fractal and fractional

Article

A Robust and Versatile Numerical Framework for Modeling Complex Fractional Phenomena: Applications to Riccati and Lorenz Systems

Waleed Mohammed Abdelfattah [1,*], Ola Ragb [2], Mohamed Salah [2] and Mokhtar Mohamed [3,*]

1 College of Engineering, University of Business and Technology, Jeddah 23435, Saudi Arabia
2 Department of Engineering Mathematics and Physics, Faculty of Engineering, Zagazig University, Zagazig 44519, Egypt; ormohamed@eng.zu.edu.eg (O.R.); msalaheldin@zu.edu.eg (M.S.)
3 Basic Science Department, Faculty of Engineering, Delta University for Science and Technology, Gamasa 11152, Egypt
* Correspondence: w.abdelfattah@ubt.edu.sa (W.M.A.); mokhtar.alsaidi@deltauniv.edu.eg (M.M.)

Abstract: The fractional differential quadrature method (FDQM) with generalized Caputo derivatives is used in this paper to show a new numerical way to solve fractional Riccati equations and fractional Lorenz systems. Unlike previous FDQM applications that have primarily focused on linear problems, our work pioneers the use of this method for nonlinear fractional initial value problems. By combining Lagrange interpolation polynomials and discrete singular convolution (DSC) shape functions with the generalized Caputo operator, we effectively transform nonlinear fractional equations into algebraic systems. An iterative method is then utilized to address the nonlinearity. Our numerical results, obtained using MATLAB, demonstrate the exceptional accuracy and efficiency of this approach, with convergence rates reaching 10^{-8}. Comparative analysis with existing methods highlights the superior performance of the DSC shape function in terms of accuracy, convergence speed, and reliability. Our results highlight the versatility of our approach in tackling a wider variety of intricate nonlinear fractional differential equations.

Keywords: fractional derivative; generalized Caputo; differential quadrature technique; discrete singular convolution; fractional Riccati; fractional Lorenz system

Citation: Abdelfattah, W.M.; Ragb, O.; Salah, M.; Mohamed, M. A Robust and Versatile Numerical Framework for Modeling Complex Fractional Phenomena: Applications to Riccati and Lorenz Systems. *Fractal Fract.* **2024**, *8*, 647. https://doi.org/10.3390/fractalfract8110647

Academic Editors: Faranak Rabiei, Dongwook Kim and Zeeshan Ali

Received: 16 September 2024
Revised: 30 October 2024
Accepted: 5 November 2024
Published: 6 November 2024

1. Introduction

Many phenomena in chemistry, biology, acoustics, psychology, control theory, rheology, damping laws, diffusion processes, and other fields of science have been successfully modeled using fractional-order derivatives in recent years. This is because fractional calculus can be used to successfully model a physical phenomenon that is dependent not only on the time instant but also on the prior time history [1–6]. Hence, numerous physical problems are defined by fractional differential equations (FDEs), and solving these equations has been the focus of several studies in recent years. Several techniques have recently been developed to solve FDEs, including numerical and analytical techniques. Various methods, including homotopy perturbation [7–9], homotopy analysis [10,11], Taylor matrix [12], Adomian decomposition [11], and Haar wavelet [13] methods, have been employed to solve the fractional-order Riccati differential equation. Unfortunately, the convergence region for the corresponding outcomes is relatively limited. The fractional derivative operator's unique properties can make the numerical solution of fractional equations challenging, particularly in high-dimensional spaces. To address this challenge, numerical methods such as the Finite Difference Method (FDM) [14–17], Galerkin [18–21], Collocation [22–25], and finite volume element methods [26–28] have been utilized to tackle such fractional equations.

Liu et al. [29] introduced a radial basis function finite difference approach for studying the time fractional convection equation. Saadeh [30] employed the finite-difference and space finite-volume techniques to solve fractional diffusion equations. To address a two-dimensional space fractional diffusion equation, Tuan et al. [31] employed finite difference discretization with Caputo derivatives. Devshali and Arora [32] proposed differential transform and differential quadrature methods for solving the fractional diffusion equation. Odibat and Momani [9] utilized a modified homotopy perturbation technique to solve fractional Riccati differential equations (FRDEs). Khader [33] applied the fractional Chebyshev FDM to solve FRDEs. Li et al. [34] solved FRDEs via the quasi-linearization method. Sakar et al. [34] explained an iterative reproducing kernel Hilbert space technique to obtain the solutions of FRDEs. Agheli [35] explained an iterative reproducing kernel Hilbert space technique to obtain the solutions of FRDEs. Agheli [36] presented numerical solutions for solving FRDEs using trigonometric basic functions. Liu et al. [37] offered the Laplace transform and quadrature rule with Caputo sense to solve FRDEs.

In the last ten years, chaos has emerged as a popular topic in fractional calculus [38]. The chaotic behavior becomes more complicated because the equation contains fractional orders. Numerical methods were developed to analyze nonlinear dynamics to better understand physical phenomena. To find numerical solutions to various nonlinear fractional differential equations, the predictor–corrector method (P-C) was developed [39–41]. Fuzzy fractional differential and fractional delay equations [42–45] demonstrate the emergence of new trends in fractional differential equations.

Despite yielding fruitful results, finding more general chaotic differential equations remains an intriguing task. A generalized fractional derivative was recently proposed in [46]. Fractional derivatives have demonstrated superior performance compared to regular derivatives in several respects and may have even more real-world applications. One such application is in image encryption, where fractional differential equations have been suggested as a means of introducing chaos [47,48]. This means that image encryption results can be made more secure by using fractional chaotic equations with two parameters. This derivative was recently proposed in quantum mechanics [49]. Furthermore, two-parameter models in control theory and diffusion issues can have degrees of freedom in control and fitting. This derivative and its applications are depicted as a new direction in fractional calculus. Li and Chen [50] demonstrated the chaotic behaviors in the fractional order Chen system. Alomari [51] used the step homotopy analysis technique to solve the fractional chaotic Chen system. Luo and Wang [44] solved chaos in the fractional-order complex Lorenz system and its synchronization. Petráš [2] introduced a new classification of the fractional-order Lorenz-type systems. Erturka and Kumar [52] presented a solution for a COVID-19 model using new generalized Caputo-type fractional derivatives. Xu et al. [53] studied numerical and analytical solutions of a new generalized fractional diffusion equation. Kumar et al. [54] proposed a new technique to solve generalized Caputo-type FDEs with the example of a computer virus model.

The primary goal of this paper is to apply the novel fractional Differential Quadrature Method (FDQM) with generalized Caputo definition fractional to solve nonlinear initial value fractional problems. Two different shape functions, the Lagrange interpolation [55,56] and the regularized Shannon kernel [57–60], have been successfully employed to address initial value problems involving fractional derivatives. To exhibit the efficacy, efficiency, and capabilities of the proposed algorithm, two test problems were investigated, including FRDEs and the fractional Lorenz system. Then, by the proposed methods, the given problems are reduced to a system of nonlinear algebraic equations, and by solving this system via the iterative method, we obtain the solution of FRDEs and the fractional Lorenz system. Furthermore, we create a MATLAB code for each approach to obtain a numerical solution for the two problems under consideration. A comparison between the computed results and previous analytical and numerical [61–65] methodologies is included to demonstrate the validity and applicability of the proposed methods. Furthermore, we conducted some parametric investigations to showcase the reliability of our techniques in the presence of fractional order derivatives.

This paper introduces a novel fractional Differential Quadrature Method (FDQM) to solve nonlinear initial value fractional problems. This method employs the generalized Caputo fractional derivative and utilizes Lagrange interpolation and the Regularized Shannon kernel as shape functions. Numerical simulations demonstrate the method's superior accuracy, efficiency, and versatility in handling fractional Riccati and Lorenz systems. The FDQM's potential applications extend to various fields where it can be used to model complex systems with memory effects and nonlinearities [66].

2. Formulation of the Problem

The following two nonlinear fractional differential equations serve as examples to illustrate the capabilities of our proposed methods:

2.1. Our 1st Example Is the Fractional Riccati Equation

$$\frac{d^{\alpha,\rho}v(t)}{dt} = 2v(t) - v^2(t) + 1, \qquad \text{when } (0 < \ t \text{ and } 0 < \alpha \ \leq 1) \tag{1}$$

where $\frac{d^{\alpha,\rho}}{dt}$ is the operator of the generalized Caputo-type fractional derivative [61].

Also, the initial condition for FRDE is:

$$v(0) = 0 \tag{2}$$

In addition, the exact solution of FRDE at $\alpha = 1$, and $\rho = 1$ is given by [67]:

$$v(x,t) = 1 + \sqrt{2}\tanh\left[\sqrt{2}t + \log\sqrt{\frac{\sqrt{2}-1}{\sqrt{2}+1}}\right] \tag{3}$$

2.2. Our 2nd Example Is the Fractional Lorenz System

$$\frac{d^{\alpha,\rho}X(t)}{dt} = \lambda(Y(t) - X(t)) \tag{4}$$

$$\frac{d^{\alpha,\rho}Y(t)}{dt} = (\phi - \lambda)X(t) - X(t)Z(t) + \phi Y(t) \tag{5}$$

$$\frac{d^{\alpha,\rho}Z(t)}{dt} = X(t)Y(t) - \beta\, Z(t) \tag{6}$$

where λ, ϕ, and β are constant parameters $\in \mathbb{R}$ that affect chaotic behavior.

Consequently, the fractional Lorenz system is subject to the following initial condition:

$$X(0) = x_0,\ Y(0) = y_0,\ Z(0) = z_0 \tag{7}$$

3. Method of Solution

This paper introduces a novel application of the FDQM to solve initial value fractional problems. We employ two distinct shape functions, Lagrange interpolation and the regularized Shannon kernel, in conjunction with the generalized Caputo fractional derivative to transform fractional problems into nonlinear algebraic systems.

We begin by defining a fractional derivative, of which several definitions exist. In this work, we utilize the recently proposed generalized Caputo definition.

3.1. Generalized Caputo-Kind Fractional Derivative [63]

The fractional derivative has good memory effects compared to ordinary calculus. FDEs are realized in model problems in fluid flow, viscoelasticity, finance, engineering, and other areas of applications.

Caputo's Fractional Derivative

A concise overview of Caputo's fractional derivative is presented in this section. This definition, which is derived from the Riemann–Liouville Fractional Derivative [68], is explained in greater detail in our prior publication [69].

Suppose $\alpha \in R^{+}$, If $\mathbb{N}$ is a positive integer, and $\mathbb{N}-1 < \alpha \le \mathbb{N}$. According to Riemann–Liouville fractional, which is one of the most researched definitions, the fraction derivative of a function $v(t)$ of order α is defined as follows:

$$D_c^{\alpha} v(t) = \frac{1}{\Gamma(\mathbb{N}-\alpha)} \frac{d^{\mathbb{N}}}{dt^{\mathbb{N}}} \int_c^t (t-x)^{\mathbb{N}-\alpha-1}\, v^{\mathbb{N}}(x)\, dx \tag{8}$$

Generalized Caputo's Fractional Derivative of operator $D_{c+}^{\alpha,\rho}$ and order α is defined as [52–54]:

$$D_{c+}^{\alpha,\rho} v(t) = \frac{\rho^{\alpha-\mathbb{N}+1}}{\Gamma(\mathbb{N}-\alpha)} \int_c^t x^{\rho-1}(t^{\rho}-x^{\rho})^{\mathbb{N}-\alpha-1} \left(x^{1-\rho}\frac{d}{dx}\right)^{\mathbb{N}} v(x)\, dx, \quad \mathbb{N}-1 < \alpha < \mathbb{N},\ \ \rho > 0,\ \ c \ge 0 \tag{9}$$

where c is the lower limit of integration.

Consequently, the solution to Equation (9) can be written as [52–54]:

$$D_{c+}^{\alpha,\rho} (t^{\rho}-x^{\rho})^{\gamma} = \rho^{\alpha} \frac{\Gamma(\gamma+1)}{\Gamma(\gamma-\alpha+1)} (t^{\rho}-c^{\rho})^{\gamma-\alpha} \tag{10}$$

Next, we will discuss the differential quadrature technique using the specified shape functions.

3.2. Our First Shape Function Is Lagrange Interpolation Polynomial Based DQM (PDQM)

Within this partition, the functional values of any unknown P at a specific set of N grid points can be represented using this shape function, as described in [55,56]:

$$\mathrm{P}(\mathrm{t_i}) = \sum_{\mathrm{j}=1}^{\mathrm{N}} \frac{1}{\mathrm{t_i}-\mathrm{t_j}} \times \frac{\prod_{\mathrm{k}=1}^{\mathrm{N}} (\mathrm{t_i}-\mathrm{t_k})}{\prod_{\mathrm{j}=1,\mathrm{j}\ne\mathrm{k}}^{\mathrm{N}} (\mathrm{t_j}-\mathrm{t_k})} \mathrm{P}(t_j), \quad (\mathrm{i}=1:\mathrm{N}) \tag{11}$$

As a result, the following are the different derivatives of this unknown P:

$$\left.\frac{\partial^{\mathrm{r}}\mathrm{P}}{\partial \mathrm{t}^{\mathrm{r}}}\right|_{\mathrm{t}\,=\,\mathrm{t_i}} = \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{a}_{\mathrm{ij}}^{(\mathrm{r})} \mathrm{P}(\mathrm{t_j}), \quad (\mathrm{i}=1:\mathrm{N}) \tag{12}$$

where $\mathrm{a}_{\mathrm{ij}}^{(\mathrm{r})}$ is the rth derivative weighting coefficient. However, determining the weighting coefficients is critical to DQM accuracy. As a result, they differ based on the shape function.

Differentiating Equation (12) results in the calculation of the weighting coefficients $\mathrm{a}_{\mathrm{ij}}^{(1)}$ and $\mathrm{a}_{\mathrm{ij}}^{(2)}$, representing the first and second derivatives.

$$\mathrm{a}_{\mathrm{ij}}^{(1)} = \begin{cases} \dfrac{1}{(\mathrm{t_i}-\mathrm{t_j})} \displaystyle\prod_{\substack{\mathrm{k}=1,\\ \mathrm{k}\ne\mathrm{i,j}}}^{\mathrm{N}} \dfrac{(\mathrm{t_i}-\mathrm{t_k})}{(\mathrm{t_j}-\mathrm{t_k})} & \mathrm{i}\ne\mathrm{j} \\ -\displaystyle\sum_{\substack{\mathrm{j}=1,\\ \mathrm{j}\ne\mathrm{i}}}^{\mathrm{N}} \mathrm{a}_{\mathrm{ij}}^{(1)} & \mathrm{i}=\mathrm{j} \end{cases}, \quad \mathrm{a}_{\mathrm{ij}}^{(2)} = \left[\mathrm{a}_{\mathrm{ij}}^{(1)}\right]\left[\mathrm{a}_{\mathrm{ij}}^{(1)}\right] \tag{13}$$

The distribution of grid points N, whether uniform or non-uniform, significantly influences the accuracy of the PDQM results. The non-uniform distribution is defined by the following equation, based on Chebyshev's distribution:

$$x_i = \frac{1}{2}L_x\left(1 - \cos\left(\frac{\pi(i-1)}{N-1}\right)\right), \qquad (i = 1:N) \tag{14}$$

3.3. Discrete Singular Convolution-Based DQM (DSCDQM)

In this part, according to previous research [59,70,71], singular convolution can be expressed as follows:

$$Y(t) = (F * H)(t) = \int_{-\infty}^{\infty} F(t-s)\, H(s)\, ds \tag{15}$$

$F(t-s)$ and $H(t)$ denote a singular kernel and a test function space element, respectively.

The choice of kernel type determines the shape function used in this technique. Given the variety of available kernels, we have selected the kernel demonstrated to have the highest accuracy [69] to represent the functional values of the unknown P and its derivatives at a specified number of grid points N:

- Our second shape function is the Regularized Shannon kernel (DSCDQM–RSK)

$$P(t_i) = \sum_{j=-M}^{M} \left\langle \frac{\sin\left[\frac{\pi(t_i - t_j)}{\Delta}\right]}{\frac{\pi(t_i - t_j)}{\Delta}} \exp\left(\frac{-(t_i - t_j)^2}{2\sigma^2}\right) \right\rangle P(x_j), \qquad (i = -N:N), \sigma = (h \times \Delta) > 0 \tag{16}$$

The parameters σ, h, and Δ represent the Regularized Shannon factor, the computational parameter, and the mesh size, respectively.

$$\frac{\partial^r P}{\partial t^r}\Big|t = t_i = \sum_{j=1}^{N} a_{ij}^{(r)} P(t_j) \qquad (i = -N:N) \tag{17}$$

Differentiating Equation (15) allows us to determine the coefficients $a_{ij}^{(1)}$ and $a_{ij}^{(2)}$, as described in [72]:

$$a_{ij}^{(1)} = \begin{cases} \frac{(-1)^{i-j}}{\Delta(i-j)} \exp\left(-\Delta^2\left(\frac{(i-j)^2}{2\sigma^2}\right)\right), & i \neq j \\ 0 & i = j \end{cases}, \quad a_{ij}^{(2)} = \begin{cases} \left[2\frac{(-1)^{i-j+1}}{a^2(i-j)^2} + \frac{1}{\sigma^2}\right] \exp\left(-\Delta^2\left(\frac{(i-j)^2}{2\sigma^2}\right)\right), & i \neq j \\ -\frac{1}{\sigma^2} - \frac{\pi^2}{3\Delta^2} & i = j \end{cases} \tag{18}$$

The kernel type, grid points (N), and bandwidth (2K + 1) are all important parameters that affect the convergence and accuracy of the solutions, as our analysis shows.

Now, after mentioning DQM based on two shape functions, we will demonstrate the effect of the generalized Caputo's fractional derivative, which is shown in Equation (9) on the FDQM in Equation (12), to determine the weighting coefficients a_{ij}^{α} for $\alpha \in (0,1]$ and $\rho > 0$, as follows [69]:

$$D_{c_+}^{\alpha,\rho} P(t) = \begin{cases} \sum_{j=1}^{N} a_{ij}^{\alpha,\rho}\, P(t_j, x), & 0 < \alpha \leq 1\ ,\ \rho > 0 \\ \sum_{j=1}^{N} a_{ij}^{(1)}\, P(t_j, x) & \alpha = \rho = 1 \end{cases} \tag{19}$$

Hence, the weighting coefficient $a_{ij}^{\alpha,\rho}$ is calculated as:

$$a_{ij}^{\alpha,\rho} = A^{1-\alpha}\rho^{\alpha}\, a_{ij}^{(1)} + \frac{\rho^{\alpha} a_{1,j}^{(1)}}{\Gamma(2-\alpha)}\left(t^{\rho} - c^{\rho}\right)^{1-\alpha}, \quad A_{ij} = a_{ij}^{(1)} - a_{1j}^{(1)} \tag{20}$$

3.4. Algorithm: Fractional Differential Quadrature Method (FDQM) for Nonlinear Initial Value Problems

This pseudo-code (Algorithm 1) outlines the key steps for implementing the proposed numerical framework for solving nonlinear fractional differential equations using the FDQM approach.

Algorithm 1: Fractional Differential Quadrature Method (FDQM) for Nonlinear Initial Value Problems

---

Input:

- Fractional order, constants in the fractional differential equation
- Nonlinear fractional differential equation.
- Initial conditions.
- Grid points (N)
- Shape functions (Lagrange, Regularized Shannon)
- Tolerance for convergence (ε)

Output:

- The solution of the fractional differential equation numerically.

1. Define the generalized Caputo fractional derivative operator.
2. Initialize grid points $x_1, x_2, \ldots, x_N$ based on Chebyshev distribution or uniform distribution.
3. For each time step "t:
   - Construct the shape functions using Lagrange interpolation and Regularized Shannon kernel.
   - Formulate the algebraic system from the fractional differential equation using FDQM:

   a. Apply shape functions to approximate the unknown function and its derivatives.
   b. Substitute into the original FDE to derive a system of nonlinear algebraic equations.
4. Initialize solution guess for the unknown function.
5. While not converged ($||f_{new} - f_{old}|| > \varepsilon$)
6. Solve the algebraic system iteratively:
   a. Use a numerical method (iterative differential quadrature method) to update the solution.
   b. Update f_{old} with f_{new}
7. End while
8. Return the numerical solution for the fractional differential equation at the specified time steps.

End Algorithm

---

4. Numerical Results

Now that it is easier to understand FDQM with different shape functions such as PDQM [55,56], and DSCDQM–RSK [59,70,71] based on the generalized Caputo definition fractional derivative, two examples will be given here and then will be discussed. In all these examples, MATLAB software(R2022b) is used for computations and graphs. The primary goal of this article is to learn about the performance, validity, efficiency, and accuracy of developed techniques by comparing the computed results to previous numerical and analytical solutions.

4.1. Problem 4.1

We introduce the first example fractional Riccati equation after substituting the Equations (19) and (20) for the proposed methods in Equation (1) as follows:

$$\sum_{j=1}^{L} a_{ij}^{\alpha,\rho}\,\upsilon(t_j) = 2\sum_{j=1}^{N}\delta_{ij}\,\upsilon(t_j) - \left(\sum_{j=1}^{N}\delta_{ij}\,\upsilon(t_j)\right)^2 + 1 \tag{21}$$

The governing Equation (21) is also used to deal with the initial condition (2). To solve the nonlinear problem, the iterative method is applied [55,72,73]. As a first step, the governing equation is solved as a linear system. Then we solve them iteratively as a nonlinear system until we reach the requisite convergence, which is as follows:

$$\left|\frac{\upsilon_{m+1}}{\upsilon_m}\right| < 1, \qquad \text{where } m = 0, 1, 2, \ldots \tag{22}$$

Also, to assess the convergence and accuracy of the developed methods, we use the error computation method:

$$L_\infty \text{ Error} = \max_{1\le i\le N} \left|\upsilon_{numerical} - \upsilon_{exact}\right| \tag{23}$$

Now, the obtained results will be demonstrated as follows:

The effect of applying PDQM with uniform and non-uniform grid distributions on the computation of the fractional Riccati equation with fractions $(\alpha = 1,\ \rho = 1)$ at different grid points (N) and times (T) is shown in Table 1. Hence, it is found that non-uniform grid results are higher and more consistent with earlier solutions than uniform ones with an error $\le 10^{-8}$, and execution time of about (0.024 s). Also, when the grid points increase with time, the accuracy increases; for example, at time (t = 1), we make (N = 13), and at time (t = 2), we make (N = 26) Furthermore, the maximum number of grids we use is significantly less than in previous studies (N = 3200).

Table 2 compares non-uniform PDQM and DSCDQM–RSK for the fractional Riccati equation under various conditions of time (T = 1), fraction $(\alpha = 1,\ \rho = 1)$, different grid points (N), regularized Shannon factor $(\sigma = h \times \Delta)$, and bandwidth (2K + 1). To begin, Table 2 ensures that the best value of the regularized Shannon factor is $\sigma = 0.45 \times \Delta$, with results matching previous studies and PDQM shown in Table 1 at the fewest grid points (N = 9), bandwidth (2K + 1 = 7) and performance time of about (0.018 s). DSCDQM–RSK is the best method overall, based on low grid points and performance time when compared to PDQM (N = 16) and previous studies (N = 640).

The efficiency, validity, and accuracy of the created methodologies are presently being explored by comparing the calculated results to earlier numerical and analytical solutions at various powers of fraction (α, ρ), as shown in Tables 3–6. Tables 3–6 show that increasing the fraction power $(\alpha \text{ or } \rho)$ decreases the value of $\upsilon(t)$, but increases with time for the fractional Riccati equation. Furthermore, the results show that DSCDQM–RSK outperforms non-uniform PDQM in terms of efficiency, validity, and accuracy.

In addition, the dynamic behaviors of the fractional Riccati equation with respect to the parameters $(\alpha \text{ and } \rho)$ and against the time variable t are depicted in Figures 1 and 2.

Table 1. Computation of $\upsilon(t)$ via uniform and non-uniform PDQM for fractional Riccati equation with fraction $(\alpha = 1,\ \rho = 1)$ at different grid points (N), and times (T).

T	PDQ Solutions							Previous Solutions		
	N	Uniform	CPU (s)	Error	Non-Uniform	CPU (s)	Error	N	Earlier Numerical [64]	Exact [61]
1	4	1.59030488	0.016	0.0971	1.64023865	0.013	0.04721	10	1.68745117	1.68949839
	6	1.68745374	0.018	0.0015	1.67122383	0.017	0.01774	20	1.68896723	
	8	1.68821427	0.018	0.0011	1.68941502	0.017	5.16×10^{-5}	40	1.68936339	
	9	1.68921673	0.019	0.0002	1.68948616	0.018	2.18×10^{-5}	80	1.68946438	
	11	1.68941775	0.020	7.2×10^{-5}	1.68949815	0.018	8.29×10^{-6}	160	1.68948986	
	12	1.68948043	0.021	1.6×10^{-5}	1.68949820	0.020	1.95×10^{-6}	320	1.68949625	
	13	1.68949755	0.021	3.1×10^{-7}	1.68949839	0.020	1.2×10^{-8}	640	1.68949786	
2	8	2.34267643	0.029	0.001168	2.35647559	0.019	0.012631	20	2.35530727	2.35777165
	12	2.35721628	0.031	0.00056	2.35777266	0.020	3.73×10^{-6}	40	2.35721255	
	16	2.35757661	0.032	0.00014	2.35777175	0.021	6.14×10^{-5}	80	2.35763805	
	18	2.35773329	0.033	3.27×10^{-5}	2.35777169	0.021	5.68×10^{-6}	160	2.35773897	
	22	2.35777129	0.034	8.08×10^{-6}	2.35777165	0.022	7.72×10^{-6}	320	2.35776357	
	24	2.35777151	0.035	2.01×10^{-6}	2.35777165	0.023	1.87×10^{-6}	640	2.35776964	
	26	2.35777167	0.036	5×10^{-7}	2.35777165	0.024	2×10^{-8}	1280	2.35777115	
5	20	2.41421578	0.033	1.743×10^{-5}	2.41420169	0.021	3.34×10^{-6}	50	2.41419835	2.41420167
	21	2.41420238	0.033	1.37×10^{-6}	2.41420169	0.021	6.8×10^{-7}	100	2.41420101	
	22	2.41420214	0.034	6.2×10^{-7}	2.41420168	0.022	1.6×10^{-7}	200	2.41420152	
	23	2.41420177	0.034	1.4×10^{-7}	2.41420167	0.022	4×10^{-8}	400	2.41420163	
	24	2.41420175	0.035	9×10^{-8}	2.41420167	0.023	1×10^{-8}	800	2.41420166	
	25	2.41420171	0.035	4×10^{-8}	2.41420167	0.023	1×10^{-8}	1600	2.41420167	
	26	2.41420169	0.036	2×10^{-8}	2.41420167	0.024	1×10^{-8}	3200	2.41420167	

Table 2. Computation of $\upsilon(t)$ via non-uniform PDQM and DSCDQM–RSK for fractional Riccati equation with time (T = 1) and fraction $(\alpha = 1,\ \rho = 1)$ at various grid points (N), regularized Shannon factor $(\sigma = h \times \Delta)$, and bandwidth (2K + 1).

N	2K + 1	Non-Uniform PDQM	DSCDQM–RSK				
			σ=0.2×Δ	σ=0.4×Δ	σ=0.45×Δ	σ=0.5×Δ	CPU (s)
9	3	1.68948616	1.69745751	1.689880214	1.68950741	1.64023989	0.008
	5	1.68948616	1.69745647	1.689876547	1.68949956	1.64023942	0.01
	7	1.68948616	1.69745559	1.689875120	1.68949839	1.64023865	0.012
	9	1.68948616	1.69745559	1.689875120	1.68949839	1.64023865	0.014
	11	1.68948616	1.69745559	1.689875120	1.68949839	1.64023865	0.016
11	3	1.68949815	1.69745666	1.689877415	1.68949951	1.64023937	0.009
	5	1.68949815	1.69745559	1.689875120	1.68949839	1.64023865	0.01
	7	1.68949815	1.69745559	1.689875120	1.68949839	1.64023865	0.011
	9	1.68949815	1.69745559	1.689875120	1.68949839	1.64023865	0.012
	11	1.68949815	1.69745559	1.689875120	1.68949839	1.64023865	0.014
13	3	1.68949839	1.69745578	1.689876014	1.6894990	1.64023900	0.01
	5	1.68949839	1.69745559	1.689875120	1.68949839	1.64023865	0.012
	7	1.68949839	1.69745559	1.689875120	1.68949839	1.64023865	0.014
	9	1.68949839	1.69745559	1.689875120	1.68949839	1.64023865	0.016
	11	1.68949839	1.69745559	1.689875120	1.68949839	1.64023865	0.018
Earlier numerical solutions [64]			1.68949786 at (N = 640)				
Exact [61]			1.68949839				

Table 3. Computation of $\upsilon(t)$ via non-uniform PDQM and DSCDQM–RSK for fractional Riccati equation at various times (T), fractions (α), and $\rho = 1$.

T	Non-Uniform PDQM		DSCDQM–RSK		Previous Results [61–65]	
	α = 0.5	α = 0.75	α = 0.5	α = 0.75	α = 0.5	α = 0.75
0.1	0.59149371	0.24512554	0.59149373	0.24512556	0.59149373	0.24512556
0.2	0.93141486	0.47450194	0.93141488	0.47450196	0.93141488	0.47450196
0.3	1.171926469	0.709154008	1.171926471	0.709154010	1.171926471	0.709154010
0.4	1.344407759	0.937441050	1.344407761	0.937441052	1.344407761	0.937441052
0.5	1.471501155	1.147807349	1.471501157	1.147807351	1.471501157	1.147807351
0.6	1.568070430	1.332985034	1.568070432	1.332985036	1.568070432	1.332985036
0.7	1.643596282	1.490535187	1.643596284	1.490535189	1.643596284	1.490535189
0.8	1.704182955	1.621592245	1.704182957	1.621592247	1.704182957	1.621592247
0.9	1.753855962	1.729220378	1.753855964	1.729220380	1.753855964	1.729220380
1	1.817133594	1.795344168	1.817133596	1.795344170	1.817133596	1.795344170

Table 4. Computation of $\upsilon(t)$ via non-uniform PDQM and DSCDQM–RSK for fractional Riccati equation with fraction $(\alpha = 1)$ at various times (T), and fractions (ρ).

T	Non-Uniform PDQM		DSCDQM–RSK		Previous Results [61–65]	
	ρ = 0.8	ρ = 1.2	ρ = 0.8	ρ = 1.2	ρ = 0.8	ρ = 1.2
0.1	0.14117992	0.09045268	0.14117994	0.09045271	0.14117994	0.09045271
0.2	0.31592641	0.195667845	0.31592645	0.19566787	0.31592645	0.19566787
0.3	0.52298485	0.315926409	0.52298488	0.315926411	0.52298488	0.315926411
0.4	0.75601439	0.450653813	0.75601442	0.450653816	0.75601442	0.450653816
0.5	1.00354951	0.59824597	1.00354953	0.59824599	1.00354953	0.59824599
0.6	1.25086733	0.75601439	1.25086736	0.75601442	1.25086736	0.75601442
0.7	1.48329584	0.92030072	1.48329586	0.92030075	1.48329586	0.92030075
0.8	1.68949839	1.08677371	1.68949842	1.08677374	1.68949842	1.08677374
0.9	1.86328744	1.25086733	1.86328746	1.25086736	1.86328746	1.25086736
1	2.00353694	1.40827080	2.00353696	1.40827081	2.00353696	1.40827081

Table 5. Computation of $\upsilon(t)$ via non-uniform PDQM for fractional Riccati equation at different grid points (N), and fractions $(\alpha,\ \rho)$.

T	Non-Uniform				Earlier Numerical Solutions [62]			
	N	α = 1, ρ = 0.9	α = 0.95, ρ = 0.75	α = 0.9, ρ = 1.2	N	α = 1, ρ = 0.9	α = 0.95, ρ = 0.75	α = 0.9, ρ = 1.2
1	4	1.80602802	1.96263496	1.39368786	10	1.84281224	2.06729863	1.52944766
	6	1.84319793	2.04896144	1.49050147	20	1.84491385	2.07202706	1.53119172
	8	1.84556010	2.06510470	1.51019655	40	1.84546411	2.07322261	1.53167452
	9	1.84561319	2.06904164	1.52085146	80	1.84560424	2.07352741	1.53180584
	11	1.84565025	2.07362649	1.52696219	160	1.84563955	2.07360571	1.53184129
	12	1.84565137	2.07363589	1.52945738	320	1.84564841	2.07362592	1.53185082
	13	1.84565137	2.07363256	1.53185408	640	1.84565063	2.07363115	1.53185339
	14	1.84565137	2.07363256	1.53185408	1280	1.84565119	2.07363250	1.53185407
2	8	2.28430436	2.32839830	2.20693947	20	2.36576348	2.34646084	2.26631061
	12	2.32381757	2.33437164	2.21000463	40	2.36763874	2.34834846	2.26840179
	16	2.34381756	2.34307946	2.22083045	80	2.36805246	2.34876916	2.26890814
	18	2.36382620	2.34836151	2.23103727	160	2.36815011	2.34887135	2.26903810
	22	2.36832617	2.34863032	2.24089052	320	2.36817385	2.34889710	2.26907235
	24	2.36818255	2.34889017	2.26879047	640	2.36817971	2.34890369	2.26908148
	26	2.36818153	2.34890584	2.26908459	1280	2.36818116	2.34890540	2.26908393
	27	2.36818153	2.34890584	2.26908459	2560	2.36818153	2.34890584	2.26908459

Table 6. Computation of $\upsilon(t)$ via DSCDQM–RSK for fractional Riccati equation at different values of $(\alpha,\ \rho)$ and times.

α	ρ	T = 0.5	T = 1	T = 2	T = 2.5	T = 3
0.4	1	1.58967600	1.77525996	2.09537035	2.21247852	2.22642314
	1.1	1.54739283	1.74403298	2.07553739	2.19214756	2.20554878
	1.2	1.50756398	1.71405017	2.05317313	2.17021456	2.18875143
	1.4	1.43438179	1.65736347	2.02666948	2.14654879	2.16214787
	1.9	1.28182554	1.53134020	2.01520768	2.12958092	2.13478462
0.5	1	1.68300421	1.79935747	2.12065899	2.25983372	2.27664509
	1.1	1.66554391	1.75798866	2.10285480	2.23984120	2.25471201
	1.2	1.65154233	1.71762640	2.09852919	2.21874621	2.23789123
	1.4	1.62320069	1.66991063	2.07719592	2.20997411	2.21987423
	1.9	1.58526142	1.56342314	2.05661402	2.20278414	2.17645789
0.7	1	1.71613371	1.82286926	2.14596688	2.29139917	2.31617003
	1.1	1.69532577	1.79031758	2.12529159	2.27075056	2.29157030
	1.2	1.67462609	1.77811144	2.10299487	2.25941935	2.26895529
	1.4	0.65934200	1.73806497	2.08048316	2.22364828	2.24318890
	1.9	0.60930946	1.65504250	2.05990416	2.20861195	2.20862313
0.85	1	0.73587574	1.84052325	2.24870001	2.30150738	2.33501568
	1.1	0.71892188	1.82175475	2.21842773	2.28982006	2.31048992
	1.2	0.69728893	1.80614145	2.18322274	2.27519396	2.29534657
	1.4	0.67897490	1.77168741	2.16870824	2.23587987	2.27096577
	1.9	0.62629628	1.72273733	2.13184852	2.20356265	2.22777374
0.95	1	0.82414933	1.87127626	2.31830946	2.36707298	2.37441750
	1.1	0.73300489	1.84609224	2.28379937	2.35373760	2.35930723
	1.2	0.71834905	1.82980945	2.24083446	2.33559233	2.33776914
	1.4	0.68447266	1.80982469	2.21190242	2.28289347	2.31913452
	1.9	0.63618443	1.77131197	2.19457437	2.23111789	2.29258450

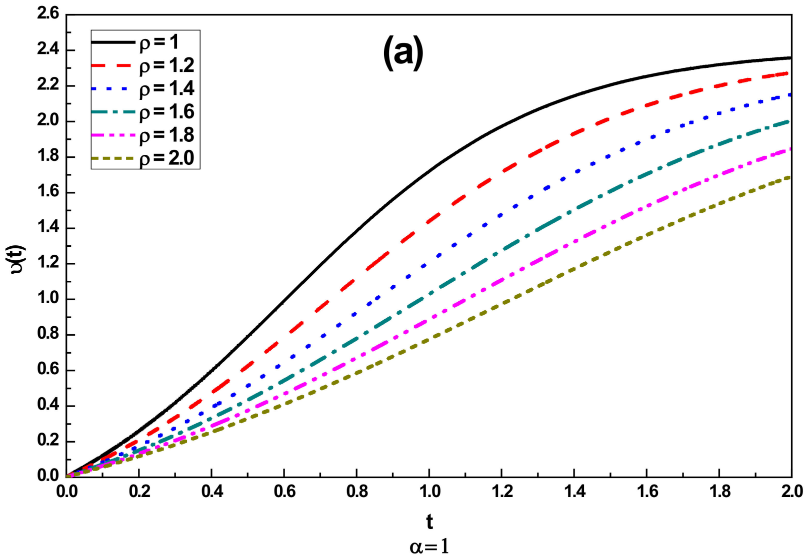

Figure 1. *Cont.*

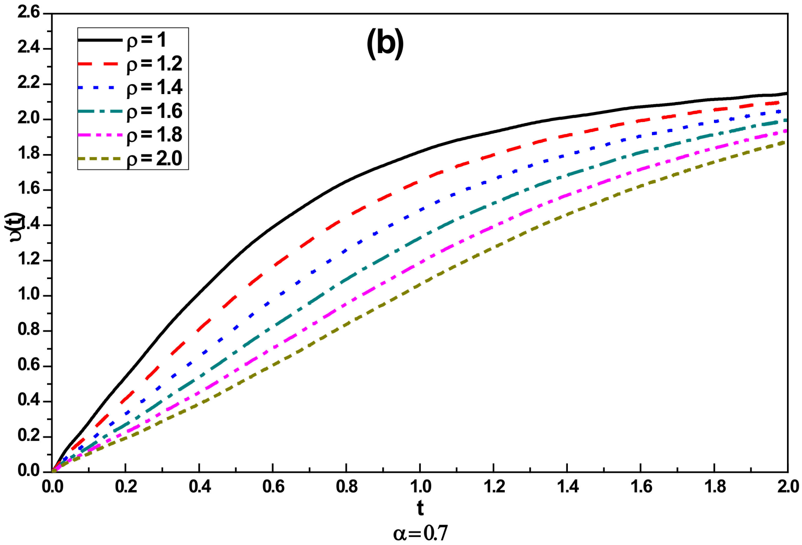

Figure 1. Numerical simulation of $\upsilon(t)$ using DSCDQM–RSK for fractional Riccati equation at different times and fraction power $(\alpha,\ \rho)$ for (**a**) $\alpha = 1$, and (**b**) $\alpha = 0.7$.

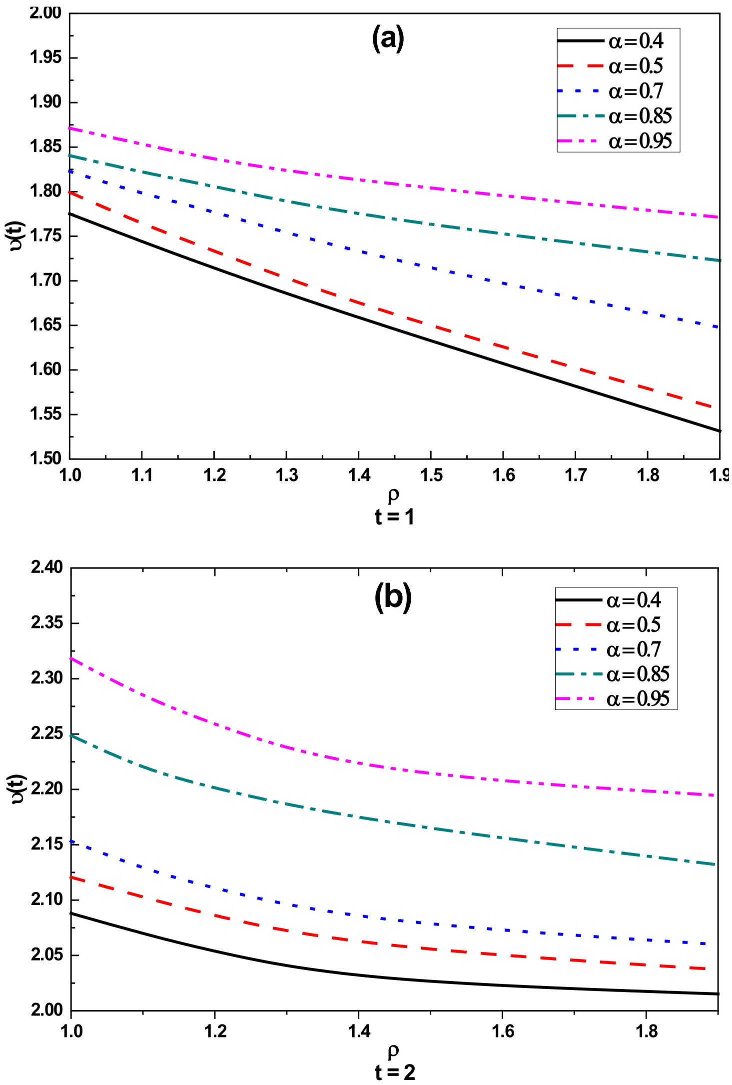

Figure 2. Numerical simulation of $\upsilon(t)$ using DSCDQM–RSK for fractional Riccati equation at different fraction power $(\alpha,\ \rho)$ for (**a**) t = 1, and (**b**) t = 2.

4.2. Problem 4.2

We deal with the fractional Lorenz system after substituting Equations (4)–(6) with Equations (19) and (20) of the proposed methods:

$$\sum_{j=1}^{L} a_{ij}^{\alpha,\rho}\, X(t_j) = \lambda\left[\sum_{j=1}^{N} \delta_{ij}\, Y(t_j) - \sum_{j=1}^{N} \delta_{ij}\, X(t_j)\right] \tag{24}$$

$$\begin{aligned}&\sum_{j=1}^{L} a_{ij}^{\alpha,\rho}\, Y(t_j) = (\phi-\lambda)\sum_{j=1}^{N} \delta_{ij}\, X(t_j) - \\ &\sum_{j=1}^{N} \delta_{ij}\, X(t_j)\sum_{j=1}^{N} \delta_{ij}\, Z(t_j) + \phi\sum_{j=1}^{N} \delta_{ij}\, Y(t_j)\end{aligned} \tag{25}$$

$$\sum_{j=1}^{L} a_{ij}^{\alpha,\rho}\, Z(t_j) = \sum_{j=1}^{N} \delta_{ij}\, X(t_j)\sum_{j=1}^{N} \delta_{ij}\, Y(t_j) - \beta\sum_{j=1}^{N} \delta_{ij}\, Z(t_j) \tag{26}$$

Dealing with the initial condition (7) is also done by substituting in the governing Equations (24)–(26). After that, we use Equation (22) to solve this system.

We will now begin to demonstrate the obtained results to explain the stability, reliability, convergence, and performance of FDQM using two types of shape functions with generalized Caputo sense, as follows:

Table 7 explains the impact of grid points (N) on the obtained results X, Y, and Z via PDQM with uniform and non-uniform grid distributions of the fractional Lorenz system with fraction $(\alpha = 1,\ \rho = 1)$. It is remarkable that non-uniform grid results are higher and more consistent with earlier solutions [62,63] and RK4 [64] at N = 11 than uniform ones at N = 13 with error $\leq 10^{-8}$, and execution time of about (0.027 s). Table 8 demonstrates the effect of control values like grid points (N), regularized Shannon factor $(\sigma = h \times \Delta)$, and bandwidth (2K + 1) on the obtained results by DSCDQM–RSK at time (T = 2), fraction $(\alpha = 1,\ \rho = 1)$. We found the best value of the regularized Shannon factor is $\sigma = 0.47 \times \Delta$, with results matching previous studies and PDQM shown in Table 7 at the fewest grid points (N = 9), bandwidth (2K + 1 = 3), and performance time of about (0.022 s). Furthermore, from Tables 7 and 8, it is noted that the maximum number of grids we use is significantly less than in previous studies (N = 1280).

Table 7. Computation of numerical solutions X, Y, Z via uniform and non-uniform PDQM for fractional Chen system (Lorenz system) with time (T = 2) and fraction $(\alpha = 1,\ \rho = 1)$ at various grid points (N). $(x_0 = 2.5, y_0 = 1, z_0 = 0.5)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

N	Uniform PDQM			CPU (s)	Non-uniform PDQM			CPU (s)
	X	Y	Z		X	Y	Z	
4	0.837919933	0.623902547	0.302407988	0.02	0.837919846	0.623902388	0.302407749	0.02
5	0.771720200	0.527485798	0.248194100	0.021	0.771720184	0.527485752	0.248193998	0.021
6	0.761378603	0.49832669	0.2415685802	0.022	0.763378566	0.522326584	0.2475685753	0.022
7	0.760223666	0.497601296	0.2409074397	0.023	0.762223591	0.515601283	0.2469074378	0.023
9	0.762203989	0.500122597	0.242400811	0.025	0.762203974	0.500122578	0.2424007951	0.024
10	0.762216222	0.500169299	0.2424081878	0.026	0.762216161	0.500169278	0.2424081821	0.025
11	0.76221575	0.500167396	0.242407838	0.027	0.76221572	0.500167392	0.242407832	0.025
12	0.76221573	0.500167394	0.2424078434	0.028	0.76221572	0.500167392	0.2424078432	0.026
13	0.76221572	0.500167392	0.2424078432	0.029	0.76221572	0.500167392	0.2424078432	0.027
	Earlier numerical solutions [62,63]							
	X			Y			Z	
1280	0.76221649			0.50016919			0.24240833	
	RK4 [64]							
	0.76221572			0.50016739			0.24240783	

Table 8. Computation of numerical solutions X, Y, Z via DSCDQM–RSK for fractional Chen system (Lorenz system) with time (T = 2) and fraction $(\alpha = 1,\ \rho = 1)$ at various grid points (N), regularized Shannon factor $(\sigma = h \times \Delta)$, and bandwidth (2K + 1). $(x_0 = 2.5, y_0 = 1, z_0 = 0.5)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

		DSCDQM–RSK									
N	**2K + 1**	**σ = 0.4 × Δ**			**σ = 0.45 × Δ**			**σ = 0.47 × Δ**			**CPU (s)**
		X	**Y**	**Z**	**X**	**Y**	**z**	**X**	**Y**	**Z**	
9	3	0.7738	0.5133	0.2549	0.7625	0.5008	0.2430	0.76225	0.5004	0.2425	0.01
	5	0.7735	0.5127	0.2540	0.7623	0.5006	0.2427	0.76223	0.5003	0.2424	0.012
	7	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.014
	9	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.016
	11	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.018
11	3	0.7347	0.5127	0.2540	0.7623	0.5006	0.2427	0.76222	0.50023	0.2425	0.012
	5	0.734	0.5124	0.2537	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.014
	7	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.016
	9	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.018
	11	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.020
13	3	0.7733	0.5122	0.2537	0.76231	0.5006	0.2427	0.762215	0.50026	0.24247	0.014
	5	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.016
	7	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.018
	9	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.020
	11	0.7732	0.5122	0.2536	0.76229	0.5005	0.2426	0.76221	0.5002	0.2424	0.022
	Earlier numerical solutions [62,63]										
	X			Y				Z			
1280	0.76221649			0.50016919				0.24240833			
	RK4 [64]										
	0.76221572			0.50016739				0.24240783			

Figures 3 and 4 present the influence of time and fraction α on the numerical results X, Y, and Z via non-uniform PDQM and DSCDQM–RSK at $(x_0 = 2.5, y_0 = 1, z_0 = 0.5)$ and $(\lambda = 1, \beta = 2, \phi = 1)$. Thus, it is found that the dynamic behaviors of X, Y, and Z differ when the fraction α change. This means that when the value α decreases, the values of X, Y, and Z increase.

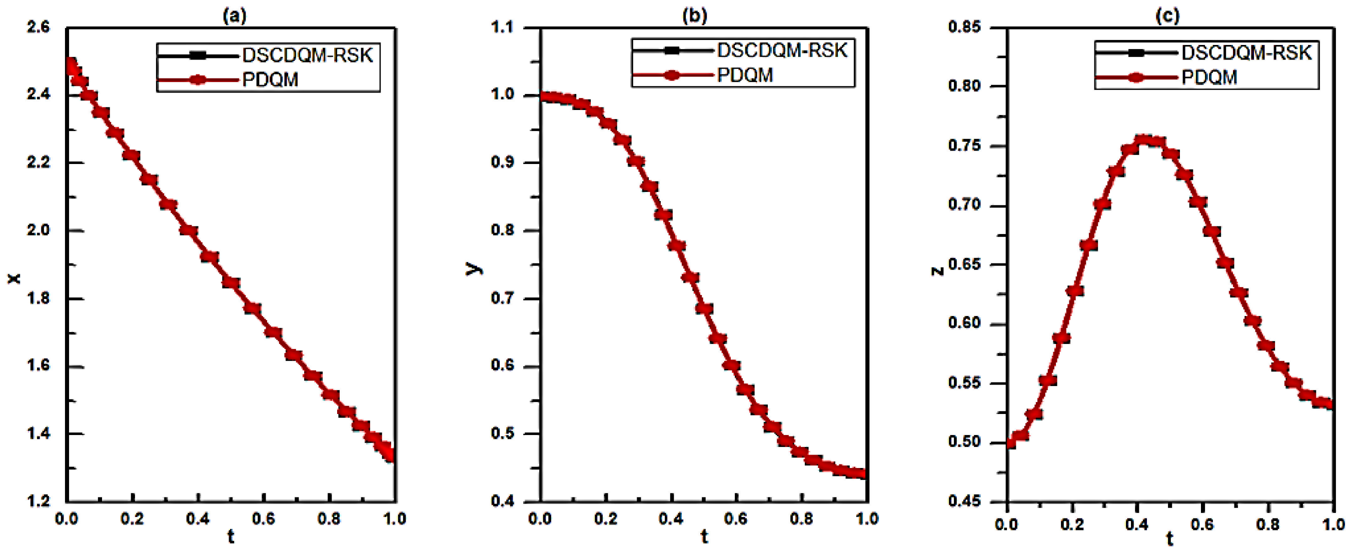

Figure 3. Variance of (**a**) X, (**b**) Y, and (**c**) Z with time (t) via non-uniform PDQM and DSCDQM–RSK for fractional Chen system with time (T = 1), fraction $(\alpha = 1,\ \rho = 1)$, $(x_0 = 2.5, y_0 = 1, z_0 = 0.5)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

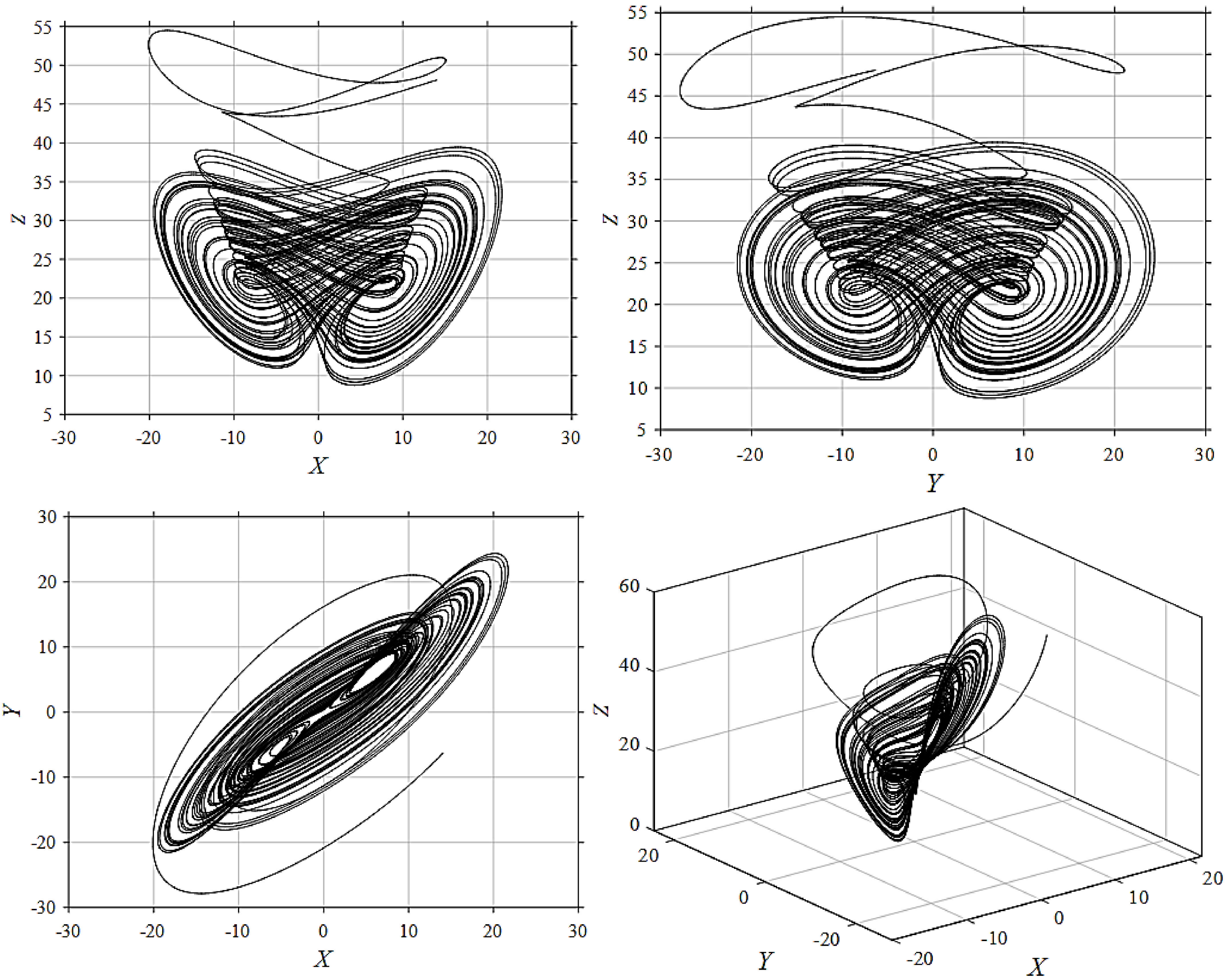

Figure 4. Chaotic attractor of fractional Lorenz system using DSCDQM–RSK with time (T = 100), fraction $(\alpha = 1,\ \rho = 1.2)$, $(x_0 = 2.5, y_0 = 1, z_0 = 0.5)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

Also, Figures 3 and 4 show that the Lorenz system, a classic model for chaotic dynamics, can be extended to fractional-order systems. This extension introduces significant differences in the observed chaotic behavior [74,75]:

Chaotic Regimes:

- The integer-order Lorenz system exhibits a well-defined chaotic regime within a specific parameter range.
- Fractional-order Lorenz systems often exhibit chaotic behavior over a wider range of parameters and fractional orders. This can lead to more complex and diverse chaotic dynamics.

Attractor Structure:

- The integer-order Lorenz system typically has a single strange attractor.
- Fractional-order Lorenz systems can exhibit multiple strange attractors or even the coexistence of different attractors, depending on the fractional order and system parameters.

Fractal Dimension:

- The fractal dimension of the strange attractor in the integer-order Lorenz system is generally between 2 and 3.
- The fractal dimension of the strange attractors in fractional-order Lorenz systems can vary more widely, often exceeding 3. This indicates a more complex and convoluted structure.

Sensitivity to Initial Conditions:

- The integer-order Lorenz system is highly sensitive to initial conditions, leading to the butterfly effect.
- Fractional-order Lorenz systems can exhibit even greater sensitivity to initial conditions, making long-term predictions even more challenging.

Memory Effects:

- The integer-order Lorenz system does not have memory effects.
- The fractional-order Lorenz system incorporates memory effects, which can influence the system's dynamics and make it more resilient to perturbations.

So, fractional-order Lorenz systems can exhibit more complex and diverse chaotic behaviors compared to their integer-order counterparts. The increased sensitivity to initial conditions in fractional-order systems makes long-term predictions even more challenging. Fractional-order Lorenz systems can be used to model real-world phenomena with memory effects or nonlinearities that are not adequately captured by integer-order models.

Figures 4–7 show the fractional Lorenz system's dynamic behaviors as the values of fractions $(\alpha, \ \rho)$ at $\left(x_0 = 2.5, y_0 = 1, z_0 = 0.5\right)$ and $(\lambda = 1, \beta = 2, \phi = 1)$ changes. Also, in these figures, we show projections of the fractional Lorenz system attractors calculated via DSCDQM–RSK when T = 100 for some values of the fractions $(\alpha, \ \rho)$. It is observed that when the fraction ρ increases, the chaotic behavior increases more than the fraction α changes. Consequently, it is noted that the fractional Lorenz system may exhibit chaotic attractors similar to those of its integer-order counterpart when $(\alpha = 0.9, \ \rho = 0.8)$ and $(\alpha = 0.8, \ \rho = 1.2)$. Also, for smaller values of the fractions $(\alpha, \ \rho)$ the system loses its chaotic character.

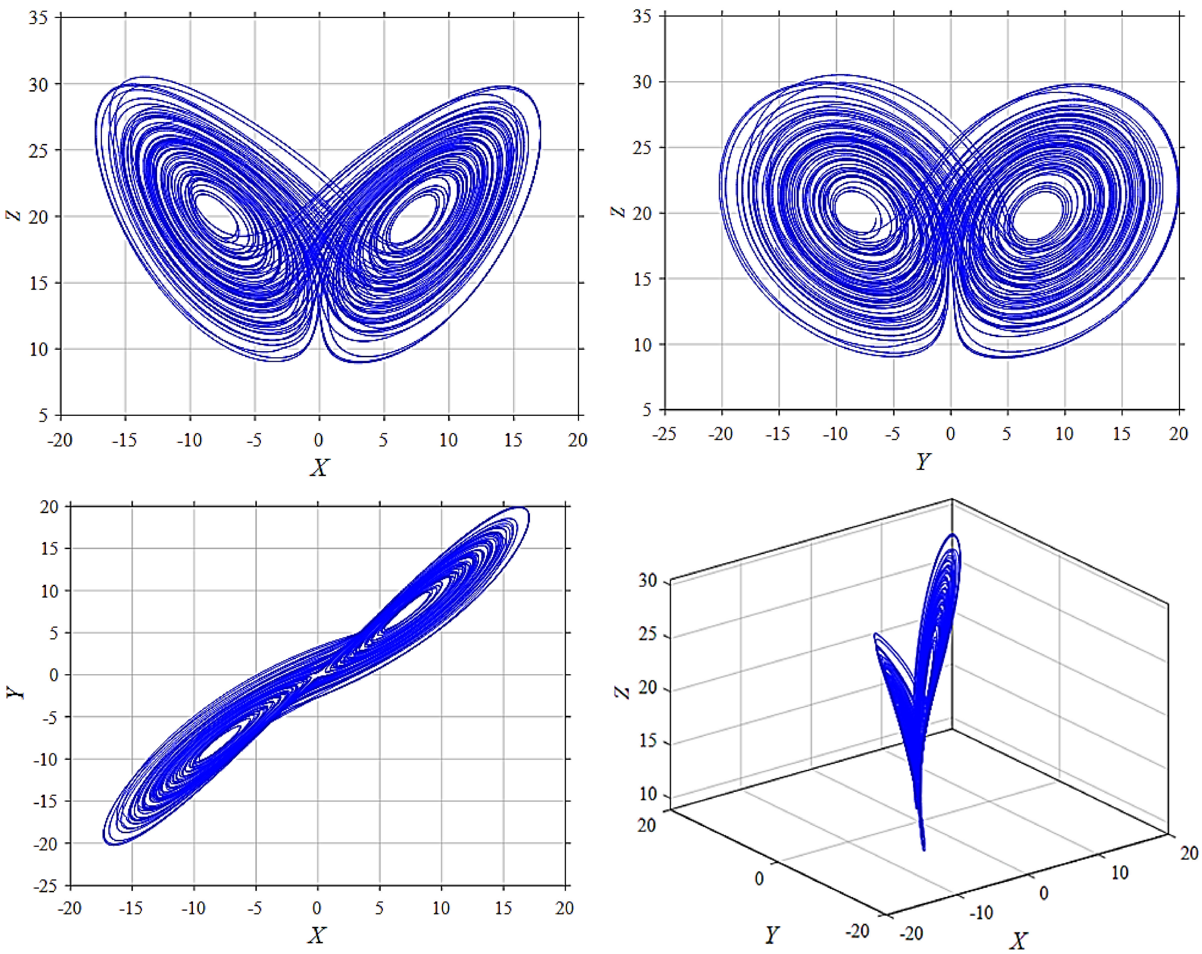

Figure 5. Chaotic attractor of fractional Lorenz system using DSCDQM–RSK with time (T = 100), fraction $(\alpha = 0.8, \ \rho = 1.2)$, $\left(x_0 = 2.5, y_0 = 1, z_0 = 0.5\right)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

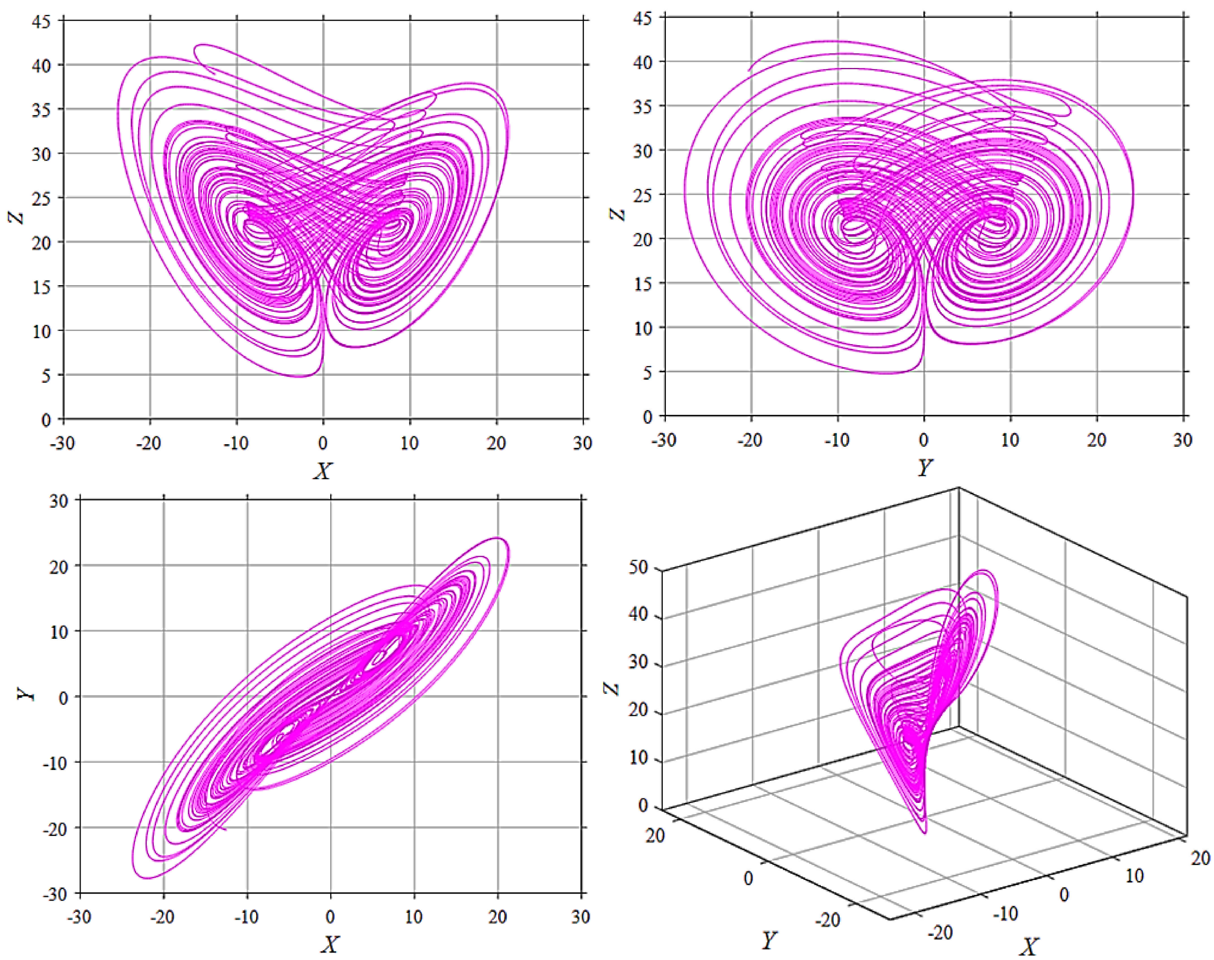

Figure 6. Chaotic attractor of fractional Lorenz system using DSCDQM–RSK with time (T = 100), fraction $(\alpha = 0.97,\ \rho = 1)$, $\left(x_0 = 2.5, y_0 = 1, z_0 = 0.5\right)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

To analyze the computational complexity of the provided code for solving the Riccati equation and the Lorenz system using the Fractional Differential Quadrature Method (FDQM), we will focus on memory space and simulation time complexities in detail. The configuration of the computer used to perform the simulation results is HP Probook 450 G8 Laptop—11th Intel Core i5-1135G7, 8 GB RAM, 512 GB PCIe NVMe SSD, 15.6″ FHD (1920 × 1080), and Intel Iris X Graphics.

1. Memory Space Complexity

Variables:

- Grid Points: The function Chebyshev grid (N) generates N Chebyshev nodes, requiring O(N) space.

The DSCDQM–RSK method demonstrated optimal performance with N = 9 grid points, a bandwidth of 2k + 1 = 7, and a regularized Shannon factor of $\sigma = 0.45 \times \Delta$ for the fractional Riccati equation. This configuration yielded a CPU time of 0.018 s. Similarly, for the fractional Lorenz system, the best results were obtained with N = 9 grid points, a bandwidth of 2k + 1 = 3, and a regularized Shannon factor of $\sigma = 0.47 \times \Delta$, achieving a CPU time of 0.02 s.

In contrast, earlier numerical methods typically required significantly more grid points, often reaching N = 640, 1280, or even 3200.

- Weighting Coefficients: The function PDQM weights (N, t) creates a matrix A of size $N \times N$. Therefore, it requires $O(N^2)$ space.
- Solution Vectors:

- The solution v, X, Y, Z vectorsfor the Riccati equation and Lorenz system are initialized as zero vectors of size $N \times 1$, each requiring O(N) space.
- Total for all solution vectors combined: $4 \times$ O(N) = O(N).

Total Memory Space Complexity:

- The dominant term is O(N^2) from the weighting coefficients matrix. Thus, the total memory space complexity is:

$$O(N^2)$$

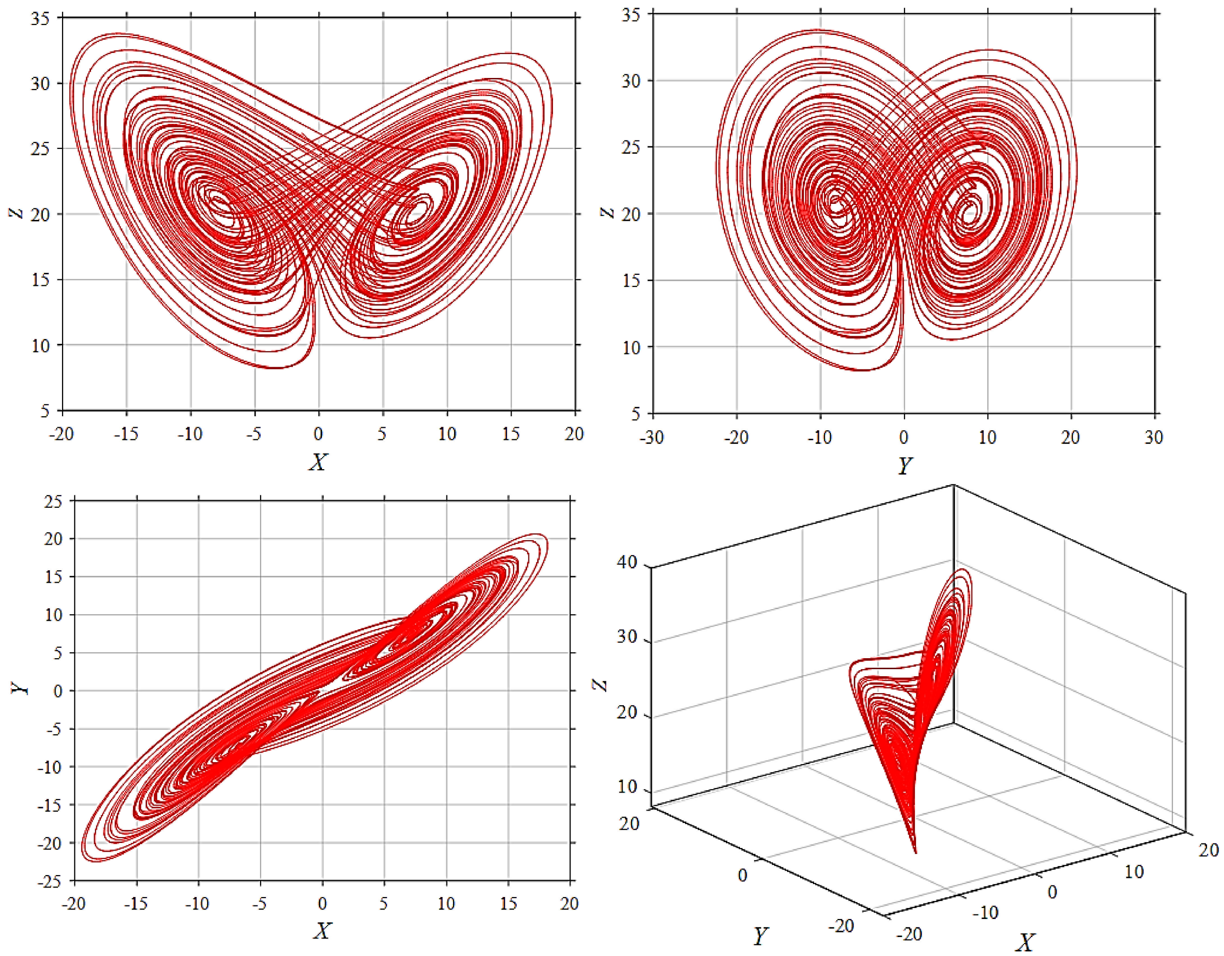

Figure 7. Chaotic attractor of fractional Lorenz system using DSCDQM–RSK with time (T = 100), fraction $(\alpha = 0.9,\ \rho = 0.8)$, $(x_0 = 2.5, y_0 = 1, z_0 = 0.5)$ and $(\lambda = 1, \beta = 2, \phi = 1)$.

2. Simulation Time Complexity
For Solving the Riccati Equation:
Grid Point Generation:

- The grid points are generated in O(N) time.

Weighting Coefficients Calculation:

- The PDQM weights(N, t, 1) function computes the coefficients with a nested loop over N, resulting in O(N^2) time complexity:
- Each entry in matrix A involves calculations that depend on N, leading to O(N^2) complexity for the entire matrix.

Time Integration:

- The time integration loop runs for N-1 iterations, performing a constant time calculation for each iteration:

$$O(N)$$

Total Time Complexity for Riccati Equation:

- Combining these, we get:

$$O(N^2) \text{ (weighting coefficients)} + O(N) \text{ (grid points)} + O(N) \text{ (integration)} = O(N^2)$$

For Solving the Lorenz System:
Grid Point Generation:

- Again, this takes O(N) time.

Weighting Coefficients Calculation:

- The calculations for Ax, Ay, and Az each take $O(N^2)$:

$$3 \times O(N^2) = O(N^2)$$

Time Integration:

- Similar to the Riccati equation, the integration loop runs for N-1 iterations:

$$O(N)$$

Total Time Complexity for Lorenz System:

- Again combining these:

$$O(N^2) \text{ (coefficients)} + O(N) \text{ (grid points)} + O(N) \text{ (integration)} = O(N^2)$$

3. Overall Complexity Summary

- Memory Complexity: $O(N^2)$
- Time Complexity for Riccati Equation: $O(N^2)$
- Time Complexity for Lorenz System: $O(N^2)$

4. Real Numbers Example
For practical evaluation, consider the following:

- For N = 9:

Memory for weighting coefficients: 81 entries.
Memory for solution vectors: 36 entries.
Assuming each entry takes 8 bytes (for double precision), the memory usage would be:

- Weighting coefficients: $81 \times 8 = 648$ bytes (approximately 0.64 KB).
- Solution vectors: $36 \times 8 = 288$ bytes (approximately 0.28 KB).

Total memory usage for N = 9 would be approximately 1 KB.
Execution Time:

- If the operations in the loops take, say, 0.001 s per iteration:

For N = 9, the time for solving the Riccati equation and the Lorenz system would be dominated by the $O(N^2)$ term, leading to an estimated execution time of about 0.1 s (for illustration).

This detailed analysis provides insights into the computational complexity of the code, which is crucial for assessing performance in practical scenarios. The findings underscore the importance of optimizing the weighting coefficients and the iterative solvers for larger values of N.

The choice of time step in numerical methods significantly affects the accuracy of the solutions for differential equations, including those solved using the Differential Quadrature Method (DQM). Here's how:

5. Stability and Convergence:

Our numerical methods are stable because small perturbations in the initial conditions lead to small perturbations in the numerical solution.

Consider the fractional differential equation (FDE) represented in the form:

$$D_c^{\alpha}U(t) = f(t, U(t)), t \in [0, T] \tag{27}$$

where D_c^{α} is the generalized Caputo fractional derivative.

Let $U(t)$ be the exact solution and $U_n(t)$ be the numerical solution obtained through the proposed methods. Introduce a perturbation ϵ_n such that:

$$U_n(t) = U(t) + \epsilon_n \tag{28}$$

By analyzing how the perturbation evolves over time:

$$D_c^{\alpha}(U_n(t) + \epsilon_n) = f(t, U_n(t) + \epsilon_n) \tag{29}$$

This leads to the error equation:

$$D_c^{\alpha}\epsilon_n = f(t, U_n(t) + \epsilon_n) - f(t, U(t)) \tag{30}$$

Assume f satisfies a Lipschitz condition:

$$|f(t, U_1) - f(t, U_2)| \leq L|U_1 - U_2| \tag{31}$$

where L is a constant.

By applying Gronwall's inequality, show that:

$$|\epsilon_n| \leq C \cdot \Delta t^p$$

where C is a constant and p is the order of the method, ensuring that the solution remains bounded as $n \to \infty$.

Our numerical method converges because the numerical solution approaches the exact solution as the grid refinement increases.

Let:

$$\lim_{N\to\infty} \| U_n(t) - U(t) \| = 0 \tag{32}$$

By conducting an error analysis between the numerical solution and the exact

$$E_n = U_n(t) - U(t) \tag{33}$$

By using Taylor expansion around t_n to express $U(t)$:

$$U(t_n + \Delta t) = U(t_n) + \Delta t U'(t_n) + O\left(\Delta t^2\right) \tag{34}$$

Relate this to the discretized version derived from the method:

$$U_n(t_n + \Delta t) = U_n(t_n) + O(\Delta t^p) \tag{35}$$

The method is stable and the achieved convergence rates reached 10^{-8}, indicating a high level of precision in solving the nonlinear fractional initial value problems. The truncation error in our numerical methods is very small due to the Gaussian regularizer $\sigma = 0.45 \times \Delta$ which depends on a small computational domain:

6. Accuracy of the Solution:

- Each step introduces local error, which accumulates over time. Smaller time steps help minimize this accumulation, resulting in a more accurate final solution (error $\leq 10^{-8}$)
- The global error, which is the total error over the entire integration period, also tends to decrease with smaller time steps, leading to better overall accuracy.

4.3. Stability Analysis

After applying our discretization schemes to Equations (1)–(6), we obtained an equivalent set of ordinary differential equations in the time domain:

$$\frac{\mathrm{d}[U]}{\mathrm{dt}} = \mathrm{R}[U] + [\mathrm{K}] \tag{36}$$

where

1. The vector {U} represents the unknown variables at the internal grid points, where υ, X, Y, and Z are the individual components;
    - The initial conditions are stored in the vector [K];
2. R[U] is the right-hand side of Equations (1)∓(6); and
3. $a_{ij}^{(1)}$ is the weighting coefficient matrix of the first derivative:

$$a_{ij}^{(1)} = \begin{bmatrix} a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2(n-1)}^{(1)} \\ a_{32}^{(1)} & a_{33}^{(1)} & & a_{3(n-1)}^{(1)} \\ \vdots & & \ddots & \vdots \\ a_{(n-1)2}^{(1)} & a_{(n-1)3}^{(1)} & \cdots & a_{(n-1)(n-1)}^{(1)} \end{bmatrix}_{(N-2)\times(N-2)} \tag{37}$$

The stability of our technique was evaluated by examining system (35). We employed eigenvalue analysis of the coefficient matrices (a) to determine stability.

For the fractional differential equation represented in the form:

$$D_c^{\alpha} U(t) = f(t, U(t))$$

discretizing this using the FDQM leads to a system of equations expressed as:

$$[a][\mathrm{U}] = \mathrm{R}[U]$$

To analyze stability, we need to compute the eigenvalues λ of the matrix $[a]$. The eigenvalue problem is given by:

$$[a][\mathrm{v}] = \lambda[\mathrm{v}]$$

where v is the eigenvector associated with the eigenvalue λ.

The characteristic polynomial is obtained from:

$$\det(a - \lambda I) = 0$$

where I is the identity matrix.

The stability of the numerical method hinges on the eigenvalues of $[a]$:

1. If all eigenvalues satisfy $|\lambda| < 1$, the method is stable.
2. If any eigenvalue has $|\lambda| \geq 1$, the method may be unstable.

Figure 8 displays the stability regions for a numerical method at various fractional orders (α). Each subplot represents a different α value: 0.5, 0.7, 0.85, and 1. The plots show the eigenvalues of the coefficient matrix (a) in the complex plane (Real vs. Imaginary). At α = 0.5, the eigenvalues are all located on the real axis near −0.5. Since the magnitude of these eigenvalues is less than 1, this indicates stability for this fractional order. At α = 0.7, the eigenvalues form a V-shape, extending into both the positive and negative real axis. A portion of the eigenvalues have magnitudes greater than 1 (outside the unit circle), indicating instability in this region. The region of stability is limited to the portion of the V-shape within the unit circle. At α = 0.85, the eigenvalues form an inverted V-shape. Similar to (b), portions of the eigenvalues are outside the unit circle, indicating instability. The region of stability is again limited to the portion within the unit circle. At α = 1, the eigenvalues lie entirely on the negative real axis, forming a vertical line. All eigenvalues appear to be within the unit circle, suggesting stability for this fractional order (which corresponds to the standard integer-order case). Furthermore, Figure 9 depicts the error propagation in relation to time and fractional order.

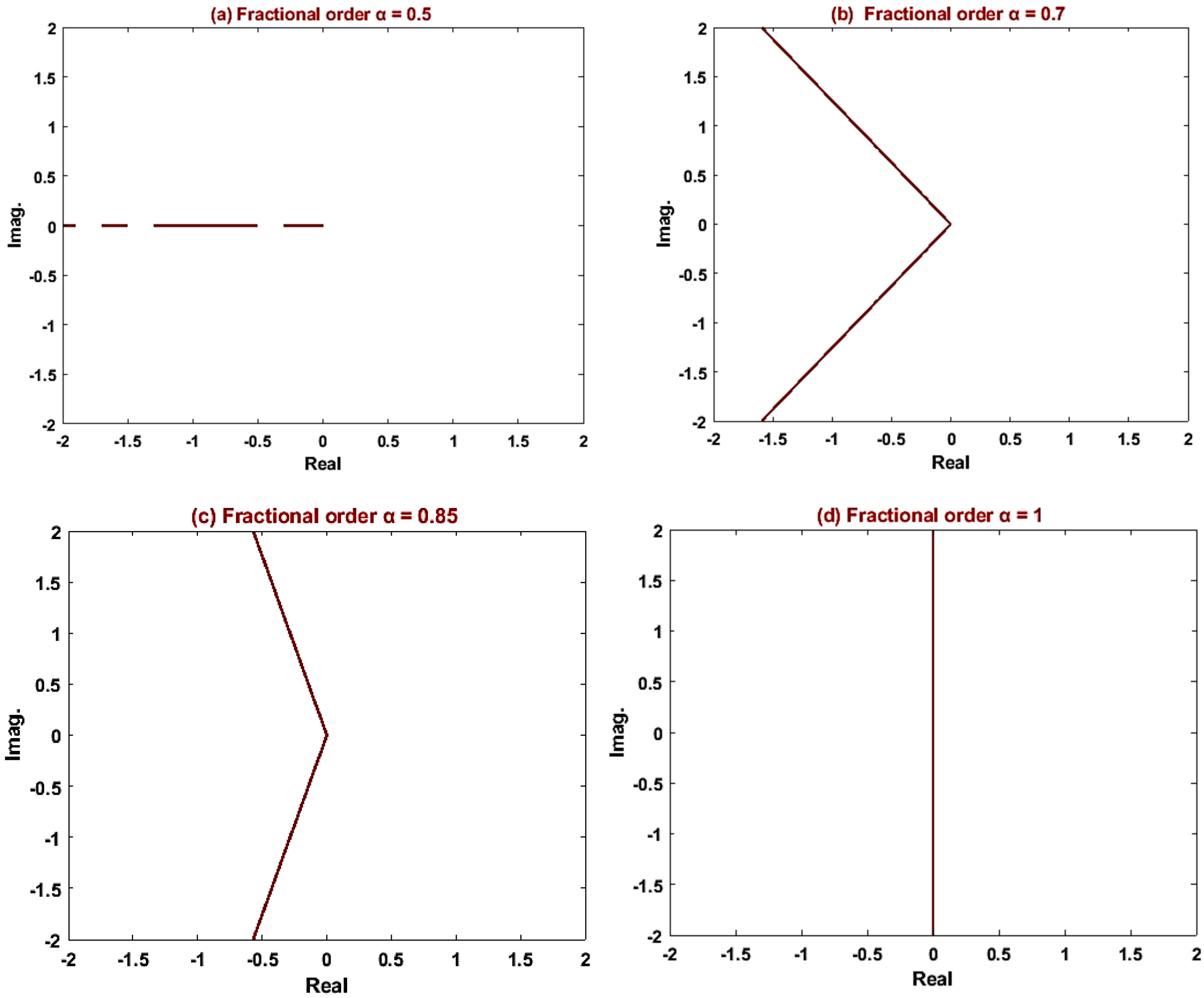

Figure 8. Regions of stability at varying fractional orders.

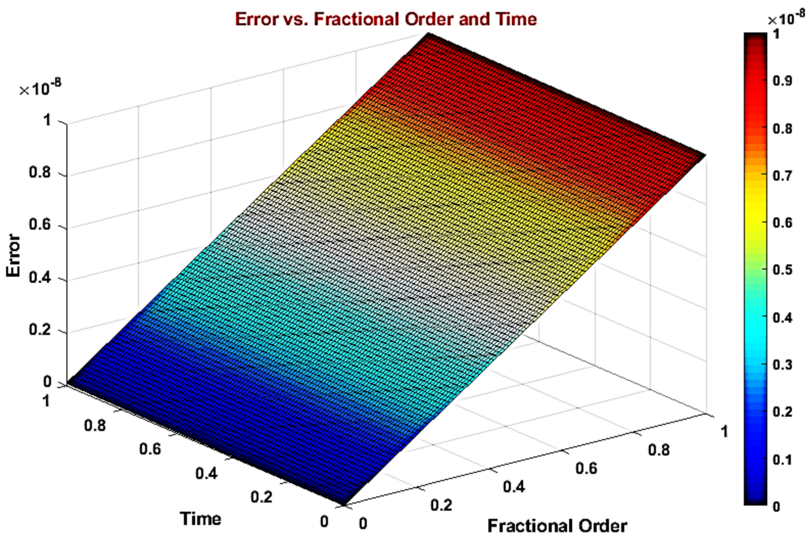

Figure 9. Propagation of errors in relation to time and fractional order.

5. Conclusions

In this present work, we have successfully investigated new numerical methods for solving the fractional Riccati equation and fractional Lorenz system. The novel numerical method is FDQM, which is based on two base functions: Lagrange interpolation polynomial and discrete singular convolution-Regularized Shannon kernel with a new generalized Caputo kind. These methods are used to transform the proposed problems into a nonlinear algebraic system. Then, the iterative method is employed to deal with the problem of nonlinearity. All numerical results were obtained using MATLAB. By comparing our results with those of existing methods, we demonstrated the superior accuracy, efficiency, and overall performance of our proposed techniques. The achieved convergence rates reached 10^{-8}, indicating a high level of precision in solving the nonlinear fractional initial value problems. Error analysis showed that non-uniform grid distributions consistently outperformed uniform distributions, with maximum errors diminishing significantly as grid points increased. Our numerical results demonstrate that the DSC-RSK method achieved significantly faster convergence rates compared to other techniques. The best results of the DSC-RSK method are achieved when grid points are N = 9, bandwidth is 2k + 1 = 7, and the regularized Shannon factor is $\sigma = 0.45 \times \Delta$ at CPU time = 0.018 s for the fractional Riccati equation. But for the fractional Lorenz system, the best results are when grid points are N = 9, bandwidth is 2k + 1 = 3, and the regularized Shannon factor is $\sigma = 0.47 \times \Delta$ at CPU time = 0.02 s. Also, the proposed techniques have been successfully employed to explain the fractional systems' dynamic behaviors. The numerical results demonstrate a strong dependence of the solution on the fractional derivative. The fractional parameters, α and ρ, offer significant advantages in studying the proposed problems with greater accuracy compared to traditional approaches. These techniques hold promise for solving more complex nonlinear equations and other differential applications involving fractional derivatives. The versatility of the FDQM was highlighted through its successful application to both the fractional Riccati equations and the fractional Lorenz system, demonstrating its potential for broader applications in fields requiring the modeling of complex dynamics. The findings suggest that further exploration of fractional orders could uncover even richer dynamics, with potential applications extending to areas such as control theory, physics, and finance.

Author Contributions: Conceptualization, M.M., O.R., M.S. and W.M.A.; methodology, M.M., O.R. and M.S.; software, M.M.; validation, M.M. and W.M.A.; formal analysis, investigation, O.R., M.S. and M.M.; resources, W.M.A.; data curation, writing—original draft preparation, O.R., M.S. and M.M.; writing—review and editing, W.M.A.; visualization, supervision, O.R. and M.S.; Funding, W.M.A. and M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in the article.

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

References

1. Cafagna, D.; Grassi, G. Bifurcation and Chaos in the Fractional-Order Chen System via a Time-Domain Approach. *Int. J. Bifurc. Chaos* **2008**, *18*, 1845–1863. [CrossRef]
2. Petráš, I. The Fractional-Order Lorenz-Type Systems: A Review. *Fract. Calc. Appl. Anal.* **2022**, *25*, 362–377. [CrossRef]
3. Grigorenko, I.; Grigorenko, E. Chaotic Dynamics of the Fractional Lorenz System. *Phys. Rev. Lett.* **2003**, *91*, 034101. [CrossRef]
4. Deng, W.H.; Li, C.P. Chaos Synchronization of the Fractional Lü System. *Phys. A Stat. Mech. Its Appl.* **2005**, *353*, 61–72. [CrossRef]
5. Lü, J.; Chen, G. A New Chaotic Attractor Coined. *Int. J. Bifurc. Chaos* **2002**, *12*, 659–661. [CrossRef]
6. Rahman, Z.-A.S.A.; Jasim, B.H.; Al-Yasir, Y.I.A.; Hu, Y.-F.; Abd-Alhameed, R.A.; Alhasnawi, B.N. A New Fractional-Order Chaotic System with Its Analysis, Synchronization, and Circuit Realization for Secure Communication Applications. *Mathematics* **2021**, *9*, 2593. [CrossRef]
7. Aminikhah, H.; Hemmatnezhad, M. An Efficient Method for Quadratic Riccati Differential Equation. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 835–839. [CrossRef]

8. Khan, N.A.; Ara, A.; Jamil, M. An Efficient Approach for Solving the Riccati Equation with Fractional Orders. *Comput. Math. Appl.* **2011**, *61*, 2683–2689. [CrossRef]
9. Odibat, Z.; Momani, S. Modified Homotopy Perturbation Method: Application to Quadratic Riccati Differential Equation of Fractional Order. *Chaos Solitons Fractals* **2008**, *36*, 167–174. [CrossRef]
10. Cang, J.; Tan, Y.; Xu, H.; Liao, S.-J. Series Solutions of Non-Linear Riccati Differential Equations with Fractional Order. *Chaos Solitons Fractals* **2009**, *40*, 1–9. [CrossRef]
11. Tan, Y.; Abbasbandy, S. Homotopy Analysis Method for Quadratic Riccati Differential Equation. *Commun. Nonlinear Sci. Numer. Simul.* **2008**, *13*, 539–546. [CrossRef]
12. Gülsu, M.; Sezer, M. On the Solution of the Riccati Equation by the Taylor Matrix Method. *Appl. Math. Comput.* **2006**, *176*, 414–421. [CrossRef]
13. Li, Y.; Hu, L. Solving Fractional Riccati Differential Equations Using Haar Wavelet. In Proceedings of the 2010 Third International Conference on Information and Computing, Wuxi, China, 4–6 June 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 1, pp. 314–317.
14. Cui, M. Combined Compact Difference Scheme for the Time Fractional Convection–Diffusion Equation with Variable Coefficients. *Appl. Math. Comput.* **2014**, *246*, 464–473. [CrossRef]
15. Gu, X.-M.; Huang, T.-Z.; Ji, C.-C.; Carpentieri, B.; Alikhanov, A.A. Fast Iterative Method with a Second-Order Implicit Difference Scheme for Time-Space Fractional Convection–Diffusion Equation. *J. Sci. Comput.* **2017**, *72*, 957–985. [CrossRef]
16. Wang, Y.-M. A Compact Finite Difference Method for a Class of Time Fractional Convection-Diffusion-Wave Equations with Variable Coefficients. *Numer. Algorithms* **2015**, *70*, 625–651. [CrossRef]
17. Salman, W.; Gavriilidis, A.; Angeli, P. A Model for Predicting Axial Mixing during Gas–Liquid Taylor Flow in Microchannels at Low Bodenstein Numbers. *Chem. Eng. J.* **2004**, *101*, 391–396. [CrossRef]
18. Xu, Q.; Hesthaven, J.S. Discontinuous Galerkin Method for Fractional Convection-Diffusion Equations. *SIAM J. Numer. Anal.* **2014**, *52*, 405–423. [CrossRef]
19. Aboelenen, T. A Direct Discontinuous Galerkin Method for Fractional Convection-Diffusion and Schrödinger-Type Equations. *Eur. Phys. J. Plus* **2018**, *133*, 316. [CrossRef]
20. Jin, B.; Lazarov, R.; Zhou, Z. A Petrov–Galerkin Finite Element Method for Fractional Convection-Diffusion Equations. *SIAM J. Numer. Anal.* **2016**, *54*, 481–503. [CrossRef]
21. Li, C.; Zeng, F. *Numerical Methods for Fractional Calculus*; CRC Press: Boca Raton, FL, USA, 2015; Volume 24, ISBN 148225381X.
22. Parvizi, M.; Eslahchi, M.R.; Dehghan, M. Numerical Solution of Fractional Advection-Diffusion Equation with a Nonlinear Source Term. *Numer. Algorithms* **2015**, *68*, 601–629. [CrossRef]
23. Saadatmandi, A.; Dehghan, M.; Azizi, M.-R. The Sinc–Legendre Collocation Method for a Class of Fractional Convection–Diffusion Equations with Variable Coefficients. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4125–4136. [CrossRef]
24. Bhrawy, A.H.; Baleanu, D. A Spectral Legendre–Gauss–Lobatto Collocation Method for a Space-Fractional Advection Diffusion Equations with Variable Coefficients. *Rep. Math. Phys.* **2013**, *72*, 219–233. [CrossRef]
25. Tian, W.; Deng, W.; Wu, Y. Polynomial Spectral Collocation Method for Space Fractional Advection–Diffusion Equation. *Numer. Methods Partial Differ. Equ.* **2014**, *30*, 514–535. [CrossRef]
26. Badr, M.; Yazdani, A.; Jafari, H. Stability of a Finite Volume Element Method for the Time-fractional Advection-diffusion Equation. *Numer. Methods Partial Differ. Equ.* **2018**, *34*, 1459–1471. [CrossRef]
27. Gao, F.; Yuan, Y.; Du, N. An Upwind Finite Volume Element Method for Nonlinear Convection Diffusion Problem. *Am. J. Comput. Math.* **2011**, *1*, 264–270. [CrossRef]
28. Bi, Y.; Jiang, Z. The Finite Volume Element Method for the Two-Dimensional Space-Fractional Convection–Diffusion Equation. *Adv. Differ. Equ.* **2021**, *2021*, 379. [CrossRef]
29. Liu, J.; Zhang, J.; Zhang, X. Semi-Discretized Numerical Solution for Time Fractional Convection–Diffusion Equation by RBF-FD. *Appl. Math. Lett.* **2022**, *128*, 107880. [CrossRef]
30. Saadeh, R. Numerical Solutions of Fractional Convection-Diffusion Equation Using Finite-Difference and Finite-Volume Schemes. *J. Math. Comput. Sci.* **2021**, *11*, 7872–7891.
31. Tuan, N.H.; Aghdam, Y.E.; Jafari, H.; Mesgarani, H. A Novel Numerical Manner for Two-dimensional Space Fractional Diffusion Equation Arising in Transport Phenomena. *Numer. Methods Partial Differ. Equ.* **2021**, *37*, 1397–1406. [CrossRef]
32. Devshali, P.; Arora, G. Solution of Two-Dimensional Fractional Diffusion Equation by a Novel Hybrid D (TQ) Method. *Nonlinear Eng.* **2022**, *11*, 135–142. [CrossRef]
33. Khader, M.M. Numerical Treatment for Solving Fractional Riccati Differential Equation. *J. Egypt. Math. Soc.* **2013**, *21*, 32–37. [CrossRef]
34. Li, X.Y.; Wu, B.Y.; Wang, R.T. Reproducing Kernel Method for Fractional Riccati Differential Equations. *Abstr. Appl. Anal.* **2014**, *2014*, 970967. [CrossRef]
35. Sakar, M.G.; Akgül, A.; Baleanu, D. On Solutions of Fractional Riccati Differential Equations. *Adv. Differ. Equ.* **2017**, *2017*, 39. [CrossRef]
36. Agheli, B. Approximate Solution for Solving Fractional Riccati Differential Equations via Trigonometric Basic Functions. *Trans. A. Razmadze Math. Inst.* **2018**, *172*, 299–308. [CrossRef]
37. Liu, X.; Kamran; Yao, Y. Numerical Approximation of Riccati Fractional Differential Equation in the Sense of Caputo-Type Fractional Derivative. *J. Math.* **2020**, *2020*, 1274251. [CrossRef]

38. Hartley, T.T.; Lorenzo, C.F.; Qammer, H.K. Chaos in a Fractional Order Chua's System. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **1995**, *42*, 485–490. [CrossRef]
39. Lu, J.-G. Chaotic Dynamics of the Fractional-Order Ikeda Delay System and Its Synchronization. *Chin. Phys.* **2006**, *15*, 301–305.
40. Li, C.; Peng, G. Chaos in Chen's System with a Fractional Order. *Chaos Solitons Fractals* **2004**, *22*, 443–450. [CrossRef]
41. Diethelm, K.; Ford, N.J.; Freed, A.D. A Predictor-Corrector Approach for the Numerical Solution of Fractional Differential Equations. *Nonlinear Dyn.* **2002**, *29*, 3–22. [CrossRef]
42. Ahmadian, A.; Suleiman, M.; Salahshour, S.; Baleanu, D. A Jacobi Operational Matrix for Solving a Fuzzy Linear Fractional Differential Equation. *Adv. Differ. Equ.* **2013**, *2013*, 104. [CrossRef]
43. Yuan, L.; Yang, Q.; Zeng, C. Chaos Detection and Parameter Identification in Fractional-Order Chaotic Systems with Delay. *Nonlinear Dyn.* **2013**, *73*, 439–448. [CrossRef]
44. Luo, C.; Wang, X. Chaos in the Fractional-Order Complex Lorenz System and Its Synchronization. *Nonlinear Dyn.* **2013**, *71*, 241–257. [CrossRef]
45. Bhalekar, S.; Daftardar-Gejji, V. Fractional Ordered Liu System with Time-Delay. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 2178–2191. [CrossRef]
46. Katugampola, U.N. New Approach to a Generalized Fractional Integral. *Appl. Math. Comput.* **2011**, *218*, 860–865. [CrossRef]
47. Wang, Z.; Huang, X.; Li, N.; Song, X.-N. Image Encryption Based on a Delayed Fractional-Order Chaotic Logistic System. *Chin. Phys. B* **2012**, *21*, 050506. [CrossRef]
48. Wu, G.-C.; Baleanu, D.; Lin, Z.-X. Image Encryption Technique Based on Fractional Chaotic Time Series. *J. Vib. Control* **2016**, *22*, 2092–2099. [CrossRef]
49. Anderson, D.R.; Ulness, D.J. Properties of the Katugampola Fractional Derivative with Potential Application in Quantum Mechanics. *J. Math. Phys.* **2015**, *56*, 063502. [CrossRef]
50. Li, C.; Chen, G. Chaos in the Fractional Order Chen System and Its Control. *Chaos Solitons Fractals* **2004**, *22*, 549–554. [CrossRef]
51. Alomari, A.K. A Novel Solution for Fractional Chaotic Chen System. *J. Nonlinear Sci. Appl.* **2015**, *8*, 478–488. [CrossRef]
52. Erturk, V.S.; Kumar, P. Solution of a COVID-19 Model via New Generalized Caputo-Type Fractional Derivatives. *Chaos Solitons Fractals* **2020**, *139*, 110280. [CrossRef]
53. Xu, Y.; He, Z.; Agrawal, O.P. Numerical and Analytical Solutions of New Generalized Fractional Diffusion Equation. *Comput. Math. Appl.* **2013**, *66*, 2019–2029. [CrossRef]
54. Kumar, P.; Erturk, V.S.; Kumar, A. A New Technique to Solve Generalized Caputo Type Fractional Differential Equations with the Example of Computer Virus Model. *J. Math. Ext.* **2021**, *15*, 1–23.
55. Ragb, O.; Mohamed, M.; Matbuly, M.S. Vibration Analysis of Magneto-Electro-Thermo NanoBeam Resting on Nonlinear Elastic Foundation Using Sinc and Discrete Singular Convolution Differential Quadrature Method. *Mod. Appl. Sci.* **2019**, *13*, 49–79. [CrossRef]
56. Shu, C. *Differential Quadrature and Its Application in Engineering*; Springer Science & Business Media: London, UK, 2012; ISBN 1447104072.
57. Civalek, Ö. Free Vibration of Carbon Nanotubes Reinforced (CNTR) and Functionally Graded Shells and Plates Based on FSDT via Discrete Singular Convolution Method. *Compos. Part B Eng.* **2017**, *111*, 45–59. [CrossRef]
58. Civalek, Ö.; Kiracioglu, O. Free Vibration Analysis of Timoshenko Beams by DSC Method. *Int. J. Numer. Methods Biomed. Eng.* **2010**, *26*, 1890–1898. [CrossRef]
59. Wan, D.C.; Zhou, Y.C.; Wei, G.W. Numerical Solution of Incompressible Flows by Discrete Singular Convolution. *Int. J. Numer. Methods Fluids* **2002**, *38*, 789–810. [CrossRef]
60. Zhang, L.; Xiang, Y.; Wei, G.W. Local Adaptive Differential Quadrature for Free Vibration Analysis of Cylindrical Shells with Various Boundary Conditions. *Int. J. Mech. Sci.* **2006**, *48*, 1126–1138. [CrossRef]
61. Daftardar-Gejji, V.; Bhalekar, S.; Gade, P. Dynamics of Fractional-Ordered Chen System with Delay. *Pramana* **2012**, *79*, 61–69. [CrossRef]
62. Katugampola, U.N. Existence and Uniqueness Results for a Class of Generalized Fractional Differential Equations. *arXiv* **2014**, arXiv:1411.5229.
63. Khan, N.A.; Ara, A.; Alam Khan, N. Fractional-Order Riccati Differential Equation: Analytical Approximation and Numerical Results. *Adv. Differ. Equ.* **2013**, *2013*, 185. [CrossRef]
64. Odibat, Z.; Baleanu, D. Numerical Simulation of Initial Value Problems with Generalized Caputo-Type Fractional Derivatives. *Appl. Numer. Math.* **2020**, *156*, 94–105. [CrossRef]
65. Baleanu, D.; Wu, G.; Zeng, S. Chaos Analysis and Asymptotic Stability of Generalized Caputo Fractional Differential Equations. *Chaos Solitons Fractals* **2017**, *102*, 99–105. [CrossRef]
66. Alemu, S.; Alluvada, P.; Thelkar, A.R.; Samanvita, N.; Lemma, T.; Heyi, K. Fractional Order Sliding Mode Controller Modelling for Fractional Order SEPIC Converter. *Authorea* **2024**. [CrossRef]
67. Podlubny, I. *Fractional Differential Equations*; Academic Press: New York, NY, USA, 1999.
68. Caputo, M. Linear Models of Dissipation Whose Q Is Almost Frequency Independent—II. *Geophys. J. Int.* **1967**, *13*, 529–539. [CrossRef]
69. Ragb, O.; Wazwaz, A.; Mohamed, M.; Matbuly, M.S.; Salah, M. Fractional Differential Quadrature Techniques for Fractional Order Cauchy Reaction-diffusion Equations. *Math. Methods Appl. Sci.* **2023**, *46*, 10216–10233. [CrossRef]

70. Wei, G.W. Discrete Singular Convolution for the Solution of the Fokker–Planck Equation. *J. Chem. Phys.* **1999**, *110*, 8930–8942. [CrossRef]
71. Shao, Z.; Wei, G.W.; Zhao, S. DSC Time-Domain Solution of Maxwell's Equations. *J. Comput. Phys.* **2003**, *189*, 427–453. [CrossRef]
72. Ragb, O.; Mohamed, M.; Matbuly, M.S.; Civalek, O. An Accurate Numerical Approach for Studying Perovskite Solar Cells. *Int. J. Energy Res.* **2021**, *45*, 16456–16477. [CrossRef]
73. Ragb, O.; Mohamed, M.; Matbuly, M.S. Free Vibration of a Piezoelectric Nanobeam Resting on Nonlinear Winkler-Pasternak Foundation by Quadrature Methods. *Heliyon* **2019**, *5*, e01856. [CrossRef]
74. Gu, Y.; Li, G.; Xu, X.; Song, X.; Zhong, H. Solution of a New High-Performance Fractional-Order Lorenz System and Its Dynamics Analysis. *Nonlinear Dyn.* **2023**, *111*, 7469–7493. [CrossRef]
75. Li, H.; Shen, Y.; Han, Y.; Dong, J.; Li, J. Determining Lyapunov Exponents of Fractional-Order Systems: A General Method Based on Memory Principle. *Chaos Solitons Fractals* **2023**, *168*, 113167. [CrossRef]