

Article

Two-Parameter General-Shift Lucas Polynomial Spectral Collocation for Time-Fractional Nonlinear Partial Integro-Differential Equations

Waleed Mohamed Abd-Elhameed ^{1,*} , Naher Mohammed A. Alsafri ² , Omar Mazen Alqubori ² ,
Amr Kamel Amin ³ , Mohammed Atef Abdallah ⁴  and Ahmed Gamal Atta ⁵ 

- ¹ Department of Mathematics, Faculty of Science, Cairo University, Giza 12613, Egypt
² Department of Mathematics and Statistics, College of Science, University of Jeddah, Jeddah 23831, Saudi Arabia; nmalsafri@uj.edu.sa (N.M.A.A.); ommohamad3@uj.edu.sa (O.M.A.)
³ Department of Mathematics, Adham University College, Umm Al-Qura University, Makkah 28653, Saudi Arabia; akgadelrab@uqu.edu.sa
⁴ Department of Mathematics, Al-Lith University College, Umm Al-Qura University, Al-Lith 24382, Saudi Arabia; maabdallah@uqu.edu.sa
⁵ Department of Mathematics, Faculty of Education, Ain Shams University, Roxy, Cairo 11341, Egypt; ahmed_gamal@edu.asu.edu.eg
* Correspondence: waleed@cu.edu.eg

Abstract

The present study introduces a two-parameter general-shift Lucas basis from the classical Lucas family via an affine argument transformation, which then forms the basis of a spectral collocation approach for the numerical treatment of nonlinear time-fractional partial integro-differential equations. This parametrized development of the ordinary Lucas basis provides a unified framework for both the classical Lucas polynomials and the standard shifted forms of the Lucas basis. We obtain closed-form power and inverse-power representations in terminating hypergeometric form, which are used to derive the operational matrices of derivatives for ordinary and Caputo fractional differentiation. A tensor-product expansion discretizes the space–time variables. Quadrature approximation is applied to the Volterra integral using Gauss–Legendre quadrature, while collocation of the governing equation, along with its underlying conditions, leads to a system of nonlinear algebraic equations that can be handled by Newton’s method. To evaluate the performance of our proposed basis in terms of accuracy, sensitivity to parameters, and computational efficiency, we provide error bounds and conduct numerical experiments. Our results demonstrate that our proposed basis achieves high levels of accuracy while offering the flexibility provided by its two affine parameters.

Keywords: Lucas polynomials; time-fractional partial integro-differential equations; collocation method; operational matrices; Caputo fractional derivative; Gauss–Legendre quadrature

MSC: 35R11; 65M70; 33C45



Academic Editor: Dimitris F. Papadopoulos

Received: 23 September 2026

Revised: 20 October 2026

Accepted: 22 October 2026

Published: 26 October 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Fractional differential equations prove particularly effective at representing memory or nonlocal temporal dependencies. Because they depend on the solution’s history, fractional derivatives require a more sophisticated discretization than integer-order counterparts.

References that may provide further insight into this topic include [1–4]. There have been several new approaches to the numerical solution of fractional differential equations. These include using barycentric collocation in conjunction with Gauss quadrature [5]; using high-order predictor–corrector algorithms on non-uniform grids for initially singular solutions [6]; utilizing Gegenbauer-wavelet Tau schemes [7]; employing Jacobi–Galerkin discretizations [8]; and employing non-uniform L1 spectral element formulations [9]. Additionally, a shifted generalized Chebyshev basis was used to treat a fractional gas dynamics model [10]. A recent alternative for rational-order fractional differential equations combines rational-power functions with rational Taylor expansions and operational matrices; that study also proves convergence directly for the resulting approximation [11].

The standard Lucas sequence polynomials and their modified counterparts also give rise to several polynomial bases used in computational mathematics. They can serve as basis functions in various models in the applied science. For example, classical Lucas polynomials have been used to construct operational matrices and a collocation procedure for multidimensional stochastic Itô–Volterra integral equations [12]. Generalized Lucas wavelets with two free parameters, together with a pseudo-operational matrix and a Ritz procedure, were developed for nonlinear fractal–fractional optimal control problems [13]. An orthonormal piecewise modification of the Lucas basis (Vieta–Lucas functions) was introduced for one- and two-dimensional fractional Galilei-invariant advection–diffusion equations [14], while a shifted variant of these modified Lucas-type polynomials was applied to variable-order fractional reaction–advection–diffusion equations [15]. Generalized Lucas wavelets were also combined with least-squares and operational-matrix techniques for fractional optimal control problems [16], and a spectral collocation scheme built from Vieta–Lucas functions was formulated for a variable-order fractional Gray–Scott model [17]. Fractional shifted Lucas polynomials were introduced for multi-order and multi-term FDEs [18]; Lucas operational matrices were applied to the fractional Klein–Gordon equation [19]; and a Lucas spectral matrix collocation scheme was employed for Bratu, Riccati, and higher-order delay differential equations [20]. The approximation properties of Lucas wavelets were further compared with Chebyshev and Legendre wavelets [21], and generalized Lucas wavelets were recently incorporated into an operational-matrix method for certain fractional integro-differential optimal control problems [22]. Shifted Pell–Lucas polynomials were also employed in numerical algorithms for two Rosenau–Hyman models [23]. These numerical developments motivate the consideration of the more flexible linear transformation used in the present paper.

Lucas, generalized Lucas, Vieta–Lucas, and other related bases have already appeared in fractional differential, integral, and optimal-control computations. Nevertheless, those formulations are predominantly built on a fixed classical, shifted, or specially tailored Lucas-type basis. In general, the nonlinear time-fractional partial integro-differential class that is being studied in this work does not seem to have been studied previously with a two-parameter affine Lucas basis that can simultaneously recover both the traditional form of the Lucas family and the classic shifted form. Our new representations of the Lucas polynomials (which include an explicit power series, its inverse representation, integer-order operational relations, and Caputo fractional operational relations) will play an essential role in our research. The current development is designed to provide a means to study the properties of the nonlinear time-fractional partial integro-differential classes using a two-parameter affine Lucas basis, as the two affine parameters change the basis while maintaining the basic recurrence and algebraic properties of the Lucas family.

This study uses Lucas polynomials for their simple recurrence relation, closed algebraic forms, and easy differentiation, useful in creating operational matrices. We consider the affine transformation of the argument as $L_n(c\vartheta + d)$, with $c \neq 0$ and $d \in \mathbb{R}$, which maintains

the polynomials' key features while allowing basis adjustment. This enables the function class to include both the standard and shifted Lucas bases, adaptable during spectral discretization without altering the overall algorithm.

In a fractional integro-differential model, the fractional derivative and the integral term introduce distinct forms of nonlocal dependence. Such coupled operators arise, for example, when hereditary effects must be described alongside transport or diffusion, and they are encountered in physical, biological and financial applications. Time-fractional partial integro-differential equations arising in Black–Scholes jump-diffusion frameworks were investigated in [24]. Fractional Volterra equations were used in the setting of tumor-immune interactions [25]. In viscoelastic contexts, variable-order integro-partial differential equations with weakly singular kernels have been investigated [26]. Exact solutions are seldom accessible when these equations combine a memory integral with nonlinear or spatially varying terms; accordingly, a range of discretization strategies has been investigated. For weakly singular nonlinear time-fractional partial integro-differential equations, a parametric finite-difference and orthogonal-spline method was introduced [27], and a second-order method based on L1-2 approximation, central differences, and product trapezoidal quadrature was established for time-fractional Volterra equations [28]. Cubic B-spline collocation was utilized with L1 and L1-2 approximations for reaction–convection–diffusion Volterra problems [29]; hexic shifted Chebyshev polynomials were employed for variable-order fractional partial integro-differential equations [30]; and shifted Chebyshev projection methodologies were formulated for Caputo fractional differential and integro-differential frameworks [31]. Recently, concurrent space–time Hermite wavelets were utilized in nonlinear fractional integro-partial differential equations involving weakly singular Volterra and Fredholm operators [32]; bidimensional Gegenbauer polynomials were employed to formulate pseudo-operational matrices for variable-order issues [33]; a finite-block spectral collocation method was devised for nonlinear fractional partial integro-differential equations [34]; and high-order Legendre spectral methods were examined for nonlocal weakly singular integro-PDEs [35].

The method is formulated for nonlinear time-fractional partial integro-differential equations that couple a Caputo derivative in time with a variable-coefficient second spatial derivative and a nonlinear Volterra contribution, subject to initial and boundary data. Thus, temporal memory, spatial diffusion, and history-dependent nonlocality are treated within one computational model.

More specifically, we consider nonlinear time-fractional partial integro-differential equations of the form

$$\mathcal{D}_t^\eta y(\vartheta, t) = \sigma(\vartheta, t) \frac{\partial^2 y(\vartheta, t)}{\partial \vartheta^2} + \int_0^t \mathcal{K}(\vartheta, t, r, y(\vartheta, r)) dr + g(\vartheta, t), \quad 0 < \eta \leq 1, \quad (1)$$

subject to an initial condition and two boundary conditions in the spatial variable. Here, $\mathcal{D}_t^\eta$ represents the fractional derivative in the Caputo sense, $\sigma(\vartheta, t)$ is a variable coefficient, $\mathcal{K}$ is a nonlinear Volterra kernel, and $g(\vartheta, t)$ is a prescribed source term.

The proposed model is an example of an evolutionary process where diffusion is combined with memory and historical accumulation effects. The Caputo operator models the effect of time-based history, while the Volterra term models the effect of past states. The spatial variation is modeled by the coefficient $\sigma(\vartheta, t)$, and the nonlinear kernel $\mathcal{K}$ allows for the modeling of interaction based on previous history. Examples of similar models can be found in viscoelasticity, anomalous transport, heat conduction with memory, biology with hereditary responses, and other nonlocal evolution problems. Thus, this equation provides a wide range of applications to test the validity of the proposed approximation method.

A spectral approximation represents the solution in a finite span of globally defined functions. For sufficiently regular solutions, increasing the span can yield rapid error decay; see for instance [36–38]. Tau, collocation, and Galerkin schemes differ chiefly in the way the residual is constrained. A Tau discretization reserves selected algebraic equations for the prescribed initial or boundary conditions. In this respect, the authors of [39] developed the fractional Romanovski–Jacobi Tau method for time-fractional problems with non-smooth solutions. A collocation discretization instead makes the residual vanish at a chosen collection of points. Examples include the adaptive rational Chebyshev collocation [40] and generalized spectral collocation with adjustable precision for variable-order FDEs [41]. A shifted-Dickson spectral collocation scheme was also proposed for the time-fractional Huxley equation [42]. A collocation algorithm based on shifted Pell polynomials was also developed for third-order Rosenau–Hyman and Gilson–Pickering equations [43]. In a Galerkin construction, orthogonality against the selected test functions supplies the residual equations. Typical examples are the Chebyshev–Legendre nonlinear Galerkin method [44], spectral–Galerkin methods for variable-coefficient fractional PDEs [45], and the Fibonacci-coefficient Galerkin method for another fractional model [46].

The main objective of the present study is to define and investigate the two-parameter general-shift Lucas polynomials (GSLPs)

$$\mathcal{L}_n^{c,d}(\vartheta) = L_n(c\vartheta + d), \quad c \neq 0,$$

as an adaptable basis for spectral discretization. The affine argument $c\vartheta + d$ contains the ordinary Lucas case when $(c, d) = (1, 0)$ and the familiar shifted basis $L_n(2\vartheta - 1)$ when $(c, d) = (2, -1)$. In addition to series form and its inverse formula, we establish the operational matrices for integer and fractional derivatives. These ingredients are then assembled into a tensor-product collocation algorithm for nonlinear time-fractional partial integro-differential equations. Gauss–Legendre quadrature is used for the Volterra contribution, while Newton iteration is applied to the nonlinear algebraic equations generated by the collocation conditions.

The main contributions of the present study can be listed in the following items:

- A two-parameter general-shift Lucas family $\mathcal{L}_n^{c,d}(\vartheta) = L_n(c\vartheta + d)$ is introduced, containing the classical and standard shifted Lucas polynomials as special cases.
- Explicit direct- and inverse-power representations are derived, together with integer-order and Caputo fractional differentiation formulas and their associated operational matrices.
- The developed formulas are utilized to create a tensor-product spectral collocation method for solving nonlinear time-fractional partial integro-differential equations.
- The nonlinear Volterra integral will be evaluated with Gauss–Legendre quadrature, while the resulting nonlinear algebraic system will be solved via Newton iteration.
- The approximation properties of the general-shift Lucas expansion are studied, and several numerical experiments were conducted to evaluate both the approximation capability and versatility of the developed method.

The outline of how our results will be organized is as follows. First, Section 2 presents the basic concepts of fractional calculus used throughout this work. Also, in Section 2, we discuss several standard Lucas identities that are used in developing other formulas. Next, in Section 3, we develop formulas for the analytical, inverse, integer-derivative, fractional-derivative, and operational properties of the general-shift Lucas family. Then, in Section 4, we develop a spectral collocation algorithm for solving the nonlinear time-fractional partial integro-differential equation. Following that, in Section 5, we find an upper bound on the general-shift Lucas basis. Finally, in Section 6, we conduct numerous experiments and

compare the results. Lastly, in Section 8, we conclude the paper and outline possible areas for future research.

2. Fundamental and Essential Preliminaries

This section provides an overview of definitions and useful identities needed later. We first recall some useful Lucas properties used in the derivations, then introduce the two-parameter affine basis, and finally state the Caputo definition together with its action on monomials.

2.1. An Overview of Lucas Polynomials

The classical sequence $\{L_k(\vartheta)\}_{k \geq 0}$ is linked to the Fibonacci polynomials; see [47]. They satisfy the recurrence relation

$$L_k(\vartheta) = \begin{cases} 2, & k = 0, \\ \vartheta, & k = 1, \\ \vartheta L_{k-1}(\vartheta) + L_{k-2}(\vartheta), & k \geq 2. \end{cases} \quad (2)$$

The coefficient calculations below rely on the following finite Lucas expansion:

$$L_k(\vartheta) = \sum_{j=0}^{\lfloor k/2 \rfloor} \frac{k}{k-j} \binom{k-j}{j} \vartheta^{k-2j}, \quad k \geq 1. \quad (3)$$

The corresponding generating function can be written as

$$\sum_{k=0}^{\infty} L_k(\vartheta) t^k = \frac{2 - \vartheta t}{1 - \vartheta t - t^2}. \quad (4)$$

An equivalent closed expression is given by the Binet representation

$$L_k(\vartheta) = \left(\frac{\vartheta + \sqrt{\vartheta^2 + 4}}{2} \right)^k + \left(\frac{\vartheta - \sqrt{\vartheta^2 + 4}}{2} \right)^k. \quad (5)$$

The inversion formula expressing the monomial ϑ^k in terms of Lucas polynomials is

$$\vartheta^k = \sum_{j=0}^{\lfloor k/2 \rfloor} (-1)^j \epsilon_{k-2j} \binom{k}{j} L_{k-2j}(\vartheta), \quad k \geq 0, \quad (6)$$

where

$$\epsilon_\ell = \begin{cases} \frac{1}{2}, & \ell = 0, \\ 1, & \ell > 0. \end{cases} \quad (7)$$

The series representation (3), the Binet formula (5), and the inversion formula (6) will be used repeatedly below.

2.2. Caputo's Fractional Derivative

Definition 1. Let $Q \in C^k([0, T])$, where $\kappa = \lceil \eta \rceil$ for $\eta > 0$. The Caputo fractional derivative of order η for $\vartheta > 0$ is defined as [1]

$$\mathcal{D}_{\vartheta}^{\eta} Q(\vartheta) = \begin{cases} \frac{1}{\Gamma(\kappa - \eta)} \int_0^{\vartheta} (\vartheta - t)^{\kappa - \eta - 1} Q^{(\kappa)}(t) dt, & \text{if } \kappa - 1 < \eta < \kappa \quad (\eta \notin \mathbb{N}), \\ \frac{d^{\kappa} Q(\vartheta)}{d\vartheta^{\kappa}}, & \text{if } \eta = \kappa \in \mathbb{N}. \end{cases} \quad (8)$$

For constants and nonnegative integer monomials, the Caputo rules reduce to

$$\mathcal{D}_{\vartheta}^{\eta} C = 0, \quad (C \text{ is a constant}), \quad (9)$$

$$\mathcal{D}_{\vartheta}^{\eta} \vartheta^q = \begin{cases} 0, & \text{if } q \in \mathbb{N}_0 \quad \text{and} \quad q < \lceil \eta \rceil, \\ \frac{q!}{\Gamma(q+1-\eta)} \vartheta^{q-\eta}, & \text{if } q \in \mathbb{N}_0 \quad \text{and} \quad q \geq \lceil \eta \rceil, \end{cases} \quad (10)$$

where $\mathbb{N} = \{1, 2, \dots\}$, $\mathbb{N}_0 = \{0, 1, 2, \dots\}$, and $\lceil \eta \rceil$ represents the ceiling function.

2.3. General-Shift Lucas Polynomials and Their Special Cases

The basis functions are generated by evaluating the standard Lucas polynomials at an affine argument. For $n \in \mathbb{N}_0$, define the general-shift Lucas polynomials (GSLPs) by

$$\mathcal{L}_n^{c,d}(\vartheta) = L_n(c\vartheta + d), \quad c \neq 0. \quad (11)$$

The two principal special cases are

- $\mathcal{L}_n^{1,0}(\vartheta) = L_n(\vartheta)$, which reduces to the standard Lucas polynomials.
- $\mathcal{L}_n^{2,-1}(\vartheta) = L_n(2\vartheta - 1)$, which reduces to the standard shifted Lucas polynomials on $[0, 1]$.

From now on, the notation $\mathcal{L}_n^{c,d}$ is used consistently for GSLPs, while L_n is retained for classical Lucas polynomials.

3. Some New Formulas for the General-Shift Lucas Polynomials

We will derive the following pivotal formulas in the derivation of our proposed algorithm:

- The power representation of the GSLPs and its coefficient form.
- The inverse-power representation and its coefficient form.
- The integer-order derivative formula and the corresponding operational matrix.
- The Caputo fractional-derivative formula and its operational representation.

3.1. Power and Inversion Formulas

The hypergeometric coefficient expressions below use $d \neq 0$. This restriction applies only to that coefficient parametrization: direct finite-sum identities are given separately when $d = 0$.

Theorem 1. Let $r \in \mathbb{N}$, and let $c \neq 0$ and $d \neq 0$. Then

$$\mathcal{L}_r^{c,d}(\vartheta) = \sum_{m=0}^r c^m d^{r-m} \binom{r}{m} {}_2F_1 \left(\begin{matrix} \frac{m-r}{2}, \frac{m-r+1}{2} \\ 1-r \end{matrix} \middle| -\frac{4}{d^2} \right) \vartheta^m. \quad (12)$$

Proof. From (3), with $y = c\vartheta + d$,

$$\mathcal{L}_r^{c,d}(\vartheta) = \sum_{j=0}^{\lfloor r/2 \rfloor} \frac{r}{r-j} \binom{r-j}{j} (c\vartheta + d)^{r-2j}.$$

Expand the affine power by the binomial theorem and collect like powers of ϑ . The resulting coefficient representation is

$$\mathcal{L}_r^{c,d}(\vartheta) = \sum_{m=0}^r c^m d^{r-m} S_{r,m} \vartheta^m,$$

where

$$S_{r,m} = \sum_{j=0}^{\lfloor (r-m)/2 \rfloor} \frac{r(r-j-1)!}{j! m! (r-m-2j)!} d^{-2j}.$$

The factorials in this coefficient may be converted to Pochhammer symbols using

$$\left(\frac{m-r}{2}\right)_j \left(\frac{m-r+1}{2}\right)_j = \frac{(r-m)!}{4^j (r-m-2j)!}, \quad (1-r)_j = (-1)^j \frac{(r-1)!}{(r-j-1)!}.$$

Upon applying these relations to each summand, we obtain

$$\frac{r(r-j-1)!}{j! m! (r-m-2j)!} d^{-2j} = \binom{r}{m} \frac{\left(\frac{m-r}{2}\right)_j \left(\frac{m-r+1}{2}\right)_j}{(1-r)_j j!} \left(-\frac{4}{d^2}\right)^j.$$

Hence,

$$S_{r,m} = \binom{r}{m} {}_2F_1\left(\frac{m-r}{2}, \frac{m-r+1}{2} \middle| 1-r; -\frac{4}{d^2}\right).$$

Substituting this expression for $S_{r,m}$ into the collected-power expansion establishes (12). $\square$

Remark 1. In (12), the terminating hypergeometric series has final summation index

$$j = \left\lfloor \frac{r-m}{2} \right\rfloor,$$

since one of its two upper parameters is a non-positive integer.

Remark 2. When $d = 0$, substitution into the finite Lucas series gives a formula without a singular parameter. In particular, $c = 1$ returns the classical basis. For $r \geq 1$,

$$\mathcal{L}_r^{c,0}(\vartheta) = \sum_{j=0}^{\lfloor r/2 \rfloor} \frac{r}{r-j} \binom{r-j}{j} c^{r-2j} \vartheta^{r-2j}. \quad (13)$$

Accordingly, the exclusion of $d = 0$ in (12) does not exclude that case from the definition of the basis.

Theorem 2. Let $r \in \mathbb{N}_0$, and let $c \neq 0$ and $d \neq 0$. Then

$$\vartheta^r = \sum_{m=0}^r \epsilon_m (-1)^{r-m} c^{-r} d^{r-m} \binom{r}{m} {}_2F_1\left(\frac{m-r}{2}, \frac{m-r+1}{2} \middle| m+1; -\frac{4}{d^2}\right) \mathcal{L}_m^{c,d}(\vartheta). \quad (14)$$

Proof. For the inverse transformation, start from the classical Lucas expansion of a monomial:

$$y^k = \sum_{j=0}^{\lfloor k/2 \rfloor} (-1)^j \epsilon_{k-2j} \binom{k}{j} L_{k-2j}(y),$$

and put $y = c\vartheta + d$. Since $\vartheta = c^{-1}(y - d)$,

$$\vartheta^r = c^{-r} \sum_{k=0}^r \binom{r}{k} (-d)^{r-k} y^k.$$

Replacing each power of the affine argument by its Lucas expansion produces

$$\vartheta^r = c^{-r} \sum_{k=0}^r \binom{r}{k} (-d)^{r-k} \sum_{j=0}^{\lfloor k/2 \rfloor} (-1)^j \epsilon_{k-2j} \binom{k}{j} L_{k-2j}(y).$$

Collect the terms multiplying the same Lucas polynomial by setting $k = m + 2j$. By reversing the finite summation order, we obtain

$$\vartheta^r = \sum_{m=0}^r \epsilon_m (-1)^{r-m} c^{-r} d^{r-m} T_{r,m} L_m(y),$$

where

$$T_{r,m} = \sum_{j=0}^{\lfloor (r-m)/2 \rfloor} (-1)^j \frac{r!}{j!(m+j)!(r-m-2j)!} d^{-2j}.$$

For the hypergeometric coefficient, the required Pochhammer–factorial conversions are

$$\left(\frac{m-r}{2}\right)_j \left(\frac{m-r+1}{2}\right)_j = \frac{(r-m)!}{4^j (r-m-2j)!}, \quad (m+1)_j = \frac{(m+j)!}{m!}.$$

With these conversions, a summand of $T_{r,m}$ becomes

$$(-1)^j \frac{r!}{j!(m+j)!(r-m-2j)!} d^{-2j} = \binom{r}{m} \frac{\left(\frac{m-r}{2}\right)_j \left(\frac{m-r+1}{2}\right)_j}{(m+1)_j j!} \left(-\frac{4}{d^2}\right)^j.$$

Thus,

$$T_{r,m} = \binom{r}{m} {}_2F_1\left(\frac{m-r}{2}, \frac{m-r+1}{2} \middle| -\frac{4}{d^2}\right).$$

Finally, $L_m(y) = L_m(c\vartheta + d) = \mathcal{L}_m^{c,d}(\vartheta)$, and the stated inversion formula follows. $\square$

Remark 3. The hypergeometric function in (14) is terminating, since one of

$$\frac{m-r}{2}, \quad \frac{m-r+1}{2},$$

is a non-positive integer.

Remark 4. When $d = 0$, the inversion formula has the nonsingular form

$$\vartheta^r = c^{-r} \sum_{j=0}^{\lfloor r/2 \rfloor} (-1)^j \epsilon_{r-2j} \binom{r}{j} \mathcal{L}_{r-2j}^{c,0}(\vartheta), \quad r \geq 0. \quad (15)$$

Choosing $c = 1$ recovers the usual inversion identity for the unshifted Lucas sequence.

Remark 5. For later matrix assembly, write Theorem 1 by collecting the monomial coefficients:

$$\mathcal{L}_i^{c,d}(\vartheta) = \sum_{r=0}^i M_{r,i} \vartheta^r, \quad (16)$$

where $M_{0,0} = 2$. For $i \geq 1$ and $d \neq 0$, the coefficients are given by

$$M_{r,i} = c^r d^{i-r} \binom{i}{r} {}_2F_1\left(\frac{r-i}{2}, \frac{r-i+1}{2} \middle| -\frac{4}{d^2}\right), \quad 0 \leq r \leq i. \quad (17)$$

For $d = 0$, the corresponding nonsingular coefficient form is

$$M_{r,i} = \begin{cases} \frac{i}{i-\ell} \binom{i-\ell}{\ell} c^r, & i \geq 1, \quad r = i - 2\ell, \quad 0 \leq \ell \leq \lfloor i/2 \rfloor, \\ 0, & \text{otherwise.} \end{cases} \quad (18)$$

This coefficient representation will be useful in constructing the operational formulas developed in the subsequent results.

Remark 6. The inverse transformation admits a parallel coefficient representation, obtained from Theorem 2:

$$\vartheta^j = \sum_{s=0}^j H_{s,j} \mathcal{L}_s^{c,d}(\vartheta), \quad (19)$$

where, for $d \neq 0$,

$$H_{s,j} = \epsilon_s (-1)^{j-s} c^{-j} d^{j-s} \binom{j}{s} {}_2F_1 \left(\frac{s-j}{2}, \frac{s-j+1}{2} \middle| -\frac{4}{d^2} \right), \quad 0 \leq s \leq j. \quad (20)$$

For $d = 0$, the coefficients reduce to the nonsingular form

$$H_{s,j} = \begin{cases} c^{-j} (-1)^\ell \epsilon_s \binom{j}{\ell}, & s = j - 2\ell, \quad 0 \leq \ell \leq \lfloor j/2 \rfloor, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

This representation will be used together with (16) in deriving the fractional-order differentiation formula.

3.2. Integer-Order Derivative Formulas

The differentiation result for the affine basis follows from the classical Lucas derivative expansion. We therefore establish that expansion first.

Theorem 3. Let $k, r \in \mathbb{N}$ with $1 \leq r \leq k$. Then, the r th derivative of the Lucas polynomial $L_k(\vartheta)$ may be expended in Lucas polynomials as

$$\frac{d^r}{d\vartheta^r} L_k(\vartheta) = k \sum_{j=0}^{\lfloor \frac{k-r}{2} \rfloor} (-1)^j \epsilon_{k-r-2j} \binom{r+j-1}{j} \frac{(k-j-1)!}{(k-r-j)!} L_{k-r-2j}(\vartheta), \quad (22)$$

where ϵ_ℓ is defined in (7).

Proof. The calculation starts with the generating series of the classical Lucas sequence:

$$\sum_{k=0}^{\infty} L_k(\vartheta) t^k = 1 + \frac{1+t^2}{1-\vartheta t - t^2}. \quad (23)$$

Applying r ordinary derivatives in ϑ to (23) gives

$$\sum_{k=r}^{\infty} \frac{d^r}{d\vartheta^r} L_k(\vartheta) t^k = \frac{r! t^r (1+t^2)}{(1-\vartheta t - t^2)^{r+1}}. \quad (24)$$

Set

$$\rho_1 = \frac{\vartheta + \sqrt{\vartheta^2 + 4}}{2}, \quad \rho_2 = \frac{\vartheta - \sqrt{\vartheta^2 + 4}}{2}. \quad (25)$$

Then

$$1 - \vartheta t - t^2 = (1 - \lambda_1 t)(1 - \lambda_2 t), \quad \lambda_1 \lambda_2 = -1. \quad (26)$$

By the generalized binomial theorem,

$$(1 - \lambda_1 t)^{-r-1} = \sum_{s=0}^{\infty} \binom{r+s}{s} \lambda_1^s t^s, \quad (27)$$

and

$$(1 - \lambda_2 t)^{-r-1} = \sum_{q=0}^{\infty} \binom{r+q}{q} \lambda_2^q t^q. \quad (28)$$

Therefore,

$$(1 - \vartheta t - t^2)^{-r-1} = \left(\sum_{s=0}^{\infty} \binom{r+s}{s} \lambda_1^s t^s \right) \left(\sum_{q=0}^{\infty} \binom{r+q}{q} \lambda_2^q t^q \right). \quad (29)$$

Let

$$A_N = [t^N](1 - \vartheta t - t^2)^{-r-1}. \quad (30)$$

A contribution to the coefficient of t^N occurs precisely when

$$s + q = N. \quad (31)$$

Taking

$$q = j, \quad s = N - j, \quad (32)$$

we obtain

$$A_N = \sum_{j=0}^N \binom{r+N-j}{N-j} \binom{r+j}{j} \lambda_1^{N-j} \lambda_2^j. \quad (33)$$

Pairing the j th and $(N-j)$ th terms in (33), and using $\lambda_1 \lambda_2 = -1$, gives

$$\lambda_1^{N-j} \lambda_2^j + \lambda_1^j \lambda_2^{N-j} = (-1)^j (\lambda_1^{N-2j} + \lambda_2^{N-2j}). \quad (34)$$

Since

$$L_m(\vartheta) = \lambda_1^m + \lambda_2^m, \quad (35)$$

it follows that

$$A_N = \sum_{j=0}^{\lfloor N/2 \rfloor} (-1)^j \epsilon_{N-2j} \binom{r+N-j}{N-j} \binom{r+j}{j} L_{N-2j}(\vartheta). \quad (36)$$

Moreover,

$$[t^N](1 + t^2)(1 - \vartheta t - t^2)^{-r-1} = A_N + A_{N-2}, \quad (37)$$

where $A_m = 0$ for $m < 0$. From (36), the coefficient of $L_{N-2j}(\vartheta)$ in (37) is

$$(-1)^j \epsilon_{N-2j} \left[\binom{r+N-j}{N-j} \binom{r+j}{j} - \binom{r+N-j-1}{N-j-1} \binom{r+j-1}{j-1} \right], \quad (38)$$

where the second term is taken to be zero for $j = 0$.

Since the factor t^r appears in (24), we set

$$N = k - r. \quad (39)$$

A direct simplification gives

$$r! \left[\binom{k-j}{k-r-j} \binom{r+j}{j} - \binom{k-j-1}{k-r-j-1} \binom{r+j-1}{j-1} \right] = k \binom{r+j-1}{j} \frac{(k-j-1)!}{(k-r-j)!}. \quad (40)$$

Identifying the t^k coefficient on both sides of (24) therefore gives

$$\frac{d^r}{d\vartheta^r} L_k(\vartheta) = k \sum_{j=0}^{\lfloor \frac{k-r}{2} \rfloor} (-1)^j \epsilon_{k-r-2j} \binom{r+j-1}{j} \frac{(k-j-1)!}{(k-r-j)!} L_{k-r-2j}(\vartheta). \quad (41)$$

The proof is therefore complete. $\square$

Theorem 4. Let $i, s \in \mathbb{N}$ with $i \geq s$. We have

$$D^s \mathcal{L}_i^{c,d}(\vartheta) = \sum_{n=0}^{i-s} \rho_{i,n}^s \mathcal{L}_n^{c,d}(\vartheta), \quad (42)$$

where

$$\rho_{i,n}^s = \begin{cases} c^s i (-1)^{\frac{i-s-n}{2}} \epsilon_n \binom{\frac{i+s-n}{2}-1}{\frac{i-s-n}{2}} \frac{(\frac{i+s+n}{2}-1)!}{(\frac{i-s+n}{2})!}, & 0 \leq n \leq i-s, \quad i-s-n \text{ even}, \\ 0, & \text{otherwise.} \end{cases} \quad (43)$$

Here, ϵ_n is defined in (7).

Proof. By the chain rule,

$$D^s \mathcal{L}_i^{c,d}(\vartheta) = c^s \frac{d^s}{dy^s} L_i(y) \Big|_{y=c\vartheta+d}.$$

Applying Theorem 3 and using $L_n(c\vartheta + d) = \mathcal{L}_n^{c,d}(\vartheta)$ gives the required expansion. Re-indexing with $n = i - s - 2\ell$ and substituting $\ell = (i - s - n)/2$ yields the coefficient $\rho_{i,n}^s$ displayed above. $\square$

Corollary 1. As a consequence of Theorem 4, considering the vector

$$\mathcal{L}^{c,d}(\vartheta) = [\mathcal{L}_0^{c,d}(\vartheta), \mathcal{L}_1^{c,d}(\vartheta), \dots, \mathcal{L}_{\mathfrak{L}}^{c,d}(\vartheta)]^T, \quad (44)$$

In matrix form, the derivatives of order s of the vector $\mathcal{L}^{c,d}(\vartheta)$ can be represented as

$$\frac{d^s \mathcal{L}^{c,d}(\vartheta)}{d\vartheta^s} = D^{(s)} \mathcal{L}^{c,d}(\vartheta), \quad (45)$$

where $D^{(s)} = (\rho_{i,n}^s)_{0 \leq i, n \leq \mathfrak{L}}$ is the $(\mathfrak{L} + 1) \times (\mathfrak{L} + 1)$ operational matrix of the s th derivative.

3.3. Caputo Fractional-Derivative Formula

Theorem 5. Let $\eta > 0$, $j \in \mathbb{N}_0$, and set $\kappa = \lceil \eta \rceil$. Then, for $t > 0$, the Caputo fractional derivative of $\mathcal{L}_j^{c,d}(t)$ is

$$\mathcal{D}_t^\eta \mathcal{L}_j^{c,d}(t) = t^{-\eta} \sum_{s=0}^j \theta_{j,s}^{(\eta)} \mathcal{L}_s^{c,d}(t), \quad (46)$$

where

$$\theta_{j,s}^{(\eta)} = \sum_{q=\kappa}^j (q - \eta + 1)_{\eta} H_{s,q} M_{q,j}. \quad (47)$$

Proof. The Caputo monomial rule is used for the expansion in (16), yielding the derivative (10).

$$\begin{aligned} \mathcal{D}_t^{\eta} \mathcal{L}_j^{c,d}(t) &= \sum_{q=\kappa}^j M_{q,j} \frac{\Gamma(q+1)}{\Gamma(q+1-\eta)} t^{q-\eta} \\ &= t^{-\eta} \sum_{q=\kappa}^j M_{q,j} (q - \eta + 1)_{\eta} t^q. \end{aligned} \quad (48)$$

Using the inversion Formula (19),

$$t^q = \sum_{s=0}^q H_{s,q} \mathcal{L}_s^{c,d}(t).$$

Substituting this expansion into (48) and reversing the finite sums give

$$\mathcal{D}_t^{\eta} \mathcal{L}_j^{c,d}(t) = t^{-\eta} \sum_{s=0}^j \left[\sum_{q=\kappa}^j (q - \eta + 1)_{\eta} H_{s,q} M_{q,j} \right] \mathcal{L}_s^{c,d}(t). \quad (49)$$

The expression in brackets is $\theta_{j,s}^{(\eta)}$, which proves (46). $\square$

Remark 7. In vector notation, the coefficient identity from Theorem 5 takes the operational form

$$\mathcal{D}_t^{\eta} \mathcal{L}^{c,d}(t) = t^{-\eta} \Theta^{(\eta)} \mathcal{L}^{c,d}(t), \quad (50)$$

where $\Theta^{(\eta)} = (\theta_{j,s}^{(\eta)})_{0 \leq j,s \leq \mathfrak{L}}$.

4. A Collocation Approach for the Time-Fractional Nonlinear Partial Integro-Differential Equation

The approximate solution uses a tensor-product affine Lucas basis in space and time. Derivative terms are approximated with developed operational matrices. A Gauss–Legendre quadrature evaluates the Volterra term. Expansion coefficients are found by solving a nonlinear algebraic system based on collocation at internal nodes and initial and boundary conditions.

For the numerical construction, take the time-fractional partial integro-differential model in [48], posed for $(\vartheta, t) \in (0, 1] \times (0, 1]$:

$$\mathcal{D}_t^{\eta} y(\vartheta, t) = \sigma(\vartheta, t) \frac{\partial^2 y(\vartheta, t)}{\partial \vartheta^2} + \int_0^t \mathcal{K}(\vartheta, t, r, y(\vartheta, r)) dr + g(\vartheta, t), \quad 0 < \eta \leq 1. \quad (51)$$

The initial profile is prescribed by

$$y(\vartheta, 0) = f_0(\vartheta), \quad \vartheta \in (0, 1]. \quad (52)$$

At the two spatial endpoints, we prescribe

$$y(0, t) = h_0(t), \quad y(1, t) = h_1(t), \quad t \in (0, 1]. \quad (53)$$

Here, $\mathcal{D}_t^\eta$ denotes the Caputo fractional-derivative operator of order η , $\sigma(\vartheta, t)$ is a continuous variable diffusion coefficient, $g(\vartheta, t)$ is the source term, and $\mathcal{K}(\vartheta, t, r, y)$ must be a smooth (C^∞) or polynomial nonlinear Volterra kernel function.

Remark 8. We want to mention that the well-posedness of time-fractional partial integro-differential equations has been established in the literature as in [1,49]. However, a general existence and uniqueness analysis for this model considered here is not available to the best of our knowledge. Such a well-posedness analysis is left for future work.

We use the finite-dimensional tensor-product trial space

$$\Omega_{\mathfrak{L}}(\Lambda) = \text{span}\{\mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t) : i, j = 0, 1, 2, \dots, \mathfrak{L}\}, \quad (54)$$

where $\Lambda = (0, 1] \times (0, 1]$.

An element $y_{\mathfrak{L}}(\vartheta, t) \in \Omega_{\mathfrak{L}}(\Lambda)$ is parameterized by

$$y_{\mathfrak{L}}(\vartheta, t) = \sum_{i=0}^{\mathfrak{L}} \sum_{j=0}^{\mathfrak{L}} \chi_{ij} \mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t) = \mathcal{L}^{c,d}(\vartheta)^T \mathbf{X} \mathcal{L}^{c,d}(t), \quad (55)$$

where $\mathcal{L}^{c,d}(\vartheta) = [\mathcal{L}_0^{c,d}(\vartheta), \mathcal{L}_1^{c,d}(\vartheta), \dots, \mathcal{L}_{\mathfrak{L}}^{c,d}(\vartheta)]^T$, and $\mathbf{X} = (\chi_{ij})_{0 \leq i, j \leq \mathfrak{L}}$ is the matrix of unknown expansion coefficients.

Substitution of this trial function into (51) defines the residual $\mathcal{R}_{\mathfrak{L}}(\vartheta, t)$:

$$\mathcal{R}_{\mathfrak{L}}(\vartheta, t) = \mathcal{D}_t^\eta y_{\mathfrak{L}}(\vartheta, t) - \sigma(\vartheta, t) \frac{\partial^2 y_{\mathfrak{L}}(\vartheta, t)}{\partial \vartheta^2} - I_{\mathfrak{L}}(\vartheta, t) - g(\vartheta, t), \quad (56)$$

where

$$I_{\mathfrak{L}}(\vartheta, t) = \int_0^t \mathcal{K}\left(\vartheta, t, r, y_{\mathfrak{L}}(\vartheta, r)\right) dr. \quad (57)$$

For quadrature, transfer the integration interval $r \in [0, t]$ to $\xi \in [-1, 1]$ through the affine substitution

$$r = \frac{t}{2}(\xi + 1). \quad (58)$$

The integral on the reference interval is therefore

$$I_{\mathfrak{L}}(\vartheta, t) = \frac{t}{2} \int_{-1}^1 \mathcal{K}\left(\vartheta, t, \frac{t}{2}(\xi + 1), y_{\mathfrak{L}}\left(\vartheta, \frac{t}{2}(\xi + 1)\right)\right) d\xi. \quad (59)$$

Applying Gauss–Legendre nodes ξ_k and weights w_k to this reference-interval integral gives

$$\begin{aligned} I_{\mathfrak{L}}(\vartheta, t) &\approx \frac{t}{2} \sum_{k=1}^{2\mathfrak{L}+2} w_k \mathcal{K}\left(\vartheta, t, \frac{t}{2}(\xi_k + 1), y_{\mathfrak{L}}\left(\vartheta, \frac{t}{2}(\xi_k + 1)\right)\right) \\ &= \frac{t}{2} \sum_{k=1}^{2\mathfrak{L}+2} w_k \mathcal{K}\left(\vartheta, t, \frac{t}{2}(\xi_k + 1), \mathcal{L}^{c,d}(\vartheta)^T \mathbf{X} \mathcal{L}^{c,d}\left(\frac{t}{2}(\xi_k + 1)\right)\right), \end{aligned} \quad (60)$$

where

$$w_k = \frac{2}{(1 - \xi_k^2) [P'_{2\mathfrak{L}+2}(\xi_k)]^2}, \quad k = 1, 2, \dots, (2\mathfrak{L} + 2). \quad (61)$$

Using Corollary 1 with $s = 2$, together with Remark 7 and the expansion (55), the required derivative and nonlinear terms $\mathcal{D}_t^\eta y_\mathfrak{L}(\vartheta, t)$, and $\frac{\partial^2 y_\mathfrak{L}(\vartheta, t)}{\partial \vartheta^2}$ are obtained:

$$\mathcal{D}_t^\eta y_\mathfrak{L}(\vartheta, t) = \mathcal{L}^{c,d}(\vartheta)^T \mathbf{X} \left[t^{-\eta} \boldsymbol{\Theta}^{(\eta)} \mathcal{L}^{c,d}(t) \right], \quad (62)$$

$$\frac{\partial^2 y_\mathfrak{L}(\vartheta, t)}{\partial \vartheta^2} = \left[D^{(2)} \mathcal{L}^{c,d}(\vartheta) \right]^T \mathbf{X} \mathcal{L}^{c,d}(t). \quad (63)$$

Insert the quadrature rule (60) and the two differential identities (62) and (63) into (56). The residual then has the form

$$\begin{aligned} \mathcal{R}_\mathfrak{L}(\vartheta, t) = & \mathcal{L}^{c,d}(\vartheta)^T \mathbf{X} \left[t^{-\eta} \boldsymbol{\Theta}^{(\eta)} \mathcal{L}^{c,d}(t) \right] - \sigma(\vartheta, t) \left(\left[D^{(2)} \mathcal{L}^{c,d}(\vartheta) \right]^T \mathbf{X} \mathcal{L}^{c,d}(t) \right) \\ & - \frac{t}{2} \sum_{k=1}^{2\mathfrak{L}+2} w_k \mathcal{K} \left(\vartheta, t, \frac{t}{2}(\xi_k + 1), \mathcal{L}^{c,d}(\vartheta)^T \mathbf{X} \mathcal{L}^{c,d} \left(\frac{t}{2}(\xi_k + 1) \right) \right) - g(\vartheta, t). \end{aligned} \quad (64)$$

The coefficients χ_{ij} are fixed by requiring $\mathcal{R}_\mathfrak{L}(\vartheta, t)$ to vanish on the interior grid $(\vartheta_r, t_s) = \left(\frac{r+1}{\mathfrak{L}+2}, \frac{s+1}{\mathfrak{L}+2} \right)$:

$$\mathcal{R}_\mathfrak{L}(\vartheta_r, t_s) = 0, \quad 1 \leq r \leq \mathfrak{L}-1, \quad 1 \leq s \leq \mathfrak{L}. \quad (65)$$

At the remaining nodes, the data from (52) and (53) furnish the supplementary equations:

$$\mathcal{L}^{c,d}(\vartheta_r)^T \mathbf{X} \mathcal{L}^{c,d}(0) = f_0(\vartheta_r), \quad 1 \leq r \leq \mathfrak{L}+1, \quad (66)$$

$$\mathcal{L}^{c,d}(0)^T \mathbf{X} \mathcal{L}^{c,d}(t_s) = h_0(t_s), \quad 1 \leq s \leq \mathfrak{L}, \quad (67)$$

$$\mathcal{L}^{c,d}(1)^T \mathbf{X} \mathcal{L}^{c,d}(t_s) = h_1(t_s), \quad 1 \leq s \leq \mathfrak{L}. \quad (68)$$

Together, the interior relations (65) and the constraints (66)–(68) form $(\mathfrak{L}+1)^2$ nonlinear equations. Newton iteration supplies the coefficient matrix, which is then substituted into (55) to evaluate the numerical approximation.

The role of the two parameters c and d may be summarized as follows:

- The choice $(c, d) = (1, 0)$ recovers the classical Lucas polynomials, whereas $(c, d) = (2, -1)$ yields the standard shifted Lucas polynomials on $[0, 1]$. Hence, the proposed general-shift family unifies these two commonly used cases within a single framework.
- Other admissible choices of c and d generate different general-shift Lucas bases and therefore provide additional flexibility in constructing the spectral approximation.
- For the same truncation degree, different parameter pairs may lead to different numerical approximations and consequently different error levels. This behavior is illustrated in the numerical experiments presented below.
- No universal optimal pair (c, d) is expected to exist for all problems. The parameters should therefore be viewed as adjustable basis parameters, and suitable values may be selected according to the numerical behavior observed for the problem under consideration.

Remark 9. The resulting system of $N = (\mathfrak{L}+1)^2$ nonlinear equations is solved using the built-in *FindRoot* function in Wolfram Mathematica (which implements Newton's method):

- **Memory Complexity:** $\mathcal{O}(\mathfrak{L}^4)$, required to store the $N \times N$ dense Jacobian matrix.
- **Time Complexity:** $\mathcal{O}(I_{\text{iter}} \cdot \mathfrak{L}^6)$, where I_{iter} is the number of *FindRoot* iterations. Each Newton step requires $\mathcal{O}(N^3) = \mathcal{O}(\mathfrak{L}^6)$ operations for dense linear solving.

Due to spectral convergence, small choices of $(\mathfrak{L} \leq 12)$ achieve high precision within a few seconds.

Remark 10. Figure 1 presents the steps of the method and illustrates the computational process until obtaining the approximate solution.

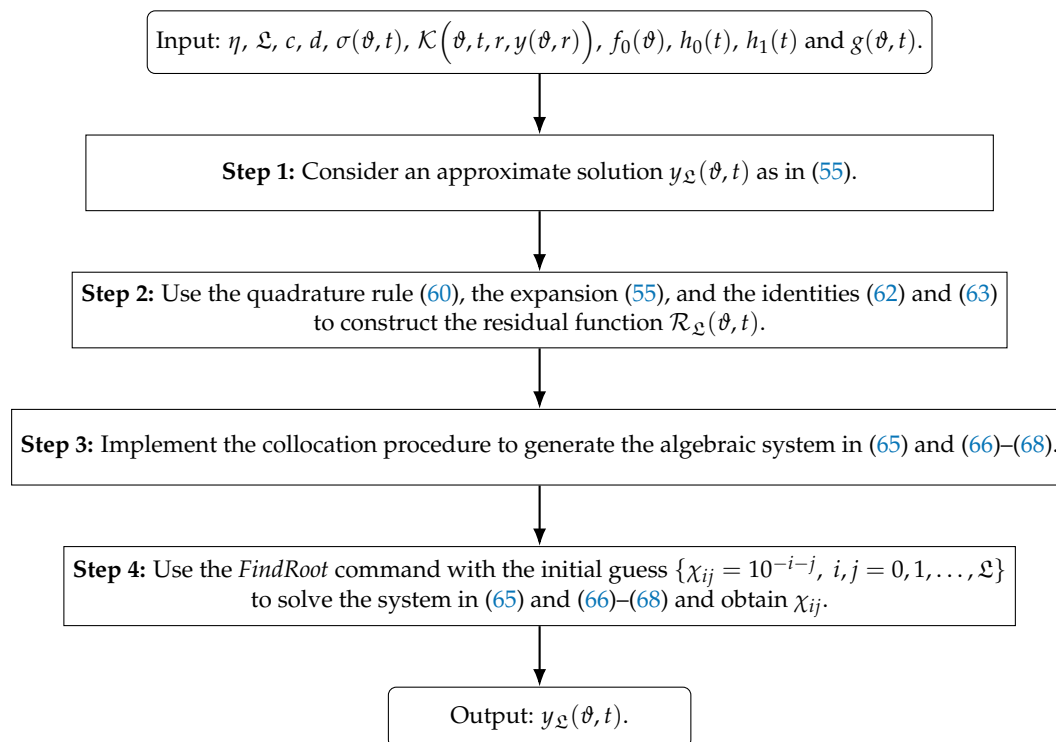


Figure 1. Flowchart of the main computational steps of the proposed collocation algorithm.

5. Convergence and Truncation Error of the Lucas Polynomial Expansion

We will analyze the error through two steps: the first is to obtain a uniform bound for each affine Lucas basis function on the spatial interval, and the second is to combine this with expansion coefficient estimates to bound the tensor-product approximation's truncation error.

Lemma 1. Let $c, d \in \mathbb{R}$ and $\nu = \max_{\vartheta \in [0,1]} |c\vartheta + d| = \max\{|d|, |c + d|\}$. The polynomials $\mathcal{L}_i^{c,d}(\vartheta)$ satisfy

$$|\mathcal{L}_i^{c,d}(\vartheta)| \leq 2 \left(\frac{\nu + \sqrt{\nu^2 + 4}}{2} \right)^i, \quad \vartheta \in [0, 1], \quad i \geq 0. \quad (69)$$

Proof. The affine Lucas basis satisfies the following recursive formula:

$$\mathcal{L}_i^{c,d}(\vartheta) = (c\vartheta + d) \mathcal{L}_{i-1}^{c,d}(\vartheta) + \mathcal{L}_{i-2}^{c,d}(\vartheta), \quad i \geq 2, \quad (70)$$

with initial conditions

$$\mathcal{L}_0^{c,d}(\vartheta) = 2, \quad \mathcal{L}_1^{c,d}(\vartheta) = c\vartheta + d.$$

The associated characteristic equation is

$$\rho^2 - (c\vartheta + d)\rho - 1 = 0, \quad (71)$$

whose characteristic roots are

$$\lambda_{\pm} = \frac{(c\vartheta + d) \pm \sqrt{(c\vartheta + d)^2 + 4}}{2}. \quad (72)$$

In virtue of the initial conditions, $\mathcal{L}_i^{c,d}(\vartheta)$ can be represented explicitly as

$$\mathcal{L}_i^{c,d}(\vartheta) = \lambda_+^i + \lambda_-^i. \quad (73)$$

For $\vartheta \in [0, 1]$, $|c\vartheta + d| \leq \nu = \max(|d|, |c + d|)$. The magnitude of the roots is bounded by

$$|\lambda_{\pm}| \leq \frac{|c\vartheta + d| + \sqrt{(c\vartheta + d)^2 + 4}}{2} \leq \frac{\nu + \sqrt{\nu^2 + 4}}{2}. \quad (74)$$

Therefore,

$$\begin{aligned} |\mathcal{L}_i^{c,d}(\vartheta)| &= |\lambda_+^i + \lambda_-^i| \\ &\leq |\lambda_+|^i + |\lambda_-|^i \\ &\leq 2 \left(\frac{\nu + \sqrt{\nu^2 + 4}}{2} \right)^i. \end{aligned} \quad (75)$$

The asserted uniform estimate follows. $\square$

Lemma 2. Let $u \in C^\infty((0, 1])$. $u(\vartheta)$ can be expanded in shifted Lucas basis polynomials as

$$u(\vartheta) = \sum_{n=0}^{\infty} \sum_{s=n}^{\infty} \frac{u^{(s)}(0) H_{n,s}}{s!} \mathcal{L}_n^{c,d}(\vartheta), \quad (76)$$

where $H_{n,s}$ is defined in Equation (20).

Proof. Expand $u(\vartheta)$ first in its Maclaurin series:

$$u(\vartheta) = \sum_{n=0}^{\infty} \frac{u^{(n)}(0)}{n!} \vartheta^n. \quad (77)$$

Substitute the inversion (19) into the previous series and interchange the sums; collecting the coefficient of $\mathcal{L}_i^{c,d}(\vartheta)$ gives (76). $\square$

Theorem 6. Let $u \in C^\infty((0, 1])$ be expanded as $u(\vartheta) = \sum_{n=0}^{\infty} \chi_n \mathcal{L}_n^{c,d}(\vartheta)$, $|u^{(s)}(0)| \leq \rho^s$ for $s > 0$, where $\rho > 0$, then

$$|\chi_n| < \frac{e^{\frac{\rho(|d|+3)}{|c|}} \left(\frac{\rho(|d|+3)}{|c|} \right)^n}{n!}, \quad \forall n > 0. \quad (78)$$

Absolute convergence of the series also follows.

Proof. Lemma 2 enables us to write $|\chi_n|$ as

$$|\chi_n| = \left| \sum_{s=n}^{\infty} \frac{u^{(s)}(0) H_{n,s}}{s!} \right| \leq \sum_{s=n}^{\infty} \frac{|u^{(s)}(0)| |H_{n,s}|}{s!}. \quad (79)$$

The inverse coefficients from (20) satisfy the following estimate whenever $s \geq n$:

$$|H_{n,s}| < \left(\frac{|d| + 3}{|c|} \right)^s, \quad (80)$$

where $\mathbb{N}_0 = \{0, 1, 2, \dots\}$.

Combining this estimate with the assumed derivative growth gives

$$|\chi_n| \leq \sum_{s=n}^{\infty} \frac{\left(\frac{\rho(|d|+3)}{|c|}\right)^s}{s!} = \frac{e^{\frac{\rho(|d|+3)}{|c|}} ((n-1)! - \Gamma(n, \frac{\rho(|d|+3)}{|c|}))}{(n-1)!}, \quad (81)$$

where $\Gamma(n, \frac{\rho(|d|+3)}{|c|})$ is the incomplete gamma function.

Setting $x = \frac{\rho(|d|+3)}{|c|}$, this inequality follows from the integral definition of the lower incomplete gamma function, $\gamma(n, x) = (n-1)! - \Gamma(n, x) = \int_0^x v^{n-1} e^{-v} dv < \int_0^x v^{n-1} dv = \frac{x^n}{n}$, which yields

$$\frac{e^{\frac{\rho(|d|+3)}{|c|}} ((n-1)! - \Gamma(n, \frac{\rho(|d|+3)}{|c|}))}{(n-1)!} < \frac{e^{\frac{\rho(|d|+3)}{|c|}} \left(\frac{\rho(|d|+3)}{|c|}\right)^n}{n!}, \quad \forall n > 0. \quad (82)$$

Hence, (81) implies

$$|\chi_n| < \frac{e^{\frac{\rho(|d|+3)}{|c|}} \left(\frac{\rho(|d|+3)}{|c|}\right)^n}{n!}, \quad \forall n > 0. \quad (83)$$

Together, the basis bound (69) and the coefficient estimate (78) yield an exponential-series majorant:

$$\begin{aligned} \left| \sum_{n=0}^{\infty} \chi_n \mathcal{L}_n^{c,d}(\vartheta) \right| &\leq \sum_{n=0}^{\infty} |\chi_n| \left| \mathcal{L}_n^{c,d}(\vartheta) \right| \\ &< \sum_{n=0}^{\infty} \frac{2^{1-n} e^{\frac{\rho(|d|+3)}{|c|}} \left(\frac{(\sqrt{v^2+4}+v)\rho(|d|+3)}{|c|} \right)^n}{\Gamma(n+1)} \\ &= 2e^{\frac{(\sqrt{v^2+4}+v+2)\rho(|d|+3)}{2|c|}}, \end{aligned} \quad (84)$$

which proves absolute convergence. $\square$

Theorem 7. Let $y_1, y_2 \in C^\infty((0, 1])$ such that $y(\vartheta, t) = y_1(\vartheta)y_2(t) \in C^\infty((0, 1] \times (0, 1])$, with expansion $y(\vartheta, t) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \chi_{ij} \mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t)$. If $|y_1^{(i)}(0)| \leq \rho_1^i$ and $|y_2^{(j)}(0)| \leq \rho_2^j$ for all $i, j > 0$ with positive constants ρ_1, ρ_2 , then

$$|\chi_{ij}| < \frac{e^{\frac{(\rho_1+\rho_2)(|d|+3)}{|c|}} \rho_1^i \rho_2^j \left(\frac{|d|+3}{|c|}\right)^{i+j}}{i! j!}. \quad (85)$$

Furthermore, the series is absolutely convergent.

Proof. Based on Lemma 2, along with the stated assumption $y(\vartheta, t) = y_1(\vartheta)y_2(t)$, we get

$$\chi_{ij} = \sum_{s=i}^{\infty} \sum_{q=j}^{\infty} \frac{y_1^{(s)}(0) y_2^{(q)}(0) H_{j,q} H_{i,s}}{s! q!}. \quad (86)$$

Utilizing the assumptions $|y_1^{(s)}(0)| \leq \rho_1^s$ and $|y_2^{(q)}(0)| \leq \rho_2^q$, we can write

$$|\chi_{ij}| \leq \sum_{s=i}^{\infty} \frac{\rho_1^s |H_{i,s}|}{s!} \times \sum_{q=j}^{\infty} \frac{\rho_2^q |H_{j,q}|}{q!}. \quad (87)$$

Apply the one-variable-coefficient estimate from Theorem 6 to the two factors to arrive at

$$|\chi_{ij}| < \frac{e^{\frac{(\rho_1+\rho_2)(|d|+3)}{|c|}} \rho_1^i \rho_2^j \left(\frac{(|d|+3)}{|c|} \right)^{i+j}}{i! j!}. \quad (88)$$

□

Theorem 8. For the finite tensor approximation, the hypotheses of Theorem 7 yield the following truncation-error estimate:

$$|y(\vartheta, t) - y_{\mathfrak{L}}(\vartheta, t)| < \frac{2^{1-\mathfrak{L}} (\rho_1 + \rho_2)^{\mathfrak{L}+1} e^{\frac{(\sqrt{v^2+4}+\nu+2)(\rho_1+\rho_2)(|d|+3)}{2|c|}} \left(\frac{(\sqrt{v^2+4}+\nu)(|d|+3)}{|c|} \right)^{\mathfrak{L}+1}}{\mathfrak{L}!}. \quad (89)$$

Proof. In view of the definitions of $y(\vartheta, t)$ and $y_{\mathfrak{L}}(\vartheta, t)$, we obtain

$$\begin{aligned} |y(\vartheta, t) - y_{\mathfrak{L}}(\vartheta, t)| &= \left| \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \chi_{ij} \mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t) - \sum_{i=0}^{\mathfrak{L}} \sum_{j=0}^{\mathfrak{L}} \chi_{ij} \mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t) \right| \\ &\leq \left| \sum_{i=0}^{\mathfrak{L}} \sum_{j=\mathfrak{L}+1}^{\infty} \chi_{ij} \mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t) \right| + \left| \sum_{i=\mathfrak{L}+1}^{\infty} \sum_{j=0}^{\infty} \chi_{ij} \mathcal{L}_i^{c,d}(\vartheta) \mathcal{L}_j^{c,d}(t) \right|. \end{aligned} \quad (90)$$

Combining Theorem 7 with Lemma 1 yields the following three estimates and one identity:

$$\begin{aligned} &\sum_{i=0}^{\mathfrak{L}} \frac{2^{1-i} e^{\frac{\rho_1(|d|+3)}{|c|}}}{i!} \left(\frac{(\sqrt{v^2+4}+\nu)\rho_1(|d|+3)}{|c|} \right)^i \\ &= \frac{2e^{\frac{\rho_1(\sqrt{v^2+4}+\nu+2)(|d|+3)}{2|c|}}}{\mathfrak{L}!} \Gamma\left(\mathfrak{L}+1, \frac{\rho_1(\nu+\sqrt{v^2+4})(|d|+3)}{2|c|}\right) \\ &< 2e^{\frac{(\sqrt{v^2+4}+\nu+2)\rho_1(|d|+3)}{2|c|}}, \end{aligned} \quad (91)$$

$$\begin{aligned} &\sum_{j=\mathfrak{L}+1}^{\infty} \frac{2^{1-j} e^{\frac{\rho_2(|d|+3)}{|c|}}}{j!} \left(\frac{(\sqrt{v^2+4}+\nu)\rho_2(|d|+3)}{|c|} \right)^j \\ &= \frac{2e^{\frac{\rho_2(\sqrt{v^2+4}+\nu+2)(|d|+3)}{2|c|}}}{\mathfrak{L}!} \left[\mathfrak{L}! - \Gamma\left(\mathfrak{L}+1, \frac{\rho_2(\nu+\sqrt{v^2+4})(|d|+3)}{2|c|}\right) \right] \\ &< \frac{2^{-\mathfrak{L}} e^{\frac{\rho_2(\sqrt{v^2+4}+\nu+2)(|d|+3)}{2|c|}}}{\mathfrak{L}!} \left(\frac{\rho_2(\sqrt{v^2+4}+\nu)(|d|+3)}{|c|} \right)^{\mathfrak{L}+1}, \end{aligned} \quad (92)$$

$$\begin{aligned} &\sum_{i=\mathfrak{L}+1}^{\infty} \frac{2^{1-i} e^{\frac{\rho_1(|d|+3)}{|c|}}}{i!} \left(\frac{(\sqrt{v^2+4}+\nu)\rho_1(|d|+3)}{|c|} \right)^i \\ &= \frac{2e^{\frac{\rho_1(\sqrt{v^2+4}+\nu+2)(|d|+3)}{2|c|}}}{\mathfrak{L}!} \left[\mathfrak{L}! - \Gamma\left(\mathfrak{L}+1, \frac{\rho_1(\nu+\sqrt{v^2+4})(|d|+3)}{2|c|}\right) \right] \\ &< \frac{2^{-\mathfrak{L}} e^{\frac{\rho_1(\sqrt{v^2+4}+\nu+2)(|d|+3)}{2|c|}}}{\mathfrak{L}!} \left(\frac{\rho_1(\sqrt{v^2+4}+\nu)(|d|+3)}{|c|} \right)^{\mathfrak{L}+1}, \end{aligned} \quad (93)$$

$$\sum_{j=0}^{\infty} \frac{2^{1-j} e^{\frac{\rho_2(|d|+3)}{|c|}}}{j!} \left(\frac{(\sqrt{\nu^2+4}+\nu)\rho_2(|d|+3)}{|c|} \right)^j = 2e^{\frac{(\sqrt{\nu^2+4}+\nu+2)\rho_2(|d|+3)}{2|c|}}. \quad (94)$$

Using the estimates and identity in Equations (91)–(94), together with $\rho_1^{\mathfrak{L}+1} + \rho_2^{\mathfrak{L}+1} \leq (\rho_1 + \rho_2)^{\mathfrak{L}+1}$, we obtain the following bound:

$$|y(\vartheta, t) - y_{\mathfrak{L}}(\vartheta, t)| < \frac{2^{1-\mathfrak{L}} (\rho_1 + \rho_2)^{\mathfrak{L}+1} e^{\frac{(\sqrt{\nu^2+4}+\nu+2)(\rho_1+\rho_2)(|d|+3)}{2|c|}} \left(\frac{(\sqrt{\nu^2+4}+\nu)(|d|+3)}{|c|} \right)^{\mathfrak{L}+1}}{\mathfrak{L}!}. \quad (95)$$

□

6. Illustrative Problems

The numerical tests report pointwise and norm-based errors for numerous truncation degrees and affine-parameter selections. Published benchmark results are compared where available. All calculations were performed in Wolfram Mathematica 11 (Version 11.0, released in 2016) on an HP Z420 workstation with an Intel(R) Xeon(R) E5-1620 v2 3.70 GHz processor and 16 GB RAM. All CPU times reported in the tables are expressed in seconds (s). Numerical errors were recorded in Mathematica.

Problem 1 ([48]). In the first benchmark, consider the equation

$$\mathcal{D}_t^\eta y(\vartheta, t) = \left(\vartheta^5 + 4\vartheta^2 \right) \frac{\partial^2 y(\vartheta, t)}{\partial \vartheta^2} + \int_0^t \left(r^2 y^4 \left(t^3 \vartheta^3 + 1 \right) \right) dr + g(\vartheta, t), \quad (96)$$

subject to

$$y(\vartheta, 0) = \sin(\vartheta), \quad \vartheta \in (0, 1], \quad (97)$$

$$y(0, t) = 0, \quad y(1, t) = e^{-t} \sin(1), \quad t \in (0, 1]. \quad (98)$$

The source term $g(\vartheta, t)$ is defined so that $y(\vartheta, t) = e^{-t} \sin(\vartheta)$ satisfies the model exactly.

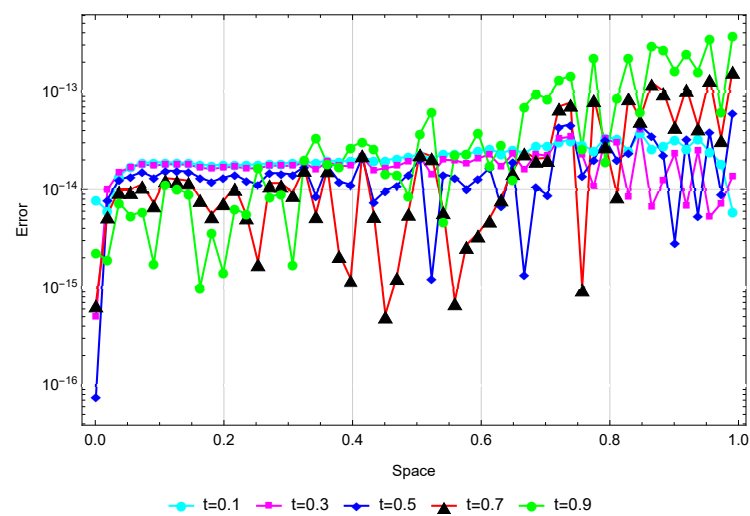
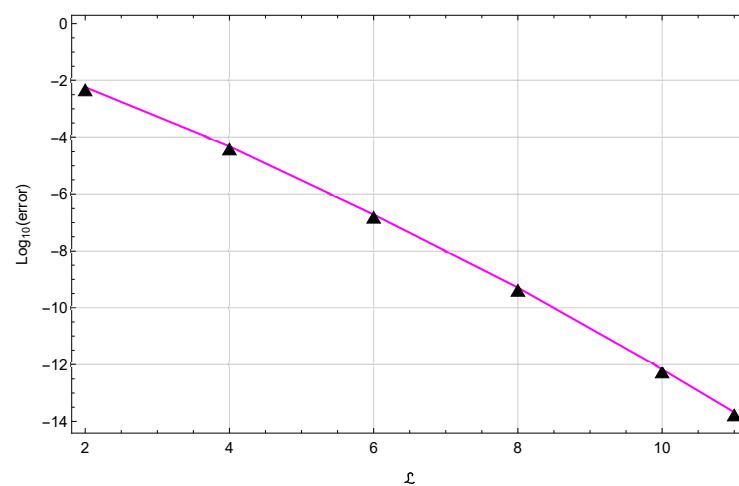
Table 1 compares the L_∞ and L_2 errors from our approximation with the published values of [48] at $\eta = 0.25$ and $t = 1$. Also, Table 2 compares the L_∞ and L_2 errors from our approximation with the published values of [50] at $\eta = 0.95$ and different values of t . Figure 2 depicts the absolute errors (AEs) at different values of t when $\eta = 0.25$, $c = 3$, $d = 2$ and $\mathfrak{L} = 11$. Figure 3 plots the $\log_{10}$ of the L_2 errors at different values of $\mathfrak{L}$ when $\eta = 0.9$, $c = 4$ and $d = 3$. Figure 4 displays the AEs at different values of $\mathfrak{L}$ when $\eta = 0.5$ and $c = d = 1$. Finally, Table 3 presents the CPU time used (in seconds) and L_∞ errors at different values of $\mathfrak{L}$, c and d when $\eta = 0.9$.

Table 1. Comparison of L_∞ and L_2 errors for Problem 1 at $\eta = 0.25$ and $t = 1$.

Method in [48] at $\delta t = 0.0001$			Proposed Technique at $\mathfrak{L} = 11$		
h	L_2	L_∞	(c, d)	L_2	L_∞
0.1	1.3828×10^{-6}	5.9314×10^{-5}	(2, −1)	7.05193×10^{-15}	2.35815×10^{-14}
0.05	4.2153×10^{-6}	1.8092×10^{-5}			
0.025	1.2814×10^{-7}	5.5022×10^{-6}			
0.0125	3.8853×10^{-8}	1.6688×10^{-6}			
0.00625	1.1769×10^{-8}	5.0553×10^{-7}			

Table 2. Comparison of L_∞ and L_2 errors for Problem 1 at $\eta = 0.95$.

t	Method in [50] at $\delta t = 0.0005, h = 0.025$		Proposed Technique at $(c, d) = (2, -1)$ and $\mathcal{L} = 11$	
	L_2	L_∞	L_2	L_∞
0.1	1.21×10^{-5}	1.82×10^{-5}	2.70417×10^{-14}	3.43059×10^{-14}
0.2	1.40×10^{-5}	2.02×10^{-5}	2.5194×10^{-14}	3.08642×10^{-14}
0.3	1.43×10^{-5}	2.02×10^{-5}	2.42673×10^{-14}	2.83107×10^{-14}
0.4	1.39×10^{-5}	1.94×10^{-5}	2.33608×10^{-14}	2.58682×10^{-14}
0.5	1.32×10^{-5}	1.83×10^{-5}	2.23133×10^{-14}	2.38698×10^{-14}
0.6	1.24×10^{-5}	1.71×10^{-5}	2.10897×10^{-14}	2.24265×10^{-14}
0.7	1.16×10^{-5}	1.58×10^{-5}	1.98264×10^{-14}	2.12885×10^{-14}
0.8	1.08×10^{-5}	1.46×10^{-5}	1.84943×10^{-14}	2.00950×10^{-14}
0.9	1.00×10^{-5}	1.35×10^{-5}	1.72246×10^{-14}	1.88738×10^{-14}
1	9.20×10^{-5}	1.26×10^{-5}	1.28503×10^{-14}	2.35589×10^{-14}

**Figure 2.** The AEs for Problem 1 at $\eta = 0.25, c = 3, d = 2$ and $\mathcal{L} = 11$.**Figure 3.** The $\log_{10}$ of the L_2 errors for Problem 1 at $\eta = 0.9, c = 4$ and $d = 3$.

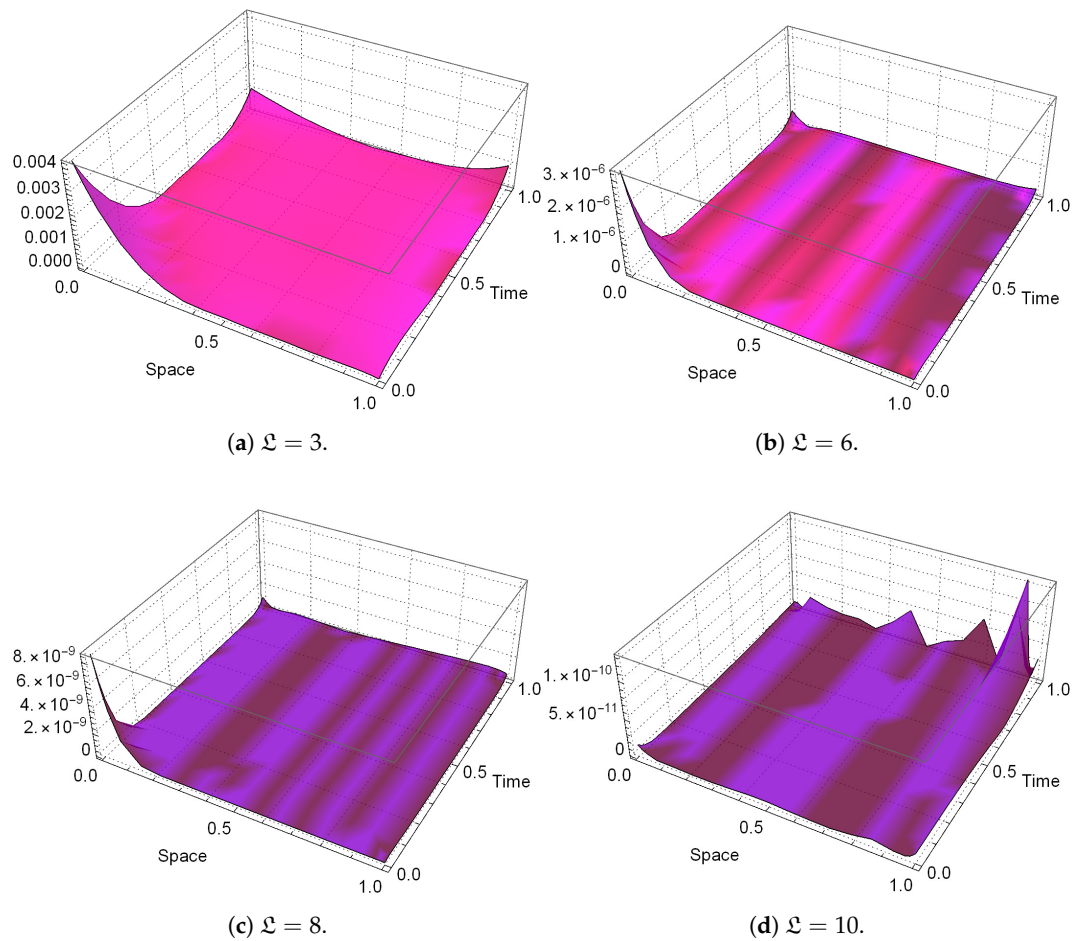


Figure 4. AEs for Problem 1 at $\eta = 0.5$ and $c = d = 1$.

Table 3. The L_∞ errors of Problem 1 at $\eta = 0.9$.

$\mathcal{L}$	4	6	8	10	11
$c = 1, d = 0$	7.75619×10^{-5}	3.24070×10^{-6}	8.22446×10^{-10}	1.38456×10^{-12}	1.01478×10^{-13}
CPU time	4.187	5.03	15.703	55.766	102.203
$c = 2, d = -1$	1.30099×10^{-4}	3.24070×10^{-6}	8.22449×10^{-10}	1.38156×10^{-12}	1.01400×10^{-13}
CPU time	4.532	5.203	12.891	44.501	84.14
$c = 4, d = 3$	7.75619×10^{-5}	3.24070×10^{-6}	8.22496×10^{-10}	5.17275×10^{-12}	2.58710×10^{-12}
CPU time	4.563	5.562	19.905	68.532	99.469

Problem 2 ([48]). We consider the test problem

$$\mathcal{D}_t^\eta y(\vartheta, t) = \left(\sin(\vartheta) + \cos^2(\vartheta) \right) \frac{\partial^2 y(\vartheta, t)}{\partial \vartheta^2} + \int_0^t \left(y^2 e^{r+t+\vartheta} \right) dr + g(\vartheta, t), \quad (99)$$

subject to

$$y(\vartheta, 0) = \vartheta, \quad \vartheta \in (0, 1], \quad (100)$$

$$y(0, t) = 0, \quad y(1, t) = \cos(t), \quad t \in (0, 1], \quad (101)$$

where $g(\vartheta, t)$ is prescribed to yield the exact solution $y(\vartheta, t) = \vartheta \cos(t)$.

For the second benchmark, Table 4 places our L_∞ and L_2 errors alongside the results in [48] at $\eta = 0.75$ and $t = 1$. Figure 5 shows the AEs at different values of t when $\eta = 0.5$, $c = 2$, $d = -1$

and $\mathfrak{L} = 12$. Table 5 presents the L_∞ errors at different values of $\mathfrak{L}$, c and d when $\eta = 0.3$. Finally, Figure 6 presents the AEs at different values of $\mathfrak{L}$ when $\eta = 0.9$ and $c = 4, d = -3$.

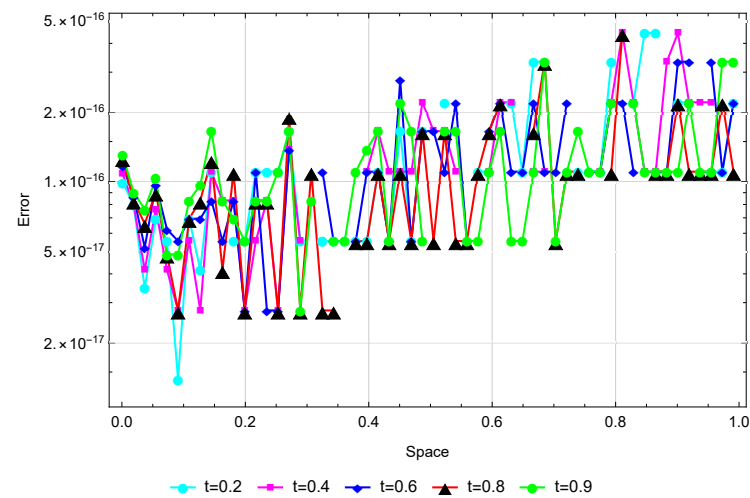


Figure 5. The AEs for Problem 2 at $\eta = 0.5, c = 2, d = -1$ and $\mathfrak{L} = 12$.

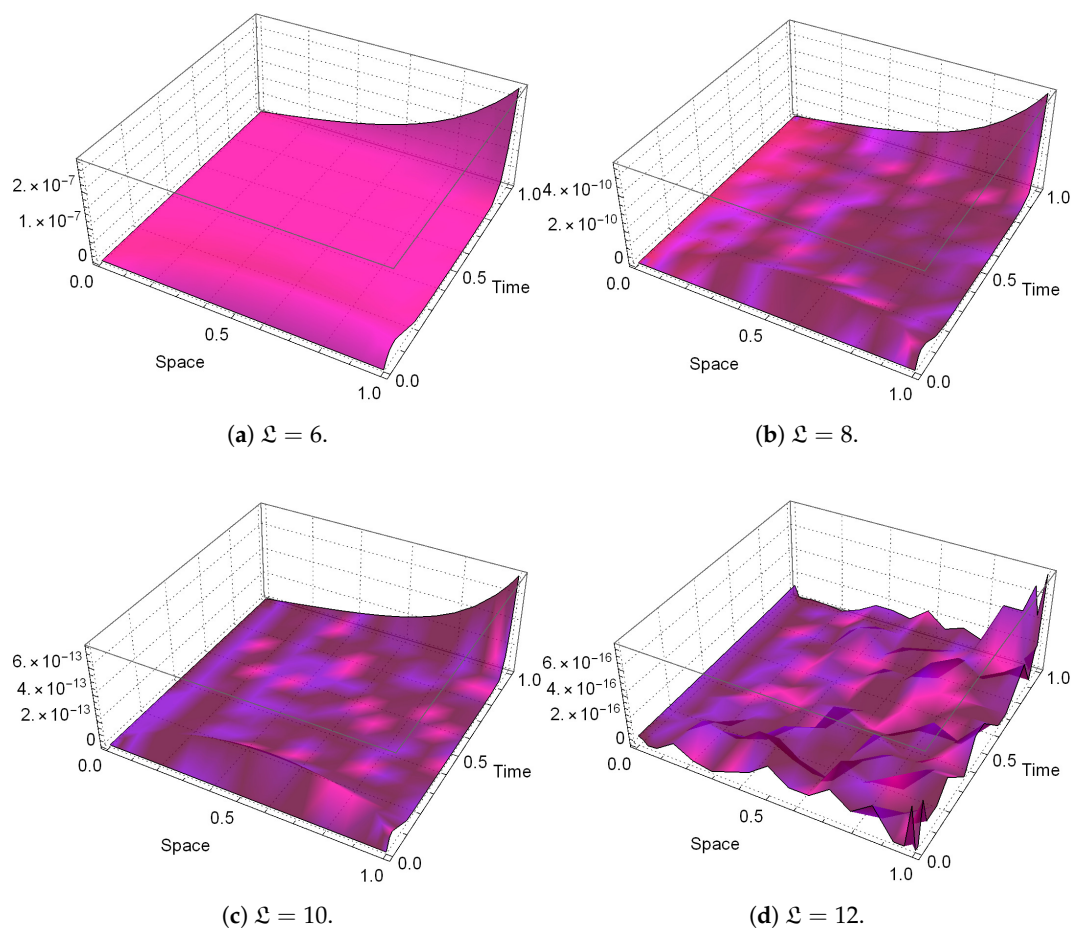


Figure 6. AEs for Problem 2 at $\eta = 0.9$ and $c = 4, d = -3$.

Table 4. Comparison of L_∞ and L_2 errors for Problem 2 at $\eta = 0.75$ and $t = 1$.

Method in [48] at $h = 0.001$			Proposed Technique at $\mathfrak{L} = 12$		
δt	L_2	L_∞	(c, d)	L_2	L_∞
0.1	6.8418×10^{-5}	2.6324×10^{-3}	$(2, -1)$	2.83206×10^{-16}	1.33227×10^{-15}
0.05	3.1900×10^{-5}	1.2200×10^{-3}			
0.025	1.4071×10^{-5}	5.3662×10^{-4}			
0.0125	6.0642×10^{-6}	2.3095×10^{-4}			
0.00625	2.2548×10^{-6}	9.8363×10^{-5}			

Table 5. The L_∞ errors of Problem 2 at $\eta = 0.3$.

$\mathfrak{L}$	4	6	8	10	12
$c = 1, d = 1$	8.07749×10^{-5}	2.81872×10^{-7}	4.32328×10^{-11}	-	-
$c = 1, d = 2$	8.07749×10^{-5}	2.81872×10^{-7}	-	-	-
$c = 2, d = -1$	8.07749×10^{-5}	2.81872×10^{-7}	4.32327×10^{-11}	9.37028×10^{-14}	1.66533×10^{-16}
$c = 3, d = -2$	8.07749×10^{-5}	2.81872×10^{-7}	4.32325×10^{-11}	9.34808×10^{-14}	1.11022×10^{-16}

Problem 3 ([48]). We consider the test problem

$$\mathcal{D}_t^\eta y(\vartheta, t) = (\vartheta^3 + 1) \frac{\partial^2 y(\vartheta, t)}{\partial \vartheta^2} + \int_0^t (3r^2 t e^y \vartheta^2) dr + g(\vartheta, t), \quad (102)$$

subject to

$$y(\vartheta, 0) = 0, \quad \vartheta \in (0, 1], \quad (103)$$

$$y(0, t) = 0, \quad y(1, t) = \frac{1}{3} t^3, \quad t \in (0, 1], \quad (104)$$

where $g(\vartheta, t)$ is prescribed so that the exact solution becomes $y(\vartheta, t) = \frac{1}{3} t^3 \vartheta^3$.

For the third benchmark, Table 6 reports both error norms and the corresponding reference data from [48] at $\eta = 0.5$ and $t = 1$. Figure 7 displays the AEs at different values of t when $\eta = 0.3$, $c = 2$, $d = -1$ and $\mathfrak{L} = 4$. Table 7 presents the CPU time used (in seconds) and AEs at $\eta = 0.8$, $c = 1$, $d = -2$ and $\mathfrak{L} = 4$.

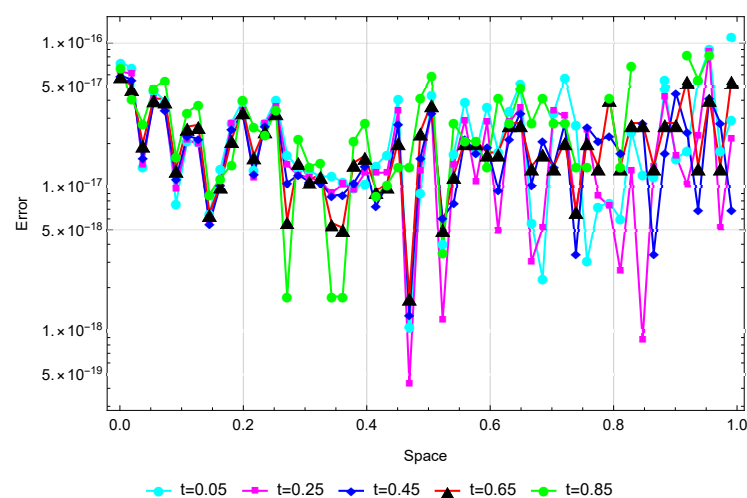
**Figure 7.** The AEs for Problem 3 at $\eta = 0.3$, $c = 2$, $d = -1$ and $\mathfrak{L} = 4$.

Table 6. Comparison of L_∞ and L_2 errors for Problem 3 at $\eta = 0.5$ and $t = 1$.

Method in [48] at $\delta t = 0.0001$			Proposed Technique at $\mathfrak{L} = 4$		
h	L_2	L_∞	(c, d)	L_2	L_∞
0.1	5.4063×10^{-5}	2.3880×10^{-4}	(1,1)	3.34184×10^{-23}	2.57572×10^{-14}
0.05	9.5330×10^{-5}	5.9546×10^{-5}			
0.025	1.6835×10^{-6}	1.4871×10^{-5}			
0.0125	2.9706×10^{-7}	3.7115×10^{-6}			
0.00625	5.2162×10^{-8}	9.2164×10^{-7}			

Table 7. The AEs for Problem 3 at $\eta = 0.8$, $c = 1$, $d = -2$ and $\mathfrak{L} = 4$.

ϑ	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	CPU Time
0.1	5.67025×10^{-16}	8.75964×10^{-16}	9.38919×10^{-16}	7.74446×10^{-16}	2.202
0.2	4.51541×10^{-15}	3.88871×10^{-15}	3.44516×10^{-15}	3.21358×10^{-15}	
0.3	5.83997×10^{-15}	4.82275×10^{-15}	4.05166×10^{-15}	3.54057×10^{-15}	
0.4	1.99418×10^{-14}	1.71764×10^{-14}	1.49325×10^{-14}	1.31214×10^{-14}	
0.5	4.47765×10^{-15}	3.79124×10^{-15}	3.15546×10^{-15}	2.76862×10^{-15}	
0.6	5.54902×10^{-14}	4.81273×10^{-14}	4.15102×10^{-14}	3.60753×10^{-14}	
0.7	7.15324×10^{-15}	6.37771×10^{-15}	5.16948×10^{-15}	5.05845×10^{-15}	
0.8	1.43895×10^{-15}	8.53484×10^{-16}	8.88178×10^{-16}	4.99600×10^{-16}	
0.9	1.38214×10^{-14}	1.24276×10^{-14}	1.07900×10^{-14}	9.27036×10^{-15}	
1.0	5.17902×10^{-15}	4.57273×10^{-15}	3.48332×10^{-15}	3.71925×10^{-15}	

Remark 11. Comparisons in Tables 1–6 focus on numerical accuracy, as the suggested technique employs $\mathfrak{L}$ truncation degree, while [48] reports mesh width h and not execution times.

Remark 12. It is worth mentioning that the condition number grows as a function of $\mathfrak{L}$. But the new basis limits how fast the condition number increases; it will remain within reasonable limits for values of $\mathfrak{L} \leq 12$. Wolfram Mathematica was used to calculate the matrix inverse to 50 decimal digits to minimize round-off sensitivity.

Remark 13. Smooth solutions with rapid error reduction work well with the scheme. Lower regularity or initial singularities slow convergence, while higher truncation degrees increase conditioning and computing cost. Strong nonlinearities may also make Newton iteration more sensitive to initial guesses.

7. Discussion

Computations show a consistent pattern for the general-shift Lucas collocation approach. In smooth benchmark tasks, boosting $\mathfrak{L}$ significantly reduces error. This trend matches the global spectral expansion and earlier approximation estimates.

The data demonstrates that the achievable error depends on the affine parameters, c and d , especially at higher truncation levels. Changing c or d alters the Lucas basis functions' argument, affecting the polynomial representation. Therefore, it is better to consider c and d as adjustable basis parameters rather than fixed values.

The extremely small errors observed in some tests should also be interpreted in light of the high regularity and simple analytic form of the manufactured exact solutions. Comparable accuracy should not automatically be expected for problems with reduced smoothness, pronounced initial singularities, or more complicated nonlinear dynamics.

Comparisons with published data show that moderate truncation degrees can achieve great accuracy. These results should be considered as accuracy benchmarks rather than

equal-cost performance tests because the competing algorithms use different discretization parameters and implementation choices.

8. Concluding Remarks

This article introduces a general-shift Lucas family with two parameters. The introduced polynomials serve as the spectral basis for the numerical treatment of the nonlinear time-fractional partial integro-differential model. This affine representation generalizes both the standard Lucas basis and the standard shifted Lucas basis $L_n(2\theta - 1)$. New identities of the utilized polynomials, including the series form, its inverse form, and operational identities for integer and Caputo differentiation, have been developed and integrated into a space–time tensor collocation scheme. In addition, Gauss–Legendre quadrature is employed to evaluate the nonlinear Volterra term, and Newton iteration is used to solve the resulting nonlinear system. The numerical results show a high degree of precision and illustrate how adjusting the values of c and d may alter the numerical behavior without requiring changes to the general computational framework. As far as we know, this is the first spectral investigation of the class of equations considered using the two-parameter general-shift Lucas family. In the near future, we hope to investigate and utilize other affine bases in a variety of important models in applied science.

Author Contributions: Conceptualization, W.M.A.-E. and A.G.A.; Methodology, W.M.A.-E., N.M.A.A., O.M.A., A.K.A. and A.G.A.; Software, W.M.A.-E. and A.G.A.; Validation, W.M.A.-E., N.M.A.A., O.M.A., A.K.A., M.A.A. and A.G.A.; Formal analysis, W.M.A.-E., N.M.A.A. and A.G.A.; Investigation, W.M.A.-E., N.M.A.A., O.M.A., A.K.A., M.A.A. and A.G.A.; Writing—original draft, W.M.A.-E., O.M.A. and A.G.A.; Writing—review & editing, W.M.A.-E. and A.G.A.; Supervision, W.M.A.-E.; Project administration, W.M.A.-E.; Funding acquisition, M.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by Umm Al-Qura University, Saudi Arabia, under grant number: 26UQU4410195GSSR06.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors extend their appreciation to Umm Al-Qura University, Saudi Arabia, for funding this research work through grant number: 26UQU4410195GSSR06.

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

References

1. Podlubny, I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*; Mathematics in Science and Engineering; Academic Press: San Diego, CA, USA, 1999; Volume 198.
2. Kilbas, A.A.; Srivastava, H.M.; Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*; North-Holland Mathematics Studies; Elsevier: Amsterdam, The Netherlands, 2006; Volume 204.
3. Mainardi, F. *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*; Imperial College Press: London, UK, 2010. [[CrossRef](#)]
4. Atanacković, T.M.; Pilipović, S.; Stanković, B.; Zorica, D. *Fractional Calculus with Applications in Mechanics: Vibrations and Diffusion Processes*; ISTE–Wiley: London, UK, 2014. [[CrossRef](#)]
5. Li, J.; Su, X.; Zhao, K. Barycentric interpolation collocation algorithm to solve fractional differential equations. *Math. Comput. Simul.* **2023**, *205*, 340–367. [[CrossRef](#)]
6. Mokhtarnezhadazar, F. A high order predictor-corrector method with non-uniform meshes for fractional differential equations. *Fract. Calc. Appl. Anal.* **2024**, *27*, 2577–2605. [[CrossRef](#)]
7. Marasi, H.R.; Derakhshan, M.H.; Ghuraibawi, A.A.; Kumar, P. A novel method based on fractional order Gegenbauer wavelet operational matrix for the solutions of the multi-term time-fractional telegraph equation of distributed order. *Math. Comput. Simul.* **2024**, *217*, 405–424. [[CrossRef](#)]

8. Chen, L.; Lv, W.; Zhang, J. Jacobi–Galerkin spectral method for the fractional diffusion equation. *Numer. Methods Partial Differ. Equ.* **2026**, *42*, e70076. [\[CrossRef\]](#)
9. Cai, M.; Li, C.; Tong, W. Nonuniform L1/spectral element scheme for the time-space fractional diffusion equation: Numerical analysis and implementation. *Appl. Numer. Math.* **2026**, *224*, 303–328. [\[CrossRef\]](#)
10. Abd-Elhameed, W.M.; Al-Mehmadi, A.H.; Alsafri, N.M.A.; Alqubori, O.M.; Amin, A.K.; Atta, A.G. Numerical solution for gas dynamics equation involving Caputo-time fractional derivative using a family of shifted Chebyshev polynomials. *Fractal Fract.* **2026**, *10*, 299. [\[CrossRef\]](#)
11. Shiri, B.; Shloof, A.; Senu, N.; Baleanu, D. A rational power function-based approach for solving rational fractional differential equations. *Int. J. Optim. Control Theor. Appl.* **2026**, *16*, 370–382. [\[CrossRef\]](#)
12. Singh, P.K.; Saha Ray, S. An efficient numerical method based on Lucas polynomials to solve multi-dimensional stochastic Itô–Volterra integral equations. *Math. Comput. Simul.* **2023**, *203*, 826–845. [\[CrossRef\]](#)
13. Sabermahani, S.; Ordokhani, Y.; Rahimkhani, P. Application of generalized Lucas wavelet method for solving nonlinear fractal-fractional optimal control problems. *Chaos Solitons Fractals* **2023**, *170*, 113348. [\[CrossRef\]](#)
14. Heydari, M.H.; Razzaghi, M.; Baleanu, D. Orthonormal piecewise Vieta–Lucas functions for the numerical solution of the one- and two-dimensional piecewise fractional Galilei invariant advection-diffusion equations. *J. Adv. Res.* **2023**, *49*, 175–190. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Sharma, R.; Rajeev. A numerical approach to solve 2D fractional RADE of variable-order with Vieta–Lucas polynomials. *Chin. J. Phys.* **2023**, *86*, 433–446. [\[CrossRef\]](#)
16. Sabermahani, S.; Ordokhani, Y.; Razzaghi, M. Numerical solution of different kinds of fractional-order optimal control problems using generalized Lucas wavelets and the least squares method. *Optim. Control Appl. Methods* **2024**, *45*, 2702–2721. [\[CrossRef\]](#)
17. Kashif, M.; Singh, M.; Som, T.; Craciun, E.M. Numerical study of variable order model arising in chemical processes using operational matrix and collocation method. *J. Comput. Sci.* **2024**, *80*, 102339. [\[CrossRef\]](#)
18. Koundal, R. Treatment of fractional multi-order/multi-term differential equations: Utilizing fractional shifted Lucas polynomials. *Indian J. Pure Appl. Math.* **2025**, *56*, 1429–1442. [\[CrossRef\]](#)
19. Khajehnasiri, A.A.; Afshar Kermani, M.; Allahviranloo, T. Lucas operational matrix approach for solving the fractional Klein–Gordon equation. *Iran. J. Sci.* **2025**, *49*, 771–780. [\[CrossRef\]](#)
20. Savaşaneril, N.B.; Çevik, M. A versatile spectral matrix collocation method for tackling Bratu, Riccati, and higher order linear and nonlinear differential equations containing delay variables. *Math. Methods Appl. Sci.* **2025**, *48*, 16592–16605. [\[CrossRef\]](#)
21. Koundal, R. Comparative study on Chebyshev, Legendre, and Lucas wavelets: A review. *Numer. Heat Transf. B Fundam.* **2025**, *86*, 3450–3467. [\[CrossRef\]](#)
22. Singh, A.; Kanauiya, A.; Mohapatra, J. An efficient generalized Lucas wavelet operational matrix method for optimal control problems. *J. Comput. Appl. Math.* **2026**, *471*, 116749. [\[CrossRef\]](#)
23. 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. [\[CrossRef\]](#)
24. Mohapatra, J.; Santra, S.; Ramos, H. Analytical and numerical solution for the time fractional Black–Scholes model under jump-diffusion. *Comput. Econ.* **2024**, *63*, 1853–1878. [\[CrossRef\]](#)
25. Hamood, M.M.; Sharif, A.A.; Ghadle, K.P. A unified Haar wavelet collocation framework for fractional Volterra integro-differential equations with application to tumor-immune dynamics modeling. *Sci. Rep.* **2026**, *16*, 12552. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Pramanick, S.; Saha Ray, S. Variable-order time-fractional integro-partial differential equation with a weakly singular kernel appearing in viscoelasticity. *Comput. Appl. Math.* **2026**, *45*, 363. [\[CrossRef\]](#)
27. Alavi, J.; Aminikhah, H. An efficient parametric finite difference and orthogonal spline approximation for solving the weakly singular nonlinear time-fractional partial integro-differential equation. *Comput. Appl. Math.* **2023**, *42*, 350. [\[CrossRef\]](#)
28. Ghosh, B.; Mohapatra, J. Analysis of a second-order numerical scheme for time-fractional partial integro-differential equations with a weakly singular kernel. *J. Comput. Sci.* **2023**, *74*, 102157. [\[CrossRef\]](#)
29. Ghosh, B.; Mohapatra, J. Efficient numerical schemes based on the cubic B-spline collocation method for time-fractional partial integro-differential equations of Volterra type. *J. Appl. Math. Comput.* **2024**, *70*, 741–769. [\[CrossRef\]](#)
30. Babaei, A.; Banihashemi, S.; Parsa Moghaddam, B.; Dabiri, A. An algorithmic exploration of variable order fractional partial integro-differential equations via hexic shifted Chebyshev polynomials. *Comput. Appl. Math.* **2024**, *43*, 172. [\[CrossRef\]](#)
31. Moumen, A.; Mennouni, A.; Bouye, M. Contributions to the numerical solutions of a Caputo fractional differential and integro-differential system. *Fractal Fract.* **2024**, *8*, 201. [\[CrossRef\]](#)
32. Santra, S.; Behera, R. Simultaneous space–time Hermite wavelet method for time-fractional nonlinear weakly singular integro-partial differential equations. *Commun. Nonlinear Sci. Numer. Simul.* **2025**, *140*, 108324. [\[CrossRef\]](#)
33. Yaghoubi, S.; Aminikhah, H.; Sadri, K. Bidimensional Gegenbauer polynomials for variable-order time-fractional integro-partial differential equation with a weakly singular kernel. *Math. Methods Appl. Sci.* **2025**, *48*, 5570–5585. [\[CrossRef\]](#)

34. Ghoreyshi, A.; Abbaszadeh, M.; Zaky, M.A.; Dehghan, M. Finite block method for nonlinear time-fractional partial integro-differential equations: Stability, convergence, and numerical analysis. *Appl. Numer. Math.* **2025**, *214*, 82–103. [[CrossRef](#)]
35. Sabawi, Y.A.; Santra, S.; Das, P.; Mohammad, N.A. H^1 stability and convergence analysis of the L1/L1-2 Legendre spectral method for non-local weakly singular integro-PDEs. *Calcolo* **2025**, *62*, 42. [[CrossRef](#)]
36. Boyd, J.P. *Chebyshev and Fourier Spectral Methods*, 2nd ed.; Dover Publications: Mineola, NY, USA, 2001.
37. Shen, J.; Tang, T.; Wang, L.L. *Spectral Methods: Algorithms, Analysis and Applications*; Springer Series in Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 2011; Volume 41. [[CrossRef](#)]
38. Canuto, C.; Hussaini, M.Y.; Quarteroni, A.; Zang, T.A. *Spectral Methods: Fundamentals in Single Domains*; Springer: Berlin/Heidelberg, Germany, 2006. [[CrossRef](#)]
39. Abo-Gabal, H.; Zaky, M.A.; Doha, E.H. Fractional Romanovski–Jacobi tau method for time-fractional partial differential equations with nonsmooth solutions. *Appl. Numer. Math.* **2022**, *182*, 214–234. [[CrossRef](#)]
40. Tee, T.W.; Trefethen, L.N. A rational spectral collocation method with adaptively transformed Chebyshev grid points. *SIAM J. Sci. Comput.* **2006**, *28*, 1798–1811. [[CrossRef](#)]
41. Zeng, F.; Zhang, Z.; Karniadakis, G.E. A generalized spectral collocation method with tunable accuracy for variable-order fractional differential equations. *SIAM J. Sci. Comput.* **2015**, *37*, A2710–A2732. [[CrossRef](#)]
42. Alqubori, O.M.; Alsulami, S.A.; Atta, A.G.; Amin, A.K.; Abd-Elhameed, W.M. Numerical scheme for solving the time-fractional Huxley equation using shifted Dickson polynomials. *AIMS Math.* **2026**, *11*, 13744–13766. [[CrossRef](#)]
43. Abdelkawy, M.A.; Abd-Elhameed, W.M.; Alzahrani, S.S.; Atta, A.G.; Biswas, A. Numerical solutions via shifted Pell polynomials for third-order Rosenau–Hyman and Gilson–Pickering equations. *Mathematics* **2026**, *14*, 582. [[CrossRef](#)]
44. Shen, J.; Temam, R. Nonlinear Galerkin method using Chebyshev and Legendre polynomials I. The one-dimensional case. *SIAM J. Numer. Anal.* **1995**, *32*, 215–234. [[CrossRef](#)]
45. Mao, Z.; Shen, J. Efficient spectral–Galerkin methods for fractional partial differential equations with variable coefficients. *J. Comput. Phys.* **2016**, *307*, 243–261. [[CrossRef](#)]
46. Adel, M.; Abd-Elhameed, W.M.; Alsafri, N.M.A.; El-Naggar, M.A.; Atta, A.G. Highly accurate Galerkin approach for modified anomalous time-fractional sub-diffusion equations based on Fibonacci coefficient polynomials. *Fractal Fract.* **2026**, *10*, 514. [[CrossRef](#)]
47. Koshy, T. *Fibonacci and Lucas Numbers with Applications*; John Wiley & Sons: New York, NY, USA, 2001. [[CrossRef](#)]
48. Saleem, M.; Ali, A.; i Haq, F.; Khan, H. Numerical approximation of time-fractional nonlinear partial integro-differential equation using fractional Euler and cubic trigonometric B-spline methods. *Partial Differ. Equ. Appl. Math.* **2025**, *15*, 101223. [[CrossRef](#)]
49. Guo, J.; Xu, D.; Qiu, W. A finite difference scheme for the nonlinear time-fractional partial integro-differential equation. *Math. Methods Appl. Sci.* **2020**, *43*, 3392–3412. [[CrossRef](#)]
50. Ali, A.; Ahmad, S.; i Haq, F.; Hussain, I.; Khan, H.; Bushnaq, S. Numerical simulation of nonlinear parabolic type Volterra partial integro-differential equations using quartic B-spline collocation method. *Nonlinear Stud.* **2020**, *27*, 621–636.

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.