algorithms

MDPI

Article

Structure Approximation-Based Preconditioning for Solving Tempered Fractional Diffusion Equations

Xuan Zhang and Chaojie Wang *

School of Mathematics and Statistics, Beijing Technology and Business University, Beijing 100048, China; 13770615793@163.com

* Correspondence: ikeyan_btbu@163.com

Abstract: Tempered fractional diffusion equations constitute a critical class of partial differential equations with broad applications across multiple physical domains. In this paper, the Crank–Nicolson method and the tempered weighted and shifted Grünwald formula are used to discretize the tempered fractional diffusion equations. The discretized system has the structure of the sum of the identity matrix and a diagonal matrix multiplied by a symmetric positive definite (SPD) Toeplitz matrix. For the discretized system, we propose a structure approximation-based preconditioning method. The structure approximation lies in two aspects: the inverse approximation based on the row-by-row strategy and the SPD Toeplitz approximation by the τ matrix. The proposed preconditioning method can be efficiently implemented using the discrete sine transform (DST). In spectral analysis, it is found that the eigenvalues of the preconditioned coefficient matrix are clustered around 1, ensuring fast convergence of Krylov subspace methods with the new preconditioner. Numerical experiments demonstrate the effectiveness of the proposed preconditioner.

Keywords: tempered fractional diffusion equations; Krylov subspace method; structure approximation; preconditioning

check for updates

Academic Editors: Dunhui Xiao and Shuai Li

Received: 14 April 2025
Revised: 20 May 2025
Accepted: 21 May 2025
Published: 23 May 2025

Citation: Zhang, X.; Wang, C. Structure Approximation-Based Preconditioning for Solving Tempered Fractional Diffusion Equations. *Algorithms* **2025**, *18*, 307. https://doi.org/10.3390/a18060307

Copyright: © 2025 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

This paper discusses the numerical solution of variable coefficient tempered fractional diffusion equations (tempered FDEs) with initial boundary value problems

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = d(x)\left({}_a\boldsymbol{D}_x^{\boldsymbol{\beta},\boldsymbol{\lambda}} + {}_x\boldsymbol{D}_b^{\boldsymbol{\beta},\boldsymbol{\lambda}}\right)u(x,t) + f(x,t), \\ u(a,t) = 0,\ u(b,t) = 0,\ t \in [0,T], \\ u(x,0) = u_0(x),\ x \in [a,b], \end{cases} \tag{1}$$

where $f(x,t)$ is the source term, $d(x) \geq 0$ is the diffusion coefficient function, and λ is a non-negative parameter. ${}_a\boldsymbol{D}_x^{\boldsymbol{\beta},\boldsymbol{\lambda}}$ and ${}_x\boldsymbol{D}_b^{\boldsymbol{\beta},\boldsymbol{\lambda}}$ denote the left and right Riemann–Liouville tempered fractional derivatives of the function $u(x,t)$ with order $\beta(1 < \beta < 2)$, respectively. They are defined by (see Baeumer and Meerschaert [1])

$${}_a\boldsymbol{D}_x^{\boldsymbol{\beta},\boldsymbol{\lambda}}u(x) = {}_aD_x^{\beta,\lambda}u(x) - \beta\lambda^{\beta-1}\partial_x u(x) - \lambda^\beta u(x)$$

and

$${}_x\boldsymbol{D}_b^{\boldsymbol{\beta},\boldsymbol{\lambda}}u(x) = {}_xD_b^{\beta,\lambda}u(x) + \beta\lambda^{\beta-1}\partial_x u(x) - \lambda^\beta u(x)$$

where

$${}_aD_x^{\beta,\lambda}u(x) = \mathrm{e}^{-\lambda x}\,{}_aD_x^\beta\left(\mathrm{e}^{\lambda x}u(x)\right) = \frac{\mathrm{e}^{-\lambda x}}{\Gamma(2-\beta)}\frac{\partial^2}{\partial x^2}\int_a^x \frac{\mathrm{e}^{\lambda s}u(s)}{(x-s)^{\beta-1}}ds$$

and

$$ {}_xD_b^{\beta,\lambda}u(x) = \mathrm{e}^{\lambda x}\,{}_xD_b^{\beta}\left(\mathrm{e}^{-\lambda x}u(x)\right) = \frac{\mathrm{e}^{\lambda x}}{\Gamma(2-\beta)}\frac{\partial^2}{\partial x^2}\int_x^b \frac{\mathrm{e}^{-\lambda s}u(s)}{(s-x)^{\beta-1}}ds. $$

Tempered FDEs are a crucial class of equations widely applied in fields such as biology, geophysics, and finance [2–4]. These equations are characterized by incorporating tempered fractional derivatives, modifying standard fractional diffusion equations by introducing an exponential tempering factor. This modification allows tempered FDEs to better simulate processes with finite propagation speed, accurately describing actual physical phenomena. However, analytical solutions to tempered FDEs and FDEs are generally difficult to obtain, leading to extensive research on the numerical solutions of FDEs in recent years [5–11].

For tempered fractional derivatives, stability of the resulting finite difference schemes can only be ensured if the spatial step size is sufficiently small when using standard fractional derivative approximations directly [12]. A novel shifted approximation method for tempered fractional derivatives was proposed in [13], successfully developing an unconditionally stable numerical scheme for one-dimensional (1D) tempered fractional diffusion equations with constant coefficients. Additionally, a Crank–Nicolson scheme was proposed [14] for solving initial boundary value problems of a class of variable coefficient tempered fractional diffusion equations, providing the discretization method used in this paper. The inherent non-locality of tempered fractional operators manifests in the algebraic structure of discretized systems, producing coefficient matrices with dense or fully populated configurations. The system's coefficient matrix assumes the configuration $I + DT$, with I denoting the identity matrix, D a diagonal matrix, and T an SPD Toeplitz matrix.

Traditional direct solution methods, such as Gaussian elimination, require large computational cost and storage space. The Krylov subspace methods are typically used to solve these kinds of linear system. Meanwhile, effective preconditioning strategies can significantly accelerate the convergence rate of the Krylov subspace methods [15–18]. For 1D and 2D tempered FDEs, Lei, Fan and Chen [19] proposed a fast-iterative method and a fast-direct method. And they used the circulant preconditioner to accelerate the iteration process. In [20], a robust preconditioner was developed for the discretized scheme of nonlinear tempered FDEs. In [21], a circulant and skew-circulant splitting iteration method was used to solve the discretized system of tempered FDEs. The induced preconditioning form was also investigated. Nevertheless, emerging methodologies demonstrate superior efficacy compared to conventional preconditioning approaches. For 1D tempered FDEs, Tang and Huang [22] approximated the SPD Toeplitz matrix by the τ matrix [15] and proposed a scaled diagonal and Toeplitz approximate splitting (SDTAS) preconditioning method. The equations examined in this study address a notably underexplored class of tempered FDEs in the contemporary literature, characterized by discretization schemes that yield coefficient matrices incorporating SPD Toeplitz structures. This distinctive matrix configuration, possessing both theoretically significant properties and practical computational advantages, constitutes the principal focus of our numerical investigation. Also, some of the latest numerical solution methods for FDEs are described in [23,24]. In addition, it is noticed that a row-by-row inverse approximation strategy has been proposed [25] and applied [26–28]. Inspired by this strategy, we will try to construct an efficient preconditioner for the linear system arising from the numerical solution of tempered FDEs with variable coefficients.

In this paper, we use the second-order finite difference scheme to discretize the tempered FDEs. Specifically, we employ the Crank–Nicolson method for time discretization and the tempered weighted and shifted Grünwald formula for spatial discretization. We observe that the coefficient matrix of the resulting linear system is of the form $I + DT$. We approximate the SPD Toeplitz matrix T by the τ matrix and construct a novel approximate inverse preconditioner inspired by the row-by-row approximation strategy. The precon-

ditioner can be efficiently computed using discrete sine transforms, with each step of the preconditioned Krylov subspace method having a computational complexity of $O(N \log N)$. Spectral analysis establishes eigenvalue clustering near unity for the preconditioned system, which theoretically guarantees the superlinear convergence rate of Krylov subspace iterations—a property rigorously verified through our mathematical proofs. Numerical experiments are carried out to assess the performance of the developed preconditioner.

The remainder of this paper is organized as follows. Section 2 presents the derivation of the discretized system for tempered FDEs. Section 3 details the formulation of an approximate inverse preconditioner. Section 4 performs spectral analysis of the preconditioned matrices. Section 5 validates the computational efficiency of the proposed preconditioner through numerical experiments. Concluding remarks summarizing the key findings are provided in Section 6.

2. Discretization of Tempered FDEs

Let N and M be positive integers, representing the number of spatial and temporal partitions, respectively. We define the spatial step size $h = \frac{a}{N+1}$ and the time step size $\Delta t = \frac{T}{M}$. The spatial and temporal grids are given by $x_i = ih$ for $i = 0, 1, \ldots, N+1$ and $t_j = j\Delta t$ for $j = 0, 1, \ldots, M$.

We review that the tempered fractional derivatives in (1) can be approximated using the tempered weighted and shifted Grünwald formula (tempered WSGD) [14]

$$ {}_aD_x^{\beta,\lambda} u(x_i, t_j) - \lambda^\beta u(x_i, t_j) = \frac{1}{h^\beta} \sum_{k=0}^{i+1} g_k^{(\beta)} u(x_{i-k+1}, t_j) - \frac{1}{h^\beta} \rho_\beta u(x_i, t_j) + O\left(h^2\right), $$

and

$$ {}_xD_b^{\beta,\lambda} u(x_i, t_j) - \lambda^\beta u(x_i, t_j) = \frac{1}{h^\beta} \sum_{k=0}^{N-i+2} g_k^{(\beta)} u(x_{i+k-1}, t_j) - \frac{1}{h^\beta} \rho_\beta u(x_i, t_j) + O\left(h^2\right), $$

where $\rho_\beta = \left(\gamma_1 e^{h\lambda} + \gamma_2 + \gamma_3 e^{-h\lambda}\right)\left(1 - e^{-h\lambda}\right)^\beta$; the weights $g_k^{(\beta)}$ are given by

$$ g_k^{(\beta)} = \begin{cases} \gamma_1 w_0^{(\beta)} e^{h\lambda}, & k = 0, \\ \gamma_1 w_1^{(\beta)} + \gamma_2 w_0^{(\beta)}, & k = 1, \\ \left(\gamma_1 w_k^{(\beta)} + \gamma_2 w_{k-1}^{(\beta)} + \gamma_3 w_{k-2}^{(\beta)}\right) e^{-(k-1)h\lambda}, & k \geq 2, \end{cases} \tag{2} $$

where $w_0^{(\beta)} = 1, w_k^{(\beta)} = \left(1 - \frac{1+\beta}{k}\right) w_{k-1}^{(\beta)}$, for $k \geq 1$, and $\gamma_1, \gamma_2, \gamma_3$ satisfy the linear system

$$ \begin{cases} \gamma_1 + \gamma_2 + \gamma_3 = 1, \\ \gamma_1 - \gamma_3 = \beta/2. \end{cases} \tag{3} $$

Let $u_i^j \approx u(x_i, t_j)$, $f_i^{j+\frac{1}{2}} = f\left(x_i, t_{j+\frac{1}{2}}\right)$, $t_{j+\frac{1}{2}} = \frac{1}{2}(t_j + t_{j+1})$ and $u_i^{j+\frac{1}{2}} = \frac{1}{2}\left(u_i^j + u_i^{j+1}\right)$ for $i = 1, \ldots, N, j = 0, 1, \ldots, M$. At the mesh point $\left(x_i, t_{j+\frac{1}{2}}\right)$, we consider the Crank–Nicolson technique for time discretization and the tempered WSGD approximation for the tempered fractional derivatives to discretize the tempered FDEs in (1). Under the truncation-

error-free assumption [23,24], the second-order finite difference scheme is formulated as follows:

$$\begin{aligned}\frac{u_i^{j+1}-u_i^j}{\Delta t} =& \frac{d_i}{2}\left(\frac{1}{h^\beta}\sum_{k=0}^{i+1} g_k^{(\beta)} u_{i-k+1}^{j+1} - \frac{1}{h^\beta}\rho_\beta u_i^{j+1} + \frac{1}{h^\beta}\sum_{k=0}^{N-i+2} g_k^{(\beta)} u_{i+k-1}^{j+1} - \frac{1}{h^\beta}\rho_\beta u_i^{j+1}\right)\\ &+ \frac{d_i}{2}\left(\frac{1}{h^\beta}\sum_{k=0}^{i+1} g_k^{(\beta)} u_{i-k+1}^{j} - \frac{1}{h^\beta}\rho_\beta u_i^{j} + \frac{1}{h^\beta}\sum_{k=0}^{N-i+2} g_k^{(\beta)} u_{i+k-1}^{j} - \frac{1}{h^\beta}\rho_\beta u_i^{j}\right)\\ &+ f_i^{j+\frac{1}{2}}.\end{aligned} \tag{4}$$

We define the diagonal matrix as

$$D = \mathrm{diag}(d_1, d_2, \ldots, d_N)$$

and the vectors as

$$u^j = \left[u_1^j, u_2^j, \ldots, u_N^j\right]^T, f^{j+\frac{1}{2}} = \left[f_1^{j+\frac{1}{2}}, f_2^{j+\frac{1}{2}}, \ldots, f_N^{j+\frac{1}{2}}\right]^T.$$

For $j = 0, 1, \ldots, M$ and a given u^0, the differential scheme (4) admits a matrix representation expressed as

$$(I + DG)u^{j+1} = (I - DG)u^j + \Delta t f^{j+\frac{1}{2}}, \tag{5}$$

where I is the identity matrix, and G is an SPD Toeplitz matrix, defined as $G = G_\beta + G_\beta^T$, where G_β^T represents the transpose of G_β. G_β can be represented as

$$G_\beta = -\frac{\Delta t}{2h^\beta}\begin{pmatrix} g_1^{(\beta)} - \rho_\beta & g_0^{(\beta)} & 0 & \ldots & 0 & 0 \\ g_2^{(\beta)} & g_1^{(\beta)} - \rho_\beta & g_0^{(\beta)} & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ g_{N-1}^{(\beta)} & \ddots & \ddots & \ddots & & g_0^{(\beta)} \\ g_N^{(\beta)} & g_{N-1}^{(\beta)} & \cdots & \cdots & g_2^{(\beta)} & g_1^{(\beta)} - \rho_\beta \end{pmatrix}.$$

Consequently, the coefficient matrix A associated with the tempered FDEs (4) is structured as a superposition of the identity matrix and an SPD Toeplitz matrix premultiplying a diagonal matrix, analytically expressed as

$$A = I + DG. \tag{6}$$

Since the system (3) is underdetermined, the following Lemma 1 [14] establishes additional constraints on parameters $\gamma_1, \gamma_2, \gamma_3$ to ensure that the finite difference scheme (4) remains unconditionally stable while attaining second-order accuracy in both spatial and temporal domains.

Lemma 1. *Let S be the solution set of the linear system (3), if* $1 < \beta < 2$, $\lambda \geq 0$, $(\gamma_1, \gamma_2, \gamma_3) \in \mathcal{S}$, *and*

1. $\max\left\{\frac{2(\beta^2+3\beta-4)}{\beta^2+3\beta+2}, \frac{\beta^2+3\beta}{\beta^2+3\beta+4}\right\} < \gamma_1 < \frac{3(\beta^2+3\beta-2)}{2(\beta^2+3\beta+2)}$; *or*
2. $\frac{(\beta-4)(\beta^2+3\beta+2)+24}{2(\beta^2+3\beta+2)} < \gamma_2 < \min\left\{\frac{(\beta-2)(\beta^2+3\beta+4)+16}{2(\beta^2+3\beta+4)}, \frac{(\beta-6)(\beta^2+3\beta+2)+48}{2(\beta^2+3\beta+2)}\right\}$; *or*
3. $\max\left\{\frac{(2-\beta)(\beta^2+\beta-8)}{\beta^2+3\beta+2}, \frac{(1-\beta)(\beta^2+2\beta)}{2(\beta^2+3\beta+4)}\right\} < \gamma_3 < \frac{(2-\beta)(\beta^2+2\beta-3)}{2(\beta^2+3\beta+2)}$

we have

$$g_1^{(\beta)} < 0, \quad g_2^{(\beta)} + g_0^{(\beta)} > 0, \quad g_k^{(\beta)} > 0, \quad \forall k \geq 3, \quad \sum_{k=0}^{\infty} g_k^{(\beta)} = \rho_\beta \geq 0. \tag{7}$$

3. Structure Approximation-Based Preconditioning

In this section, we first construct an inverse approximation for the coefficient matrix based on its structure and the row-by-row strategy. In addition, we approximated the involved SPD Toeplitz matrix by the corresponding τ matrix, thereby introducing a structure approximation-based preconditioner to address the targeted linear system (5).

The row-by-row approximation strategy [25] takes advantage of the fact that $e_i^T A = e_i^T K_i$. Here we take the approximation as $e_i^T A^{-1} \approx e_i^T K_i^{-1}$, where e_i represents the i-th column of the identity matrix, $K_i = I + d_i G, i = 1, 2, \ldots, N$. Based on this, we obtain the approximate inverse preconditioner P_1

$$P_1^{-1} = \sum_{i=1}^{N} e_i e_i^T K_i^{-1}.$$

Recognizing that G exhibits an SPD Toeplitz structure, we employ the τ-based matrix to derive an efficient approximation of G [15]. For convenience, the first column of the G matrix can be expressed as $-\frac{\Delta t}{2h^\beta}\left[2\left(g_1^{(\beta)} - \rho_\alpha\right), g_0^{(\beta)} + g_2^{(\beta)}, g_3^{(\beta)}, \ldots, g_N^{(\beta)}\right]^T$. According to [15], we can compute the τ matrix approximation using Hankel correction

$$\tau(G) = G - HC(G), \tag{8}$$

where $HC(G)$ is a symmetric Hankel matrix, and its anti-diagonal elements can be computed from the corresponding elements in the following first column and last column:

$$-\frac{\Delta t}{2h^\beta}\left(g_3^{(\beta)}, g_4^{(\beta)}, \ldots, g_N^{(\beta)}, 0, 0\right)^T, \quad -\frac{\Delta t}{2h^\beta}\left(0, 0, g_N^{(\beta)}, \ldots, g_4^{(\beta)}, g_3^{(\beta)}\right)^T. \tag{9}$$

It is well-known that $\tau(G)$ defined in (8) admits diagonalization via the discrete sine transform:

$$\tau(G) = S\Lambda S, \quad S = \left(\sqrt{\frac{2}{N+1}} \cdot \sin\frac{\pi ij}{N+1}\right), \quad i, j = 1, 2, \ldots, N, \tag{10}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$, and S is the discrete sine transform matrix.

Clearly, $G_j = I + d_j \tau(G)$ remains a τ matrix. Using the τ matrix G_j to approximate the SPD Toeplitz matrix K_j, we obtain the τ matrix-based approximate preconditioner

$$P_2^{-1} = \sum_{i=1}^{N} e_i e_i^T G_i^{-1}. \tag{11}$$

With the preconditioner P_2^{-1}, we need to compute $O(N)$ discrete sine transforms for each iteration, which is still computationally intensive. As carried out in [25], we consider using interpolation methods to reduce computational complexity.

Define $\{x_i\}_{i=1}^N$ as the set of all discrete points in the interval $[a, b]$, and the function $q_k(x) = \frac{1}{1+\lambda_k d(x)}$, where $\lambda_k \in sp(\tau(G)) = \{\lambda_1, \lambda_2, \ldots, \lambda_N\}, k = 1, 2, \ldots, N$. Select $l (l \ll N)$

interpolation points $\{(\tilde{x}_j, q_k(\tilde{x}_j))\}_{j=1}^{l}$ in $(x_i, q_k(x_i))\}_{i=1}^{N}$, and then piecewise linear interpolation is applied to construct the interpolating function

$$\begin{aligned} p_k(x) &= \phi_1(x)q_k(\tilde{x}_1) + \phi_2(x)q_k(\tilde{x}_2) + \cdots + \phi_l(x)q_k(\tilde{x}_l) \\ &= \sum_{s=1}^{l} \phi_s(x)q_k(\tilde{x}_s). \end{aligned} \tag{12}$$

By leveraging Equation (10), each G_j is diagonalized into the form $G_j = S\Lambda_j S$ for $j = 1, 2, \ldots, N$ where S denotes the discrete sine transform matrix, and Λ_j represents a diagonal matrix containing the eigenvalues of G_j. Then, applying interpolation (12) to approximate G_j^{-1}, we have

$$G_j^{-1} \approx S\left(\sum_{s=1}^{l} \phi_s(x_j)\Lambda_s^{-1}\right)S. \tag{13}$$

Replacing G_j^{-1} in (11) with approximation values (13), we propose the approximate inverse preconditioner as

$$\begin{aligned} P_3^{-1} &= \sum_{i=1}^{N} e_i e_i^T S\left(\sum_{s=1}^{l} \phi_s(x_i)\Lambda_s^{-1}\right)S \\ &= \sum_{s=1}^{l}\left(\sum_{i=1}^{N} e_i e_i^T \phi_s(x_i)\right)S\left(\Lambda_s^{-1}\right)S \\ &= \sum_{s=1}^{l} \Phi_s S\Lambda_s^{-1} S, \end{aligned} \tag{14}$$

where $\Phi_s = \mathrm{diag}(\varphi_s(x_1), \varphi_s(x_2), \ldots, \varphi_s(x_N))$, $\Lambda_s^{-1} = \mathrm{diag}(q_1(\tilde{x}_s), q_2(\tilde{x}_s), \ldots, q_N(\tilde{x}_s))$, $s = 1, 2, \ldots, l$ are diagonal matrices. For a suitable number of interpolation points l, the computation of the preconditioner (14) requires $O(lN \log N)$ operations. We denote P_3^{-1} as $P_{TAI}^{-1}(l)$, where l represents the number of interpolation points, and have summarized the Algorithm 1 in the table below.

Algorithm 1 Solving system (5) by the PGMRES method with preconditioner $P_{TAI}^{-1}(l)$.

Input: Matrix $A = I + DG$ defined in (6) and the interpolation point l.
Output: The solution $u = [u_1, u_2, \ldots, u_m]$.

1: Calculate the eigenvalues of G by FFT and calculate the eigenvalues of $\tau(G)$ by DST;
2: Construct the preconditioner $P_{TAI}^{-1}(l)$ by diagonalizing $\tau(G)$ using Formula (14);
3: Set the initial vector u^0;
4: **For** $k = 1, 2, \ldots, m$ **do**
5: Use the right-hand term of Formula (5) to calculate b_k by FFT.
6: Calculate $P_{TAI}^{-1}(l)v$ by DST and calculate Av by FFT.
7: Obtain u_k by performing MATLAB's gmres.
8: **end for**

4. Spectral Analysis

In this section, we discuss the spectral properties of the preconditioned matrix $P_3^{-1}A$. First, we review some off-diagonal decay property reasoning.

Lemma 2 ([25]). *Let $\beta \in (1, 2)$ and $A = (a_{i,j}) \in \mathcal{L}_{\beta+1}$ be a matrix, which satisfies*

$$|a_{i,j}| \leq \frac{c}{(1 + |i - j|)^{\beta+1}}$$

for $\beta > 0$, and some constant $c > 0$. Then $\exists$ a constant ϖ s.t. $\|L\|_\infty \leq \varpi$.

Lemma 3 ([25]). *Let A and P_1 be as defined in Section 3, $d(x) \in C^1[x_L, x_R]$. For any given $\varepsilon > 0$, there exists a constant c_1 and an integer N_1, such that for $l \geq N_1$,*

$$\|P_1^{-1} - A^{-1}\|_\infty \leq c_1 \max_{1 \leq i \leq N} \Delta(x_i, l) + \varepsilon,$$

where

$$\Delta(x_i, l) = \max_{i-l \leq k \leq i+l} |x_k - x_i| = (l-1)h.$$

Next, we discuss the spectral properties of $P_3^{-1}A$ by analyzing the approximation $P_3^{-1} - A^{-1}$. Due to

$$P_3^{-1} - A^{-1} = P_3^{-1} - P_2^{-1} + P_2^{-1} - P_1^{-1} + P_1^{-1} - A^{-1},$$

we will respectively analyze the properties of $P_3^{-1} - P_2^{-1}, P_2^{-1} - P_1^{-1}$ and $P_1^{-1} - A^{-1}$. And the approximation property of P_1 and A is given in Lemma 3 above. Then, we give the results of the approximation property of P_2^{-1} and P_1^{-1}.

Theorem 1. *Let P_1 and P_2 be defined as in Section 3. The approximation P_2^{-1} to P_1^{-1} satisfies*

$$P_2^{-1} - P_1^{-1} = E_2 + F_2,$$

where E_2 and F_2 are matrices of small norm and low rank, respectively, i.e., $\|E_2\| < c_2 \cdot \epsilon$ and $rank(F_2) \leq 4\varsigma$ for some positive constant c_2, ϵ and ς.

Proof. We can write $P_2^{-1} - P_1^{-1}$ as

$$\begin{aligned} P_2^{-1} - P_1^{-1} &= \sum_{i=1}^{N} e_i e_i^T \left(G_i^{-1} - K_i^{-1} \right) \\ &= \sum_{i=1}^{N} e_i e_i^T K_i^{-1} (K_i - G_i) G_i^{-1} \\ &= \sum_{i=1}^{N} d_i e_i e_i^T K_i^{-1} (G - \tau(G)) G_i^{-1}, \end{aligned}$$

where $\|G_i^{-1}\|_\infty < \eta^{-1}$; for detailed proof, refer to Lemma 4.7 in [25].

As shown in [15], the matrix difference $G - \tau(G)$ admits a decomposition $E_1 + F_1$, where E_1 coincides with $G - \tau(G)$ within the upper-left and lower-right $(N-1) \times (N-1)$ submatrices and vanishes elsewhere, while satisfying $rank(E_1) \leq 2(N-1)$. Further, if we let $\varepsilon > 0$ be fixed, then we have $\|F_1\|_2 < \varepsilon$.

Given that K_i^{-1} exhibits off-diagonal decay, it admits a splitting $\tilde{K}_i + \hat{K}_i$, where

$$\tilde{K}_i = \begin{bmatrix} * & \cdots & * & 0 & \cdots & 0 \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ * & & \ddots & & \ddots & 0 \\ 0 & \ddots & & \ddots & & * \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ 0 & \cdots & 0 & * & \cdots & * \end{bmatrix} \quad \text{and} \quad \hat{K}_i = \begin{bmatrix} 0 & \cdots & 0 & * & \cdots & * \\ \vdots & \ddots & & \ddots & \ddots & \vdots \\ 0 & & \ddots & & \ddots & * \\ * & \ddots & & \ddots & & 0 \\ \vdots & \ddots & \ddots & & \ddots & \vdots \\ * & \cdots & * & 0 & \cdots & 0 \end{bmatrix},$$

where '*' denotes the nonzero entries.

Considering

$$\begin{aligned} K_i^{-1}(G-\tau(G))G_i^{-1} &= K_i^{-1}(E_1+F_1)G_i^{-1} \\ &= K_i^{-1}E_1G_i^{-1}+K_i^{-1}F_1G_i^{-1} \\ &= K_i^{-1}E_1G_i^{-1}+(\tilde{K}_i+\hat{K}_i)F_1G_i^{-1} \\ &= (K_i^{-1}E_1+\hat{K}_iF_1)G_i^{-1}+\tilde{K}_iF_1G_i^{-1}, \end{aligned}$$

we derive

$$\begin{aligned} \|(K_i^{-1}E_1+\hat{K}_iF_1)G_i^{-1}\|_\infty &\le (\|K_i^{-1}\|_\infty\cdot\|E_1\|_\infty+\|\hat{K}_i\|_\infty\cdot\|F_1\|_\infty)\cdot\|G_i^{-1}\|_\infty \\ &\le \varepsilon(\|K_i^{-1}\|_\infty+\|F_1\|_\infty)\cdot\|G_i^{-1}\|_\infty \end{aligned}$$

and

$$\begin{aligned} \|\sum_{i=1}^{N}e_ie_i^T(K_i^{-1}E_1+\hat{K}_iF_1)G_i^{-1}\|_\infty &= \max_{1\le i\le N}\|e_i^T(K_i^{-1}E_1+\hat{K}_iF_1)G_i^{-1}\|_1 \\ &= \max_{1\le i\le N}\|(K_i^{-1}E_1+\hat{K}_iF_1)G_i^{-1}\|_\infty \\ &\le \varepsilon\cdot\max_{1\le i\le N}(\|K_i^{-1}\|_\infty+\|F_1\|_\infty)\cdot\|G_i^{-1}\|_\infty. \end{aligned}$$

Since $\|K_i^{-1}\|_\infty$ and $\|F_1\|_\infty$ are limited due to Lemma 2, it follows that $\|G_i^{-1}\|_\infty$ is also bounded. In other words, there is a constant $c_2>0$ such that

$$\left\|\sum_{i=1}^{N}e_ie_i^T(K_i^{-1}E_1+\hat{K}_iF_1)G_i^{-1}\right\|_\infty \le c_2\cdot\epsilon.$$

Next, we focus on the matrix product $\tilde{K}_iF_1G_i^{-1}$. We assume the block dimensions of $\tilde{K}_i$ align with those of F_1; otherwise, the smaller block is extended accordingly. Through structural analysis of $\tilde{K}_i$ and F_1, the subsequent outcomes are derived:

$$\tilde{K}_iF_1G_i^{-1}=\begin{pmatrix} 0 & 0 & \dots & \dots & 0 & + \\ 0 & 0 & \dots & \dots & 0 & + \\ 0 & 0 & \dots & \dots & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ 0 & 0 & \dots & \dots & 0 & 0 \\ + & 0 & \dots & \dots & 0 & 0 \\ + & 0 & \dots & \dots & 0 & 0 \end{pmatrix},\quad G_i^{-1}=\begin{pmatrix} + & + & \dots & \dots & + & + \\ + & + & \dots & \dots & + & + \\ 0 & 0 & \dots & \dots & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ 0 & 0 & \dots & \dots & 0 & 0 \\ + & + & \dots & \dots & + & + \\ + & + & \dots & \dots & + & + \end{pmatrix},$$

where '+' denotes non-negative entries. Therefore, we have

$$\operatorname{rank}\left(\sum_{i=1}^{N}e_ie_i^T\tilde{K}_iF_1G_i^{-1}\right)\le 4\varsigma,$$

where ς is the dimension of the blocks in $\tilde{K}_iF_1G_i^{-1}$. Consequently,

$$\begin{aligned} \sum_{i=1}^{N}e_ie_i^T\left(K_i^{-1}-K_i^{-1}\right) &= \sum_{i=1}^{N}e_ie_i^T\left(K_i^{-1}E_1+\hat{K}_iF_1\right)G_i^{-1}+\sum_{i=1}^{N}e_ie_i^T\tilde{K}_iF_1G_i^{-1} \\ &= E_2+F_2, \end{aligned}$$

where E_2 and F_2 are matrices of small norm and low rank, respectively. □

Finally, we focus on the approximation property of P_3 and A.

Theorem 2. *Let P_3, P_2, P_1 and A be defined as shown previously. Supposing l is sufficiently smaller than N, there exists N_2 such that for $N > N_2$, it holds that*

$$P_3^{-1} - A^{-1} = E + F,$$

where E and F are of small norm and of low rank, respectively.

Proof. By Theorem 4.2 in [25], we know that $P_3^{-1} - P_2^{-1} = E_3 - F_3$, where E_3 is of small norm, satisfying $\|E_3\|_\infty < c_3$ and F_3 is of small rank. Due to Lemma 3, we know

$$\|P_1^{-1} - A^{-1}\|_\infty \leq \max_{1 \leq i \leq N} \Delta(x_i, N_1) + \varepsilon = c_1(N_1 - 1)h + \varepsilon.$$

Let N_2 be an integer such that $(N_1 - 1)h < \varepsilon$, we have

$$\begin{aligned} P_3^{-1} - A^{-1} &= P_3^{-1} - P_2^{-1} + P_2^{-1} - P_1^{-1} + P_1^{-1} - A^{-1} \\ &= E_3 + F_3 + E_2 + F_2 + P_1^{-1} - A^{-1} \\ &= E_3 + E_2 + (P_1^{-1} - A^{-1}) + F_3 + F_2 \\ &\triangleq E + F, \end{aligned}$$

where $E = E_3 + E_2 + (P_1^{-1} - A^{-1})$ and $F = F_3 + F_2$. As

$$\|E\|_\infty = \|E_3\|_\infty + \|E_2\|_\infty + \|P_1^{-1} - A^{-1}\|_\infty < (c_1 + c_2 + c_3 + 1) \cdot \varepsilon,$$

then the conclusion follows. □

5. Numerical Experiments

This section evaluates the computational efficiency of the proposed preconditioning strategy through numerical experiments. All computations were executed in MATLAB (R2022b) on a workstation equipped with a 4.00 GHz AMD Ryzen 9 7940H processor, 16.00 GB RAM, and the Windows 11 OS. We solve the variable-coefficient tempered FDEs:

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = d(x)\left({}_0\mathbf{D}_x^{\beta,\lambda} + {}_x\mathbf{D}_1^{\beta,\lambda}\right)u(x,t) + f(x,t), \\ u(0,t) = 0, u(1,t) = 0, t \in [0,1], \\ u(x,0) = 0, x \in [0,1]. \end{cases} \tag{15}$$

The source term is given as follows:

$$\begin{aligned} f(x,t) = e^{-\lambda x}x^3(1-x)^3 - td(x)\Bigg[e^{-\lambda x}\sum_{m=0}^{3}\left((-1)^m\binom{3}{m}\frac{\Gamma(4+m)}{\Gamma(4+m-\beta)}x^{3+m-\beta}\right) \\ +e^{\lambda(x-2)}\sum_{j=0}^{30}\frac{3^j}{j!}\left(\sum_{m=0}^{3}\left((-1)^m\binom{3}{m}\frac{\Gamma(4+m+j)}{\Gamma(4+m+j-\beta)}(1-x)^{j+3+m-\beta}\right)\right)\Bigg] \\ +2td(x)\left(\lambda^\beta\right)e^{-\lambda x}x^3(1-x)^3 \end{aligned}$$

and the exact solution is $u(x,t) \simeq te^{-\lambda x}x^3(1-x)^3$.

The resulting linear system of the discretization is solved by the GMRES method with preconditioners. We test P_3^{-1} and denote it as $P_{TAI}^{-1}(l)$, where l represents the number of interpolation points. For comparison, we also test the diagonal and circulant splitting preconditioner [17], the skew-diagonal Toeplitz splitting preconditioner based on τ ma-

trix [22], and the Strang circulant approximate inverse preconditioner [25]. These three preconditioner are denoted as P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$ and $P_{CAI}^{-1}(l)$, respectively.

In all experiments, $\lambda = 1.5$, $\beta = 1.2$. We searched for the optimal number of interpolation points l in the proposed preconditioning matrices $P_{TAI}^{-1}(l)$. For the two experiments, we choose the optimal number of interpolation points l to be 8 and 12, respectively. We specify the same grid density in both space and time $\Delta t = h$, and for a certain initial point u^0. The iterative procedure is initialized with the zero vector and halts once the relative residual error satisfies $\|r_k\|/\|r_0\| < 10^{-7}$, with a cap of 1000 iterations imposed to prevent divergence.

5.1. Test Problem 1

Consider the coefficient

$$d(x) = \frac{e^{5x}}{1+x}.$$

An illustration of the corresponding analytical solution and the numerical solution is shown in Figure 1. We use GMRES with P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ preconditioners as well as GMRES without preconditioning to test. The results of the computations at discretization levels $N = 10^8, 10^9, 10^{10}, 10^{11}, 10^{12}$ are shown in Table 1. Here, "IT" denotes the average number of iterations per step, and "CPU" denotes the total computation time(s) of the algorithm. To facilitate the comparison of the performance of different preconditioning matrices, we visualized the results from Table 1, as shown in Figure 2. From Table 1 and Figure 2, we can see that the performance of $P_{TAI}^{-1}(l)$ is significantly better than that of $P_{CAI}^{-1}(l)$, P_{DCS}^{-1} and $P_{SDTAS_\tau}^{-1}$. The number of iterations for each preconditioning method remains relatively stable as the discretization level increases, but the number of iterations for $P_{TAI}^{-1}(l)$ is notably lower than that for the other three. Similarly, in terms of computation time, the computation time for $P_{TAI}^{-1}(l)$ is much lower than that for the other three, and the growth in computation time for $P_{TAI}^{-1}(l)$ is slower as the discretization level increases.

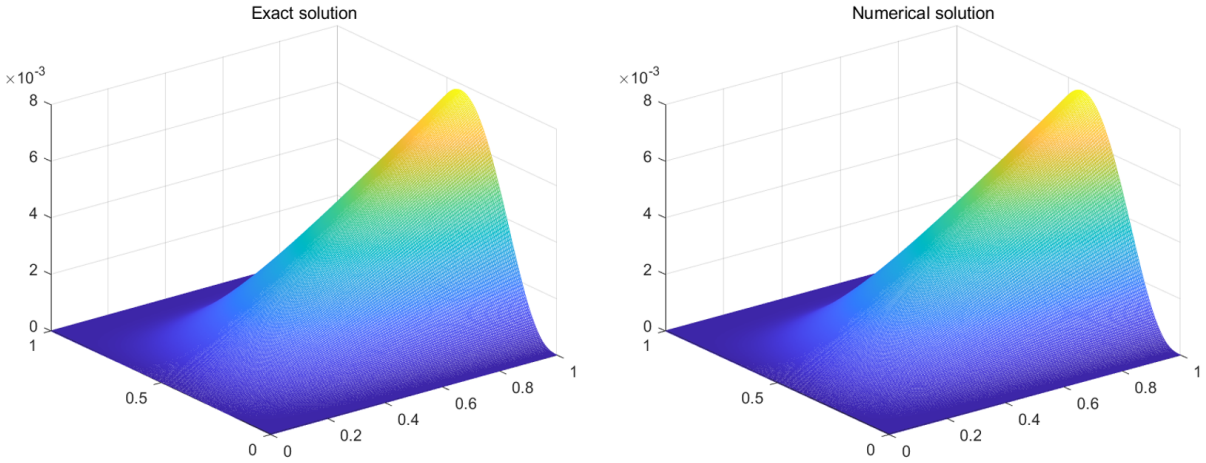


Figure 1. Comparison of the exact solutions and $P_{TAI}^{-1}(10)$ numerical solutions ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

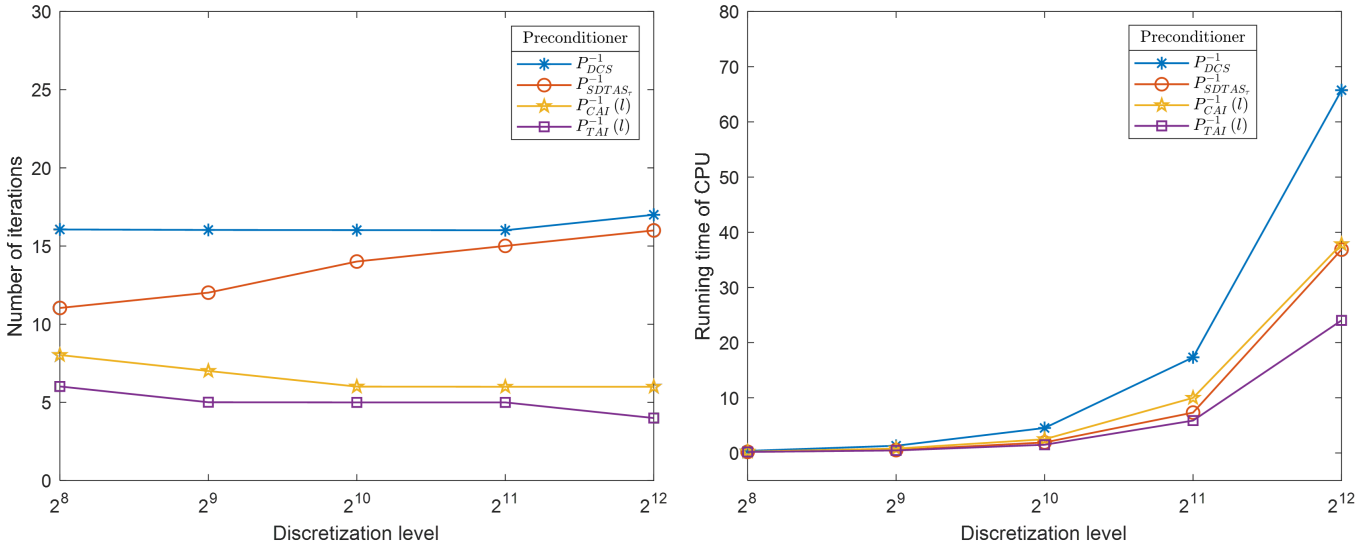


Figure 2. Comparison of average iterations and computation time for P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

Table 1. Numerical results of I, P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

Method	**Index**	**N**				
		$\mathbf{2^8}$	$\mathbf{2^9}$	$\mathbf{2^{10}}$	$\mathbf{2^{11}}$	$\mathbf{2^{12}}$
I	IT	87.34	108.21	123.12	134.07	140.03
	CPU	0.82	3.66	14.33	70.64	555.72
P_{DCS}^{-1} [17]	α	0.1	0.1	0.1	0.1	0.1
	IT	16.06	16.03	16.02	16.01	17.00
	CPU	0.38	1.31	4.53	17.33	65.74
$P_{SDTAS_\tau}^{-1}$ [22]	α	0.1	0.1	0.1	0.1	0.1
	IT	11.04	12.02	14.01	15.01	16.00
	CPU	0.20	0.52	1.91	7.32	36.86
$P_{CAI}^{-1}(l)$ [25]	l	8	8	8	8	8
	IT	8.03	7.01	6.01	6.00	6.00
	CPU	0.25	0.79	2.49	9.99	37.84
$\boldsymbol{P_{TAI}^{-1}(l)}$	l	8	8	8	8	8
	IT	**6.02**	**5.01**	**5.00**	**5.00**	**4.00**
	CPU	**0.19**	**0.46**	**1.49**	**5.87**	**24.03**

The proposed preconditioner and optimal result have been bolded.

To further illustrate the effectiveness of the proposed preconditioning matrices, we plot the eigenvalue distributions of the original coefficient matrix A, the preconditioner P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ when $N = 10^8$, as shown in Figure 3. It can be observed that the eigenvalue distribution of the coefficient matrix A without preconditioning is highly dispersed, while the eigenvalue distributions of the preconditioned matrices $AP_{TAI}^{-1}(l)$ are concentrated around 1. Notably, the eigenvalue distribution of $AP_{TAI}^{-1}(l)$ is more concentrated than those of the other preconditioners.

We searched for the optimal number of interpolation points l in the proposed preconditioner $P_{TAI}^{-1}(l)$. As shown in Table 2, we tested GMRES using $P_{TAI}^{-1}(l)$ with the number of interpolation points $l = 4, 6, 8, 10, 12$. To determine the optimal number of interpolation points l, we visualized the number of iterations, as shown in Figure 4. By analyzing Figure 4, we found that when the number of interpolation points $l \geq 8$, the number of iterations tends to stabilize. With this observation, the number of interpolation points $l = 8$ was recommended for test problem 1.

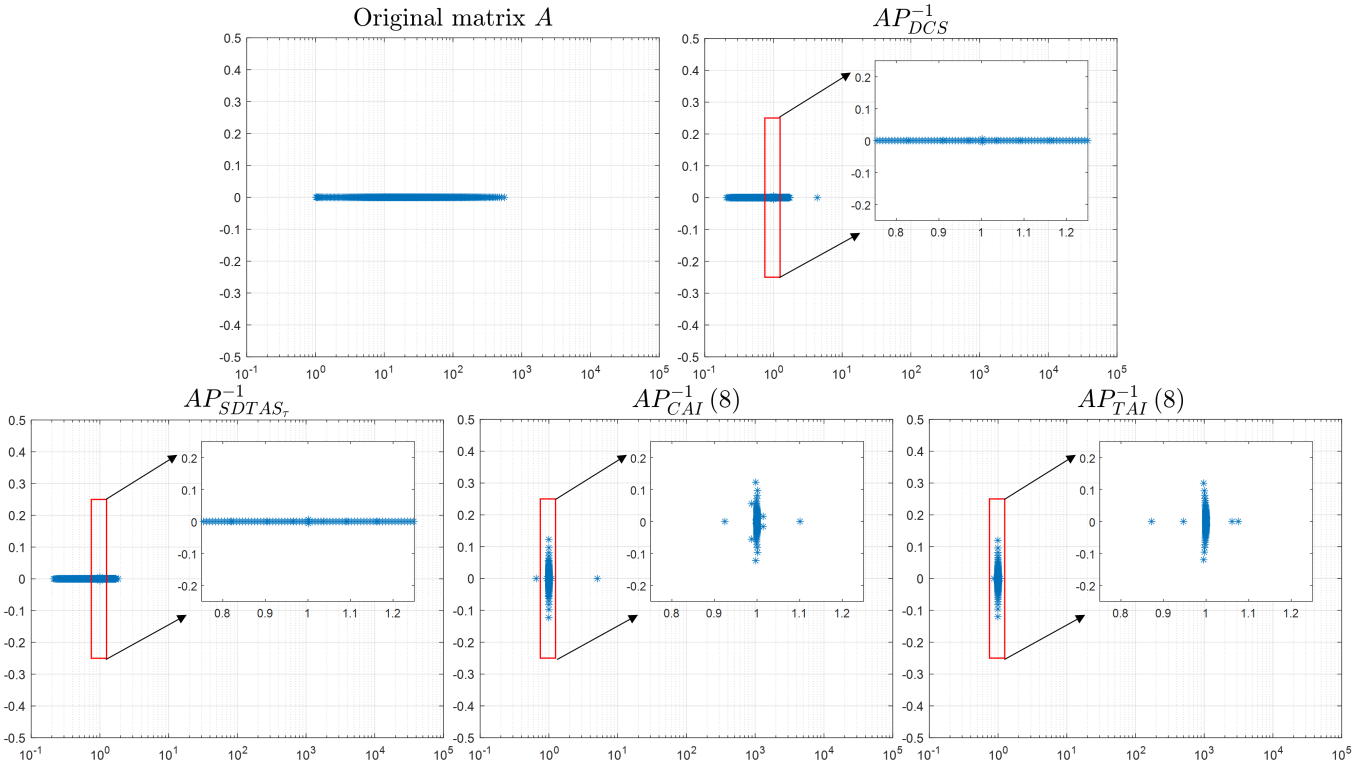


Figure 3. The eigenvalue distributions of the original coefficient matrix A, the preconditioner P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

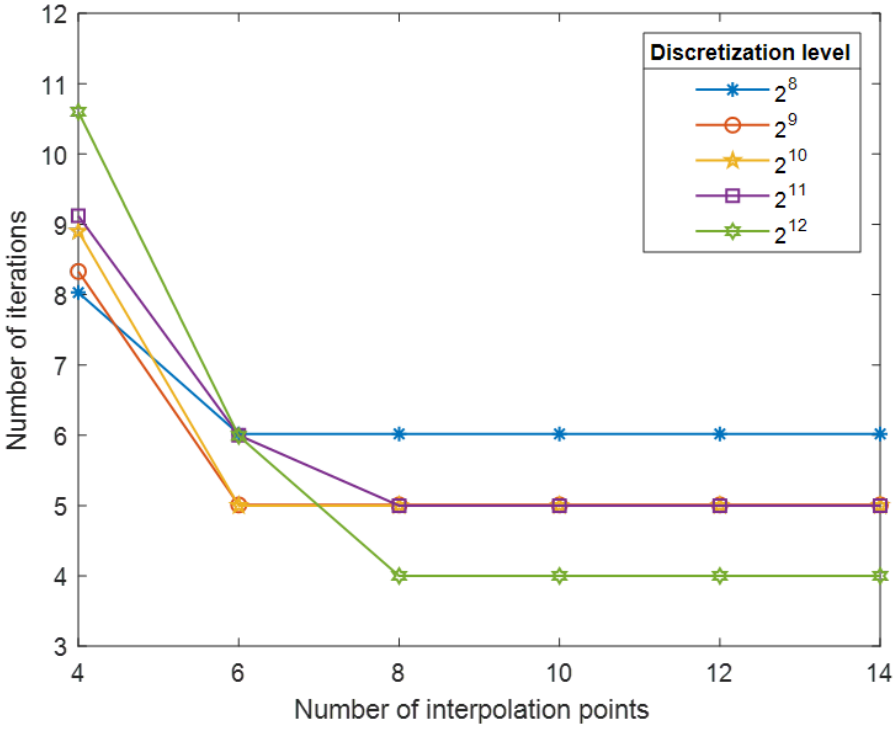


Figure 4. Number of iteration variation curves when $l = 4, 6, 8, 10, 12$ ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

Table 2. Numerical results of $P_{TAI}^{-1}(l)$ at different numbers of interpolation points l ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

N	Index	Number of Interpolation Points l					
		4	**6**	**8**	**10**	**12**	**14**
2^8	IT	8.03	6.02	6.02	6.02	6.02	6.02
	CPU	0.17	0.16	0.18	0.22	0.25	0.24
2^9	IT	8.33	5.01	5.01	5.01	5.01	5.01
	CPU	0.49	0.39	0.46	0.47	0.49	0.54
2^{10}	IT	8.90	5.00	5.00	5.00	5.00	5.00
	CPU	1.74	1.45	1.62	1.82	1.80	1.95
2^{11}	IT	9.12	6.00	5.00	5.00	5.00	5.00
	CPU	8.89	6.20	5.70	6.30	6.51	6.79
2^{12}	IT	10.60	5.99	4.00	4.00	4.00	4.00
	CPU	60.43	41.71	42.19	44.64	40.37	50.46

5.2. Test Problem 2

Consider the coefficient

$$d(x) = \frac{e^{3x} + 0.2}{x(1-x)}.$$

An illustration of the corresponding analytical solution and the numerical solution are shown in Figure 5. We compared P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$. The results of the computations at discretization levels $N = 10^8, 10^9, 10^{10}, 10^{11}, 10^{12}$ are shown in Table 3 and visualized in Figure 6. In terms of the number of iterations, all preconditioning methods show relatively stable iteration numbers as the discretization level increases, but the number of iterations for $P_{TAI}^{-1}(l)$ is notably lower than that for the other three. In terms of computation time, the computation time for $P_{TAI}^{-1}(l)$ is slightly higher than that for $P_{SDTAS_\tau}^{-1}$ at discretization level $N = 10^8, 10^9, 10^{10}$, but the difference is not significant. As the discretization level increases, the computation time for $P_{TAI}^{-1}(l)$ grows more slowly and is significantly lower than that for the other three preconditioners. Additionally, we note that this coefficient is more complex than the last one as it has singularities at point $x = 0$ and $x = 1$. And this is reflected in the higher number of iterations in Table 3 and the more dispersed eigenvalue distribution in Figure 7.

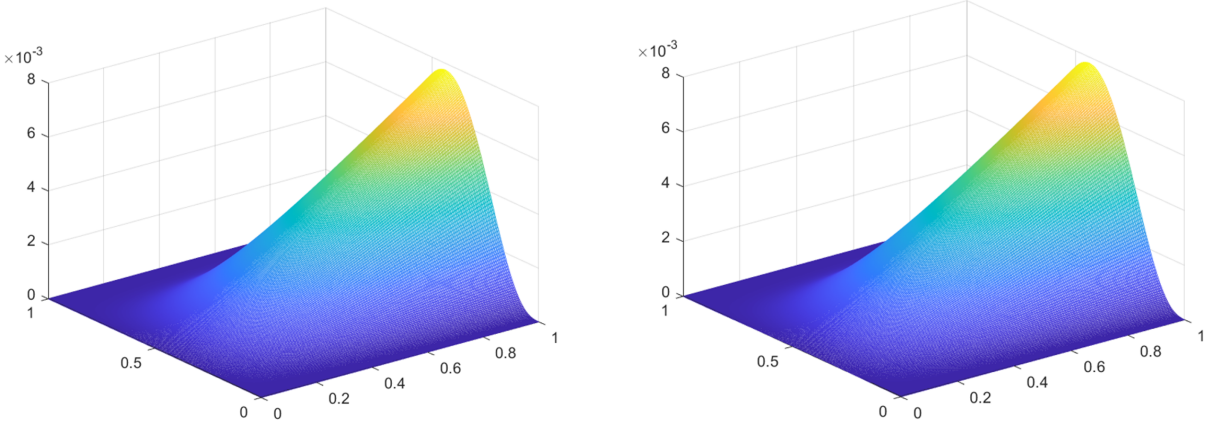


Figure 5. Comparison of the exact solutions and $P_{TAI}^{-1}(l)$ numerical solutions ($d = d_2(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

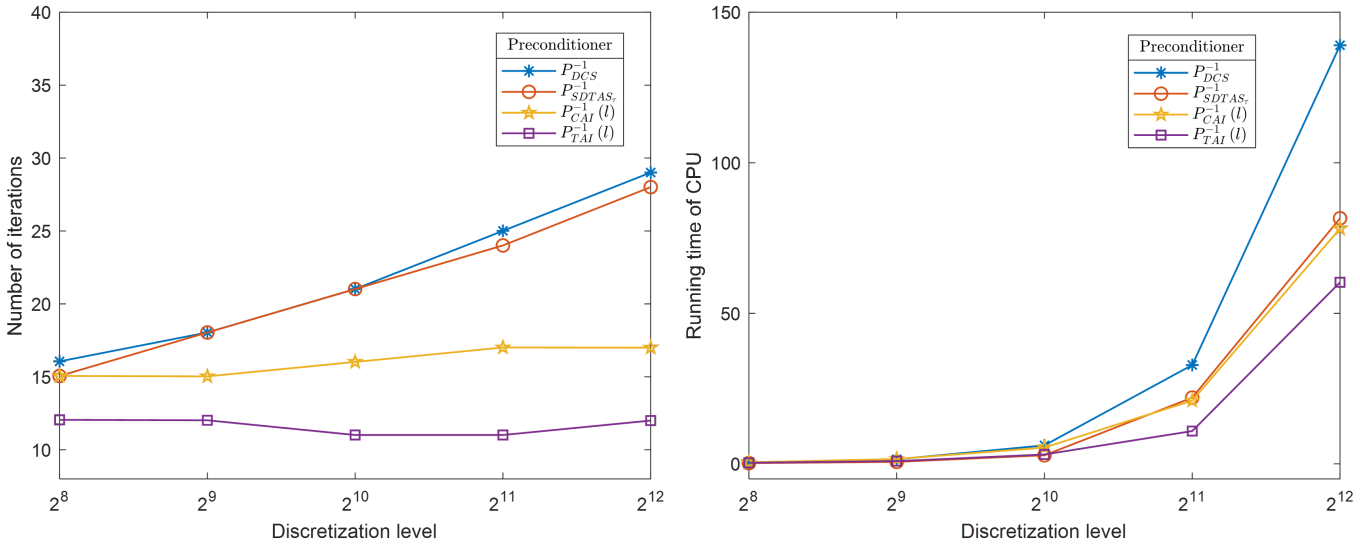


Figure 6. Comparison of average iterations and computation time for P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ ($d = d_1(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

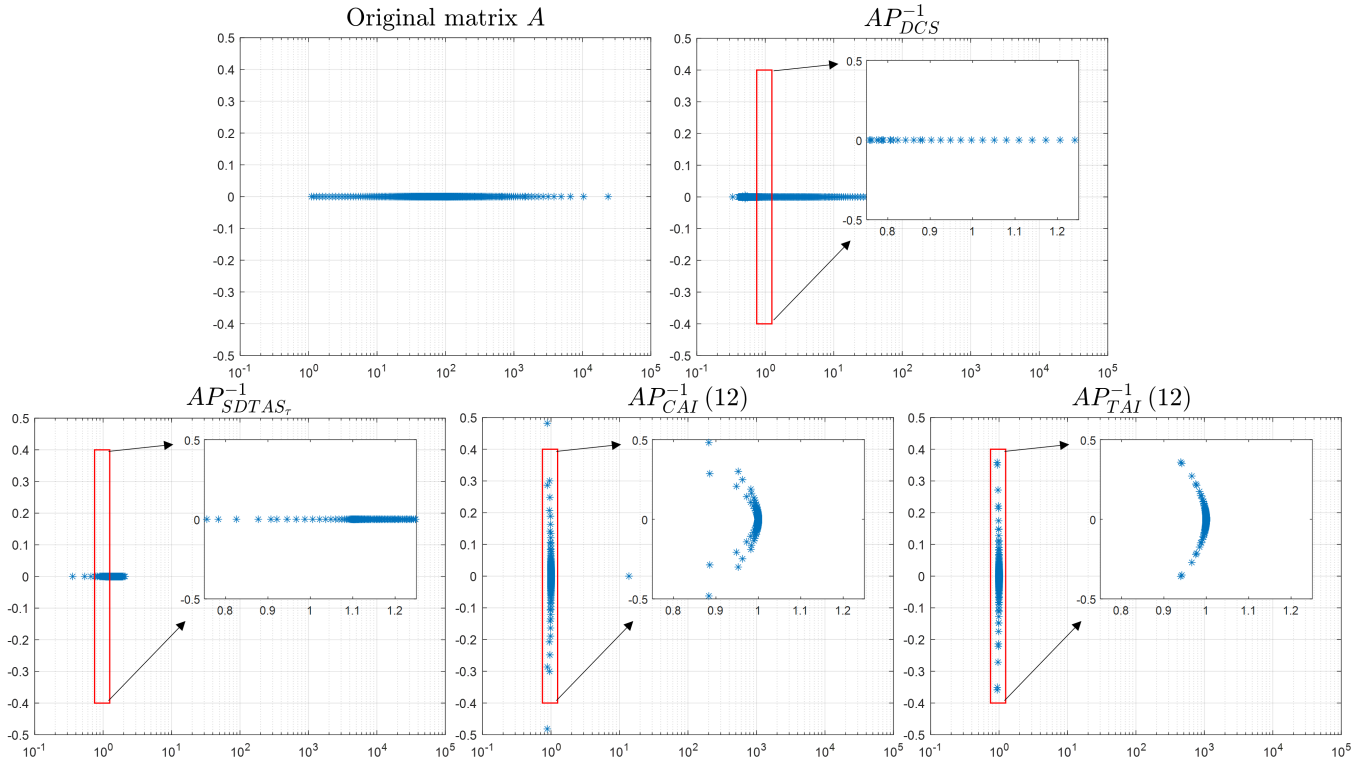


Figure 7. The eigenvalue distributions of the original coefficient matrix A, the preconditioner P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ ($d = d_2(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

We also plot the eigenvalue distributions of A, the preconditioner P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ when $N = 10^8$, as shown in Figure 7. It can be observed that the eigenvalue distribution of the coefficient matrix without preconditioning is highly dispersed, while the eigenvalue distributions of the preconditioned matrices are concentrated around 1. Notably, the eigenvalue distribution of $P_{TAI}^{-1}(l)$ is more concentrated than those of the other preconditioners.

Similarly, we searched for the optimal number of interpolation points l in the proposed preconditioner $P_{TAI}^{-1}(l)$. As shown in Table 4, we tested GMRES using $P_{TAI}^{-1}(l)$ with the number of interpolation points $l = 4, 6, 8, 10, 12$. To determine the optimal number of

interpolation points l, we visualized the number of iterations, as shown in Figure 8. By analyzing Figure 8, we found that when the number of interpolation points $l \geq 12$, the number of iterations tends to stabilize. In this view, we selected the optimal number of interpolation points $l = 12$ for the above test.

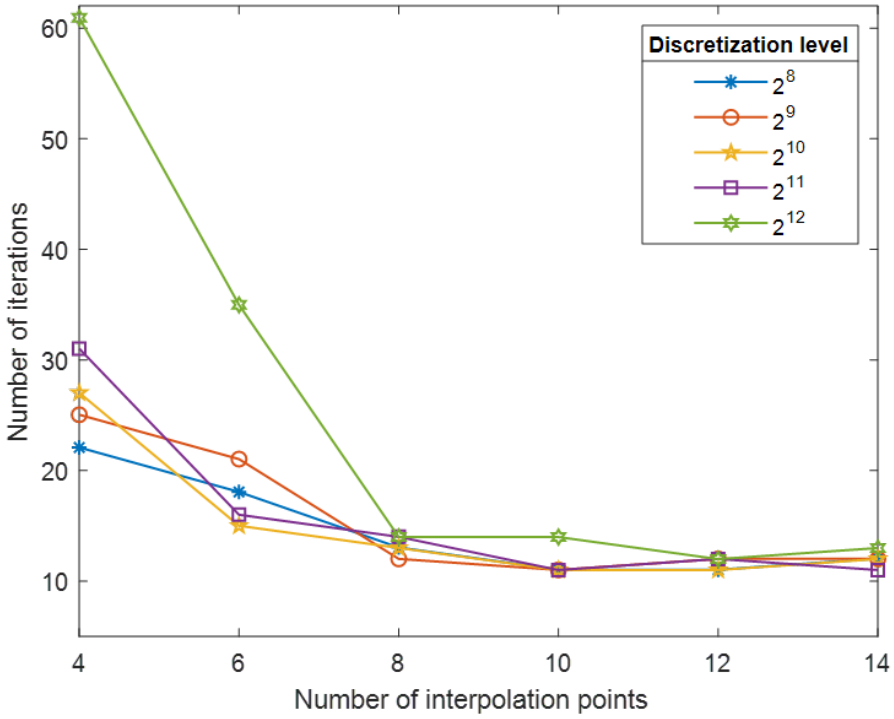


Figure 8. Number of iteration variation curves when $l = 4, 6, 8, 10, 12$ ($d = d_2(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

Table 3. Numerical results of I, P_{DCS}^{-1}, $P_{SDTAS_\tau}^{-1}$, $P_{CAI}^{-1}(l)$ and $P_{TAI}^{-1}(l)$ ($d = d_2(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

Method	**Index**	**N**				
		$\mathbf{2^8}$	$\mathbf{2^9}$	$\mathbf{2^{10}}$	$\mathbf{2^{11}}$	$\mathbf{2^{12}}$
I	IT	125.49	174.34	233.23	300.15	$\sim$
	CPU	1.46	7.41	46.45	225.74	$\sim$
P_{DCS}^{-1} [17]	α	0.15	0.15	0.15	0.15	0.15
	IT	16.06	18.04	21.02	25.01	29.01
	CPU	0.48	1.52	6.14	32.83	139.03
$P_{SDTAS_\tau}^{-1}$ [22]	α	0.15	0.15	0.15	0.15	0.15
	IT	15.06	18.04	21.02	24.01	28.01
	CPU	0.24	0.65	2.84	22.00	81.56
$P_{CAI}^{-1}(l)$ [25]	l	12	12	12	12	12
	IT	15.06	15.03	16.02	17.01	17.00
	CPU	0.49	1.60	5.42	21.00	78.15
$\boldsymbol{P_{TAI}^{-1}(l)}$	l	12	12	12	12	12
	IT	**12.05**	**12.02**	**11.01**	**11.01**	**12.00**
	CPU	**0.35**	**0.90**	**3.11**	**10.89**	**60.30**

The proposed preconditioner and optimal result have been bolded.

Table 4. Numerical results of $P_{TAI}^{-1}(l)$ at different numbers of interpolation points l ($d = d_2(x)$, $\lambda = 1.5$, $\beta = 1.2$, $\gamma_1 = 0.75$).

N	Index	Number of Interpolation Points l					
		4	**6**	**8**	**10**	**12**	**14**
2^8	IT	22.09	18.07	13.05	11.04	11.04	12.05
	CPU	0.41	0.40	0.35	0.32	0.36	0.48
2^9	IT	25.05	21.04	12.02	11.02	12.02	12.02
	CPU	1.27	1.29	0.92	0.92	1.00	1.00
2^{10}	IT	27.03	15.01	13.01	11.01	11.01	12.01
	CPU	10.22	3.16	3.10	3.04	3.26	3.53
2^{11}	IT	31.02	16.01	14.01	11.01	12.01	11.01
	CPU	20.98	15.90	12.04	11.23	12.51	11.79
2^{12}	IT	60.93	34.98	14.00	13.99	12.00	13.00
	CPU	626.27	259.42	121.80	112.15	76.09	109.79

6. Concluding Remarks

In this paper, we consider solving the tempered fractional diffusion equations with variable coefficients in a numerical way. The problem is discretized by the Crank–Nicolson method and the tempered weighted and shifted Grünwald formula. For the resulting linear system, we approximate the SPD Toeplitz matrix by the τ matrix and take the advantage of the row-by-row approximation strategy to construct an approximate inverse preconditioner. A spectral analysis reveals that the eigenvalues of the preconditioned system matrix are tightly clustered near unity. Experiments further validate the enhanced convergence enabled by the developed preconditioner.

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

Funding: This work was supported by the National Natural Science Foundation of China (No. 12001022) and the Disciplinary funding of Beijing Technology and Business University (No. STKY202508).

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no competing interests.

References

1. Baeumer, B.; Meerschaert, M.M. Tempered stable Lévy motion and transient super-diffusion. *J. Comput. Appl. Math.* **2010**, *233*, 2438–2448. [CrossRef]
2. Magin, R.L. Fractional calculus in bioengineering. *Crit. Rev. Biomed. Eng.* **2004**, *32*, 1–104. [CrossRef]
3. Meerschaert, M.M.; Zhang, Y.; Baeumer, B. Tempered anomalous diffusion in heterogeneous systems. *Geophys. Res. Lett.* **2008**, *35*, L17403. [CrossRef]
4. Zhang, H.; Liu, F.; Turner, I.; Chen, S. The numerical simulation of the tempered fractional Black-Scholes equation for European double barrier option. *Appl. Math. Model.* **2016**, *40*, 5819–5834. [CrossRef]
5. Huang, X.; Sun, H.-W. A preconditioner based on sine transform for two-dimensional semi-linear Riesz space fractional diffusion equations in convex domains. *Appl. Numer. Math.* **2021**, *169*, 289–302. [CrossRef]
6. Qin, H.; Pang, H.; Sun, H. Sine transform based preconditioning techniques for space fractional diffusion equations. *Numer. Linear Algebra Appl.* **2023**, *30*, e2474. [CrossRef]
7. Sun, L.-Y.; Lei, S.-L.; Sun, H.-W.; Zhang, J.-L. An α-robust fast algorithm for distributed-order time–space fractional diffusion equation with weakly singular solution. *Math. Comput. Simul.* **2023**, *207*, 437–452. [CrossRef]

8. Zhang, C.-H.; Yu, J.-W.; Wang, X. A fast second-order scheme for nonlinear Riesz space-fractional diffusion equations. *Numer. Algorithms* **2023**, *92*, 1813–1836. [CrossRef]
9. Liu, J.; Fu, H.; Wang, H.; Chai, X. A preconditioned fast quadratic spline collocation method for two-sided space-fractional partial differential equations. *J. Comput. Appl. Math.* **2019**, *360*, 138–156. [CrossRef]
10. Lu, X.; Fang, Z.-W.; Sun, H.-W. Splitting preconditioning based on sine transform for time-dependent Riesz space fractional diffusion equations. *J. Appl. Math. Comput.* **2021**, *66*, 673–700. [CrossRef]
11. Shao, X.-H.; Kang, C.-B. A preconditioner based on sine transform for space fractional diffusion equations. *Appl. Numer. Math.* **2022**, *178*, 248–261. [CrossRef]
12. Qu, W.; Liang, Y. Stability and convergence of the Crank-Nicolson scheme for a class of variable-coefficient tempered fractional diffusion equations. *Adv. Differ. Equ.* **2017**, *2017*, 108. [CrossRef]
13. Wang, W.; Chen, X.; Ding, D.; Lei, S.-L. Circulant preconditioning technique for barrier options pricing under fractional diffusion models. *Int. J. Comput. Math.* **2015**, *92*, 2596–2614. [CrossRef]
14. Li, C.; Deng, W. High order schemes for the tempered fractional diffusion equations. *Adv. Comput. Math.* **2016**, *42*, 543–572. [CrossRef]
15. Serra, S. Superlinear PCG methods for symmetric Toeplitz systems. *Math. Comput.* **1999**, *68*, 793–803. [CrossRef]
16. Wang, C.; Li, H.; Zhao, D. Preconditioning Toeplitz-plus-diagonal linear systems using the Sherman-Morrison-Woodbury formula. *J. Comput. Appl. Math.* **2017**, *309*, 312–319. [CrossRef]
17. Bai, Z.-Z.; Lu, K.-Y.; Pan, J.-Y. Diagonal and Toeplitz splitting iteration methods for diagonal-plus-Toeplitz linear systems from spatial fractional diffusion equations. *Numer. Linear Algebra Appl.* **2017**, *24*, e2093. [CrossRef]
18. Wang, C.; Li, H.; Zhao, D. Improved block preconditioners for linear systems arising from half-quadratic image restoration. *Appl. Math. Comput.* **2019**, *363*, e124614. [CrossRef]
19. Lei, S.; Fan, D.; Chen, X. Fast solution algorithms for exponentially tempered fractional diffusion equations. *Numer. Methods Partial Differ. Equ.* **2018**, *34*, 1301–1323. [CrossRef]
20. Zhao, Y.-L.; Zhu, P.-Y.; Gu, X.-M.; Zhao, X.-L.; Jian, H.-Y.A preconditioning technique for all-at-once system from the nonlinear tempered fractional diffusion equation. *J. Sci. Comput.* **2020**, *83*, 10. [CrossRef]
21. Qu, W.; Lei, S.-L. On CSCS-based iteration method for tempered fractional diffusion equations. *Jpn. J. Ind. Appl. Math.* **2016**, *33*, 583–597. [CrossRef]
22. Tang, S.-P.; Huang, Y.-M. A matrix splitting preconditioning method for solving the discretized tempered fractional diffusion equations. *Numer. Algorithms* **2023**, *92*, 1311–1333. [CrossRef]
23. Feng, Y.; Zhang, X.; Chen, Y.; Wei, L. A compact finite difference scheme for solving fractional Black-Scholes option pricing model. *J. Inequal. Appl.* **2025**, *2025*, 36. [CrossRef]
24. Zhang, X.; Wei, L.; Liu, J. Application of the LDG method using generalized alternating numerical flux to the fourth-order time-fractional sub-diffusion model. *Appl. Math. Lett.* **2025**, *168*, 109580. [CrossRef]
25. Pan, J.; Ke, R.; Ng, M.K.; Sun, H.-W. Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations. *SIAM J. Sci. Comput.* **2014**, *36*, A2698–A2719. [CrossRef]
26. Chou, L.-K.; Lei, S.-L. Fast ADI method for high dimensional fractional diffusion equations in conservative form with preconditioned strategy. *Comput. Math. Appl.* **2017**, *73*, 385–403. [CrossRef]
27. Du, N.; Sun, H.-W.; Wang, H. A preconditioned fast finite difference scheme for space-fractional diffusion equations in convex domains. *Comput. Appl. Math.* **2019**, *38*, 14. [CrossRef]
28. Gan, D.; Zhang, G.-F. Efficient ADI schemes and preconditioning for a class of high-dimensional spatial fractional diffusion equations with variable diffusion coefficients. *J. Comput. Appl. Math.* **2023**, *423*, 114938. [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.