A fast implicit difference scheme for solving the generalized time–space fractional diffusion equations with variable coefficients

Xian-Ming Gu1,2, Ting-Zhu Huang3, Yong-Liang Zhao3, Pin Lyu1, Bruno Carpentieri4

1School of Economic Mathematics/Institute of Mathematics, Southwestern University of Finance and Economics, Chengdu, Sichuan, China
2Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen, AK Groningen, The Netherlands
3School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, China
4Facoltà di Scienze e Tecnologie informatiche, Libera Università di Bolzano, Italy, Bozen-Bolzano

Correspondence
Ting-Zhu Huang, School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China.
Email: tingzhuhuang@126.com

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 11801463, 61772003. Applied Basic Research Project of Sichuan Province, Grant/Award Number: 2020YJ0007. Fundamental Research Funds for the Central Universities, Grant/Award Number: JBK1902028. Ministry of Education of Humanities and Social Science Layout Project, Grant/Award Number: 19JYA790094.

Abstract
In this article, we first propose an unconditionally stable implicit difference scheme for solving generalized time–space fractional diffusion equations (GTSFDEs) with variable coefficients. The numerical scheme utilizes the $L_1$-type formula for the generalized Caputo fractional derivative in time discretization and the second-order weighted and shifted Grünwald difference (WSGD) formula in spatial discretization, respectively. Theoretical results and numerical tests are conducted to verify the $(2-\gamma)$-order and 2-order of temporal and spatial convergence with $\gamma \in (0, 1)$ the order of Caputo fractional derivative, respectively. The fast sum-of-exponential approximation of the generalized Caputo fractional derivative and Toeplitz-like coefficient matrices are also developed to accelerate the proposed implicit difference scheme. Numerical experiments show the effectiveness of the proposed numerical scheme and its good potential for large-scale simulation of GTSFDEs.

KEYWORDS
fast Fourier transform, generalized Caputo fractional derivative, GTSFDEs, implicit difference scheme, Krylov subspace method, WSGD
1 | INTRODUCTION

In recent years, there has been a growing interest in the field of fractional calculus. For instance, Podlubny [43], Samko et al. [47], and Kilbas et al. [24] provide the history and a comprehensive treatment of this subject. Many phenomena in engineering, physics, chemistry and other sciences can be described very successfully by using fractional partial differential equations (FPDEs). Diffusion with an additional velocity field and diffusion under the influence of a constant external force field are, in the Brownian case, both modeled by the diffusion equation. In the case of anomalous diffusion this is no longer true, that is, the space fractional generalization may be different for the transport in external force field [38]. Under the framework of the continuous time random walks (CTRWs) model, the fractional diffusion, Fokker–Planck and Feynman–Kac equations [38, 56] can be derived with power-law waiting time distribution (WTD), assuming the particles may exhibit long waiting time. However, for some practical physical processes, it is necessary to make the first moment of the waiting time finite. This leads to the generalized time fractional diffusion equation corresponding to the CTRWs model with some more complicated WTDs (beyond the power-law limit) [1, 48, 49], for example, the tempered [2, 15, 16, 46, 59] and the scale-weight [12, 57] power law WTDs. In one word, the generalization of time–space fractional diffusion equations where the subdiffusion in time and the power-law waiting time distribution (WTD), assuming the particles may exhibit long waiting time.

Based on the above considerations, in this work, we are interested in developing fast numerical methods for solving the initial-boundary value problem of the generalized time–space fractional diffusion equation (GTSFDE) with variable coefficients

\[
\begin{align*}
\left\{ \begin{array}{ll}
C_0 D_t^{\alpha(t)} u(x, t) &= \xi(x, t)[p_{xL} D_x^{\gamma(t)} u(x, t) + (1-p_{x}) D_x^{\gamma(t)} u(x, t)] + f(x, t), & (x, t) \in (x_L, x_R) \times (0, T), \\
u(x, 0) &= \phi(x), & x \in [x_L, x_R], \\
u(x_L, t) &= \varphi(t), & x_L \in (0, T), \\
u(x_R, t) &= \psi(t), & x_R \in (0, T),
\end{array} \right.
\end{align*}
\]

where \( \alpha \in (1, 2) \) and \( \gamma \in (0, 1) \). The parameter \( p \in [0, 1] \), called “skewness,” represents the proportion of high-velocity “jets” in the direction of flow and also indicates the relative weight of forward versus backward transition probability [37, 58]. The function \( u(x, t) \) can be interpreted as representing the concentration of a particle plume undergoing anomalous diffusion. The diffusion coefficient \( \xi(x, t) \) depending on both time and space variables satisfies the condition \( 0 < \xi_{\min} \leq \xi(x, t) < \xi_{\max} < +\infty \), \( \forall (x, t) \in [x_L, x_R] \times [0, T] \), and the forcing function \( f(x, t) \) represents the source or sink term. In the current study, we assume that the problem (1.1) has a unique and (sufficiently) smooth solution unless otherwise specified [1, 4, 39, 48].

The GTSFDE (1.1) can be regarded as a generalization of classical diffusion equations where the first-order time derivative is replaced by the generalized Caputo fractional derivative of order \( \gamma \in (0, 1] \) with weighting function \( \lambda(t) > 0 \) for \( t \in [0, T] \), and the second-order spatial derivative is replaced by the two-sided Riemann–Liouville (R-L) fractional derivative of order \( \alpha \in (1, 2] \). Specifically, the time fractional derivative in Equation (1.1) is the generalized Caputo fractional derivative of order \( \gamma \) [1] denoted by

\[
C_0 D_t^{\lambda(t)} u(x, t) = \frac{1}{\Gamma(1-\gamma)} \int_0^t \lambda(t-\eta) \frac{\partial u(x, \eta)}{\partial \eta} d\eta,
\]

which collapses to the widely recognized Caputo or Caputo-tempered fractional derivatives when \( \lambda(t) \equiv 1 \) or \( \lambda(t) = e^{-bt} \) \((b > 0) \) [2, 56], respectively. It implies that the weighting function is indeed often chosen as \( \lambda(t) > 0 \) (even with the certain monotonicity) in real-world applications. Meanwhile, the left-handed \( (\_L D_x^{\gamma}) \) and the right-handed \( (\_R D_x^{\gamma}) \) space fractional derivatives in Equation (1.1) are
the R-L fractional derivatives of order \( \alpha \) [43] which are defined as

\[
x_L D^\alpha_x u(x, t) = \frac{1}{\Gamma(2 - \alpha)} \frac{\partial^2}{\partial x^2} \int_x^x u(\xi, t) d\xi, \quad \text{and} \quad x_R D^\alpha_x u(x, t) = \frac{1}{\Gamma(2 - \alpha)} \frac{\partial^2}{\partial x^2} \int_x^x u(\xi, t) d\xi \frac{1}{(\xi - x)^{\alpha - 1}},
\]

where \( \Gamma(\cdot) \) denotes the Gamma function. Note that the above equation reduces to the classical diffusion equation for \( \gamma = \lambda(t) \equiv 1 \) and \( \alpha = 2 \).

Generally speaking, although the (semi-)analytical (or closed-form) solutions of particular (generalized) space–time FPDEs on the entire real line are accessible via the Laplace or Fourier transforms, yet these solutions are expressed in terms of special functions which are usually difficult for the numerical evaluation in practice. Moreover, if we define the problem (1.1) on a bounded domain, one cannot obtain any known equations for its fundamental solution; refer to [5, 36]. These naturally promote the rapid development of numerical methods for FPDEs. Therefore, the current study will focus on developing the numerical approaches for solving the problem (1.1).

If \( \gamma = \lambda(t) \equiv 1 \), the problem (1.1) collapses to the space fractional diffusion equation (SFDE) with variable coefficients. For such SFDEs, various robust numerical schemes are proposed by exploiting the shifted Grünwald discretization and the implicit Euler (or Crank–Nicolson) time-stepping procedure for two-sided R-L fractional derivatives and the first-order time derivative, respectively; refer to [28, 37, 52] for details. To improve the convergence order of such numerical methods, several studies combined different second-order accurate approximations for discretizing two-sided R-L fractional derivatives with the Crank–Nicolson technique in order to obtain the second-order finite difference schemes for solving the SFDEs with variable coefficients. However, the unconditional convergence of such second-order finite difference schemes is not easy to prove, refer to [8, 27, 31, 32, 44, 50, 54, 62] for discussions on this issue. However, these studies verified the unconditional convergence of second-order finite difference schemes often restrict diffusion coefficients positively bounded and relied on the spatial variable \( x \). Besides, other numerical treatments including the Chebyshev-tau, finite volume and finite element methods are proposed to solve the SFDEs with variable coefficients, refer, for example, to [6, 9, 19, 33–35, 42, 45] for details.

When \( \alpha = 2 \), the problem (1.1) is equivalent to the generalized time fractional diffusion equation (GTFDE) with variable coefficients. Such GTFDEs were first derived and studied by Sandev et al. in [48]. Later, Alikhanov adapted the classical \( L^1 \) formula [43] and employed the second-order weighted and shifted Grünwald difference (WSGD) formula [17] to approximate the generalized Caputo fractional derivative and the spatial R-L fractional derivative respectively for solving such GTFDEs with variable coefficients. Moreover, the convergence of his implicit difference schemes is proved to be unconditionally stable, refer to [1] for details. In addition, Khibie [23] has extended Alikhanov’s work to establish the stable implicit difference scheme for solving the multiterm GTFDE with variable coefficients.

On the other hand, although there are several numerical schemes about solving TSFDEs with variable coefficients –cf. \( \lambda(t) \equiv 1 \), however those that are proved to be unconditionally convergent [3, 10, 11, 30, 61] are only first- and \((2 - \gamma)\)-order accurate in space and time directions, respectively. It means that proving the unconditional convergence of implicit difference schemes with high-order spatial discretizations is often very challenging. Moreover, there are few results on numerical solutions of GTSFDEs with variable coefficients via finite difference methods in the literature. Such GTSFDEs can be regarded as a generalization of the GTFDEs introduced in [1, 48] and their numerical solutions should be more difficult due to lots of computational cost arising from the nonlocal properties in both spatial and temporal fractional derivatives. Therefore, establishing an unconditionally stable numerical scheme with low computational cost for solving such GTSFDEs with variable coefficients is a promising topic and also the main motivation of our current study. In this article, we develop
the implicit difference schemes for GTSFDEs with variable coefficients, then the implicit schemes are strictly proved to be unconditionally stable and convergent with second- and \((2 - \gamma)\)-order accuracy in space and time directions, respectively. Moreover, the implicit difference schemes lead to the solutions of the resulting linear systems with Toeplitz-like coefficient matrices which can be solved via direct method in \(O(N^3)\) operations along with \(O(N^2)\) storage. However, the efficient preconditioned Krylov subspace solvers are employed to reduce the above computational and memory cost to \(O(N \log N)\) and \(O(N)\), respectively, where \(N\) is the number of spatial grid nodes. Furthermore, the fast sum-of-exponential (SOE) approximation \([20]\) is extended to reduce computational and memory cost arising from the nonlocal property in the generalized Caputo fractional derivative with special function \(\lambda(t)\)'s. To the best of our knowledge, this is the first successful attempt to derive such a fast and stable numerical scheme of GTSFDEs with variable coefficients. Meanwhile, numerical experiments are reported to support our theoretical finding and effectiveness of the proposed schemes.

The rest of this article is organized as follows. In Section 2, the approximations of the generalized Caputo and R-L fractional derivatives are recalled to establish the implicit difference scheme. Meanwhile, the stability and convergence of the proposed difference scheme are proved in details. In Section 3, the practical implementation of the proposed schemes requires to solve a sequence of linear systems with Toeplitz-like coefficient matrices. The efficient preconditioned Krylov subspace solvers are adapted and investigated to handle such Toeplitz-like resultant linear systems. In Section 4, numerical experiments are reported to demonstrate the efficiency of the proposed method. Some concluding remarks are given in Section 5.

2 | AN IMPLICIT DIFFERENCE SCHEME FOR GTSFDES

In this section, we first review the approximation of the generalized Caputo fractional derivative and employ the second-order WSGD approximation \([17]\) to derive the implicit difference scheme to problem (1.1). Moreover, we have to provide the certain smoothness and monotonicity \([1]\) for the weighting function \(\lambda(t)\), then we can derive in details both the stability and convergence of our implicit difference scheme.

2.1 | The approximation for the generalized Caputo fractional derivative

We first briefly recall the generalized \(L1\) formula for approximating the temporal fractional derivative \(\dot{C}^\alpha_0 D^\gamma_t\) proposed in \([1]\) and denote its approximation result by \(\Delta^\alpha_0 D^\gamma_t\). To derive the difference scheme, we first introduce a rectangle \(Q_T = \{(x, t) : x_L \leq x \leq x_R, 0 \leq t \leq T\}\) discretized on the mesh \(\sigma_x = \sigma_x \times \sigma_t\), where \(\sigma_x = \left\{ x_i = x_L + ih, 0 \leq i \leq N, h = \frac{x_R - x_L}{N} \right\}\) and \(\sigma_t = \left\{ t_j = j\tau, j = 0, 1, \ldots, M, \tau = \frac{T}{M} \right\}\). We also denote by \(v = \{v_i \mid i = 0, 1, \ldots, N\}\) any grid function. Moreover, we denote the linear interpolation over the time interval \((t_j, t_{j+1})\) with \(0 \leq j \leq M - 1\) by

\[\Pi_{1,\delta} v(t) = v(t_{j+1}) \frac{t - t_j}{\tau} + v(t_j) \frac{t_{j+1} - t}{\tau} .\]

At each time step \(t_{j+1}\) with \(j = 0, 1, \ldots, M - 1\), the generalized \(L1\) formula is defined by

\[\dot{C}^\alpha_0 D^\gamma_t v(t) \bigg|_{t = t_{j+1}} = \frac{1}{\Gamma(1 - \gamma)} \int_0^{t_{j+1}} \frac{\lambda(t_{j+1} - \eta) v'(\eta) d\eta}{(t_{j+1} - \eta)^\gamma} .\]
where $\lambda = \lambda(t_i)$ and

$$v_{i,s} = \frac{v(t_{s+1}) - v(t_s)}{\tau}, \quad a_\ell = (\ell + 1)^{1-\gamma} - \ell^{1-\gamma},$$

$$b_\ell = \frac{1}{2 - \gamma}[(\ell + 1)^{2-\gamma} - \ell^{2-\gamma}] - \frac{1}{2}[(\ell + 1)^{1-\gamma} + \ell^{1-\gamma}], \quad \ell \geq 1,$$

and the definition of $R_1^j, R_2^j$ and their estimations can be separately found in [1]. The truncation error and property of the generalized $L1$ formula are also analyzed in [1, Lemma 4.1] as follows.

**Lemma 2.1** Assume that $\gamma \in (0, 1)$, $\lambda(t) > 0$, $\lambda'(t) \leq 0$, and $\lambda(t), v(t) \in C^2[0, t_{j+1}]$. Then

$$c_\ell D_t^{\gamma, \lambda(t)} v(t_{j+1}) = \Delta_{0,t_{j+1}}^{\gamma, \lambda(t)} + O(\tau^{2-\gamma}), \quad (2.1)$$

where $\Delta_{0,t_{j+1}}^{\gamma, \lambda(t)} w^{j+1} = \sum_{s=0}^j c_{j-s}[u(t_{s+1}) - u(t_s)]$ and $c_\ell = \frac{\tau^\gamma}{\Gamma(2-\gamma)}[\lambda_{j+1/2} a_\ell + (\lambda_\ell - \lambda_{\ell+1}) b_\ell] (\ell \geq 0)$. Moreover, the following inequalities hold:

$$a_0 > a_1 > \cdots > a_\ell > \frac{1 - \gamma}{(\ell + 1)^{\gamma}}, \quad b_0 > b_1 > \cdots > b_\ell > 0.$$

Based on the property of $a_\ell$ and $b_\ell$, we can obtain the following result for the coefficients $c_\ell$, which is absolutely vital for our theoretical analysis in the next subsection.

**Lemma 2.2** For all $\ell = 0, 1, \ldots, \gamma \in (0, 1)$ and $\lambda(t) \in C^2[0, T]$, where $\lambda(t) > 0$, $\lambda'(t) \leq 0$ for all $t \in [0, T]$, the following inequalities hold:

$$c_0 > c_1 > \cdots > c_\ell > \frac{\lambda(t_{\ell+1/2})}{\Gamma(1-\gamma)t_{\ell+1}^\gamma}.$$

After we introduce the temporal discretization, it is the time to characterize the discretization in the space variable. First of all, we denote by

$$\mathcal{L}^{n+a}(\mathbb{R}) = \left\{ v \mid v \in L_1(\mathbb{R}) \text{ and } \int_{-\infty}^{+\infty} (1 + k)^{n+a} |\hat{v}(k)| dk < \infty \right\},$$

where $\hat{v}(k) = \int_{-\infty}^{+\infty} e^{ikx} v(x) dx$ is the Fourier transformation of $v(x)$, and by $i = \sqrt{-1}$ the imaginary unit. Then we introduce the following preliminary lemma, which provides numerical approximations for the spatial R-L fractional derivatives:

**Lemma 2.3** Let $v(x) \in \mathcal{L}^{2+a}(\mathbb{R})$ and define the following difference operators

$$\delta_{k,+}^{a} v(x) = \frac{1}{h^a} \sum_{k=0}^{\left[\frac{x-k}{h}\right]} w_k^{(a)} v(x - (k - 1)h) \quad \text{and} \quad \delta_{k,-}^{a} v(x) = \frac{1}{h^a} \sum_{k=0}^{\left[\frac{x-k}{h}\right]} w_k^{(a)} v(x + (k - 1)h).$$
Then, for a fixed $h$, we have
\[
\alpha D_x^\alpha v(x) = \delta_{x,1}^\alpha v(x) + \mathcal{O}(h^2) \quad \text{and} \quad \beta D_x^\beta v(x) = \delta_{x,1}^\beta v(x) + \mathcal{O}(h^2),
\]
where $[\cdot]$ is the floor function and
\[
\begin{aligned}
\{w_0^{(a)} = \kappa_1 \delta_{0}^{(a)} + \kappa_0 g_0^{(0)}, \quad w_1^{(a)} = \kappa_1 \delta_{1}^{(a)} + \kappa_0 g_0^{(0)}, \quad w_k^{(a)} = \kappa_1 \delta_{k}^{(a)} + \kappa_0 g_{k-1}^{(a)} + \kappa_{-1} g_{k-2}^{(a)}, \quad k \geq 2,
\end{aligned}
\]
with
\[
\kappa_1 = \frac{a^2 + 3a + 2}{12}, \quad \kappa_0 = \frac{4 - a}{6}, \quad \kappa_{-1} = \frac{a^2 - 3a + 2}{12}, \quad \text{and} \quad g_k^{(a)} = (-1)^k \binom{a}{k}.
\]

At this stage, the numerical approximations of both the temporal and spatial fractional derivatives have been set for the derivation of the targeted implicit difference scheme. Let $u(x, t) \in C_{x,t}^4([x_L, x_R] \times [0, T])$ be a solution to the problem (1.1). Then we consider Equation (1.1) at the set of grid points $(x, t) = (x_i, t_j+1) \in \Omega_T, i = 1, 2, \ldots, N-1, j = 0, 1, \ldots, M-1$:
\[
C_0 D_t^{\lambda(t)} u(x, t) = \sum_{\gamma, \lambda} \left( \sum_{i=0}^{N} \sum_{j=0}^{M} g_i^{(a)} f_j^{(b)} u(x, t) \right)_{i, j} + f(x_i, t_{j+1}).
\]
Let $U$ be a grid function defined by
\[
U_i^j := u(x_i, t_j) \quad \text{and} \quad f_i^j = f(x_i, t_j), \quad 0 \leq i \leq N, \quad 0 \leq j \leq M.
\]
Using this notation and recalling Lemma 2.1 and Lemma 2.3, we can write the problem (1.1) at the grid points $(x_i, t_{j+1})$ as follows
\[
\Delta_{0, t_j+1}^{\lambda(t)} U_i^{j+1} = \xi_i^{j+1} (\delta_{x} U_i^{j+1}) + f_i^{j+1} + R_i^{j+1}, \quad 1 \leq i \leq N - 1, \quad 0 \leq j \leq M - 1, \tag{2.2}
\]
where $\{R_i^{j+1}\}$ are small and satisfy the relation $|R_i^{j+1}| = \mathcal{O}(r_{2-\gamma} + h^2)$ for $1 \leq i \leq N - 1, 0 \leq j \leq M - 1$.

We omit them and use the initial-boundary value conditions
\[
\begin{aligned}
U_i^0 &= \phi(x_i), \quad 1 \leq i \leq N - 1, \\
U_0^j &= \phi(t_j), \quad U_N^j = \psi(t_j), \quad 0 \leq j \leq M.
\end{aligned}
\]

For the sake of clarity, we introduce the notations
\[
\xi_i^j = \xi(x_i, t_j), \quad \delta_{x} U_i^{j+1} = \frac{1}{h^2} \left[ \sum_{k=0}^{N-1} \sum_{j=0}^{M} \delta_{x}^{(a)} u_{i-k}^{j+1} + (1 - p) \sum_{k=0}^{N} \delta_{x}^{(a)} u_{i+1}^{j+1} \right],
\]
and then we arrive at the implicit difference scheme with (local) truncation errors of $\mathcal{O}(r_{2-\gamma} + h^2)$:
\[
\begin{aligned}
\Delta_{0, t_j+1}^{\lambda(t)} U_i^{j+1} = \xi_i^{j+1} (\delta_{x} U_i^{j+1}) + f_i^{j+1}, \quad i = 1, 2, \ldots, N - 1, \quad j = 0, 1, \ldots, M - 1, \\
U_i^0 &= \phi(x_i), \quad i = 0, 1, \ldots, N, \\
U_0^j &= \phi(t_j), \quad U_N^j = \psi(t_j), \quad j = 0, 1, \ldots, M.
\end{aligned} \tag{2.3}
\]

It is interesting to note that for $\lambda(t) \equiv 1$ and $\gamma \rightarrow 1$, Equation (2.3) reduces to the classical backward Euler scheme for solving the SFDEs with variable coefficients [25, 37]. Similarly, if $\alpha = 2$, the above scheme (2.3) collapses to the implicit difference scheme introduced in [1] for solving the variable-coefficient GTFDEs.
2.2 Stability and convergence analysis

In this subsection, we are committed to analyzing both the stability and convergence for the implicit difference scheme (2.3). We define

\[ V_h = \{v \mid v = \{v_i\} \text{ is a grid function on } \mathbb{R} \} \]

and, for all \( u, v \in V_h \), the discrete inner product and corresponding discrete \( L^2 \)-norms

\[ (u, v) = h \sum_{i=1}^{N-1} u_i v_i, \quad \text{and} \quad \|u\| = \sqrt{(u, u)}. \]

The starting point of our analysis is the following theoretical result.

**Lemma 2.4** ([14, 55]). Let \( \alpha \in (1, 2) \) and \( g_k^{(a)} \) be defined in Lemma 2.3, then we obtain

\[
\begin{aligned}
W_0^{(a)} &= k_1 > 0, \quad w_1^{(a)} < 0, \quad w_k^{(a)} > 0, \quad k \geq 3, \\
\sum_{k=0}^{\infty} w_k^{(a)} &= 0, \quad \sum_{k=0}^{N} w_k^{(a)} < 0, \quad N > 1, \\
w_0^{(a)} + w_2^{(a)} &\geq 0.
\end{aligned}
\]

In fact, this lemma does not show whether \( w_2^{(a)} \) is positive or negative. After simple calculations, we obtain

\[
w_2^{(a)} = \frac{\alpha^4}{24} + \frac{\alpha^3}{12} + \frac{5\alpha^2}{24} - \alpha + \frac{1}{6}, \quad (2.4)
\]

where \( \alpha \in (1, 2] \) and it can be plotted as in Figure 1. As seen from Figure 1, the following proposition can be derived, which is helpful to analyze the property of the coefficient matrices appearing in Equation (3.3) in the next section.

**Proposition 2.1** When \( \alpha \in (1, \alpha_0) \), then \( w_2^{(a)} < 0 \). Similarly, when \( \alpha \in [\alpha_0, 2) \), then \( w_2^{(a)} \geq 0 \) with \( \alpha_0 \approx 1.8223 \). Moreover, the sufficient condition for \( W_a \) and \( W_a^T \) to be diagonally dominant is \( \alpha \in [\alpha_0, 2] \), where the matrix

\[
W_a = \begin{pmatrix}
W_0^{(a)} & W_0^{(a)} & 0 & \cdots & 0 & 0 \\
W_2^{(a)} & W_1^{(a)} & W_0^{(a)} & 0 & \cdots & 0 \\
: & W_2^{(a)} & W_1^{(a)} & \ddots & \ddots & : \\
: & \vdots & \ddots & \ddots & \ddots & \vdots \\
W_{N-2}^{(a)} & \ddots & \ddots & \ddots & \ddots & 0 \\
W_{N-1}^{(a)} & W_{N-2}^{(a)} & \cdots & \cdots & W_1^{(a)} & W_0^{(a)}
\end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)}. \quad (2.5)
\]

**Proof.** Since \( \alpha \in [\alpha_0, 2] \), it holds \( w_1^{(a)} < 0 \) and \( w_k^{(a)} \geq 0 \) (\( k \neq 1 \)). According to \( \sum_{k=0}^{\infty} w_k^{(a)} = 0 \), it holds that both \( W_a \) and \( W_a^T \) are diagonally dominant [27].

Based on Lemma 2.4, the first two properties of the discrete inner product related to two approximate operators \( \delta_{k+}^{(a)} \) and \( \delta_{k-}^{(a)} \) can be shown below.
Lemma 2.5. For $\alpha \in (1, 2)$ and $N \geq 5$, and any $v \in V_h$, it holds that

$$(-\delta_{\alpha, h}^+ v, v) = (-\delta_{\alpha, h}^- v, v) > c \ln 2 \|v\|^2,$$

where $c$ is a positive constant independent of the spatial step size $h$.

**Theorem 2.1** For $\alpha \in (1, 2)$, and any $v \in V_h$, it holds that

$$(\delta_{\alpha, h}^- v, v) < -c \ln 2 \|v\|^2,$$

where $c$ is the same constant appearing in Lemma 2.5.

**Proof.** The concrete expression of $(\delta_{\alpha, h}^+ v, v)$ can be written as

$$(\delta_{\alpha, h}^+ v, v) = p(\delta_{\alpha, h}^+ v, v) + (1 - p)(\delta_{\alpha, h}^- v, v) \leq -c \ln 2 \|v\|_2,$$

and this completes the proof of Theorem 2.1.

To establish the stability of the difference scheme, we still need to introduce the following lemma.

**Lemma 2.6** For any function $v(t)$ defined on the discrete grid $\sigma_T = \{t_j = j\tau : j = 0, 1, \ldots, M\}$, the following inequality holds

$$v^{j+1}(K^{j+1})^{-1} \Delta_{0,j+1}^{r,d(t)} v \geq \frac{1}{2} \Delta_{0,j+1}^{r,d(t)} \|v\|_{(K^{j+1})^{-1}}^2,$$

where $K^{j+1} = \text{diag}(\xi_1^{j+1}, \xi_2^{j+1}, \ldots, \xi_n^{j+1}) > 0$ and $\|v\|_{(K^{j+1})^{-1}}^2 = v^T(K^{j+1})^{-1} v$. 

---

**FIGURE 1** The plot of $\omega_2^{(\alpha)}$ with $\alpha \in (1, 2)$ [Color figure can be viewed at wileyonlinelibrary.com]
Proof. We rewrite the following inner product
\[ \nu^{j+1}(K^{j+1})^{-1} \Delta^{y,t}_{0,\tau_{j+1}} \nu = \tilde{\nu}^{j+1} \Delta^{y,t}_{0,\tau_{j+1}} \tilde{\nu} \geq \frac{1}{2} \Delta^{y,t}_{0,\tau_{j+1}} \| \tilde{\nu} \|_2, \]
where \( \tilde{\nu} = (K^{j+1})^{-\frac{1}{2}} \nu \) regarded as a (weighted) function \( \nu(t) \) defined on the discrete grid \( \omega_t \). Meanwhile, the inequality (2.7) is correct due to [1, Lemma 4.4].

Another ingredient, introduced as the following lemma, is also required to describe the diagonally weighted norm that will be used in the next theorem.

**Lemma 2.7** ([29]). Let \( H \in \mathbb{R}^{n \times n} \) be a symmetric matrix with eigenvalues \( \tilde{\lambda}_1 \geq \tilde{\lambda}_2 \geq \cdots \geq \tilde{\lambda}_n \). Then for all \( w \in \mathbb{R}^{n \times 1} \),
\[ \tilde{\lambda}_n w^T w \leq w^T H w \leq \tilde{\lambda}_1 w^T w. \]

Now we can conclude the stability and convergence of the implicit difference scheme (2.3). For simplicity of presentation, we denote \( a^{j+1}_s = c_{j-s} \), then \( \Delta^{y,t}_{0,\tau_{j+1}} u = \sum_{s=0}^{j} (u^{s+1} - u^{s}) a^{j+1}_s \).

**Theorem 2.2** If we define \( \| f^{j+1} \|^{2} = h \sum_{i=1}^{N-1} f^2(x_i, t_{j+1}) \), then the implicit difference scheme (2.3) is unconditionally stable and the following a priori estimate holds:
\[ \| u^{j+1} \|^{2}_{(K^{j+1})^{-1}} \leq \frac{1}{\xi_{\min}} \left[ \| u^{0} \|^{2} + \frac{\Gamma(1-\gamma)T^\gamma}{2c_{\xi_{\min}} \ln 2} \max_{0 \leq s \leq M-1} \| f^{j+1} \|^{2} \right] \]
where \( u^{j+1} = [u^{j+1}_1, u^{j+1}_2, \ldots, u^{j+1}_{N-1}]^{T} \).

Proof. To make an inner product of Equation (2.3) with \( u^{j+1} \), we have
\[ (\Delta^{y,t}_{0,\tau_{j+1}} u, (K^{j+1})^{-1} u^{j+1}) = (\delta^y u^{j+1}, u^{j+1}) + (f^{j+1}, (K^{j+1})^{-1} u^{j+1}). \]
It follows from Theorem 2.1 and Lemma 2.6 that
\[ (\delta^y u^{j+1}, u^{j+1}) \leq -c \ln 2 \| u^{j+1} \|^{2} \]
and
\[ (\Delta^{y,t}_{0,\tau_{j+1}} u, (K^{j+1})^{-1} u^{j+1}) \geq \frac{1}{2} \Delta^{y,t}_{0,\tau_{j+1}} \| u \|^{2}_{(K^{j+1})^{-1}}. \]
Substituting (2.11)–(2.12) into (2.10) and using the Cauchy–Schwarz and Young’s inequalities, we obtain
\[ \frac{1}{2} \Delta^{y,t}_{0,\tau_{j+1}} \| u \|^{2}_{(K^{j+1})^{-1}} \leq -c \ln 2 \| u^{j+1} \|^{2} + (f^{j+1}, (K^{j+1})^{-1} u^{j+1}) \]
\[ \leq -c \ln 2 \| u^{j+1} \|^{2} + c_{\xi_{\min}} \ln 2 \| u^{j+1} \|^{2}_{(K^{j+1})^{-1}} + \frac{1}{4c_{\xi_{\min}} \ln 2} \| f^{j+1} \|^{2}_{(K^{j+1})^{-1}} \]
\[ \leq -c \ln 2 \| u^{j+1} \|^{2} + c \ln 2 \| u^{j+1} \|^{2} + \frac{1}{4c_{\xi_{\min}} \ln 2} \| f^{j+1} \|^{2}_{(K^{j+1})^{-1}} \quad \text{(cf. Lemma 2.7)} \]
\[ = \frac{1}{4c_{\xi_{\min}} \ln 2} \| f^{j+1} \|^{2}_{(K^{j+1})^{-1}}. \]
Next, we have the following inequality
\[
d_{j}^{i+1} \|u^{i+1}\|_{(K^{i+1})^{-1}}^2 \leq \sum_{s=1}^{j} (d_{s}^{i+1} - d_{s-1}^{i+1})\|u^i\|_{(K^{i+1})^{-1}}^2 + d_{0}^{i+1} \|u^0\|_{(K^{i+1})^{-1}}^2 + \frac{1}{2\xi_{\min}}\|f^{i+1}\|_{(K^{i+1})^{-1}}^2.
\]  
(2.13)

Employing the inequality \(d_{0}^{i+1} = c_{j} \geq \frac{\alpha(T)}{\Gamma(1-\gamma)T^\gamma}\) (cf. [1, Theorem 5.1]), we obtain
\[
d_{j}^{i+1} \|u^{i+1}\|_{(K^{i+1})^{-1}}^2 \leq \sum_{s=1}^{j} (d_{s}^{i+1} - d_{s-1}^{i+1})\|u^i\|_{(K^{i+1})^{-1}}^2
\]
\[
+ d_{0}^{i+1}\left[\|u^0\|_{(K^{i+1})^{-1}}^2 + \frac{\Gamma(1-\gamma)T^\gamma}{2\xi_{\min}\ln 2T} \|f^{i+1}\|_{(K^{i+1})^{-1}}^2\right].
\]  
(2.14)

Suppose \(h < 1\) and denote
\[
P \triangleq \frac{1}{\xi_{\min}} \left[\|u^0\|^2 + \frac{\Gamma(1-\gamma)T^\gamma}{2\xi_{\min}\ln 2T} \max_{0 \leq s \leq M-1} \|f^{i+1}\|^2\right].
\]

Then, Equation (2.14) can be rewritten as
\[
d_{j}^{i+1} \|u^{i+1}\|_{(K^{i+1})^{-1}}^2 \leq \sum_{s=1}^{j} (d_{s}^{i+1} - d_{s-1}^{i+1})\|u^i\|_{(K^{i+1})^{-1}}^2 + d_{0}^{i+1} P.
\]  
(2.15)

At this stage, by mathematical induction we prove that
\[
\|u^i\|_{(K^{i+1})^{-1}}^2 \leq P, \quad 0 \leq s \leq j + 1,
\]  
(2.16)

is valid for the fixed \(j\). The result is obviously true for \(s = 0\) from (2.14). Assuming that (2.16) holds for all \(0 \leq s \leq j\) (\(0 \leq j \leq M - 1\)), then from (2.14) at \(0 \leq s \leq j + 1\), one has
\[
d_{j}^{i+1} \|u^{i+1}\|_{(K^{i+1})^{-1}}^2 \leq \sum_{s=1}^{j} (d_{s}^{i+1} - d_{s-1}^{i+1})\|u^i\|_{(K^{i+1})^{-1}}^2 + d_{0}^{i+1} P
\]
\[
\leq \sum_{s=1}^{j} (d_{s}^{i+1} - d_{s-1}^{i+1})P + d_{0}^{i+1} P
\]
\[
= d_{j}^{i+1} P.
\]

This completes the proof of Theorem 2.2. \(\blacksquare\)

The following theorem shows that our proposed implicit difference scheme achieves \((2-\gamma)\)-order and quadratic-order convergence in time and space variables, respectively, when the solution of Equation (1.1) is sufficiently smooth. To our knowledge, it is the first theoretical result on the convergence of implicit difference schemes for solving the variable-coefficient GTSFDEs (1.1).

**Theorem 2.3** Suppose that \(u(x,t) \in C_x^{4,2}([x_L,x_R] \times [0,T])\) is the solution of Equation (1.1) and \(\{u^i|x_i \in \varpi_h, 0 \leq j \leq M\}\) is the solution of the implicit difference scheme (2.3).

Define
\[
E_j^i = u(x_i,t_j) - u^i, \quad x_i \in \varpi_h, 0 \leq j \leq M,
\]  
(2.17)

where \(\varpi_h = \{x_i = ih, i = 0, 1, \ldots, N\}; Nh = b - a\), then there exists a positive constant \(\tilde{c}\) such that
\[
\|E^i\| \leq \tilde{c}(\tau^{2-\gamma} + h^2), \quad 0 \leq j \leq M.
\]
It can be easily obtained that $E^j$ satisfies the following error equation

$$
\begin{align*}
\Delta x^{j+1}(E^j) = \frac{\partial^+}{h^2} (e^r E^j) + R^j,
\end{align*}
$$

where $R^j$ is a Toeplitz matrix; refer to [37, 40]. Therefore, it can be stored with $N$ entries and the matrix-vector product involving the matrix $\Delta x$ is inherently sequential, thus it is difficult to parallelize it over time. This implies that we need to solve the sequence of linear systems

$$
\begin{align*}
\text{Theorem 2.3 implies that our numerical scheme converges to the optimal order } & \mathcal{O}(\tau^{2-\gamma} + h^2) \text{ in the } L^2 \text{-norm, when the solution of Equation (1.1) is sufficiently smooth. Besides, if the solution of Equation (1.1) is nonsmooth, several useful alternatives utilizing the nonuniform temporal step or initial correction techniques [21, 22, 26, 51] can be adapted to address this problem. However, that is not the emphasis of this current study and we point the reader to the next section for a short discussion. In addition, the above analysis can be similarly adapted to remedy defects in our previous work [14], which only focuses on the model problem with time-varying diffusion coefficients.}
\end{align*}
$$


3 | EFFICIENT IMPLEMENTATION OF THE PROPOSED IMPLICIT DIFFERENCE SCHEME

In order to develop an efficient implementation of the proposed scheme, we rewrite the implicit difference scheme (2.3) into the following form with $i = 1, 2, \ldots, N - 1$ and $j = 0, 1, \ldots, M - 1$:

$$
\begin{align*}
(c_0 u^{j+1}_i - c_j u^0_i) - \sum_{s=1}^{j} (c_{s-1} - c_s) u^{j+1-s}_i = \frac{\varphi^{j+1}}{h^2} \left[ \sum_{k=0}^{i+1} p \sum_{k=0}^{j=N+i-1} w_k^{(a)} u^{j+1}_{i-k+1} + (1 - p) \sum_{k=0}^{j=N+i-1} w_k^{(a)} u^{j+k-1} \right] + f_i^{j+1},
\end{align*}
$$

or, equivalently,

$$
\begin{align*}
(c_0 u^{j+1}_i - \frac{\varphi^{j+1}}{h^2} \left[ \sum_{k=0}^{i+1} p \sum_{k=0}^{j=N+i-1} w_k^{(a)} u^{j+1}_{i-k+1} + (1 - p) \sum_{k=0}^{j=N+i-1} w_k^{(a)} u^{j+k-1} \right]) = c_j u^0_i + \sum_{s=1}^{j} (c_{s-1} - c_s) u^{j+1-s} + f_i^{j+1}.
\end{align*}
$$

At this stage, the above implicit difference scheme can be reformulated as the following sequence of linear systems,

$$
\begin{align*}
\mathcal{M}^{(j+1)} u^{j+1} = c_0 u^0 + \sum_{s=1}^{j} (c_{s-1} - c_s) u^{j+1-s} + f^{j+1}, \quad j = 0, 1, 2, \ldots, M - 1,
\end{align*}
$$

where $\mathcal{M}^{(j+1)} = c_0 I - \frac{K^{(j+1)}}{h^2} [p W_a + (1 - p) W_f]$, $u^j = [u^j, u^j_2, \ldots, u^j_{N-1}]^T$, $f^j = [f^j_1, f^j_2, \ldots, f^j_{N-1}]^T$, $K^{(j+1)} = \text{diag}(\varphi^{j+1}_1, \varphi^{j+1}_2, \ldots, \varphi^{j+1}_N)$ and $I$ is the identity matrix of order $(N-1)$. Meanwhile, it is obvious that $W_a$ and $W_f$ is a Toeplitz matrix; refer to [37, 40]. Therefore, it can be stored with $N$ entries and the matrix-vector product involving the matrix $\mathcal{M}^{(j)}$ can be evaluated via fast Fourier transforms (FFTs) in $\mathcal{O}(N \log N)$ operations [25, 40]. On the other hand, it is meaningful to remark that the sequence of linear systems (3.3) corresponding to the implicit scheme (2.3) is inherently sequential, thus it is difficult to parallelize it over time. This implies that we need to solve the sequence of linear systems
(3.3) one by one. Then Krylov subspace methods with suitable preconditioners [7, 25, 41] can be efficient candidates for solving Toeplitz-like linear systems since their complexity is of only $O(N \log N)$ arithmetic operations per iteration step.

In order to solve Equation (3.3) effectively, we consider two specific classes of problems:

(i) When the diffusion coefficient $\xi(x, t) \equiv \xi$, the coefficient matrix of Equation (3.3) will be a time-independent Toeplitz matrix, that is, $M^{j+1} = M$; then we can compute its matrix inverse via the Gohberg–Semencul formula (GSF) [13] using only its first and last columns. Such a strategy does not need to call the preconditioned Krylov subspace solvers at each time level $0 \leq j \leq M - 1$, and the solution at each time level (i.e., $M^{-1} u^{j+1}$) can be calculated via about six FFTs, thus saving considerable computational cost; refer to [14, 18, 32, 60] for detail.

(ii) When the diffusion coefficient is just a function related to both $x$ and $t$, that is, $\xi(x, t)$, the coefficient matrix of Equation (3.3) becomes the sum of a scalar matrix and of a diagonal-multiply-Toeplitz matrix, which is time-dependent. In this case, Equation (3.3) has to be solved via a preconditioned Krylov subspace solver at each time level $j$.

Based on the above considerations, we still require to solve several nonsymmetric Toeplitz(-like) linear systems, whose matrix–vector products can be efficiently calculated via FFTs, thus we utilize the biconjugate gradient stabilized (BiCGSTAB) method which has a fast and smooth convergence [53]. For accelerating BiCGSTAB, we consider the following skew-circulant and banded preconditioners:

\[
P_{sk} = \begin{cases} 
  c_0I - \frac{\xi}{h} [p \cdot sk(W_a) + (1 - p)sk(W_a^T)], & \xi(x, t) \equiv \xi, \\
  c_0I - \frac{\xi^{(j+1)}}{h} [p \cdot sk(W_a) + (1 - p)sk(W_a^T)], & \xi^{(j+1)} = \frac{1}{N-1} \sum_{i=1}^{N-1} \xi(x_i, t_{j+1}), 
\end{cases}
\]

where the vector $\delta = [w_1^{(a)}, w_2^{(a)}, \ldots, w_{N-2}^{(a)}, -w_0^{(a)}]^T$ is the first column of the skew-circulant matrix $sk(W_a)$ [60], and

\[
P_b = \begin{cases} 
  c_0I - \frac{\xi}{h} [pW_{a, \ell} + (1 - p)W_{a, \ell}^T], & \xi(x, t) \equiv \xi, \\
  c_0I - \frac{\xi^{(j+1)}}{h} [pW_{a, \ell} + (1 - p)W_{a, \ell}^T], & \text{(general case)},
\end{cases}
\]

with the band matrix

\[
W_{a, \ell} = \begin{bmatrix} 
  w_1^{(a)} & w_0^{(a)} \\
  : & : \\
  w_\ell^{(a)} & w_0^{(a)} \\
  : & : \\
  w_0^{(a)} & \cdots & w_0^{(a)} \\
  \cdots & \cdots & \cdots \\
  w_\ell^{(a)} & \cdots & w_1^{(a)} 
\end{bmatrix}, \quad \ell \in \mathbb{N}^+,
\]

respectively. Meanwhile, the high efficiency of skew-circulant and banded preconditioners for (time-)space FDEs has been shown in [28, 60, 61].

In practical implementations, when $P_{sk}$ or $P_b$ is employed as the preconditioner, a fast preconditioned version of the BiCGSTAB method is obtained. During each BiCGSTAB iteration, two preconditioning steps are added in which one has to solve either the linear system $P_{sk} \mathbf{z} = \mathbf{y}$ or $P_b \mathbf{z} = \mathbf{y}$ for some given vector $\mathbf{y}$. Thus, some additional storage and computational cost are still required. However, we point out that $P_{sk}$ (resp., $P_b$) can also be efficiently stored in $\mathcal{O}(N)$ (resp., $\mathcal{O}(\ell N)$) memory by only storing the $(N - 1)$-dimensional vector $\delta$ in (3.4) (resp., the band matrix $W_{a, \ell}$ in (3.5)). Besides,
as $P_{sk}$ is the skew matrix, we observe that
\[
P_{sk} = \Omega^* F^s \left\{ c_0 I - \frac{\xi(i+1)}{h^2} [p\Lambda_s + (1 - p)\Lambda_y] \right\} F\Omega, \quad sk(W_a) = \Omega^* F^s \Lambda_s F\Omega, \tag{3.6}
\]
where $\Omega = \text{diag} \left( 1, (-1)^{-\frac{1}{N-1}}, \ldots, (-1)^{-\frac{N-2}{N-1}} \right)$. $F$ is the discrete Fourier matrix and its conjugate transpose $F^*$. According to Equation (3.6), the inverse-matrix–vector product $z = P_{sk}^{-1} y$ can be carried out in $\mathcal{O}(N \log N)$ operations via the (inverse) FFTs. Most importantly, the diagonal matrix $\Lambda_s$ can be computed in advance and only once per time step. On the other hand, since $W_a, r$ is a band matrix, then $P_b$ should be a band matrix of bandwidth $2r+1$ and $z = P_b^{-1} y$ can be computed by the banded LU decomposition [28, 61] in $\mathcal{O}(\ell N)$ arithmetic operations ($\ell \ll N$). In one word, we employ a fast preconditioned BiCGSTAB solution method with low memory requirement and computational cost per iteration, while the number of iterations and thus the total computational cost are greatly reduced. Compared to the skew–circulant preconditioner, the banded preconditioner needs more computational cost to update at each time level; refer to the next section for a discussion.

On the other hand, it is worthwhile to note that when $a \geq a_0$, the coefficient matrix $\mathcal{M}^{(j+1)}$ are diagonally dominant with positive diagonal elements [27] due to Proposition 2.1 and $\xi(x, t) > 0$. Meanwhile, the banded preconditioner was shown to be considerably efficient for solving the linear systems with diagonally dominant coefficient matrix, which arise from the numerical discretization of (time-)space FDEs; refer, for example, to [27, 28, 61] for a discussion.

4 | NUMERICAL EXPERIMENTS

The numerical experiments presented in this section have a twofold objective. They illustrate that the proposed implicit difference scheme (IDS) for the GTSFDE (1.1) can indeed converge with the order of $O(\varepsilon^2 + h^2)$. Meanwhile, they assess the computational efficiency of the fast solution techniques described in Section 3. Our choice of Krylov subspace method and direct solver in Example 2 (where $\xi(x, t) \equiv \xi$ and $\mathcal{M}^{(j+1)} \equiv \mathcal{M}$ will be independent of time levels) are the built-in MATLAB implementations of the preconditioned BiCGSTAB method and of the LU factorization, respectively, while in Example 1 with variable coefficients (where the coefficient matrices $\mathcal{M}^{(j+1)}$ change at each time level) we use the MATLAB’s backslash operator. The stopping criterion for the BiCGSTAB method with the two different preconditioners is $\|r^{(k)}\|_2/\|r^{(0)}\|_2 \leq 10^{-12}$, where $r^{(k)}$ is the residual vector of the linear system after $k$ iterations; the initial guess is chosen as the zero vector. All experiments were performed on a Windows 10 (64 bit) PC-Intel(R) Core(TM) i5-8250U CPU @1.60 GHz–1.80GHz, 8GB of RAM using MATLAB 2017b with machine epsilon $10^{-16}$ in double precision floating point arithmetic. By the way, all timings (measured in seconds) are averages calculated over 20 runs of our algorithms. Before we report the numerical results of the IDS for the problem (1.1), we introduce the following notations that are adopted throughout this section:

\[
\text{Error}_{\infty} = \max_{0 \leq j \leq M} \|E_j\|_{\infty} \quad \text{and} \quad \text{Error}_2 = \max_{0 \leq j \leq M} \|E_j\|_2,
\]
then

\[
\text{Rate}_{\infty} = \begin{cases} \log_{r_1/r_2} \left( \frac{\text{Error}_{\infty,r_1}}{\text{Error}_{\infty,r_2}} \right), & \text{(temporal convergence order)}, \\ \log_{h_1/h_2} \left( \frac{\text{Error}_{\infty,h_1}}{\text{Error}_{\infty,h_2}} \right), & \text{(spatial convergence order)}, \end{cases}
\]

1If the diffusion coefficient $\xi(x, t) \equiv \xi(t)$, then are time-varying constants, which is available for other similar cases $\xi^{(j+1)}$. 


In Tables 1 and 2, we display the maximum-norm errors and $L_2$-norm errors of the IDS scheme for solving the problem (1.1) with variable diffusion coefficients in spatial and temporal variables, respectively. More precisely, the results of Table 1 with different $(\gamma, \alpha, b)$’s and $h = 2^{-12}$ show, as expected, a reduction in the maximum- or $L_2$-norm error as the number of time steps of our IDS is increased, and the temporal convergence order of IDS is $O(\tau^{2-\gamma})$. At the same time, our experiments displayed in Table 2 with different $(\gamma, \alpha, b)$’s and $\tau = 2^{-10}$ show a reduction in the maximum- or 

| $(\gamma, \alpha)$ | $\tau$ | $b = 1.0$ | $\tau$ | $b = 2.0$ |
|-------------------|--------|-----------|--------|-----------|
|                    |        | Error$_{\infty}$ | Rate$_{\infty}$ | Error$_{\infty}$ | Rate$_{\infty}$ |
| $(0.2, 1.1)$      | 1/8    | 5.9655e-4   | --        | 3.1311e-4   | --         |
|                   | 1/16   | 1.7385e-4   | 1.7788   | 9.0388e-5   | 1.7925     |
|                   | 1/32   | 5.0703e-5   | 1.7777   | 2.6194e-5   | 1.7869     |
|                   | 1/64   | 1.4813e-5   | 1.7752   | 7.6240e-6   | 1.7806     |
| $(0.5, 1.5)$      | 1/8    | 1.0328e-3   | 1.0162e-3| 5.1328e-4   | 1.0164     |
|                   | 1/16   | 3.7458e-4   | 1.4627   | 1.8639e-4   | 1.4614     |
|                   | 1/32   | 1.3450e-4   | 1.4777   | 6.7060e-5   | 1.4748     |
|                   | 1/64   | 4.8098e-5   | 1.4835   | 2.4016e-5   | 1.4815     |
| $(0.9, 1.9)$      | 1/8    | 2.9303e-3   | 2.8851e-3| 1.3940e-3   | 1.3710e-3  |
|                   | 1/16   | 1.3909e-3   | 5.5779e-4| 5.0407e-4   | 5.0407e-4  |
|                   | 1/32   | 6.5575e-4   | 1.0848   | 3.1678e-4   | 1.0767     |
|                   | 1/64   | 1.3450e-4   | 1.0750   | 6.5664e-4   | 1.0621     |

Example 1 In this example, we solve the initial-boundary value problem of GTSFDE (1.1) with variable coefficients and $\lambda(t) = e^{-\beta t}$, $b \geq 0$, the spatial domain $[x_L, x_R] = [0, 2]$ and the time interval is $[0, T] = [0, 1]$. The diffusion coefficient function is given as $\xi(x, t) = 1 + x^2 + \sin t$. The source term is

$$f(x, t) = \frac{2x^2}{2x^2 + \sin t} \left( 1 + x^2 + \sin t \right)$$

and

$$\text{Rate}_2 = \begin{cases} \log_{r_1/r_2} \left( \frac{\text{Error}_{r_1}}{\text{Error}_{r_2}} \right), & \text{(temporal convergence order)}, \\ \log_{h_1/h_2} \left( \frac{\text{Error}_{h_1}}{\text{Error}_{h_2}} \right), & \text{(spatial convergence order)}. \end{cases}$$

The exact (and smooth) solution to this problem is $u(x, 0) = g(0)x^2(2 - x)^2$, where $g(t)$ is given as follows:

$$g(t) = 1 + \frac{2 - (2 + 2bt + b^2t^2)e^{-bt}}{b^3}$$

for any $\alpha \in (1, 2)$ and $b \in \mathbb{R}^+$. Numerical experiments with our proposed difference scheme are reported in the following Tables 1–4.
**TABLE 2**  $L_2$-norm and maximum norm errors versus grid size reduction when $\tau = 2^{-10}$ and $p = 0.7$ in Example 1

| $(\gamma, \alpha)$ | $b = 1.0$ | | $b = 2.0$ | |
|-------------------|----------|---|----------|---|
| $h$ | $Error_\infty$ | Rate | $Error_2$ | Rate | $Error_\infty$ | Rate | $Error_2$ | Rate |
| (0.2,1.1) | 2/8 | 1.033e-1 | – | 9.578e-2 | – | 1.075e-1 | – | 9.668e-2 | – |
| | 2/16 | 2.419e-2 | 2.094 | 2.330e-2 | 2.093 | 2.391e-2 | 2.169 | 2.351e-2 | 2.040 |
| | 2/32 | 7.354e-3 | 1.718 | 5.568e-3 | 2.065 | 6.679e-3 | 1.840 | 5.617e-3 | 2.065 |
| | 2/64 | 2.033e-3 | 1.855 | 1.335e-3 | 2.059 | 1.835e-3 | 1.863 | 1.347e-3 | 2.059 |
| (0.5,1.5) | 2/8 | 7.041e-2 | – | 6.703e-2 | – | 6.902e-2 | – | 6.564e-2 | – |
| | 2/16 | 1.652e-2 | 2.091 | 1.568e-2 | 2.095 | 1.611e-2 | 2.098 | 1.531e-2 | 2.099 |
| | 2/32 | 3.924e-3 | 2.074 | 3.712e-3 | 2.079 | 3.829e-3 | 2.072 | 3.615e-3 | 2.082 |
| | 2/64 | 1.032e-3 | 2.014 | 8.843e-4 | 2.062 | 9.584e-4 | 1.998 | 8.628e-4 | 2.067 |
| (0.9,1.9) | 2/8 | 6.996e-2 | – | 7.062e-2 | – | 6.693e-2 | – | 7.553e-2 | – |
| | 2/16 | 1.706e-2 | 2.036 | 1.714e-2 | 2.042 | 1.630e-2 | 2.037 | 1.639e-2 | 2.043 |
| | 2/32 | 4.182e-3 | 2.027 | 4.180e-3 | 2.036 | 3.988e-3 | 2.031 | 3.987e-3 | 2.039 |
| | 2/64 | 1.035e-3 | 2.012 | 1.028e-3 | 2.023 | 9.792e-4 | 2.026 | 9.727e-4 | 2.035 |

**TABLE 3**  Numerical comparisons of the direct, iterative, and preconditioned iterative methods for solving Example 1 with $\tau = 2^{-12}$, $b = 1.0$, and $p = 0.7$

| $(\gamma, \alpha)$ | $N$ | Direct | Noprec | Banded($\ell = 8$) | Skew-cir |
|-------------------|-----|--------|--------|--------------------|---------|
| $\tau$ | | CPU(s) | Iter | CPU(s) | Iter | CPU(s) | Iter | CPU(s) |
| (0.2,1.1) | 128 | 25.895 | 225.0 | 84.890 | 6.0 | 23.462 | 13.8 | 26.027 |
| | 256 | 55.777 | 994.5 | 893.026 | 7.3 | 32.111 | 14.8 | 31.613 |
| | 512 | 249.966 | 1926 | 311449 | 11.7 | 23.111 | 14.8 | 31.613 |
| | 1024 | 1859.940 | 4102.047 | 77.540 |
| (0.5,1.5) | 128 | 25.666 | 65.0 | 38.702 | 6.0 | 23.111 | 14.8 | 26.180 |
| | 256 | 52.250 | 127.6 | 68.884 | 8.0 | 32.794 | 14.8 | 31.585 |
| | 512 | 249.797 | 261.8 | 311449 | 11.7 | 23.111 | 14.8 | 31.613 |
| | 1024 | 1836.925 | 4102.047 | 77.540 |
| (0.9,1.9) | 128 | 24.118 | 39.0 | 29.815 | 6.0 | 23.111 | 14.8 | 26.180 |
| | 256 | 52.021 | 71.8 | 49.841 | 8.0 | 23.111 | 14.8 | 31.585 |
| | 512 | 260.412 | 147.4 | 183.047 | 11.7 | 23.111 | 14.8 | 31.613 |
| | 1024 | 1835.804 | 4102.047 | 77.540 |

$L_2$-norm error as the size of time steps of our IDS is decreased, and thus the convergence order in space is $O(h^2)$. In conclusion, the numerical convergence orders are consistent with the theoretical estimate $O(\tau^{2-\gamma} + h^2)$ presented in Section 2.2.

In Tables 3 and 4, the performance of the direct, iterative, and preconditioned iterative methods for Equation (3.3) are illustrated along with the elapsed CPU time and the average number of iterations. Here the symbols “Direct,” “Noprec,” “Banded($\ell = 8$),” and “Skew-cir” mean that the sequence of linear systems (3.3) is consecutively solved by using the MATLAB’s backslash operator or the BiCGSTAB routine available in MATLAB with no preconditioner, banded preconditioner and skew-circulant preconditioner, respectively. As can be seen from Tables 3 and 4, both banded and skew-circulant preconditioners are fairly efficient to accelerate the BiCGSTAB method for solving Equation (3.3) in terms of the elapsed CPU time and the number of iterations, especially when
TABLE 4 Numerical comparisons of the direct, iterative, and preconditioned iterative methods for solving Example 1 with $\tau = 2^{-12}$, $b = 2.0$, and $p = 0.3$

| $(\gamma, \alpha)$ | $N$  | Direct CPU | Noprec Iter | CPU | Banded($\phi = 8$) Iter | CPU | Skew-circ Iter | CPU |
|---------------------|------|------------|-------------|-----|------------------------|-----|----------------|-----|
| $(0.2, 1.1)$        | 128  | 26.045     | 204.3       | 80.795 | 5.5                     | 23.267 | 15.9          | 26.447 |
|                    | 256  | 54.872     | 981.7       | 1017.883 | 7.3                     | 30.687 | 15.9          | 32.356 |
|                    | 512  | 250.823    | $\dagger$   | $\dagger$ | 11.0                     | 57.626 | 15.8          | 57.388 |
|                    | 1024 | 1822.899   | $\dagger$   | $\dagger$ | 17.8                     | 107.799 | 16.6          | 77.719 |
| $(0.5, 1.5)$        | 128  | 26.470     | 60.0        | 36.100 | 5.4                     | 23.145 | 14.2          | 26.188 |
|                    | 256  | 54.736     | 123.1       | 67.480 | 7.8                     | 33.156 | 14.8          | 31.489 |
|                    | 512  | 260.543    | 246.8       | 293.753 | 11.3                     | 57.874 | 14.9          | 55.981 |
|                    | 1024 | 1824.693   | 502.1       | 658.004 | 18.9                     | 110.868 | 15.8          | 76.082 |
| $(0.9, 1.9)$        | 128  | 24.914     | 39.4        | 30.340 | 3.0                     | 22.314 | 11.9          | 24.145 |
|                    | 256  | 52.028     | 70.3        | 47.876 | 3.0                     | 27.789 | 13.6          | 30.749 |
|                    | 512  | 250.682    | 144.2       | 175.927 | 5.0                     | 42.018 | 14.9          | 55.671 |
|                    | 1024 | 1823.420   | 298.9       | 410.570 | 7.0                     | 70.534 | 15.9          | 75.984 |

the number of grid nodes increases. Moreover, we remark that $P_{sk}$ exhibits more robust performance than $P_b$ in terms of average number of iterations, that is, compared to BiCGSTAB with $P_b$, and the average of number of BiCGSTAB with $P_{sk}$ is weakly sensitive to the spatial grid size. In addition, it can be observed that the performance of BiCGSTAB with $P_{sk}$ becomes better when $\alpha = 1.9$, because the banded preconditioner has been proved to be very efficient for solving Equation (3.3), whose coefficient matrices are diagonally dominant—cf. Proposition 2.1 and Section 3 for a discussion. In conclusion, the skew-circulant preconditioner is still recommended for enhancing the convergence of BiCGSTAB applied to solve Equation (3.3), when $1 < \alpha < \alpha_0$, whereas the banded preconditioner is recommended if $\alpha \geq \alpha_0$, because the coefficient matrices are diagonally dominant—cf. Section 3.

Example 2 In this numerical example, we show the effect on the solution due to the presence of a singularity in the temporal derivative. For clarity, we assume that we can isolate a nonsmooth part from $u(x, t)$ as is shown in [1]. We suppose that the solution to the problem (1.1) with $f = 0$, $u(0, t) = u(1, t) = 0$, $u(x, 0) = 5x^3(1 - x)^3$, $\lambda(t) = e^{-bt}$, $\xi(x, t) \equiv \xi$ and $[x_L, x_R] \times [0, T] = [0, 1]^2$ has the following form

$$u(x, t) = 5x^3(1 - x)^3 \left[ 1 - \frac{\sqrt{t}}{\Gamma(1.5)} e^{-bt} \right] + v(x, t), \quad (4.1)$$

where $v(x, t)$ is the exact solution to the problem

$$C_0 D_t^{\gamma, \lambda(t)} v(x, t) = \xi [pD_t^\alpha v(x, t) + (1 - p)_t D_t^\alpha v(x, t)] + 5x^3(1 - x)^3 \left[ \frac{t^{1-\gamma}}{\Gamma\left(\frac{3}{2} - \gamma\right)} - \frac{bt^{3-\gamma}}{\Gamma\left(\frac{5}{2} - \gamma\right)} \right] e^{-bt}$$

$$+ 5\xi \cdot q(x) \left[ 1 - \frac{\sqrt{t}}{\Gamma(1.5)} e^{-bt} \right], \quad (x, t) \in (0, 1) \times (0, T), \quad (4.2)$$
along with the initial condition \( v(x, 0) = 0 \) (\( x \in [0, 1] \)), the boundary conditions \( v(0, t) = v(1, t) = 0 \) (\( t \in (0, T) \)), and with

\[
q(x) = \frac{\Gamma(4)}{\Gamma(4 - \alpha)} [px^{3-a} + (1-p)(1-x)^{3-a}] - \frac{3\Gamma(5)}{\Gamma(5 - \alpha)} [px^{4-a} + (1-p)(1-x)^{4-a}] + \frac{3\Gamma(6)}{\Gamma(6 - \alpha)} [px^{5-a} + (1-p)(1-x)^{5-a}] - \frac{\Gamma(7)}{\Gamma(7 - \alpha)} [px^{6-a} + (1-p)(1-x)^{6-a}]. \tag{4.3}
\]

Since \( u(x, t) \) is not available in this case, we estimate both the temporal and the spatial convergence orders of our difference scheme by computing the approximate solution on two different grids \( \mathcal{O}_{h_1} \) and \( \mathcal{O}_{h_2} \), where \( \mathcal{O}_{h_1} \subset \mathcal{O}_{h_2} \). The numerical calculations of \( u(x, t) \) by formula (4.1) are reported in Tables 5–8, where we focus on the convergence order of numerical solutions at the final time point.\(^2\)

As is seen from Table 5, in this example the temporal convergence order of our proposed difference scheme is almost 1 but is smaller than the theoretically estimated order—\(2 - \gamma\), except for the values \((\gamma, \alpha) = (0.9, 1.9)\). This can be explained by observing that we selected only the part of the solution that yields a singularity in the first derivative of \( u(x, t) \). This occurs when \( v_1(x, t) \) is continuous and \( v_2(x, t) \) has a singularity at the initial point \( t = 0 \). Moreover, Table 6 shows that the spatial convergence order can still reach the theoretical estimate \( O(h^2) \), especially in the \( L^2 \) norm, when both the solution and the initial data are sufficiently smooth in the space variable \( x \). In addition, the convergence order will be slightly better if \( b \) increases, because the larger value of \( b \) will make the solution \( u(x, t) \) behave more smoothly. We conclude that in cases when we do not have enough information on the smoothness of the solution, we can calculate the convergence order as suggested above. If the estimated convergence order is smaller than \( O(\tau^{2-\gamma} + h^2) \), then we should represent the solution as the sum of two functions, one of which is nonsmooth while the other is smooth but unknown [1]. Of course, finding such a representation may be difficult for the problem (1.1) with variable coefficients, and this is an interesting research issue in its own right.

---

\(^2\)It is worth noting that the results of our experiments showing that the temporal convergence order of our proposed scheme in the only \( L^2 \)-norm (which is the same as Table 5) in the whole domain is 1 but less than \(2 - \gamma\) still remains consistent with the stability analysis and error estimates based on the \( L^2 \)-norm introduced in Section 2.2.
TABLE 6  \(L_2\)-norm and maximum norm errors versus grid size reduction when \(\tau = 2^{-11}\) and \(p = 0.4\) in Example 2

| \((\gamma, \alpha)\) | \(h_1\) | \(h_2\) | \(b = 3.0\) | \(b = 4.0\) |
|------------------|------|------|-------------|-------------|
|                  |      |      | \(\text{Error}_\infty\) | \(\text{Rate}_\infty\) | \(\text{Error}_2\) | \(\text{Rate}_2\) | \(\text{Error}_\infty\) | \(\text{Rate}_\infty\) | \(\text{Error}_2\) | \(\text{Rate}_2\) |
| (0.2,1.2)        | 1/10 | 1/20 | 2.318e-3     | –            | 1.500e-3     | –            | 2.424e-3     | –            | 1.555e-3     | –            |
|                  | 1/20 | 1/40 | 5.285e-4     | 2.133        | 3.437e-4     | 2.126        | 5.517e-4     | 2.135        | 3.557e-4     | 2.128        |
|                  | 1/40 | 1/80 | 1.295e-4     | 2.029        | 8.427e-5     | 2.028        | 1.350e-4     | 2.031        | 8.687e-5     | 2.034        |
|                  | 1/80 | 1/160| 5.271e-5     | 1.297        | 2.141e-5     | 1.977        | 3.369e-5     | 2.003        | 2.167e-5     | 2.003        |
| (0.5,1.5)        | 1/10 | 1/20 | 2.003e-3     | –            | 1.274e-3     | –            | 2.084e-3     | –            | 1.324e-3     | –            |
|                  | 1/20 | 1/40 | 4.577e-4     | 2.130        | 2.949e-4     | 2.111        | 4.760e-4     | 2.130        | 3.063e-4     | 2.111        |
|                  | 1/40 | 1/80 | 1.122e-4     | 2.028        | 7.208e-5     | 2.033        | 1.167e-4     | 2.029        | 7.487e-5     | 2.033        |
|                  | 1/80 | 1/160| 2.799e-5     | 2.003        | 1.790e-5     | 2.009        | 2.908e-5     | 2.004        | 1.860e-5     | 2.009        |
| (0.9,1.9)        | 1/10 | 1/20 | 1.445e-3     | –            | 8.723e-4     | –            | 1.501e-3     | –            | 9.057e-4     | –            |
|                  | 1/20 | 1/40 | 3.256e-4     | 2.150        | 2.059e-4     | 2.083        | 3.382e-4     | 2.150        | 2.138e-4     | 2.083        |
|                  | 1/40 | 1/80 | 7.940e-5     | 2.036        | 5.069e-5     | 2.022        | 8.245e-5     | 2.036        | 5.262e-5     | 2.022        |
|                  | 1/80 | 1/160| 1.975e-5     | 2.007        | 1.262e-5     | 2.006        | 2.051e-5     | 2.007        | 1.310e-5     | 2.006        |

Tables 7 and 8 show the elapsed CPU time, the number of iterations and the memory cost (measured in megabytes) required by different solution techniques for solving the class (i) of problem (1.1) described in Section 3. Overall, the performance of the BiCGSTAB method with skew-circulant preconditioner are the best in terms of elapsed CPU time and memory cost, the banded preconditioner being a good alternative. However, the direct method (based on only one LU decomposition) is non-competitive due to the large elapsed CPU time and memory costs, especially for the fine discretized meshes, because the large dense matrix and its LU decomposition factors need to be explicitly stored.

In addition, according to numerical results of Examples 1 and 2, it is interesting to observe that, although we employ the fast preconditioned BiCGSTAB method to solve Equation (3.3) corresponding to the IDS (2.3), the total CPU time is still high. In fact, the solution time comes from two main computations: (a) solving the sequence of linear systems (3.3); (b) evaluating the right-hand
side vector of (3.3) by repeatedly summing the solutions at previous time levels. Our precondi-
tioned BiCGSTAB method can only alleviate the first cost, while we should analyze further the
degradation of CPU time due to handling the nonlocal property of the discrete temporal fractional
derivative. However, such analysis is always difficult in the general case of $\lambda(t)$. In particular, if we
set $\lambda(t) = e^{-bt}$ like in Examples 1 and 2, we can further alleviate the computational and memory
cost of the proposed IDS. The derivation of such a more cost-effective scheme (A.5) is presented
in Appendix.

5 | CONCLUSIONS

In this article, the stability and convergence of an IDS scheme for solving the GTSFDEs with vari-
able coefficients are studied via the diagonally weighted energy norm analysis. The proposed IDS can
be proved to reach second order convergence in space and $(2^{\gamma} - $)-th approximation order in time for
the GTSFDEs with variable coefficients. Moreover, numerical experiments involving problem (1.1)
with nonsmooth solution are carried out yielding results completely in line with our theoretical
analysis. The method can be easily extended to solve the variable coefficient GTSFDEs with other
boundary conditions. Although the focus of the article is on the case of one-dimensional spatial
domains, the results can be extended to two- and three-dimensional domain; refer, for example,
to [55].

In addition, we have also shown an efficient implementation of the proposed IDS based on pre-
conditioned iterative solvers, achieving about $O(N \log N)$ computational complexity and $O(N)$ storage
cost. Numerical evidence of the efficiency of the proposed preconditioning methods is reported.
For the special choice of $\lambda(t) = e^{-bt}$, the fast sum-of-exponential approximations of the kernel in
(1.2) can be used to derive a cost-effective version of IDS (A.5); then, numerical experiments are
illustrated to show that the rate of the truncation error of this new IDS is about $O(\tau^{2-\gamma} + h^2)$. However, its rigorous
stability and convergence analyses remain an open question. Meanwhile, numerical results show the fast IDS (A.5) requires less CPU time and memory cost than the proposed IDS (2.3).
ACKNOWLEDGMENTS
The authors are grateful to Prof. Jiwei Zhang and Dr. Hong-Lin Liao for their constructive discussions and insightful comments. This research is supported by NSFC (11801463 and 61772003), the Applied Basic Research Project of Sichuan Province (2020YJ0007), the Fundamental Research Funds for the Central Universities (JBK1902028), and the Ministry of Education of Humanities and Social Science Layout Project (19JYA790094). Meanwhile, the first author would like to thank Prof. Hai-Wei Sun and Dr. Siu-Long Lei for their helpful discussions during his visiting to the University of Macau.

CONFLICT OF INTEREST
The authors declared that they have no conflicts of interest to this work.

AUTHOR CONTRIBUTIONS
All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication. More precisely, Xian-Ming Gu: writing—original draft preparation, design, analysis, methodology; Ting-Zhu Huang: conceptualization, supervision, data curation; Yong-Liang Zhao: coding, performance analysis, software visualization; Pin Lyu: design, analysis, methodology; Bruno Carpentieri: writing—reviewing, revising, editing.

ORCID
Xian-Ming Gu https://orcid.org/0000-0001-7895-2050

REFERENCES
[1] A. A. Alikhanov, A time-fractional diffusion equation with generalized memory kernel in differential and difference settings with smooth solutions, Comput. Methods Appl. Math. 17 (2017), 647–660.
[2] M. Chen and W. Deng, High order algorithm for the time-tempered fractional Feynman–Kac equation, J. Sci. Comput. 76 (2018), 867–887.
[3] G. Chi et al., Numerical solution to the space-time fractional diffusion equation and inversion for the space-dependent diffusion coefficient, J. Comput. Theor. Transp. 46 (2017), 122–146.
[4] W. Deng et al., Boundary problems for the fractional and tempered fractional operators, Multiscale Model. Simul. 16 (2018), 125–149.
[5] W. Deng and Z. Zhang, High accuracy algorithms for the differential equations governing anomalous diffusion: Algorithm and models for anomalous diffusion, World Scientific Publishing, Singapore, 2019.
[6] E. H. Doha, A. H. Bhrawy, and S. S. Ezz-Eldien, Numerical approximations for fractional diffusion equations via a Chebyshev spectral-tau method, Cent. Eur. J. Phys. 11 (2013), 1494–1503.
[7] M. Donatelli, M. Mazza, and S. Serra-Capizzano, Spectral analysis and structure preserving preconditioners for fractional diffusion equations, J. Comput. Phys. 307 (2016), 262–279.
[8] L. Feng et al., Second-order approximation for the space fractional diffusion equation with variable coefficient, Progr. Fract. Differ. Appl. 1 (2015), 23–35.
[9] L. B. Feng et al., Stability and convergence of a new finite volume method for a two-sided space-fractional diffusion equation, Appl. Math. Comput. 257 (2015), 52–65.
[10] M. A. Firoozjaee, S. A. Yousefi, and H. Jafari, A numerical approach to Fokker–Planck equation with space- and time-fractional and non fractional derivatives, Commun. Math. Comput. Chem. 74 (2015), 449–464.
[11] H. Fu and H. Wang, A preconditioned fast parareal finite difference method for space-time fractional partial differential equation, J. Sci. Comput. 78 (2019), 1724–1743.
[12] G.-H. Gao, A. A. Alikhanov, and Z.-Z. Sun, The temporal second order difference schemes based on the interpolation approximation for solving the time multi-term and distributed-order fractional sub-diffusion equations, J. Sci. Comput. 73 (2017), 93–121.
[13] I. Gohberg and V. Olshevsky, Circulants, displacements and decompositions of matrices, Integr. Equ. Oper. Theory 15 (1992), 730–743.
Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations.

J. Pan et al., M. K. Ng, Fractional differential equations, Mathematics in Science and Engineering, Academic Press, 2014.

Z.-P. Hao, Z.-Z. Sun, and W.-R. Cao, A fourth-order approximation of fractional derivatives with its applications, J. Comput. Phys. 281 (2015), 787–805.

X.-M. Gu et al., Fast iterative method with a second-order implicit difference scheme for time-space fractional convection-diffusion equation, J. Sci. Comput. 72 (2017), 957–985.

F.-R. Lin and W.-D. Liu, A circulant preconditioner for fractional diffusion equations, J. Comput. Phys. 242 (2013), 715–725.

S.-L. Lei and H.-W. Sun, A circulant preconditioner for fractional diffusion equations, J. Comput. Phys. 242 (2013), 715–725.
How to cite this article: Gu X-M, Huang T-Z, Zhao Y-L, Lyu P, Carpentieri B. A fast implicit difference scheme for solving the generalized time–space fractional diffusion equations with variable coefficients. Numer Methods Partial Differential Eq. 2021;37:1136–1162. https://doi.org/10.1002/num.22571

APPENDIX A: FAST SOE APPROXIMATION OF THE GENERALIZED CAPUTO FRACTIONAL DERIVATIVE

Due to the nonlocality of the generalized Caputo fractional derivative (1.2), the proposed scheme (3.3) requires the storage of the solution at all previous time steps which leads to huge computational cost. This phenomenon also can be observed from the numerical experiments reported in Section 4. To reduce the computational cost, we follow the work about fast $L^1$ formula [20] for

[44] W. Qu, S.-L. Lei, and S.-W. Vong, A note on the stability of a second order finite difference scheme for space fractional diffusion equations, Numer. Algebra Control Optim. 4 (2014), 317–325. https://doi.org/10.3934/naco.2014.4.317.
[45] R.-F. Ren et al., An efficient Chebyshev-tau method for solving the space fractional diffusion equations, Appl. Math. Comput. 224 (2013), 259–267.
[46] F. Sabzikar, M. M. Meerschaert, and J. Chen, Tempered fractional calculus, J. Comput. Phys. 293 (2015), 14–28.
[47] S. G. Samko, A. A. Kilbas, and O. I. Marichev, Fractional integrals and derivatives: Theory and applications, Gordon and Breach Science Publishers, Yverdonn, Switzerland, 1993.
[48] T. Sandev et al., Diffusion and Fokker–Planck–Smoluchowski equations with generalized memory kernel, Fract. Calc. Appl. Anal. 18 (2015), 1006–1038.
[49] M. A. F. Santos, Mittag-Leffler memory kernel in Lévy flights, Mathematics 7 (2019), 766. https://doi.org/10.3390/math7090766.
[50] E. Sousa and C. Li, A weighted finite difference method for the fractional diffusion equation based on the Riemann–Liouville derivative, Appl. Numer. Math. 90 (2015), 22–37.
[51] M. Stynes, E. O’Riordan, and J. L. Gracia, Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation, SIAM J. Numer. Anal. 55 (2017), 1057–1079.
[52] C. Tadjeran, M. M. Meerschaert, and H.-P. Scheffler, A second-order accurate numerical approximation for the fractional diffusion equation, J. Comput. Phys. 213 (2006), 205–213.
[53] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992), 631–644.
[54] S. Vong and P. Lyu, On a second order scheme for space fractional diffusion equations with variable coefficients, Appl. Numer. Math. 137 (2019), 34–48.
[55] S. Vong et al., High order finite difference method for time-space fractional differential equations with Caputo and Riemann–Liouville derivatives, Numer. Algorithms 72 (2016), 195–210.
[56] X. Wu, W. Deng, and E. Barkai, Tempered fractional Feynman–Kac equation: Theory and examples, Phys. Rev. E 93 (2016), 032151. https://doi.org/10.1103/PhysRevE.93.032151.
[57] Y. Xu, Z. He, and O. P. Agrawal, Numerical and analytical solutions of new generalized fractional diffusion equation, Comput. Math. Appl. 66 (2013), 2019–2029.
[58] Y. Zhang, D. A. Benson, and D. M. Reeves, Time and space nonlocalities underlying fractional-derivative models: Distinction and literature review of field applications, Adv. Water Resour. 32 (2009), 561–581.
[59] Z. Zhang and W. Deng, Numerical approaches to the functional distribution of anomalous diffusion with both traps and flights, Adv. Comput. Math. 43 (2017), 699–732.
[60] Y.-L. Zhao et al., A limited-memory block bi-diagonal Toeplitz preconditioner for block lower triangular Toeplitz system from time-space fractional diffusion equation, J. Comput. Appl. Math. 362 (2019), 99–115.
[61] Z. Zhao, X.-Q. Jin, and M. M. Lin, Preconditioned iterative methods for space-time fractional advection-diffusion equations, J. Comput. Phys. 319 (2016), 266–279.
[62] X. Zheng, V. J. Ervin, and H. Wang, Spectral approximation of a variable coefficient fractional diffusion equation in one space dimension, Appl. Math. Comput. 361 (2019), 98–111.
developing the SOE approximation of the generalized Caputo fractional derivative with \( \lambda(t) = e^{-bt} \), which is adopted in Section 4. More precisely,

\[
C_0 D_t^{\lambda(t)} u(t)|_{t=t_j} = \frac{1}{\Gamma(1 - \gamma)} \int_{0}^{t_j} e^{-b(t_j-s)} u'(s) ds (t_j - s)^\gamma \\
= \frac{1}{\Gamma(1 - \gamma)} \int_{t_{j-1}}^{t_j} e^{-b(t_j-s)} u'(s) ds (t_j - s)^\gamma + \frac{1}{\Gamma(1 - \gamma)} \int_{0}^{t_{j-1}} e^{-b(t_j-s)} u'(s) ds (t_j - s)^\gamma \\
= C_l(t_j) + C_h(t_j),
\]

where the last equality defines the local part and the history part, respectively. For the local part, we employ the generalized \( L_1 \) approximation recalled in Section 2.1, which approximates \( u(s) \) on \([t_{j-1}, t_j] \) via a linear polynomial (with \( u(t_{j-1}) \) and \( u(t_j) \) as the interpolation nodes) or \( u'(s) \) via a constant \( \frac{u(t_j) - u(t_{j-1})}{\tau} \). We have

\[
C_l(t_j) \approx \frac{u(t_j) - u(t_{j-1})}{\tau \Gamma(1 - \gamma)} \int_{t_{j-1}}^{t_j} e^{-b(t_j-s)} ds (t_j - s)^\gamma \\
= \frac{u(t_j) - u(t_{j-1})}{\tau \Gamma(2 - \gamma)} \left( e^{-bt_j} t_j^{1-\gamma} + b \int_{0}^{\tau} e^{-b\theta} \theta^{1-\gamma} d\theta \right), \tag{A.1}
\]

where the second integral can be evaluated via the MATLAB built-in function “integral.m.” For the history part, we first recall the following lemma \[20\] to approximate the history part \( C_h(t_j) \).

**Lemma A.1** Let \( \epsilon \) denote tolerance error, \( \delta \) cut-off time restriction and \( T \) final time. Then there are a natural number \( N_{exp} \) and positive numbers \( s_k \) and \( w_k \), \( k = 1, 2, \ldots, N_{exp} \), such that

\[
\left| \frac{1}{\tau} - \sum_{k=1}^{N_{exp}} w_k e^{-s_k \tau} \right| < \epsilon, \quad \text{for any } t \in [\delta, T],
\]

where \( N_{exp} = \mathcal{O}((\log \epsilon^{-1})(\log \log \epsilon^{-1} + \log(T\delta^{-1})) + (\log \delta^{-1})(\log \log \epsilon^{-1} + \log \delta^{-1})) \).

Therefore, when we set \( \delta = \tau \) and apply Lemma A.1, then we obtain

\[
C_h(t_j) \approx \frac{1}{\Gamma(1 - \gamma)} \int_{0}^{t_{j-1}} e^{-b(t_{j-1}-s)} \sum_{k=1}^{N_{exp}} w_k e^{-s_k (t_{j-1}-s)} u'(s) ds \\
\triangleq \frac{1}{\Gamma(1 - \gamma)} \sum_{k=1}^{N_{exp}} \int_{0}^{t_{j-1}} w_k e^{-\tilde{s}_k (t_{j-1}-s)} u'(s) ds \\
\triangleq \frac{1}{\Gamma(1 - \gamma)} \sum_{k=1}^{N_{exp}} w_k U_{hist,k}(t_j) \\
= \frac{1}{\Gamma(1 - \gamma)} \sum_{k=1}^{N_{exp}} w_k \left[ e^{-\tilde{s}_k \tau} U_{hist,k}(t_{j-1}) + \int_{t_{j-2}}^{t_j} e^{-\tilde{s}_k (t_j-s)} u'(s) ds \right], \tag{A.2}
\]

where \( \tilde{s}_k = s_k + b \). To evaluate \( U_{hist,k}(t_j) \) for \( j = 1, 2, \ldots, N_{exp} \), it observes the following simple recurrence relation:

\[
U_{hist,k}(t_j) = e^{-\tilde{s}_j \tau} U_{hist,k}(t_{j-1}) + \int_{t_{j-2}}^{t_{j-1}} e^{-\tilde{s}_k (t_j-s)} u'(s) ds
\]
fractional derivative by our fast evaluation scheme. It can be seen from Table A.1 that when difference schemes for the variable coefficient GTSFDEs with different (levels) and inverting the linear system have At each time step, the scheme of the following form

\[ u_{\text{hist}, k(t_j)} = e^{-\frac{\tau}{\gamma}k\tau} U_{\text{hist}, k(t_j)} + \frac{u(t_{j-1}) - u(t_{j-2})}{\tau} \int_{t_{j-2}}^{t_{j-1}} e^{-\frac{\tau}{\gamma}s} ds \]

\[ \approx e^{-\frac{\tau}{\gamma}k\tau} U_{\text{hist}, k(t_j)} + \frac{u(t_{j-1}) - u(t_{j-2})}{\tau} \int_{t_{j-2}}^{t_{j-1}} e^{-\frac{\tau}{\gamma}s} ds \]

Noting that \( U_{\text{hist}, k(t_1)} \equiv 0 \) when \( n = 1 \), we have

\[ \frac{\partial^\gamma}{\partial t^\gamma} u^1 = \frac{u(t_1) - u(t_0)}{\tau} \left( e^{-\beta \tau^{1-\gamma}} + b \int_0^\tau e^{-\beta \theta^{1-\gamma}} d\theta \right) \]

where we define

\[ \frac{\partial^\gamma}{\partial t^\gamma} u^1 = \frac{u(t_1) - u(t_{j-1})}{\tau} \left( e^{-\beta \tau^{1-\gamma}} + b \int_0^\tau e^{-\beta \theta^{1-\gamma}} d\theta \right) + \frac{1}{\Gamma(1-\gamma)} \sum_{k=1}^{N_{\text{exp}}} \omega_k U_{\text{hist}, k(t)} \]

as the approximate discrete operator for evaluating the fractional derivative quickly and \( U_{\text{hist}, k(t_j)} \) can be computed via Equation (A.3). At each time step, we only need \( O(1) \) work to compute \( U_{\text{hist}, k(t_j)} \) since \( U_{\text{hist}, k(t_{j-1})} \) is known at that point. Thus, the total work is reduced from \( O(M^2) \) to \( O(MN_{\text{exp}}) \), and the total memory requirement is reduced from \( O(M) \) to \( O(N_{\text{exp}}) \).

Similar to [20], replacing the \( L^1 \)-type approximation (cf. Lemma 2.1) for the generalized Caputo fractional derivative by our fast evaluation scheme, we obtain a novel implicit difference scheme of the following form

\[
\begin{align*}
F_0^C D_t^{\gamma, \lambda(t)} u^1_i & = \frac{u(t_{j+1}) - u(t_j)}{\tau} \left( e^{-\beta \tau^{1-\gamma}} + b \int_0^\tau e^{-\beta \theta^{1-\gamma}} d\theta \right) \\
& + \frac{1}{\Gamma(1-\gamma)} \sum_{k=1}^{N_{\text{exp}}} \omega_k U_{\text{hist}, k(t)}
\end{align*}
\]

which nearly reaches the approximation order of \( O(\tau^{2-\gamma} + h^2) \); see numerical results in the next context. At each time step \( t_{j+1} \), evaluating the right hand side (i.e., the known solutions at the previous time levels) and inverting the linear system have \( O(NN_{\text{exp}}) \) and \( O(I_{\text{avg}} N \log N) \) computational complexity, respectively, which leads to an overall computational complexity of \( O(MN(N_{\text{exp}} + I_{\text{avg}} \log N)) \), where \( I_{\text{avg}} \approx N \) is the average number of iterations required for solving the resulting linear system at each time step. By contrast, if we use the Gaussian elimination method to solve the resulting linear systems of Equation (3.3), the overall computational complexity of the implicit difference scheme (2.3) is about \( O(MN^3 + M^2 N) \) operations. In addition, it is meaningful to note that the above fast difference scheme has an overwhelming advantage when the number of temporal discretization steps (i.e., \( M \)) is relatively large.

**Example A.1** In this example, we test the fast difference scheme (A.5) and the direct difference scheme (2.3) for solving the same model problem in Example 1 except different diffusion coefficient \( \xi(x, t) = 10(1/2 + x^2 + \sin t) \). Let the tolerance error \( \epsilon = 10^{-9} \) for fast difference scheme (A.5) and Tables A.1 and A.2 are reported to evaluate the accuracy and efficiency of the proposed algorithms.

Tables A.1 and A.2 illustrate the temporal/spatial errors, convergence orders and CPU time of the methods. It can be seen from Table A.1 that when \( \tau = 2^{-11} \), both “Error\(_{\infty}\)” and “Error\(_2\)” of two implicit difference schemes for the variable coefficient GTSFDEs with different \((\gamma, \alpha, b)\)’s decreases steadily.

\[ 3 \text{In our experiments, it always finds that } N_{\text{exp}} < 80. \]
TABLE A.1  \(L_2\)-norm and maximum norm errors versus grid size reduction when \(\tau = 2^{-11}\) and \(p = 0.7\) in Example A.1

| \((y, a, b)\) | \(N\) | Direct scheme (2.3) | | | | | Fast scheme (A.5) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | \(\text{Error}_\infty\) | \(\text{Rate}_\infty\) | \(\text{Error}_2\) | \(\text{Rate}_2\) | CPU(s) | \(\text{Error}_\infty\) | \(\text{Rate}_\infty\) | \(\text{Error}_2\) | \(\text{Rate}_2\) | CPU(s) |
| \((0.2, 1.1, 1.0)\) | 10 | 7.3589e-2 | – | 7.0444e-2 | – | 3.578 | 7.3581e-2 | – | 7.0438e-2 | – | 0.335 |
| | 20 | 1.7410e-2 | 2.0796 | 1.7101e-2 | 2.0424 | 3.913 | 1.7404e-2 | 2.0799 | 1.7095e-2 | 2.0428 | 0.408 |
| | 40 | 4.1567e-3 | 2.0664 | 4.1035e-3 | 2.0592 | 4.159 | 4.1515e-3 | 2.0677 | 4.0983e-3 | 2.0605 | 0.469 |
| | 80 | 1.1354e-3 | 1.8722 | 9.8777e-4 | 2.0546 | 5.910 | 1.1443e-3 | 1.8592 | 9.8257e-4 | 2.0604 | 1.157 |
| \((0.5, 1.5, 1.0)\) | 10 | 4.8279e-2 | – | 4.6207e-2 | – | 3.586 | 4.8274e-2 | – | 4.6202e-2 | – | 0.343 |
| | 20 | 1.1381e-2 | 2.0848 | 1.0791e-2 | 2.0983 | 3.967 | 1.1377e-2 | 2.0851 | 1.0787e-2 | 2.0987 | 0.417 |
| | 40 | 2.7033e-3 | 2.0738 | 2.5503e-3 | 2.0811 | 4.278 | 2.6990e-3 | 2.0756 | 2.5463e-3 | 2.0828 | 0.482 |
| | 80 | 6.7243e-4 | 2.0073 | 6.0900e-4 | 2.0662 | 5.935 | 6.7315e-4 | 2.0034 | 6.0516e-4 | 2.0730 | 1.161 |
| \((0.9, 1.9, 1.0)\) | 10 | 4.6595e-2 | – | 4.6972e-2 | – | 3.584 | 4.6593e-2 | – | 4.6969e-2 | – | 0.328 |
| | 20 | 1.1365e-2 | 2.0356 | 1.1402e-2 | 2.0425 | 3.966 | 1.1363e-2 | 2.0358 | 1.1400e-2 | 2.0427 | 0.396 |
| | 40 | 2.7816e-3 | 2.0306 | 2.7753e-3 | 2.0386 | 4.197 | 2.7797e-3 | 2.0313 | 2.7734e-3 | 2.0393 | 0.473 |
| | 80 | 6.8243e-4 | 2.0272 | 6.7697e-4 | 2.0355 | 5.857 | 6.8051e-4 | 2.0302 | 6.7506e-4 | 2.0386 | 1.149 |
| \((0.2, 1.1, 2.0)\) | 10 | 6.9685e-2 | – | 6.6419e-2 | – | 3.285 | 6.9679e-2 | – | 6.6415e-2 | – | 0.324 |
| | 20 | 1.6438e-2 | 2.0838 | 1.6132e-2 | 2.0417 | 3.652 | 1.6432e-2 | 2.0840 | 1.6128e-2 | 2.0419 | 0.373 |
| | 40 | 3.9206e-3 | 2.0679 | 3.8721e-3 | 2.0587 | 3.938 | 3.9167e-3 | 2.0690 | 3.8682e-3 | 2.0598 | 0.461 |
| | 80 | 1.0747e-3 | 1.8671 | 9.3218e-4 | 2.0544 | 5.597 | 1.0747e-3 | 1.8657 | 9.2820e-4 | 2.0592 | 1.112 |
| \((0.5, 1.5, 2.0)\) | 10 | 4.5268e-2 | – | 4.3322e-2 | – | 3.265 | 4.5265e-2 | – | 4.3318e-2 | – | 0.321 |
| | 20 | 1.0665e-2 | 2.0856 | 1.0114e-2 | 2.0987 | 3.512 | 1.0662e-2 | 2.0859 | 1.0110e-2 | 2.0992 | 0.369 |
| | 40 | 2.5332e-3 | 2.0739 | 2.3895e-3 | 2.0816 | 3.898 | 2.5300e-3 | 2.0753 | 2.3865e-3 | 2.0828 | 0.458 |
| | 80 | 6.2593e-4 | 2.0169 | 5.7014e-4 | 2.0673 | 5.697 | 6.2648e-4 | 2.0138 | 5.6752e-4 | 2.0722 | 1.116 |
| \((0.9, 1.9, 2.0)\) | 10 | 4.3556e-2 | – | 4.3908e-2 | – | 3.298 | 4.3554e-2 | – | 4.3906e-2 | – | 0.319 |
| | 20 | 1.0623e-2 | 2.0357 | 1.0657e-2 | 2.0427 | 3.557 | 1.0621e-2 | 2.0359 | 1.0656e-2 | 2.0428 | 0.372 |
| | 40 | 2.5992e-3 | 2.0311 | 2.5935e-3 | 2.0388 | 3.935 | 2.5978e-3 | 2.0316 | 2.5921e-3 | 2.0395 | 0.463 |
| | 80 | 6.3724e-4 | 2.0282 | 6.3217e-4 | 2.0365 | 5.713 | 6.3580e-4 | 2.0306 | 6.3074e-4 | 2.0390 | 1.126 |
TABLE A.2  \( L_2 \)-norm, maximum norm errors and CPU time (in seconds) versus grid size reduction when \( N = \lceil 2M^{2-\gamma/2} \rceil \) and \( p = 0.7 \) in Example A.1

| \((\gamma, \alpha, b)\) | \( M \) | \( (r, \alpha, b) \) | Direct scheme (2.3) | Fast scheme (A.5) |
|------------------------|-------|------------------|-------------------|-------------------|
|                        |       | \( Error_\infty \) | \( Rate_\infty \) | \( Error_2 \) | \( Rate_2 \) | \( CPU(s) \) | \( Error_\infty \) | \( Rate_\infty \) | \( Error_2 \) | \( Rate_2 \) | \( CPU(s) \) |
| \((0.2,1.1,1.0)\)      | 2\(^5\) | 3.2759e-3 | – | 3.2241e-3 | – | 0.008 | 3.7666e-3 | – | 2.9594e-3 | – | 0.007 |
|                        | 2\(^6\) | 1.0351e-3 | 1.6621 | 8.9483e-4 | 1.8492 | 0.029 | 1.2961e-3 | 1.5391 | 7.6594e-4 | 1.9500 | 0.031 |
|                        | 2\(^7\) | 3.1910e-4 | 1.6977 | 2.4837e-4 | 1.8491 | 0.264 | 4.4060e-4 | 1.5566 | 2.0299e-4 | 1.9158 | 0.254 |
|                        | 2\(^8\) | 9.5866e-5 | 1.7349 | 6.8823e-5 | 1.8515 | 2.873 | 1.5160e-4 | 1.5392 | 5.8937e-5 | 1.7842 | 2.859 |
| \((0.5,1.5,1.0)\)      | 2\(^5\) | 2.1236e-3 | – | 2.0025e-3 | – | 0.012 | 2.0134e-3 | – | 1.9017e-3 | – | 0.013 |
|                        | 2\(^7\) | 7.4051e-4 | 1.5199 | 6.7840e-4 | 1.5616 | 0.052 | 7.5084e-4 | 1.4231 | 6.2573e-4 | 1.6037 | 0.053 |
|                        | 2\(^8\) | 2.7349e-4 | 1.4370 | 2.3228e-4 | 1.5463 | 0.356 | 2.7741e-4 | 1.4365 | 2.0565e-4 | 1.6053 | 0.334 |
|                        | 2\(^9\) | 9.9638e-5 | 1.4567 | 8.0300e-5 | 1.5324 | 2.703 | 1.0111e-4 | 1.4561 | 6.7254e-5 | 1.6125 | 2.442 |
| \((0.9,1,1.0)\)        | 2\(^7\) | 5.7500e-3 | – | 5.7539e-3 | – | 0.019 | 5.7304e-3 | – | 5.7345e-3 | – | 0.021 |
|                        | 2\(^8\) | 2.5255e-3 | 1.1870 | 2.5185e-3 | 1.1920 | 0.078 | 2.5143e-3 | 1.1885 | 2.5073e-3 | 1.1935 | 0.054 |
|                        | 2\(^9\) | 1.1846e-3 | 1.0922 | 1.1776e-3 | 1.0967 | 0.303 | 1.1782e-3 | 1.0936 | 1.1713e-3 | 1.0980 | 0.161 |
|                        | 2\(^{10}\) | 5.3828e-4 | 1.1380 | 5.3354e-4 | 1.1422 | 1.776 | 5.3475e-4 | 1.1396 | 5.3004e-4 | 1.1439 | 0.705 |
| \((0.2,1,1.2)\)        | 2\(^5\) | 3.1058e-3 | – | 3.0955e-3 | – | 0.007 | 3.3736e-3 | – | 2.8288e-3 | – | 0.008 |
|                        | 2\(^6\) | 9.7378e-4 | 1.6733 | 8.4355e-4 | 1.8493 | 0.028 | 1.1535e-3 | 1.5483 | 7.4229e-4 | 1.9301 | 0.032 |
|                        | 2\(^7\) | 2.9435e-4 | 1.7261 | 2.3410e-4 | 1.8493 | 0.261 | 3.8784e-4 | 1.5725 | 2.0336e-4 | 1.8679 | 0.255 |
|                        | 2\(^8\) | 8.8505e-5 | 1.7337 | 6.4856e-5 | 1.8518 | 2.901 | 1.3145e-4 | 1.5609 | 5.5811e-5 | 1.8654 | 2.861 |
| \((0.5,1.5,2.0)\)      | 2\(^5\) | 1.9855e-3 | – | 1.8722e-3 | – | 0.011 | 1.9019e-3 | – | 1.7954e-3 | – | 0.012 |
|                        | 2\(^7\) | 6.8955e-4 | 1.5258 | 6.3406e-4 | 1.5620 | 0.053 | 6.9740e-4 | 1.4474 | 5.9410e-4 | 1.5955 | 0.054 |
|                        | 2\(^8\) | 2.5469e-4 | 1.4369 | 2.1704e-4 | 1.5467 | 0.360 | 2.5766e-4 | 1.4365 | 1.9681e-4 | 1.5939 | 0.255 |
|                        | 2\(^9\) | 9.2793e-5 | 1.4567 | 7.5014e-5 | 1.5327 | 2.699 | 9.3904e-5 | 1.4562 | 6.6809e-5 | 1.5587 | 2.434 |
| \((0.9,1.9,2.0)\)      | 2\(^7\) | 5.3618e-3 | – | 5.3656e-3 | – | 0.020 | 5.3470e-3 | – | 5.3510e-3 | – | 0.021 |
|                        | 2\(^8\) | 2.3546e-3 | 1.1872 | 2.3482e-3 | 1.1922 | 0.077 | 2.3462e-3 | 1.1884 | 2.3398e-3 | 1.1934 | 0.055 |
|                        | 2\(^9\) | 1.0144e-3 | 1.0922 | 1.0980e-3 | 1.0967 | 0.302 | 1.0996e-3 | 1.0933 | 1.0933e-3 | 1.0977 | 0.255 |
|                        | 2\(^{10}\) | 5.0182e-4 | 1.1380 | 4.9742e-4 | 1.1423 | 1.769 | 4.9917e-4 | 1.1394 | 4.9480e-4 | 1.1438 | 0.694 |
for smaller $h$, and the order of accuracy in space is about two. Fixing $N = \lceil 2M^{(2 - \gamma)/2} \rceil$, Table A.2 lists the maximum-norm and $L_2$-norm errors and illustrates that the order of temporal accuracy is of $(2 - \gamma)$. Therefore, Tables A.1 and A.2 confirm that the rate of the truncation errors of numerical schemes (2.3) and (A.5) is $\mathcal{O}(\tau^{2 - \gamma} + h^2)$. However, it seems that the temporal errors based on the maximum norm of fast scheme (A.5) change slightly irregularly compared to those of the direct scheme (2.3), especially for the case of $(0.2, 1.1, 2.0)$. Moreover, the fast scheme (A.5) requires less CPU time than the direct scheme (2.3) for the variable-coefficient GTSFDEs with different $(\gamma, \alpha, b)$'s. The time reduction between the direct scheme (2.3) and the fast scheme (A.5) shown in Table A.2 is not distinct, because the number of temporal discretization steps is less than the size of spatially discretized linear systems which are preponderantly time-consuming. In conclusion, although the derived fast scheme (A.5)\(^4\) needs less CPU time and memory cost than the direct scheme (2.3). Further analysis is still required to assess its stability and convergence properties.

\(^4\)In fact, the above fast scheme can easily utilize the non-uniform temporal steps [26, 51], which can enhance its (numerical) temporal convergence order for solving the variable-coefficient GTSFDEs (even with the weak singularity at initial time). MATLAB codes of all the numerical tests is available from the authors’ emails.