. *Alex. Eng. J.* **57**(2), 1137–1156 (2018)
30. Dehghan, M., Abbaszadeha, M.: Variational multiscale element-free Galerkin method combined with the moving Kriging interpolation for solving some partial differential equations with discontinuous solutions. *Comput. Appl. Math.* **37**(3), 3869–3905 (2018)
31. Srivastava, V.K., Awasthi, M.K., Tamsir, M.: A fully implicit Finite-difference solution to one dimensional Coupled Nonlinear Burgers’ equations. *Int. J. Math. Comput. Sci. Eng.*, 7 (2013)
32. Abd-Elhameed, W.M., Youssri, Y.H.: New connection formulae between Chebyshev and Lucas polynomials: new expressions involving Lucas numbers via hypergeometric functions. *Adv. Stud. Contemp. Math.* **28**(3), 357–367 (2018)
33. Abd-Elhameed, W.M., Youssri, Y.H.: Connection formulae between generalized Lucas polynomials and some Jacobi polynomials: application to certain types of fourth-order BVPs. *Int. J. Appl. Comput. Math.* **6**(2), 1–19 (2020)
34. Abd-Elhameed, W.M., Youssri, Y.H.: Spectral solutions for fractional differential equations via a novel Lucas operational matrix of fractional derivatives. *Rom. J. Phys.* **61**, 795–813 (2016)
35. Abd-Elhameed, W.M., Youssri, Y.H.: Spectral tau algorithm for certain coupled system of fractional differential equations via generalized Fibonacci polynomial sequence. *Iran. J. Sci. Technol. Trans. A, Sci.* **43**, 543–554 (2019)
36. Abd-Elhameed, W.M., Youssri, Y.H.: Generalized Lucas polynomial sequence approach for fractional differential equations. *Nonlinear Dyn.* **89**(2), 1341–1355 (2017)
37. Nadir, M.: Lucas polynomials for solving linear integral equations. *J. Theor. Appl. Comput. Sci.* **11**, 13–19 (2017)
38. Cetin, M., Sezer, M., Guler, C.: Lucas polynomial approach for system of high-order linear differential equations and residual error estimation. *Math. Probl. Eng.* **2015**, Article ID 625984 (2015)
39. Mirzaee, F., Hoseini, S.F.: Application of Fibonacci collocation method for solving Volterra–Fredholm integral equations. *Appl. Math. Comput.* **273**, 637–644 (2016)
40. Baykus, N., Sezer, M.: Hybrid Taylor–Lucas collocation method for numerical solution of high-order pantograph type delay differential equations with variables delays. *Appl. Math. Inf. Sci.* **11**, 1795–1801 (2017)
41. Oruc, O.: A new algorithm based on Lucas polynomials for approximate solution of 1D and 2D nonlinear generalized Benjamin–Bona–Mahony–Burgers equation. *Comput. Math. Appl.* **74**(12), 3042–3057 (2017)
42. Oruc, O.: A new numerical treatment based on Lucas polynomials for 1D and 2D sinh-Gordon equation. *Commun. Nonlinear Sci. Numer. Simul.* **57**, 14–25 (2018)
43. Burden, R.L., Faires, J.D.: *Numerical Analysis*, Ninth Edition
44. Nicholson, W.K.: *Linear Algebra with Applications*, Open Edition

45. Gumgum, S., Savasaneiril, N.B., Kurkcu, O.K., Sezer, M.: Lucas polynomial solution for neutral differential equations with proportional delays. *TWMS J. Appl. Eng. Math.* **10**(1), 259–269 (2020)
46. Duan, Y., Kong, L., Zhang, R.: A lattice Boltzmann model for the generalized Burgers–Huxley equation. *Physica A* **391**(3), 625–632 (2012)
47. Mittal, R.C., Arora, G.: Numerical solution of the coupled viscous Burgers' equation. *Commun. Nonlinear Sci. Numer. Simul.* **16**, 1304–1313 (2011)
48. Rashid, A., Ismail, A.I.B.: A Fourier pseudospectral method for solving coupled viscous Burgers equations. *Comput. Methods Appl. Math.* **9**(4), 412–420 (2009)
49. Arminjon, P., Beauchamp, C.: Numerical solution of Burgers' equations in two-space dimensions. *Comput. Methods Appl. Mech. Eng.* **19**(3), 351–365 (1979)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
