A preconditioning technique for all-at-once system from the nonlinear tempered fractional diffusion equation

Yong-Liang Zhao[a], Pei-Yong Zhu[a], Xian-Ming Gu[b], Xi-Le Zhao[a], Huan-Yan Jian[a]

[a] *School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, P.R. China*

[b] *School of Economic Mathematics/Institute of Mathematics, Southwestern University of Finance and Economics, Chengdu, Sichuan 611130, P.R. China*

Abstract

An all-at-once linear system arising from the nonlinear tempered fractional diffusion equation with variable coefficients is studied. Firstly, the nonlinear and linearized implicit schemes are proposed to approximate such the nonlinear equation with continuous/discontinuous coefficients. The stabilities and convergences of the two schemes are proved under several suitable assumptions, and numerical examples show that the convergence orders of these two schemes are 1 in both time and space. Secondly, a nonlinear all-at-once system is derived based on the nonlinear implicit scheme, which may suitable for parallel computations. Newton's method, whose initial value is obtained by interpolating the solution of the linearized implicit scheme on the coarse space, is chosen to solve such the nonlinear all-at-once system. To accelerate the speed of solving the Jacobian equations appeared in Newton's method, a robust preconditioner is developed and analyzed. Numerical examples are reported to demonstrate the effectiveness of our proposed preconditioner. Meanwhile, they also imply that such the initial guess for Newton's method is more suitable.

Keywords: Nonlinear tempered fractional diffusion equation, All-at-once system; Newton's method, Krylov subspace method, Toeplitz matrix, banded Toeplitz preconditioner

2010 MSC: 65L05, 65N22, 65F10

1. Introduction

In this work, we mainly focus on solving the all-at-once system arising from the nonlinear tempered fractional diffusion equation (NL-TFDE):

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} = d_+(x)\, {}_a\mathbf{D}_x^{\alpha,\lambda}\, u(x,t) + d_-(x)\, {}_x\mathbf{D}_b^{\alpha,\lambda}\, u(x,t) + f(u(x,t),x,t), & (x,t) \in [a,b] \times (0,T], \\ u(a,t) = u(b,t) = 0, & 0 \le t \le T, \\ u(x,0) = u_0(x), & a \le x \le b, \end{cases} \tag{1.1}$$

Email addresses: uestc_ylzhao@sina.com (Yong-Liang Zhao), zpy6940@uestc.edu.cn (Pei-Yong Zhu), guxianming@live.cn (Xian-Ming Gu), xlzhao122003@163.com (Xi-Le Zhao), uestc_hyjian@sina.com (Huan-Yan Jian)

where $\alpha \in (1,2)$, $\lambda \geq 0$, $d_+(x) \geq d_-(x) > 0$ and $f(u(x,t),x,t)$ is a source term which satisfies the Lipschitz condition:

$$\mid f(r_1,x,t) - f(r_2,x,t) \mid \leq L \mid r_1 - r_2 \mid, \text{ for all } r_1, r_2 \text{ over } [a,b]\times[0,T].$$

The variants of the left and right Riemann-Liouville tempered fractional derivatives are respectively defined as [1–3]

$$ {}_a\mathbf{D}_x^{\alpha,\lambda}\, u(x,t) = {}_aD_x^{\alpha,\lambda}\, u(x,t) - \alpha\lambda^{\alpha-1}\frac{\partial u(x,t)}{\partial x} - \lambda^{\alpha} u(x,t), \tag{1.2}$$

and

$$ {}_x\mathbf{D}_b^{\alpha,\lambda}\, u(x,t) = {}_xD_b^{\alpha,\lambda}\, u(x,t) + \alpha\lambda^{\alpha-1}\frac{\partial u(x,t)}{\partial x} - \lambda^{\alpha} u(x,t), \tag{1.3}$$

where ${}_aD_x^{\alpha,\lambda}\, u(x,t)$ and ${}_xD_b^{\alpha,\lambda}\, u(x,t)$ are the left and right Riemann-Liouville tempered fractional derivatives defined respectively as [1, 2]

$$ {}_aD_x^{\alpha,\lambda}\, u(x,t) = \frac{e^{-\lambda x}}{\Gamma(2-\alpha)}\frac{\partial^2}{\partial x^2}\int_a^x \frac{e^{\lambda\xi}u(\xi,t)}{(x-\xi)^{1-\alpha}}d\xi,$$

$$ {}_xD_b^{\alpha,\lambda}\, u(x,t) = \frac{e^{\lambda x}}{\Gamma(2-\alpha)}\frac{\partial^2}{\partial x^2}\int_x^b \frac{e^{-\lambda\xi}u(\xi,t)}{(\xi-x)^{1-\alpha}}d\xi.$$

If $\lambda = 0$, they reduce to the Riemann-Liouville fractional derivatives [4].

Tempered fractional diffusion equations (TFDEs) are exponentially tempered extension of fractional diffusion equations. In recent several decades, the TFDEs are widely used across various fields, such as statistical physics [1, 5, 6], finance [7–10] and geophysics [2, 11–14]. Unfortunately, it is difficult to obtain the analytical solutions of TFDEs, or the obtained analytical solutions are less practical. Hence, numerical methods such as finite difference method [15, 16] and finite element method [17] become important approaches to solve TFDEs. There are limited works addressing the finite difference schemes for the TFDEs. Baeumera and Meerschaert [2] provided finite difference and particle tracking methods for solving the TFDEs on a bounded interval. The stability and second-order accuracy of the resulted schemes are discussed. Cartea and del-Castillo-Negrete [18] proposed a general finite difference scheme to numerically solve a Black-Merton-Scholes model with tempered fractional derivatives. Marom and Momoniat [19] compared the numerical solutions of three fractional partial differential equations (FDEs) with tempered fractional derivatives that occur in finance. However, the stabilities of their proposed schemes are not proved. Recently, Li and Deng [20] derived a series of high order difference approximations (called tempered-WSGD operators) for the tempered fractional calculus. They also used such operators to numerically solve the TFDE, and the stability and convergence of the obtained numerical schemes are proved.

Similar to the fractional derivatives, the tempered fractional derivatives are nonlocal. Thus the discretized systems for TFDEs usually accompany a full (or dense) coefficient matrix. Traditional methods (e.g., Gaussian elimination) to solve such systems need computational cost is of $\mathcal{O}(N^3)$ and storage requirement is of $\mathcal{O}(N^2)$, where N is the number of space grid points. Fortunately, the coefficient matrix always holds a Toeplitz-like structure. It is well known that Toeplitz matrices possess great structures and properties, and their matrix-vector multiplications can be computed in $\mathcal{O}(N \log N)$ operations via fast Fourier transform (FFT) [21, 22]. With this truth, the memory requirement and computational cost of Krylov subspace methods are $\mathcal{O}(N)$ and $\mathcal{O}(N \log N)$, respectively. However, the convergence rate of the Krylov subspace methods will be slow, if the coefficient matrix is ill-conditioned. To address this problem, Wang et al. [9] proposed a circulant preconditioned generalized minimal residual method (PGMRES) to solve the discretized linear system, whose computational cost is of $\mathcal{O}(N \log N)$. Lei et al. [23] proposed fast solution algorithms for solving TFDEs in one-dimensional (1D) and two-dimensional (2D). In their article, for 1D case, a circulant preconditioned iterative method and a fast-direct method are developed, and the computational complexity of both methods are $\mathcal{O}(N \log N)$ in each time step. For 2D case, such two methods were extended to fast solve their alternating direction implicit (ADI) scheme, and the complexity of both methods are $\mathcal{O}(N^2 \log N)$ in each time step. For many other studies about Toeplitz-like systems, see [24–27] and the references therein.

Actually, all the aforementioned fast implementations for TFDEs are developed based on the time-stepping schemes, which are not suitable for parallel computations. If all the time steps are stacked in a vector, the all-at-once system is obtained and it is suitable for parallel computations, see [28, 29]. To the best of our knowledge, such the system arising from the FDEs or the partial differential equations have been studied by many researchers [30–37]. However, the all-at-once system arising from the TFDEs is less studied. In this work, a preconditioning technique is designed for such the system arising from the NL-TFDE (1.1). The rest of this paper is organized as follows: in Section 2, the nonlinear and linearized implicit schemes are derived by utilizing the finite difference method. Then, the nonlinear all-at-once system is obtained from the nonlinear implicit scheme. The stabilities and convergences of such two schemes are analyzed in Section 3. A preconditioning technique is designed in Section 4 to accelerate solving such the all-at-once system. In Section 5, numerical examples are provided to illustrate the first-order convergences of the two implicit schemes and show the performance of our preconditioning strategy for solving such the system. Concluding remarks are given in Section 6.

2. Two implicit schemes and all-at-once system

In this section, the nonlinear and linearized implicit schemes are proposed to approach Eq. (1.1). Then, the all-at-once system is obtained from the nonlinear one.

2.1. Two implicit schemes

In order to derive the proposed schemes, we first introduce the mesh $\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau$, where $\bar{\omega}_h = \{x_i = a + ih,\ i = 0, 1, \cdots, N;\ x_0 = a, x_N = b\}$ and $\bar{\omega}_\tau = \{t_j = j\tau,\ j = 0, 1, \cdots, M;\ t_M = T\}$. Let u_i^j represents the numerical approximation of $u(x_i,t_j)$. Then the variants of the Riemann-Liouville tempered fractional derivatives defined in Eqs. (1.2)-(1.3) for $(x,t)=(x_i,t_j)$ can be approximated respectively as [38, 20]:

$$
{}_a\mathbf{D}_x^{\alpha,\lambda}\, u(x,t)|_{(x,t)=(x_i,t_j)} = \frac{1}{h^\alpha}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^j - \alpha\lambda^{\alpha-1}\delta_x u_i^j + \mathcal{O}(h); \tag{2.1}
$$

$$
{}_x\mathbf{D}_b^{\alpha,\lambda}\, u(x,t)|_{(x,t)=(x_i,t_j)} = \frac{1}{h^\alpha}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^j + \alpha\lambda^{\alpha-1}\delta_x u_i^j + \mathcal{O}(h), \tag{2.2}
$$

where

$$
\delta_x u_i^j = \frac{u_i^j - u_{i-1}^j}{h} \quad \text{and} \quad g_k^{(\alpha)} = \begin{cases} \tilde{g}_1^{(\alpha)} - e^{h\lambda}\left(1-e^{-h\lambda}\right)^\alpha, & k=1, \\ \tilde{g}_k^{(\alpha)} e^{-(k-1)h\lambda}, & k\neq 1 \end{cases}
$$

with $\tilde{g}_k^{(\alpha)} = (-1)^k\binom{\alpha}{k}$ $(k\geq 0)$.

As for the time discretization, the backward Euler method is used. Combining Eqs. (2.1) and (2.2), the following first-order nonlinear implicit Euler scheme (NL-IES) is obtained:

$$
\begin{aligned}
&u_i^j - w_1\left(d_{+,i}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^j + d_{-,i}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^j\right) + w_2\left(d_{+,i}-d_{-,i}\right)\left(u_i^j - u_{i-1}^j\right) \\
&= u_i^{j-1} + \tau f_{u,i}^j,
\end{aligned} \tag{2.3}
$$

in which $w_1 = \frac{\tau}{h^\alpha}$, $w_2 = \frac{\alpha\lambda^{\alpha-1}\tau}{h}$, $d_{\pm,i} = d_\pm(x_i)$ and $f_{u,i}^j = f(u(x_i,t_j),x_i,t_j)$. Applying the formula $f(u(x_i,t_j),x_i,t_j) = f(u(x_i,t_{j-1}),x_i,t_{j-1}) + \mathcal{O}(\tau)$ to Eq. (2.3) and omitting the small term, it gets the first-order linearized implicit Euler scheme (L-IES):

$$
\begin{aligned}
&u_i^j - w_1\left(d_{+,i}\sum_{k=0}^{i+1} g_k^{(\alpha)} u_{i-k+1}^j + d_{-,i}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} u_{i+k-1}^j\right) + w_2\left(d_{+,i}-d_{-,i}\right)\left(u_i^j - u_{i-1}^j\right) \\
&= u_i^{j-1} + \tau f_{u,i}^{j-1}.
\end{aligned} \tag{2.4}
$$

The stabilities and first-order convergences of schemes (2.3)-(2.4) will be discussed in Section 3.

2.2. *The all-at-once system*

Several auxiliary notations are introduced before deriving the all-at-once system: I and $\mathbf{0}$ represent the identity and zero matrices of suitable orders, respectively.

$$
\boldsymbol{u}^j = \left[u_1^j, u_2^j, \cdots, u_{N-1}^j\right]^T, \qquad \boldsymbol{f}_u^j = \left[f_{u,1}^j, f_{u,2}^j, \cdots, f_{u,N-1}^j\right]^T,
$$

$$
D_\pm = \mathrm{diag}(d_{\pm,1}, d_{\pm,2}, \cdots, d_{\pm,N-1}), \qquad B = \mathrm{tridiag}(-1,1,0),
$$

$$
\boldsymbol{u}=\begin{bmatrix}\boldsymbol{u}^1\\ \boldsymbol{u}^2\\ \vdots\\ \boldsymbol{u}^M\end{bmatrix},\ \boldsymbol{f}(\boldsymbol{u})=\begin{bmatrix}\boldsymbol{f}_u^1\\ \boldsymbol{f}_u^2\\ \vdots\\ \boldsymbol{f}_u^M\end{bmatrix},\ \boldsymbol{v}=\begin{bmatrix}\boldsymbol{u}^0\\ \boldsymbol{0}\\ \vdots\\ \boldsymbol{0}\end{bmatrix},\ G=\begin{bmatrix}g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \cdots & 0\\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ g_{N-2}^{(\alpha)} & \cdots & \ddots & \ddots & g_0^{(\alpha)}\\ g_{N-1}^{(\alpha)} & g_{N-2}^{(\alpha)} & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)}\end{bmatrix}.
$$

In this work, the all-at-once system is derived based on Eq. (2.3), which can be expressed as:

$$
\mathcal{A}\boldsymbol{u}=\tau\boldsymbol{f}(\boldsymbol{u})+\boldsymbol{v}, \tag{2.5}
$$

in which $\mathcal{A}=\mathrm{blktridiag}(-I,A,\boldsymbol{0})$ is a bi-diagonal block matrix with $A=I-w_1\left(D_+G+D_-G^T\right)+w_2\left(D_+-D_-\right)B$. Obviously, A is a Toeplitz-like matrix and its storage requirement is of $\mathcal{O}(N)$.

For this nonlinear all-at-once system, we prefer to utilize Newton's method [39]. Such the method requires to solve the equation with Jacobian matrix at each iterative step, and the computation of solving these equations consumes most of the method. Before applying Newton's method to solve the system (2.5), two essential problems need to be addressed:

1. How to find a good enough initial value?

2. How to solve the Jacobian equations efficiently?

Here, a strategy is provided to address these two problems. For the first problem, the initial value of Newton's method is constructed by interpolating the solution of L-IES (2.4) on the coarse mesh. Numerical experiences in Section 5 show that it is a good enough initial value. For the second problem, the Jacobian matrix of (2.5) is a bi-diagonal block matrix. More precisely, such the matrix is the sum of a diagonal block matrix and a bi-diagonal block matrix, whose blocks are Toeplitz-like matrices. Based on this special structure, the preconditioned Krylov subspace methods, such as preconditioned biconjugate gradient stabilized (PBiCGSTAB) method [40], are employed to solve the Jacobian equations appeared in Newton's method. The details will be discussed in Section 4.

3. Stabilities and convergences of (2.3) and (2.4)

In this section, the stabilities and convergences of NL-IES (2.3) and L-IES (2.4) are studied. Let U_i^j be the approximation solution of u_i^j in Eqs. (2.3) or (2.4) and $e_i^j=U_i^j-u_i^j$ $(i=1,\cdots,N-1;\ j=1,\cdots,M)$

be the error satisfying equation

$$e_i^j - w_1\left(d_{+,i}\sum_{k=0}^{i+1} g_k^{(\alpha)} e_{i-k+1}^j + d_{-,i}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} e_{i+k-1}^j\right) + w_2\left(d_{+,i} - d_{-,i}\right)\left(e_i^j - e_{i-1}^j\right)$$
$$= e_i^{j-1} + \tau\left(f_{U,i}^j - f_{u,i}^j\right) \text{ (for NL-IES)}$$

or

$$e_i^j - w_1\left(d_{+,i}\sum_{k=0}^{i+1} g_k^{(\alpha)} e_{i-k+1}^j + d_{-,i}\sum_{k=0}^{N-i+1} g_k^{(\alpha)} e_{i+k-1}^j\right) + w_2\left(d_{+,i} - d_{-,i}\right)\left(e_i^j - e_{i-1}^j\right)$$
$$= e_i^{j-1} + \tau\left(f_{U,i}^{j-1} - f_{u,i}^{j-1}\right) \text{ (for L-IES)},$$

in which $f_{U,i}^k = f(U_i^k, x_i, t_k)$. To prove the stabilities and convergences of (2.3) and (2.4), the following results given in [38, 41] are required.

Lemma 3.1. *([38]) The coefficients $g_k^{(\alpha)}$, for $k = 0, 1, \cdots$, satisfy:*

$$\begin{cases} g_1^{(\alpha)} < 0,\ g_k^{(\alpha)} > 0 \text{ (for } k \neq 1); \\ \sum\limits_{k=0}^{\infty} g_k^{(\alpha)} = 0,\ \sum\limits_{k=0}^{j} g_k^{(\alpha)} < 0 \text{ (for } j \geq 1). \end{cases}$$

Lemma 3.2. *([41], discrete Gronwall inequality) Suppose that $\tilde{f}_k \geq 0$, $\eta_k \geq 0$ $(k = 0, 1, \cdots)$, and*

$$\eta_{k+1} \leq \rho\eta_k + \tau\tilde{f}_k,\ \rho = 1 + C_0\tau,\ \eta_0 = 0,$$

where $C_0 \geq 0$ is a constant, then

$$\eta_{k+1} \leq \exp(C_0 t_k)\sum_{j=0}^{k} \tau\tilde{f}_j.$$

3.1. The stabilities of (2.3) and (2.4)

Denote $E^j = \left[e_1^j, e_2^j, \cdots, e_{N-1}^j\right]^T$, and assume that

$$\left\|E^j\right\|_\infty = \mid e_{\ell_j} \mid = \max_{1\leq \ell \leq N-1} \mid e_\ell^j \mid \ (0 \leq j \leq M, 1 \leq \ell_j \leq N-1).$$

Then the next theorem is established.

Theorem 3.1. *Suppose $d_+(x) \geq d_-(x) \geq 0$, then the L-IES (2.4) is stable, and we have*

$$\left\|E^j\right\|_\infty \leq \exp(TL)\left\|E^0\right\|_\infty, \text{ for } j = 1, \cdots, M.$$

Proof. From Lemma 3.1, $\sum\limits_{k=0}^{\ell_j+1} g_k^{(\alpha)} < 0$ and $\sum\limits_{k=0}^{N-\ell_j+1} g_k^{(\alpha)} < 0$. Then

$$\begin{aligned}
\mid e_{\ell_j}^j \mid \leq &\left[1 - w_1\left(d_{+,\ell_j}\sum_{k=0}^{\ell_j+1} g_k^{(\alpha)} + d_{-,\ell_j}\sum_{k=0}^{N-\ell_j+1} g_k^{(\alpha)}\right)\right] \mid e_{\ell_j}^j \mid + w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right) \times \\
&\left(\mid e_{\ell_j}^j \mid - \mid e_{\ell_j}^j \mid\right) \\
\leq &\mid e_{\ell_j}^j \mid - w_1\left(d_{+,\ell_j}\sum_{k=0}^{\ell_j+1} g_k^{(\alpha)} \mid e_{\ell_j-k+1}^j \mid + d_{-,\ell_j}\sum_{k=0}^{N-\ell_j+1} g_k^{(\alpha)} \mid e_{\ell_j+k-1}^j \mid\right) \\
&+ w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right)\left(\mid e_{\ell_j}^j \mid - \mid e_{\ell_j-1}^j \mid\right) \\
= &\mid e_{\ell_j}^j \mid + \mid -w_1 d_{+,\ell_j} g_1^{(\alpha)} e_{\ell_j}^j \mid + \mid -w_1 d_{-,\ell_j} g_1^{(\alpha)} e_{\ell_j}^j \mid + \mid w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right) e_{\ell_j}^j \mid \\
&- \sum_{k=0,k\neq 1}^{\ell_j+1} \mid -w_1 d_{+,\ell_j} g_k^{(\alpha)} e_{\ell_j-k+1}^j \mid - \sum_{k=0,k\neq 1}^{N-\ell_j+1} \mid -w_1 d_{-,\ell_j} g_k^{(\alpha)} e_{\ell_j+k-1}^j \mid \\
&- \mid -w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right) e_{\ell_j-1}^j \mid \\
= &\mid e_{\ell_j}^j - w_1\left(d_{+,\ell_j} g_1^{(\alpha)} e_{\ell_j}^j + d_{-,\ell_j} g_1^{(\alpha)} e_{\ell_j}^j\right) + w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right) e_{\ell_j}^j \mid \\
&- \sum_{k=0,k\neq 1}^{\ell_j+1} \mid -w_1 d_{+,\ell_j} g_k^{(\alpha)} e_{\ell_j-k+1}^j \mid - \sum_{k=0,k\neq 1}^{N-\ell_j+1} \mid -w_1 d_{-,\ell_j} g_k^{(\alpha)} e_{\ell_j+k-1}^j \mid \\
&- \mid -w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right) e_{\ell_j-1}^j \mid \\
\leq &\mid e_{\ell_j}^j - w_1\left(d_{+,\ell_j}\sum_{k=0}^{\ell_j+1} g_k^{(\alpha)} e_{\ell_j-k+1}^j + d_{-,\ell_j}\sum_{k=0}^{N-\ell_j+1} g_k^{(\alpha)} e_{\ell_j+k-1}^j\right) \\
&+ w_2\left(d_{+,\ell_j} - d_{-,\ell_j}\right)\left(e_{\ell_j}^j - e_{\ell_j-1}^j\right) \mid \\
= &\mid e_{\ell_j}^{j-1} + \tau\left(f_{U,\ell_j}^{j-1} - f_{u,\ell_j}^{j-1}\right) \mid \leq (1+\tau L) \mid e_{\ell_j}^{j-1} \mid \\
\leq &(1+\tau L)\left\|E^{j-1}\right\|_\infty.
\end{aligned}$$

The above inequality implies $\left\|E^j\right\|_\infty \leq (1+\tau L)\left\|E^{j-1}\right\|_\infty$. Then

$$\left\|E^j\right\|_\infty \leq (1+\tau L)^j \left\|E^0\right\|_\infty \leq \exp(TL)\left\|E^0\right\|_\infty.$$

□

From the above proof, it can be find that if $\tau L < 1$, the following result is true.

Corollary 1. *Suppose $d_+(x) \geq d_-(x) \geq 0$ and $\tau L < 1$, then the NL-IES (2.3) is stable, and it obtains*

$$\left\|E^k\right\|_\infty \leq C_1 \left\|E^0\right\|_\infty, \text{ for } k = 1, \cdots, M,$$

where C_1 is a positive constant.

Proof. Based on the proof of Theorem 3.1, it yields

$$(1 - \tau L)\left\|E^j\right\|_\infty \leq \left\|E^{j-1}\right\|_\infty.$$

Summing up for j from 1 to k and using Lemma 3.4 in [15], it gets

$$\left\|E^k\right\|_\infty \leq \frac{\exp(\frac{TL}{1-\tau L})}{1 - \tau L}\left\|E^0\right\|_\infty.$$

Note that

$$\lim_{\tau \to 0} \frac{\exp(\frac{TL}{1-\tau L})}{1 - \tau L} = \exp(TL).$$

Hence, there is a positive constant C_1 such that

$$\frac{\exp(\frac{TL}{1-\tau L})}{1 - \tau L} \leq C_1,$$

thereby $\left\|E^k\right\|_\infty \leq C_1 \left\|E^0\right\|_\infty$, for $k = 1, \cdots, M$. □

In Corollary 1, it is worth to notice that the assumption $\tau < \frac{1}{L}$ is independent of the spatial size h. Actually, the smaller time step size τ is, the easier such assumption can be satisfied.

3.2. The convergences of (2.3) and (2.4)

In this subsection, the convergences of (2.3) and (2.4) are studied. Let $\xi_i^j = u(x_i, t_j) - u_i^j$ satisfies

$$\begin{aligned}
&\xi_i^j - w_1\left(d_{+,i}\sum_{k=0}^{i+1} g_k^{(\alpha)}\xi_{i-k+1}^j + d_{-,i}\sum_{k=0}^{N-i+1} g_k^{(\alpha)}\xi_{i+k-1}^j\right) + w_2\left(d_{+,i} - d_{-,i}\right)\left(\xi_i^j - \xi_{i-1}^j\right) \\
&= \xi_i^{j-1} + \tau\left(f(u(x_i, t_j), x_i, t_j) - f_{u,i}^j\right) + R_i^j \text{ (for NL-IES)}
\end{aligned}$$

or

$$\begin{aligned}
&\xi_i^j - w_1\left(d_{+,i}\sum_{k=0}^{i+1} g_k^{(\alpha)}\xi_{i-k+1}^j + d_{-,i}\sum_{k=0}^{N-i+1} g_k^{(\alpha)}\xi_{i+k-1}^j\right) + w_2\left(d_{+,i} - d_{-,i}\right)\left(\xi_i^j - \xi_{i-1}^j\right) \\
&= \xi_i^{j-1} + \tau\left(f(u(x_i, t_{j-1}), x_i, t_{j-1}) - f_{u,i}^{j-1}\right) + R_i^j \text{ (for L-IES)},
\end{aligned}$$

where $\left|R_i^j\right| \leq C_2\left(\tau^2+\tau h\right)$ (C_2 is a positive constant). Denote $\boldsymbol{\xi^j}=\left[\xi_1^j, \xi_2^j, \cdots, \xi_{N-1}^j\right]^T$ and $\left\|\boldsymbol{\xi}^j\right\|_{\infty}=\mid \xi_{\ell_j} \mid=\max\limits_{1 \leq \ell \leq N-1} \mid \xi_{\ell}^j \mid \quad (0 \leq j \leq M, 1 \leq \ell_j \leq N-1)$. Similar to the proof of Theorem 3.1, the following theorem about the convergence of L-IES can be established.

Theorem 3.2. *Assume that $d_{+}(x) \geq d_{-}(x) \geq 0$ and the problem* (1.1) *has a sufficiently smooth solution $u(x, t)$. u_i^j is the numerical solution of* (2.4). *Then there is a positive constant C such that*

$$\left\|\boldsymbol{\xi}^j\right\|_{\infty} \leq C(\tau+h), \ j=1,2, \cdots, M.$$

Proof. Same technique in Theorem 3.1 is utilized, then it yields

$$\begin{aligned}
\left\|\boldsymbol{\xi}^j\right\|_{\infty}=\mid \xi_{\ell_j}^j \mid & \leq \mid \xi_{\ell_j}^{j-1}+\tau\left(f(u(x_{\ell_j}, t_{j-1}), x_{\ell_j}, t_{j-1})-f_{u, \ell_j}^{j-1}\right)+R_{\ell_j}^j \mid \\
& \leq(1+\tau L) \mid \xi_{\ell_j}^{j-1} \mid+C_2\left(\tau^2+\tau h\right) \\
& \leq(1+\tau L)\left\|\boldsymbol{\xi}^{j-1}\right\|_{\infty}+C_2\left(\tau^2+\tau h\right).
\end{aligned}$$

Using Lemma 3.2, it gets

$$\left\|\boldsymbol{\xi}^j\right\|_{\infty} \leq \exp(T L) T C_2(\tau+h) \leq C(\tau+h).$$

□

Corollary 2. *Assume that $d_{+}(x) \geq d_{-}(x) \geq 0$, $\tau L<1$ and the problem* (1.1) *has a sufficiently smooth solution $u(x, t)$. u_i^j is the numerical solution of* (2.3). *Then*

$$\left\|\boldsymbol{\xi}^j\right\|_{\infty} \leq C(\tau+h), \ j=1,2, \cdots, M,$$

where C is a positive constant.

Proof. According to Corollary 1, we have

$$(1-\tau L)\left\|\boldsymbol{\xi}^j\right\|_{\infty} \leq\left\|\boldsymbol{\xi}^{j-1}\right\|_{\infty}+C_2\left(\tau^2+\tau h\right).$$

Similarly, it arrives

$$\left\|\boldsymbol{\xi}^j\right\|_{\infty} \leq \frac{\exp(\frac{T L}{1-\tau L})}{1-\tau L} T C_2(\tau+h) \leq C(\tau+h), \ j=1,2, \cdots, M,$$

in which $\frac{\exp(\frac{T L}{1-\tau L})}{1-\tau L} \leq C_1$ is employed. □

It is interesting to note that if ξ_i^j represents the error between NL-IES (2.3) and L-IES (2.4), then it also satisfies $\left\|\boldsymbol{\xi}^j\right\|_\infty \leq C(\tau + h)$ $(j = 1, 2, \cdots, M)$. This can be proved easily through the condition $f_{u,i}^j = f_{u,i}^{j-1} + \mathcal{O}(\tau)$. In the next section, fast implementations are designed to solve (2.5).

4. The preconditioned iterative method

The Jacobian matrix of (2.5) can be treated as the sum of a diagonal block matrix and a block bi-diagonal matrix with Toeplitz-like blocks, then its matrix-vector multiplication can be done by FFT in $\mathcal{O}(MN \log MN)$ operations. Such the technique truly reduces the computational cost of a Krylov subspace method, such as the biconjugate gradient stabilized (BiCGSTAB) method, but the convergence rate of this method is slow when the Jacobi matrix is very ill-conditioned. In order to speed up the convergence rate of the Krylov subspace method, a preconditioner $P_\ell = \text{blktridiag}(-I, A_\ell, \mathbf{0})$ $(\ell > 2)$ is proposed and analyzed in this section, in which

$$A_\ell = I - w_1\left(D_+ G_\ell + D_- G_\ell^T\right) + w_2\left(D_+ - D_-\right) B \text{ with } G_\ell = \begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & & & & \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & & & \\ \vdots & \ddots & \ddots & \ddots & & \\ g_\ell^{(\alpha)} & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & g_\ell^{(\alpha)} & \cdots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix}.$$

Noticing the properties of $g_k^{(\alpha)}$ given in Lemma 3.1, the following result about P_ℓ is true.

Theorem 4.1. *The preconditioner P_ℓ is a nonsingular matrix.*

Proof. Obviously, we only need to proof the nonsingularity of A_ℓ. Let

$$H = \frac{A_\ell + A_\ell^T}{2} = I - \frac{\omega_1}{2}\left(D_+ + D_-\right)\left(G_\ell + G_\ell^T\right) + \frac{\omega_2}{2}\left(D_+ - D_-\right)\left(B + B^T\right)$$

and θ be the arbitrary eigenvalue of H. According to Lemma 3.1 and Gershgorin circle theorem [42], it arrives at

$$\begin{aligned}
&\left|\theta - \left[1 - \omega_1\left(d_{+,i} + d_{-,i}\right) g_1^{(\alpha)} + \omega_2\left(d_{+,i} - d_{-,i}\right)\right]\right| \\
&\leq r_i = \left|-\omega_1\left(d_{+,i} + d_{-,i}\right)\left(g_0^{(\alpha)} + g_2^{(\alpha)}\right) - \omega_2\left(d_{+,i} - d_{-,i}\right)\right| + \sum_{k=3}^{\ell}\left|-\omega_1\left(d_{+,i} + d_{-,i}\right) g_k^{(\alpha)}\right| \\
&\leq \omega_1\left(d_{+,i} + d_{-,i}\right) \sum_{k=0,\ k\neq 1}^{\ell} g_k^{(\alpha)} + \omega_2\left(d_{+,i} - d_{-,i}\right) < -\omega_1\left(d_{+,i} + d_{-,i}\right) g_1^{(\alpha)} + \omega_2\left(d_{+,i} - d_{-,i}\right).
\end{aligned}$$

This implies that all eigenvalues of H are larger than 1. Then, the desired result is achieved. $\square$

Unfortunately, it is difficult to theoretically investigate the eigenvalue distributions of the preconditioned Jacobian matrix, but we still can give some figures to illustrate the clustering eigenvalue distributions of several specified preconditioned matrices in the next section. For convenience, let $\boldsymbol{u}^{(k+1)}$ be the approximation of $\boldsymbol{u}$ obtained in the k-th Newton iterative step, the Jacobian matrix in the k-th Newton iterative step is denoted as J^k. With these auxiliary notations, the preconditioned Newton's method can be summarized as below:

Algorithm 1 Solve $\boldsymbol{u}$ from Eq. (2.5)

1: Given maximum iterative step $maxit$, tolerance tol_{out} and initial vector $\boldsymbol{u}^{(0)}$, which is obtained by interpolating the solution of L-IES (2.4) on the coarse grid (here $M = N = 16$)
2: **for** $k = 1, \cdots, maxit$ **do**
3: Solve $J^k \boldsymbol{z} = -\boldsymbol{f}(\boldsymbol{u}^{(k)})$ via PBiCGSTAB method with preconditioner P_ℓ ($\ell = 8$ is chosen experimentally to balance the number of iterations and CPU time)
4: $\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{z}$
5: **if** $\|\boldsymbol{z}\|_2 \le tol_{out}$ **then**
6: $\boldsymbol{u} = \boldsymbol{u}^{(k+1)}$
7: **break**
8: **end if**
9: **end for**

In fact, this algorithm can be viewed as a simple two-grid method, our readers can refer to [43–45] for details.

5. Numerical examples

Two numerical experiments presented in this section have a two-fold objective. On the one hand, they illustrate that the convergence orders of our two implicit schemes (2.3)-(2.4) are 1. On the other hand, they show the performance of the preconditioner P_ℓ proposed in Section 4. In Algorithm 1, for generating the initial guess $\boldsymbol{u}^{(0)}$, the MATLAB build-in function "`interp2`" is employed in this work. The $maxit$ and tol_{out} in Algorithm 1 are fixed as 100 and 10^{-12}, respectively. For the PBiCGSTAB method (or the BiCGSTAB method), it terminates if the relative residual error satisfies $\frac{\|r^k\|_2}{\|r^0\|_2} \le 10^{-6}$ or the iteration number is more than 1000, where r^k is the residual vector of the linear system after k iteration, and the initial guess of the PBiCGSTAB method (or the BiCGSTAB method) is chosen as the zero vector. "$\mathcal{P}$" represents that our proposed preconditioned iterative method in Section 4 is utilized to solve (2.5). "BS" (or "$\mathcal{I}$") means that the PBiCGSTAB method in Step 3 of Algorithm 1 is replaced by the MATLAB's backslash method (or the BiCGSTAB method). Some other notations, which will appear in later, are given:

$$Err(\tau, h) = \max_{0 \le j \le M} \|\boldsymbol{\xi}^j\|_\infty,$$

$$Order1 = \log_2 Err(2\tau, h)/Err(\tau, h), \quad Order2 = \log_2 Err(\tau, 2h)/Err(\tau, h).$$

"Iter1" represents the number of iterations required by Algorithm 1. "Iter2" denotes the average number of iterations required by the PBiCGSTAB method (or the BiCGSTAB method) in Algorithm 1, i.e.,

$$\text{Iter2} = \sum_{m=1}^{\text{Iter1}} \text{Iter2}(m)/\text{Iter1},$$

where Iter2(m) is the number of iterations required by such the method in the m-th iterative step of Algorithm 1. "Time" denotes the total CPU time in seconds for solving the system (2.5). "‡" means the maximum iterative step is reached but not convergence, and"†" means out of memory.

All experiments were performed on a Windows 10 (64 bit) PC-Intel(R) Core(TM) i7-8700k CPU 3.70 GHz, 16 GB of RAM using MATLAB R2016a.

Example 1. We consider Eq. (1.1) with $T = 1$, the initial value $u(x, 0) = (\cos \pi x - 1) \sin \pi x$ ($x \in [-1, 1]$), the nonlinear source term $f(u(x, t), x, t) = u(x, t) - 3u(x, t)$ and the continuous coefficients $d_+(x) = 1.5 \exp(-x)$ and $d_-(x) = \exp(x)$. Obviously, it is hard to obtain the exact solution of Eq. (1.1). Thus, the numerical solution computed from the finer mesh ($M = N = 1024$) is treated as the exact solution.

Table 1: The maximum norm errors and convergence orders for Example 1 with $h = 2^{-10}$.

		$\lambda = 0$				$\lambda = 1$				$\lambda = 5$				$\lambda = 10$			
		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)	
α	M	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$
1.1	64	8.0277E-02	–	8.1498E-02	–	1.1908E-02	–	1.1390E-02	–	6.6057E-03	–	6.3807E-03	–	4.8182E-03	–	4.6460E-03	–
	128	4.0105E-02	1.0012	4.1591E-02	0.9705	6.1196E-03	0.9604	5.8931E-03	0.9507	3.6702E-03	0.8479	3.5897E-03	0.8299	2.7628E-03	0.8024	2.7029E-03	0.7815
	256	1.7837E-02	1.1689	1.8690E-02	1.1540	2.8629E-03	1.0960	2.7890E-03	1.0793	1.8010E-03	1.0271	1.7783E-03	1.0134	1.4017E-03	0.9790	1.3849E-03	0.9647
	512	6.0553E-03	1.5586	6.3917E-03	1.5480	1.0254E-03	1.4813	1.0039E-03	1.4741	6.7680E-04	1.4120	6.6838E-04	1.4118	5.2752E-04	1.4099	5.2131E-04	1.4096
1.5	64	4.7666E-02	–	5.3762E-02	–	3.5875E-02	–	4.3134E-02	–	2.4027E-02	–	3.1109E-02	–	1.8286E-02	–	2.5143E-02	–
	128	3.6430E-02	0.3878	3.5974E-02	0.5796	1.9300E-02	0.8944	2.1332E-02	1.0158	1.1881E-02	1.0160	1.5200E-02	1.0333	8.9830E-03	1.0255	1.2218E-02	1.0411
	256	1.9950E-02	0.8687	1.9854E-02	0.8575	1.2203E-02	0.6614	1.2140E-02	0.8133	6.9339E-03	0.7769	6.9000E-03	1.1394	4.7950E-03	0.9057	5.3432E-03	1.1932
	512	7.5241E-03	1.4068	7.5004E-03	1.4044	5.2003E-03	1.2306	5.1850E-03	1.2274	3.2837E-03	1.0783	3.2750E-03	1.0751	2.3859E-03	1.0070	2.3798E-03	1.1669
1.9	64	1.1241E-01	–	1.1930E-01	–	1.1075E-01	–	1.1761E-01	–	1.0545E-01	–	1.1229E-01	–	1.0186E-01	–	1.0871E-01	–
	128	6.0869E-02	0.8850	6.3952E-02	0.8995	5.9987E-02	0.8846	6.3072E-02	0.8989	5.7145E-02	0.8839	6.0268E-02	0.8978	5.5212E-02	0.8835	5.8362E-02	0.8974
	256	2.8789E-02	1.0802	3.0258E-02	1.0797	2.8379E-02	1.0798	2.9832E-02	1.0801	2.6822E-02	1.0912	2.8264E-02	1.0924	2.5776E-02	1.0990	2.7172E-02	1.1029
	512	1.0164E-02	1.5020	1.0654E-02	1.5059	1.0020E-02	1.5019	1.0506E-02	1.5056	9.4678E-03	1.5023	9.9521E-03	1.5059	9.1113E-03	1.5003	9.5782E-03	1.5043
1.99	64	1.3122E-01	–	1.3757E-01	–	1.3117E-01	–	1.3752E-01	–	1.3105E-01	–	1.3739E-01	–	1.3119E-01	–	1.3753E-01	–
	128	7.1292E-02	0.8802	7.4522E-02	0.8844	7.1318E-02	0.8791	7.4540E-02	0.8836	7.1297E-02	0.8782	7.4508E-02	0.8828	7.1453E-02	0.8766	7.4661E-02	0.8813
	256	3.5500E-02	1.0059	3.6809E-02	1.0176	3.5495E-02	1.0067	3.6804E-02	1.0182	3.5471E-02	1.0072	3.6779E-02	1.0185	3.5526E-02	1.0081	3.6833E-02	1.0194
	512	1.2807E-02	1.4709	1.3271E-02	1.4718	1.2809E-02	1.4705	1.3271E-02	1.4716	1.2803E-02	1.4702	1.3265E-02	1.4713	1.2828E-02	1.4696	1.3289E-02	1.4708

Table 2: The maximum norm errors and convergence orders for Example 1 with $\tau = h$.

		$\lambda = 0$				$\lambda = 1$				$\lambda = 5$				$\lambda = 10$			
		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)	
α	N	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$	$Err(\tau, h)$	$Order2$
1.1	64	2.4026E-01	–	2.3738E-01	–	2.8807E-01	–	2.8824E-01	–	4.8922E-01	–	4.9031E-01	–	8.3296E-01	–	8.4089E-01	–
	128	1.9494E-01	0.3016	1.9416E-01	0.2900	1.7183E-01	0.7454	1.7102E-01	0.7531	2.2356E-01	1.1298	2.2247E-01	1.1401	3.8624E-01	1.1087	3.8625E-01	1.1224
	256	1.5375E-01	0.3424	1.5350E-01	0.3390	1.0246E-01	0.7459	1.0222E-01	0.7425	9.4040E-02	1.2493	9.3496E-02	1.2506	1.6193E-01	1.2541	1.6150E-01	1.2580
	512	9.6958E-02	0.6652	9.6904E-02	0.6636	4.7287E-02	1.1155	4.7236E-02	1.1137	3.1677E-02	1.5698	3.1520E-02	1.5686	5.4094E-02	1.5818	5.3945E-02	1.5820
1.5	64	7.3489E-02	–	7.2202E-02	–	6.0097E-02	–	5.9292E-02	–	1.3157E-01	–	1.3375E-01	–	3.0816E-01	–	3.1344E-01	–
	128	4.8998E-02	0.5848	4.8626E-02	0.5703	3.7724E-02	0.6718	3.7582E-02	0.6578	6.7287E-02	0.9674	6.8079E-02	0.9743	1.7000E-01	0.8581	1.7235E-01	0.8628
	256	2.5042E-02	0.9684	2.4956E-02	0.9623	1.9558E-02	0.9477	1.9515E-02	0.9455	3.0301E-02	1.1510	3.0576E-02	1.1548	8.0107E-02	1.0855	8.0929E-02	1.0906
	512	9.2783E-03	1.4324	9.2628E-03	1.4299	7.5546E-03	1.3723	7.5455E-03	1.3709	1.0369E-02	1.5471	1.0449E-02	1.5490	2.7766E-02	1.5286	2.7995E-02	1.5315
1.9	64	1.1145E-01	–	1.1829E-01	–	1.0157E-01	–	1.0836E-01	–	9.6149E-02	–	1.0316E-01	–	1.0809E-01	–	1.1351E-01	–
	128	6.0209E-02	0.8883	6.3279E-02	0.9025	5.4910E-02	0.8873	5.7978E-02	0.9023	5.2520E-02	0.8724	5.5832E-02	0.8857	6.0055E-02	0.8479	6.3014E-02	0.8491
	256	2.8477E-02	1.0802	2.9948E-02	1.0793	2.6153E-02	1.0701	2.7608E-02	1.0704	2.4581E-02	1.0953	2.6008E-02	1.1021	2.8130E-02	1.0942	2.9453E-02	1.0973
	512	1.0045E-02	1.5033	1.0538E-02	1.5069	9.2257E-03	1.5032	9.7132E-03	1.5071	8.5957E-03	1.5159	9.0714E-03	1.5196	9.8009E-03	1.5211	1.0259E-02	1.5215
1.99	64	1.3155E-01	–	1.3788E-01	–	1.2545E-01	–	1.3173E-01	–	1.2630E-01	–	1.3326E-01	–	1.3911E-01	–	1.4597E-01	–
	128	7.1323E-02	0.8832	7.4580E-02	0.8866	6.8822E-02	0.8662	7.2074E-02	0.8700	6.8825E-02	0.8759	7.1952E-02	0.8891	7.7047E-02	0.8524	8.0114E-02	0.8655
	256	3.5498E-02	1.0066	3.6803E-02	1.0190	3.4089E-02	1.0136	3.5389E-02	1.0262	3.2523E-02	1.0815	3.3934E-02	1.0843	3.6251E-02	1.0877	3.7635E-02	1.0900
	512	1.2802E-02	1.4714	1.3265E-02	1.4722	1.2329E-02	1.4673	1.2790E-02	1.4683	1.1514E-02	1.4981	1.1989E-02	1.5010	1.2871E-02	1.4939	1.3329E-02	1.4975

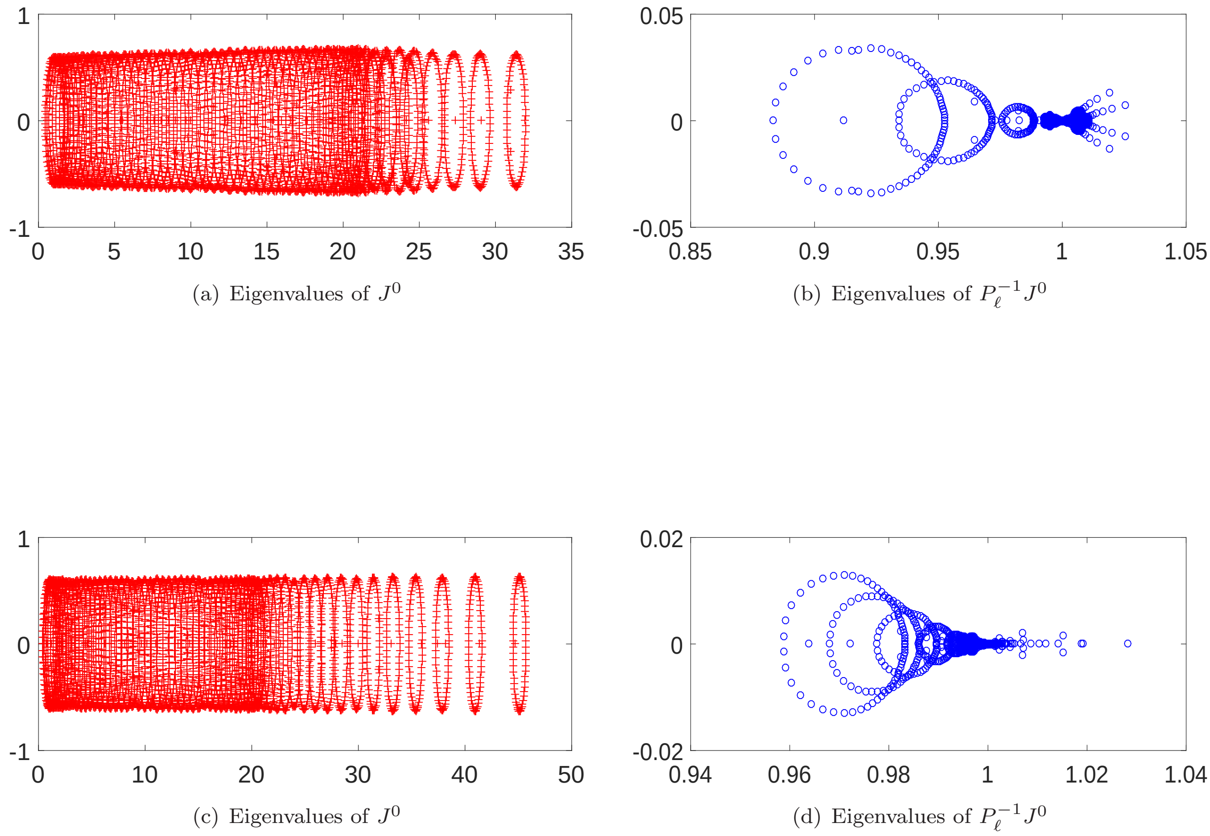
Table 3: The maximum norm errors between NL-IES (2.3) and L-IES (2.4) for Example 1 with $h = 2^{-10}$.

		$\lambda = 0$		$\lambda = 1$		$\lambda = 5$		$\lambda = 10$	
α	M	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$
1.1	64	1.1812E-02	–	7.9283E-03	–	6.5488E-03	–	5.8862E-03	–
	128	6.5390E-03	0.8531	3.9934E-03	0.9894	3.2863E-03	0.9948	2.9518E-03	0.9957
	256	3.4857E-03	0.9076	2.0043E-03	0.9945	1.6467E-03	0.9969	1.4784E-03	0.9976
	512	1.8064E-03	0.9483	1.0041E-03	0.9972	8.2423E-04	0.9985	7.3988E-04	0.9987
1.5	64	8.9765E-03	–	8.8547E-03	–	8.6193E-03	–	8.5281E-03	–
	128	4.6220E-03	0.9576	4.5559E-03	0.9587	4.4082E-03	0.9674	4.3454E-03	0.9727
	256	2.4014E-03	0.9446	2.3235E-03	0.9714	2.2390E-03	0.9773	2.1958E-03	0.9847
	512	1.2244E-03	0.9718	1.1764E-03	0.9819	1.1280E-03	0.9891	1.1049E-03	0.9908
1.9	64	8.1415E-03	–	8.1710E-03	–	8.2299E-03	–	8.2611E-03	–
	128	4.1759E-03	0.9632	4.1567E-03	0.9751	4.2044E-03	0.9690	4.2457E-03	0.9603
	256	2.2051E-03	0.9212	2.2097E-03	0.9116	2.2166E-03	0.9236	2.2180E-03	0.9367
	512	1.1399E-03	0.9519	1.1377E-03	0.9577	1.1312E-03	0.9705	1.1329E-03	0.9692
1.99	64	7.7831E-03	–	7.7909E-03	–	7.8126E-03	–	7.8311E-03	–
	128	4.1471E-03	0.9082	4.1464E-03	0.9099	4.1452E-03	0.9144	4.1443E-03	0.9181
	256	2.1825E-03	0.9261	2.1816E-03	0.9265	2.1800E-03	0.9271	2.1790E-03	0.9275
	512	1.1267E-03	0.9539	1.1264E-03	0.9537	1.1260E-03	0.9531	1.1258E-03	0.9527

Table 4: The maximum norm errors between NL-IES (2.3) and L-IES (2.4) for Example 1 with $\tau = h$.

		$\lambda = 0$		$\lambda = 1$		$\lambda = 5$		$\lambda = 10$	
α	N	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$
1.1	64	1.0573E-02	–	8.3127E-03	–	7.9461E-03	–	8.1361E-03	–
	128	6.0969E-03	0.7942	4.0972E-03	1.0207	3.7134E-03	1.0975	3.6526E-03	1.1554
	256	3.3629E-03	0.8584	2.0262E-03	1.0159	1.7471E-03	1.0878	1.6332E-03	1.1612
	512	1.7831E-03	0.9153	1.0076E-03	1.0079	8.4131E-04	1.0543	7.6363E-04	1.0968
1.5	64	8.9043E-03	–	8.8174E-03	–	8.5584E-03	–	8.4458E-03	–
	128	4.6102E-03	0.9497	4.5477E-03	0.9552	4.4088E-03	0.9570	4.3427E-03	0.9596
	256	2.3971E-03	0.9435	2.3200E-03	0.9710	2.2381E-03	0.9781	2.2016E-03	0.9800
	512	1.2235E-03	0.9703	1.1758E-03	0.9805	1.1287E-03	0.9876	1.1061E-03	0.9931
1.9	64	8.1434E-03	–	8.1893E-03	–	8.2678E-03	–	8.2806E-03	–
	128	4.1773E-03	0.9631	4.1606E-03	0.9769	4.2237E-03	0.9690	4.2707E-03	0.9553
	256	2.2049E-03	0.9219	2.2099E-03	0.9128	2.2176E-03	0.9295	2.2178E-03	0.9453
	512	1.1399E-03	0.9518	1.1378E-03	0.9577	1.1310E-03	0.9714	1.1330E-03	0.9690
1.99	64	7.7944E-03	–	7.8275E-03	–	7.9290E-03	–	8.0097E-03	–
	128	4.1447E-03	0.9112	4.1426E-03	0.9180	4.1333E-03	0.9398	4.1162E-03	0.9604
	256	2.1827E-03	0.9252	2.1817E-03	0.9251	2.1795E-03	0.9233	2.1772E-03	0.9188
	512	1.1267E-03	0.9540	1.1264E-03	0.9537	1.1260E-03	0.9528	1.1257E-03	0.9517

Tables 1-2 show that the convergence orders of the implicit schemes NL-IES (2.3) and L-IES (2.4) for different α and λ can indeed reach 1 in both time and space. The errors between NL-IES (2.3) and L-IES (2.4) are listed in Tables 3-4. From such tables, the $Order1$ and $Order2$ are almost 1, which in accord with the result at the end of Section 3. In Table 5, the CPU time and number of iterations of the methods BS, $\mathcal{I}$ and $\mathcal{P}$ are reported. The method $\mathcal{I}$ in most cases needs more than 1000 iterative steps to obtain the solutions of $J^k \boldsymbol{z} = -\boldsymbol{f}(\boldsymbol{u}^{(k)})$, which implies that the Jacobian matrices are very ill-conditioned. For the method $\mathcal{P}$, the Iter2 is greatly reduced compared with the method $\mathcal{I}$. This means that our preconditioner P_ℓ is efficient for solving the Jacobian equations in Algorithm 1, but the Iter2 grows slightly fast in several cases such as $(\alpha, \lambda) = (1.9, 5)$. On the other hand, as seen from Table 5, the total CPU time of the method $\mathcal{P}$ is the smallest one among them. The eigenvalues of the initial Jacobian matrix J^0 and its preconditioned



(a) Eigenvalues of J^0 (b) Eigenvalues of $P_\ell^{-1}J^0$

(c) Eigenvalues of J^0 (d) Eigenvalues of $P_\ell^{-1}J^0$

Fig. 1: Spectra of J^0 and $P_\ell^{-1}J^0$, when $\alpha = 1.5$, $M = N = 65$ in Example 1. Top row: $\lambda = 0$; Bottom row: $\lambda = 5$.

matrix $P_\ell^{-1}J^0$ are drawn in Fig. 1. As can be seen, the eigenvalues of $P_\ell^{-1}J^0$ are clustered around 1.

Example 2. In this example, we consider Eq. (1.1) with $T = 1$, the initial value $u(x, 0) = \frac{4\exp(10x)}{(\exp(10x)+1)^2}$ ($x \in [-1, 1]$), the nonlinear source term $f(u(x,t), x, t) = -u(x,t)\,(1 - u(x,t))$ and the discontinuous coefficients

$$d_+(x) = \begin{cases} 1.5\exp(-x), & -1 \le x < 0, \\ 2\mathrm{sech}(x), & 0 \le x \le 1, \end{cases} \qquad d_-(x) = \begin{cases} \exp(x), & -1 \le x < 0, \\ 0.1 + \mathrm{sech}(-x), & 0 \le x \le 1. \end{cases}$$

Similar to Example 1, we regard the numerical solution on the finer mesh ($M = N = 1024$) as our exact solution.

It can be seen from Tables 6-7 that the convergence orders of the two implicit schemes are 1 in both time and space for the discontinuous coefficients. Tables 8 and 9 display the errors between NL-IES (2.3) and L-IES (2.4), and the rates of such errors can indeed reach 1. The performance of the method $\mathcal{P}$ shown in Table 10 is the best one among them in aspects of CPU time and the number of iterations. The Iter1 of the methods BS, $\mathcal{I}$ and $\mathcal{P}$ is small, which indicates that the initial vector $\boldsymbol{u}^{(0)}$ is a good enough initial value. As for the Iter2, the method $\mathcal{P}$ requires less iterative steps than the method $\mathcal{I}$ under the same termination condition. This illustrates that our proposed preconditioner P_ℓ is efficient and can accelerate solving the Jacobian equations in Algorithm 1. Furthermore, Fig. 2 displays the eigenvalues of J^0 and $P_\ell^{-1}J^0$.

Table 5: Results of different methods when $M = N$ for Example 1.

		BS		$\mathcal{I}$		$\mathcal{P}$	
(α, λ)	N	Iter1	Time	(Iter1, Iter2)	Time	(Iter1, Iter2)	Time
(1.1, 0)	129	5.0	0.509	(5.0, 230.4)	5.980	(5.0, 4.8)	0.199
	257	5.0	3.997	(5.0, 521.0)	53.085	(5.0, 6.6)	0.928
	513	5.0	35.110	(6.0, 688.8)	315.968	(5.0, 9.8)	4.704
	1025	†	†	‡	‡	(5.0, 16.2)	26.179
(1.5, 0)	129	4.0	0.448	(4.0, 483.3)	10.255	(4.0, 12.8)	0.399
	257	4.0	3.336	(5.0, 817.4)	84.057	(4.0, 25.5)	2.380
	513	4.0	29.445	‡	‡	(4.0, 62.3)	22.153
	1025	†	†	‡	‡	(4.0, 169.5)	207.366
(1.9, 0)	129	4.0	0.441	(5.0, 913.2)	27.381	(4.0, 9.3)	0.290
	257	4.0	3.393	‡	‡	(4.0, 22.3)	2.248
	513	4.0	29.532	‡	‡	(4.0, 66.5)	23.564
	1025	†	†	‡	‡	(4.0, 215.0)	262.269
(1.1, 5)	129	6.0	0.595	(6.0, 252.7)	9.361	(6.0, 3.8)	0.178
	257	6.0	4.666	(24.0, 332.5)	161.449	(6.0, 4.5)	0.760
	513	6.0	41.446	‡	‡	(6.0, 7.0)	4.022
	1025	†	†	‡	‡	(7.0, 13.4)	29.704
(1.5, 5)	129	4.0	0.454	(5.0, 628.8)	19.502	(5.0, 7.6)	0.283
	257	5.0	4.002	‡	‡	(5.0, 18.8)	2.381
	513	5.0	35.842	‡	‡	(5.0, 52.2)	23.144
	1025	†	†	‡	‡	(5.0, 150.4)	227.671
(1.9, 5)	129	4.0	0.457	(5.0, 944.4)	28.843	(4.0, 6.3)	0.190
	257	4.0	3.350	‡	‡	(4.0, 17.0)	1.685
	513	4.0	29.932	‡	‡	(4.0, 56.3)	19.871
	1025	†	†	‡	‡	(4.0, 218.3)	263.273
(1.1, 10)	129	6.0	0.691	(6.0, 242.8)	9.137	(6.0, 3.7)	0.171
	257	7.0	5.341	(7.0, 519.9)	74.813	(7.0, 4.3)	0.727
	513	7.0	47.171	‡	‡	(7.0, 6.1)	4.011
	1025	†	†	‡	‡	(7.0, 11.7)	26.043
(1.5, 10)	129	5.0	0.542	(5.0, 664.4)	20.579	(5.0, 5.4)	0.221
	257	5.0	4.007	(19.0, 373.5)	146.726	(5.0, 14.4)	1.820
	513	5.0	36.374	‡	‡	(5.0, 43.6)	19.333
	1025	†	†	‡	‡	(5.0, 143.2)	215.481
(1.9, 10)	129	4.0	0.484	(5.0, 898.4)	28.227	(4.0, 4.8)	0.153
	257	4.0	3.327	‡	‡	(4.0, 12.3)	1.253
	513	4.0	30.498	‡	‡	(4.0, 51.3)	18.470
	1025	†	†	‡	‡	(4.0, 186.0)	225.510

Table 6: The maximum norm errors and convergence orders for Example 2 with $h = 2^{-10}$.

		$\lambda = 0$				$\lambda = 1$				$\lambda = 5$				$\lambda = 10$			
		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)	
α	M	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$	$Err(\tau,h)$	$Order1$
1.1	64	3.2431E-02	–	3.3730E-02	–	6.8046E-03	–	4.9594E-03	–	4.3160E-03	–	2.3456E-03	–	3.4101E-03	–	1.3524E-03	–
	128	1.6340E-02	0.9890	1.6918E-02	0.9955	3.2092E-03	1.0843	2.3436E-03	1.0814	2.0253E-03	1.0916	1.1032E-03	1.0883	1.5961E-03	1.0953	6.3392E-04	1.0931
	256	7.3028E-03	1.1619	7.5422E-03	1.1655	1.3830E-03	1.2144	1.0110E-03	1.2129	8.7035E-04	1.2185	4.7469E-04	1.2166	6.8508E-04	1.2202	2.7231E-04	1.2191
	512	2.4886E-03	1.5531	2.5666E-03	1.5551	4.6229E-04	1.5809	3.3812E-04	1.5802	2.9052E-04	1.5830	1.5855E-04	1.5820	2.2853E-04	1.5839	9.0900E-05	1.5829
1.5	64	5.2358E-02	–	5.0868E-02	–	4.3947E-02	–	4.2489E-02	–	3.1695E-02	–	3.0122E-02	–	2.6224E-02	–	2.4699E-02	–
	128	2.7642E-02	0.9215	2.6909E-02	0.9187	2.3157E-02	0.9243	2.2433E-02	0.9215	1.6578E-02	0.9350	1.5849E-02	0.9264	1.3353E-02	0.9737	1.2661E-02	0.9641
	256	1.2690E-02	1.1232	1.2368E-02	1.1215	1.0624E-02	1.1241	1.0302E-02	1.1227	7.5608E-03	1.1327	7.2512E-03	1.1281	5.9884E-03	1.1569	5.6980E-03	1.1519
	512	4.4268E-03	1.5194	4.3191E-03	1.5178	3.6814E-03	1.5290	3.5748E-03	1.5270	2.6047E-03	1.5374	2.5021E-03	1.5351	2.0525E-03	1.5448	1.9541E-03	1.5440
1.9	64	8.5313E-02	–	8.3849E-02	–	8.3615E-02	–	8.2142E-02	–	8.0497E-02	–	7.9014E-02	–	7.8786E-02	–	7.7299E-02	–
	128	5.8082E-02	0.5547	5.7382E-02	0.5472	5.6620E-02	0.5625	5.5923E-02	0.5547	5.3634E-02	0.5858	5.2945E-02	0.5776	5.1839E-02	0.6039	5.1155E-02	0.5956
	256	2.9567E-02	0.9741	2.9299E-02	0.9697	2.8709E-02	0.9798	2.8444E-02	0.9753	2.6837E-02	0.9989	2.6580E-02	0.9942	2.5645E-02	1.0154	2.5392E-02	1.0105
	512	1.1116E-02	1.4114	1.1021E-02	1.4106	1.0791E-02	1.4117	1.0697E-02	1.4109	1.0083E-02	1.4123	9.9912E-03	1.4116	9.6308E-03	1.4130	9.5406E-03	1.4122
1.99	64	8.7279E-02	–	8.5852E-02	–	8.7128E-02	–	8.5699E-02	–	8.6886E-02	–	8.5455E-02	–	8.6792E-02	–	8.5360E-02	–
	128	6.2942E-02	0.4716	6.2243E-02	0.4639	6.2805E-02	0.4723	6.2106E-02	0.4645	6.2554E-02	0.4740	6.1855E-02	0.4663	6.2449E-02	0.4749	6.1750E-02	0.4671
	256	3.4261E-02	0.8775	3.3986E-02	0.8730	3.4176E-02	0.8779	3.3901E-02	0.8734	3.4008E-02	0.8792	3.3734E-02	0.8747	3.3932E-02	0.8800	3.3658E-02	0.8755
	512	1.2865E-02	1.4131	1.2768E-02	1.4124	1.2833E-02	1.4131	1.2736E-02	1.4124	1.2769E-02	1.4132	1.2673E-02	1.4124	1.2741E-02	1.4132	1.2644E-02	1.4125

Table 7: The maximum norm errors and convergence orders for Example 2 with $\tau = h$.

		$\lambda = 0$				$\lambda = 1$				$\lambda = 5$				$\lambda = 10$			
		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)		L-IES (2.4)		NL-IES (2.3)	
α	N	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$	$Err(\tau,h)$	$Order2$
1.1	64	1.0317E-01	–	1.0452E-01	–	1.3346E-01	–	1.3487E-01	–	2.1802E-01	–	2.1913E-01	–	3.1633E-01	–	3.1681E-01	–
	128	6.8449E-02	0.5919	6.8706E-02	0.6053	7.7164E-02	0.7904	7.7912E-02	0.7917	1.3089E-01	0.7361	1.3152E-01	0.7365	1.9729E-01	0.6811	1.9769E-01	0.6804
	256	4.5280E-02	0.5962	4.5395E-02	0.5979	3.8083E-02	1.0188	3.8428E-02	1.0197	6.6996E-02	0.9662	6.7307E-02	0.9665	1.0488E-01	0.9116	1.0511E-01	0.9113
	512	2.6552E-02	0.7701	2.6588E-02	0.7718	1.3841E-02	1.4602	1.3963E-02	1.4605	2.4969E-02	1.4239	2.5081E-02	1.4242	4.0448E-02	1.3746	4.0540E-02	1.3745
1.5	64	4.5283E-02	–	4.3847E-02	–	2.8941E-02	–	2.7535E-02	–	3.5242E-02	–	3.4077E-02	–	9.9995E-02	–	9.8520E-02	–
	128	2.3622E-02	0.9388	2.2923E-02	0.9357	1.4966E-02	0.9514	1.4277E-02	0.9476	1.8483E-02	0.9311	1.7949E-02	0.9249	5.3729E-02	0.8962	5.3096E-02	0.8918
	256	1.0920E-02	1.1132	1.0622E-02	1.1097	7.0537E-03	1.0852	6.7675E-03	1.0770	8.3827E-03	1.1407	8.1585E-03	1.1375	2.4510E-02	1.1323	2.4265E-02	1.1297
	512	3.8072E-03	1.5202	3.7073E-03	1.5186	2.4552E-03	1.5225	2.3625E-03	1.5183	2.8736E-03	1.5446	2.8000E-03	1.5429	8.4061E-03	1.5439	8.3295E-03	1.5426
1.9	64	8.2087E-02	–	8.0640E-02	–	7.8230E-02	–	7.6775E-02	–	6.5789E-02	–	6.4329E-02	–	5.0179E-02	–	4.8724E-02	–
	128	5.6383E-02	0.5419	5.5686E-02	0.5342	5.3996E-02	0.5349	5.3301E-02	0.5265	4.7243E-02	0.4777	4.6554E-02	0.4666	4.0154E-02	0.3215	3.9467E-02	0.3040
	256	2.8883E-02	0.9650	2.8616E-02	0.9605	2.7681E-02	0.9640	2.7417E-02	0.9591	2.4505E-02	0.9470	2.4248E-02	0.9410	2.1569E-02	0.8966	2.1315E-02	0.8888
	512	1.0852E-02	1.4123	1.0757E-02	1.4115	1.0404E-02	1.4118	1.0310E-02	1.4110	9.2250E-03	1.4095	9.1333E-03	1.4087	8.1620E-03	1.4020	8.0715E-03	1.4010
1.99	64	8.4400E-02	–	8.2988E-02	–	8.2930E-02	–	8.1518E-02	–	7.6184E-02	–	7.4777E-02	–	6.5522E-02	–	6.4128E-02	–
	128	6.1409E-02	0.4588	6.0712E-02	0.4509	6.0689E-02	0.4505	5.9992E-02	0.4423	5.7807E-02	0.3982	5.7111E-02	0.3888	5.3735E-02	0.2861	5.3038E-02	0.2739
	256	3.3581E-02	0.8708	3.3306E-02	0.8662	3.3280E-02	0.8668	3.3005E-02	0.8621	3.2186E-02	0.8448	3.1911E-02	0.8397	3.0799E-02	0.8030	3.0524E-02	0.7971
	512	1.2614E-02	1.4126	1.2517E-02	1.4119	1.2502E-02	1.4125	1.2405E-02	1.4118	1.2106E-02	1.4107	1.2009E-02	1.4099	1.1626E-02	1.4055	1.1529E-02	1.4047

Table 8: The maximum norm errors between NL-IES (2.3) and L-IES (2.4) for Example 2 with $h = 2^{-10}$.

		$\lambda = 0$		$\lambda = 1$		$\lambda = 5$		$\lambda = 10$	
α	M	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$
1.1	64	2.8218E-03	–	2.1570E-03	–	2.1603E-03	–	2.2519E-03	–
	128	1.5181E-03	0.8943	1.0877E-03	0.9877	1.0841E-03	0.9947	1.1289E-03	0.9962
	256	7.9372E-04	0.9356	5.4617E-04	0.9939	5.4301E-04	0.9974	5.6515E-04	0.9982
	512	4.0689E-04	0.9640	2.7367E-04	0.9969	2.7175E-04	0.9987	2.8275E-04	0.9991
1.5	64	1.6696E-03	–	1.6967E-03	–	1.6847E-03	–	1.6839E-03	–
	128	8.8651E-04	0.9133	8.7906E-04	0.9487	8.5788E-04	0.9736	8.5169E-04	0.9834
	256	4.5771E-04	0.9537	4.4768E-04	0.9735	4.3266E-04	0.9875	4.2827E-04	0.9918
	512	2.3261E-04	0.9765	2.2606E-04	0.9858	2.1752E-04	0.9921	2.1498E-04	0.9943
1.9	64	1.5498E-03	–	1.5600E-03	–	1.5723E-03	–	1.5782E-03	–
	128	8.0342E-04	0.9479	8.0002E-04	0.9634	7.9196E-04	0.9894	7.8715E-04	1.0036
	256	4.0968E-04	0.9717	4.0908E-04	0.9677	4.0733E-04	0.9592	4.0636E-04	0.9539
	512	2.0755E-04	0.9810	2.0649E-04	0.9863	2.0549E-04	0.9871	2.0549E-04	0.9837
1.99	64	1.5094E-03	–	1.5108E-03	–	1.5130E-03	–	1.5140E-03	–
	128	7.9818E-04	0.9192	7.9805E-04	0.9208	7.9781E-04	0.9233	7.9779E-04	0.9243
	256	4.0060E-04	0.9946	4.0067E-04	0.9941	4.0079E-04	0.9932	4.0089E-04	0.9928
	512	2.0409E-04	0.9730	2.0405E-04	0.9735	2.0399E-04	0.9743	2.0400E-04	0.9746

Table 9: The maximum norm errors between NL-IES (2.3) and L-IES (2.4) for Example 2 with $\tau = h$.

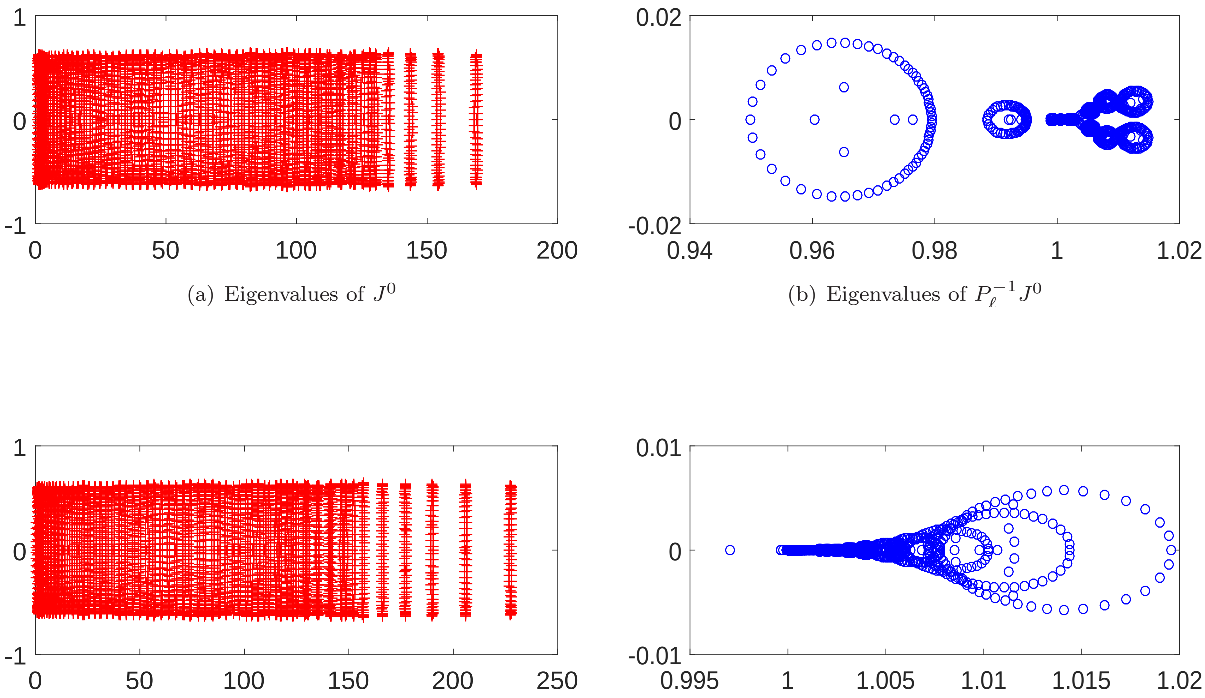
		$\lambda = 0$		$\lambda = 1$		$\lambda = 5$		$\lambda = 10$	
α	N	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$	$Err(\tau, h)$	$Order1$
1.1	64	2.6283E-03	–	1.8233E-03	–	1.7754E-03	–	2.0306E-03	–
	128	1.4530E-03	0.8551	9.7882E-04	0.8974	9.5216E-04	0.8989	9.4495E-04	1.1036
	256	7.7570E-04	0.9055	5.1535E-04	0.9255	5.0437E-04	0.9167	5.0857E-04	0.8938
	512	4.0344E-04	0.9431	2.6755E-04	0.9457	2.6380E-04	0.9350	2.7052E-04	0.9107
1.5	64	1.6456E-03	–	1.6295E-03	–	1.6426E-03	–	1.6259E-03	–
	128	8.7865E-04	0.9053	8.6650E-04	0.9112	8.4421E-04	0.9603	8.3915E-04	0.9542
	256	4.5527E-04	0.9486	4.4503E-04	0.9613	4.3027E-04	0.9724	4.2567E-04	0.9792
	512	2.3213E-04	0.9718	2.2560E-04	0.9801	2.1709E-04	0.9869	2.1452E-04	0.9886
1.9	64	1.5331E-03	–	1.5425E-03	–	1.5498E-03	–	1.5465E-03	–
	128	8.0010E-04	0.9382	7.9730E-04	0.9521	7.9150E-04	0.9694	7.8974E-04	0.9696
	256	4.0891E-04	0.9684	4.0829E-04	0.9655	4.0667E-04	0.9607	4.0596E-04	0.9600
	512	2.0742E-04	0.9792	2.0636E-04	0.9844	2.0531E-04	0.9861	2.0534E-04	0.9833
1.99	64	1.4936E-03	–	1.4938E-03	–	1.4887E-03	–	1.4758E-03	–
	128	7.9506E-04	0.9097	7.9510E-04	0.9098	7.9545E-04	0.9042	7.9606E-04	0.8905
	256	3.9970E-04	0.9921	3.9974E-04	0.9921	3.9979E-04	0.9925	3.9976E-04	0.9937
	512	2.0396E-04	0.9706	2.0392E-04	0.9711	2.0387E-04	0.9716	2.0388E-04	0.9714

Table 10: Results of different methods when $M = N$ for Example 2.

		BS		$\mathcal{I}$		$\mathcal{P}$	
(α, λ)	N	Iter1	Time	(Iter1, Iter2)	Time	(Iter1, Iter2)	Time
(1.1, 0)	129	5.0	0.539	(5.0, 207.0)	6.609	(5.0, 3.8)	0.167
	257	5.0	4.022	(5.0, 436.2)	42.799	(5.0, 6.0)	0.815
	513	5.0	37.223	‡	‡	(5.0, 9.2)	4.360
	1025	†	†	‡	‡	(5.0, 15.4)	24.702
(1.5, 0)	129	4.0	0.471	(4.0, 426.8)	10.178	(4.0, 11.3)	0.331
	257	4.0	3.413	‡	‡	(4.0, 24.3)	2.394
	513	4.0	30.196	‡	‡	(4.0, 62.0)	22.520
	1025	†	†	‡	‡	(4.0, 154.8)	190.537
(1.9, 0)	129	4.0	0.462	(4.0, 880.5)	22.810	(4.0, 8.0)	0.309
	257	4.0	3.426	‡	‡	(4.0, 21.8)	3.258
	513	4.0	30.240	‡	‡	(4.0, 66.3)	31.259
	1025	†	†	‡	‡	(4.0, 212.5)	362.669
(1.1, 5)	129	5.0	0.553	(5.0, 271.8)	8.562	(5.0, 2.8)	0.122
	257	5.0	4.105	(5.0, 584.2)	60.355	(5.0, 4.2)	0.536
	513	5.0	36.435	‡	‡	(5.0, 7.6)	3.632
	1025	†	†	‡	‡	(5.0, 14.2)	22.479
(1.5, 5)	129	4.0	0.470	(4.0, 542.5)	13.903	(4.0, 6.5)	0.189
	257	4.0	3.460	‡	‡	(4.0, 17.5)	1.704
	513	4.0	30.523	‡	‡	(5.0, 46.4)	20.737
	1025	†	†	‡	‡	(5.0, 136.2)	206.973
(1.9, 5)	129	4.0	0.452	(4.0, 907.0)	24.141	(4.0, 4.5)	0.167
	257	4.0	3.437	‡	‡	(4.0, 15.5)	2.241
	513	4.0	30.595	‡	‡	(4.0, 54.3)	25.433
	1025	†	†	‡	‡	(4.0, 197.5)	338.909
(1.1, 10)	129	5.0	0.568	(5.0, 277.0)	8.515	(5.0, 2.8)	0.109
	257	5.0	4.069	(5.0, 598.8)	58.637	(5.0, 3.6)	0.409
	513	5.0	36.612	‡	‡	(5.0, 6.6)	3.083
	1025	†	†	‡	‡	(6.0, 12.3)	23.298
(1.5, 10)	129	5.0	0.549	(5.0, 571.2)	17.305	(5.0, 4.2)	0.158
	257	5.0	4.097	‡	‡	(5.0, 12.2)	1.524
	513	5.0	36.738	‡	‡	(5.0, 38.6)	17.222
	1025	†	†	‡	‡	(5.0, 124.0)	191.098
(1.9, 10)	129	4.0	0.459	(5.0, 841.2)	26.614	(4.0, 3.3)	0.126
	257	4.0	3.434	‡	‡	(4.0, 11.5)	1.689
	513	4.0	30.590	‡	‡	(4.0, 46.0)	21.659
	1025	†	†	‡	‡	(4.0, 179.0)	307.717

6. Concluding remarks

The nonlinear all-at-once system arising from the nonlinear tempered fractional diffusion equations is studied. Firstly, the two implicit schemes (i.e., NL-IES (2.3) and L-IES (2.4)) in Section 2 are obtained through applying the finite difference method. Then, the stabilities and first-order convergences of such schemes are analyzed in Section 3 under several suitable assumptions. Secondly, for solving all the time steps in Eq. (2.3) simultaneously, the nonlinear all-at-once system (2.5) is derived from it. Then, Newton's method is employed to solve this system (2.5). Once the method is used to solve such the nonlinear system, the following two basic problems need to be addressed: 1. How to find a good initial value for Newton's method? 2. How to fast solving the Jacobian equations in Newton's method? As for the first problem, the value, which is constructed by interpolating the solution of L-IES (2.4) on the coarse grid, is chosen as



(a) Eigenvalues of J^0 (b) Eigenvalues of $P_\ell^{-1}J^0$

(c) Eigenvalues of J^0 (d) Eigenvalues of $P_\ell^{-1}J^0$

Fig. 2: Spectra of J^0 and $P_\ell^{-1}J^0$, when $\alpha = 1.9$, $M = N = 65$ in Example 2. Top row: $\lambda = 0$; Bottom row: $\lambda = 10$.

the initial guess. For the second problem, the PBiCGSTAB method with the preconditioner P_ℓ is employed to accelerate solving the Jacobian equations, which is discussed in Section 4. On the one hand, numerical examples in Section 5 show that the convergence orders of two proposed schemes (both in continuous and discontinuous cases) can indeed reach 1 in both time and space. On the other hand, they also indicate that our preconditioning strategy is effective for solving (2.5) with continuous or discontinuous coefficients. However, the performance of P_ℓ are not satisfactory. The reason may be that the diagonal block matrix in the Jacobian matrix (can be rewritten as $\mathcal{A}$ plus this diagonal block matrix), which is resulted from $-\tau\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial \boldsymbol{u}}$, is not considered when designing P_ℓ in this work. Thus, a preconditioner designed with considering such the diagonal block matrix may be more effective to solve Eq. (2.5). In the future work, we will study along with this direction and give some relative theoretical analysis.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Nos. 61772003, 61876203 and 11801463) and the Fundamental Research Funds for the Central Universities (No. ZYGX2016J132).

References

[1] Á. Cartea, D. del Castillo-Negrete, Fluid limit of the continuous-time random walk with general Lévy jump distribution functions, Phys. Rev. E 76 (2007) 041105. doi:10.1103/PhysRevE.76.041105.

[2] B. Baeumer, M. M. Meerschaert, Tempered stable Lévy motion and transient super-diffusion, J. Comput. Appl. Math. 233 (2010) 2438–2448.

[3] M. M. Meerschaert, A. Sikorskii, Stochastic Models for Fractional Calculus, De Gruyter, Berlin, 2012.

[4] I. Podlubny, Fractional Differential Equations, Vol. 198, Academic Press, San Diego, CA, 1998.

[5] A. Chakrabarty, M. M. Meerschaert, Tempered stable laws as random walk limits, Stat. Probab. Lett. 81 (2011) 989–997.

[6] M. Zheng, G. E. Karniadakis, Numerical methods for SPDEs with tempered stable processes, SIAM J. Sci. Comput. 37 (2015) A1197–A1217.

[7] P. Carr, H. Geman, D. B. Madan, M. Yor, The fine structure of asset returns: An empirical investigation, J. Business 75 (2002) 305–332.

[8] P. Carr, H. Geman, D. B. Madan, M. Yor, Stochastic volatility for Lévy processes, Math. Financ. 13 (2003) 345–382.

[9] W. Wang, X. Chen, D. Ding, S.-L. Lei, Circulant preconditioning technique for barrier options pricing under fractional diffusion models, Int. J. Comput. Math. 92 (2015) 2596–2614.

[10] H. Zhang, F. Liu, I. Turner, S. Chen, The numerical simulation of the tempered fractional Black–Scholes equation for European double barrier option, Appl. Math. Model. 40 (2016) 5819–5834.

[11] M. M. Meerschaert, Y. Zhang, B. Baeumer, Tempered anomalous diffusion in heterogeneous systems, Geophys. Res. Lett. 35 (2008) L17403. doi:10.1029/2008GL034899.

[12] R. Metzler, J. Klafter, The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics, J. Phys. A-Math. Theor. 37 (2004) R161. doi:10.1088/0305-4470/37/31/R01.

[13] Y. Zhang, M. M. Meerschaert, Gaussian setting time for solute transport in fluvial systems, Water Resour. Res. 47 (2011) W08601. doi:10.1029/2010WR010102.

[14] Y. Zhang, M. M. Meerschaert, A. I. Packman, Linking fluvial bed sediment transport across scales, Geophys. Res. Lett. 39 (2012) L20404. doi:10.1029/2012GL053476.

[15] Y.-L. Zhao, P.-Y. Zhu, W.-H. Luo, A fast second-order implicit scheme for non-linear time-space fractional diffusion equation with time delay and drift term, Appl. Math. Comput. 336 (2018) 231–248.

[16] X.-M. Gu, T.-Z. Huang, C.-C. Ji, B. Carpentieri, A. A. Alikhanov, Fast iterative method with a second-order implicit difference scheme for time-space fractional convection–diffusion equation, J. Sci. Comput. 72 (2017) 957–985.

[17] M. Li, X.-M. Gu, C. Huang, M. Fei, G. Zhang, A fast linearized conservative finite element method for the strongly coupled nonlinear fractional Schrödinger equations, J. Comput. Phys. 358 (2018) 256–282.

[18] A. Cartea, D. del Castillo-Negrete, Fractional diffusion models of option prices in markets with jumps, Physica A 374 (2007) 749–763.

[19] O. Marom, E. Momoniat, A comparison of numerical solutions of fractional diffusion models in finance, Nonlinear Anal.-Real World Appl. 10 (2009) 3435–3442.

[20] C. Li, W. Deng, High order schemes for the tempered fractional diffusion equations, Adv. Comput. Math. 42 (2016) 543–572.

[21] M. K. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, New York, NY, 2004.

[22] R. Chan, X.-Q. Jin, An Introduction to Iterative Toeplitz Solvers, SIAM, Philadelphia, PA, 2007.

[23] S.-L. Lei, D. Fan, X. Chen, Fast solution algorithms for exponentially tempered fractional diffusion equations, Numer. Meth. Part. Differ. Equ. 34 (2018) 1301–1323.

[24] W. Qu, S.-L. Lei, On CSCS-based iteration method for tempered fractional diffusion equations, Jpn. J. Ind. Appl. Math. 33 (2016) 583–597.

[25] X.-M. Gu, T.-Z. Huang, H.-B. Li, L. Li, W.-H. Luo, On k-step CSCS-based polynomial preconditioners for Toeplitz linear systems with application to fractional diffusion equations, Appl. Math. Lett. 42 (2015) 53–58.

[26] X.-M. Gu, T.-Z. Huang, X.-L. Zhao, H.-B. Li, L. Li, Strang-type preconditioners for solving fractional diffusion equations by boundary value methods, J. Comput. Appl. Math. 277 (2015) 73–86.

[27] X.-L. Zhao, T.-Z. Huang, S.-L. Wu, Y.-F. Jing, DCT-and DST-based splitting methods for Toeplitz systems, Int. J. Comput. Math. 89 (2012) 691–700.

[28] M. J. Gander, L. Halpern, Time parallelization for nonlinear problems based on diagonalization, in: C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, O. B. Widlund (Eds.), Domain Decomposition Methods in Science and Engineering XXIII, Springer-Verlag, 2017, pp. 163–170.

[29] S. Wu, Toward parallel coarse grid correction for the parareal algorithm, SIAM J. Sci. Comput. 40 (2018) A1446–A1472.

[30] M. J. Gander, 50 years of time parallel time integration, in: T. Carraro, M. Geiger, S. Körkel, R. Rannacher (Eds.), Multiple Shooting and Time Domain Decomposition Methods, Springer-Verlag, 2015, pp. 69–114.

[31] L. Banjai, D. Peterseim, Parallel multistep methods for linear evolution problems, IMA J. Numer. Anal. 32 (2012) 1217C1240.

[32] E. McDonald, J. Pestana, A. Wathen, Preconditioning and iterative solution of all-at-once systems for evolutionary partial differential equations, SIAM J. Sci. Comput. 40 (2) (2018) A1012–A1033.

[33] R. Ke, M. K. Ng, H.-W. Sun, A fast direct method for block triangular Toeplitz-like with tri-diagonal block systems from time-fractional partial differential equations, J. Comput. Phys. 303 (2015) 203–211.

[34] X. Lu, H.-K. Pang, H.-W. Sun, Fast approximate inversion of a block triangular Toeplitz matrix with applications to fractional sub-diffusion equations, Numer. Linear Algebr. Appl. 22 (2015) 866–882.

[35] Y.-C. Huang, S.-L. Lei, A fast numerical method for block lower triangular Toeplitz with dense Toeplitz blocks system with applications to time-space fractional diffusion equations, Numer. Algorithms 76 (2017) 605–616.

[36] X. Lu, H.-K. Pang, H.-W. Sun, S.-W. Vong, Approximate inversion method for time-fractional subdiffusion equations, Numer. Linear Algebr. Appl. 25 (2018) e2132.

[37] Y.-L. Zhao, P.-Y. Zhu, X.-M. Gu, X.-L. Zhao, J. Cao, A limited-memory block bi-diagonal Toeplitz preconditioner for block lower triangular Toeplitz system from time-space fractional diffusion equation, J. Comput. Appl. Math. (2018) 17 pages (in revision).

[38] F. Sabzikar, M. M. Meerschaert, J. Chen, Tempered fractional calculus, J. Comput. Phys. 293 (2015) 14–28.

[39] C. T. Kelley, Solving Nonlinear Equations with Newton's Method, SIAM, Philadelphia, PA, 2003.

[40] H. A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.

[41] P. Zhuang, F. Liu, V. Anh, I. Turner, Numerical methods for the variable-order fractional advection-diffusion equation with a nonlinear source term, SIAM J. Numer. Anal. 47 (2009) 1760–1781.

[42] R. S. Varga, Geršgorin and His Circles, Springer-Verlag, Berlin, 2004.

[43] J. Xu, A novel two-grid method for semilinear elliptic equations, SIAM J. Sci. Comput. 15 (1994) 231–237.

[44] J. Xu, Two-grid discretization techniques for linear and nonlinear pdes, SIAM J. Numer. Anal. 33 (1996) 1759–1777.

[45] D. Kim, E.-J. Park, B. Seo, A unified framework for two-grid methods for a class of nonlinear problems, Calcolo 55 (2018) 45. `doi:10.1007/s10092-018-0287-y`.