fractal and fractional

Article

An Implicit Numerical Method for the Riemann–Liouville Distributed-Order Space Fractional Diffusion Equation

Mengchen Zhang [1], Ming Shen [2,*] and Hui Chen [3]

1 School of Mathematical Sciences, Queensland University of Technology (QUT), GPO Box 2434, Brisbane, QLD 4001, Australia; m64.zhang@hdr.qut.edu.au
2 School of Mathematics and Statistics, Fuzhou University, Fuzhou 350108, China
3 School of Mechanical Engineering and Automation, Fuzhou University, Fuzhou 350108, China; felixceng@fzu.edu.cn
* Correspondence: shenming0516@fzu.edu.cn

Abstract: This paper investigates a two-dimensional Riemann–Liouville distributed-order space fractional diffusion equation (RLDO-SFDE). However, many challenges exist in deriving analytical solutions for fractional dynamic systems. Efficient and reliable methods need to be explored for solving the RLDO-SFDE numerically. We develop an alternating direction implicit scheme and prove that the numerical method is unconditionally stable and convergent with an accuracy of $O(\sigma^2 + \rho^2 + \tau + h_x + h_y)$. After employing an extrapolated technique, the convergence order is improved to second order in time and space. Furthermore, a fast algorithm is constructed to reduce computational costs. Two numerical examples are presented to verify the effectiveness of the numerical methods. This study may provide more possibilities for simulating diffusion complexities by fractional calculus.

Keywords: distributed-order operator; diffusion equation; fractional derivative; implicit alternating direction method; extrapolated technique

MSC: 35R11; 60J60; 65R10

Citation: Zhang, M.; Shen, M.; Chen, H. An Implicit Numerical Method for the Riemann–Liouville Distributed-Order Space Fractional Diffusion Equation. *Fractal Fract.* **2023**, *7*, 382. https://doi.org/10.3390/fractalfract7050382

Academic Editor: Rodica Luca

Received: 27 March 2023
Revised: 28 April 2023
Accepted: 29 April 2023
Published: 2 May 2023

1. Introduction

In past centuries, the classical differential equation was regarded as a powerful tool to describe many physical phenomena. However, there exist many challenges in modelling some abnormal diffusion processes and physical behaviours in complex systems only by integer-order differential equations. Fractional derivative operators can be used to describe the memory effect and heterogeneous properties that are not covered by integer-order models, which give a more realistic description of complex physical processes [1]. Increasing interest has been given to the application of fractional differential equations in various fields, such as finance [2], information security [3], viscoelasticity [4], imaging processing [5,6], and control problems [7,8]. By virtue of the non-local property, fractional derivatives have shown great potential for simulating diffusion processes in complex and heterogeneous media [9,10]. In recent years, extensive attention has been drawn to distributed-order differential equations since they are more effective in modelling phenomena exhibiting varying non-local properties over multiple scales.

The distributed-order operator, generalised by integrating the order of fractional derivatives over a given range, has accumulated considerable popularity during recent years [11]. Due to the utility of this operator in describing behaviours with multi-scale varying characteristics, extensive attention has been attracted to the investigation of distributed-order differential equations [12,13]. In 1969, Caputo [14] first presented the idea of the distributed-order equation, which was solved in 1995 [15]. Li et al. [16] studied fractional differential equations involving the distributed-order operator to investigate ultra-slow diffusion. Some physical processes exhibit non-local behaviours that are variable with time or space. It was found that the distributed-order fractional diffusion equation is an efficient tool to describe complex diffusion processes, especially when the diffusion exponent varies over multi-scales [17]. Lorenzo and Hartley [18] concluded that both integer-order and fractional-order systems are special cases of distributed-order systems. Space distributed-order fractional equations are more flexible in their ability to capture diffusion phenomena with multi-scale memory characteristics [19]. However, studies on space distributed-order diffusion equations are still limited, especially for the Riemann–Liouville fractional derivative. Furthermore, high-dimensional partial differential equations (PDEs) will be more suitable for the simulation of diffusion phenomena in the real world. Motivated by this, we consider the Riemann–Liouville distributed-order space fractional diffusion equation (RLDO-SFDE) in two dimensions.

Since it is difficult to derive the analytical solutions of fractional dynamic systems, many numerical methods for solving distributed-order fractional PDEs have been proposed [20–23]. By using numerical integration techniques, one-dimensional time distributed-order fractional models have been transformed into multi-term fractional systems which were finally solved by spectral methods [20,21]. A meshless method was proposed for solving the time distributed-order anomalous sub-diffusion equation [22]. However, efficient and robust numerical schemes are still under development, especially for high-dimensional cases. The computational cost of solving two-dimensional distributed-order fractional equations is very large, and highly accurate techniques are demanded [24]. Further research needs to be focused on exploring effective numerical methods.

The discretised fractional diffusion equation generates a dense coefficient matrix; it requires high computational costs and memory storage, especially for high-dimensional problems. To tackle this issue, we employ the alternating direction implicit method (ADIM) [25] to solve a complex two-dimension fractional system. Peaceman and Rachford [26] proposed the ADIM to solve classical parabolic and elliptic PDEs efficiently. This efficient method has gained considerable popularity for solving large matrix equations numerically, such as diffusion equations in two or higher dimensions [26]. Meerschaert et al. [25] developed a variation on the classical ADIM to solve two-dimensional fractional PDEs with variable coefficients. The theory of this fractional ADIM is based on splitting the finite difference equations into two iteration systems, first solving the x-direction iteration to calculate the intermediate solution and then dealing with the y-direction iteration system. Compared with the Gaussian elimination, the ADIM has the advantage of reducing the two-dimensional problem into two one-dimensional problems, which improves the efficiency and reduces the computational cost remarkably [27,28]. Therefore, we develop the ADIM to solve the two-dimensional RLDO-SFDE. Furthermore, to improve the convergence order of the numerical method, an extrapolated technique [29] is employed to solve the fractional system efficiently. This Richardson extrapolated technique is a more general approach that can also be combined with other numerical methods to improve the convergence order further. For example, the modified Crank–Nicolson scheme with Richardson extrapolation has been investigated for solving many PDEs [30,31]. This extrapolated technique may offer more possibilities for the research on developing numerical methods with higher-order accuracy.

The coefficient matrices in iterated systems of the ADIM are very dense for high-dimensional problems with fine meshes, which leads to high computational costs and memory storage. In view of the Toeplitz-like structure of these coefficient matrices, we develop a fast algorithm by constructing circulant matrices and employing fast Fourier transform. The idea of this fast method has been investigated extensively [32–34]. The fast ADIM does not require the assembly of coefficient matrices, which saves memory storage remarkably. The implemented fast Fourier transform accelerates the calculation with lower computational costs.

In this paper, we investigate the two-dimensional RLDO-SFDE. Firstly, some preliminary knowledge related to this research is introduced in Section 2. Section 3 discretises the RLDO-SFDE as a multi-term fractional equation by employing the midpoint quadrature rule. Distributed-order fractional PDEs can be regarded as the limiting case of multi-term fractional PDEs [35]. Diethelm and Ford [36] have observed that small changes in the order of a fractional PDE lead to only slight changes in the final solution, which gives initial support to the employed numerical integration method. Then, we use an implicit Euler method, which includes a left-shifted Grünwald–Letnikov approximation [29] to discretise the Riemann–Liouville fractional derivative. Moreover, the discretised equation is solved by the ADIM. In Section 4, we provide the stability and convergence of the numerical scheme. More importantly, an extrapolated technique and a fast algorithm are constructed to improve the ADIM in Section 5. Section 6 verifies the effectiveness of the developed numerical schemes by comparing the error between the exact solutions and numerical results. Finally, some conclusions are drawn in Section 7.

2. Preliminaries

When we discretise a space fractional diffusion equation by applying the standard Grünwald–Letnikov fractional derivative, the finite difference approximation is unstable whether the difference method is the explicit Euler method, implicit Euler method, or the Crank–Nicholson method. Thus, we take the use of a left-shifted Grünwald–Letnikov approximation, which is unconditionally stable, to discretise the Riemann–Liouville fractional derivative.

Theorem 1 ([1])**.** *Suppose $\alpha \in (1,2)$, $f(x) \in C^2[a,b]$; then, the definite left-shifted Grünwald –Letnikov fractional derivative is defined as:*

$$ {}_a^G D_{x,1}^{\alpha} f(x) = \frac{1}{h^\alpha} \sum_{j=0}^{(x-a)/h+1} g_j^{(\alpha)} f(x-(j-1)h) \tag{1} $$

Equation (1) is the approximation of the left Riemann–Liouville fractional derivative if and only if $f(a) = 0$, i.e.,

$$ {}_a^G D_{x,1}^{\alpha} f(x) = {}_a^R D_x^{\alpha} f(x) + O(h). $$

where the left Riemann–Liouville fractional derivative is defined as [1]

$$ {}_a^R D_x^{\alpha} f(x) = \frac{1}{\Gamma(2-\alpha)} \frac{d^2}{dx^2} \int_a^x \frac{f(\xi)}{(x-\xi)^{\alpha-1}} d\xi, $$

where $\Gamma(\cdot)$ refers to the gamma function.

When $f(a) \neq 0$, then

$$ {}_a^G D_{x,1}^{\alpha} f(x) = {}_a^R D_x^{\alpha} f(x) + O(h) + O(f(a)). $$

Lemma 1 ([37])**.** *When $1 < \alpha < 2$, the coefficients $g_k^{(\alpha)} = (-1)^k \binom{\alpha}{k} (k = 0, \pm 1, \pm 2, \cdots)$ satisfy*

1. $g_0^{(\alpha)} = 1, g_1^{(\alpha)} = -\alpha, g_k^{(\alpha)} > 0 (k = 0, 2, 3, \cdots);$
2. $\sum\limits_{k=0}^{\infty} g_k^{(\alpha)} = 0;$
3. $\sum\limits_{k=0}^{N} g_k^{(\alpha)} < 0$ *for any positive integer N.*

Lemma 2 ([29])**.** *Let $X = [x_1, x_2, \cdots, x_m]^T$, $\|X\|_\infty = \max_{1 \le i \le m} |x_i|$. If the matrix $D = (d_{i,l})_{m \times m} \in \mathbb{R}^{m \times m}$ satisfies the following conditions,*

$$ \sum_{l=1, l \neq i}^{m} |d_{i,l}| \le |d_{i,i}| - 1, \qquad (i = 1, 2, \cdots, m), $$

then

$$\|X\|_\infty \leq \|DX\|_\infty.$$

3. Numerical Methods

Meerschaert et al. [25] investigated the following two-dimensional fractional diffusion equation with variable coefficients:

$$\frac{\partial u(x,y,t)}{\partial t} = K_1(x,y)\frac{\partial^\alpha u(x,y,t)}{\partial x^\alpha} + K_2(x,y)\frac{\partial^\beta u(x,y,t)}{\partial y^\beta} + f(x,y,t), \quad (2)$$

where $K_1(x,y)$ and $K_2(x,y)$ refer to variable diffusion coefficients. $\partial^\alpha u(x,y,t)/\partial x^\alpha$ and $\partial^\beta u(x,y,t)/\partial y^\beta$ are Riemann–Liouville fractional derivative operators.

It is difficult to simulate the diffusion processes in heterogeneous media with varying characteristics only by the fixed fractional parameters [38]. Based on the above fractional diffusion model (2), this study generalises space fractional derivatives by using distributed-order operators, which can describe diffusion behaviours exhibited varying non-local properties. Regarding diffusion coefficients as constants, we consider the following two-dimensional RLDO-SFDE:

$$\frac{\partial u(x,y,t)}{\partial t} = K_x \int_1^2 P(\alpha)\frac{\partial^\alpha u(x,y,t)}{\partial x^\alpha}\, d\alpha + K_y \int_1^2 Q(\beta)\frac{\partial^\beta u(x,y,t)}{\partial y^\beta}\, d\beta + f(x,y,t), \quad (3)$$

$$x \in (a,b), y \in (c,d), \Omega = (a,b) \times (c,d), t \in (0,T],$$

where K_x and K_y refer to diffusion coefficients. $P(\alpha)$ and $Q(\beta)$ are weight functions satisfying the following conditions:

$$P(\alpha) \geq 0, \quad P(\alpha) \not\equiv 0, \quad \int_1^2 P(\alpha)\, d\alpha < \infty, \quad \alpha \in [1,2], \quad (4)$$

$$Q(\beta) \geq 0, \quad Q(\beta) \not\equiv 0, \quad \int_1^2 Q(\beta)\, d\beta < \infty, \quad \beta \in [1,2]. \quad (5)$$

The boundary and initial conditions of the two-dimensional fractional diffusion Equation (3) are

$$u(x,y,t) = 0, (x,y,t) \in (R^2 \setminus \Omega) \times (0,T], \quad (6)$$

$$u(x,y,0) = u_0(x,y), (x,y) \in \Omega \cup \partial\Omega, \quad (7)$$

where $\partial\Omega$ is the boundary of the domain Ω.

3.1. The Discretisation of Integral Intervals

Firstly, the integral intervals [1, 2] of α and [1, 2] of β are discretised as $1 = \xi_0 < \xi_1 < \cdots < \xi_S = 2$, $1 = \eta_0 < \eta_1 < \cdots < \eta_{\bar{S}} = 2$. We denote $\Delta\xi_j = \xi_j - \xi_{j-1} = 1/S = \sigma$, $\alpha_j = (\xi_j + \xi_{j-1})/2 = 1 + (2j-1)/2S, j = 1,2,\cdots,S, S \in N$, and $\Delta\eta_j = \eta_j - \eta_{j-1} = 1/\bar{S} = \rho$, $\beta_j = (\eta_j + \eta_{j-1}) = 1 + (2j-1)/2\bar{S}, j = 1,2,\cdots,\bar{S}, \bar{S} \in N$. Then, adopting the midpoint quadrature formula [39], we can obtain

$$\begin{aligned}&\int_1^2 P(\alpha)\frac{\partial^\alpha u(x,y,t)}{\partial x^\alpha}\, d\alpha + \int_1^2 Q(\beta)\frac{\partial^\beta u(x,y,t)}{\partial y^\beta}\, d\beta \\ &= \sum_{j=1}^{S} P(\alpha_j)\frac{\partial_j^\alpha u(x,y,t)}{\partial x_j^\alpha}\Delta\xi_j + \sum_{j=1}^{\bar{S}} Q(\beta_j)\frac{\partial_j^\beta u(x,y,t)}{\partial y_j^\beta}\Delta\eta_j + O(\sigma^2 + \rho^2). \quad (8)\end{aligned}$$

Then, the two-dimensional Riemann–Liouville space distributed-order diffusion Equation (3) is discretised as follows:

$$\begin{aligned}\frac{\partial u(x,y,t)}{\partial t} &= K_x \sum_{j=1}^{S} P(\alpha_j) \frac{\partial_j^{\alpha} u(x,y,t)}{\partial x_j^{\alpha}} \Delta\xi_j + K_y \sum_{j=1}^{\bar{S}} Q(\beta_j) \frac{\partial_j^{\beta} u(x,y,t)}{\partial y_j^{\beta}} \Delta\eta_j \\ &+ f(x,y,t) + O(\sigma^2+\rho^2).\end{aligned} \tag{9}$$

3.2. The ADIM

The interval $\Omega \times (0,T]$ is divided by $x_i = a + ih_x(i = 0,1,2,\cdots,M_1)$, $y_l = c + lh_y(l = 0,1,2,\cdots,M_2)$, and $t_n = n\tau(n = 0,1,2,\cdots,N)$, where $h_x = \frac{b-a}{M_1}$, $h_y = \frac{d-c}{M_2}$ are space steps, and $\tau = T/N$ is the time step. According to the backward difference, the integer-order term can be written as:

$$\frac{\partial u(x_i,y_l,t_{n-1/2})}{\partial t} = \frac{u(x_i,y_l,t_n) - u(x_i,y_l,t_{n-1})}{\tau} + O(\tau). \tag{10}$$

Using the implicit Euler method [29], which contains a left-shifted Grünwald–Letnikov approximation, the Riemann–Liouville space fractional derivative operators can be discretised as the following forms:

$$\frac{\partial^{\alpha_j} u(x_i,y_l,t_n)}{\partial x^{\alpha_j}} = \frac{1}{h_x^{\alpha_j}} \sum_{k=0}^{i+1} g_k^{\alpha_j} u(x_{i-k+1},y_l,t_n) + O(h_x), \tag{11}$$

$$\frac{\partial^{\beta_j} u(x_i,y_l,t_n)}{\partial y^{\beta_j}} = \frac{1}{h_y^{\beta_j}} \sum_{k=0}^{l+1} g_k^{\beta_j} u(x_i,y_{l-k+1},t_n) + O(h_y), \tag{12}$$

where $g_k^{\alpha_j} = (-1)^k \binom{\alpha_j}{k}$.

Thus, the implicit difference method of Equation (3) can be approximated by the following equation:

$$\begin{aligned}&\frac{u(x_i,y_l,t_n) - u(x_i,y_l,t_{n-1})}{\tau} \\ =&K_x \sum_{j=0}^{S} \frac{P(\alpha_j)}{S} \frac{1}{h_x^{\alpha_j}} \sum_{k=0}^{i+1} g_k^{\alpha_j} u(x_{i-k+1},y_l,t_n) + K_y \sum_{j=1}^{\bar{S}} \frac{Q(\beta_j)}{\bar{S}} \frac{1}{h_y^{\beta_j}} \sum_{k=0}^{l+1} g_k^{\beta_j} u(x_i,y_{l-k+1},t_n) \\ &+ f(x_i,y_l,t_n) + O(\sigma^2+\rho^2+\tau+h_x+h_y),\end{aligned} \tag{13}$$

which can be written as:

$$\begin{aligned}&\frac{u(x_i,y_l,t_n) - u(x_i,y_l,t_{n-1})}{\tau} \\ =&K_x \sum_{j=0}^{S} \frac{P(\alpha_j)}{S} \frac{1}{h_x^{\alpha_j}} \sum_{k=0}^{i+1} g_{i-k+1}^{\alpha_j} u(x_k,y_l,t_n) + K_y \sum_{j=1}^{\bar{S}} \frac{Q(\beta_j)}{\bar{S}} \frac{1}{h_y^{\beta_j}} \sum_{k=0}^{l+1} g_{l-k+1}^{\beta_j} u(x_i,y_k,t_n) \\ &+ f(x_i,y_l,t_n) + R_{i,l}^n, \\ &(i = 0,1,2,\cdots,M_1, l = 0,1,2,\cdots,M_2, n = 0,1,2,\cdots,N),\end{aligned} \tag{14}$$

where the truncation error is denoted as $R_{i,l}^n = O(\sigma^2+\rho^2+\tau+h_x+h_y)$.

After the truncation error is neglected, the implicit numerical method for Equation (3) is obtained as:

$$\frac{u_{i,l}^n - u_{i,l}^{n-1}}{\tau} = K_x \sum_{j=0}^{S} \frac{P(\alpha_j)}{S} \frac{1}{h_x^{\alpha_j}} \sum_{k=0}^{i+1} g_{i-k+1}^{\alpha_j} u_{k,l}^n + K_y \sum_{j=1}^{\bar{S}} \frac{Q(\beta_j)}{\bar{S}} \frac{1}{h_y^{\beta_j}} \sum_{k=0}^{l+1} g_{l-k+1}^{\beta_j} u_{i,k}^n + f_{i,l}^n, \tag{15}$$

$$u_{0,l}^n = u_{M_1,l}^n = 0, u_{i,0}^n = u_{i,M_2}^n = 0, (0 \leq n \leq N), \tag{16}$$

$$u_{i,l}^0 = u_0(x_i, y_l), (0 \leq i \leq M_1, 0 \leq l \leq M_2). \tag{17}$$

In order to solve the above Equations (15)–(17), two fractional partial difference operators are defined as follows:

$$\delta_x u_{i,l}^n = K_x \sum_{j=0}^{S} \frac{P(\alpha_j)}{S} \frac{1}{h_x^{\alpha_j}} \sum_{k=0}^{i+1} g_{i-k+1}^{\alpha_j} u_{k,l}^n, \tag{18}$$

$$\delta_y u_{i,l}^n = K_y \sum_{j=1}^{\bar{S}} \frac{Q(\beta_j)}{\bar{S}} \frac{1}{h_y^{\beta_j}} \sum_{k=0}^{l+1} g_{l-k+1}^{\beta_j} u_{i,k}^n. \tag{19}$$

Thus, the difference method can be rewritten as the following form:

$$(1 - \tau\delta_x - \tau\delta_y) u_{i,l}^n = u_{i,l}^{n-1} + \tau f_{i,l}^n. \tag{20}$$

After introducing the truncation error $\tau^2 \delta_x \delta_y u_{i,l}^n$, which is far smaller than the original error, Equation (20) is written as:

$$(1 - \tau\delta_x)(1 - \tau\delta_y) u_{i,l}^n = u_{i,l}^{n-1} + \tau f_{i,l}^n, \ \ 0 \leq i \leq M_1, 0 \leq l \leq M_2. \tag{21}$$

The above Equation (21) is the ADIM of the two-dimensional Riemann–Liouville space distributed-order diffusion equation, and the perturbation error is $O(\sigma^2 + \rho^2 + \tau + h_x + h_y)$.

Then, we can employ the following iteration techniques to find the numerical solution of the ADIM (21):

$$(1 - \tau\delta_x) u_{i,l}^* = u_{i,l}^{n-1} + \tau f_{i,l}^n, \tag{22}$$

$$(1 - \tau\delta_y) u_{i,l}^n = u_{i,l}^*, \tag{23}$$

$$1 \leq i \leq M_1, 1 \leq l \leq M_2.$$

1. Solve Equation (22) in the x-direction for each fixed value of l, so the intermediate solution $u_{i,l}^*$ can be obtained;
2. Solve Equation (23) in the y-direction for each fixed i, so we can obtain the final solution $u_{i,l}^n$.

4. The Stability and Convergence of the ADIM

The ADIM Equation (21) can be rewritten as

$$GHU^n = U^{n-1} + \tau F^n, \tag{24}$$

where $U^n = (u_{1,1}^n, \cdots, u_{M_1-1,1}^n, u_{1,2}^n, \cdots, u_{M_1-1,2}^n, \cdots, u_{1,M_2-1}^n, \cdots, u_{M_1-1,M_2-1}^n)^T$, $F^n = (f_{1,1}^n, \cdots, f_{M_1-1,1}^n, f_{1,2}^n, \cdots, f_{M_1-1,2}^n, \cdots, f_{1,M_2-1}^n, \cdots, f_{M_1-1,M_2-1}^n)^T$.

Additionally, the matrices G and H represent the difference operators $(1 - \tau\delta_x)$ and $(1 - \tau\delta_y)$, respectively. The matrix $G = diag(A_1, A_2, \cdots, A_{M_2-1})$ is a block diagonal matrix where the $(M_1 - 1) \times (M_1 - 1)$ block A_l is obtained by Equation (22). Similarly, the $(M_2 - 1) \times (M_2 - 1)$ blocks B_i resulting from Equation (23) lead to the diagonal matrix $H = diag(B_1, B_2, \cdots, B_{M_1-1})$.

Here, the matrices $A_l = (a_{i,k})_{(M_1-1)\times(M_1-1)}^{(l)}$, $1 \leq l \leq M_2 - 1$, note $r_1 = \frac{\tau K_x}{S} > 0$,

$$a_{i,k} = \begin{cases} 0 & \text{for } k > i+1, \\ -r_1 \sum\limits_{j=1}^{S} \dfrac{P(\alpha_j)}{h_x^{\alpha_j}} g_0^{(\alpha_j)} & \text{for } k = i+1, \\ 1 - r_1 \sum\limits_{j=1}^{S} \dfrac{P(\alpha_j)}{h_x^{\alpha_j}} g_1^{(\alpha_j)} & \text{for } k = i, \\ -r_1 \sum\limits_{j=1}^{S} \dfrac{P(\alpha_j)}{h_x^{\alpha_j}} g_{i-k+1}^{(\alpha_j)} & \text{for } k < i. \end{cases} \tag{25}$$

The matrices $B_i = (b_{k,l})^{(i)}_{(M_2-1)\times(M_2-1)}$, $1 \le i \le M_2 - 1$, note $r_2 = \frac{\tau K_y}{\bar{S}} > 0$,

$$b_{k,l} = \begin{cases} 0 & \text{for } l > k+1, \\ -r_2 \sum\limits_{j=1}^{\bar{S}} \dfrac{Q(\beta_j)}{h_y^{\beta_j}} g_0^{(\beta_j)} & \text{for } l = k+1, \\ 1 - r_2 \sum\limits_{j=1}^{\bar{S}} \dfrac{Q(\beta_j)}{h_y^{\beta_j}} g_1^{(\beta_j)} & \text{for } l = k, \\ -r_2 \sum\limits_{j=1}^{\bar{S}} \dfrac{Q(\beta_j)}{h_y^{\beta_j}} g_{k-l+1}^{(\beta_j)} & \text{for } l < k. \end{cases} \tag{26}$$

Equation (22) can be rewritten as

$$A_l U_l^* = U_l^{n-1} + \tau F_l^n + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ r_1 \sum\limits_{j=1}^{S} \dfrac{P(\alpha_j)}{h_x^{\alpha_j}} g_0^{(\alpha_j)} u^*_{M_1,l} \end{pmatrix}, \qquad 1 \le l \le M_2 - 1, \tag{27}$$

where $U_l^* = (u^*_{1,l}, u^*_{2,l}, \cdots, u^*_{M_1-1,l})^T$, $F_l^n = (f^n_{1,l}, f^n_{2,l}, \cdots, u^n_{M_1-1,l})^T$.

Similarly, Equation (23) can also be written as

$$B_i U_i^n = U_i^* + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ r_2 \sum\limits_{j=1}^{\bar{S}} \dfrac{Q(\beta_j)}{h_y^{\beta_j}} g_0^{(\beta_j)} u^n_{i,M_2} \end{pmatrix}, \qquad 1 \le i \le M_1 - 1, \tag{28}$$

where $U_i^n = (u^n_{i,1}, u^n_{i,2}, \cdots, u^n_{i,M_2-1})^T$, $U_i^* = (u^*_{i,1}, u^*_{i,2}, \cdots, u^*_{i,M_2-1})^T$.

The framework of the numerical method is listed in the following Algorithm 1.

Algorithm 1 Framework of the ADIM for the RLDO-SFDE.

Input: Mesh generation
The total number of time steps N, the total number of spatial nodes M_1 (x-direction) and M_2 (y-direction);
The partitions of integral intervals $\alpha_j = 1 + (2j-1)/2S, (j = 1,2,\dots,S)$ and $\beta_j = 1 + (2j-1)/2\bar{S}, (j = 1,2,\dots,\bar{S})$;
Output: Numerical solutions U^n
- *Generation of weight functions* $P(\alpha_j), Q(\beta_j), g_k^{\alpha_j}$ *and* $g_k^{\beta_j}$;
- *Assemble matrices* A_l *(by Equation (25)),* B_i *(by Equation (26)), and the vector* F_l^n;

for $n = 1 : N$ **do**
  for $l = 1 : M_2 - 1$ **do**
    Compute U_l^* by the system (27).
  end for
  for $i = 1 : M_1 - 1$ **do**
    Compute U_i^n by the system (28).
  end for
  Obtain numerical solutions U^n.
end for

4.1. Stability

Theorem 2. *The ADIM (Equation (21)) is unconditionally stable.*

Proof. In order to prove the stability of the numerical method, we assume that $u_{i,l}^n$ and $\bar{u}_{i,l}^n$ are the numerical solutions and the approximation solutions, respectively. Their corresponding error is $\varepsilon_{i,l}^n = \bar{u}_{i,l}^n - u_{i,l}^n$, noting that

$$Y^n = (\varepsilon_{1,1}^n, \varepsilon_{2,1}^n, \cdots, \varepsilon_{M_1-1,1}^n, \cdots, \varepsilon_{1,M_2-1}^n, \varepsilon_{2,M_2-1}^n, \cdots, \varepsilon_{M_1-1,M_2-1}^n)^T.$$

According to Lemma 1, we can obtain that $a_{i,l} < 0\ (i \neq l)$, $a_{i,i} > 0$. Thus,

$$\begin{aligned}
\sum_{l=1,l\neq i}^{M_1-1} |a_{i,l}| &= r_1 \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_0^{(\alpha_j)} + r_1 \sum_{l=1}^{i-1} \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_{i-k+1}^{(\alpha_j)} \\
&= r_1 \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_0^{(\alpha_j)} + r_1 \sum_{k=2}^{i} \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_k^{(\alpha_j)} \\
&= -r_1 \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_1^{(\alpha_j)} - r_1 \sum_{k=i+1}^{\infty} \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_k^{(\alpha_j)} \\
&= |a_{i,i}| - 1 - r_1 \sum_{k=i+1}^{\infty} \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_k^{(\alpha_j)}.
\end{aligned}$$

It is obtained from Lemma 1 that $g_k^{(\alpha_j)} > 0 (k = 0,2,3,\cdots)$, which leads to the expression $r_1 \sum_{k=i+1}^{\infty} \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_k^{(\alpha_j)} > 0$, so there yields

$$\sum_{l=1,l\neq i}^{M_1-1} |a_{i,l}| = |a_{i,i}| - 1 - r_1 \sum_{k=i+1}^{\infty} \sum_{j=1}^{S} \frac{P(\alpha_j)}{h_x^{\alpha_j}} g_k^{(\alpha_j)} < |a_{i,i}| - 1.$$

Therefore, the matrices $A_l (1 \leq l \leq M_2 - 1)$ satisfy Lemma 2. Similarly, the matrices $B_i (1 \leq i \leq M_1 - 1)$ also satisfy Lemma 2.

According to the relationships between A_l and G, as well as B_i and H, it is clear that matrices G and H meet the conditions of Lemma 2 as well. Thus,

$$\|Y^n\|_\infty \le \|HY^n\|_\infty \le \|GHY^n\|_\infty = \|Y^{n-1}\|_\infty \le \|Y^0\|_\infty.$$

Therefore, the ADIM is unconditionally stable. □

4.2. Convergence

Theorem 3. *The ADIM (Equation (21)) is convergent, and the convergence order is $O(\sigma^2 + \rho^2 + \tau + h_x + h_y)$.*

Proof. Let $u(x_i, y_l, t_n)$ be the exact solutions of Equations (3), (6), and (7), let $u^n_{i,l}$ be the numerical solutions of the ADIM, and note that

$$e^n_{i,l} = u(x_i, y_l, t_n) - u^n_{i,l},$$
$$E^n = (e^n_{1,1}, e^n_{2,1}, \cdots, e^n_{M_1-1,1}, \cdots, e^n_{1,M_2-1}, e^n_{2,M_2-1}, \cdots, e^n_{M_1-1,M_2-1})^T;$$

then, the error E^n satisfies the following perturbed equations:

$$GHE^n = E^{n-1} + \tau R^n, \tag{29}$$

$$e^n_{0,l} = e^n_{M_1,l} = 0,\ e^n_{i,0} = e^n_{i,M_2} = 0,\ 0 \le n \le N, \tag{30}$$

$$e^0_{i,l} = 0,\ 0 \le i \le M_1,\ 0 \le l \le M_2, \tag{31}$$

where $R^n_{i,l} = O(\sigma^2 + \rho^2 + \tau + h_x + h_y)$.

Using Lemma 2, we can obtain

$$\begin{aligned}\|E^n\|\infty &\le \|HE^n\|\infty \le \|GHE^n\|\infty = \|E^{n-1} + \tau R^n\|\infty \\ &\le \|E^{n-1}\|\infty + \tau\|R^n\|\infty \\ &\le \|E^{n-2}\|\infty + 2\tau\|R^n\|\infty \\ &\le \|E^0\|\infty + n\tau\|R^n\|\infty \\ &\le C(\sigma^2 + \rho^2 + \tau + h_x + h_y),\end{aligned}$$

where C is a positive constant.

Therefore, the ADIM is convergent, and the convergence order is $O(\sigma^2 + \rho^2 + \tau + h_x + h_y)$. □

5. Improved Algorithms

This section develops two improved numerical algorithms. An extrapolated technique is employed to increase the convergence order of the ADIM. Then, we provide a fast algorithm to solve the fractional diffusion model more efficiently with lower memory usage and computational costs.

5.1. An Extrapolated Technique

In order to improve the convergence of the numerical method discussed above, we adopt an extrapolated technique [29] at a time step (τ) and space steps (h_x *and* h_y). An extrapolated technique of the Euler method for a one-dimensional fractional diffusion equation has been analysed [35]. The procedures of the extended extrapolation for this research are summarized as follows:

- Employ the ADIM that contains a left-shifted Grünwald–Letnikov approximation to obtain the numerical solutions $u_{i,l}^n(h_x, h_y, \tau)$ on the coarse grid $\Delta t = \tau, \Delta x = h_x, \Delta y = h_y$.
- Solve Equation (3) by the implicit Euler method on the fine grid $\Delta t = \tau/2$, $\Delta x = h_x/2, \Delta y = h_y/2$; then, the numerical solutions $u_{2i,2l}^{2n}(h_x/2, h_y/2, \tau/2)$ can be obtained.
- The new extrapolated solutions can be written as:

$$\begin{aligned}&\tilde{u}_{i,l}^n \approx 2u_{2i,2l}^{2n}(h_x/2, h_y/2, \tau/2) - u_{i,l}^n(h_x, h_y, \tau),\\ &i = 1, 2, \cdots, M_1 - 1, \quad l = 1, 2, \cdots, M_2 - 1.\end{aligned}$$

Therefore, we can obtain the second-order-accuracy numerical solutions by applying the proposed extrapolated technique, and the convergence order of the new extrapolated solution $\tilde{u}_{i,l}^n$ is $O(\tau^2 + h_x^2 + h_y^2)$.

5.2. A Fast Algorithm for the ADIM

The numerical solutions of the RLDO-SFDE can be obtained by solving two matrix systems ((27)–(28)). The Gaussian elimination for solving the system $Ax = b$ with $A \in \mathbb{R}^{M\times M}$ requires $O(M^2)$ memory and $O(M^3)$ operations per time step when A is a full matrix. The coefficient matrices $A_l \in \mathbb{R}^{(M_1-1)\times(M_1-1)}$ and $B_i \in \mathbb{R}^{(M_2-1)\times(M_2-1)}$ in systems ((27) and (28)) are lower Hessenberg–Toeplitz-like matrices. The computational work for solving the matrix system $A_l x = b$ is approximately $M_1 \times (M_1 - 1)$ arithmetic operations [25]. Although it requires less computational work compared with the Gaussian elimination for a full coefficient matrix system, further efficient algorithms need to be explored to reduce the computational complexity for fine meshes. Since the coefficient matrices A_l and B_i are Toeplitz-like matrices, we develop a fast algorithm based on the circulant matrix and the fast Fourier transform to accelerate the computation. This fast algorithm is implemented with the Bi-CGSTAB iterative method, which is suitable for solving non-symmetric linear systems [40]. The main tasks of the fast Bi-CGSTAB ADIM are matrix-vector multiplications $A_l v$ and $B_i \bar{v}$ with $v \in \mathbb{R}^{M_1-1}$ and $\bar{v} \in \mathbb{R}^{M_2-1}$. Taking the matrix-vector multiplication $A_l v$ as an example, we can obtain the following lemma.

Lemma 3. *The matrix-vector multiplication $A_l v$ can be performed in $O((M_1 - 1)\log(M_1 - 1))$ operations for any $v \in \mathbb{R}^{M_1-1}$.*

Proof. The coefficient matrix A_l of the ADIM is a lower Hessenberg–Toeplitz-like matrix formulated in the following form:

$$A_l := \begin{bmatrix} 1 - r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_1^{(\alpha_j)} & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_0^{(\alpha_j)} & 0 & \cdots & 0 \\ -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_2^{(\alpha_j)} & 1 - r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_1^{(\alpha_j)} & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_0^{(\alpha_j)} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_{M_1-2}^{(\alpha_j)} & \ddots & \ddots & \ddots & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_0^{(\alpha_j)} \\ -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_{M_1-1}^{(\alpha_j)} & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_{M_1-2}^{(\alpha_j)} & \cdots & \cdots & 1 - r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_1^{(\alpha_j)} \end{bmatrix}. \tag{32}$$

The Toeplitz matrix can be embedded into the following $2(M_1-1)\times 2(M_1-1)$ circulant matrix $C^{\alpha,2(M_1-1)}$:

$$C^{\alpha,2(M_1-1)} := \begin{bmatrix} A_l & S^{\alpha,M_1-1} \\ S^{\alpha,M_1-1} & A_l \end{bmatrix}, \tag{33}$$

where

$$S^{\alpha,M_1-1} = \begin{bmatrix} 0 & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_{M_1-1}^{(\alpha_j)} & \cdots & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_3^{(\alpha_j)} & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_2^{(\alpha_j)} \\ 0 & 0 & \ddots & \ddots & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_3^{(\alpha_j)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \ddots & 0 & -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_{M_1-1}^{(\alpha_j)} \\ -r_1\sum_{j=1}^{S}\frac{P(\alpha_j)}{h_x^{\alpha_j}}g_0^{(\alpha_j)} & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

For any $\hat{v} := [v^T, 0^T]^T \in \mathbb{R}^{2(M_1-1)}$, we have

$$C^{\alpha,2(M_1-1)}\hat{v} = \begin{bmatrix} A_l v \\ S^{\alpha,M_1-1}v \end{bmatrix}. \tag{34}$$

Then, we can obtain the matrix-vector multiplication $A_l v$ from the first half of $C^{\alpha,2(M_1-1)}\hat{v}$. The circulant matrix $C^{\alpha,2(M_1-1)}$ can be diagonalised as follows by using a discrete Fourier transform [41,42]:

$$C^{\alpha,2(M_1-1)} = F^{-1}_{2(M_1-1)}\mathrm{diag}(\Lambda)F_{2(M_1-1)}, \tag{35}$$

where $\Lambda = F_{2(M_1-1)}c^{\alpha,2(M_1-1)}$, with $c^{\alpha,2(M_1-1)}$ being the first column vector of the matrix $C^{\alpha,2(M_1-1)}$, and $F_{2(M_1-1)}$ is a $2(M_1-1)\times 2(M_1-1)$ discrete Fourier transform. For any $w \in \mathbb{R}^{2(M_1-1)}$, it is known that $F_{2(M_1-1)}w$ and $F^{-1}_{2(M_1-1)}w$ can be performed in $O(2(M_1-1)\log(2(M_1-1))) = O((M_1-1)\log(M_1-1))$ operations via the fast Fourier transform and the inverse fast Fourier transform, respectively [43]. Thus, the computation work of $A_l v$ requires $O((M_1-1)\log(M_1-1))$ operations. □

Similarly, the matrix-vector multiplication $B_i\bar{v}$ can be performed in $O((M_2-1)\log(M_2-1))$ operations for any $\bar{v} \in \mathbb{R}^{M_2-1}$.

Therefore, this fast algorithm based on the fast Fourier transform reduces computational costs. Furthermore, coefficient matrices A_l and B_i do not need to be assembled in the fast algorithm. Only the first column vectors of circulant matrices $C^{\alpha,2(M_1-1)}$ and $C^{\beta,2(M_2-1)}$ are required to be stored in this fast ADIM. Both the memory requirement and computational costs can be reduced in solving the fractional system by the fast Bi-CGSTAB ADIM compared to the classical Gaussian elimination and standard Bi-CGSTAB method.

The algorithm of the fast Bi-CGSTAB iterative method is illustrated in Algorithm 2.

Algorithm 2 Framework of the fast Bi-CGSTAB iterative method.

Input: The eigenvalue vector Λ of the corresponding circulant matrix;
The numerical solution at the previous time level U^{n-1};
The right-hand-side vector b;
Output: Numerical solutions U^n
• $x_0 = U^{n-1}$ is an initial guess, $M = length(x_0)$, defining a zero vector $I_1 = zeros(M, 1)$;
• Compute the product of the circulant matrix C and the vector $\hat{x}_0 = [x_0; I_1]$ using Equations (34) and (35), and let the first M elements be the Toeplitz-like matrix-vector multiplication $A_l x_0 = C\hat{x}_0(1:M)$;
• Compute $r_0 = b - A_l x_0$, where $\hat{r}_0$ is an arbitrary vector such that $(\hat{r}_0, r_0) \neq 0$, e.g., $\hat{r}_0 = r_0$;
• Let $\rho_0 = \alpha_0 = \omega_0 = 1$, $v_0 = p_0 = 0$;
for $i = 1, 2, 3, \cdots$, **do**
$\rho_i = (\hat{r}_0, r_{i-1})$; $\beta = (\rho_i / rho_{i-1})(\alpha_{i-1}/\omega_{i-1})$;
$p_i = r_{i-1} + \beta(p_{i-1} - \omega_{i-1 v_{i-1}})$;
Compute the product of the circulant matrix C and the vector $\hat{p}_i = [p_i; I_1]$ using Equations (34) and (35);
$v_i = C\hat{p}_i(1:M)$; $\alpha_i = \rho_i/(\hat{r}_0, v_i)$;
$s = r_{i-1} - \alpha_i v_i$;
Compute the product of the circulant matrix C and the vector $\hat{s} = [s; I_1]$ using Equations (34) and (35);
$t = C\hat{s}(1:M)$; $x_i = x_{i-1} + \alpha_i p_i + \omega_i s$;
if x_i is accurate enough, then quit;
$r_i = s - \omega_i t$;
end for
Obtain numerical solutions $U^n = x_i$.

6. Numerical Example

This section provides two numerical examples. The effectiveness and reliability of the developed numerical methods are discussed. The numerical computations were performed using MATLAB R2021b on a MacBook Air with the following configuration: Apple M1 chip, 8-core CPU, 16 GB RAM, 512 GB storage, and macOS Ventura 13.1.

In the following numerical examples, the L_∞ error refers to the maximum value of the absolute value of the difference between the numerical solution and the exact solution at $t_n = T$, which is calculated by

$$L_\infty \ error = \max_{(x,y)\in\Omega} |u_{i,j}^n - u(x,y,t_n)|. \tag{36}$$

Example 1. *We consider the two-dimensional RLDO-SFDE:*

$$\frac{\partial u}{\partial t} = \int_1^2 P(\alpha)\frac{\partial^\alpha u}{\partial x^\alpha}\, d\alpha + \int_1^2 Q(\beta)\frac{\partial^\beta u}{\partial y^\beta}\, d\beta + f(x,y,t), \tag{37}$$

in a finite domain $\Omega \in (0,1)\times(0,1)$, $t \in (0,1]$, where $P(\alpha) = \Gamma(5-\alpha)$, $Q(\beta) = \Gamma(5-\beta)$.

The boundary and initial conditions are

$$\begin{aligned} &u(x,y,t) = 0, \ \ (x,y,t) \in (R^2 \setminus \Omega)\times(0,1], \\ &u(x,y,0) = x^2(1-x)^2y^2(1-y)^2, \ \ (x,y) \in \Omega \cup \partial\Omega. \end{aligned}$$

The exact solution of Equation (29) is

$$u(x,y,t) = (t^2+1)x^2(1-x)^2y^2(1-y)^2.$$

Employing the Riemann–Liouville fractional derivative of power functions, we can obtain

$$\begin{aligned}
\frac{\partial^\alpha u}{\partial x^\alpha} &= (t^2+1)y^2(1-y)^2 {}_a^R D_x^\alpha (x^2(1-x)^2) \\
&= (t^2+1)y^2(1-y)^2 {}_a^R D_x^\alpha (x^4-2x^3+x^2) \\
&= (t^2+1)y^2(1-y)^2 \left[{}_a^R D_x^\alpha x^4 - 2 {}_a^R D_x^\alpha x^3 + {}_a^R D_x^\alpha x^2 \right] \\
&= (t^2+1)y^2(1-y)^2 \left[\frac{\Gamma(5)x^{4-\alpha}}{\Gamma(5-\alpha)} - 2\frac{\Gamma(4)x^{3-\alpha}}{\Gamma(4-\alpha)} + \frac{\Gamma(3)x^{2-\alpha}}{\Gamma(3-\alpha)} \right].
\end{aligned}$$

Then,

$$\begin{aligned}
&\int_1^2 P(\alpha)\frac{\partial^\alpha u}{\partial x^\alpha}\,d\alpha \\
=&(t^2+1)y^2(1-y)^2 \int_1^2 \left[\Gamma(5)x^{4-\alpha} - 2(4-\alpha)\Gamma(4)x^{3-\alpha} + (4-\alpha)(3-\alpha)\Gamma(3)x^{2-\alpha}\right] d\alpha \\
=&(t^2+1)y^2(1-y)^2 \int_1^2 \left[24x^{4-\alpha} - 12(4-\alpha)x^{3-\alpha} + 2(\alpha^2-7\alpha+12)x^{2-\alpha}\right] d\alpha \\
=&2(t^2+1)y^2(1-y)^2 \frac{1}{\ln^3 x}\left[2(x-1) + (6x^2-11x+3)\ln x + 2(6x^3-15x^2+9x-1)\ln^2 x\right].
\end{aligned}$$

Likewise,

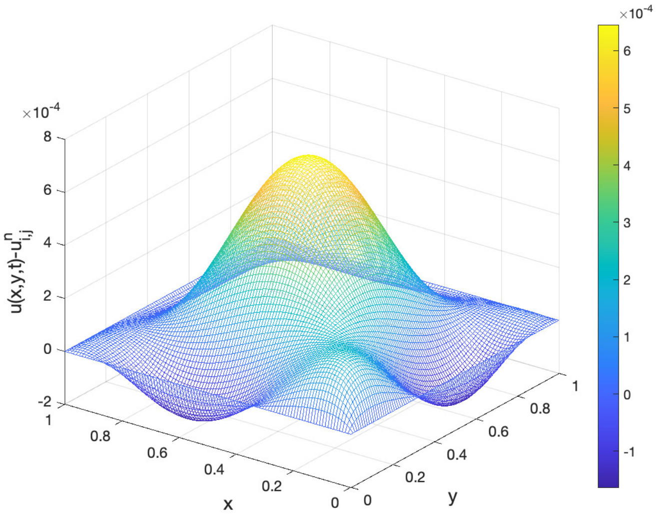
$$\int_1^2 Q(\beta)\frac{\partial^\beta u}{\partial y^\beta}\,d\beta = 2(t^2+1)x^2(1-x)^2 \frac{1}{\ln^3 y}\left[2(y-1) + (6y^2-11y+3)\ln y + 2(6y^3-15y^2+9y-1)\ln^2 y\right].$$

Therefore,

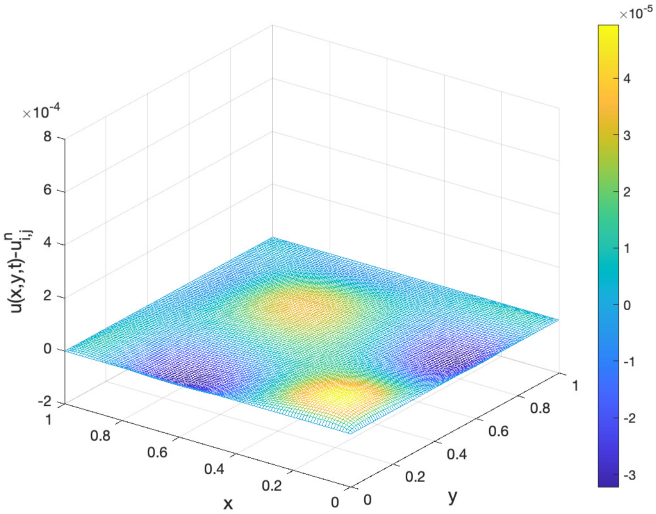
$$\begin{aligned}
f(x,y,t) =& \frac{\partial u(x,y,t)}{\partial t} - \int_1^2 P(\alpha)\frac{\partial^\alpha u}{\partial x^\alpha}\,d\alpha - \int_1^2 Q(\beta)\frac{\partial^\beta u}{\partial y^\beta}\,d\beta \\
=&2tx^2(1-x)^2y^2(1-y)^2 \\
&-2(t^2+1)y^2(1-y)^2 \frac{1}{\ln^3 x}\left[2(x-1) + (6x^2-11x+3)\ln x + 2(6x^3-15x^2+9x-1)\ln^2 x\right] \\
&-2(t^2+1)x^2(1-x)^2 \frac{1}{\ln^3 y}\left[2(y-1) + (6y^2-11y+3)\ln y + 2(6y^3-15y^2+9y-1)\ln^2 y\right].
\end{aligned}$$

To verify the effectiveness of the proposed numerical methods, Figure 1 displays the error between the exact solutions $u(x,y,t)$ and numerical solutions $u_{i,j}^n$ with $\sigma = \rho = 1/200$ and $\tau = h_x = h_y = 1/100$ at $t = 1$. Both error distributions in Figure 1 suggest that the results produced by the proposed two numerical methods are in great agreement with the exact solutions. Furthermore, it is demonstrated that the error distributions in Figure 1b decrease dramatically after the implementation of the extrapolated technique as compared to the error without the extrapolated technique shown in Figure 1a. The result implies that the applied extrapolated technique is very efficient in solving the fractional system numerically.

The L_∞ error and the corresponding convergence order of implicit alternating direction methods without and with the extrapolated technique for different mesh partitions with $\sigma = \rho = 1/200$ at $t = 1$ are compared in Table 1. It is observed that the L_∞ error of the original ADIM maintains $O(\tau + h_x + h_y)$ accuracy. The L_∞ error decreases dramatically after applying the extrapolated technique, as shown in Table 1. More importantly, the convergence order is improved to second-order accuracy by implementing the extrapolated technique.



(**a**) Without the extrapolated technique.



(**b**) Applying the extrapolated technique.

Figure 1. The error between exact solutions $u(x, y, t)$ and numerical solutions $u^n_{i,j}$ for different numerical methods with $\sigma = \rho = 1/200$ and $\tau = h_x = h_y = 1/100$ at $t = 1$. (**a**) The error of the numerical method without the extrapolated technique. (**b**) The error of the numerical method after applying the extrapolated technique.

Table 1. The comparison of L_∞ error and convergence order between ADIM without and with the extrapolated method for different meshes with $\sigma = \rho = 1/200$ at $t = 1$.

	Without Extrapolated Technique		**Extrapolated Technique**	
$\tau = h_x = h_y$	L_∞ **Error**	**Order**	L_∞ **Error**	**Order**
$1/50$	1.2000×10^{-3}	-	1.3159×10^{-4}	-
$1/100$	6.4413×10^{-4}	0.90	4.9333×10^{-5}	1.42
$1/200$	3.3766×10^{-4}	0.93	1.5396×10^{-5}	1.68
$1/400$	1.7308×10^{-4}	0.96	4.3243×10^{-6}	1.83

Example 2. *Consider the two-dimensional RLDO-SFDE in* $\Omega \in (0,1) \times (0,1)$*:*

$$\begin{aligned}
&\frac{\partial u}{\partial t} = \int_1^2 P(\alpha) \frac{\partial^\alpha u}{\partial x^\alpha}\, d\alpha + \int_1^2 Q(\beta) \frac{\partial^\beta u}{\partial y^\beta}\, d\beta + f(x,y,t),\\
&u(x,y,t) = 0, \ \ (x,y,t) \in (R^2 \setminus \Omega) \times (0,1],\\
&u(x,y,0) = x^2(1-x)^2 y^2 (1-y)^2, \ \ (x,y) \in \Omega \cup \partial\Omega.
\end{aligned}$$

where $t \in (0,1]$, $P(\alpha) = \Gamma(5-\alpha)$, $Q(\beta) = \Gamma(5-\beta)$.

The exact solution of this example is given by

$$u(x,y,t) = e^t x^2 (1-x)^2 y^2 (1-y)^2.$$

The source term is formulated as

$$\begin{aligned}
f(x,y,t) =& \frac{\partial u(x,y,t)}{\partial t} - \int_1^2 P(\alpha) \frac{\partial^\alpha u}{\partial x^\alpha}\, d\alpha - \int_1^2 Q(\beta) \frac{\partial^\beta u}{\partial y^\beta}\, d\beta\\
=& e^t x^2 (1-x)^2 y^2 (1-y)^2\\
&- 2e^t y^2 (1-y)^2 \frac{1}{\ln^3 x} \Big[2(x-1) + (6x^2 - 11x + 3) \ln x + 2(6x^3 - 15x^2 + 9x - 1) \ln^2 x \Big]\\
&- 2e^t x^2 (1-x)^2 \frac{1}{\ln^3 y} \Big[2(y-1) + (6y^2 - 11y + 3) \ln y + 2(6y^3 - 15y^2 + 9y - 1) \ln^2 y \Big].
\end{aligned}$$

To test the effectiveness of the proposed fast algorithm, we compare the computational costs of three different ADIMs (Gaussian elimination, Bi-CGSTAB, and fast Bi-CGSTAB). The stopping criterion on the relative residual was 10^{-9} for the standard and fast Bi-CGSTAB algorithms. The L_∞ error between exact solutions $u(x,y,t)$ and numerical solutions $u_{i,j}^n$ and the corresponding convergence order of two different algorithms (Gaussian elimination and fast Bi-CGSTAB) are illustrated in Table 2 with $\sigma = \rho = 1/200$ at $t = 1$. Table 2 shows that the L_∞ error of the numerical methods can achieve first-order convergence both for Gaussian elimination and the fast Bi-CGSTAB method. Furthermore, the comparison of CPU time consumed by Gaussian elimination and the standard and fast Bi-CGSTAB is demonstrated in Table 3. The fast Bi-CGSTAB method requires less CPU time than Gaussian elimination, as displayed in Table 3. More importantly, it is observed from Table 3 that the fast Bi-CGSTAB algorithm leads to a significant reduction of the CPU running time compared to the CPU time consumed by the standard Bi-CGSTAB method. This result indicates that the fast algorithm plays a significant role in reducing computational costs, especially compared with the standard Bi-CGSTAB method. Due to the fact that coefficient matrices in ADIM systems are lower Hessenberg–Toeplitz-like matrices, these systems can be effectively solved by first reducing the system to a lower triangular system using a backward sweep. The solution can be obtained by using Gaussian elimination in a relatively efficient way. This leads to the result that the fast Bi-CGSTAB method shows a slight acceleration of the computation than Gaussian elimination. When the coefficient matrix is a full matrix, the fast Bi-CGSTAB method might show better performance in reducing computational costs. Additionally, this fast algorithm offers great potential to speed up the calculation of the standard Bi-CGSTAB method. Some preconditioned techniques and parallel algorithms can further improve the efficiency of the fast Bi-CGSTAB method. The study of these issues and more effective numerical algorithms will be the goal of our future research.

Table 2. The comparison of L_∞ error and convergence order between the ADIM and fast Bi-CGSTAB ADIM for different meshes with $\sigma = \rho = 1/200$ at $t = 1$.

	Gauss		Fast Bi-CGSTAB	
$\tau = h_x = h_y$	L_∞ **Error**	**Order**	L_∞ **Error**	**Order**
100	8.7548×10^{-4}	-	8.7548×10^{-4}	-
200	4.5894×10^{-4}	0.93	4.5901×10^{-4}	0.93
400	2.3524×10^{-4}	0.96	2.3523×10^{-4}	0.96
600	1.5817×10^{-4}	0.98	1.5814×10^{-4}	0.98

Table 3. The comparison of the consumed CPU time for three ADIMs (Gauss, Bi-CGSTAB, and fast Bi-CGSTAB) for different meshes with $\sigma = \rho = 1/200$ at $t = 1$.

	Gauss	Bi-CGSTAB	Fast Bi-CGSTAB
$\tau = h_x = h_y$	**CPU Time**	**CPU Time**	**CPU Time**
25	0.73 s	0.75 s	0.31 s
50	2.65 s	3.03 s	1.53 s
100	11.99 s	13.84 s	9.58 s
200	89.15 s	5 min 14 s	82.63 s
400	14 min 54 s	32 min 39 s	10 min 52 s
600	1 h 6 min 4 s	3 h 7 min 15 s	1 h 1 min 12 s

7. Conclusions

In this work, we investigated the two-dimensional RLDO-SFDE by the ADIM. The stability and convergence of the numerical method were provided. An extrapolated technique was implemented to improve the convergence order of the implicit scheme. More importantly, a fast algorithm applied to the ADIM was developed to accelerate the computation. Finally, two numerical examples were provided to verify the effectiveness of the proposed numerical methods. The main highlights of this paper can be summarised as follows:

- The distributed-order operator allows for simulating anomalous diffusion processes exhibiting varying memory effects.
- The ADIM is unconditionally stable and convergent with an accuracy of $O(\sigma^2 + \rho^2 + \tau + h_x + h_y)$.
- An extrapolated scheme maintains the higher-order accuracy of the implicit numerical method.
- The fast Bi-CGSTAB method shows excellent performance in reducing computational costs.
- This research may offer more insights into the application of fractional calculus and numerical techniques for solving fractional systems.

The offered work in this paper can be extended for more advanced fractional definitions (e.g., generalised fractal-fractional Caputo derivatives [44]). This research provides more possibilities for improving the effectiveness of the standard Bi-CGSTAB method. Further investigations into exploring efficient preconditioned techniques and parallel algorithms will be a potential direction in our future research.

Author Contributions: Conceptualization, M.Z., M.S. and H.C.; methodology, M.Z. and M.S.; software, M.Z. and H.C.; validation, M.Z., M.S. and H.C.; formal analysis, M.Z.; investigation, M.Z. and M.S.; resources, M.Z.; data curation, M.Z.; writing—original draft preparation, M.Z.; writing—review and editing, M.Z., M.S. and H.C.; visualization, M.Z.; supervision, M.S. and H.C.; project administration, H.C.; funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Natural Science Foundation of Fujian Province (grant number 2021J01618).

Data Availability Statement: Not applicable.

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

References

1. Podlubny, I. *Fractional Differential Equations*; Academic Press: New York, NY, USA, 1999.
2. Chen, T.; Wang, D. Combined application of blockchain technology in fractional calculus model of supply chain financial system. *Chaos Solitons Fractals* **2020**, *131*, 109461. [CrossRef]
3. Liu, Z.; Xia, T.; Wang, T. Dynamic analysis of fractional-order six-order discrete chaotic mapping and its application in information security. *Optik* **2023**, *272*, 170356. [CrossRef]
4. Peng, W.; Chen, L.; Tianhu, H.E. A modified fractional-order thermo-viscoelastic model and its application to a polymer micro-rod heated by a moving heat source. *Appl. Math. Mech.* **2022**, *43*, 507–522. [CrossRef]
5. Yang, Q.; Chen, D.; Zhao, T.; Chen, Y. Fractional Calculus in Image Processing: A Review. *Fract. Calc. Appl. Anal.* **2016**, *19*, 1222–1249. [CrossRef]
6. Guo, L.; Zhao, X.L.; Gu, X.M.; Zhao, Y.L.; Zheng, Y.B.; Huang, T.Z. Three-dimensional fractional total variation regularized tensor optimized model for image deblurring. *Appl. Math. Comput.* **2021**, *404*, 126224. [CrossRef]
7. Sabermahani, S.; Ordokhani, Y.; Rahimkhani, P. Application of generalized Lucas wavelet method for solving nonlinear fractal-fractional optimal control problems. *Chaos Solitons Fractals* **2023**, *170*, 113348. [CrossRef]
8. Liu, X.; Gan, H.; Luo, Y.; Chen, Y.; Gao, L. Digital-Twin-Based Real-Time Optimization for a Fractional Order Controller for Industrial Robots. *Fractal Fract.* **2023**, *7*, 167. [CrossRef]
9. Kamran; Irfan, M.; Alotaibi, F.M.; Haque, S.; Mlaiki, N.; Shah, K. RBF-Based Local Meshless Method for Fractional Diffusion Equations. *Fractal Fract.* **2023**, *7*, 143. [CrossRef]
10. Zhang, M.; Liu, F.; Turner, I.W.; Anh, V.V.; Feng, L. A finite volume method for the two-dimensional time and space variable-order fractional Bloch-Torrey equation with variable coefficients on irregular domains. *Comput. Math. Appl.* **2021**, *98*, 81–98. [CrossRef]
11. Ding, W.; Patnaik, S.; Sidhardh, S.; Semperlotti, F. Applications of distributed-order fractional operators: A review. *Entropy* **2021**, *23*, 110. [CrossRef] [PubMed]
12. Jia, J.; Wang, H. A fast finite difference method for distributed-order space-fractional partial differential equations on convex domains. *Comput. Math. Appl.* **2018**, *75*, 2031–2043.
13. Zhang, A.; Ganji, R.M.; Jafari, H.; Ncube, M.N.; Agamalieva, L.; Taylor, T. Numerical Solution of Distributed-Order Integro-Differential Equations. *Fractals* **2022**, *30*, 2240123. [CrossRef]
14. Caputo, M. *Elasticitá e Dissipazione*; Zanichelli: Bologna, Italy, 1969.
15. Caputo, M. Mean fractional-order-derivatives differential equations and filters. *Annali dell'Universitá Di Ferrara* **1995**, *41*, 73–84. [CrossRef]
16. Li, Z.; Fujishiro, K.; Li, G. Uniqueness in the inversion of distributed orders in ultraslow diffusion equations. *J. Comput. Appl. Math.* **2020**, *369*, 112564. [CrossRef]
17. Ye, H.; Liu, F.; Anh, V. Compact difference scheme for distributed-order time-fractional diffusion-wave equation on bounded domains. *J. Comput. Phys.* **2015**, *298*, 652–660. [CrossRef]
18. Lorenzo, C.F.; Hartley, T.T. Variable order and distributed order fractional operators. *Nonlinear Dyn.* **2002**, *29*, 57–98. [CrossRef]
19. Caputo, M. Diffusion with space memory modelled with distributed order space fractional differential equations. *Ann. Geophys.* **2003**, *46*, 223–234. [CrossRef]
20. Pourbabaee, M.; Saadatmandi, A. A novel Legendre operational matrix for distributed order fractional differential equations. *Appl. Math. Comput.* **2019**, *361*, 215–231. [CrossRef]
21. Abdelkawy, M.A.; Babatin, M.M.; Lopes, A.M. Highly accurate technique for solving distributed-order time-fractional-sub-diffusion equations of fourth order. *Comput. Appl. Math.* **2020**, *39*, 1–22. [CrossRef]
22. Abbaszadeh, M.; Dehghan, M. Numerical investigation of reproducing kernel particle Galerkin method for solving fractional modified distributed-order anomalous sub-diffusion equation with error estimation. *Appl. Math. Comput.* **2021**, *392*, 125718. [CrossRef]
23. Chou, L.K.; Lei, S.L. Finite volume approximation with ADI scheme and low-rank solver for high dimensional spatial distributed-order fractional diffusion equations. *Comput. Math. Appl.* **2021**, *89*, 116–126. [CrossRef]
24. Bu, W.; Ji, L.; Tang, Y.; Zhou, J. Space-time finite element method for the distributed-order time fractional reaction diffusion equations. *Appl. Numer. Math.* **2020**, *152*, 446–465. [CrossRef]
25. Meerschaert, M.M.; Scheffler, H.P.; Tadjeran, C. Finite difference methods for two-dimensional fractional dispersion equation. *J. Comput. Phys.* **2006**, *211*, 249–261. [CrossRef]
26. Peaceman, D.W.; Rachford, H.H., Jr. The Numerical Solution of Parabolic and Elliptic Differential Equations. *J. Soc. Ind. Appl. Math.* **1955**, *3*, 28–41. [CrossRef]
27. Yu, Q.; Liu, F.; Turner, I.; Burrage, K. A computationally effective alternating direction method for the space and time fractional Bloch–Torrey equation in 3-D. *Appl. Math. Comput.* **2012**, *219*, 4082–4095. [CrossRef]
28. Chen, R.; Liu, F.; Anh, V. A fractional alternating-direction implicit method for a multi-term time–space fractional Bloch–Torrey equations in three dimensions. *Comput. Math. Appl.* **2019**, *78*, 1261–1273. [CrossRef]

29. Liu, F.; Zhuang, P.; Liu, Q. *Numerical Methods of Fractional Partial Differential Equations and Applications*; Science Press: Beijing, China, 2015.
30. Zlatev, Z.; Dimov, I.; Faragó, I.; Georgiev, K.; Havasi, Á.; Ostromsky, T. Application of Richardson extrapolation for multi-dimensional advection equations. *Comput. Math. Appl.* **2014**, *67*, 2279–2293.
31. Wang, Y.M. A Crank-Nicolson-type compact difference method and its extrapolation for time fractional Cattaneo convection-diffusion equations with smooth solutions. *Numer. Algorithms* **2019**, *81*, 489–527. [CrossRef]
32. Wang, H.; Wang, K. An $O(Nlog2N)$ alternating-direction finite difference method for two-dimensional fractional diffusion equations. *J. Comput. Phys.* **2011**, *230*, 7830–7839. [CrossRef]
33. Fu, H.; Ng, M.K.; Wang, H. A divide-and-conquer fast finite difference method for space–time fractional partial differential equation. *Comput. Math. Appl.* **2017**, *73*, 1233–1242.
34. Sun, L.Y.; Fang, Z.W.; Lei, S.L.; Sun, H.W.; Zhang, J.L. A fast algorithm for two-dimensional distributed-order time-space fractional diffusion equations. *Appl. Math. Comput.* **2022**, *425*, 127095. [CrossRef]
35. Diethelm, K.; Ford, N.J. Numerical analysis for distributed-order differential equations. *J. Comput. Appl. Math.* **2009**, *225*, 96–104. [CrossRef]
36. Diethelm, K.; Ford, N.J. Analysis of Fractional Differential Equations. *J. Math. Anal. Appl.* **2002**, *265*, 229–248. [CrossRef]
37. Liu, F.; Zhuang, P.; Anh, V.; Turner, I.; Burrage, K. Stability and convergence of the difference methods for the space–time fractional advection–diffusion equation. *Appl. Math. Comput.* **2007**, *191*, 12–20. [CrossRef]
38. Liu, L.; Feng, L.; Xu, Q.; Zheng, L.; Liu, F. Flow and heat transfer of generalized Maxwell fluid over a moving plate with distributed order time fractional constitutive models. *Int. Commun. Heat Mass Transf.* **2020**, *116*, 104679. [CrossRef]
39. Burden, R.L.; Faires, J.D.; Burden, A.M. *Numerical Analysis*; PWS Publishing Company: Boston, MA, USA, 1985.
40. Van der Vorst, H.A. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **1992**, *13*, 631–644. [CrossRef]
41. Davis, P. *Circulant Matrices*; Wiley-Intersciences: New York, NY, USA, 1979.
42. Gray, R.M. Toeplitz and Circulant Matrices: A Review. *Found. Trends Commun. Inf. Theory* **2006**, *2*, 155–239. [CrossRef]
43. Wang, H.; Basu, T.S. A Fast Finite Difference Method for Two-Dimensional Space-Fractional Diffusion Equations. *SIAM J. Sci. Comput.* **2012**, *34*, A2444–A2458. [CrossRef]
44. Youssri, Y.H.; Atta, A.G. Spectral Collocation Approach via Normalized Shifted Jacobi Polynomials for the Nonlinear Lane-Emden Equation with Fractal-Fractional Derivative. *Fractal Fract.* **2023**, *7*, 133. [CrossRef]