AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(5): 8471–8490.
DOI: 10.3934/math.2022472
Received: 31 December 2021
Revised: 12 February 2022
Accepted: 21 February 2022
Published: 28 February 2022

Research article

Approximate solutions of the 2D space-time fractional diffusion equation via a gradient-descent iterative algorithm with Grünwald-Letnikov approximation

Adisorn Kittisopaporn[1] **and Pattrawut Chansangiam**[1,*]

[1] Department of Mathematics, School of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand.

* **Correspondence:** Email: pattrawut.ch@kmitl.ac.th; Tel: +66935266600.

Abstract: We consider the two-dimensional space-time fractional differential equation with the Caputo's time derivative and the Riemann-Liouville space derivatives on bounded domains. The equation is subjected to the zero Dirichlet boundary condition and the zero initial condition. We discretize the equation by finite difference schemes based on Grünwald-Letnikov approximation. Then we linearize the discretized equations into a sparse linear system. To solve such linear system, we propose a gradient-descent iterative algorithm with a sequence of optimal convergence factor aiming to minimize the error occurring at each iteration. The convergence analysis guarantees the capability of the algorithm as long as the coefficient matrix is invertible. In addition, the convergence rate and error estimates are provided. Numerical experiments demonstrate the efficiency, the accuracy and the performance of the proposed algorithm.

Keywords: fractional derivatives; fractional diffusion equation; Grünwald-Letnikov approximation; gradient descent; iterative method
Mathematics Subject Classification: 65F10, 65N22, 34A08, 15A60

1. Introduction

In modern days, fractional differential equations (FDEs) play an important role in applied mathematics, science, and engineering; see e.g. [1–4]. FDEs are powerful to model many physical phenomena due to a noninteger order in time and space.

Let us consider the standard two-dimensional (2D) diffusion equation (see e.g. [5]).

$$\frac{\partial u(\mathbf{x},t)}{\partial t} - \Delta u(\mathbf{x},t) \;=\; 0, \tag{1.1}$$

where $\mathbf{x} \in V \subseteq \mathbb{R}^2$ and $t \in [0, T] \subseteq \mathbb{R}^+$. Here, Δ denotes the Laplacian operator with respect to the spatial variable $\mathbf{x} = (x, y) \in \mathbb{R}^2$. In physical contexts, the function $u(\mathbf{x}, t)$ represents the density of a diffusing material at location $\mathbf{x}$ and time t. Let us denote the Gamma function by Γ. From Eq (1.1), if we replace the first-order time derivative with the Caputo's derivative [2]

$$ {}^{C}D_t^1 u(x,y,t) \;=\; \frac{\partial u}{\partial t}(x,y,t), \text{ and } {}^{C}D_t^{\beta} u(x,y,t) \;=\; \frac{1}{\Gamma(1-\beta)} \int_0^t \frac{\partial u(x,y,\tau)}{\partial \tau} \frac{d\tau}{(t-\tau)^{\beta}} \text{ for } 0 < \beta < 1, $$

and replace the second-order space derivative with the Riemann-Liouville derivatives [6].

$$ {}^{RL}D_x^2 u(x,y,t) \;=\; \frac{\partial^2 u}{\partial x^2}(x,y,t), \text{ and } {}^{RL}D_x^{\alpha_1} u(x,y,t) \;=\; \frac{1}{\Gamma(2-\alpha_1)} \frac{\partial^2}{\partial x^2} \int_0^x \frac{u(\xi,y,t)}{(x-\xi)^{\alpha_1-1}}\, d\xi \text{ for } 1 < \alpha_1 < 2, $$

and similarly for ${}^{RL}D_y^{\alpha_2} u(x,y,t)$, then we get the 2D space-time fractional diffusion equation (e.g. [7]).

$$ {}^{C}D_t^{\beta} u(x,y,t) \;-\; {}^{RL}D_x^{\alpha_1} u(x,y,t) - {}^{RL}D_y^{\alpha_2} u(x,y,t) \;=\; f(x,y,t). \tag{1.2} $$

Here, we impose the function $f(x, y, t)$ on the right hand side of the equation. Physical interpretations restrict the fractional orders so that $\alpha_1 \in (1, 2]$, $\alpha_2 \in (1, 2]$, and $\beta \in (0, 1]$. When $\beta = 1$, Eq (1.2) is known as the 2D time fractional diffusion equation.

In recent years, there are many studies about theory of (1D or 2D) space-time (or time) fractional diffusion equations and related equations, e.g. [8–13]. These differential equations were successfully used to describe many physical phenomena, e.g. random walk models [14–17], and anomalous diffusion processes [18–21].

In general, FDEs do not posses exact solutions in closed froms. Thus, numerical methods have been implemented for several types of FDEs, e.g. the variational iteration method [22], the homotopy analysis method [23, 24], Adomian decomposition method [25, 26], B-spline collocation schemes [27–30], and the collocation method based on fractional Legendre functions [31]. There are also numerical methods using either finite difference or finite element methods to a discretization of certain FDEs; see e.g. [32–36]. Finite difference techniques based on Grünwald-Letnikov approximation for fractional derivatives were investigated by many authors, e.g. [1, 37, 38].

The main objective of the present paper is to propose an iterative algorithm to produce well-approximated solutions of the 2D space-time fractional diffusion Eq (1.2), subjected to the zero Dirichlet boundary condition and the zero initial condition. First, we discretize Eq (1.2) via Grünwald-Letnikov approximation for the Riemann-Liouville and the Caputo derivatives (see Section 3). Then, we linearize the discretized equations into a linear system with a sparse cofficient matrix. In particular, we discuss the numerical scheme for its interesting special case, namely, the 2D time fractional diffusion equation. To solve the linear system, we apply the gradient-descent technique to derive an iterative procedure with suitable directions and step sizes (see Section 4). We show that the produced approximate solutions converge to a unique solution for any given initial vector (see Section 5). Theoretical performance of the proposed algorithm are discussed through the convergence rate and error estimates. We verify the capability and theoretical performance of the proposed algorithm by making two numerical experiments (see Section 6). We compare the performance of the proposed algorithm with well-known iterative methods for linear systems in the literature, e.g. GI, LSI, SOR, and JOR algorithms.

2. Preliminaries

In this section, we recall relevant background that used in later discussions.

2.1. Grünwald-Letnikov approximation for fractional derivatives

The discretization of fractional derivatives is often done by finite difference schemes based on Grünwald-Letnikov type (see e.g. [37,38]). For the 2D problem, we have the following approximation for Riemann-Liouville fractional derivatives of order $\alpha > 0$:

$$
\begin{aligned}
{}^{RL}D_x^{\alpha} u(x, y, t) &= \lim_{N_x \to \infty} \frac{1}{h_x^{\alpha}} \sum_{k=0}^{N_x} g_{\alpha,k} u(x - (k-1)h_x, y, t), && (2.1)\\
{}^{RL}D_y^{\alpha} u(x, y, t) &= \lim_{N_y \to \infty} \frac{1}{h_y^{\alpha}} \sum_{k=0}^{N_y} g_{\alpha,k} u(x, y - (k-1)h_y, t), && (2.2)
\end{aligned}
$$

where the coefficients $g_{\alpha,k}$ are defined as

$$
g_{\alpha,k} = \frac{\Gamma(k-\alpha)}{\Gamma(-\alpha)\Gamma(k+1)}.
$$

Alternatively, we have the recusive formula for $g_{\alpha,k}$:

$$
g_{\alpha,0} = 1, \quad g_{\alpha,k} = \Big(1 - \frac{\alpha+1}{k}\Big) g_{\alpha,k-1}.
$$

We can compute the Grünwald-Letnikov approximation for Caputo's fractional derivatives via the difference formula.

$$
{}^{RL}D_t^{\alpha} f - {}^{C}D_t^{\alpha} f = \sum_{\nu=0}^{\lfloor \alpha \rfloor} r_\nu^{\alpha}(t) f^{(\nu)}(0), \quad \text{where} \quad r_\nu^{\alpha}(t) = \frac{t^{\nu-\alpha}}{\Gamma(\nu+1-\alpha)}. \qquad (2.3)
$$

Here, $\lfloor \cdot \rfloor$ denotes the floor function. When $\alpha \in (0, 1]$, the correction term on the right hand side of (2.3) is equal to zero when we consider the zero initial value, i.e., $u(x, y, t = 0) = 0$; see e.g. [38].

2.2. Mittag-Leffler functions

The Mittag-Leffler function $E_{a,b}$ on two parameters $a, b > 0$ is a generalization of the exponential function defined by the Taylor series.

$$
E_{a,b}(x) = \sum_{k=0}^{\infty} \frac{x^k}{\Gamma(ak+b)}.
$$

Mittag-Leffler functions arise naturally in certain fractional differential equations, in particular, in our numerical examples (Section 6).

2.3. Auxiliary results from matrix analysis

All matrices and vectors considered throughout this paper are real. The Frobenius norm of a rectangular matrix A is defined by $\|A\|_F = \sqrt{\operatorname{tr}(A^T A)}$. The Frobenius norm of a vector coincides with the Euclidean vector norm. The spectral norm $\|\cdot\|_2$ of a square matrix is defined to be the largest singular value of the matrix, or equivalently, the matrix norm induced by the Euclidean vector norm. Thus, we have the following relation (see e.g. [39]).

$$\|Ax\|_F \leqslant \|A\|_2 \, \|x\|_F \tag{2.4}$$

for any square matrix A and vector x of conformable sizes. The condition number $\kappa(A)$ of a square matrix A is defined to be the ratio between the largest and the smallest singular values of A. In particular, if A is invertible, we have

$$\kappa(A) = \|A\|_2 \, \|A^{-1}\|_2. \tag{2.5}$$

We recall gradient formulas for the trace of certain matrix products:

Lemma 2.1. *(e.g. [39]) For matrices A, B, X of compatible sizes, we have*

$$\begin{aligned} \frac{d}{dX}\operatorname{tr}(AX) &= A^T, \\ \frac{d}{dX}\operatorname{tr}(XAX^TB) &= BXA + B^TXA^T. \end{aligned}$$

3. Discretization and linearization of the 2D space-time fractional diffusion equation

In this section, we consider the 2D space-time fractional diffusion equation, and its interesting special case, namely, the 2D space fractional diffusion equation. We discretize these differential equations in which the Riemann-Liouville and the Caputo' derivatives are approximated by Grünwald-Letnikov approximation. Then we form a linear system from the discretized equations.

Let us consider the 2D space-time fractional fractional diffusion equation of the form

$${}^{C}D_t^{\beta}u(x,y,t) - {}^{RL}D_x^{\alpha_1}u(x,y,t) - {}^{RL}D_y^{\alpha_2}u(x,y,t) = f(x,y,t), \tag{3.1}$$

on bounded domains $x \in [a_x, b_x]$, $y \in [a_y, b_y]$, and $t \in [0, T]$. The equation is subjected to the zero Dirichlet boundary condition

$$u(x = a_x, y, t) = u(x = b_x, y, t) = u(x, y = a_y, t) = u(x, y = b_y, t) = 0,$$

and the zero initial condition, i.e., $u(x, y, t = 0) = 0$.

3.1. The 2D space-time fractional diffusion equation

Consider Eq (3.1) with fractional orders $\beta \in (0, 1]$ and $\alpha_1, \alpha_2 \in (1, 2]$. We discretize $x_i = a_x + ih_x$, $y_j = a_y + jh_y$ and $t_k = kh_t$ where h_x, h_y and h_t are defined by

$$h_x = \frac{b_x - a_x}{N_x}, \; h_y = \frac{b_y - a_y}{N_y}, \; \text{ and } h_t = \frac{T}{N_t}. \tag{3.2}$$

The approximated solution $u(x_i, y_j, t_k)$ at a point (x_i, y_j, t_k) is written by u_{ijk}. For convenience, we define

$$\xi_l = \frac{g_{\beta,l}}{g_{\beta,0}},\ \varphi_p = \frac{g_{\alpha_1,p}h_t^\beta}{g_{\beta,0}h_x^{\alpha_1}},\ \psi_q = \frac{g_{\alpha_2,q}h_t^\beta}{g_{\beta,0}h_y^{\alpha_2}} \text{ and } \sigma = \frac{h_t^\beta}{g_{\beta,0}}.$$

From the difference Formula (2.3) and the Grünwald-Letnikov Formulae (2.1) and (2.2), we obtain

$$\frac{1}{h_t^\beta}\sum_{l=0}^{k+1} g_{\beta,l}u_{i,j,k-l+1} - \frac{1}{h_x^{\alpha_1}}\sum_{p=0}^{i+1} g_{\alpha_1,p}u_{i-p+1,j,k+1} - \frac{1}{h_y^{\alpha_2}}\sum_{q=0}^{j+1} g_{\alpha_2,q}u_{i,j-q+1,k+1} = f_{i,j,k+1},$$

or equivalently,

$$u_{i,j,k+1} + \sum_{l=1}^{k+1} \xi_l u_{i,j,k-l+1} - \sum_{p=0}^{i+1} \varphi_p u_{i-p+1,j,k+1} - \sum_{q=0}^{j+1} \psi_q u_{i,j-q+1,k+1} = \sigma f_{i,j,k+1},$$

for each $i \in \{1,\dots,N_x-1\}$, $j \in \{1,\dots,N_y-1\}$ and $k \in \{1,\dots,N_t\}$. For convenience, let us denote $N_{xy} = (N_x-1)(N_y-1)$ and $N = N_{xy}N_t$. Thus, Eq (1.2) can be linearized into a system of N linear equations in N unknowns $u_{111},\dots,u_{N_x-1,N_y-1,N_t}$.

To form a linear system, let

$$U = \begin{bmatrix} u_{1,1,1} & u_{1,1,2} & \cdots & u_{1,1,N_t} \\ \vdots & \vdots & \cdots & \vdots \\ u_{1,N_y-1,1} & u_{1,N_y-1,2} & \cdots & u_{1,N_y-1,N_t} \\ \vdots & \vdots & \cdots & \vdots \\ u_{N_x-1,N_y-1,1} & u_{N_x-1,N_y-1,2} & \cdots & u_{N_x-1,N_y-1,N_t} \end{bmatrix}, F = \begin{bmatrix} f_{1,1,1} & f_{1,1,2} & \cdots & f_{1,1,N_t} \\ \vdots & \vdots & \cdots & \vdots \\ f_{1,N_y-1,1} & f_{1,N_y-1,2} & \cdots & f_{1,N_y-1,N_t} \\ \vdots & \vdots & \cdots & \vdots \\ f_{N_x-1,N_y-1,1} & f_{N_x-1,N_y-1,2} & \cdots & f_{N_x-1,N_y-1,N_t} \end{bmatrix}.$$

Then put $\mathbf{u} = \mathrm{Vec}(U)$ and $\mathbf{f} = \sigma\,\mathrm{Vec}(F)$. Here, $\mathrm{Vec}(\cdot)$ is an operator that turns a matrix into a column vector by stacking columns of the matrix consecutively; see e.g. [40, Ch. 4]. Hence, we obtain a linear system

$$P\mathbf{u} = \mathbf{f}, \tag{3.3}$$

where the coefficient P is a block lower-triangular matrix

$$P = \begin{bmatrix} A & 0 & 0 & \cdots & 0 \\ \xi_1 I_{N_{xy}} & A & 0 & \cdots & 0 \\ \xi_2 I_{N_{xy}} & \xi_1 I_{N_{xy}} & A & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \xi_{N_t-1} I_{N_{xy}} & \cdots & \xi_2 I_{N_{xy}} & \xi_1 I_{N_{xy}} & A \end{bmatrix},$$

and

$$A = \begin{bmatrix} B & -\varphi_0 I_{N_y-1} & 0 & \cdots & 0 \\ -\varphi_2 I_{N_y-1} & B & -\varphi_0 I_{N_y-1} & \cdots & 0 \\ -\varphi_3 I_{N_y-1} & -\varphi_2 I_{N_y-1} & B & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\varphi_{N_x-1} I_{N_y-1} & \cdots & -\varphi_3 I_{N_y-1} & -\varphi_2 I_{N_y-1} & B \end{bmatrix},$$

$$B = \begin{bmatrix} 1-\varphi_1-\psi_1 & -\psi_0 & 0 & \cdots & 0 \\ -\psi_2 & 1-\varphi_1-\psi_1 & -\psi_0 & \cdots & 0 \\ -\psi_3 & -\psi_2 & 1-\varphi_1-\psi_1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\psi_{N_y-1} & \cdots & -\psi_3 & -\psi_2 & 1-\varphi_1-\psi_1 \end{bmatrix}.$$

We shall seek for a well-approximated solution $\mathbf{u}$ of the linear system (3.3). Note that the system (3.3) has a unique solution if and only if P is invertible, or equivalently, A is invertible. Once we find the vector $\mathbf{u}$, we can obtain the matrix U due to the injectivity of the operator $\mathrm{Vec}(\cdot)$.

3.2. An interesting special case: The 2D space fractional diffusion equation

Here, we consider the 2D space fractional diffusion equation

$$\frac{\partial u(x,y,t)}{\partial t} - {}^{RL}D_x^{\alpha_1} u(x,y,t) - {}^{RL}D_y^{\alpha_2} u(x,y,t) = f(x,y,t), \tag{3.4}$$

where $\alpha_1, \alpha_2 \in (1,2]$, the spatial domain is $[a_x, b_x]\times[a_y, b_y]$, and the temporal domain is $[0,T]$. Eq (3.4) is a special case of Eq (3.1) for which the fractional order β is equal to 1.

We discretize Eq (3.4) by computing an approximated solution at (x_i, y_j, t_k) with $x_i = a_x + ih_x$, $y_j = a_y + jh_y$ and $t_k = kh_t$ where h_x, h_y and h_t are defined as (3.2). The first-order and the fractional-order derivatives are approximated by the forward time difference method and the Grünwald-Letnikov approximation (2.1) and (2.2), respectively. For convenience, we define

$$\gamma_p = \frac{g_{\alpha_1,p}h_t}{h_x^{\alpha_1}} \text{ and } \delta_q = \frac{g_{\alpha_2,q}h_t}{h_y^{\alpha_2}}.$$

From Eq (3.4), we obtain that for each $i \in \{1,\ldots,N_x-1\}$, $j \in \{1,\ldots,N_y-1\}$ and $k \in \{1,\ldots,N_t\}$,

$$\frac{u_{i,j,k+1}-u_{i,j,k}}{h_t} - \frac{1}{h_x^{\alpha_1}}\sum_{p=0}^{i+1} g_{\alpha_1,p}u_{i-p+1,j,k+1} - \frac{1}{h_y^{\alpha_2}}\sum_{q=0}^{j+1} g_{\alpha_2,q}u_{i,j-q+1,k+1} = f_{i,j,k+1},$$

or equivalently,

$$u_{i,j,k+1} - u_{i,j,k} - \sum_{p=0}^{i+1}\gamma_p u_{i-p+1,j,k+1} - \sum_{q=0}^{j+1}\delta_q u_{i,j-q+1,k+1} = h_t f_{i,j,k+1}.$$

We use the notations N_{xy}, N, U, F, $\mathbf{u}$, and $\mathbf{f}$ as same as those in Subsection 3.1. Then we can put the discretized equations into a system of N linear equations in N variables $u_{111}, \ldots, u_{N_x-1,N_y-1,N_t}$ and displayed as

$$P\mathbf{u} = \mathbf{f}. \tag{3.5}$$

In this case, P is a block lower-triangular matrix as follows:

$$P = \begin{bmatrix} \tilde{A} & 0 & 0 & \cdots & 0 \\ -I_{N_{xy}} & \tilde{A} & 0 & \cdots & 0 \\ 0 & -I_{N_{xy}} & \tilde{A} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -I_{N_{xy}} & \tilde{A} \end{bmatrix},$$

where

$$\tilde{A} = \begin{bmatrix} \tilde{B} & -\gamma_0 I_{N_y-1} & 0 & \cdots & 0 \\ -\gamma_2 I_{N_y-1} & \tilde{B} & -\gamma_0 I_{N_y-1} & \cdots & 0 \\ -\gamma_3 I_{N_y-1} & -\gamma_2 I_{N_y-1} & \tilde{B} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\gamma_{N_x-1} I_{N_y-1} & \cdots & -\gamma_3 I_{N_y-1} & -\gamma_2 I_{N_y-1} & \tilde{B} \end{bmatrix},$$

$$\tilde{B} = \begin{bmatrix} 1-\gamma_1-\delta_1 & -\delta_0 & 0 & \cdots & 0 \\ -\delta_2 & 1-\gamma_1-\delta_1 & -\delta_0 & \cdots & 0 \\ -\delta_3 & -\delta_2 & 1-\gamma_1-\delta_1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\delta_{N_y-1} & \cdots & -\delta_3 & -\delta_2 & 1-\gamma_1-\delta_1 \end{bmatrix}.$$

4. Proposing a gradient-descent iterative algorithm

In this section, we apply the gradient-descent technique to derive an iterative procedure for solving the linear system (3.3) and, in particular, the system (3.5). Assume that the coefficient matrix P is invertible. Then we can solve for the analytical solution $\mathbf{u}^*$ directly by

$$\mathbf{u}^* = P^{-1}\mathbf{f}. \tag{4.1}$$

For approximate solutions, we measure an error using the quadratic norm-error function

$$\Omega : \mathbb{R}^N \to \mathbb{R}, \quad \Omega(\mathbf{u}) = \frac{1}{2}\|P\mathbf{u} - \mathbf{f}\|_F^2. \tag{4.2}$$

We introduce a gradient-descent iterative solver described by the recursive formula

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \tau_{k+1}\nabla\Omega(\mathbf{u}_k). \tag{4.3}$$

We pick an arbitrary initial solution $\mathbf{u}_0$, so that Eq (4.3) iteratively computes the next solution $\mathbf{u}_{k+1}$ and form a sequence of approximate solutions. Due to the gradient direction $\nabla\Omega(\mathbf{u}_k)$ and a convergence factor τ_{k+1}, the sequence $\{\mathbf{u}_k\}_{k=0}^{\infty}$ would converge to the exact solution $\mathbf{u}^*$.

To find the gradient of Ω, we apply Lemma 2.1 to derive

$$\begin{aligned} \nabla\Omega(u) &= \frac{1}{2}\frac{d}{d\mathbf{u}}(P\mathbf{u} - \mathbf{f})^T(P\mathbf{u} - \mathbf{f}) \\ &= \frac{1}{2}\frac{d}{d\mathbf{u}}\,\mathrm{tr}((P\mathbf{u} - \mathbf{f})(P\mathbf{u} - \mathbf{f})^T) \\ &= \frac{1}{2}\frac{d}{d\mathbf{u}}\,\mathrm{tr}(P\mathbf{u}\mathbf{u}^T P^T - \mathbf{f}\mathbf{u}^T P^T - P\mathbf{u}\mathbf{f}^T + \mathbf{f}\mathbf{f}^T) \\ &= P^T(P\mathbf{u} - \mathbf{f}). \end{aligned}$$

Hence, the iterative equation takes the form

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \tau_{k+1}P^T(\mathbf{f} - P\mathbf{u}).$$

According to the gradient-descent, the convergence factor τ_{k+1} is generated in order to minimize the error occurring at each iteration. Thus, we define a new function $s_{k+1} : [0, \infty) \to \mathbb{R}$ to be an error at step $k + 1$:

$$s_{k+1}(\tau) := \Omega(\mathbf{u}_{k+1}) = \frac{1}{2}\|P(\mathbf{u}_k + \tau P^T(\mathbf{f} - P\mathbf{u})) - \mathbf{f}\|_F^2.$$

Using properties of the matrix trace, we obtain the derivative of s_{k+1} as follows:

$$\frac{d}{d\tau}s_{k+1}(\tau) \;=\; \tau\,\mathrm{tr}(PP^T(\mathbf{f} - T\mathbf{u}_k)(\mathbf{f} - T\mathbf{u}_k)^T PP^T) - \mathrm{tr}(PP^T(\mathbf{f} - P\mathbf{u}_k)(\mathbf{f} - P\mathbf{u}_k)^T).$$

It is easy to check that the second-order derivative $\frac{d^2}{d\tau^2}s_{k+1}(\tau)$ is positive. Setting $\frac{d}{d\tau}s_{k+1}(\tau) = 0$, we have the minimizer of the function $s_{k+1}(\tau)$ as follows:

$$\begin{aligned}\tau_{k+1} &= \frac{\mathrm{tr}(PP^T(\mathbf{f} - P\mathbf{u}_k)(\mathbf{f} - P\mathbf{u}_k)^T)}{\mathrm{tr}(PP^T(\mathbf{f} - P\mathbf{u}_k)(\mathbf{f} - P\mathbf{u}_k)^T PP^T)} \\ &= \frac{\|P^T(\mathbf{f} - P\mathbf{u}_k)\|_F^2}{\|PP^T(\mathbf{f} - P\mathbf{u}_k)\|_F^2}.\end{aligned}$$

We call $\{\tau_{k+1}\}_{k=0}^{\infty}$ the sequence of optimal convergence factors. We now describe the Frobenius norms $\|P^T(\mathbf{f} - P\mathbf{u}_k)\|_F$ and $\|PP^T(\mathbf{f} - P\mathbf{u}_k)\|_F$ more precisely. To avoid duplicated multiplications, let $\mathbf{y} = P^T\mathbf{f}$ and $V = P^T P$. Consider

$$P^T(\mathbf{f} - P\mathbf{u}_k) \;=\; \mathbf{y} - V\mathbf{u}_k \;=\; \begin{bmatrix}\mathbf{y}_1\\ \vdots\\ \mathbf{y}_N\end{bmatrix} - \begin{bmatrix}v_{11} & \dots & v_{1N}\\ \vdots & & \vdots\\ v_{N1} & \dots & v_{NN}\end{bmatrix}\begin{bmatrix}\mathbf{u}_1\\ \vdots\\ \mathbf{u}_N\end{bmatrix} = \begin{bmatrix}\mathbf{y}_1 - \sum_{i=1}^N v_{1i}\mathbf{u}_i\\ \vdots\\ \mathbf{y}_N - \sum_{i=1}^N v_{Ni}\mathbf{u}_i\end{bmatrix}.$$

Let us denote $\mathbf{y}_i$ and v_{ij} for the ith entry of $\mathbf{y}$ and the (i, j)th entry of V, respectively. It follows that

$$\|P^T(\mathbf{f} - P\mathbf{u}_k)\|_F^2 \;=\; \sum_{i=1}^N(\mathbf{y}_i - \sum_{j=1}^N v_{ij}\mathbf{u}_j)^2.$$

In a similar way, by denoting $\mathbf{z} = P\mathbf{y}$ and $W = PV$, we obtain

$$\|PP^T(\mathbf{f} - P\mathbf{u}_k)\|_F^2 \;=\; \sum_{i=1}^N(\mathbf{z}_i - \sum_{j=1}^N w_{ij}\mathbf{u}_j)^2.$$

We combine the direction and the step size altogether to obtain the following iterative algorithm for solving Eqs (3.3) and (3.5), see Algorithm 1.

Algorithm 1: A gradient-descent iterative solver for the linear system (3.3) arising from the 2D space-time fractional diffusion equation.

```
**Input:** $P$, $\mathbf{f}$, $\mathbf{u}_0$
$\mathbf{y} = P^T\mathbf{f}$;
$V = P^T P$;
$\mathbf{z} = P\mathbf{y}$;
$W = PV$;
for $k = 1, \dots, n$ do
    $\tau_{k+1} = \sum_{i=1}^{N}(\mathbf{y}_i - \sum_{j=1}^{N} v_{ij}\mathbf{u}_{k_j})^2 / \sum_{i=1}^{N}(\mathbf{z}_i - \sum_{j=1}^{N} w_{ij}\mathbf{u}_{k_j})^2$;
    $\mathbf{u}_{k+1} = \mathbf{u}_k + \tau_{k+1}(\mathbf{y} - V\mathbf{u}_k)$;
end
```

To break the procedure, one can impose a stopping rule $\|P\mathbf{u}_k - \mathbf{f}\|_F < \epsilon$ or $\|\mathbf{u}_k - \mathbf{u}_{k-1}\|_F < \epsilon$, where ϵ is a satisfactory error. Note that the coefficient matrix P is sparse, so that the computational procedures need not so much time.

5. Capability and theoretical performance of the proposed algorithm

In this section, we investigate the capability and theoretical performance of Algorithm 1 through error esimates and the convergence rate.

To show that Algorithm 1 is applicable for any initial vector, let us recall the following approximations for strongly convex functions on $\mathbb{R}^n$. Recall also that for real symmetric matrices A and B of the same size, the matrix ordering $A \leqslant B$ means that the difference $B - A$ is positive semidefinite.

Lemma 5.1. *([41]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a strongly convex function, i.e., there exist two numbers $\phi, \Psi \geqslant 0$ such that the matrix orderings $\phi I \leqslant \nabla^2 f(x) \leqslant \Psi I$ hold for all $x \in \mathbb{R}^n$. Then, for any $u, v \in \mathbb{R}^n$, we have*

$$f(v) \geqslant f(u) + \nabla f(u)^T(v-u) + \frac{\phi}{2}\|v-u\|_F^2, \tag{5.1}$$

$$f(v) \leqslant f(u) + \nabla f(u)^T(v-u) + \frac{\Psi}{2}\|v-u\|_F^2. \tag{5.2}$$

Theorem 5.2. *Consider the linear system* (3.3) *where the coefficient matrix P is invertible. Denote by κ the condition number of P. Then the following statements hold:*

(i) For any initial vector $\mathbf{u}_0$, the sequence of approximate solutions $\{\mathbf{u}_k\}$ generated by Algorithm 1 converges to a unique solution $\mathbf{u}^*$.

(ii) Error estimates of $\|P\mathbf{u}_k - \mathbf{f}\|_F$ compared to the preceding step and the initial step are described by the following inequalities:

$$\|P\mathbf{u}_k - \mathbf{f}\|_F \leqslant (1-\kappa^{-2})^{\frac{1}{2}}\,\|P\mathbf{u}_{k-1} - \mathbf{f}\|_F, \tag{5.3}$$

$$\|P\mathbf{u}_k - \mathbf{f}\|_F \leqslant (1-\kappa^{-2})^{\frac{k}{2}}\|P\mathbf{u}_0 - \mathbf{f}\|_F. \tag{5.4}$$

In particular, the relative error decreases at every iteration.

(iii) Algorithm 1 has the convergence rate (in regard to the relative error $\|P\mathbf{u}_k - \mathbf{f}\|_F$) governed by $\sqrt{1-\kappa^{-2}}$.

(iv) Moreover, the following error estimates for $\|\mathbf{u}_k - \mathbf{u}^*\|_F$ *hold:*

$$\|\mathbf{u}_k - \mathbf{u}^*\|_F \leqslant \sqrt{\kappa^2 - 1}\,\|\mathbf{u}_{k-1} - \mathbf{u}^*\|_F, \tag{5.5}$$

$$\|\mathbf{u}_k - \mathbf{u}^*\|_F \leqslant \kappa(1 - \kappa^{-2})^{\frac{k}{2}}\|\mathbf{u}_0 - \mathbf{u}^*\|_F. \tag{5.6}$$

Proof. If there exists an integer $k > 0$ so that a vector $\mathbf{u}_k$ makes $\nabla\Omega(\mathbf{u}_k)$ to be zero, we have $\mathbf{u}_k = \mathbf{u}^*$ and the result holds. So we assume that $\nabla\Omega(\mathbf{u}_k) \neq 0$ for all k. Now, we compute the second-order derivative $\nabla^2\Omega(\mathbf{u}) = P^TP$, which is a positive semidefinite matrix. Let $\lambda_{\min}$ and $\lambda_{\max}$ be the smallest and the largest eigenvalues of P^TP, respectively. From the spectral theory of matrices, we obtain

$$\lambda_{\min}I \leqslant P^TP \leqslant \lambda_{\max}I,$$

where I is the identity matrix of compatible size. Hence, Ω is a strongly convex function in which $\phi = \lambda_{\min}$ and $\Psi = \lambda_{\max}$. Applying Eq (5.2) to the function $s_{k+1}(\tau)$, we have

$$\Omega(\mathbf{u}_{k+1}) \leqslant \Omega(\mathbf{u}_k) - \tau\|\nabla\Omega(\mathbf{u}_k)\|_F^2 + \frac{\lambda_{\max}\tau^2}{2}\|\nabla\Omega(\mathbf{u}_k)\|_F^2. \tag{5.7}$$

Let us define the right-hand side (RHS) of (5.7) to be a function of τ, namely,

$$f(\tau) := \Omega(\mathbf{u}_k) - \tau\|\nabla\Omega(\mathbf{u}_k)\|_F^2 + \frac{\lambda_{\max}\tau^2}{2}\|\nabla\Omega(\mathbf{u}_k)\|_F^2.$$

We can verify that $1/\lambda_{\max}$ is a minimizer of f. Minimizing (5.7) both sides by $\tau = 1/\lambda_{\max}$, we obtain

$$\Omega(\mathbf{u}_{k+1}) \leqslant \Omega(\mathbf{u}_k) - \frac{1}{2\lambda_{\max}}\|\nabla\Omega(\mathbf{u}_k)\|_F^2. \tag{5.8}$$

Applying Eq (5.1) to the function $\Omega k + 1(\tau)$, we have

$$\Omega(\mathbf{u}_{k+1}) \geqslant \Omega(\mathbf{u}_k) - \tau\|\nabla\Omega(\mathbf{u}_k)\|_F^2 + \frac{\lambda_{\min}\tau^2}{2}\|\nabla\Omega(\mathbf{u}_k)\|_F^2. \tag{5.9}$$

Similarly, we have that $\tau = 1/\lambda_{\min}$ is a minimizer of the RHS of (5.9). We continue in this fashion to obtain

$$\begin{aligned} 0 &\geqslant \Omega(\mathbf{u}_k) - \frac{1}{\lambda_{\min}}\|\nabla\Omega(\mathbf{u}_k)\|_F^2 + \frac{1}{2\lambda_{\min}}\|\nabla\Omega(\mathbf{u}_k)\|_F^2 \\ &= \Omega(\mathbf{u}_k) - \frac{1}{2\lambda_{\min}}\|\nabla\Omega(\mathbf{u}_k)\|_F^2. \end{aligned}$$

Now, $\|\nabla\Omega(\mathbf{u}_k)\|_F^2 \geqslant 2\lambda_{\min}\Omega(\mathbf{u}_k)$, and hence by taking account of (5.8), we get

$$\Omega(\mathbf{u}_{k+1}) \leqslant (1 - \frac{\lambda_{\min}}{\lambda_{\max}})\Omega(\mathbf{u}_k) = (1 - \kappa^{-2})\Omega(\mathbf{u}_k). \tag{5.10}$$

Since P is an invertible matrix, all eigenvalues of P^TP are strictly positive, and thus $1 - \kappa^{-2} < 1$. By recurring the above inequality, it follows that for any $k \in \mathbb{N}$,

$$\Omega(\mathbf{u}_k) \leqslant (1 - \kappa^{-2})^k\Omega(\mathbf{u}_0) \tag{5.11}$$

Therefore, $\Omega(\mathbf{u}_k) \to 0$, i.e., $\mathbf{u}_k \to \mathbf{u}^*$ as $k \to \infty$. Moreover, from the error bounds (5.10) and (5.11), we get the inequalities (5.3) and (5.4), respectively. Thus, the asymptotic behavior of $\mathbf{u}_k$ with respect to the relative error $\|P\mathbf{u}_k - \mathbf{f}\|_F$ is governed by $\sqrt{1-\kappa^{-2}}$.

From the estimate (5.4) and the norm properties (2.4) and (2.5), we derive

$$\begin{aligned}
\|\mathbf{u}_k - \mathbf{u}^*\|_F &= \|P^{-1}P\mathbf{u}_k - P^{-1}P\mathbf{u}^*\|_F \\
&\leqslant \|P^{-1}\|_2\|P\mathbf{u}_k - \mathbf{f}\|_F \\
&\leqslant \|P^{-1}\|_2(1-\kappa^{-2})^{\frac{k}{2}}\|P\mathbf{u}_0 - \mathbf{f}\|_F \\
&\leqslant \|P^{-1}\|_2\|P\|_2(1-\kappa^{-2})^{\frac{k}{2}}\|\mathbf{u}_0 - \mathbf{u}^*\|_F \\
&= \kappa(1-\kappa^{-2})^{\frac{k}{2}}\|\mathbf{u}_0 - \mathbf{u}^*\|_F.
\end{aligned}$$

Similarly, we can have the error estimate $\|\mathbf{u}_k - \mathbf{u}^*\|_F$ compared to the preceding step $\|\mathbf{u}_{k-1} - \mathbf{u}^*\|_F$ using Eqs (2.4) and (2.5) together with the bound (5.3). □

Theorem 5.2 implies that the condition number κ of the sparse matrix P effects the convergence behaviour. Indeed, Algorithm 1 converges faster to the exact solution if κ is close to 1.

6. Capability and performance of the algorithm through numerical experiments

In this section, we implement numerical experiments to perform the capability and performance of Algorithm 1, which is denoted by *TauOpt*. All experiments have been carried out by MATLAB R2020b with PC environment Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, RAM 8.00 GB. To measure the time consuming (in seconds) taken for each simulation, we apply the *tic* and *toc* functions in MATLAB and abbreviate them TC. At the kth iteration step, we concern the relative error $\|\mathbf{f} - P\mathbf{u}_k\|_F$. In each example, we provide tables and figures to demonstrate the accuracy and efficiency of algorithms. We compare the performance of the proposed algorithm with famous iterative methods for the linear system, for instance, GI algorithm [42], LSI algorithm [42], SOR algorithm, and JOR algorithm [43].

Example 6.1. *Consider the* 2*D space-time fractional diffusion equation*

$$^{C}D_t^{1/3}u(x,y,t) - {}^{RL}D_x^{3/2}u(x,y,t) - {}^{RL}D_y^{3/2}u(x,y,t) = f(x,y,t), \tag{6.1}$$

where

$$f(x,y,t) = 1.1077t^{2/3}\sin x \sin y - tx^{-1/2}(\sin y)E_{2,0.5}(-x^2) - ty^{-1/2}(\sin x)E_{2,0.5}(-y^2).$$

The domains are $x \in [0,\pi]$, $y \in [0,\pi]$, *and* $t \in [0, 0.1]$. *Indeed, the exact solution for this equation subjected to the zero Dirichlet boundary condition and the zero initial condition is given by*

$$u^*(x,y,t) = t\sin x \sin y.$$

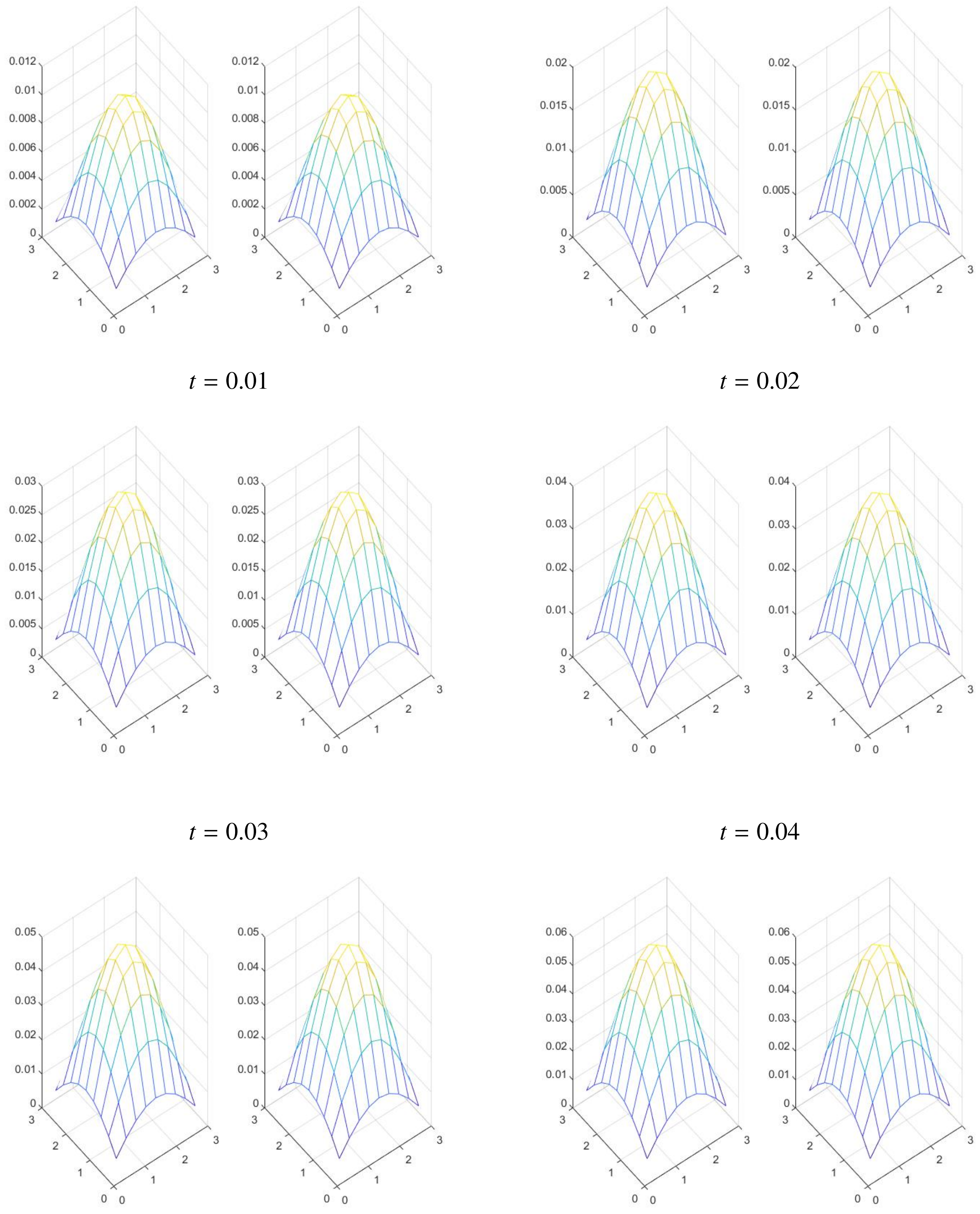
According to the numerical scheme explained in Subsect. 3.1, we discretize Eq (6.1) *by choosing partition numbers* $N_x = 10$, $N_y = 10$, *and* $N_t = 20$. *Then the coefficient matrix P of the associated linear system* (3.3) *is of size* 1620×1620. *In this case, the condition number of P is* $\kappa = 10.2976$. *Theoretically, Algorithm 1 has the convergence rate governed by* $\sqrt{1-\kappa^{-2}} = 0.9953$. *Table 1 shows* *the comparisons between numerical solutions and the exact solution for certain values of x, y, t in the domain. After running 250 iterations, we see that the 4-digits numerical solutions are very close to the exact solution. Also, Figure 1 illustrates the 3D plot of the exact and numerical solutions in the cases that $t = 0.01, 0.02, \ldots, 0.06$. Besides, we compare the performance of Algorithm 1 with SOR, JOR, GI and LSI algorithms. The results, shown in Table 2 and Figure 2, indicate that Algorithm 1 performs well in both iteration numbers and computational time.*

Table 1. The numerical and analytical solutions for Example 6.1.

	$u(x,y,t)$							
	$y = 0.4\pi$				$y = 0.6\pi$			
	$x = 0.1\pi$		$x = 0.3\pi$		$x = 0.5\pi$		$x = 0.7\pi$	
t	numerical	analytical	numerical	analytical	numerical	analytical	numerical	analytical
0.005	0.0029	0.0029	0.0078	0.0079	0.0097	0.0098	0.0079	0.0080
0.010	0.0042	0.0043	0.0115	0.0116	0.0144	0.0145	0.0117	0.0118
0.015	0.0055	0.0056	0.0152	0.0153	0.0190	0.0192	0.0155	0.0156
0.020	0.0068	0.0069	0.0188	0.0190	0.0237	0.0239	0.0193	0.0195
0.025	0.0081	0.0082	0.0225	0.0227	0.0283	0.0286	0.0231	0.0233
0.030	0.0094	0.0095	0.0261	0.0264	0.0330	0.0332	0.0269	0.0271
0.035	0.0107	0.0108	0.0298	0.0300	0.0377	0.0379	0.0307	0.0309
0.040	0.0120	0.0121	0.0335	0.0337	0.0423	0.0426	0.0345	0.0347
0.045	0.0133	0.0134	0.0371	0.0374	0.0470	0.0472	0.0383	0.0385
0.050	0.0146	0.0147	0.0408	0.0410	0.0516	0.0519	0.0421	0.0424

Table 2. Relative error and time consuming for Example 6.1.

Method	Relative error	TC
TauOpt	0.0037	6.1169
SOR	0.9177	116.6804
JOR	0.9010	96.6348
GI	0.8103	4.7887
LSI	0.7223	90.4681



$t = 0.01$

$t = 0.02$

$t = 0.03$

$t = 0.04$

$t = 0.05$

$t = 0.06$

Figure 1. The 3D plot of the exact (left) and numerical (right) solutions for Example 6.1.

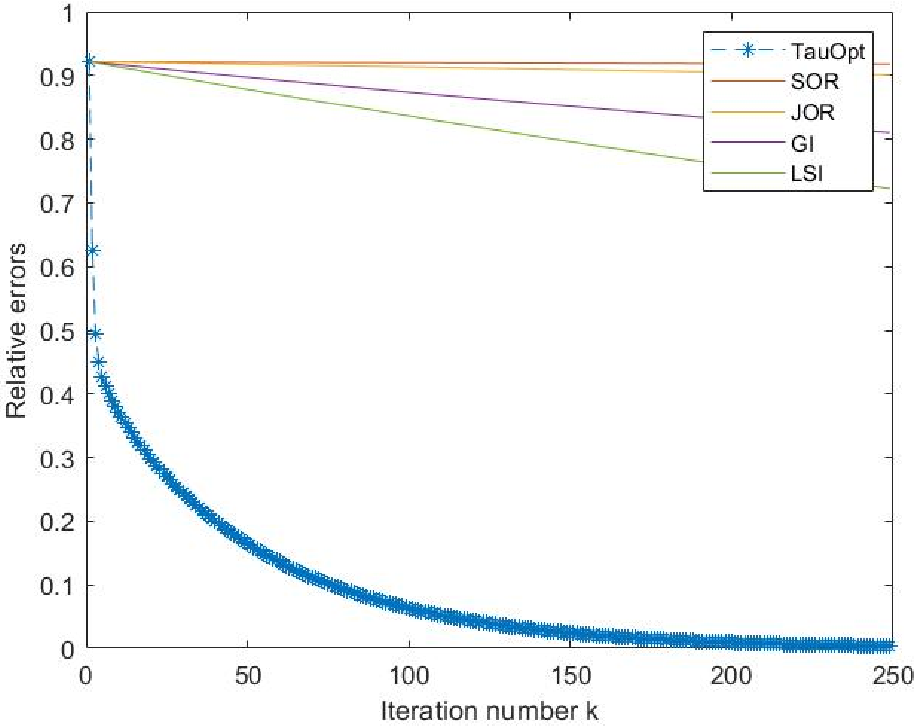


Figure 2. The relative error plot for Example 6.1.

Example 6.2. *Consider the* 2*D space fractional diffusion equation*

$$\frac{\partial u(x,y,t)}{\partial t} - {}^{RL}D_x^{4/3}u(x,y,t) - {}^{RL}D_y^{4/3}u(x,y,t) = f(x,y,t), \tag{6.2}$$

where

$$f(x,y,t) = (y^2 - y - 2.2155ty^{2/3} + 0.7385ty^{-1/3})(\sin 2x) - 2tx^{-1/3}(y^2 - y)E_{2,2/3}(-4x^2),$$

subjected to the zero Dirichlet boundary condition and the zero initial condition. The domains are $x \in [0,\pi]$, $y \in [0,1]$, *and* $t \in [0,0.1]$. *Indeed, the exact solution for this problem is given by*

$$u^*(x,y,t) = t(y^2 - y)(\sin 2x).$$

According to the discussions in Subsection 3.2, we discretize Eq (6.2) *using partition numbers* $N_x = 10$, $N_y = 10$ *and* $N_t = 20$. *In this case, the condition number of the coefficient matrix* $P \in \mathbb{R}^{1690\times1690}$ *is* $\kappa = 26.7484$. *Hence, Algorithm 1 has the convergence rate governed by* $\sqrt{1-\kappa^{-2}} = 0.9993$. *We run Algorithm 1 for* 500 *iterations and report the comparison between numerical solutions and the exact solution in Table 3. Figure 3 illustrates the* 3D *plot of the exact and numerical solutions in the cases that* $t = 0.01, 0.02, \ldots, 0.06$. *The results from Table 3 reveal that numerical solutions are very close to the exact solution.*

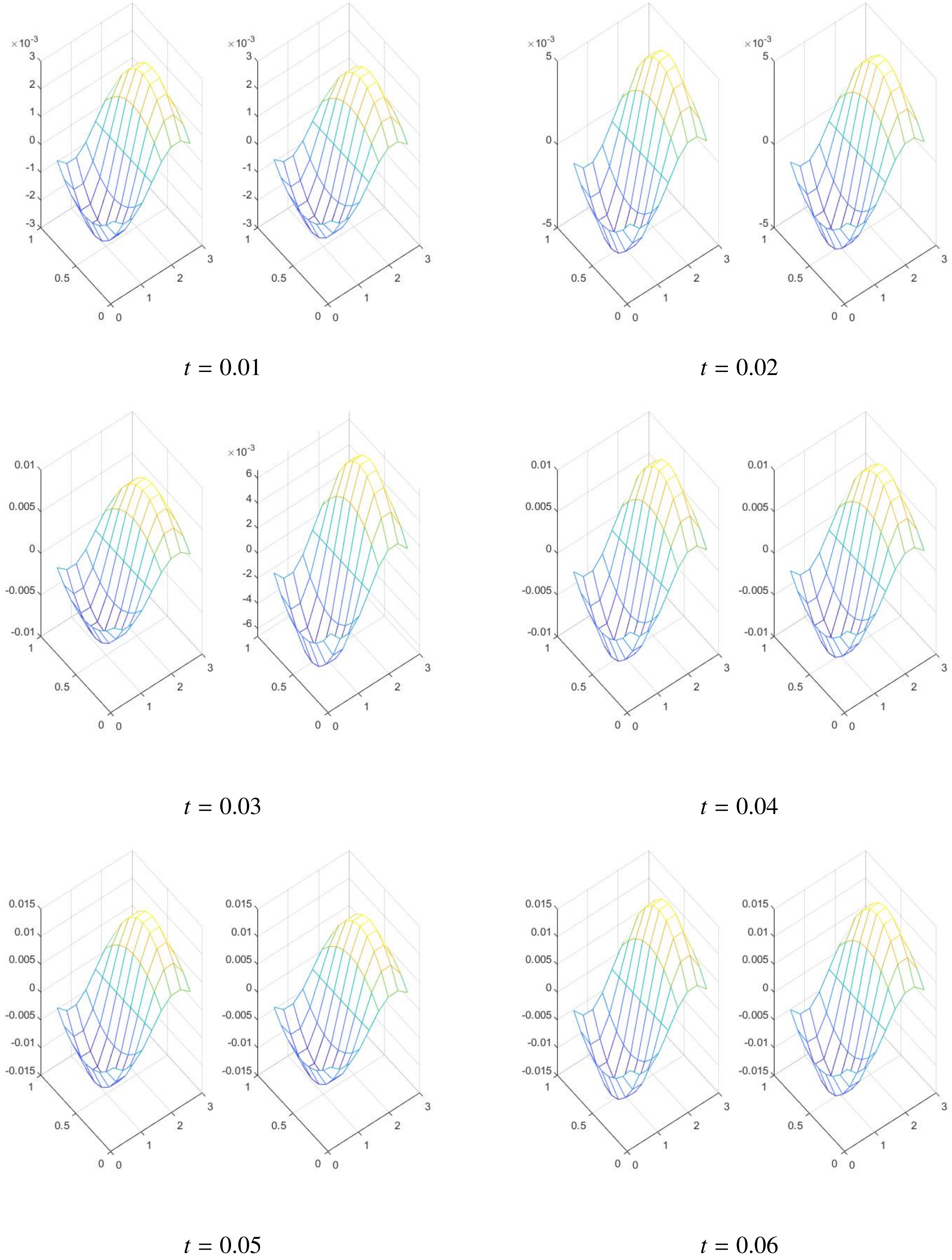
In addition, we compare the performance of Algorithm 1 (TauOpt) with SOR, JOR, GI, and LSI algorithms. The results of running 500 *iterations are shown in Table 4. Figure 4 displays the logarithmic relative error of each algorithm. It is seen that Algorithm 1 performs well in both iteration numbers and time consuming.*

Table 3. The numerical and analytical solutions for Example 6.2.

	$u(x, y, t)$							
	$y = 0.4\pi$				$y = 0.6\pi$			
	$x = 0.2\pi$		$x = 0.4\pi$		$x = 0.6\pi$		$x = 0.8\pi$	
t	numerical	analytical	numerical	analytical	numerical	analytical	numerical	analytical
0.005	-0.0021	-0.0022	-0.0013	-0.0014	0.0013	0.0013	0.0021	0.0022
0.010	-0.0032	-0.0033	-0.0019	-0.0021	0.0019	0.0020	0.0031	0.0034
0.015	-0.0042	-0.0045	-0.0026	-0.0028	0.0026	0.0027	0.0042	0.0045
0.020	-0.0053	-0.0056	-0.0032	-0.0035	0.0032	0.0034	0.0053	0.0056
0.025	-0.0064	-0.0067	-0.0040	-0.0042	0.0039	0.0041	0.0064	0.0067
0.030	-0.0074	-0.0078	-0.0046	-0.0049	0.0046	0.0048	0.0074	0.0079
0.035	-0.0085	-0.0089	-0.0053	-0.0056	0.0052	0.0055	0.0085	0.0090
0.040	-0.0095	-0.0100	-0.0060	-0.0063	0.0059	0.0061	0.0096	0.0101
0.045	-0.0106	-0.0111	-0.0066	-0.0070	0.0065	0.0068	0.0106	0.0111
0.050	-0.0116	-0.0122	-0.0073	-0.0077	0.0071	0.0075	0.0117	0.0123

Table 4. Relative error and time consuming for Example 6.2.

Method	Relative error	TC
TauOpt	0.0062	20.7160
SOR	0.0343	202.5196
JOR	0.0338	189.5905
GI	0.0342	10.6095
LSI	0.0211	178.5967



$t = 0.01$

$t = 0.02$

$t = 0.03$

$t = 0.04$

$t = 0.05$

$t = 0.06$

Figure 3. The 3D plot of the exact (left) and numerical (right) solutions for Example 6.2.

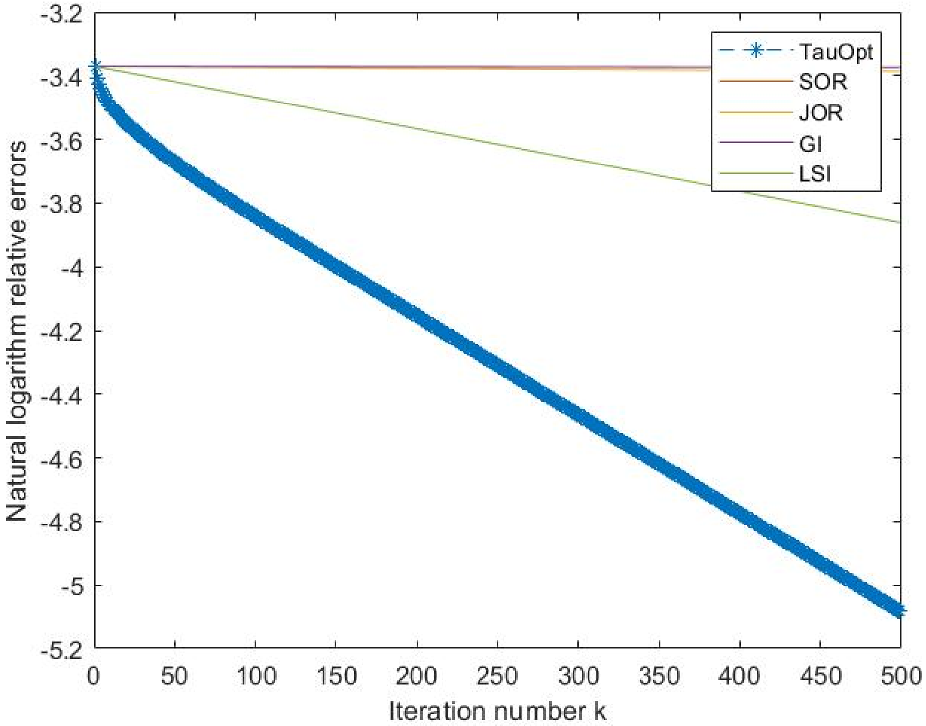


Figure 4. The logarithmic relative error plot for Example 6.2.

Note that in Examples 6.1 and 6.2, the condition numbers κ seem to be too much, so that the convergences of the proposed algorithm would be not fast. However, due to the fact that the coefficient matrix P is sparse and the convergence factor is chosen by an optimazation technique, the iterative procedures lead to a desire solution with a satisfactory error in a short time.

7. Conclusions

We discretize the 2D space-time fractional and the 2D space fractional diffusion equations via the finite difference scheme of Grünwald-Letnikov approximation. We transform the discretized equations into a sparse linear system with coefficient matrix P. We propose an iterative solver based on the technique of gradient-descent. The analysis confirms the capability of the proposed algorithm as long as the matrix P is invertible with the convergence rate governed by $\sqrt{1-\kappa^{-2}}$, where κ is the condition number of P. The numerical experiments indicate the accuracy and the efficiency of the proposed algorithm compared to well-known iterative solvers for linear systems. Indeed, the iterative procedures of the proposed algorithm lead to a desire solution with a satisfactory error in a short time since the coefficient matrix P is sparse, the convergence factor is chosen by an optimazation technique, and the procedures avoid duplicated computations.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. I. Podlubny, *Fractional differential equations: An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, New York: Academic Press, 1998.

2. I. Podlubny, *Fractional differential equations*, New York: Academic Press, 1999.

3. R. Hilfer, *Applications of fractional calculus in physics*, Singapore: World Scientific Publishing, 2000.

4. V. V. Kulish, J. L. Lage, Application of fractional calculus to fluid mechanics, *J. Fluids Eng.,* **124** (2002), 803–806. http://dx.doi.org/10.1115/1.1478062

5. X. J. Jiang, P. J. Scott, *Free-form surface filtering using the diffusion equation*, Adv. Metrol., 2020, 129–142. https://doi.org/10.1016/B978-0-12-821815-0.00006-X

6. Q. Liu, F. Liu, I. Turner, V. Anh, Approximation of the Levy-Feller advection-dispersion process by random walk and flnite difierence method, *J. Comput. Phys.,* **222** (2007), 57–70.

7. P. Zhuang, F. Liu, Implicit difference approximation for the two-dimensional space-time fractional diffusion equation, *J. Appl. Math. Comput.*, **25** (2007), 269–282.

8. J. Huang, F. Liu, The space-time fractional diffusion equation with Caputo derivatives, *J. Appl. Math. Comput.,* **19** (2005), 179–190.

9. Z. Q. Chen, M. M. Meerschaert, E. Nane, Space-time fractional diffusion on bounded domains, *J. Math. Anal. Appl.,* **393** (2012), 479–488. http://dx.doi.org/10.1016/j.jmaa.2012.04.032

10. J. Mua, B. Ahmad, S. Huang, Existence and regularity of solutions to time-fractional diffusion equations, *Comput. Math. Appl.,* **73** (2017), 985–996. https://doi.org/10.1016/j.camwa.2016.04.039

11. M. M. Meerschaert, D. A. Benson, H. P. Scheffler, B. Baeumer, Stochastic solution of space-time fractional diffusion equations, *Phys. Rev. E,* **65** (2002), 041103. http://dx.doi.org/10.1103/PhysRevE.65.041103

12. L. Chen, R. H. Nochetto, E. Otárola, A. J. Salgado, A PDE approach to fractional diffusion: A posteriori error analysis, *J. Comput. Phys.,* **293** (2015), 339–358. http://dx.doi.org/10.1016/j.jcp.2015.01.001

13. R. H. Nochetto, E. Otarola, A. J. Salgado, A PDE approach to fractional diffusion in general domains: A prior error analysis, *Found. Comput. Math.,* **15** (2015), 733–791. http://dx.doi.org/10.1007/s10208-014-9208-x

14. R. Gorenflo, F. Mainardi, Random walk models for space-fractional diffusion processes, *Fract. Calc. Appl. Anal.,* **1** (1998), 167–191.

15. R. Gorenflo, F. Mainardi, Approximation of Lévy-Feller diffusion by random walk, *Z. für Anal. Anwend.,* **18** (1999), 231–246. http://dx.doi.org/10.4171/ZAA/879

16. R. Gorenflo, F. Mainardi, M. P. Paradisi, Time fractional diffusion: A discrete random walk approach, *Nonlinear Dyn.,* **29** (2002), 129–143. http://dx.doi.org/10.1023/A:1016547232119

17. R. Gorenflo, F. Mainardi, D. Moretti, G. Pagnini, P. Paradisi, Discrete random walk models for space-time fractional diffusion, *Chem. Phys.,* **284** (2002), 521–541. http://dx.doi.org/10.1016/S0301-0104(02)00714-0

18. R. L. Magin, O. Abdullah, D. Baleanu, X. J. Zhou, Anomalous diffusion expressed through fractional order differential operators in the Bloch-Torrey equation, *J. Magn. Reson.,* **190** (2008), 255–270. http://dx.doi.org/10.1016/j.jmr.2007.11.007

19. A. V. Chechkin, R. Gorenflo, I. M. Sokolov, Fractional diffusion in inhomogeneous media, *J. Phys. Math. Gen.,* **38** (2005), 679–684. http://dx.doi.org/10.1088/0305-4470/38/42/L03

20. F. Santamaria, S. Wils, E. D. Schutter, G. J. Augustine, Anomalous diffusion in Purkinje cell dendrites caused by spines, *Neuron,* **52** (2006), 635–648. http://dx.doi.org/10.1016/j.neuron.2006.10.025

21. S. Umarov, S. Steinberg, Variable order differential equations with piecewise constant order-function and diffusion with changing modes, *Z. für Anal. Anwend.,* **28** (2009), 431–450. https://doi.org/10.1016/j.poly.2008.11.015

22. M. Inc, The approximate and exact solutions of the space and time-fractional Burgers equations with initial conditions by variational iteration method, *J. Math. Anal. Appl.,* **345** (2008), 476–484. https://doi.org/10.1016/j.jmaa.2008.04.007

23. N. H. Sweilam, M. M. Khader, R. F. Al-Bar, Numerical studies for a multi-order fractional differential equation, *Phys. Lett. A,* **371** (2007), 26–33. https://doi.org/10.1016/j.physleta.2007.06.016

24. L. Song, H. Zhang, Application of homotopy analysis method to fractional KdV-Burgers-KURamoto equation, *Phys. Lett. A,* **367** (2007), 88–94. https://doi.org/10.1016/j.physleta.2007.02.083

25. H. Jafari, V. Daftardar-Gejji, Solving linear and nonlinear fractional diffusion and wave equations by Adomian decomposition, *Appl. Math.Comput.,* **180** (2006), 488–497. https://doi.org/10.1016/j.amc.2005.12.031

26. C. Yang, J. Hou, An approximate solution of nonlinear fractional differential equation by Laplace transform and Adomian polynomials, *J. Inf. Comput. Sci.,* **10** (2013), 213–222.

27. T. Akram, M. Abbas, A. I. Ismail, *An extended cubic B-spline collocation scheme for time fractional sub-diffusion equation*, AIP Conference Proceedings, 2019. https://doi.org/10.1063/1.5136449

28. T. Akram, M. Abbas, A. I. Ismail, *Numerical solution of fractional cable equation via extended cubic B-spline,* AIP Conference Proceedings, 2019. https://doi.org/10.1063/1.5121041

29. U. Ali, A. Iqbal, M. Sohail, F. A. Abdull, Z. Khan, Compact implicit difference approximation for time-fractional diffusion-wave equation, *Alex. Eng. J.,* **61** (2022), 4119–4126. https://doi.org/10.1016/j.aej.2021.09.005

30. A. Iqbal, M. J. Siddiqui, I. Muhia, M. Abbasb, T. Akram, Nonlinear waves propagation and stability analysis for planar waves at far field using quintic B-spline collocation method, *Alex. Eng. J.,* **59** (2020), 2695–2703. https://doi.org/10.1016/j.aej.2020.05.011

31. M. Syam, M. A. Refai, Solving fractional diffusion equation via the collocation method based on fractional Legendre functions, *J. Comput. Meth. Phys.,* 381074, 2014. http://dx.doi.org/10.1155/2014/381074

32. M. R. Cui, Compact finite difference method for the fractional diffusion equation, *J. Comput. Phys.,* **228** (2009), 7792–7804. http://dx.doi.org/10.1016/j.jcp.2009.07.021

33. K. Diethelm, J. M. Ford, N. J. Fordc, M. Weilbeer, Pitfalls in fast numerical solvers for fractional differential equations, *J. Comput. Appl. Math.,* **186** (2006), 482–503. http://dx.doi.org/10.1016/j.cam.2005.03.023

34. R. Du, W. R. Cao, Z. Z. Sun, A compact difference scheme for the fractional diffusion-wave equation, *Appl. Math. Model.,* **34** (2010), 2998–3007. http://dx.doi.org/10.1016/j.apm.2010.01.008

35. P. Kumar, O. P. Agrawal, Numerical scheme for the solution of fractional differential equations of order greater than one, *J. Comput. Nonlin. Dyn.,* **1** (2006), 178–185. http://dx.doi.org/10.1115/1.2166147

36. A. Kittisopaporn, P. Chansangiam, Gradient-descent iterative algorithm for solving a class of linear matrix equations with applications to heat and Poisson equations, *Adv. Differ. Equ.,* **2020** (2020). https://doi.org/10.1186/s13662-020-02785-9

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

38. R. Scherer, S. L. Kalla, Y. F. Tang, J. F. Huang, The Grunwald-Letnikov method for fractional differential equations, *Comput. Math. Appl.,* **62** (2011), 813–823. http://dx.doi.org/10.1016/j.camwa.2011.03.054

39. H. Lütkepohl, *Handbook of matrices*, Chichester: John Wiley & Sons, 1996.

40. R. A. Horn, C. R. Johnson, *Topics in matrix analysis*, New York: Cambridge University Press, 1991.

41. P. B. Stephen, V. Lieven, *Convex optimization*, New York: Cambridge University Press, 2004.

42. F. Ding, T. Chen, Iterative least-squares solutions of coupled Sylvester matrix equations, *Syst. Control. Lett.,* **54** (2005), 95–107. http://dx.doi.org/10.1016/j.sysconle.2004.06.008

43. D. M. Young, *Iterative solution of large linear systems*, New York: Academic Press, 1971.