



---

*Research article*

## **High dimensional Riesz space distributed-order advection-dispersion equations with ADI scheme in compression format**

**Lot-Kei Chou and Siu-Long Lei\***

Department of Mathematics, University of Macau, Avenida da Universidade, Taipa, Macau, China

\* **Correspondence:** Email: [slei@um.edu.mo](mailto:slei@um.edu.mo); Tel: +(853)88224449.

**Abstract:** A second order alternating direction implicit scheme for time-dependent Riesz space distributed-order advection-dispersion equations is applied to higher dimensions with the Tensor-Train decomposition technique. The solutions are solved in compressed format, the Tensor-Train format, and the errors accumulated due to compressions are analyzed to ensure convergence. Problems with low-rank data are tested, the results illustrated a steeper growth in the ranks of the numerical solutions than that in related works.

**Keywords:** high dimension; advection-dispersion equations; Riesz fractional derivative; distributed-order; alternating direction implicit method; compression format

---

### **1. Introduction**

In recent decades, fractional differential equations (FDEs) have been widely considered in modelling anomalous diffusion. A class of FDEs, the distributed-order FDEs (DO-FDEs), have been applied in modelling ultraslow diffusion [1, 2], mixture of delay sources [3], and dielectric induction and diffusion [4].

To solve DO-FDEs with numerical method, one may discretize the integral with certain quadrature rule, followed by applying an approximation for ordinary fractional derivative [5–7]. In particular, in [8], a form of DO-FDE called Riesz space distributed-order advection-diffusion equations (RSDO-ADE) is studied, with second order schemes proposed for one-dimensional and two-dimensional cases.

On the other hand, a data compression format called the Tensor-Train format (TT-format) [9] has been employed in various high dimensional problems. In particular, a TT-format iterative method called the TT-GMRES method is widely used in solving FDE related problems, especially for those which discretized linear systems possess low-rank structure [10–12]. Benefiting from the Toeplitz-like structure of the linear systems, many studies have been made to explore the underlying properties and to the development of preconditioners [13, 14]. Another TT-format approach for solving FDE

problems is the application of the alternating direction implicit (ADI) method [15, 16], which reduces the linear systems into one-dimensional systems. Although this approach probably limits the classes of problems to be solved, the convergence analysis is relatively simple to perform. These results provide practical ground for the implementation of RSDO-ADE with higher dimensions, and suggest that the implementation is a trial for the latter approach.

In this work, the RSDO-ADE from [8] is considered in high dimensional form. As the concerning fractional orders are distributed over  $(0, 1) \cup (1, 2)$  instead of subintervals of  $(1, 2)$ , and the derivatives are of Riesz type instead of weighted two-sided type, we consider the finite volume approximation in [16] not more suitable here than the approximation in [8]. Following the proposed discretization, we adopt the midpoint quadrature rule for the integrals, a second order approximation for the Riesz space fractional derivatives, and the Crank-Nicolson method, thus obtaining a second order scheme. Further, similar to [15] and [16], we apply the ADI method for the reduction of the dimensionality, and TT-format is adopted so that the resulting scheme can be solved in compressed form, provided that the given data possess low TT-ranks.

As compression format introduces perturbations, error analysis is performed to estimate and maintain the overall convergence order. For efficiency, consider a  $d$ -dimensional case, discretized as linear systems of size  $N$  with  $N$  time steps, and suppose the numerical solutions possess TT-ranks around  $r$ , then the proposed method requires storage of  $O(2N^2 + dr^2N)$  and operation cost of  $O(dr^2N^3 + dr^3N^2)$ , provided that the Gaussian Elimination (GE) method is adopted for solving the linear systems. The analysis is testified by some numerical examples with  $d \leq 20$ .

The content is briefly described as follows. The  $d$ -dimensional RSDO-ADE problem is first presented in Section 2, then the implementation of the TT-format method is described in Section 3, and some numerical results are presented in Section 4.

## 2. The $d$ -dimensional RSDO-ADE

### 2.1. The problem

For  $\mathbf{x} = [x^{(1)}, \dots, x^{(d)}]^\top$  and  $\Omega = \prod_{k=1}^d [x_a^{(k)}, x_b^{(k)}]$ , we consider the  $d$ -dimensional RSDO-ADE of the following form.

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \sum_{k=1}^d \left( \int_0^1 P(\alpha) \frac{\partial^\alpha u(\mathbf{x}, t)}{\partial |x^{(k)}|^\alpha} d\alpha + \int_1^2 Q(\beta) \frac{\partial^\beta u(\mathbf{x}, t)}{\partial |x^{(k)}|^\beta} d\beta \right) + f(\mathbf{x}, t),$$

$$(\mathbf{x}, t) \in \Omega \times [0, T], \quad (2.1)$$

$$u(\mathbf{x}, t) = 0, \quad (\mathbf{x}, t) \in (\mathbb{R}^d \setminus \Omega) \times [0, T],$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

where

$$P(\alpha) \geq 0, \quad P(\alpha) \not\equiv 0, \quad \alpha \in [0, 1], \quad 0 < \int_0^1 P(\alpha) d\alpha < \infty,$$

$$Q(\beta) \geq 0, \quad Q(\beta) \not\equiv 0, \quad \beta \in [1, 2], \quad 0 < \int_1^2 Q(\beta) d\beta < \infty.$$

Here,  $\partial^\alpha u / \partial |x^{(k)}|^\alpha$  and  $\partial^\beta u / \partial |x^{(k)}|^\beta$  denote the  $\alpha$ -order and  $\beta$ -order Riesz space fractional derivatives of  $u(x, t)$  respectively, with the forms:

$$\frac{\partial^\alpha u(\mathbf{x}, t)}{\partial |x^{(k)}|^\alpha} = -c_\alpha \left( {}_{x_a^{(k)}}D_{x^{(k)}}^\alpha u(\mathbf{x}, t) + {}_{x^{(k)}}D_{x_b^{(k)}}^\alpha u(\mathbf{x}, t) \right),$$

$$\frac{\partial^\beta u(\mathbf{x}, t)}{\partial |x^{(k)}|^\beta} = -c_\beta \left( {}_{x_a^{(k)}}D_{x^{(k)}}^\beta u(\mathbf{x}, t) + {}_{x^{(k)}}D_{x_b^{(k)}}^\beta u(\mathbf{x}, t) \right),$$

where

$$c_\alpha = \frac{1}{2 \cos(\pi\alpha/2)}, \quad 0 < \alpha < 1,$$

$$c_\beta = \frac{1}{2 \cos(\pi\beta/2)}, \quad 1 < \beta < 2,$$

$${}_{x_a^{(k)}}D_{x^{(k)}}^\alpha = \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial x^{(k)}} \int_{x_a^{(k)}}^{x^{(k)}} (x^{(k)} - \xi)^{-\alpha} u(\mathbf{x}, t)|_{x^{(k)}=\xi} d\xi,$$

$${}_{x^{(k)}}D_{x_b^{(k)}}^\alpha = \frac{-1}{\Gamma(1-\alpha)} \frac{\partial}{\partial x^{(k)}} \int_{x^{(k)}}^{x_b^{(k)}} (\xi - x^{(k)})^{-\alpha} u(\mathbf{x}, t)|_{x^{(k)}=\xi} d\xi,$$

$${}_{x_a^{(k)}}D_{x^{(k)}}^\beta = \frac{1}{\Gamma(2-\beta)} \frac{\partial^2}{\partial (x^{(k)})^2} \int_{x_a^{(k)}}^{x^{(k)}} (x^{(k)} - \xi)^{1-\beta} u(\mathbf{x}, t)|_{x^{(k)}=\xi} d\xi,$$

$${}_{x^{(k)}}D_{x_b^{(k)}}^\beta = \frac{1}{\Gamma(2-\beta)} \frac{\partial^2}{\partial (x^{(k)})^2} \int_{x^{(k)}}^{x_b^{(k)}} (\xi - x^{(k)})^{1-\beta} u(\mathbf{x}, t)|_{x^{(k)}=\xi} d\xi,$$

where  $\Gamma(\cdot)$  denotes the gamma function.

We remark that in this problem form, the boundary condition is modified as described in [17].

## 2.2. Discretization

The discretization of Problem 2.1 is described below.

For the integrals, by taking a positive integer  $S$ , we define  $\sigma = 1/S$ , grid points  $\xi_j = j\sigma$  for  $0 \leq j \leq S$ ,  $\alpha_j = (\xi_j + \xi_{j-1})/2 = (j-0.5)\sigma$  for  $1 \leq j \leq S$ ,  $\beta_j = ((1 + \xi_j) + (1 + \xi_{j-1}))/2 = 1 + (j-0.5)\sigma$  for  $1 \leq j \leq S$ . Then we define the midpoint quadrature operators

$$I_S^P(v(\alpha)) \triangleq \sigma \sum_{j=1}^S P(\alpha_j)v(\alpha_j), \quad \text{and} \quad I_S^Q(v(\beta)) \triangleq \sigma \sum_{j=1}^S Q(\beta_j)v(\beta_j),$$

where  $v$  can be a function or a matrix. Like the integrals, the operators are linear.

With the operators, we have

$$\int_0^1 P(\alpha) \frac{\partial^\alpha u(\mathbf{x}, t)}{\partial |x^{(k)}|^\alpha} d\alpha \approx I_S^P \left( \frac{\partial^\alpha u(\mathbf{x}, t)}{\partial |x^{(k)}|^\alpha} \right),$$

$$\int_1^2 Q(\beta) \frac{\partial^\beta u(\mathbf{x}, t)}{\partial |x^{(k)}|^\beta} d\beta \approx I_S^Q \left( \frac{\partial^\beta u(\mathbf{x}, t)}{\partial |x^{(k)}|^\beta} \right).$$

For the derivatives, take positive integers  $N, M$ , and grid points  $x_i^{(k)} = x_a^{(k)} + ih_k$  and  $t_m = m\tau$ , where  $h_k = (x_b^{(k)} - x_a^{(k)})/(N + 1)$ ,  $\tau = T/M$ ,  $0 \leq i_k \leq N + 1$ ,  $1 \leq k \leq d$ , and  $0 \leq m \leq M$ . Then, denote  $u_{i_1, \dots, i_d}^{(m)} = u(x_{i_1}^{(1)}, \dots, x_{i_d}^{(d)}, t_m)$ ,  $f_{i_1, \dots, i_d}^{(m)} = f(x_{i_1}^{(1)}, \dots, x_{i_d}^{(d)}, t_m)$ .

As presented in [8, 18], the second-order approximations for the Riesz space fractional derivatives have the forms

$$\frac{\partial^\alpha u(x_{i_1}^{(1)}, \dots, x_{i_k}^{(k)}, \dots, x_{i_d}^{(d)}, t_m)}{\partial |x^{(k)}|^\alpha} \approx -\frac{1}{h_k^\alpha} \sum_{j=1}^N g_{i_k-j}^{(\alpha)} u_{i_1, \dots, j, \dots, i_d}^{(m)},$$

$$\frac{\partial^\beta u(x_{i_1}^{(1)}, \dots, x_{i_k}^{(k)}, \dots, x_{i_d}^{(d)}, t_m)}{\partial |x^{(k)}|^\beta} \approx -\frac{1}{h_k^\beta} \sum_{j=1}^N g_{i_k-j}^{(\beta)} u_{i_1, \dots, j, \dots, i_d}^{(m)},$$

where

$$g_0^{(\alpha)} = \frac{\Gamma(1 + \alpha)}{(\Gamma(\alpha/2 + 1))^2}, \quad g_k^{(\alpha)} = g_{-k}^{(\alpha)} = \frac{(-1)^k \Gamma(1 + \alpha)}{\Gamma(\alpha/2 - k + 1) \Gamma(\alpha/2 + k + 1)}$$

$$= \left(1 - \frac{1 + \alpha}{\alpha/2 + k}\right) g_{k-1}^{(\alpha)} \quad \text{for } k \geq 1,$$

$$g_0^{(\beta)} = \frac{\Gamma(1 + \beta)}{(\Gamma(\beta/2 + 1))^2}, \quad g_k^{(\beta)} = g_{-k}^{(\beta)} = \frac{(-1)^k \Gamma(1 + \beta)}{\Gamma(\beta/2 - k + 1) \Gamma(\beta/2 + k + 1)}$$

$$= \left(1 - \frac{1 + \beta}{\beta/2 + k}\right) g_{k-1}^{(\beta)} \quad \text{for } k \geq 1.$$

With the notations, for  $k = 1, \dots, d$ , define the operators

$$\delta_k v_{i_1, \dots, i_k, \dots, i_d} = I_S^P \left( \frac{1}{h_k^\alpha} \sum_{j=1}^N g_{i_k-j}^{(\alpha)} v_{i_1, \dots, j, \dots, i_d} \right) + I_S^Q \left( \frac{1}{h_k^\beta} \sum_{j=1}^N g_{i_k-j}^{(\beta)} v_{i_1, \dots, j, \dots, i_d} \right)$$

$$= \sum_{j=1}^N I_S^P \left( \frac{1}{h_k^\alpha} g_{i_k-j}^{(\alpha)} \right) v_{i_1, \dots, j, \dots, i_d} + \sum_{j=1}^N I_S^Q \left( \frac{1}{h_k^\beta} g_{i_k-j}^{(\beta)} \right) v_{i_1, \dots, j, \dots, i_d}.$$

With the approximations and operators, the Crank-Nicolson method is applied to obtain the second-order implicit finite difference scheme

$$\left(1 + \frac{\tau}{2} \sum_{k=1}^d \delta_k\right) u_{i_1, \dots, i_d}^{(m)} = \left(1 - \frac{\tau}{2} \sum_{k=1}^d \delta_k\right) u_{i_1, \dots, i_d}^{(m-1)} + \tau f_{i_1, \dots, i_d}^{m-\frac{1}{2}}, \quad (2.2)$$

$$1 \leq i_k \leq N, 1 \leq k \leq d, 1 \leq m \leq M.$$

By complementing appropriate cross terms of  $\mathcal{O}(\tau^2)$ , we obtain the ADI scheme

$$\left(1 + \frac{\tau}{2} \delta_1\right) \cdots \left(1 + \frac{\tau}{2} \delta_d\right) u_{i_1, \dots, i_d}^{(m)} = \left(1 - \frac{\tau}{2} \delta_1\right) \cdots \left(1 - \frac{\tau}{2} \delta_d\right) u_{i_1, \dots, i_d}^{(m-1)} + \tau f_{i_1, \dots, i_d}^{m-\frac{1}{2}}, \quad (2.3)$$

$$1 \leq i_k \leq N, 1 \leq k \leq d, 1 \leq m \leq M.$$

2.3. Matrix form

Denote

$$\begin{aligned} \mathbf{u}^{(m)} &= [u_{1,1,\dots,1}^{(m)}, \dots, u_{N,1,\dots,1}^{(m)}, \dots, u_{1,N,\dots,N}^{(m)}, \dots, u_{N,N,\dots,N}^{(m)}]^\top, \\ \mathbf{f}^{(m)} &= [f_{1,1,\dots,1}^{(m)}, \dots, f_{N,1,\dots,1}^{(m)}, \dots, f_{1,N,\dots,N}^{(m)}, \dots, f_{N,N,\dots,N}^{(m)}]^\top, \end{aligned}$$

$I$  as the  $N \times N$  identity matrix, and  $\hat{A}$  as the Toeplitz matrix

$$\hat{A} = \begin{bmatrix} \tilde{g}_0 & \tilde{g}_{-1} & \cdots & \cdots & \tilde{g}_{-N+2} & \tilde{g}_{-N+1} \\ \tilde{g}_1 & \tilde{g}_0 & \tilde{g}_{-1} & \ddots & \ddots & \tilde{g}_{-N+2} \\ \vdots & \tilde{g}_1 & \tilde{g}_0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \tilde{g}_{N-2} & \ddots & \ddots & \ddots & \tilde{g}_0 & \tilde{g}_{-1} \\ \tilde{g}_{N-1} & \tilde{g}_{N-2} & \cdots & \cdots & \tilde{g}_1 & \tilde{g}_0 \end{bmatrix},$$

where  $\tilde{g}_k = I_S^P \left( \frac{1}{h^\alpha} g_k^{(\alpha)} \right) + I_S^Q \left( \frac{1}{h^\beta} g_k^{(\beta)} \right) = \tilde{g}_{-k}$ , so  $\hat{A}$  is symmetric.

Also, denote

$$\tilde{I} := \underbrace{I_N \otimes \cdots \otimes I_N}_d \quad \text{and} \quad \tilde{A}_k := \underbrace{I_N \otimes \cdots \otimes I_N}_{d-k} \otimes \hat{A} \otimes \underbrace{I_N \otimes \cdots \otimes I_N}_{k-1}.$$

Together with the extended zero boundary condition, the scheme (2.2) can be rewritten as the matrix form

$$\left( \tilde{I} + \frac{\tau}{2} \sum_{k=1}^d \tilde{A}_k \right) \mathbf{u}^{(m)} = \left( \tilde{I} - \frac{\tau}{2} \sum_{k=1}^d \tilde{A}_k \right) \mathbf{u}^{(m-1)} + \tau \mathbf{f}^{(m-\frac{1}{2})}, \quad 1 \leq m \leq M.$$

while the ADI scheme (2.3) can be rewritten as

$$\left( \tilde{I} + \frac{\tau}{2} \tilde{A}_1 \right) \cdots \left( \tilde{I} + \frac{\tau}{2} \tilde{A}_d \right) \mathbf{u}^{(m)} = \left( \tilde{I} - \frac{\tau}{2} \tilde{A}_1 \right) \cdots \left( \tilde{I} - \frac{\tau}{2} \tilde{A}_d \right) \mathbf{u}^{(m-1)} + \tau \mathbf{f}^{(m-\frac{1}{2})}, \quad 1 \leq m \leq M,$$

or equivalently,

$$\mathring{A} \mathbf{u}^{(m)} = \mathring{A}' \mathbf{u}^{(m-1)} + \tau \mathbf{f}^{(m-\frac{1}{2})}, \quad 1 \leq m \leq M, \tag{2.4}$$

where  $\mathring{A} = \underbrace{A \otimes \cdots \otimes A}_d$  and  $\mathring{A}' = \underbrace{A' \otimes \cdots \otimes A'}_d$ , with  $A = I + (\tau/2)\hat{A}$  and  $A' = I - (\tau/2)\hat{A}$ .

In [8], it is proved that  $\rho(A^{-1}A') = \|A^{-1}A'\|_2 < 1$ . With this property, we have

$$\begin{aligned} \|\mathring{A}^{-1}\mathring{A}'\|_2 &= \|(A^{-1} \otimes \cdots \otimes A^{-1})(A' \otimes \cdots \otimes A')\|_2 \\ &= \|A^{-1}A' \otimes \cdots \otimes A^{-1}A'\|_2 = \prod_{k=1}^d \|A^{-1}A'\|_2 < 1, \end{aligned}$$

such that

$$\|\mathring{A}^{-1}\mathring{A}'\mathbf{x}\|_2 < \|\mathbf{x}\|_2, \quad \forall \mathbf{x}. \tag{2.5}$$

### 3. Solving process

#### 3.1. TT-format method

Following [15] and [16], the linear system (2.4) can be solved in compressed form if  $\mathbf{u}^{(0)}$  and  $\mathbf{f}^{(m-1/2)}$  are in TT-format, or  $u_0(\mathbf{x})$  and  $f(\mathbf{x}, t_{m-1/2})$  possess functional TT-format (FTT-format).

Suppose  $u_0(\mathbf{x})$  and  $f(\mathbf{x}, t_{m-1/2})$  have FTT-formats

$$u_0(\mathbf{x}) = G_1(x^{(1)}) \cdots G_d(x^{(d)}) \quad \text{and} \quad f(\mathbf{x}, t_{m-\frac{1}{2}}) = H_1^{(m-\frac{1}{2})}(x^{(1)}) \cdots H_d^{(m-\frac{1}{2})}(x^{(d)}),$$

where  $G_k$  and  $H_k$  are of sizes  $r_{k-1} \times r_k$  and  $r'_{k-1} \times r'_k$  respectively for  $1 \leq k \leq d$ , then  $\mathbf{u}^{(0)}$  and  $\mathbf{f}^{(m-1/2)}$  can be reshaped to tensors  $\mathcal{U}^{(0)}$  and  $\mathcal{F}^{(m-1/2)}$  with TT-formats

$$\begin{aligned} \mathcal{U}^{(0)}(i_1, \dots, i_d) &= u_0(x_{i_1}^{(1)}, \dots, x_{i_d}^{(d)}) \\ &= G_1(x_{i_1}^{(1)}) \cdots G_d(x_{i_d}^{(d)}) := U_1^{(0)}(i_1) \cdots U_d^{(0)}(i_d) \quad \text{and} \\ \mathcal{F}^{(m-\frac{1}{2})}(i_1, \dots, i_d) &= f(x_{i_1}^{(1)}, \dots, x_{i_d}^{(d)}, t_{m-\frac{1}{2}}) \\ &= H_1^{(m-\frac{1}{2})}(x_{i_1}^{(1)}) \cdots H_d^{(m-\frac{1}{2})}(x_{i_d}^{(d)}) := F_1^{(m-\frac{1}{2})}(i_1) \cdots F_d^{(m-\frac{1}{2})}(i_d), \end{aligned}$$

where  $U_k^{(0)}$  and  $F_k^{(m-1/2)}$  are of sizes  $r_{k-1} \times N \times r_k$  and  $r'_{k-1} \times N \times r'_k$  respectively for  $1 \leq k \leq d$ .

After this, the numerical solutions can be obtained in TT-format through the following algorithm.

---

#### Algorithm 1

---

**Input:** Matrix  $\hat{A}$ , TT-cores  $U_1^{(0)}, \dots, U_d^{(0)}$  of  $\mathcal{U}^{(0)}$ , TT-cores  $F_1^{(m-1/2)}, \dots, F_d^{(m-1/2)}$  of  $\mathcal{F}^{(m-1/2)}$  for  $m = 1, 2, \dots, M$ , and relative error  $\epsilon$ .

**Output:** TT-cores  $\tilde{U}_1^{(m)}, \dots, \tilde{U}_d^{(m)}$  of  $\tilde{\mathcal{U}}^{(m)}$  for  $m = 0, 1, 2, \dots, M$ .

- 1: Compute  $\mathcal{S} = \left(I + \frac{\tau}{2}\hat{A}\right)^{-1}$  and  $\mathcal{S}' = \mathcal{S} \left(I - \frac{\tau}{2}\hat{A}\right)$
  - 2: **for**  $k = 1 : d$  **do**
  - 3:    $\tilde{U}_k^{(0)} = U_k^{(0)}$
  - 4: **end for**
  - 5: **for**  $m = 1 : M$  **do**
  - 6:    $\tilde{U}_1^{(m)} = \left(\tilde{U}_1^{(m-1)} \times_2 \mathcal{S}' \quad F_1^{(m-\frac{1}{2})} \times_2 \mathcal{S}\right)$
  - 7:   **for**  $k = 2 : d - 1$  **do**
  - 8:      $\tilde{U}_k^{(m)} = \begin{pmatrix} \tilde{U}_k^{(m-1)} \times_2 \mathcal{S}' & 0 \\ 0 & F_k^{(m-\frac{1}{2})} \times_2 \mathcal{S} \end{pmatrix}$
  - 9:   **end for**
  - 10:    $\tilde{U}_d^{(m)} = \begin{pmatrix} \tilde{U}_d^{(m-1)} \times_2 \mathcal{S}' \\ \tau F_d^{(m-\frac{1}{2})} \times_2 \mathcal{S} \end{pmatrix}$
  - 11:   Round  $\tilde{\mathcal{U}}^{(m)}$  to  $\hat{\mathcal{U}}^{(m)}$  with relative error  $\epsilon$  by rounding process.
  - 12: **end for**
- 

In the algorithm, the notation  $U \times_2 A$  denotes the multiplication of the matrix  $A$  with  $U$  along the second dimension, or with the mode-2 fibers of  $U$ ; while “Round” denotes a recompression process called rounding for TT-format, expecting to reduce TT-ranks by introducing relative error.

### 3.2. Computational efficiency

Here, we discuss the efficiency of Algorithm 3.1.

Suppose the GE method is adopted in line 1 to compute  $\mathcal{S}$ , and the numerical solutions possess TT-ranks  $\approx r$ .

For storage, as the major components are the matrices and TT-cores of the numerical solutions, the GE method requires storage of  $\mathcal{O}(N^2 + dr^2N)$ .

For operation cost, there are three major parts as shown in Algorithm 3.1. The first part is line 1, the matrix inversion; the second part is line 6 to line 10, matrix-vector multiplications in single time step; and the third part is line 11, rounding process in single time step. Thus the GE method requires totally  $\mathcal{O}(N^3 + M(dr^2N^2 + dr^3N))$  operations.

We can see that the TT-ranks  $r$  is a crucial factor affecting the efficiency. Theoretically,  $r$  may range from small numbers to powers of  $N$ , so that compressed format method may be less efficient than full storage method. With rough estimation, if the problem data possess TT-ranks  $\approx r'$ , then the numerical solutions can be computed with TT-ranks of at most  $\mathcal{O}(Mr')$ , which is attained when rounding process does not compress at all. Numerical examples indicate a tentative form of growth that  $r \approx \mathcal{O}(\log^k N)$  for some  $k > 0$ , which implies a possible dominance of the rounding process over the whole solving process. Further studies about the possible forms of growth of TT-ranks are to be made.

**Remark 1.** By virtue of the Toeplitz structure of  $\hat{A}$ , hence  $I + (\tau/2)\hat{A}$ , the operation cost may be reduced to  $\mathcal{O}(N \log N + M(dr^2N \log N + dr^3N))$  in the optimal case. This may be accomplished by adopting the circulant-and-skew-circulant representation method [19], equipped with preconditioned conjugate gradient method and the Fast Fourier Transform [20].

### 3.3. Error analysis

For the overall convergence of the proposed method, we have the following proposition.

**Proposition 1.** Suppose  $P(\alpha)$ ,  $Q(\beta)$ , and the exact solution  $u(\mathbf{x}, t)$  of Problem 2.1 satisfy the following conditions.

- i.  $\frac{\partial^2 P(\alpha)}{\partial \alpha^2}$ ,  $\frac{\partial^2}{\partial \alpha^2} \frac{\partial^\alpha u(\mathbf{x}, t)}{\partial |x^{(k)}|^\alpha} \in C(\Omega \times [0, T] \times [0, 1])$  with respect to  $(\mathbf{x}, t, \alpha)$ , and  $\frac{\partial^2 Q(\beta)}{\partial \beta^2}$ ,  $\frac{\partial^2}{\partial \beta^2} \frac{\partial^\beta u(\mathbf{x}, t)}{\partial |x^{(k)}|^\beta} \in C(\Omega \times [0, T] \times [1, 2])$  with respect to  $(\mathbf{x}, t, \beta)$ .

- ii. For some  $\rho > 0$ , for  $1 \leq k \leq d$ , the mixed partial derivatives

$$\frac{\partial^{\gamma_1}}{\partial |x^{(1)}|^{\gamma_1}} \cdots \frac{\partial^{\gamma_d}}{\partial |x^{(d)}|^{\gamma_d}} \frac{\partial^{\gamma_*}}{\partial t^{\gamma_*}} u(\mathbf{x}, t)$$

are in  $C(\Omega \times [0, T]) \cap \mathcal{L}^1(\Omega \times [0, T])$  for all  $\gamma_k \in [0, 5 + \rho]$ ,  $\gamma_* \in [0, 3 + \rho]$ , and vanish at infinity for all  $\gamma_k \in [0, 4 + \rho]$ ,  $\gamma_* \in [0, 2 + \rho]$ .

Then for rounding relative error  $\epsilon$ , we have

$$E_m \leq K(1 + \epsilon)^M(\tau^2 + \hat{h}^2 + \sigma^2 + M\epsilon).$$

where  $E_m$  is the discrete 2-norm error between the exact solution in Eq (2.4) and the perturbed numerical solution at  $t_m$  due to rounding,  $K > 0$  is a constant independent of  $m, \tau, h_k, \sigma$ , and  $\hat{h} = \max_{1 \leq k \leq d} h_k$ .

*Proof.* Denote  $\mathbf{u}^{(m)}$  as the exact solution in Problem 2.1,  $\hat{\mathbf{u}}^{(m)}$  as the scheme solution in Eq (2.4),  $\tilde{\mathbf{u}}^{(m)}$  as an intermediate solution, and  $\hat{\mathbf{u}}^{(m)}$  as the perturbed numerical solution. Then they have the following relations.

- By similar arguments in [8] and [16], from the given conditions, we have  $(h_1 \cdots h_d)^{1/2} \|\mathbf{u}^{(m)}\|_2 \leq B$ , and

$$\mathring{A}\mathbf{u}^{(m)} = \mathring{A}'\mathbf{u}^{(m-1)} + \tau\mathbf{f}^{(m-\frac{1}{2})} + \mathbf{R}^{(m-\frac{1}{2})}, \text{ with } (h_1 \cdots h_d)^{1/2} \|\mathbf{R}^{(m-\frac{1}{2})}\|_2 \leq K_0\tau(\tau^2 + \hat{h}^2 + \sigma^2)$$

for some constants  $B, K_0 > 0$  independent of  $m, \tau, h_k, \sigma$ .

- $\hat{\mathbf{u}}^{(0)} = \hat{\mathbf{u}}^{(0)}$ .
- $\mathring{A}\hat{\mathbf{u}}^{(m)} = \mathring{A}'\hat{\mathbf{u}}^{(m-1)} + \tau\mathbf{f}^{(m-\frac{1}{2})}$ .
- $\tilde{\mathbf{u}}^{(m)} = \mathring{A}^{-1}\mathring{A}'\hat{\mathbf{u}}^{(m-1)} + \tau\mathring{A}^{-1}\mathbf{f}^{(m-\frac{1}{2})}$ .
- $\|\tilde{\mathbf{u}}^{(m)} - \hat{\mathbf{u}}^{(m)}\|_F \leq \epsilon\|\tilde{\mathbf{u}}^{(m)}\|_F$ .

By similar arguments in [15], combining the relations with the norm property in inequality (Eq 2.5), we have

$$\begin{aligned} E_m &= (h_1 \cdots h_d)^{1/2} \|\mathbf{u}^{(m)} - \hat{\mathbf{u}}^{(m)}\|_2 \\ &\leq (h_1 \cdots h_d)^{1/2} \left( \|\mathbf{u}^{(m)} - \hat{\mathbf{u}}^{(m)}\|_2 + \|\hat{\mathbf{u}}^{(m)} - \tilde{\mathbf{u}}^{(m)}\|_2 + \|\tilde{\mathbf{u}}^{(m)} - \hat{\mathbf{u}}^{(m)}\|_2 \right) \\ &\leq (h_1 \cdots h_d)^{1/2} \left( (1 + \epsilon) \|\mathbf{u}^{(m)} - \hat{\mathbf{u}}^{(m)}\|_2 + (1 + \epsilon) \|\hat{\mathbf{u}}^{(m)} - \tilde{\mathbf{u}}^{(m)}\|_2 \right) + B\epsilon \\ &\leq 2(1 + \epsilon)K_0\tau(\tau^2 + \hat{h}^2 + \sigma^2) + (1 + \epsilon)E_{m-1} + B\epsilon \\ &\leq \dots \\ &\leq 2M(1 + \epsilon)^M K_0\tau(\tau^2 + \hat{h}^2 + \sigma^2) + (1 + \epsilon)^M E_0 + M(1 + \epsilon)^M B\epsilon \\ &\leq K(1 + \epsilon)^M (\tau^2 + \hat{h}^2 + \sigma^2 + M\epsilon), \end{aligned}$$

where  $K = \max\{2K_0T, B\} > 0$  is independent of  $m, \tau, h_k, \sigma$ .

This allows us to take  $\epsilon = 1/M^3$  such that the overall convergence is of second order.

**Remark 2.** In [15], the rounding perturbation analysis for a non-Crank-Nicolson scheme is demonstrated; while in [16], the ADI perturbation for a Crank-Nicolson scheme is analyzed, with the corresponding rounding perturbation analysis omitted. The above proof is supplemented for the rounding perturbation analysis for a Crank-Nicolson scheme.

#### 4. Results and discussion

In this section, some numerical examples are presented to test the methods described above. For simplicity, we set  $[x_a^{(k)}, x_b^{(k)}] = [0, 1]$  for  $1 \leq k \leq d$ ,  $[0, T] = [0, 1]$ , and  $S = M = N$ . We will show the numerical errors (Error), corresponding convergence rates (Rate), CPU times in the whole process (CPU(s)), CPU times in rounding process (rCPU(s)), and (mean, mode, maximum) of the involving TT-ranks ( $M_r$ ).

The numerical examples are tested in MATLAB R2014b with the configuration: Intel(R) Core(TM) i5-8300H CPU 2.30 GHz and 8 GB RAM.



**Example 1.** This example is tested with a TT-ranks 1 exact solution, with

$$u(\mathbf{x}, t) = e^{-t} \prod_{k=1}^d 20(x^{(k)})^2(1 - x^{(k)})^2,$$

$$P(\alpha) = 2\Gamma(5 - \alpha) \cos\left(\frac{\pi\alpha}{2}\right), \quad Q(\beta) = -2\Gamma(5 - \beta) \cos\left(\frac{\pi\beta}{2}\right),$$

$$f(\mathbf{x}, t) = e^{-t} \sum_{k=1}^d 20g(x^{(k)}) \prod_{\ell \neq k} 20(x^{(\ell)})^2(1 - x^{(\ell)})^2 - e^{-t} \prod_{k=1}^d 20(x^{(k)})^2(1 - x^{(k)})^2,$$

$$u_0(\mathbf{x}) = \prod_{k=1}^d 20(x^{(k)})^2(1 - x^{(k)})^2,$$

where

$$g(x) = 2 \left[ \frac{2x^2 - 2x}{\ln^3(1-x)} + \frac{-6x^3 + 5x^2 + 3x - 2}{\ln^2(1-x)} + \frac{6x(2x^3 - 2x^2 - x + 1)}{\ln(1-x)} \right. \\ \left. + \frac{2x(x-1)}{\ln^3 x} + \frac{x(6x^2 - 13x + 5)}{\ln^2 x} + \frac{6x(x-1)(2x^2 - 4x + 1)}{\ln x} \right] \\ - 2 \left[ \frac{-2x}{\ln^3(1-x)} + \frac{6x^2 - x - 2}{\ln^2(1-x)} + \frac{2(-6x^3 + 3x^2 + 3x - 1)}{\ln(1-x)} \right. \\ \left. + \frac{2(x-1)}{\ln^3 x} + \frac{6x^2 - 11x + 3}{\ln^2 x} + \frac{2(6x^3 - 15x^2 + 9x - 1)}{\ln x} \right]$$

The performance is summarized in Table 1, with “Error” column showing  $E_M$  at  $t_M = 1$ .

**Example 2.** This example is tested with an exact solution of TT-ranks 3, with

$$u(\mathbf{x}, t) = e^{-t} G_1(x^{(1)}) G_2(x^{(2)}) \cdots G_{d-1}(x^{(d-1)}) G_d(x^{(d)}),$$

$$P(\alpha) = \frac{2\Gamma(9 - \alpha)}{\Gamma(9)} \cos\left(\frac{\pi\alpha}{2}\right), \quad Q(\beta) = -\frac{2\Gamma(9 - \beta)}{\Gamma(8)} \cos\left(\frac{\pi\beta}{2}\right)$$

$$f(\mathbf{x}, t) = -e^{-t} \begin{bmatrix} G'_1(x^{(1)}) & G_1(x^{(1)}) \end{bmatrix} \begin{bmatrix} G_2(x^{(2)}) & 0 \\ G'_2(x^{(2)}) & G_2(x^{(2)}) \end{bmatrix} \cdots \\ \cdots \begin{bmatrix} G_{d-1}(x^{(d-1)}) & 0 \\ G'_{d-1}(x^{(d-1)}) & G_{d-1}(x^{(d-1)}) \end{bmatrix} \begin{bmatrix} G_d(x^{(d)}) \\ G'_d(x^{(d)}) + G_d(x^{(d)}) \end{bmatrix}$$

$$u_0(\mathbf{x}) = G_1(x^{(1)}) G_2(x^{(2)}) \cdots G_{d-1}(x^{(d-1)}) G_d(x^{(d)}),$$

where

$$G_1(x^{(1)}) = \begin{bmatrix} y^{[1,2]} & y^{[1,3]} & y^{[1,4]} \end{bmatrix}, \quad G_k(x^{(k)}) = \begin{bmatrix} y^{[k,2]} & y^{[k,3]} & y^{[k,4]} \\ y^{[k,3]} & y^{[k,2]} & y^{[k,3]} \\ y^{[k,4]} & y^{[k,3]} & y^{[k,2]} \end{bmatrix}, \quad G_d(x^{(d)}) = \begin{bmatrix} y^{[d,2]} \\ y^{[d,3]} \\ y^{[d,4]} \end{bmatrix},$$

for  $2 \leq k \leq d-1$ ,  $y^{[k,\ell]} = 10(x^{(k)})^\ell(1 - x^{(k)})^\ell$  for  $1 \leq k \leq d$ ,  $3 \leq \ell \leq 5$ , and

$$G'_k(x^{(k)}) = \int_0^1 P(\alpha) \frac{\partial^\alpha G_k(x^{(k)})}{\partial |x^{(k)}|^\alpha} d\alpha + \int_1^2 Q(\beta) \frac{\partial^\beta G_k(x^{(k)})}{\partial |x^{(k)}|^\beta} d\beta$$

which is defined entrywisely.

As the closed form of  $G'_k(x^{(k)})$  is numerically unstable, each entry of  $G'_k(x^{(k)})$  is first reduced to the following form, then the remaining integrals are approximated by the midpoint quadrature rule.

$$\int_0^1 P(\alpha) \frac{\partial^\alpha y^{[k,\ell]}}{\partial |x^{(k)}|^\alpha} d\alpha + \int_1^2 Q(\beta) \frac{\partial^\beta y^{[k,\ell]}}{\partial |x^{(k)}|^\beta} d\beta$$

$$= -10 \sum_{r=0}^{\ell} (-1)^r \binom{\ell}{r} \frac{\Gamma(\ell+r+1)}{\Gamma(9)} \sum_{r'=0}^{8-\ell-r} c_{\ell+r+1,r'} \int_0^1 \alpha^{r'} \left( (x^{(k)})^{\ell+r-\alpha} + (1-x^{(k)})^{\ell+r-\alpha} \right) d\alpha$$

$$+ 10 \sum_{r=0}^{\ell} (-1)^r \binom{\ell}{r} \frac{\Gamma(\ell+r+1)}{\Gamma(8)} \sum_{r'=0}^{8-\ell-r} c_{\ell+r+1,r'} \int_1^2 \beta^{r'} \left( (x^{(k)})^{\ell+r-\beta} + (1-x^{(k)})^{\ell+r-\beta} \right) d\beta$$

with  $\Gamma(9-\gamma)/\Gamma(\ell'-\gamma) = \sum_{r'=0}^{\ell'-\ell} c_{\ell',r'} \gamma^{r'}$ ,  $\gamma = \alpha, \beta$ .

The performance is summarized in Table 2, with “Error” column showing  $E_M$  at  $t_M = 1$ .

**Example 3.** This example is tested with uncertain exact solution, and the same  $u_0(\mathbf{x})$  and  $f(\mathbf{x}, t)$  of TT-ranks 3 as in Example 3 in [15], with

$$P(\alpha) = 2\Gamma(5-\alpha) \cos\left(\frac{\pi\alpha}{2}\right), \quad Q(\beta) = -2\Gamma(5-\beta) \cos\left(\frac{\pi\beta}{2}\right).$$

The performance is summarized in Table 3, with “Error” column showing  $E_M$  at  $t_M = 1$ , assuming the numerical solution of the finest grid as the exact solution.

**Table 1.** Performance in Example 1.

$d = 5$						$d = 10$					
$N$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	
$2^5$	1.67E-05	-	0.37	0.04	(3.48, 3, 4)	6.21E-06	-	0.45	0.12	(3.75, 4, 4)	
$2^6$	4.16E-06	2.00	0.17	0.10	(3.49, 3, 4)	1.60E-06	1.96	0.49	0.32	(5.29, 6, 6)	
$2^7$	1.04E-06	2.01	0.60	0.36	(4.97, 4, 6)	4.02E-07	1.99	1.28	0.88	(5.75, 6, 6)	
$2^8$	2.58E-07	2.00	2.09	1.25	(5.48, 5, 6)	1.01E-07	2.00	5.17	3.24	(6.52, 7, 7)	
$2^9$	6.45E-08	2.00	9.31	3.99	(5.49, 5, 6)	2.52E-08	2.00	23.53	11.12	(6.76, 7, 7)	
$2^{10}$	1.61E-08	2.00	64.06	23.14	(5.49, 5, 6)	6.30E-09	2.00	172.49	75.74	(7.42, 8, 8)	
$2^{11}$	4.05E-09	2.00	441.25	109.79	(5.98, 5, 7)	1.57E-09	2.00	1209.30	405.57	(8.28, 9, 9)	
$2^{12}$	1.02E-09	1.98	2964.89	446.61	(5.99, 5, 7)	3.93E-10	2.00	7329.04	1678.53	(8.42, 9, 9)	
$d = 15$						$d = 20$					
$N$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	
$2^5$	6.99E-07	-	0.24	0.13	(3.83, 4, 4)	5.69E-08	-	0.32	0.18	(3.87, 4, 4)	
$2^6$	1.86E-07	1.91	0.76	0.53	(5.52, 6, 6)	1.58E-08	1.85	1.05	0.75	(5.53, 6, 6)	
$2^7$	4.72E-08	1.98	2.16	1.52	(5.83, 6, 6)	4.06E-09	1.96	3.35	2.35	(6.63, 7, 7)	
$2^8$	1.19E-08	1.99	8.13	5.30	(6.83, 7, 7)	1.02E-09	1.99	11.08	7.21	(6.87, 7, 7)	
$2^9$	2.97E-09	2.00	43.12	21.69	(7.61, 8, 8)	2.56E-10	2.00	68.53	36.76	(8.48, 9, 9)	
$2^{10}$	7.42E-10	2.00	345.27	167.96	(8.53, 9, 9)	6.40E-11	2.00	449.05	220.34	(8.71, 9, 9)	
$2^{11}$	1.86E-10	2.00	1964.05	686.63	(8.69, 9, 9)	1.60E-11	2.00	2598.83	924.04	(8.77, 9, 9)	
$2^{12}$	4.64E-11	2.00	11496.46	2760.87	(8.70, 9, 9)	4.00E-12	2.00	17432.30	4701.29	(9.71, 10, 10)	

**Table 2.** Performance in Example 2.

$d = 5$						$d = 10$					
$N$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	
$2^5$	6.25E-02	-	0.34	0.03	(1.98, 2, 3)	3.57E-02	-	0.07	0.04	(1.97, 2, 2)	
$2^6$	2.60E-02	1.26	0.15	0.08	(3.45, 3, 4)	2.60E-02	0.46	0.29	0.17	(2.97, 3, 3)	
$2^7$	7.98E-03	1.71	0.36	0.20	(4.45, 4, 5)	1.21E-02	1.11	0.67	0.38	(3.97, 4, 4)	
$2^8$	2.13E-03	1.91	1.39	0.77	(5.96, 5, 7)	3.88E-03	1.64	2.85	1.70	(5.74, 6, 6)	
$2^9$	5.42E-04	1.97	8.96	3.49	(7.47, 9, 9)	1.05E-03	1.89	21.86	9.67	(8.30, 9, 9)	
$2^{10}$	1.36E-04	1.99	72.06	24.44	(9.34, 11, 11)	2.68E-04	1.97	195.19	77.67	(10.30, 11, 11)	
$2^{11}$	3.40E-05	2.00	543.10	139.50	(10.48, 8, 13)	6.74E-05	1.99	1606.92	511.30	(12.86, 14, 14)	
$2^{12}$	8.51E-06	2.00	3724.36	687.59	(11.49, 9, 14)	1.69E-05	2.00	11369.61	2906.70	(14.42, 16, 16)	
$d = 15$						$d = 20$					
$N$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	
$2^5$	1.21E-02	-	0.09	0.05	(1.97, 2, 2)	3.91E-03	-	0.33	0.08	(1.97, 2, 2)	
$2^6$	1.10E-02	0.14	0.30	0.18	(2.00, 2, 3)	3.81E-03	0.04	0.44	0.26	(1.98, 2, 2)	
$2^7$	7.03E-03	0.64	0.94	0.55	(3.83, 4, 4)	2.99E-03	0.35	1.08	0.62	(2.98, 3, 3)	
$2^8$	2.81E-03	1.32	4.52	2.72	(5.68, 6, 6)	1.46E-03	1.03	5.63	3.38	(4.82, 5, 5)	
$2^9$	8.29E-04	1.76	31.15	13.69	(7.61, 8, 8)	4.79E-04	1.61	38.31	15.51	(6.76, 7, 7)	
$2^{10}$	2.17E-04	1.93	301.84	121.85	(10.26, 11, 11)	1.30E-04	1.88	379.80	153.41	(9.66, 10, 10)	
$2^{11}$	5.50E-05	1.98	2312.51	728.29	(12.34, 13, 13)	3.32E-05	1.97	3315.56	1066.01	(12.24, 13, 13)	
$2^{12}$	1.38E-05	2.00	19495.91	5279.55	(14.91, 16, 16)	8.36E-06	1.99	24377.55	6392.04	(14.46, 15, 15)	

**Table 3.** Performance in Example 3.

$d = 5$						$d = 10$					
$N$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	
$2^5$	4.85E-03	-	0.29	0.10	(4.18, 4, 7)	7.92E-03	-	0.21	0.05	(3.00, 3, 4)	
$2^6$	2.34E-03	1.05	0.24	0.11	(5.45, 6, 7)	6.15E-03	0.36	0.32	0.17	(3.97, 4, 5)	
$2^7$	8.35E-04	1.49	0.78	0.51	(9.31, 8, 12)	3.16E-03	0.96	1.35	0.89	(6.85, 7, 8)	
$2^8$	2.47E-04	1.76	5.02	3.08	(13.72, 12, 17)	1.12E-03	1.50	8.30	5.25	(12.21, 13, 14)	
$2^9$	6.71E-05	1.88	67.28	38.93	(19.57, 17, 24)	3.18E-04	1.82	121.94	69.83	(18.16, 20, 20)	
$2^{10}$	1.77E-05	1.92	473.79	242.42	(25.96, 31, 32)	7.58E-05	2.07	1140.53	613.71	(25.61, 28, 29)	
$2^{11}$	-	-	3887.31	1901.15	(33.29, 28, 40)	-	-	10220.83	5018.83	(34.22, 38, 38)	
$d = 15$						$d = 20$					
$N$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	Error	Rate	CPU(s)	rCPU(s)	$M_r$	
$2^5$	1.21E-02	-	1.19	0.23	(2.97, 3, 4)	2.05E-02	-	0.70	0.14	(2.97, 3, 4)	
$2^6$	1.14E-02	0.09	0.64	0.35	(3.64, 4, 4)	2.03E-02	0.01	0.48	0.31	(2.95, 3, 4)	
$2^7$	7.95E-03	0.52	2.00	1.34	(6.45, 7, 7)	1.70E-02	0.25	1.61	1.05	(3.93, 4, 5)	
$2^8$	3.53E-03	1.17	8.59	5.34	(9.45, 10, 11)	9.41E-03	0.86	9.31	5.98	(7.78, 8, 9)	
$2^9$	1.09E-03	1.69	154.03	90.33	(15.94, 17, 17)	3.31E-03	1.51	121.22	73.28	(13.39, 14, 15)	
$2^{10}$	2.58E-04	2.08	1591.50	845.58	(23.85, 26, 26)	8.08E-04	2.03	1682.07	863.74	(21.49, 23, 24)	
$2^{11}$	-	-	14325.51	6792.25	(32.46, 35, 35)	-	-	16035.60	7337.32	(29.86, 32, 32)	

In the tables, second order convergence is observed, agreeing with the perturbation analysis. In Tables 1 and 2, the  $M_r$  illustrate that the problems with low TT-ranks exact solutions may have numerical solutions of higher TT-ranks, yet the ranks still grow steadily of about  $O(\log N)$ , and the rounding process occupy about one-fourth of the whole solving process. While in Table 3, there seems to be a nonlinear growth of  $O(\log N)$  in the TT-ranks, and the rounding process occupy about half of the whole solving process. These agree with the discussion about the effect of the TT-ranks in operation cost. Besides, the deviations of TT-ranks of the numerical solutions from the exact solutions also happen in the related works [15] and [16] where the ADI method is applied, implicating that the ADI treatment is a potential factor for this phenomenon. The growth of TT-ranks is to be furthered studied.

## 5. Conclusions

In this paper, high dimensional RSDO-ADEs are discretized with a second order ADI scheme, and TT-format method is introduced to solve the scheme in compressed form. The perturbation error analysis is performed, and it is claimed that under certain conditions, with rounding relative error  $\epsilon = 1/N^3$ , Algorithm 3.1 can maintain the second order convergence of the scheme. Numerical experiments with low TT-ranks data are conducted, the results agree with the claim. Meanwhile, the TT-ranks appear to be indicating a growth of power of  $\log N$  or other nonlinear function of  $\log N$ , hence implying a possible dominance of the rounding process over the whole solving process. Further studies about the TT-ranks of the numerical solutions are to be made.

## Acknowledgments

This work was supported by University of Macau [MYRG2020-00208-FST, MYRG2018-00025-FST], the Science and Technology Development Fund, Macau SAR under Funding Scheme for Postdoctoral Researchers of Higher Education Institutions 2021 (File No. 0032/APD/2021).

## Conflict of interest

All authors declare no conflicts of interest in this paper.

## References

1. A. V. Chechkin, J. Klafter, I. M. Sokolov, Fractional Fokker-Planck equation for ultraslow kinetics, *Europhys. Lett.*, **63** (2003), 326–332. <https://doi.org/10.1209/epl/i2003-00539-0>
2. Y. G. Sinai, The limiting behavior of a one-dimensional random walk in a random medium, *Theory Probab. Appl.*, **27** (1983), 256–268. <https://doi.org/10.1137/1127028>
3. M. M. Meerschaert, E. Nane, P. Vellaisamy, Distributed-order fractional diffusions on bounded domains, *J. Math. Anal. Appl.*, **379** (2011), 216–228. <https://doi.org/10.1016/j.jmaa.2010.12.056>
4. M. Caputo, Distributed order differential equations modelling dielectric induction and diffusion, *Fract. Calc. Appl. Anal.*, **4** (2001), 421–442.

5. X. Hu, F. Liu, V. Anh, I. Turner, A numerical investigation of the time distributed-order diffusion model, *Anziam J.*, **5** (2014), C464–C478. <https://doi.org/10.21914/anziamj.v55i0.7888>
6. J. Jia, H. Wang, A fast finite difference method for distributed-order space-fractional partial differential equations on convex domains, *Comput. Math. Appl.*, **75** (2018), 2031–2043. <https://doi.org/10.1016/j.camwa.2017.09.003>
7. W. Fan, F. Liu, A numerical method for solving the two-dimensional distributed order space-fractional diffusion equation on an irregular convex domain, *Appl. Math. Lett.*, **77** (2018), 114–121. <https://doi.org/10.1016/j.aml.2017.10.005>
8. X. Wang, F. Liu, X. Chen, Novel second-order accurate implicit numerical methods for the Riesz space distributed-order advection-dispersion equations, *Adv. Math. Phys.*, **2015** (2015), 1–14. <https://doi.org/10.1155/2015/590435>
9. I. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.*, **33** (2011), 2295–2317. <https://doi.org/10.1137/090752286>
10. D. Bertaccini, F. Durastante, Block structured preconditioners in tensor form for the all-at-once solution of a finite volume fractional diffusion equation, *Appl. Math. Lett.*, **95** (2019), 92–97. <https://doi.org/10.1016/j.aml.2019.03.028>
11. T. Breiten, V. Simoncini, M. Stoll, Low-rank solvers for fractional differential equations, *Electron. Trans. Numer. Anal.*, **45** (2016), 107–132. <https://doi.org/10.17617/2.2270973>
12. S. Dolgov, J. Pearson, D. Savostyanov, M. Stoll, Fast tensor product solvers for optimization problems with fractional differential equations as constraints, *Appl. Math. Comput.*, **273** (2016), 604–623. <https://doi.org/10.1016/j.amc.2015.09.042>
13. I. Oseledets, E. Tyrtyshnikov, N. Zamarashkin, Tensor-train ranks for matrices and their inverses, *Comput. Methods Appl. Math.*, **11** (2011), 394–403. <https://doi.org/10.2478/cmam-2011-0022>
14. V. Kazeev, B. Khoromskij, E. Tyrtyshnikov, Multilevel Toeplitz matrices generated by tensor-structured vectors and convolution with logarithmic complexity, *SIAM J. Sci. Comput.*, **35** (2013), A1511–A1536. <https://doi.org/10.1137/110844830>
15. L. Chou, S. Lei, Tensor-train format solution with preconditioned iterative method for high dimensional time-dependent space-fractional diffusion equations with error analysis, *J. Sci. Comput.*, **80** (2019), 1731–1763. <https://doi.org/10.1007/s10915-019-00994-3>
16. L. Chou, S. Lei, Finite volume approximation with ADI scheme and low-rank solver for high dimensional spatial distributed-order fractional diffusion equations, *Comput. Math. Appl.*, **89** (2021), 116–126. <https://doi.org/10.1016/j.camwa.2021.02.014>
17. W. Deng, B. Li, W. Tian, P. Zhang, Boundary problems for the fractional and tempered fractional operators, *Multiscale Model. Simul.*, **16** (2018), 125–149. <https://doi.org/10.1137/17M1116222>
18. C. Çelik, M. Duman, Crank-Nicolson method for the fractional diffusion equation with the Riesz fractional derivative, *J. Comput. Phys.*, **231** (2012), 1743–1750. <https://doi.org/10.1016/j.jcp.2011.11.008>
19. I. Gohberg, V. Olshevsky, Circulants, displacements and decompositions of matrices, *Integr. Equat. Oper. Theory*, **15** (1992), 730–743. <https://doi.org/10.1007/bf01200697>

- 
20. R. Chan, M. Ng, Conjugate gradient methods for Toeplitz systems, *SIAM Rev.*, **38** (1996), 427–482. <https://doi.org/10.1137/S0036144594276474>



AIMS Press

©2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)