Received: 19 July 2018 | Revised: 9 February 2019 | Accepted: 22 October 2019
DOI: 10.1002/nla.2274

RESEARCH ARTICLE

WILEY

On regularized Hermitian splitting iteration methods for solving discretized almost-isotropic spatial fractional diffusion equations

Zhong-Zhi Bai[1,2] | **Kang-Ya Lu**[3]

[1]State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

[2]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China

[3]School of Applied Science, Beijing Information Science and Technology University, Beijing, China

Correspondence
Zhong-Zhi Bai, State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, PO Box 2719, Beijing 100190, China.
Email: bzz@lsec.cc.ac.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 11671393 and 11911530082

Summary

The shifted finite-difference discretization of the one-dimensional almost-isotropic spatial fractional diffusion equation results in a discrete linear system whose coefficient matrix is a sum of two diagonal-times-Toeplitz matrices. For this kind of linear systems, we propose a class of regularized Hermitian splitting iteration methods and prove its asymptotic convergence under mild conditions. For appropriate circulant-based approximation to the corresponding regularized Hermitian splitting preconditioner, we demonstrate that the induced fast regularized Hermitian splitting preconditioner possesses a favorable preconditioning property. Numerical results show that, when used to precondition Krylov subspace iteration methods such as generalized minimal residual and biconjugate gradient stabilized methods, the fast preconditioner significantly outperforms several existing ones.

KEYWORDS

convergence property, matrix splitting iteration, preconditioning, spatial fractional diffusion equation

1 | INTRODUCTION

We consider numerical solutions of one-dimensional almost-isotropic spatial fractional diffusion equations of the form

$$\begin{cases} -\omega(x)\frac{d^\beta u(x)}{d_+x^\beta} - \gamma(x)\frac{d^\beta u(x)}{d_-x^\beta} = f(x), \quad x \in (0,1), \\ u(0) = u(1) = 0, \end{cases} \tag{1}$$

where $\omega(x)$ and $\gamma(x)$ are nonnegative and bounded diffusion coefficients satisfying $\omega(x) \approx \gamma(x)$ for $\forall x \in (0, 1)$, and $\frac{d^\beta u(x)}{d_+x^\beta}$ and $\frac{d^\beta u(x)}{d_-x^\beta}$ are, respectively, the left and right Riemann–Liouville fractional derivatives,[1,2] with the order being $\beta \in (1, 2)$.

By discretizing the spatial fractional derivatives with the shifted finite-difference formulas of the Grünwald–Letnikov type[3,4] on the spatial grid $\{x_i = ih \mid i = 0, 1, \ldots, n+1\}$ with the equidistant step size $h = \frac{1}{n+1}$, we can obtain the discrete linear system corresponding to the spatial fractional diffusion equation in Equation (1) as follows:

$$Au^h := (\Omega T + \Gamma T^*)u^h = h^\beta f^h := b. \tag{2}$$

Here

$$u^h = \left[u_1^h, u_2^h, \ldots, u_n^h\right]^* \quad \text{and} \quad f^h = \left[f_1^h, f_2^h, \ldots, f_n^h\right]^*$$

are n-dimensional vectors, with $u_i^h \approx u(x_i)$, $f_i^h = f(x_i)$, and $(\cdot)^*$ indicating the transpose in the real vector or matrix space, say, $\mathbb{R}^n$ or $\mathbb{R}^{n\times n}$;

$$\Omega = \mathrm{diag}(\omega_1, \omega_2, \ldots, \omega_n) \quad \text{and} \quad \Gamma = \mathrm{diag}(\gamma_1, \gamma_2, \ldots, \gamma_n)$$

are diagonal matrices of nonnegative diagonal entries, with $\omega_i = \omega(x_i)$ and $\gamma_i = \gamma(x_i)$; and

$$T = -\begin{pmatrix} g_1^{(\beta)} & g_0^{(\beta)} & 0 & \cdots & 0 & 0 \\ g_2^{(\beta)} & g_1^{(\beta)} & g_0^{(\beta)} & \ddots & \ddots & 0 \\ \vdots & g_2^{(\beta)} & g_1^{(\beta)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{n-1}^{(\beta)} & \ddots & \ddots & \ddots & g_1^{(\beta)} & g_0^{(\beta)} \\ g_n^{(\beta)} & g_{n-1}^{(\beta)} & \cdots & \cdots & g_2^{(\beta)} & g_1^{(\beta)} \end{pmatrix} \tag{3}$$

is a Toeplitz matrix of lower Hessenberg form, with $g_j^{(\beta)} = (-1)^j \binom{\beta}{j}$, $j = 0, 1, 2, \ldots$, satisfying

$$\begin{cases} g_0^{(\beta)} = 1, \quad g_1^{(\beta)} = -\beta, \quad 1 \geq g_2^{(\beta)} \geq g_3^{(\beta)} \geq \cdots \geq 0, \\ \sum_{j=0}^{\infty} g_j^{(\beta)} = 0, \quad \sum_{j=0}^{s} g_j^{(\beta)} < 0, \quad \forall s \geq 1, \end{cases}$$

where $\binom{\beta}{j}$ is the fractional binomial coefficients.

Note that the Toeplitz matrix T and, hence, the coefficient matrix A of the discrete linear system in Equation (2) are strictly diagonally dominant M-matrices.[5,6] Thereby, both T and A are monotone matrices,* that is, $T^{-1} \geq 0$ and $A^{-1} \geq 0$. In addition, according to Lemma 2 in the work of Bai,[7] the condition number of matrix A with respect to the infinity norm, denoted as $\kappa_\infty(A)$, is bounded as $\kappa_\infty(A) \leq \mathcal{O}(h^{-\beta})$. However, as matrix A is full and dense, the direct method based on the LU factorization[8] for solving the discrete linear system in Equation (2) has the computational complexity $\mathcal{O}(n^3)$ and storage complexity $\mathcal{O}(n^2)$, which are practically prohibitive, especially when step size h is significantly small (see, e.g., the works of Bai and Lu,[9] Meerschaert et al.,[10] and Meerschaert and Tadjeran[11] for more details).

When it is assumed that one of the diffusion coefficients $\omega(x)$ and $\gamma(x)$ of the fractional diffusion equation in Equation (1) is sufficiently larger than the other, Bai[7] proposed a class of *Hermitian and skew-Hermitian splitting*† iteration methods based on respective scaling, and established the asymptotic convergence theory for this class of *respectively scaled Hermitian and skew-Hermitian splitting* (RSHSS) iteration methods under certain conditions imposed on the involved iteration parameter. If the matrix splitting preconditioner induced by the RSHSS iteration method is further approximated by replacing the involved Toeplitz matrices with certain circulant matrices, the resulting *fast RSHSS* (FRSHSS) preconditioner can be executed with a computational complexity of $\mathcal{O}(n\log(n))$ by fast Fourier transforms. It was demonstrated in the work of Bai[7] that the FRSHSS preconditioner possesses a favorable property in improving the convergence properties of the Krylov subspace iteration methods such as *generalized minimal residual* (GMRES) and *biconjugate gradient stabilized* (BiCGSTAB) methods, and the FRSHSS-preconditioned GMRES can compute an approximate solution for the target discrete linear system accurately and stably, even if the diffusion coefficients $\omega(x)$ and $\gamma(x)$ of the spatial fractional diffusion equation in Equation (1) are nonsmooth, are drastically oscillating, or have big jumps. However, as pointed out in the work of Bai,[7] if the diffusion coefficients $\omega(x)$ and $\gamma(x)$ are about equal or, in other words, the fractional diffusion equation in Equation (1) is almost isotropic, then the RSHSS iteration method and the corresponding FRSHSS preconditioner are infeasible when being used to solve the discrete linear system in Equation (2).

Motivated by the idea in the work of Bai[7] and focusing particularly on the situation that the diffusion coefficients $\omega(x)$ and $\gamma(x)$ are almost equal, in this paper we propose a class of *regularized Hermitian splitting* (RHS) iteration methods for solving the discrete linear system in Equation (2). Under mild conditions, we prove the asymptotic convergence property

*Here and in the sequel, a real $m \times n$ matrix $B = (b_{ij}) \in \mathbb{R}^{m\times n}$ is said to be nonnegative, that is, $B \geq 0$, if all of its entries satisfy $b_{ij} \geq 0$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.

†In the real matrix space, the Hermitian and skew-Hermitian parts of a matrix are simply called the symmetric and skew-symmetric parts, respectively.

of this iterative method, derive a tight upper bound on the contraction factor in the Euclidean norm, and compute the optimal iteration parameter that minimizes the upper bound of the corresponding contraction factor. With a circulant-based approximation to the Toeplitz matrix involved in the corresponding RHS preconditioner, we obtain a *fast RHS* (FRHS) preconditioner for the discrete linear system in Equation (2). We demonstrate that the FRHS-preconditioned matrix is approximated well by the RHS-preconditioned matrix up to a low-rank matrix plus a small-norm matrix. Numerical results show that, when used to precondition Krylov subspace methods such as GMRES and BiCGSTAB, the FRHS preconditioner significantly outperforms several existing ones, so that the FRHS-preconditioned GMRES and BiCGSTAB methods can compute an approximate solution for the target discrete linear system accurately and stably, even if the fractional diffusion equation in Equation (1) is almost isotropic, with its diffusion coefficients $\omega(x)$ and $\gamma(x)$ being nonsmooth, drastically oscillating, and of big jumps.

This paper is organized as follows. In Section 2, we describe the RHS iteration method and prove its asymptotic convergence. In Section 3, we present a fast variant of the RHS preconditioning matrix and some properties about the spectrum and the rank of the corresponding preconditioned matrix. Section 4 shows the numerical results. Finally, in Section 5, we end this paper with a few remarks and conclusions.

2 | THE RHS ITERATION METHOD

In this section, we construct the RHS iteration method for solving the discrete linear system in Equation (2) and establish its asymptotic convergence theory.

Let

$$H = \frac{1}{2}(T + T^*) \quad \text{and} \quad S = \frac{1}{2}(T - T^*) \tag{4}$$

be the symmetric and skew-symmetric parts of the Toeplitz matrix T in Equation (3). Then

$$T = H + S \quad \text{and} \quad T^* = H - S,$$

and the coefficient matrix A of the discrete linear system in Equation (2) can be equivalently reformulated as

$$A = (\Omega + \Gamma)H + (\Omega - \Gamma)S.$$

For any positive parameter α, as

$$H = \frac{1}{2}(\alpha I + H) - \frac{1}{2}(\alpha I - H), \tag{5}$$

matrix A admits the so-called RHS

$$A = \frac{1}{2}(\Omega + \Gamma)(\alpha I + H) - \left[\frac{1}{2}(\Omega + \Gamma)(\alpha I - H) - (\Omega - \Gamma)S\right] := M(\alpha) - N(\alpha), \tag{6}$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix, and

$$\begin{cases} M(\alpha) = \frac{1}{2}(\Omega + \Gamma)(\alpha I + H), \\ N(\alpha) = \frac{1}{2}(\Omega + \Gamma)(\alpha I - H) - (\Omega - \Gamma)S. \end{cases}$$

We remark that the splitting in Equation (5) was called quasi-symmetric splitting in the work of Axelsson et al.[12] (see also the work of Golub and Overton[13]) and shift splitting in the work of Bai et al.[14]

With the RHS defined in Equation (6), we can construct the following RHS iteration method for solving the discrete linear system in Equation (2).

The RHS iteration method. Given an initial guess $u_0^h \in \mathbb{R}^n$, for $k = 0, 1, 2, \ldots$ until the iteration sequence $\{u_k^h\} \subset \mathbb{R}^n$ converges, compute the next iterate $u_{k+1}^h \in \mathbb{R}^n$ according to the following procedure:

$$(\alpha I + H)u_{k+1}^h = \left[(\alpha I - H) - 2(\Omega + \Gamma)^{-1}(\Omega - \Gamma)S\right] u_k^h + 2(\Omega + \Gamma)^{-1}b, \tag{7}$$

where α is a prescribed positive constant.

The iteration scheme in Equation (7) can be equivalently rewritten as

$$u_{k+1}^h = L(\alpha)u_k^h + G(\alpha)b, \quad k = 0, 1, 2, \dots,$$

where

$$L(\alpha) = M(\alpha)^{-1}N(\alpha) = (\alpha I + H)^{-1}(\alpha I - H) - 2(\alpha I + H)^{-1}(\Omega + \Gamma)^{-1}(\Omega - \Gamma)S \tag{8}$$

is the corresponding iteration matrix, and

$$G(\alpha) = M(\alpha)^{-1} = 2(\alpha I + H)^{-1}(\Omega + \Gamma)^{-1}.$$

The RHS $A = M(\alpha) - N(\alpha)$ naturally leads to an RHS preconditioner $M(\alpha)$ for the coefficient matrix A of the discrete linear system in Equation (2). As a result, incorporated with this RHS preconditioning matrix, possibly approximated further by a certain replacement strategy using an appropriate circulant matrix, the Krylov subspace iteration methods[15] could be fast and effective solvers to compute an approximate solution for the discrete linear system in Equation (2).

For the asymptotic convergence property of the RHS iteration method, we have the following result. Here and in the sequel, we use $\|\cdot\|_2$ to denote the Euclidean norm.

Theorem 1. *Assume that the diffusion coefficients $\omega(x)$ and $\gamma(x)$ of the spatial fractional diffusion equation in Equation (1) are uniformly positive in the interval $(0, 1)$. For the Toeplitz matrix $T \in \mathbb{R}^{n\times n}$ defined in Equation (3), let the symmetric and skew-symmetric Toeplitz matrices $H \in \mathbb{R}^{n\times n}$ and $S \in \mathbb{R}^{n\times n}$ be defined in Equation (4), with the eigenvalues of H being bounded from below and above by $\eta_{\min}$ and $\eta_{\max}$, respectively. Define*

$$\phi = \max_{1\le i\le n}\left\{\frac{|\omega_i - \gamma_i|}{\omega_i + \gamma_i}\right\},$$

and assume $\phi\|S\|_2 < \eta_{\min}$. Then, the iteration sequence $\{u_k^h\}_{k=0}^{\infty}$, yielded by the RHS iteration method starting from any initial vector $u_0^h \in \mathbb{R}^n$, converges to the unique solution of the discrete linear system in Equation (2), provided the iteration parameter α satisfies $\alpha > \alpha_+$ with

$$\alpha_+ = \frac{\eta_{\min} - \phi\|S\|_2}{2}\left(\sqrt{1 + \frac{4\phi\eta_{\max}\|S\|_2}{(\eta_{\min} - \phi\|S\|_2)^2}} - 1\right).$$

Proof. Let $\Theta = (\Omega + \Gamma)^{-1}(\Omega - \Gamma)$. Then we know that $\|\Theta\|_2 \le \phi$. It follows from the definition in Equation (8) of the RHS iteration matrix $L(\alpha)$ that

$$\begin{aligned}
\|L(\alpha)\|_2 &\le \|(\alpha I + H)^{-1}(\alpha I - H)\|_2 + 2\|(\Omega + \Gamma)^{-1}(\Omega - \Gamma)\|_2\|(\alpha I + H)^{-1}\|_2\|S\|_2 \\
&\le \max_{\lambda\in \mathrm{sp}(H)}\frac{|\alpha - \lambda|}{\alpha + \lambda} + 2\phi\|S\|_2 \max_{\lambda\in \mathrm{sp}(H)}\frac{1}{\alpha + \lambda} \\
&\le \max_{\eta_{\min}\le\lambda\le\eta_{\max}}\frac{|\alpha - \lambda|}{\alpha + \lambda} + \frac{2\phi\|S\|_2}{\alpha + \eta_{\min}} \\
&:= \varsigma(\alpha),
\end{aligned}$$

where $\mathrm{sp}(\cdot)$ indicates the set of the eigenvalues of the corresponding square matrix. Because

$$\begin{aligned}
\varsigma(\alpha) &= \max\left\{\frac{|\alpha - \eta_{\min}|}{\alpha + \eta_{\min}}, \quad \frac{|\alpha - \eta_{\max}|}{\alpha + \eta_{\max}}\right\} + \frac{2\phi\|S\|_2}{\alpha + \eta_{\min}} \\
&= \begin{cases} \frac{\eta_{\max}-\alpha}{\eta_{\max}+\alpha} + \frac{2\phi\|S\|_2}{\eta_{\min}+\alpha}, & \text{for} \quad \alpha \le \sqrt{\eta_{\min}\eta_{\max}}, \\ \frac{\alpha+2\phi\|S\|_2-\eta_{\min}}{\alpha+\eta_{\min}}, & \text{for} \quad \alpha > \sqrt{\eta_{\min}\eta_{\max}}, \end{cases}
\end{aligned}$$

after straightforward computations we can demonstrate that $\varsigma(\alpha) < 1$ is implied by the condition $\alpha > \alpha_+$. Hence, for the spectral radius $\rho(L(\alpha))$ of the RHS iteration matrix $L(\alpha)$, under the condition $\alpha > \alpha_+$ it holds that

$$\rho(L(\alpha)) \leq \|L(\alpha)\|_2 < 1,$$

which, in accordance with the work of Varga,[6] shows that the RHS iteration method is asymptotically convergent. □

From the proof of Theorem 1, we see that the contraction factor of the RHS iteration method in the Euclidean norm is bounded by

$$\varsigma(\alpha) = \begin{cases} \frac{\eta_{\max}-\alpha}{\eta_{\max}+\alpha} + \frac{2\phi\|S\|_2}{\eta_{\min}+\alpha}, & \text{for} \quad \alpha_+ < \alpha \leq \alpha_o, \\ \frac{\alpha+2\phi\|S\|_2-\eta_{\min}}{\alpha+\eta_{\min}}, & \text{for} \quad \alpha > \alpha_o, \end{cases}$$

with $\alpha_o = \sqrt{\eta_{\min}\eta_{\max}}$. It then follows from the monotonicity of the function $\varsigma(\alpha)$ that its minimum is attained at the point $\alpha = \alpha_o$. Moreover, it holds that

$$\varsigma(\alpha_o) = \frac{\sqrt{\kappa(H)} - 1}{\sqrt{\kappa(H)} + 1} + \frac{2\phi\|S\|_2}{\eta_{\min}\left(\sqrt{\kappa(H)} + 1\right)},$$

where $\kappa(H) = \frac{\eta_{\max}}{\eta_{\min}}$ represents an upper bound of the Euclidean condition number of matrix H.

The above arguments show that when ϕ is sufficiently small or, in other words, the fractional diffusion equation in Equation (1) is almost isotropic, with the use of the optimal iteration parameter α_o the convergence rate of the RHS iteration method is about equal to $\frac{\sqrt{\kappa(H)}-1}{\sqrt{\kappa(H)}+1}$, which is the same as that of the conjugate gradient method applied to solve a linear system of the coefficient matrix H.[8]

In accordance with Theorem 1, we can provide a more precise description about the implication of the term "almost isotropic" with respect to the spatial fractional diffusion equation (Equation (1)) in its discretized form (Equation (2)). Intuitively, we say that this discrete linear system is almost isotropic, if at least the RHS iteration method is guaranteed to be convergent without imposing any further condition, that is, if $\phi\|S\|_2 < \eta_{\min}$ or, equivalently, in the matrix form, $\phi < \frac{1}{\|H^{-1}\|_2\|S\|_2}$. In turn, if the discrete linear system in Equation (2) is almost isotropic, then we say that the spatial fractional diffusion equation in Equation (1) is almost isotropic. Admittedly, such a classification criterion is pessimistically too strict to be used in actual applications, as there are frequently many examples that the RHS iteration method is convergent even if the abovementioned criterion is violated.

Note that when $n \geq 3$, from Lemma 1 in the work of Bai,[7] we have

$$\eta_{\min} = æh^{\beta}, \quad \eta_{\max} = 2\beta \quad \text{and} \quad \|S\|_2 \leq \beta,$$

where

$$æ = \frac{e^{\beta+1}(4-\beta)^{9/2}}{e^{13/12}5^{9/2}2^{\beta}\beta\Gamma(-\beta)}, \tag{9}$$

with $\Gamma(\cdot)$ being the Gamma function. Hence, it holds that

$$\alpha_o = \sqrt{2æ\beta}\, h^{\beta/2} \quad \text{and} \quad \varsigma(\alpha_o) = \frac{\sqrt{2\beta} - \sqrt{æ}h^{\beta/2}}{\sqrt{2\beta} + \sqrt{æ}h^{\beta/2}} + \frac{2\beta\phi}{\sqrt{æ}h^{\beta/2}(\sqrt{2\beta} + \sqrt{æ}h^{\beta/2})}.$$

As a result, the discrete linear system in Equation (2) and the corresponding spatial fractional diffusion equation in Equation (1) are said to be almost isotropic if $\phi < \frac{æ}{\beta}h^{\beta} = \mathcal{O}(h^{\beta})$. Once again, we emphasize that this criterion is of theoretical meaning only rather than of practical one.

3 | THE FRHS PRECONDITIONING

The action of the RHS preconditioner $M(\alpha)$ can be accomplished in $\mathcal{O}(n\log^2(n))$ complexity by employing the superfast direct Toeplitz solvers developed in other works.[16–19] However, as pointed out in the work of Bai,[7] this approach has its own limitations and stability issues. In turn, we may replace the Toeplitz matrix H in $M(\alpha)$ by a circulant matrix, for example, H_C, obtaining the so-called FRHS preconditioner

$$M_C(\alpha) = \frac{1}{2}(\Omega + \Gamma)(\alpha I + H_C)$$

for the coefficient matrix A of the discrete linear system in Equation (2). Evidently, the FRHS preconditioner $M_C(\alpha)$ can be executed more cheaply in $\mathcal{O}(n\log(n))$ complexity and less restrictively than the RHS preconditioner by making use of fast Fourier transform.

In actual applications, from the works of Bai et al.[20] and Strang,[21] we know that the best possible choice for the circulant matrix H_C is Strang's circulant approximation given by

$$H_C = -\frac{1}{2}\begin{pmatrix} 2g_1^{(\beta)} & g_0^{(\beta)} + g_2^{(\beta)} & \cdots & g_{(n+1)/2}^{(\beta)} & g_{(n+1)/2}^{(\beta)} & \cdots & g_0^{(\beta)} + g_2^{(\beta)} \\ g_2^{(\beta)} + g_0^{(\beta)} & 2g_1^{(\beta)} & \ddots & \ddots & g_{(n+1)/2}^{(\beta)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & g_{(n+1)/2}^{(\beta)} \\ g_{(n+1)/2}^{(\beta)} & \ddots & \ddots & \ddots & \ddots & \ddots & g_{(n+1)/2}^{(\beta)} \\ g_{(n+1)/2}^{(\beta)} & g_{(n+1)/2}^{(\beta)} & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & g_0^{(\beta)} + g_2^{(\beta)} \\ g_0^{(\beta)} + g_2^{(\beta)} & \cdots & g_{(n+1)/2}^{(\beta)} & g_{(n+1)/2}^{(\beta)} & \cdots & g_2^{(\beta)} + g_0^{(\beta)} & 2g_1^{(\beta)} \end{pmatrix} \tag{10}$$

when n is odd, and by

$$H_C = -\frac{1}{2}\begin{pmatrix} 2g_1^{(\beta)} & g_0^{(\beta)} + g_2^{(\beta)} & \cdots & g_{n/2}^{(\beta)} & 0 & g_{n/2}^{(\beta)} & \cdots & g_0^{(\beta)} + g_2^{(\beta)} \\ g_2^{(\beta)} + g_0^{(\beta)} & 2g_1^{(\beta)} & \ddots & \ddots & g_{n/2}^{(\beta)} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & g_{n/2}^{(\beta)} \\ g_{n/2}^{(\beta)} & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & g_{n/2}^{(\beta)} \\ g_{n/2}^{(\beta)} & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & g_0^{(\beta)} + g_2^{(\beta)} \\ g_0^{(\beta)} + g_2^{(\beta)} & \cdots & g_{n/2}^{(\beta)} & 0 & g_{n/2}^{(\beta)} & \cdots & g_2^{(\beta)} + g_0^{(\beta)} & 2g_1^{(\beta)} \end{pmatrix} \tag{11}$$

when n is even.

We now analyze the favorable property of the FRHS-preconditioned matrix $M_C(\alpha)^{-1}A$. To this end, we need to utilize the positive constants æ in Equation (9), ϕ in Theorem 1, and

$$\theta = \frac{3^{\beta+1}}{\beta(3-\beta)^{\beta+1}\Gamma(-\beta)}.$$

Also, by rank$(\cdot)$ and $\|\cdot\|_\infty$, we indicate the rank and the infinity norm of a matrix; and with $\lfloor\cdot\rfloor$, we represent the floor of the corresponding real number, which is the largest integer not larger than that real number.

Theorem 2. *Suppose that the conditions in Theorem 1 are satisfied, and the diffusion coefficients $\omega(x)$ and $\gamma(x)$ of the spatial fractional diffusion equation in Equation (1) are uniformly bounded as $\nu_{\min} \le \omega(x), \gamma(x) \le \nu_{\max}$, $\forall x \in (0,1)$, with both $\nu_{\min}$ and $\nu_{\max}$ being positive constants. For the Toeplitz matrix $T \in \mathbb{R}^{n\times n}$ defined in Equation (3), let the symmetric and skew-symmetric Toeplitz matrices $H \in \mathbb{R}^{n\times n}$ and $S \in \mathbb{R}^{n\times n}$ be defined in Equation (4), and the circulant matrix*

$H_C \in \mathbb{R}^{n\times n}$ be defined in Equation (10) or in Equation (11). Assume that $n \geq 6$ and ϵ is a positive constant such that $\epsilon \geq \theta\left(\frac{2}{n}\right)^{\beta}$. Denote by $k_o = \left\lfloor \left(\frac{\theta}{\epsilon}\right)^{1/\beta} \right\rfloor$. Then, the following statements hold:

(i) there exist two matrices $E_{\mathrm{C}}(\alpha) \in \mathbb{R}^{n\times n}$ and $F_{\mathrm{C}}(\alpha) \in \mathbb{R}^{n\times n}$ such that

$$M(\alpha) - M_{\mathrm{C}}(\alpha) = E_{\mathrm{C}}(\alpha) + F_{\mathrm{C}}(\alpha),$$

with $E_{\mathrm{C}}(\alpha)$ and $F_{\mathrm{C}}(\alpha)$ satisfying $\mathrm{rank}(E_{\mathrm{C}}(\alpha)) \leq 2k_o$ and

$$\|F_{\mathrm{C}}(\alpha)\|_2 \leq \frac{1}{2}\nu_{\max}\epsilon;$$

(ii) there exist two matrices $E(\alpha) \in \mathbb{R}^{n\times n}$ and $F(\alpha) \in \mathbb{R}^{n\times n}$ such that

$$M_{\mathrm{C}}(\alpha)^{-1}A = M(\alpha)^{-1}A + E(\alpha) + F(\alpha),$$

with $E(\alpha)$ and $F(\alpha)$ satisfying $\mathrm{rank}(E(\alpha)) \leq 2k_o$ and $\|F(\alpha)\|_2 \leq \frac{\epsilon}{\alpha + 2^{\beta} æ h^{\beta}}$.

Proof. We only prove the case that n is even, for instance. Since

$$M(\alpha) - M_{\mathrm{C}}(\alpha) = \frac{1}{2}(\Omega + \Gamma)(H - H_C)$$

and

$$H - H_C = -\frac{1}{2}\begin{pmatrix} 0 & \hat{E}_{12} & \hat{E}_{13} \\ \hat{E}_{12}^* & 0 & 0 \\ \hat{E}_{13}^* & 0 & 0 \end{pmatrix},$$

where

$$\hat{E}_{12} = \begin{pmatrix} g_{n/2+1}^{(\beta)} & g_{n/2+2}^{(\beta)} - g_{n/2}^{(\beta)} & \cdots & \cdots & \cdots & g_{n-k_o}^{(\beta)} - g_{k_o+2}^{(\beta)} \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & g_{n/2+1}^{(\beta)} \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{n/2\times(n/2-k_o)}$$

and

$$\hat{E}_{13} = \begin{pmatrix} g_{n-k_o+1}^{(\beta)} - g_{k_o+1}^{(\beta)} & \cdots & \cdots & \cdots & \cdots & g_n^{(\beta)} - \left(g_0^{(\beta)} + g_2^{(\beta)}\right) \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ g_{n/2+1}^{(\beta)} & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & g_{n-k_o+1}^{(\beta)} - g_{k_o+1}^{(\beta)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & g_{n/2+1}^{(\beta)} \end{pmatrix} \in \mathbb{R}^{n/2\times k_o},$$

with the notation

$$\hat{E}_a = -\frac{1}{2}\begin{pmatrix} 0 & \hat{E}_{12} & 0 \\ \hat{E}_{12}^* & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \hat{E}_b = -\frac{1}{2}\begin{pmatrix} 0 & 0 & \hat{E}_{13} \\ 0 & 0 & 0 \\ \hat{E}_{13}^* & 0 & 0 \end{pmatrix},$$

we see that

$$H - H_C = \hat{E}_a + \hat{E}_b$$

and

$$M(\alpha) - M_{\mathrm{C}}(\alpha) = \frac{1}{2}(\Omega + \Gamma)(\hat{E}_a + \hat{E}_b) = E_{\mathrm{C}}(\alpha) + F_{\mathrm{C}}(\alpha),$$

where

$$E_{\mathrm{C}}(\alpha) = \frac{1}{2}(\Omega + \Gamma)\hat{E}_b \quad \text{and} \quad F_{\mathrm{C}}(\alpha) = \frac{1}{2}(\Omega + \Gamma)\hat{E}_a.$$

It then follows immediately from $\mathrm{rank}(\hat{E}_b) = 2k_o$ that $\mathrm{rank}(E_{\mathrm{C}}(\alpha)) \leq 2k_o$.

In addition, according to Geršgorin's circle theorem,[6,22] from Lemma 4.1 in the work of Bai et al.[20] we have

$$
\begin{aligned}
\left\|\widehat{E}_a\right\|_2 &\leq \left\|\widehat{E}_a\right\|_\infty = \frac{1}{2}\max\left\{\left\|\widehat{E}_{12}\right\|_\infty, \left\|\widehat{E}_{12}^*\right\|_\infty\right\} = \frac{1}{2}\left\|\widehat{E}_{12}\right\|_\infty \\
&\leq \frac{1}{2}\sum_{j=k_o+2}^{n/2+1} g_j^{(\beta)} \leq \frac{1}{2}\sum_{j=k_o+2}^{\infty} g_j^{(\beta)} < \frac{3^{\beta+1}}{2\beta(3-\beta)^{\beta+1}\Gamma(-\beta)}\frac{1}{(k_o+1)^\beta} \\
&= \frac{\theta}{2(k_o+1)^\beta} < \frac{1}{2}\epsilon.
\end{aligned} \tag{12}
$$

Hence, we can obtain the bound

$$
\begin{aligned}
\|F_{\mathrm{C}}(\alpha)\|_2 &\leq \frac{1}{2}\|\Omega+\Gamma\|_2\left\|\widehat{E}_a\right\|_2 \leq \frac{1}{2}\|\Omega+\Gamma\|_\infty\left\|\widehat{E}_a\right\|_2 \\
&\leq \frac{1}{2}(\|\Omega\|_\infty + \|\Gamma\|_\infty)\left\|\widehat{E}_a\right\|_2 \leq \frac{1}{2}\nu_{\max}\epsilon.
\end{aligned}
$$

Now we turn to demonstrate (ii). On the basis of conclusion (i), we know that

$$
\begin{aligned}
M_{\mathrm{C}}(\alpha)^{-1}A &= M(\alpha)^{-1}A + \left(M_{\mathrm{C}}(\alpha)^{-1} - M(\alpha)^{-1}\right)A \\
&= M(\alpha)^{-1}A + M_{\mathrm{C}}(\alpha)^{-1}(M(\alpha) - M_{\mathrm{C}}(\alpha))M(\alpha)^{-1}A \\
&= M(\alpha)^{-1}A + M_{\mathrm{C}}(\alpha)^{-1}(E_{\mathrm{C}}(\alpha) + F_{\mathrm{C}}(\alpha))M(\alpha)^{-1}A \\
&= M(\alpha)^{-1}A + E(\alpha) + F(\alpha),
\end{aligned}
$$

where

$$
\begin{cases}
E(\alpha) = M_{\mathrm{C}}(\alpha)^{-1}E_{\mathrm{C}}(\alpha)M(\alpha)^{-1}A, \\
F(\alpha) = M_{\mathrm{C}}(\alpha)^{-1}F_{\mathrm{C}}(\alpha)M(\alpha)^{-1}A.
\end{cases}
$$

Hence, it also holds that $\mathrm{rank}(E(\alpha)) \leq 2k_o$.

Moreover, it follows from

$$
M(\alpha)^{-1}A = I - L(\alpha) \quad \text{and} \quad \|L(\alpha)\|_2 < 1 \quad (\forall \alpha > \alpha_+)
$$

that

$$
\|M(\alpha)^{-1}A\|_2 \leq 1 + \|L(\alpha)\|_2 < 2, \tag{13}
$$

where α_+ is the positive constant defined in Theorem 1. Also, for $n \geq 6$, by a straightforward application of the bound given in the proof of Theorem 2 in the work of Bai,[7] we can achieve the estimate

$$
\begin{aligned}
\|(\alpha I + H_C)^{-1}\|_2 \leq \|(\alpha I + H_C)^{-1}\|_\infty &\leq \frac{1}{\alpha - \sum_{j=0}^{n/2} g_j^{(\beta)}} \\
&= \frac{1}{\alpha + \sum_{j=n/2+1}^{\infty} g_j^{(\beta)}} \leq \frac{1}{\alpha + 2^\beta æ h^\beta}.
\end{aligned} \tag{14}
$$

Recalling

$$
F(\alpha) = (\alpha I + H_C)^{-1}\widehat{E}_a M(\alpha)^{-1}A,
$$

from Equations (12), (13), and (14), we can obtain the bound

$$
\|F(\alpha)\|_2 \leq \|(\alpha I + H_C)^{-1}\|_2\left\|\widehat{E}_a\right\|_2\|M(\alpha)^{-1}A\|_2 \leq \frac{\epsilon}{\alpha + 2^\beta æ h^\beta}.
$$

□

Theorem 2 indicates that the FRHS-preconditioned matrix $M_C(\alpha)^{-1}A$ is a good approximation to the RHS-preconditioned matrix $M(\alpha)^{-1}A$ in terms of both rank and norm. Hence, if the eigenvalues of $M(\alpha)^{-1}A$ are tightly clustered and its (normalized) eigenvectors are well conditioned, then the Krylov subspace iteration methods, when incorporated with the FRHS preconditioner, are expected to converge to the exact solution of the discrete linear system in Equation (2) accurately and stably (see the work of Bai[15] for more details). In fact, from Theorem 1, we see that this situation happens when the diffusion coefficients $\omega(x)$ and $\gamma(x)$ are almost equal and the matrix H is well conditioned.

4 | NUMERICAL RESULTS

In this section, we test and examine the numerical behavior of the FRHS preconditioner by solving the discrete linear system in Equation (2) with the FRHS-GMRES and FRHS-BiCGSTAB methods. The advantages of the FRHS-GMRES and FRHS-BiCGSTAB methods are shown in aspects of number of iteration steps (IT) and computing time in seconds (CPU). As comparisons, we also implement the *geometric multigrid* (GMG) method, as well as GMRES and BiCGSTAB incorporated with the *circulant-based approximate inverse* (CAI) preconditioner,[23] the *tridiagonal approximation* (TA) preconditioner (for the one-dimensional fractional diffusion equation) and the *block tridiagonal approximation* (BTA) preconditioner (for the two-dimensional fractional diffusion equation),[24] and the *circulant-based approximated and scaled Hermitian splitting* (CASHS) preconditioner

$$P_{\text{CASHS}} = \frac{1}{2}(\Omega + \Gamma)H_C$$

obtained through the intuitive approximations

$$\frac{1}{2}(\Omega + \Gamma) \approx \Omega, \Gamma \quad \text{and} \quad H_C \approx H,$$

where H_C is Strang's circulant approximation defined in Equation (10) or in Equation (11). These methods are abbreviated as CAI-GMRES(BiCGSTAB), (B)TA-GMRES(BiCGSTAB), and CASHS-GMRES(BiCGSTAB), respectively. Note that P_{CASHS} is the limit of $M_C(\alpha)$ when $\alpha \to 0_+$, so the CASHS preconditioner is a special case of the FRHS preconditioner for $\alpha = 0$.

The GMG is a V-cycle multigrid method[25] with the banded splitting iteration of half-bandwidth 2 as the pre- and post-smoothing processes.[26] It was demonstrated in the work of Lin et al.[26] that such a banded splitting iteration smoother is much better than either the damped Jacobi smoother or the Gauss–Seidel smoother in iteration steps or computing times. We remark that the multigrid methods used in the works of Pang and Sun[25] and Zhao et al.[27] are also the V-cycle version with only their smoothers being different; in the work of Pang and Sun,[25] they are the damped Jacobi iterations, whereas in the work of Zhao et al.,[27] they are the Gauss–Seidel iterations.

In our computations, all experiments are started from the initial vector $u_0^h = 0$, and terminated once either the relative residual error at the current iterate u_k^h satisfies $\|b - Au_k^h\|_2 \leq 10^{-5}\|b\|_2$ or the iteration sequence does not converge within 3,000 steps (denoted as "—"). In addition, all experiments are carried out using MATLAB (version R2014a) on a personal computer with 3.60 GHz central processing unit (Intel(R) Core(TM) i7-4790 CPU @3.60GHz), 8.00 GB memory, and Windows 7 operating system.

In addition, according to the parameter α involved in the FRHS preconditioner, we set it to be the experimentally computed optimal value that minimizes the total number of iteration steps of the FRHS-GMRES or the FRHS-BiCGSTAB method.

Also, we remark that for $\beta = 1.1$ the (B)TA preconditioning matrix is constructed by approximating the Toeplitz matrix T with the shifted first-order derivative operator,[24] that is, $\text{tridiag}(0, 1 + h, -1)$, in order to avoid its singularity. And for the other derivative order β, we approximate the Toeplitz matrix T with the Laplacian matrix $\text{tridiag}(-1, 2, -1)$, that is, the tridiagonal matrix with 2 on the main diagonal and -1 on the first lower and upper off-diagonals.

Example 1. We consider the fractional diffusion equation in Equation (1) of the order $\beta = 1.1, 1.3, 1.5, 1.7$, or 1.9. The nonnegative coefficient functions $\omega(x)$ and $\gamma(x)$ are given by

$$\omega(x) = \begin{cases} 1 + (x+3)^2 + x, & \text{if} \quad x \in \left[0, \frac{1}{8}\right), \\ 1 + \frac{8(x+9)^2}{x^3}, & \text{if} \quad x \in \left[\frac{1}{8}, 1\right] \end{cases}$$

and

$$\gamma(x) = \begin{cases} 1 + (x+3)^2, & \text{if} \quad x \in \left[0, \frac{1}{8}\right), \\ 1 + \frac{8(x+9)^2}{x^3} + 4(2-x), & \text{if} \quad x \in \left[\frac{1}{8}, 1\right], \end{cases}$$

and the right-hand side function $f(x)$ is taken as

$$f(x) = -\frac{1}{\Gamma(2-\beta)}\left[\omega(x)x^{1-\beta} + \gamma(x)(1-x)^{1-\beta}\right] + \frac{2}{\Gamma(3-\beta)}\left[\omega(x)x^{2-\beta} + \gamma(x)(1-x)^{2-\beta}\right],$$

so that the exact solution of the fractional diffusion equation in Equation (1) is $u(x) = x(1-x)$.

For this example, the coefficient functions $\omega(x)$ and $\gamma(x)$ are discontinuous and have very big jumps at $x = \frac{1}{8}$, although $\omega(x) \approx \gamma(x)$, $\forall x \in [0,\ 1]$, as it holds that

$$|\omega(x) - \gamma(x)| \leq \begin{cases} x, & \text{if} \quad x \in \left[0, \frac{1}{8}\right), \\ 4(2-x), & \text{if} \quad x \in \left[\frac{1}{8}, 1\right]. \end{cases}$$

For both FRHS-GMRES and FRHS-BiCGSTAB methods, the experimentally computed optimal values of the parameter α are listed in Table 1. Correspondingly, in Tables 2–6 we report the numerical results for all implemented methods with respect to various tested derivative orders of β. From Tables 2–6, we see that the CAI-GMRES and CAI-BiCGSTAB methods take much more iteration steps and CPU times than the FRHS-GMRES, FRHS-BiCGSTAB, TA-GMRES, and TA-BiCGSTAB methods, and they even fail to converge for almost all β with respect to a few large values of n. This

TABLE 1 The experimentally computed optimal values of the parameter α in the FRHS-GMRES and FRHS-BiCGSTAB methods for Example 1

β	Method	n						
		8,191	**16,383**	**32,767**	**65,535**	**131,071**	**262,143**	**524,287**
1.1	FRHS-GMRES	1E−7	1E−6	1E−7	1E−7	1E−7	1E−8	1E−8
	FRHS-BiCGSTAB	1E−6	1E−6	5E−7	7E−8	1E−8	2E−8	3E−8
1.3	FRHS-GMRES	1E−6	1E−7	1E−8	1E−7	1E−8	1E−8	1E−9
	FRHS-BiCGSTAB	7E−6	7E−6	1E−7	1E−7	1E−7	1E−8	2E−8
1.5	FRHS-GMRES	1E−7	1E−7	1E−8	1E−8	1E−8	1E−9	1E−9
	FRHS-BiCGSTAB	1E−6	1E−7	8E−8	1E−8	9E−9	2E−9	2E−10
1.7	FRHS-GMRES	1E−7	1E−8	1E−8	1E−9	1E−9	1E−9	1E−9
	FRHS-BiCGSTAB	1E−6	1E−7	4E−8	2E−8	1E−8	5E−9	4E−10
1.9	FRHS-GMRES	1E−8	1E−8	1E−9	1E−9	1E−9	1E−9	2E−10
	FRHS-BiCGSTAB	3E−7	3E−7	3E−9	1E−10	2E−10	1E−10	3E−10

TABLE 2 Iteration steps and computing times of the experimented methods for Example 1 when $\beta = 1.1$

Method	Index	n						
		8,191	**16,383**	**32,767**	**65,535**	**131,071**	**262,143**	**524,287**
CAI-GMRES	IT	854	958	1,008	1,031	1,048	1,064	1,082
	CPU	32.52	56.00	125.04	309.80	1,824.63	3,786.49	10,255.08
TA-GMRES	IT	20	25	32	41	56	76	104
	CPU	0.35	0.50	1.13	2.46	10.02	29.61	161.13
FRHS-GMRES	IT	5	6	6	6	6	6	7
	CPU	0.22	0.36	0.70	1.34	2.94	5.53	16.50
CAI-BiCGSTAB	IT	1,461.5	1,540.0	1,407.0	2,246.5	2,780.5	1,973.5	—
	CPU	16.17	32.33	70.44	173.37	802.44	747.04	—
TA-BiCGSTAB	IT	14.0	16.5	23.0	28.5	39.5	71.0	114.5
	CPU	0.31	0.59	1.55	3.00	10.10	31.95	250.26
FRHS-BiCGSTAB	IT	4.5	5.0	4.5	4.5	6.0	6.0	6.5
	CPU	0.22	0.40	0.79	1.46	3.61	6.61	26.00
GMG	IT	19	47	106	231	495	1,053	2,194
	CPU	21.73	54.79	159.70	434.96	1,461.52	5,256.40	28,133.26

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	860	927	1,088	1,125	1,149	1,180	1,208
	CPU	33.25	55.93	139.11	352.72	1,989.33	4,348.19	11,784.94
TA-GMRES	IT	21	25	29	33	39	45	52
	CPU	0.29	0.53	1.19	2.40	7.99	18.22	65.85
FRHS-GMRES	IT	6	6	6	6	6	7	7
	CPU	0.23	0.36	0.70	1.36	2.94	5.65	16.55
CAI-BiCGSTAB	IT	980.5	1,562.5	1,195.5	2,442.0	—	—	—
	CPU	11.49	35.00	67.04	197.50	—	—	—
TA-BiCGSTAB	IT	20.5	23.0	25.5	33.0	34.0	53.5	73.5
	CPU	0.39	0.74	1.79	3.56	10.59	26.67	170.36
FRHS-BiCGSTAB	IT	4.5	5.0	5.5	6.0	6.0	6.5	6.5
	CPU	0.23	0.40	0.85	1.54	3.63	6.82	25.90
GMG	IT	22	63	163	409	1,014	2,497	—
	CPU	26.17	78.17	239.81	795.66	3,172.99	12,911.90	—

TABLE 3 Iteration steps and computing times of the experimented methods for Example 1 when $\beta = 1.3$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	919	1,072	1,172	1,076	1,242	—	—
	CPU	39.47	72.73	160.48	312.42	2,300.96	—	—
TA-GMRES	IT	17	19	22	25	29	33	37
	CPU	0.27	0.48	1.04	2.06	6.19	13.54	45.85
FRHS-GMRES	IT	5	6	6	6	6	6	7
	CPU	0.22	0.36	0.71	1.34	2.92	5.53	16.48
CAI-BiCGSTAB	IT	1,058.5	1,028.5	2,894.5	2,733.5	—	—	—
	CPU	11.83	21.69	146.74	205.76	—	—	—
TA-BiCGSTAB	IT	17.5	17.5	17.0	18.0	24.0	25.5	32.0
	CPU	0.36	0.64	1.38	2.53	8.37	15.57	85.09
FRHS-BiCGSTAB	IT	5.5	5.5	5.5	5.5	5.5	7.0	7.0
	CPU	0.23	0.40	0.83	1.51	3.51	6.95	27.32
GMG	IT	27	86	251	718	2,037	—	—
	CPU	31.83	106.28	369.03	1,321.79	5,855.97	—	—

TABLE 4 Iteration steps and computing times of the experimented methods for Example 1 when $\beta = 1.5$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	965	1,131	1,230	1,286	—	—	—
	CPU	42.94	80.60	175.36	439.94	—	—	—
TA-GMRES	IT	12	13	15	16	17	19	21
	CPU	0.25	0.42	0.92	1.75	4.43	9.22	28.48
FRHS-GMRES	IT	5	5	6	6	6	6	7
	CPU	0.22	0.35	0.71	1.36	2.97	5.54	16.69
CAI-BiCGSTAB	IT	1,124.5	1,490.5	2,769.0	—	—	—	—
	CPU	12.53	31.99	139.66	—	—	—	—
TA-BiCGSTAB	IT	8.0	9.5	11.0	10.5	12.5	15.0	15.0
	CPU	0.27	0.50	1.13	2.02	5.65	11.34	45.31
FRHS-BiCGSTAB	IT	5.0	5.0	5.0	6.5	7.0	7.5	8.0
	CPU	0.23	0.40	0.82	1.60	3.81	7.04	29.72
GMG	IT	32	116	386	1,262	—	—	—
	CPU	37.35	142.25	568.92	2,325.09	—	—	—

TABLE 5 Iteration steps and computing times of the experimented methods for Example 1 when $\beta = 1.7$

phenomenon may result from poor approximation of the CAI preconditioning matrix to the discrete matrix A in Equation (2) when the coefficient functions have big jumps. As for the GMG method, it also results in much more iteration steps and CPU times than the FRHS-GMRES, FRHS-BiCGSTAB, TA-GMRES, and TA-BiCGSTAB methods. Moreover, GMG also fails to converge for almost all β with respect to a few large values of n. These numerical behaviors may be

TABLE 6 Iteration steps and computing times of the experimented methods for Example 1 when $\beta = 1.9$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	1,002	—	—	—	—	—	—
	CPU	44.02	—	—	—	—	—	—
TA-GMRES	IT	7	7	7	8	8	9	9
	CPU	0.23	0.38	0.77	1.52	3.42	6.93	17.79
FRHS-GMRES	IT	5	5	5	5	6	7	7
	CPU	0.22	0.36	0.69	1.35	2.98	5.72	17.42
CAI-BiCGSTAB	IT	—	—	—	—	—	—	—
	CPU	—	—	—	—	—	—	—
TA-BiCGSTAB	IT	4.0	4.0	4.5	4.5	5.0	5.0	5.0
	CPU	0.23	0.39	0.84	1.60	3.82	7.20	22.20
FRHS-BiCGSTAB	IT	5.0	5.5	5.0	5.5	6.0	7.5	8.0
	CPU	0.23	0.40	0.80	1.51	3.59	7.04	29.58
GMG	IT	37	155	590	2,212	—	—	—
	CPU	49.80	219.66	976.27	4,516.90	—	—	—

caused by the ineffective smoothers due to the cloudy diagonal dominance of the discrete matrix $A := (a_{ij}) \in \mathbb{R}^{n\times n}$ given in Equations (2) and (3) when $\beta \in (1, 2)$, in the sense of the following facts:

- for all row indices $i \in \{1, 2, \ldots, n\}$, it holds that

$$\frac{1}{\beta} \leq \frac{1}{|a_{ii}|}\sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}| \leq 1 - \mathcal{O}(h^{\beta}); \tag{15}$$

- for a number of row indices, the bounds of the inequality in Equation (15) are approached with a fair degree of accuracy; and
- for some of the row indices, the bounds of the inequality in Equation (15) are even achieved exactly.

Note that the upper bound of the inequality in Equation (15) is approaching to 1 when step size h becomes smaller and smaller.

In addition, when $\beta \leq 1.5$, the iteration steps and CPU times of the FRHS-GMRES and FRHS-BiCGSTAB methods are much less than those of the TA-GMRES and TA-BiCGSTAB methods; and the CPU times of the FRHS-GMRES method are less than those of the FRHS-BiCGSTAB method, although they both have about the same iteration counts. When $\beta = 1.7$, the TA-GMRES and FRHS-BiCGSTAB methods cost about the same CPU times, which are more than the CPU times of the FRHS-GMRES method. Hence, when $\beta \leq 1.7$, the FRHS-GMRES method is the most effective among all of these methods. Furthermore, when $\beta = 1.9$, the CPU times of the TA-GMRES and FRHS-GMRES methods are about equal, and they are less than those of the TA-BiCGSTAB and FRHS-BiCGSTAB methods. Hence, for this case, both TA-GMRES and FRHS-GMRES methods are the most effective among all of these methods.

It is shown in Table 1 that the experimentally computed optimal values of the parameter α involved in the FRHS preconditioner are very close to 0. Therefore, we can expect that the parameter-free methods CASHS-GMRES and CASHS-BiCGSTAB should perform as good as the optimal FRHS-GMRES and FRHS-BiCGSTAB methods. This deduction is confirmed by the numerical results in Table 7. As a matter of fact, by comparing the results in this table with those in Tables 2–6, we see that the iteration steps and CPU times of the CASHS-GMRES and CASHS-BiCGSTAB methods are about the same as those of the FRHS-GMRES and FRHS-BiCGSTAB methods, respectively, when $\beta \leq 1.7$. However, when $\beta = 1.9$, the CASHS-preconditioned methods take more CPU times than the corresponding FRHS-preconditioned methods, especially for the largest grid number $n = 524,287$. For $\beta = 1.9$ and $n = 524,287$, the CASHS-BiCGSTAB method requires much more iteration steps, implying that the CASHS preconditioner can be unstable which, however, does not happen to the FRHS preconditioner.

Example 2. We consider the fractional diffusion equation in Equation (1) of the order $\beta = 1.1, 1.3, 1.5, 1.7$, or 1.9. The nonnegative coefficient functions $\omega(x)$ and $\gamma(x)$ are given by

$$\omega(x) = 10|\sin(2\pi x)| + 1 - 0.5\sin(\pi x), \qquad \gamma(x) = 10|\sin(2\pi x)| + 1 + (x - 0.8)\sin(\pi x),$$

β	Method	Index	n						
			8,191	**16,383**	**32,767**	**65,535**	**131,071**	**262,143**	**524,287**
1.1	CASHS-GMRES	IT	5	6	6	6	6	6	7
		CPU	0.24	0.38	0.73	1.40	3.09	5.74	17.07
	CASHS-BiCGSTAB	IT	4.5	5.0	5.0	5.0	7.0	7.0	7.0
		CPU	0.26	0.40	0.82	1.49	4.40	6.72	26.38
1.3	CASHS-GMRES	IT	6	6	6	6	6	7	7
		CPU	0.23	0.37	0.72	1.40	3.12	5.87	17.18
	CASHS-BiCGSTAB	IT	5.5	5.5	5.5	6.0	6.0	9.0	9.0
		CPU	0.28	0.42	0.84	1.56	4.11	7.31	30.96
1.5	CASHS-GMRES	IT	5	6	6	6	6	6	7
		CPU	0.22	0.37	0.72	1.39	3.03	5.65	16.94
	CASHS-BiCGSTAB	IT	6.5	5.5	5.5	5.5	6.0	8.0	7.0
		CPU	0.30	0.41	0.85	1.54	4.11	6.96	26.02
1.7	CASHS-GMRES	IT	5	5	6	6	6	6	6
		CPU	0.23	0.36	0.72	1.39	3.01	5.73	16.05
	CASHS-BiCGSTAB	IT	5.0	6.5	7.0	8.0	10.0	12.0	11.0
		CPU	0.27	0.44	0.91	1.69	5.33	8.18	35.54
1.9	CASHS-GMRES	IT	5	5	5	5	6	9	6
		CPU	0.23	0.36	0.71	1.37	3.19	7.76	29.91
	CASHS-BiCGSTAB	IT	5.5	5.5	5.5	6.0	12.0	34.0	239.5
		CPU	0.28	0.42	0.86	1.55	5.77	14.12	613.62

TABLE 7 Iteration steps and computing times of the CASHS-GMRES and CASHS-BiCGSTAB methods for Example 1

and the right-hand side function $f(x)$ is taken as

$$f(x) = 8 + (200 + x)^2 x^{1-\beta} + (10 + x)^2 x^{2-\beta}.$$

For this example, the coefficient functions $\omega(x)$ and $\gamma(x)$ are continuous but strongly oscillating, although $\omega(x) \approx \gamma(x)$, $\forall x \in [0,\ 1]$, as it holds that

$$|\omega(x) - \gamma(x)| = |0.3 - x|\,|\sin(\pi x)|.$$

For both FRHS-GMRES and FRHS-BiCGSTAB methods, the experimentally computed optimal values of the parameter α are listed in Table 8. Correspondingly, in Tables 9–13 we report the numerical results for all implemented methods with respect to various tested derivative orders of β. In Tables 9–13, we see that the GMG method takes much more iteration steps and CPU times than the other methods. Comparatively, the iteration counts and CPU times of both CAI-GMRES and CAI-BiCGSTAB methods are moderate due to continuity of the diffusion coefficients $\omega(x)$ and $\gamma(x)$, which are, however, still more than those of the FRHS-GMRES and FRHS-BiCGSTAB methods. Note that the CAI-GMRES method fails to converge when $\beta = 1.7$ with $n = 524{,}287$, and when $\beta = 1.9$ with $n = 262{,}143$ and $524{,}287$, because of the stagnation occurred in the iteration history.

When $\beta \leq 1.7$, the iteration steps and CPU times of the TA-GMRES and TA-BiCGSTAB methods are more than those of the FRHS-GMRES and FRHS-BiCGSTAB methods; and when $\beta = 1.9$, they are almost the same for both TA-GMRES and FRHS-GMRES methods but less than those of the TA-BiCGSTAB and FRHS-BiCGSTAB methods. Hence, the FRHS-GMRES and FRHS-BiCGSTAB methods are the most effective for solving the discrete linear system when

β	Method	n						
		8,191	**16,383**	**32,767**	**65,535**	**131,071**	**262,143**	**524,287**
1.1	FRHS-GMRES	1E−6	1E−6	1E−6	1E−6	1E−7	1E−7	1E−7
	FRHS-BiCGSTAB	1E−5	2E−6	1E−6	4E−8	4E−8	1E−9	2E−9
1.3	FRHS-GMRES	1E−6	1E−6	1E−7	1E−7	1E−7	1E−7	1E−8
	FRHS-BiCGSTAB	7E−6	3E−6	1E−6	5E−7	2E−8	4E−9	4E−9
1.5	FRHS-GMRES	1E−6	1E−8	1E−8	1E−7	1E−8	8E−9	3E−9
	FRHS-BiCGSTAB	2E−6	1E−7	1E−7	1E−7	1E−9	1E−9	4E−9
1.7	FRHS-GMRES	1E−7	1E−7	1E−8	8E−9	1E−9	1E−9	1E−9
	FRHS-BiCGSTAB	2E−7	1E−7	4E−8	8E−9	5E−9	1E−9	1E−10
1.9	FRHS-GMRES	1E−8	1E−8	1E−9	1E−9	2E−10	1E−10	1E−10
	FRHS-BiCGSTAB	4E−8	1E−8	1E−9	1E−9	3E−10	1E−10	3E−11

TABLE 8 The experimentally computed optimal values of the parameter α in the FRHS-GMRES and FRHS-BiCGSTAB methods for Example 2

TABLE 9 Iteration steps and computing times of the experimented methods for Example 2 when $\beta = 1.1$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	34	36	37	37	38	38	39
	CPU	1.03	1.04	1.94	2.43	12.34	12.76	255.40
TA-GMRES	IT	51	73	103	142	194	267	368
	CPU	0.31	0.81	2.62	8.32	60.59	252.03	1,204.80
FRHS-GMRES	IT	13	14	15	16	16	17	17
	CPU	0.19	0.19	0.38	0.63	2.71	3.29	17.73
CAI-BiCGSTAB	IT	21.5	24.0	26.5	28.5	40.0	30.5	32.5
	CPU	0.30	0.60	1.47	2.33	11.92	11.96	111.86
TA-BiCGSTAB	IT	36.5	63.0	266.0	—	—	—	—
	CPU	0.40	1.17	11.25	—	—	—	—
FRHS-BiCGSTAB	IT	10.0	12.0	12.5	13.0	14.5	15.5	16.5
	CPU	0.24	0.29	0.66	0.96	4.54	5.22	40.79
GMG	IT	19	48	108	235	503	1,070	2,269
	CPU	21.17	56.16	164.41	446.26	1,502.13	5,388.44	28,956.04

TABLE 10 Iteration steps and computing times of the experimented methods for Example 2 when $\beta = 1.3$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	28	28	29	29	29	29	29
	CPU	0.55	0.50	1.01	1.72	6.98	6.84	39.87
TA-GMRES	IT	41	50	60	72	86	104	128
	CPU	0.28	0.55	1.47	3.45	18.08	50.84	215.43
FRHS-GMRES	IT	9	10	10	10	11	11	11
	CPU	0.16	0.15	0.29	0.49	2.00	2.33	12.07
CAI-BiCGSTAB	IT	20.0	19.0	25.5	23.5	26.5	31.0	27.0
	CPU	0.28	0.48	1.41	1.99	8.10	12.13	93.84
TA-BiCGSTAB	IT	31.0	41.5	50.0	70.5	86.0	121.5	150.5
	CPU	0.37	0.86	2.44	5.47	21.86	51.53	356.37
FRHS-BiCGSTAB	IT	7.0	7.5	8.0	8.5	9.0	10.0	10.0
	CPU	0.19	0.21	0.47	0.72	3.06	3.70	25.62
GMG	IT	23	64	166	417	1,030	2,535	—
	CPU	25.05	84.31	265.19	777.49	3,012.27	12,424.58	—

TABLE 11 Iteration steps and computing times of the experimented methods for Example 2 when $\beta = 1.5$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	27	27	26	26	27	27	27
	CPU	0.48	0.40	0.78	1.31	6.50	6.39	37.01
TA-GMRES	IT	30	35	41	48	55	64	73
	CPU	0.21	0.38	0.95	2.12	9.28	23.38	169.30
FRHS-GMRES	IT	8	8	9	9	9	9	10
	CPU	0.15	0.14	0.27	0.45	1.70	2.00	11.29
CAI-BiCGSTAB	IT	24.5	24.5	24.5	27.0	32.5	33.5	30.5
	CPU	0.34	0.60	1.39	2.25	9.82	13.02	105.41
TA-BiCGSTAB	IT	19.5	25.0	30.5	33.5	39.5	54.0	61.5
	CPU	0.26	0.55	1.53	2.73	10.32	23.51	146.95
FRHS-BiCGSTAB	IT	7.0	8.5	8.5	8.5	9.5	10.0	10.0
	CPU	0.19	0.23	0.49	0.70	3.10	3.64	25.60
GMG	IT	26	84	245	701	1,985	—	—
	CPU	28.30	101.33	366.08	1,306.84	5,794.64	—	—

$\beta \le 1.7$, whereas when $\beta = 1.9$, the TA-GMRES and FRHS-GMRES methods are the best choices. We should point out that the TA-BiCGSTAB method fails to converge when $\beta = 1.1$ with $n \ge 65,535$, which may result from singularity of the TA preconditioning matrix.

Note that the experimentally computed optimal values of the parameter α involved in the FRHS preconditioner are very close to 0, especially when the grid number n becomes large (see Table 8). Therefore, we can expect again that the

TABLE 12 Iteration steps and computing times of the experimented methods for Example 2 when $\beta = 1.7$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	24	24	24	24	25	28	—
	CPU	0.45	0.36	0.71	1.18	6.02	7.35	—
TA-GMRES	IT	18	20	23	25	28	30	36
	CPU	0.14	0.21	0.51	0.94	3.41	7.16	32.12
FRHS-GMRES	IT	7	8	8	8	8	8	9
	CPU	0.15	0.14	0.26	0.43	1.63	1.90	10.44
CAI-BiCGSTAB	IT	26.5	25.5	29.5	27.0	39.5	47.0	49.0
	CPU	0.36	0.62	1.64	2.28	11.81	17.96	168.42
TA-BiCGSTAB	IT	10.5	12.0	14.0	16.0	17.5	19.5	21.5
	CPU	0.17	0.30	0.77	1.46	4.95	9.32	53.06
FRHS-BiCGSTAB	IT	7.5	7.5	8.5	8.5	9.5	10.5	11.5
	CPU	0.20	0.21	0.48	0.71	3.12	3.80	29.12
GMG	IT	29	103	341	1,099	—	—	—
	CPU	33.59	126.51	500.13	2,013.02	—	—	—

TABLE 13 Iteration steps and computing times of the experimented methods for Example 2 when $\beta = 1.9$

Method	Index	n						
		8,191	16,383	32,767	65,535	131,071	262,143	524,287
CAI-GMRES	IT	22	22	22	38	33	—	—
	CPU	0.44	0.35	0.69	1.82	6.15	—	—
TA-GMRES	IT	8	8	9	9	9	10	10
	CPU	0.10	0.13	0.28	0.50	1.33	3.00	10.48
FRHS-GMRES	IT	6	6	6	6	6	7	8
	CPU	0.14	0.12	0.22	0.38	1.32	1.78	9.41
CAI-BiCGSTAB	IT	28.0	31.0	40.0	39.5	44.5	48.0	55.0
	CPU	0.37	0.74	2.15	3.17	13.48	18.24	188.37
TA-BiCGSTAB	IT	4.5	5.0	5.5	5.5	6.0	7.0	7.5
	CPU	0.11	0.17	0.39	0.69	2.12	4.11	20.22
FRHS-BiCGSTAB	IT	6.0	7.0	8.0	8.0	8.0	9.5	10.5
	CPU	0.17	0.20	0.46	0.68	2.72	3.53	26.77
GMG	IT	31	123	452	1,635	—	—	—
	CPU	32.93	150.19	660.48	2,990.84	—	—	—

parameter-free methods CASHS-GMRES and CASHS-BiCGSTAB should perform as good as the optimal FRHS-GMRES and FRHS-BiCGSTAB methods. This deduction is confirmed by the numerical results in Table 14.

Example 3. We consider the following two-dimensional counterpart of the fractional diffusion equation in Equation (1):

$$-\omega(x,y)\left(\frac{\partial^{\beta}u(x,y)}{\partial_{+}x^{\beta}}+\frac{\partial^{\beta}u(x,y)}{\partial_{+}y^{\beta}}\right)-\gamma(x,y)\left(\frac{\partial^{\beta}u(x,y)}{\partial_{-}x^{\beta}}+\frac{\partial^{\beta}u(x,y)}{\partial_{-}y^{\beta}}\right)=f(x,y), \tag{16}$$

in which the nonnegative coefficient functions $\omega(x,y)$ and $\gamma(x,y)$ are given by

$$\omega(x,y)=\begin{cases}1+(x+3)(y+3)+xy, & \text{if} \quad (x,y)\in\left[0,\frac{1}{8}\right)\times\left[0,\frac{1}{8}\right),\\ 1+\frac{8(x+9)(y+9)}{xy}, & \text{otherwise,}\end{cases}$$

and

$$\gamma(x,y)=\begin{cases}1+(x+3)(y+3), & \text{if} \quad (x,y)\in\left[0,\frac{1}{8}\right)\times\left[0,\frac{1}{8}\right),\\ 1+\frac{8(x+9)(y+9)}{xy}+(2-x)(2-y), & \text{otherwise,}\end{cases}$$

TABLE 14 Iteration steps and computing times of the CASHS-GMRES and CASHS-BiCGSTAB methods for Example 2

β	Method	Index	n						
			8,191	16,383	32,767	65,535	131,071	262,143	524,287
1.1	CASHS-GMRES	IT	13	14	15	16	16	17	17
		CPU	0.21	0.20	0.41	0.70	2.78	3.60	18.23
	CASHS-BiCGSTAB	IT	11.0	12.5	12.5	13.5	15.5	16.5	17.5
		CPU	0.26	0.31	0.67	1.05	4.81	5.51	42.65
1.3	CASHS-GMRES	IT	9	10	10	10	11	11	11
		CPU	0.17	0.17	0.32	0.54	2.05	2.56	12.47
	CASHS-BiCGSTAB	IT	8.0	8.5	9.5	10.0	10.5	13.5	12.5
		CPU	0.21	0.23	0.53	0.86	3.42	4.69	31.30
1.5	CASHS-GMRES	IT	8	8	9	9	9	10	10
		CPU	0.16	0.16	0.32	0.55	1.82	2.46	11.46
	CASHS-BiCGSTAB	IT	8.5	8.5	8.5	10.0	11.0	10.0	11.5
		CPU	0.22	0.23	0.49	0.86	3.54	3.66	29.03
1.7	CASHS-GMRES	IT	7	8	8	8	8	9	9
		CPU	0.16	0.16	0.30	0.50	1.74	2.37	10.79
	CASHS-BiCGSTAB	IT	8.0	8.5	9.0	10.5	11.5	11.5	11.5
		CPU	0.21	0.23	0.51	0.82	3.71	4.10	29.11
1.9	CASHS-GMRES	IT	6	6	6	6	7	6	6
		CPU	0.15	0.14	0.26	0.46	1.73	3.44	18.17
	CASHS-BiCGSTAB	IT	8.0	8.0	8.5	9.5	10.5	11.5	12.5
		CPU	0.21	0.22	0.50	0.79	3.41	4.09	31.33

and the right-hand side function $f(x,y)$ is taken as

$$\begin{aligned} f(x,y) = & -2y^2(1-y)^2[\omega(x,y)\,g(x,\beta)+\gamma(x,y)\,g(1-x,\beta)] \\ & -2x^2(1-x)^2[\omega(x,y)\,g(y,\beta)+\gamma(x,y)\,g(1-y,\beta)], \end{aligned}$$

with

$$g(\varsigma,\beta)=\frac{12\varsigma^{4-\beta}}{\Gamma(5-\beta)}-\frac{6\varsigma^{3-\beta}}{\Gamma(4-\beta)}+\frac{\varsigma^{2-\beta}}{\Gamma(3-\beta)},$$

so that the exact solution of the fractional diffusion equation in Equation (16) is

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

When a discretization scheme analogously to that used for the one-dimensional equation is applied to this two-dimensional one, we can obtain a discrete linear system of the form

$$Au^h := \left[\Omega(I\otimes T+T\otimes I)+\Gamma(I\otimes T^*+T^*\otimes I)\right]u^h = h^\beta f^h := b, \tag{17}$$

where $\otimes$ is the Kronecker product symbol, T is the Toeplitz matrix defined in Equation (3), and u^h and f^h are two-dimensional extensions of their one-dimensional counterparts.

Because the GMG method performs poorly in terms of iteration steps and computing times for the one-dimensional fractional diffusion equations in Examples 1 and 2, for this example we do not compare it with the other methods.

In Tables 16–20, we list the numerical results of the FRHS-GMRES, FRHS-BiCGSTAB, CAI-GMRES, CAI-BiCGSTAB, BTA-GMRES, and BTA-BiCGSTAB methods with respect to various tested derivative orders of β: 1.1, 1.3, 1.5, 1.7, or 1.9; among them, for the FRHS-GMRES and FRHS-BiCGSTAB methods, we adopt the experimentally computed optimal values of the parameter α given in Table 15. In Tables 16–20, we see that the CAI-GMRES and CAI-BiCGSTAB methods take much more iteration steps and CPU times than the FRHS-GMRES and FRHS-BiCGSTAB methods, and that both methods BTA-GMRES and BTA-BiCGSTAB cost more CPU times than the methods FRHS-GMRES and FRHS-BiCGSTAB, although they have less iteration steps when $\beta \geq 1.5$. Moreover, the BTA-GMRES and BTA-BiCGSTAB methods fail to compute an approximate solution for the discrete linear system in Equation (17) when $n = 2,047$ because of insufficient memory in the LU factorization process of the BTA preconditioning matrix of a block-tridiagonal with tridiagonal-block structure.

β	Method	n					
		63	127	255	511	1,023	2,047
1.1	FRHS-GMRES	1E−3	1E−3	1E−4	9E−5	1E−4	1E−6
	FRHS-BiCGSTAB	2E−3	1E−3	1E−3	2E−4	7E−5	7E−5
1.3	FRHS-GMRES	1E−3	1E−3	1E−4	1E−4	1E−4	5E−5
	FRHS-BiCGSTAB	3E−2	5E−3	5E−4	5E−4	9E−5	1E−4
1.5	FRHS-GMRES	1E−3	1E−3	1E−4	1E−4	1E−5	1E−5
	FRHS-BiCGSTAB	5E−3	9E−4	1E−4	9E−5	9E−5	6E−5
1.7	FRHS-GMRES	1E−3	3E−4	1E−4	2E−5	5E−6	3E−6
	FRHS-BiCGSTAB	9E−3	2E−4	9E−4	4E−5	4E−5	4E−6
1.9	FRHS-GMRES	2E−4	6E−5	1E−5	5E−6	1E−6	5E−7
	FRHS-BiCGSTAB	4E−3	5E−4	2E−4	5E−6	1E−6	8E−7

TABLE 15 The experimentally computed optimal values of the parameter α in the FRHS-GMRES and FRHS-BiCGSTAB methods for Example 3

Method	Index	n					
		63	127	255	511	1,023	2,047
CAI-GMRES	IT	23	28	37	50	72	102
	CPU	0.73	4.17	12.17	70.92	511.67	4,977.66
BTA-GMRES	IT	8	11	14	17	22	—
	CPU	0.62	2.17	10.00	52.96	280.41	—
FRHS-GMRES	IT	11	13	14	16	19	21
	CPU	0.58	1.86	7.68	31.64	132.10	549.84
CAI-BiCGSTAB	IT	22.0	20.0	30.5	39.5	66.5	100.0
	CPU	0.47	1.23	6.12	40.30	317.98	3,537.24
BTA-BiCGSTAB	IT	5.0	6.5	9.5	14.5	19.5	—
	CPU	0.55	2.26	10.70	63.76	340.99	—
FRHS-BiCGSTAB	IT	9.5	11.0	12.0	14.0	17.0	18.0
	CPU	0.56	2.00	7.94	35.96	147.73	637.78

TABLE 16 Iteration steps and computing times of the experimented methods for Example 3 when $\beta = 1.1$

Method	Index	n					
		63	127	255	511	1,023	2,047
CAI-GMRES	IT	21	25	35	49	70	102
	CPU	0.81	2.82	11.69	70.04	498.30	5,242.94
BTA-GMRES	IT	9	11	14	18	22	—
	CPU	1.23	2.28	9.97	53.87	279.69	—
FRHS-GMRES	IT	10	12	13	15	17	20
	CPU	0.58	1.86	7.57	32.56	134.44	570.71
CAI-BiCGSTAB	IT	15.5	20.0	32.0	43.0	73.5	107.0
	CPU	0.34	1.22	6.18	42.93	346.96	3,983.84
BTA-BiCGSTAB	IT	6.0	7.0	10.0	13.5	17.0	—
	CPU	0.64	2.25	11.05	61.42	318.62	—
FRHS-BiCGSTAB	IT	7.5	10.0	11.5	14.5	16.0	20.0
	CPU	0.55	1.95	7.83	36.19	144.39	658.70

TABLE 17 Iteration steps and computing times of the experimented methods for Example 3 when $\beta = 1.3$

Analogously, as the experimentally computed optimal values of the parameter α involved in the FRHS preconditioner are very small (see Table 15), we also expect that the parameter-free methods CASHS-GMRES and CASHS-BiCGSTAB should perform as good as the optimal FRHS-GMRES and FRHS-BiCGSTAB methods when they are employed to solve the discrete linear system in Equation (17) resulting from the two-dimensional fractional diffusion equation in Equation (16). From the numerical results in Table 21, we find that the CASHS-GMRES and CASHS-BiCGSTAB methods only take a little more iteration steps and CPU times than the FRHS-GMRES and FRHS-BiCGSTAB methods. Therefore, although FRHS-GMRES and FRHS-BiCGSTAB are the most effective methods for solving the discrete linear system in Equation (17), CASHS-GMRES and CASHS-BiCGSTAB are also computationally attractive as they do not need to turn any parameter.

TABLE 18 Iteration steps and computing times of the experimented methods for Example 3 when $\beta = 1.5$

Method	Index	*n*					
		63	**127**	**255**	**511**	**1,023**	**2,047**
CAI-GMRES	IT	21	28	38	54	79	117
	CPU	0.67	2.48	11.42	75.72	564.72	5,856.98
BTA-GMRES	IT	8	10	12	14	17	—
	CPU	0.59	2.20	9.77	49.83	255.84	—
FRHS-GMRES	IT	10	12	14	16	20	24
	CPU	0.58	1.95	7.43	31.78	133.38	579.31
CAI-BiCGSTAB	IT	15.5	23.5	36.0	56.0	97.5	154.5
	CPU	0.35	1.35	6.90	55.34	492.55	5,925.50
BTA-BiCGSTAB	IT	5.0	6.0	8.5	10.5	14.0	—
	CPU	0.55	2.20	10.58	55.94	295.82	—
FRHS-BiCGSTAB	IT	7.5	10.0	13.5	16.0	20.5	28.5
	CPU	0.55	1.94	7.99	38.66	163.16	806.24

TABLE 19 Iteration steps and computing times of the experimented methods for Example 3 when $\beta = 1.7$

Method	Index	*n*					
		63	**127**	**255**	**511**	**1,023**	**2,047**
CAI-GMRES	IT	23	32	44	63	93	140
	CPU	0.69	2.67	12.37	86.64	679.13	7,472.02
BTA-GMRES	IT	7	8	9	10	12	—
	CPU	0.55	2.07	9.24	46.39	236.61	—
FRHS-GMRES	IT	10	12	14	18	22	28
	CPU	0.59	1.86	7.36	32.31	136.69	612.43
CAI-BiCGSTAB	IT	18.5	28.5	46.5	79.5	147.0	298.5
	CPU	0.36	1.46	7.95	70.95	656.08	10,204.81
BTA-BiCGSTAB	IT	4.0	5.0	5.5	7.5	8.5	—
	CPU	0.55	2.17	9.58	50.75	253.20	—
FRHS-BiCGSTAB	IT	8.5	11.0	16.0	18.0	26.0	34.5
	CPU	0.55	1.98	8.41	38.55	168.32	820.91

TABLE 20 Iteration steps and computing times of the experimented methods for Example 3 when $\beta = 1.9$

Method	Index	*n*					
		63	**127**	**255**	**511**	**1,023**	**2,047**
CAI-GMRES	IT	27	37	52	77	117	179
	CPU	0.72	2.79	13.81	106.25	898.39	13,387.44
BTA-GMRES	IT	5	5	5	6	6	—
	CPU	0.55	2.00	8.70	43.12	212.85	—
FRHS-GMRES	IT	10	12	15	19	26	31
	CPU	0.56	1.84	7.58	32.74	144.10	645.06
CAI-BiCGSTAB	IT	23.0	34.5	67.5	134.0	340.5	676.0
	CPU	0.39	1.57	9.51	113.44	1,614.10	25,923.06
BTA-BiCGSTAB	IT	3.0	3.0	3.5	3.5	4.0	—
	CPU	0.52	2.04	8.97	43.21	218.01	—
FRHS-BiCGSTAB	IT	8.5	10.0	16.0	20.0	30.5	44.0
	CPU	0.55	1.97	8.32	39.73	178.53	931.01

For Examples 1–3, we have also implemented the GMRES and BiCGSTAB methods incorporated with the FRSHSS preconditioner proposed and analyzed in the work of Bai.[7] Analogously, these methods are abbreviated as FRSHSS-GMRES and FRSHSS-BiCGSTAB. From the numerical results, we see that the performance of these two methods is the worst among all of the tested methods in terms of both iteration counts and computing times.

More specifically, for Example 1, the FRSHSS-GMRES method is only convergent for $\beta = 1.1$, 1.3, and 1.5 when $n = 8,191$, with the iteration steps being 1,465, 1,563, and 1,731 and the computing times being 142.89, 161.38, and 200.17; whereas it is convergent only for $\beta = 1.1$ and 1.3 when $n = 16,383$, with the iteration steps being 2,105 and 2,273 and the computing times being 463.84 and 541.32. For Example 2, the FRSHSS-GMRES method is convergent for all tested values of β when $n = 8,191$, with the number of iteration steps ranging from 1,775 to 2,031 and the computing time ranging

β	Method	Index	n					
			63	**127**	**255**	**511**	**1,023**	**2,047**
1.1	CASHS-GMRES	IT	11	13	14	16	19	21
		CPU	0.69	1.84	7.30	31.39	130.34	544.35
	CASHS-BiCGSTAB	IT	10.0	11.0	17.0	20.5	20.5	22.5
		CPU	0.58	1.98	8.49	40.06	154.96	687.46
1.3	CASHS-GMRES	IT	11	12	13	16	18	21
		CPU	0.55	1.84	7.24	31.42	129.40	554.66
	CASHS-BiCGSTAB	IT	9.5	12.0	13.0	18.0	17.5	23.0
		CPU	0.56	2.01	8.03	38.19	148.31	689.85
1.5	CASHS-GMRES	IT	11	12	14	17	20	26
		CPU	0.56	1.81	7.27	31.28	131.27	589.29
	CASHS-BiCGSTAB	IT	9.5	11.5	14.5	19.0	27.5	35.0
		CPU	0.55	1.98	8.19	39.00	170.48	822.18
1.7	CASHS-GMRES	IT	10	13	15	19	24	28
		CPU	0.56	1.84	7.36	32.48	139.64	611.41
	CASHS-BiCGSTAB	IT	11.0	13.5	17.5	25.5	28.0	40.0
		CPU	0.58	2.04	8.55	42.96	171.94	879.73
1.9	CASHS-GMRES	IT	11	13	16	21	26	48
		CPU	0.55	1.84	7.44	33.67	144.25	835.16
	CASHS-BiCGSTAB	IT	10.0	16.5	20.0	34.0	43.0	74.5
		CPU	0.56	2.11	8.78	48.30	205.72	1,265.62

TABLE 21 Iteration steps and computing times of the CASHS-GMRES and CASHS-BiCGSTAB methods for Example 3

from 262.47 to 209.13; it is convergent for the values of β within the interval $[1.1,\ 1.7]$ when $n = 16,383$, with the number of iteration steps ranging from 2,483 to 2,761 and the computing time ranging from 638.03 to 793.60; and it is convergent only for $\beta = 1.1$ and 1.5 when $n = 32,767$, with the iteration steps being 2,816 and 2,779 and the computing times being 1,387.50 and 1,380.07. For Example 3, the FRSHSS-GMRES method is convergent for all tested values of β when $n = 63$ and 127, with the number of iteration steps decreasing from 439 down to 371 and the computing time decreasing from 9.46 down to 7.26 when $n = 63$, and with the number of iteration steps ranging from 1,452 to 1,360 and the computing time ranging from 244.50 to 211.36 when $n = 127$. For all other cases with respect to the tested grid number n and derivative order β, the FRSHSS-GMRES method cannot compute a satisfactory numerical solution for the discrete linear system in Equation (2) or for its two-dimensional counterpart in Equation (17). The computing results for the FRSHSS-BiCGSTAB method are similar to those for the FRSHSS-GMRES method.

In fact, this numerical phenomenon happens naturally, as both FRHS and FRSHSS are tailor-made preconditioners for solving the discrete linear system in Equation (2) or in Equation (17) corresponding to the spatial fractional diffusion equation in Equation (1) or its two-dimensional counterpart in Equation (16). In accordance with the theoretical analyses, FRHS can be employed to effectively solve the discretized almost-isotropic problems, whereas FRSHSS can only be applied to solve the discretized strongly anisotropic problems. Examples 1–3 belong to the first class of problems, so using the FRSHSS-GMRES and FRSHSS-BiCGSTAB methods to solve the corresponding discrete linear systems is not an appropriate choice.

5 | CONCLUDING REMARKS

A Toeplitz matrix can be uniquely decomposed into the sum of a Hermitian Toeplitz matrix and a skew-Hermitian Toeplitz matrix. Based on this representation, by making use of the arithmetic mean of the diffusion coefficients and the scalar shift of the Hermitian Toeplitz matrix, we have constructed a class of RHS iteration methods for solving the discrete linear system in Equation (2). With appropriate circulant-based approximation to the involved Hermitian Toeplitz matrix in the correspondingly induced RHS preconditioner, we have obtained the so-called FRHS preconditioner for the discrete linear system in Equation (2). Both theoretical analysis and numerical experiments have shown that the FRHS-preconditioned GMRES and BiCGSTAB iteration methods are economic and fast linear solvers for the discretized almost-isotropic spatial fractional diffusion equation in Equation (1). This approach can be straightforwardly applied to treat the higher-dimensional or time-dependent fractional diffusion equations, which also produces satisfactory computational results.

For anisotropic spatial fractional diffusion equations of the form in Equation (1), if one of the diffusion coefficients $\omega(x)$ and $\gamma(x)$ is assumed to be sufficiently larger than the other, for example, $\omega(x) \gg \gamma(x)$ (for all $x \in [0, 1]$), which can be understood in the sense of the finite-difference discretization of the equidistant step size h that either of the restrictions

$$\max_{1\le i\le n}\left\{\frac{\gamma_i}{\omega_i+\gamma_i}\right\} \ll \mathcal{O}(h^{\beta/2}) \quad \text{and} \quad \max_{1\le i\le n}\left\{\frac{\gamma_i}{|\omega_i-\gamma_i|}\right\} \ll \mathcal{O}(h^{\beta/2})$$

is further satisfied, the FRSHSS-preconditioned Krylov subspace methods, such as FRSHSS-GMRES and FRSHSS-BiCGSTAB proposed and analyzed in the work of Bai,[7] provide effective linear iterative solvers for the corresponding discrete linear system in Equation (2), and if, in addition, both $\omega(x)$ and $\gamma(x)$ are sufficiently smooth and almost linear, then CAI-GMRES(BiCGSTAB)[23] and TA-GMRES(BiCGSTAB)[24] should still be competitive methods for solving the discrete linear system in Equation (2). This conclusion is equally true for the two-dimensional spatial fractional diffusion equation in Equation (16) and the corresponding discrete linear system in Equation (17).

ACKNOWLEDGEMENTS

The authors are thankful to the referees for their constructive comments and valuable suggestions, which greatly improved the original manuscript of this paper. This paper was supported by the National Natural Science Foundation of China under Grants 11671393 and 11911530082. There are no conflicts of interest to this work.

ORCID

Zhong-Zhi Bai https://orcid.org/0000-0001-8353-3803

REFERENCES

1. Podlubny I. Fractional differential equations. San Diego, CA: Academic Press; 1999. (Mathematics in Science and Engineering; No. 198).
2. Samko SG, Kilbas AA, Marichev OI. Fractional integrals and derivatives: Theory and applications. Yverdon, Switzerland: Gordon and Breach Science Publishers; 1993.
3. Meerschaert MM, Tadjeran C. Finite difference approximations for fractional advection–dispersion flow equations. J Comput Appl Math. 2004;172:65–77.
4. Wang H, Wang K-X, Sircar T. A direct $\mathcal{O}(N \log^2 N)$ finite difference method for fractional diffusion equations. J Comput Phys. 2010;229:8095–8104.
5. Berman A, Plemmons RJ. Nonnegative matrices in the mathematical sciences. Revised reprint of the 1979 original. Philadelphia, PA: SIAM; 1994.
6. Varga RS. Matrix iterative analysis. Englewood Cliffs, NJ: Prentice-Hall; 1962.
7. Bai Z-Z. Respectively scaled HSS iteration methods for solving discretized spatial fractional diffusion equations. Numer Linear Algebra Appl. 2018;25:e2157.
8. Golub GH, Van Loan CF. Matrix computations. 3rd ed. Baltimore, MD: The Johns Hopkins University Press; 1996.
9. Bai Z-Z, Lu K-Y. On banded *M*-splitting iteration methods for solving discretized spatial fractional diffusion equations. BIT Numer Math. 2019;59:1–33.
10. Meerschaert MM, Scheffler H-P, Tadjeran C. Finite difference methods for two-dimensional fractional dispersion equation. J Comput Phys. 2006;211:249–261.
11. Meerschaert MM, Tadjeran C. Finite difference approximations for two-sided space-fractional partial differential equations. Appl Numer Math. 2006;56:80–90.
12. Axelsson O, Bai Z-Z, Qiu S-X. A class of nested iteration schemes for linear systems with a coefficient matrix with a dominant positive definite symmetric part. Numer Algorithms. 2004;35:351–372.
13. Golub GH, Overton ML. The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems. Numer Math. 1988;53:571–593.
14. Bai Z-Z, Yin J-F, Su Y-F. A shift-splitting preconditioner for non-Hermitian positive definite matrices. J Comput Math. 2006;24:539–552.
15. Bai Z-Z. Motivations and realizations of Krylov subspace methods for large sparse linear systems. J Comput Appl Math. 2015;283:71–78.
16. Ammar GS, Gragg WB. Superfast solution of real positive definite Toeplitz systems. SIAM J Matrix Anal Appl. 1988;9:61–76.
17. Bitmead RR, Anderson BDO. Asymptotically fast solution of Toeplitz and related systems of linear equations. Linear Algebra Appl. 1980;34:103–116.
18. Brent RP, Gustavson FG, Yun DYY. Fast solution of Toeplitz systems of equations and computation of Padé approximants. J Algorithms. 1980;1:259–295.
19. de Hoog FR. A new algorithm for solving Toeplitz systems of equations. Linear Algebra Appl. 1987;88–89:123–138.

20. Bai Z-Z, Lu K-Y, Pan J-Y. Diagonal and Toeplitz splitting iteration methods for diagonal-plus-Toeplitz linear systems from spatial fractional diffusion equations. Numer Linear Algebra Appl. 2017;24:e2093.
21. Strang G. A proposal for Toeplitz matrix calculations. Stud Appl Math. 1986;74:171–176.
22. Varga RS. Geršgorin and his circles. Berlin, Germany: Springer-Verlag; 2004.
23. Pan J-Y, Ke R-H, Ng MK, Sun H-W. Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations. SIAM J Sci Comput. 2014;36:A2698–A2719.
24. Donatelli M, Mazza M, Serra-Capizzano S. Spectral analysis and structure preserving preconditioners for fractional diffusion equations. J Comput Phys. 2016;307:262–279.
25. Pang H-K, Sun H-W. Multigrid method for fractional diffusion equations. J Comput Phys. 2012;231:693–703.
26. Lin X-L, Ng MK, Sun H-W. A multigrid method for linear systems arising from time-dependent two-dimensional space-fractional diffusion equations. J Comput Phys. 2017;336:69–86.
27. Zhao X, Hu X-Z, Cai W, Karniadakis GE. Adaptive finite element method for fractional differential equations using hierarchical matrices. Comput Methods Appl Mech Engrg. 2017;325:56–76.

How to cite this article: Bai Z-Z, Lu K-Y. On regularized Hermitian splitting iteration methods for solving discretized almost-isotropic spatial fractional diffusion equations. *Numer Linear Algebra Appl*. 2020;27:e2274. https://doi.org/10.1002/nla.2274