



Ihteram Ali · Sirajul Haq · Kottakkaran Sooppy Nisar ·  
Shams Ul Arifeen

# Numerical study of 1D and 2D advection-diffusion-reaction equations using Lucas and Fibonacci polynomials

Received: 26 February 2021 / Accepted: 30 June 2021 / Published online: 27 July 2021  
© The Author(s) 2021

**Abstract** In this work, a numerical scheme based on combined Lucas and Fibonacci polynomials is proposed for one- and two-dimensional nonlinear advection–diffusion–reaction equations. Initially, the given partial differential equation (PDE) reduces to discrete form using finite difference method and  $\theta$ – weighted scheme. Thereafter, the unknown functions have been approximated by Lucas polynomial while their derivatives by Fibonacci polynomials. With the help of these approximations, the nonlinear PDE transforms into a system of algebraic equations which can be solved easily. Convergence of the method has been investigated theoretically as well as numerically. Performance of the proposed method has been verified with the help of some test problems. Efficiency of the technique is examined in terms of root mean square (RMS),  $L_2$  and  $L_\infty$  error norms. The obtained results are then compared with those available in the literature.

**Mathematics Subject Classification** 65Mxx

## 1 Introduction

Partial differential equations (PDEs) have dominant applications in various physical, chemical, and biological processes. These processes are mostly modeled in the form of heat, Laplace, and wave-type differential equations. Finding analytical solution of these PDEs is not an easy task. Therefore, researchers are trying to develop an accurate and efficient numerical method for solving these PDEs. Here we construct general heat type time dependent PDEs of the following form:

$$\partial_t Y(\xi, \eta, t) + \alpha Y(\xi, \eta, t) \nabla Y(\xi, \eta, t) + \gamma Y(\xi, \eta, t) = \beta \Delta Y(\xi, \eta, t) + g(\xi, \eta, t), \quad \xi, \eta \in \Gamma, \quad t \in [0, T], \quad (1)$$

with initial and boundary conditions

$$Y(\xi, \eta, 0) = \phi_1(\xi, \eta), \quad \xi, \eta \in \Gamma \quad \text{and} \quad Y(\xi, \eta, t) = \phi_2(\xi, \eta, t), \quad \xi, \eta \in \partial\Gamma,$$

I. Ali, S. Haq, S. U. Arifeen  
Faculty of Engineering Sciences, GIK Institute, Topi, KPK 23640, Pakistan  
E-mail: ihteramali60@gmail.com

S. Haq  
E-mail: siraj@giki.edu.pk

S. U. Arifeen  
E-mail: shamsularifeen054@gmail.com

K. S. Nisar (✉)  
Department of Mathematics, College of Arts and Sciences, Prince Sattam Bin Abdulaziz University, Wadi Aldawaser 11991,  
Kingdom of Saudi Arabia  
E-mail: n.sooppy@psau.edu.sa



where  $g$  is source term depends on space and time,  $\Gamma$ ,  $\partial\Gamma$  are spatial domain and boundary of the domain, respectively,  $\alpha$  and  $\gamma$  are positive constants,  $\nabla = \partial_\xi + \partial_\eta$  and  $\Delta = \partial_{\xi\xi} + \partial_{\eta\eta}$  represents Laplacian operator. When  $\alpha = 0$  and  $\beta = 1$  Eq. (1) becomes heat equation which is used to model thermal conductivity over a solid body. This equation is also used to model other real-world phenomenons like Black–Scholes model; it is applied for price estimation, and in chemical process it is designed in the form of diffusion equation [18]. In the form of advection-diffusion equation it describes heat transfer in a solid body and dissipation of salt in ground water. In image processing it handles image at different scales [28]. The solution of this equation plays a significant role to examine behavior of such physical processes. Different numerical methods have been used for solution of heat equation by many researchers. A comparative study between the classical finite difference and finite element method was investigated by Benyam Mebrate [25]. B-spline and finite element method have been used by Darbel et al. [12]. Hooshmandasl et al. [18] used wavelet methods to solve one dimensional heat equations, while Dehghan [13] used second order finite difference scheme for two-dimensional heat equations. In [34], the authors applied haar wavelet method for two-dimensional fractional reaction–diffusion equations. In [2], the authors applied fifth-kind Chebyshev polynomials for the spectral solution of convection–diffusion equation.

When  $\alpha = 1$  and  $\beta = 1/Re$  ('Re' is Reynolds number), then Eq. (1) becomes well-known Burger equation which was experienced in the field of turbulence, shock wave theory, viscous fluid flow, gas dynamics, cosmology, traffic flow, quantum field, and heat conduction [27]. The low-kinematic viscosity shocks and the relation between cellular and large-scale structure of the universe have been described by one- and three-dimensional Burger equations. When traffic is treated as one-dimensional incompressible fluid, then the density wave in traffic flow which changes from non-uniform to uniform distribution is described by Burger equation [10]. The Burger equation was first introduced by Bateman in viscous fluid flow, which was then extended by Burger in (1948) to examine turbulence phenomena and that is why it is known as Burger equation [27]. Due to wide applications of Burger equation many numerical methods have been implemented to study behavior of the model. One-dimensional Burger equation has been solved using various techniques in [23, 35]. Mittal and Jiwari [27] implemented differential quadrature method for solution of Burger-type equation. Similarly El-sayed and Kaya [20] solved two-dimensional Burger equation using decomposition method. Liao [24] used fourth-order finite difference technique for the study of two-dimensional Burger equation. Oruc [33] applied meshless pseudo-spectral method to modified Burgers equations. The same author in [30] studied three-dimensional fractional convection–diffusion equation using meshless method based on multiple-scale polynomials. In this work, we study afore mentioned models by using combination of Lucas and Fibonacci polynomials. These polynomials can be directly obtained as a special case from the work done in [4, 6]. These polynomials are non-orthogonal and do not require domain and problem transformation which is important point of the proposed scheme. Also the higher order derivative of unknown functions can be easily approximated via Lucas and Fibonacci Polynomials. Second, it is straightforward and produces better accuracy for less number of nodal points. Many researchers applied these polynomials for the solution of fractional differential equations (FDEs) such as Elhameed and Youssri [3], who applied Lucas polynomial in a Caputo sense to FDEs. Moreover they computed the solution of fractional pantograph differential equations (FPDEs) using generalized Lucas polynomials [39]. Other polynomials applied for approximation of FDEs are studied in [5, 7]. Cetin [11] used Lucas polynomial approach to study a system of higher order differential equation, whereas Bayku [9] applied hybrid Taylor–Lucas collocation technique for delay differential equations. Mostefa [29] obtained solution of integro differential equation using Lucas series. Farshid et al. [26] applied Fibonacci polynomials for solution of Volterra–Fredholm integral equations. Omer Oruc [31, 32] applied a combined Lucas and Fibonacci polynomials approach for numerical solution of evolutionary equation for the first time. Recently, Ali et al. [8, 15, 17] applied Lucas polynomials coupled with finite differences and obtained accurate solution of various classes of one- and two-dimensional PDEs. In this paper, we implement the proposed method to one- and two-dimensional Burger and heat equations. The simulation is carried out with the help of MATLAB 2013 using Intel core-i7 machine with 4GB RAM. The error bound of the scheme is also investigated in this work. The paper is organized as follows: In Sect. 2 we define basic definitions and important concept which will be used in this work. In Sect. 3 methodology of the proposed scheme is formulated. Error analysis of the method is described in Sect. 4. Numerical experiments are presented in Sect. 5 followed by conclusion of the paper.

## 2 Basic concepts and definition

In this section, we describe some necessary definitions and concepts required for our subsequent development.



**Definition 2.1** [22] Lucas polynomials are the generalization of well-known Lucas numbers which are generated by linear recursive relation as follows:

$$L_k(\xi) = \xi L_{k-1}(\xi) + L_{k-2}(\xi), \quad \text{for } k \geq 2, \tag{2}$$

with  $L_0(\xi) = 2$  and  $L_1(\xi) = \xi$ , by putting  $\xi = 1$ , Eq (2) gives Lucas numbers.

**Definition 2.2** [22] Fibonacci polynomials can be generated by the following linear recursive relation:

$$\mathcal{F}_k(\xi) = \xi \mathcal{F}_{k-1}(\xi) + \mathcal{F}_{k-2}(\xi) \quad \text{for } k \geq 2, \tag{3}$$

where  $\mathcal{F}_0(\xi) = 0$ , and  $\mathcal{F}_1(\xi) = 1$ , by putting  $\xi = 1$ , Eq. (3) generate Fibonacci numbers.

**Function Approximation.** Let  $Y(\xi)$  be square integrable on  $(0, 1)$  and suppose that it can be expressed in terms of the Lucas polynomials given as follows:

$$Y(\xi) = \sum_{k=0}^N c_k L_k(\xi) = \mathbf{C}\mathbf{L}(\xi), \tag{4}$$

where  $\mathbf{C} = [c_0, c_1, \dots, c_N]^T$  is  $(N + 1) \times 1$  vector of unknown coefficients and  $\mathbf{L}(\xi) = [L_0(\xi), L_1(\xi), \dots, L_N(\xi)]$  is  $(N + 1) \times (N + 1)$  matrix of Lucas polynomials. Similarly the  $m$ th order derivative of the function  $Y(\xi)$  can be approximated in terms of finite Lucas series is given as follows:

$$Y^{(m)}(\xi) = \sum_{k=0}^N c_k L_k^{(m)}(\xi) = \mathbf{C}\mathbf{L}^{(m)}(\xi), \tag{5}$$

in which  $\mathbf{L}^{(m)}(\xi) = [L_0^{(m)}(\xi), L_1^{(m)}(\xi), \dots, L_N^{(m)}(\xi)]$  is square matrix.

**Corollary 2.3** [22]: Let  $L_k^{(m)}(\xi)$  be the  $m$ th order derivative of Lucas polynomials; Then it can be expanded in terms of Fibonacci polynomials by the following relation:

$$L_k^{(m)}(\xi) = k(\mathcal{F}_k(\xi)\mathbf{D}^{m-1}), \quad m \geq 1, \tag{6}$$

where  $\mathbf{D}$  is  $(N + 1) \times (N + 1)$  differentiation matrix such that:

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

in which  $\mathbf{d}$  is square matrix of order  $N \times N$  defined as:

$$d_{m,k} = \begin{cases} m \sin \frac{(k-m)\pi}{2}, & \text{if } k > m, \\ 0, & \text{otherwise.} \end{cases}$$

For example, if we choose  $N = 3$ , then we have

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and for  $m = 2$  the second-order derivative of  $\mathbf{L}(\xi)$  can be obtained using Eq. (6)

$$\mathbf{L}''(\xi) = k(\mathcal{F}(\xi)\mathbf{D}) = k[0, 1, \xi, \xi^2 + 1] \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{7}$$

where  $k = [0, 1, 2, 3]$  after element wise multiplication we get

$$L''(\xi) = [0, 0, 2, 6x] \quad (8)$$

Now assume a two-dimensional continuous function  $Y(\xi, \eta)$  can be written in terms of Lucas polynomials as follows:

$$Y(\xi, \eta) \approx \sum_{k=0}^N \sum_{m=0}^N c_{km} L_k(\xi) L_m(\eta) = \sum_{k=0}^N \sum_{m=0}^N c_{km} L_{km}(\xi, \eta) = \mathbf{C}\mathbf{L}(\xi, \eta), \quad (9)$$

where  $\mathbf{C} = [c_{00}, \dots, c_{0N}, c_{N0}, \dots, c_{NN}]^T$  is Lucas unknown coefficients vector of order  $(N + 1)^2 \times 1$ , and  $\mathbf{L}(\xi, \eta) = [L_{00}(\xi, \eta), \dots, L_{0N}(\xi, \eta), L_{N0}(\xi, \eta), \dots, L_{NN}(\xi, \eta)]$  is square matrix of order  $(N + 1)^2$ .

### 3 Solution methodology

Consider the following evolutionary equation:

$$\partial_t Y(\mathbf{x}, t) + \mathfrak{L}Y(\mathbf{x}, t) = g(\mathbf{x}, t) \quad \mathbf{x} \in \Gamma, \quad t \in [0, T], \quad (10)$$

where  $\mathfrak{L}$  is differential operator and  $g(\mathbf{x}, t)$  is a given smooth function. The initial and boundary conditions are given as follows:

$$Y(\mathbf{x}, 0) = Y_0(\mathbf{x}), \quad \mathbf{x} \in \Gamma \quad \text{and} \quad \mathfrak{B}Y(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Gamma, \quad (11)$$

where  $\mathfrak{B}$  is boundary operator. To approximate Eq. (10), first we define time discretization given by the following:

$$t^n = (n - 1)\delta t, \quad n = 1, 2, \dots, M$$

where  $\delta t = T/M$  is time step size for variable  $t$  and  $T$  is final time. Now applying finite difference scheme to temporal part and  $\theta$ -weighted scheme to spatial part of Eq. (10), one can write

$$\frac{1}{\delta t} [Y^{n+1}(\mathbf{x}) - Y^n(\mathbf{x})] + \theta \mathfrak{L}Y^{n+1}(\mathbf{x}) + (1 - \theta) \mathfrak{L}Y^n(\mathbf{x}) = g^{n+1}(\mathbf{x}), \quad (12)$$

where  $Y^{n+1}(\mathbf{x}) = Y(\mathbf{x}, t^{n+1})$  and so forth. By putting  $\theta = 0.5$ , the scheme in Eq. (12) represents Crank–Nicolson which is  $O(\delta t^2)$  accurate in time.

For discretization of spatial domain  $\Gamma = [a, b]$  we use regular grid point defined as follows:

$$\mathbf{x}_i = a + \frac{b - a}{M}(i - 1), \quad i = 1, 2, \dots, N + 1, \quad \mathbf{x}_i \in \Gamma$$

For  $(n + 1)$  time level and at the collocation point  $\mathbf{x} = (\xi_i, \eta_j)$  Eq. (9) can be written as follows:

$$Y^{n+1}(\xi_i, \eta_j) = \sum_{k=0}^N \sum_{m=0}^N c_{km}^{n+1} L_k(\xi_i) L_m(\eta_j), \quad i, j = 0, 1, \dots, N. \quad (13)$$

Using Eq. (13) in Eq. (12), we have

$$\begin{aligned} \frac{1}{\delta t} \left[ \sum_{k=0}^N \sum_{m=0}^N c_{km}^{n+1} L_k(\xi_i) L_m(\eta_j) - \sum_{k=0}^N \sum_{m=0}^N c_{km}^n L_k(\xi_i) L_m(\eta_j) \right] + \theta \left[ \sum_{k=0}^N \sum_{m=0}^N c_{km}^{n+1} \mathfrak{L}(L_k(\xi_i) L_m(\eta_j)) \right] \\ + (1 - \theta) \left[ \sum_{k=0}^N \sum_{m=0}^N c_{km}^n \mathfrak{L}(L_k(\xi_i) L_m(\eta_j)) \right] = g^{n+1}(\xi_i, \eta_j), \quad (\xi_i, \eta_j) \in \Gamma. \end{aligned}$$



The above equation can also be written as

$$\begin{aligned} & \sum_{k=0}^N \sum_{m=0}^N [L_k(\xi_i)L_m(\eta_j) + \theta \delta t \mathfrak{L}(L_k(\xi_i)L_m(\eta_j))]c_{km}^{n+1} \\ &= \sum_{k=0}^N \sum_{m=0}^N [L_k(\xi_i)L_m(\eta_j) + (1 - \theta) \delta t \mathfrak{L}(L_k(\xi_i)L_m(\eta_j))]c_{km}^n + g^{n+1}(\xi_i, \eta_j), \quad (\xi_i, \eta_j) \in \Gamma. \end{aligned} \tag{14}$$

The boundary conditions (11) transform to

$$\sum_{k=0}^N \sum_{m=0}^N c_{km}^{n+1} \mathfrak{B}(L_k(\xi_i)L_m(\eta_j)) = f^{n+1}(\xi_i, \eta_j), \quad (\xi_i, \eta_j) \in \partial\Gamma. \tag{15}$$

Matrix form of Eqs. (14) and (15) is given by

$$\mathbf{HC}^{n+1} = \mathbf{GC}^n + \mathbf{B}^{n+1}. \tag{16}$$

For  $k, m = 0, \dots, N$  elements for the above matrices can be obtained in the following:

$$\mathbf{H} = \begin{cases} L_k(\xi_i)L_m(\eta_j) + \delta t \theta \mathfrak{L}(L_k(\xi_i)L_m(\eta_j)), & (\xi_i, \eta_j) \in \Gamma, \\ \mathfrak{B}(L_k(\xi_i)L_m(\eta_j)), & (\xi_i, \eta_j) \in \partial\Gamma, \end{cases} \tag{17}$$

$$\mathbf{G} = \begin{cases} L_k(\xi_i)L_m(\eta_j) + \delta t (1 - \theta) \mathfrak{L}(L_k(\xi_i)L_m(\eta_j)), & (\xi_i, \eta_j) \in \Gamma, \\ 0, & (\xi_i, \eta_j) \in \partial\Gamma, \end{cases} \tag{18}$$

$$\mathbf{B}^{n+1} = \begin{cases} g^{n+1}(\xi_i, \eta_j), & (\xi_i, \eta_j) \in \Gamma, \\ f^{n+1}(\xi_i, \eta_j), & (\xi_i, \eta_j) \in \partial\Gamma, \end{cases} \tag{19}$$

where  $\mathbf{H}$ ,  $\mathbf{G}$  and  $\mathbf{B}$  are  $(N + 1)^2$  order square matrices. The unknown coefficient vector  $\mathbf{C}$  can be obtained by solving Eq. (16). Once the values of unknown coefficient are computed the solution of the problem under consideration can be obtained from Eq. (9).

### 4 Truncation error estimate

To study the error estimate of the proposed scheme, we follow the approach of Abd-Elhameed and Youssri [3].

**Theorem 4.1** *Let  $Y(\xi, \eta)$  and  $Y^*(\xi, \eta)$  be exact and approximate solutions of Eq. (1). Moreover, we expand  $Y(\xi, \eta)$  in terms of Lucas sequence. Then truncation error is given as*

$$|e| < \frac{4 \exp(2\kappa) \cosh^2(2P)\kappa^{2(N+1)}}{((N + 1)!)^2}. \tag{20}$$

*Proof* Consider the absolute error between exact and approximate solution

$$|e| = |Y(\xi, \eta) - Y^*(\xi, \eta)| \tag{21}$$

where  $Y(\xi, \eta) = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} \lambda_k \lambda_m L_k(\xi)L_m(\eta)$  and  $Y^*(x, y) = \sum_{k=0}^M \sum_{m=0}^M \lambda_k \lambda_m L_k(\xi)L_m(\eta)$ . Then the truncated term is given as

$$|e| = \sum_{k=N+1}^{\infty} \sum_{m=N+1}^{\infty} \lambda_k \lambda_m L_k(\xi)L_m(\eta). \tag{22}$$

It is shown in [3]

$$|L_m(\xi)| \leq 2\vartheta^m, \quad |\lambda_m| \leq \frac{P^m \cosh(2P)}{m!},$$

where  $\vartheta$  is well-known golden ratio. Therefore, Eq. (22) implies that

$$|e| \leq 4 \cosh^2(2P) \sum_{k=N+1}^{\infty} \sum_{m=N+1}^{\infty} \frac{\kappa^{k+m}}{k!m!},$$

where  $\kappa = P\vartheta$ . The above inequality can also be written as

$$|e| \leq 4 \exp(2\kappa) \cosh^2(2P) \left[ 1 - \frac{\Gamma(N+1, \kappa)}{\Gamma(N+1)} \right]^2. \quad (23)$$

Here,  $\Gamma(N+1, \kappa)$  is the incomplete gamma function and  $\Gamma(N+1)$  is complete gamma function [36]. In integral form Eq. (23) can be written as

$$|e| \leq \frac{4 \exp(2\kappa) \cosh^2(2P)}{(N!)^2} \left[ \int_0^\kappa t^N \exp(-t) dt \right]^2.$$

As,  $\exp(-t) < 1$ , therefore, for all  $t > 0$ , we have

$$|e| < \frac{4 \exp(2\kappa) \cosh^2(2P) \kappa^{2(N+1)}}{((N+1)!)^2}.$$

This completes the proof.  $\square$

## 5 Numerical examples

In this section, one- and two-dimensional Burgers' and diffusion equations have been solved using the proposed method. Performance of the technique is examined by computing  $L_2$ ,  $L_\infty$ , and root mean square (RMS) error norms for different collocation points  $N$  and time  $T$ . The obtained results are then compared with available results in the literature.

*Example 5.1* When  $\alpha = 0$ ,  $\beta = 1$ ,  $\gamma = 0$  and  $g(\xi, t) = 0$  then Eq. (1) becomes heat equation given by

$$\partial_t Y(\xi, t) - \partial_{\xi\xi} Y(\xi, t) = 0, \quad (24)$$

with initial and boundary conditions

$$Y(\xi, 0) = \sin(\xi), \quad 0 \leq \xi \leq 1 \quad \text{and} \quad Y(0, t) = 0, \quad Y(1, t) = \sin(1)e^{-t} \quad \text{for} \quad t \geq 0,$$

and exact solution [18].

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

Comparison of the above equations with Eqs. (10) and (11) gives

$$\mathfrak{L} = -\partial_{\xi\xi}, \quad g(\xi, t) = 0, \quad Y_0(\xi) = \sin(\xi), \quad \text{and} \quad f(\xi, t) = 0.$$

Applying the technique discussed in Sect. 2, Eq. (14) takes the following form:

$$\sum_{k=0}^N c_k^{n+1} L_k(\xi) - \delta t \theta \sum_{k=0}^N c_k^{n+1} L_k''(\xi) = \sum_{k=0}^N c_k^n L_k(\xi) + \delta t (1 - \theta) \sum_{k=0}^N c_k^n L_k''(\xi), \quad (25)$$

where  $L_k''(\xi)$  represents second-order derivative of Lucas polynomials which can be obtained using Eq. (6). The matrices in Eq. (17)–(19) takes the following form:

$$\mathbf{H} = \begin{cases} L_k(\xi_i) - \delta t \theta k \mathcal{F}_k(\xi_i) \mathbf{D}, & i = 1, \dots, N-1, \\ L_k(\xi_i), & i = 0, N, \end{cases} \quad (26)$$

$$\mathbf{G} = \begin{cases} L_k(\xi_i) + \delta t (1 - \theta) k \mathcal{F}_k(\xi_i) \mathbf{D}, & i = 1, \dots, N-1, \\ 0, & i = 0, N, \end{cases} \quad (27)$$



**Table 1** Error norms for  $N = 16, \delta t = 0.001$  of Example 5.1

| Present method |            |            |            |          | [18]       |            |
|----------------|------------|------------|------------|----------|------------|------------|
| T              | $L_2$      | $L_\infty$ | RMS        | CPU time | $L_2$      | $L_\infty$ |
| 0.1            | $2.14E-09$ | $3.01E-09$ | $2.08E-09$ | 0.09500  | $4.46E-06$ | $1.07E-05$ |
| 0.3            | $2.75E-09$ | $3.82E-09$ | $2.66E-09$ | 10.11596 | $2.15E-07$ | $3.45E-06$ |
| 0.5            | $2.39E-09$ | $3.32E-09$ | $2.31E-09$ | 0.15753  | $3.18E-06$ | $5.13E-06$ |
| 0.7            | $1.97E-09$ | $2.74E-09$ | $1.91E-09$ | 0.16876  | $4.71E-06$ | $7.45E-06$ |
| 0.9            | $1.62E-09$ | $2.25E-09$ | $1.57E-09$ | 0.19902  | $6.04E-06$ | $9.47E-06$ |
| 1              | $1.47E-09$ | $2.04E-09$ | $1.42E-09$ | 0.21293  | $6.55E-06$ | $1.02E-05$ |

**Table 2** Spatial convergence when  $T = 1, \delta t = 0.001$  for Example 5.1

| $d\xi$ | CPU time | $L_\infty$ | $L_2$      | RMS        |
|--------|----------|------------|------------|------------|
| 1/3    | 0.0316   | $5.20E-04$ | $3.67E-04$ | $3.00E-04$ |
| 1/4    | 0.0169   | $2.19E-04$ | $2.31E-04$ | $1.42E-04$ |
| 1/5    | 0.0164   | $1.50E-05$ | $1.42E-05$ | $8.56E-06$ |
| 1/6    | 0.0201   | $9.99E-07$ | $1.62E-06$ | $7.42E-07$ |
| 1/7    | 0.0223   | $5.96E-08$ | $6.90E-08$ | $3.37E-08$ |
| 1/8    | 0.0271   | $1.25E-09$ | $1.60E-09$ | $7.04E-10$ |
| 1/9    | 0.0339   | $1.01E-09$ | $1.71E-09$ | $1.38E-09$ |

**Table 3** Error norm for  $N = 20, \delta t = 0.001$  of Example 5.1

| T          | 1          | 2          | 3          | 4          | 5          |
|------------|------------|------------|------------|------------|------------|
| $L_\infty$ | $2.04E-09$ | $7.52E-10$ | $2.76E-10$ | $1.02E-10$ | $3.74E-11$ |
| $L_2$      | $5.92E-09$ | $2.18E-09$ | $8.01E-10$ | $2.95E-10$ | $1.08E-10$ |
| RMS        | $1.43E-09$ | $5.26E-10$ | $1.94E-10$ | $7.12E-11$ | $2.62E-11$ |

$$\mathbf{B}^{n+1} = \begin{cases} 0, & i = 1, 2, \dots, N - 1, \\ \sin(\xi_i)e^{-t}, & i = 0, N. \end{cases} \tag{28}$$

The problem has been solved for different values of nodal points  $N$ , and time  $T$ . Computed solutions are compared with the results provided by Chebyshev wavelet method in the form of various error norms which are shown in Table 1. It is clear from the table that proposed scheme gives excellent results than cited work. Numerical convergence and Cpu time have been reported in Table 2 for various values of  $N$ . From the table it can be observed that the solution converges as the nodal point  $N$  increases. i.e. ( $d\xi$  decreases). Error norms for large time level are recorded in Table 3 and better results noticed than that for small time level. Solution profile and absolute errors are presented in Fig. 1 which show that exact and approximate solution are well matched with each other showing efficiency of the proposed scheme.

*Example 5.2* Consider Eq. (1), with  $\alpha = 0, \beta = 1, \gamma = -2$  and  $g(\xi, t) = 0$ , defined as homogeneous heat equation given as follows

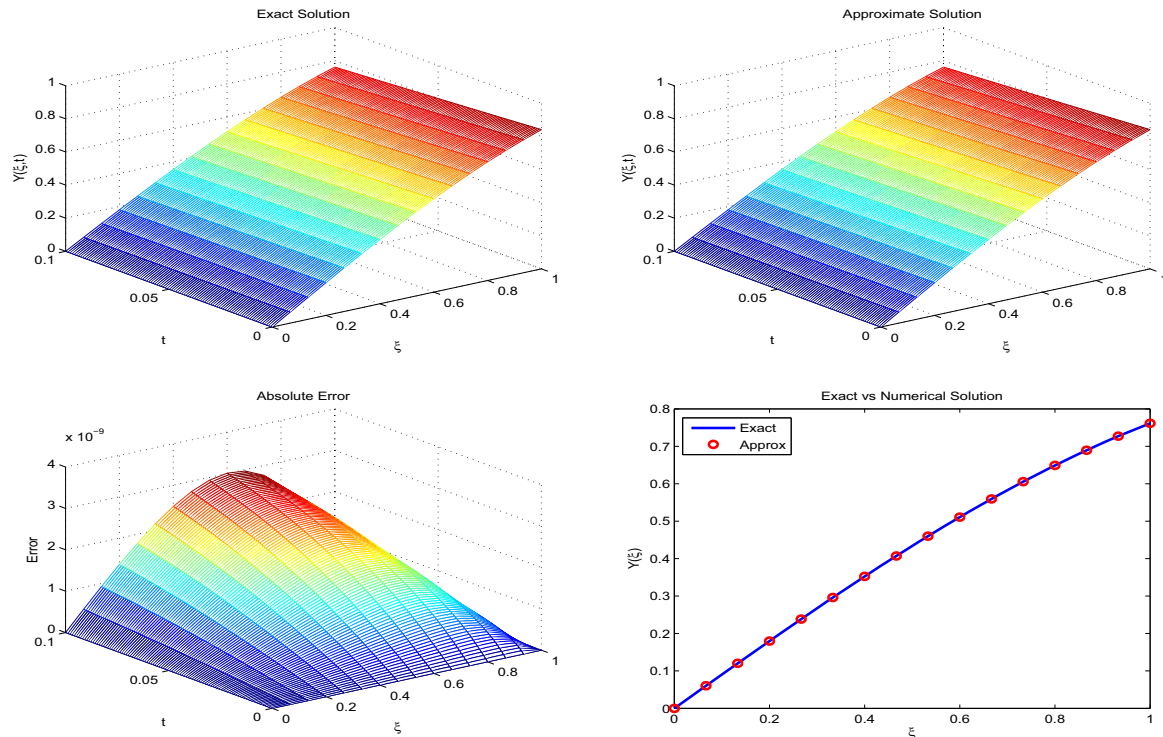
$$\partial_t Y(\xi, t) = \partial_{\xi\xi} Y(\xi, t) + 2Y(\xi, t), \quad 0 \leq \xi \leq 1, \quad 0 \leq t \leq 1.$$

with exact solution

$$Y(\xi, t) = \sinh(\xi)e^{-t}$$

Initial and boundary conditions are extracted from exact solution. The problem has been solved by adopting the procedure discussed in Sect. 2. The results are computed for different values of  $N$  and  $T$ . RMS,  $L_2$ , and  $L_\infty$  error norms have been calculated and comparison is made with available results in literature [18] for different values of  $T, N = 16, \delta t = 0.001$  that are shown in Table 4. From the table, it is straightforward that the results achieved using the proposed method are better than those available in literature which show proficiency of the method used. For convergence in space the results obtained are shown in Table 5 for different values of  $d\xi$  showing that the solution converges as the value  $d\xi$  decreases. Graphs of solutions along with absolute errors for  $T = 1$  and  $N=15$  are plotted in Fig. 2. The figure shows that exact and approximate solutions are closely matched.





**Fig. 1** Solution profile when  $T = 0.1$ ,  $\delta t = 0.001$ ,  $N = 15$  of Example 5.1

**Table 4** Error norms when  $N = 16$ ,  $\delta t = 0.001$  of Example 5.2

| Present method |            |            |            |          | [18]       |            |
|----------------|------------|------------|------------|----------|------------|------------|
| T              | $L_\infty$ | $L_2$      | RMS        | CPU time | $L_\infty$ | $L_2$      |
| 0.1            | $3.20E-09$ | $2.26E-09$ | $2.20E-09$ | 0.09610  | $8.44E-03$ | $6.01E-03$ |
| 0.3            | $3.69E-09$ | $2.66E-09$ | $2.58E-09$ | 0.15702  | $8.10E-03$ | $2.59E-03$ |
| 0.5            | $3.13E-09$ | $2.25E-09$ | $2.18E-09$ | 0.13122  | $7.43E-03$ | $4.78E-03$ |
| 0.7            | $2.57E-09$ | $1.85E-09$ | $1.79E-09$ | 0.17758  | $9.18E-03$ | $5.88E-03$ |
| 0.9            | $2.11E-09$ | $1.51E-09$ | $1.47E-09$ | 0.20781  | $1.07E-03$ | $6.89E-03$ |
| 1              | $1.91E-09$ | $1.37E-09$ | $1.33E-09$ | 0.22148  | $1.15E-03$ | $7.39E-03$ |

**Table 5** Space convergence for  $T = 1$  and  $\delta t = 0.001$  of Example 5.2

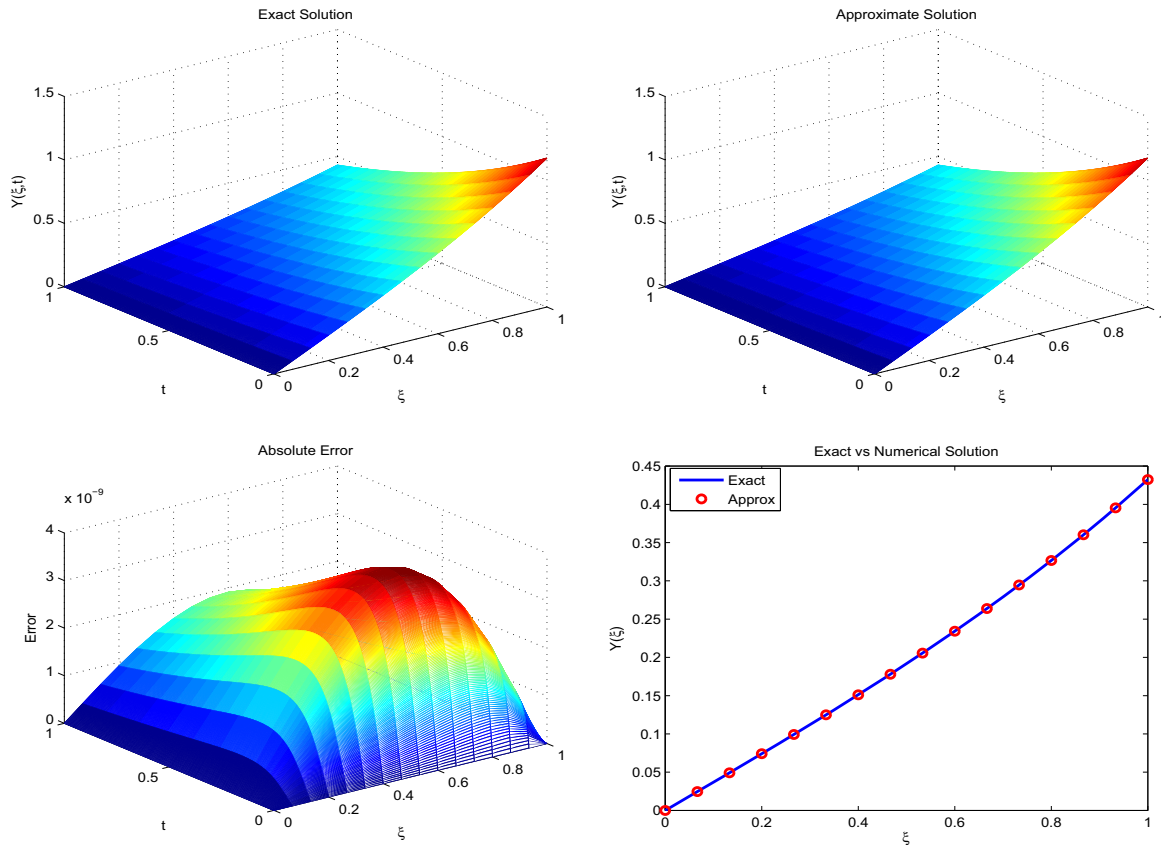
| $d\xi$ | CPU time | $L_\infty$ | $L_2$      | RMS        |
|--------|----------|------------|------------|------------|
| 1/3    | 0.0395   | $4.47E-04$ | $3.16E-04$ | $2.58E-04$ |
| 1/4    | 0.0420   | $2.03E-04$ | $1.48E-04$ | $1.28E-04$ |
| 1/5    | 0.0594   | $1.81E-05$ | $1.17E-05$ | $1.05E-05$ |
| 1/6    | 0.0922   | $9.70E-07$ | $7.37E-07$ | $6.73E-07$ |
| 1/7    | 0.1274   | $7.31E-08$ | $4.47E-08$ | $4.14E-08$ |
| 1/8    | 0.1643   | $1.55E-09$ | $7.11E-09$ | $6.65E-09$ |
| 1/9    | 0.2232   | $1.49E-09$ | $1.35E-09$ | $1.27E-09$ |

**Example 5.3** Consider  $\alpha = 1$ ,  $\gamma = 0$ ,  $g(\mathbf{x}, t) = 0$  in Eq. (1); we get the following two-dimensional nonlinear Burger equation:

$$\partial_t Y(\xi, \eta, t) + Y(\xi, \eta, t) \{ \partial_\xi Y(\xi, \eta, t) + \partial_\eta Y(\xi, \eta, t) \} = \beta \{ \partial_{\xi\xi} Y(\xi, \eta, t) + \partial_{\eta\eta} Y(\xi, \eta, t) \}. \quad (29)$$

There are two cases.





**Fig. 2** Solution profile when  $T = 1, \delta t = 0.001, N = 15$  of Example 5.2

**Case 1:** In this case the exact solution is given as [16]

$$Y(\xi, \eta, t) = \frac{1}{1 + e^{\frac{\xi + \eta - t}{2\beta}}}, \quad 0 \leq \xi, \eta \leq 1. \tag{30}$$

Initial and boundary conditions are extracted from the exact solution. The nonlinear part in Eq. (29) can be linearized by the following formula [8]:

$$(Y \partial_\xi Y)^{n+1} = Y^{n+1} \partial_\xi Y^n. \tag{31}$$

Applying the technique discussed in Sect. 2 Eq. (14) takes the following form:

$$\begin{aligned} & \sum_{k=0}^N \sum_{m=0}^N \left[ L_k(\xi_i) L_m(\eta_j) - \delta t \theta \beta \{ (k \mathcal{F}_k(\xi_i) \mathbf{D}) L_m(\eta_j) + L_k(\xi_i) (m \mathcal{F}_m(\eta_j) \mathbf{D}) \} + \delta t \theta L_k(\xi_i) L_m(\eta_j) \right. \\ & \quad \left. \{ \partial_\xi Y^n(\xi_i, \eta_j) + \partial_\eta Y^n(\xi_i, \eta_j) \} \right] c_{km}^{n+1} \\ & = \sum_{k=0}^N \sum_{m=0}^N \left[ L_k(\xi_i) L_m(\eta_j) + \delta t (1 - \theta) \beta \{ (k \mathcal{F}_k(\xi_i) \mathbf{D}) L_m(\eta_j) + L_k(\xi_i) (m \mathcal{F}_m(\eta_j) \mathbf{D}) \} - \right. \\ & \quad \left. - \delta t (1 - \theta) L_k(\xi_i) L_m(\eta_j) \{ \partial_\xi Y^n(\xi_i, \eta_j) + \partial_\eta Y^n(\xi_i, \eta_j) \} \right] c_{km}^n. \end{aligned}$$

The matrices **H**, **G** and **B** in Eq. (16) for  $k, m = 0, 1, \dots, N$ , are given as follows:

$$\mathbf{H} = \begin{cases} L_k(\xi_i) L_m(\eta_j) - \delta t \theta \beta \{ (k \mathcal{F}_k(\xi_i) \mathbf{D}) L_m(\eta_j) + L_k(\xi_i) (m \mathcal{F}_m(\eta_j) \mathbf{D}) \} \\ + \delta t \theta L_k(\xi_i) L_m(\eta_j) \{ \partial_\xi Y^n(\xi_i, \eta_j) + \partial_\eta Y^n(\xi_i, \eta_j) \}, & (\xi_i, \eta_j) \in \Gamma, \\ L_k(\xi_i) L_m(\eta_j), & (\xi_i, \eta_j) \in \partial\Gamma, \end{cases} \tag{32}$$

**Table 6** Error norms of Example 5.3 case 1 for  $\delta t = 0.0025$

| Present method |         |     |            |            |            |          | [19]       |            |
|----------------|---------|-----|------------|------------|------------|----------|------------|------------|
| T              | $\beta$ | N   | $L_\infty$ | $L_2$      | RMS        | CPU time | $L_\infty$ | $L_2$      |
| 0.25           | 1       | 100 | $6.50E-05$ | $1.03E-04$ | $1.00E-04$ | 0.36900  | $7.88E-04$ | $4.03E-03$ |
|                |         | 400 | $2.12E-05$ | $1.85E-05$ | $1.07E-05$ | 3.21356  | $8.55E-05$ | $8.47E-04$ |
|                | 0.1     | 100 | $1.79E-04$ | $1.69E-04$ | $1.29E-04$ | 0.34902  | $7.18E-04$ | $3.68E-03$ |
|                |         | 400 | $5.88E-05$ | $2.03E-05$ | $1.12E-05$ | 3.13794  | $3.54E-04$ | $1.96E-03$ |
| Present method |         |     |            |            |            |          | [16]       |            |
| T              | $\beta$ | N   | $L_\infty$ | $L_2$      | RMS        | CPU time | $L_\infty$ | $L_2$      |
| 0.5            | 1       | 16  | $1.32E-05$ | $1.25E-05$ | $1.36E-04$ | 0.08960  | $5.81E-05$ | $1.53E-04$ |
|                |         | 64  | $1.98E-05$ | $3.07E-05$ | $6.75E-05$ | 0.36905  | $6.36E-05$ | $2.77E-04$ |
|                |         | 256 | $1.56E-05$ | $3.33E-05$ | $1.99E-05$ | 2.51117  | $6.30E-05$ | $5.42E-04$ |

**Table 7** Error norms when  $N = 15$ , and  $\delta t = 0.01$  for case 1 Example 5.3

| T          | 0.1        | 0.25       | 0.5        | 1          |
|------------|------------|------------|------------|------------|
| $L_\infty$ | $7.17E-04$ | $4.65E-05$ | $3.66E-05$ | $3.46E-05$ |
| $L_2$      | 0.001247   | $4.96E-05$ | $1.87E-05$ | $1.79E-05$ |
| RMS        | $1.37E-04$ | $2.74E-05$ | $1.68E-05$ | $1.65E-05$ |

$$\mathbf{G} = \begin{cases} L_k(\xi_i)L_m(\eta_j) + \delta t(1 - \theta)\beta\{(k\mathcal{F}_k(\xi_i)\mathbf{D})L_m(\eta_j) + L_k(\xi_i)(m\mathcal{F}_m(\eta_j)\mathbf{D})\} \\ -\delta t(1 - \theta)L_k(\xi_i)L_m(\eta_j)\{\partial_\xi Y^n(\xi_i, \eta_j) + \partial_\eta Y^n(\xi_i, \eta_j)\}, & (\xi_i, \eta_j) \in \Gamma, \\ 0, & (\xi_i, \eta_j) \in \partial\Gamma, \end{cases} \quad (33)$$

$$\mathbf{B}^{n+1} = \begin{cases} 0, & (\xi_i, \eta_j) \in \Gamma, \\ \frac{1}{1+e^{-\frac{\xi_i+\eta_j-t}{2\beta}}}, & (\xi_i, \eta_j) \in \partial\Gamma. \end{cases} \quad (34)$$

The approximate solution of the problem has been computed for different values of viscosity  $\beta$ , time  $T$ , and nodal points  $N$ . Error norms  $L_2$ ,  $L_\infty$  and RMS have been calculated and the obtained results compared with the results of Haar wavelet [16] and differential quadrature [19]. The results are presented in Tables 6 and 7. From these tables it is obvious that proposed method works pretty well than those available in the literature. The solution profile and error plots for  $T = 2$ ,  $\delta t = 0.001$ ,  $\beta = 1$ , and nodal points  $[20 \times 20]$  are shown in Fig. 3 showing efficiency of the proposed technique.

**Case 2:**

In this case exact solution of Eq. (29) is given as [19]

$$Y(\xi, \eta, t) = 0.5 - \tanh\left(\frac{\xi + \eta - t}{2\beta}\right) \quad -0.5 \leq \xi, \eta \leq 0.5, \quad t \geq 0. \quad (35)$$

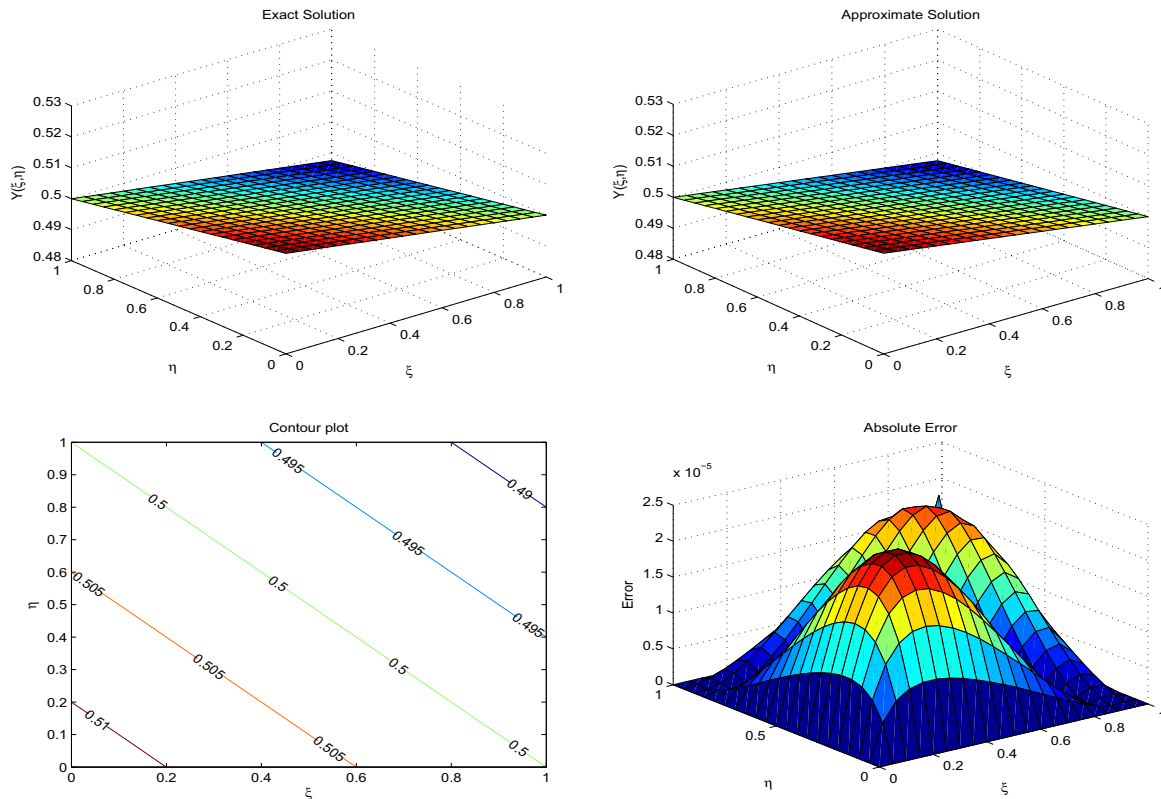
Here also initial and boundary conditions are extracted from the exact solutions. The method discussed in previous example is applied to solve this example for different value of viscosity  $\beta$ . Here also the error norms  $L_2$ ,  $L_\infty$  and RMS have been computed and are compared with the results of meshless collocation method [19] available in literature for different values of  $N$ . The obtained results presented in Table 8 which shows that the present method gives better results than those available in literature. The graph of numerical and exact solution are shown in Fig. 4 which shows efficiency of the current technique.

*Example 5.4* Finally, consider the case when  $\alpha = 0$ ,  $\beta = 1$ ,  $\gamma = 0$  and  $g(\xi, \eta, t) = 0$ ; Then Eq. (1) takes the following form:

$$\partial_t Y(\xi, \eta, t) = \partial_{\xi\xi} Y(\xi, \eta, t) + \partial_{\eta\eta} Y(\xi, \eta, t). \quad (36)$$

The initial and boundary conditions are extracted from exact solution [37]

$$Y(\xi, \eta, t) = \sin(\pi\xi) \sin(\pi\eta) e^{-2\pi^2 t}.$$



**Fig. 3** Solution profile when  $T = 1, \delta t = 0.01, N = 20$  for case 1 Example 5.3

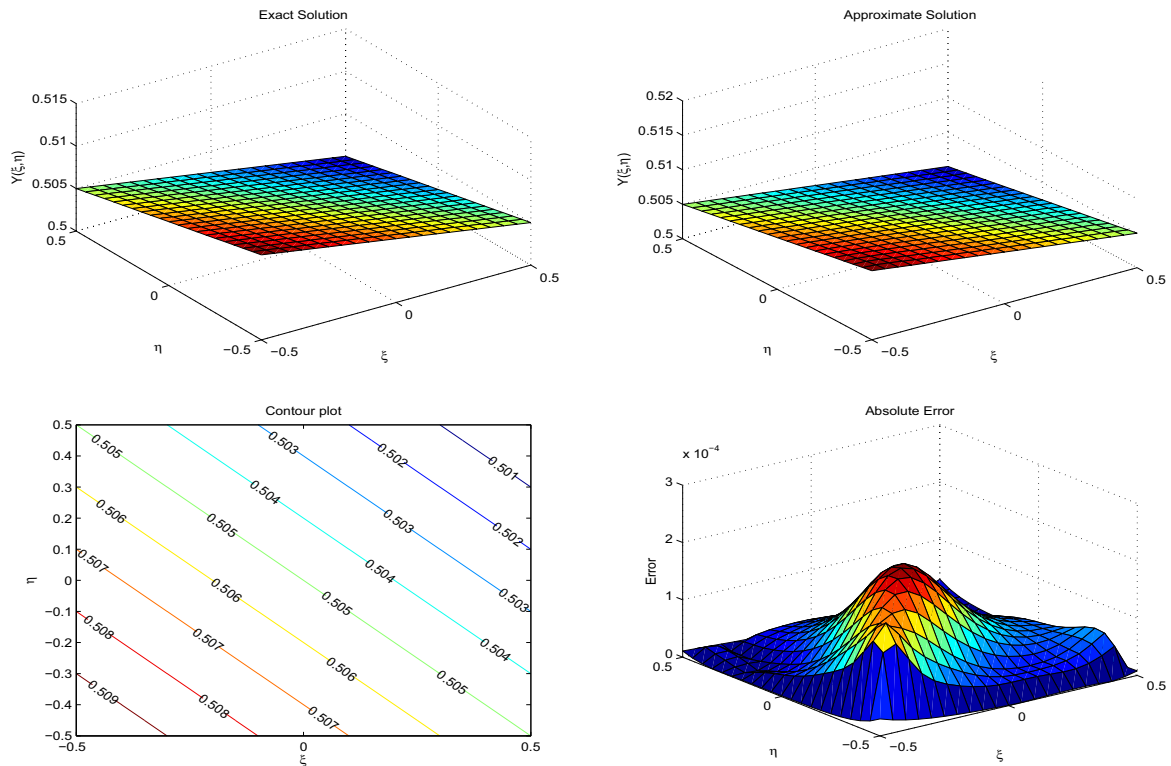
**Table 8** Error norms for  $T = 0.1, \delta t = 0.001$  of case 2 Example 5.3

| Present method |     |          |            |          |           | [19]     |            |
|----------------|-----|----------|------------|----------|-----------|----------|------------|
| $\beta$        | N   | $L_2$    | $L_\infty$ | RMS      | CPU time  | $L_2$    | $L_\infty$ |
| 0.02           | 100 | 2.41E-03 | 1.38E-03   | 4.03E-04 | 0.439244  | 2.49E-01 | 6.86E-01   |
|                | 400 | 2.63E-04 | 7.55E-04   | 3.67E-05 | 4.468202  | 1.42E-02 | 5.69E-02   |
|                | 625 | 8.70E-05 | 2.32E-04   | 1.38E-05 | 14.085021 | 1.44E-04 | 1.65E-03   |
| 0.013          | 100 | 2.58E-03 | 2.37E-03   | 3.99E-04 | 0.484267  | 7.69E-01 | 2.07E+00   |
|                | 400 | 2.64E-04 | 9.52E-04   | 3.68E-05 | 6.261195  | 8.61E-02 | 3.48E-01   |
|                | 625 | 5.96E-05 | 1.83E-04   | 1.14E-05 | 14.188564 | 8.53E-03 | 5.68E-02   |
| 0.01           | 100 | 2.43E-03 | 2.35E-01   | 4.04E-04 | 0.366495  | 1.83E+00 | 4.96E+00   |
|                | 400 | 2.99E-04 | 8.36E-04   | 3.91E-05 | 3.328385  | 2.10E-01 | 1.09E+01   |
|                | 625 | 7.19E-05 | 3.11E-04   | 1.25E-05 | 10.062396 | 4.31E-02 | 4.29E-01   |

The problem has been solved in domain  $[0, 1] \times [0, 1]$  over the time interval  $[0, 1]$  for different values of  $T$  and  $M$ . Error norms are computed and compared with the results of RBFs available in literature given in Tables 9 and 10. From these tables it is clear that the results got using the proposed technique are comparable with those of multiquadric (MQ) RBFs and better than those of wendland (WL) RBFs [37]. The solution profile is plotted in Fig. 5 when  $T = 0.2$  showing efficiency of the method suggested.

### 6 Conclusion

In this paper, we studied a numerical technique based on Lucas and Fibonacci polynomials. First, we discretized temporal part of PDEs by finite difference and spatial part by  $\theta$ - weighted scheme with  $\theta = 1/2$  (Crank Nicolson method). Thenceforth, the unknown functions are expanded by Lucas series while their derivatives are replaced by Fibonacci polynomials. Performance and convergence of the method are investigated by several



**Fig. 4** Solution Profile when  $T = 1$ ,  $\delta t = 0.01$ ,  $\beta = 0.01$ ,  $N = 20$  of case 2 Example 5.3

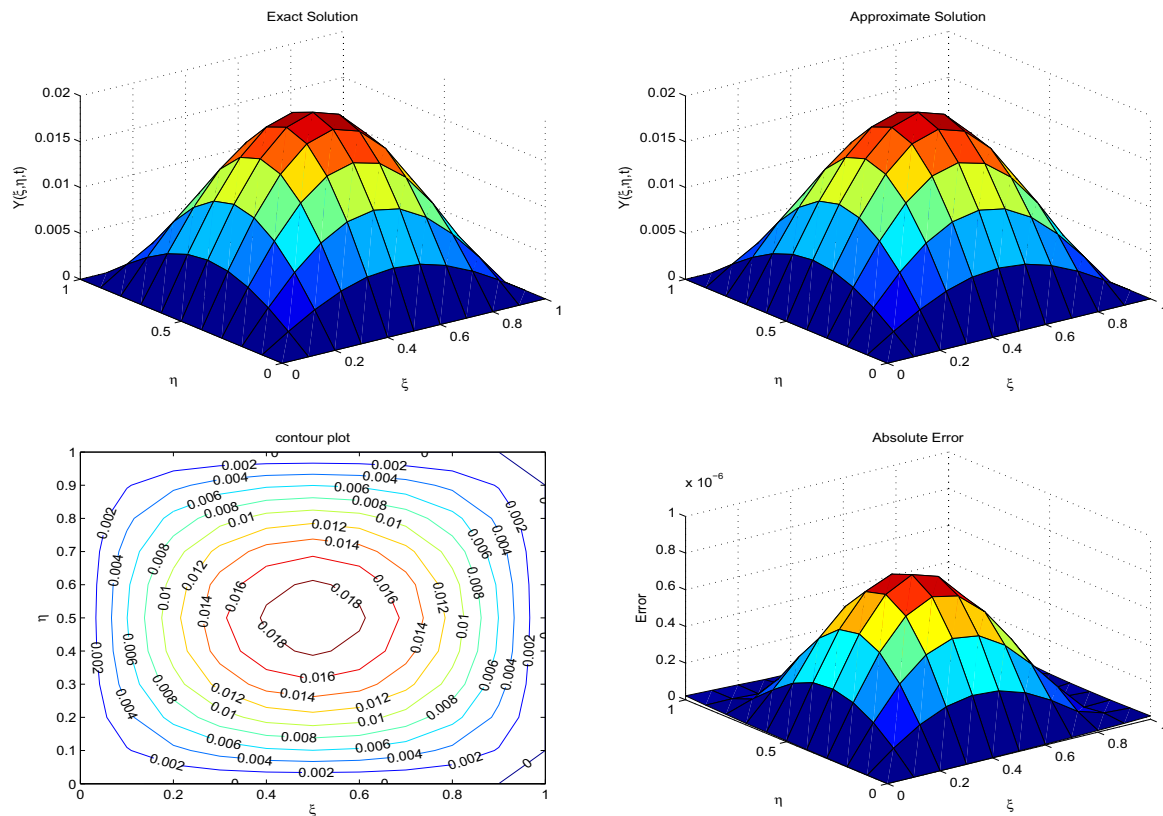
**Table 9** Error norms when  $T = 0.2$ , and  $\delta t = 0.001$  for Example 5.4

| Present method |            |            |            |          | [37]       |            |            |            |
|----------------|------------|------------|------------|----------|------------|------------|------------|------------|
|                |            |            |            |          | MQ RBF     | WL RBF     |            |            |
| N              | $L_\infty$ | $L_2$      | RMS        | CPU time | $L_\infty$ | RMS        | $L_\infty$ | RMS        |
| 25             | $3.15E-04$ | $3.83E-04$ | $5.29E-04$ | 0.137004 | $2.59E-04$ | $1.21E-04$ | $1.96E-03$ | $9.23E-04$ |
| 64             | $1.40E-05$ | $1.86E-05$ | $5.25E-05$ | 0.412875 | $2.35E-05$ | $1.22E-05$ | $3.49E-04$ | $1.82E-04$ |
| 100            | $8.17E-07$ | $8.60E-07$ | $7.60E-06$ | 0.671877 | $6.43E-06$ | $3.26E-06$ | $1.18E-04$ | $6.27E-05$ |

**Table 10** Error norms when  $N = 64$  and  $\delta t = 0.001$  of Example 5.4

| T          | 0.1        | 0.2        | 0.3        | 0.4        | 0.5        |
|------------|------------|------------|------------|------------|------------|
| $L_\infty$ | $5.01E-05$ | $1.40E-05$ | $2.98E-06$ | $5.52E-07$ | $9.54E-08$ |
| $L_2$      | $6.95E-05$ | $1.86E-05$ | $3.88E-06$ | $7.11E-07$ | $1.22E-07$ |
| RMS        | $1.02E-04$ | $5.25E-05$ | $2.40E-05$ | $1.03E-05$ | $4.26E-06$ |

test problems including one- and two-dimensional linear and nonlinear equations. The results are compared with exact as well numerical results available in the literature. The comparison of the results justify demonstrates efficiency and applicability of the proposed methodology.



**Fig. 5** Solution profile when  $T = 0.2$ ,  $\delta = 0.001$ ,  $N = 10$  of Example 5.4

**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 Faculty of Engineering Sciences, GIK Institute, Topi 23640, KPK, Pakistan.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

**Declarations**

**Conflict of interests** It is stated that we have no conflict of interest with anyone.

**Authors’ contributions** The idea of this paper was proposed by IA and prepared the manuscript initially. All authors read and approved the final manuscript.

**References**

1. Abd-Elhameed, W.M.; Youssri, Y.: A novel operational matrix of caputo fractional derivatives of Fibonacci polynomials: spectral solutions of fractional differential equations. *Entropy* **18**(10), 345 (2016)
2. Abd-Elhameed, W. M.; Youssri, H.: New formulas of the high-order derivatives of fifth-kind Chebyshev polynomials: Spectral solution of the convection-diffusion equation. *Numer. Methods Partial Differ. Equ.* (2021)
3. 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)
4. Abd-Elhameed, W.M.; Youssri, Y.H.: Generalized Lucas polynomial sequence approach for fractional differential equations. *Nonlinear Dyn.* **89**, 1341–1355 (2017)
5. Abd-Elhameed, W.M.; Youssri, Y.H.: Fifth-kind orthonormal Chebyshev polynomial solutions for fractional differential equations. *Comput. Appl. Math.* **37**(3), 2897–2921 (2018)
6. 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)

7. Abd-Elhameed, W.M.; Youssri, Y.H.: Sixth-kind Chebyshev spectral approach for solving fractional differential equations. *Int. J. Nonlinear Sci. Numer. Simul.* **20**(2), 191–203 (2019)
8. Ali, I.; Haq, S.; Nisar, K.S.; Baleanu, D.: An efficient numerical scheme based on Lucas polynomials for the study of multidimensional Burgers-type equations. *Adv. Differ. Equ.* **2021**(1), 1–24 (2021)
9. 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)
10. Bonkile, M.P.; Awasthi, A.; Lakshmi, C.; Mukundan, V.; Aswin, V.S.: A systematic literature review of Burgers' equation with recent advances. *Pramana.* **69**, (2018)
11. Cetin, M.; Sezer, M.; Guler, C.: Lucas polynomial approach for system of high-order linear differential equations and residual error estimation. *Math. Prob. Eng.* (2015)
12. Dabral, V.; Kapoor, S.; Dhawan, S.: Numerical simulation of one dimensional heat equation B-spline finite element method. *Indian J. Comput. Sci. Eng.* **2**, 222–235 (2011)
13. Dehghan, M.: A finite difference method for a non-local boundary value problem for two-dimensional heat equation. *Appl. Math. Comput.* **112**, 133–142 (2000)
14. El-Sayed, S.M.; Kaya, D.: On the numerical solution of the system of two-dimensional Burgers equations by decomposition method. *Appl. Math. Comput.* **158**, 101–109 (2004)
15. Haq, S.; Ali, I.: Approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials. *Eng. Comput.* 1–10 (2021)
16. Haq, S.; Ghafoor, A.: An efficient numerical algorithm for multi-dimensional time dependent partial differential equations. *Comput. Math. Appl.* **75**(8), 2723–2734 (2018)
17. Haq, S.; Ali, I.; Nisar, K.S.: A computational study of two-dimensional reaction-diffusion Brusselator system with applications in chemical processes. *Alexandria Eng. J.* **6**(5), 4381–4392 (2021)
18. Hooshmandasl, M.R.; Heydari, M.H.; Ghaini, F.M.: Numerical solution of the one-dimensional heat equation by using Chebyshev wavelets method. *Appl. Comput. Math.* **1**, 6 (2012)
19. Hussain, I.; Mukhtar, S.; Ali, A.: A numerical meshless technique for the solution of the two dimensional Burger's equation using collocation method. *World Appl. Sci. J.* **23**(12), 29–40 (2013)
20. Kaya, D.: An explicit solution of coupled Burgers' equations by decomposition method. *Int. J. Math. Math. Sci.* **27**, 675–680 (2001)
21. Koc, A.B.; Cakmak, M.; Kurnaz, A.; Uslu, K.: A new Fibonacci type collocation procedure for boundary value problems. *Adv. Differ. Equ.* **1**, 262 (2013)
22. Koshy, T.: *Fibonacci and Lucas numbers with applications.* Wiley, Hoboken (2019)
23. Kutluay, S.; Esen, A.; Dag, I.: Numerical solution of the Burgers' equation by the least-squares quadratic B-spline finite element method. *J. Comput. Appl. Math.* **167**, 21–33 (2004)
24. Liao, W.: A fourth-order finite-difference method for solving the system of two-dimensional Burgers equations. *Int. J. Numer. Meth. Fluids* **64**(5), 565–590 (2009)
25. Mebrate, B.: Numerical solution of a one dimensional heat equation with Dirichlet boundary conditions. *Am. J. Appl. Math.* **3**(6), 305–311 (2015)
26. Mirzaee, F.; Hoseini, S.F.: Application of Fibonacci collocation method for solving Volterra-Fredholm integral equations. *Appl. Math. Comput.* **273**, 637–44 (2016)
27. Mittal, R.C.; Jiware, R.: A differential quadrature method for numerical solutions of Burgers'-type equations. *Int. J. Numer. Meth. Heat Fluid Flow* **22**(7), 880–895 (2012)
28. Mohebbi, A.; Dehghan, M.: High-order compact solution of the one-dimensional heat and advection-diffusion equations. *Appl. Math. Model.* **10**, 3071–84 (2010)
29. Nadir, M.: Lucas polynomials for solving linear integral equations. *J. Theor. Appl. Comput. Sci.* **11**, 13–19 (2017)
30. Oruç, O.: A meshless multiple-scale polynomial method for numerical solution of 3d convection-diffusion problems with variable coefficients. *Eng. Comput.* 1–14 (2019)
31. 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)
32. 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)
33. Oruç, 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.* **3**(98), 461–479 (2021)
34. Oruç, O.; Esen, A.; Bulut, F.: A Haar wavelet approximation for two-dimensional time fractional reaction-subdiffusion equation. *Eng. Comput.* **1**(35), 75–86 (2019)
35. Ozis, T.; Aksan, E.N.; Ozdes, A.: A finite element approach for solution of Burger's equation. *Appl. Math. Comput.* **139**, 417–428 (2003)
36. Rainville, D.E.: *Special functions.* New York. (1960)
37. Uddin, M.: On the selection of a good value of shape parameter in solving time-dependent partial differential equations using RBF approximation method. *Appl. Math. Model.* **38**(1), 135–144 (2014)
38. Wazwaz, A.M.: *Partial differential equations and solitary waves theory.* Springer Science and Business Media, Berlin (2010)
39. Youssri, Y.H.; Abd-Elhameed, W.M.; Mohamed, A.S.; Sayed, S.M.: Generalized Lucas polynomial sequence treatment of fractional pantograph differential equation. *Int. J. Appl. Comput. Math.* **7**(2), 1–16 (2021)

