mathematics

MDPI

Article

Pell–Lucas Collocation Method for Time-Fractional Partial Integro-Differential Equations

Javad Damirchi [1], Ahmadreza Yazdanian [2], Shohreh Mehranpour [1] and Mahdi Mahmoudi [3,*]

[1] Department of Mathematics, Faculty of Mathematics, Statistics and Computer Science, Semnan University, Semnan 35196-45399, Iran
[2] Faculty of Finance Sciences, Kharazmi University, Tehran 15719-14911, Iran
[3] Faculty of Medicine, Sigmund Freud University, 1020 Vienna, Austria
* Correspondence: mahdi.mahmoudi@med.sfu.ac.at

Abstract

This paper presents a Pell–Lucas collocation method for time-fractional partial integro-differential equations (TFPIDEs) with initial and boundary conditions, where the fractional derivative is taken in the Caputo sense. The unknown solution is approximated by a truncated bivariate Pell–Lucas series, and the governing equation together with the initial and boundary conditions is discretized at uniform collocation points. Operational matrices are employed to represent the fractional-time derivative, spatial derivatives, and Volterra integral term, thereby reducing the continuous problem to a system of algebraic equations. An upper error bound in the infinity norm and a residual-based error-estimation procedure are also discussed. Numerical examples illustrate the applicability and accuracy of the proposed approach and include comparisons with exact solutions and with results reported in the literature for alternative methods.

Keywords: Pell–Lucas polynomials; time-fractional partial integro-differential equation; Caputo fractional derivative; collocation method; error estimation

MSC: 65N35; 45K05

1. Introduction

Over the past decade, partial integro-differential equations (PIDEs) have attracted increasing attention in applied sciences, including mathematics, engineering, physics, electromagnetic theory, geophysics, heat conduction, radiation propagation, poro-viscoelastic medium compression, reaction–diffusion problems, and nuclear reactor dynamics [1–4]. Because PIDEs combine integral, spatial, and temporal terms, they provide a flexible framework for modeling phenomena with memory and nonlocal effects.

The theory of partial integro-differential equations is a significant area within the study of differential and integral equations. For example, in dynamical systems with automatic control, the inclusion of an integral term in differential equations of the first and second orders is essential [5,6]. A PIDE involves partial differentiation and integration of the unknown function and is frequently employed when memory effects must be taken into account. PIDEs can be solved by a variety of methods, including finite difference schemes, collocation methods, the variational iteration method, and spline collocation methods [5,7,8].

Fractional calculus extends classical differentiation and integration to non-integer orders and has become increasingly important in science and engineering [9]. It provides a natural

Check for updates

Academic Editors: Jorge Sánchez-Ortiz, Martin P. Arciga-Alejandre and Ricardo Abreu-Blaya

Received: 22 August 2026
Revised: 20 October 2026
Accepted: 25 October 2026
Published: 1 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

mathematical setting for many physical and engineering processes [10,11], and fractional differential equations have been studied extensively in the literature [12–14]. Because complex phenomena are often nonlocal, fractional derivatives have proved useful in modeling memory-affecting processes. Time-fractional partial integro-differential equations (TFPIDEs) have therefore received considerable attention for their ability to describe a wide range of physical, biological, financial, and control-theoretic systems [15–24]. Representative applications include viscoelastic materials, anomalous diffusion in porous media, population dynamics, delayed control systems, and asset pricing models with long-memory effects.

Since most TFPIDE problems cannot be solved exactly in closed form, numerical methods play a central role in their study. Several approaches have been proposed for fractional partial integro-differential equations, including compact finite difference methods [25,26], the homotopy analysis method [27], and variational iteration techniques [7,28]. In particular, the variational iteration method (VIM) and the homotopy perturbation method (HPM) are well-established iterative schemes for fractional integro-differential equations; they construct approximate solutions through successive corrections or homotopy expansions [7,28,29]. Additional numerical schemes for fractional PIDEs are discussed in [30–32]. Relative to these semi-analytical series methods and to grid-based finite-difference discretizations, spectral/collocation schemes based on polynomial operational matrices reduce the continuous TFPIDE to a single algebraic system, which is the approach taken here. The numerical treatment remains challenging because fractional operators are nonlocal, memory terms introduce long-range dependencies, and the interaction between fractional order and approximation must be handled carefully.

Although many fractional models can be formulated with alternative derivative definitions, the present work employs the classical Caputo derivative throughout. Other operators used in the recent literature include the two-scale fractal derivative [33] and related modified Caputo-type constructions in Chun-Hui He's fractal–fractional framework [34]. We mention these alternatives briefly for context; extending the present collocation scheme to such operators is left for future work.

The present Pell–Lucas collocation scheme is conceptually related to the classical Ritz method of variational theory [35]: both seek approximate solutions in a finite-dimensional polynomial space and lead to algebraic equations for the unknown coefficients. The essential distinction is that the Ritz method arises from a variational (energy) formulation, whereas the scheme developed below is a residual collocation method driven by operational matrices for the Caputo time derivative, the spatial derivatives, and the Volterra integral term.

Related Pell–Lucas collocation studies address different equation classes. In particular, Yüzbaşı and Yıldırım applied Pell–Lucas collocation with residual correction to nonlinear population models [36] and later to a fractional HIV/AIDS compartmental system [24], while Abd-Elhameed et al. developed shifted Pell–Lucas algorithms with convergence estimates for fractional and classical Rosenau–Hyman equations [37]. Closely related spectral/collocation work for fractional PDEs includes Chebyshev–Legendre treatments of Lévy–Feller diffusion and variable-order advection–dispersion equations [38,39] and related Riesz–Feller fractional Schrödinger models [40]. The present contribution targets a linear time-fractional partial integro-differential equation with a Volterra kernel $K(x,t,s)$ and builds a unified collocation system from operational matrices for the Caputo time derivative, the spatial second derivative, and the Volterra integral term. The novelty therefore rests on this combined nonlocal treatment, not on the use of Pell–Lucas polynomials alone.

In this work, we combine Pell–Lucas (PL) polynomials with a collocation method to obtain numerical approximations of TFPIDEs with initial and boundary conditions.

PL polynomials are related to Pell and Lucas numbers and have been used in number theory, combinatorics, and numerical analysis [24,41–45]. Their recurrence relation provides a convenient basis for constructing operational matrices of integer and fractional derivatives. Following the operational-matrix approach widely used in spectral and collocation methods [46], we approximate the solution by a truncated bivariate Pell–Lucas expansion, transform the governing equation and the Volterra integral term into matrix form, and enforce the initial and boundary conditions at uniform collocation points. This procedure reduces the original problem to a system of algebraic equations that can be treated numerically.

The main contributions of this paper are threefold: (i) the development of a Pell–Lucas collocation formulation for the TFPIDE (1) and (2); (ii) the derivation and use of operational matrices for the fractional-time derivative, spatial derivatives, and Volterra integral term, leading to an algebraic collocation system; and (iii) an error analysis comprising an upper error bound and a residual-based error-estimation procedure, together with numerical illustrations on representative test problems.

In this paper, we investigate the time-fractional partial integro-differential equation

$$ {}^{C}D_t^{\beta} v(x,t) = h(x,t)v_{xx}(x,t) + \int_0^t K(x,t,s)v(x,s)ds + f(x,t), \quad a \le x \le b,\ 0 \le t \le T. \tag{1} $$

with the following initial and boundary conditions:

$$ \begin{aligned} v(a,t) &= q_1(t), && t \in [0,T], \\ v(b,t) &= q_2(t), && t \in [0,T], \\ v(x,0) &= q_3(x), && x \in [a,b], \end{aligned} \tag{2} $$

where $q_1(t), q_2(t), q_3(x)$, $h(x,t)$, and $f(x,t)$ are known continuous functions, $K(x,t,s)$ is the kernel of the Volterra integral term defined on $\mathbb{R}^3$, and ${}^{C}D_t^{\beta} v(x,t)$ denotes the Caputo fractional derivative of order $0 < \beta \le 1$.

The approximate solution is constructed as a truncated bivariate Pell–Lucas series:

$$ v(x,t) = \sum_{m=0}^{\infty}\sum_{s=0}^{\infty} a_{m,s}\phi_{m,s}(x,t), \qquad \phi_{m,s}(x,t) = \phi_m(x)\phi_s(t), \tag{3} $$

where $\phi_m(x)$ and $\phi_s(t)$ are PL polynomials and $a_{m,s}$, $m,s = 0,1,2,\ldots$, denote the unknown coefficients.

If the infinite series (3) is truncated after $(M+1)$ terms in each variable, it can be written as

$$ v(x,t) \simeq v_{2M}(x,t) = \sum_{m=0}^{M}\sum_{s=0}^{M} a_{m,s}\phi_{m,s}(x,t), \qquad m,s = 0,1,2,\ldots,M. \tag{4} $$

Equation (1) is treated as a fractional integro-differential equation in which a Volterra integral term with a general kernel appears from the outset, rather than being obtained by converting a partial differential equation into an equivalent integral form.

The remainder of the paper is organized as follows. Section 2 reviews fractional calculus and Pell–Lucas polynomials. Section 3 derives operational matrices for integer- and fractional-order derivatives. Section 4 constructs the matrix representation of the TFPIDE together with the initial and boundary conditions. Section 5 presents the collocation system and its assembly. Section 6 discusses error bounds and error estimation. Section 7 reports numerical examples, and Section 8 gives concluding remarks.

2. Preliminaries

This section introduces the fractional operators and Pell–Lucas polynomials used throughout the paper.

2.1. Fractional Calculus

The basic notions of fractional calculus required in the subsequent analysis are reviewed below [14,15].

Definition 1. *Left-sided and right-sided Riemann–Liouville fractional integral operators* $I^{\beta}_{a^+}$, $I^{\beta}_{b^-}$ *where* $\mathrm{Re}(\beta) > 0$ *and* $v \in L_1(a,b)$ *are defined as:*

$$I^{\beta}_{a^+} v(t) = \frac{1}{\Gamma(\beta)} \int_a^t (t-\zeta)^{\beta-1} v(\zeta) d\zeta, \qquad \zeta \in (a,b], \quad t > a, \tag{5}$$

$$I^{\beta}_{b^-} v(t) = \frac{1}{\Gamma(\beta)} \int_t^b (\zeta - t)^{\beta-1} v(\zeta) d\zeta, \qquad \zeta \in [a,b), \quad t < b, \tag{6}$$

Definition 2. *For* $\beta > 0, t > a$ *and* $\beta, a, t \in \mathbb{R}$, *the Caputo fractional derivative of order* β, *of function* $v(t)$ *on the interval* $[a,t]$ *is defined as follows:*

$${}^{C}_{a}D^{\beta}_{t} v(t) = \begin{cases} \frac{1}{\Gamma(n-\beta)} \int_a^t (t-\zeta)^{n-\beta-1} v^{(n)}(\zeta) d\zeta, & n-1 < \beta < n \in \mathbb{N}, \\ \frac{d^n}{dt^n} v(t), & \beta = n \in \mathbb{N}, \end{cases} \tag{7}$$

where ${}^{C}_{a}D^{\beta}_{t}$ *represents the Caputo derivative of fractional order* β.

The Caputo operator is related to the Riemann–Liouville integrals of Definition 1 by

$${}^{C}_{a}D^{\beta} v(t) = I^{n-\beta}_{a^+} D^{(n)} v(t),$$

and therefore

$${}^{C}_{a}D^{\beta} v(t) = I^{n-\beta}_{a^+} v^{(n)}(t) = \frac{1}{\Gamma(n-\beta)} \int_a^t (t-\zeta)^{n-\beta-1} v^{(n)}(\zeta) d\zeta.$$

In particular, for $0 < \beta \leq 1$, *the Caputo derivative of* $v(x,t)$ *with respect to* t *is defined as follows.*

$${}^{C}_{0}D^{\beta}_{t} v(x,t) = \begin{cases} \frac{1}{\Gamma(1-\beta)} \int_0^t (t-s)^{-\beta} v_t(x,s) ds, & 0 < \beta < 1, \\ v_t(x,t), & \beta = 1, \end{cases} \tag{8}$$

The following properties are satisfied by fractional differential and integral operators:

$$\begin{aligned}
&(1)\ I^{\beta} I^{\gamma} v = I^{\beta+\gamma} v, \quad \beta, \gamma > 0, \\
&(2)\ I^{\beta} t^{\gamma} = \frac{\Gamma(\gamma+1)}{\Gamma(\beta+\gamma+1)} t^{\beta+\gamma}, \quad \beta > 0,\ \gamma > -1,\ t > 0, \\
&(3)\ I^{\beta}\ {}^{C}D^{\beta}_{t} v(t) = v(t) - \sum_{k=0}^{m-1} v^{(k)}(0^+) \frac{t^k}{k!}, \quad t > 0,\ m-1 < \beta \leq m, \\
&(4)\ {}^{C}D^{\beta}_{t} I^{\beta} v(t) = v(t), \quad t > 0,\ m-1 < \beta \leq m, \\
&(5)\ {}^{C}D^{\beta}_{t} k = 0, \quad k \text{ is constant}, \\
&(6)\ {}^{C}D^{\beta}_{t} t^{\gamma} = \begin{cases} 0, & \gamma \in \mathbb{N}_0,\ \gamma < \lceil \beta \rceil, \\ \frac{\Gamma(\gamma+1)}{\Gamma(\gamma-\beta+1)} t^{\gamma-\beta}, & \gamma \in \mathbb{N}_0,\ \gamma \geq \lceil \beta \rceil. \end{cases}
\end{aligned} \tag{9}$$

where $\lceil \beta \rceil$ denotes the smallest integer greater than or equal to β and $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

2.2. Review of the Pell–Lucas Polynomials

Polynomial bases play a central role in many applied mathematics techniques. According to the literature [47], Pell–Lucas polynomials provide effective tools for the numerical treatment of differential equations. Further properties and results can be found in [48,49].

The Pell–Lucas polynomials $\phi_m(x)$ are defined by $\phi_0(x) = 2$, $\phi_1(x) = 2x$, and, for $m \geq 2$,

$$\phi_m(x) = \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{m\Gamma(m-j)}{\Gamma(j+1)\Gamma(m-2j+1)} (2x)^{m-2j}, \tag{10}$$

where $\lfloor \frac{m}{2} \rfloor$ denotes the greatest integer less than or equal to $\frac{m}{2}$. Equivalently, they satisfy the recurrence

$$\phi_{m+1}(x) = 2x\phi_m(x) + \phi_{m-1}(x), \quad m \geq 1, \ x \in \mathbb{R},$$

with the same initial polynomials ϕ_0 and ϕ_1 [48,49].

The derivative of $\phi_m(x)$ with respect to x is defined recursively by:

$$\phi'_m(x) = 2x\phi'_{m-1}(x) + \phi'_{m-2}(x) + 2\phi_{m-1}(x), \quad m \geq 2, \tag{11}$$

where $\phi'_0(x) = 0$ and $\phi'_1(x) = 2$.

Lemma 1. *The expansion of the following power function in the sense of PL polynomials can be expressed as follows*

$$x^m = \frac{1}{2^m} \sum_{j=0}^{\lfloor \frac{m}{2} \rfloor} \frac{(-1)^j \Gamma(m+1)\zeta_{m-2j}}{\Gamma(j+1)\Gamma(m-j+1)} \phi_{m-2j}(x), \tag{12}$$

where

$$\zeta_{m-2j} = \begin{cases} \frac{1}{2}, & j = \frac{m}{2}, \\ 1, & j < \frac{m}{2} \end{cases}. \tag{13}$$

Proof. See [50] for more details. □

Lemma 2. *The first derivative of Pell–Lucas polynomials can be written as follows.*

$$\frac{d}{dx}\phi_m(x) = 2m \sum_{j=0}^{\lfloor \frac{m-1}{2} \rfloor} (-1)^j \xi_{m-2j-1} \phi_{m-2j-1}(x), \qquad m > 2, \tag{14}$$

where the auxiliary coefficients ξ_k are defined by

$$\xi_k = \begin{cases} \frac{1}{2}, & k = 0, \\ 1, & k \geq 1. \end{cases}$$

Proof. See [50]. □

The Pell–Lucas polynomial representation in Equation (10) in terms of power functions x^j can be rewritten with the following formula:

$$\phi_m(x) = m \sum_{j=0}^{m} \frac{2^{j+1}\Gamma(\frac{m+j+2}{2})\delta_{m+j}}{(m+j)\Gamma(j+1)\Gamma\left(\frac{m-j+2}{2}\right)} x^j, \quad m \geq 1, \tag{15}$$

where

$$\delta_z = \begin{cases} 1, & z\ even, \\ 0, & z\ odd. \end{cases} \tag{16}$$

The equivalent representation of the power function in Equation (12) in terms of Pell–Lucas polynomials is given as follows:

$$x^m = \frac{\Gamma(m+1)}{2^m} \sum_{j=0,\,(j+m)\ even}^{m} \frac{(-1)^{\frac{m-j}{2}} \xi_j}{\Gamma\left(\frac{m+j+2}{2}\right)\Gamma\left(\frac{m-j+2}{2}\right)} \phi_j(x), \qquad m \geq 1. \tag{17}$$

One can see [48,51] for further details on Pell–Lucas polynomials.

3. Pell–Lucas Operational Matrices

In this section, integer- and fractional-order differential operators are represented as matrix operations associated with Pell–Lucas polynomials.

3.1. Operational Matrix of Classical Derivatives

A function $v(x) \in L_2(a,b)$ can be approximated by an infinite Pell–Lucas series as follows:

$$v(x) = \sum_{m=0}^{\infty} a_m \phi_m(x). \tag{18}$$

where a_m, $m = 0, 1, 2, \ldots$, are the unknown coefficients. Truncating the series after the first $M+1$ terms, Equation (18) can be rewritten as

$$v(x) \simeq v_M(x) = \sum_{m=0}^{M} a_m \phi_m(x) = A^T \Phi_M(x), \tag{19}$$

where the unknown coefficients vector A^T and vector of Pell–Lucas polynomials $\Phi_M(x)$ are defined by:

$$A^T = \begin{bmatrix} a_0 & a_1 & \cdots & a_M \end{bmatrix}, \tag{20}$$

and

$$\Phi_M(x) = \begin{bmatrix} \phi_0(x) & \phi_1(x) & \cdots & \phi_M(x) \end{bmatrix}^T. \tag{21}$$

Lemma 3. *The operational matrix of first derivative of the vector $\Phi_M(x)$ can be expressed as follows:*

$$\frac{d}{dx}\Phi_M(x) = W^{(1)}\Phi_M(x), \tag{22}$$

where $W^{(1)} = (w_{lm}^{(1)})$ is the operational matrix of the first derivative and has dimension $(M+1) \times (M+1)$.

Proof. In accordance with the property of the first order derivative of Pell–Lucas polynomials, the elements of matrix $W^{(1)}$ are represented as follows:

$$w_{lm}^{(1)} = \begin{cases} 2l(-1)^{\frac{l-m+1}{2}} \xi_m, & l > m, (l+m)\ odd, \\ 0, & l \leq m, (l+m)\ even. \end{cases} \tag{23}$$

The Pell–Lucas operational matrix of the nth-order derivative of the vector $\Phi_M(x)$ can be expressed as

$$\frac{d^n}{dx^n}\Phi_M(x) = W^{(n)}\Phi_M(x) = (W^{(1)})^n \Phi_M(x), \tag{24}$$

where $n \in \mathbb{Z}^+$. For further details, see [49]. □

3.2. Operational Matrix of Derivative of Fractional Order

Theorem 1. *Assume $\Phi_M(x)$ is the Pell–Lucas vector defined in Equation (21). The Caputo fractional derivative of order β of $\Phi_M(x)$ can be represented as follows:*

$$ {}^{C}D_x^{\beta}\Phi_M(x) = x^{-\beta}H^{(\beta)}\Phi_M(x), \tag{25}$$

where $H^{(\beta)} = (h_{p,l}^{\beta})$ is an $(M+1)\times(M+1)$ matrix representing the Pell–Lucas operational matrix of the Caputo derivative of order β and can be expressed as follows:

$$H^{(\beta)} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \zeta_\beta(\lceil\beta\rceil,0) & \zeta_\beta(\lceil\beta\rceil,1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \zeta_\beta(p,0) & \cdots & \zeta_\beta(p,p) & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ \zeta_\beta(M,0) & \zeta_\beta(M,1) & \zeta_\beta(M,2) & \cdots & \zeta_\beta(M,M) \end{pmatrix}, \tag{26}$$

where the elements of $H^{(\beta)}$ can be given as

$$h_{p,l}^{\beta} = \begin{cases} \zeta_\beta(p,l), & p \geq l, \quad p \geq \lceil\beta\rceil, \quad l = 0,1,\ldots,p, \\ 0, & p < l, \quad p < \lceil\beta\rceil, \end{cases} \tag{27}$$

where

$$\zeta_\beta(p,l) = p \sum_{n=\lceil\beta\rceil, n-l+2>0}^{p} \frac{(-1)^{\frac{n-l}{2}}\delta_{p+n}\delta_{l+n}\xi_l\Gamma(n+1)\Gamma\left(\frac{p+n}{2}\right)}{\Gamma\left(\frac{p-n+2}{2}\right)\Gamma\left(\frac{n-l+2}{2}\right)\Gamma\left(\frac{n+l+2}{2}\right)\Gamma(n-\beta+1)}. \tag{28}$$

Proof. Using the Pell–Lucas polynomials definition in Equation (15) and definition of δ_z in Equation (16). The power function x^m can be represented as:

$$x^m = \frac{\Gamma(m+1)}{2^m}\sum_{j=0}^{m} \frac{(-1)^{\frac{m-j}{2}}\xi_j\delta_{j+m}}{\Gamma\left(\frac{m+j+2}{2}\right)\Gamma\left(\frac{m-j+2}{2}\right)}\phi_j(x), \qquad m \geq 1. \tag{29}$$

By calculating the fractional derivative of $\phi_p(x)$ in the Caputo sense of order β of Equation (21) is defined as follows:

$$ {}^{C}D_x^{\beta}\phi_p(x) = \sum_{j=0}^{p} \frac{2^{j+1}\Gamma(\frac{p+j+2}{2})\delta_{p+j}}{(j+p)\Gamma(j+1)\Gamma(\frac{p-j+1}{2})}\ {}^{C}D_x^{\beta}x^j. \tag{30}$$

Now, using Equation (7), we write Equation (30) for values greater than $\lceil\beta\rceil$ as follows:

$$ {}^{C}D_x^{\beta}\phi_p(x) = px^{-\beta}\sum_{j=\lceil\beta\rceil}^{p} \frac{2^{j+1}\Gamma(\frac{p+j+2}{2})\delta_{p+j}}{(j+p)\Gamma(\frac{p-j+1}{2})\Gamma(j-\beta+1)}x^j. \tag{31}$$

By changing the index from j to n, we have

$$ {}^{C}D_x^{\beta}\phi_p(x) = px^{-\beta}\sum_{n=\lceil\beta\rceil}^{p}\frac{2^{n+1}\Gamma\left(\frac{p+n+2}{2}\right)\delta_{p+n}}{(p+n)\Gamma\left(\frac{p-n+2}{2}\right)\Gamma(n-\beta+1)}x^n. \tag{32}$$

By replacing x^n of Equation (29) in Equation (32) and using the properties of the Gamma function, we will have

$$ {}^{C}D_x^{\beta}\phi_p(x) = x^{-\beta}\sum_{\substack{l=0\\ n-l+2>0}}^{p}\left(p\sum_{n=\lceil\beta\rceil}^{p}\frac{(-1)^{\frac{n-l}{2}}\delta_{p+n}\delta_{l+n}\zeta_l\Gamma(n+1)\Gamma(\frac{p+n}{2})}{\Gamma(\frac{p-n+2}{2})\Gamma(\frac{n+l+2}{2})\Gamma(\frac{n-l+2}{2})\Gamma(n-\beta+1)}\right)\phi_l(x), \tag{33}$$

then Equation (33) can be written as:

$$ {}^{C}D_x^{\beta}\phi_p(x) = x^{-\beta}\sum_{l=0}^{p}\zeta_{\beta}(p,l)\phi_l(x), \tag{34}$$

where $\zeta_{\beta}(p,l)$ is given in Equation (28).

Equation (34) can also be expressed as the vector form

$$ {}^{C}D_x^{\beta}\phi_p(x) = x^{-\beta}[\zeta_{\beta}(p,0),\zeta_{\beta}(p,1),\ldots,\zeta_{\beta}(p,p),0,0,\ldots,0]\Phi(x), \quad \lceil\beta\rceil \le j \le N, \tag{35}$$

Moreover, we can write

$$ {}^{C}D_x^{\beta}\phi_p(x) = x^{-\beta}[0,0,\ldots,0], \quad 0 \le p \le \lceil\beta\rceil - 1. \tag{36}$$

Combining Equations (35) and (36) yields the desired result. □

4. Matrix Representations

To obtain a matrix formulation of Equation (1) in terms of Pell–Lucas polynomials, we rewrite it as

$$ D_1(x,t) - D_2(x,t) = f(x,t), \tag{37}$$

where the derivative and integral parts are expressed as follows:

$$ D_1(x,t) = {}^{C}D_t^{\beta}v(x,t) - h(x,t)v_{xx}(x,t), \tag{38}$$

and

$$ D_2(x,t) = \int_0^t K(x,t,s)v(x,s)ds. \tag{39}$$

The problem is thereby transformed into a system of algebraic equations, and the approximate solution is obtained by the collocation method described below.

Lemma 4. *The approximate solution* (3) *in terms of Pell–Lucas polynomials can be represented as:*

$$ v_{2M}(x,t) = \Phi_M(x)\overline{\Phi}_M(t)\overline{A}_M, \tag{40}$$

where

$$ \overline{A}_M = [A_0, A_1, \ldots, A_M]^T, \quad A_i = [a_{i,0}, a_{i,1}, \ldots, a_{i,M}]^T, \quad i = 0,1,2,\ldots,M, $$

$$ \Phi_M(x) = [\phi_0(x), \phi_1(x), \ldots, \phi_M(x)], $$

and $\overline{\Phi}_M(t) = diag(\Phi_M(t))$.

Proof. The expansion of approximate solution $v_{2M}(x,t)$ with respect to Pell–Lucas vectors $\Phi_M(x)$ and $\Phi_M(t)$ can be easily expressed by Equation (40). □

Lemma 5. *The second-order spatial partial derivative of Equation (3) in terms of PL polynomials in matrix structure can be represented by:*

$$(v_{2M}(x,t))_{xx} = \Phi_M(x)\boldsymbol{N}^2\overline{\Phi}_M(t)\overline{\boldsymbol{A}}_M, \tag{41}$$

where if M is odd

$$\boldsymbol{N} = \begin{pmatrix} 0 & 1 & 0 & -3 & \cdots & (-1)^{\frac{M-1}{2}}M \\ 0 & 0 & 4 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 6 & \cdots & (-1)^{\frac{M-3}{2}}2M \\ \vdots & \vdots & \vdots & \vdots & \ddots & 2M \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}_{(M+1)\times(M+1)},$$

and if M is even

$$\boldsymbol{N} = \begin{pmatrix} 0 & 1 & 0 & -3 & \cdots & 0 \\ 0 & 0 & 4 & 0 & \cdots & (-1)^{\frac{M-2}{2}}2M \\ 0 & 0 & 0 & 6 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}_{(M+1)\times(M+1)},$$

where, $\Phi_M(x), \overline{\Phi}_M(t)$ and $\overline{\boldsymbol{A}}_M$ are defined in Lemma 4. The two displays represent the same truncated differentiation map $\Phi_M \mapsto \Phi'_M$ (and thence $\Phi''_M = \Phi_M\mathbf{N}^2$); the even/odd cases differ only in which entries of the highest-degree columns remain nonzero after truncation of the Pell–Lucas derivative recurrence.

Proof. The first and second spatial derivatives of approximate solution in Equation (40) in matrix form can be expressed as follows:

$$(v_{2M}(x,t))_x = \Phi'_M(x)\overline{\Phi}_M(t)\overline{\mathbf{A}}_M, \tag{42}$$

and

$$(v_{2M}(x,t))_{xx} = \Phi''_M(x)\overline{\Phi}_M(t)\overline{\mathbf{A}}_M, \tag{43}$$

In order to construct the matrix structure of $\Phi''_M(x)$ in Equation (43), we use the recurrence formula of first derivative of Pell–Lucas polynomials. So, the matrix structure of the second derivative of $\Phi_M(x)$ can be written as:

$$\Phi''_M(x) = \Phi_M(x)\mathbf{N}^2. \tag{44}$$

Finally, by substituting Equation (44) into the expression (43), the matrix structure (41) is derived. The parity dependence of the displayed pattern of **N** arises because the highest retained index generated by the derivative recurrence depends on whether the truncation degree M is even or odd; both forms encode the same linear map on the truncated basis. □

Lemma 6. *The matrix structure of $K(x,t,s)$ in Equation (39) can be represented in terms of PL polynomials as follows:*

$$[K(x,t,s)] = \boldsymbol{X}_M(x)\overline{\boldsymbol{X}}_M(t)\boldsymbol{K}\boldsymbol{X}_M^T(s), \quad \boldsymbol{K} = [k^T_{p_1,p_2,p_3}], \quad p_1,p_2,p_3 = 0,1,\ldots,M, \tag{45}$$

where

$$k^T_{p_1,p_2,p_3} = \frac{1}{p_1!p_2!p_3!}\frac{\partial^{p_1+p_2+p_3}K(0,0,0)}{\partial x^{p_1}\partial t^{p_2}\partial s^{p_3}}, \qquad p_1,p_2,p_3 = 0,1,\ldots,M, \tag{46}$$

$$\boldsymbol{X}_M(x) = \begin{bmatrix}1 & x & \cdots & x^M\end{bmatrix}, \ \boldsymbol{X}^T_M(s) = \begin{bmatrix}1 & s & \cdots & s^M\end{bmatrix}^T, \ \overline{\boldsymbol{X}}_M(t) = diag(\boldsymbol{X}_M(t)).$$

Proof. The kernel $K(x,t,s)$ based on the truncated Maclaurin series, can be expressed as follows:

$$K(x,t,s) = \sum_{p_1=0}^{M}\sum_{p_2=0}^{M}\sum_{p_3=0}^{M} k^T_{p_1,p_2,p_3}x^{p_1}t^{p_2}s^{p_3}. \tag{47}$$

Multiplying by $\mathbf{K}\mathbf{X}^T_M(x)$, from the right side in vector $\mathbf{X}_M(x)\overline{\mathbf{X}}_M(t)$, the relation (45) is obtained. □

In the collocation scheme, the Volterra integral term is therefore not evaluated by numerical quadrature of the original kernel at each step. Instead, after truncating K by its Maclaurin polynomial, the integral against the Pell–Lucas expansion of v_{2M} is converted into an exact matrix product through the operational representation of Lemma 7 (and the associated integrated basis factors).

Lemma 7. *In the Volterra integral Equation* (39)*, the expression* $v_{2M}(x,s)$ *has the following matrix representation:*

$$v_{2M}(x,s) = \Phi_M(s)\overline{\Phi}_M(x)\overline{A}^*_M, \tag{48}$$

where

$$\overline{A}^*_M = \begin{bmatrix}\boldsymbol{A}^*_0 & \boldsymbol{A}^*_1 & \cdots & \boldsymbol{A}^*_M\end{bmatrix}^T, \qquad \boldsymbol{A}^*_i = \begin{bmatrix}a_{0,i} & a_{1,i} & \cdots & a_{M,i}\end{bmatrix}^T,$$

$$\Phi_M(s) = \begin{bmatrix}\phi_0(s) & \phi_1(s) & \cdots & \phi_M(s)\end{bmatrix}, \qquad \overline{\Phi}_M(x) = diag(\Phi_M(x)).$$

Proof. By substituting s instead of t in Lemma 4, we have

$$v_{2M}(x,s) = \Phi_M(x)\overline{\Phi}_M(s)\overline{\mathbf{A}}_M. \tag{49}$$

This expression can be more exactly written as follows:

$$v_{2M}(x,s) = [a_{0,0} \quad sa_{0,1} \quad s^2a_{0,2} \quad xa_{1,0} \quad xsa_{1,1} \quad xs^2a_{1,2} \quad x^2a_{2,0} \quad x^2sa_{2,1} \quad x^2s^2a_{2,2}] \tag{50}$$

If $\Phi_M(s)$ and $\overline{\Phi}_M(x)$ are substituted in place of $\Phi_M(x)$ and $\overline{\Phi}_M(s)$ in Equation (49), then $\overline{\mathbf{A}}^*_M$ is written instead of $\overline{\mathbf{A}}_M$ to obtain the equality in Equation (50). Consequently, Equation (48) is obtained.

Explicitly, both stacked vectors have length $(M+1)^2$ and contain the same scalars $a_{i,j}$. Writing

$$\overline{\mathbf{A}}_M = \left(a_{0,0},\ldots,a_{0,M},a_{1,0},\ldots,a_{1,M},\ldots,a_{M,0},\ldots,a_{M,M}\right)^T$$

(spatial index outer) and

$$\overline{\mathbf{A}}^*_M = \left(a_{0,0},\ldots,a_{M,0},a_{0,1},\ldots,a_{M,1},\ldots,a_{0,M},\ldots,a_{M,M}\right)^T$$

(temporal index outer), one has the exact identity

$$\overline{\mathbf{A}}^*_M = P\,\overline{\mathbf{A}}_M, \tag{51}$$

where $P \in \mathbb{R}^{N\times N}$ with $N = (M+1)^2$ is the (symmetric involutory) commutation matrix defined by

$$P\,e_{i(M+1)+j} = e_{j(M+1)+i}, \qquad i,j = 0,\ldots,M$$

(equivalently, if $A = (a_{i,j})$ then $\overline{\mathbf{A}}_M = \mathrm{vec}(A)$ and $\overline{\mathbf{A}}^*_M = \mathrm{vec}(A^T)$ in row-major stacking). Thus P has dimension $(M+1)^2 \times (M+1)^2$. In particular $P = P^T = P^{-1}$, and $\overline{\mathbf{A}}^*_M$ does not introduce independent unknowns: every occurrence of $\overline{\mathbf{A}}^*_M$ in Lemmas 8 and 9 is the same coefficient vector as $\overline{\mathbf{A}}_M$, only reordered for the Volterra/time layout. □

Lemma 8. *The matrix representation of time-fractional derivative of order β of Equation (3) in Caputo sense has the following matrix structure:*

$$ {}^{C}D_t^{\beta} v_{2M}(x,t) = t^{-\beta}\Phi_M(t)(H^{(\beta)})^T\overline{\Phi}_M(x)\overline{\boldsymbol{A}}^*_M, \tag{52}$$

Here, $\Phi_M(t), \overline{\Phi}_M(x)$ and $\overline{\boldsymbol{A}}^*_M$ are as in Lemma 7, and $H^{(\beta)}$ in Equation (26).

Proof. By taking the Caputo fractional derivative of order β of the approximate solution (48) with respect to t, we obtain

$$ {}^{C}D_t^{\beta} v_{2M}(x,t) = {}^{C}D_t^{\beta}\left(\Phi_M(t)\overline{\Phi}_M(x)\overline{\mathbf{A}}^*_M\right), \tag{53}$$

then, ${}^{C}D_t^{\beta}\Phi_M(t)$ in Equation (53) in matrix form can be represented as $t^{-\beta}\Phi_M(t)(H^{(\beta)})^T$ based on Equation (25). Inserting this relationship into Equation (53) gives the desired result. □

Lemma 9. *The term $D_1(x,t)$ has the following matrix structure:*

$$ [D_1(x,t)] = t^{-\beta}\Phi_M(t)(H^{(\beta)})^T\overline{\Phi}_M(x)\overline{\boldsymbol{A}}^*_M - [h(x,t)]\Phi_M(x)\boldsymbol{N}^2\overline{\Phi}_M(t)\overline{\boldsymbol{A}}_M. \tag{54}$$

Here, $\Phi_M(t), \overline{\Phi}_M(x)$ and $\overline{\boldsymbol{A}}^*_M$ are as in Lemma 8, $\Phi_M(x), \overline{\Phi}_M(t)$ and $\overline{\boldsymbol{A}}_M$ are as in Theorem 4 and $\boldsymbol{N}$ in Lemma 5.

Proof. By substituting Equations (41) and (52) into Equation (38), the desired result is obtained. □

Lemma 10. *The matrix representation of $D_2(x,t)$ in Equation (38) can be expressed as*

$$ [D_2(x,t)] = \boldsymbol{X}_M(x)\overline{\boldsymbol{X}}_M(t)\boldsymbol{K}\boldsymbol{S}(t)\overline{\Phi}_M(x)\overline{\boldsymbol{A}}^*_M, \tag{55}$$

where

$$ \boldsymbol{S}(t) = \int_0^t \boldsymbol{X}_M^T(s)\Phi_M(s)ds. $$

Here, $\boldsymbol{X}_M(x), \overline{\boldsymbol{X}}_M(t), \boldsymbol{K}$ and $\boldsymbol{X}_M^T(s)$ are in Lemma 6 and $\Phi_M(s), \overline{\Phi}_M(x), \overline{\boldsymbol{A}}^*_M$ are in Lemma 7.

Proof. By employing Equations (45) and (48) into Equation (39), the matrix form of $D_2(x,t)$ can be easily obtained. □

In the following, the initial and boundary conditions (2) are written in matrix form.

Lemma 11. *The matrix form of the conditions (2) is given by*

$$\begin{aligned} \Phi_M(a)\overline{\Phi}_M(t)\overline{\boldsymbol{A}}_M &= q_1(t), \\ \Phi_M(b)\overline{\Phi}_M(t)\overline{\boldsymbol{A}}_M &= q_2(t), \\ \Phi_M(x)\overline{\Phi}_M(0)\overline{\boldsymbol{A}}_M &= q_3(x). \end{aligned} \tag{56}$$

Here, $\Phi_M(x), \overline{\Phi}_M(t)$ and $\overline{\boldsymbol{A}}_M$ are as in Lemma 4.

Proof. By inserting a, b instead of x and 0 instead of t in Lemma 4, we obtain $v_{2M}(a,t) = \Phi_M(a)\overline{\Phi}_M(t)\overline{\mathbf{A}}_M$, $v_{2M}(b,t) = \Phi_M(b)\overline{\Phi}_M(t)\overline{\mathbf{A}}_M$ and $v_{2M}(x,0) = \Phi_M(x)\overline{\Phi}_M(0)\overline{\mathbf{A}}_M$, respectively. By using Equation (2), the desired result is easily obtained. □

5. Procedure of the Solution

This section describes the Pell–Lucas collocation method on uniform collocation points. The operational matrices derived in the previous sections are used to assemble a linear algebraic system for the unknown expansion coefficients.

The truncated series expansion (3) is substituted into Equation (1) and into the initial and boundary conditions (2). Using the matrix representations of the fractional-time derivative, spatial derivatives, and Volterra integral term, the governing equation can be written as

$$\begin{aligned}&\left\{t^{-\beta}\Phi_M(t)(H^{(\beta)})^T\overline{\Phi}_M(x) - \mathbf{X}_M(x)\overline{\mathbf{X}}_M(t)\mathbf{KS}(t)\overline{\Phi}_M(x)\right\}\overline{\mathbf{A}}_M^*\\&- [h(x,t)]\left\{\Phi_M(x)\mathbf{N}^2\overline{\Phi}_M(t)\right\}\overline{\mathbf{A}}_M = [f(x,t)].\end{aligned}$$

We define uniformly spaced collocation points by

$$x_i = a + \frac{b-a}{M}i, \quad i = 0,1,\ldots,M, \qquad t_j = \frac{T}{M}j, \quad j = 0,1,\ldots,M. \tag{57}$$

We assume that the matrix representation for the approximate solution of Equations (1) and (2) is as follows:

$$\mathbf{Z}^*\overline{\mathbf{A}}_M^* - \mathbf{Z}\overline{\mathbf{A}}_M = \mathbf{F}. \tag{58}$$

Using the structure of $\overline{\mathbf{A}}_M$ and $\overline{\mathbf{A}}_M^*$, Equation (58) may be recast as a linear system in the coefficients of $\overline{\mathbf{A}}_M$. Substituting the collocation points (57) into (58) yields the algebraic system

$$\mathbf{Z}^*(x_i,t_j)\overline{\mathbf{A}}_M^* - \mathbf{Z}(x_i,t_j)\overline{\mathbf{A}}_M = [f(x_i,t_j)], \tag{59}$$

where

$$\mathbf{Z}^* = \begin{pmatrix}\mathbf{Z}_0^*\\ \mathbf{Z}_1^*\\ \vdots\\ \mathbf{Z}_M^*\end{pmatrix}, \quad \mathbf{Z}_i^* = \begin{pmatrix}\mathbf{Z}^*(x_i,t_0)\\ \mathbf{Z}^*(x_i,t_1)\\ \vdots\\ \mathbf{Z}^*(x_i,t_M)\end{pmatrix}, \quad \mathbf{Z} = \begin{pmatrix}\mathbf{Z}_0\\ \mathbf{Z}_1\\ \vdots\\ \mathbf{Z}_M\end{pmatrix}, \quad \mathbf{Z}_i = \begin{pmatrix}\mathbf{Z}(x_i,t_0)\\ \mathbf{Z}(x_i,t_1)\\ \vdots\\ \mathbf{Z}(x_i,t_M)\end{pmatrix}, \tag{60}$$

and

$$\mathbf{F} = \begin{pmatrix}\mathbf{F}_0\\ \mathbf{F}_1\\ \vdots\\ \mathbf{F}_M\end{pmatrix}, \qquad \mathbf{F}_j = \begin{pmatrix}f(x_j,t_0)\\ f(x_j,t_1)\\ \vdots\\ f(x_j,t_M)\end{pmatrix}, \; j = 0,1,\ldots,M. \tag{61}$$

Lemma 12. *The initial and boundary conditions* (2) *in collocation points* (57) *can be represented as follows:*

$$\boldsymbol{U}\overline{\boldsymbol{A}}_M = \eta_j, \quad \boldsymbol{V}\overline{\boldsymbol{A}}_M = \vartheta_j, \quad \boldsymbol{W}\overline{\boldsymbol{A}}_M = \delta_i, \tag{62}$$

where

$$U = \begin{pmatrix} U_0 \\ U_1 \\ \vdots \\ U_M \end{pmatrix}, \quad V = \begin{pmatrix} V_0 \\ V_1 \\ \vdots \\ V_M \end{pmatrix}, \quad W = \begin{pmatrix} W_0 \\ W_1 \\ \vdots \\ W_M \end{pmatrix}, \quad \delta_i = \begin{pmatrix} w_{j,1} & w_{j,2} & \dots & w_{j,(M+1)^2} \end{pmatrix},$$

$$\eta_j = \begin{pmatrix} u_{j,1} & u_{j,2} & \dots & u_{j,(M+1)^2} \end{pmatrix}, \qquad \vartheta_j = \begin{pmatrix} v_{j,1} & v_{j,2} & \dots & v_{j,(M+1)^2} \end{pmatrix},$$

$$U_j = \Phi_M(a)\overline{\Phi}_M(t_j), \qquad V_j = \Phi_M(b)\overline{\Phi}_M(t_j), \qquad W_i = \Phi_M(x_i)\overline{\Phi}_M(0).$$

Proof. The stated block rows follow by evaluating the matrix forms of the initial and boundary conditions in Lemma 11 at the collocation nodes (57). □

The condition number of the collocation algebraic system in Equation (59) grows rapidly with M; for Example 1 with $\beta = 0.55$ and $M = 6$, the unscaled matrix condition number is already large (of order 10^8–10^{18}, depending on whether the overdetermined matrix or the normal equations are monitored). Adding Equation (62) to the system (59) yields the generalized matrix system

$$[\tilde{\mathbf{Z}}^*; \tilde{\mathbf{Z}}; \tilde{\mathbf{F}}]. \tag{63}$$

Solving the system (63) yields the Pell–Lucas coefficient vector $\overline{\mathbf{A}}_M$, and the approximate solution is obtained by substituting these coefficients into Equation (3).

6. Error Analysis

This section presents an upper bound for the approximation error and a residual-based error-estimation procedure for the Pell–Lucas collocation method. The first result provides an error bound in the infinity norm, and the second characterizes the error through an auxiliary problem that can be used when an exact solution is unavailable.

Lemma 13 (Upper Bound of Error)**.** *Let v denote the exact solution of Equation (1), and let v_{2M} denote its Pell–Lucas polynomial approximation on the domain $[a,b] \times [0,T]$. Let v_{2M}^{Mac} be the truncated Maclaurin series of the exact solution $v(x,t)$ in the domain $[a,b] \times [0,T]$. The absolute error of the proposed method has an upper bound as follows:*

$$\begin{aligned} \|v - v_{2M}\|_\infty \le k_M \|\tilde{\tilde{A}}_M\|_\infty + q_M \overline{q}_M \|\overline{A}_M\|_\infty \\ + \frac{1}{(2M+1)!} \left\| \left(x\frac{\partial}{\partial x} + t\frac{\partial}{\partial t} \right)^{2M+1} v(c_x, c_t) \right\|_\infty, \\ a \le x \le b, \quad 0 \le t \le T, \end{aligned} \tag{64}$$

where $\tilde{\tilde{A}}_M$ shows the coefficient matrix of $v_{2M}^{Mac}(x,t)$ and

$$k_M = \|X_M(x,t)\|_\infty, \qquad q_M = \|\Phi_M(x)\|_\infty, \qquad \overline{q}_M = \|\overline{\Phi}_M(t)\|_\infty.$$

Proof. See [5]. □

Theorem 2 (Error Estimation). *Let v and v_{2M} denote the exact solution of Equations (1) and (2) and its Pell–Lucas collocation approximation, respectively. For the approximate solution v_{2M}, define the residual function $R_{2M}(x,t)$ by*

$$R_{2M}(x,t) = L[v_{2M}(x,t)] - f(x,t). \tag{65}$$

Writing $e_{2M} = v - v_{2M}$, linearity of the operator L together with $L[v] = f$ yields the residual-driven error equation

$${}^{C}D_t^{\beta} e_{2M} - h(x,t)(e_{2M})_{xx} - \int_0^t K(x,t,s) e_{2M}(x,s) ds = -R_{2M}. \tag{66}$$

If, in addition, v_{2M} coincides with the prescribed data on the lateral boundaries and at the initial time, then the error satisfies the homogeneous traces

$$\begin{cases} e_{2M}(a,t) = 0, \\ e_{2M}(b,t) = 0, \\ e_{2M}(x,0) = 0, \qquad a \le x \le b, \qquad 0 \le t \le T. \end{cases} \tag{67}$$

In the overdetermined, column-equilibrated least-squares collocation implementation of Section 5, the governing equation and the initial/boundary conditions are enforced only in the discrete least-squares sense; consequently the residual R_{2M} need not vanish identically, and the homogeneous conditions (67) hold only approximately whenever the corresponding discrete mismatches are nonzero.

Proof. According to the definitions of v and v_{2M}, Equation (1) can be written in operator form as

$$L[v(x,t)] = f(x,t), \quad L[v] = {}^{C}D_t^{\beta} v - h(x,t) v_{xx} - \int_0^t K(x,t,s) v(x,s) ds. \tag{68}$$

The approximate solution $v_{2M}(x,t)$ is constructed to satisfy the discrete collocation equations and the imposed initial and boundary conditions at the selected collocation nodes. In particular, it is not asserted that v_{2M} solves the continuous problem (1) and (2) pointwise throughout the domain. Defining

$$L[v_{2M}] = {}^{C}D_t^{\beta}(v_{2M}) - h(x,t)(v_{2M})_{xx} - \int_0^t K(x,t,s) v_{2M}(x,s) ds \tag{69}$$

and the residual $R_{2M} = L[v_{2M}] - f$ as in (65), subtraction from (68) gives (66). The boundary and initial traces of e_{2M} equal the mismatches between the exact data q_1, q_2, q_3 and the corresponding traces of v_{2M}; they reduce to (67) precisely when those mismatches vanish. Thus, this completes the proof. □

Convergence Discussion

The error analysis presented in this section provides an upper bound on the approximation error (Lemma 13) and a residual-based procedure for error estimation (Theorem 2). For an exact solution $v(x,t)$ that is sufficiently smooth, Lemma 13 bounds $\|v - v_{2M}\|_{\infty}$ in terms of the truncation parameter M and a remainder term involving derivatives of v; this result establishes an error bound under the stated assumptions rather than a formal convergence theorem for the Pell–Lucas collocation method. Theorem 2 characterizes the error $e_{2M} = v - v_{2M}$ through an auxiliary problem driven by the residual R_{2M}, thereby offering a practical means to estimate the error when an exact solution is unavailable.

Concerning stability with respect to the kernel K and the diffusion coefficient h, several points should be distinguished. The continuous problem is assumed to admit a sufficiently regular solution for the given data; in particular, boundedness and smoothness of h and of the Volterra kernel K (together with the Caputo operator) are part of the standing hypotheses underlying the approximation bound in Lemma 13. Lemma 13 and Theorem 2 do not by themselves yield a uniform discrete stability estimate for arbitrary perturbations of K and h. Separately, the collocation discretization produces an algebraic matrix whose condition number may grow rapidly with M; this discrete conditioning is a numerical property of the basis and assembly and must be monitored on its own. The numerical experiments of Section 7 document the observed behavior after column equilibration: for the smooth manufactured Example 1 the L_2 and L_∞ errors decrease systematically for $M = 4, 6, 8, 10$, with large successive rates, while the numerical experiments in Section 7 report both the unscaled and column-scaled condition numbers. Column equilibration is a standard diagonal scaling of the least-squares system; in exact arithmetic it yields the same minimizer of $\|Ac - b\|_2$, but it is not a theoretical stability theorem.

In particular, the bound does not by itself guarantee monotone decrease of the discrete error for every increase of M in finite precision. Convergence with respect to M should therefore be assessed together with residual norms and condition numbers. In Section 7 we report the truncation degrees $M = 4, 6, 8, 10$ and apply column equilibration to the least-squares collocation matrix, which reduces the observed condition numbers while preserving the least-squares solution of the discrete system in exact arithmetic.

7. Numerical Results

This section illustrates the proposed Pell–Lucas collocation method on two test problems for Equations (1) and (2). The numerical computations were performed using Python with NumPy, employing the collocation assembly and least-squares solve with column equilibration described above; CPU times therefore refer to this implementation. The reported numerical results were subsequently reproduced using Python 3.13.3 and NumPy 2.5.3. The exact software versions used in the original computations were not recorded. No parallel computing or GPU acceleration was used.

The exact and approximate solutions are denoted by $v(x,t)$ and $v_{2M}(x,t)$, respectively. The quantities $|v - v_{2M}|$ and e_{2M} denote the actual error and the estimated error function. The L_2- and L_∞-norms of the absolute error are defined by

$$L_2 = \|v - v_{2M}\|_2 = \left(\int_0^T \int_a^b (v - v_{2M})^2 dx dt \right)^{\frac{1}{2}},$$

and

$$L_\infty = \|v - v_{2M}\|_\infty = \max\{|v(x,t) - v_{2M}(x,t)|\}, \qquad a \le x \le b,\ 0 \le t \le T.$$

Example 1. *We consider Equations* (1) *and* (2) *on the domain* $[-1,1] \times [0,1]$, *with*

$$h(x,t) = 1,$$
$$K(x,t,s) = e^{x(t-s)},$$
$$f(x,t) = (1-x^2)t^{1-\beta}E_{2,2-\beta}(-t^2) + 2\sin t + \frac{(1-x^2)(x\sin t - e^{xt} + \cos t)}{x^2+1},$$

and the initial and boundary conditions $v(x,0) = 0, v(-1,t) = 0, v(1,t) = 0$ *[52]. The forcing term is consistent with the exact solution below for* $h(x,t) = 1$ *and* $K(x,t,s) = e^{x(t-s)}$.

In $f(x,t)$, $E_{\eta,\zeta}$ denotes the two-parameter Mittag–Leffler function with $\mathrm{Re}(\eta) > 0$, *defined by the series [53]*

$$E_{\eta,\zeta}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(k\eta + \zeta)}, \quad t \in \mathbb{C}.$$

The exact solution of this problem is

$$v(x,t) = (1 - x^2)\sin t.$$

In the numerical experiments below, the overdetermined collocation system is solved in the least-squares sense after column equilibration of the discrete matrix (a standard diagonal scaling that, in exact arithmetic, preserves the least-squares minimizer of $\|Ac - b\|_2$). Table 1 reports the exact solution, the approximate solution, and the absolute error for $M = 6$ and $\beta = 0.55$ at selected time levels along the diagonal $x = t$. Figure 1 compares the exact and approximate solutions over the computational domain, and Figure 2 shows the corresponding absolute-error field.

Table 1. Numerical results for Example 1 with $M = 6$, $\beta = 0.55$, and selected time levels along $x = t$.

t	$v(x,t)$	$v_{2M}(x,t)$	$\lvert v(x,t) - v_{2M}(x,t)\rvert$
0.00	0.0000	1.8239×10^{-11}	1.8239×10^{-11}
0.10	9.8835×10^{-2}	9.8835×10^{-2}	1.2417×10^{-7}
0.20	1.9072×10^{-1}	1.9072×10^{-1}	3.5802×10^{-8}
0.30	2.6892×10^{-1}	2.6892×10^{-1}	1.2395×10^{-8}
0.40	3.2711×10^{-1}	3.2711×10^{-1}	1.8462×10^{-8}
0.50	3.5957×10^{-1}	3.5957×10^{-1}	1.1093×10^{-8}
0.60	3.6137×10^{-1}	3.6137×10^{-1}	3.9263×10^{-9}
0.70	3.2855×10^{-1}	3.2855×10^{-1}	1.2481×10^{-8}
0.80	2.5825×10^{-1}	2.5825×10^{-1}	1.1042×10^{-8}
0.90	1.4883×10^{-1}	1.4883×10^{-1}	2.1527×10^{-8}
1.00	0.0000	1.7023×10^{-9}	1.7023×10^{-9}

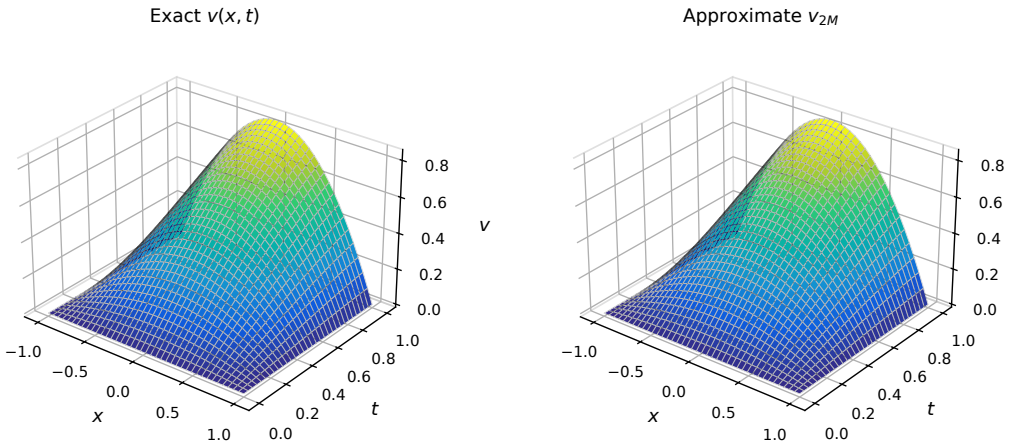


Figure 1. Exact and approximate solution surfaces for Example 1 with $M = 6$ and $\beta = 0.55$.

Table 2 presents the absolute errors along $x = t$ for $M = 4, 6, 8, 10$ with $\beta = 0.25$; the errors decrease systematically as M increases. Figure 3 displays these diagonal errors on a logarithmic scale, and Figure 4 reports the L_2 and L_∞ norms together with the growth of the (unscaled and column-scaled) condition numbers. Table 3 lists the global error norms, the CPU time, and the observed successive L_∞ rates for the same values of M. The large empirical rates indicate rapid error decay for the smooth manufactured solution considered here. Without column equilibration the discrete system at $M = 10$ becomes severely ill-conditioned; with the scaling used here the accuracy continues to improve, while Figure 4 makes the residual growth of $\kappa(A)$ transparent.

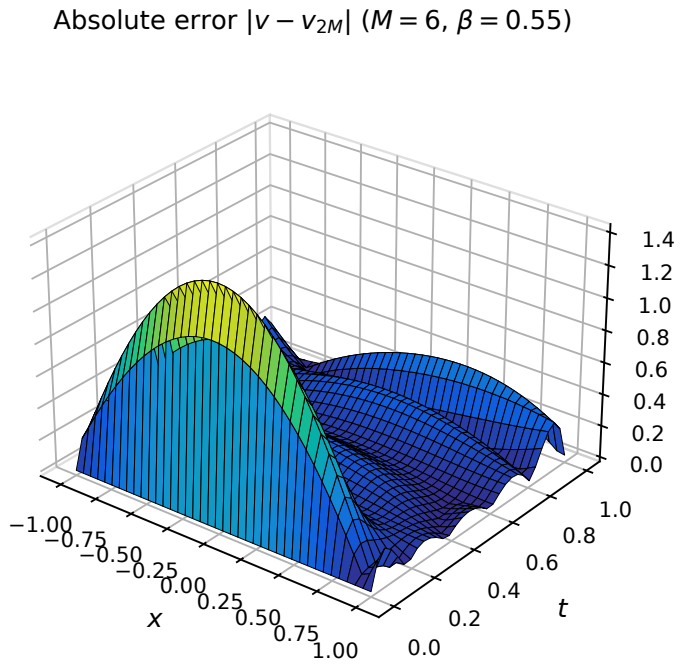


Figure 2. Absolute-error surface $|v - v_{2M}|$ for Example 1 with $M = 6$ and $\beta = 0.55$.

Table 2. Absolute errors for Example 1 with $M = 4, 6, 8, 10$, $\beta = 0.25$, and selected time levels along $x = t$.

t	$M = 4$	$M = 6$	$M = 8$	$M = 10$
0.00	1.3970×10^{-7}	3.0153×10^{-11}	4.0779×10^{-15}	1.7418×10^{-15}
0.10	3.3177×10^{-5}	6.0593×10^{-8}	4.9866×10^{-11}	1.8041×10^{-14}
0.20	1.6798×10^{-5}	1.5115×10^{-9}	5.8591×10^{-12}	5.1625×10^{-15}
0.30	1.7609×10^{-6}	3.1217×10^{-9}	1.0024×10^{-11}	4.0523×10^{-15}
0.40	5.9769×10^{-6}	9.2619×10^{-9}	3.0775×10^{-12}	3.4417×10^{-15}
0.50	4.8102×10^{-7}	3.7452×10^{-9}	3.2287×10^{-12}	1.2212×10^{-15}
0.60	6.6602×10^{-6}	2.2538×10^{-9}	4.8442×10^{-12}	4.9960×10^{-16}
0.70	5.4517×10^{-6}	4.4350×10^{-9}	4.6324×10^{-13}	2.8866×10^{-15}
0.80	7.1436×10^{-7}	5.1404×10^{-9}	7.7791×10^{-12}	1.5543×10^{-15}
0.90	2.9504×10^{-6}	7.6538×10^{-9}	3.6566×10^{-12}	1.4738×10^{-14}
1.00	7.5738×10^{-7}	2.0767×10^{-10}	1.1310×10^{-14}	1.0697×10^{-14}

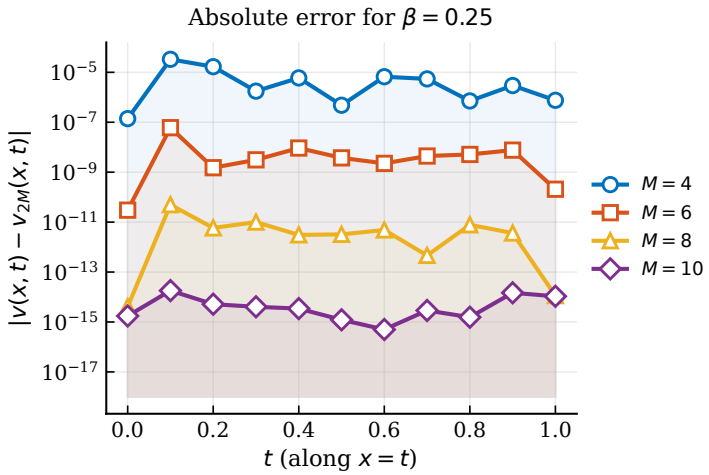


Figure 3. Absolute error along $x = t$ for Example 1 with $\beta = 0.25$ and $M = 4, 6, 8, 10$.

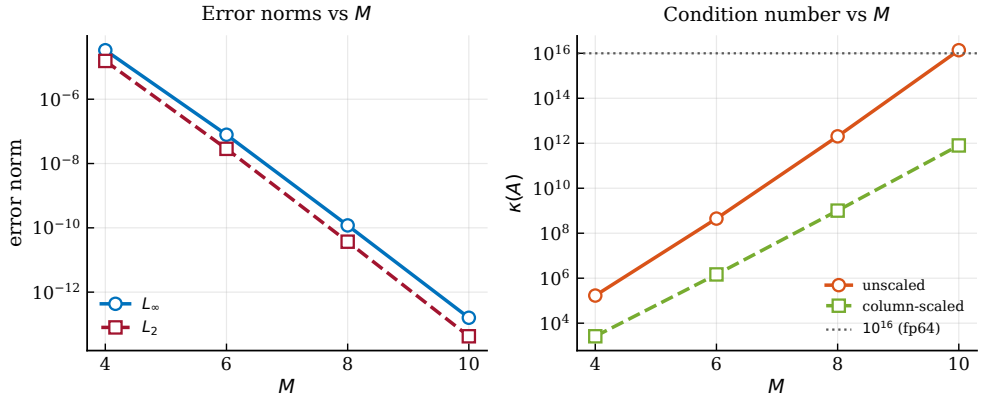


Figure 4. Error norms (**left**) and condition numbers (**right**) versus M for Example 1 with $\beta = 0.25$.

Table 3. L_2- and L_∞-norms, CPU time, and observed successive rates for Example 1 with $\beta = 0.25$. The rate from M to M' is $\varrho = \log(E_M/E_{M'})/\log(M'/M)$ with $E_M = \|v - v_{2M}\|_\infty$.

M	L_2	L_∞	CPU Time (s)	ϱ (L_∞)
4	1.5499×10^{-5}	3.3436×10^{-5}	0.008	—
6	2.8635×10^{-8}	7.8641×10^{-8}	0.029	14.93
8	3.7078×10^{-11}	1.1899×10^{-10}	0.078	22.57
10	4.2386×10^{-14}	1.5965×10^{-13}	0.181	29.64

Example 2. *We consider Equations (1) and (2) on the domain* $[0,1] \times (0,1]$, *with*

$$h(x,t) = 1,$$
$$K(x,t,s) = -x(t-s),$$
$$f(x,t) = (1-x^2)\left(\frac{t^{1-\beta}}{\Gamma(2-\beta)} + \Gamma(1+\beta)\right) + 2(t+t^\beta) + x(1-x^2)\left(\frac{t^3}{6} + \frac{t^{2+\beta}}{(1+\beta)(2+\beta)}\right),$$

exact solution $v(x,t) = (1-x^2)(t+t^\beta)$, *and initial and boundary conditions* $v(x,0) = 0$, $v(0,t) = t + t^\beta$, $v(1,t) = 0$ *[25]. The forcing term is consistent with this data for* $h(x,t) = 1$ *and* $K(x,t,s) = -x(t-s)$.

Reference results obtained by the Adomian decomposition method (ADM) with $n = 20$ *terms and the finite difference method (FDM) with* $M = N = 10$ *are taken from [25]; those reference solvers were not re-implemented here. The proposed Pell–Lucas collocation method is applied with* $M = 5$ *and* $\beta = 0.5, 0.7$, *using the same least-squares collocation solve with column equilibration as in Example 1. Table 4 compares solution values at selected sample points, and Table 5 reports the corresponding absolute errors. The comparisons with ADM and FDM are based on pointwise error values reported in the literature; corresponding CPU timings are unavailable or are not directly comparable across implementations and computing environments.*

Table 4. Comparison of the proposed method ($M = 5$) with ADM ($n = 20$) and FDM ($M = N = 10$) for selected (x,t) and $\beta = 0.5, 0.7$ in Example 2. ADM/FDM values are from [25].

(x,t)	$\beta = 0.5$				$\beta = 0.7$			
	Exact	**ADM**	**FDM**	**Proposed**	**Exact**	**ADM**	**FDM**	**Proposed**
$(0.9, 0.1)$	0.0791	0.0790	0.0748	0.0705	0.0569	0.0569	0.0538	0.0532
$(0.7, 0.3)$	0.4323	0.4241	0.4302	0.4358	0.3726	0.3706	0.3701	0.3754
$(0.5, 0.5)$	0.9053	0.8680	0.9042	0.9075	0.8367	0.8248	0.8353	0.8392
$(0.3, 0.7)$	1.3984	1.3218	1.3979	1.4015	1.3459	1.3166	1.3453	1.3476
$(0.1, 0.9)$	1.8302	1.7661	1.8301	1.8266	1.8106	1.7824	1.8105	1.8097

Figure 5 shows the exact and approximate solutions for $\beta = 0.5$, Figure 6 the absolute-error field, and Figure 7 a logarithmic comparison of absolute errors at the Table 5 nodes. Because the exact solution contains the weakly singular factor t^{β}, global polynomial accuracy is limited near $t = 0$; nevertheless, Table 6 and Figure 8 show that the L_2 and L_∞ norms decrease as M increases from 4 to 6, with modest CPU times.

Table 5. Absolute errors of ADM, FDM, and the proposed method ($M = 5$) for selected (x, t) and $\beta = 0.5, 0.7$ in Example 2. ADM/FDM errors are from [25].

(x,t)	$\beta = 0.5$			$\beta = 0.7$		
	ADM	**FDM**	**Proposed**	**ADM**	**FDM**	**Proposed**
$(0.8, 0.2)$	2.1713×10^{-3}	3.0136×10^{-3}	4.6057×10^{-4}	4.0422×10^{-4}	3.1136×10^{-3}	1.8780×10^{-3}
$(0.6, 0.4)$	1.9929×10^{-2}	1.5344×10^{-3}	3.0836×10^{-3}	5.5585×10^{-3}	1.8397×10^{-3}	2.8855×10^{-3}
$(0.4, 0.6)$	5.8106×10^{-2}	7.4680×10^{-4}	2.3809×10^{-3}	2.0409×10^{-2}	9.4712×10^{-4}	1.9256×10^{-3}
$(0.2, 0.8)$	8.3181×10^{-2}	2.5420×10^{-4}	1.6212×10^{-3}	4.4305×10^{-2}	3.6554×10^{-4}	1.1497×10^{-3}

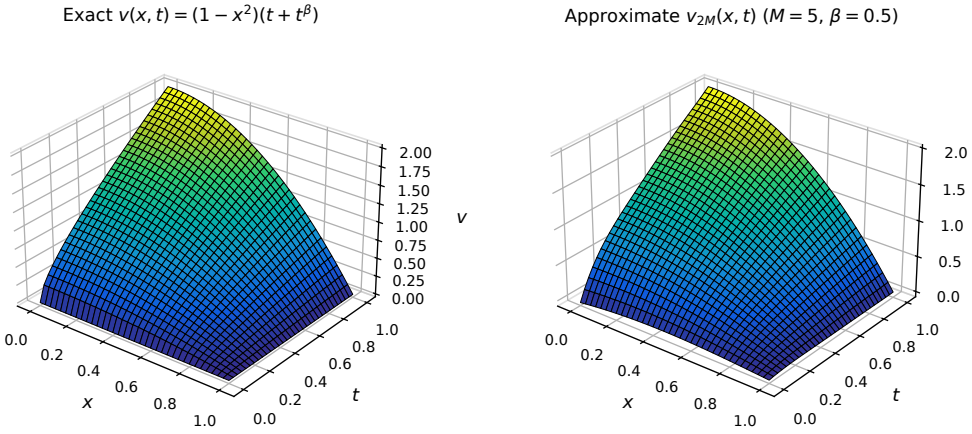


Figure 5. Exact and approximate solution surfaces for Example 2 with $M = 5$ and $\beta = 0.5$.

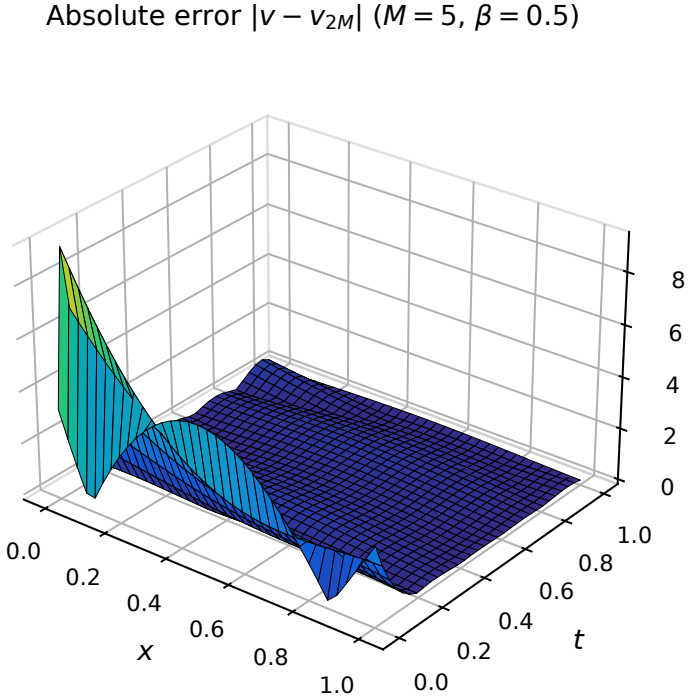


Figure 6. Absolute-error surface $|v - v_{2M}|$ for Example 2 with $M = 5$ and $\beta = 0.5$.

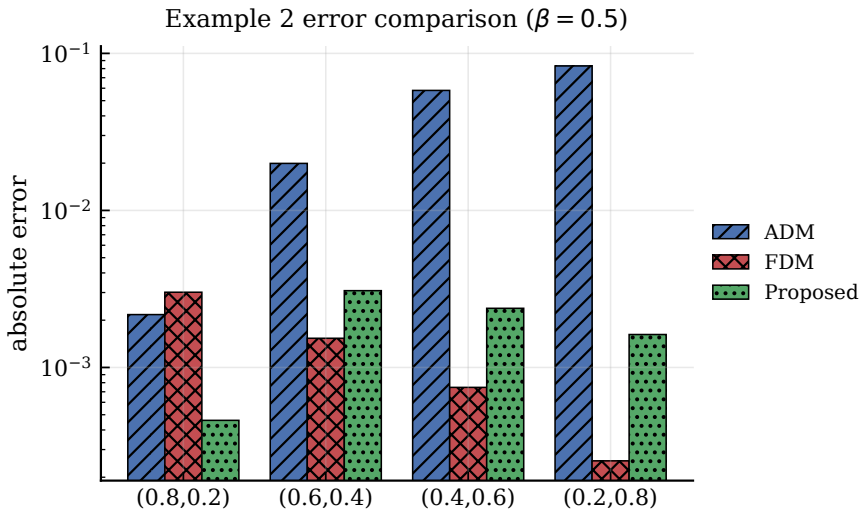


Figure 7. Absolute-error comparison at selected nodes for Example 2 ($\beta = 0.5$; ADM/FDM from [25]).

Table 6. L_2- and L_∞-norms and CPU time for Example 2.

M	β	L_2	L_∞	CPU Time (s)
4	0.5	1.5985×10^{-2}	1.0655×10^{-1}	0.007
5	0.5	1.2120×10^{-2}	9.6501×10^{-2}	0.016
6	0.5	9.8802×10^{-3}	8.6653×10^{-2}	0.026
4	0.7	7.8039×10^{-3}	5.2520×10^{-2}	0.007
5	0.7	6.8937×10^{-3}	4.9138×10^{-2}	0.014
6	0.7	6.3568×10^{-3}	4.6584×10^{-2}	0.025

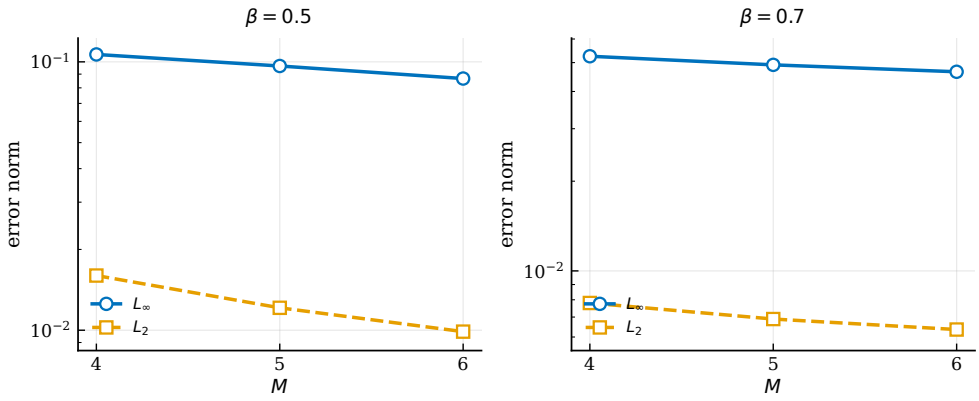


Figure 8. Global L_2 and L_∞ error norms versus M for Example 2.

8. Conclusions

This paper developed a Pell–Lucas collocation method for time-fractional partial integro-differential equations with Caputo fractional derivatives, spatial diffusion, and a Volterra memory term. The unknown solution was approximated by a truncated bivariate Pell–Lucas expansion, and operational matrices were used to represent the fractional-time derivative, spatial derivatives, and integral term. The governing equation together with the initial and boundary conditions was discretized at uniform collocation points, yielding a system of algebraic equations for the expansion coefficients.

An upper error bound in the infinity norm and a residual-based error-estimation procedure were also presented. In Example 1, the approximate solution agreed closely with the exact solution, and the reported error norms decreased systematically for $M = 4, 6, 8, 10$ after column equilibration. In Example 2, recomputed collocation values were compared

with ADM and FDM results taken from the literature; the method remains competitive at the reported sample points, while global norms improve mildly with M despite the limited regularity induced by t^{β}. Further study for additional equation classes and comparison settings remains a natural direction for future work.

Author Contributions: Conceptualization, J.D. and M.M.; methodology, J.D.; software, A.Y.; validation, A.Y.; formal analysis, A.Y.; investigation, A.Y.; data curation, S.M.; writing—original draft, J.D. and S.M.; writing—review and editing, J.D., A.Y., S.M. and M.M.; supervision, J.D. All authors contributed equally to the conception, development, preparation, and review of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No external datasets were used in this study. The numerical results are reported in the article, and the Python code used to generate them is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Appell, J.; Kalitvin, A.; Zabrejko, P. *Partial Integral Operators and Integro-Differential Equations: Pure and Applied Mathematics*; CRC Press: Boca Raton, FL, USA, 2000.
2. Grasselli, M.; Kabanikhin, S.; Lorenzi, A. An inverse hyperbolic integrodifferential problem arising in geophysics ii. *Nonlinear Anal. Theory Methods Appl.* **1990**, *15*, 283–298. [CrossRef]
3. Miller, R. An integrodifferential equation for rigid heat conductors with memory. *J. Math. Anal. Appl.* **1978**, *66*, 313–332. [CrossRef]
4. Pao, C. Solution of a nonlinear integrodifferential system arising in nuclear reactor dynamics. *J. Math. Anal. Appl.* **1974**, *48*, 470–492. [CrossRef]
5. Yüzbaşı, Ş.; Yıldırım, G. A collocation method to solve the parabolic-type partial integro-differential equations via pell–lucas polynomials. *Appl. Math. Comput.* **2022**, *421*, 126956. [CrossRef]
6. Yuldashev, T. Nonlocal boundary value problem for a nonlinear fredholm integro-differential equation with degenerate kernel. *Differ. Equ.* **2018**, *54*, 1646–1653. [CrossRef]
7. Dehghan, M.; Shakeri, F. Solution of parabolic integro-differential equations arising in heat conduction in materials with memory via he's variational iteration technique. *Int. J. Numer. Methods Biomed. Eng.* **2010**, *26*, 705–715. [CrossRef]
8. Fairweather, G. Spline collocation methods for a class of hyperbolic partial integro-differential equations. *SIAM J. Numer. Anal.* **1994**, *31*, 444–460. [CrossRef]
9. Samko, S. *Fractional Integrals and Derivatives: Theory and Applications*; Gordan and Breach Science: London, UK, 1993.
10. Mainardi, F. *Fractional Calculus: Some Basic Problems in Continuum and Statistical Mechanics*; Springer: Berlin/Heidelberg, Germany, 1997.
11. Kumar, D.; Baleanu, D. *Fractional Calculus and Its Applications in Physics*; Frontiers Media SA: Lausanne, Switzerland, 2019.
12. Diethelm, K.; Ford, N. *The Analysis of Fractional Differential Equations*; Lecture Notes in Mathematics; Springer: Berlin/Heidelberg, Germany, 2010.
13. Hilfer, R. *Applications of Fractional Calculus in Physics*; World Scientific: Singapore, 2000.
14. Podlubny, I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 198.
15. Abd Elaziz El-Sayed, A.; Boulaaras, S.; Sweilam, N. Numerical solution of the fractional-order logistic equation via the first-kind dickson polynomials and spectral tau method. *Math. Methods Appl. Sci.* **2023**, *46*, 8004–8017. [CrossRef]
16. Noeiaghdam, S.; Dreglea, A.; Işık, H.; Suleman, M. A comparative study between discrete stochastic arithmetic and floating-point arithmetic to validate the results of fractional order model of malaria infection. *Mathematics* **2021**, *9*, 1435. [CrossRef]
17. Ali, Z.; Nia, S.N.; Rabiei, F.; Shah, K.; Tan, M.K. A semianalytical approach to the solution of time-fractional navier-stokes equation. *Adv. Math. Phys.* **2021**, *2021*, 5547804. [CrossRef]
18. Atangana, A.; Secer, A. A note on fractional order derivatives and table of fractional derivatives of some special functions. *Abstr. Appl. Anal.* **2013**, *2013*, 279681. [CrossRef]

19. Noeiaghdam, S.; Micula, S.; Nieto, J.J. A novel technique to control the accuracy of a nonlinear fractional order model of covid-19: Application of the cestac method and the cadna library. *Mathematics* **2021**, *9*, 1321. [CrossRef]
20. Mainardi, F. *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*; World Scientific: Singapore, 2022.
21. Janmohammadi, A.; Damirchi, J.; Mahmoudi, S.M.; Esfandiari, A. Numerical solutions of inverse time fractional coupled burgers' equations by the chebyshev wavelet method. *J. Appl. Math. Comput.* **2022**, *68*, 2983–3009. [CrossRef]
22. Foadian, S.; Pourgholi, R.; Tabasi, S.H.; Damirchi, J. The inverse solution of the coupled nonlinear reaction–diffusion equations by the haar wavelets. *Int. J. Comput. Math.* **2019**, *96*, 105–125. [CrossRef]
23. Aslan, E.; Kürkçü, Ö.K.; Sezer, M. A fast numerical method for fractional partial integro-differential equations with spatial-time delays. *Appl. Numer. Math.* **2021**, *161*, 525–539. [CrossRef]
24. Yıldırım, G.; Yüzbaşı, Ş. A collocation technique via Pell-Lucas polynomials to solve fractional differential equation model for HIV/AIDS with treatment compartment. *Comput. Model. Eng. Sci.* **2024**, *141*, 281–310. [CrossRef]
25. Santra, S.; Mohapatra, J. A novel finite difference technique with error estimate for time fractional partial integro-differential equation of volterra type. *J. Comput. Appl. Math.* **2022**, *400*, 113746. [CrossRef]
26. Luo, Z.; Zhang, X.; Wang, S.; Yao, L. Numerical approximation of time fractional partial integro-differential equation based on compact finite difference scheme. *Chaos Solitons Fractals* **2022**, *161*, 112395. [CrossRef]
27. Jaradat, H.; Awawdeh, F.; Rawashdeh, E. Analytic solution of fractional integro-differential equations. *Ann. Univ. Craiova-Math. Comput. Sci. Ser.* **2011**, *38*, 1–10.
28. Hussain, A.K.; Rusli, N.; Fadhel, F.S.; Yahya, Z.R. Solution of one-dimensional fractional order partial integro-differential equations using variational iteration method. *AIP Conf. Proc.* **2016**, *1775*, 030096. [CrossRef]
29. Nawaz, Y. Variational iteration method and homotopy perturbation method for fourth-order fractional integro-differential equations. *Comput. Math. Appl.* **2011**, *61*, 2330–2341. [CrossRef]
30. Dehestani, H.; Ordokhani, Y.; Razzaghi, M. Pseudo-operational matrix method for the solution of variable-order fractional partial integro-differential equations. *Eng. Comput.* **2021**, *37*, 1791–1806. [CrossRef]
31. Abu Arqub, O. Computational algorithm for solving singular fredholm time-fractional partial integrodifferential equations with error estimates. *J. Appl. Math. Comput.* **2019**, *59*, 227–243. [CrossRef]
32. Aslefallah, M.; Shivanian, E. Nonlinear fractional integro-differential reaction-diffusion equation via radial basis functions. *Eur. Phys. J. Plus* **2015**, *130*, 47. [CrossRef]
33. He, J.H.; El-Dib, Y.O. A tutorial introduction to the two-scale fractal calculus and its application to the fractal Zhiber–Shabat oscillator. *Fractals* **2021**, *29*, 2150268. [CrossRef]
34. He, C.H.; He, J.H. Fractal, fractional calculus, and AI: Advanced tools driving innovation in mechanical engineering. *Facta Univ. Ser. Mech. Eng.* **2026**, *24*, 1–20. [CrossRef]
35. He, J.H. A short review on analytical methods for a fully fourth-order nonlinear integral boundary value problem with fractal derivatives. *Int. J. Numer. Methods Heat Fluid Flow* **2020**, *30*, 4933–4943. [CrossRef]
36. Yüzbaşı, Ş.; Yıldırım, G. Pell–Lucas collocation method for numerical solutions of two population models and residual correction. *J. Taibah Univ. Sci.* **2020**, *14*, 1262–1278. [CrossRef]
37. Abd-Elhameed, W.M.; Alshehri, H.M.; Alharbi, M.H.; Alqubori, O.M.; Amin, A.K.; Atta, A.G. Shifted Pell–Lucas polynomial-based algorithms for fractional and classical Rosenau–Hyman models. *Fractal Fract.* **2026**, *10*, 359. [CrossRef]
38. Sweilam, N.H.; Abou Hasan, M.M. Numerical approximation of Lévy–Feller fractional diffusion equation via Chebyshev–Legendre collocation method. *Eur. Phys. J. Plus* **2016**, *131*, 251. [CrossRef]
39. Sweilam, N.H.; Abou Hasan, M.M. An improved method for nonlinear variable-order Lévy–Feller advection–dispersion equation. *Bull. Malays. Math. Sci. Soc.* **2019**, *42*, 3021–3046. [CrossRef]
40. Sweilam, N.H.; Abou Hasan, M.M. Numerical studies for the fractional Schrödinger equation with the quantum Riesz–Feller derivative. *Prog. Fract. Differ. Appl.* **2016**, *2*, 231–245. [CrossRef]
41. Koshy, T.; Koshy, T. *Pell and Pell–Lucas Numbers*; Springer: Berlin/Heidelberg, Germany, 2014.
42. El-Sayed, A.A.E. Pell-lucas polynomials for numerical treatment of the nonlinear fractional-order duffing equation. *Demonstr. Math.* **2023**, *56*, 20220220. [CrossRef]
43. Yüzbaşı, Ş.; Yıldırım, G. Pell–Lucas collocation method for solving a class of second order nonlinear differential equations with variable delays. *Turk. J. Math.* **2023**, *47*, 37–55. [CrossRef]
44. Yüzbaşı, Ş.; Yıldırım, G. Pell–Lucas collocation method to solve high-order linear Fredholm–Volterra integro-differential equations and residual correction. *Turk. J. Math.* **2020**, *44*, 1065–1091. [CrossRef]
45. Jafari, H.; Ganji, R.; Salati, S.; Johnston, S. A mixed-method to numerical simulation of variable order stochastic advection diffusion equations. *Alex. Eng. J.* **2024**, *89*, 60–70. [CrossRef]
46. Zhou, T.; Ncube, M.N.; Salati, S.; Jafari, H. Operational matrices for solving a class of variable-order partial differential equations. *Fractals* **2025**, *33*, 2540130. [CrossRef]

47. Şahin, M.; Sezer, M. Pell-lucas collocation method for solving high-order functional differential equations with hybrid delays. *Celal Bayar Univ. J. Sci.* **2018**, *14*, 141–149. [CrossRef]
48. Horadam, A.; Mahon, J. Pell and pell-lucas polynomials. *Fibonacci Q.* **1985**, *23*, 7–20. [CrossRef]
49. Horadam, A.; Swita, B.; Filipponi, P. Integration and derivative sequences for pell and pell-lucas polynomials. *Fibonacci Q.* **1994**, *32*, 130–135. [CrossRef]
50. Abd-Elhameed, W.; Youssri, Y. Generalized lucas polynomial sequence approach for fractional differential equations. *Nonlinear Dyn.* **2017**, *89*, 1341–1355. [CrossRef]
51. Wang, W.; Wang, H. Some results on convolved (p, q)-fibonacci polynomials. *Integral Transform. Spec. Funct.* **2015**, *26*, 340–356. [CrossRef]
52. Meybodi, F.A.A.; Heydari, M.H.; Ghaini, F.M.M. Gaussian-radial basis functions for solving fractional parabolic partial integro-differential equations. *J. Math. Ext.* **2019**, *15*, 1–21.
53. Podlubny, I. Geometric and physical interpretation of fractional integration and fractional differentiation. *arXiv* **2001**, arXiv:math/0110241.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.