J Sci Comput (2017) 72:957–985
DOI 10.1007/s10915-017-0388-9

Fast Iterative Method with a Second-Order Implicit Difference Scheme for Time-Space Fractional Convection–Diffusion Equation

Xian-Ming Gu[1,2] · **Ting-Zhu Huang**[1] · **Cui-Cui Ji**[3] · **Bruno Carpentieri**[4] · **Anatoly A. Alikhanov**[5]

Received: 1 April 2016 / Revised: 16 October 2016 / Accepted: 9 February 2017 /
Published online: 17 February 2017

Abstract In this paper we intend to establish fast numerical approaches to solve a class of initial-boundary problem of time-space fractional convection–diffusion equations. We present a new unconditionally stable implicit difference method, which is derived from the weighted and shifted Grünwald formula, and converges with the second-order accuracy in both time and space variables. Then, we show that the discretizations lead to Toeplitz-like systems of linear equations that can be efficiently solved by Krylov subspace solvers with suitable circulant preconditioners. Each time level of these methods reduces the memory requirement of the proposed implicit difference scheme from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ and the computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$ in each iterative step, where N is the

✉ Ting-Zhu Huang
tingzhuhuang@126.com

Xian-Ming Gu
guxianming@live.cn; x.m.gu@rug.nl

Cui-Cui Ji
cuicuiahuan@163.com

Bruno Carpentieri
bcarpentieri@gmail.com

Anatoly A. Alikhanov
aaalikhanov@gmail.com

[1] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, People's Republic of China

[2] Institute of Mathematics and Computing Science, University of Groningen, Nijenborgh 9, P.O. Box 407, 9700 AK Groningen, The Netherlands

[3] Department of Mathematics, Southeast University, Nanjing 210096, Jiangsu, People's Republic of China

[4] School of Science and Technology, Nottingham Trent University, Clifton Campus, Nottingham NG11 8NS, UK

[5] Institute of Applied Mathematics and Automation, Russian Academy of Sciences, ul. Shortanova 89 a, Nalchik, Russia 360000

number of grid nodes. Extensive numerical examples are reported to support our theoretical findings and show the utility of these methods over traditional direct solvers of the implicit difference method, in terms of computational cost and memory requirements.

Keywords Fractional convection–diffusion equation · Shifted Grünwald discretization · Toeplitz matrix · Fast Fourier transform · Circulant preconditioner · Krylov subspace method

Mathematics Subject Classification 65F15 · 65H18 · 15A51

1 Introduction

Recent years have been a growing body of knowledge in the theory of fractional calculus and fractional differential equations (FDEs) with applications to modelling practical scientific problems arising in engineering, physics, chemistry and other applied sciences. For example, diffusion with an additional velocity field and diffusion under the influence of a constant external force field are, in the Brownian case, both modelled by the convection–dispersion equation. In the case of anomalous diffusion this is no longer true, i.e., the fractional generalization may be different for the advection case and the transport in external force field [4]. We point the reader to, e.g., Podlubny [1], Samko et al. [2] and Kilbas et al. [3] for some history and for a comprehensive treatment of this subject. In the present study, we consider in particular a fast and stable numerical approach for solving the initial-boundary value problem of the time-space fractional convection–diffusion equation (TSFCDE) [5,6]:

$$\begin{cases} \partial_{0,t}^{\alpha} u(x,t) = \gamma(t)\frac{\partial u(x,t)}{\partial x} + d_+(t)_a D_x^{\beta} u(x,t) + d_-(t)_x D_b^{\beta} u(x,t) + f(x,t), \\ u(x,0) = \phi(x), \qquad a \le x \le b, \\ u(a,t) = u(b,t) = 0, \quad 0 < t \le T, \end{cases} \tag{1.1}$$

where $\alpha \in (0,1]$, $\beta \in (1,2]$, $a < x < b$, and $0 < t \le T$. Here, the parameters α and β appearing in (1.1) denote the order of the TSFCDE, $f(x,t)$ is the source term, and diffusion coefficient functions $d_\pm(t)$ are non-negative under the assumption that the flow is from left to right. Moreover, the coefficients $\gamma(t)$ represents the velocity, which is depend upon the time t, of the flow.

The TSFCDE (1.1) can be regarded as a generalization of classical convection–diffusion equations where the first-order time derivative is replaced by the Caputo fractional derivative of order $\alpha \in (0,1]$, and the second-order space derivative is replaced by the two-sided Riemann–Liouville fractional derivative of order $\beta \in (1,2]$. Specifically, the time fractional derivative in (1.1) is the Caputo fractional derivative of order α [1] denoted by

$$\partial_{0,t}^{\alpha} u(x,t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x,\xi)}{\partial \xi} \frac{d\xi}{(t-\xi)^{\alpha}}, \tag{1.2}$$

and the left-handed (${}_a D_x^{\beta}$) and the right-handed (${}_x D_b^{\beta}$) space fractional derivatives in (1.1) are the Riemann–Liouville fractional derivatives of order β [2,3] which are defined as

$${}_a D_x^{\beta} u(x,t) = \frac{1}{\Gamma(2-\beta)} \frac{\partial^2}{\partial x^2} \int_a^x \frac{u(\eta,t)d\eta}{(x-\eta)^{\beta-1}} \tag{1.3}$$

and

$${}_x D_b^{\beta} u(x,t) = \frac{1}{\Gamma(2-\beta)} \frac{\partial^2}{\partial x^2} \int_x^b \frac{u(\eta,t)d\eta}{(\eta-x)^{\beta-1}}, \tag{1.4}$$

$\Gamma(\cdot)$ denoting the Gamma function. Note that the above equation reduces to the classical convection–diffusion equation (CDE) for $\alpha = 1$ and $\beta = 2$.

Many research studies in the last two decades have shown that fractional CDEs can be a valuable theoretical tool for modelling transport dynamics in complex systems governed by anomalous diffusion and non-exponential relaxation patterns, see e.g., [4,5]. Moreover, the fractional CDE is also popular in groundwater hydrology research to model the transport of passive tracers carried by fluid flow in a porous medium [7,8].

Despite the growing body of development of analytic approaches based on the Fourier transform method, the Laplace transform method, and the Mellin transform method for seeking closed-form solutions of FDEs [1,6,9,10], to date the overwhelming majority of FDEs of practical interest in computational science do not have solutions that can be expressed in terms of simple function. Therefore, we must rely on numerical methods that produce approximations to the desired solutions; refer, e.g., to [11–16] and references therein for the description of some approaches.

Most of the early established numerical methods for fractional convection–diffusion equations were developed for solving either the space factional CDE or the time fractional CDE. For space fractional CDE, many of the available numerical schemes are based on the conventional shifted Grünwald discretization [17] and the implicit Euler (or Crank–Nicolson) time-stepping discretization for two-sided Riemann–Liouville fractional derivatives and the first-order time derivative, respectively; see for example [8,17–25] and references therein for details. By combining the second-order spatial discretization and the Crank–Nicolson temporal discretization, Chen and Deng have produced the second-order accurate numerical methods, which achieve the second-order accuracy in both time and space for space fractional CDE [26,27]. Even Chen and Deng and Qu et al. separately designed the fast computational techniques, which can also reduce the required algorithmic storage, for implementing the above mentioned second-order numerical scheme; see [27,28] for details. Additionally, there are also some other interesting numerical methods for the space fractional CDE, refer, e.g., to [29–33] for a discussion of these issues.

On the other hand, many of the early developed implicit methods for time fractional CDEs are derived by combining the $L1$ approximate formula [34] for Caputo fractional derivative with the first/second-order spatial discretization. These numerical methods are unconditionally convergent with the accuracy of $\mathcal{O}(\tau^{2-\alpha} + h)$ or $\mathcal{O}(\tau^{2-\alpha} + h^2)$, where τ and h are the time-stepping size and the spatial grid size, respectively; see for example [35] and reference therein for details. It is remarkable that Cui [36,37] and Mohebbi and Abbaszadeh [38] have proposed two compact exponential methods and a compact finite difference method that achieve fourth-order spatial accuracy for a time fractional convection–subdiffusion equation. The analyses in [36,38] are derived for the equations with constant coefficients; specifically, the diffusion and convection coefficients are assumed to be one in [38]. We point the reader to [39–44] for some numerical approaches for handling the time fractional CDE.

In contrast, although the numerical methods for space (or time) fractional CDE are extensively investigated in the past research, the work about numerically handling the TSFCDE is not too much. Firstly, Zhang [45,46], Shao and Ma [47], Qin and Zhang [48] and Liu et al. [24] have worked out a series of studies about constructing the implicit difference scheme (IDS) for TSFCDE, however all these numerical schemes can achieve the convergence with first-order accuracy in both space and time from both the theoretical and numerical perspectives. Moreover, Liu et al. [24,49], Zhao et al. [50] and Shen et al. [51] had considered to solve the more general form of TSFCDE, in which the first-order space derivative is replaced by the two-sided Riemann–Liouville fractional derivative of order $\nu \in (0, 1)$. Again, their numerical methods cannot enjoy the convergence with second-order accuracy in both space and time. In addition, some other efficient approaches are also developed for dealing with TSFCDE numerically. Moreover, most of these numerical methods have no complete theoretical analysis for both convergence and stability; e.g., refer to [31,52–58] for details.

One problem in the development of efficient numerical schemes for solving FDEs is that conventional methods tend to generate fully populated systems whose solutions incur $\mathcal{O}(N^3)$ arithmetic operations and $\mathcal{O}(N^2)$ storage requirement, N denoting the number of grid points [8,28]. To optimize this huge cost, Meerschaert and Tadjeran have proposed to approximate the space fractional CDE by a shifted Grünwald discretization scheme, which is unconditional stability [17]. Later, Wang and Wang made the important observation that the linear system arising from this discretization has a special Toeplitz-like coefficient matrix, more precisely, this coefficient matrix can be expressed as a sum of diagonal–multiply–Toeplitz matrices [8]. Exploting this structure the storage requirement can be reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$, and the fast Fourier transform (FFT) [59] can be used to carry out the matrix–vector product in only $\mathcal{O}(N\log N)$ operations. Wang and Wang solve the linear systems arising from the discretization in $\mathcal{O}(N\log^2 N)$ arithmetic operations using the conjugate gradient on the normal equations (known as CGNR) method. The convergence of the CGNR method turns out to be fast when the diffusion coefficients are very small and the discretized systems are well-conditioned, but it can be rather slow when the diffusion coefficients are not small [8]. As an attempt of a possible remedy, Zhao et al. have extended the preconditioned technique, which is introduced by Lin et al. in the context of space fractional diffusion equations [60] to the case of solving Toeplitz-like linear systems arising from the discretization of TSFCDE [50]. Their results related to the promising acceleration of the convergence of the iterative methods, while solving (1.1). At the same time, it remarked that these mentioned fast solution techniques and the corresponding eigenvalue analyses (of preconditioned matrices) are constructed via using the shifted Grünwald formula, in which its coefficients have many essential properties for theoretical results; refer to [50,60,61] for details.

The novelty of this paper compared to earlier literature in this field is to present a new unconditionally stable implicit difference scheme that is second-order accurate in both space and time, i.e., $\mathcal{O}(\tau^2+h^2)$, for solving problem (1.1). These properties of our proposed scheme will be proved both analytically and numerically. To the best of our knowledge, this is the successfully attempt to derive a second-order numerical scheme of TSFCDEs without using extrapolation. Especially, we also investigate the performance of our scheme for TSFCDEs when its convection term becomes dominant and our scheme also works well for this case. On the other side, the time marching of the scheme gives rise to a sequence of linear systems with different Toeplitz coefficient matrices. Those linear systems can be solved efficiently one after one by using Krylov subspace methods with suitable circulant preconditioners [59,61], then it can reduce the computational cost and memory deeply. Especially for TSFCDE with constant coefficients, we turn to represent the inverse of the Toeplitz coefficient matrix as a sum of products of Toeplitz triangular matrices [59,62], so that the solution of each linear system for time marching can be obtained by several FFTs. To obtain the explicit inversion of Toeplitz matrix, only two specific linear systems with the same Toeplitz coefficient matrix are needed to be solved by the preconditioned Krylov subspace methods [63] with complexity $\mathcal{O}(N\log N)$.

An outline of this paper is as follows. In Sect. 2, we establish a novel implicit difference scheme for (1.1) and we prove that this scheme is unconditionally stable and convergent with the accuracy of $\mathcal{O}(\tau^2+h^2)$. In Sect. 3, we investigate that the resulting linear systems have the nonsymmetric Toeplitz matrices, then we design the fast solution techniques based on preconditioned Krylov subspace methods to solve (1.1) by exploiting the Toeplitz matrix property of the implicit difference scheme. Finally, we present numerical experiments to illustrate the effectiveness of our numerical approaches in Sect. 4 and provide concluding remarks in Sect. 5.

2 An Implicit Difference Scheme for TSFCDEs

In this section, we present an implicit difference method for discretizing the TSFCDE defined by (1.1). Unlike former numerical approaches with the first-order accuracy in both time and space [45–47,49,50], we exploit henceforth second-order approximate operators to discretize the Riemann–Liouville derivatives in (1.3) and (1.4). We can show that, by two-sided fractional derivatives, this proposed method is also unconditionally stable and convergent under second-order accuracy in time and space.

2.1 Numerical Discretization of the TSFCDE

To derive the proposed scheme, we first consider a rectangle $\bar{Q}_T = \{(x,t) : a \le x \le b, 0 \le t \le T\}$ discretized on the mesh $\varpi_{h\tau} = \varpi_h \times \varpi_\tau$, where $\varpi_h = \{x_i = ih,\ i = 0, 1, \dots, N;\ hN = b - a\}$, and $\varpi_\tau = \{t_j = j\tau,\ j = 0, 1, \dots, M;\ \tau = T/M\}$. We denote by $\boldsymbol{v} = \{v_i \mid 0 \le i \le N\}$ any grid function. Then, the following lemma introduced in [64] gives a complete description of the discretization in the time variable.

Lemma 2.1 *Suppose* $0 < \alpha < 1, \sigma = 1 - \frac{\alpha}{2}, u(t) \in \mathcal{C}^3[0,T]$, *and* $t_{j+\sigma} = (j+\sigma)\tau$. *Then*

$$\partial_{0,t}^{\alpha} u(t_{j+\sigma}) - \Delta_{0,t_{j+\sigma}}^{\alpha} u(t) = \mathcal{O}(\tau^{3-\alpha}),$$

where $\Delta_{0,t_{j+\sigma}}^{\alpha} u(t) = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \sum_{s=0}^{j} c_{j-s}^{(\alpha,\sigma,j)} [u(t_{s+1}) - u(t_s)]$, *and* $c_0^{(\alpha,\sigma,0)} = a_0^{(\alpha,\sigma)}$ *for* $j = 0$,

$$c_m^{(\alpha,\sigma,j)} = \begin{cases} a_0^{(\alpha,\sigma)} + b_1^{(\alpha,\sigma)}, & m = 0, \\ a_m^{(\alpha,\sigma)} + b_{m+1}^{(\alpha,\sigma)} - b_m^{(\alpha,\sigma)}, & 1 \le m \le j-1, \\ a_j^{(\alpha,\sigma)} - b_j^{(\alpha,\sigma)}, & m = j, \end{cases}$$

for $j \ge 1$, *in which* $a_0^{(\alpha,\sigma)} = \sigma^{1-\alpha}$, $a_\ell^{(\alpha,\sigma)} = (\ell+\sigma)^{1-\alpha} - (\ell-1+\sigma)^{1-\alpha}$, *for* $\ell \ge 1$; *and* $b_\ell^{(\alpha,\sigma)} = \frac{1}{2-\alpha}[(\ell+\sigma)^{2-\alpha} - (\ell-1+\sigma)^{2-\alpha}] - \frac{1}{2}[(\ell+\sigma)^{1-\alpha} + (\ell-1+\sigma)^{1-\alpha}]$.

To characterize the discretization in the space variable, first we denote by $\mathfrak{L}^{n+\beta}(\mathbb{R}) = \{v | v \in L_1(\mathbb{R})$ and $\int_{-\infty}^{+\infty} (1+|k|)^{n+\beta} |\hat{v}(k)| dk < \infty\}$, where $\hat{v}(k) = \int_{-\infty}^{+\infty} e^{\iota k x} v(x) dx$ is the Fourier transformation of $v(x)$, and by $\iota = \sqrt{-1}$ the imaginary unit. Then we introduce the following preliminary lemma:

Lemma 2.2 ([65,66]) *Suppose that* $v \in \mathfrak{L}^{2+\beta}(\mathbb{R})$, *and let*

$$\delta_{x,+}^{\beta} v(x) = \frac{1}{h^\beta} \sum_{k=0}^{[\frac{x-a}{h}]} \omega_k^{(\beta)} v(x - (k-1)h),$$

$$\delta_{x,-}^{\beta} v(x) = \frac{1}{h^\beta} \sum_{k=0}^{[\frac{b-x}{h}]} \omega_k^{(\beta)} v(x + (k-1)h),$$

then for a fixed h, we have

$$
\begin{aligned}
{}_aD_x^{\beta} v(x) &= \delta_{x,+}^{\beta} v(x) + \mathcal{O}(h^2),\\
{}_xD_b^{\beta} v(x) &= \delta_{x,-}^{\beta} v(x) + \mathcal{O}(h^2),
\end{aligned}
$$

where

$$
\begin{cases}
\omega_0^{(\beta)} = \lambda_1 g_0^{(\beta)}, \quad \omega_1^{(\beta)} = \lambda_1 g_1^{(\beta)} + \lambda_0 g_0^{(\beta)},\\
\omega_k^{(\beta)} = \lambda_1 g_k^{(\beta)} + \lambda_0 g_{k-1}^{(\beta)} + \lambda_{-1} g_{k-2}^{(\beta)}, \qquad k \geq 2,
\end{cases}
$$

with

$$
\lambda_1 = \frac{\beta^2+3\beta+2}{12}, \quad \lambda_0 = \frac{4-\beta^2}{6}, \quad \lambda_{-1} = \frac{\beta^2-3\beta+2}{12}, \quad \textit{and} \ \ g_k^{(\beta)} = (-1)^k \binom{\beta}{k}.
$$

At the stage, let $u(x,t) \in \mathcal{C}_{x,t}^{4,3}([a,b]\times[0,T])$ be a solution of the problem (1.1). Then we define the parameter $\sigma = 1 - \frac{\alpha}{2}$ and consider Eq. (1.1) at the set of grid points $(x,t) = (x_i, t_{j+\sigma}) \in \bar{Q}_T$, $i = 1,2,\ldots,N-1$, $j = 0,1,\ldots,M-1$:

$$
\begin{aligned}
\partial_{0,t}^{\alpha} u(x, t_{j+\sigma}) = \gamma(t_{j+\sigma}) \Big(\frac{\partial u(x,t)}{\partial x}\Big)_{(x_i,t_{j+\sigma})} + d_+(t_{j+\sigma}) \Big({}_aD_x^{\beta} u(x,t)\Big)_{(x_i,t_{j+\sigma})}\\
+ d_-(t_{j+\sigma}) \Big({}_xD_b^{\beta} u(x,t)\Big)_{(x_i,t_{j+\sigma})} + f(x_i, t_{j+\sigma}).
\end{aligned}
$$

For simplicity, we define

$$
\begin{aligned}
u_i^{(\sigma)} &= \sigma u_i^{j+1} + (1-\sigma) u_i^j, \quad \gamma^{(j+\sigma)} = \gamma(t_{j+\sigma}),\\
D_{\pm}^{(j+\sigma)} &= d_{\pm}(t_{j+\sigma}), \quad f_i^{j+\sigma} = f(x_i, t_{j+\sigma})
\end{aligned}
$$

and

$$
\delta_h^{\beta} u_i^{(\sigma)} = \gamma^{(j+\sigma)} \frac{u_{i+1}^{(\sigma)} - u_{i-1}^{(\sigma)}}{2h} + \frac{D_+^{(j+\sigma)}}{h^{\beta}} \sum_{k=0}^{i+1} \omega_k^{(\beta)} u_{i-k+1}^{(\sigma)} + \frac{D_-^{(j+\sigma)}}{h^{\beta}} \sum_{k=0}^{N-i+1} \omega_k^{(\beta)} u_{i+k-1}^{(\sigma)}.
$$

By Lemma 2.1, the following implicit difference scheme with the approximation order $\mathcal{O}(h^2 + \tau^2)$ is derived:

$$
\begin{cases}
\Delta_{0,t_{j+\sigma}}^{\alpha} u_i = \delta_h^{\beta} u_i^{(\sigma)} + f_i^{j+\sigma}, & 1 \leq i \leq N-1, \quad 0 \leq j \leq M-1,\\
u_i^0 = \phi(x_i), & 1 \leq i \leq N-1,\\
u_0^j = u_N^j = 0, & 0 \leq j \leq M.
\end{cases}
\tag{2.1}
$$

It is interesting to note that for $\alpha \to 1$, Eq. (2.1) reduces to the classical Crank–Nicolson difference scheme.

2.2 Stability and Convergence Analysis

In this section, we need to analyze the stability and convergence for the implicit difference scheme (2.1). We define

$$
V_h = \{\boldsymbol{v} \mid \boldsymbol{v} = \{v_i\} \text{ is a grid function on } \varpi_h \text{ and } v_i = 0 \text{ if } i = 0, N\},
$$

and, for all $\boldsymbol{u},\ \boldsymbol{v} \in V_h$, the discrete inner product and corresponding discrete L_2-norms

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

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

Lemma 2.3 ([17,60,65]) *Let* $1 < \beta < 2$ *and* $g_k^{(\beta)}$ *be defined in Lemma 2.2. Then we have*

$$\begin{cases} g_0^{(\beta)} = 1, \quad g_1^{(\beta)} = -\beta, \quad g_2^{(\beta)} > g_3^{(\beta)} > \cdots > 0, \\ \sum\limits_{k=0}^{\infty} g_k^{(\beta)} = 0, \quad \sum\limits_{k=0}^{N} g_k^{(\beta)} < 0, \quad N > 1, \\ g_k^{(\beta)} = \mathcal{O}(k^{-(\beta+1)}), \quad g_k^{(\beta)} = \left(1 - \frac{\beta+1}{k}\right) g_{k-1}^{(\beta)}, \quad k = 1, 2, \ldots \end{cases}$$

Lemma 2.3 ensures the following properties of the coefficients $\omega_k^{(\beta)}$, since the second-order approximate formulae of two two-sided Riemann–Liouville fractional derivatives are constructed by combining the distinct shifted Grünwald–Letnikov formulae with their corresponding weights [65].

Lemma 2.4 ([65,66])*Let* $1 < \beta < 2$ *and* $g_k^{(\beta)}$ *be defined in Lemma 2.2. Then we have*

$$\begin{cases} \omega_0^{(\beta)} = \lambda_1, \quad \omega_1^{(\beta)} < 0, \quad \omega_k^{(\beta)} > 0, \quad k \geq 3, \\ \sum\limits_{k=0}^{\infty} \omega_k^{(\beta)} = 0, \quad \sum\limits_{k=0}^{N} \omega_k^{(\beta)} < 0, \quad N > 1, \\ \omega_0^{(\beta)} + \omega_2^{(\beta)} \geq 0. \end{cases}$$

Lemma 2.4 ensures the first property of the discrete inner product related to two approximate operators $\delta_{x,+}^{\beta}$ and $\delta_{x,-}^{\beta}$ shown below.

Lemma 2.5 ([65,66]) *For* $1 < \beta < 2$*, and any* $\boldsymbol{v} \in V_h$*, it holds that*

$$(\delta_{x,+}^{\beta} \boldsymbol{v}, \boldsymbol{v}) = (\delta_{x,-}^{\beta} \boldsymbol{v}, \boldsymbol{v}) \leq \left(\frac{1}{h^{\beta}} \sum_{k=0}^{N-1} \omega_k^{(\beta)} \right) \|\boldsymbol{v}\|^2.$$

A more accurate estimate of the discrete inner product associated to two approximate operators $\delta_{x,+}^{\beta}$ and $\delta_{x,-}^{\beta}$ can be obtained from Lemma 2.5 as follows.

Lemma 2.6 *For* $1 < \beta < 2$*,* $N \geq 5$*, and any* $\boldsymbol{v} \in V_h$*, there exists a positive constant* c_1*, such that*

$$(-\delta_{x,+}^{\beta} \boldsymbol{v}, \boldsymbol{v}) = (-\delta_{x,-}^{\beta} \boldsymbol{v}, \boldsymbol{v}) > c_1 \ln 2 \|\boldsymbol{v}\|^2.$$

Proof Since $N \geq 5$ and

$$\sum_{k=N}^{2N+2} \omega_k^{(\beta)} = \sum_{k=N}^{2N} g_k^{(\beta)} + (\lambda_1 + \lambda_0) g_{2N+1}^{(\beta)} + \lambda_1 g_{2N+2}^{(\beta)} + \zeta(\beta),$$

where

$$\begin{aligned} \zeta(\beta) = (\lambda_0 + \lambda_{-1}) g_{N-1}^{(\beta)} + \lambda_{-1} g_{N-2}^{(\beta)} &= \left[(\lambda_0 + \lambda_{-1}) \frac{N-2-\beta}{N-1} + \lambda_{-1} \right] g_{N-2}^{(\beta)} \\ &= \frac{(12-6\beta)N + \beta^3 + 4\beta^2 - \beta - 22}{12(N-1)} g_{N-2}^{(\beta)}, \\ &\triangleq \frac{\vartheta(\beta)}{12(N-1)} g_{N-2}^{(\beta)} \end{aligned}$$

then $\zeta(2) = 0$, $\vartheta(2) = 0$ and $\vartheta'(\beta) = -6N + 3\beta^2 + 8\beta - 1 \leq 27 - 6N < 0$ due to $N \geq 5$, which implies $\zeta(\beta)$ is a decreasing function for $\beta \in [1, 2]$. Hence $\zeta(\beta) > 0$ when $N \geq 5$ (i.e., $g_{N-2}^{(\beta)} > 0$).

Next, by Lemma 2.3, there exist two positive constants $\tilde{c}_1$ and c_1, such that

$$\begin{aligned} \frac{1}{h^\beta}\sum_{k=N}^{\infty}\omega_k^{(\beta)} &> \frac{1}{h^\beta}\sum_{k=N}^{2N+2}\omega_k^{(\beta)} > \frac{1}{h^\beta}\sum_{k=N}^{2N} g_k^{(\beta)} \geq \tilde{c}_1 \sum_{k=N}^{2N} k^{-(\beta+1)} N^\beta \\ &> \tilde{c}_1 \sum_{k=N}^{2N} k^{-(\beta+1)}\left(\frac{k}{2}\right)^\beta = \frac{\tilde{c}_1}{2^\beta}\sum_{k=N}^{2N}\frac{1}{k} \\ &\geq c_1 \int_N^{2N+1}\frac{1}{x}dx \geq c_1 \int_N^{2N}\frac{1}{x}dx = c_1 \ln 2, \qquad N \geq 5. \end{aligned} \tag{2.2}$$

The penultimate and antepenult inequalities in (2.2) hold because $J_1(x) = \frac{1}{x}$ is a lower convex function and $J_2(x) = \ln(x) > 0$, $x \in [N, 2N+1]$ is an increasing function. Then, applying Lemmas 2.4 and 2.5, we obtain the desired result

$$(-\delta_{x,+}^\beta \boldsymbol{v}, \boldsymbol{v}) = (-\delta_{x,-}^\beta \boldsymbol{v}, \boldsymbol{v}) > \left(\frac{1}{h^\beta}\sum_{k=N}^{\infty}\omega_k^{(\beta)}\right)\|\boldsymbol{v}\|^2 > c_1 \ln 2 \|\boldsymbol{v}\|^2.$$

The following bound, derived from the above lemmas, is an essential ingredient for the stability analysis of this section. □

Theorem 2.1 *For any $\boldsymbol{v} \in V_h$, it holds that*

$$(\delta_h^\beta \boldsymbol{v}, \boldsymbol{v}) \leq -c \ln 2 \|\boldsymbol{v}\|^2,$$

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

Proof We start from the expression of $(\delta_h^\beta \boldsymbol{v}, \boldsymbol{v})$

$$(\delta_h^\beta \boldsymbol{v}, \boldsymbol{v}) = \gamma^{(j+\sigma)} h \sum_{i=1}^{N-1}\frac{v_{i+1} - v_{i-1}}{2h} v_i + D_+^{(j+\sigma)}(\delta_{x,+}^\beta \boldsymbol{v}, \boldsymbol{v}) + D_-^{(j+\sigma)}(\delta_{x,-}^\beta \boldsymbol{v}, \boldsymbol{v}). \tag{2.3}$$

Since $v_0 = v_N = 0$, we have

$$\gamma^{(j+\sigma)} h \sum_{i=1}^{N-1}\frac{v_{i+1} - v_{i-1}}{2h} v_i = 0. \tag{2.4}$$

Moreover, according to Lemma 2.6, there exists a positive constant c_1 independent of the spatial step size h such that, for any non-vanishing vector $\boldsymbol{v} \in V_h$, it is

$$D_+^{(j+\sigma)}(\delta_{x,+}^\beta \boldsymbol{v}, \boldsymbol{v}) + D_-^{(j+\sigma)}(\delta_{x,-}^\beta \boldsymbol{v}, \boldsymbol{v}) \leq -c_1 \ln 2\left(D_+^{(j+\sigma)} + D_-^{(j+\sigma)}\right)\|\boldsymbol{v}\|^2 \tag{2.5}$$

Denote $c = c_1\left(D_+^{(j+\sigma)} + D_-^{(j+\sigma)}\right)$. Then the main result follows by inserting (2.4) and (2.5) into (2.3). □

Before proving the most important result of this section on the unconditional stability and quadratic-order convergence property of the implicit difference scheme (2.1), we first need to recall the following useful lemma.

Lemma 2.7 ([64,66]) *Let* $V_\tau = \{\boldsymbol{u}|\boldsymbol{u} = (u^0, u^1, \ldots, u^M)\}$ *For any* $\boldsymbol{u} \in V_\tau$*; one has the following inequality*

$$[\sigma u^{j+1} + (1-\sigma)u^j]\Delta^\alpha_{0,t_{j+\sigma}}\boldsymbol{u} \geq \frac{1}{2}\Delta^\alpha_{0,t_{j+\sigma}}(\boldsymbol{u})^2.$$

For simplicity of presentation, in our proof, we denote $a_s^{j+1} = \frac{c_{j-s}^{(\alpha,\sigma,j)}}{\tau^\alpha\Gamma(2-\alpha)}$. Then $\Delta^\alpha_{0,t_{j+\sigma}}u = \sum\limits_{s=0}^{j}(u^{s+1}-u^s)a_s^{j+1}$.

Theorem 2.2 *Denote* $\|f^{j+\sigma}\|^2 = h\sum\limits_{i=1}^{N-1} f^2(x_i, t_{j+\sigma})$. *Then the implicit difference scheme* (2.1) *is unconditionally stable with respects to the initial value* u^0 *and the source term* f*, and the following priori estimate holds*:

$$\|u^{j+1}\|^2 \leq \|u^0\|^2 + \frac{T^\alpha\Gamma(1-\alpha)}{c\ln 2}\|f^{j+\sigma}\|^2, \quad 0 \leq j \leq M-1, \tag{2.6}$$

where $u^{j+1} = (u_1^{j+1}, u_2^{j+1}, \ldots, u_{N-1}^{j+1})^T$.

Proof To make an inner product of (2.1) with $u^{(\sigma)}$, we have

$$(\Delta^\alpha_{0,t_{j+\sigma}}u, u^{(\sigma)}) = (\delta_h^\beta u^{(\sigma)}, u^{(\sigma)}) + (f^{j+\sigma}, u^{(\sigma)}). \tag{2.7}$$

It follows from Theorem 2.1 and Lemma 2.6 that

$$(\delta_h^\beta u^{(\sigma)}, u^{(\sigma)}) \leq -c\ln 2\|u^{(\sigma)}\|^2, \tag{2.8}$$

$$(\Delta^\alpha_{0,t_{j+\sigma}}u, u^{(\sigma)}) \geq \frac{1}{2}\Delta^\alpha_{0,t_{j+\sigma}}(\|u\|^2). \tag{2.9}$$

Inserting (2.8) and (2.9) into (2.7) and using the Cauchy–Schwarz and Youngs' inequalities, we can write

$$\begin{aligned}\frac{1}{2}\Delta^\alpha_{0,t_{j+\sigma}}(\|u\|^2) &\leq -c\ln 2\|u^{(\sigma)}\|^2 + (f^{j+\sigma}, u^{(\sigma)})\\ &\leq -c\ln 2\|u^{(\sigma)}\|^2 + c\ln 2\|u^{(\sigma)}\|^2 + \frac{1}{4c\ln 2}\|f^{j+\sigma}\|^2\\ &\leq \frac{1}{4c\ln 2}\|f^{j+\sigma}\|^2.\end{aligned} \tag{2.10}$$

Next, we have the following inequality

$$a_j^{j+1}\|u^{j+1}\|^2 \leq \sum_{s=1}^{j}(a_s^{j+1} - a_{s-1}^{j+1})\|u^s\|^2 + a_0^{j+1}\|u^0\|^2 + \frac{1}{2c\ln 2}\|f^{j+\sigma}\|^2.$$

Exploiting $a_0^{j+1} > \frac{1}{2T^\alpha\Gamma(1-\alpha)}$ (cf. [64]), we obtain

$$a_j^{j+1}\|u^{j+1}\|^2 \leq \sum_{s=1}^{j}(a_s^{j+1} - a_{s-1}^{j+1})\|u^s\|^2 + a_0^{j+1}\Big(\|u^0\|^2 + \frac{T^\alpha\Gamma(1-\alpha)}{c\ln 2}\|f^{j+\sigma}\|^2\Big). \tag{2.11}$$

Suppose $h < 1$ and denote

$$\check{\mathcal{P}} \triangleq \|u^0\|^2 + \frac{T^\alpha \Gamma(1-\alpha)}{c \ln 2} \|f^{j+\sigma}\|^2.$$

Then, Eq. (2.11) can be rewritten as

$$a_j^{j+1} \|u^{j+1}\|^2 \leq \sum_{s=1}^{j} (a_s^{j+1} - a_{s-1}^{j+1}) \|u^s\|^2 + a_0^{j+1} \check{\mathcal{P}}. \tag{2.12}$$

At this stage, by mathematical induction we prove that the estimate relation (2.6) is valid for $j = 0, 1 \ldots, M-1$. The result is obviously true for $j = 0$ from (2.12). Assuming that (2.6) holds for all $0 \leq j \leq k$ $(0 \leq k \leq M-1)$, meaning that

$$\|u^j\| \leq \check{\mathcal{P}}, \qquad j = 0, 1, \ldots, k.$$

From (2.12) at $j = k$, one has

$$\begin{aligned} a_k^{k+1} \|u^{k+1}\|^2 &\leq \sum_{s=1}^{k} (a_s^{k+1} - a_{s-1}^{k+1}) \|u^s\|^2 + a_0^{k+1} \check{\mathcal{P}} \\ &\leq \sum_{s=1}^{k} (a_s^{k+1} - a_{s-1}^{k+1}) \check{\mathcal{P}} + a_0^{k+1} \check{\mathcal{P}} = a_k^{k+1} \check{\mathcal{P}}. \end{aligned}$$

This completes the proof of Theorem 2.2. □

The following theorem shows that our implicit difference scheme achieves quadratic-order convergence in both time and space variables.

Theorem 2.3 *Suppose that* $u(x,t) \in C_{x,t}^{4,3}([a,b] \times [0,T])$ *is the solution of* (1.1) *and* $\{u_i^j \mid x_i \in \varpi_h, \;\; 0 \leq j \leq M\}$ *is the solution of the implicit difference scheme* (2.1). *Denote*

$$E_i^j = u(x_i, t_j) - u_i^j, \quad x_i \in \varpi_h, \quad 0 \leq j \leq M.$$

Then there exists a positive constant $\tilde{c}$ *such that*

$$\|E^j\| \leq \tilde{c}(\tau^2 + h^2), \quad 0 \leq j \leq M.$$

Proof It can be easily obtained that E^j satisfies the following error equation

$$\begin{cases} \Delta_{0,t_{j+\sigma}}^{\alpha} E_i = \delta_h^{\beta} E_i^{(\sigma)} + R_i^{j+\sigma}, & 1 \leq i \leq N-1, \quad 0 \leq j \leq M-1, \\ E_i^0 = 0, & 1 \leq i \leq N-1, \\ E_0^j = E_N^j = 0, & 0 \leq j \leq M. \end{cases}$$

where $R_i^{j+\sigma} = \mathcal{O}(\tau^2 + h^2)$. In virtue of Theorem 2.2, we can write

$$\|E^{j+1}\|^2 \leq \frac{T^\alpha \Gamma(1-\alpha)}{c \ln 2} \|R^{j+\sigma}\|^2 \leq \tilde{c}(\tau^2 + h^2), \quad 0 \leq j \leq M-1,$$

which proves the theorem. □

3 Fast Solution Techniques Based on Preconditioned Iterative Solvers

In the section, we analyze both the implementation and the computational complexity of IDS (2.1) and we propose an efficient implementation based on preconditioned Krylov subspace solvers. We start the development from the following matrix form of the implicit difference scheme (2.1) at the time level j:

$$\begin{cases} \Big[\eta_j I - \sigma\Big(\frac{\gamma^{(\sigma)}}{2h} Q + \frac{D_+^{(\sigma)}}{h^\beta} W_\beta + \frac{D_-^{(\sigma)}}{h^\beta} W_\beta^T\Big)\Big]\boldsymbol{u}^1 = \Big[\eta_j I + (1-\sigma)\Big(\frac{\gamma^{(\sigma)}}{2h} Q + \frac{D_+^{(\sigma)}}{h^\beta} W_\beta \\ \quad + \frac{D_-^{(\sigma)}}{h^\beta} W_\beta^T\Big)\Big]\boldsymbol{u}^0 + \boldsymbol{f}^\sigma, \qquad j = 0, \\ \Big[\eta_j I - \sigma\Big(\frac{\gamma^{(j+\sigma)}}{2h} Q + \frac{D_+^{(j+\sigma)}}{h^\beta} W_\beta + \frac{D_-^{(j+\sigma)}}{h^\beta} W_\beta^T\Big)\Big]\boldsymbol{u}^{j+1} = \Big[\eta_j I + (1-\sigma)\Big(\frac{\gamma^{(j+\sigma)}}{2h} Q + \\ \frac{D_+^{(j+\sigma)}}{h^\beta} W_\beta + \frac{D_-^{(j+\sigma)}}{h^\beta} W_\beta^T\Big)\Big]\boldsymbol{u}^j - \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \sum_{s=0}^{j-1} c_{j-s}^{(\alpha,\sigma,j)} (\boldsymbol{u}^{s+1} - \boldsymbol{u}^s) + \boldsymbol{f}^{j+\sigma}, \\ \qquad j = 1, 2, \ldots, M-1, \end{cases} \tag{3.1}$$

where the coefficients η_j are defined as

$$\eta_j = \begin{cases} \frac{c_0^{(\alpha,\sigma,0)}}{\tau^\alpha \Gamma(2-\alpha)} = \frac{a_0^{(\alpha,\sigma)}}{\tau^\alpha \Gamma(2-\alpha)}, & j = 0 \\ \frac{c_0^{(\alpha,\sigma,j)}}{\tau^\alpha \Gamma(2-\alpha)} = \frac{a_0^{(\alpha,\sigma)} + b_1^{(\alpha,\sigma)}}{\tau^\alpha \Gamma(2-\alpha)}, & j = 1, 2, \ldots, M-1, \end{cases}$$

and Q and W are real matrices of size $(N-1) \times (N-1)$ with the following form

$$Q = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \ddots & \vdots \\ 0 & -1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & -1 & 0 \end{bmatrix}, \qquad W_\beta = \begin{bmatrix} \omega_1^{(\beta)} & \omega_0^{(\beta)} & 0 & \cdots & 0 \\ \omega_2^{(\beta)} & \omega_1^{(\beta)} & \omega_0^{(\beta)} & \ddots & \vdots \\ \vdots & \omega_2^{(\beta)} & \omega_1^{(\beta)} & \ddots & 0 \\ \omega_{N-2}^{(\beta)} & \cdots & \ddots & \ddots & \omega_0^{(\beta)} \\ \omega_{N-1}^{(\beta)} & \omega_{N-2}^{(\beta)} & 0 & \omega_2^{(\beta)} & \omega_1^{(\beta)} \end{bmatrix}. \tag{3.2}$$

It is obvious that W_β is a Toeplitz matrix (see [8,59,66]), and therefore it can be stored with only $N+1$ entries [59].

3.1 Resulting Problems from the Discretized Scheme

According to (3.1) and (3.2), there is a sequence of nonsymmetric Toeplitz linear systems to be solved at each time level j of the form

$$A^{(j+\sigma)} \boldsymbol{u}^{(j+1)} = B^{(j+\sigma)} \boldsymbol{u}^{(j)} + \delta \boldsymbol{u}^{(j)} + \boldsymbol{f}^{(j+\sigma)} \tag{3.3}$$

where we have denoted by $\delta \boldsymbol{u}^{(j)} = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \sum_{s=0}^{j-1} c_{j-s}^{(\alpha,\sigma,j)} (\boldsymbol{u}^{s+1} - \boldsymbol{u}^s)$, and by

$$A^{(j+\sigma)} = \eta_j I - \sigma\Big(\frac{\gamma^{(j+\sigma)}}{2h} Q + \frac{D_+^{(j+\sigma)}}{h^\beta} W_\beta + \frac{D_-^{(j+\sigma)}}{h^\beta} W_\beta^T\Big),$$

$$B^{(j+\sigma)} = \eta_j I + (1-\sigma)\Big(\frac{\gamma^{(j+\sigma)}}{2h} Q + \frac{D_+^{(j+\sigma)}}{h^\beta} W_\beta + \frac{D_-^{(j+\sigma)}}{h^\beta} W_\beta^T\Big)$$

for all $j = 0, 1, \dots, M-1$ and $A^{(j+1)}$ varies with j; $\boldsymbol{f}^{(j+\sigma)} \in \mathbb{R}^{N-1}$ also varies with j. The solutions of the sequence of linear systems (3.3) corresponds to the time-stepping scheme (2.1) is inherently sequential, thus it is difficult to parallelize over time.

However, if the coefficients $\gamma(t) = \gamma$ and $d_{\pm}(t) = d_{\pm}$ are constant, then the scalars η_j do not depend on $j = 1, 2, \dots, M-1$, and consequently the coefficient matrices $A^{(j+\sigma)}$

$$A^{(j+\sigma)} = \begin{cases} A^{(\sigma)}, & j = 0, \\ A, & j = 1, 2, \dots, M-1 \end{cases} \tag{3.4}$$

with $A^{(j+\sigma)} = A = \eta_j I - \sigma\left(\frac{\gamma}{2h} Q + \frac{d_+}{h^\beta} W_\beta + \frac{d_-}{h^\beta} W_\beta^T\right)$ are time independent. In this case of $j = 1, 2, \dots, M-1$, the solutions simply write as $\boldsymbol{u}^{(j+\sigma)} = A^{-1}\Big(B^{(j+\sigma)}\boldsymbol{u}^{(j)} + \delta\boldsymbol{u}^{(j)} + \boldsymbol{f}^{(j+\sigma)}\Big)$ and can be computed at the cost of only one LU factorization [67, pp. 44–54]. This approach, however, may be prohibitively expensive if the Toeplitz matrix is large and rather dense. Fortunately, since A is also a Toeplitz matrix, its inverse can be computed with limited memory and algorithmic cost by the Gohberg–Semencul formula (GSF) [59,62] using only its first and last columns. More precisely, denote by $\boldsymbol{e}_1, \boldsymbol{e}_{N-1}$ the first and the last column of the $(N-1)$-by-$(N-1)$ identity matrix, and let $\boldsymbol{x} = [\xi_0, \xi_1, \dots, \xi_{N-2}]^T$ and $\boldsymbol{y} = [\eta_0, \eta_1, \dots, \eta_{N-2}]^T$ be the solutions of the following two Toeplitz systems

$$A\boldsymbol{x} = \boldsymbol{e}_1 \quad \text{and} \quad A\boldsymbol{y} = \boldsymbol{e}_{N-1}. \tag{3.5}$$

If $\xi_0 \neq 0$, then the Gohberg–Semencul formula can be expressed as

$$\begin{aligned} A^{-1} = \frac{1}{\xi_0}\Bigg(& \begin{bmatrix} \xi_0 & 0 & \cdots & 0 \\ \xi_1 & \xi_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{N-2} & \xi_{N-3} & \cdots & \xi_0 \end{bmatrix} \cdot \begin{bmatrix} \eta_{N-2} & \eta_{N-3} & \cdots & \eta_0 \\ 0 & \eta_{N-2} & \cdots & \eta_1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \eta_{N-2} \end{bmatrix} \\ & - \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \eta_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{N-3} & \cdots & \eta_0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & \xi_{N-2} & \cdots & \xi_1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi_{N-2} \\ 0 & 0 & \cdots & 0 \end{bmatrix} \Bigg) = \frac{1}{\xi_0}(L_p R_p - L_p^0 R_p^0), \end{aligned} \tag{3.6}$$

where L_p, L_p^0 are both lower Toeplitz matrices, and R_p, R_p^0 are upper Toeplitz matrices. Consequently, the Toeplitz matrix–vector multiplication $A^{-1}\Big(B^{(j+\sigma)}\boldsymbol{u}^{(j)} + \delta\boldsymbol{u}^{(j)} + \boldsymbol{f}^{(j+\sigma)}\Big)$ can be achieved in several FFTs of length $N-1$ [59]. The following fast algorithm sketches the product of A^{-1} times a vector $\boldsymbol{v}$ using the GSF.

Algorithm 1 Compute $z = A^{-1}\boldsymbol{v}$

```
Solve two linear systems in Eq. (3.5)
Compute $z_1 = R_p^0 \boldsymbol{v}$ and $z_2 = R_p \boldsymbol{v}$ via FFTs
Compute $z_3 = L_p^0 z_1$ and $z_4 = L_p z_2$ via FFTs
Compute $z = \frac{1}{\xi_0}(z_4 - z_3)$
```

In summary, we need to search some efficient solvers for the nonsymmetric resulted Toeplitz linear systems, whether to solve (3.3) or (3.5). In next subsection, we will introduce how to build efficient preconditioned iterative solvers for nonsymmetric Toeplitz linear systems.

3.2 Fast Implementation of IDS Based on Preconditioned Iterative Solvers

In this subsection, we discuss the detailed framework about implementing the proposed implicit difference scheme (2.1). For the sake of clarity, an algorithm for implementing the implicit difference scheme is given in Algorithm 2.

Algorithm 2 Practical implementation of IDS

```
for $j = 0, 1, \ldots, M-1$, do
    Compute $\boldsymbol{g}^{(j+1)} = B^{(j+\sigma)}\boldsymbol{u}^{(j)} + \delta\boldsymbol{u}^{(j)} + \boldsymbol{f}^{(j+\sigma)}$
    Solve $A^{(j+\sigma)}\boldsymbol{u}^{(j+1)} = \boldsymbol{g}^{(j+1)}$
end for
```

From Algorithm 2, M real linear systems are needed to be solved. Direct solvers based on variants of Gaussian elimination [67, pp. 33–44]) are prohibitively expensive when N is large, due to $\mathcal{O}(MN^3)$. Exploiting the Toeplitz form of matrices $A^{(j+\sigma)}$ and $B^{(j+\sigma)}$, the matrix–vector product $B^{(j+\sigma)}\boldsymbol{u}^{(j)}$ in Step 2 can be evaluated by FFTs in $\mathcal{O}(N \log N)$ operations, and fast iterative Toeplitz methods can be used to solve the linear systems in Step 3 of Algorithm 2 in $\mathcal{O}(N \log N)$ arithmetic operations per iteration step. A thorough discussion on itertive methods for Toeplitz systems can be found in [59] and in [8,50,60,61,66,69] for their application to solving space fractional diffusion equations. In this study, we employ a Krylov subspace solver namely the conjugate gradient squared (CGS) method [63, pp. 241-244] with the circulant preconditoner

$$P^{(j+\sigma)} = \eta_j I - \sigma\Big(\frac{\gamma^{(j+\sigma)}}{2h}s(Q) + \frac{D_+^{(j+\sigma)}}{h^\beta}s(W_\beta) + \frac{D_-^{(j+\sigma)}}{h^\beta}s(W_\beta^T)\Big), \tag{3.7}$$

where $s(\cdot)$ means the well-known Strang circulant approximation of a given Toeplitz matrix [59,61] for solving $A^{(j+\sigma)}\boldsymbol{u}^{(j+1)} = \boldsymbol{g}^{(j+1)}$. The high efficiency of Strang circulant preconditioner for space FDEs has been demonstrated in [61]. Below we show that the preconditioner $P^{(j+\sigma)}$ defined in (3.7) is nonsingular, thus is well-defined.

Lemma 3.1 *All eigenvalues of $s(W_\beta)$ and $s(W_\beta^T)$ fall inside the open disc*

$$\{z \in \mathbb{C} : |z - \omega_1^{(\beta)}| < -\omega_1^{(\beta)}\}. \tag{3.8}$$

Proof All the Gershgorin disc [63, pp. 119–122] of the circulant matrices $s(W_\beta)$ and $s(W_\beta^T)$ are centered at $-\omega_1^{(\beta)} > 0$ with radius

$$r_N = \omega_0^{(\beta)} + \sum_{k=2}^{\lfloor \frac{N}{2} \rfloor} \omega_k^{(\beta)} < \sum_{k=0,k\neq 1}^{\infty} \omega_k^{(\beta)} = -\omega_1^{(\beta)}. \tag{3.9}$$

by the properties of the sequence $\{\omega_k^{(\beta)}\}$; refer to Lemmas 2.3 and 2.4. □

Remark 3.1 It is worth to mention that:

1. The real parts of all eigenvalues of $s(W_\beta)$ and $s(W_\beta^T)$ are strictly negative for all N;

2. The absolute values of all eigenvalues of $s(W_\beta)$ and $s(W_\beta^T)$ are bounded above by $2|\omega_1^{(\beta)}|$ for all N.

As we know, a circulant matrix can be quickly diagonalized by the Fourier matrix F [28,59,61]. Then it follows that $s(Q) = F^* \Lambda_q F$, $s(W_\beta) = F^* \Lambda_\beta F$, and $s(W_\beta^T) = F^* \bar{\Lambda}_\beta F$, where $\bar{\Lambda}_\beta$ is the complex conjugate of Λ_β. Decompose the circulant matrix $P^{(j+\sigma)} = F^* \Lambda_p F$ with the diagonal matrix $\Lambda_p = \eta_j I - \sigma\Big(\frac{\gamma^{(j+\sigma)}}{2h}\Lambda_q + \frac{D_+^{(j+\sigma)}}{h^\beta}\Lambda_\beta + \frac{D_-^{(j+\sigma)}}{h^\beta}\bar{\Lambda}_\beta\Big)$. Then $P^{(j+\sigma)}$ is invertible if all diagonal entries of Λ_p are nonzero. Moreover, we can obtain the following conclusion about the invertibility of $P^{(j+\sigma)}$ in (3.7).

Theorem 3.1 *The circulant preconditioners $P^{(j+\sigma)}$ defined as in* (3.7) *are nonsingular.*

Proof We already know that Q is a skew-symmetric Toeplitz matrix. However, $s(Q)$ is also a skew-symmetric circulant matrix, thus the real part of Λ_q is equal to zero, i.e., $\mathrm{Re}([\Lambda_q]_{k,k}) = 0$. On the other hand, by Part 1 of Remark 3.1, we have $\mathrm{Re}([\Lambda_\beta]_{k,k}) < 0$. Noting that $\eta_j > 0$, $\sigma > 0$, and $D_\pm^{(j+\sigma)} \geq 0$, thus we obtain

$$\mathrm{Re}([\Lambda_p]_{k,k}) = \eta_j - \sigma\Big(0 + \frac{D_+^{(j+\sigma)}}{h^\beta}\mathrm{Re}([\Lambda_\beta]_{k,k}) + \frac{D_-^{(j+\sigma)}}{h^\beta}\mathrm{Re}([\bar{\Lambda}_\beta]_{k,k})\Big) \neq 0,$$

for each $k = 1, 2, \ldots, N-1$. Therefore, $P^{(j+\sigma)}$ are invertible. □

Although we do not investigate the eigenvalue distributions of preconditioned matrices $(P^{(j+\sigma)})^{-1}A^{(j+\sigma)}$ theoretically, we still analyse in next section the favourable clustering of the spectra for several preconditioned matrices numerically. In our experiments, the iteration numbers always fluctuate between 4 and 11, ensuring an $\mathcal{O}(N\log N)$ complexity for solving the linear system in Step 3 and an overall computational complexity of $\mathcal{O}(MN\log N)$ for implementing the proposed IDS.

Beside, if we have the coefficients $\gamma(t) = \gamma$ and $d_\pm(t) = d_\pm$ in Eq. (1.1), the matrix $B^{(j+\sigma)}$ also has the similar form as $B^{(j+\sigma)}$ in Eq. (3.4), then we can simplify Algorithm 2 as the following Algorithm 3.

Algorithm 3 Practical implementation of IDS with constant coefficients

```
1: for $j = 0, 1, \ldots, M-1$, do
2:   if $j = 0$ then
3:     Compute $\boldsymbol{g}^{(1)} = B^{(\sigma)}\boldsymbol{u}^{(0)} + \boldsymbol{f}^{(\sigma)}$
4:     Solve $A^{(\sigma)}\boldsymbol{u}^{(1)} = \boldsymbol{g}^{(1)}$
5:   else
6:     Compute $\boldsymbol{g}^{(j+1)} = B\boldsymbol{u}^{(j)} + \delta\boldsymbol{u}^{(j)} + \boldsymbol{f}^{(j+\sigma)}$
7:     Solve $\boldsymbol{u}^{(j+1)} = A^{-1}\boldsymbol{g}^{(j+1)}$
8:   end if
9: end for
```

Again, if a direct method is employed to solve the linear system in Step 4 of Algorithm 3, the LU decomposition can be reused in Step 7 of Algorithm 3; however, the complexity will be still $\mathcal{O}(MN^3)$, which is too costly if N is large. On the other hand, utilizing the Toeplitz structure of those four matrices in Steps 3, 4, 6 and 7 of Algorithm 3, two matrix–vector multiplications $B^{(\sigma)}\boldsymbol{u}^{(0)}$ and $B\boldsymbol{u}^{(j)}$ in Steps 3 and 6 can be evaluated by FFTs in $\mathcal{O}(N\log N)$ operations. Then fast Toeplitz iterative solvers with suitable circulant preconditioners can be applied to solve the Toeplitz system in Step 4 and (3.5). Here we can construct two circulant preconditioners defined as

$$P^{(\sigma)} = \eta_0 I - \sigma\Big(\frac{\gamma}{2h}s(Q) + \frac{D_+}{h^\beta}s(W_\beta) + \frac{D_-}{h^\beta}s(W_\beta^T)\Big), \tag{3.10}$$

$$P = \eta_j I - \sigma\Big(\frac{\gamma}{2h}s(Q) + \frac{D_+}{h^\beta}s(W_\beta) + \frac{D_-}{h^\beta}s(W_\beta^T)\Big), \tag{3.11}$$

for the linear systems in Step 4 and (3.5), respectively. Note that the invertibility of the two circulant preconditioners introduced in (3.10–3.11) can be similarly proved from Theorem 3.1. Then we can employ Algorithm 1 to evaluate the matrix–vector multiplication $A^{-1}\boldsymbol{g}^{(j+1)}$ in Step 7 of Algorithm 3. Similarly, we will show in next section that the iteration numbers required by preconditioned Krylov subspace methods always range between 4 and 11.[1] In this case, the algorithmic complexity of preconditioned Krylov subspace methods is only $\mathcal{O}(N\log N)$ at each iteration step, leading to the total complexity of $\mathcal{O}(MN\log N)$ operations for implementing the IDS with constant coefficients.

4 Numerical Results

The numerical experiments presented in this section have a two-fold objective. They illustrate that the proposed IDS can indeed converge with the second-order accuracy in both space and time. At the same time, they assess the computational efficiency of the fast solution techniques (i.e., Algorithms 1, 2, and 3) described in Sect. 3. For Krylov subspace method and direct solver, we choose built-in functions for the preconditioned CGS (PCGS) method, LU factorization of MATLAB in Example 1 and MATLAB's backslash in Example 2, respectively. For the CGS method with circulant preconditioners, the stopping criterion of those methods is $\|\boldsymbol{r}^{(k)}\|_2/\|\boldsymbol{r}^{(0)}\|_2 < 10^{-12}$, where $\boldsymbol{r}^{(k)}$ is the residual vector of the linear system after k iterations, and the initial guess is chosen as the zero vector. All experiments were performed on a Windows 7 (32 bit) PC-Intel(R) Core(TM) i5-3470 CPU 3.20 GHz, 4 GB of RAM using MATLAB 2014a with machine epsilon 10^{-16} in double precision floating point arithmetic. By the way, all timings are averages over 20 runs of our algorithms.

Example 1 We solved Eq. (1.1) on the space interval $[a,b] = [0,1]$ and time interval $[0,T] = [0,1]$, with diffusion coefficients $d_+(t) = d_+ = 0.8,\ d_-(t) = d_- = 0.5$, convection coefficient $\gamma(t) = \gamma = -0.1$. The initial condition was $u(x,0) = x^2(1-x)^2$, and the source term was

$$\begin{aligned} f(x,t) = {} & \frac{\Gamma(3+\alpha)}{2}x^2(1-x)^2t^2 - (t^{2+\alpha}+1)\Big\{2\gamma x(1-x)(1-2x) + \frac{\Gamma(3)}{\Gamma(3-\beta)}[d_+x^{2-\beta} \\ & + d_-(1-x)^{2-\beta}] - \frac{2\Gamma(4)}{\Gamma(4-\beta)}[d_+x^{3-\beta} + d_-(1-x)^{3-\beta}] + \frac{\Gamma(5)}{\Gamma(5-\beta)}[d_+x^{4-\beta} \\ & + d_-(1-x)^{4-\beta}]\Big\}. \end{aligned}$$

This problem is modified from [24, Example 6]. and has exact solution $u(x,t) = (t^{2+\alpha} + 1)x^2(1-x)^2$. In the finite difference discretization, the space step and time step are taken to be $h = 1/N$ and $\tau = h$, respectively.

[1] In this case, it should mention that we only need to solve three nonsymmetric Toeplitz systems, i.e., equations with the form like (3.5) and in Step 4 of Algorithm 3, for implementing the whole `for` loop.

Table 1 L_2-norm and maximum norm error behavior versus grid size reduction when $\tau = h$ and $\beta = 1.8$ in Example 1

α	h	$\max_{0\le n\le M} \|E^n\|_0$	CO in $\|\cdot\|_0$	$\|E\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$	CO in $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$
0.10	1/32	2.7954e−4	–	4.0880e−4	–
	1/64	6.6775e−5	2.0657	9.8580e−5	2.0520
	1/128	1.6010e−5	2.0603	2.3815e−5	2.0494
	1/256	3.8514e−6	2.0556	5.7630e−6	2.0470
0.50	1/32	2.6670e−4	–	3.8874e−4	–
	1/64	6.3583e−5	2.0685	9.3590e−5	2.0544
	1/128	1.5219e−5	2.0628	2.2573e−5	2.0518
	1/256	3.6558e−6	2.0576	5.4539e−6	2.0492
0.90	1/32	2.4972e−4	–	3.6255e−4	–
	1/64	5.9441e−5	2.0708	8.7173e−5	2.0562
	1/128	1.4206e−5	2.0650	2.0993e−5	2.0540
	1/256	3.4078e−6	2.0596	5.0762e−6	2.0481
0.99	1/32	2.5899e−4	–	3.7959e−4	–
	1/64	6.2121e−5	2.0598	9.1923e−5	2.0460
	1/128	1.4944e−5	2.0555	2.2275e−5	2.0450
	1/256	3.6057e−6	2.0512	5.4042e−6	2.0433

The quantities reported in Tables 1 and 2 are the error ($E = U - u$) and the convergence order (CO = $\log_{h_1/h_2} \frac{\|E_1\|}{\|E_2\|}$, where E_i is the error corresponding to h_i) computed in the norms $\|\cdot\|_0$ and $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$ with $\|U\|_{\mathcal{C}(\bar{\omega}_{h\tau})} = \max_{(x_i,t_j)\in\bar{\omega}_{h\tau}} |U|$. In Tables 5 and 6 we report on the performance of the fast solvers presented in Sect. 3. The quantity denoted as "Speed-up" is defined as

$$\texttt{Speed-up} = \frac{\texttt{Time1}}{\texttt{Time2}}.$$

where Time2 is the computing time required by the fast IDS for solving problem, and Time1 is the time required by Algorithm 3 that reuses the LU decomposition. Obviously, when $\texttt{Speed-up} > 1$, it means that Time2 needed by our proposed method is more competitive than Time1 required by Algorithm 3 with reusing LU decomposition in aspects of the CPU time elapsed.

We notice from Table 1 that the maximum error decreased when the space grid size h and time step size $h = \tau$ were reduced. The convergence order of the approximate scheme is $\mathcal{O}(h^2) = \mathcal{O}(\tau^2)$. In Table 2, we fixed $h = 1/1000$ and increasing the number of time steps (i.e., M); a reduction in the maximum error was observed, and the convergence order of time is $\mathcal{O}(\tau^2)$, where the convergence order is given by the following formula: CO = $\log_{\tau_1/\tau_2} \frac{\|E_1\|}{\|E_2\|}$. Finally, to further illustrate the reliability of our proposed scheme, Fig. 1 are plotted to show that the 'quadratic-type' order of accuracy is achieved in both time and space directions.

Since it is important to investigate convergence behaviors of the proposed method in the cast that the convection term of Eq. (1.1) becomes more dominant, so we choose a large coefficient $\gamma = -400$ and we reduced the step-size $h = \tau$ in Table 3, while in Table 4 we used a constant $h = 1/1600$ and decreased τ. The same numerical trend was observed as in the case of a less dominant convection term. This is evident both from Tables 3 and

Table 2 L_2-norm and maximum norm error behavior versus τ-grid size reduction when $h = 1/1000$ and $\beta = 1.8$ in Example 1

α	τ	$\max_{0\leq n\leq M} \|E^n\|_0$	CO in $\|\cdot\|_0$	$\|E\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$	CO in $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$
0.10	1/5	7.6009e−5	–	1.2059e−4	–
	1/10	1.9209e−5	1.9843	3.0437e−5	1.9861
	1/20	4.6741e−6	2.0390	7.4069e−6	2.0389
	1/40	1.0134e−6	2.2054	1.6095e−6	2.2023
0.50	1/5	5.0068e−4	–	7.9189e−4	–
	1/10	1.2639e−4	1.9860	1.9985e−4	1.9864
	1/20	3.1564e−5	2.0015	4.9914e−5	2.0014
	1/40	7.7315e−6	2.0295	1.2232e−5	2.0288
0.90	1/5	9.9593e−4	–	1.5728e−3	–
	1/10	2.4927e−4	1.9983	3.9380e−4	1.9978
	1/20	6.2151e−5	2.0039	9.8203e−5	2.0036
	1/40	1.5356e−5	2.0170	2.4272e−5	2.0165
0.99	1/5	1.0964e−3	–	1.7304e−3	–
	1/10	2.7402e−4	2.0005	4.3269e−4	1.9997
	1/20	6.8333e−5	2.0036	1.0791e−4	2.0035
	1/40	1.6913e−5	2.0145	2.6714e−5	2.0142

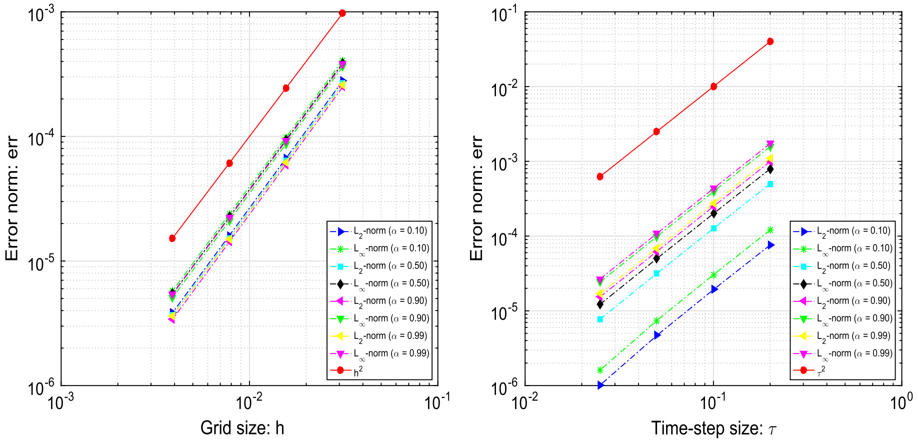

Fig. 1 Comparison the order of accuracy obtained by our proposed schemes for Example 1 in space and time variables. *Left* space direction; *Right* time direction

4 and from the plot in 2. In our experiments, no noticeable instabilities have arisen in the approximation solution when the convection term becomes more dominant.

In Figs. 3 and 4, the eigenvalues of both the original matrix $A^{(j+\sigma)}$ and the preconditioned matrix $(P^{(j+\sigma)})^{-1}A^{(j+\sigma)}$ are plotted. These two figures confirm that the circulant preconditioning exhibits very nice clustering properties. The eigenvalues of $(P^{(j+\sigma)})^{-1}A^{(j+\sigma)}$ are well grouped around 1 expect for a few (about $5 \sim 11$) outliers. The vast majority of the eigenvalues are well separated away from 0. It may be interpreted as that in our whole implementation the number of iterations required by preconditioned Krylov subspace methods for

Table 3 L_2-norm and maximum norm error behavior versus grid size reduction when $\tau = h,\ d_1 = 0.6,\ d_2 = 0.7,\ \gamma = -400$ and $\beta = 1.9$ in Example 1

α	h	$\max_{0\le n\le M} \|E^n\|_0$	CO in $\|\cdot\|_0$	$\|E\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$	CO in $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$
0.10	1/32	7.0004e−4	–	9.6167e−4	–
	1/64	1.7386e−4	2.0095	2.3901e−4	2.0085
	1/128	4.3044e−5	2.0141	5.9287e−5	2.0113
	1/256	1.0626e−5	2.0182	1.4673e−5	2.0146
0.50	1/32	6.8858e−4	–	9.4341e−4	–
	1/64	1.7101e−4	2.0095	2.3446e−4	2.0015
	1/128	4.2330e−5	2.0143	5.8151e−5	2.0014
	1/256	1.0448e−5	2.0185	1.4389e−5	2.0017
0.90	1/32	6.7575e−4	–	9.2302e−4	–
	1/64	1.6780e−4	2.0098	2.2936e−4	2.0088
	1/128	4.1529e−5	2.0145	5.6878e−5	2.0117
	1/256	1.0247e−5	2.0185	1.4071e−5	2.0152
0.99	1/32	7.1465e−4	–	9.9367e−4	–
	1/64	1.7239e−4	2.0516	2.3711e−4	2.0672
	1/128	4.2694e−5	2.0136	5.8841e−5	2.0107
	1/256	1.0542e−5	2.0179	1.4566e−5	2.0142

Table 4 L_2-norm and maximum norm error behavior versus τ-grid size reduction when $h = 1/1600,\ d_1 = 0.6,\ d_2 = 0.7,\ \gamma = -400$ and $\beta = 1.9$ in Example 1

α	τ	$\max_{0\le n\le M} \|E^n\|_0$	CO in $\|\cdot\|_0$	$\|E\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$	CO in $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$
0.10	1/5	8.6540e−5	–	1.3583e−4	–
	1/10	2.1512e−5	2.0083	3.3795e−5	2.0070
	1/20	5.1923e−6	2.0507	8.1879e−6	2.0452
	1/40	1.1252e−6	2.2062	1.7960e−6	2.1887
0.50	1/5	5.5079e−4	–	8.6424e−4	–
	1/10	1.3856e−4	1.9910	2.1744e−4	1.9908
	1/20	3.4587e−5	2.0022	5.4308e−5	2.0014
	1/40	8.4703e−6	2.0297	1.3331e−5	2.0264
0.90	1/5	1.0809e−3	–	1.6961e−3	–
	1/10	2.7042e−4	1.9989	4.2431e−4	1.9991
	1/20	6.7513e−5	2.0020	1.0597e−4	2.0015
	1/40	1.6696e−5	2.0157	2.6236e−5	2.0140
0.99	1/5	1.1826e−3	–	1.8559e−3	–
	1/10	2.9603e−4	1.9982	4.6443e−4	1.9986
	1/20	7.3789e−5	2.0043	1.1582e−4	2.0036
	1/40	1.8239e−5	2.0164	2.8650e−5	2.0152

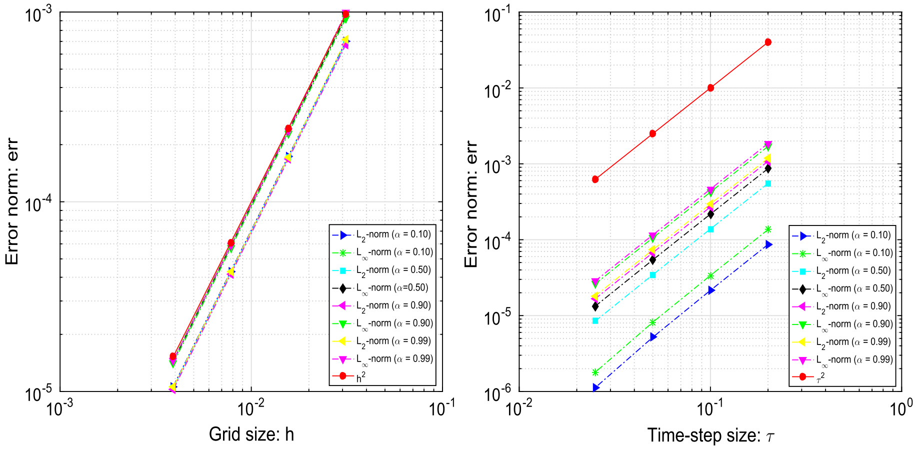

Fig. 2 Comparison the order of accuracy obtained by our proposed schemes for Example 1 in space and time variables. *Left* space direction; *Right* time direction

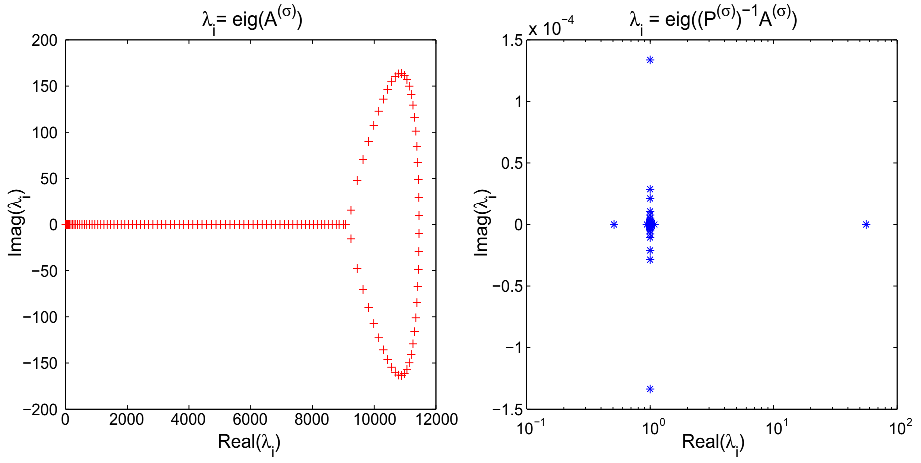

Fig. 3 Spectrum of both original and preconditioned matrices at the time level $j = 0$, respectively, when $N = M = 128$, $\alpha = 0.9$ and $\beta = 1.8$. *Left* original matrix; *Right* circulant preconditioned matrix

solving three targeted nonsymmetric Toeplitz systems[2] (i.e., equations with the form like (3.5) and in Step 4 of Algorithm 3) almost ranges from 5 to 11. We validate the effectiveness and robustness of the designed circulant preconditioner from the perspective of clustering spectra distribution.

Tables 5 and 6 illustrate that the proposed fast direct solver for different discretized problems takes much less CPU time elapsed as M and N become large. When $M = N = 2^{10}$ and different discretized parameters, the CPU time of Algorithm 3 is about 14.5 seconds, the speedup is more than three times. Meanwhile, although `Time1` required by Algorithm 3 for small test problems ($M = N = 32, 64, 128$) is lower than `Time2` needed by Algorithm 3, our proposed method is still more attractive in terms of the lower memory requirement.

[2] For the sake of clarity, here we do not list the number of iterations required for solving those three linear systems one by one.

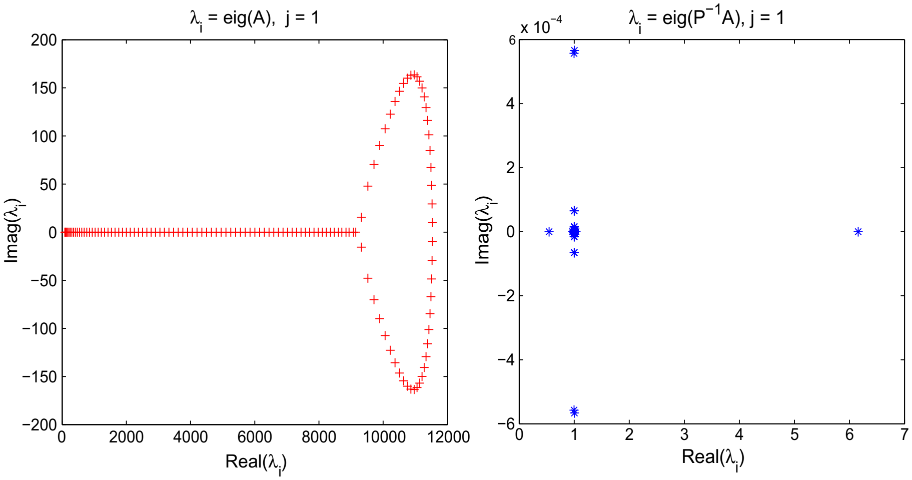

Fig. 4 Spectrum of both original and preconditioned matrices at the time level $j = 1$, respectively, when $N = M = 128$, $\alpha = 0.9$ and $\beta = 1.8$. *Left* original matrix; *Right* circulant preconditioned matrix

Table 5 CPU time in seconds for solving Example 1 with $\alpha = 0.9$ that `Time1` is done by for Algorithm 3 (LU decomposition) and `Time2` is done by Algorithm 3 with Algorithm 1

$h = \tau$	$\beta = 1.2$			$\beta = 1.5$			$\beta = 1.8$		
	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`
2^{-5}	0.003	0.010	0.30	0.003	0.010	0.30	0.003	0.010	0.30
2^{-6}	0.011	0.017	0.65	0.011	0.017	0.65	0.011	0.017	0.69
2^{-7}	0.051	0.061	0.84	0.052	0.061	0.85	0.051	0.061	0.84
2^{-8}	0.492	0.253	1.91	0.493	0.255	1.93	0.496	0.257	1.93
2^{-9}	4.544	1.714	2.65	4.548	1.713	2.65	4.550	1.714	2.65
2^{-10}	44.532	14.436	3.08	44.562	14.571	3.06	44.548	14.446	3.08

Table 6 CPU time in seconds for solving Example 1 with $\beta = 1.8$ that `Time1` is done by for Algorithm 3 (LU decomposition) and `Time2` is done by Algorithm 3 with Algorithm 1

$h = \tau$	$\alpha = 0.1$			$\alpha = 0.5$			$\alpha = 0.9$		
	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`
2^{-5}	0.003	0.010	0.30	0.003	0.010	0.30	0.003	0.010	0.30
2^{-6}	0.011	0.017	0.65	0.011	0.017	0.65	0.011	0.017	0.65
2^{-7}	0.051	0.061	0.84	0.052	0.061	0.85	0.051	0.061	0.85
2^{-8}	0.494	0.254	1.94	0.495	0.254	1.95	0.494	0.253	1.28
2^{-9}	4.562	1.716	2.66	4.547	1.716	2.65	4.549	1.717	2.65
2^{-10}	44.560	14.452	3.08	44.553	14.448	3.08	44.547	14.466	3.08

Table 7 CPU time in seconds for solving Example 1 with $\alpha = 0.99$ that `Time1` is done by for Algorithm 3 (LU decomposition) and `Time2` is done by Algorithm 3 with Algorithm 1

$h = \tau$	$\beta = 1.8$			$\beta = 1.9$			$\beta = 1.99$		
	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`
2^{-5}	0.003	0.009	0.33	0.003	0.009	0.33	0.003	0.009	0.33
2^{-6}	0.012	0.016	0.75	0.012	0.016	0.75	0.012	0.016	0.75
2^{-7}	0.060	0.061	0.98	0.060	0.059	1.02	0.060	0.059	1.02
2^{-8}	0.536	0.255	2.10	0.539	0.255	2.11	0.531	0.252	2.11
2^{-9}	5.018	1.711	2.93	4.725	1.701	2.78	4.653	1.693	2.75
2^{-10}	44.568	14.408	3.09	44.574	14.393	3.10	44.587	14.401	3.10

Table 8 CPU time in seconds for solving Example 1 with $\beta = 1.99$ that `Time1` is done by for Algorithm 3 (LU decomposition) and `Time2` is done by Algorithm 3 with Algorithm 1

$h = \tau$	$\alpha = 0.1$			$\alpha = 0.5$			$\alpha = 0.9$		
	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`
2^{-5}	0.003	0.009	0.33	0.003	0.009	0.33	0.003	0.009	0.33
2^{-6}	0.012	0.016	0.75	0.012	0.016	0.75	0.012	0.016	0.75
2^{-7}	0.059	0.059	1.00	0.060	0.059	1.02	0.060	0.059	1.02
2^{-8}	0.538	0.256	2.10	0.540	0.255	2.11	0.536	0.254	2.11
2^{-9}	4.704	1.707	2.76	4.676	1.698	2.75	4.669	1.700	2.75
2^{-10}	44.559	14.389	3.10	44.565	14.392	3.10	44.548	14.378	3.10

Compared to Algorithm 3 with reusing the LU decomposition, it highlighted that in the whole implementation the proposed solution technique does not require to store the full matrices (e.g., some matrices $A^{(\sigma)}$, A and their LU decomposition factors) at all. In short, we can conclude that our proposed IDS with fast implementation is still more competitive than the IDS with reusing the conventional LU decomposition.

In Tables 7 and 8, we investigate the performance of our proposed preconditioners for handling discretized nonsymmetric Toeplitz systems if the convection term appearing in Eq. (1.1) becomes more dominate. Here we take $d_1 = 0.6$, $d_2 = 0.7$ and $\gamma = -400$ for making the discretized linear systems very ill-conditioned. It illustrates that the proposed fast solver still requires much less CPU time as M and N become larger. When $M = N = 2^{10}$, the CPU time of Algorithm 3 is only 14.4 seconds, the speedup factor is more than 3. On the other hand, although `Time1` required by Algorithm 3 with the LU factorization for small test problems ($M = N = 32, 64$) is lower than `Time2` needed by Algorithm 3 with Algorithm 1, our proposed method is still more attractive memory-wise. Compared to Algorithm 3 with reusing the LU decomposition, it remarked that the practical implementation of the proposed solution techniques does not need to store the full matrices (e.g., some matrices $A^{(\sigma)}$, A and their LU decomposition factors) at all.

Similarly, here we present two eigenvalue plots about both original and preconditioned matrices in Figs. 5 and 6. Compare with Figs. 3 and 4, we can see that when the convection term of Eq. (1.1) become more dominate, the discretized nonsymmetric linear systems indeed tend to be more ill-conditioned in terms of "scattering" eigenvalue distribution plots (i.e.,

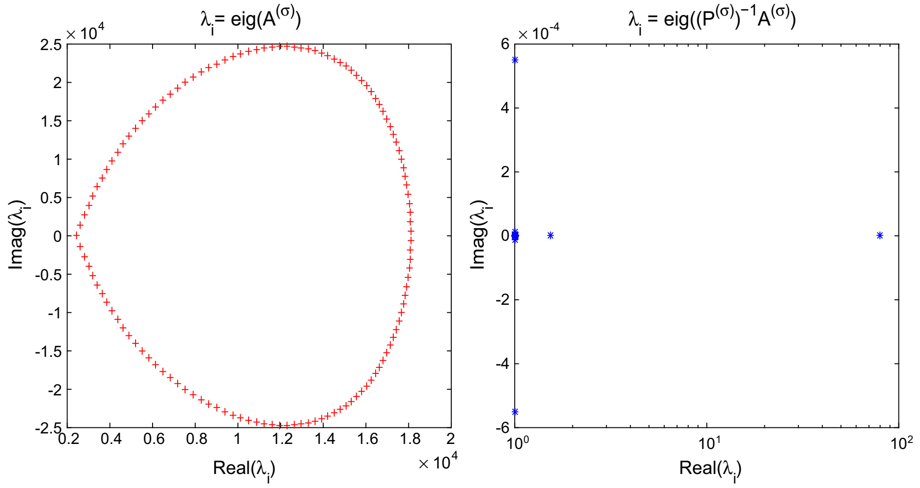

Fig. 5 Spectrum of both original and preconditioned matrices at the time level $j = 0$, respectively, when $d_+ = 0.6$, $d_- = 0.7$, $\gamma = -400$, $N = M = 128$, $\alpha = 0.99$ and $\beta = 1.9$. *Left* original matrix; *Right* circulant preconditioned matrix

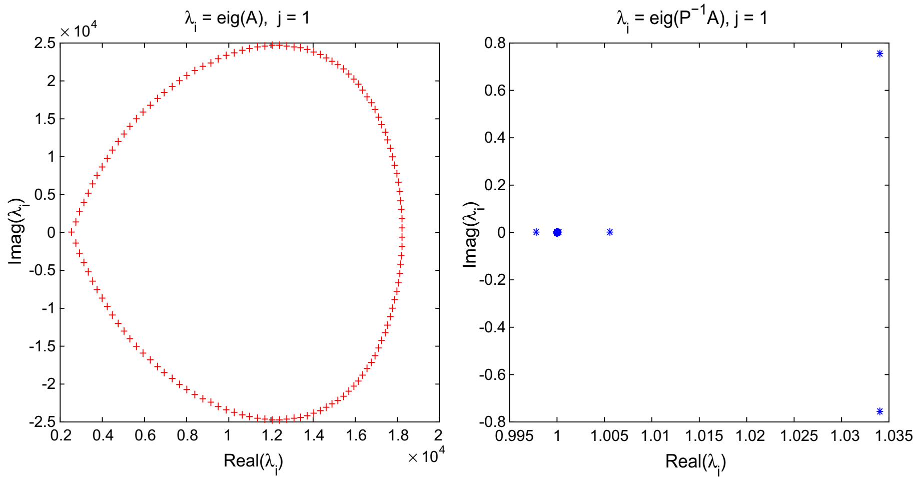

Fig. 6 Spectrum of both original and preconditioned matrices at the time level $j = 1$, respectively, when $d_+ = 0.6$, $d_- = 0.7$, $\gamma = -400$, $N = M = 128$, $\alpha = 0.99$ and $\beta = 1.9$. *Left* original matrix; *Right* circulant preconditioned matrix

Figs. 5, 6). These two figures confirm that for circulant preconditioners, the eigenvalues of preconditioned matrices are clustered at 1, expect for a few (about $4 \sim 8$) outliers. The vast majority of the eigenvalues are well separated away from 0. It may guarantee that our proposed preconditioners can deeply accelerate Krylov subspace methods for solving three targeted nonsymmetric Toeplitz systems (i.e., equations with the form like (3.5) and in Step 4 of Algorithm 3). We validate the effectiveness and robustness of the designed circulant preconditioner from the perspective of clustering spectrum distribution.

Table 9 L_2-norm and maximum norm error behavior versus grid size reduction when $\tau = h$ and $\beta = 1.3$ in Example 2

α	h	$\max_{0\leq n\leq M}\|E^n\|_0$	CO in $\|\cdot\|_0$	$\|E\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$	CO in $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$
0.10	1/32	3.1941e−4	–	5.6886e−4	–
	1/64	7.6298e−5	2.0657	1.6055e−4	1.8250
	1/128	1.8397e−5	2.0521	4.2694e−5	1.9110
	1/256	4.4694e−6	2.0414	1.1036e−5	1.9519
0.50	1/32	3.0866e−4	–	5.6897e−4	–
	1/64	7.3673e−5	2.0668	1.6054e−4	1.8254
	1/128	1.7757e−5	2.0527	4.2689e−5	1.9110
	1/256	4.3137e−6	2.0414	1.1035e−5	1.9518
0.90	1/32	2.9880e−4	–	5.6951e−4	–
	1/64	7.1478e−5	2.0636	1.6058e−4	1.8264
	1/128	1.7232e−5	2.0524	4.2691e−5	1.9113
	1/256	4.1814e−6	2.0430	1.1034e−5	1.9519
0.99	1/32	3.2304e−4	–	5.7367e−4	–
	1/64	7.7278e−5	2.0638	1.6119e−4	1.8314
	1/128	1.8633e−5	2.0522	4.2748e−5	1.9149
	1/256	4.5227e−6	2.0426	1.1035e−5	1.9538

Example 2 In the last test, we investigate the proposed method for solving Eq. (1.1) on the space interval $[a, b] = [0, 1]$ and the time interval $[0, T] = [0, 1]$ with diffusion coefficients $d_+(t) = 9\sin(t)$, $d_-(t) = 4\sin(t)$, convection coefficient $\gamma(t) = -t$, initial condition $u(x, 0) = x^2(1-x)^2$, and source term

$$\begin{aligned} f(x,t) =& \frac{\Gamma(3+\alpha)}{2}t^2x^2(1-x)^2 - (t^{2+\alpha}+1)\Big\{-2tx(1-x)(1-2x) + \frac{\Gamma(3)\sin(t)}{\Gamma(3-\beta)}[9x^{2-\beta} \\ &+ 4(1-x)^{2-\beta}] - \frac{2\Gamma(4)\sin(t)}{\Gamma(4-\beta)}[9x^{3-\beta} + 4(1-x)^{3-\beta}] + \frac{\Gamma(5)\sin(t)}{\Gamma(5-\beta)}[9x^{4-\beta} \\ &+ 4(1-x)^{4-\beta}]\Big\}. \end{aligned}$$

This example can be viewed as a variant of [56, Example 1]. The exact solution of this example is defined as $u(x,t) = (t^{2+\alpha}+1)x^2(1-x)^2$ For the implicit finite difference discretization, the space step and time step are taken to be $h = 1/N$ and $\tau = h$, respectively. The experiment results about the proposed IDS for Example 2 are reported in Tables 9 and 10. Furthermore, the effectiveness of fast solution techniques presented in Sect. 3 for this example will be illustrated in Tables 11 and 12.

According to the numerical results illustrated in Table 9, it finds that as the number of the spatial subintervals and time steps is increased keeping $h = \tau$, a reduction in the maximum error takes place, as expected and the convergence order of the approximate scheme is $\mathcal{O}(h^2) = \mathcal{O}(\tau^2)$, where the convergence order is given by the formula: CO = $\log_{h_1/h_2}\frac{\|E_1\|}{\|E_2\|}$ (E_i is the error corresponding to h_i). On the other hand, Table 10 illustrates that if $h = 1/1800$, then as the number of time steps of our approximate scheme is increased, a reduction in the maximum error takes place, as expected and the convergence order of time is $\mathcal{O}(\tau^2)$, where the convergence order is given by the following formula: CO = $\log_{\tau_1/\tau_2}\frac{\|E_1\|}{\|E_2\|}$. In order to

Table 10 L_2-norm and maximum norm error behavior versus τ-grid size reduction when $h = 1/1800$ and $\beta = 1.3$ in Example 2

α	τ	$\max_{0\le n\le M}\|E^n\|_0$	CO in $\|\cdot\|_0$	$\|E\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$	CO in $\|\cdot\|_{\mathcal{C}(\bar{\omega}_{h\tau})}$
0.10	1/6	5.7382e−5	–	9.0504e−5	–
	1/12	1.4405e−5	1.9940	2.2699e−5	1.9954
	1/24	3.5630e−6	2.0154	5.6036e−6	2.0182
	1/48	8.4188e−7	2.0814	1.3128e−6	2.0937
0.50	1/6	3.6999e−4	–	5.8276e−4	–
	1/12	9.3087e−5	1.9908	1.4660e−4	1.9910
	1/24	2.3295e−5	1.9986	3.6679e−5	1.9989
	1/48	5.7803e−6	2.0108	9.0927e−6	2.0122
0.90	1/6	7.2848e−4	–	1.1469e−3	–
	1/12	1.8230e−4	1.9986	2.8697e−4	1.9988
	1/24	4.5551e−5	2.0008	7.1699e−5	2.0009
	1/48	1.1337e−5	2.0065	1.7836e−5	2.0071
0.99	1/6	7.9927e−4	–	1.2586e−3	–
	1/12	1.9978e−4	2.0006	3.1431e−4	2.0015
	1/24	4.9882e−5	2.0014	7.8492e−5	2.0016
	1/48	1.2419e−5	2.0060	1.9533e−5	2.0066

Table 11 CPU time in seconds for Example 2 with $\alpha = 0.9$ that `Time1` is done by for Algorithm 2 (MATLAB's backslash) and `Time2` (*Iters*) is done by Algorithm 2 with PCGS solver

$h = \tau$	$\beta = 1.3$			$\beta = 1.5$			$\beta = 1.9$		
	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`
2^{-5}	0.03	0.06 (6.0)	0.50	0.03	0.06 (6.8)	0.50	0.03	0.06 (6.0)	0.50
2^{-6}	0.09	0.13 (6.0)	0.69	0.09	0.13 (6.8)	0.69	0.09	0.13 (6.0)	0.69
2^{-7}	0.46	0.32 (6.0)	1.44	0.47	0.35 (7.0)	1.34	0.46	0.35 (7.0)	1.31
2^{-8}	4.04	0.88 (7.0)	4.59	4.05	0.91 (7.7)	4.45	4.07	0.89 (7.0)	4.57
2^{-9}	36.91	3.53 (7.0)	10.46	36.97	3.71 (8.0)	9.96	37.03	3.57 (7.0)	10.37
2^{-10}	375.68	20.10 (7.0)	18.69	376.59	20.89 (8.0)	18.03	377.02	20.28 (7.0)	18.59

further verify the reliability of our proposed scheme, Fig. 7 are plotted to illustrate that the 'quadratic-type' order of accuracy is achieved in both time and space variables.

Again, for the case of variable time coefficients, several eigenvalue plots about both original and preconditioned matrices are similarly displayed in Figs. 8 and 9. These two figures confirm that for circulant preconditioning, the eigenvalues of preconditioned matrices are clustered at 1, expect for a few (about 6 $\sim$ 10) outliers. The vast majority of the eigenvalues are well separated away from 0. It may be mainly interpreted as that in our implementation the number of iterations needed by PCGS with circulant preconditioners almost ranges from 6 to 10. We validate the effectiveness and robustness of the proposed circulant preconditioner from the perspective of clustering spectrum.

In Tables 11 and 12, it verifies that the proposed fast direct solver for different discretized problems takes much less CPU time elapsed as M and N become large . Here we mention

Table 12 CPU time in seconds for Example 2 with $\beta = 1.8$ that `Time1` is done by for Algorithm 2 (MATLAB's backslash) and `Time2` (*Iters*) is done by Algorithm 2 with PCGS solver

$h = \tau$	$\alpha = 0.5$			$\alpha = 0.9$			$\alpha = 0.99$		
	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`	`Time1`	`Time2`	`Speed-up`
2^{-5}	0.03	0.06 (6.0)	0.50	0.03	0.06 (6.0)	0.50	0.03	0.06 (6.0)	0.50
2^{-6}	0.09	0.14 (7.0)	0.64	0.09	0.14 (7.0)	0.64	0.09	0.13 (7.0)	0.69
2^{-7}	0.45	0.36 (7.3)	1.28	0.45	0.35 (7.0)	1.29	0.45	0.35 (7.0)	1.29
2^{-8}	4.09	0.98 (9.0)	4.13	4.05	0.89 (7.0)	4.55	4.07	0.88 (7.0)	4.63
2^{-9}	37.41	3.89 (9.0)	9.62	37.26	3.56 (7.0)	10.47	37.39	3.54 (7.0)	8.21
2^{-10}	379.28	21.04 (9.0)	18.03	380.01	20.81 (8.3)	18.26	378.59	20.42 (7.0)	18.54

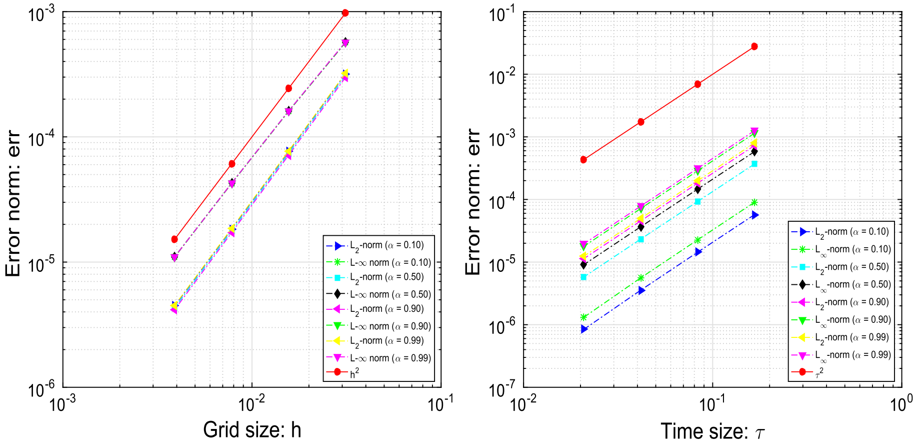

Fig. 7 Comparison the order of accuracy obtained by our proposed schemes for Example 2 in space and time variables. *Left* space direction; *Right* time direction

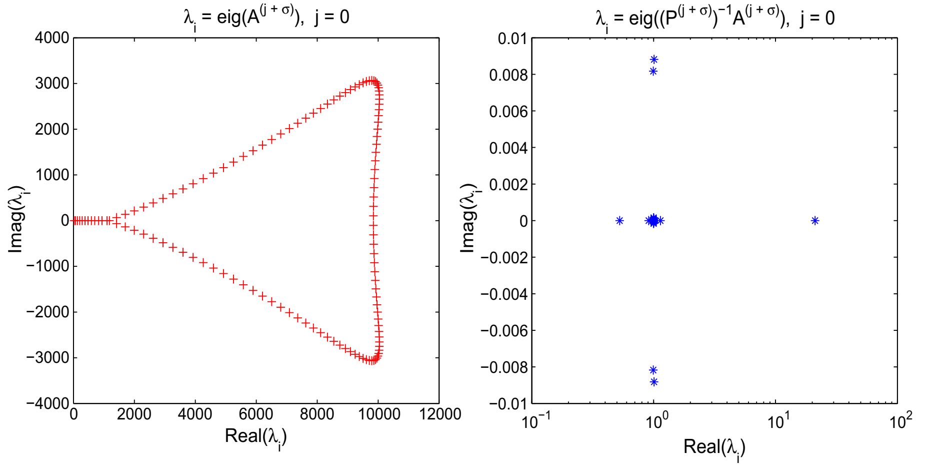

Fig. 8 Spectrum of both original and preconditioned matrices at the time level $j = 0$, respectively, when $N = M = 128$, $\alpha = 0.9$ and $\beta = 1.5$. *Left* original matrix; *Right* circulant preconditioned matrix

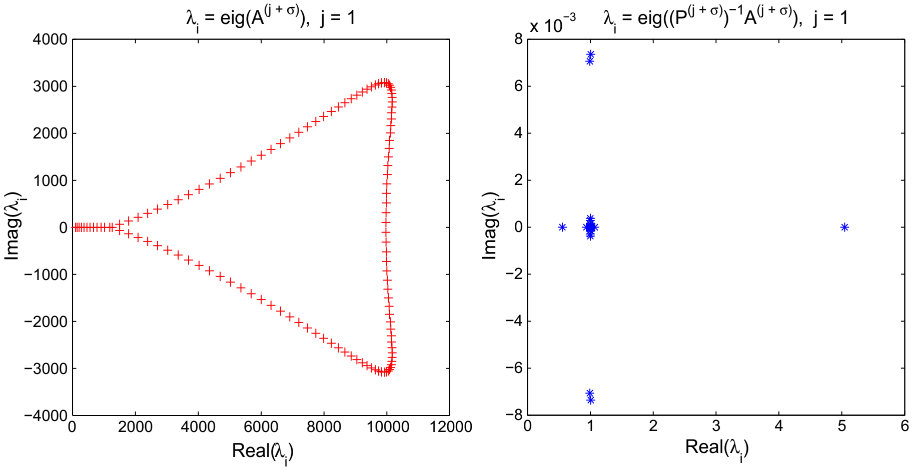

Fig. 9 Spectrum of both original and preconditioned matrices at the time level $j = 1$, respectively, when $N = M = 128$, $\alpha = 0.9$ and $\beta = 1.5$. *Left* original matrix; *Right* circulant preconditioned matrix

that "*Iters*" in brackets denotes the average number of iterations required for solving the TSFCDE problem (1.1); i.e.,

$$Iters = \frac{1}{M}\sum_{j=1}^{M} Iter(j),$$

where $Iter(j)$ denotes the number of iterations required for solving Eqs. (3.1). When $M = N = 2^{10}$ and different discretized parameters, the CPU time of Algorithm 3 by PCGS with circulant preconditioners is about 21 s, the speedup factor is more than 18. For all listed cases, we find that the average number of iterations required for solving a series of nonsymmetric Toeplitz systems is less than 10. Meanwhile, although `Time1` required by Algorithm 3 with MATLAB's backslash for small test problems ($M = N = 32, 64$) is cheaper than `Time2` needed by Algorithm 3 with using the PCGS method, our proposed method is still more attractive in aspects of the lower memory requirement. Compared to Algorithm 3 with MATLAB's backslash, it highlighted that in the whole procedure the proposed solution technique does not require to store a series of full matrices (e.g., coefficient matrices $A^{(j+\sigma)}$, $j = 0, 1, \ldots, M - 1$) at all. All in all, we can conclude that our proposed IDS with fast solution techniques is still more promising than the IDS with conventional implementations, in which direct solvers are straightforwardly employed.

5 Conclusions

In this paper, the stability and convergence of a new implicit difference scheme approximating the time-space fractional convection–diffusion equation are studied. Sufficient conditions for the unconditional stability of such difference schemes are obtained. For proving the stability of a wide class of difference schemes approximating the time fractional diffusion equation, it is simple enough to check the stability conditions obtained in this paper. Meanwhile, the new difference schemes of the second approximation order in space and the second approximation order in time for the TSFCDE with variable coefficients (in terms of t) are constructed as well. The stability and convergence of these implicit schemes in the mesh L_2-norm with the rate equal to the order of the approximation error are proved. The method can be easily adopted to other TSFCDEs with other boundary conditions (refer, e.g., to [68]). Numerical tests completely confirming the obtained theoretical results are carried out.

More significantly, we have also shown an efficient implementation of the proposed IDS based on preconditioned iterative solvers, achieving $\mathcal{O}(N \log N)$ computational complexity and $\mathcal{O}(N)$ storage cost. Extensive numerical results fully support the theoretical findings and prove the efficiency of the proposed preconditioning methods. Future work will include the extension of the proposed IDS with fast solution techniques for two and three-dimensional TSFCDEs subject to various boundary value conditions. Meanwhile, we will also focus on the development of other efficient preconditioners for accelerating the convergence of Krylov subspace solvers in this context; refer, e.g., to our recent work [69] for this topic.

Acknowledgements The authors would like to thank the Prof. Zhi-Zhong Sun and Dr. Zhao-Peng Hao for their insightful discussions about the convergence analysis of the proposed implicit difference scheme. The authors are also grateful to the anonymous referees for their useful suggestions and comments that improved the presentation of this paper. The work of Gu and Huang is supported by 973 Program (2013CB329404), NSFC (61370147 and 61402082), the Fundamental Research Funds for the Central Universities (ZYGX2014J084). Ji's work is supported by the Fundamental Research Funds for the Central Universities and the Research and Innovation Project for College Graduates of Jiangsu Province (Grant No. KYLX15_0106). The work of the last author has been partially implemented with the financial support of the Russian Presidential grant for young scientists MK-3360.2015.1.

References

1. Podlubny, I.: Fractional Differential Equations, vol. 198 of Mathematics in Science. Academic Press Inc., San Diego (1999)
2. Samko, S.G., Kilbas, A.A., Marichev, O.I.: Fractional Integrals and Derivatives: Theory and Applications. Gordon and Breach Science Publishers, Yverdonn (1993)
3. Kilbas, A.A., Srivastava, H.M., Trujillo, J.J.: Theory and Applications of Fractional Differential Equations. Elsevier, Amsterdam (2006)
4. Metzler, R., Klafter, J.: The random walk's guide to anomalous diffusion: a fractional dynamics approach. Phys. Rep. **339**, 1–77 (2000)
5. Saichev, A.I., Zaslavsky, G.M.: Fractional kinetic equations: solutions and applications. Chaos **7**, 753–764 (1997). doi:10.1063/1.166272
6. Povstenko, Y.: Space-time-fractional advection diffusion equation in a plane. In: Latawiec, K.J., Łukaniszyn, M., Stanisławski, R. (eds.) Advances in Modelling and Control of Non-Integer-Order Systems. Volume 320 of the Series Lecture Notes in Electrical Engineering, pp. 275–284. Springer, New York (2015)
7. Benson, D.A., Wheatcraft, S.W., Meerschaert, M.M.: Application of a fractional advection–dispersion equation. Water Resour. Res. **36**, 1403–1412 (2000)
8. Wang, K., Wang, H.: A fast characteristic finite difference method for fractional advection–diffusion equations. Adv. Water Resour. **34**, 810–816 (2011)
9. Dehghan, M., Manafian, J., Saadatmandi, A.: Solving nonlinear fractional partial differential equations using the homotopy analysis method. Numer. Methods Partial Differ. Equ. **26**, 448–479 (2010)
10. Povstenko, Y.Z.: Fundamental solutions to time-fractional advection diffusion equation in a case of two space variables. Math. Probl. Eng. (2014). doi:10.1155/2014/705364
11. Saadatmandi, A., Dehghan, M., Azizi, M.-R.: The Sinc–Legendre collocation method for a class of fractional convection–diffusion equations with variable coefficients. Commun. Nonlinear Sci. Numer. Simul. **17**, 4125–4136 (2012)
12. Dehghan, M., Abbaszadeh, M., Mohebbi, A.: Error estimate for the numerical solution of fractional reaction-subdiffusion process based on a meshless method. J. Comput. Appl. Math. **280**, 14–36 (2015)
13. Luo, W.-H., Huang, T.-Z., Wu, G.-C., Gu, X.-M.: Quadratic spline collocation method for the time fractional subdiffusion equation. Appl. Math. Comput. **276**, 252–265 (2016)

14. Dehghan, M., Safarpoor, M., Abbaszadeh, M.: Two high-order numerical algorithms for solving the multi-term time fractional diffusion-wave equations. J. Comput. Appl. Math. **290**, 174–195 (2015)
15. Mohebbi, A., Abbaszadeh, M., Dehghan, M.: A high-order and unconditionally stable scheme for the modified anomalous fractional sub-diffusion equation with a nonlinear source term. J. Comput. Phys. **240**, 36–48 (2013)
16. Gu, X.-M., Huang, T.-Z., Zhao, X.-L., Li, H.-B., Li, L.: Strang-type preconditioners for solving fractional diffusion equations by boundary value methods. J. Comput. Appl. Math. **277**, 73–86 (2015)
17. Meerschaert, M.M., Tadjeran, C.: Finite difference approximations for fractional advection–dispersion flow equations. J. Comput. Appl. Math. **172**, 65–77 (2004)
18. Su, L., Wang, W., Xu, Q.: Finite difference methods for fractional dispersion equations. Appl. Math. Comput. **216**, 3329–3334 (2010)
19. Su, L., Wang, W., Yang, Z.: Finite difference approximations for the fractional advection–diffusion equation. Phys. Lett. A **373**, 4405–4408 (2009)
20. Sousa, E.: Finite difference approximations for a fractional advection–diffusion problem. J. Comput. Phys. **228**, 4038–4054 (2009)
21. Su, L., Wang, W., Wang, H.: A characteristic difference method for the transient fractional convection–diffusion equations. Appl. Numer. Math. **61**, 946–960 (2011)
22. Deng, Z., Singh, V., Bengtsson, L.: Numerical solution of fractional advection–dispersion equation. J. Hydraul. Eng. **130**, 422–431 (2004)
23. Ding, Z., Xiao, A., Li, M.: Weighted finite difference methods for a class of space fractional partial differential equations with variable coefficients. J. Comput. Appl. Math. **233**, 1905–1914 (2010)
24. Liu, F., Zhuang, P., Burrage, K.: Numerical methods and analysis for a class of fractional advection–dispersion models. Comput. Math. Appl. **64**, 2990–3007 (2012)
25. Momani, S., Rqayiq, A.A., Baleanu, D.: A nonstandard finite difference scheme for two-sided space-fractional partial differential equations. Int. J. Bifurcat. Chaos **22**(1250079), 5 (2012). doi:10.1142/S0218127412500794
26. Chen, M., Deng, W.: A second-order numerical method for two-dimensional two-sided space fractional convection diffusion equation. Appl. Math. Model. **38**, 3244–3259 (2014)
27. Deng, W., Chen, M.: Efficient numerical algorithms for three-dimensional fractional partial differential equations. J. Comput. Math. **32**, 371–391 (2014)
28. Qu, W., Lei, S.-L., Vong, S.-W.: Circulant and skew-circulant splitting iteration for fractional advection–diffusion equations. Int. J. Comput. Math. **91**, 2232–2242 (2014)
29. Ford, N.J., Pal, K., Yan, Y.: An algorithm for the numerical solution of two-sided space-fractional partial differential equations. Comput. Methods Appl. Math. **15**, 497–514 (2015)
30. Bhrawy, A.H., Baleanu, D.: A spectral Legendre–Gauss–Lobatto collocation method for a space-fractional advection–diffusion equations with variable coefficients. Rep. Math. Phys. **72**, 219–233 (2013)
31. Bhrawy, A.H., Zaky, M.A.: A method based on the Jacobi tau approximation for solving multi-term time-space fractional partial differential equations. J. Comput. Phys. **281**, 876–895 (2015)
32. Hejazi, H., Moroney, T., Liu, F.: Stability and convergence of a finite volume method for the space fractional advection–dispersion equation. J. Comput. Appl. Math. **255**, 684–697 (2014)
33. Tian, W.Y., Deng, W., Wu, Y.: Polynomial spectral collocation method for space fractional advection–diffusion equation. Numer. Methods Partial Differ. Equ. **30**, 514–535 (2014)
34. Lin, Y., Xu, C.: Finite difference/spectral approximations for the time-fractional diffusion equation. J. Comput. Phys. **225**, 1533–1552 (2007)
35. Zhang, H., Liu, F., Phanikumar, M.S., Meerschaert, M.M.: A novel numerical method for the time variable fractional order mobile–immobile advection–dispersion model. Comput. Math. Appl. **66**, 693–701 (2013)
36. Cui, M.: A high-order compact exponential scheme for the fractional convection–diffusion equation. J. Comput. Appl. Math. **255**, 404–416 (2014)
37. Cui, M.: Compact exponential scheme for the time fractional convection–diffusion reaction equation with variable coefficients. J. Comput. Phys. **280**, 143–163 (2015)
38. Mohebbi, A., Abbaszadeh, M.: Compact finite difference scheme for the solution of time fractional advection–dispersion equation. Numer. Algorithms **63**, 431–452 (2013)
39. Momani, S.: An algorithm for solving the fractional convection–diffusion equation with nonlinear source term. Commun. Nonlinear Sci. Numer. Simul. **12**, 1283–1290 (2007)
40. Wang, Z., Vong, S.: A high-order exponential ADI scheme for two dimensional time fractional convection–diffusion equations. Comput. Math. Appl. **68**, 185–196 (2014)
41. Fu, Z.-J., Chen, W., Yang, H.-T.: Boundary particle method for Laplace transformed time fractional diffusion equations. J. Comput. Phys. **235**, 52–66 (2013)
42. Zhai, S., Feng, X., He, Y.: An unconditionally stable compact ADI method for three-dimensional time-fractional convection–diffusion equation. J. Comput. Phys. **269**, 138–155 (2014)

43. Zhuang, P., Gu, Y.T., Liu, F., Turner, I., Yarlagadda, P.K.D.V.: Time-dependent fractional advection–diffusion equations by an implicit MLS meshless method. Int. J. Numer. Methods Eng. **88**, 1346–1362 (2011)
44. Wang, Y.-M.: A compact finite difference method for solving a class of time fractional convection–subdiffusion equations. BIT Numer. Math. **55**, 1187–1217 (2015)
45. Zhang, Y.: A finite difference method for fractional partial differential equation. Appl. Math. Comput. **215**, 524–529 (2009)
46. Zhang, Y.: Finite difference approximations for space-time fractional partial differential equation. J. Numer. Math. **17**, 319–326 (2009)
47. Shao, Y., Ma, W.: Finite difference approximations for the two-side space-time fractional advection–diffusion equations. J. Comput. Anal. Appl. **21**, 369–379 (2016)
48. Qin, P., Zhang, X.: A numerical method for the space-time fractional convection–diffusion equation. Math. Numer. Sin. **30**, 305–310 (2008). (in Chinese)
49. Liu, F., Zhuang, P., Anh, V., Turner, I., Burrage, K.: Stability and convergence of the difference methods for the space-time fractional advection–diffusion equation. Appl. Math. Comput. **191**, 12–20 (2007)
50. Zhao, Z., Jin, X.-Q., Lin, M.M.: Preconditioned iterative methods for space-time fractional advection–diffusion equations. J. Comput. Phys. **319**, 266–279 (2016)
51. Shen, S., Liu, F., Anh, V.: Numerical approximations and solution techniques for the space-time Riesz–Caputo fractional advection–diffusion equation. Numer. Algorithms **56**, 383–403 (2011)
52. Parvizi, M., Eslahchi, M.R., Dehghan, M.: Numerical solution of fractional advection–diffusion equation with a nonlinear source term. Numer. Algorithms **68**, 601–629 (2015)
53. Chen, Y., Wu, Y., Cui, Y., Wang, Z., Jin, D.: Wavelet method for a class of fractional convection–diffusion equation with variable coefficients. J. Comput. Sci. **1**, 146–149 (2010)
54. Irandoust-pakchin, S., Dehghan, M., Abdi-mazraeh, S., Lakestani, M.: Numerical solution for a class of fractional convection–diffusion equations using the flatlet oblique multiwavelets. J. Vib. Control **20**, 913–924 (2014)
55. Bhrawy, A.H., Zaky, M.A., Tenreiro-Machado, M.A.: Efficient Legendre spectral tau algorithm for solving the two-sided space-time Caputo fractional advection–dispersion equation. J. Vib. Control **22**, 2053–2068 (2016)
56. Hejazi, H., Moroney, T., Liu, F.: A finite volume method for solving the two-sided time-space fractional advection–dispersion equation. Cent. Eur. J. Phys. **11**, 1275–1283 (2013)
57. Jiang, W., Lin, Y.: Approximate solution of the fractional advection–dispersion equation. Comput. Phys. Commun. **181**, 557–561 (2010)
58. Wei, J., Chen, Y., Li, B., Yi, M.: Numerical solution of space-time fractional convection–diffusion equations with variable coefficients using Haar wavelets. Comput. Model. Eng. Sci. (CMES) **89**, 481–495 (2012)
59. Ng, M.: Iterative Methods for Toeplitz Systems. Oxford University Press, New York (2004)
60. Lin, F.-R., Yang, S.-W., Jin, X.-Q.: Preconditioned iterative methods for fractional diffusion equation. J. Comput. Phys. **256**, 109–117 (2014)
61. Lei, S.-L., Sun, H.-W.: A circulant preconditioner for fractional diffusion equations. J. Comput. Phys. **242**, 715–725 (2013)
62. Gohberg, I., Semencul, A.: On the inversion of finite Toeplitz matrices and their continuous analogues. Matem. Issled. **7**, 201–223 (1972). (in Russian)
63. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
64. Alikhanov, A.A.: A new difference scheme for the time fractional diffusion equation. J. Comput. Phys. **280**, 424–438 (2015)
65. Hao, Z.-P., Sun, Z.-Z., Cao, W.-R.: A fourth-order approximation of fractional derivatives with its applications. J. Comput. Phys. **281**, 787–805 (2015)
66. Vong, S., Lyu, P., Chen, X., Lei, S.-L.: High order finite difference method for time-space fractional differential equations with Caputo and Riemann–Liouville derivatives. Numer. Algorithms **72**, 195–210 (2016)
67. Björck, Å.: Numerical Methods in Matrix Computations, volume 59 of the series Texts in Applied Mathematics. Springer, Switzerland (2014)
68. Jia, J., Wang, H.: Fast finite difference methods for space-fractional diffusion equations with fractional derivative boundary conditions. J. Comput. Phys. **293**, 359–369 (2015)
69. Gu, X.-M., Huang, T.-Z., Li, H.-B., Li, L., Luo, W.-H.: On k-step CSCS-based polynomial preconditioners for Toeplitz linear systems with application to fractional diffusion equations. Appl. Math. Lett. **42**, 53–58 (2015)