Fast Numerical Solution for Fractional Diffusion Equations by Exponential Quadrature Rule☆

Lu Zhang[a], Hai-Wei Sun[a,*], Hong-Kui Pang[b]

[a]*Department of Mathematics, University of Macau, Macao, China*
[b]*School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou, 221116, Jiangsu, China*

Abstract

After spatial discretization to the fractional diffusion equation by the shifted Grünwald formula, it leads to a system of ordinary differential equations, where the resulting coefficient matrix possesses the Toeplitz-like structure. An exponential quadrature rule is employed to solve such a system of ordinary differential equations. The convergence by the proposed method is theoretically studied. In practical computation, the product of a Toeplitz-like matrix exponential and a vector is calculated by the shift-invert Arnoldi method. Meanwhile, the coefficient matrix satisfies a condition that guarantees the fast approximation by the shift-invert Arnoldi method. Numerical results are given to demonstrate the efficiency of the proposed method.

Keywords: Fractional diffusion equation, Toeplitz-like structure, Exponential quadrature rule, Matrix exponential, Shift-invert Arnoldi, Preconditioned GMRES
2000 MSC: 65F10, 65L05, 65L20, 65T50, 26A33

1. Introduction

The fractional diffusion equation (FDE) is used in extensive applications, such as modeling chaotic dynamics of classical conservative systems [53], groundwater contaminant transport [2, 3], turbulent flow [5, 41] and applications in finance [39], image processing [1] and physics [42]. Unlike the second-order diffusion equation, there are very few cases of FDEs in which the closed-form analytical solutions are available. Therefore, many numerical methods have been developed for these problems [4, 10, 11, 13, 26, 28, 29, 30, 43, 44, 45, 48]. Since the fractional differential operators are nonlocal, however, the stabilities of numerical approximations become very sensitive [29, 30]. Moreover, numerical methods for FDEs

☆This research was supported by research grants MYRG206(Y3-L4)-FST11-SHW from University of Macau, 105/2012/A3 from FDCT of Macao, the National Natural Science Foundation of China under grant 11201192, the Natural Science Foundation of Jiangsu Province under grant BK2012577, and the Natural Science Foundation for Colleges and Universities in Jiangsu Province under grant 12KJB110004.

*Corresponding author
Email addresses: `yulu7517@126.com` (Lu Zhang), `HSun@umac.mo` (Hai-Wei Sun), `panghongkui@163.com` (Hong-Kui Pang)

tend to generate full coefficient matrices. These features introduce significant computational difficulties for the numerical methods for FDEs.

Meerschaert and Tadjeran in [29, 30] proposed the shifted Grünwald discretization to approximate the FDEs which has been proved to be unconditionally stable. Later, Wang, Wang, and Sircar in [52] showed that the resulting coefficient matrix by the Meerschaert-Tadjeran method possesses a Toeplitz-like structure. Therefore, the computational cost for such a matrix-vector multiplication can be carried out in $\mathcal{O}(n \log n)$ operations using a fast algorithm based on the fast Fourier transform (FFT), and the storage requirement is reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, where n is the number of spatial grid points in the discretization. As the resulting coefficient matrix is still ill-conditioned [34], many fast iterative methods have been proposed to speed up the convergence rate; see [24, 27, 32, 34, 38]. The complexity by those methods for solving the resulting system at each time step is of order $\mathcal{O}(n \log n)$. Nevertheless, since the temporal accuracies of those discretized methods are only first- or second-order, we need many time steps in the practical computation and hence the computational cost is too expensive.

In recent years, exponential integrators have been employed to various large-scale computations [17]. They constitute an interesting class of higher-order accurate and stable numerical methods for the time integration of stiff systems of differential equations. In this paper, we develop a fast and accurate numerical method for solving the FDE by the exponential integrator method. The main contributions of this work are as follows. First, we develop the exponential integrator for the FDE, which is a higher-order temporal accurate method. Second, we propose the shift-invert Arnoldi method for evaluating the matrix exponential in the exponential integrator. More precisely, an exponential quadrature rule is proposed to solve the system of ordinary differential equations (ODEs) obtained by spatially discretizing the FDE with the shifted Grünwald formula. Theoretically, we prove that the coefficient matrix satisfies the condition which leads to the convergence of order s in time if s non-confluent quadrature nodes are used. In computation, the product of a Toeplitz-like matrix (the coefficient matrix) exponential and a vector is involved in the exponential integrator method. Note that the norm of such a Toeplitz-like matrix is very large. Therefore, the shift-invert Arnoldi method [23, 33, 35] is exploited to approximate the Toeplitz-like matrix exponential. Furthermore, the coefficient matrix is proved to be sectorial and hence the convergence of the approximation by the shift-invert Arnoldi method is independent of the size of the matrix norm. With this advantage, the calculating for the Toeplitz-like matrix exponential can be done in $\mathcal{O}(n \log n)$ complexity [23]. Therefore, the computational cost by the exponential quadrature rule to compute the FDE at each time step is of order $\mathcal{O}(n \log n)$. Due to the higher-order temporal accurate discretization by the exponential quadrature rule, the number of time steps is much less than that of other lower-order temporal accurate schemes. Thus, the total computational cost by the proposed method is much cheaper than that by other lower-order methods.

The rest of this paper is organized as follows. In Section 2, we discretize the FDE spatially to a system of ODEs. In Section 3, we propose an exponential quadrature rule to solve the resulting system of ODEs. The shift-invert Arnoldi method is employed to approximate the

Toeplitz-like matrix exponential involved in the exponential quadrature rule in Section 4. In Section 5, we report the numerical results to demonstrate the efficiency of the proposed method. At last, the concluding remarks are given in Section 6.

2. Spatial discretization of FDEs

Consider an initial-boundary value problem of the FDE [30]:

$$\begin{aligned}
&\frac{\partial u(x,t)}{\partial t} - d_+(x)\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} - d_-(x)\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} = f(x,t),\\
&x \in (x_L, x_R), \quad t \in (0,T],\\
&u(x_L,t) = u(x_R,t) = 0, \quad t \in [0,T],\\
&u(x,0) = u^0(x), \quad x \in [x_L, x_R],
\end{aligned} \tag{1}$$

where $1 < \alpha < 2$, $f(x,t)$ is the source term, and $d_\pm(x) \geq 0$. Here the left-sided and the right-sided fractional derivatives $\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha}$ and $\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha}$ are defined in the Grünwald-Letnikov form [36]:

$$\frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} = \lim_{\varepsilon \to 0^+} \frac{1}{\varepsilon^\alpha} \sum_{k=0}^{\lfloor (x-x_L)/\varepsilon \rfloor} g_k^{(\alpha)} u(x - k\varepsilon, t),$$

$$\frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} = \lim_{\varepsilon \to 0^+} \frac{1}{\varepsilon^\alpha} \sum_{k=0}^{\lfloor (x_R-x)/\varepsilon \rfloor} g_k^{(\alpha)} u(x + k\varepsilon, t),$$

where $\lfloor x \rfloor$ denotes the floor of x, $g_0^{(\alpha)} = 1$ and

$$g_k^{(\alpha)} = \frac{(-1)^k}{k!}\alpha(\alpha-1)\cdots(\alpha-k+1). \tag{2}$$

The existence and uniqueness of the weak solution to (1) can be found in [25]. We note that $g_k^{(\alpha)}$ satisfy the following proposition.

Proposition 1. (see [29, 30, 34, 52]) *Let $1 < \alpha < 2$ and $g_k^{(\alpha)}$ be defined in* (2)*. Then we have*

$$g_0^{(\alpha)} = 1, \quad g_1^{(\alpha)} = -\alpha < 0, \quad g_2^{(\alpha)} > g_3^{(\alpha)} > \cdots > 0,$$

$$\sum_{k=0}^{\infty} g_k^{(\alpha)} = 0, \quad \sum_{k=0}^{i} g_k^{(\alpha)} < 0 \quad \text{for} \quad i \geq 1.$$

Let n be positive integer and $h = (x_R - x_L)/n$ be the size of spatial grid. We define a spatial partition $x_i = x_L + ih$ for $i = 0, 1, \ldots, n$. Let $u_i(t) = u(x_i, t)$, $d_{\pm,i} = d_\pm(x_i)$, and $f_i(t) = f(x_i, t)$. In [29, 30], Meerschaert and Tadjeran proposed the following shifted Grünwald approximations,

$$\begin{aligned} \frac{\partial^\alpha u(x_i,t)}{\partial_+ x^\alpha} &= \frac{1}{h^\alpha}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}(t) + \mathcal{O}(h), \\ \frac{\partial^\alpha u(x_i,t)}{\partial_- x^\alpha} &= \frac{1}{h^\alpha}\sum_{k=0}^{n-i+1} g_k^{(\alpha)} u_{i+k-1}(t) + \mathcal{O}(h), \end{aligned} \tag{3}$$

where $g_k^{(\alpha)}$ are defined in (2). Using (3), the FDE (1) can be spatially semi-discretized as

$$\frac{\mathrm{d}u_i(t)}{\mathrm{d}t} - \frac{d_{+,i}}{h^\alpha}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}(t) - \frac{d_{-,i}}{h^\alpha}\sum_{k=0}^{n-i+1} g_k^{(\alpha)} u_{i+k-1}(t) = f_i(t), \quad u_i(0) = u^0(x_i), \tag{4}$$

where $1 \le i \le n-1$. Let $\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_{n-1}(t)]^\intercal$, $\mathbf{f}(t) = [f_1(t), f_2(t), \ldots, f_{n-1}(t)]^\intercal$, and $\mathbf{u}_0 = [u^0(x_1), u^0(x_2), \ldots, u^0(x_{n-1})]^\intercal$. The above semi-discretized formulation (4) can be further expressed in the following matrix form

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} + A_h\mathbf{u}(t) = \mathbf{f}(t), \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{5}$$

where

$$A_h = D_{+,h}G_\alpha + D_{-,h}G_\alpha^\intercal, \tag{6}$$

with $D_{\pm,h} = -(1/h^\alpha)\cdot \mathrm{diag}(d_{\pm,1}, \ldots, d_{\pm,n-1})$ and

$$G_\alpha = \begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0 \\ \vdots & g_2^{(\alpha)} & g_1^{(\alpha)} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{n-2}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_{n-1}^{(\alpha)} & g_{n-2}^{(\alpha)} & \cdots & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}. \tag{7}$$

It is obvious that G_α is Toeplitz and A_h in (6) possesses the Toeplitz-like structure [52]. Therefore, the matrix-vector multiplication for A_h can be obtained in $\mathcal{O}(n\log n)$ operations by the FFT; see [34, 52] for details.

The following lemma gives an important property of A_h by Proposition 1.

Lemma 1. *If $d_+(x) + d_-(x) > 0$, then the matrix A_h in* (6) *is strictly diagonally dominant.*

Proof: From Proposition 1, we have

$$\begin{aligned}
[A_h]_{ii} - \sum_{j=1, j\neq i}^{n} |[A_h]_{ij}| &= -\frac{1}{h^\alpha}\left[(d_{+,i}+d_{-,i})g_1^{(\alpha)} + d_{+,i}\sum_{j=0,j\neq 1}^{i} g_j^{(\alpha)} + d_{-,i}\sum_{j=0,j\neq 1}^{n-i} g_j^{(\alpha)}\right] \\
&= \frac{1}{h^\alpha}\left[d_{+,i}\sum_{j=i+1}^{\infty} g_j^{(\alpha)} + d_{-,i}\sum_{j=n-i+1}^{\infty} g_j^{(\alpha)}\right] \\
&\geq \frac{d_{+,i}+d_{-,i}}{h^\alpha}\sum_{j=n+1}^{\infty} g_j^{(\alpha)} \\
&> 0.
\end{aligned}$$

□

3. Exponential quadrature

In this section, the exponential quadrature rule [16, 17] is employed to solve the semi-discretized system (5).

3.1. Exponential quadrature rule

First, we consider the general linear system of ODEs,

$$\frac{\mathrm{d}\mathbf{u}(t)}{\mathrm{d}t} + A\mathbf{u}(t) = \mathbf{f}(t), \quad \mathbf{u}(0) = \mathbf{u}_0, \tag{8}$$

with a time-invariant matrix A. The solution of (8) at time

$$t_{i+1} = t_i + \Delta t, \quad t_0 = 0, \quad i = 0, 1, \ldots$$

is given by the variation-of-constants formula [16, 17]

$$\mathbf{u}(t_{i+1}) = \exp(-\Delta t A)\mathbf{u}(t_i) + \int_0^{\Delta t} \exp\big(-(\Delta t - \tau)A\big)\mathbf{f}(t_i+\tau)\mathrm{d}\tau. \tag{9}$$

If the function $\mathbf{f}$ within the integral is approximated by its interpolation polynomial in certain non-confluent quadrature nodes $c_1, \ldots, c_s$, the exponential quadrature rule for (9) is obtained

$$\mathbf{u}_{i+1} = \exp(-\Delta t A)\mathbf{u}_i + \Delta t\sum_{j=1}^{s} b_j(-\Delta t A)\mathbf{f}(t_i + c_j\Delta t), \tag{10}$$

where the weights

$$b_j(-\Delta t A) = \int_0^1 \exp\big(-\Delta t(1-\theta)A\big)\ell_j(\theta)\mathrm{d}\theta, \tag{11}$$

in which $\ell_j(\theta)$ are the Lagrange interpolation polynomials

$$\ell_j(\theta) = \prod_{i=1, i\neq j}^{s} \frac{\theta - c_i}{c_j - c_i}, \quad j = 1, \ldots, s. \tag{12}$$

In order to carry out the exponential quadrature rule, we assume the coefficient matrix A in (8) satisfies the following assumption.
Assumption A (see [17]) There exist constants C and ω such that

$$\| \exp(-tA)\| \leq C \exp(\omega t), \quad t \geq 0, \tag{13}$$

with a matrix norm $\| \cdot \|$.
The following theorem gives the convergence result for the exponential quadrature scheme.

Theorem 1. (see [17, Theorem 2.7 & Corollary 2.8]) *Let* (13) *be fulfilled and* $\mathbf{f}^{(s)} \in L^1(0, T)$. *Then the exponential quadrature rule* (10) *is convergent with order s. More precisely, the error bound*

$$\|\mathbf{u}_i - \mathbf{u}(t_i)\| \leq C_T (\Delta t)^s \int_0^{t_i} \|\mathbf{f}^{(s)}(\tau)\| \mathrm{d}\tau$$

holds, uniformly on $0 \leq t_i \leq T$, with a constant C_T that depends on T, but is independent of Δt.

Therefore, we only need to verify that A_h in (6) satisfies the condition (13) of Assumption A to obtain the desired convergence result. To this aim, we need the following lemma.

Lemma 2. (see [37]) *Let A be a diagonally dominant matrix with $[A]_{ii} \geq 0$, then*

$$\| \exp(-tA)\|_\infty \leq 1, \quad t \geq 0. \tag{14}$$

By Lemma 1, A_h is diagonally dominant with $[A_h]_{ii} = (d_{+,i} + d_{-,i})\alpha/h^\alpha > 0$, providing that $d_+(x) + d_-(x) > 0$. Furthermore, the condition (14) is a special case of (13) in which $C = 1$ and $\omega = 0$. By Theorem 1, we immediately have the following corollary.

Corollary 1. *Let $d_+(x) + d_-(x) > 0$ and $\mathbf{f}^{(s)} \in L^1(0, T)$. Then exponential quadrature rule* (10) *for solving* (5) *is convergent with order s; i.e.,*

$$\|\mathbf{u}_i - \mathbf{u}(t_i)\|_\infty \leq C_T (\Delta t)^s \int_0^{t_i} \|\mathbf{f}^{(s)}(\tau)\|_\infty \mathrm{d}\tau$$

holds, uniformly on $0 \leq t_i \leq T$, with a constant C_T that depends on T, but is independent of Δt.

3.2. Implementation

Now we consider the implementation of the exponential quadrature rule (10). Obviously, the weights $b_j(z)$ in (11) are linear combinations of the entire functions [17]

$$\varphi_k(z) = \int_0^1 \exp\big((1-\theta)z\big)\frac{\theta^{k-1}}{(k-1)!}\mathrm{d}\theta, \quad k \geq 1.$$

Those functions satisfy $\varphi_k(0) = 1/k!$ and the recurrence relation

$$\varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}, \quad \varphi_0(z) = \exp(z).$$

In this paper, we choose $s = 4$ in (10) with $c_1 = 0$, $c_2 = 1/3$, $c_3 = 2/3$, and $c_4 = 1$. Once the quadrature nodes c_j are known, we can determine the Lagrange interpolation polynomials $\ell_j(\theta)$ in (12) for $j = 1, \ldots, 4$. Thus, according to the formulations of weights $b_j(z)$ and entire functions $\varphi_k(z)$, we obtain the following weights by simple calculation,

$$\begin{aligned}
b_1(z) &= \varphi_1(z) - (11/2)\varphi_2(z) + 18\varphi_3(z) - 27\varphi_4(z),\\
b_2(z) &= 9\varphi_2(z) - 45\varphi_3(z) + 81\varphi_4(z),\\
b_3(z) &= -(9/2)\varphi_2(z) + 36\varphi_3(z) - 81\varphi_4(z),\\
b_4(z) &= \varphi_2(z) - 9\varphi_3(z) + 27\varphi_4(z).
\end{aligned}$$

Taking the above weights into (10) and denoting

$$\begin{aligned}
&\mathbf{a}_1 = \mathbf{f}(t_i),\\
&\mathbf{a}_2 = -(11/2)\mathbf{f}(t_i) + 9\mathbf{f}(t_{i+1/3}) - (9/2)\mathbf{f}(t_{i+2/3}) + \mathbf{f}(t_{i+1}),\\
&\mathbf{a}_3 = 18\mathbf{f}(t_i) - 45\mathbf{f}(t_{i+1/3}) + 36\mathbf{f}(t_{i+2/3}) - 9\mathbf{f}(t_{i+1}),\\
&\mathbf{a}_4 = -27\mathbf{f}(t_i) + 81\mathbf{f}(t_{i+1/3}) - 81\mathbf{f}(t_{i+2/3}) + 27\mathbf{f}(t_{i+1}),
\end{aligned} \tag{15}$$

with $t_{i+1/3} \equiv t_i + (1/3)\Delta t$ and $t_{i+2/3} \equiv t_i + (2/3)\Delta t$, the exponential quadrature rule (10) (by setting $A = A_h$) can be written as

$$\mathbf{u}_{i+1} = \frac{1}{6}A_h^{-1}\mathbf{a}_4 + \frac{1}{2}A_h^{-1}\mathbf{c}_1 + A_h^{-1}\mathbf{c}_2 + A_h^{-1}\mathbf{c}_3 + \exp(-\Delta t A_h)(\mathbf{u}_i - A_h^{-1}\mathbf{c}_3), \tag{16}$$

where

$$\mathbf{c}_1 = -\Delta t^{-1}A_h^{-1}\mathbf{a}_4 + \mathbf{a}_3, \quad \mathbf{c}_2 = -\Delta t^{-1}A_h^{-1}\mathbf{c}_1 + \mathbf{a}_2, \quad \mathbf{c}_3 = -\Delta t^{-1}A_h^{-1}\mathbf{c}_2 + \mathbf{a}_1.$$

The implementation formula (16) can be carried out as the following algorithm.

Algorithm 1: Exponential quadrature rule with order $s = 4$

1.	Input: $\mathbf{u}_i$, $\mathbf{f}$, Δt, and A_h
2.	Compute $\mathbf{a}_j$, $j = 1, \ldots, 4$, using (15)
3.	Compute $\mathbf{v}_1 = A_h^{-1}\mathbf{a}_4$
4.	Compute $\mathbf{c}_j = -\frac{1}{\Delta t}\mathbf{v}_j + \mathbf{a}_{4-j}$ and $\mathbf{v}_{j+1} = A_h^{-1}\mathbf{c}_j$ for $j = 1, 2, 3$
5.	Compute $\mathbf{c}_4 = \exp\left(-\Delta t A_h\right)(\mathbf{u}_i - \mathbf{v}_4)$
6.	Compute $\mathbf{u}_{i+1} = \frac{1}{6}\mathbf{v}_1 + \frac{1}{2}\mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_4 + \mathbf{c}_4$

The main computational workloads in Algorithm 1 are at steps 3–5, where we need to solve four linear systems for obtaining $\mathbf{v}_j$ and one matrix exponential for getting $\mathbf{c}_4$, respectively.

For steps 3 and 4 in Algorithm 1, note that A_h in (6) is diagonally dominant Toeplitz-like. There are many fast algorithms for solving such a linear system in $\mathcal{O}(n \log n)$ operations; see [24, 27, 32, 34, 38] for more discussions. In this paper, we employ the preconditioned GMRES method [40] with the generalized Strang's circulant preconditioner proposed in [24] to solve the linear system iteratively. Strang's circulant matrix $S(B) = [s_{j-k}]_{0 \le j,k<n}$ for a real Toeplitz matrix $B = [b_{j-k}]_{0 \le j,k<n}$ is obtained by copying the central entries of B and bringing them around to complete the circulant requirement [6, 7]. More precisely, the first column of $S(B)$ are given by

$$s_j = \begin{cases} b_j, & 0 \le j < n/2, \\ 0, & j = n/2 \text{ if } n \text{ is even}, \\ b_{j-n}, & n/2 < j < n, \\ s_{j+n}, & 0 < -j < n. \end{cases}$$

Recall that $A_h = D_{+,h} G_\alpha + D_{-,h} G_\alpha^\intercal$ in (6), let

$$d_\pm = -\frac{1}{n-1} \sum_{i=1}^{n-1} \frac{d_{\pm,i}}{h^\alpha}.$$

As in [24], the generalized Strang circulant preconditioner for A_h is defined by

$$\mathcal{S} = d_+ S(G_\alpha) + d_- S(G_\alpha^\intercal). \tag{17}$$

At step 5 in Algorithm 1, we need to calculate the product of a Toeplitz-like matrix exponential and a vector. In next section, we will exploit the shift-invert Arnoldi method [23, 33, 35] to approximate such a matrix exponential that only needs $\mathcal{O}(n \log n)$ operations.

4. Toeplitz-like matrix exponential

4.1. Shift-invert Arnoldi method

Recently, the Krylov subspace methods have been applied to approximate the matrix exponential [22, 23, 31, 33], especially when the matrix is with a very large size and special structure (sparse or Toeplitz). The main idea is to approximately project the exponential of a large matrix onto a small Krylov subspace. This can be achieved by the Arnoldi process for nonsymmetric matrices or Lanczos process for symmetric matrices. Nevertheless, the convergence of such a Krylov subspace method is often very slow. In [31], a shift-invert technique is proposed to speed up the Arnoldi process.

We consider the product of matrix exponential with a vector, i.e.,

$$\mathbf{w}(t) = \exp(-tA)\mathbf{v}$$

with an $n \times n$ matrix A, a scalar t, and a vector $\mathbf{v}$. Denote I_n as the $n \times n$ identity matrix and $\mathbf{e}_j$ as jth column of I_n. Let $\gamma > 0$ be the shift parameter. The shift-invert Arnoldi method for approximating $\exp(-tA)\mathbf{v}$ is described as follows [22, 23, 31].

Algorithm 2: Shift-invert Arnoldi method for matrix exponential

1. Initialize: Compute $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|_2$
2. Iterate: Do $j = 1, \dots, m$
   (a) Compute $\hat{\mathbf{v}} := (I_n + \gamma A)^{-1}\mathbf{v}_j$
   (b) Do $k = 1, \dots, j$
      i. Compute $h_{k,j} := \hat{\mathbf{v}}^\intercal \mathbf{v}_k$
      ii. Compute $\hat{\mathbf{v}} := \hat{\mathbf{v}} - h_{k,j}\mathbf{v}_k$
   (c) Compute $h_{j+1,j} := \|\hat{\mathbf{v}}\|_2$ and $\mathbf{v}_{j+1} := \hat{\mathbf{v}}/h_{j+1,j}$
3. Approximate: (a) Set $V_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$ and upper Hessenberg matrix $H_m = [h_{ij}]_{m\times m}$
   (b) Compute $\mathbf{w}_m(t) = \beta V_m \exp\big(-\tau(H_m^{-1} - I_m)\big)\mathbf{e}_1$
   where $\tau = t/\gamma$ and $\beta = \|\mathbf{v}\|_2$

In Algorithm 2, $\mathbf{w}_m(t)$ is the approximation to the matrix exponential $\mathbf{w}(t) = \exp(-tA)\mathbf{v}$ that a small $m\,(\ll n\,)$ is expected to guarantee the cheaper computational cost. In the following, we will study a certain condition which makes m independent of $\|tA\|_2$.

We first give the definition of numerical range that will come into use afterward. The numerical range of a matrix A is defined as

$$\mathcal{W}(A) \equiv \{\mathbf{v}^*A\mathbf{v} : \mathbf{v} \in \mathbb{C}^n, \mathbf{v}^*\mathbf{v} = 1\},$$

which is a compact and convex subset of $\mathbb{C}$ [18, p.8]. Let $\Sigma_{\xi,\rho}$ be the following set:

$$\Sigma_{\xi,\rho} \equiv \left\{ z \in \mathbb{C} : |\arg(z-\xi)| < \rho, \xi \geq 0, 0 < \rho < \frac{\pi}{2} \right\};$$

i.e., $\Sigma_{\xi,\rho}$ is an unbounded sector in the right-half plane with semiangle $\rho < \pi/2$ and vertex lying on the nonnegative real axis. If $\mathcal{W}(A) \subseteq \overline{\Sigma_{\xi,\rho}}$ (the closure of $\Sigma_{\xi,\rho}$), then A is called a sectorial operator [22, 23, 31].

In [31], a sufficient condition is provided for error estimate in terms of $\mathcal{W}(A)$. According to [22, 31] and the discussions therein, we can summarize the theoretical result in [31] by the following lemma.

Lemma 3. (see [31, Proposition 2.1 & Proposition 3.2]) *For Algorithm 2, let A be sectorial in which $\mathcal{W}(A) \subseteq \overline{\Sigma_{0,\rho}}$ with $\rho < \pi/2$. Then the following error bound holds:*

$$\|\mathbf{w}(t) - \mathbf{w}_m(t)\|_2 \leq \left[\pi \sin\left(\frac{\pi}{4} - \frac{\rho}{2}\right)\right]^{-1} \Phi_m,$$

where $\Phi_m \to 0$ as $m \to \infty$ and its convergence is independent of $\|A\|_2$.

Lemma 3 states that $\|A\|_2$ is not involved in the error bound of the shift-invert Arnoldi approximation $\mathbf{w}_m(t)$ if A is sectorial. Therefore, the iteration number m is not disturbed by $\|A\|_2$.

Now we consider how to calculate the Toeplitz-like matrix exponential in Algorithm 1 by the shift-invert Arnoldi method. Let $A = A_h$ and $t = \Delta t$ in Algorithm 2. Then the main computational cost should be at step 2(a) where m linear systems are inverted. Note that A_h is diagonally dominant Toeplitz-like. So is $I_{n-1} + \gamma A_h$. Therefore, as in [24] and Section 3.2, the preconditioned GMRES method with the circulant preconditioner $I_{n-1}+\gamma\mathcal{S}$, where $\mathcal{S}$ is defined in (17), can be employed to solve those linear systems. Thus, the total computational cost for calculating $\mathbf{w}_m(t)$ in Algorithm 2 is of $\mathcal{O}(mn\log n)$.

The remaining issue is to prove that the Toeplitz-like matrix A_h is sectorial and hence m is independent of $\|A_h\|_2$ by Lemma 3.

4.2. Error estimate for Toeplitz-like matrix exponential

We first introduce the definition of the generating function for a Toeplitz matrix. If the entries of a Toeplitz matrix $A = [a_{k-j}]_{n\times n}$ are the Fourier coefficients of a 2π periodic function $p(\theta)$ as follows,

$$a_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} p(\theta)\exp(-\mathbf{i}j\theta)\mathrm{d}\theta, \quad \mathbf{i} \equiv \sqrt{-1}, \quad 1-n \le j \le n-1,$$

then $p(\theta)$ is called the *generating function* of A. We denote the real and imaginary parts of p by $\mathrm{Re}(p)$ and $\mathrm{Im}(p)$ respectively. The following theorem shows a sufficient condition by the generating function to ensure that a Toeplitz matrix is sectorial.

Theorem 2. (see [23, Theorem 2.3 and Lemma 4.1]) *Let p be the generating function of a Toeplitz matrix A. If $\mathrm{Re}(p)$ is even, $\mathrm{Im}(p)$ is odd, and p satisfies the following assumptions:*

$$\mathrm{Re}(p) \ge 0 \quad \text{and} \quad |\mathrm{Im}(p)| \le C_p \mathrm{Re}(p), \quad \forall\theta \in [-\pi,\pi], \tag{18}$$

where $C_p > 0$ is a constant, then A is sectorial. Moreover,

$$\mathcal{W}(A) \subseteq \overline{\Sigma_{0,\rho}},$$

where $\rho = \arctan(C_p) < \pi/2$ and

$$\overline{\Sigma_{0,\rho}} = \left\{ z\in\mathbb{C}: \ |\arg z| \le \rho, \ 0<\rho<\frac{\pi}{2} \right\}.$$

Next, we will show that the generating function of A_h satisfies the assumptions (18) in Theorem 2 when $d_\pm(x)$ in (1) are constants. Some properties of the generating function of G_α in (7) are introduced in advance before we obtain the main result.

Lemma 4. *Let $p_\alpha(\theta)$ be the generating function of G_α. Then*

$$p_\alpha(\theta) = \begin{cases} |2\sin(\theta/2)|^\alpha \left[\cos\left(\alpha(\theta-\pi)/2-\theta\right) + \mathbf{i}\sin\left(\alpha(\theta-\pi)/2-\theta\right)\right], & \theta\in[0,\pi], \\ |2\sin(\theta/2)|^\alpha \left[\cos\left(\alpha(\theta+\pi)/2-\theta\right) + \mathbf{i}\sin\left(\alpha(\theta+\pi)/2-\theta\right)\right], & \theta\in[-\pi,0). \end{cases} \tag{19}$$

Proof: It is obvious that the generating functions of G_α can be written as the following series form

$$p_\alpha(\theta) = \sum_{k=0}^{\infty} g_k^{(\alpha)} \exp\Big(\mathbf{i}(k-1)\theta\Big).$$

On the other hand, $g_k^{(\alpha)}$ in (2) are coefficients of the power series of the function $(1-z)^\alpha$,

$$(1-z)^\alpha = \sum_{k=0}^{\infty} g_k^{(\alpha)} z^k$$

for all $|z| \le 1$. Using the following relation

$$\exp(\mathbf{i}\theta) - \exp(\mathbf{i}\phi) = 2\mathbf{i}\sin\left(\frac{\theta-\phi}{2}\right)\exp\left(\frac{\mathbf{i}(\theta+\phi)}{2}\right),$$

we obtain

$$\begin{aligned} p_\alpha(\theta) &= \sum_{k=0}^{\infty} g_k^{(\alpha)} \exp\Big(\mathbf{i}(k-1)\theta\Big) \\ &= \exp(-\mathbf{i}\theta)\sum_{k=0}^{\infty} g_k^{(\alpha)} \exp(\mathbf{i}k\theta) \\ &= \exp(-\mathbf{i}\theta)[1-\exp(\mathbf{i}\theta)]^\alpha \\ &= \exp(-\mathbf{i}\theta)[-2\mathbf{i}\sin(\theta/2)\exp(\mathbf{i}\theta/2)]^\alpha, \end{aligned}$$

which can be reformulated into (19).

$\square$

We further consider the even and odd features of p_α in the following lemma.

Lemma 5. $\mathrm{Re}(p_\alpha(\theta)) \le 0$ *is even and* $\mathrm{Im}(p_\alpha(\theta))$ *is odd on* $[-\pi, \pi]$.

Proof: By the formula (19) in Lemma 4, we have

$$\mathrm{Re}(p_\alpha(\theta)) = \begin{cases} |2\sin(\theta/2)|^\alpha \cos\left(\alpha(\theta-\pi)/2-\theta\right), & \theta \in [0,\pi], \\ |2\sin(\theta/2)|^\alpha \cos\left(\alpha(\theta+\pi)/2-\theta\right), & \theta \in [-\pi,0), \end{cases}$$

and

$$\mathrm{Im}(p_\alpha(\theta)) = \begin{cases} |2\sin(\theta/2)|^\alpha \sin\left(\alpha(\theta-\pi)/2-\theta\right), & \theta \in [0,\pi], \\ |2\sin(\theta/2)|^\alpha \sin\left(\alpha(\theta+\pi)/2-\theta\right), & \theta \in [-\pi,0). \end{cases}$$

For $\theta \in [-\pi, 0)$, we have $-\theta \in (0, \pi]$ and

$$\alpha(\theta+\pi)/2-\theta = -\Big(\alpha[(-\theta)-\pi]/2-(-\theta)\Big).$$

Therefore,

$$\cos\Big(\alpha(\theta+\pi)/2-\theta\Big) = \cos\Big(\alpha[(-\theta)-\pi]/2-(-\theta)\Big)$$

and

$$\sin\Big(\alpha(\theta+\pi)/2-\theta\Big) = -\sin\Big(\alpha[(-\theta)-\pi]/2-(-\theta)\Big).$$

Thus, they imply that $\mathrm{Re}(p_\alpha(\theta)) = \mathrm{Re}(p_\alpha(-\theta))$ is even, and $\mathrm{Im}(p_\alpha(\theta)) = -\mathrm{Im}(p_\alpha(-\theta))$ is odd.

For $\theta \in [0, \pi]$, note that $\alpha \in (1, 2)$, we have

$$\alpha(\theta - \pi)/2 - \theta \in [-\pi, -\pi/2]. \tag{20}$$

Therefore,

$$\mathrm{Re}(p_\alpha(\theta)) \leq 0, \quad \theta \in [0, \pi].$$

The above inequality also holds for $\theta \in [-\pi, 0)$ since $\mathrm{Re}(p_\alpha(\theta))$ is even. We then complete the proof. □

Lemma 6. *p_α satisfies the following condition:*

$$\left|\frac{\mathrm{Im}(p_\alpha)}{\mathrm{Re}(p_\alpha)}\right| \leq \tan\big((1-\alpha/2)\pi\big), \quad \forall \theta \in [-\pi, \pi].$$

Proof: For $\theta \in [0, \pi]$, noting that $\alpha(\theta - \pi)/2 - \theta \in [-\pi, -\pi/2]$ by (20), we have

$$\left|\frac{\mathrm{Im}(p_\alpha)}{\mathrm{Re}(p_\alpha)}\right| = |\tan(\alpha(\theta - \pi)/2 - \theta)| \leq \tan\big((1-\alpha/2)\pi\big).$$

By Lemma 5, the above inequality also holds for $\theta \in [-\pi, 0)$. The proof is completed. □

Finally, we give the main result about the matrix A_h in this subsection.

Theorem 3. *Assume that $d_\pm(x) = d_\pm$ are constants with $d_+ + d_- > 0$. Then A_h is sectorial and*

$$\mathcal{W}(A_h) \subseteq \overline{\Sigma_{0,\rho}},$$

where $\rho = (1 - \alpha/2)\pi < \pi/2$ and

$$\overline{\Sigma_{0,\rho}} = \left\{ z \in \mathbb{C} : \ |\arg z| \leq \rho, \ 0 < \rho < \frac{\pi}{2} \right\}.$$

Proof: According to the assumption, A_h can be written as

$$A_h = -h^{-\alpha}(d_+ G_\alpha + d_- G_\alpha^\intercal).$$

With the fact that the generating function of $G_\alpha^\intercal$ is $\overline{p_\alpha(\theta)}$, the generating function of A_h can be written as $p_h(\theta) = -h^{-\alpha}[d_+ p_\alpha(\theta) + d_- \overline{p_\alpha(\theta)}]$. Therefore, by Lemma 5,

$$\mathrm{Re}(p_h) = -h^{-\alpha}(d_+ + d_-)\mathrm{Re}(p_\alpha) \geq 0$$

is even and $\mathrm{Im}(p_h) = h^{-\alpha}(d_- - d_+)\mathrm{Im}(p_\alpha)$ is odd. Furthermore, using Lemma 6, noting that $|d_- - d_+| \leq d_+ + d_-$, we have

$$\left|\frac{\mathrm{Im}(p_h)}{\mathrm{Re}(p_h)}\right| = \left|\frac{(d_- - d_+)\mathrm{Im}(p_\alpha)}{(d_+ + d_-)\mathrm{Re}(p_\alpha)}\right| \leq \tan\big((1-\alpha/2)\pi\big),$$

which implies that

$$|\mathrm{Im}(p_h)| \leq \tan\big((1-\alpha/2)\pi\big)\mathrm{Re}(p_h).$$

According to Theorem 2, it concludes that A_h is a sectorial operator and

$$\mathcal{W}(A_h) \subseteq \overline{\Sigma_{0,\rho}},$$

where $\rho = (1-\alpha/2)\pi < \pi/2$ and

$$\overline{\Sigma_{0,\rho}} = \left\{ z \in \mathbb{C} : \ |\arg z| \leq \rho, \ 0 < \rho < \frac{\pi}{2} \right\}.$$

□

By Theorem 3 and Lemma 3, it is resulted that the error bound of the shift-invert Arnoldi approximation is independent of $\|A_h\|_2$ which is related to n. This implies that the number of iteration m in Algorithm 2 is independent of mesh size h or the number of spatial nodes n. Therefore, the Toeplitz-like matrix exponential can be done in $\mathcal{O}(mn\log n)$ with small m.

5. Numerical experiments

In the following numerical tests, we employ the exponential quadrature rule to solve the FDE numerically. As comparisons, we also carry out the implicit finite difference scheme [24, 34] to solve the FDE. In this scheme, the FDE is discretized in space by the shifted Grünwald formula (3) and in time using an implicit Euler method. Then, one has to solve a Toeplitz-like linear systems in each time step of the implicit finite difference scheme. For the linear systems appearing in the implementation of above both methods, the preconditioned GMRES method with the generalized Strang circulant preconditioner (17) is utilized to solve them. Moreover, the stopping criterion for solving those linear systems is

$$\frac{\|\mathbf{r}^{(k)}\|_2}{\|\mathbf{r}^{(0)}\|_2} < 10^{-7},$$

where $\mathbf{r}^{(k)}$ is the residual vector of linear systems after k iterations.

For all tables, "M" denotes the number of time steps for the implicit finite difference scheme, "Δt" denotes the time-step size for the exponential quadrature rule, "Error" denotes the error between the true solution and the approximation under the infinity norm at the last time step, and "CPU" denotes the total CPU time in seconds for solving the whole discretized systems. For the shift parameter γ in Algorithm 2, we choose $\gamma = \Delta t/10$ which is suggested in [22, 23]. The iteration number of the shift-invert Arnoldi process (i.e., Algorithm 2) is set to $m = 7$ in all examples. We also denote "EQR" as the exponential quadrature rule (16) and "IFD" as the implicit finite difference scheme proposed in [24], respectively. All numerical experiments are run in MATLAB 7.11(R2010a) on a PC with Intel(R) Core(TM)i7-2600 3.40 GHz processor and 16.0 GB RAM.

Example 1. (see [47, 54]) In this example, we consider the FDE (1) whose data are given as follows: $\alpha = 1.5$, $(x_L, x_R) = (0, 1)$, $T = 1$, $d_+(x) = d_-(x) = 1$, and the source term

$$f(x,t) = -\exp(-t)\Big\{x^3(1-x)^3 + \frac{\Gamma(4)}{\Gamma(4-\alpha)}\left[x^{3-\alpha} + (1-x)^{3-\alpha}\right]$$
$$-\frac{3\Gamma(5)}{\Gamma(5-\alpha)}\left[x^{4-\alpha} + (1-x)^{4-\alpha}\right] + \frac{3\Gamma(6)}{\Gamma(6-\alpha)}\left[x^{5-\alpha} + (1-x)^{5-\alpha}\right]$$
$$-\frac{\Gamma(7)}{\Gamma(7-\alpha)}\left[x^{6-\alpha} + (1-x)^{6-\alpha}\right]\Big\}.$$

The initial condition is chosen as

$$u(x,0) = x^3(1-x)^3.$$

The true solution to the corresponding FDE is given by

$$u(x,t) = \exp(-t)x^3(1-x)^3.$$

Table 1: Comparisons for Example 1 by the EQR and the IFD, respectively.

	EQR			IFD		
n	Δt	CPU	Error	M	CPU	Error
2^6	T	0.02	1.1244e-4	2^6	0.06	1.0800e-4
2^7		0.03	5.6542e-5	2^7	0.14	5.5273e-5
2^8		0.04	2.7604e-5	2^8	0.32	2.7948e-5
2^9	$T/2$	0.07	1.4777e-5	2^9	1.23	1.4050e-5
2^{10}		0.09	7.3622e-6	2^{10}	3.64	7.0437e-6
2^{11}		0.24	3.6404e-6	2^{11}	20.10	3.5263e-6
2^{12}		0.29	1.7760e-6	2^{12}	35.49	1.7642e-6
2^{13}		1.03	8.4287e-7	2^{13}	282.10	8.8233e-7
2^{14}		2.73	3.7611e-7	2^{14}	1558.24	4.4116e-7
2^{15}		4.09	1.4268e-7	2^{15}	3666.23	2.2052e-7

In order to get the similar magnitude of error, we let $\Delta t = T$ for $n = 2^6, 2^7, 2^8$ and $\Delta t = T/2$ for $n = 2^9, \ldots, 2^{15}$ in the exponential quadrature rule (16) and $M = n$ for the implicit finite difference scheme in this example. Table 1 reports the numerical results. The implicit finite difference scheme in [24, 34] is a temporal first-order scheme, while our new method is a temporal higher-order accurate scheme. Therefore, to achieve the similar magnitude of error, the number of time steps of the implicit finite difference scheme is much larger than that of the exponential quadrature rule. Correspondingly, we see from Table 1 that the CPU times of the proposed method are much less than those of the implicit finite difference scheme, especially when n is very large. Moreover, the favorable numerical results can be obtained with the exponential quadrature rule only after several time steps.

Example 2. (see [24, 34, 52]) In this example, we consider the FDE (1) whose data are given as follows: $\alpha = 1.3$, $(x_L, x_R) = (0, 2)$, $T = 1$, $d_+(x) = \Gamma(3-\alpha)x^\alpha$, $d_-(x) = \Gamma(3-\alpha)(2-x)^\alpha$, and the source term

$$f(x,t) = -32\exp(-t)\left\{x^2 + \frac{1}{8}(2-x)^2(8+x^2) - \frac{3}{3-\alpha}\left[x^3 + (2-x)^3\right]\right.$$
$$\left.+\frac{3}{(4-\alpha)(3-\alpha)}\left[x^4 + (2-x)^4\right]\right\}.$$

The initial condition is chosen as

$$u(x,0) = 4x^2(2-x)^2.$$

The true solution to the corresponding FDE is given by

$$u(x,t) = 4\exp(-t)x^2(2-x)^2.$$

Table 2: Comparisons for Example 2 by the EQR and the IFD, respectively.

	EQR			IFD		
n	Δt	CPU	Error	M	CPU	Error
2^6		0.06	2.4581e-2	2^5	0.09	2.7287e-2
2^7	T	0.07	1.2452e-2	2^6	0.23	1.3738e-2
2^8		0.12	6.3198e-3	2^7	0.52	6.8923e-3
2^9		0.45	3.1010e-3	2^8	1.85	3.4520e-3
2^{10}		0.68	1.5554e-3	2^9	5.58	1.7275e-3
2^{11}		1.96	7.8159e-4	2^{10}	29.34	8.6412e-4
2^{12}	$T/2$	2.29	3.9444e-4	2^{11}	48.69	4.3209e-4
2^{13}		8.56	2.0086e-4	2^{12}	365.39	2.1597e-4
2^{14}		22.92	1.0419e-4	2^{13}	1784.74	1.0791e-4
2^{15}		33.37	5.6088e-5	2^{14}	4284.43	5.3565e-5

For the same reason as Example 1, we also set $\Delta t = T$ for $n = 2^6, 2^7, 2^8$ and $\Delta t = T/2$ for $n = 2^9, \ldots, 2^{15}$ in the exponential quadrature rule (16). The numerical results are presented in Table 2. It shows that the CPU times by the proposed method are much less than those by the implicit finite difference scheme, especially when n gets larger. Since the exponential quadrature rule used here is a higher-order method in time compared with the implicit finite difference scheme, better numerical results can be obtained only after a few time steps.

Table 3: Comparisons for Example 2 by the EQR and the IFD, respectively.

	EQR			IFD		
n	Δt	CPU	Error	M	CPU	Error
2^5	T	0.05	1.3633e-2	2^8	0.21	1.4772e-2
2^6		0.08	3.6019e-3	2^{10}	0.71	3.7197e-3
2^7	$T/2$	0.19	8.5809e-4	2^{12}	2.60	9.2719e-4
2^8		0.22	2.2355e-4	2^{14}	11.77	2.3385e-4

We remark that our new method of this paper is still valid if the left-sided and the right-sided fractional derivatives in (1) are discretized by an existing second-order discretization scheme given in [9, 12]. Table 3 reports the numerical results in this case.

Example 1 and 2 show the efficiency by the exponential quadrature rule (16) to solve the one dimensional FDE. Analogously, the proposed method also can be extended to solve the two dimensional FDE. In the following, we study the two dimensional FDE.

Example 3. We consider the two dimensional FDE as follows:

$$\begin{aligned}
&\frac{\partial u(x,y,t)}{\partial t} - d_+(x,y)\frac{\partial^\alpha u(x,y,t)}{\partial_+ x^\alpha} - d_-(x,y)\frac{\partial^\alpha u(x,y,t)}{\partial_- x^\alpha}\\
&-e_+(x,y)\frac{\partial^\beta u(x,y,t)}{\partial_+ y^\beta} - e_-(x,y)\frac{\partial^\beta u(x,y,t)}{\partial_- y^\beta} = f(x,y,t),\\
&(x,y) \in \Omega \equiv (x_L, x_R) \times (y_L, y_R), \quad t \in (0,T],\\
&u(x,y,t) = 0, \quad (x,y) \in \partial\Omega, \quad t \in [0,T],\\
&u(x,y,0) = u^0(x,y), \quad (x,y) \in \bar{\Omega}.
\end{aligned}$$

In the numerical tests, we choose $\alpha = 1.8$, $\beta = 1.6$, $\Omega = (0,1) \times (0,1)$, $T = 1$, $d_+(x,y) = x^{\alpha-1}y$, $d_-(x,y) = (1-x)^{\alpha-1}y$, $e_+(x,y) = xy^{\beta-1}$, $e_-(x,y) = x(1-y)^{\beta-1}$, and the source term

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

The initial condition is chosen as

$$u(x,y,0) = x^2(1-x)^2y^2(1-y)^2.$$

The true solution to the corresponding FDE is given by

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

Assume the numbers of spatial discretization points in x-direction and y-direction are n_1 and n_2, respectively. Let $h_1 = (x_R - x_L)/n_1$ and $h_2 = (y_R - y_L)/n_2$ be the sizes of spatial grids and denote $d_{i,j}^{\pm} = d_{\pm}(x_i, y_j)$, $e_{i,j}^{\pm} = e_{\pm}(x_i, y_j)$. For convenience, we denote $N_1 = n_1 - 1$ and $N_2 = n_2 - 1$. Let $D_j^{\pm}$ and $E_j^{\pm}$ be diagonal matrices defined by

$$D_j^{\pm} = -\frac{1}{h_1^{\alpha}} \cdot \mathrm{diag}(d_{1,j}^{\pm}, \dots, d_{N_1,j}^{\pm}), \quad E_j^{\pm} = -\frac{1}{h_2^{\beta}} \cdot \mathrm{diag}(e_{1,j}^{\pm}, \dots, e_{N_1,j}^{\pm}),$$

where $1 \le j \le N_2$. Then the coefficient matrix in this example is given by

$$A_{h_1,h_2} = A^x + A^y,$$

with

$$A^x = \begin{bmatrix} D_1^+ & & & \\ & D_2^+ & & \\ & & \ddots & \\ & & & D_{N_2}^+ \end{bmatrix} (I_{N_2} \otimes G_\alpha) + \begin{bmatrix} D_1^- & & & \\ & D_2^- & & \\ & & \ddots & \\ & & & D_{N_2}^- \end{bmatrix} (I_{N_2} \otimes G_\alpha^\intercal),$$

and

$$A^y = \begin{bmatrix} E_1^+ & & & \\ & E_2^+ & & \\ & & \ddots & \\ & & & E_{N_2}^+ \end{bmatrix} (G_\beta \otimes I_{N_1}) + \begin{bmatrix} E_1^- & & & \\ & E_2^- & & \\ & & \ddots & \\ & & & E_{N_2}^- \end{bmatrix} (G_\beta^\intercal \otimes I_{N_1}).$$

Here "$\otimes$" denotes the Kronecker product.

Similar to (17), let

$$\mathcal{S}_{h_1,h_2} = I_{N_2} \otimes \left[d^+ S(G_\alpha) + d^- S(G_\alpha^\intercal)\right] + \left[e^+ S(G_\beta) + e^- S(G_\beta^\intercal)\right] \otimes I_{N_1},$$

where

$$d^{\pm} = -\frac{1}{N_1 N_2} \sum_{j=1}^{N_2} \sum_{i=1}^{N_1} \frac{d_{i,j}^{\pm}}{h_1^{\alpha}}, \quad e^{\pm} = -\frac{1}{N_1 N_2} \sum_{j=1}^{N_2} \sum_{i=1}^{N_1} \frac{e_{i,j}^{\pm}}{h_2^{\beta}}.$$

Then $\mathcal{S}_{h_1,h_2}$ is a block circulant matrix with circulant blocks. As a result, we consider $\mathcal{S}_{h_1,h_2}$ as the preconditioner of A_{h_1,h_2}, and $I_{N_1N_2} + \gamma\mathcal{S}_{h_1,h_2}$ as the preconditioner of $I_{N_1N_2} + \gamma A_{h_1,h_2}$. Note that the product of $\mathcal{S}_{h_1,h_2}^{-1}$ and a vector requires $\mathcal{O}(N_1 N_2 (\log N_1 + \log N_2))$ operations by two dimensional FFT; see [6, 21].

We let $\Delta t = T$ in the exponential quadrature rule (16) and $M = n_1 = n_2$ in the implicit finite difference scheme. The numerical test is reported in Table 4. From this table, we see that the CPU times of the proposed method are less than those by the implicit finite difference scheme due to the higher-order temporal accurate discretization by the exponential quadrature rule.

Table 4: Comparisons for Example 3 by the EQR and the IFD, respectively.

$n_1 = n_2$	Δt	EQR		M	IFD	
		CPU	Error		CPU	Error
2^5		0.88	1.6151e-5	2^5	1.42	1.7371e-5
2^6	T	2.91	7.8767e-6	2^6	6.54	8.3592e-6
2^7		24.25	3.9356e-6	2^7	74.97	4.1075e-6
2^8		135.43	2.0186e-6	2^8	525.70	2.0368e-6

6. Concluding remarks

In this paper, we have developed an exponential quadrature rule to solve the space-discretized system of the FDE. The convergence of the proposed method has been studied. Theoretical analysis reveals that the exponential quadrature rule is convergent of order s if s non-confluent quadrature nodes are used. In implementation, the Toeplitz-like matrix exponential, which is involved in the exponential integrator method, is approximated by the shift-invert Arnoldi method. The discretized matrix is proved to be sectorial when $d_{\pm}(x)$ are constants, which ensures the error bound of the shift-invert Arnoldi approximation is independent of the discretized matrix norm. Numerical examples have been given to illustrate the efficiency and robustness of the proposed method.

Generally, the exponential quadrature rule of this paper is used to solve linear parabolic problems (8), and the matrix A is time-invariant. Therefore, the method developed in the current paper can be extended to solve the three-dimensional problem [49]. If the matrix A is time-dependent, Magnus integrators have been studied for this problem; please refer to [14, 15] for details. However, the exponential quadrature rule in this present paper cannot be extended to the fractional partial differential equations [19, 20] or other types of non-local models [46, 50, 51], since these problems do not belong to linear parabolic problems (8).

We also want to remark that although this current paper focus on using the first-order discretization in space, the exponential quadrature rule of this paper is still valid if applying the higher-order discretizations, e.g., [8, 9, 12]; in fact the validity is already verified numerically in Table 3, where we exploit the second-order discretization in [9, 12]. In our future consideration, we will extend the exponential integrator methods, with the Runge-Kutta methods, to solve the semilinear problems.

Acknowledgements

The authors wish to thank the referees for their many constructive comments and suggestions on improving the paper.

References

[1] J. Bai, X. Feng, Fractional-order anisotropic diffusion for image denoising, IEEE Trans. Image Proc. 16 (2007) 2492–2502.

[2] D. Benson, S.W. Wheatcraft, M.M. Meerschaert, Application of a fractional advection-dispersion equation, Water Resour. Res. 36 (2000) 1403–1413.

[3] D. Benson, S.W. Wheatcraft, M.M. Meerschaert, The fractional-order governing equation of Lévy motion, Water Resour. Res. 36 (2000) 1413–1423.

[4] B. Beumer, M. Kovács, M.M. Meerschaert, Numerical solutions for fractional reaction-diffusion equations, Comput. Math. Appl. 55 (2008) 2212–2226.

[5] B.A. Carreras, V.E. Lynch, G.M. Zaslavsky, Anomalous diffusion and exit time distribution of particle tracers in plasma turbulence model, Phys. Plasma 8 (2001) 5096–5103.

[6] R. Chan, X. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, 2007.

[7] R. Chan, M. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev. 38 (1996) 427–482.

[8] M. Chen, W. Deng, Fourth Order Difference Approximations for Space Riemann-Liouville Derivatives Based on Weighted and Shifted Lubich Difference Operators, Commun. Comput. Phys. 16 (2014) 516–540.

[9] M. Chen, Y. Wang, X. Cheng, W. Deng, Second-order LOD multigrid method for multidimensional Riesz fractional diffusion equation, BIT 54 (2014) 623–647.

[10] M. Cui, Compact finite difference method for the fractional diffusion equation, J. Comput. Phys. 228 (2009) 7792–7804.

[11] W. Deng, Finite element method for the space and time fractional Fokker-Planck equation, SIAM J. Numer. Anal. 47 (2008) 204–226.

[12] W. Deng, M. Chen, Efficient Numerical Algorithms for Three-Dimensional Fractional Partial Differential Equations, J. Comp. Math. 32 (2014) 371–391.

[13] V.J. Ervin, N. Heuer, J.P. Roop, Numerical approximation of a time dependent, non-linear, space-fractional diffusion equation, SIAM J. Numer. Anal. 45 (2007) 572–591.

[14] C. González, A. Ostermann, M. Thalhammer, A second-order Magnus-type integrator for nonautonomous parabolic problems, J. Comput. Appl. Math. 189 (2006) 142-156.

[15] C. González, M. Thalhammer, A second-order Magnus-type integrator for quasi-linear parabolic problems, Math. Comp. 76 (2007) 205-231.

[16] M. Hochbruck, A. Ostermann, Exponential Runge-Kutta methods for parabolic problems, Appl. Numer. Math. 53 (2005) 323–339.

[17] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286.

[18] R.A. Horn, C.R. Johnson, Topics in Matrix Analysis, Cambridge University Press, 1991.

[19] J. Jia, H. Wang, Fast finite difference methods for space-fractional diffusion equations with fractional derivative boundary conditions, J. Comput. Phys., Available online 19 August 2014.

[20] J. Jia, C. Wang, H. Wang, A fast locally refined method for a space-fractional diffusion equation, # IE0147, ICFDA'14 Catania, 23–25 June 2014 Copyright 2014 IEEE ISBN 978-1-4799-2590-2.

[21] X. Jin, Developments and Applications of Block Toeplitz Iterative Solvers, Science Press & Kluwer Academic Publishers, Beijing/Dordrecht, The Netherlands, 2002.

[22] S. Lee, X. Liu, H. Sun, Fast exponential time integration scheme for option pricing with jumps, Numer. Linear Algebra Appl. 19 (2012) 87–101.

[23] S. Lee, H. Pang, H. Sun, Shift-invert Arnoldi approximation to the Toeplitz matrix exponential, SIAM J. Sci. Comput. 32 (2010) 774–792.

[24] S. Lei, H. Sun, A circulant preconditioner for fractional diffusion equations, J. Comput. Phys. 242 (2013) 715–725.

[25] X. Li, C. Xu, The existence and uniqueness of the week solution of the space-time fractional diffusion equation and a spectral method approximation, Commun. Comput. Phys. 8 (2010) 1016–1051.

[26] Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, J. Comput. Phys. 225 (2007) 1533–1552.

[27] F. Lin, S. Yang, X. Jin, Preconditioned iterative methods for fractional diffusion equation, J. Comput. Phys. 256 (2014) 109–117.

[28] M.M. Meerschaert, H.P. Scheffler, C. Tadjeran, Finite difference methods for two-dimensional fractional dispersion equation, J. Comput. Phys. 211 (2006) 249–261.

[29] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for fractional advection-dispersion flow equations, J. Comput. Appl. Math. 172 (2004) 65–77.

[30] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math. 56 (2006) 80–90.

[31] I. Moret, P. Novati, RD-rational approximations of the matrix exponential, BIT 44 (2004) 595–615.

[32] J. Pan, R. Ke, M. Ng, H. Sun, Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations, SIAM J. Sci. Comput. 36 (2014) A2698-A2719.

[33] H. Pang, H. Sun, Shift-invert Lanczos method for the symmetric positive semidefinite Toeplitz matrix exponential, Numer. Linear Algebra Appl. 18 (2011) 603–614.

[34] H. Pang, H. Sun, Multigrid method for fractional diffusion equations, J. Comput. Phys. 231 (2012) 693–703.

[35] H. Pang, H. Sun, Fast exponential time integration for pricing options in stochastic volatility jump diffusion models, East Asia Journal of Applied Mathematics 4 (2014) 53–68.

[36] I. Podlubny, Fractional Differential Equations, Academic Press, New York, 1999.

[37] D.L. Powers, R. Jeltsch, Problem 74-5: On the norm of a matrix exponential, SIAM Rev. 17 (1975) 174–176.

[38] W. Qu, S. Lei, S. Vong, Circulant and skew-circulant splitting iteration for fractional advection-diffusion equations, Int. J. Comput. Math. 91 (2014) 2232–2242.

[39] M. Raberto, E. Scalas, F. Mainardi, Waiting-times and returns in high-frequency financial data: an empirical study, Phys. A 314 (2002) 749–755.

[40] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, Philadelphia, 2003.

[41] M.F. Shlesinger, B.J. West, J. Klafter, Lévy dynamics of enhanced diffusion: application to turbulence, Phys. Rev. Lett. 58 (1987) 1100–1103.

[42] I.M. Sokolov, J. Klafter, A. Blumen, Fractional kinetics, Phys. Today Nov. 55 (2002) 48–54.

[43] E. Sousa, Finite difference approximates for a fractional advection diffusion problem, J. Comput. Phys. 228 (2009) 4038–4054.

[44] L. Su, W. Wang, Z. Yang, Finite difference approximations for the fractional advection-diffusion equation, Phys. Lett. A 373 (2009) 4405–4408.

[45] C. Tadjeran, M.M. Meerschaert, H.P. Scheffler, A second-order accurate numerical approximation for the fractional diffusion equation, J. Comput. Phys. 213 (2006) 205–213.

[46] H. Tian, H. Wang, W. Wang, An efficient collocation method for a non-local diffusion model, Int. J. Numer. Anal. Model. 10 (2013) 815–825.

[47] W. Tian, H. Zhou, W. Deng, A class of second order difference approximations for solving space fractional diffusion equations, Math. Comp. in press (arXiv:1201.5949 [math.NA]).

[48] H. Wang, T. Basu, A fast finite difference method for two-dimensional space-fractional diffusion equations, SIAM J. Sci. Comput. 34 (2012) A2444–A2458.

[49] H. Wang, N. Du, A fast finite difference method for three-dimensional time-dependent space-fractional diffusion equations and its efficient implementation, J. Comput. Phys. 253 (2013) 50–63.

[50] H. Wang, H. Tian, A fast Galerkin method with efficient matrix assembly and storage for a peridynamic model, J. Comput. Phys. 231 (2012) 7730–7738.

[51] H. Wang, H. Tian, A fast and faithful collocation method with efficient matrix assembly for a two dimensional nonlocal diffusion model, Comput. Methods Appl. Mech. Engrg. 273 (2014) 19–36.

[52] H. Wang, K. Wang, T. Sircar, A direct $\mathcal{O}(N \log^2 N)$ finite difference method for fractional diffusion equations, J. Comput. Phys. 229 (2010) 8095–8104.

[53] G.M. Zaslavsky, D. Stevens, H. Weitzner, Self-similar transport in incomplete chaos, Phys. Rev. E 48 (1993) 1683–1694.

[54] H. Zhou, W. Tian, W. Deng, Quasi-compact finite difference schemes for space fractional diffusion equations, J. Sci. Comput. 56 (2013) 45-66.