Alexandria Engineering Journal 82 (2023) 426–436

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original Article

Computational analysis of time-fractional models in energy infrastructure applications

Imtiaz Ahmad [a,*], Asmidar Abu Bakar [a,b,**], Ihteram Ali [c], Sirajul Haq [d], Salman Yussof [a,b], Ali Hasan Ali [e,f,g]

[a] Institute of Informatics and Computing in Energy (IICE), Universiti Tenaga Nasional (UNITEN), Kajang, Selangor 43000, Malaysia
[b] Department of Computing, College of Computing and Informatics (CCI), Universiti Tenaga Nasional (UNITEN), Kajang, Selangor 43000, Malaysia
[c] Department of Mathematics, Women University Swabi, Swabi, 23430, Pakistan
[d] Faculty of Engineering Sciences, GIK Institute of Engineering Sciences & Technology Topi, Pakistan
[e] Institute of Mathematics, University of Debrecen, Pf. 400, H-4002 Debrecen, Hungary
[f] College of Engineering Technology, National University of Science and Technology, 64001 Dhi Qar, Iraq
[g] Technical Engineering College, Al-Ayen University, 64001 Dhi Qar, Iraq

A R T I C L E   I N F O

Keywords:
Caputo derivative
Convection-diffusion equation
Finite differences
Lucas polynomials
Fibonacci polynomials
Energy infrastructure

A B S T R A C T

In this paper, we propose an effective numerical method to solve the one- and two-dimensional time-fractional convection-diffusion equations based on the Caputo derivative. The presented approach employs a hybrid method that combines Lucas and Fibonacci polynomials with the Caputo derivative definition. The main objective is to transform the problem into a time-discrete form utilizing the Caputo derivative technique and then approximate the function's derivative using Fibonacci polynomials. To evaluate the efficiency and accuracy of the proposed technique, we apply it to one- and two-dimensional problems and compare the results with the exact as well as with existing methods in recent literature. The comparison demonstrates that the proposed approach is highly efficient, accurate and ease to implement.

1. Introduction

Fractional calculus is a field of mathematics that extends the concept of differentiation and integration to non-integer orders, encompassing all real numbers. It finds practical applications in describing physical systems that exhibit non-local behavior, notably in phenomena such as diffusion in porous media [1–13]. Fractional differential operators play a crucial role in describing phenomena by enabling fractional differentiation and integration. The fractional derivative is a powerful tool for modeling materials and processes with memory-dependent properties, such as viscoelastic materials and fluids. Unlike the integer-order derivative, it considers the system's ability to retain its past behavior. As a result, fractional derivatives have been successfully applied to model intricate systems including turbulent flow [14], chaotic dynamics of classical conservation systems, and financial markets [15].

The family of convection-diffusion type partial differential equations (PDEs) is widely relevant across physical and biological sciences, finance, oceanography, environmental sciences, and energy infrastructure, where the transport and diffusion of energy-related substances play a crucial role. Notable examples include the Navier-Stokes equations [16] and Burgers' equation [17]. In practical applications, coupled systems of PDEs are used to model simultaneous reaction and transport processes in energy and environmental research. One of the specific applications of the convection-diffusion equation is in modeling heat transfer in buildings. By considering factors such as temperature gradients, airflow, and building materials, this equation aids in optimizing energy consumption and designing efficient heating, ventilation, and air conditioning (HVAC) systems [18]. These equations also play a crucial role in air pollution dispersion modeling, allowing researchers to assess environmental impacts and implement effective pollution control measures by simulating pollutant dispersion from industrial facilities, power plants, and transportation sources [19,20]. Convection-diffusion equations have versatile scientific and engineering applications, including weather prediction, simulating chemical dispersion in reactors, and modeling mass and energy transport in oil reservoirs [21]. These equations have numerous potential applications in the fields of oceanography, environmental sciences and also used to simulate the transport of

* Corresponding author.
** Corresponding author at: Institute of Informatics and Computing in Energy (IICE), Universiti Tenaga Nasional (UNITEN), Kajang, Selangor 43000, Malaysia.
E-mail addresses: imtiazkakakhil@gmail.com, imtiaz.ahmad@uniten.edu.my (I. Ahmad), Asmidar@uniten.edu.my (A. Bakar).

https://doi.org/10.1016/j.aej.2023.09.057
Received 1 June 2023; Received in revised form 1 September 2023; Accepted 22 September 2023

Available online 18 October 2023

contaminants in groundwater, especially in areas with energy-related activities like mining and oil extraction. Understanding the dispersion and migration of pollutants helps in managing and mitigating potential environmental risks. These equations can be utilized to create models that predict the spread of pollutants in the ocean and evaluate the effectiveness of different management strategies for reducing pollution levels [22,23]. Additionally, they can aid in assessing the potential risks associated with the release of pollutants in the ocean and identifying areas that are most susceptible to pollution [24,25]. Furthermore, the convection-diffusion equations can be employed in designing marine structures, such as offshore platforms and pipelines, that are resistant to pollution and can minimize the impact of pollutant dispersion [24,26]. Besides this, convection-diffusion PDEs can be used in electrical power transmission systems, to model the flow of electrical energy [27,28]. The convection term accounts for the movement of electricity due to differences in voltage potential (analogous to fluid flow), whereas the diffusion term describes how the energy spreads out and scatters. This can help analyze power losses, voltage drops, and optimization of transmission line configurations [28,29]. When electricity flows through cables, heat is generated due to resistive losses [30]. The convection-diffusion equation can be applied to study the temperature distribution along the cable's length, considering both heat conduction and heat dissipation through convection. This is crucial for ensuring cables operate within safe temperature limits [31,32]. Furthermore, such PDEs can be interpreted in terms of a probability distribution of underlying stochastic processes, which has applications in finance and option pricing [33]. These PDE models are therefore versatile and important mathematical tools with widespread applications. The fractional convection-diffusion model is derived from Fick's first law [34]. The focus of our consideration is the time-fractional convection-diffusion equation (TFCDE) [35] with variable coefficients, which can be expressed as follows:

$$\frac{\partial^\alpha w(\bar{\mathbf{s}},t)}{\partial t^\alpha}+\beta_1(\bar{\mathbf{s}})\Delta w(\bar{\mathbf{s}},t)-\beta_2(\bar{\mathbf{s}})\nabla w(\bar{\mathbf{s}},t)=g(\bar{\mathbf{s}},t),\quad \bar{\mathbf{s}}\in\Omega\subset\mathbb{R}^n,\quad t>0, \tag{1}$$

with initial condition

$$w(\bar{\mathbf{s}},0)=w_0, \tag{2}$$

and boundary conditions

$$\mathcal{B}w(\bar{\mathbf{s}},t)=f_1(\bar{\mathbf{s}},t),\quad \bar{\mathbf{s}}\in\partial\Omega,\quad t>0, \tag{3}$$

where β_1 (velocity variable coefficient), β_2 (diffusion variable coefficient) and $w(\bar{\mathbf{s}},t)$ either represents concentration or temperature for mass or heat transfer respectively. Numerous researchers have examined the existence and characteristics of solutions for convection-diffusion equations, as evidenced in the works of several authors [36–39].

The TFCDE finds various applications in science and engineering fields, including fluid dynamics, heat transfer, and chemical engineering [40–45]. In fluid dynamics, it models transport phenomena in porous media, such as groundwater flow and oil recovery, atmospheric dispersion, oceanic mixing, and blood flow in biological systems [46]. In heat transfer, the TFCDE explores materials with non-Fourier heat conduction, exhibiting time lag, indispensable for temperature control in electronics and aerospace [47]. In chemical engineering, it models solute transport through porous membranes, catalyst behavior in reaction-diffusion systems, drug delivery systems, and energy storage materials design [48].

Due to its diverse applications, a wide range of numerical and analytical methods [49–54] has been explored by researchers to solve the TFCDE. These methods includ the Laplace transform technique [55], Adomian decomposition method [56], homotopy perturbation method [57], alternating direction implicit method [58], finite difference methods (FDM) [59], finite element methods (FEM) [60], Galerkin mixed finite element [61], spectral methods [62], and meshless methods [63]. FDM, FEM and meshless methods are particularly popular due to their simplicity. Recently, some hybrid methods, such as the combination of FDM and FEM or FDM, meshless methods and method of line based on Fibonacci polynomials, have also been proposed to improve the accuracy and efficiency of the numerical solutions.

This study employs an efficient hybrid approach to compute numerical solutions for the underlying model equations in the sense of Caputo derivative. The method utilizes a combination of Fibonacci polynomials and the well-known Caputo derivative definition. A notable advantage of this method is its ease of implementation in higher-order derivatives through the use of the relationship between Lucas and Fibonacci polynomials. In addition, the suggested method enhances the accuracy of computations even when using a limited number of collocation points, leading to reduced computing expenses. These polynomials have various practical applications in the field of differential equations. For instance, the connection between Chebyshev and Lucas polynomials is examined in [64,65], which facilitates the acquisition of accurate solutions for boundary value problems. The Lucas sequence was employed in [66] as an approximation technique for integro-differential equations. Meanwhile, in [67], the Lucas polynomial technique was utilized to solve higher-order differential equations. Additionally, in [68], a Fibonacci polynomial approach was employed to solve Volterra-Fredholm integral differential equations. Lastly, a hybrid Taylor-Lucas polynomial approach was used to solve delay difference equations, as described in [69]. A novel method was introduced by the authors in their publications [70,71], which utilizes a hybrid Lucas and Fibonacci polynomial scheme to solve time-dependent PDEs. Other researchers have also utilized Lucas polynomials and finite-differences to obtain efficient numerical solutions for various types of PDE models, as evidenced by publications such as [72–74].

1.1. Motivation

Computing the analytical solutions of nonlinear PDEs is a challenging task. Therefore, researchers continuously seek to develop efficient and accurate numerical solutions. The convection-diffusion model, a crucial PDE, and is utilized in electrical power transmission to model energy flow: convection accounts for electricity movement due to voltage differences, and diffusion depicts energy dispersion. This aids in analyzing power losses, voltage drops, and optimizing transmission. In cable electricity flow, convection-diffusion equation studies temperature distribution, vital for safe cable operation by considering heat conduction and dissipation. Diverse numerical approaches have been explored and evaluated in the literature, each with its advantages and limitations. In this article, our objective is to propose an efficient numerical scheme for the aforementioned time-fractional PDEs. The suggested scheme is hybrid, combining finite difference methods with Fibonacci and Lucas Polynomials. Unlike orthogonal polynomials such as Chebyshev polynomials, Fibonacci and Lucas Polynomials are non-orthogonal and do not require interval transformations. Moreover, these polynomials enable easy approximation of higher-order derivatives for the unknown functions. Furthermore, the scheme exhibits simplicity and achieves improved accuracy even with fewer nodal points. Through this approach, we aim to present an effective and robust numerical solution for tackling the complexities of nonlinear PDEs.

The paper is structured as follows: Section 2 presents an overview of essential concepts and definitions. Section 3 explains the proposed methodology for the scheme whereas theoretical results related to the error analysis and stability are given in Section 4. Section 5 verifies the effectiveness of the method through numerical experiments while the paper concludes with a summary of findings and concluding remarks in Section 6.

1.2. Theory of Lucas and Fibonacci polynomials

The definitions and useful results for Lucas and Fibonacci polynomials, along with their use for approximation of unknown function and its derivatives, are presented as follows.

Fibonacci polynomials [72]

The Fibonacci polynomial is generalization of Fibonacci numbers and can be defined by the three-term recurrence relation

$$F_k(s) = kF_{k-1}(s) + F_{k-2}(s), \quad k \geq 2, \tag{4}$$

with initial values $F_0(s) = 0$ and $F_1(s) = 1$. For $s = 1$ Eq. (4) generate a sequence of well known Fibonacci numbers.

Lucas polynomials [72]

The Lucas polynomials can be defined by the three-term recurrence relation

$$L_k(s) = kL_{k-1}(s) + L_{k-2}(s), \quad k \geq 2, \tag{5}$$

with initial values $L_0(s) = 2$ and $L_1(s) = s$. By letting $s = 1$ Eq. (5) generate a sequence of Lucas numbers.

Lemma *([72]). The mth-order derivative of kth Lucas polynomial $L_k(s)$ can be expressed in terms of kth Fibonacci polynomial $F_k(s)$ by the following relation*

$$L_k^{(m)}(s) = kF_k(s)D^{m-1}, \quad D^{m-1} = \underbrace{D \times D \times D \cdots D}_{(m-1)time} \tag{6}$$

where D represents a square matrix of order $(M+1)\times(M+1)$ be defined as

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

where d can be obtained as [72]

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

1.3. Function approximation

Let $w(\bar{\mathbf{s}})$ be a continuous function with $w \in L^2(\mathbb{R})$. Then one can expand w using linear combination of kth Lucas polynomials as

$$w(\bar{\mathbf{s}}) = \sum_{k=0}^{\infty} \Lambda_k L_k(\bar{\mathbf{s}}), \tag{7}$$

where Λ_k are the unknown coefficients and $L_k(\bar{\mathbf{s}})$ are Lucas polynomials.

Similarly, under the same conditions, $w(\bar{\mathbf{s}})$ can also be expanded using linear combination of kth Fibonacci polynomials as follows

$$w(\bar{\mathbf{s}}) = \sum_{k=0}^{\infty} \Lambda_k F_k(\bar{\mathbf{s}}),$$

where Λ_k are the unknown coefficients and $F_k(\bar{\mathbf{s}})$ are Fibonacci polynomials.

The first-order derivative of the function $w(\bar{\mathbf{s}})$ can be expanded by Lucas polynomial series as

$$w'(\bar{\mathbf{s}}) = \sum_{k=0}^{\infty} \Lambda_k L'_k(\bar{\mathbf{s}}), \tag{8}$$

and the corresponding mth order derivative of the function $w(\bar{\mathbf{s}})$ is expressed as

$$w^m(\bar{\mathbf{s}}) = \sum_{k=0}^{\infty} \Lambda_k L_k^{(m)}(\bar{\mathbf{s}}), \tag{9}$$

where

$$w^m(\bar{\mathbf{s}}) = \frac{\mathrm{d}^m w(\bar{\mathbf{s}})}{\mathrm{d}\bar{\mathbf{s}}^m}, \quad L_k^{(m)}(\bar{\mathbf{s}}) = \frac{\mathrm{d}^m L_k(\bar{\mathbf{s}})}{\mathrm{d}\bar{\mathbf{s}}^m}.$$

By incorporating the relation (6) in to Eqs. (8) and (9), we can write then

$$w'(\bar{\mathbf{s}}) = \sum_{k=0}^{\infty} \Lambda_k k F_k(\bar{\mathbf{s}}), \tag{10}$$

and the corresponding mth order derivative of the function $w(\bar{\mathbf{s}})$ as

$$w^{(m)}(\bar{\mathbf{s}}) = \sum_{k=0}^{\infty} \Lambda_k k F_k(\bar{\mathbf{s}}) D^{m-1}, \tag{11}$$

where D and D^{m-1} as defined before.

Remark 1. During numerical calculations, one usually expand the unknown function $w(\bar{\mathbf{s}})$ and its mth order derivative by truncated Lucas and Fibonacci polynomial series. That is, we take

$$w(\bar{\mathbf{s}}) \simeq \sum_{k=0}^{M} \Lambda_k L_k(\bar{\mathbf{s}}), \quad M \in \mathbb{N}$$

and

$$w^{(m)}(\bar{\mathbf{s}}) \simeq \sum_{k=0}^{M} \Lambda_k L_k^{(m)}(\bar{\mathbf{s}}) = \sum_{k=0}^{M} \Lambda_k k F_k(\bar{\mathbf{s}}) D^{m-1}, \quad M \in \mathbb{N} \tag{12}$$

2. Fractional calculus

Fractional derivatives are essential in fractional calculus. The following are some fundamental definitions of fractional derivatives that are commonly utilized.

Definition 1. The Riemann-Liouville derivative [75,76]

$$\frac{\partial^\alpha w(\bar{\mathbf{s}}, t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_t^T \frac{(w(\bar{\mathbf{s}}, \vartheta) - w(\bar{\mathbf{s}}, T))}{(\vartheta - t)^\alpha} d\vartheta, \quad 0 < \alpha < 1. \tag{13}$$

Definition 2. The Caputo's fractional derivative [77]

$$\frac{\partial^\alpha w(\bar{\mathbf{s}}, t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial w(\bar{\mathbf{s}}, \zeta)}{\partial \zeta} (t-\zeta)^{-\alpha} d\zeta, \quad 0 < \alpha < 1. \tag{14}$$

Definition 3. The Atangana and Baleanu fractional derivative [78]

$${}_a^{ABC}\frac{\partial^\alpha w(\bar{\mathbf{s}}, t)}{\partial t^\alpha} = \frac{B(\alpha)}{1-\alpha} \int_a^t w'(\bar{\mathbf{s}}) E_\alpha\left(-\frac{\alpha(t-\bar{\mathbf{s}})^\alpha}{1-\alpha}\right) d\bar{\mathbf{s}}, \quad 0 < \alpha < 1. \tag{15}$$

Definition 4. He's fractional derivative [79]

$$\frac{\partial^\alpha w(\bar{\mathbf{s}}, t)}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)} \frac{d}{d\bar{\mathbf{s}}} \int_{t_0}^t (t-\zeta)^{-\alpha}[w_0(\zeta) - w(\zeta)], \quad 0 < \alpha < 1. \tag{16}$$

3. Proposed solution methodology

In this section, we formulate the proposed numerical scheme for approximation of 2D TFDE given in Eqs. (1)-(3). For ease of notations, we introduce the following throughout the discussion in this section:

$$w^{n+1}(\bar{\mathbf{s}}) = w(s, r, t^{n+1}), \quad w_{ij}^{n+1} = w(s_i, r_j, t^{n+1})$$

where $t^n = n \times \tau$ with τ being time-step size, and s_i, r_j are the collocation points defined as

$$s_i = a + ih_s, \quad r_j = c + jh_r \quad (i, j = 1, 2, \cdots, M, \ M \in \mathbb{N}),$$

where $h_s = (b-a)/M$ and $h_r = (d-c)/M$ are mesh-step sizes (equal in size) in s and r spatial direction over the spatial domain $\Xi = [a, b] \times [c, d] \subset \mathbb{R}^2$.

3.1. Time discretization

The discrete form of time-fractional derivative is obtained by using the well-known $L1$-formula with the order of approximation error $\mathcal{O}(\tau^{2-\alpha})$, $(0<\alpha\le 1)$ [80]. This results in the following discretization at $(n+1)^{th}$ time level ([80,81]):

$$\frac{\partial^\alpha w(\bar{\mathbf{s}},t^{n+1})}{\partial t^\alpha} = \frac{1}{\Gamma(1-\alpha)}\int_0^{t^{n+1}} \frac{\partial w(\bar{\mathbf{s}},\zeta)}{\partial \zeta}(t^{n+1}-\zeta)^{-\alpha}\mathrm{d}\zeta,$$

$$= \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{n}\int_{j\times\tau}^{(j+1)\times\tau} \frac{\partial w(\bar{\mathbf{s}},\zeta)}{\partial \zeta}(t^{n+1}-\zeta)^{-\alpha}\mathrm{d}\zeta,$$

$$= \frac{1}{\Gamma(1-\alpha)}\sum_{j=0}^{n}\left[\frac{w^{j+1}(\bar{\mathbf{s}})-w^{j}(\bar{\mathbf{s}})}{\tau}+O(\tau)\right]\int_{j\times\tau}^{(j+1)\times\tau}((j+1)\tau-\zeta)^{-\alpha}\mathrm{d}\zeta.$$

After simple integration it yields

$$\frac{\partial^\alpha w(s,r,t^{n+1})}{\partial t^\alpha} = \begin{cases} A_\alpha \sum_{j=0}^{n} K_\alpha(j)\left[w^{n-j+1}(\bar{\mathbf{s}})-w^{n-j}(\bar{\mathbf{s}})\right]+O(\tau^{2-\alpha}), \\ \quad 0<\alpha<1, \\ \frac{w^{n+1}(\bar{\mathbf{s}})-w^{n}(\bar{\mathbf{s}})}{\tau}+O(\tau), \quad \alpha=1, \end{cases} \tag{17}$$

where $A_\alpha = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}$ and $K_\alpha(j)=(j+1)^{1-\alpha}-(j)^{1-\alpha}$. Hence for $0<\alpha<1$ we can write then after neglecting the error term

$$\frac{\partial^\alpha w(\bar{\mathbf{s}},t^{n+1})}{\partial t^\alpha} = A_\alpha\left[w^{n+1}(\bar{\mathbf{s}})-w^{n}(\bar{\mathbf{s}})\right] + A_\alpha\sum_{j=1}^{n}K_\alpha(j)\left[w^{n-j+1}(\bar{\mathbf{s}})-w^{n-j}(\bar{\mathbf{s}})\right], \tag{18}$$

with $K_\alpha(j)=1, j=0$.

Now by using the $\theta-$ weighted rule Eq. (1) becomes

$$\frac{\partial^\alpha w(\bar{\mathbf{s}},t^{n+1})}{\partial t^\alpha}+\theta\Big[\beta_1(\bar{\mathbf{s}})\Delta w^{n+1}(\bar{\mathbf{s}})+\beta_2(\bar{\mathbf{s}})\nabla w^{n+1}(\bar{\mathbf{s}})\Big] = (\theta-1)\Big[\beta_1(\bar{\mathbf{s}})\Delta w^{n}(\bar{\mathbf{s}})+\beta_2(\bar{\mathbf{s}})\nabla w^{n}(\bar{\mathbf{s}})\Big]+\theta g^{n+1}(\bar{\mathbf{s}})+(1-\theta)g^{n}(\bar{\mathbf{s}}),$$

which after putting relation (18) and simplification gives

$$A_\alpha w^{n+1}(\bar{\mathbf{s}})+\theta\Big[\beta_1(\bar{\mathbf{s}})\Delta w^{n+1}(\bar{\mathbf{s}})+\beta_2(\bar{\mathbf{s}})\nabla w^{n+1}(\bar{\mathbf{s}})\Big] = A_\alpha w^{n}(\bar{\mathbf{s}})+(\theta-1)\Big[\beta_1(\bar{\mathbf{s}})\Delta w^{n}(\bar{\mathbf{s}})+\beta_2(\bar{\mathbf{s}})\nabla w^{n}(\bar{\mathbf{s}})\Big] + \theta g^{n+1}(\bar{\mathbf{s}})+(1-\theta)g^{n}(\bar{\mathbf{s}})-A_\alpha\sum_{j=1}^{n}K_\alpha(j)\left[w^{n-j+1}(\bar{\mathbf{s}})-w^{n-j}(\bar{\mathbf{s}})\right]. \tag{19}$$

Letting

$$B_\alpha^n(s,r)=A_\alpha\sum_{j=1}^{n}K_\alpha(j)\left[w^{n-j+1}(s,r)-w^{n-j}(s,r)\right],$$

and

$$g_\theta^n(\bar{\mathbf{s}})=\theta g^{n+1}(\bar{\mathbf{s}})+(1-\theta)g^{n}(\bar{\mathbf{s}}),$$

Eq. (19) can be further simplified as

$$A_\alpha w^{n+1}(\bar{\mathbf{s}})+\theta\Big[\beta_1(\bar{\mathbf{s}})\Delta w^{n+1}(\bar{\mathbf{s}})+\beta_2(\bar{\mathbf{s}})\nabla w^{n+1}(\bar{\mathbf{s}})\Big] = A_\alpha w^{n}(\bar{\mathbf{s}})+(\theta-1)\Big[\beta_1(\bar{\mathbf{s}})\Delta w^{n}(\bar{\mathbf{s}})+\beta_2(\bar{\mathbf{s}})\nabla w^{n}(\bar{\mathbf{s}})\Big]+g_\theta^n(\bar{\mathbf{s}})-B_\alpha^n(\bar{\mathbf{s}}). \tag{20}$$

At this end, Eq. (20) is the time discrete form of Eq. (1).

3.2. Space discretization

Let us expand a 2D function $w(\bar{\mathbf{s}})=w(s,r)$ as a product of truncated Lucas polynomials series in the following form

$$w(s,r)\simeq\sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}L_k(s)L_m(r)=\sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}\mathcal{L}_{km}(s,r), \tag{21}$$

where $\mathcal{L}_{km}(s,r)=L_k(s)L_m(r)$. In compact form (21) can be written as

$$w(s,r)\simeq\mathbf{L}^{\mathrm{T}}(s,r)\mathbf{\Lambda}, \tag{22}$$

with $\mathbf{\Lambda}=[\Lambda_{00},\Lambda_{01},\Lambda_{10},\cdots,\Lambda_{MM}]^{\mathrm{T}}$ and $\mathbf{L}^{\mathrm{T}}(s,r)=[\mathcal{L}_{00}(s,r),\mathcal{L}_{01}(s,r),\mathcal{L}_{10}(s,r),\cdots,\mathcal{L}_{MM}(s,r)]$.

The corresponding partial derivatives of $w(s,r)$ can be computed and expressed in compact as follows:

$$\frac{\partial w(s,r)}{\partial s}\simeq\sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}\frac{\mathrm{d}L_k(s)}{\mathrm{d}s}L_m(r) = \sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}k(F_k(s))L_m(r)=\mathbf{L}_s^{\mathrm{T}}(s,r)\mathbf{\Lambda},$$

and

$$\frac{\partial^2 w(s,r)}{\partial s^2}\simeq\sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}\frac{\mathrm{d}^2L_k(s)}{\mathrm{d}s^2}L_m(r) = \sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}k(F_k(s)D)L_m(r)=\mathbf{L}_{ss}^{\mathrm{T}}(s,r)\mathbf{\Lambda},$$

where

$$\mathbf{L}_s^{\mathrm{T}}(s,r)=\{\frac{\partial}{\partial s}\mathcal{L}_{km}(s,r)\}_{k,m=0}^{M} = \{\frac{\mathrm{d}L_k(s)}{\mathrm{d}s}L_m(r)\}_{k,m=0}^{M}=\{k(F_k(s))L_m(r)\}_{k,m=0}^{M},$$

$$\mathbf{L}_{ss}^{\mathrm{T}}(s,r)=\{\frac{\partial^2}{\partial s^2}\mathcal{L}_{km}(s,r)\}_{k,m=0}^{M} = \{\frac{\mathrm{d}^2L_k(s)}{\mathrm{d}s^2}L_m(r)\}_{k,m=0}^{M}=\{k(F_k(s)D)L_m(r)\}_{k,m=0}^{M}.$$

Similarly

$$\frac{\partial w(s,r)}{\partial r}\simeq\sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}L_k(s)\frac{\mathrm{d}L_m(r)}{\mathrm{d}r} = \sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}mL_k(s)(F_m(r))=\mathbf{L}_r^{\mathrm{T}}(s,r)\mathbf{\Lambda},$$

and

$$\frac{\partial^2 w(s,r)}{\partial s^2}\simeq\sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}L_k(s)\frac{\mathrm{d}^2L_m(r)}{\mathrm{d}r^2} = \sum_{k=0}^{M}\sum_{m=0}^{M}\Lambda_{km}mL_k(s)(F_m(r)D)=\mathbf{L}_{rr}^{\mathrm{T}}(s,r)\mathbf{\Lambda},$$

with

$$\mathbf{L}_r^{\mathrm{T}}(s,r)=\{\frac{\partial}{\partial r}\mathcal{L}_{km}(s,r)\}_{k,m=0}^{M} = \{L_k(s)\frac{\mathrm{d}^2L_m(r)}{\mathrm{d}r^2}\}_{k,m=0}^{M}=\{mL_k(s)(F_m(r))\}_{k,m=0}^{M}.$$

$$\mathbf{L}_{rr}^{\mathrm{T}}(s,r)=\{\frac{\partial^2}{\partial r^2}\mathcal{L}_{km}(s,r)\}_{k,m=0}^{M} = \{L_k(s)\frac{\mathrm{d}^2L_m(r)}{\mathrm{d}r^2}\}_{k,m=0}^{M}=\{mL_k(s)(F_m(r)D)\}_{k,m=0}^{M}.$$

This results in the following expanded form of 1D and 2D Laplace operator

$$\nabla w(s,r)=\frac{\partial w}{\partial s}+\frac{\partial w}{\partial r}\simeq\left(\mathbf{L}_s^{\mathrm{T}}(s,r)+\mathbf{L}_r^{\mathrm{T}}(s,r)\right)\mathbf{\Lambda}. \tag{23}$$

$$\Delta w(s,r)=\frac{\partial^2 w}{\partial s^2}+\frac{\partial^2 w}{\partial r^2}\simeq\left(\mathbf{L}_{ss}^{\mathrm{T}}(s,r)+\mathbf{L}_{rr}^{\mathrm{T}}(s,r)\right)\mathbf{\Lambda}. \tag{24}$$

Accordingly, we can express the time-dependent function $w(s,r,t)$ and its partial derivatives in compact form as follows:

$$w(s,r,t) \approx \mathbf{L}^{\mathrm{T}}(s,r)\boldsymbol{\Lambda}(t), \quad \frac{\partial^2 w(s,r,t)}{\partial s^2} \approx \mathbf{L}_{ss}^{\mathrm{T}}(s,r)\boldsymbol{\Lambda}(t),$$
$$\frac{\partial^2 w(s,r,t)}{\partial r^2} \approx \mathbf{L}_{rr}^{\mathrm{T}}(s,r)\boldsymbol{\Lambda}(t), \tag{25}$$

with

$$\nabla w(s,r,t) \approx \left(\mathbf{L}_{s}^{\mathrm{T}}(s,r) + \mathbf{L}_{r}^{\mathrm{T}}(s,r)\right)\boldsymbol{\Lambda}(t),$$
$$\Delta w(s,r,t) \approx \left(\mathbf{L}_{ss}^{\mathrm{T}}(s,r) + \mathbf{L}_{rr}^{\mathrm{T}}(s,r)\right)\boldsymbol{\Lambda}(t), \tag{26}$$

where $\boldsymbol{\Lambda}(t)$ is the vector of time-dependent unknown coefficients. On processing the information from Eqs. (25) and (26) to Eq. (20), we have

$$\begin{aligned} &A_\alpha \mathbf{L}(s,r)\boldsymbol{\Lambda}^{n+1} + \theta\Big[\beta_1(s,r)\Big(\mathbf{L}_{ss}^{\mathrm{T}}(s,r) + \mathbf{L}_{rr}^{\mathrm{T}}(s,r)\Big) \\ &\quad + \beta_2(s,r)\Big(\mathbf{L}_{s}^{\mathrm{T}}(s,r) + \mathbf{L}_{r}^{\mathrm{T}}(s,r)\Big)\Big]\boldsymbol{\Lambda}^{n+1} \\ &= A_\alpha \mathbf{L}(s,r)\boldsymbol{\Lambda}^{n} + (\theta-1)\Big[\beta_1(s,r)\Big(\mathbf{L}_{ss}^{\mathrm{T}}(s,r) + \mathbf{L}_{rr}^{\mathrm{T}}(s,r)\Big) \\ &\quad + \beta_2(s,r)\Big(\mathbf{L}_{s}^{\mathrm{T}}(s,r) + \mathbf{L}_{r}^{\mathrm{T}}(s,r)\Big)\Big]\boldsymbol{\Lambda}^{n} \\ &\quad + g_\theta^{n+1}(s,r) - B_\alpha^n(s,r), \quad \forall (s,r) \in \Omega. \end{aligned} \tag{27}$$

The boundary conditions are expressed as

$$\mathbb{B}\mathbf{L}(s,r)\boldsymbol{\Lambda}^{n+1} \approx f_1(s,r,t^{n+1}) = f_1^{n+1}(s,r), \quad \forall (s,r) \in \partial\Omega \tag{28}$$

where $\boldsymbol{\Lambda}^n = \boldsymbol{\Lambda}(t^n)$.

3.3. Full discretization

In order to obtain the full discrete form of the TFDE given in Eq. (1), we collocate Eq. (27) and (28) at discrete mesh points s_i and r_j at the respective time-levels. This gives us the following system of equations

$$\begin{aligned} &A_\alpha \mathbf{L}(s_i,r_j)\boldsymbol{\Lambda}^{n+1} + \theta\Big[\beta_1(s_i,r_j)\Big(\mathbf{L}_{ss}^{\mathrm{T}}(s_i,rj) + \mathbf{L}_{rr}^{\mathrm{T}}(s_i,r_j)\Big) \\ &\quad + \beta_2(s_i,r_j)\Big(\mathbf{L}_{s}^{\mathrm{T}}(s_i,r_j) + \mathbf{L}_{r}^{\mathrm{T}}(s_i,r_j)\Big)\Big]\boldsymbol{\Lambda}^{n+1} \\ &= A_\alpha \mathbf{L}(s_i,r_j)\boldsymbol{\Lambda}^{n} + (\theta-1)\Big[\beta_1(s_i,r_j)\Big(\mathbf{L}_{ss}^{\mathrm{T}}(s_i,r_j) + \mathbf{L}_{rr}^{\mathrm{T}}(s_i,r_j)\Big) \\ &\quad + \beta_2(s_i,r_j)\Big(\mathbf{L}_{s}^{\mathrm{T}}(s_i,r_j) + \mathbf{L}_{r}^{\mathrm{T}}(s_i,r_j)\Big)\Big]\boldsymbol{\Lambda}^{n} \\ &\quad + g_\theta^{n+1}(s_i,r_j) - B_\alpha^n(s_i,r_j), \quad \forall (s_i,r_j) \in \Omega, \end{aligned}$$

and

$$\mathbb{B}\mathbf{L}(s_i,r_j)\boldsymbol{\Lambda}^{n+1} \approx f_1(s_i,r_j,t^{n+1}) = f_1^{n+1}(s_i,r_j), \quad \forall (s_i,r_j) \in \partial\Omega$$

where $i,j = 0,1,2,\cdots,M$. In matrix-vector form this implies

$$\mathbf{G}\boldsymbol{\Lambda}^{n+1} = \mathbf{H}\boldsymbol{\Lambda}^{n} + \mathbf{Q}^{n+1}, \tag{29}$$

where the entries of the matrices are given by

$$(\mathbf{G})_{ij} = \begin{cases} A_\alpha \mathbf{L}(s_i,r_j) + \theta\Big[\beta_1(s_i,r_j)\Big(\mathbf{L}_{ss}^{\mathrm{T}}(s_i,rj) + \mathbf{L}_{rr}^{\mathrm{T}}(s_i,r_j)\Big) & \\ + \beta_2(s_i,r_j)\Big(\mathbf{L}_{s}^{\mathrm{T}}(s_i,r_j) + \mathbf{L}_{r}^{\mathrm{T}}(s_i,r_j)\Big)\Big], & (s_i,r_j) \in \Omega, \\ \mathbb{B}\mathbf{L}(s_i,r_j), & (s_i,r_j) \in \partial\Omega, \end{cases} \tag{30}$$

$$(\mathbf{H})_{ij} = \begin{cases} A_\alpha \mathbf{L}(s_i,r_j) + (\theta-1)\Big[\beta_1(s_i,r_j)\Big(\mathbf{L}_{ss}^{\mathrm{T}}(s_i,r_j) + \mathbf{L}_{rr}^{\mathrm{T}}(s_i,r_j)\Big) \\ + \beta_2(s_i,r_j)\Big(\mathbf{L}_{s}^{\mathrm{T}}(s_i,r_j) + \mathbf{L}_{r}^{\mathrm{T}}(s_i,r_j)\Big)\Big], \\ \quad (s_i,r_j) \in \Omega, \\ 0, \quad (s_i,r_j) \in \partial\Omega, \end{cases} \tag{31}$$

and

$$(\mathbf{Q})_{ij} = \begin{cases} g^{n+1}(s_i,r_j) - B_\alpha^n(s_i,r_j), & (s_i,r_j) \in \Omega, \\ f_1(s_i,r_j,t^{n+1}), & (s_i,r_j) \in \partial\Omega. \end{cases} \tag{32}$$

This completes the scheme for the two-dimensional model equation. The same procedure can be applied for the one-dimensional case by disregarding the variable r, i.e. $\bar{\mathbf{s}} = s$. The existence of a solution to the linear system (29) is guaranteed by the works of [70,82]. By solving the linear system (29) at each time-level, the vector of unknown coefficients can be determined. For initialization we use the initial condition as follows:

$$\mathbf{L}^{\mathrm{T}}(s_i,r_j)\boldsymbol{\Lambda}^0 = w(s_i,r_j,0) = w_0 \Longrightarrow \mathbf{A}\boldsymbol{\Lambda}^0 = \mathbf{w_0},$$

which on solving gives $\boldsymbol{\Lambda}^0$, and then the iteration is repeated up to desired time-level using Eq. (29). Once the coefficients vector is available, solution then can be obtained at the desired time-level by

$$w(s_i,r_j,t^n) = \mathbf{L}^{\mathrm{T}}(s_i,r_j)\boldsymbol{\Lambda}^n, \quad \forall (s_i,r_j) \in \Omega, \quad (n \geq 0).$$

This complete the scheme for two dimension diffusion equation which can be reduce to one dimension by ignoring the variable r, i.e. $\bar{\mathbf{s}} = s$.

4. Error analysis

We will consult the following theorem for the error analysis.

Theorem. *If we denote the exact and approximate solutions of the given problem as E and w respectively, the error can be bounded as follows:*

$$\mathcal{E} \leq \frac{4exp(2\kappa)\cosh^2(2P)\kappa^2(M+1)}{((M+1)!)2}.$$

Proof. See [83].

4.1. Stability analysis

In this study, the stability analysis is conducted using the matrix method. To accomplish this, we adopt the approach outlined in previous works [84,85].

Theorem. *Let w represent the approximate solution to the given problem. In this case, the amplification matrix can be expressed as $_F = L\mathbf{G}^{-1}HL^{-1}$. To ensure stability, we need to verify that the absolute maximum eigenvalue value of the matrix $_F$, denoted as $\rho(_F)$, is less than or equal to 1.*

Proof. To show that $_F = L\mathbf{G}^{-1}\mathbf{H}L^{-1}$. We write from Eq. (12)

$$\boldsymbol{\Lambda}^{n+1} = L^{-1}w^n. \tag{33}$$

One can express the relationship as given in Eq. (29).

$$w^{n+1} = L\mathbf{G}^{-1}\mathbf{H}L^{-1}w^n + L\mathbf{G}^{-1}\mathbf{Q}^{n+1}. \tag{34}$$

Further the error vector

$$\mathcal{E} = E - w, \tag{35}$$

will meet the following condition

$$\mathcal{E}^{n+1} = {}_F\mathcal{E}^n.$$

The amplification matrix, denoted by $_F$, is defined as $_F = L\mathbf{G}^{-1}\mathbf{H}L^{-1}$. The stability of the method can be ensured by satisfying the Lax–Richtmyer criterion of stability [86], which requires that

$$||_F|| \leq 1.$$

In the following section, example is presented to demonstrate that the spectral radius $\rho(_F)$ must be less than or equal to 1 to satisfy the requirement.

Table 1
The outcomes obtained from utilizing the suggested approach to address Problem 1.

	Current scheme			Cited work [87]		
s	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.8$
0.1	5.8079E-07	2.0118E-06	2.2748E-06	4.7422E-04	7.2102E-05	7.5119E-05
0.2	1.1507E-06	3.9789E-06	4.4790E-06	4.9962E-04	1.1687E-04	1.4758E-04
0.3	1.6439E-06	5.6710E-06	6.3461E-06	5.2219E-04	1.5602E-04	2.1062E-04
0.4	2.0017E-06	6.8868E-06	7.6553E-06	5.3907E-04	1.8481E-04	2.5675E-04
0.5	2.1782E-06	7.4726E-06	8.2483E-06	5.4795E-04	1.9944E-04	2.7994E-04
0.6	2.1443E-06	7.3353E-06	8.0395E-06	5.4727E-04	1.9743E-04	2.7630E-04
0.7	1.8903E-06	6.4490E-06	7.0202E-06	5.3633E-04	1.7781E-04	2.4437E-04
0.8	1.4265E-06	4.8553E-06	5.2537E-06	5.1540E-04	1.4119E-04	1.8525E-04
0.9	7.8178E-07	2.6563E-06	2.8612E-06	4.8564E-04	8.9735E-05	1.0250E-04

5. Numerical results and discussion

In this section, we present the numerical solution for time-fractional convection-diffusion equations in both one- and two-dimensional cases, using the proposed method. The accuracy and convergence of the method were evaluated by utilizing the error norms outlined below.

$$L_\infty = max|\, E_k - w_k \,|_{k=1}^{M+1},$$

$$L_2 = \left(h \sum_{k=1}^{M+1} \left(E_k - w_k\right)^2 \right)^{1/2},$$

$$L_{rms} = \left(1/M \sum_{k=1}^{M+1} \left(E_k - w_k\right)^2 \right)^{1/2},$$

where E is the exact solution.

Problem 1. The following form is used to express the time-fractional convection-diffusion equation.

$$\frac{\partial^\alpha w(s,t)}{\partial t^\alpha} + \frac{\partial w(s,t)}{\partial s} - \frac{\partial^2 w(s,t)}{\partial s^2} = g(s,t), \quad s \in [0,1], \quad t > 0, \tag{36}$$

where

$$g(s,t) = \frac{\Gamma(4+\alpha)}{6} t^3 \sin(\pi s) + \pi^2 t^{3+\alpha} \sin(\pi s) + \pi t^{3+\alpha} \cos(\pi s),$$

and the exact solution is $w(s,t) = t^{3+\alpha} \sin(\pi s)$.

The recommended approach on Test Problem 1 is employed to produce numerical outcomes, presented in Table 1 in terms of absolute error. Diverse values of α were applied to obtain these results whereas final time $T = 0.1$ and time step size $\tau = 0.001$. A comparison of the suggested approach with [87] indicates that the proposed technique is more accurate than the ones published in the latest literature. The suggested approach is employed to compute numerical results for various values of M and α whereas final time $T = 0.1$ and time step size $\tau = 0.001$ in Table 2, using L_∞, L_2 and L_{rms} error measures and the CPU times in seconds. Furthermore, the calculation presented in Table 2 indicates that the spectral radius is approximately one, demonstrating the fulfillment of the stability condition. The results show that increasing the number of collocation points leads to improved accuracy with the current approach. In addition, the proposed technique is demonstrated to be efficient through a comparison between exact and approximate solutions and their corresponding absolute errors, as shown in Fig. 1.

Problem 2. The following form is used to express time-fractional convection-diffusion equation.

$$\frac{\partial^\alpha w(s,t)}{\partial t^\alpha} + s\frac{\partial w(s,t)}{\partial s} - \frac{\partial^2 w(s,t)}{\partial s^2} = g(s,t), \quad s \in [0,1], \quad t > 0, \tag{37}$$

where

$$g(s,t) = \frac{\Gamma(1+2\alpha)}{\Gamma(1+\alpha)} t^\alpha (s - s^3) + (1 + t^{2\alpha})(7s - 3s^3),$$

and the exact solution is $w(s,t) = (1 + t^{2\alpha})(s - s^3)$.

To obtain numerical results for Test Problem 2, we utilized the methodology suggested in this paper. The obtained outcomes are presented in Table 3 as the absolute error, where various values of α were considered whereas final time $T = 0.1$ and time step size $\tau = 0.001$. Our proposed approach was compared to the method proposed in [87], and the results show that our technique outperforms the method published in the latest literature in terms of accuracy. To calculate the numerical results for various values of M and α in Table 4, we utilized the suggested approach and evaluated the accuracy using L_∞, L_2 and L_{rms} error measures, as well as CPU times in seconds. The final time $T = 1$ and time step size $\tau = 0.001$. The obtained results indicate that increasing the number of collocation points can enhance the accuracy of the current approach. Furthermore, Fig. 2 illustrates a comparison between the exact and approximate solutions, and their corresponding absolute errors at various time instances, demonstrating the efficiency of the proposed technique.

Problem 3. The following form is used to express time-fractional convection-diffusion equation.

$$\frac{\partial^\alpha w(s,t)}{\partial t^\alpha} + s\frac{\partial w(s,t)}{\partial s} - \frac{\partial^2 w(s,t)}{\partial s^2} = g(s,t), \quad s \in [0,1], \quad t > 0, \tag{38}$$

where

$$g(s,t) = \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)}(s^2 - s^3) + (1 + t^2)(2s^2 - 3s^3 + 6s - 2),$$

and the exact solution is $w(s,t) = (1 + t^2)(s^2 - s^3)$.

Similar to the previous two problems, we employed the proposed technique for Test Problem 3. The results, which consist of the absolute error for various values of α, $T = 0.1$ and $\tau = 0.001$ are reported in Table 5. We compared our approach with the method proposed by [87] and found that our method achieves greater accuracy. To obtain the numerical outcomes presented in Table 6 for different values of M and α, $T = 1$ and $\tau = 0.01$, we applied the recommended method and assessed its precision using various error measures, including L_∞, L_2, and L_{rms}, along with CPU times measured in seconds. The outcomes reveal that increasing the number of collocation points can lead to more accurate results using the proposed method. Additionally, the effectiveness of the proposed technique is demonstrated in Fig. 3, which depicts a comparison of the exact and approximate solutions, and their respective absolute errors at different time.

Problem 4. The following form is used to express time-fractional convection-diffusion equation.

$$\frac{\partial^\alpha w(s,r,t)}{\partial t^\alpha} + 0.001\frac{\partial w(s,r,t)}{\partial s} - \left(\frac{\partial^2 w(s,r,t)}{\partial s^2} + \frac{\partial^2 w(s,r,t)}{\partial r^2} \right) = g(s,r,t),$$
$$(s,r) \in [0,1], \quad t > 0, \tag{39}$$

where

$$g(s,r,t) = \frac{t^{1-\alpha}}{\Gamma(2-\alpha)} sr(s-1)(r-1)$$
$$- 2tr(r-1) - 2ts(r-1) + 0.001t(2s-1)r(r-1),$$

Table 2
The outcomes obtained from utilizing the suggested approach to address Problem 1.

α	M	L_∞	L_2	L_{rms}	CPU time	$\rho(\digamma)$
0.3	5	2.2757E-06	1.3955E-06	3.1049E-07	0.104730	0.9998
	10	3.7838E-06	2.6870E-06	8.4548E-07	0.127747	0.9998
	15	3.8115E-06	2.6921E-06	1.0375E-06	0.158500	0.9999
	20	3.8027E-06	2.6891E-06	1.1966E-06	0.245428	1.0000
	25	3.8017E-06	2.6868E-06	1.3367E-06	1.487935	1.0002
0.5	5	6.6391E-06	4.6075E-06	1.0251E-06	0.094082	0.9998
	10	7.4726E-06	5.3041E-06	1.6690E-06	0.128429	0.9998
	15	7.5029E-06	5.3005E-06	2.0427E-06	0.146154	0.9999
	20	7.5063E-06	5.3116E-06	2.3636E-06	0.251146	1.0000
	25	7.5048E-06	5.3042E-06	2.6389E-06	1.460811	1.0002
0.7	5	8.8345E-06	6.2931E-06	1.4002E-06	0.106572	0.9998
	10	9.3094E-06	6.6026E-06	2.0776E-06	0.132660	0.9998
	15	9.3414E-06	6.6012E-06	2.5439E-06	0.154339	0.9999
	20	9.3251E-06	6.6056E-06	2.9394E-06	0.240029	1.0000
	25	9.3157E-06	6.5736E-06	3.2705E-06	1.541608	1.0002

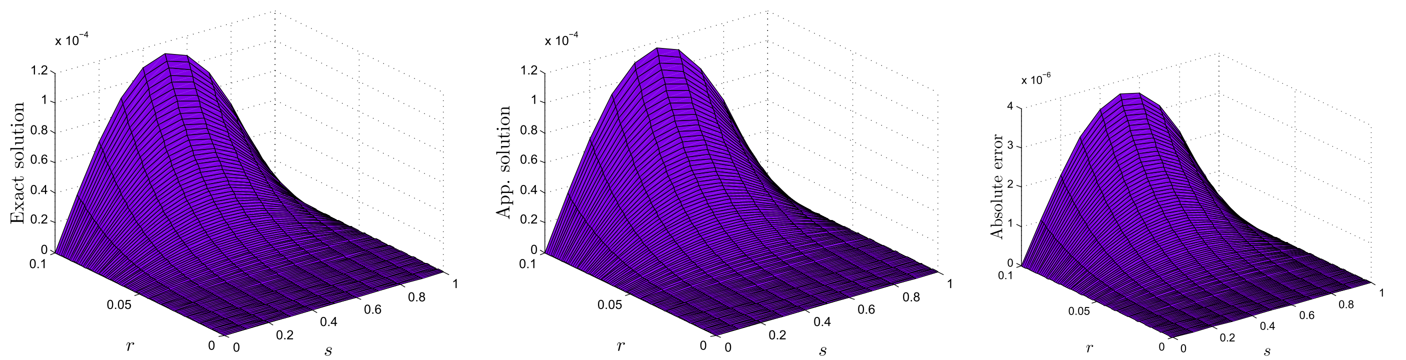

Fig. 1. Exact and numerical solutions, and absolute error at $t = 0.1$ from left to right respectively, for Test Problem 1.

Table 3
The outcomes obtained from utilizing the suggested approach to address Problem 2.

	Current scheme			Cited work [87]		
s	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 0.95$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 0.95$
0.1	2.4404E-05	1.0308E-05	4.7213E-06	3.0250E-03	2.4473E-03	2.3521E-03
0.2	9.4312E-05	3.9798E-05	1.8222E-05	5.8222E-03	4.7146E-03	4.5138E-03
0.3	1.9477E-04	8.2331E-05	3.7660E-05	8.1614E-03	6.6114E-03	6.3227E-03
0.4	2.9320E-04	1.2442E-04	5.6773E-05	9.8394E-03	7.9728E-03	7.6213E-03
0.5	3.4588E-04	1.4768E-04	6.7024E-05	1.0675E-02	8.6566E-03	8.2740E-03
0.6	3.2244E-04	1.3865E-04	6.2283E-05	1.0492E-02	8.5537E-03	8.1765E-03
0.7	2.3108E-04	9.9927E-05	4.4099E-05	9.3727E-03	7.5997E-03	7.2674E-03
0.8	1.1705E-04	5.0697E-05	2.1809E-05	7.1396E-03	5.7900E-03	5.5422E-03
0.9	3.0835E-05	1.3328E-05	5.5810E-06	3.9436E-03	3.1971E-03	3.0699E-03

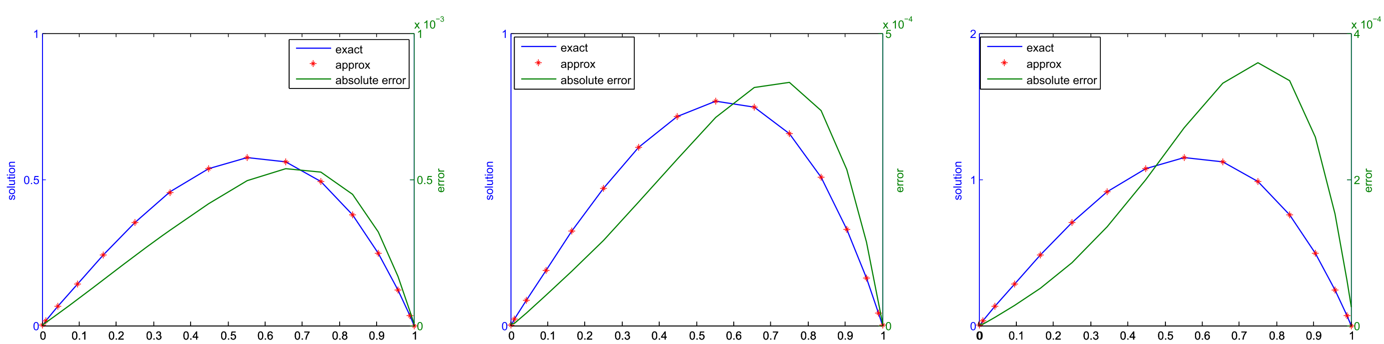

Fig. 2. Exact versus numerical results and absolute error at $t = 0.5$, $t = 1$ and $t = 2$ from left to right respectively, for Test Problem 2.

Table 4
The outcomes obtained from utilizing the suggested approach to address Problem 2.

α	M	L_∞	L_2	L_{rms}	CPU time
0.3	5	6.492E-04	3.954E-04	8.797E-05	0.11023
	10	6.492E-04	3.953E-04	1.244E-04	0.13906
	15	6.476E-04	3.941E-04	1.519E-04	0.17176
	20	6.525E-04	3.954E-04	1.759E-04	0.23261
	25	6.542E-04	3.945E-04	4.618E-04	1.42035
0.5	5	4.100E-04	2.407E-04	5.355E-05	0.09487
	10	4.098E-04	2.405E-04	7.568E-05	0.14267
	15	4.245E-04	2.444E-04	9.420E-05	0.15270
	20	4.222E-04	2.410E-04	1.072E-04	0.22670
	25	4.226E-04	2.417E-04	1.202E-04	1.54943
0.7	5	4.798E-04	2.651E-04	5.899E-05	0.09415
	10	4.798E-04	2.651E-04	8.342E-05	0.14148
	15	5.128E-04	2.762E-04	1.065E-04	0.14236
	20	5.001E-04	2.668E-04	1.187E-04	0.24842
	25	4.963E-04	2.644E-04	1.315E-04	1.51036

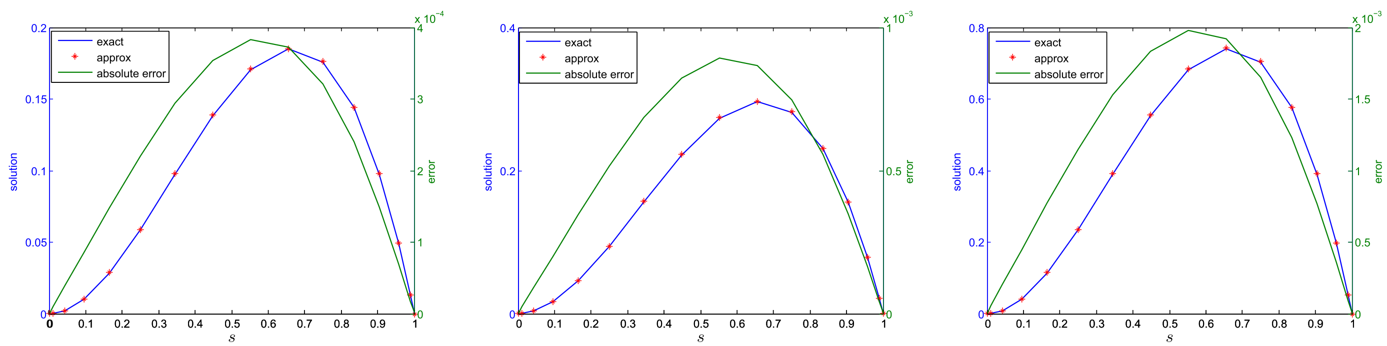

Fig. 3. Exact versus numerical results and absolute error at $t = 0.5$, $t = 1$ and $t = 2$ from left to right respectively, for Test Problem 3.

Table 5
The outcomes obtained from utilizing the suggested approach to address Problem 3.

	Current scheme			Cited work [87]
s	$\alpha = 0.2$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.5$
0.1	1.9632E-07	1.2461E-06	2.4526E-06	1.3845E-04
0.2	7.6720E-07	4.8810E-06	9.6389E-06	1.5784E-04
0.3	1.6460E-06	1.0546E-05	2.1022E-05	1.3859E-04
0.4	2.6394E-06	1.7093E-05	3.4551E-05	9.3740E-05
0.5	3.3525E-06	2.1962E-05	4.5039E-05	3.5000E-05
0.6	3.3536E-06	2.2188E-05	4.6059E-05	2.5614E-05
0.7	2.5507E-06	1.6996E-05	3.5585E-05	7.4063E-05
0.8	1.3523E-06	9.0459E-06	1.9028E-05	9.2366E-05
0.9	3.6701E-07	2.4579E-06	5.1779E-06	5.6406E-05

and the exact solution is $w(s,r,t) = tsr(s-1)(r-1)$.

Table 7 presents the results of the suggested method for solving Test Problem 4 in two dimensions, and is compared with the method presented in [59], whereas the value of $M = 10$, $\alpha = 0.3$, $T = 01$ and $\tau = 0.001$. The table clearly shows that the suggested method achieves good accuracy even with a small number of nodes. This comparison indicates that the proposed approach outperforms the method presented in [59] in terms of accuracy.
Fig. 4 illustrates the approximate solution obtained using the suggested method, and compares it with the exact solution. The absolute errors between the approximate and exact solutions are also shown in the figure. It is evident that the numerical solution obtained using the suggested method is in excellent agreement with the exact solution.

Problem 5. The following form is used to express time-fractional convection-diffusion equation.

Table 6
The outcomes obtained from utilizing the suggested approach to address Problem 3.

α	M	L_∞	L_2	L_{rms}	CPU time
0.3	5	3.956E-04	2.402E-04	5.344E-05	1.777E-01
	10	3.968E-04	2.402E-04	7.559E-05	1.249E-01
	15	4.083E-04	2.410E-04	9.286E-05	1.763E-01
	20	4.089E-04	2.403E-04	1.069E-04	2.514E-01
	25	4.088E-04	2.411E-04	1.199E-04	1.551E+00
0.6	5	1.129E-03	6.850E-04	1.524E-04	1.248E-01
	10	1.132E-03	6.851E-04	2.156E-04	1.284E-01
	15	1.172E-03	6.924E-04	2.668E-04	1.530E-01
	20	1.165E-03	6.842E-04	3.045E-04	2.608E-01
	25	1.166E-03	6.870E-04	3.418E-04	1.495E+00
0.9	5	1.101E-03	6.682E-04	1.487E-04	1.229E-01
	10	1.104E-03	6.683E-04	2.103E-04	1.268E-01
	15	1.161E-03	6.861E-04	2.644E-04	1.674E-01
	20	1.140E-03	6.698E-04	2.980E-04	2.526E-01
	25	1.132E-03	6.660E-04	3.313E-04	1.479E+00

$$\frac{\partial^\alpha w(s,r,t)}{\partial t^\alpha} + \frac{\partial w(s,r,t)}{\partial s} - \left(\frac{\partial^2 w(s,r,t)}{\partial s^2} + \frac{\partial^2 w(s,r,t)}{\partial r^2}\right) = g(s,r,t),$$
$$(s,r) \in [0,1], \quad t > 0, \tag{40}$$

where

$$g(s,r,t) = \frac{t^{1-\alpha}}{\Gamma(2-\alpha)}\sin(\pi s)\sin(\pi r) + t\left(2\pi^2\sin(\pi s)\sin(\pi r) + \pi\cos(\pi s)\sin(\pi r)\right),$$

and the exact solution is $w(s,r,t) = t\sin(\pi s)\sin(\pi r)$.

Table 8 displays the results of using the proposed method to solve Test Problem 5, with respect to L_∞ and L_2 error norms, for differ-

Table 7
The outcomes obtained from utilizing the suggested approach to address Problem 4.

		Current scheme			Cited work [59]	
s	r	Exact	App.	Absolute error	App.	Absolute error
0.1	0.1	0.0081	0.0081	1.50E-06	0.0080	6.77E-05
0.2	0.2	0.0256	0.0256	5.32E-06	0.0255	9.65E-05
0.3	0.3	0.0441	0.0441	9.88E-06	0.0440	6.93E-05
0.4	0.4	0.0576	0.0576	1.34E-05	0.0575	8.43E-05
0.5	0.5	0.0625	0.0625	1.47E-05	0.0624	9.08E-05
0.6	0.6	0.0576	0.0576	1.32E-05	0.0575	8.43E-05
0.7	0.7	0.0441	0.0441	9.29E-06	0.0440	6.63E-05
0.8	0.8	0.0256	0.0256	4.26E-06	0.0256	4.12E-05
0.9	0.9	0.0081	0.0081	3.44E-07	0.0080	9.60E-05

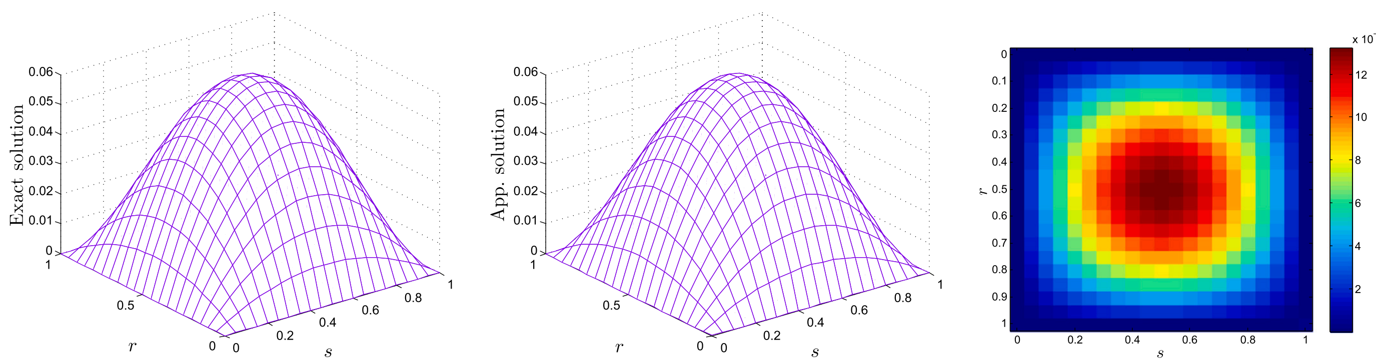

Fig. 4. Exact and numerical solutions, and absolute error at $t = 1$ from left to right respectively, for Test Problem 4.

Table 8
The outcomes obtained from utilizing the suggested approach to address Problem 5.

α	M	L_∞	L_2	CPU time
0.25	4	5.65E-03	4.05E-03	3.1316
	8	1.59E-04	7.84E-05	4.7225
	12	1.60E-04	7.93E-05	8.6797
	16	1.62E-04	8.10E-05	16.9765
	20	1.62E-04	8.34E-05	38.2507
0.75	4	4.79E-03	3.37E-03	3.1740
	8	1.63E-03	8.18E-04	4.7656
	12	1.60E-03	8.05E-04	8.6114
	16	1.59E-03	7.96E-04	16.9824
	20	1.51E-03	5.85E-04	38.9179

Table 9
The outcomes obtained from utilizing the suggested approach to address Problem 5.

	T = 0.5		T = 1	
α	L_∞	L_2	L_∞	L_2
0.2	1.04E-04	5.22E-05	1.11E-04	5.56E-05
0.4	3.97E-04	1.99E-04	4.12E-04	2.06E-04
0.6	1.06E-03	5.34E-04	1.09E-03	5.45E-04
0.8	2.66E-03	1.33E-04	2.74E-03	1.38E-03

ent numbers of nodes whereas $T = 01$ and $\tau = 0.001$. The table also includes the computation time. The results demonstrate that the suggested method is both accurate and efficient, even when using a small number of nodes. Additionally, Table 9 presents the numerical results for various fractional orders and time instances whereas $M = 16$ and $\tau = 0.001$, indicating that the suggested method achieves better accuracy.
Fig. 5 depicts the approximate solution obtained by the suggested method, and compares it with the exact solution. The figure also shows the absolute errors between the approximate and exact solutions. The results clearly demonstrate that the numerical solution produced by the proposed method agrees excellently with the exact solution.

6. Conclusion

This study presents an efficient technique for solving one- and two-dimensional time-fractional convection-diffusion equations using the Caputo derivative. The method combines Lucas and Fibonacci polynomials with the Caputo derivative definition to discretize the problem in temporal direction and approximate the function's derivative. To evaluate the accuracy of the suggested approach, three different error norms were computed and compared with exact solutions and previously published methods. The results demonstrate that the hybrid method achieves higher accuracy, while the employed polynomials enable convenient approximation of higher-order derivatives for unknown functions. Additionally, the study highlights the effectiveness of reducing spatial and time steps in minimizing errors. The proposed technique can be adapted for various complex fractional partial differential equations with minor modifications.

Ethical approval

All the authors demonstrating that they have adhered to the accepted ethical standards of a genuine research study.

Consent to participate

Being the corresponding author, I have consent to participate of all the authors in this research work.

Consent to publish

All the authors are agreed to publish this research work.

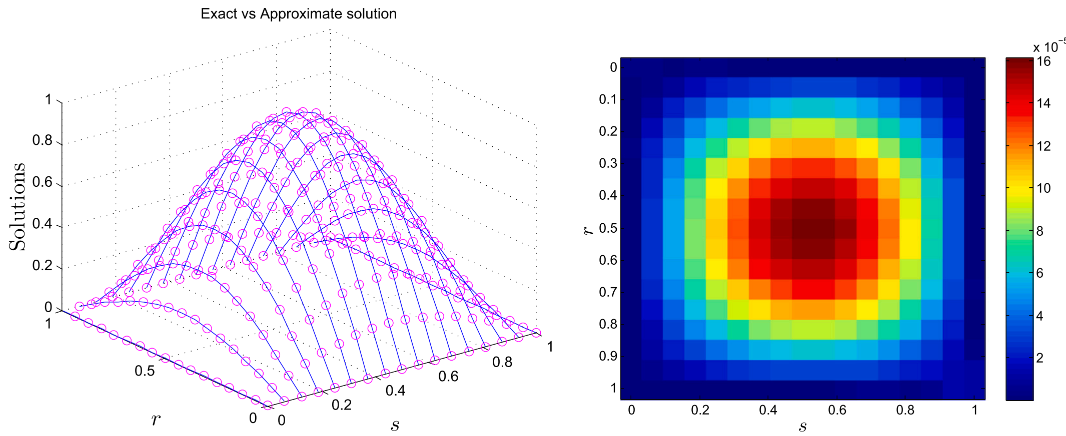

Fig. 5. Exact versus numerical solutions and absolute error at $t = 1$ from left to right respectively, for Test Problem 5.

Declaration of competing interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Data availability

Data will be provided on request to the corresponding author.

Acknowledgement and Funding

This work was supported by Tenaga Nasional Berhad (TNB) and UNITEN through the BOLD Refresh Publication Fund under the project code of J510050002-IC-6 BOLDREFRESH2025-Centre of Excellence. Also, this work was supported by Tenaga Nasional Berhad (TNB) and UNITEN through the BOLD Refresh Postdoctoral Fellowships under the project code of J510050002-IC-6 BOLDREFRESH2025-Centre of Excellence.

References

[1] A. Atangana, On the new fractional derivative and application to nonlinear Fisher's reaction–diffusion equation, Appl. Math. Comput. 273 (2016) 948–956.
[2] J. Singh, A. Gupta, D. Baleanu, Fractional dynamics and analysis of coupled Schrödinger-kdv equation with Caputo-Katugampola type memory, J. Comput. Nonlinear Dynam. 18 (9) (2023).
[3] A. Atangana, Non validity of index law in fractional calculus: a fractional differential operator with Markovian and non-Markovian properties, Phys. A, Stat. Mech. Appl. 505 (2018) 688–706.
[4] A. Atangana, B.S.T. Alkahtani, Analysis of the Keller–Segel model with a fractional derivative without singular kernel, Entropy 17 (6) (2015) 4439–4453.
[5] D. Baleanu, Z.B. Güvenç, J.T. Machado, et al., New Trends in Nanotechnology and Fractional Calculus Applications, vol. 10, Springer, 2010.
[6] S. Priyadarshana, J. Mohapatra, S. Pattanaik, A second order fractional step hybrid numerical algorithm for time delayed singularly perturbed 2d convection-diffusion problems, Appl. Numer. Math. 189 (2023) 107–129.
[7] P. Sawangtong, K. Logeswari, C. Ravichandran, K.S. Nisar, V. Vijayaraj, Fractional order geminivirus impression in capsicum annuum model with Mittag-Leffler kernal, Fractals (2023) 2340049.
[8] K.S. Nisar, R. Jagatheeshwari, C. Ravichandran, P. Veeresha, High performance computational method for fractional model of solid tumour invasion, Ain Shams Eng. J. (2023) 102226.
[9] C. Ravichandran, K. Jothimani, K.S. Nisar, E.E. Mahmoud, I.S. Yahia, An interpretation on controllability of Hilfer fractional derivative with nondense domain, Alex. Eng. J. 61 (12) (2022) 9941–9948.
[10] B. Ghanbari, A fractional system of delay differential equation with nonsingular kernels in modeling hand-foot-mouth disease, Adv. Differ. Equ. 2020 (1) (2020) 536.
[11] B. Ghanbari, On novel nondifferentiable exact solutions to local fractional Gardner's equation using an effective technique, Math. Methods Appl. Sci. 44 (6) (2021) 4673–4685.
[12] B. Ghanbari, On approximate solutions for a fractional prey–predator model involving the Atangana–Baleanu derivative, Adv. Differ. Equ. 2020 (1) (2020) 679.
[13] B. Ghanbari, A. Atangana, Some new edge detecting techniques based on fractional derivatives with non-local and non-singular kernels, Adv. Differ. Equ. 2020 (1) (2020) 1–19.
[14] B. Carreras, V. Lynch, G. Zaslavsky, Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence model, Phys. Plasmas 8 (12) (2001) 5096–5103.
[15] Siraj-ul-Islam, I. Ahmad, A comparative analysis of local meshless formulation for multi-asset option models, Eng. Anal. Bound. Elem. 65 (2016) 159–176.
[16] D. McLean, Understanding Aerodynamics: Arguing from the Real Physics, John Wiley & Sons, 2012.
[17] J.M. Burgers, A mathematical model illustrating the theory of turbulence, Adv. Appl. Mech. 1 (1948) 171–199.
[18] H.K. Jalghaf, E. Kovács, B. Bolló, Comparison of old and new stable explicit methods for heat conduction, convection, and radiation in an insulated wall with thermal bridging, Buildings 12 (9) (2022) 1365.
[19] D.-S. Li, Convection-diffusion modelling for chemical pollutant dispersion in the joint of artificial lake using finite element method, Bulg. Chem. Commun. 47 (2015) 949–958.
[20] D. Parra-Guevara, Y.N. Skiba, Industrial pollution transport. part 2. control of industrial emissions, Environ. Model. Assess. 5 (3) (2000) 177–184.
[21] H. Khan, S. Mustafa, I. Ali, P. Kumam, D. Baleanu, M. Arif, Approximate analytical fractional view of convection–diffusion equations, Open Phys. 18 (1) (2020) 897–905.
[22] S. Sankaranarayanan, N. Shankar, H. Cheong, Three-dimensional finite difference model for transport of conservative pollutants, Ocean Eng. 25 (6) (1998) 425–442.
[23] J. Sierra, M. Mestres, A. Rodriguez, A. Arcilla, A lagrangian 3D numerical model of pollutant dispersion in coastal waters, WIT Transactions on Ecology and the Environment 25 (1970).
[24] L. Li, Z. Yin, Numerical simulation of groundwater pollution problems based on convection diffusion equation, Am. J. Comput. Math. 7 (3) (2017) 350–370.
[25] A. Siripong, Hydrodynamic and oil-spill modelling for the East Asian seas region, Ambio (1988) 183–185.
[26] S.-D. Wang, Y.-M. Shen, Y.-K. Guo, J. Tang, Three-dimensional numerical simulation for transport of oil spills in seas, Ocean Eng. 35 (5–6) (2008) 503–510.
[27] A. Kennedy, W.J. O'Connor, A transmission line modelling (tlm) method for steady-state convection–diffusion, Int. J. Numer. Methods Eng. 72 (9) (2007) 1009–1028.
[28] O.M. Bamigbola, M.M. Ali, K.O. Awodele, et al., Predictive models of current, voltage, and power losses on electric transmission lines, J. Appl. Math. (2014).
[29] T. Lan, K. Strunz, Modeling of the enthalpy transfer using electric circuit equivalents: theory and application to transients of multi-carrier energy systems, IEEE Trans. Energy Convers. 34 (4) (2019) 1720–1730.
[30] C.-Z. Fu, W.-R. Si, L. Quan, J. Yang, Numerical study of convection and radiation heat transfer in pipe cable, Math. Probl. Eng. 2018 (2018) 1–12.
[31] C.-Z. Fu, W.-R. Si, D. Yang, J. Yang, Numerical study of heat transfer and optimum design for trench laying cables with ceramic plates, Math. Probl. Eng. 2021 (2021) 1–13.
[32] D. Enescu, P. Colella, A. Russo, Thermal assessment of power cables and impacts on cable current rating: an overview, Energies 13 (20) (2020) 5319.
[33] A.S.A. Heryudono, E. Larsson, A radial basis function partition of unity collocation method for convection-diffusion equations arising in financial applications, J. Sci. Comput. 64 (2015) 341–367.
[34] Z.-Q. Deng, V.P. Singh, L. Bengtsson, Numerical solution of fractional advection-dispersion equation, J. Hydraul. Eng. 130 (5) (2004) 422–431.
[35] M. Srivastava, H. Ahmad, I. Ahmad, P. Thounthong, N. Khan, Numerical simulation of three-dimensional fractional-order convection-diffusion PDEs by a local meshless method, Therm. Sci. 25 (2020) 347–358.
[36] X.-B. Shu, Q. Wang, The existence and uniqueness of mild solutions for fractional differential equations with nonlocal conditions of order $1 < \alpha < 2$, Comput. Math. Appl. 64 (6) (2012) 2100–2110.

[37] B. Jin, N. Su, Existence and uniqueness of solutions of reaction–convection equations with non-Lipschitz nonlinearity, J. Math. Anal. Appl. 410 (1) (2014) 267–274.
[38] Y. Gambo, R. Ameen, F. Jarad, T. Abdeljawad, Existence and uniqueness of solutions to fractional differential equations in the frame of generalized Caputo fractional derivatives, Adv. Differ. Equ. 2018 (1) (2018) 1–13.
[39] A. Kilbas, Theory and applications of fractional differential equations, 2006.
[40] B. Ghosh, J. Mohapatra, A novel numerical technique for solving time fractional nonlinear diffusion equations involving weak singularities, Math. Methods Appl. Sci. (2023).
[41] N. Sene, K. Abdelmalek, Analysis of the fractional diffusion equations described by Atangana-Baleanu-Caputo fractional derivative, Chaos Solitons Fractals 127 (2019) 158–164.
[42] C. Ravichandran, K. Logeswari, A. Khan, T. Abdeljawad, J. Gómez-Aguilar, An epidemiological model for computer virus with Atangana-Baleanu fractional derivative, Results Phys. 51 (2023) 106601.
[43] B. Ghanbari, On approximate solutions for a fractional prey–predator model involving the Atangana–Baleanu derivative, Adv. Differ. Equ. 2020 (1) (2020) 679.
[44] B. Ghanbari, On the modeling of the interaction between tumor growth and the immune system using some new fractional and fractional-fractal operators, Adv. Differ. Equ. 2020 (1) (2020) 1–32.
[45] B. Ghanbari, A new model for investigating the transmission of infectious diseases in a prey-predator system using a non-singular fractional derivative, Math. Methods Appl. Sci. 46 (7) (2023) 8106–8125.
[46] V.R. Hosseini, M. Koushki, W.-N. Zou, The meshless approach for solving 2D variable-order time-fractional advection–diffusion equation arising in anomalous transport, Eng. Comput. (2021) 1–19.
[47] A. Zhokh, P. Strizhak, Advection-diffusion in a porous medium with fractal geometry: fractional transport and crossovers on time scales, Meccanica 57 (4) (2022) 833–843.
[48] A. Jannelli, Numerical solutions of fractional differential equations arising in engineering sciences, Mathematics 8 (2) (2020) 215.
[49] C. Ravichandran, K. Munusamy, K.S. Nisar, N. Valliammal, Results on neutral partial integrodifferential equations using monch-Krasnosel'skii fixed point theorem with nonlocal conditions, Fractal Fract. 6 (2) (2022) 75.
[50] E.F.D. Goufo, C. Ravichandran, G.A. Birajdar, Self-similarity techniques for chaotic attractors with many scrolls using step series switching, Math. Model. Anal. 26 (4) (2021) 591–611.
[51] K.S. Nisar, K. Munusamy, C. Ravichandran, Results on existence of solutions in nonlocal partial functional integrodifferential equations with finite delay in nondense domain, Alex. Eng. J. 73 (2023) 377–384.
[52] N. Sene, Analytical solutions and numerical schemes of certain generalized fractional diffusion models, Eur. Phys. J. Plus 134 (5) (2019) 199.
[53] B. Ghosh, J. Mohapatra, Analysis of finite difference schemes for Volterra integro-differential equations involving arbitrary order derivatives, J. Appl. Math. Comput. 69 (2) (2023) 1865–1886.
[54] A. Panda, J. Mohapatra, A robust finite difference method for the solutions of singularly perturbed Fredholm integro-differential equations, Mediterr. J. Math. 20 (4) (2023) 198.
[55] F.M. Salama, N.H.M. Ali, N.N. Abd Hamid, Fast O(N) hybrid Laplace transform-finite difference method in solving 2D time fractional diffusion equation, J. Math. Comput. Sci. 23 (2021) 110–123.
[56] C. Gong, W. Bao, G. Tang, Y. Jiang, J. Liu, A domain decomposition method for time fractional reaction-diffusion equation, Sci. World J. (2014).
[57] X. Li, M. Xu, X. Jiang, Homotopy perturbation method to time-fractional diffusion equation with a moving boundary condition, Appl. Math. Comput. 208 (2) (2009) 434–439.
[58] G. Dong, Z. Guo, W. Yao, Numerical methods for time-fractional convection-diffusion problems with high-order accuracy, Open Math. 19 (1) (2021) 782–802.
[59] L. Li, Z. Jiang, Z. Yin, Compact finite-difference method for 2D time-fractional convection–diffusion equation of groundwater pollution problems, Comput. Appl. Math. 39 (3) (2020) 142.
[60] J. Zhang, X. Zhang, B. Yang, An approximation scheme for the time fractional convection–diffusion equation, Appl. Math. Comput. 335 (2018) 305–312.
[61] J. Wang, T. Liu, H. Li, Y. Liu, S. He, Second-order approximation scheme combined with H1-Galerkin MFE method for nonlinear time fractional convection–diffusion equation, Comput. Math. Appl. 73 (6) (2017) 1182–1196.
[62] M. Dehghan, M. Abbaszadeh, A. Mohebbi, Legendre spectral element method for solving time fractional modified anomalous sub-diffusion equation, Appl. Math. Model. 40 (5–6) (2016) 3635–3654.
[63] H. Qiao, A. Cheng, A fast finite difference/RBF meshless approach for time fractional convection-diffusion equation with non-smooth solution, Eng. Anal. Bound. Elem. 125 (2021) 280–289.
[64] W.M. Abd-Elhameed, Y.H. Youssri, N. El-Sissi, M. Sadek, New hypergeometric connection formulae between Fibonacci and Chebyshev polynomials, Ramanujan J. 42 (2) (2017) 347–361.
[65] W. Abd-Elhameed, Y. Youssri, 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) (2020) 1–19.
[66] M. Nadir, Lucas polynomials for solving linear integral equations, J. Theor. Appl. Comput. Sci. 11 (1) (2017) 13–19.
[67] M. Çetin, M. Sezer, C. Güler, Lucas polynomial approach for system of high-order linear differential equations and residual error estimation, Math. Probl. Eng. (2015).
[68] F. Mirzaee, S.F. Hoseini, Application of Fibonacci collocation method for solving Volterra–Fredholm integral equations, Appl. Math. Comput. 273 (2016) 637–644.
[69] N. Baykuş-Savaşaneril, M. Sezer, Hybrid Taylor-Lucas collocation method for numerical solution of high-order Pantograph type delay differential equations with variables delays, Appl. Math. Inf. Sci. 11 (6) (2017) 1795–1801.
[70] Ö. Oruç, 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) (2017) 3042–3057.
[71] Ö. Oruç, A new numerical treatment based on Lucas polynomials for 1D and 2D sinh-Gordon equation, Commun. Nonlinear Sci. Numer. Simul. 57 (2018) 14–25.
[72] I. Ali, S. Haq, K.S. Nisar, D. Baleanu, An efficient numerical scheme based on Lucas polynomials for the study of multidimensional Burgers-type equations, Adv. Differ. Equ. 2021 (1) (2021) 1–24.
[73] S. Haq, I. Ali, Approximate solution of two-dimensional Sobolev equation using a mixed Lucas and Fibonacci polynomials, Eng. Comput. (2021) 1–10.
[74] S. Haq, I. Ali, K.S. Nisar, A computational study of two-dimensional reaction-diffusion Brusselator system with applications in chemical processes, Alex. Eng. J. 60 (5) (2021) 4381–4392.
[75] G. Jumarie, Stock exchange fractional dynamics defined as fractional exponential growth driven by (usual) Gaussian white noise. application to fractional Black-Scholes equations, Insur. Math. Econ. 42 (1) (2008) 271–287.
[76] G. Jumarie, Derivation and solutions of some fractional Black-Scholes equations in coarse-grained space and time. application to Merton's optimal portfolio, Comput. Math. Appl. 59 (3) (2010) 1142–1164.
[77] M. Caputo, Linear models of dissipation whose Q is almost frequency independent-II, Geophys. J. Int. 13 (5) (1967) 529–539.
[78] A. Atangana, D. Baleanu, New fractional derivatives with non-local and nonsingular kernel theory and application to heat transfer model, Therm. Sci. 20 (2016) 763.
[79] J.-H. He, A new fractal derivation, Therm. Sci. 15 (suppl. 1) (2011) 145–147.
[80] M. Hussain, S. Haq, A. Ghafoor, I. Ali, Numerical solutions of time-fractional coupled viscous Burgers' equations using meshfree spectral method, Comput. Appl. Math. 39 (1) (2020) 6.
[81] I. Ali, S. Haq, S.F. Aldosary, K.S. Nisar, F. Ahmad, Numerical solution of one-and two-dimensional time-fractional Burgers equation via Lucas polynomials coupled with finite difference method, Alex. Eng. J. 61 (8) (2022) 6077–6087.
[82] S. Gumgum, N.B. Savasaneril, O. Kurkcu, M. Sezer, Lucas polynomial solution for neutral differential equations with proportional delays, TWMS J. Appl. Eng. Math. 10 (1) (2020) 259–269.
[83] I. Ali, S. Haq, K.S. Nisar, S.U. Arifeen, Numerical study of 1d and 2d advection-diffusion-reaction equations using Lucas and Fibonacci polynomials, Arab. J. Math. 10 (3) (2021) 513–526.
[84] G. Garmanjani, R. Cavoretto, M. Esmaeilbeigi, A rbf partition of unity collocation method based on finite difference for initial–boundary value problems, Comput. Math. Appl. 75 (11) (2018) 4066–4090.
[85] M. Hussain, S. Haq, A. Ghafoor, Meshless rbfs method for numerical solutions of two-dimensional high order fractional Sobolev equations, Comput. Math. Appl. 79 (3) (2020) 802–816.
[86] P.D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Commun. Pure Appl. Math. 7 (1) (1954) 159–193.
[87] V. Saw, S. Kumar, The Chebyshev collocation method for a class of time fractional convection-diffusion equation with variable coefficients, Math. Methods Appl. Sci. 44 (8) (2021) 6666–6678.