



# An efficient numerical method on modified space-time sparse grid for time-fractional diffusion equation with nonsmooth data

Bi-Yun Zhu<sup>1</sup> · Ai-Guo Xiao<sup>1</sup> · Xue-Yang Li<sup>1</sup>

Received: 9 August 2022 / Accepted: 25 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

In this paper, we focus on developing a high efficient algorithm for solving  $d$ -dimension time-fractional diffusion equation (TFDE). For TFDE, the initial function or source term is usually not smooth, which can lead to the low regularity of exact solution. And such low regularity has a marked impact on the convergence rate of numerical method. In order to improve the convergence rate of the algorithm, we introduce the space-time sparse grid (STSG) method to solve TFDE. In our study, we employ the sine basis and the linear element basis for spatial discretization and temporal discretization, respectively. The sine basis can be divided into several levels, and the linear element basis can lead to the hierarchical basis. Then, the STSG can be constructed through a special tensor product of the spatial multilevel basis and the temporal hierarchical basis. Under certain conditions, the function approximation on standard STSG can achieve the accuracy order  $\mathcal{O}(2^{-J}J)$  with  $\mathcal{O}(2^{JJ})$  degrees of freedom (DOF) for  $d = 1$  and  $\mathcal{O}(2^{Jd})$  DOF for  $d > 1$ , where  $J$  denotes the maximal level of sine coefficients. However, if the solution changes very rapidly at the initial moment, the standard STSG method may reduce accuracy or even fail to converge. To overcome this, we integrate the full grid into the STSG, and obtain the modified STSG. Finally, we obtain the fully discrete scheme of STSG method for solving TFDE. The great advantage of the modified STSG method can be shown in the comparative numerical experiment.

---

✉ Xue-Yang Li  
lix1217@xtu.edu.cn

Bi-Yun Zhu  
420737336@qq.com

Ai-Guo Xiao  
xag@xtu.edu.cn

<sup>1</sup> School of Mathematics and Computational Science & National Center for Applied Mathematics in Hunan & Hunan Key Laboratory for Computation and Simulation in Science and Engineering, Xiangtan University, Xiangtan, Hunan 411105, China

**Keywords** Time-fractional diffusion equation · Space-time sparse grid · Sine pseudospectral method · L1 difference method · Hierarchical basis

**Mathematics Subject Classification (2010)** 35R11 · 65M70 · 65T40 · 68Q25

## 1 Introduction

Consider the  $d$ -dimensional ( $d = 1, 2, 3$ ) TFDE

$$D_t^\alpha u = a\Delta u + f(u, \mathbf{x}, t), \quad (\mathbf{x}, t) \in \Omega \times (0, T] \tag{1}$$

with the homogeneous boundary condition

$$u(\mathbf{x}, t) = 0, \quad (\mathbf{x}, t) \in \partial\Omega \times (0, T] \tag{2}$$

and the initial condition

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

where  $T > 0$  is a fixed final time,  $a > 0$  is the diffusion coefficient,  $\Delta$  is the Laplacian,  $f(u, \mathbf{x}, t)$  is the source term or reaction term, and  $\partial\Omega$  denotes the boundary of the rectangular domain  $\Omega$ . Without loss of generality, we always assume

$$\Omega = (0, 1)^d$$

in this paper. Here the operator  $D_t^\alpha$  denotes the Caputo fractional derivative of order  $\alpha \in (0, 1)$  in temporal direction

$$D_t^\alpha u(\cdot, t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} \frac{\partial}{\partial s} u(\cdot, s) ds, \tag{3}$$

where  $\Gamma(z)$  is the Gamma function defined by

$$\Gamma(z) = \int_0^{+\infty} s^{z-1} e^{-s} ds, \quad \text{Re}(z) > 0.$$

The classical diffusion model is based on the assumption of Brownian motion. In Brownian motion, the distribution of substance has linear growth of the mean squared particle displacement with the time  $t$ . However, a lot of physical processes [6, 28] cannot be accurately described by Brownian motion, and they have superlinear or sub-linear mean square particle displacements with the time  $t$ . In contrast, the fractional diffusion equation can describe these problems more accurately than integer-order diffusion equation, and it has great application value. In particular, for TFDE, it has been used to describe many models of practical applications, such as the electron transport in Xerox photocopier [28], Modeling the epidemic control measures in overcoming COVID-19 outbreaks [34], simulating the viscoelastic liquid [16], the dynamics of latent liquidity in financial markets [2], and so on.

In recent years, there many researches about the numerical methods for fractional problem. The key for solving TFDE is how to discretize the time-fractional derivative. A classical discrete method is L1 difference method [17]. Kopteva [14] gave a complete convergence analysis of L1 difference method for standard TFDE. Mehandiratta et al. [19–22] extended the method to TFDE on metric star graph. Ali et al. [1] consider the variable-order fractional modified sub-diffusion equation. Moreover, other discretization of time-fractional derivative such as generalized Jacobi spectral method [4, 18] has also been used for solving TFDE. The time memory property of TFDE imply that it requires a lot of computational cost to approximate the time-fractional derivative, and so the research of fast algorithm is important. Jiang et al. [11] used the sum of exponentials approximation to obtain the fast convolution algorithm. Salama et al. [24–27] used hybrid Laplace transform-finite difference method to obtain the high efficient scheme.

In this paper, we mainly consider the TFDE with nonsmooth data, and there are two considerable problems. One problem is the memory of time-fractional derivatives, which means that the nonsmooth data will affect the spatial regularity of solutions for a long time. Thus, some spatial discrete methods, such as spectral method, high-order finite element method, etc, will be difficult to give full play to their high-accuracy advantages. The other problem is that the solution has singularity at the initial time if the source term  $f(u, \mathbf{x}, t)$  is not compatible with the initial function [31]. For general cases, the direct difference discretization cannot exceed the first-order accuracy in temporal direction, even if the higher-order temporal basis function is selected. Therefore, many researches are interested in the improved method of temporal discretization. A famous method is to use the graded meshes instead of uniform meshes [14, 32, 37], and such method can effectively deal with the initial singularity by using the non-equidistant time stepsize. More recently, Yan et al. [35] developed the corrected L1 difference method, and this method corrects the first step of the standard L1 difference scheme and effectively overcomes the influence of initial singularity on the accuracy of the algorithm. Based on such idea, the high-order BDF method can also be obtained [12]. These corrected schemes are also introduce in [13]. However, the algorithm with high-order accuracy requires that the solution of the equation satisfy the corresponding regularity. For some cases, for example, if the source term  $f(u, \mathbf{x}, t)$  is singular in the temporal direction, the advantages of the higher-order algorithm may be greatly weakened.

For the conventional time-stepping algorithm, which is called the full grid (FG) method in this paper, the total DOF is equal to the spatial DOF multiplied by the temporal DOF. When the convergence rate of the algorithm in spatial direction and temporal direction is seriously limited, the high-accuracy computation of FG method implies the expensive computational cost. In order to reduce the total DOF of high-accuracy computation, we introduce the STSG method.

The sparse grid method is presented by Smolyak [30], and it has been used to break “the curse of dimensionality” [3]. For sparse grid method, the selection of multilevel basis is necessary. The difference among different levels can be represented by hierarchical basis [36], which can lead to the hierarchical transform and hierarchical coefficients. The hierarchical coefficients are generally decay from level to level, and the decaying rate depends on the regularity of function and on the selection of

basis functions. The numerical approximation requires a proper truncation of hierarchical coefficients indices. If the decaying rate of hierarchical coefficients is slow and anisotropic, the hyperbolic-type truncation may achieve nice results, and such hyperbolic-type approximation can lead to the construction of sparse grid. In particular, for time-dependent problems, if the decaying rate of hierarchical coefficient is anisotropic between spatial direction and temporal direction, then the STSG method can be considered in approximating these problems [8, 9]. The STSG method does not directly improve the convergence order in spatial or temporal direction. It only makes an appropriate choice for the basis functions of tensor product type (the product of spatial basis function and time basis function), so as to reduce the total DOF of the fully discrete scheme.

As far as we know, although there are a few researches about sparse grid method for space-fractional problems [10], there is no literature about STSG method for time-fractional problems or other time memory problems. Therefore, our purpose is to introduce the STSG method into the evolution equations with time memory property, and the TFDE considered in this paper is an important class of these equations. In the work of this paper, we combine the STSG method with the conventional algorithm of TFDE, and obtain a new scheme for solving TFDE. When the regularity of TFDE in the spatial direction is relatively weak, the STSG method has great advantages over the FG method. Compared with some temporal higher-order algorithms for solving TFDE, such as corrected L1 difference method, the comparative advantages of STSG are reflected when the value of  $\alpha$  is relatively large or the source term has strong singularity in temporal direction. Moreover, if the solution changes very rapidly at the initial moment, the standard STSG method may reduce accuracy or even fail to converge. Therefore, we constructed the modified STSG. The numerical results show that the modified STSG method has wider applicability than the standard STSG method.

Besides, the sine pseudospectral method is used for spatial discretization in our research. For low regularity problem, although the high-accuracy advantage is greatly weakened, the sine pseudospectral method still has its value, because the obtained algebraic equations are more convenient for computation. Moreover, we use the hierarchical basis of linear element for temporal discretization. Such selection of spatial basis and temporal basis is very common in FG method, but it is very rare in the existing literatures about STSG researches. Therefore, the algorithm of function approximation is also a research point of this paper.

The rest of the paper is organized as follows. In Section 2, we illustrate why it is necessary to use STSG method for solving TFDE. In Section 3, by introducing the temporal hierarchical basis and the spatial multilevel basis, the standard STSG is constructed. Then, we obtain the discrete sine transform (DST) algorithm and the function approximation error estimate on STSG. In Section 4, we obtain the modified STSG, and the L1 difference method is used to obtain the algorithm for computing the Caputo derivative. In Section 5, by using the L1 difference/sine pseudospectral method on modified STSG, we obtain the fully discrete scheme for solving TFDE. In Section 6, several numerical experiments are given to show the great advantage of STSG method.

## 2 Motivation

Let  $v(x)$  be a periodic odd function with a period of 2, and  $v \in \mathcal{L}_0^2(0, 1)$ . Then, the sine coefficients are defined as

$$\hat{v}_k = 2 \int_0^1 v(x) \sin k\pi x dx, \quad k \in \mathbb{N}_+,$$

where  $\mathbb{N}_+$  denotes the set of positive integer numbers. Here the absolute value of the  $k$ -th coefficient  $|\hat{v}_k|$  is also called the  $k$ -th frequency spectrum for  $k \in \mathbb{N}_+$ . It is well-known that the decaying rate of the frequency spectrum is directly related to the regularity of the function in periodic domain. Then, we consider the following problem.

**Example 1** Consider the one-dimensional TFDE

$$D_t^\alpha u = 0.1 \frac{\partial^2}{\partial x^2} u, \quad (x, t) \in (0, 1) \times (0, 1],$$

with the homogeneous boundary condition (2) and the initial function

$$u_0(x) = \delta(x - 0.5), \quad x \in (0, 1),$$

where  $\delta(x)$  denotes the Dirac  $\delta$  function.

The diffusion equation with Dirac  $\delta$  initial function is very common in practical problems, and it means that the diffusing substance is concentrated in one point at the initial time. From the mathematical point of view, the Dirac  $\delta$  function can be defined as an element of dual space. According to Theorem 2.1 of [13], the weak solution of Example 1 satisfies a certain regularity. For integer-order diffusion equation ( $\alpha = 1$ ), the solution becomes a smooth function (infinitely differentiable function) after a short moment. But for fractional problem ( $0 < \alpha < 1$ ), because the Caputo fractional derivative (3) has the property of time memory, the singularity of the initial function will have a lasting effect on the regularity of the solution. Figure 1 clearly shows that the solution of equation in the case  $\alpha = 0.5$  is not smooth. Here we are very concerned about the decaying property of frequency spectrum. Figure 2 shows the decaying property of frequency spectrum at  $t = 0.1$ . For integer-order diffusion equation, the frequency spectrum has exponentially decaying rate, and it implies that we can achieve very high spatial approximation accuracy with only a few spatial basis functions. But for TFDE, the decaying rate of frequency spectrum is greatly affected by the low regularity of solution. Therefore, if the conventional algorithm is used to solve such problem, the high-accuracy approximation requires a large number of spatial basis functions, which may bring an expensive computational cost.

However, the algebraically decaying rate of frequency spectrum also implies that the high-frequency coefficients are relatively small. For high-accuracy approximation, although the high-frequency coefficients cannot be ignored, their computation can allow larger relative error. The main idea of STSG method is to discretize the

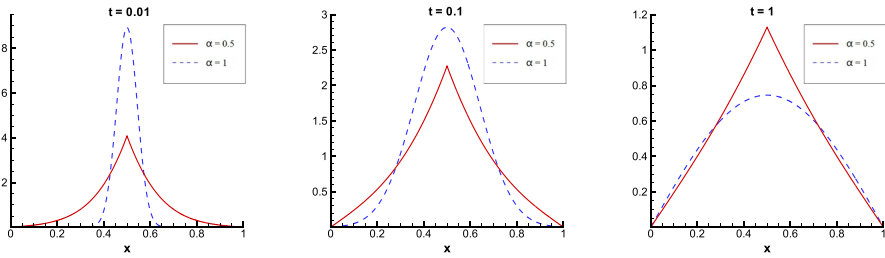


Fig. 1 The solution at different time in Example 1 with  $\alpha = 1$  and  $\alpha = 0.5$

lower-frequency coefficients with smaller time stepsize and to discretize the higher-frequency coefficients with larger time stepsize, whereas the accuracy is close to that by discretizing all coefficients with smallest time stepsize. In this way, the computational cost of high-accuracy computation can be greatly reduced.

### 3 Standard space-time sparse grid

Without loss of generality, we only consider the function defined on time interval  $[0, 1]$  in this section. Let  $\theta_{s,\tau}(t)$  be a linear element with center point  $s$  and width  $2\tau$ , i.e.,

$$\theta_{s,\tau}(t) = \begin{cases} 1 - \frac{1}{\tau}|t - s|, & \text{for } t \in (s - \tau, s + \tau), \\ 0, & \text{for else,} \end{cases} \quad s \in \mathbb{R}, \tau \in \mathbb{R}_+.$$

For any level  $J \in \mathbb{N}_0$  ( $\mathbb{N}_0$  denotes the natural number including zero), let  $\{\theta_{s,2^{-J}}\}_{s \in \mathcal{T}_J}$  be temporal basis of level  $J$ , where

$$\mathcal{T}_J = \{n2^{-J}\}_{n=0}^{2^J}.$$

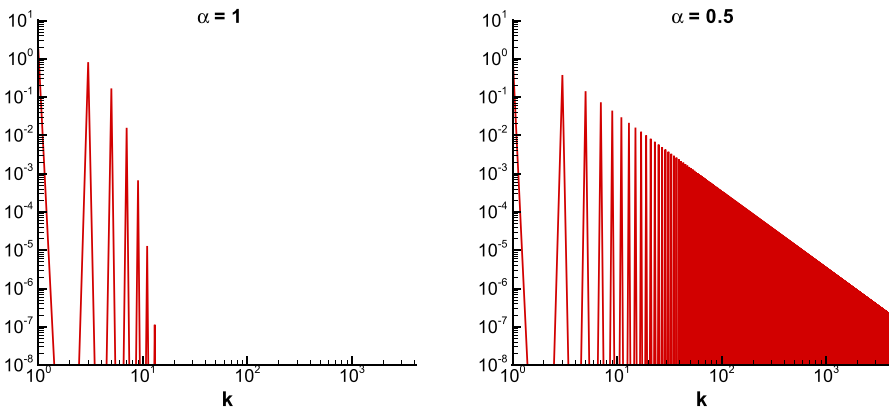


Fig. 2 The frequency spectrum at  $t = 0.1$  in Example 1

Obviously, the basis functions of adjacent levels satisfy the relation

$$\theta_{s,2^{1-J}}(t) = \theta_{s,2^{-J}}(t) + \frac{1}{2}\theta_{s-2^{-J},2^{-J}}(t) + \frac{1}{2}\theta_{s+2^{-J},2^{-J}}(t), \quad s \in \mathcal{T}_{J-1}. \quad (4)$$

Then, we can define the temporal linear element spaces  $\mathcal{W}_J$  and their hierarchical increment spaces  $\check{\mathcal{W}}_J$  as

$$\mathcal{W}_J = \text{span} \{ \theta_{s,2^{-J}} \}_{s \in \mathcal{T}_J}, \quad \check{\mathcal{W}}_J = \text{span} \{ \theta_{s,2^{-j}} \}_{s \in \check{\mathcal{T}}_J}, \quad J \in \mathbb{N}_0,$$

where

$$\check{\mathcal{T}}_J = \begin{cases} \mathcal{T}_0, & \text{for } J = 0, \\ \mathcal{T}_J \setminus \mathcal{T}_{J-1}, & \text{for } J > 0. \end{cases}$$

Here  $\{ \theta_{s,2^{-j}} \}_{s \in \check{\mathcal{T}}_j}$  is the temporal hierarchical basis of level  $J$ . For any function  $W \in \mathcal{W}_J$ , we have

$$W(t) = \sum_{s \in \mathcal{T}_J} W(s)\theta_{s,2^{-J}}(t).$$

Then, using the relation (4), we can easily prove that the function  $W(t)$  can be written as the linear combination of multilevel hierarchical basis

$$W(t) = \sum_{j=0}^J \sum_{s \in \check{\mathcal{T}}_j} \check{W}_s \theta_{s,2^{-j}}(t), \quad (5)$$

where the hierarchical coefficients are given by the hierarchical transform

$$\check{W}_s = \begin{cases} W(s), & \text{for } s \in \mathcal{T}_0, \\ W(s) - \frac{1}{2}(W(s - 2^{-j}) + W(s + 2^{-j})), & \text{for } s \in \check{\mathcal{T}}_j, j > 0. \end{cases} \quad (6)$$

Let  $\{ \varphi_{\mathbf{k}} \}_{\mathbf{k} \in \mathbb{N}_+^d}$  be the  $d$ -dimensional sine basis, i.e.,

$$\varphi_{\mathbf{k}}(\mathbf{x}) = \prod_{j=1}^d \sin \pi k_j x_j, \quad \mathbf{k} \in \mathbb{N}_+^d, \quad \mathbf{x} \in \Omega,$$

where  $\mathbb{N}_+$  denotes the set of positive integers. Then, any function  $v \in \mathcal{L}_0^2 = \mathcal{L}_0^2(\Omega)$  can be expanded as

$$v(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_+^d} \hat{v}_{\mathbf{k}} \varphi_{\mathbf{k}}(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

where the multi-dimensional sine coefficients are given by

$$\hat{v}_{\mathbf{k}} = 2^d \int_{\Omega} v(\mathbf{x}) \varphi_{\mathbf{k}}(\mathbf{x}) d\mathbf{x}, \quad k \in \mathbb{N}_+^d. \quad (7)$$

For any level  $J \in \mathbb{N}_0$ , we can define the index set

$$\mathcal{K}_J = \{1, 2, \dots, 2^J - 1\}^d,$$

the spatial grid points

$$\mathcal{X}_J = \{2^{-J}, 2 \cdot 2^{-J}, \dots, (2^J - 1)2^{-J}\}^d,$$

and the spatial function space

$$\mathcal{V}_J = \text{span}\{\varphi_{\mathbf{k}}\}_{\mathbf{k} \in \mathcal{K}_J}.$$

In particular, we assume that

$$\mathcal{K}_0 = \emptyset, \quad \mathcal{X}_0 = \emptyset, \quad \mathcal{V}_0 = \{0\}.$$

Then, the projection operator and the interpolation operator are defined as

$$\begin{aligned} \mathcal{P}_J : \mathcal{L}_0^2 \Omega &\rightarrow \mathcal{V}_J \\ : v(\mathbf{x}) &\mapsto \mathcal{P}_J v(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}_J} \hat{v}_{\mathbf{k}} \varphi_{\mathbf{k}}(\mathbf{x}), \end{aligned} \tag{8}$$

and

$$\begin{aligned} \mathcal{I}_J : \mathcal{L}_0^2 \Omega &\rightarrow \mathcal{V}_J \\ : v(\mathbf{x}) &\mapsto V(\mathbf{x}) = \mathcal{I}_J v(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}_J} \hat{V}_{\mathbf{k}} \varphi_{\mathbf{k}}(\mathbf{x}), \end{aligned} \tag{9}$$

respectively, where  $\{\hat{v}_{\mathbf{k}}\}_{\mathbf{k} \in \mathcal{K}_J}$  is given by (7), and the discrete sine coefficients  $\{\hat{V}_{\mathbf{k}}\}_{\mathbf{k} \in \mathcal{K}_J}$  are given by the DST

$$\hat{V}_{\mathbf{k}} = 2^{-Jd} \sum_{\xi \in \mathcal{X}_J} v(\xi) \varphi_{\mathbf{k}}(\xi).$$

It is well-know that the algorithm of fast Fourier transform can be used in the computation of DST, so the computational cost of DST is of order  $\mathcal{O}(2^{Jd} J)$ .



Using the tensor product construction of the multilevel basis in space and the hierarchical basis in time, we can construct the standard STSG as

$$\begin{aligned}
 \text{index set: } \mathcal{C}_J &= \bigcup_{j=0}^{J-1} \mathcal{K}_{J-j} \times \check{\mathcal{T}}_j, \\
 \text{grid points: } \mathcal{G}_J &= \bigcup_{j=0}^{J-1} \mathcal{X}_{J-j} \times \check{\mathcal{T}}_j, \\
 \text{function space: } \mathcal{U}_J &= \bigoplus_{j=0}^{J-1} \mathcal{V}_{J-j} \times \check{\mathcal{W}}_j,
 \end{aligned} \tag{10}$$

with the level  $J$ . Figure 3 gives an example of STSG.

Any function  $U \in \mathcal{U}_J$  can be written as the form

$$U(\mathbf{x}, t) = \sum_{j=0}^{J-1} \sum_{s \in \check{\mathcal{T}}_j} \theta_{s, 2^{-j}}(t) \sum_{\mathbf{k} \in \mathcal{K}_{J-j}} \varphi_{\mathbf{k}}(\mathbf{x}) \check{U}_{\mathbf{k}, s}, \tag{11}$$

where  $\{\check{U}_{\mathbf{k}, s}\}_{\mathbf{k} \in \mathcal{K}_{J-j}, s \in \check{\mathcal{T}}_j}$  are the sine coefficients of  $U(\mathbf{x}, t)$  on the space  $\mathcal{V}_{J-j} \times \check{\mathcal{W}}_j$  for  $j = 0, 1, \dots, J - 1$ . Let

$$\check{\mathcal{K}}_j = \mathcal{K}_j \setminus \mathcal{K}_{j-1}, \quad j = 1, 2, \dots, J,$$

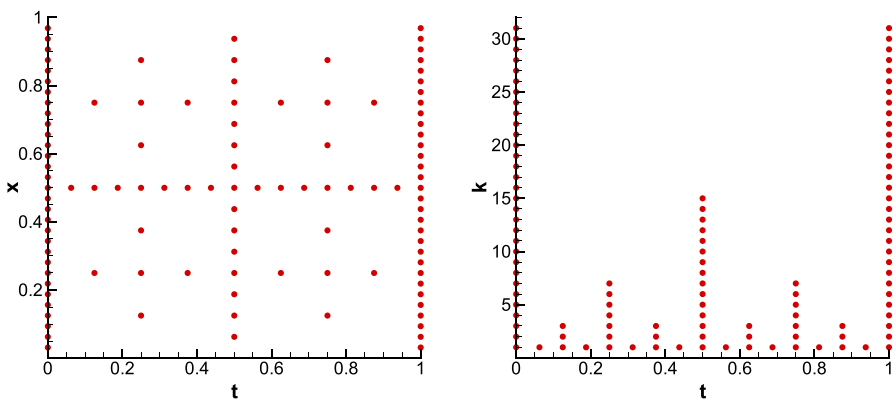


Fig. 3 Grid points (left) and index set (right) of STSG with  $d = 1$  and  $J = 5$

then

$$\begin{aligned}
 U(\mathbf{x}, t) &= \sum_{j=0}^{J-1} \sum_{s \in \check{T}_j} \theta_{s,2^{-j}}(t) \sum_{j'=1}^{J-j} \sum_{\mathbf{k} \in \check{K}_{j'}} \varphi_{\mathbf{k}}(\mathbf{x}) \check{U}_{\mathbf{k},s} \\
 &= \sum_{j'=1}^J \sum_{\mathbf{k} \in \check{K}_{j'}} \varphi_{\mathbf{k}}(\mathbf{x}) \sum_{j=0}^{J-j'} \sum_{s \in \check{T}_j} \theta_{s,2^{-j}}(t) \check{U}_{\mathbf{k},s} \\
 &= \sum_{\mathbf{k} \in \mathcal{K}_J} \varphi_{\mathbf{k}}(\mathbf{x}) \hat{U}_{\mathbf{k}}(t).
 \end{aligned} \tag{12}$$

Therefore, the sine coefficients of  $U(\mathbf{x}, t)$  satisfy

$$\hat{U}_{\mathbf{k}}(t) = \sum_{j=0}^{J-j'} \sum_{s \in \check{T}_j} \theta_{s,2^{-j}}(t) \check{U}_{\mathbf{k},s}, \quad \mathbf{k} \in \check{K}_{j'}, \quad j' = 1, 2, \dots, J.$$

From (5) and (6), it follows from (12) that

$$\check{U}_{\mathbf{k},s} = \begin{cases} \hat{U}_{\mathbf{k}}(s), & \text{for } s \in \check{T}_0, \\ \hat{U}_{\mathbf{k}}(s) - \frac{1}{2}(\hat{U}_{\mathbf{k}}(s - 2^{-j}) + \hat{U}_{\mathbf{k}}(s + 2^{-j})), & \text{for } s \in \check{T}_j, \quad j > 0. \end{cases} \tag{13}$$

Moreover, let

$$\check{U}_s(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}_{J-j}} \varphi_{\mathbf{k}}(\mathbf{x}) \check{U}_{\mathbf{k},s}, \quad s \in \check{T}_j, \quad j = 0, 1, \dots, J - 1.$$

Then, (11) yields

$$U(\mathbf{x}, t) = \sum_{j=0}^{J-1} \sum_{s \in \check{T}_j} \check{U}_s(\mathbf{x}) \theta_{s,2^{-j}}(t) = \sum_{s \in \mathcal{T}_{J-1}} U(\mathbf{x}, s) \theta_{s,2^{1-j}}(t), \tag{14}$$

and it is easy to obtain the hierarchical transform on grid function as

$$\check{U}_s(\mathbf{x}) = \begin{cases} U(\mathbf{x}, s), & \text{for } s \in \check{T}_0, \\ U(\mathbf{x}, s) - \frac{1}{2}(U(\mathbf{x}, s - 2^{-j}) + U(\mathbf{x}, s + 2^{-j})), & \text{for } s \in \check{T}_j, \quad j > 0, \end{cases} \tag{15}$$

For simplicity, we use the form of vector to represent the relevant data. Let

$$\mathbf{U} = (U(\boldsymbol{\xi}, s))_{(\boldsymbol{\xi}, s) \in \mathcal{G}_J}, \quad \hat{\mathbf{U}} = (\hat{U}_{\mathbf{k}}(s))_{(\mathbf{k}, s) \in \mathcal{C}_J}.$$

Then, the derivations (11), (12) and (14) show a way to compute the coefficients  $\hat{\mathbf{U}}$  from the grid function  $\mathbf{U}$ . Such process is called the DST on standard STSG, and it is

specifically described by Algorithm 1. Here we use the following linear operators to represent the DST on standard STSG

$$\begin{aligned} \mathbf{F} &: \mathbb{R}^{\mathcal{G}_J} \rightarrow \mathbb{R}^{\mathcal{C}_J} \\ &: \mathbf{U} \mapsto \hat{\mathbf{U}} = \mathbf{F}\mathbf{U}. \end{aligned} \tag{16}$$

Algorithm 1 is obviously reversible, and so there exists an operator  $\mathbf{F}^{-1}$  such that

$$\mathbf{U} = \mathbf{F}^{-1}\hat{\mathbf{U}}.$$

---

**Algorithm 1** Discrete sine transform (16) on standard STSG.

---

- 1: Compute the hierarchical grid function  $(\check{U}_s(\xi))_{(\xi,s) \in \mathcal{G}_J}$  via the hierarchical transform (15);
- 2: Compute the hierarchical sine coefficients  $\{\hat{U}_{\mathbf{k},s}\}_{\mathbf{k} \in \mathcal{K}_{J-j}}$  from  $(\check{U}_s(\xi))_{\xi \in \mathcal{X}_{J-j}}$  via spatial DST for  $s \in \check{T}_j, j = 0, 1, \dots, J - 1$ .
- 3: Compute the sine coefficients via the inverse of hierarchical transform (13) as

$$\hat{U}_{\mathbf{k}(s)} = \begin{cases} \hat{U}_{\mathbf{k},s}, & \text{for } s \in \mathcal{T}_0, \\ \hat{U}_{\mathbf{k},s} + \frac{1}{2} (\hat{U}_{\mathbf{k}(s-2^{-j})} + \hat{U}_{\mathbf{k}(s+2^{-j})}), & \text{for } s \in \check{T}_j, j > 0. \end{cases}$$


---

Thanks to Lemma 2.5 in [9], the DOF of STSG is of order  $\mathcal{O}(2^J J)$  and  $\mathcal{O}(2^{Jd})$  for  $d = 1$  and  $d > 1$ , respectively. Moreover, the computational cost of Algorithm 1 can be estimated by the following theorem.

**Theorem 1** *The computational cost of Algorithm 1 is of order  $\mathcal{O}(2^J J^2)$  for  $d = 1$  and  $\mathcal{O}(2^{Jd} J)$  for  $d > 1$ .*

**Proof** For Algorithm 1, the computational costs of step 1 and step 3 have the same order as the number of grid points, and so we just need to care about step 2. The number of elements in  $\check{T}_j$  is 2 and  $2^{j-1}$  for  $j = 0$  and  $j = 1, 2, \dots, J - 1$ , respectively. The computational cost of spatial DST in grid  $\mathcal{X}_{J-j}$  is less than  $C2^{(J-j)d}(J - j)$ , where  $C$  is a constant independent of  $J$  and  $j$ .

Then, for one-dimensional case, the computational cost is less than

$$C2^{J+1}J + C \sum_{j=1}^{J-1} 2^{j-1}2^{J-j}(J - j) = C2^{J+1}J + C2^{J-1} \sum_{j=1}^{J-1} (J - j),$$

and for multi-dimensional case, the computational cost is less than

$$C2^{Jd+1}J + C \sum_{j=1}^{J-1} 2^{j-1}2^{(J-j)d}(J - j) \leq C2^{Jd+1}J + C2^{Jd-1}J \sum_{j=1}^{J-1} 2^{j(1-d)}.$$

Note that  $\sum_{j=1}^{J-1} (J - j) = \mathcal{O}(J^2)$  and  $\sum_{j=1}^{J-1} 2^{j(1-d)} = \mathcal{O}(1)$ , and we can obtain the result of original proposition.  $\square$

Define the Hilbert space

$$\mathcal{H}_0^\beta = \{v : \|v\|_\beta < \infty\}, \quad \text{where } \|v\|_\beta = \left( \sum_{\mathbf{k} \in \mathbb{N}_+^d} |\mathbf{k}|^{2\beta} |\hat{v}_{\mathbf{k}}|^2 \right)^{\frac{1}{2}}. \quad (17)$$

Especially, it is obvious that  $\mathcal{H}_0^0 = \mathcal{L}_0^2$  since

$$\|v\|_0 = \sum_{\mathbf{k} \in \mathbb{N}_+^d} |\hat{v}_{\mathbf{k}}|^2 = \int_{\Omega} |v(\mathbf{x})|^2 \, d\mathbf{x} = \|v\|_{\mathcal{L}^2}.$$

Then, the function approximation on STSG is defined as

$$\begin{aligned} \tilde{\mathcal{I}}_J : \mathcal{L}^2([0, 1], \mathcal{H}_0^0) &\rightarrow \mathcal{U}_J \\ : u(\mathbf{x}, t) &\mapsto U(\mathbf{x}, t) = \tilde{\mathcal{I}}_J u(\mathbf{x}, t) = \sum_{j=0}^{J-1} \sum_{s \in \tilde{\mathcal{I}}_j} \theta_{s, 2^{-j}}(t) \mathcal{I}_{J-j} \check{u}_s(\mathbf{x}), \end{aligned} \quad (18)$$

where the notation  $\check{u}_s(\mathbf{x})$  is given by (15). For (18), it is easy to prove

$$U(\xi, s) = u(\xi, s), \quad \forall (\xi, s) \in \mathcal{G}_J.$$

**Theorem 2** Suppose that  $u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$  for a fixed real number  $\beta > \frac{d}{2}$ , and the approximation  $U = \tilde{\mathcal{I}}_J u$  is shown in (18). Then,

$$\|u - U\|_0 \lesssim 2^{-J \min\{\beta, 1\}} J \left\| \frac{\partial}{\partial t} u \right\|_\beta + 2^{-J\beta} (\|u(\cdot, 0)\|_\beta + \|u(\cdot, 1)\|_\beta) \quad (19)$$

if  $\frac{\partial}{\partial t} u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$ , and

$$\|u - U\|_0 \lesssim 2^{-J \min\{\beta, 2\}} J \left\| \frac{\partial^2}{\partial t^2} u \right\|_\beta + 2^{-J\beta} (\|u(\cdot, 0)\|_\beta + \|u(\cdot, 1)\|_\beta) \quad (20)$$

if  $\frac{\partial^2}{\partial t^2} u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$ , where

$$\|u\|_\beta = \left( \int_0^1 \|u(\cdot, t)\|_\beta^2 \, dt \right)^{\frac{1}{2}},$$

and the notation  $A \lesssim B$  means that there exists a number  $C$  independent of  $u(\mathbf{x}, t)$  and  $J$ , such that  $A \leq CB$ .

**Proof** Here we only prove (20), because the proof of (19) is similar.

At first, the conditions  $u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$  and  $\frac{\partial^2}{\partial t^2} u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$  can easily yield

$$\max_{0 \leq t \leq 1} \|u(\cdot, t)\|_\beta < \infty, \quad \max_{0 \leq t \leq 1} \left\| \frac{\partial}{\partial t} u(\cdot, t) \right\|_\beta < \infty.$$

Let  $u^*$  be the temporal discretization of  $u$ , i.e.,

$$u^*(\mathbf{x}, t) = \sum_{s \in \mathcal{T}_{J-1}} \theta_{s, 2^{1-j}}(t) u(\mathbf{x}, s).$$

Through the error estimate of linear element approximation [15], we can obtain

$$\|u - u^*\|_0 \lesssim 2^{-2J} \left\| \frac{\partial^2 u}{\partial t^2} \right\|_0. \tag{21}$$

Using the similar proof of Theorem 2.3 in [29], we can obtain the error estimate of sine interpolation

$$\|v - \mathcal{I}_j v\|_0 \lesssim 2^{-j\beta} \|v\|_\beta, \quad \forall j > 0, \beta > \frac{d}{2}.$$

Thus,

$$\begin{aligned} \|u^* - U\|_0 &\leq \sum_{j=0}^{J-1} \left\| \sum_{s \in \check{\mathcal{T}}_j} \theta_{s, 2^{-j}} (\check{u}_s - \mathcal{I}_{J-j} \check{u}_s) \right\|_0 \\ &= \sum_{j=0}^{J-1} \left( \sum_{s \in \check{\mathcal{T}}_j} \int_{s-2^{-j}}^{s+2^{-j}} \|\theta_{s, 2^{-j}}(t) (\check{u}_s - \mathcal{I}_{J-j} \check{u}_s)\|_0^2 dt \right)^{\frac{1}{2}} \\ &= \sum_{j=0}^{J-1} \left( \sum_{s \in \check{\mathcal{T}}_j} \|\check{u}_s - \mathcal{I}_{J-j} \check{u}_s\|_0^2 \int_{s-2^{-j}}^{s+2^{-j}} |\theta_{s, 2^{-j}}(t)|^2 dt \right)^{\frac{1}{2}} \\ &\leq \sum_{j=0}^{J-1} 2^{-\frac{j}{2}} \left( \sum_{s \in \check{\mathcal{T}}_j} \|\check{u}_s - \mathcal{I}_{J-j} \check{u}_s\|_0^2 \right)^{\frac{1}{2}} \\ &\lesssim \sum_{j=0}^{J-1} 2^{-\frac{j}{2} - 2(J-j)\beta} \left( \sum_{s \in \check{\mathcal{T}}_j} \|\check{u}_s\|_\beta^2 \right)^{\frac{1}{2}}. \end{aligned} \tag{22}$$

For  $j \geq 1$ , from the Taylor expansion of integral remainder, we have

$$\begin{aligned} \check{u}_s(\mathbf{x}) &= \frac{1}{2} \int_{s-2^{-j}}^{s+2^{-j}} (|s-t| - 2^{-j}) \frac{\partial^2}{\partial t^2} u(\mathbf{x}, t) dt \\ &= \frac{1}{2} \sum_{\mathbf{k} \in \mathbb{N}_+^d} \varphi_{\mathbf{k}}(\mathbf{x}) \int_{s-2^{-j}}^{s+2^{-j}} (|s-t| - 2^{-j}) \hat{u}_{\mathbf{k}}''(t) dt, \end{aligned}$$

and so

$$\begin{aligned} \|\check{u}_s\|_{\beta} &= \frac{1}{2} \left( \sum_{\mathbf{k} \in \mathbb{N}_+^d} |\mathbf{k}|^{2\beta} \left| \int_{s-2^{-j}}^{s+2^{-j}} (|s-t| - 2^{-j}) \hat{u}_{\mathbf{k}}''(t) dt \right|^2 \right)^{\frac{1}{2}} \\ &\leq \frac{1}{2} \left( \sum_{\mathbf{k} \in \mathbb{N}_+^d} |\mathbf{k}|^{2\beta} \int_{s-2^{-j}}^{s+2^{-j}} | |s-t| - 2^{-j} |^2 dt \int_{s-2^{-j}}^{s+2^{-j}} |\hat{u}_{\mathbf{k}}''(t)|^2 dt \right)^{\frac{1}{2}} \\ &\leq \frac{1}{2} \left( 2^{-3j} \sum_{\mathbf{k} \in \mathbb{N}_+^d} |\mathbf{k}|^{2\beta} \int_{s-2^{-j}}^{s+2^{-j}} |\hat{u}_{\mathbf{k}}''(t)|^2 dt \right)^{\frac{1}{2}}. \end{aligned}$$

Substituting the above inequality into (22), it follows that

$$\begin{aligned} \|u^* - U\|_0 &\lesssim \sum_{j=1}^{J-1} 2^{-2j-(J-j)\beta} \left( \sum_{s \in \check{\mathcal{I}}_j} \sum_{\mathbf{k} \in \mathbb{N}_+^d} |\mathbf{k}|^{2\beta} \int_{s-2^{-j}}^{s+2^{-j}} |\hat{u}_{\mathbf{k}}''(t)|^2 dt \right)^{\frac{1}{2}} \\ &\quad + 2^{-J\beta} (\|u(\cdot, 0)\|_{\beta} + \|u(\cdot, 1)\|_{\beta}) \\ &= 2^{-J\beta} \left\| \frac{\partial^2 u}{\partial t^2} \right\|_{\beta} \sum_{j=1}^{J-1} 2^{j(\beta-2)} + 2^{-J\beta} (\|u(\cdot, 0)\|_{\beta} + \|u(\cdot, 1)\|_{\beta}) \\ &\leq 2^{-J \min\{\beta, 2\}} J \left\| \frac{\partial^2 u}{\partial t^2} \right\|_{\beta} + 2^{-J\beta} (\|u(\cdot, 0)\|_{\beta} + \|u(\cdot, 1)\|_{\beta}). \quad (23) \end{aligned}$$

From (21), (23) and the norm inequality

$$\|u - U\|_0 \leq \|u - u^*\|_0 + \|u^* - U\|_0,$$

the conclusion of the original proposition can be obtained. □

Compare the discretization (18) with the FG discretization

$$U^*(\mathbf{x}, t) = \sum_{m=0}^M \theta_{\frac{m}{M}, \frac{1}{M}}(t) \mathcal{I}_J u\left(\mathbf{x}, \frac{m}{M}\right).$$

The FG approximation requires  $\mathcal{O}(2^{Jd}M)$  DOF, and it is easy to prove the error estimate

$$\|u - U^*\|_0 \lesssim \frac{1}{M^2} \left\| \left\| \frac{\partial^2 u}{\partial t^2} \right\| \right\|_0 + 2^{-J\beta} \max_{0 \leq t \leq 1} \|u(\cdot, t)\|_\beta. \tag{24}$$

If the function  $u(\mathbf{x}, t)$  has low regularity in spatial direction, the constant  $\beta$  shown in Theorem 2 or in inequality (24) cannot be too large. In particular, if  $\beta$  is equal to 2 and  $M$  is chosen to be proportional to  $2^J$ , then the accuracy order in (24) is the same to that in (20). But the FG approximation requires  $\mathcal{O}(2^{J(d+1)})$  DOF, whereas the STSG approximation only requires  $\mathcal{O}(2^J J)$  DOF for  $d = 1$  and  $\mathcal{O}(2^{Jd})$  DOF for  $d > 1$ . When  $\beta$  is not equal to 2, the approximation effects for such two kinds of grids need to be discussed in detail. In general, the advantages of STSG method in saving DOF are very significant if  $\beta$  is not too far from 2. However, it should be noted that the additional condition  $\frac{\partial}{\partial t} u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$  or  $\frac{\partial^2}{\partial t^2} u \in \mathcal{L}^2([0, 1], \mathcal{H}_0^\beta)$  shown in Theorem 2 is necessary. Such condition implies that the spatial regularity is not very related to the temporal derivative, and it is an important factor for STSG method to reflect its advantages.

### 4 Approximating Caputo derivative on modified space-time sparse grid

Unfortunately, for TFDE (1), the condition  $\frac{\partial}{\partial t} u \in \mathcal{L}^2([0, T], \mathcal{H}_0^\beta)$  or  $\frac{\partial^2}{\partial t^2} u \in \mathcal{L}^2([0, T], \mathcal{H}_0^\beta)$  in Theorem 2 is usually not satisfied, because the solution  $u(\mathbf{x}, t)$  usually has the singularity at the initial time [31]. From another point of view, the high-frequency coefficients of the exact solution may change rapidly at the initial moment. In the construction of standard STSG (10), the large time stepsizes are employed to approximate the high-frequency coefficients, and the computation of large time stepsizes is not suitable to simulate the rapid change. In fact, the advantage of the standard STSG is reflected in simulating the spatial low regularity, but it is worse than FG in dealing with the singularity of initial time. Therefore, for some cases of TFDE, the standard STSG method may bring bad results.

For this problem, our approach is to compute the TFDE through the FG method in the first time step, and then the standard STSG is used to compute the equation in the remaining time intervals. And such approach can lead to the modified STSG, which will be introduced later.

The time interval  $[0, T]$  can be divided as

$$0 \leq T_0 < T_1 < \dots < T_L = T, \tag{25}$$

where  $L$  is a given positive integer. For the approach of modified STSG, we select  $T_0$  as the minimum time stepsize in STSG. Then, for any given maximal level  $J$ , we have

$$\begin{aligned} T_n - T_{n-1} &= \Delta T, \quad n = 1, 2, \dots, L, \\ T_0 &= \frac{\Delta T}{2^{J-1}}. \end{aligned} \tag{26}$$

Then,  $T_0$  and  $\Delta T$  can be solved as

$$T_0 = \frac{T}{2^{J-1}L + 1}, \quad \Delta T = \frac{2^{J-1}T}{2^{J-1}L + 1}.$$

According to the related definitions of standard STSG (10), we can define the index set and the grid points of the modified STSG as

$$\begin{aligned} \text{index set: } \mathcal{C}^{(J,L)} &= \bigcup_{n=0}^L \mathcal{C}_J^n, \\ \text{grid points: } \mathcal{G}^{(J,L)} &= \bigcup_{n=0}^L \mathcal{G}_J^n, \end{aligned} \tag{27}$$

where

$$\begin{aligned} \mathcal{C}_J^0 &= \{(\mathbf{k}, s) : \mathbf{k} \in \mathcal{K}_J, s = 0 \text{ or } T_0\}, \\ \mathcal{G}_J^0 &= \{(\xi, s) : \xi \in \mathcal{X}_J, s = 0 \text{ or } T_0\}, \end{aligned}$$

and

$$\begin{aligned} \mathcal{C}_J^n &= \{(\mathbf{k}, s\Delta T + T_{n-1}) : (\mathbf{k}, s) \in \mathcal{C}_J\}, \\ \mathcal{G}_J^n &= \{(\xi, s\Delta T + T_{n-1}) : (\xi, s) \in \mathcal{G}_J\}, \end{aligned} \quad n = 1, 2, \dots, L.$$

It is obvious that  $\mathcal{G}_J^0$  is the FG with only one time step, and  $\mathcal{G}_J^n$  ( $n = 1, 2, \dots, L$ ) are a group of standard STSG. Figure 4 proposes an example of the modified STSG.

To ensure the feasibility of the modified STSG (27), we need to assume that the exact solution  $u(\mathbf{x}, t)$  is singular only at the initial time, and so the norm  $\|\frac{\partial}{\partial t} u(\cdot, t)\|_\beta$  or  $\|\frac{\partial^2}{\partial t^2} u(\cdot, t)\|_\beta$  is bounded with respect to  $t \in [T_0, T]$ . For the computation of interval  $[0, T_0]$ , we employ the FG method with maximal spatial level. And the FG method

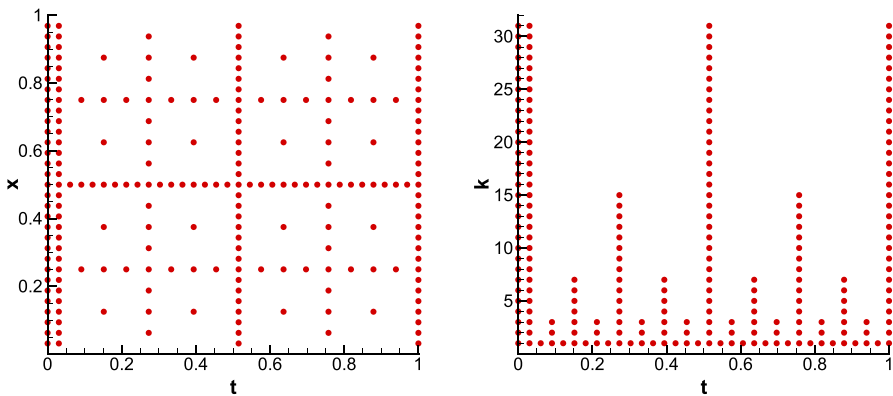


Fig. 4 Grid points (left) and index set (right) of the modified STSG with  $d = 1, J = 5, L = 2$  and  $T = 1$



only needs to be used in the computation of the first time step, because the solution is no longer singular after the first time step.

Let

$$\mathcal{U}_J^{(0)} = \text{span}\{\varphi_{\mathbf{k}}(\mathbf{x})\theta_{s,T_0}(t) : (\mathbf{k}, s) \in \mathcal{C}_J^0\}.$$

Then, we say

$$U \in \mathcal{U}^{(J,L)},$$

if there exists  $U^0 \in \mathcal{U}_J^{(0)}$  and  $U^1, U^2, \dots, U^L \in \mathcal{U}_J$  such that

$$U^{n-1}(\mathbf{x}, 1) = U^n(\mathbf{x}, 0), \quad n = 1, 2, \dots, L$$

and

$$U(\mathbf{x}, t) = \begin{cases} U^0(\mathbf{x}, t), & \text{for } t \in [0, T_0], \\ U^n\left(\mathbf{x}, \frac{t-T_{n-1}}{\Delta T}\right), & \text{for } t \in [T_{n-1}, T_n], \quad n = 1, 2, \dots, L. \end{cases}$$

Any function  $U \in \mathcal{U}^{(J,L)}$  can be written as the form

$$U(\mathbf{x}, t) = \sum_{\mathbf{k} \in \mathcal{K}_J} \hat{U}_{\mathbf{k}}(t)\varphi_{\mathbf{k}}(\mathbf{x}).$$

Let

$$\begin{aligned} \mathbf{U}^n &= (U(\boldsymbol{\xi}, s))_{(\boldsymbol{\xi}, s) \in \mathcal{G}_J^n}, \\ \hat{\mathbf{U}}^n &= (\hat{U}_{\mathbf{k}}(s))_{(\mathbf{k}, s) \in \mathcal{C}_J^n}, \end{aligned} \quad n = 0, 1, \dots, L. \tag{28}$$

Similar to Algorithm 1, we can also obtain the DST on general STSG

$$\hat{\mathbf{U}}^n = \mathbf{F}\mathbf{U}^n, \quad n = 1, 2, \dots, L.$$

Then, we will discuss the solution of the Caputo derivative  $D_t^\alpha \hat{U}_{\mathbf{k}}(t)$  for  $\mathbf{k} \in \mathcal{K}_J$  in this paragraph. Let

$$\tau_j = \frac{\Delta T}{2^j}, \quad j = 0, 1, \dots, J - 1.$$

According to the index set in (27),  $\hat{U}_{\mathbf{k}}(t)$  can be expanded as

$$\hat{U}_{\mathbf{k}}(t) = \begin{cases} \hat{U}_{\mathbf{k}}(0)\theta_{0,T_0}(t) + \hat{U}_{\mathbf{k}}(T_0)\theta_{T_0,T_0}(t), & \text{for } t \in [0, T_0], \\ \sum_{m=0}^{2^{J-j}-1} \hat{U}_{\mathbf{k}}(T_0 + m\tau_{J-j})\theta_{T_0+m\tau_{J-j},\tau_{J-j}}(t), & \text{for } t \in [T_0, T], \end{cases}$$

for  $\mathbf{k} \in \check{\mathcal{K}}_j$ . Note that

$$\frac{d}{dt} \hat{U}_{\mathbf{k}}(t) = \frac{\hat{U}_{\mathbf{k}}(T_0)}{T_0} - \frac{\hat{U}_{\mathbf{k}}(0)}{T_0}$$

for  $t \in (0, T_0)$ , and

$$\frac{d}{dt} \hat{U}_{\mathbf{k}}(t) = \frac{\hat{U}_{\mathbf{k}}(T_0 + m\tau_{J-j})}{\tau_{J-j}} - \frac{\hat{U}_{\mathbf{k}}(T_0 + (m-1)\tau_{J-j})}{\tau_{J-j}}$$

for  $t \in (T_0 + (m-1)\tau_{J-j}, T_0 + m\tau_{J-j})$ ,  $m = 1, 2, \dots, 2^{J-j}L$  and  $\mathbf{k} \in \check{\mathcal{K}}_j$ . Thus, by substituting the expression of  $\frac{d}{dt} \hat{U}_{\mathbf{k}}(t)$  into the Caputo fractional derivative (3), we can obtain

$$\begin{aligned} & D_t^\alpha \hat{U}_{\mathbf{k}}(t) \Big|_{t=T_0+M\tau_{J-j}} \\ &= \frac{1}{\Gamma(1-\alpha)} \int_0^{T_0} (T_0 + M\tau_{J-j} - r)^{-\alpha} \frac{d}{dr} \hat{U}_{\mathbf{k}}(r) dr \\ &+ \sum_{m=1}^M \frac{1}{\Gamma(1-\alpha)} \int_{T_0+(m-1)\tau_{J-j}}^{T_0+m\tau_{J-j}} (T_0 + M\tau_{J-j} - r)^{-\alpha} \frac{d}{dr} \hat{U}_{\mathbf{k}}(r) dr \\ &= \frac{(T_0 + M\tau_{J-j})^{1-\alpha} - (M\tau_{J-j})^{1-\alpha}}{\Gamma(2-\alpha)T_0} (\hat{U}_{\mathbf{k}}(T_0) - \hat{U}_{\mathbf{k}}(0)) \\ &+ \sum_{m=1}^M \frac{b_{M-m}}{\tau_{J-j}^\alpha} (\hat{U}_{\mathbf{k}}(T_0 + m\tau_{J-j}) - \hat{U}_{\mathbf{k}}(T_0 + (m-1)\tau_{J-j})) \\ &= \frac{(T_0 + M\tau_{J-j})^{1-\alpha} - (M\tau_{J-j})^{1-\alpha}}{\Gamma(2-\alpha)T_0} (\hat{U}_{\mathbf{k}}(T_0) - \hat{U}_{\mathbf{k}}(0)) - \frac{b_{M-1}}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0) \\ &+ \sum_{m=1}^{M-1} \frac{b_{M-m} - b_{M-m-1}}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0 + m\tau_{J-j}) + \frac{b_0}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0 + M\tau_{J-j}) \quad (29) \end{aligned}$$

for  $M = 1, 2, \dots, 2^{J-j}L$ , where

$$b_l = \frac{(l+1)^{1-\alpha} - l^{1-\alpha}}{\Gamma(2-\alpha)}, \quad l \in \mathbb{N}_0.$$

The formula (29) is called the L1 difference method [17]. For convenience of computation, we assume that  $2^{J-j}(n-1) \leq M \leq 2^{J-j}n$  for  $n = 1, 2, \dots, L$ , and (29) can be written as

$$D_t^\alpha \hat{U}_{\mathbf{k}}(t) \Big|_{t=s=T_0+M\tau_{J-j}} = L^\alpha \hat{U}_{\mathbf{k}}(s) + \hat{R}_{\mathbf{k},s}, \quad (30)$$

where

$$\begin{aligned}
 L^\alpha \hat{U}_{\mathbf{k}}(s) &= \sum_{m=2^{J-j_n}}^{M-1} \frac{b_{M-m} - b_{M-m-1}}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0 + m\tau_{J-j}) + \frac{b_0}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0 + M\tau_{J-j}) \\
 \hat{R}_{\mathbf{k},s} &= \frac{(T_0 + M\tau_{J-j})^{1-\alpha} - (M\tau_{J-j})^{1-\alpha}}{\Gamma(2-\alpha)T_0} (\hat{U}_{\mathbf{k}}(T_0) - \hat{U}_{\mathbf{k}}(0)) - \frac{b_{M-1}}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0) \\
 &\quad + \sum_{m=1}^{2^{J-j_n}-1} \frac{b_{M-m} - b_{M-m-1}}{\tau_{J-j}^\alpha} \hat{U}_{\mathbf{k}}(T_0 + m\tau_{J-j}).
 \end{aligned}$$

Using the definition of the vector (28), the discretization of the Caputo derivative (30) in the function space  $\mathcal{U}^{(J,L)}$  can be written in the form of vector as

$$\mathbf{D}_t^\alpha \hat{\mathbf{U}}^n = \mathbf{L}^\alpha \hat{\mathbf{U}}^n + \hat{\mathbf{R}}^n,$$

where

$$\mathbf{D}_t^\alpha \hat{\mathbf{U}}^n = (\mathbf{D}_t^\alpha \hat{U}_{\mathbf{k}}(s))_{(\mathbf{k},s) \in \mathcal{C}_J^n}, \quad \mathbf{L}^\alpha \hat{\mathbf{U}}^n = (L^\alpha \hat{U}_{\mathbf{k}}(s))_{(\mathbf{k},s) \in \mathcal{C}_J^n}, \quad \hat{\mathbf{R}}^n = (\hat{R}_{\mathbf{k},s})_{(\mathbf{k},s) \in \mathcal{C}_J^n},$$

for  $n = 1, 2, \dots, L$ . And it is obvious that  $\hat{\mathbf{R}}^n$  is independent of  $\hat{\mathbf{U}}^n$ .

### 5 Fully discrete algorithm

In this section, the fully discrete algorithm of TFDE (1) is obtained. As described in the previous section, the time interval  $[0, T]$  can be divided as (25). We construct a FG with only one time step on  $[0, T_0]$  if  $T_0 > 0$ , and we construct several standard STSGs on  $[T_0, T]$ . Therefore, the numerical solution  $U \in \mathcal{U}^{(J,L)}$  of TFDE (1) can be obtained via the following three steps:

1. Compute  $\hat{U}_{\mathbf{k}}(0)$  for  $\mathbf{k} \in \mathcal{K}_J$ ;
2. Compute  $\hat{U}_{\mathbf{k}}(T_0)$  for  $\mathbf{k} \in \mathcal{K}_J$ , if  $T_0 > 0$ ;
3. Compute  $\hat{U}_{\mathbf{k}}(s)$  for  $(\mathbf{k}, s) \in \mathcal{C}_J^n, n = 1, 2, \dots, L$ .

Step 1 can be obtained by the projection (8) or the interpolation (9) of initial function  $u_0(\mathbf{x})$ . Step 2 can be obtained by L1 difference method with only one time step, and the scheme is

$$\frac{\hat{U}_{\mathbf{k}}(T_0) - \hat{U}_{\mathbf{k}}(0)}{T_0^\alpha \Gamma(2-\alpha)} + a\pi^2 |\mathbf{k}|_2^2 \hat{U}_{\mathbf{k}}(T_0) = \hat{f}_{\mathbf{k}}(T_0),$$

where  $\{\hat{f}_{\mathbf{k}}(t)\}_{\mathbf{k} \in \mathcal{K}_J}$  are the discrete sine coefficients of function  $f(u(\mathbf{x}, t), \mathbf{x}, t)$ .

Thus we mainly discuss step 3. the negative Laplacian  $-\Delta$  corresponds to the linear operator  $\mathbf{\Lambda}$  as

$$\mathbf{\Lambda} \hat{\mathbf{U}}^n = (\pi^2 |\mathbf{k}|_2^2 \hat{U}_{\mathbf{k}}(s))_{(\mathbf{k},s) \in \mathcal{C}_J^n}, \quad n = 1, 2, \dots, L,$$

where  $\|\mathbf{k}\|_2 = (k_1^2 + k_2^2 + \dots + k_d^2)^{\frac{1}{2}}$ , and  $\hat{\mathbf{U}}^n$  is defined in (28). Let

$$\mathbf{f}^n(\mathbf{U}^n) = (f(U(\boldsymbol{\xi}, s), \boldsymbol{\xi}, s))_{(\boldsymbol{\xi}, s) \in \mathcal{C}_J^n},$$

then the numerical scheme of STSG method for TFDE (1) is

$$\mathbf{L}^\alpha \hat{\mathbf{U}}^n + a\Lambda \hat{\mathbf{U}}^n = \mathbf{F}\mathbf{f}^n(\mathbf{F}^{-1}\hat{\mathbf{U}}^n) - \hat{\mathbf{R}}^n, \quad n = 1, 2, \dots, L, \tag{31}$$

where  $\mathbf{L}^\alpha$  and  $\hat{\mathbf{R}}^n$  are given by (30). The algebraic equations of the scheme (31) can be solved by the iteration method

$$(\mathbf{L}^\alpha + a\Lambda + c_{\text{opt}}^{n,p}\mathbf{I}) \hat{\mathbf{U}}^{n,p} = \mathbf{F}\tilde{\mathbf{f}}^{n,p}(\mathbf{F}^{-1}\hat{\mathbf{U}}^{n,p-1}) - \hat{\mathbf{R}}^n, \quad p = 1, 2, \dots, \tag{32}$$

where  $\hat{\mathbf{U}}^{n,p} = (\hat{U}_{\mathbf{k},s}^p)_{(\mathbf{k},s) \in \mathcal{C}_J^n}$  is the approximation of  $\mathbf{U}^n$  obtained by the  $p$ -th iteration,  $c_{\text{opt}}^{n,p}$  is an optimal factor given by

$$c_{\text{opt}}^{n,p} = \max \left\{ 0, \max_{\boldsymbol{\xi} \in \mathcal{X}_J} \left\{ -\frac{\partial}{\partial u} f(u, \boldsymbol{\xi}, T_n) \Big|_{u=U^p(\boldsymbol{\xi}, T_n)} \right\} \right\},$$

and

$$\tilde{\mathbf{f}}^{n,p}(\mathbf{F}^{-1}\hat{\mathbf{U}}^{n,p-1}) = \mathbf{f}^n(\mathbf{F}^{-1}\hat{\mathbf{U}}^{n,p-1}) + c_{\text{opt}}^{n,p}\mathbf{F}^{-1}\hat{\mathbf{U}}^{n,p-1}.$$

The initial vector  $\hat{\mathbf{U}}^{n,0}$  of the iteration (32) is computed by

$$\mathbf{L}^\alpha \hat{\mathbf{U}}^{n,0} + a\Lambda \hat{\mathbf{U}}^{n,0} = -\hat{\mathbf{R}}^n,$$

and the termination condition is given by

$$\left( \sum_{\mathbf{k} \in \mathcal{K}_J} |\hat{U}_{\mathbf{k},T_n}^p - \hat{U}_{\mathbf{k},T_n}^{p-1}|^2 \right)^{\frac{1}{2}} \leq \epsilon \tag{33}$$

with a fixed threshold  $\epsilon$ .

For the numerical scheme (31), the corresponding stability analysis and convergence analysis can be carried out in three cases. These three cases correspond to the three forms of the source term  $f(u, \mathbf{x}, t)$ , and that is the form without  $u$ , the general linear form and the nonlinear form. For the first case, the analysis is relatively simple since the basis function in the equation is separable. For the second case, the stability of the numerical solution can be obtained by proving the positive definiteness of the fully discrete matrix, and then the convergence can be proved by using the results obtained in the first case. For the third case, we mainly focus on several special problems, and the analysis strategy depends on the specific form of each problem. By using such analysis method, we can prove that the scheme (31) is stable and convergent of order  $\mathcal{O}(2^{-J}J)$ . The detailed proving process is relatively complicated, and it will be given elsewhere. In the present paper, we only concentrate on the algorithm design and the numerical tests of the STSG method.

## 6 Numerical experiment

In this section, we present several numerical examples to show the advantage of modified STSG method in solving TFDE (1). All simulations of this section are accomplished with C++ programming language (Microsoft Visual Studio 2010 development environment), and run on the computer with an Intel (R) Core (TM) i5-4590 CPU and 8.00 GB of RAM. We both consider the standard STSG method and the modified STSG method, which are characterized as  $T_0 = 0$  and  $T_0$  satisfies (26) in the division of time interval (25). For comparison, we also consider the results computed by the standard L1 difference [13, 35]/sine pseudospectral method and the corrected L1 difference/sine pseudospectral method on FG, and these two methods are referred to as standard FG method and corrected FG method respectively in the following content. For FG method, we select the index set

$$(1, 2, \dots, K)^d$$

for spatial discretization and the grid points

$$\{0, T/M, \dots, T(M - 1)/M, T\}$$

for temporal discretization, where  $K$  and  $M$  are given positive integers. Moreover, for the iterative termination condition (33) of algebraic equations, we apply the threshold  $\epsilon = 10^{-10}$  in all numerical experiments, and this threshold is also used for the corresponding FG method.

For the error results, we mainly consider the relative error at the final time  $T$ , which is computed by

$$E_T = \frac{\|u_{\text{ref}}(\cdot, T) - u^*(\cdot, T)\|_0}{\|u_{\text{ref}}(\cdot, T)\|_0},$$

where the norm  $\|\cdot\|_0$  is defined in (17),  $u^*$  is test solution, and  $u_{\text{ref}}$  is the reference solution solved by the numerical method with enough accuracy. We also concerned about the convergence order. For an algorithm with two sets of computational parameters, suppose that  $E_1, E_2$  and  $N_1, N_2$  are the relative errors and the total DOFs, respectively. Then, the convergence order is calculated by

$$\text{Order} = \frac{\ln E_1 - \ln E_2}{\ln N_2 - \ln N_1}.$$

**Example 2** Consider the one-dimensional TFDE

$$D_t^\alpha u = 0.1 \frac{\partial^2}{\partial x^2} u + (1 - t^\gamma)(1 - u)(1 - \cos 2\pi x), \quad \gamma > 0, (x, t) \in (0, 1) \times (0, 1]$$

with the homogeneous boundary condition (2) and the linear element initial function

$$u_0(x) = \begin{cases} 2x, & \text{for } 0 \leq x \leq 0.5, \\ 2(1 - x), & \text{for } 0.5 < x \leq 1. \end{cases}$$

We firstly consider the case of  $\gamma = 1$ . Figure 5 shows that the solutions at  $t = 1$  are similar in the case of  $\alpha = 0.5$  and  $\alpha = 0.2$ . From Fig. 6, their frequency spectrum is about fourth-power decaying rate, and this implies that the solutions are not infinitely smooth.

In this example, we test the error of the numerical method through the reference solutions obtained by modified STSG method with  $J = 17, L = 16$ . Figure 7 shows the convergence behavior. For the standard FG method, the convergence rate is very slow for  $M/K = 1$ , whereas the starting accuracy is too low for  $M/K = 16$ . On the whole, the accuracy of modified STSG method is significantly higher than that of standard FG method for similar DOF. Figure 8 shows the relationship between the DOF and the CPU time cost. When the DOF are similar, the CPU time cost of modified STSG method may be more. This is because the algebraic system obtained modified STSG method is more complex than that obtained by standard FG method, and such algebraic system requires more iterative steps to compute.

We subjectively select some “optimized” computational parameters in the numerical method. Tables 1 and 2 give the representative results in the case of  $\alpha = 0.2$  and  $\alpha = 0.8$ , respectively. From the whole convergence process, the convergence order of STSG method is close to first order, and such order is similar to that of L1 difference approximation. In contrast, the convergence order of standard FG method is significantly lower than first order. In this example, we give the comparison of the STSG method and the corrected FG method. For the case of  $\alpha = 0.2$ , the STSG method requires fewer DOF to achieve similar accuracy, but the corrected FG method requires less CPU time cost, so the advantages of STSG method are not significant. But for the case of  $\alpha = 0.8$ , the advantage of corrected FG method is greatly weakened, this is because the corrected L1 difference scheme is of order  $\mathcal{O}(\tau^{2-\alpha})$ , which decrease with the increase of  $\alpha$ . By comparison, the convergence behavior of STSG method is basically not affected by the value of  $\alpha$ . Moreover, the convergence behavior of the standard STSG method and the modified STSG method is very similar for both  $\alpha = 0.2$  and  $\alpha = 0.8$ .

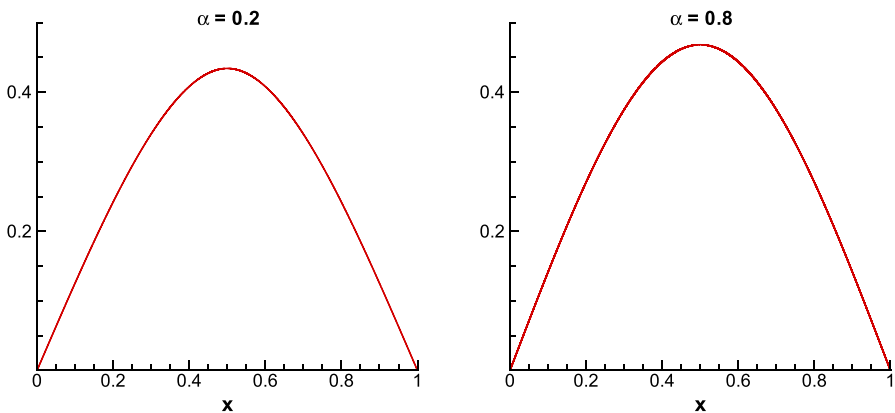


Fig. 5 The solution at  $t = 1$  in Example 2 ( $\gamma = 1$ )

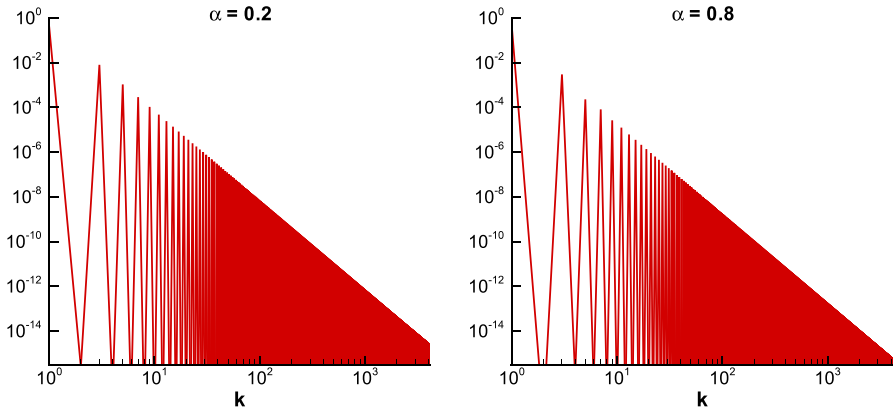


Fig. 6 The frequency spectrum at  $t = 1$  in Example 2 ( $\gamma = 1$ )

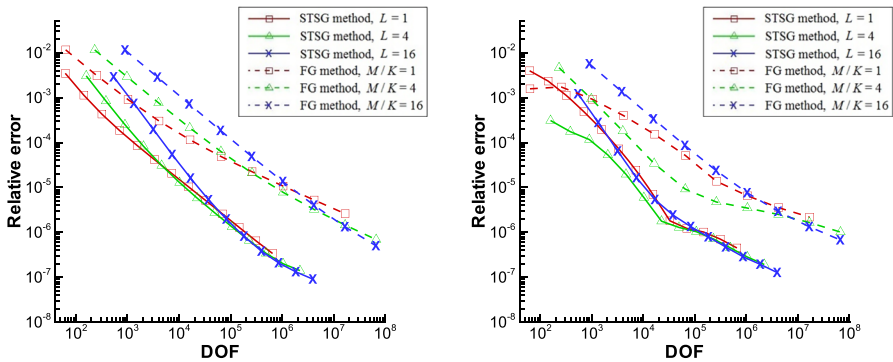


Fig. 7 Convergence behavior of modified STSG method and standard FG method in Example 2 ( $\gamma = 1$ ) with  $\alpha = 0.2$  (left) and  $\alpha = 0.8$  (right)

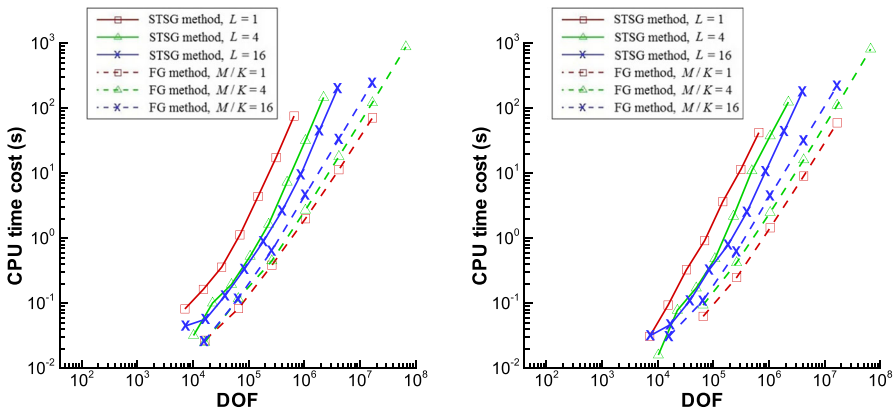


Fig. 8 The CPU time cost of modified STSG method and standard FG method in Example 2 ( $\gamma = 1$ ) with  $\alpha = 0.2$  (left) and  $\alpha = 0.8$  (right)

**Table 1** Representative numerical results in Example 2 with  $\alpha = 0.2, \gamma = 1$

	CPU (s)	DOF	$E_T$	Order
Standard STSG method				
$J = 5, L = 1$	–	38	1.614E-4	–
$J = 7, L = 2$	0.01	1023	1.235E-4	0.78
$J = 9, L = 4$	0.07	9727	1.313E-5	1.00
$J = 11, L = 8$	1.20	92159	1.585E-6	0.94
$J = 14, L = 16$	56.39	1851391	1.529E-7	0.78
Modified STSG method				
$J = 5, L = 1$	–	142	1.136E-3	–
$J = 7, L = 2$	0.01	1150	1.189E-4	1.08
$J = 9, L = 4$	0.03	10238	1.287E-5	1.02
$J = 11, L = 8$	0.40	94206	1.455E-6	0.98
$J = 14, L = 16$	45.45	1867774	1.317E-7	0.80
Standard FG method				
$K = 32, M = 32$	–	1023	9.084E-4	–
$K = 128, M = 128$	0.03	16383	1.148E-4	0.75
$K = 512, M = 1024$	0.96	523775	1.277E-5	0.63
$K = 1024, M = 8192$	63.67	8381439	1.963E-6	0.68
Corrected FG method				
$K = 16, M = 8$	–	135	1.334E-3	–
$K = 32, M = 16$	–	527	2.554E-4	1.22
$K = 64, M = 32$	–	2079	4.382E-5	1.28
$K = 512, M = 256$	0.19	131327	1.334E-6	0.84
$K = 2048, M = 2048$	10.85	4194303	1.461E-7	0.64

Then, we consider the smaller value of  $\gamma$ . Many numerical experiments show that the STSG method is better than the Corrected FG method when  $\gamma < 0.5$ . In particular, the result of the case  $\alpha = 0.2, \gamma = 0.1$  can be seen in Table 3. Indeed, when the source term is singularity at  $t = 0$ , the regularity of the solution in temporal direction becomes very low. In this way, the high-order accuracy algorithm in temporal direction will not give full play to its effect, and so the corrected FG method has no obvious advantages. In contrast, the STSG method can show nice convergence behavior in the case of low regular solution. Besides, in this case, the accuracy of the modified STSG method is slightly better than the standard STSG method. This is because the source term changes quickly at the initial moment, which reduces the accuracy of the standard STSG method.



**Table 2** Representative numerical results in Example 2 with  $\alpha = 0.8, \gamma = 1$

	CPU (s)	DOF	$E_T$	Order
Standard STSG method				
$J = 5, L = 4$	–	351	1.631E-4	–
$J = 6, L = 8$	–	1599	2.574E-5	1.34
$J = 8, L = 8$	0.05	8447	4.913E-6	1.00
$J = 12, L = 32$	17.02	790527	2.858E-7	0.63
Modified STSG method				
$J = 5, L = 4$	–	382	1.751E-4	–
$J = 6, L = 8$	–	1662	2.501E-5	1.32
$J = 8, L = 8$	0.03	8702	3.967E-6	1.11
$J = 12, L = 32$	7.65	794622	2.997E-7	0.57
Standard FG method				
$K = 16, M = 32$	–	495	2.634E-4	–
$K = 64, M = 256$	0.03	16191	3.424E-5	0.58
$K = 512, M = 1024$	1.45	1048575	2.121E-6	0.67
$K = 2048, M = 16384$	873.04	33522687	9.240E-7	0.24
Corrected FG method				
$K = 16, M = 64$	–	975	2.916E-4	–
$K = 64, M = 512$	0.05	32319	1.199E-5	0.91
$K = 256, M = 4096$	4.44	1044735	2.452E-6	0.46
$K = 1024, M = 32768$	875.08	33522687	3.577E-7	0.55

**Example 3** Consider the one-dimensional homogeneous problem

$$D_t^\alpha u = 0.1 \frac{\partial^2}{\partial x^2} u + (1 - t)(1 - u)(1 - \cos 2\pi x), \quad (x, t) \in (0, 1) \times (0, 1],$$

with the homogeneous boundary condition (2) and the Dirac  $\delta$  initial function

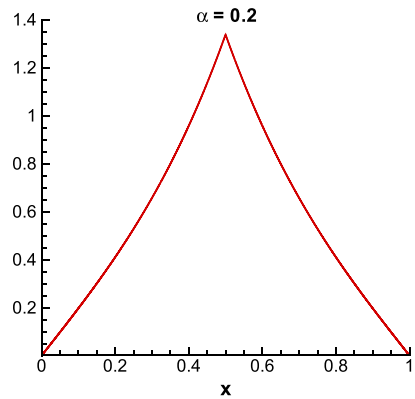
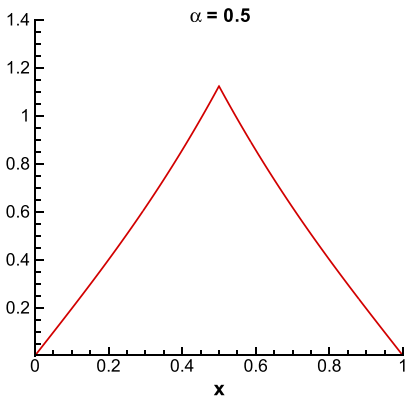
$$u_0(x) = \delta(x - 0.5).$$

Note that the Dirac  $\delta$  function cannot be interpolated, so we choose its projection  $\mathcal{P}_J u_0$  (8) as the initial vector of numerical solution. Figures 9 and 10 show the information of solution at  $t = 1$ . The regularity of the solution in Example 3 is obviously weaker than that in Example 2, and the frequency spectrum is about second-power decaying. In contrast, the regularity of the solution in the case of  $\alpha = 0.2$  is slightly lower than that in the case of  $\alpha = 0.5$ .

In this example, we use the same reference solution as that in Example 2. At first, we observe the relative error results of the standard STSG method, which are shown in Fig. 11. In this example, the standard method even fails to converge. This is because the solution changes very quickly at the initial moment, and the convergence of the

**Table 3** Representative numerical results in Example 2 with  $\alpha = 0.2, \gamma = 0.1$

	CPU (s)	DOF	$E_T$	Order
<b>Standard STSG method</b>				
$J = 6, L = 2$	–	447	1.123E-3	–
$J = 8, L = 4$	0.02	4351	1.071E-4	1.03
$J = 10, L = 8$	0.41	41983	1.079E-5	1.01
$J = 12, L = 16$	7.80	397311	1.115E-6	1.01
$J = 13, L = 64$	99.58	3416063	1.121E-7	1.07
<b>Modified STSG method</b>				
$J = 6, L = 2$	–	510	6.723E-4	–
$J = 8, L = 4$	0.02	4606	6.510E-5	1.06
$J = 10, L = 8$	0.56	43006	6.743E-6	1.01
$J = 12, L = 16$	7.92	401406	7.209E-7	1.00
$J = 13, L = 64$	101.22	3424254	7.358E-8	1.06
<b>FG method</b>				
$K = 64, M = 64$	–	4095	6.761E-4	–
$K = 128, M = 512$	0.14	65151	1.018E-4	0.68
$K = 512, M = 4096$	10.62	2093567	9.007E-6	0.70
$K = 2048, M = 32768$	1696.18	67078143	8.606E-7	0.68
<b>Corrected FG Method</b>				
$K = 64, M = 32$	–	527	9.084E-4	–
$K = 256, M = 256$	0.14	65535	5.149E-5	0.51
$K = 1024, M = 2048$	6.78	2096127	4.681E-6	0.69
$K = 2048, M = 16384$	452.27	33540095	5.700E-7	0.76



**Fig. 9** The solution at  $t = 1$  in Example 3

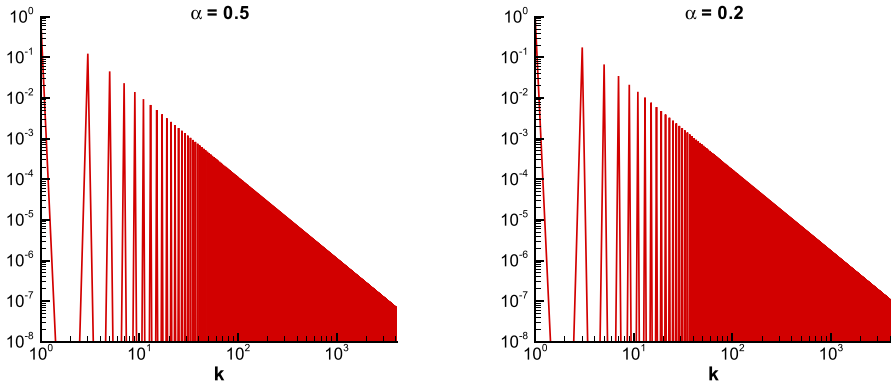


Fig. 10 The frequency spectrum at  $t = 1$  in Example 3

standard STSG method will become extremely difficult. In contrast, from Fig. 12 and Table 4, the modified STSG method can still achieve convergence of about first order, and its convergence behavior is still significantly better than the standard FG method. Therefore, for some special cases, modified technology of STSG can play a decisive role. On the other hand, for general cases, the modified STSG method is not weaker than the standard STSG method, so we can completely replace the standard STSG method with the modified STSG method.

**Example 4** Extending the problem of Example 3 to two-dimensional case, and we have

$$D_t^{0.5} u = 0.1 \Delta u + (1 - t)(1 - u)(1 - \cos 2\pi x)(1 - \cos 2\pi y),$$

$$u_0(x, y) = \delta(x - 0.5)\delta(y - 0.5),$$

for  $(x, y, t) \in (0, 1)^2 \times (0, 1]$ .

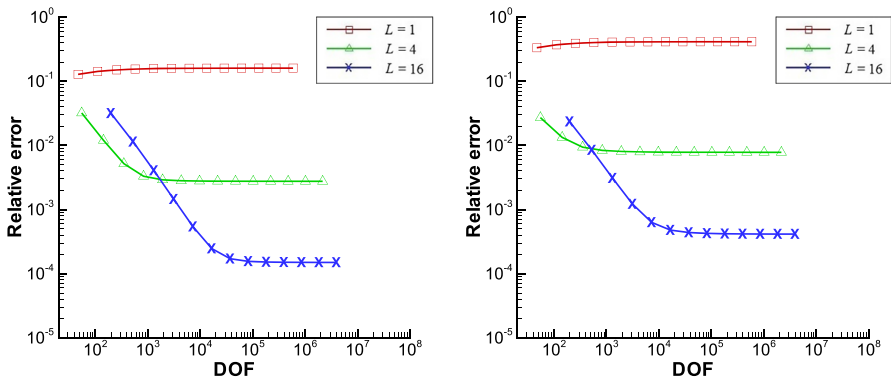
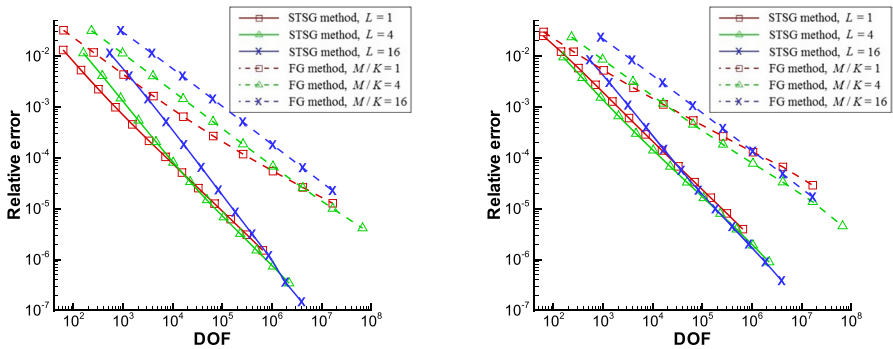


Fig. 11 Relative error of standard STSG method in Example 3 with  $\alpha = 0.2$  (left) and  $\alpha = 0.5$  (right)



**Fig. 12** Convergence behavior of modified STSG method and standard FG method in Example 3 with  $\alpha = 0.2$  (left) and  $\alpha = 0.5$  (right)

Multi-dimensional problems with Dirac  $\delta$  initial function will encounter greater difficulties in computation. On the one hand, multi-dimensional problems require much more DOF in spatial direction; on the other hand, the singularity of solution caused by multi-dimensional Dirac  $\delta$  function is stronger than that caused by one-dimensional Dirac  $\delta$  function. From Fig. 13, although after a time of length 1, the solution still has a marked singularity at the point  $(x, y) = (0.5, 0.5)$ .

In this example, we consider the reference solution obtained by modified STSG method with  $J = 13, L = 2$ . From Fig. 14, the convergence behavior of modified STSG method is still significantly better than that of standard FG method. From Table 5, the standard FG method needs more than 500 million DOF to reach the relative error of about  $9 \cdot 10^{-4}$ , and the required storage is close to the upper limit of our computer memory. For achieving such accuracy, The modified STSG method requires only one

**Table 4** Representative numerical results in Example 3 with  $\alpha = 0.2$

	CPU (s)	DOF	$E_T$	Order
Modified STSG method				
$J = 4, L = 1$	–	62	1.325E-2	–
$J = 7, L = 1$	–	702	9.910E-4	1.07
$J = 9, L = 2$	0.02	5630	1.197E-4	1.02
$J = 12, L = 4$	0.50	106494	6.817E-6	0.97
$J = 13, L = 4$	1.69	229374	3.222E-6	0.98
$J = 15, L = 16$	213.01	3997694	1.465E-7	1.08
Standard FG method				
$K = 16, M = 16$	–	255	1.184E-2	–
$K = 64, M = 64$	–	4095	1.641E-3	0.71
$K = 512, M = 512$	0.369	262143	1.185E-4	0.63
$K = 2048, M = 8192$	123.39	16771071	1.005E-5	0.59
$K = 4096, M = 16384$	916.02	67096575	4.148E-6	0.64

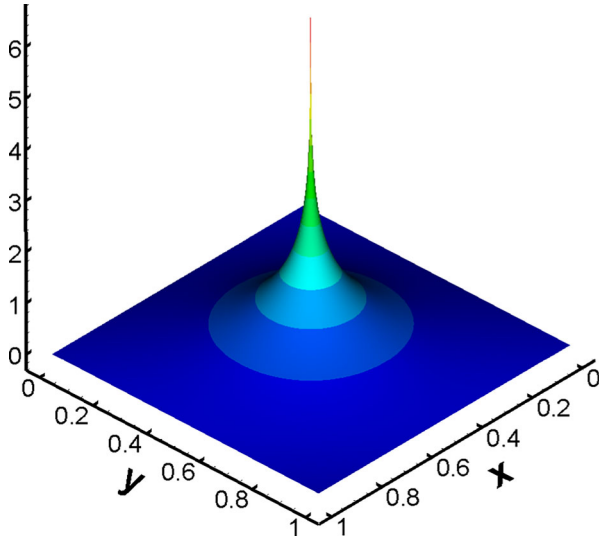


Fig. 13 The solution at  $t = 1$  in Example 4

percent of the DOF. Moreover, the convergence order obtained by computing two-dimensional problems is about half of that in computing one-dimensional problems, which is true for both the modified STSG method and the standard FG method.

**Example 5** Consider the time-fractional Allen-Cahn equation

$$D_t^\alpha u = 0.01 \frac{\partial^2}{\partial x^2} u + u - u^3, \quad (x, t) \in (0, 1) \times (0, 100],$$

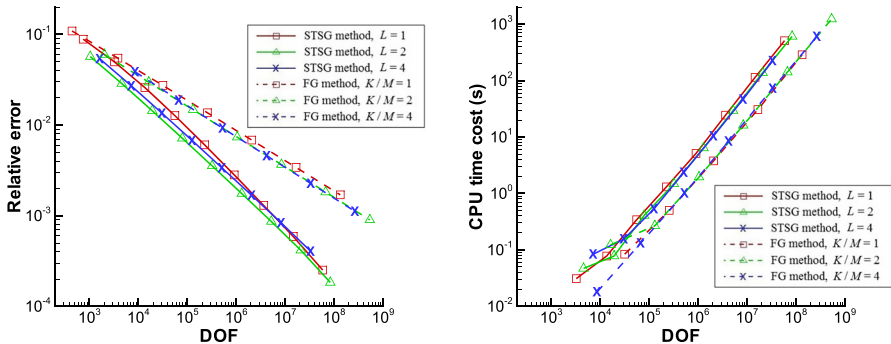


Fig. 14 The convergence behavior (left) and the CPU time cost (right) of modified STSG method and standard FG method in Example 4

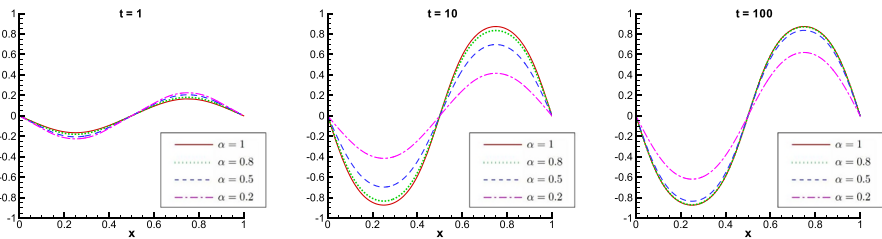
**Table 5** Representative numerical results in Example 4

	CPU (s)	DOF	$E_T$	Order
Modified STSG method				
$J = 6, L = 2$	0.08	19266	1.438E-2	–
$J = 8, L = 2$	1.48	321794	3.548E-3	0.50
$J = 10, L = 2$	29.40	5215234	8.636E-4	0.51
$J = 12, L = 2$	607.01	83759106	1.834E-4	0.56
Standard FG method				
$K = 64, M = 32$	0.27	130977	1.478E-2	–
$K = 256, M = 128$	16.13	8388225	3.667E-3	0.34
$K = 1024, M = 512$	1232.734	536869377	9.005E-4	0.34

with the homogeneous boundary condition (2) and the bidirectional pulse initial function

$$u_0(x) = \begin{cases} 0.1, & \text{for } \frac{1}{8} \leq x \leq \frac{3}{8}, \\ -0.1, & \text{for } \frac{5}{8} \leq x \leq \frac{7}{8}, \\ 0, & \text{for } 0 < x < \frac{1}{8} \text{ or } \frac{3}{8} < x < \frac{5}{8} \text{ or } \frac{7}{8} < x < 1. \end{cases}$$

The Allen-Cahn equation is an important kind of reaction-diffusion equation, and it is also one of basic equations in phase-field theory. The numerical method of time-fractional Allen-Cahn equation has also been studied in recent years [5, 33]. Figure 15 shows the snapshots of solutions at different times. In the beginning, the change is faster for smaller  $\alpha$ . However, for long time observation, the change with smaller  $\alpha$  is obviously slower. The solution of the integer-order case has basically arrived at the steady state when  $t = 10$ , but the solution of the case  $\alpha = 0.2$  is still far from the steady state when  $t = 100$ . Figure 16 shows the frequency spectrum at  $t = 100$ . The frequency spectrum is exponentially decaying in the integer-order case. But in the fractional case of  $\alpha = 0.8$  or  $\alpha = 0.5$ , although their solution functions look similar to



**Fig. 15** The solution at different time in Example 5

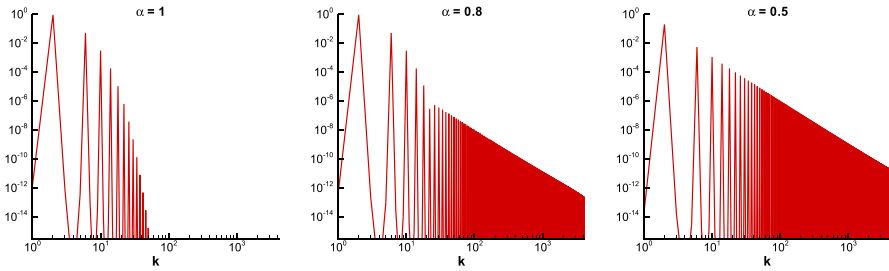


Fig. 16 The frequency spectrum at  $t = 100$  in Example 5

that in the integer-order case, their frequency spectrums are not exponentially decaying. Therefore, the solutions at  $t = 100$  in the fractional case are not infinitely smooth.

Then, we consider the convergence behavior of numerical method for  $\alpha = 0.5$ , and the reference solution is obtained by modified STSG method with  $J = 19, L = 2$ . From Fig. 17, the modified STSG method can still greatly reduce DOF in high-accuracy approximation. However, the CPU time costs of modified STSG method in this example are obviously more than those in Examples 2 and 3. The nonlinear property brings more difficult to solve the algebraic system, which is more obvious in the modified STSG method. Nevertheless, from Table 6, the modified STSG method still has obvious advantages in time efficiency and storage efficiency. In particular, when the required relative error is not more than  $3 \cdot 10^{-7}$ , the DOF and the CPU time cost saved by the modified STSG method are about 500 times and 100 times, respectively. Therefore, the modified STSG method is still suitable for nonlinear problems.

**Example 6** Consider the TFDE

$$D_t^{0.5} u = 0.1 \frac{\partial^2}{\partial x^2} u + f(u, x, t), \quad (x, t) \in (0, 1) \times (0, 1],$$

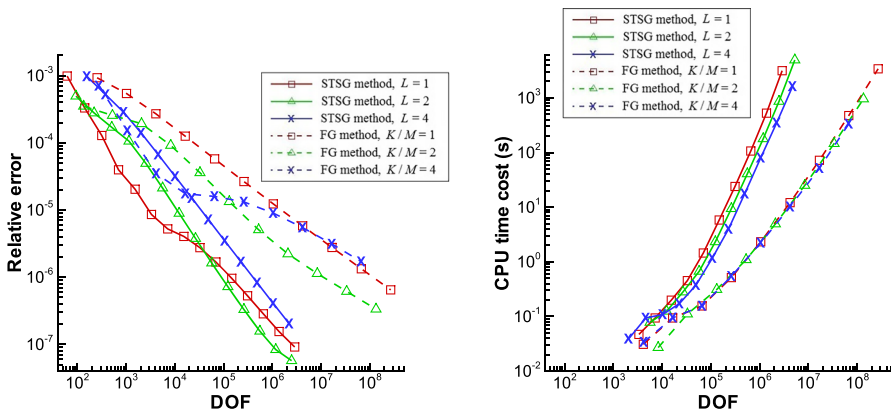


Fig. 17 The convergence behavior (left) and the CPU time cost (right) of modified STSG method and standard FG method in Example 5 with  $\alpha = 0.5$

**Table 6** Representative numerical results in Example 5 with  $\alpha = 0.5$

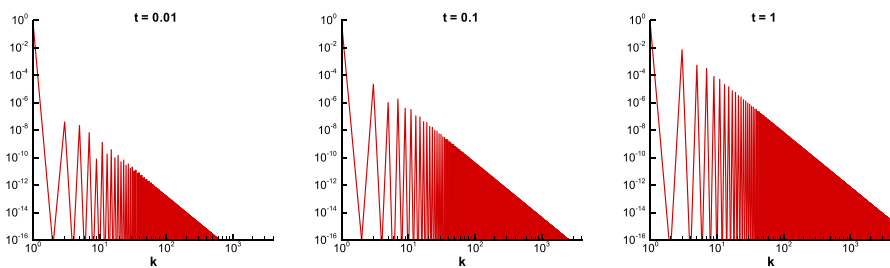
	CPU (s)	DOF	$E_T$	Order
Modified STSG method				
$J = 6, L = 1$	–	318	1.284E-4	–
$J = 9, L = 1$	0.05	3326	8.575E-6	1.15
$J = 13, L = 2$	2.33	122878	7.220E-7	0.69
$J = 14, L = 2$	9.42	262142	3.302E-7	1.03
$J = 16, L = 2$	178.74	1179646	8.361E-8	0.91
Standard FG method				
$K = 64, M = 16$	–	1071	1.516E-4	–
$K = 256, M = 64$	0.10	16575	1.747E-5	0.79
$K = 4096, M = 2048$	25.04	8390655	1.132E-6	0.44
$K = 16384, M = 8192$	957.77	134225919	3.348E-7	0.44

with the homogeneous boundary condition (2) and the sine initial function  $u_0(x) = \sin \pi x$ , where the source term is

$$f(u, x, t) = \begin{cases} (1 - u)(2xt)^2, & \text{for } 0 \leq x \leq 0.5, \\ (1 - u)(2(1 - x)t)^2, & \text{for } 0.5 < x \leq 1. \end{cases}$$

Different from the previous examples, we consider smooth initial function and nonsmooth source term in this example. Figure 18 shows that the frequency spectrums are about fourth-power decaying rate at different time. Note that the sine coefficient of the initial function is equal to 0 for  $k > 1$ , and it implies that the frequency spectrums of index  $k > 1$  have a fast growth at the initial moment.

In this example, the reference solution is obtained by the modified STSG method with  $J = 17, L = 32$ . Figure 19 and Table 7 show the results of numerical comparison. When the required relative error is not more than  $1 \cdot 10^{-6}$ , the modified STSG method saves more than 100 times of total DOF.



**Fig. 18** The frequency spectrum at different time in Example 6



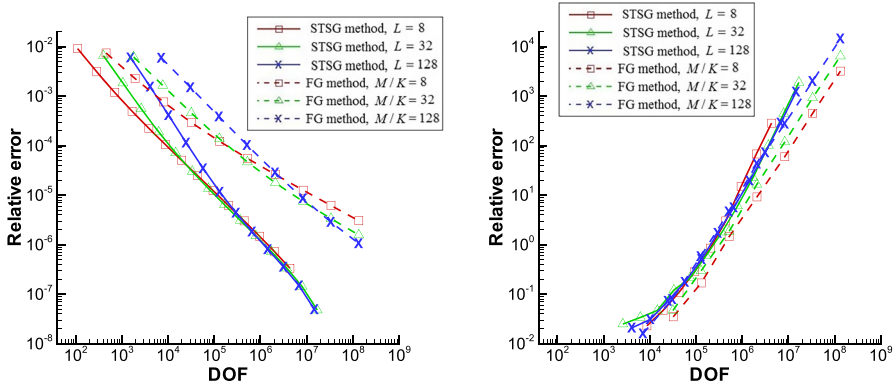


Fig. 19 The convergence behavior (left) and the CPU time cost (right) of modified STSG method and standard FG method in Example 6

### 7 Conclusion

In this paper, the modified STSG method is constructed for solving TFDE (1). In numerical experiments, we give the comparison for different cases (one-dimensional problem or multi-dimensional problem, linear problem or nonlinear problem, nonsmooth initial function or nonsmooth source term). When the solution is not smooth enough in spatial direction, the modified STSG method has obvious advantages over the standard FG method. Furthermore, if the value of  $\alpha$  is relatively large or the source term has strong singularity at the initial point, the modified STSG method is also obviously better than the corrected FG method. In addition, compared with the standard STSG method, the modified STSG method can well deal with the fast change at the initial moment. To sum up, we can determine that the modified STSG method has special advantages in solving TFDE.

Table 7 Representative numerical results in Example 6

	CPU (s)	DOF	$E_T$
Modified STSG method			
$J = 5, L = 8$	–	702	1.182E-3
$J = 8, L = 8$	0.02	8702	1.057E-4
$J = 9, L = 32$	0.21	74750	1.366E-5
$J = 10, L = 256$	21.08	1312766	1.061E-6
$J = 13, L = 128$	289.70	6832126	1.488E-7
Standard FG method			
$K = 16, M = 128$	–	1935	2.265E-3
$K = 128, M = 1024$	0.17	130175	1.223E-4
$K = 512, M = 8192$	30.78	4186623	1.366E-5
$K = 1024, M = 131072$	14478.26	134087679	1.061E-6

Moreover, the STSG method has great promotion value. On the one hand, the idea of STSG method can be extended to other evolution equation with time memory property, such as delay diffusion equation [7], Volterra diffusion equation [23], and so on. On the other hand, the STSG technique is also feasible for most spatial discretization method, such as finite difference method, finite element method and other Fourier-like spectral method. Besides, in order to improve further the efficiency of the algorithm, the STSG method can also be combined with other fast algorithm or high-order accuracy method for solving TFDE.

**Funding** This work is supported by NSF of China (Nos. 12071403, 11971412), Scientific Research Foundation of Hunan Provincial Education Department of China (Nos. 21A0108, 20B566).

**Data Availability** All data generated or analyzed during this study are included in this published article.

## Declarations

**Ethics approval and consent to participate** Not applicable

**Consent for publication** Not applicable

**Human and animal ethics** Not applicable

**Competing interests** The authors declare no competing interests.

## References

1. Ali, U., Naeem, M., Abdullah, F.A., Wang, M.-K., Salama, F.M.: Analysis and implementation of numerical scheme for the variable-order fractional modified sub-diffusion equation. *Fractals* **30**(10), 2240253 (2022)
2. Benzaquen, M., Bouchaud, J.P.: A fractional reaction-diffusion description of supply and demand. *Eur. Phys. J. B* **91**, 23 (2018)
3. Bungartz, H., Griebel, M.: Sparse Grids. *Acta Numer.* **13**, 147–269 (2004)
4. Chen, S., Shen, J., Wang, L.-L.: Generalized Jacobi functions and their applications to fractional differential equations. *Math. Comput.* **85**(300), 1603–1638 (2016)
5. Du, Q., Yang, J., Zhou, Z.: High-order time stepping Fourier spectral method for multi-dimensional space-fractional reaction-diffusion equations. *J. Sci. Comput.* **85**(2), 42 (2019)
6. Ginoa, M., Cerbelli, S., Roman, H.E.: Fractional diffusion equation and relaxation in complex viscoelastic materials. *Phys. A* **191**(1–4), 449–453 (1992)
7. Gourley, S.A., Kuang, Y.: A delay reaction-diffusion model of the spread of bacteriophage infection. *SIAM J. Math. Anal.* **65**(2), 550–566 (2005)
8. Griebel, M., Oeltz, D.: A sparse grid space-time discretization scheme for parabolic problems. *Computing* **81**, 1–34 (2007)
9. Griebel, M., Oeltz, D., Vassilevski, P.: Space-time approximation with sparse grids. *SIAM J. Sci. Comput.* **28**(2), 701–727 (2006)
10. Huang, Y.-Q., Li, X.-Y., Xiao, A.-G.: Fourier pseudospectral method on generalized sparse grids for the space-fractional Schrödinger equation. *Comput. Math. Appl.* **75**(12), 4241–4255 (2018)
11. Jiang, S.-D., Zhang, J.-W., Zhang, Q., Zhang, Z.-M.: Fast Evaluation of the Caputo Fractional Derivative and its Applications to Fractional Diffusion Equations. *Commun. Comput. Phys.* **21**(3), 650–678 (2017)

12. Jin, B.-T., Li, B.-Y., Zhou, Z.: Correction of high-order BDF convolution quadrature for fractional evolution equations. *SIAM J. Sci. Comput.* **39**(6), A3129–A3152 (2017)
13. Jin, B.-T., Lazarovb, R., Zhou, Z.: Numerical methods for time-fractional evolution equations with nonsmooth data: A concise overview. *Comput. Methods Appl. Mech. Engrg.* **346**, 332–358 (2019)
14. Kopteva, N.: Error analysis of the L1 method on graded and uniform meshes for a fractional-derivative problem in two and three dimensions. *Math. Comp.* **88**, 2135–2155 (2019)
15. Larsson, S., Thomée, V.: *Partial Differential Equations with Numerical Methods*. Springer-Verlag, Berlin (2003)
16. Leugering, G.: Time optimal boundary controllability of a simple linear viscoelastic liquid. *Math. Methods Appl. Sci.* **9**, 413–430 (1987)
17. Li, C.-P., Zeng, F.-H.: Finite difference methods for fractional differential equations. *Int. J. Bifurcat. Chaos* **22**(4), 1230014 (2012)
18. Li, X.-J., Xu, C.-J.: A space-time spectral method for the time fractional diffusion equation. *SIAM J. Numer. Anal.* **47**(3), 2108–2131 (2009)
19. Mehendiratta, V., Mehra, M.: A difference scheme for the time-fractional diffusion equation on a metric star graph. *Appl. Numer. Math.* **158**, 152–163 (2020)
20. Mehendiratta, V., Mehra, M., Leugering, G.: Existence and uniqueness of time-fractional diffusion equation on a metric star graph. In: Awasthi, A., John, S.J., Panda, S. (eds.) *Computational Sciences - Modelling, Computing and Soft Computing*, pp. 25–41. Springer, Singapore (2020)
21. Mehendiratta, V., Mehra, M., Leugering, G.: Optimal control problems driven by time-fractional diffusion equations on metric graphs: optimality system and finite difference approximation. *SIAM J. Control. Optim.* **59**(6), 4216–4242 (2021)
22. Mehendiratta, V., Mehra, M., Leugering, G.: Well-posedness, optimal control and discretization for time-fractional parabolic equations with time-dependent coefficients on metric graphs. *Asian J. Control* (2022). <https://doi.org/10.1002/asjc.2958>
23. Redlinger, R.: On Volterra's population equation with diffusion. *SIAM J. Math. Anal.* **16**(1), 135–142 (1985)
24. Salama, F.M., Ali, N.H.M.: Computationally efficient hybrid method for the numerical solution of the 2D time fractional advection-diffusion equation. *Int. J. Math. Eng. Manag.* **5**(3), 432–446 (2020)
25. Salama, F.M., Ali, N.H.M., Hamid, N.N.A.: Fast  $O(N)$  hybrid Laplace transform-finite difference method in solving 2D time fractional diffusion equation. *J. Math. Comput. Sci.* **23**, 110–123 (2021)
26. Salama, F.M., Hamid, N.N.A., Ali, N.H.M., Ali, U.: An efficient modified hybrid explicit group iterative method for the time-fractional diffusion equation in two space dimensions. *AIMS Math.* **7**(2), 2370–2392 (2021)
27. Salama, F.M., Hamid, N.N.A., Ali, U., Ali, N.H.M.: Fast hybrid explicit group methods for solving 2D fractional advection-diffusion equation. *AIMS Math.* **7**(9), 15854–15880 (2022)
28. Scher, H., Montroll, E.: Anomalous transit-time dispersion in amorphous solids. *Phys. Rev. B* **12**(6), 2455–2477 (1975)
29. Shen, J., Tang, T., Wang, L.-L.: *Spectral Methods: Algorithms, Analysis and Applications*. Springer-Verlag, Heidelberg (2011)
30. Smolyak, S.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl. Akad. Nauk SSSR* **148**, 1042–1045 (1963)
31. Stynes, M.: Too much regularity may force too much uniqueness. *Fract. Calc. Appl. Anal.* **19**(6), 1554–1562 (2016)
32. Stynes, M., Riordan, E.O., Gracia, J.L.: Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. *SIAM J. Numer. Anal.* **55**(2), 1057–1079 (2017)
33. Tang, T., Yu, H.-J., Zhou, T.: On energy dissipation theory and numerical stability for time-fractional phase-field equations. *SIAM J. Sci. Comput.* **41**(6), A3757–A3778 (2019)
34. Ullah, M.S., Higazy, M., Kabir, K.A.: Modeling the epidemic control measures in overcoming COVID-19 outbreaks: A fractional-order derivative approach. *Chaos Soliton. Fract.* **155**, 111636 (2022)
35. Yan, Y.-B., Khan, M., Ford, N.J.: An analysis of the modified L1 scheme for time-fractional partial differential equations with nonsmooth data. *SIAM J. Numer. Anal.* **56**(1), 210–227 (2018)
36. Yserentant, H.: On the multi-level splitting of finite element spaces. *Numer. Math.* **49**, 379–412 (1986)

37. Zhang, Y.-N., Sun, Z.-Z., Liao, H.-L.: Finite difference methods for the time fractional diffusion equation on non-uniform meshes. *J. Comput. Phys.* **265**, 195–210 (2014)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.