



Article

Highly Efficient Numerical Algorithm for Nonlinear Space Variable-Order Fractional Reaction–Diffusion Models

Muhammad Yousuf * and Shahzad Sarwar

Department of Mathematics, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; shahzad.sarwar@kfupm.edu.sa or shahzadppn@gmail.com

* Correspondence: myousuf@kfupm.edu.sa

Abstract: In this paper, we present a new highly efficient numerical algorithm for nonlinear variable-order space fractional reaction–diffusion equations. The algorithm is based on a new method developed by using the Gaussian quadrature pole rational approximation. A splitting technique is used to address the issues related to computational efficiency and the stability of the method. Two linear systems need to be solved using the same real-valued discretization matrix. The stability and convergence of the method are discussed analytically and demonstrated through numerical experiments by solving test problems from the literature. The variable-order diffusion effects on the solution profiles are illustrated through graphs. Finally, numerical experiments demonstrate the superiority of the presented method in terms of computational efficiency, accuracy, and reliability.

Keywords: fractional variable-order; fractional reaction–diffusion equations; Riesz derivative; numerical approximation



Citation: Yousuf, M.; Sarwar, S. Highly Efficient Numerical Algorithm for Nonlinear Space Variable-Order Fractional Reaction–Diffusion Models. *Fractal Fract.* **2023**, *7*, 688. <https://doi.org/10.3390/fractalfract7090688>

Academic Editor: Jordan Hristov

Received: 13 August 2023

Revised: 9 September 2023

Accepted: 11 September 2023

Published: 15 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The numerical study of variable-order fractional partial differential equations (VOFDEs) is comparatively new and still in its early stages. It has applications in almost all areas of science and engineering, including elasticity [1], geothermal heat transport [2], biochemistry [3], porous or fractured media [4], the blood alcohol model [5], anomalous diffusion [6–8], viscoelastic mechanics [9], control systems [10], petroleum engineering [11], and many other areas of physics and engineering [12].

Constant-order fractional calculus [13,14] can handle certain extremely significant physical phenomena, but it cannot handle some important classes of physical situations where the order of the fractional derivative is variable. For example, in protein reaction kinetics, it has been found that there are relaxation processes that are adequately represented by a fractional order [15], which depends on the temperature [15]. As a result, the basic physics of reaction kinetics (represented by the order of the relaxation mechanism) change with respect to changes in the temperature. It is logical to suppose that the protein kinetics will be better described by a differential equation with operators that update their order as a function of temperature. To encourage the use of variable-order fractional operators, we can find more examples in the literature. Electroviscous or electrorheological fluids [16] and polymer gels [17] have been shown to change their properties in response to variations in the strength of an applied electric field. The properties of magnetorheological elastomers change in response to the strength of the magnetic field [18]. In the field of damage modeling, nonlinear stress/strain behavior has been shown to change as damage accumulates (over time) in a structure. Variable-order computations may be better suited to describe this phenomenon. These examples suggest that variable-order operators are better suited to represent certain types of physical problems.

The fascinating extension of fractional calculus known as variable-order fractional calculus was proposed by Samko [19] in 1993. The non-stationary power law kernel of variable-order fractional operators allows them to represent the memory and inheritance

properties of a variety of physical phenomena and processes. Variable-order fractional calculus has, therefore, been used as a viable option to provide an efficient mathematical framework for the precise characterization of complicated physical systems and processes.

A challenging task for researchers in the study of fractional differential equations is to find the exact solution of the dynamical systems. In most cases, we do not know the exact solution to the problem and must search for a numerical approximation. The numerical study of variable-order fractional partial differential equations is comparatively new and is still at a young stage. In general, numerical methods are used to approximate solutions of VOFDEs. Due to the complexity of fractional operators, finding the numerical solution of nonlinear problems is always a challenging task for researchers.

Lin et al. [20] presented the equality relation between the Riemann–Liouville fractional derivative and the Grünwald–Letnikov expansion with the variable-order fractional derivative. Exploiting this relationship, they proposed an explicit finite difference approximation for a one-dimensional variable-order fractional nonlinear reaction–diffusion problem. The convergence and stability of this approximation were proved. Zhuang et al. [8] proposed explicit and implicit Euler approximations for a one-dimensional advection–diffusion equation of a variable-order fractional derivative with a nonlinear source component on a finite domain. Sun et al. [21] studied the time variable-order fractional diffusion equation and proposed finite difference schemes. Farnaz et al. [22] applied the compact finite difference operator weighted-shifted Grünwald formula on the time fractional VOFDEs. In [23], the coupled nonlinear Ginzburg–Landau equations are formulated using the Heydari–Hosseini concept’s nonsingular variable-order fractional derivative. The shifted Vieta–Lucas polynomials are used to build a numerical scheme. The authors of [24] derived the numerical scheme based on hp-version spectral collocation methods of the variable-order fractional approach. Based on a natural generalization of the classic Riesz potential, Darve et al. [25] developed a unique definition of the variable-order fractional Laplacian on $\mathbb{R}^n$. To explain the kernels of derivatives and the solution of a relaxation equation, a numerical approach for the numerical inversion of the Laplace transform is used in [26]. In [27], the authors suggested a technique for solving a generic variable-order temporal fractional advection–reaction–diffusion equation with complicated geometries using the meshless approach. For the nonlinear variable-order space fractional reaction–diffusion equations, Li and Wu [28] proposed an iterative reproducing kernel method.

In this paper, we consider the following two-dimensional variable-order initial boundary value problem:

$$\frac{\partial u(x, y, t)}{\partial t} = \mathcal{K}_x \frac{\partial^{\alpha(x, t)} u(x, y, t)}{\partial |x|^{\alpha(x, t)}} + \mathcal{K}_y \frac{\partial^{\beta(y, t)} u(x, y, t)}{\partial |y|^{\beta(y, t)}} + F(u, t), \quad (1)$$

with initial and boundary conditions

$$u(x, y, 0) = \varphi(x, y), \quad (x, y) \in [a, b] \times [c, d], \quad (2)$$

$$u(x, y, t) = 0, \quad (x, y) \in \partial\Omega, \quad 0 < t, \quad (3)$$

where $1 < \alpha(x, t), \beta(y, t) < 2$, $\Omega = (a, b) \times (c, d)$, $\mathcal{K}_x$ and $\mathcal{K}_y$ are diffusion coefficients. For $\alpha(x, t) = \beta(y, t) = 2$, Equation (1) represents the two-dimensional classical reaction–diffusion model. The source term $F(u, t)$ is assumed to be continuous and satisfies the Lipschitz condition with $L > 0$; that is:

$$\|F(u) - F(\tilde{u})\| \leq L \|u - \tilde{u}\|.$$

The space fractional derivative in Equation (1) is considered a Riesz fractional derivative of the variable order [12,29], defined as

$$\frac{\partial^{\alpha(x, t)} u(x, y, t)}{\partial |x|^{\alpha(x, t)}} = \frac{-1}{2 \cos(\frac{\pi \alpha(x, t)}{2})} \left[{}_{RL}D_{a, x}^{\alpha(x, t)} + {}_{RL}D_{x, b}^{\alpha(x, t)} \right] u(x, y, t), \quad (4)$$

$$\frac{\partial^{\beta(y,t)} u(x, y, t)}{\partial |y|^{\beta(y,t)}} = \frac{-1}{2 \cos(\frac{\pi \beta(y,t)}{2})} \left[{}_{RL}D_{c,y}^{\beta(y,t)} + {}_{RL}D_{y,d}^{\beta(y,t)} \right] u(x, y, t), \quad (5)$$

where ${}_{RL}D_{a,x}^{\alpha(x,t)}$, ${}_{RL}D_{c,y}^{\beta(y,t)}$ are the left variable-order Riemann–Liouville fractional derivatives, defined as follows:

$${}_{RL}D_{a,x}^{\alpha(x,t)} u(x, y, t) = \frac{1}{\Gamma(2 - \alpha(x, t))} \frac{\partial^2}{\partial x^2} \int_a^x (x - \tau)^{1-\alpha(x,t)} u(\tau, y, t) d\tau, \quad (6)$$

$${}_{RL}D_{c,y}^{\beta(y,t)} u(x, y, t) = \frac{1}{\Gamma(2 - \beta(y, t))} \frac{\partial^2}{\partial y^2} \int_c^y (y - \xi)^{1-\beta(y,t)} u(x, \xi, t) d\xi. \quad (7)$$

Similarly, the right variable-order Riemann–Liouville fractional derivatives ${}_{RL}D_{x,b}^{\alpha(x,t)}$, ${}_{RL}D_{y,d}^{\beta(y,t)}$ are defined as

$${}_{RL}D_{x,b}^{\alpha(x,t)} u(x, y, t) = \frac{1}{\Gamma(2 - \alpha(x, t))} \frac{\partial^2}{\partial x^2} \int_x^b (x - \tau)^{1-\alpha(x,t)} u(\tau, y, t) d\tau, \quad (8)$$

$${}_{RL}D_{y,d}^{\beta(y,t)} u(x, y, t) = \frac{1}{\Gamma(2 - \beta(y, t))} \frac{\partial^2}{\partial y^2} \int_y^d (y - \xi)^{1-\beta(y,t)} u(x, \xi, t) d\xi, \quad (9)$$

respectively.

The variable-order fractional model (1) allows different diffusion rates of the fractional order with respect to each spatial variable as well as time. To explain the generalized diffusion fluxes that do not satisfy Fick's law, Fourier's law, or Newton's constitutive equation, the Riesz fractional derivative is introduced. Since they connect the local state of the field with the state of the whole field, these nonlocal derivatives can improve the predictability of several models. For more on fractional Riesz derivatives and fractional Laplacian, see [19,29].

To our knowledge, there is no work on variable-order space fractional problems by a third-order time-stepping method. However, for the constant-order fractional differential equations, several approaches have been developed to solve space-fractional reaction–diffusion equations. Khaliq et al. [30] presented a fourth-order technique for the space-fractional nonlinear Schrödinger equation. The authors of [31] presented a numerical approach for solving variable-order fractional differential equations utilizing fractional-order generalized Chelyshkov wavelets and the beta function. Ding and his collaborators [32,33] developed a numerical technique by discretizing the Riesz derivative; they produced interesting findings by adopting the Runge–Kutta (ETDRK) approach, the Padé approximation, or by developing new helpful generating functions in a series of articles. Furati et al. [34] developed two schemes for one-dimensional space fractional reaction–diffusion equations. Yousuf et al. [35] provided an L -stable method and an A -stable method for solving two-dimensional nonlinear fractional reaction–diffusion problems. The authors of this study [36] suggested finite difference techniques for solving Riesz-space variable-order fractional diffusion equations. The time derivatives in linear and nonlinear situations are then discretized using the Crank–Nicolson and linearly implicit difference schemes, respectively.

Almost all previously developed time-stepping methods used Padé rational approximations, which required numerical computations with complex numbers that made the methods slow and computationally expensive. Motivated by the above-mentioned work, the natural question is: Is there any way to develop a time-stepping method for variable-order fractional problems without using Padé rational approximations? The answer to this question is efficiently presented in this work. In this paper, we develop an unconditionally stable, highly efficient, third-order convergent numerical method for nonlinear variable-order fractional reaction–diffusion problems. Instead of using Padé approximations, a

rational approximation with a single Gaussian quadrature pole, called the Calahan rational approximation [37], is used to avoid complex arithmetic. To achieve high accuracy and computational efficiency, one must solve two linear systems of the form $(kA + cI)X = Y$ using the Gaussian quadrature point $c = \frac{3+\sqrt{3}}{6}$ and the same discretization matrix A . For example, the rational approximation Padé-(2, 2) has two complex conjugate poles and residues and Padé-(0, 4) has two pairs of complex conjugate poles and residues. To achieve the desired results, multiple systems with differing coefficient matrices must be solved. This process can also necessitate complex arithmetic, potentially making the schemes more computationally expensive. The computational efficiency of the new method is shown by recording the central processing unit (CPU) time for each iteration in the convergence tables.

The rest of this paper is organized as follows: Section 2 is devoted to the discretization of the Riesz derivative. In Section 3, a new third-order numerical method for nonlinear space variable-order reaction–diffusion problems is developed and an algorithm is provided to easily implement the method. The stability of the method and a linear error analysis are given in Section 4. In Section 5, numerical experiments are performed by solving nonlinear reaction–diffusion problems with the variable-order fractional Riesz derivative. The convergence results and CPU time are also presented in the same section. Finally, our conclusion is drawn in the last section.

2. Spatial Discretization

In this section, we provide the space derivative approximation of Equation (1). Suppose that $u^{(m-1)} \in C[a, b]$ and $u^{(m)}$ are integrable in $[a, b]$. Consider the spatial discretization: $x_m = a + mh_x$, $m = 0, 1, \dots, M$ with $h_x = (b - a)/M$ and $y_n = c + nh_y$, $n = 0, 1, \dots, N$ with $h_y = (d - c)/N$. Then for every $\alpha(x, t) \in (1, 2)$, the variable-order Riemann–Liouville fractional derivative exists and coincides with the Grünwald–Letnikov derivative. The relationship between the Riemann–Liouville and Grünwald–Letnikov definitions also plays an essential role in the numerical approximations of fractional differential equations, the manipulation of fractional derivatives, and the formulation of physically meaningful initial-value and boundary-value problems for fractional differential equations. As a result, the Riemann–Liouville definition can be used to formulate the problems, and the Grünwald–Letnikov definition can then be applied to arrive at a numerical solution. The left and right variable-order Grünwald–Letnikov fractional derivatives [29] are defined as follows:

$${}_G L D_{a,x}^{\alpha(x,t)} u(x, y, t) = \lim_{h_x \rightarrow 0^+} h_x^{-\alpha(x,t)} \sum_{k=0}^{\frac{x-a}{h_x}} (-1)^k \frac{\Gamma(\alpha(x,t) + 1)}{k! \Gamma(\alpha(x,t) - k + 1)} u(x - kh_x, y, t),$$

and

$${}_G L D_{x,b}^{\alpha(x,t)} u(x, y, t) = \lim_{h_x \rightarrow 0^+} h_x^{-\alpha(x,t)} \sum_{k=0}^{\frac{b-x}{h_x}} (-1)^k \frac{\Gamma(\alpha(x,t) + 1)}{k! \Gamma(\alpha(x,t) - k + 1)} u(x + kh_x, y, t),$$

respectively. In [38], the symmetrical fractional central difference operator $\Delta_{h_x}^\alpha$ for the constant fractional order α is introduced by Ortigueira. By changing the constant-order α with variable-order $\alpha(x, t)$ we have

$$\Delta_{h_x}^{\alpha(x,t)} u(x, y, t) = \sum_{k=-\infty}^{\infty} \frac{(-1)^k \Gamma(\alpha(x,t) + 1)}{\Gamma(\frac{\alpha(x,t)}{2} - k + 1) \Gamma(\frac{\alpha(x,t)}{2} + k + 1)} u(x - kh_x, y, t). \quad (10)$$

Lemma 1. Let $u \in C^5(\mathbb{R})$ and all derivatives of u up to five belong to $L_1(\mathbb{R})$, and by using the above-mentioned variable-order fractional central difference operator $\Delta_{h_x}^{\alpha(x,t)}$, the variable-order Riesz derivative can be discretized as

$$\frac{\partial^{\alpha(x,t)}}{\partial |x|^{\alpha(x,t)}} u(x_j) = \frac{-1}{h_x^{\alpha(x,t)}} \Delta_{h_x}^{\alpha(x,t)} u(x_j) + \mathcal{O}(h_x^2).$$

Proof. The proof of this Lemma can be drawn by using a method similar to [39]. Therefore, we have chosen to omit the detailed proof here, inviting interested readers to explore it on their own. $\square$

Lemma 2. Let $p_j^{\alpha(x,t)} = \frac{(-1)^j \Gamma(\alpha(x,t)+1)}{\Gamma(\frac{\alpha(x,t)}{2}-j+1)\Gamma(\frac{\alpha(x,t)}{2}+j+1)}$ be the coefficient of the variable finite-centered difference approximation in Equation (10) for $j = 0, \pm 1, \pm 2, \dots$, and $\alpha(x, t) > -1$. Then we have

- $p_0^{\alpha(x,t)} = \frac{\Gamma(\alpha(x,t)+1)}{\Gamma^2(\frac{\alpha(x,t)}{2}+1)}$, $p_{j+1}^{\alpha(x,t)} = \left(1 - \frac{\alpha(x,t)+1}{\frac{\alpha(x,t)}{2}+j+1}\right) p_j^{\alpha(x,t)}$, $j = 0, \pm 1, \pm 2, \dots$,
- $p_{-j}^{\alpha(x,t)} = p_j^{\alpha(x,t)}$, $p_0^{\alpha(x,t)} > 0$, $p_j^{\alpha(x,t)} \leq 0$ for $j = \pm 1, \pm 2, \dots$,
- $p_j^{\alpha(x,t)} = \mathcal{O}(j^{-1-\alpha(x,t)})$.

Proof. This lemma can easily be proved by using the arguments in [39], (Lemma 2.1). Therefore, we have chosen to omit the detailed proof here, inviting interested readers to explore it on their own. $\square$

When $u(x) = 0$ for $x \notin (a, b)$, then $u_j = 0$, for $j \neq 1, \dots, M-1$. At the interior mesh x_m , $m = 1, \dots, M-1$, we have

$$\begin{aligned} \Delta_{h_x}^{\alpha(x,t)} u(x_m) &= \sum_{j=m+1-M}^{m-1} p_j^{\alpha(x,t)} u(x_{m-j}) \\ &= \left[p_{m-1}^{\alpha(x,t)} u_1 + p_0^{\alpha(x,t)} u_m + \dots + p_{M-m-1}^{\alpha(x,t)} u_{M-1} \right]. \end{aligned} \quad (11)$$

The system of Equation (11) can be written in the following form:

$$\Delta_{h_x}^{\alpha(x,t)} U = P^{\alpha(x,t)} U$$

where

$$P^{\alpha(x,t)} = \begin{bmatrix} p_0^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_2^{\alpha(x,t)} & \dots & p_{M-4}^{\alpha(x,t)} & p_{M-3}^{\alpha(x,t)} & p_{M-2}^{\alpha(x,t)} \\ p_1^{\alpha(x,t)} & p_0^{\alpha(x,t)} & p_1^{\alpha(x,t)} & \dots & p_{M-5}^{\alpha(x,t)} & p_{M-4}^{\alpha(x,t)} & p_{M-3}^{\alpha(x,t)} \\ p_2^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_0^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_2^{\alpha(x,t)} & \dots & p_{M-4}^{\alpha(x,t)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ p_{M-4}^{\alpha(x,t)} & \dots & p_2^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_0^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_2^{\alpha(x,t)} \\ p_{M-3}^{\alpha(x,t)} & p_{M-4}^{\alpha(x,t)} & \dots & p_2^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_0^{\alpha(x,t)} & p_1^{\alpha(x,t)} \\ p_{M-2}^{\alpha(x,t)} & p_{M-3}^{\alpha(x,t)} & p_{M-4}^{\alpha(x,t)} & \dots & p_2^{\alpha(x,t)} & p_1^{\alpha(x,t)} & p_0^{\alpha(x,t)} \end{bmatrix}, \quad U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{M-3} \\ u_{M-2} \\ u_{M-1} \end{bmatrix}.$$

The term $\frac{\partial^{\beta(y,t)}}{\partial |y|^{\beta(y,t)}} u(x, y, t)$ is similarly discretized. The semi-discrete system for Equation (1) is

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} + F(\mathbf{u}(t), t) \quad (12)$$

where $A = K_x \Delta_{h_x}^{\alpha(x,t)} \otimes I_x + I_y \otimes K_y \Delta_{h_y}^{\beta(y,t)}$ is a square matrix of size $(M-1)(N-1) \times (M-1)(N-1)$, with I_x and I_y being identity matrices of sizes $(M-1) \times (M-1)$ and $(N-1) \times (N-1)$, respectively. Vector $\mathbf{u}$ is of the size $(M-1)(N-1) \times 1$ and consists of

columns of the matrix $[u_{i,j}]$, where each $u_{i,j} = u(x, y_j)$ for $1 \leq i \leq M-1$, $1 \leq j \leq N-1$. The nonlinear term $F(\mathbf{u}) = [F_1, \dots, F_{M-1}]^T$ is a vector of size $(M-1)(N-1) \times 1$ with each $F_j = [F(u_{1,j}), F(u_{2,j}), \dots, F(u_{M-1,j})]^T$, $j = 1, 2, \dots, N-1$.

3. The Reaction–Diffusion System and Time-Stepping Method

Consider the following standard semi-linear initial value problem:

$$\begin{aligned} \frac{\partial u}{\partial t} + Au &= F(u, t) \quad \text{in } \Omega, \quad t \in (0, T], \\ u(x, y, 0) &= u_0 \quad \text{in } \Omega, \end{aligned} \quad (13)$$

where A represents a uniformly elliptic operator, F is taken as a smooth nonlinear function defined on $\mathbf{R}^d$, and Ω is a bounded domain in $\mathbf{R}^d$.

$$Au := - \sum_{j,k=1}^d \frac{\partial}{\partial x_j} \left(a_{j,k} \frac{\partial u}{\partial x_k} \right) + \sum_{j=1}^d b_j \frac{\partial u}{\partial x_j} + b_0 u. \quad (14)$$

Coefficients $a_{j,k}$ and b_j are sufficiently smooth C^∞ functions defined over $\bar{\Omega}$; they possess the properties $a_{j,k} = a_{k,j}$, and $b_0 \geq 0$, and there exists some $c_0 > 0$,

$$\sum_{j,k=1}^d a_{j,k} \xi_j \xi_k \geq c_0 |\xi|^2, \quad \text{on } \bar{\Omega}, \quad \text{for all } \xi \in \mathbf{R}^d. \quad (15)$$

We examine the initial value problem (13) in a Hilbert space $\mathcal{H}$, where the linear, self-adjoint, and positive-definite closed operator A with a compact inverse is defined on a dense domain $D(A) \subset \mathcal{H}$; see [37]. Operator A could represent any of $\{A_{\bar{h}}\}_{0 < \bar{h} \leq \bar{h}_0'}$ resulting from a spatial discretization, and $\mathcal{H}$ could be an appropriate finite-dimensional subspace of $L_2(\Omega)$, cf. [37,40].

Assume that the resolvent set $\rho(A)$ satisfies

$$\rho(A) \supset \bar{\Sigma}_\alpha, \quad \Sigma_\alpha := \{c \in \mathbb{C} : \alpha < |\arg(c)| \leq \pi, c \neq 0\}.$$

for some $\alpha \in (0, \pi/2)$. We also assume that there exists $M \geq 1$, such that

$$\|(cI - A)^{-1}\| \leq M|c|^{-1}, \quad c \in \Sigma_\alpha. \quad (16)$$

It follows that $-A$ is the infinitesimal generator of an analytic semigroup $\{e^{-tA}\}_{t \geq 0}$, which is the solution operator for (13) with $F \equiv 0$. There is a standard representation,

$$e^{-tA} = \frac{1}{2\pi i} \int_\Lambda e^{-tc} (cI - A)^{-1} dc, \quad (17)$$

where $\Lambda := \{c \in \mathbb{C} : |\arg(c)| = \theta\}$, oriented so that $\text{Im}(c)$ decreases, for any $\theta \in (\alpha, \frac{\pi}{2})$. Using the integrating factor approach, we write the exact solution of (13) as a Volterra integral equation,

$$u(t) = e^{-tA} u_0 + \int_0^t e^{-(t-s)A} F(u(s), s) ds. \quad (18)$$

Let $0 < k \leq k_0$, for some k_0 , be the fixed time step and $t_n = nk$. We replace t with $t+k$ and apply the linear transformation $s-t = k\tau$ to transform (18) into the following nonlinear Fredholm integral equation,

$$u(t+k) = e^{-kA} u(t) + k \int_0^1 e^{-k(1-\tau)A} F(u(t+k\tau), t+k\tau) d\tau, \quad (19)$$

and set the recurrence formula,

$$u(t_{n+1}) = e^{-kA}u(t_n) + k \int_0^1 e^{-k(1-\tau)A} F(u(t_n + \tau k), t_n + \tau k) d\tau. \quad (20)$$

3.1. Time-Stepping Method

It is quite challenging to implement the recurrence Equation (20) because it involves a nonlinear function of the unknown function $u(t)$. Another computational difficulty is to efficiently compute the matrix exponential functions e^{-kA} and $e^{-k(1-\tau)A}$. Various time-stepping schemes are available in the literature, where matrix exponential functions are computed using MATLAB command $\expm(A)$. Through our numerical experiments, we find that computing the matrix exponential functions using the MATLAB command is expansive (time-consuming), especially when matrix A is non-diagonal.

These difficulties are resolved by approximating the nonlinear function $F(u, t)$ by some constant functions and approximating the matrix exponential functions using a third-order single Gaussian quadrature pole rational approximation. The method developed by using such a rational approximation does not require complex arithmetic, which definitely enhance the computational efficiency.

Development of the Time-Stepping Method

The time-stepping method is constructed using the following steps:

Step1: Assume $F(u, t)$ to be constant F_n over each sub-interval $[t_n, t_{n+1}]$; that is, $F(u(t), t) \approx F_n = F(u(t_n), t_n)$ for $t \in [t_n, t_{n+1}]$. Then the exact computation of Equation (20) results in

$$a_n \approx P_0\left(\frac{kA}{2}\right)u_n + \frac{k}{2}P_1\left(\frac{kA}{2}\right)F_n \quad (21)$$

where $P_0(kA) = e^{-kA}$, $P_1(kA) = (-kA)^{-1}(e^{-kA} - I)$.

Step2: Now, using a_n , we approximate $F(u(t), t) \approx 2F_n^a - F_n$, for $t \in [t_n, t_{n+1}]$ and evaluate the formula (20) to obtain

$$b_n \approx P_0(kA)u_n + kP_1(kA)(2F_n^a - F_n) \quad (22)$$

where $F_n^a = F(a_n, t_n + k/2)$.

Step3: Finally, using a_n and b_n , we approximate the nonlinear function as

$$F(u(t), t) \approx F_n + (t - t_n)(-3F_n + 4F_n^a - F_n^b) + \frac{(t - t_n)^2}{2}(F_n - 2F_n^a + F_n^b),$$

for $t \in [t_n, t_{n+1}]$, we evaluate the integral, rearrange the terms, and obtain the following formula to compute u_{n+1}

$$\begin{aligned} u_{n+1} = & P_0(kA)u_n + kP_1(kA)F_n + kP_2(kA)(-3F_n + 4F_n^a - F_n^b) \\ & + 4kP_3(kA)(F_n - 2F_n^a + F_n^b), \end{aligned} \quad (23)$$

where

$$\begin{aligned} F_n^b &= F(b_n, t_n + k), \\ P_2(kA) &= (-kA)^{-2}(e^{-kA} - I + kA), \\ P_3(kA) &= (-kA)^{-3}\left(e^{-kA} - I + kA - \frac{k^2A^2}{2}\right). \end{aligned}$$

This is a Runge–Kutta type time-stepping strategy and can easily be implemented when A is a scalar or a diagonal matrix without a zero entry on the diagonal. However, when A is a non-diagonal matrix, there can be computational difficulties in computing

$(-A)^{-1}$ and $(-A)^{-3}$ if eigenvalues of A are close to zero. Another issue is to efficiently compute the matrix exponential functions, e^{-kA} and $e^{-kA/2}$, when A is non-diagonal. These issues are resolved by approximating the matrix exponential functions using Calahan's single real pole rational approximation [37], as

$$R(z) = 1 - \frac{z}{1 + \frac{3+\sqrt{3}}{6}z} - \frac{\sqrt{3}}{6} \frac{z^2}{(1 + \frac{3+\sqrt{3}}{6}z)^2} = \frac{6(6 + 2\sqrt{3}z - (1 + \sqrt{3})z^2)}{(6 + (3 + \sqrt{3})z)^2}. \quad (24)$$

Replacing e^{-kA} by $R(kA)$ and $e^{-kA/2}$ by $R(kA/2)$, in Equations (21)–(23), we find that factors A^{-1} and A^{-3} are also canceled. The third-order accuracy of the approximation (24) is shown as follows

$$\begin{aligned} e^{-z} &= 1 - z + \frac{1}{2}z^2 - \frac{1}{6}z^3 + \frac{1}{24}z^4 + O(z^4), \\ R(z) &= 1 - z + \frac{1}{2}z^2 - \frac{1}{6}z^3 - \frac{\sqrt{3}}{36}z^4 + O(z^4), \end{aligned}$$

$$\|e^{-z} - R(z)\| = O(z^4).$$

L -acceptability for the method is evident from the fact that [41],

$$E(y) \equiv |D(iy)|^2 - |N(iy)|^2 = (108 + 72\sqrt{3})y^4 \geq 0, y \in \mathcal{R}.$$

A very small error constant; that is, the coefficient of the leading term in the error, 0.00644585578, shows good accuracy of the approximation. The rational approximation $R(z)$ also satisfies the condition that $|R(z)| < 1$ for all $z > 0$ and $R(z)$ is a decreasing function on $(0, \infty)$ with $|R(\infty)| < 1$. In fact, $R(\infty) = 1 - \sqrt{3} > -1$. One crucial advantage of this approximation is that its denominator is the square of a linear function with one real pole of multiplicity 2. We can easily implement the method by solving two systems of the form $(kA + cI)X = Y$, where $c = \frac{3+\sqrt{3}}{6}$. In the finite-dimensional case, when A is a positive-definite matrix, this means that the two systems have the same real values of the positive-definite matrix [37]. This is in contrast to the schemes developed by using Padé approximations. For example, Padé-(2, 2) has two complex conjugate poles and Padé-(0, 4) has two pairs of complex conjugate poles. Such schemes require complex arithmetic, which can make the scheme computationally expensive.

3.2. Modified Time-Stepping Method

We use the third-order Calahan approximation (24) of e^{-z} to modify the method (21)–(23), as:

$$u_{n+1} = R(z)u_n + \Phi_1(z)F_n + \Phi_2(z)F_n^a + \Phi_3(kA)F_n^b, \quad (25)$$

where

$$\begin{aligned} \Phi_1(z) &= \frac{6k(1 + (1 + \sqrt{3})z)}{(6 + (3 + \sqrt{3})z)^2}, \\ \Phi_2(z) &= \frac{24k}{(6 + (3 + \sqrt{3})z)^2}, \\ \Phi_3(z) &= \frac{6k(1 + (2 + \sqrt{3})z)}{(6 + (3 + \sqrt{3})z)^2}, \end{aligned}$$

and

$$a_n = \tilde{R}(z)u_n + \tilde{\Phi}_1(z)F_n, \quad (26)$$

$$b_n = R(z)u_n + \tilde{\Phi}_2(z)(2F_n^a - F_n), \quad (27)$$

with

$$\begin{aligned} \tilde{R}(z) &= \frac{6(24 + 4\sqrt{3}z - (1 + \sqrt{3})z^2)}{(12 + (3 + \sqrt{3})z)^2}, \\ \tilde{\Phi}_1(z) &= \frac{6k(12 + (3 + 2\sqrt{3})z)}{(12 + (3 + \sqrt{3})z)^2}, \\ \tilde{\Phi}_2(z) &= \frac{6k(6 + (3 + 2\sqrt{3})z)}{(6 + (3 + \sqrt{3})z)^2}. \end{aligned}$$

The rational functions $R(z)$ and $\tilde{R}(z)$ are Calahan rational approximations of e^{-z} and $e^{-z/2}$, respectively. The rational functions $\tilde{\Phi}_1(z)$ and $\tilde{\Phi}_2(z)$ are obtained by simplifying $P_1(z/2)$ and $P_1(z)$ in (21) and (22), respectively. The rational functions $\Phi_1(z)$, $\Phi_2(z)$, and $\Phi_3(z)$ are obtained from the second, third, and fourth terms of (23), respectively.

3.3. Split Version of the New Method

The use of rational approximation (24) resolves the issues of computing matrix exponential functions and computing matrix inverses A^{-1} and A^{-3} but it creates another issue of computing matrix polynomial inverses $(6I + (3 + \sqrt{3})kA)^{-2}$ and $(12I + (3 + \sqrt{3})kA)^{-2}$. This is handled by applying splitting techniques, as follows:

$$\begin{aligned} R(z) &= -0.7320508076 + \frac{2.784609691}{z + 1.267949192} - \frac{0.7461339179}{(z + 1.267949192)^2}, \\ \Phi_1(z) &= \frac{0.7320508076}{z + 1.267949192} - \frac{0.6602540378}{(z + 1.267949192)^2}, \\ \Phi_2(z) &= \frac{0.2679491924}{(z + 1.267949192)^2}, \\ \Phi_3(z) &= \frac{1.0}{z + 1.267949192} - \frac{1.0}{(z + 1.267949192)^2}, \\ \tilde{R}(z) &= -0.7320508076 + \frac{5.569219382}{z + 2.535898385} - \frac{2.984535672}{(z + 2.535898385)^2}, \\ \tilde{\Phi}_1(z) &= \frac{1.732050808}{z + 2.535898385} - \frac{1.176914536}{(z + 2.535898385)^2}, \\ \tilde{\Phi}_2(z) &= \frac{1.732050808}{z + 1.267949192} - \frac{0.5884572681}{(z + 1.267949192)^2}. \end{aligned}$$

Let poles and residues for u_{n+1} be denoted as

$$\begin{aligned} \omega_0 &= -0.7320508076, \quad \omega_1 = 2.784609691, \quad \omega_2 = -0.7461339179, \\ \omega_{11} &= 0.7320508076, \quad \omega_{12} = -0.6602540378, \\ \omega_{21} &= 0.0000000000, \quad \omega_{22} = 0.2679491924, \\ \omega_{31} &= 1.0000000000, \quad \omega_{32} = -1.0000000000, \\ c_1 &= 1.267949192, \end{aligned}$$

and let poles and residues for a_n and b_n be denoted by

$$\begin{aligned} \tilde{\omega}_0 &= -0.7320508076, \quad \tilde{\omega}_1 = 5.569219382, \quad \tilde{\omega}_2 = -2.984535672, \\ \tilde{\omega}_{11} &= 1.732050808, \quad \tilde{\omega}_{12} = -1.176914536, \end{aligned}$$

$$\begin{aligned}\tilde{\omega}_{21} &= 1.732050808, & \tilde{\omega}_{22} &= -0.5884572681, \\ \tilde{c}_1 &= 2.535898385, & \tilde{c}_2 &= 1.267949192 = c_1.\end{aligned}$$

Then we can rewrite it as

$$\begin{aligned}R(z) &= \omega_0 + \frac{\omega_1}{(z+c_1)} + \frac{\omega_2}{(z+c_1)^2}, \\ \Phi_1(z) &= \frac{\omega_{11}}{(z+c_1)} + \frac{\omega_{12}}{(z+c_1)^2}, \\ \Phi_2(z) &= \frac{\omega_{22}}{(z+c_1)^2}, \\ \Phi_3(z) &= \frac{\omega_{31}}{(z+c_1)} + \frac{\omega_{32}}{(z+c_1)^2}, \\ \tilde{R}(z) &= \tilde{\omega}_0 + \frac{\tilde{\omega}_1}{(z+\tilde{c}_1)} + \frac{\tilde{\omega}_2}{(z+\tilde{c}_1)^2}, \\ \tilde{\Phi}_1(z) &= \frac{\tilde{\omega}_{11}}{(z+\tilde{c}_1)} + \frac{\tilde{\omega}_{12}}{(z+\tilde{c}_1)^2}, \\ \tilde{\Phi}_2(z) &= \frac{\tilde{\omega}_{21}}{(z+\tilde{c}_1)} + \frac{\tilde{\omega}_{22}}{(z+\tilde{c}_1)^2},\end{aligned}$$

Using these split forms, methods (25)–(27) can be written as

$$\begin{aligned}u_{n+1} &= \left[\omega_0 + \frac{\omega_1}{(z+c_1)} + \frac{\omega_2}{(z+c_1)^2} \right] u_n + \left[\frac{\omega_{11}}{(z+c_1)} + \frac{\omega_{12}}{(z+c_1)^2} \right] F_n \\ &\quad + \left[\frac{\omega_{22}}{(z+c_1)^2} \right] F_n^a + \left[\frac{\omega_{31}}{(z+c_1)} + \frac{\omega_{32}}{(z+c_1)^2} \right] F_n^b,\end{aligned}$$

and

$$a_n = \left[\tilde{\omega}_0 + \frac{\tilde{\omega}_1}{(z+\tilde{c}_1)} + \frac{\tilde{\omega}_2}{(z+\tilde{c}_1)^2} \right] u_n + \left[\frac{\tilde{\omega}_{11}}{(z+\tilde{c}_1)} + \frac{\tilde{\omega}_{12}}{(z+\tilde{c}_1)^2} \right] F_n, \quad (28)$$

$$b_n = \left[\omega_0 + \frac{\omega_1}{(z+c_1)} + \frac{\omega_2}{(z+c_1)^2} \right] u_n + \left[\frac{\tilde{\omega}_{21}}{(z+\tilde{c}_1)} + \frac{\tilde{\omega}_{22}}{(z+\tilde{c}_1)^2} \right] (2F_n^a - F_n). \quad (29)$$

3.4. Algorithm

In this subsection, we provide the algorithm that we developed to implement the new method.

To compute a_n :

1. Solve $(kA + \tilde{c}_1 I)W = \tilde{\omega}_2 u_n + \tilde{\omega}_{12} F_n$ for W .
2. Solve $(kA + \tilde{c}_1 I)X = \tilde{\omega}_1 u_n + \tilde{\omega}_{11} F_n + W$ for X .
3. Compute $a_n = \tilde{\omega}_0 u_n + X$.

To compute b_n :

1. Solve $(kA + c_1 I)W = \omega_2 u_n + \tilde{\omega}_{22}(2F_n^a - F_n)$ for W .
2. Solve $(kA + c_1 I)X = \omega_1 u_n + \tilde{\omega}_{21}(2F_n^a - F_n) + W$ for X .
3. Compute $b_n = \omega_0 u_n + X$.

To compute u_{n+1} :

1. Solve $(kA + c_1 I)W = \omega_2 u_n + \omega_{12} F_n + \omega_{22} F_n^a + \omega_{32} F_n^b$ for W .
2. Solve $(kA + c_1 I)X = \omega_1 u_n + \omega_{11} F_n + \omega_{31} F_n^b + W$ for X .
3. Compute $u_{n+1} = \omega_0 u_n + X$.

4. Stability Regions and Linear Error Analysis

The general strategy for the stability analysis of a numerical method, which uses a variety of techniques for the linear and nonlinear parts of the equation, is provided by considering the following nonlinear autonomous ODE [42],

$$u_t = cu + F(u), \quad (30)$$

where $F(u)$ is a nonlinear function of the complex-valued function u . Let $u_0 = u(t_0)$ be a fixed point, such that $cu_0 + F(u_0) = 0$. When we consider u as a perturbation of the fixed point u_0 and $\lambda = F'(u_0)$, then the linearization results in the following equation are as follows:

$$u_t = cu + \lambda u. \quad (31)$$

We say that fixed point u_0 is stable if $\text{Re}(c + \lambda) < 0$ [42]. When a numerical method is applied to (31), the regions of stability are determined by taking into account the ratio u_{n+1}/u_n . For this, we assume that $x = \lambda k$ and $y = ck$, where k is the time step size.

When applying this new method to solve (31), we obtain the ratio

$$R(x, y) = \frac{u_{n+1}}{u_n} = \frac{c_0 + c_1x + c_2x^2 + c_3x^3}{\text{Den}},$$

with

$$\begin{aligned} \text{Den} &= (6 + (3 + \sqrt{3})y)^4(12 + (3 + \sqrt{3})y)^2, \\ c_0 &= -216((19 + 11\sqrt{3})y^6 + (120 + 70\sqrt{3})y^5 + (102\sqrt{3} + 168)y^4 \\ &\quad - (204\sqrt{3} + 360)y^3 - (1224 + 756\sqrt{3})y^2 - (1296 + 720\sqrt{3})y - 864), \\ c_1 &= -216((19\sqrt{3} + 33)y^5 - (25\sqrt{3} + 44)y^4 - (918 + 534\sqrt{3})y^3 \\ &\quad - (2220 + 1344\sqrt{3})y^2 - (2016 + 1152\sqrt{3})y - 1152), \\ c_2 &= -216((33 + 19\sqrt{3})y^4 - (127\sqrt{3} + 222)y^3 - (990 + 582\sqrt{3})y^2 \\ &\quad - (1044y + 576\sqrt{3})y - 576), \\ c_3 &= -216((-90\sqrt{3} - 156)y^3 - (474 + 276\sqrt{3})y^2 - (216\sqrt{3} + 396)y - 144). \end{aligned}$$

Linear Error Analysis

In this subsection, we prove the third-order convergence of the method for the specific case. We demonstrate that our new method has a third-order local truncation error in time using the linearization of $\mathbf{F}(\mathbf{u})$ by $\Lambda\mathbf{u}$. We consider the following linear semi-discretized linearized system

$$\frac{d\mathbf{u}}{dt} = -A\mathbf{u} + \Lambda\mathbf{u}, \quad (32)$$

where A and Λ are the linear spatial discretization matrices. The exact solution of (32) is

$$\mathbf{u}(t_{n+1}) = e^{k(\Lambda - A)}\mathbf{u}(t_n). \quad (33)$$

By applying (25) to (32), we obtain

$$\begin{aligned} \mathbf{u}_{n+1} &= \left(6I + (3 + \sqrt{3})kA\right)^{-2} \times \\ &\quad \left[36I + 12\sqrt{3}kA - (1 + \sqrt{3})k^2A^2 + 6k\Lambda + 6(1 + \sqrt{3})k^2\Lambda A\right]u_n \\ &\quad + \left(6I + (3 + \sqrt{3})kA\right)^{-2} \left[24k\Lambda a_n + 6k\Lambda(1 + (2 + \sqrt{3})kA)b_n\right] \end{aligned} \quad (34)$$

where $\mathbf{a}_n$ and $\mathbf{b}_n$ are

$$\mathbf{a}_n = \left(12I + (3 + \sqrt{3})kA\right)^{-2} \times \left[144I + 24\sqrt{3}kA - 6(1 + \sqrt{3})k^2A^2 + 72k\Lambda + 6(3 + 2\sqrt{3})k\Lambda A\right]u_n, \quad (35)$$

$$\mathbf{b}_n = \left(6I + (3 + \sqrt{3})kA\right)^{-2} \times \left[36I + 12k(\sqrt{3}A + 3\Lambda) - (1 + \sqrt{3})k^2A^2 + 6(3 + 2\sqrt{3})k^2\Lambda A\right](2a_n - u_n), \quad (36)$$

Using the following Taylor series expansions

$$\begin{aligned} \frac{1}{(12 + (3 + \sqrt{3})z)^2} &= \frac{1}{144} - \left(\frac{1}{288} \frac{\sqrt{3}}{864}\right)z + \left(\frac{1}{576} + \frac{\sqrt{3}}{1152}\right)z^2 \\ &- \left(\frac{1}{1152} + \frac{5\sqrt{3}}{10368}\right)z^3 + \left(\frac{35}{82944} + \frac{5\sqrt{3}}{20736}\right)z^4 + \mathcal{O}(z^5), \end{aligned} \quad (37)$$

$$\begin{aligned} \frac{1}{(6 + (3 + \sqrt{3})z)^2} &= \frac{1}{36} - \left(\frac{1}{36} \frac{\sqrt{3}}{108}\right)z + \left(\frac{1}{36} + \frac{\sqrt{3}}{72}\right)z^2 \\ &- \left(\frac{1}{36} + \frac{5\sqrt{3}}{324}\right)z^3 + \left(\frac{35}{1296} + \frac{5\sqrt{3}}{324}\right)z^4 + \mathcal{O}(z^5), \end{aligned} \quad (38)$$

Equation (34) is simplified as

$$\begin{aligned} \mathbf{u}_{n+1} &= \left[I + k(\Lambda - A) + k^2\left(\frac{\Lambda^2}{2} - \Lambda A + \frac{A^2}{2}\right) + k^3\left(\frac{\Lambda^3}{6} - \frac{\Lambda^2 A}{2} + \frac{\Lambda A^2}{2} - \frac{A^3}{6}\right) \right. \\ &+ \left. k^4\left(-\frac{\Lambda^4}{24} + \left(\frac{5}{24} + \frac{\sqrt{3}}{9}\right)\Lambda^3 A - \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)\Lambda^2 A^2 + \dots\right) + \dots \right] \mathbf{u}_n. \end{aligned} \quad (39)$$

We rewrite the exact solution (33) using the Taylor series expansion of e^{-z}

$$\begin{aligned} \mathbf{u}_{n+1} &= \left[I + k(\Lambda - A) + k^2\left(\frac{\Lambda^2}{2} - \Lambda A + \frac{A^2}{2}\right) + k^3\left(\frac{\Lambda^3}{6} - \frac{\Lambda^2 A}{2} + \frac{\Lambda A^2}{2} - \frac{A^3}{6}\right) \right. \\ &+ \left. k^4\left(\frac{\Lambda^4}{24} - \frac{\Lambda^3 A}{6} + \frac{\Lambda^2 A^2}{4} - \frac{\Lambda A^3}{6} + \frac{A^4}{24}\right) + \mathcal{O}(k^5) \right] \mathbf{u}_n \end{aligned} \quad (40)$$

It follows from (39) and (40) that the local truncation error is

$$\mathcal{E}_{n+1} = \left\| \mathbf{u}_{n+1} - e^{-k(A-\Lambda)} \mathbf{u}_n \right\| = \mathcal{O}(k^4). \quad (41)$$

5. Numerical Experiments

In this section, we compute and discuss the solutions obtained by applying the new method to three well-known practical test problems, an enzyme kinetics equation, the Fisher equation, and the Allen–Cahn equation. The stability region of the proposed method is shown in Figure 1. First, we solve these first two problems in one dimension and plot the solutions at $t = 1$ for each $\alpha(x, t)$. The surface plots given in Figures 2 and 3 show the behaviors of the solutions at $t = 1$ corresponding to fractional derivatives of variable-order $\alpha(x, t)$. Figure 4 shows the solution profiles of Section 5.1 using ETD Crank–Nicolson scheme at $t = 1$ with 80 time and 160 time steps. Because of the A -stability of the ETD Crank–Nicolson scheme, unwanted oscillations occur in the solution profile when initial data are non-smooth. Since the initial data are not smooth in Section 5.1 due to mismatched initial and boundary conditions, unwanted oscillations can be seen in the solution profiles

shown in Figure 4. These oscillations reduce in magnitude by decreasing the time-step size. But this makes the scheme computationally more expensive. The graph of the initial condition of Section 5.2 is showing in Figure 5. Note that the graph of the solution profile in Figure 3 is rotated to make it more visible. Next, we solve the same problems in two dimensions and plot the solutions at $t = 1$ corresponding to the fractional derivatives of variable orders $\alpha(x, t)$ and $\beta(y, t)$ with $1 \leq \alpha(x, t), \beta(y, t) \leq 2$. For simplicity, functions $\alpha(x, t)$ and $\beta(y, t)$ are taken as the same in Problem 5.2. Their graphs are the same as the graph of $\alpha(x, t)$ given in Figure 2. The graphs of $\beta(y, t)$ in Section 5.4 are the same as the graphs of $\alpha(x, t)$ in the same problem. The effects of the variable-order fractional diffusion are shown in the graphs of the solution profiles. The graphical behaviour of the initial condition used in Section 5.4 can be see in the Figure 6. The solution profiles of the Section 5.4 are computed at different times are shown in the Figures 7–9. The initial condition of Section 5.5 is plotted in the Figure 10. Numerical solutions of the Allen–Cahn Section 5.5 are computed at $t = 5$ for various values of α . These graphs are shown in Figure 11.

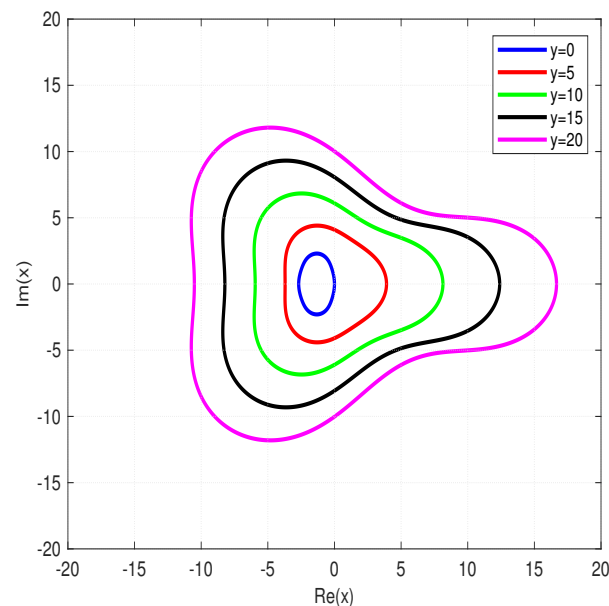


Figure 1. Stability regions of the method.

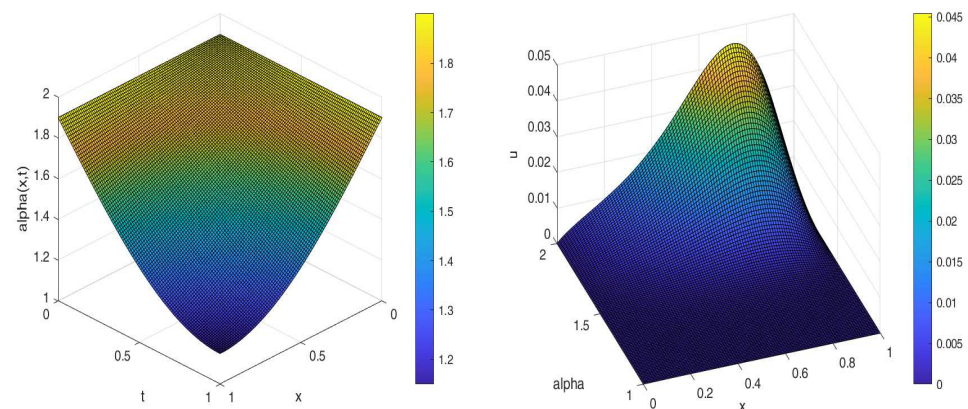


Figure 2. Section 5.1: Graph $\alpha(x, t)$ (left) and solution profile at $t = 1$ using $\alpha(x, t)$ (right).

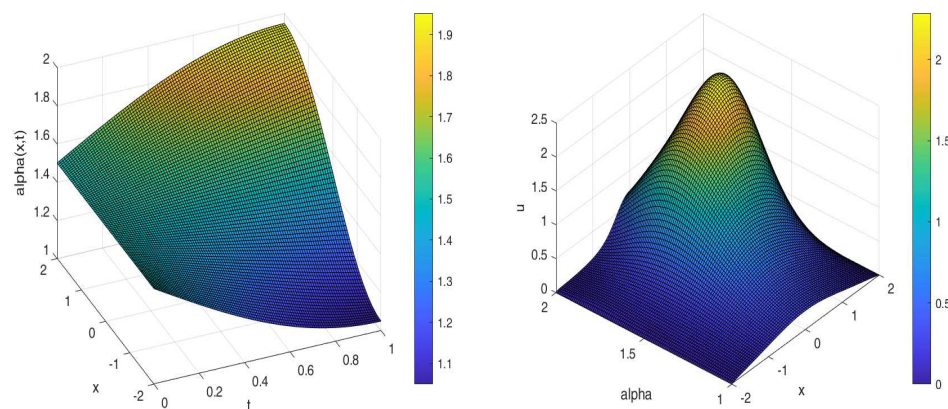


Figure 3. Section 5.3: Graph $\alpha(x, t)$ (left) and solution profile at $t = 1$ using $\alpha(x, t)$ (right).

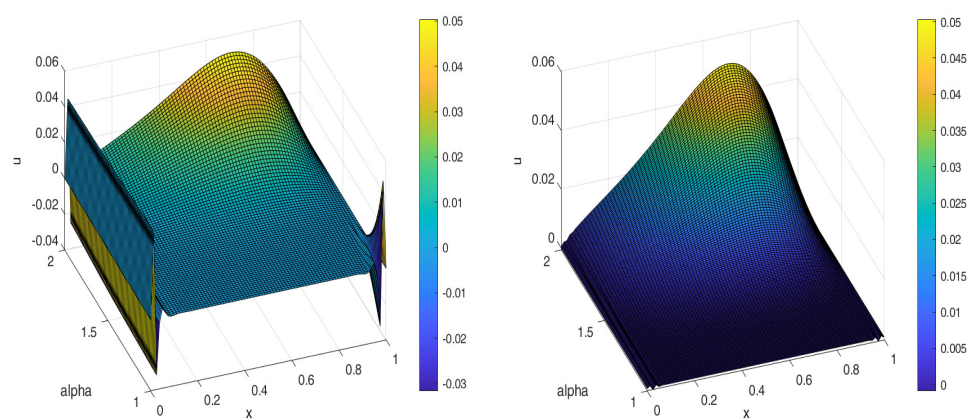


Figure 4. Section 5.1: Graphs of solution profiles using ETD Crank–Nicolson scheme at $t = 1$ using $\alpha(x, t)$ with 80 time steps (left) and 160 time steps (right).

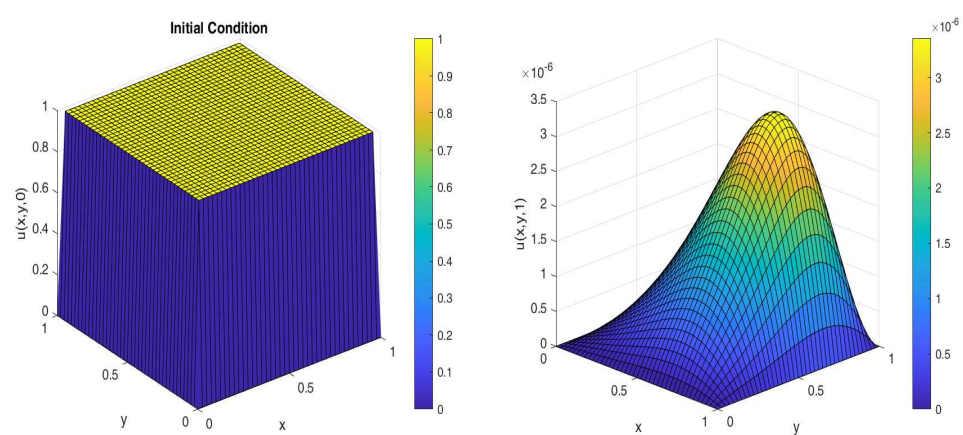


Figure 5. Section 5.2: Initial condition (left) and Solution at $t = 1$ using $\alpha(x, t)$ and $\beta(y, t)$ (right).

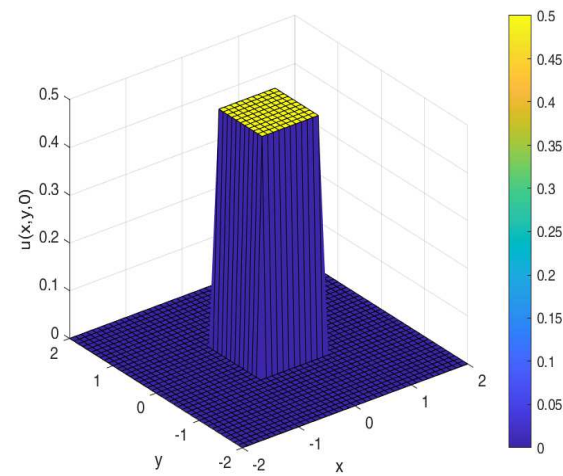


Figure 6. Section 5.4: Graph of the initial conditions.

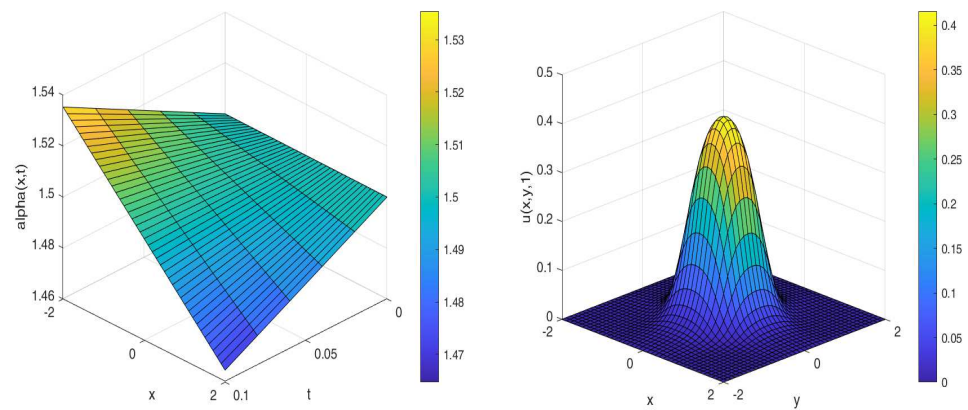


Figure 7. Section 5.4: Graph of $\alpha(x,t)$ for $0 \leq t \leq 0.1$ (left) and solution at $t = 0.1$ (right).

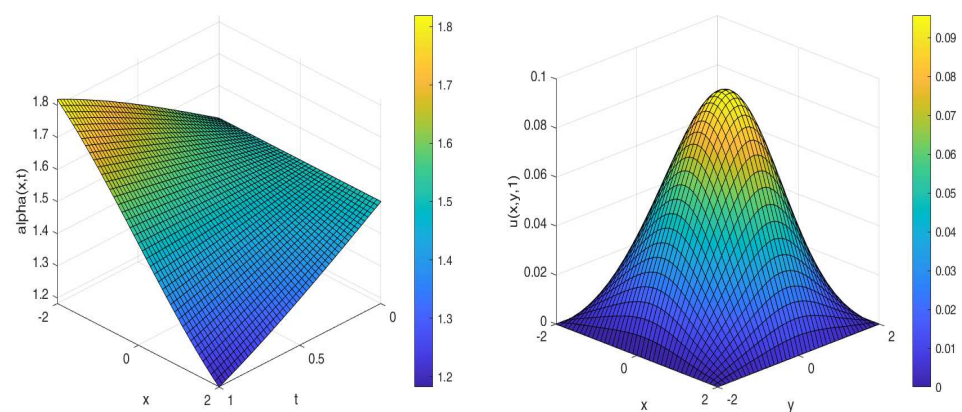


Figure 8. Section 5.4: Graph $\alpha(x,t)$ for $0 \leq t \leq 1$ (left) and the solution at $t = 1$ (right).

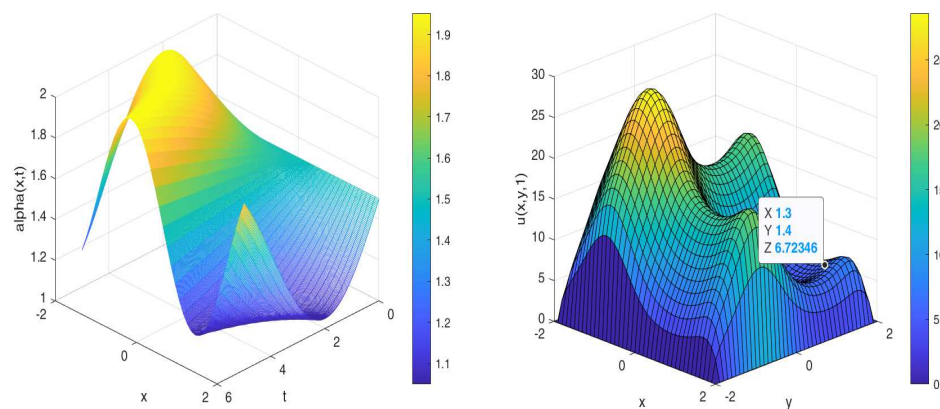


Figure 9. Section 5.4: Graph $\alpha(x, t)$ for $0 \leq t \leq 5$ (left) and solution at $t = 5$ (right).

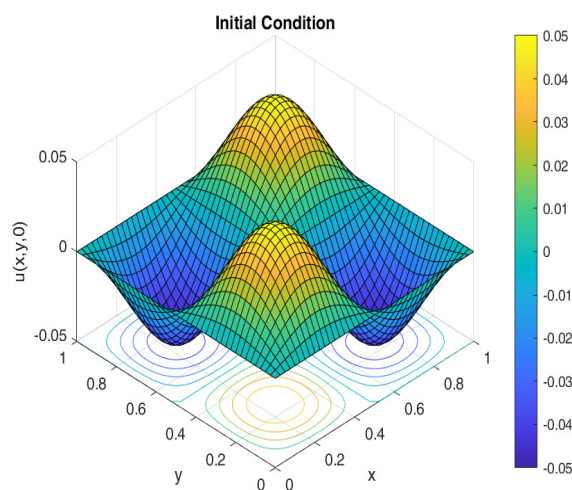


Figure 10. Section 5.5: Graph of the initial conditions.

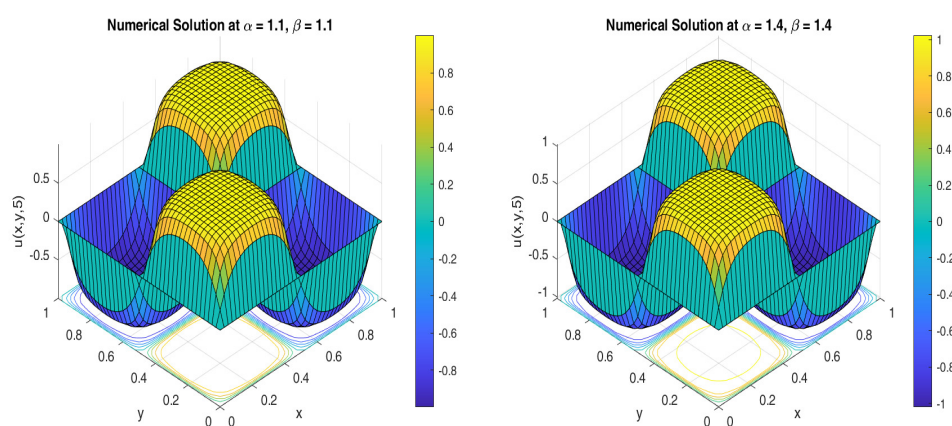


Figure 11. Cont.

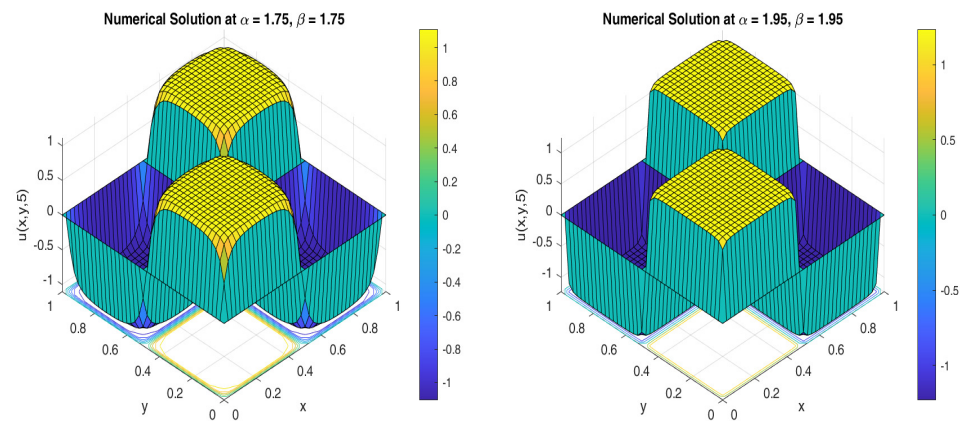


Figure 11. Section 5.5: Graph of the numerical solution of the Allen–Cahn equation at $t = 5$ for various values of α and β .

The superiority of our new scheme over an existing ETD Crank–Nicolson scheme [43] is demonstrated by solving Section 5.1 using both schemes. Third order convergence of our scheme is shown in the Table 1. The second-order convergence of the ETD Crank–Nicolson scheme is shown in Table 2. The ETD Crank–Nicolson scheme is also computationally efficient because it is developed by using the (1, 1)-Padé rational approximation with a single real pole but it is of second-order. Third order convergence of our scheme is also shown in Tables 3 by solving Section 5.3.

Table 1. Section 5.1: Convergence results for our time-stepping method.

Time Steps	Error – L_2	Error – L_∞	Order – L_2	Order – L_∞	CPU Time
20	1.0439×10^{-3}	8.6572×10^{-4}			0.0015
40	2.7328×10^{-5}	1.2955×10^{-5}	2.6555	2.6623	0.0016
80	3.9789×10^{-6}	1.8861×10^{-6}	2.7799	2.7800	0.0019
160	5.4031×10^{-7}	2.5612×10^{-7}	2.8805	2.8805	0.0022
320	7.0630×10^{-8}	3.3480×10^{-8}	2.9354	2.9354	0.0042

Table 2. Section 5.1: Convergence results for the ETD Crank–Nicolson stepping method.

Time Steps	Error – L_2	Error – L_∞	Order – L_2	Order – L_∞	CPU Time
40	8.1702×10^{-5}	3.8398×10^{-5}	0.0000	0.0000	0.0015
80	2.0658×10^{-5}	9.7932×10^{-6}	1.9837	1.9712	0.0018
160	5.1855×10^{-6}	2.4583×10^{-6}	1.9941	1.9941	0.0019
320	1.2972×10^{-6}	6.1495×10^{-7}	1.9991	1.9991	0.0012
640	3.2428×10^{-7}	1.5373×10^{-7}	2.0001	2.0001	0.0063

Table 3. Section 5.3: Convergence results for the time-stepping method.

Time Steps	Error – L_2	Error – L_∞	Order – L_2	Order – L_∞	CPU Time
20	2.3984×10^{-4}	9.7746×10^{-5}			0.0015
40	3.7514×10^{-5}	1.5335×10^{-5}	2.6766	2.6722	0.0025
80	5.3142×10^{-6}	2.1845×10^{-6}	2.8195	2.8115	0.0032
160	7.0947×10^{-7}	2.9247×10^{-7}	2.9050	2.9010	0.0043
320	9.2693×10^{-8}	3.8234×10^{-8}	2.9362	2.9353	0.0084

The order of convergence is computed using absolute errors between the consecutive iterations with the following formula:

$$\mathcal{R} = \log_2 \frac{\mathcal{E}(2k)}{\mathcal{E}(k)},$$

where $\mathcal{E}(k) = |u(2k) - u(k)|$ is the error between consecutive solutions corresponding to time steps $2k$ and k . This approach is used to compute the order of convergence when analytical solutions are not available; see, for example, [35,44]. Convergence errors are computed using the L_2 norm and L_∞ norm. The computational efficiency of the new method is shown by recording the CPU time for each iteration and is given in the convergence tables. It can be seen that the CPU time is very small even with a large number of time steps. All the calculations are performed using Matlab version R2020b on an Intel Core i7 CPU with 4.0 GHz of speed and 32 GB RAM.

5.1. Space Variable-Order Fractional Enzyme Kinetics Equation in One Dimension

$$\frac{\partial u}{\partial t} = \frac{\partial^{\alpha(x,t)} u}{\partial |x|^{\alpha(x,t)}} - \frac{u}{1+u}, \quad 0 < x < 1, \quad t > 0, \quad (42)$$

$$u(x, 0) = 1, \quad 0 < x < 1,$$

$$u(0, t) = 0, \quad u(1, t) = 0, \quad t > 0,$$

$$\alpha(x, t) = 1.95 - 0.75 \sin(0.5\pi xt). \quad (43)$$

5.2. Space Variable-Order Fractional Enzyme Kinetics Equation in Two Dimensions

$$\frac{\partial u}{\partial t} = \frac{\partial^{\alpha(x,t)} u}{\partial |x|^{\alpha(x,t)}} + \frac{\partial^{\beta(y,t)} u}{\partial |y|^{\beta(y,t)}} - \frac{u}{1+u}, \quad (x, y) \in \Omega, \quad t > 0, \quad (44)$$

$$u(x, y, 0) = 1, \quad (x, y) \in \Omega,$$

$$u(x, y, t) = 0, \quad (x, y) \in \partial\Omega, \quad t > 0,$$

$$\alpha(x, t) = 1.95 - 0.75 \sin(0.5\pi xt), \quad \beta(y, t) = 1.95 - 0.75 \sin(0.5\pi yt),$$

where $\Omega = [0, 1] \times [0, 1]$.

5.3. Space Variable-Order Fractional Fisher Equation in One Dimension

$$\frac{\partial u}{\partial t} = \frac{\partial^{\alpha(x,t)} u}{\partial |x|^{\alpha(x,t)}} - u(1-u), \quad -2 < x < 2, \quad t > 0, \quad (45)$$

$$u(x, 0) = \frac{4e^{10x}}{(e^{10x} + 1)^2}, \quad -2 < x < 2,$$

$$u(-2, t) = 0, \quad u(2, t) = 0, \quad t > 0,$$

$$\alpha(x, t) = 1.5 + 0.45 \sin(0.25\pi xt). \quad (46)$$

5.4. Space Variable-Order Fractional Fisher Equation in Two Dimensions

$$\frac{\partial u}{\partial t} = \frac{\partial^{\alpha(x,t)} u}{\partial |x|^{\alpha(x,t)}} + \frac{\partial^{\beta(y,t)} u}{\partial |y|^{\beta(y,t)}} - u(1-u), \quad (x, y) \in \Omega, \quad t > 0, \quad (47)$$

$$u(x, y, 0) = \begin{cases} 1, & (x, y) \in [-0.5, 0.5] \times [-0.5, 0.5], \\ 0, & \text{elsewhere,} \end{cases}$$

$$u(x, y, t) = 0, \quad (x, y) \in \partial\Omega, \quad t > 0, \quad (48)$$

$$\alpha(x, t) = 1.5 - 0.45 \sin(0.125\pi xt), \quad \beta(y, t) = 1.5 - 0.45 \sin(0.125\pi yt),$$

where $\Omega = [-2, 2] \times [-2, 2]$.

5.5. Allen–Cahn Equation in Two Dimensions

$$\frac{\partial u}{\partial t} = \frac{\partial^{\alpha} u}{\partial |x|^{\alpha}} + \frac{\partial^{\beta} u}{\partial |y|^{\beta}} + u(1-u^2), \quad (x, y) \in \Omega, \quad t > 0, \quad (49)$$

$$u(x, y, 0) = 0.05 \sin(2\pi x) \sin(2\pi y) \quad (50)$$

$$u(x, y, t) = 0, (x, y) \in \partial\Omega, \quad t > 0, \quad (51)$$

where $\Omega = [0, 1] \times [0, 1]$.

6. Conclusions

A novel, highly efficient numerical method has been developed using a single Gaussian quadrature pole rational approximation of the matrix exponential function. An algorithm was constructed based on the method to easily implement the method. Three variable-order fractional–reaction–diffusion problems with nonlinear reaction terms were solved in one and two dimensions. The third-order convergence of the method was proved analytically and verified numerically. Graphs of the solution profiles were plotted to demonstrate the effect of the variable-order diffusion. In the future, we plan to develop similar schemes for systems of reaction–diffusion problems, where several coupled nonlinear equations need to be solved.

Author Contributions: The numerical scheme was developed by M.Y. The stability and linear convergence analyses and numerical experiments were conducted by M.Y., S.S. worked on the introduction and spacial discretization. The writing was conducted by both M.Y. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the funding support provided by the Deanship of Research Oversight and Coordination (DROC) at King Fahd University of Petroleum & Minerals (KFUPM), Kingdom of Saudi Arabia.

Data Availability Statement: Not applicable.

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

References

1. Waurick, M. Homogenization in fractional elasticity. *SIAM J. Math. Anal.* **2014**, *46*, 1551–1576. [\[CrossRef\]](#)
2. Luchko, Y.; Punzi, A. Modeling anomalous heat transport in geothermal reservoirs via fractional diffusion equations. *GEM-Int. J. Geomath.* **2011**, *1*, 257–276. [\[CrossRef\]](#)
3. Rida, S.Z.; Yahya, A.A.; Zidan, N.A.; Bakry, H.M. Fractional order of mathematical systems for some bio-chemical application. *Fract. Calc. Appl. Anal.* **2014**, *5*, 25.
4. Maji, S.; Natesan, S. Analytical and numerical solutions of time-fractional advection-diffusion-reaction equation. *Appl. Numer. Math.* **2023**, *185*, 549–570. [\[CrossRef\]](#)
5. Singh, J. Analysis of fractional blood alcohol model with composite fractional derivative. *Chaos Solitons Fractals* **2020**, *140*, 110–127. [\[CrossRef\]](#)
6. Chechkin, A.V.; Gorenflo, R.; Sokolov, I.M. Fractional diffusion in inhomogeneous media. *J. Phys. Gen. Phys.* **2005**, *38*, 679–684. [\[CrossRef\]](#)
7. Sun, H.G.; Chen, W.; Chen, Y. Variable-order fractional differential operators in anomalous diffusion modeling. *Phys. A* **2009**, *388*, 4586–4592. [\[CrossRef\]](#)
8. Zhuang, P.; Liu, F.; Anh, V.; Turner, I. Numerical methods for the variable-order fractional advection-diffusion equation with a nonlinear source term. *SIAM J. Numer. Anal.* **2009**, *47*, 1760–1781. [\[CrossRef\]](#)
9. Diaz, G.; Coimbra, C.F.M. Nonlinear dynamics and control of a variable order oscillator with application to the van der Pol equation. *Nonlinear Dynamics* **2009**, *56*, 145–157. [\[CrossRef\]](#)
10. Kumar, P.; Chaudhary, S.K. Analysis of fractional order control system with performance and stability. *International J. Eng. Sci. Technol.* **2017**, *9*, 408–416.
11. Obembe, A.D.; Hossain, M.E.; Abu-Khamsin, S.A. Variable-order derivative time fractional diffusion model for heterogeneous porous media. *J. Pet. Sci. Eng.* **2017**, *152*, 391–405. [\[CrossRef\]](#)
12. Patnaik, S.; Hollkamp, J.P.; Semperlotti, F. Applications of variable-order fractional operator—A review. *Proc. R. Soc.* **2020**, *476*, 20190498. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Li, C.P.; Zeng, F.H. *Numerical Methods for Fractional Differential Calculus*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2015.
14. Li, C.P.; Cai, M. *Theory and Numerical Approximations of Fractional Integrals and Derivatives*; SIAM Philadelphia: Philadelphia, PA, USA, 2019.
15. Glockle, W.G.; Nonnenmacher, T.F. A fractional calculus approach to self-similar protein dynamics. *Biophys. J.* **1995**, *68*, 46–53. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Klass, D.L.; Martinek, T.W. Electroviscous fluids. I. Rheological Properties. *J. Appl. Phys.* **1967**, *38*, 67–74. [\[CrossRef\]](#)

17. Shiga, T. Deformation and viscoelastic behavior of polymer gel in electric fields. *Proc. Jpn. Acad. Ser. Phys. Biol. Sci.* **1998**, *74*, 6–11. [\[CrossRef\]](#)
18. Davis, L.C. Model of Magnetorheological Elastomers. *J. Appl. Phys.* **1999**, *85*, 3342–3351. [\[CrossRef\]](#)
19. Samko, S.G.; Ross, B. Integration and differentiation to a variable fractional order. *Integral Transform. Spec. Funct.* **1993**, *1*, 277–300. [\[CrossRef\]](#)
20. Lin, R.; Liu, F.; Anh, V.; Turner, I. Stability and convergence of a new explicit finite difference approximation for the variable-order nonlinear fractional diffusion equation. *Appl. Math. Comput.* **2009**, *2*, 435–445. [\[CrossRef\]](#)
21. Sun, H.; Chen, W.; Li, C.P.; Chen, Y. Finite difference schemes for variable order time fractional diffusion equation. *Int. J. Bifurc. Chaos* **2012**, *22*, 1250085. [\[CrossRef\]](#)
22. Kheirikhah, F.; Hajipour, M.; Baleanu, D. The performance of a numerical scheme on the variable-order time-fractional advection-reaction-subdiffusion equations. *Appl. Numer. Math.* **2022**, *178*, 25–40. [\[CrossRef\]](#)
23. Heydari, M.H.; Avazzadeh, Z.; Razzaghi, M. Vieta-Lucas polynomials for the coupled nonlinear variable-order fractional Ginzburg-Landau equations. *Appl. Numer. Math.* **2021**, *165*, 442–458. [\[CrossRef\]](#)
24. Yan, R.; Ma, Q.; Ding, X. Convergence analysis of the hp-version spectral collocation method for a class of nonlinear variable-order fractional differential equations. *Appl. Numer. Math.* **2021**, *170*, 269–297. [\[CrossRef\]](#)
25. Darve, E.; Delia, M.; Garrappa, R.; Giusti, A.; Rubio, N.L. On the fractional Laplacian of variable order. *Fract. Calc. Appl. Anal.* **2022**, *25*, 15–28. [\[CrossRef\]](#)
26. Garrappa, R.; Giusti, A.; Mainardi, F. Variable-order fractional calculus. A change of perspective. *Commun. Nonlinear Sci. Numer. Simul.* **2021**, *102*, 105904. [\[CrossRef\]](#)
27. Xu, Y.; Sun, H.; Zhang, Y.; Sun, H.W. A novel meshless method based on RBF for solving variable-order time fractional advection-diffusion-reaction equation in linear or nonlinear systems. *Image. Comput. Math. Appl.* **2023**, *142*, 107–120. [\[CrossRef\]](#)
28. Li, X.Y.; Wu, B.Y. Iterative reproducing kernel method for nonlinear variable-order space fractional diffusion equations. *Int. J. Comput. Math.* **2018**, *95*, 1210–1221.
29. Sun, H.; Chang, A.; Zhang, Y.; Chen, W. A review on variable-order fractional differential equation. Mathematical foundations, physical models, and its applications. *Fract. Calc. Appl. Anal.* **2019**, *22*, 27–59. [\[CrossRef\]](#)
30. Khaliq, A.Q.M.; Liang, X.; Furati, K.M. A fourth-order implicit-explicit scheme for the space fractional nonlinear Schrödinger equations. *Numer. Algorithms* **2017**, *75*, 147–172. [\[CrossRef\]](#)
31. Ngo, H.T.B.; Razzaghi, M.; Vo, T.N. Fractional order Chelyshkov wavelet method for solving variable order fractional differential equations and an application in variable order fractional relaxation system. *Numer. Algor.* **2023**, *92*, 1571–1588. [\[CrossRef\]](#)
32. Ding, H.F. The construction of an optimal fourth-order fractional-compact-type numerical differential formula of the Riesz derivative and its application. *Commun. Nonlinear Sci. Numer. Simul.* **2023**, *23*, 107272. [\[CrossRef\]](#)
33. Ding, H.F.; Li, C.P. High order numerical algorithm and error analysis for the two-dimensional nonlinear spatial fractional complex Ginzburg-Landau equation. *Commun. Nonlinear Sci. Numer. Simul.* **2023**, *120*, 107160. [\[CrossRef\]](#)
34. Furati, K.M.; Yousuf, M.; Khaliq, A.Q.M. Fourth-order methods for space fractional reaction-diffusion equations with non-smooth data. *Int. J. Comput. Math.* **2018**, *95*, 1240–1256. [\[CrossRef\]](#)
35. Yousuf, M.; Furati, K.M.; Khaliq, A.Q.M. High-order time-stepping methods for two-dimensional Riesz fractional nonlinear reaction-diffusion equations. *Comput. Math. Appl.* **2020**, *80*, 204–226. [\[CrossRef\]](#)
36. Wang, Q.Y.; She, Z.H.; Lao, C.X.; Lin, F.R. Fractional centered difference schemes and banded preconditioners for nonlinear Riesz space variable-order fractional diffusion equations. *Numer. Algor.* **2023**. <http://doi.org/10.1007/s11075-023-01592-z>. [\[CrossRef\]](#)
37. Thomee, V. *Galerkin Finite Element Methods for Parabolic Problems*; Series in Computational Mathematics; Springer: Berlin/Heidelberg, Germany, 2006.
38. Ortigueira, M.D. Riesz potential operators and inverses via fractional centered derivatives. *Int. J. Math. Math. Sci.* **2006**, *2006*, 48391. [\[CrossRef\]](#)
39. Celik, C.; Duman, M. Crank-Nicolson method for the fractional diffusion equation with the Riesz fractional derivative. *J. Comput. Phys.* **2012**, *231*, 1743–1750. [\[CrossRef\]](#)
40. Pazy, A. *Semigroups of Linear Operators and Applications to Partial Differential Equations*; Springer: Berlin/Heidelberg, Germany, 1983.
41. Norsett, S.P.; Wolfbrandt, A. Attainable order of rational approximations to the exponential function with only real poles. *Bit Numer. Math.* **1977**, *17*, 200–208. [\[CrossRef\]](#)
42. Cox, S.M.; Matthews, P.C. Exponential Time Differencing for Stiff Systems. *J. Comput. Phys.* **2002**, *176*, 430–455. [\[CrossRef\]](#)
43. Kleefeld, B.; Khaliq, A.Q.M.; Wade, B.A. An ETD Crank-Nicolson method for reaction-diffusion systems. *Numer. Methods Partial. Differ. Eq.* **2012**, *28*, 1309–1335. [\[CrossRef\]](#)
44. Pooley, D.M.; Vetzal, K.R.; Forsyth, P.A. Convergence Remedies for Non-Smooth Payoffs in Option Pricing. *J. Comput. Financ.* **2003**, *6*, 25–40. [\[CrossRef\]](#)

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.