Journal of Applied Analysis and Computation
Volume *, Number *, * *, 1–25
Website:http://www.jaac-online.com
DOI:10.11948/***

A FAST PARALLEL DIFFERENCE METHOD FOR SOLVING THE TIME-FRACTIONAL GENERALIZED FISHER EQUATION

Longtao Chai[1], Lifei Wu[1] and Xiaozhong Yang[1,†]

Abstract As a nonlinear fractional reaction-diffusion equation, the Time-Fractional Generalized Fisher (TFGF) equation is deeply rooted in physics, and its fast numerical methods' research has essential scientific significance and practical value. For the time-fractional generalized Fisher equation, based on the alternating segment technique, a parallel computation method for the Fast Alternating Segment Explicit-Implicit (FASE-I) difference scheme is proposed. The time-fractional derivative is approximated by the fast L1 algorithm, while the spatial derivative is discretized by the alternating segment explicit-implicit difference method, the nonlinear term is processed using extrapolation. Theoretically analyze the FASE-I method's uniqueness, stability, and convergence. Compared with the three classic difference methods, numerical experimental results show that the FASE-I method not only has good computational accuracy; but also significantly improves computational efficiency. The FASE-I method is efficient and feasible for resolving the time-fractional generalized Fisher equation.

Keywords Time-fractional generalized Fisher (TFGF) equation, fast L1 approximation, alternating segment difference scheme, parallel computing, stability and convergence.

MSC(2010) 65M06, 65M12.

1. Introduction

The nonlinear fractional reaction-diffusion equations are used in many fields widely such as fluid dynamics, control theory, biology, etc.; the Time-Fractional Generalized Fisher (TFGF) equation is a classical fractional reaction-diffusion equation [1–3]. The TFGF equation often lacks explicit analytical solution, making the need for effective numerical algorithms crucial. Currently, the primary numerical solution methods for the TFGF equation consist of the finite difference method, finite element method, spectral method, etc. [4–6].

Fractional derivatives have historical memory, and the present time value is contingent on the values maintained since the beginning, resulting in substantial computational and storage costs in solving fractional differential equations; finding fast algorithms is a hot research topic. Jiang et al. (2017) [7] introduced a technique

[†] The corresponding author.

[1] School of Mathematics and Physics, North China Electric Power University, Beijing, 102206 Beijing, China
Email: chailongtao2@163.com(L.Chai), wulf@ncepu.edu.cn(L.Wu), yxiaozh@ncepu.edu.cn(X.Yang)

using the sum-of-exponentials to estimate the fractional derivative kernel, then developed a fast algorithm relying on the L1 method. The algorithm reduces the computational and storage costs while maintaining stability and convergence. [8] Proved that the difference in solutions between the L1 method and the fast L1 method for handling semi-linear time-fractional subdiffusion equations can be arbitrarily small and independent of the size of the time and spatial grids. Wang and Huang (2018) [9] constructed a fast algorithm for solving linear and nonlinear fractional diffusion equations by using a polynomial to approximate kernel functions on non-uniform grids. Numerical experiments showed that the algorithm reduces storage requirement and computation time while maintaining convergence. Shen et al. (2020) [10] constructed a fast algorithm based on H2N2 (using quadratic Hermite interpolation and Newton interpolation) method for solving the initial boundary value problem of multi-dimensional time-fractional wave equations. The article proved the stability and convergence of the algorithm, and discussed solutions to the problem of weak regularity in initial time. Gao et al. (2021) [11] proposed a fast compact difference scheme to solve the fourth order time multi-term fractional sub-diffusion equations with the first Dirichlet boundary condition. The problem was transformed into an equivalent low order system using a reduction method, and the multi-term Caputo fractional derivatives were quickly approximated at the superconvergence points. The unconditional stability and convergence analysis of the difference scheme were given.Yuan et al. (2023) [12] proposed a fast numerical algorithm for solving the spatiotemporal fractional Schrödinger equation using a fast algorithm to approximate the time-fractional derivative and a Fourier spectral method to discretize the spatial fractional derivative on non-uniform grids. The algorithm's unconditional convergence was proved using the fractional Sobolev inequality and the boundedness of numerical solutions. Wang et al. (2024) [13] proposed a fast compact finite difference method for solving fourth-order time-fractional diffusion-wave equation. The H2N2 method was used to approximate the Caputo derivative, by using the sum-of-exponentials to approximate the derivative kernels. The stability and convergence of the algorithm were proven through discrete energy method, Cholesky decomposition method, and the reduced-order method.

Considering the rapid advancements in multi-core and cluster technologies, there's been significant progress in exploring parallel algorithms for differential equations [14–16], among which new progress has been made in the parallelization of traditional difference schemes. He and Chen (2016) [17] used the Alternating Group Explicit (AGE) method and Alternating Direction Implicit (ADI) method to solve parabolic equations. In contrast, the ADI method was used for small domain solving and provided initial conditions for the AGE method. The accuracy and effectiveness of the algorithm are verified by numerical experiments. Xue and Feng (2020) [18] constructed an Alternating Segment Explicit-Implicit (ASE-I) parallel difference method based on the alternating segment technique to solve the Burgers equation combining the classical explicit scheme, implicit scheme, and two asymmetric schemes. The linear stability of the method was theoretically proven. Wu and Yang (2020) [19] constructed Pure Alternating Segment Explicit-Implicit (PASE-I) and Implicit-Explicit (PASI-E) parallel difference methods based on the alternating segment technique by the classical explicit and implicit schemes for solving the time-fractional telegraph equation, and provided stability and convergence analysis of the difference schemes. Yan et al. (2021) [20] studied a Mixed Alternating Segment Crank-Nicolson (MASC-N) parallel difference method to solve the time-space fractional Black-Schole model. The method is constructed by using classical explicit, implicit, and C-N schemes based on the alternating segment technique. The unconditional stability and convergence of the method have been proven. Liu and Wu (2023) [21] constructed a Hybrid Alternating Band Crank-Nicolson (HABC-N) parallel difference method using the classical explicit, implicit, and C-N schemes based on alternating band technique, to solve the two-dimensional time-fractional Fisher equation. The HABC-N method's unconditional stability and convergence were proven.

Inspired by the above literature, our focus is not on parallel algorithms from the numerical algebra perspective, but rather on conventional difference scheme parallelization, aiming to surmount numerical algebra's challenges and pave the way for alternative parallelization methods. Based on the alternating segment technique, this paper combines the fast L1 approximation to provide the FASE-I parallel difference method for the TFGF equation. The unconditional stability and convergence of the FASE-I method have been theoretically proven, and the theoretical analysis has been verified through numerical experiments.

Considering the following TFGF equation [22, 23]:

$$\begin{cases} {}_0^C D_t^\alpha u(x,t) = \dfrac{\partial^2 u(x,t)}{\partial x^2} + f(u) + g(x,t), (L_l \le x \le L_r, 0 \le t \le T) \\ u(x,0) = \varphi_0(x), (L_l \le x \le L_r) \\ u(L_l,t) = \mu_1(t), u(L_r,t) = \mu_2(t), (0 \le t \le T), \end{cases} \tag{1.1}$$

where $0 < \alpha \le 1$, $f(u) = \lambda u(x,t)(1 - u^\delta(x,t))$, λ is a real number, δ is a positive integer, $g(x,t)$ is the non-homogeneous term. L_l and L_r are the left and right boundaries, $\varphi_0(x), \mu_1(t), \mu_2(t)$ are the given known functions.

When $\alpha = 1$, $\delta = 1$, Eq. (1.1) is an integer-order Fisher equation used to explain the space-time propagation of viral genes in infinite media. When $\alpha \in (0,1)$, Eq. (1.1) is the TFGF equation used to describe the diffusion process of biological populations, $u(x,t)$ represents population density. The application of this equation and its alternative versions spans numerous disciplines, including neurophysiology, chemical dynamics, and branching Brownian motion processes, etc. [24–27].

The definition of ${}_0^C D_t^\alpha u(x,t)$ is given as follows:

$$ {}_0^C D_t^\alpha u(x,t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{\partial u(x,\xi)}{\partial \xi} \frac{d\xi}{(t-\xi)^\alpha}. \tag{1.2}$$

It is usually assumed that $f(u)$ is Lipschtiz continuous with respect to u [28, 29], a constant $L > 0$ exists such that:

$$|f(u_1) - f(u_2)| \le L\,|u_1 - u_2|\,. \tag{1.3}$$

Similar to the handling method in [30, 31], it is assumed that L and T satisfy the following inequality:

$$L < \frac{1}{T^\alpha \Gamma(2-\alpha)}. \tag{1.4}$$

To ensure the stability and convergence of the numerical scheme, the numerical analysis throughout the article is based on the condition (1.4). The numerical experimental results confirm the feasibility of this hypothesis.

2. The TFGF equation's FASE-I difference scheme

2.1. Fast L1 approximation of the time-fractional derivatives

M and N are the given positive integers. Let $h = \frac{L_r - L_l}{M}$, $\tau = \frac{T}{N}$; the solution area is divided into a network with grid points (x_i, t_n), among $x_i = L_l + ih$, $(i = 0, 1, 2, \cdots, M)$, $t_k = k\tau$, $(k = 0, 1, 2, \cdots, N)$, $u_i^n = u(x_i,\ t_n)$ represents the analytical solution at (x_i, t_k), U_i^k represents the numerical solution at (x_i, t_k). $f_i^n = f(u(x_i, t_n), x_i, t_n)$, $g_i^n = g(x_i, t_n)$.

The Caputo fractional derivative's L1 approximation formula is as follows [4]:

$$\begin{aligned} {}_0^C D_t^\alpha u(t)|_{t=t_{k+1}} &\approx D_t^\alpha u(t)|_{t=t_{k+1}} \\ &= \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}\left[\overline{a}_0 u(t_{k+1}) - \sum_{j=1}^{k}(\overline{a}_{k-j} - \overline{a}_{k-j+1})u(t_j) - \overline{a}_k u(t_0)\right], \end{aligned} \tag{2.1}$$

where $\overline{a}_l = (l+1)^{1-\alpha} - l^{1-\alpha}$, $l \geq 0$.

Lemma 2.1 ([7, 11]). *When $\alpha \in (0,1)$, $\tau > 0$, $\varepsilon > 0$, $T > 0$ with $\tau < T$, there exists a positive interger $N_{\exp}$ and positive numbers s_l, w_l, $(l = 1, 2, \cdots, N_{\exp})$ satisfied the following inequality:*

$$\left| t^{-\alpha} - \sum_{l=1}^{N_{\exp}} w_l e^{-s_l t} \right| \leq \varepsilon, \forall t \in [\tau, T].$$

Furthermore, the exponential function's term count is estimated as follows:

$$N_{\exp} = O\left(\left(\log\frac{1}{\varepsilon}\right)\left(\log\log\frac{1}{\varepsilon} + \log\frac{T}{\tau}\right) + \left(\log\frac{1}{\tau}\right)\left(\log\log\frac{1}{\varepsilon} + \log\frac{T}{\tau}\right)\right).$$

By Lemma 2.1, the Caputo fractional derivative's fast L1 approximation algorithm can be obtained [4, 7]:

$$\begin{cases} {}^{\mathcal{F}}D_t^\alpha u(t)|_{t=t_{k+1}} = \frac{1}{\Gamma(1-\alpha)}\left[\sum\limits_{l=1}^{N_{\exp}} w_l F_l^{k+1} + \frac{\tau^{-\alpha}}{1-\alpha}(u(t_{k+1}) - u(t_k))\right], (k \geq 0) \\ F_l^1 = 0, (1 \leq l \leq N_{\exp}) \\ F_l^{k+1} = e^{-s_l\tau}F_l^k + B_l\left[u(t_k) - u(t_{k-1})\right], (1 \leq l \leq N_{\exp}, k \geq 1), \end{cases} \tag{2.2}$$

where $B_l = \int_0^1 e^{-s_l(1+\theta)\tau} d\theta$.

The above algorithm is used for practical programming, and in order to analyze the difference method numerically, it's essential to utilize the subsequent form [4]:

$$\begin{cases} {}^{\mathcal{F}}D_t^\alpha u(t_{k+1}) = \frac{1}{\Gamma(1-\alpha)}\left[a_0 u(t_{k+1}) - \sum\limits_{j=1}^{k}(a_{j-1} - a_j)u(t_{k+1-j}) - a_k u(t_0)\right] \\ a_0 = \frac{\tau^{-\alpha}}{1-\alpha} \\ a_j = \int_0^1 \sum\limits_{l=1}^{N_{\exp}} w_l e^{-s_l(t_j + \theta\tau)} d\theta, (j \geq 1). \end{cases} \tag{2.3}$$

The computational cost of (2.1) is $O(N^2)$, and the computational cost of (2.2) is $O(NN_{\exp})$, $N_{\exp}$ generally not exceeding 200. In cases where N is large, $O\left(NN_{\exp}\right) \ll O\left(N^2\right)$ [4].

2.2. The construction of FASE-I difference scheme

Drawing from the classical explicit and implicit difference schemes, the design of the FASE-I scheme is as follows:

Set $M-1=ql$, with q, $l \in N^{+}$, $l \geq 3$, q is odd and $q \geq 3$. Firstly, all the time layers are divided into odd and even layers, and the inner points of the spatial grid are segmented into q segments. The computation in even layers proceeds from left to right, alternating according to the "explicit difference-implicit difference-explicit difference" rule. Within the subsequent odd layers, Within the subsequent odd layers, calculations adhere to the alternating "implicit difference-explicit difference-implicit difference" rule. The node segments in the FASE-I scheme are shown in Figure 1, where ○ represents the point determined in the explicit difference and □ represents the point determined in the implicit difference:

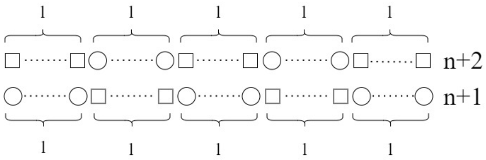


Figure 1: The node segments in FASE-I scheme

To construct the FASE-I scheme, let $b=\frac{1}{a_0}=(1-\alpha)\tau^{\alpha}$, $c=\Gamma(2-\alpha)\tau^{\alpha}$, $a=\frac{c}{h^2}, l_k=a_{k-1}-a_k$, $(k=1,2,\cdots,N-1)$.

The explicit and implicit difference schemes of Eq. (1.1) are given firstly:

$$\begin{aligned}
{}_{0}^{C}D_{t}^{\alpha}u(x_i,t_{k+1}) &= \frac{1}{h^2}(U_{i-1}^{k}-2U_{i}^{k}+U_{i+1}^{k})+f_{i}^{k}+g_{i}^{k},\\
{}_{0}^{C}D_{t}^{\alpha}u(x_i,t_{k+1}) &= \frac{1}{h^2}(U_{i-1}^{k+1}-2U_{i}^{k+1}+U_{i+1}^{k+1})+f_{i}^{k+1}+g_{i}^{k+1}.
\end{aligned} \tag{2.4}$$

Substituting the fast L1 approximation (2.2) into Eq. (2.4) converts to:

$$\begin{aligned}
&U_{i}^{k+1}=aU_{i-1}^{k}+(1-2a)U_{i}^{k}+aU_{i+1}^{k}-b\sum_{l=1}^{N_{\text{exp}}}w_lF_l^{k+1}+cf_i^k+cg_i^k,\\
&-aU_{i-1}^{k+1}+(1+2a)U_{i}^{k+1}-aU_{i+1}^{k+1}=U_i^k-b\sum_{l=1}^{N_{\text{exp}}}w_lF_l^{k+1}+cf_i^{k+1}+cg_i^{k+1}.
\end{aligned} \tag{2.5}$$

By using the extrapolation method to deal with the nonlinear term $f(u)$ [32,33]:

$$f(u_i^k)=2f(u_i^{k-1})-f(u_i^{k-2})+O\left(\tau^2\right),(k\geq 2),$$

a second-order numerical accuracy in time can be obtained.

Expression of the TFGF's FASE-I scheme is as follows: $(k=0,2,\cdots,N-1)$

$$\left\{\begin{aligned}
(I+aG_1)U^{k+1}= \quad & (I-aG_2)U^k-b\sum_{l=1}^{N_{\text{exp}}}w_lF_l^{k+1}+S_1(f^k+g^k)\\
& +S_2(f^{k+1}+g^{k+1})+d^k\\
(I+aG_2)U^{k+2}= \quad & (I-aG_1)U^{k+1}-b\sum_{l=1}^{N_{\text{exp}}}w_lF_l^{k+2}+S_1(f^{k+2}+g^{k+2})\\
& +S_2(f^{k+1}+g^{k+1})+d^{k+2}.
\end{aligned}\right. \tag{2.6}$$

To simplify the process of numerical analysis, in combination with Eq. (2.3), the expression of Eq. (2.5) can be transformed into the corresponding equivalent form:

$$U_i^{k+1} = aU_{i-1}^k + (bl_1 - 2a)U_i^k + aU_{i+1}^k + b\sum_{j=1}^{k-1} l_{k+1}U_i^{k-j} + ba_kU_i^0 + cf_i^k + cg_i^k,$$

$$-\,aU_{i-1}^{k+1} + (1+2a)U_i^{k+1} - aU_{i+1}^{k+1} = bl_1U_i^k + b\sum_{j=1}^{k-1} l_{k+1}U_i^{k-j} + ba_kU_i^0 + cf_i^{k+1} + cg_i^{k+1}.$$

Eq. (2.6) can be converted to the subsequent equivalent form: $(k = 2, 4, \cdots, N-1)$

$$\begin{cases} (I + aG_1)U^1 = (I - aG_2)U^0 + cS_1(f^0 + g^0) + cS_2(f^1 + g^1) + d^0 \\ (I + aG_2)U^2 = (bl_1I - aG_1)U^1 + ba_1U^0 + cS_1(f^2 + g^2) + cS_2(f^1 + g^1) + d^2, \end{cases} \tag{2.7}$$

$$\begin{cases} (I + aG_1)U^{k+1} = & (bl_1I - aG_2)U^k + \sum\limits_{j=1}^{k-1} bl_{j+1}U^{k-j} + ba_kU^0 \\ & +cS_1(f^k + g^k) + cS_2(f^{k+1} + g^{k+1}) + d^k \\ (I + aG_2)U^{k+2} = & (bl_1I - aG_1)U^{k+1} + \sum\limits_{j=1}^{k} bl_{j+1}U^{k+1-j} + ba_{k+1}U^0 \\ & +cS_1(f^{k+2} + g^{k+2}) + cS_2(f^{k+1} + g^{k+1}) + d^{k+2}, \end{cases} \tag{2.8}$$

where I represents the unit matrix with an order of $M-1$, $U^k = \left(U_1^k, U_2^k, \cdots, U_{M-1}^k\right)^T$, $d^k = \left(au_0^k, 0, \cdots, 0, au_{M+1}^k\right)^T$, $f^k = \left(f_1^k, f_2^k, \cdots, f_{M-1}^k\right)^T$, $g^k = \left(g_1^k, g_2^k, \cdots, g_{M-1}^k\right)^T$, $(k = 0, 1, 2, \cdots, N)$.

$$G_1 = \begin{pmatrix} Q_{l-1} & & & & & \\ & W_{l+2} & & & & \\ & & Q_{l-2} & & & \\ & & & \ddots & & \\ & & & & W_{l+2} & \\ & & & & & Q_{l-1} \end{pmatrix}_{(M-1)\times(M-1)}$$

$$G_2 = \begin{pmatrix} \overline{W}_{l+1} & & & & & \\ & Q_{l-2} & & & & \\ & & W_{l+2} & & & \\ & & & \ddots & & \\ & & & & Q_{l-2} & \\ & & & & & \overline{\overline{W}}_{l-1} \end{pmatrix}_{(M-1)\times(M-1)}$$

$$W_{l+2}=\begin{pmatrix} 0 & & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & & 0 \end{pmatrix}\quad \overline{W}_{l+1}=\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & & 0 \end{pmatrix}$$

$$\overline{\overline{W}}_{l+1}=\begin{pmatrix} 0 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

$$S_1=\begin{pmatrix} E_l & & & & & \\ & Q_l & & & & \\ & & E_l & & & \\ & & & \ddots & & \\ & & & & Q_l & \\ & & & & & E_l \end{pmatrix}\quad S_2=\begin{pmatrix} Q_l & & & & & \\ & E_l & & & & \\ & & Q_l & & & \\ & & & \ddots & & \\ & & & & E_l & \\ & & & & & Q_l \end{pmatrix}$$

Where E_l denotes the unit matrix of order l, Q_l is the zero matrix of order l, W_{l+2} is the matrix of order $l+2$, $\overline{W}_{l+1}$ and $\overline{\overline{W}}_{l+1}$ are the matrix of order $l+1$, S_1 and S_2 are the matrix of order $M-1$.

The Purely Alternating Segment Explicit-Implicit (PASE-I) difference scheme obtained by substituting the L1 approximation (2.1) into Eq. (2.4): ($k=2,4,\cdots,N-1$)

$$\begin{cases}(I+aG_1)U^1=(I-aG_2)U^0+cS_1(f^0+g^0)+cS_2(f^1+g^1)+d^0\\ (I+aG_2)U^2=(\bar{l}_1I-aG_1)U^1+\overline{a}_1U^0+cS_1(f^2+g^2)+cS_2(f^1+g^1)+d^2,\end{cases}\tag{2.9}$$

$$\begin{cases}(I+aG_1)U^{k+1}= & (\bar{l}_1I-aG_2)U^k+\sum\limits_{j=1}^{k-1}\bar{l}_{j+1}U^{k-j}+b\overline{a}_kU^0\\ & +cS_1(f^k+g^k)+cS_2(f^{k+1}+g^{k+1})+d^k\\ (I+aG_2)U^{k+2}= & (\bar{l}_1I-aG_1)U^{k+1}+\sum\limits_{j=1}^{k}\bar{l}_{j+1}U^{k+1-j}+b\overline{a}_{k+1}U^0\\ & +cS_1(f^{k+2}+g^{k+2})+cS_2(f^{k+1}+g^{k+1})+d^{k+2},\end{cases}\tag{2.10}$$

where $\bar{l}_k=\overline{a}_{k-1}-\overline{a}_k$, $\overline{a}_k=(k+1)^{1-\alpha}-k^{1-\alpha}$.

By substituting the fast L1 approximation (2.2) into the classical Explicit-Implicit (E-I) difference scheme, the Fast Explicit-Implicit (FE-I) difference scheme can be obtained: $(k = 0, 2, \cdots, N-1)$

$$\begin{cases} U_i^{k+1} = \left[I - a(G_1 + G_2)\right] U_i^k - b \sum\limits_{l=1}^{N_{\exp}} w_l F_l^{k+1} + cf_i^k + cg_i^k \\ \left[I - a(G_1 + G_2)\right] U_i^{k+2} = U_i^{k+1} - b \sum\limits_{l=1}^{N_{\exp}} w_l F_l^{k+2} + cf_i^{k+2} + cg_i^{k+2}. \end{cases} \tag{2.11}$$

3. Numerical analysis of the FASE-I parallel difference method

3.1. The existence and uniqueness of FASE-I scheme's solution

Obviously, based on the above FASE-I scheme, the following is evident: $I + aG_1$ and $I + aG_2$ are all strictly diagonal dominant matrices. The existence of the inverse matrices for $I + aG_1$ and $I + aG_2$ leads to the conclusion that:

Theorem 3.1. *The FASE-I scheme's* (2.7)-(2.8) *solution of Eq.* (1.1) *is unique.*

3.2. The stability of the FASE-I scheme

Let v_j^k represent another numerical solution within the FASE-I scheme. Define the error as $\varepsilon_j^k = v_j^k - U_j^k$. Since the edge-value conditions are known, we can deduce that $\varepsilon_0^k = \varepsilon_M^k = 0$, let $E^k = (\varepsilon_1^k, \varepsilon_2^k, \cdots, \varepsilon_{M-1}^k)$, $(k = 1, 2, \cdots, N+1)$.

From Eq. (1.3), there exists a constant $L > 0$ such that:

$$\left| f^k - \overline{f^k} \right| \le L \left| v^k - U^k \right| \le L \left| E^k \right|, \tag{3.1}$$

where $f_j^k = f(v_j^k, x_j, t_k)$, $\overline{f_j^k} = f\left(U_j^k, x_j, t_k\right)$.

Substituting v_j^k and U_j^k into (2.7)-(2.8) and making a difference, then combined with the Eq. (3.1), the following equations can be obtained:

$$\begin{cases} (I + aG_1 - cLS_2)E^1 = (I - aG_2 + cLS_1)E^0 \\ (I + aG_2 - cLS_1)E^2 = (bl_1 I - aG_1 + cLS_2)E^1 + ba_1 E^0, \end{cases} \tag{3.2}$$

when $k = 2, 4, \cdots, N-1$:

$$\begin{cases} (I + aG_1 - cLS_2)E^{k+1} = & (bl_1 I - aG_2 + cLS_1)E^k + \sum\limits_{j=1}^{k-1} bl_{j+1} E^{k-j} \\ & + ba_k E^0 \\ (I + aG_2 - cLS_1)E^{k+2} = & (bl_1 I - aG_1 + cLS_2)E^{k+1} + \sum\limits_{j=1}^{k} bl_{j+1} E^{k+1-j} \\ & + ba_{k+1} E^0. \end{cases} \tag{3.3}$$

Lemma 3.1 ([34]). *For the following N by N tridiagonal matrix:*

$$U_N = \begin{pmatrix} a & b & & & \\ c & a & b & & \\ & \ddots & \ddots & \ddots & \\ & & c & a & b \\ & & & c & a \end{pmatrix}_{N\times N}$$

Its eigenvalues are $\lambda_s = a + 2\sqrt{bc}\cos\frac{s\pi}{N+1}, (s = 1, 2, \cdots, N)$. a, b, c *are real or complex numbers.*

Lemma 3.2. *For any invertible real matrix* A, B, *the conclusion can be drawn for any matrix norm:*

$$\|A\|\left\|B^{-1}\right\| = \frac{\sup\limits_{\|x\|=1}\|Ax\|}{\inf\limits_{\|y\|=1}\|By\|}.$$

Proof. According to the definition of matrix norm:

$$\|A\| = \sup_{x\neq 0}\frac{\|Ax\|}{\|x\|} = \sup_{\|x\|=1}\|Ax\|,$$

$$\left\|B^{-1}\right\| = \sup_{y\neq 0}\frac{\left\|B^{-1}y\right\|}{\|y\|} = \sup_{By\neq 0}\frac{\left\|B^{-1}By\right\|}{\|By\|} = \sup_{y\neq 0}\frac{\|y\|}{\|By\|} = \frac{1}{\inf\limits_{y\neq 0}\frac{\|By\|}{\|y\|}} = \frac{1}{\inf\limits_{\|y\|=1}\|By\|},$$

so the conclusion can be obtained:

$$\|A\|\left\|B^{-1}\right\| = \frac{\sup\limits_{\|x\|=1}\|Ax\|}{\inf\limits_{\|y\|=1}\|By\|}.$$

The proof is complete. □

Theorem 3.2. *The FASE-I scheme* (2.7)-(2.8) *for Eq.* (1.1) *is stable when L and T meet relationship* (1.4).

Proof. It is easy to know the following relationship:

$$S_1 + S_2 = I. \tag{3.4}$$

Using the mathematical induction to prove that: $\left\|E^k\right\| \leq \left\|E^0\right\|$.

Firstly, considering the case where the number of time layers $n = 1$, combined the Lemma 3.2 and the Eq. (3.4), the Eq. (3.2) can be made simpler and apply 2-norm to each side of the equation:

$$\begin{aligned} \left\|E^1\right\| &= \left\|(I + aG_1 - cLS_2)^{-1}(I - aG_2 + cLS_1E^0)\right\| \\ &\leq \left\|(I + aG_1 - cLS_2)^{-1}\right\|\|(I - aG_2 + cLS_1)\|\left\|E^0\right\| \\ &\leq \left|\frac{1 - a\lambda_{2,k} + cL}{1 + a\lambda_{1,k} - cL}\right|\left\|E^0\right\| \\ &\leq \left|1 - \frac{a\lambda_{1,k} + a\lambda_{2,k} - 2cL}{1 + a\lambda_{1,k} - cL}\right|\left\|E^0\right\|, \end{aligned}$$

where $\lambda_{1,k}$, $\lambda_{2,k}$ are the smallest eigenvalue corresponding to G_1, G_2. From Lemma 3.1 it can be obtained: $\lambda_{1,k} \approx \lambda_{2,k} \approx 2 + 2\cos\frac{(l+1)\pi}{l+2}$.

When $\frac{a\lambda_{1,k}+a\lambda_{2,k}-2cL}{1+a\lambda_{1,k}-cL} \in (0,2)$, there is $\left\|E^1\right\| \leq \left\|E^0\right\|$.

(1) Consider the left end of the inequality:

when $L < \frac{1}{c} = \frac{1}{\tau^\alpha\Gamma(2-\alpha)} \leq \frac{1}{T^\alpha\Gamma(2-\alpha)}$, $1 + a\lambda_{1,k} - cL > 0$ always holds.

Also, when $L < \frac{a(\lambda_{1,k}+\lambda_{2,k})}{2c} = \frac{2+2\cos\frac{(l+1)\pi}{l+2}}{h^2}$, the following inequality always holds:

$$a\lambda_{1,k} + a\lambda_{2,k} - 2cL > 0.$$

Verified by numerical experiments, for any h, $\frac{1}{T^\alpha\Gamma(2-\alpha)} < \frac{2+2\cos\frac{(l+1)\pi}{l+2}}{h^2}$ holds.

In summary, when $L < \frac{1}{T^\alpha\Gamma(2-\alpha)}$:

$$\frac{a\lambda_{1,k} + a\lambda_{2,k} - 2cL}{1 + a\lambda_{1,k} - cL} > 0.$$

(2) Consider the right end of the inequality:

$$\frac{a\lambda_{1,k} + a\lambda_{2,k} - 2cL}{1 + a\lambda_{1,k} - cL} < 2,$$

$$a\lambda_{2,k} - a\lambda_{1,k} < 2,$$

the above inequality obviously holds.

So when L and T meet relationship (1.4), it can be obtained: $\left\|E^1\right\| \leq \left\|E^0\right\|$.

Due to $l_k = a_{k-1} - a_k$, $(k = 1, 2, \cdots, N)$, $b = \frac{1}{a_0}$, the following conclution can be obtained, for any $k = 1, 2, \cdots, N$:

$$b\left(l_1 + \sum_{j=1}^{k-1} l_{j+1} + a_k\right) = b(a_0 - a_1 + a_1 - a_2 + \cdots + a_{k-1} - a_k + a_k) = 1. \quad (3.5)$$

The Eq. (3.2) can be made simpler and apply 2-norm to each side of the equation:

$$\begin{aligned}\left\|E^2\right\| &\leq \left\|(I + aG_2 - cLS_1)^{-1}\right\| \left\|(bl_1 I - aG_1 + cLS_2 + ba_1 I)\right\| \left\|E^0\right\| \\ &= \left\|(I + aG_2 - cLS_1)^{-1}\right\| \left\|(I - aG_1 + cLS_2)\right\| \left\|E^0\right\| \\ &\leq \left|\frac{1 - a\lambda_{1,k} + cL}{1 - a\lambda_{2,k} - cL}\right| \left\|E^0\right\| \\ &\leq \left|1 - \frac{a\lambda_{1,k} + a\lambda_{2,k} - 2cL}{1 + a\lambda_{2,k} - cL}\right| \left\|E^0\right\|.\end{aligned}$$

Similar to the analytical process for the number of time layers $n = 1$, when L and T meet relationship (1.4), $\left\|E^2\right\| \leq \left\|E^0\right\|$ can be obtained.

Assume that when the time layer $n = k$, there is $\left\|E^k\right\| \leq \left\|E^0\right\|$, then when the number of time layers $n = k + 1, n = k + 2$, combined Lemma 3.2 and Eq. (3.5) to simplify the Eq. (3.3):

$$\begin{aligned}
\left\|E^{k+1}\right\| &= \left\|(I+aG_1-cLS_2)^{-1}\left[(bl_1I+cLS_1-aG_2)E^k+\sum_{j=1}^{k-1}bl_{j+1}E^{k-j}+ba_kE^0\right]\right\| \\
&\leq \left\|(I+aG_1-cLS_2)^{-1}\left[b(l_1+\sum_{j=1}^{k-1}l_{j+1}+a_k)I-aG_2+cLS_1\right]\right\|\left\|E^0\right\| \\
&\leq \left\|(I+aG_1-cLS_2)^{-1}\right\|\left\|(I-aG_2+cLS_1)\right\|\left\|E^0\right\| \\
&\leq \left|\frac{1-a\lambda_{2,k}+cL}{1+a\lambda_{1,k}-cL}\right|\left\|E^0\right\| \\
&\leq \left|1-\frac{a\lambda_{1,k}+a\lambda_{2,k}-2cL}{1+a\lambda_{1,k}-cL}\right|\left\|E^0\right\|,
\end{aligned}$$

$$\begin{aligned}
\left\|E^{k+2}\right\| &= \left\|(I+aG_2-cLS_1)^{-1}\left[(bl_1I+cLS_2-aG_1)E^{k+1}+\sum_{j=1}^{k}bl_{j+1}E^{k+1-j}+ba_{k+1}E^0\right]\right\| \\
&\leq \left\|(I+aG_2-cLS_1)^{-1}\left[b(l_1+\sum_{j=1}^{k}l_{j+1}+a_{k+1})I-aG_1+cLS_2\right]\right\|\left\|E^0\right\| \\
&\leq \left\|(I+aG_2-cLS_1)^{-1}\right\|\left\|(I-aG_1+cLS_2)\right\|\left\|E^0\right\| \\
&\leq \left|\frac{1-a\lambda_{1,k}+cL}{1+a\lambda_{2,k}-cL}\right|\left\|E^0\right\| \\
&\leq \left|1-\frac{a\lambda_{1,k}+a\lambda_{2,k}-2cL}{1+a\lambda_{2,k}-cL}\right|\left\|E^0\right\|.
\end{aligned}$$

Similar to the analytical process for the number of time layers $n=1$ and $n=2$, when L and T meet relationship (1.4), we can obtain $\left\|E^{k+1}\right\|\leq\left\|E^0\right\|$, $\left\|E^{k+2}\right\|\leq\left\|E^0\right\|$.

By mathematical induction, the proof is complete. □

3.3. The convergence of the FASE-I scheme

Lemma 3.3 ([4,7]). *For $\alpha\in(0,1)$, suppose $u\in C^2[t_0,t_{n+1}]$, then we have:*

$$\begin{aligned}
|R(u(t_{n+1}))| \leq & \frac{1}{2\Gamma(1-\alpha)}\left[\frac{1}{4}+\frac{\alpha}{(1-\alpha)(2-\alpha)}\right]\max_{t_0\leq t\leq t_{n+1}}|u''(t)|\tau^{2-\alpha} \\
&+\frac{\varepsilon t_{n+1}}{\Gamma(1-\alpha)}\max_{t_0\leq t\leq t_{n+1}}|u'(t)|,
\end{aligned}$$

where $R(u(t_{n+1}))={}_0^C D_t^\alpha u(t)|_{t=t_{n+1}}-{}^{\mathcal{F}} D_t^\alpha u(t_{n+1})$.

In general, ε is much smaller than $O\left(\tau^{2-\alpha}\right)$, so in accuracy analysis, we usually only consider the case of $O\left(\tau^{2-\alpha}\right)$.

Lemma 3.4 ([6]). *For $\alpha \in (0,1)$, if $u \in C^2[0,t_{n+1}]$, the conclusion can be obtained:*

$$\begin{aligned}\frac{\partial^{1+\alpha}u(t_{n+1})}{\partial t^{1+\alpha}} =&\frac{1}{\Gamma(1-\alpha)}\int_0^{t_{n+1}}\frac{\partial^2 u(\xi)}{\partial \xi^2}\frac{1}{(t_{n+1}-\xi)^\alpha}d\xi\\ =&\frac{1}{\Gamma(1-\alpha)}\sum_{k=1}^{n+1}\int_{t_{k-1}}^{t_k}\frac{\partial^2 u(\xi)}{\partial \xi^2}\frac{1}{(t_{n+1}-\xi)^\alpha}d\xi\\ \le&\frac{1}{\Gamma(1-\alpha)}\max_{t_0\le t\le t_{n+1}}\left\{\left|\frac{\partial^2 u(t)}{\partial t^2}\right|\right\}\sum_{k=1}^{n+1}\int_{t_{k-1}}^{t_k}\frac{1}{(t_{n+1}-\xi)^\alpha}d\xi\\ \le&\frac{(n+1)^{1-\alpha}C}{\Gamma(2-\alpha)}\tau^{1-\alpha}=\frac{(t_{n+1})^{1-\alpha}C}{\Gamma(2-\alpha)},\end{aligned}$$

where $C = \max\limits_{t_0\le t\le t_{n+1}}\left\{\left|\frac{\partial^2 u(t)}{\partial t^2}\right|\right\}$.

Lemma 3.5 ([4]). *The coefficients $\{a_k|0\le k\le N\}$ defined by (2.3) satisfy the following inequality:*

$$a_1 > a_2 > \cdots > a_N,$$

if $\varepsilon < \frac{2-2^{1-\alpha}}{1-\alpha}\tau^{-\alpha}$, the following conclusion can be obtained:

$$\begin{cases} a_0 > a_1, a_0 = \frac{\tau^{-\alpha}}{1-\alpha} \\ (t_{k+1})^{-\alpha}-\varepsilon < a_k < (t_k)^{-\alpha}+\varepsilon, (k\ge 1).\end{cases}$$

Firstly, the analysis of the FASE-I scheme's truncation error is given:

Case1: Combined with Lemma 3.3 to consider the truncation error of the explicit scheme at time layer $k+1$ and the implicit scheme at time layer $k+2$:

$$\begin{aligned} {}^{\mathcal{F}}D_t^\alpha u(x_i,t_{k+1}) + O\left(\tau^{2-\alpha}\right) &= \frac{1}{h^2}\left(u_{i-1}^k - 2u_i^k + u_{i+1}^k\right) + f_i^k + R_1,\\ {}^{\mathcal{F}}D_t^\alpha u(x_i,t_{k+2}) + O\left(\tau^{2-\alpha}\right) &= \frac{1}{h^2}\left(u_{i-1}^{k+2} - 2u_i^{k+2} + u_{i+1}^{k+2}\right) + f_i^{k+2} + R_2.\end{aligned} \tag{3.6}$$

Taylor expansion of each term in Eq. (3.6) separately at u_i^{k+1}, combined with Lemma 3.4 the following estimating equations can be obtained:

$$\begin{aligned} R_1 &= -\frac{h^2}{12}u_{xxxx} - \frac{\tau^2}{2}u_{xxtt} + \tau u_{xxt} - \tau\frac{\partial f}{\partial t} + O\left(\tau + h^2\right),\\ R_2 &= \tau{}_0^C D_t^{1+\alpha}u(t_{k+1}) - \frac{h^2}{12}u_{xxxx} - \frac{\tau^2}{2}u_{xxtt} - \tau u_{xxt} + \tau\frac{\partial f}{\partial t} + O\left(\tau + h^2\right).\end{aligned} \tag{3.7}$$

Case2: Combined with Lemma 3.3 to consider the truncation error of the implicit scheme at time layer $k+1$ and the explicit scheme at time layer $k+2$:

$$\begin{aligned} {}^{\mathcal{F}}D_t^\alpha u(x_i,t_{k+1}) + O\left(\tau^{2-\alpha}\right) &= \frac{1}{h^2}\left(u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}\right) + f_i^{k+1} + R_3,\\ {}^{\mathcal{F}}D_t^\alpha u(x_i,t_{k+2}) + O\left(\tau^{2-\alpha}\right) &= \frac{1}{h^2}\left(u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}\right) + f_i^{k+1} + R_4.\end{aligned} \tag{3.8}$$

Taylor expansion of each term in Eq. (3.8) separately at u_i^{k+1}, combined with Lemma 3.4 the following estimating equations can be obtained:

$$\begin{aligned} R_3 &= -\frac{h^2}{12}u_{xxxx} + O\left(h^2\right),\\ R_4 &= \tau{}_0^C D_t^{1+\alpha}u(t_{k+1}) - \frac{h^2}{12}u_{xxxx} + O\left(\tau + h^2\right).\end{aligned} \tag{3.9}$$

When the explicit and implicit schemes alternate at different time layers, it can be observed from Eq. (3.7) and Eq. (3.9) that although some truncation errors can be offset, the time accuracy of the truncation error is still $O(\tau)$ due to the presence of $\tau {}_0^C D_t^{1+\alpha} u$. For ease of analysis, we assume that:

$$R_1 = R_2 = R_3 = R_4 = O(\tau + h^2). \tag{3.10}$$

Substituting Eq. (3.10) into Eq. (3.6) and Eq. (3.8), then the following equations can be obtained:

$$U_i^{k+1} = aU_{i-1}^k + (bl_1 - 2a)U_i^k + aU_{i+1}^k + b\sum_{j=1}^{k-1} l_{k+1}U_i^{k-j} + ba_kU_i^0 + O\left(\tau^{1+\alpha} + \tau^\alpha h^2\right),$$

$$-aU_{i-1}^{k+1} + (1+2a)U_i^{k+1} - aU_{i+1}^{k+1} = bl_1U_i^k + b\sum_{j=1}^{k-1} l_{k+1}U_i^{k-j} + ba_kU_i^0 + O\left(\tau^{1+\alpha} + \tau^\alpha h^2\right).$$

Based on the preceding equations, it's possible to ascertain the computational accuracy of the FASE-I scheme (2.7)-(2.8) as $O\left(\tau^{1+\alpha} + \tau^\alpha h^2\right)$.

Nextly, let $e_j^k = u_j^k - U_j^k$, $e_0^k = e_M^k = 0$, $e^k = (e_1^k, \cdots, e_{M-1}^k)$, $e^0 = 0$, $f_j^k = f(u_j^k, x_j, t_k)$, $\overline{f_j^k} = f(U_j^k, x_j, t_k)$, $(1 \le j \le M-1)$.

From Eq. (1.3), there exists a constant $L > 0$ such that:

$$\left|f^k - \overline{f^k}\right| \le L\left|u^k - U^k\right| \le L\left|e^k\right|. \tag{3.11}$$

Theorem 3.3. *If the solution of Eq.* (1.1) *satisfies* $u \in C^2[0, t_{n+1}]$, *the FASE-I scheme* (2.7)-(2.8) *is convergent, and the following conclusion can be obtained :*

$$\|e^n\| \le C\left(\tau^{1+\alpha} + \tau^\alpha h^2\right),$$

when L *and* T *meet relationship* (1.4), $n = 1, 2, \cdots, N, C > 0$, C *is a constant.*

Proof. Substituting u_j^k, U_j^k into (2.7)-(2.8) and making the difference, then combined with the Eq. (3.11), the following equations can be obtained:

$$\begin{cases} (I + aG_1 - cLS_2)\, e^1 = (I - aG_2 + cLS_1)\, e^0 + R^1 \\ (I + aG_2 - cLS_1)\, e^2 = (bl_1I - aG_1 + cLS_2)\, e^1 + ba_1e^0 + R^2, \end{cases}$$

when $k = 2, 4, \cdots, N-1$:

$$\begin{cases} (I + aG_1 - cLS_2)\, e^{k+1} = & (bl_1I - aG_2 + cLS_1)\, e^k + \sum\limits_{j=1}^{k-1} bl_{j+1}e^{k-j} \\ & +ba_ke^0 + R^{k+1} \\ (I + aG_2 - cLS_1)\, e^{k+2} = & (bl_1I - aG_1 + cLS_2)\, e^{k+1} + \sum\limits_{j=1}^{k} bl_{j+1}e^{k+1-j} \\ & +ba_{k+1}e^0 + R^{k+2}, \end{cases}$$

where $R^k = \tau^\alpha O\left(\tau^{1+\alpha} + \tau^\alpha h^2\right)$, there exists $C_1 > 0$ such that $\left\|R^k\right\| \le C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right)$.

Using the mathematical induction to prove it:

When the time layer $n = 1$, shift the terms on both sides of the equation and take 2-norm:

$$\left\|e^1\right\| = \left\|(I + aG_1 - cLS_2)^{-1}\left[(I - aG_2 + cLS_1)e^0 + R^1\right]\right\|, \tag{3.12}$$

since $e^0 = 0$, by Lemma 3.2 we can simplify the Eq. (3.12):

$$\left\|e^1\right\| \leq \left\|(I + aG_1 - cLS_2)^{-1}\right\| \left\|R^1\right\| \leq \left|\frac{1}{1 + a\lambda_{1,k} - cL}\right| \left\|R^1\right\|,$$

combined with Lemma 3.1, it is easy to see that when:

$$1 + a\lambda_{1,k} - cL > 1,$$
$$1 + a\left(2 + 2\cos\frac{(l+1)\pi}{l+2}\right) - cL > 1,$$
$$L < \frac{2 + 2\cos\frac{(l+1)\pi}{l+2}}{h^2},$$

$\left\|e^1\right\| \leq \left\|R^1\right\| \leq (ba_0)^{-1} C_1 \left(\tau^{1+\alpha} + \tau^{\alpha} h^2\right)$ can be obtained.

When the time layer $n = 2$, shift the terms on both sides of the equation and take 2-norm, Since $e^0 = 0$, by Lemma 3.2 the following derivation can be obtained:

$$\begin{aligned}
\left\|e^2\right\| =& \left\|(I + aG_2 - cLS_1)^{-1}\left[(bl_1 I - aG_1 + cLS_2)e^1 + ba_1 e^0 + R^2\right]\right\| \\
=& \left\|(I + aG_2 - cLS_1)^{-1}\left[(bl_1 I - aG_1 + cLS_2)e^1 + R^2\right]\right\| \\
\leq& (ba_1)^{-1}\left\|(I + aG_2 - cLS_1)^{-1}\right\| \left\|(bl_1 I - aG_1 + cLS_2) + ba_1 I\right\| C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right) \\
\leq& (ba_1)^{-1}\left|\frac{1 - a\lambda_{1,k} + cL}{1 + a\lambda_{2,k} - cL}\right| C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right) \\
\leq& (ba_1)^{-1}\left|1 - \frac{a\lambda_{1,k} + a\lambda_{2,k} - 2cL}{1 + a\lambda_{2,k} - cL}\right| C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right).
\end{aligned}$$

At this point, similar to the stability analysis for the number of time layer $n = 1$, when L and T meet relationship (1.4), it can be obtained:

$$\left\|e^2\right\| \leq (ba_1)^{-1} C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right).$$

Assume that $\left\|e^k\right\| \leq (ba_{k-1})^{-1} C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right)$ holds for the first k layers, since $e^0 = 0$, then when the time layers $n = k+1, n = k+2$, shift the terms on both sides of the equations and take 2-norm:

$$\left\|e^{k+1}\right\| = \left\|(I + aG_1 - cLS_2)^{-1}\left[(bl_1 I - aG_2 + cLS_1)e^k + \sum_{j=1}^{k-1} bl_{j+1}e^{k-j} + R^{k+1}\right]\right\|,$$

$$\left\|e^{k+2}\right\| = \left\|(I + aG_2 - cLS_1)^{-1}\left[(bl_1 I - aG_1 + cLS_2)e^{k+1} + \sum_{j=1}^{k} bl_{j+1}e^{k+1-j} + R^{k+2}\right]\right\|.$$

We can simplify them by Eq. (3.5):

$$\begin{aligned}
\left\|e^{k+1}\right\| \leq& (ba_k)^{-1}\left\|(I + aG_1 - cLS_2)^{-1}\left[(bl_1 I - aG_2 + cLS_1) + \sum_{j=1}^{k-1} bl_{j+1}I + ba_k I\right]\right\| \\
& C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right) \\
\leq& (ba_k)^{-1}\left\|(I + aG_1 - cLS_2)^{-1}\right\| \left\|I - aG_2 + cLS_1\right\| C_1\left(\tau^{1+2\alpha} + \tau^{2\alpha}h^2\right) \\
\leq& (ba_k)^{-1}\left|1 - \frac{a\lambda_{1,k} + a\lambda_{2,k} - 2cL}{1 + a\lambda_{1,k} - cL}\right| C_1(\tau^{1+2\alpha} + \tau^{2\alpha}h^2),
\end{aligned}$$

$$
\begin{aligned}
\left\|e^{k+2}\right\| \leq (ba_{k+1})^{-1} & \left\|(I+aG_2-cLS_1)^{-1}\left[(bl_1I-aG_1+cLS_2)\,t+\sum_{j=1}^{k} bl_{j+1}I+ba_{k+1}I\right]\right\| \\
& C_1\left(\tau^{1+2\alpha}+\tau^{2\alpha}h^2\right) \\
& \leq (ba_{k+1})^{-1}\left\|(I+aG_2-cLS_1)^{-1}\right\|\left\|I-aG_1+cLS_2\right\| C_1\left(\tau^{1+2\alpha}+\tau^{2\alpha}h^2\right) \\
& \leq (ba_{k+1})^{-1}\left|1-\frac{a\lambda_{1,k}+a\lambda_{2,k}-2cL}{1+a\lambda_{2,k}-cL}\right| C_1(\tau^{1+2\alpha}+\tau^{2\alpha}h^2).
\end{aligned}
$$

Similar to the stability analysis, when L and T meet relationship (1.4):

$$
\left\|e^{k+1}\right\| \leq (ba_k)^{-1}C_1\left(\tau^{1+2\alpha}+\tau^{2\alpha}h^2\right),\ \left\|e^{k+2}\right\| \leq (ba_{k+1})^{-1}C_1\left(\tau^{1+2\alpha}+\tau^{2\alpha}h^2\right).
$$

By mathematical induction, the following conclusion is obtained:

$$
\|e^n\| \leq (ba_{n-1})^{-1}C_1\left(\tau^{1+2\alpha}+\tau^{2\alpha}h^2\right), (n=1,2,\cdots,N). \tag{3.13}
$$

By Lemma 3.5, the following relation holds true:

$$
\begin{gathered}
\lim_{n\to\infty}\{(1-\alpha)\tau^{\alpha}[(n-1)\tau]^{-\alpha}\}^{-1} \leq \lim_{n\to\infty}(ba_{n-1})^{-1} \leq \lim_{n\to\infty}\{(1-\alpha)\tau^{\alpha}(n\tau)^{-\alpha}\}^{-1} \\
\frac{(n-1)^{\alpha}}{1-\alpha} \leq \lim_{n\to\infty}(ba_{n-1})^{-1} \leq \frac{n^{\alpha}}{1-\alpha}, \\
\lim_{n\to\infty}(ba_{n-1})^{-1} = \frac{n^{\alpha}}{1-\alpha}.
\end{gathered}
$$

Then the Eq. (3.13) can be simplified:

$$
\begin{aligned}
\|e^n\| & \leq (ba_{n-1})^{-1}\tau^{\alpha}C_1\left(\tau^{1+\alpha}+\tau^{\alpha}h^2\right) \\
& \leq C\left(\tau^{1+\alpha}+\tau^{\alpha}h^2\right), (n=1,2,\cdots,N),
\end{aligned}
$$

where $C=\frac{T^{\alpha}}{1-\alpha}C_1 \geq \frac{n^{\alpha}\tau^{\alpha}}{1-\alpha}C_1$.

The proof is complete. □

Remark 3.1. When the solution of the Eq. (1.1) satisfies $u \in C^2[0,t_{n+1}]$, the time order of the FASE-I parallel difference scheme converges to $O(\tau^{1+\alpha})$.

Remark 3.2. When the solution of Eq. (1.1) exhibits weak regularity, it usually satisfy [29, 35]:

$$
|\partial_x^n u(x,t)| \leq C, \left|\partial_t^l u(x,t)\right| \leq C(1+t^{\sigma-l}), (n=0,1,2,3,4, l=0,1,2), \tag{3.14}
$$

where C is some fixed constants, σ is the regularity parameter.

[35] provides a detailed discussion of the error estimation and convergence of the L1 method for different regularity parameters. Specifically, when the regularity parameter $\sigma=\alpha$, the time convergence order is $O(\tau)$ for a fixed $t=T$ far away from $t=0$.

Under the above conditions, the time order of the FASE-I method converges to $O(\tau)$ while dealing with TFGF equations with weak regularity, which is consistent with the conclusions presented in [35, 36].

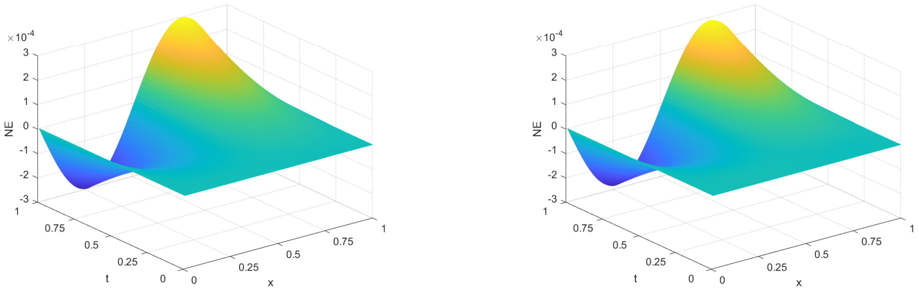
4. Numerical simulation

This section presents three numerical experiments designed to verify the correctness of the theoretical analysis for the FASE-I parallel difference method. The fast L1 approximation (2.3) is selected as $\varepsilon = 10^{-8}$ in the following examples. Numerical experiments are constructed using MATLAB (R2022b) in Intel Xeon Gold 6248R CPU.

Example 4.1. [23] Consider the Eq. (1.1), let $\lambda = 1$, $\delta = 2$, the TFGF equation is given:

$$\begin{cases} {}_0^C D_t^{\alpha} u(x,t) = \dfrac{\partial^2 u(x,t)}{\partial x^2} + u(x,t)(1 - u^2(x,t)) + g(x,t), x \in [0,1], t \in [0,1] \\ u(x,0) = 0, x \in [0,1] \\ u(0,t) = u(1,t) = 0, t \in [0,1], \end{cases}$$

where $g(x,t) = \frac{\Gamma(3)}{\Gamma(3-\alpha)} t^{2-\alpha} \sin(2\pi x) + 4\pi^2 t^2 \sin(2\pi x) - t^2 \sin(2\pi x)(1 - t^4 \sin^2(2\pi x))$, the equation's analytical solution : $u(x,t) = t^2 \sin(2\pi x)$.



(a) NE for E-I

(b) NE for FASE-I

Figure 2: NE distributions of the two methods for Example 4.1 ($\alpha = 0.6$, $M = 101$, $N = 10000$)

Figure 2 illustrates the Node Errors (NE) surfaces of the E-I and FASE-I methods in solving the Example 4.1. The NE for two difference methods are less than 3×10^{-4}, indicating that the two methods' numerical solutions can approximate exact solution well.

For assessing the stability of the FASE-I method, we define the Sum of Relative Error at each Time level (SRET) as follows:

$$SRET(k) = \sum_{j=1}^{M} \frac{\left| u_j^k - U_j^k \right|}{\left| u_j^k \right|}.$$

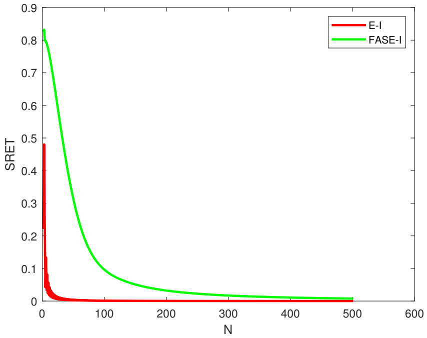


Figure 3: Trend of the two methods' SRET for Example 4.1 ($\alpha = 0.6$, $M = 501$, $N = 500$)

The diagram in Figure 3 illustrates the SRET for both the E-I and FASE-I methods; both methods gradually converge to zero as the layer of time increases. Consequently, the numerical result shows that the FASE-I method is as stable as the classical E-I method, which affirms Theorem 3.2.

To confirm the convergence orders of time and space for the FASE-I method, the following definitions are given [37,38]:

$$E_\infty(h,\tau) = \max_{1\le k\le N} \left| u_j^k - U_j^k \right|,$$

$$Order_h = \frac{\log(E_\infty(h_1,\tau)/E_\infty(h_2,\tau))}{\log(h_1/h_2)}, Order_t = \frac{\log(E_\infty(h,\tau_1)/E_\infty(h,\tau_2))}{\log(\tau_1/\tau_2)}.$$

Table 1: Space convergence orders and CPU of the two methods for Example 4.1 ($N = 10000$)

		E-I scheme			FASE-I scheme		
α	M	$E_\infty(h,\tau)$	$Order_h$	CPU time/s	$E_\infty(h,\tau)$	$Order_h$	CPU time/s
α=0.4	31	3.202635e-3		64.35	3.171612e-3		12.06
	41	1.829549e-3	2.002627	65.74	1.788650e-3	2.048677	12.38
	51	1.181448e-3	2.003764	67.91	1.132906e-3	2.092404	12.39
	61	8.250510e-4	2.005332	70.52	7.667138e-4	2.180574	13.55
α =0.6	31	3.188580e-3		63.29	3.184426e-3		12.27
	41	1.821459e-3	2.002748	66.27	1.815980e-3	2.008860	12.43
	51	1.176170e-3	2.003974	67.45	1.169656e-3	2.015616	12.83
	61	8.213178e-4	2.005651	69.59	8.134908e-4	2.028116	13.38

Table 2: Time convergence orders and CPU of the two methods for Example 4.1 ($M = 2001$)

		E-I scheme			FASE-I scheme		
α	N	$E_\infty(h,\tau)$	$Order_t$	CPU time/s	$E_\infty(h,\tau)$	$Order_t$	CPU time/s
α=0.4	800	2.575060e-5		11.53	6.857804e-2		8.80
	1600	1.352997e-5	0.928447	39.76	2.628100e-2	1.383726	17.70
	3200	6.728497e-6	1.007803	154.14	9.846797e-3	1.416294	35.52
	6400	3.099865e-6	1.118079	638.62	3.675240e-3	1.421816	72.51
α=0.6	800	3.046166e-5		11.75	1.537852e-2		8.82
	1600	1.529489e-5	0.993945	40.19	5.007811e-3	1.618664	17.72
	3200	7.408382e-6	1.045819	156.86	1.639074e-3	1.611299	35.64
	6400	3.368913e-6	1.136875	657.40	5.385275e-4	1.605789	72.73

Table 1 and Table 2 present the numerical errors, space convergence orders, time convergence orders and computational time (CPU) of the E-I and FASE-I methods for Example 4.1. According to Table 1, when the number of spatial point M is equal to a certain value and $N = 10000$, the FASE-I method improves computational efficiency by approximately 80% compared to the E-I method. According to Table 2, when the number of spatial point is fixed at $M = 2001$ and $N = 6400$, the FASE-I method enhances its computational efficiency by approximately 90% compared to the E-I method. The FASE-I method converges to $O\left(\tau^{1+\alpha} + \tau^{\alpha}h^2\right)$, and the experiments confirm the Theorem 3.3.

Table 3: CPU of the two methods for Example 4.1 ($\alpha = 0.6$, $M = 1001$)

N	400	800	1600	3200	6400
PASE-I(CPU/s)	4.49	7.67	17.69	55.89	181.77
FASE-I(CPU/s)	1.03	1.96	4.06	8.16	16.03

To demonstrate the efficiency of the fast L1 method, the FASE-I (2.7)-(2.8) method and the PASE-I (2.9)-(2.10) method for Eq. (1.1) were experimentally compared. The number of spatial points was set at $M = 1001$, and the related computation time is displayed as time layer increases in Table 3.

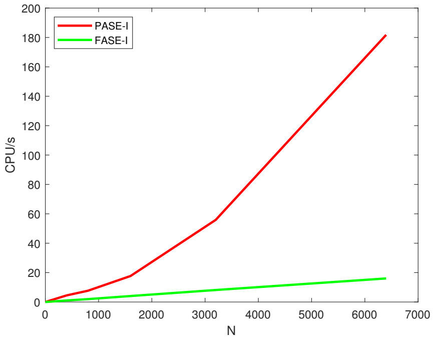


Figure 4: CPU of the PASE-I and FASE-I methods for Example 4.1 ($\alpha = 0.6$, $M = 1001$)

The changes in calculation time for the PASE-I and FASE-I methods with increasing time layer are shown in Figure 4. When the time layer $N = 6400$, the calculation efficiency of the FASE-I method is improved by about 90% compared to the PASE-I method. Numerical experiments demonstrate that the FASE-I method exhibits higher computational efficiency than the PASE-I method, especially in the context of long-term numerical simulations.

To demonstrate the efficiency of parallel algorithms, the FASE-I (2.7)-(2.8) method and the FE-I (2.11) method for Eq. (1.1) were experimentally compared, the speed-up ratio is defined [21]:

$$S_p = \frac{T_1}{T_2},$$

where T_1: calculation time of the FE-I method; T_2: calculation time of the FASE-I method.

Table 4: CPU and speed-up ratio of the two methods for Example 4.1 ($\alpha = 0.6$, $N = 1000$)

M	101	301	501	701	901	1101	1301	1501	1701	1901
FE-I(CPU/s)	0.81	1.25	2.56	4.17	7.11	9.89	13.79	17.75	22.70	27.73
FASE-I(CPU/s)	1.69	2.00	2.30	2.28	2.76	3.11	4.59	5.05	5.65	5.98
S_p	0.48	0.63	1.11	1.83	2.73	3.18	3.00	3.51	4.01	4.64

The fixed number of temporal layer is $N = 1000$. As the number of spatial grid points increases, the corresponding computation time is recorded, as shown in Table 4. When the number of spatial grid points M is greater than or equal to 500, the speed-up ratio exceeds 1. As the number of spatial grid points increases, the speed-up ratio increases linearly, and the improvement in computational efficiency of the FASE-I method becomes more pronounced. This reflects the efficiency of parallel computing using the FASE-I method.

Figure 5 illustrates the variation in computation time between the FE-I and FASE-I methods with an increasing number of spatial grid points. When the spatial grid point count is less than approximately 470, the computation time for the FASE-I method is slightly longer than for the FE-I method, due to the impact of data

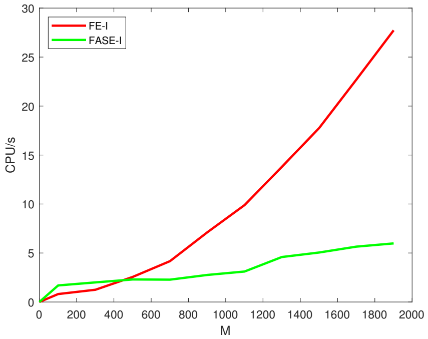


Figure 5: CPU of the FE-I and FASE-I methods for Example 4.1 ($\alpha = 0.6$, $N = 1000$)

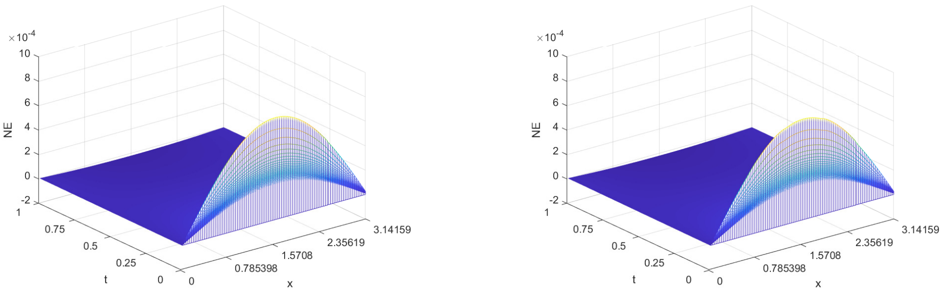
communication between different units. When the spatial grid point count exceeds approximately 400, the computation time for the FASE-I method increases linearly, whereas the computation time for the FE-I method increases exponentially with the number of spatial grid points. Specifically, when $M = 1901$, the calculation efficiency of the FASE-I method is improved by about 78% compared to the FE-I method. This demonstrates that the FASE-I method has superior computational efficiency compared to the FE-I method in scenarios with a larger number of spatial grid points.

Example 4.2. [36] Consider the Eq. (1.1), let $\lambda = 1$, $\delta = 1$, the TFGF equation is given:

$$\begin{cases} {}_0^C D_t^{\alpha} u(x,t) = \dfrac{\partial^2 u(x,t)}{\partial x^2} + u(x,t)(1 - u(x,t)) + g(x,t), x \in [0,\pi], t \in [0,1] \\ u(x,0) = 0, x \in [0,\pi] \\ u(0,t) = u(\pi,t) = 0, t \in [0,1], \end{cases}$$

where $g(x,t) = \Gamma(1+\alpha)\sin(x) + t^{\alpha}\sin(x) - t^{\alpha}\sin(x)(1 - t^{\alpha}\sin(x))$.

The analytical solution: $u(x,t) = t^{\alpha}\sin(x)$, it exhibits initial singularity.



(a) NE for E-I

(b) NE for FASE-I

Figure 6: NE distributions of the two methods for Example 4.2 ($\alpha = 0.6$, $M = 101$, $N = 10000$)

Figure 6 illustrates the NE surfaces of the E-I and FASE-I methods when solving

Example 4.2. The NE of the two difference methods are less than 8.5×10^{-4}, indicating that the numerical solutions obtained by both methods can approximate the exact solution well.

Table 5: Space convergence orders at $t = 1$ and CPU of the two methods for Example 4.2 ($N = 10000$)

		E-I scheme			FASE-I scheme		
α	M	$E_\infty(h,\tau)$	$Order_h$	CPU time/s	$E_\infty(h,\tau)$	$Order_h$	CPU time/s
α=0.4	31	3.198397e-4		72.18	3.198369e-4		12.29
	41	1.810891e-4	2.034554	73.75	1.810862e-4	2.034581	12.19
	51	1.155319e-4	2.059267	75.63	1.155289e-4	2.059313	12.49
	61	7.946320e-5	2.090236	79.02	7.946009e-5	2.090307	13.30
α=0.6	31	3.183294e-4		71.86	3.143253e-4		11.95
	41	1.796185e-4	2.046791	73.64	1.765875e-4	2.062386	12.02
	51	1.140800e-4	2.079853	75.14	1.115087e-4	2.106330	12.50
	61	7.802149e-5	2.121861	78.89	7.570309e-5	2.163012	13.07

Table 6: Time convergence orders at $t = 1$ and CPU of the two methods for Example 4.2 ($M = 2001$)

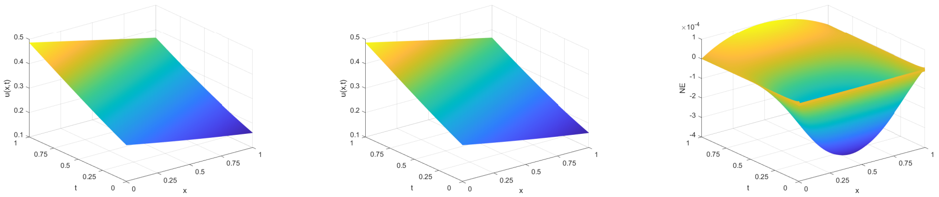
		E-I scheme			FASE-I scheme		
α	N	$E_\infty(h,\tau)$	$Order_t$	CPU time/s	$E_\infty(h,\tau)$	$Order_t$	CPU time/s
α=0.4	800	5.911031e-5		11.14	6.041954e-5		8.12
	1600	2.896203e-5	1.029247	40.36	2.809570e-5	1.104666	16.57
	3200	1.422394e-5	1.025842	154.60	1.365078e-5	1.041366	33.10
	6400	6.988012e-6	1.025367	646.89	6.723727e-6	1.021650	69.98
α=0.6	800	9.518492e-5		11.26	9.962570e-5		8.10
	1600	4.729421e-5	1.009069	39.24	4.764789e-5	1.064106	16.47
	3200	2.352180e-5	1.007665	155.46	2.346326e-5	1.022009	33.12
	6400	1.116938e-5	1.008250	649.49	1.164562e-5	1.010616	68.87

Table 5 and Table 6 present the numerical errors, space convergence orders, time convergence orders, and CPU for the E-I and FASE-I methods applied to Example 4.2. The computational accuracies of the two methods is basically same. As shown in Table 5, with $N = 10000$ and an equal number of spatial grid points M, the FASE-I method shows an approximately 85% improvement in computational efficiency compared to the E-I method. Table 6 indicates that with $M = 2001$ and $N = 6400$, the FASE-I method enhances its computational efficiency by approximately 90% compared to the E-I method. The solution of Example 4.2 features an initial singularity, the FASE-I method converges to $O\left(\tau + h^2\right)$, which is consistent with the conclusion in [35,36] and verifies the Remark 3.2.

Example 4.3. [39] Consider the Eq. (1.1), let $\lambda = 1$, $\delta = 1$, $g(x,t) = 0$, the TFGF equation is given:

$$\begin{cases} {}_0^C D_t^\alpha u(x,t) = \dfrac{\partial^2 u(x,t)}{\partial x^2} + u(x,t)(1-u(x,t)), x \in [0,1], t \in [0,1] \\ u(x,0) = \dfrac{1}{\left(1+\exp^{\frac{x}{\sqrt{6}}}\right)^2}, x \in [0,1] \\ u(0,t) = \dfrac{1}{\left(1+\exp^{-\frac{5t}{6}}\right)^2}, u(1,t) = \dfrac{1}{\left(1+\exp^{\frac{1}{\sqrt{6}}-\frac{5t}{6}}\right)^2}, t \in [0,1]. \end{cases}$$

The equation's analytical solution at $\alpha = 1$ is: $u(x,t) = \frac{1}{\left(1+\exp^{\frac{x}{\sqrt{6}}-\frac{5t}{6}}\right)^2}$, but it lacks an analytical solution when $\alpha \in (0,1)$.



(a) Exact(α=1) **(b)** FASE-I(α=0.99) **(c)** NE for FASE-I

Figure 7: Analytical solution surface, the FASE-I method's numerical solution surface and node error distribution for Example 4.3 ($\alpha = 0.99$, $M = 501$, $N = 1000$)

Figure 7 presents the exact solution surface for Example 4.3 when $\alpha = 1$, and the approximate numerical solution surface obtained by the FASE-I method when $\alpha = 0.99$. The NE is less than 4×10^{-4}, indicating that the fractional-order numerical solution accurately approximates the exact solution.

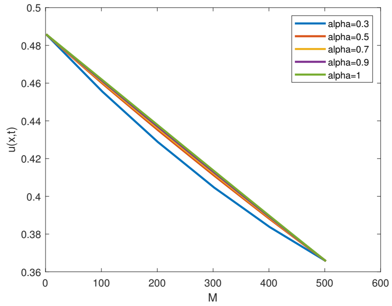


Figure 8: The FASE-I method solution curves ($\alpha = 0.3, 0.5, 0.7, 0.9$) and analytical solution ($\alpha = 1$) curve for Example 4.3 ($M = 501$, $N = 1000$)

Figure 8 presents the solution curves at $t = 1$ for $\alpha = 0.3, 0.5, 0.7, 0.9$ and $\alpha = 1$. It is observable that the solution obtained by the FASE-I method ($\alpha = 0.9$) for Example 4.3 closely approximates the solution of the integer-order generalized Fisher equation ($\alpha = 1$), indicating that the FASE-I method used to solve the TFGF equation that lacks an analytic solution, can still achieve high accuracy.

5. Conclusion

This article studies the fast parallel difference method for the TFGF equation. Utilizing the alternating segment technique and in conjunction with the fast L1 method, we propose the parallel computing method of the Fast Alternating Segment Explicit-Implicit (FASE-I) difference scheme for the TFGF equation. The stability of the FASE-I method has been theoretically established. When the solution u belongs to $C^2[0, t_{n+1}]$, the FASE-I method converges to $O\left(\tau^{1+\alpha} + \tau^\alpha h^2\right)$. In cases where the solution exhibits an initial singularity, the method's convergence rate is $O\left(\tau + h^2\right)$.

The FASE-I method approximates the time fractional derivative using the fast L1 method. As the number of time layers increases, the computational efficiency of the FASE-I method can be improved by up to 90% compared with the E-I and PASE-I methods. The FASE-I method exhibits distinct parallel computing characteristics; as the number of spatial grid points increases, its computational efficiency can be enhanced by up to 80% and 78% compared to the E-I and FE-I methods. The FASE-I method presented in this paper is also suitable for solving the TFGF equation when an analytical solution is not available. Numerical experiments have confirmed the theoretical analysis, the FASE-I method is an efficient numerical method for solving the TFGF equation.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

References

[1] W. Chen, H. Sun and X. Li, *Fractional Derivative Modeling in Mechanics and Engineering*, Springer, Singapore, 2022.

[2] S. Trifce and T. Živorad, *Fractional Equations and Models: Theory and Applications*, Springer, Cham, 2019.

[3] V. V. Uchaikin, *Fractional Derivatives for Physicists and Engineers: Volume II Applications*, Higher Education Press, Beijing, 2013.

[4] Z. Sun and G. Gao, *Fractional Differential Equations:Finite Difference Methods*, De Gruyter, Berlin, Boston, 2020.

[5] W. Deng and Z. Zhang, *High Accuracy Algorithm for the Differential Equations Governing Anomalous Diffusion.*, World Scientific, Singapore, 2019.

[6] F. Liu, P. Zhuang and Q. Liu, *Numerical Methods of Fractional Partial Differential Equations and their Applications*, Science Press, Beijing, 2015.(in Chinese).

[7] S. Jiang, J. Zhang, Q. Zhang and Z. Zhang, *Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations*, Commun. Comput. Phys., 2017, 21(3), 650–678.

[8] Y. Huang, F. Zeng and L. Guo, *Error estimate of the fast L1 method for time-fractional subdiffusion equations*, Appl. Math. Lett., 2022, 133.

[9] K. Wang and J. Huang, *A fast algorithm for the Caputo fractional derivative*, E. Asian. J. Appl. Math., 2018, 8(4), 656–677.

[10] J. Shen, C. Li and Z. Sun, *An H2N2 interpolation for Caputo derivative with order in (1,2) and its application to time-fractional wave equations in more than one space dimension*, J. Sci. Comput., 2020, 83(2).

[11] G. Gao, P. Xu and R. Tang, *Fast compact difference scheme for the fourth-order time multi-term fractional sub-diffusion equations with the first Dirichlet boundary*, J. Appl. Anal. Comput., 2021, 11(6), 2736–2761.

[12] W. Yuan, C. Zhang and D. Li, *Linearized fast time-stepping schemes for time-space fractional Schrödinger equations*, Physica. D., 2023, 454.

[13] W. Wang, H. Zhang, Z. Zhou and X. Yang, *A fast compact finite difference scheme for the fourth-order diffusion-wave equation*, Int. J. Comput. Math., 2024.

[14] A. Ghosh and C. Mishra, *A parallel cyclic reduction algorithm for pentadiagonal systems with application to a convection-dominated heston pde*, Siam. J. Sci. Comput., 2021, 43(2), C177–C202.

[15] M. K. Riahi, *Pitsbicg: Parallel in time stable bi-conjugate gradient algorithm*, Appl. Numer. Math., 2022, 181, 225–233.

[16] M. Yousuf, K. M. Furati and A. Q. M. Khaliq, *A hybrid fourth order time stepping method for space distributed order nonlinear reaction-diffusion equations*, Comput. Math. Appl., 2023, 151, 116–126.

[17] Z. He and R. Chen, *A novel parallel parabolic equation method for electromagnetic scatterings*, Ieee. T. Antenn. Propag., 2016, 64(11), 4777–4784.

[18] G. Xue and H. Feng, *An alternating segment explicit-implicit scheme with intrinsic parallelism for Burgers' equation*, J. Comput. Theor. Trans., 2020, 49(1), 15–30.

[19] L. Wu and X. Yang, *An efficient alternating segment parallel difference method for the time fractional telegraph equation*, Adv. Math. Phys., 2020, 2020.

[20] R. Yan, Y. He and Q. Zuo, *A difference method with parallel nature for solving time-space fractional Black-Schole model*, Chaos. Soliton. Fract., 2021, 151.

[21] R. Liu and L. Wu, *Numerical approach for solving two-dimensional time-fractional Fisher equation via habc-n method*, Commun. Appl. Math. Comput., 2023.

[22] R. Choudhary, S. Singh and D. Kumar, *A high-order numerical technique for generalized time-fractional Fisher's equation*, Math. Method. Appl. Sci., 2023, 46(15, SI), 16050–16071.

[23] P. Roul and V. Rohil, *A high order numerical technique and its analysis for nonlinear generalized Fisher's equation*, J. Comput. Appl. Math., 2022, 406.

[24] Z. Sun, Q. Zhang and G. Gao, *Finite Difference Methods for Nonlinear Evolution Equations*, De Gruyter, Berlin, Boston, 2023.

[25] A. K. Y. Tam and M. J. Simpson, *The effect of geometry on survival and extinction in a moving-boundary problem motivated by the Fisher-KPP equation*, Physica. D., 2022, 438.

[26] F. Wang, M. N. Khan, I. Ahmad et al., *Numerical solution of traveling waves in chemical kinetics: Time-fractional Fishers equations*, Fractals., 2022, 30(02).

[27] M. Adel, D. Baleanu, U. Sadiya et al., *Inelastic soliton wave solutions with different geometrical structures to fractional order nonlinear evolution equations*, Results. Phys., 2022, 38.

[28] B. Zhu, L. Liu and Y. Wu, *Existence and uniqueness of global mild solutions for a class of nonlinear fractional reaction-diffusion equations with delay*, Comput. Math. Appl., 2019, 78(6), 1811–1818.

[29] B. Ghosh and J. Mohapatra, *A novel numerical technique for solving time fractional nonlinear diffusion equations involving weak singularities*, Math. Method. Appl. Sci., 2023, 46(12), 12811–12825.

[30] Y. Yu, W. Deng and Y. Wu, *Positivity and boundedness preserving schemes for the fractional reaction-diffusion equation*, Sci. China. Math., 2013, 56(10), 2161–2178.

[31] D. Li, H. Liao, W. Sun et al., *Analysis of L1-galerkin fems for time-fractional nonlinear parabolic problems*, Commun. Comput. Phys., 2018, 24(1), 86–103.

[32] N. Liu, Y. Liu, H. Li and J. Wang, *Time second-order finite difference/finite element algorithm for nonlinear time-fractional diffusion problem with fourth-order derivative term*, Comput. Math. Appl., 2018, 75(10), 3521–3536.

[33] E. G. M. Elmahdi and J. Huang, *Efficient numerical solution of two-dimensional time-space fractional nonlinear diffusion-wave equations with initial singularity*, J. Appl. Anal. Comput., 2022, 12(2), 831–849.

[34] J. E. Gentle, *Matrix Algebra: Theory, Computations and Applications in Statistics*, Springer International Publishing, Cham, 2017.

[35] D. Li, H. Qin and J. Zhang, *Sharp pointwise-in-time error estimate of L1 scheme for nonlinear subdiffusion equations*, J. Comput. Math., 2024, 42(3), 662–678.

[36] D. Kumar, S. Chaudhary and V. V. K. S. Kumar, *Fractional Crank-Nicolson-Galerkin finite element scheme for the time-fractional nonlinear diffusion equation*, Numer. Meth. Part. D. E., 2019, 35(6), 2056–2075.

[37] S. Vong, P. Lyu and Z. Wang, *A compact difference scheme for fractional sub-diffusion equations with the spatially variable coefficient under Neumann boundary conditions*, J. Sci. Comput., 2016, 66(2), 725–739.

[38] Y. Wang and Y. Zhang, *A Crank-Nicolson-type compact difference method with the uniform time step for a class of weakly singular parabolic integro-differential equations*, Appl. Numer. Math., 2022, 172, 566–590.

[39] H. P. Jani and T. R. Singh, *Aboodh transform homotopy perturbation method for solving fractional-order Newell-Whitehead-Segel equation*, Math. Method. Appl. Sci., 2022.