Numerical solution to a linearized time fractional KdV equation on unbounded domains

Article in Mathematics of Computation · January 2016
DOI: 10.1090/mcom/3229

CITATIONS
0

READS
64

4 authors, including:

Qian Zhang
Beijing Computational Science Research Cen...
2 PUBLICATIONS 3 CITATIONS

Jiwei Zhang
Beijing Computational Science Research Cen...
30 PUBLICATIONS 166 CITATIONS

Zhimin Zhang
Wayne State University
21 PUBLICATIONS 90 CITATIONS

Some of the authors of this publication are also working on these related projects:

Numerical methods for time and space fractional partial differential equations View project

All content following this page was uploaded by Qian Zhang on 17 November 2016.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.
NUMERICAL SOLUTION TO A LINEARIZED TIME FRACTIONAL KDV EQUATION ON UNBOUNDED DOMAINS

QIAN ZHANG, JIWEI ZHANG, SHIDONG JIANG, AND ZHIMIN ZHANG

Abstract. An efficient numerical scheme is developed to solve a linearized time fractional KdV equation on unbounded spatial domains. First, the exact absorbing boundary conditions (ABCs) are derived which reduces the pure initial value problem into an equivalent initial-boundary value problem on a finite interval that contains the compact support of the initial data and the inhomogeneous term. Second, the stability of the reduced initial-boundary value problem is studied in detail. Third, an efficient unconditionally stable finite difference scheme is constructed to solve the initial-boundary value problem where the nonlocal fractional derivative is evaluated via a sum-of-exponentials approximation for the convolution kernel. As compared with the direct method, the resulting algorithm reduces the storage requirement from $O(MN)$ to $O(M \log^d N)$ and the overall computational cost from $O(MN^2)$ to $O(MN \log^d N)$ with $M$ the total number of spatial grid points and $N$ the total number of time steps. Here $d = 1$ if the final time $T$ is much greater than 1 and $d = 2$ if $T \approx 1$. Numerical examples are given to demonstrate the performance of the proposed numerical method.

1. Introduction

The classical Korteweg-de Vries (KdV) equation is a typical dispersive nonlinear partial differential equation (PDE). Historically, the solitary solution of the equation was first observed physically by Scott Russell in 1834 [27], and the equation itself was later derived by Korteweg and de Vries in 1895 [19]. Since then, it has been applied in many fields to describe a wide range of physical phenomena such as interaction of nonlinear waves [33], collision-free hydro-magnetic waves in a cold plasma, ion-acoustic waves, interfacial electrohydrodynamics [9], etc. As a mathematical model of the water wave, when the wave height is small compared to the water depth [32], the nonlinear equation is reduced to the linear KdV equation.

When these physical phenomena are considered non-conservative, they can be described using fractional differential equations. Over the last two decades the fractional calculus has been applied to almost every field of science, engineering, and mathematics [18, 20, 22, 24, 25]. Here we consider pure initial value problem...
(IVP) of the linearized time-fractional KdV equation in one space dimension:

\[
\begin{aligned}
C_0 D_0^\alpha u(x, t) + u_{xxx}(x, t) &= f(x, t), \quad t \in (0, T], \ x \in \mathbb{R}, \\
u(x, 0) &= u_0(x), \quad x \in \mathbb{R}, \\
u(x, t) \to 0, \quad \text{when } |x| \to \infty, \ t \in (0, T],
\end{aligned}
\]

where the source term \(f(x, t)\) and initial value \(u_0(x)\) are assumed to be compactly supported, and the operator \(C_0 D_0^\alpha(0 < \alpha < 1)\) stands for the Caputo fractional derivative and is defined by the formula

\[
C_0 D_0^\alpha u(x, t) = \frac{1}{\Gamma(1 - \alpha)} \int_0^t u'(x, s) (t - s)^{-\alpha} ds.
\]

Although there are some methods to analytically solve the fractional KdV equation in some special cases \([8, 23, 31, 35]\), one may have to resort to the numerical methods to obtain its solution in general. There are two difficulties when one tries to solve (1.1) numerically. First, the original problem is defined on the whole spatial domain, which has to be truncated to a finite domain for the purpose of numerical computation. For this, one needs to impose some artificial boundary conditions at the boundary of the truncated domain. These boundary conditions have to be carefully chosen so that there will be no reflections from the boundary and the reduced initial-boundary value problem (IBVP) is still stable and equals to the original problem in the truncated domain. Second, the fractional derivative is defined as a convolution integral from 0 to the current time \(t\), and a direct method of evaluating the fractional derivative requires the storage of the solution at all times and quadratic complexity in computational cost, which is prohibitively expensive for large-scale long-time simulations.

Among these artificial boundary conditions, the so-called exact absorbing boundary conditions (ABCs) prevent the reflections of the wave from the boundary, which leads to an equivalent IBVP formulation for the original IVP. For the linearized KdV equation of the integer order, Zheng, Han, and Wen \([37]\) derived the exact ABCs via the Laplace transform. Recently, Zhang, Li, and Wu \([34]\) developed a series of high-order local ABCs to approximate the exact ABCs using Padé expansion in the Laplace domain. Besse et al. \([5]\) constructed the discrete ABCs for the fully discrete problem using the \(Z\)-transformation.

In this paper, we first derive exact ABCs for the linearized fractional KdV equation (1.1) with standard techniques in this field. That is, we apply the Laplace transform in time to reduce the equation to an ODE; we then use the condition that the solution has to decay to zero at infinity to obtain the solution of the ODE in the left and right exterior domains, which also leads to boundary conditions at two end points in the Laplace domain; the boundary conditions in the physical space are obtained by inverting back. However, unlike many second order differential equations where only one boundary condition at each end point is needed, the KdV equation involves third order derivatives in space and one more boundary condition is required. All three absorbing boundary conditions are carefully chosen and we then show that the IBVP with our exact ABCs is stable in the \(L^2\) norm.

Next, we construct a delicate finite difference (FD) scheme to discretize the IBVP. Our construction starts from the introduction of an auxiliary variable which reduces the third-order PDE to a second-order one. We then combine the PDE and the exact ABCs to obtain more consistent discrete equations for the boundary points.
This way, we are able to show that the resulting FD scheme is unconditionally stable; whereas the naive FD schemes which discretize the spatial derivatives via central difference for interior points and one-sided differences for boundary points is only conditionally stable.

We further apply the fast algorithm in [15] to reduce the computational and storage cost for the evaluation of the fractional derivatives in the temporal variable. Here the main idea is to approximate the convolution kernel $t^{-1-\alpha}$ (obtained by integration by parts for the integral on the right hand side of (1.2)) by a sum-of-exponentials (SOE) approximation when $t$ is away from the origin. For each spatial point, the convolution with the exponential kernel can be evaluated in $O(1)$ time at each time step via standard recurrence relation or any A-stable ODE solver (see, for example, [2–4, 6, 12–15, 21, 36]). This reduces the computational cost from $O(N^2)$ for direct method to $O(N N^{\text{exp}})$ and the storage cost from $O(N)$ for direct method to $O(N_{\text{exp}})$, where $N$ is the total number of time steps and $N_{\text{exp}}$ is the total number of the exponentials needed in the SOE approximation. When similar SOE approximations are used for the exact ABCs as well, the overall computational cost of our algorithm is $O(M N N_{\text{exp}})$ as compared with $O(M N^2)$ for direct method; and the storage requirement of our algorithm is $O(M N_{\text{exp}})$ as compared with $O(M N)$ for direct method, as the direct method needs to store the solution in the whole computational spatial domain at all times. Here $M$ is the total number of discretization points in space. The number of exponentials $N_{\text{exp}}$ needed in the SOE approximation depends on the prescribed precision $\varepsilon$, the cut-off time step size $\delta$, and the final time $T$. For a fixed absolute precision, $N_{\text{exp}} = O(\log N)$ when $T \gg 1$ and $N_{\text{exp}} = O(\log^2 N)$ when $T \approx 1$ if we assume that $N = T^\alpha$, i.e., a uniform mesh is used in the temporal variable and the time step size is chosen to be $\Delta t = \delta$.

The rest of the paper is organized as follows. In Section 2, we derive the exact ABCs for the linearized fractional KdV equation. In Section 3, we present the stability analysis of the reduced IBVP. In Section 4, we develop an efficient and stable finite difference scheme for solving the IBVP. Some numerical examples are given to demonstrate the performance of our scheme in Section 5. Finally, we conclude our paper with a short summary.

2. The derivation of exact ABCs

We use the artificial boundary methods (ABMs) [11, 30] to construct the exact ABCs. We first introduce artificial boundaries $\Gamma_l := \{x \mid x = x_l\}$ and $\Gamma_r := \{x \mid x = x_r\}$ to divide the original infinite domain into three parts: the computational domain of interest $\Omega_c := (x_l, x_r)$, the left unbounded exterior domain $\Omega_l := (-\infty, x_l)$, and the right unbounded exterior domain $\Omega_r := (x_r, \infty)$. The choices of parameters $x_l$ and $x_r$ are determined such that the initial data $u_0(x)$ and the source term $f$ are compactly supported in $\Omega_c$. We consider the following two exterior problems on $\Omega_l := (-\infty, x_l)$ and $\Omega_r := (x_r, \infty)$, respectively.

\begin{align}
\tag{2.1}
\frac{\partial}{\partial t} u(x, t) + u_{xxx}(x, t) &= 0, \quad t \in (0, T], \quad x \in \Omega_r, \\
\tag{2.2}
u(x, 0) &= 0, \quad x \in \Omega_r, \\
\tag{2.3}u(x, t) &\to 0, \quad \text{when } x \to +\infty, \quad t \in (0, T],
\end{align}

and

\begin{align}
\tag{2.4}
\frac{\partial}{\partial t} u(x, t) + u_{xxx}(x, t) &= 0, \quad t \in (0, T], \quad x \in \Omega_l,
\end{align}
The Laplace transform of the Caputo derivatives is given by the formula
\[ \mathcal{L}_0^\alpha u(x, t) = s^\alpha \hat{u}(x, s) - s^{\alpha-1} u(x, 0), \quad 0 < \alpha \leq 1. \]

Applying the Laplace transform to equations (2.1) and (2.4) with zero initial values (2.2) and (2.5), we obtain an ODE in \( \hat{u}(x, s) \)
\[ s^\alpha \hat{u}(x, s) + \hat{u}_{xxx}(x, s) = 0, \quad x \in \Omega_t \cup \Omega_r. \]
The general solution of (2.7) is given by the formula
\[ \hat{u}(x, s) = c_1(s)e^{\lambda_1(s)x} + c_2(s)e^{\lambda_2(s)x} + c_3(s)e^{\lambda_3(s)x}, \]
where \( \lambda_1(s), \lambda_2(s), \lambda_3(s) \) are the roots of the cubic equation
\[ s^\alpha + \lambda^3 = 0. \]
That is,
\[ \lambda_1(s) = -s^\frac{1}{2} \lambda_2(s) = -s^\frac{1}{2} \omega, \quad \lambda_3(s) = -s^\frac{1}{2} \omega^2 \]
with \( w = e^{2\pi i/3} \). It is easy to see that \( \Re(\lambda_1(s)) < 0, \Re(\lambda_2(s)) > 0, \Re(\lambda_3(s)) > 0. \)

Applying the Laplace transform on (2.3) and (2.6), we have
\[ \hat{u}(x, s) \to 0, \quad \text{when} \quad |x| \to \infty. \]
Thus, on the interval \( \Omega_r \),
\[ \hat{u}(x, s) = c_1(s)e^{\lambda_1(s)x}. \]
Taking derivatives of this solution with respect to \( x \), we obtain
\[ \hat{u}_{xx}(x, s) = \lambda_1^2(s) \hat{u}(x, s) \quad \text{and} \quad \hat{u}_x(x, s) = \lambda_1(s) \hat{u}(x, s), \]
or equivalently
\[ \frac{1}{\lambda_1(s)} \hat{u}_{xx}(x, s) = \hat{u}(x, s) \quad \text{and} \quad \frac{1}{\lambda_1(s)} \hat{u}_x(x, s) = \hat{u}(x, s). \]

Applying the inverse Laplace transform (see [25]) to (2.11) and (2.12) yields
\[ u_{xx}(x, t) = \mathcal{L}_0^\alpha \hat{u}(x, s) \quad \text{and} \quad u_x(x, t) = \mathcal{L}_0^\alpha \hat{u}_x(x, s), \]
\[ T^\alpha_t u_{xx}(x, t) = u(x, t) \quad \text{and} \quad T^\alpha_t u_x(x, t) = -u(x, t), \]
where \( T^\alpha_t \) with \( \alpha > 0 \) is the Riemann-Liouville fractional integral defined by the formula
\[ T^\alpha_t g(t) = \frac{1}{\Gamma(\alpha)} \int_0^t g(\tau) (t - \tau)^{1-\alpha} d\tau, \quad t > 0. \]
Similarly, on the interval \( \Omega_l \), we have \( c_1 = 0 \) from the decay condition (2.8) as \( x \to -\infty \). Hence, the solution is
\[ \hat{u}(x, s) = c_2(s)e^{\lambda_2(s)x} + c_3(s)e^{\lambda_3(s)x}. \]
Taking first and second derivatives of $\hat{u}$ and eliminating $c_2$ and $c_3$, we obtain

$$\hat{u}_{xx}(x, s) - (\lambda_2 + \lambda_3)\hat{u}_x(x, s) + \lambda_2\lambda_3\hat{u}(x, s) = 0.$$  

Using the facts that $\lambda_2 + \lambda_3 = -\lambda_1$ and $\lambda_2\lambda_3 = \lambda_1^2$, we have

$$(2.16) \quad \hat{u}(x, s) + \frac{1}{\lambda_1(s)} \hat{u}_x(x, s) + \frac{1}{\lambda_1^2(s)} \hat{u}_{xx}(x, s) = 0,$$

or equivalently,

$$(2.17) \quad \lambda_1^2(s)\hat{u}(x, s) + \lambda_1(s)\hat{u}_x(x, s) + \hat{u}_{xx}(x, s) = 0.$$  

Applying the inverse Laplace transform to (2.16) and (2.17) leads to

$$(2.23) \quad \hat{u}(x, t) = \frac{1}{2\pi i} \int_{C} e^{st} \hat{u}(x, s) ds,$$

$$(2.27) \quad u(x, t) = \frac{1}{2\pi i} \int_{C} e^{st} u(x, s) ds.$$  

Using (2.24) and (2.28) as exact artificial boundary conditions, the original IVP (1.1) is reduced to the following IBVP on $\Omega_c$:

$$(2.20) \quad \frac{\partial}{\partial t} u(x, t) + u_{xxx}(x, t) = f(x, t), \quad x \in \Omega_c, \ t \in (0, T],$$

$$(2.21) \quad u(x, 0) = u_0(x), \quad x \in \Omega_c,$$

$$(2.22) \quad \frac{\partial}{\partial t} u_{xx}(x, t) = u(x, t),$$

$$(2.23) \quad u_{xx}(x, t) = -u(x, t),$$

$$(2.24) \quad u_{x}(x, t) - u_{xx}(x, t) + u(x, t) = 0.$$  

Similarly, using (2.13) and (2.19) as exact artificial boundary conditions, the original IVP (1.1) is reduced to the following IBVP on $\Omega_c$:

$$(2.25) \quad \frac{\partial}{\partial t} u(x, t) + u_{xxx}(x, t) = f(x, t), \quad x \in \Omega_c, \ t \in (0, T],$$

$$(2.26) \quad u(x, 0) = u_0(x), \quad x \in \Omega_c,$$

$$(2.27) \quad u_{xx}(x, t) = \frac{\partial}{\partial t} u_{x}(x, t),$$

$$(2.28) \quad u_{x}(x, t) = -\frac{\partial}{\partial t} u_{xx}(x, t),$$

$$(2.29) \quad u_{xx}(x, t) - \frac{\partial}{\partial t} u_{x}(x, t) + \frac{\partial}{\partial t} u(x, t) = 0.$$  

3. Stability of the exact absorbing boundary conditions

We now show that both the IBVP (2.20)–(2.24) and the IBVP (2.25)–(2.29) are stable in $L^2$ norm. We begin with some useful lemmas.

**Lemma 3.1** (see [1]). Suppose that $v(t)$ is absolutely continuous on $[0, T]$. Then

$$v(t)\frac{\partial}{\partial t}^\alpha v(t) \geq \frac{1}{2\alpha} \frac{\partial}{\partial t} v^2(t), \quad 0 < \alpha < 1.$$  

**Lemma 3.2** (see [25]). Suppose that $v(t)$ satisfies the inequality

$$v(t) \leq a(t) + b(t) \int_{0}^{t} (t - s)^{\alpha - 1} v(s) ds, \quad \alpha > 0,$$

for $t \in [0, T]$, where $a(t)$ is not decreasing, nonnegative and locally integrable over $[0, T]$, and $b(t)$ is nonnegative and nondecreasing. Then

$$v(t) \leq a(t) E_\alpha(b(t) \Gamma(\alpha)t^\alpha), \quad 0 \leq t \leq T,$$
Lemma 3.3. Suppose that \( y(t) \) is absolutely continuous on \([0, T]\). Then
\[
I_t^y C^\alpha(D^\alpha y)(t) = -\frac{t^{1-\alpha}}{\Gamma(2-\alpha)} y(0) + I_t^\alpha y(t).
\]

Proof. Using the definitions for the Caputo fractional derivative (1.2) and Riemann-Liouville integral (2.15), we obtain
\[
I_t^\alpha C^\alpha(D^\alpha y)(t) = \int_0^t \frac{1}{\Gamma(1-\alpha)} \int_0^\tau \frac{y'(\eta)}{(\tau-\eta)^\alpha} d\tau d\eta.
\]
Exchanging the order of integration and performing integration by parts, we have
\[
I_t^\alpha C^\alpha(D^\alpha y)(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t y'(\eta) \int_\eta^t \frac{1}{(t-\tau)^\alpha} d\tau d\eta
\]
\[
= \frac{1}{\Gamma(1-\alpha)(1-\alpha)} \int_0^t y'(\eta)(t-\eta)^{1-\alpha} d\eta
\]
\[
= -\frac{t^{1-\alpha}}{\Gamma(2-\alpha)} y(0) + \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{y(\eta)}{(t-\eta)^\alpha} d\eta
\]
\[
= -\frac{t^{1-\alpha}}{\Gamma(2-\alpha)} y(0) + I_t^\alpha y(t),
\]
and the proof is complete.

Lemma 3.4. For any \( T > 0 \), suppose \( f \) and \( g \) are smooth functions with \( f(0) = g(0) = 0 \). Then
\begin{align*}
(3.1) & \quad I_t^\alpha(2fI_t^{\frac{\alpha}{2}}f - (I_t^{\frac{\alpha}{2}}f)^2)|_{t=T} \geq 0, \\
(3.2) & \quad I_t^\alpha(2(I_t^{\frac{\alpha}{2}}g - I_t^{\frac{\alpha}{2}}f)g - g^2)|_{t=T} \leq 0.
\end{align*}

Proof. For any \( T > 0 \), we set \( F(t) = f(t) \mathbb{I}_{[0, T]}, G(t) = g(t) \mathbb{I}_{[0, T]}(t) \), where \( \mathbb{I}_{[0, T]}(t) \) is the characteristic function of \([0, T]\). We may then extend the integration domain as follows:
\[
I_t^\alpha(2fI_t^{\frac{\alpha}{2}}f - (I_t^{\frac{\alpha}{2}}f)^2)|_{t=T} = I_t^\alpha(2F I_t^{\frac{\alpha}{2}}F - (I_t^{\frac{\alpha}{2}}F)^2)|_{t=T} = \int_0^{+\infty} (2F I_t^{\frac{\alpha}{2}}F - (I_t^{\frac{\alpha}{2}}F)^2) dt.
\]

We now employ the Plancherel theorem to obtain
\[
\int_0^{+\infty} (2F I_t^{\frac{\alpha}{2}}F - (I_t^{\frac{\alpha}{2}}F)^2) dt
\]
\[
= \frac{1}{2\pi} \int_{-\infty}^{+\infty} (2i\xi) - \xi^\alpha - |\xi|^{-\frac{\alpha}{2}} \left| \hat{F}(i\xi) \right|^2 d\xi
\]
\[
= \frac{1}{2\pi} \int_{0}^{+\infty} \left| (2i\xi) - \xi^\alpha - |\xi|^{-\frac{\alpha}{2}} \right| \hat{F}(i\xi) \left| \hat{F}(i\xi) \right|^2 d\xi
\]
\[
= \frac{1}{2\pi} \int_{0}^{+\infty} 2(2\cos \frac{\pi\alpha}{3} - 1) |\xi|^{-\frac{\alpha}{2}} \left| \hat{F}(i\xi) \right|^2 d\xi
\]
\[
\geq 0,
\]
where we have used the fact that \(1 < 2 \cos \frac{\alpha \pi}{2} < 2\) for \(0 < \alpha < 1\) and (3.1) is proved. In addition, we have
\[
\mathcal{I}_1^\alpha(2I_4^\alpha g - \mathcal{I}_4^\alpha f) = 0
\]
\[
\mathcal{I}_1^\alpha(2I_4^\alpha g - \mathcal{I}_4^\alpha f) = 0
\]

for \(t = T\), where the second equality follows from the differentiation of (2.23). Replacing \(u\) with \(u\), multiplying both sides of (2.20) with 2
\[
\text{Theorem 3.5.} \quad \text{The IBVP (2.20)-(2.24) is } L^2\text{-stable. More precisely, for } 0 < t \leq T, \text{ the following estimate holds}
\]
\[
\int_0^t \|u(\cdot, \tau)\|^2 \, d\tau \leq \mathcal{I}_1^\alpha \left( \frac{t^{1-\alpha}}{\Gamma(2-\alpha)} E_\alpha(t^\alpha) \right) \|u(\cdot, 0)\|^2
\]
\[
+ \mathcal{I}_1^\alpha \left( E_\alpha(t^\alpha) \int_0^t \|f(\cdot, \tau)\|^2 \, d\tau \right).
\]

\textbf{Proof.} Multiplying both sides of (2.20) with 2\(u\), integrating \(x\) from \(x_l\) to \(x_r\), and applying integration by parts, we obtain
\[
\int_{x_l}^{x_r} u_0 D^\alpha_t u \, dx = -2 \int_{x_l}^{x_r} (uu_{xx} - \frac{1}{2} u_x^2) \, dx + \int_{x_l}^{x_r} 2uf \, dx
\]
\[
= (2uu_{xx} - u_x^2)(x_l, t) - (2uu_{xx} - u_x^2)(x_r, t) + \int_{x_l}^{x_r} 2uf \, dx.
\]

The exact ABCs (2.22)-(2.24) can be rewritten as follows:
\[
u(x_l, t) = -I_4^\alpha u_{xx}(x_l, t), \quad u_x(x_l, t) = I_4^\alpha u_{xx}(x_l, t),
\]
\[
u(x_r, t) = -I_4^\alpha u_{xx}(x_r, t) + I_4^\alpha u_x(x_r, t),
\]
where the second equality follows from the differentiation of (2.23). Replacing \(u(x_l, t), u(x_r, t)\) and \(u(x_0, t)\) in (3.4) by the above integrals, we have
\[
\int_{x_l}^{x_r} u_0 D^\alpha_t u \, dx = \left[ 2(I_4^\alpha u_{xx} - I_4^\alpha u_{xxxx})u_{xx} - u_x^2 \right] (x_l, t)
\]
\[
- \left[ 2uu_{xx}I_4^\alpha u_x - (I_4^\alpha u_{xx})^2 \right] (x_r, t) + \int_{x_l}^{x_r} f \, u \, dx \, d\tau.
\]

Integrating both sides of the above equation over \([0, t]\) and then applying the lemmas 3.1 and 3.4, we obtain
\[
\mathcal{I}_1^\alpha \int_{x_l}^{x_r} u^2 \, dx \leq 2 \int_0^t \int_{x_l}^{x_r} f \, u \, dx \, d\tau.
\]
The applications of the Lemma 3.3 and the Cauchy-Schwartz inequality yield
\begin{equation}
I_t^{1-\alpha} \int_{x_1}^{x_2} u^2 dx \leq \frac{t^{1-\alpha}}{\Gamma(2-\alpha)} \int_{x_1}^{x_2} u(x,0)^2 dx
\end{equation}
\begin{equation}
+ \int_0^t \int_{x_1}^{x_2} u^2 dx dr + \int_0^t \int_{x_1}^{x_2} f^2 dx dr.
\end{equation}
Let \( v(t) := I_t^{1-\alpha} \left( \int_{x_1}^{x_2} u^2 dx \right) \) and \( a(t) := \frac{t^{1-\alpha}}{\Gamma(2-\alpha)} \int_{x_1}^{x_2} u_0^2(x) dx + \int_0^t \int_{x_1}^{x_2} f^2(x,t) dx dr. \)
Using the property of the Riemann-Liouville integral, we may rewrite \( \int_0^t \int_{x_1}^{x_2} u^2 dx dr \) as follows:
\[ I_t^{1-\alpha} \left( \int_{x_1}^{x_2} u^2 dx \right) = I_t^{1-\alpha} \left( \int_{x_1}^{x_2} u^2 dx \right) = I_t^{1-\alpha} \left( \int_{x_1}^{x_2} u^2 dx \right) = I_t^{1-\alpha} v(t). \]
Hence, (3.5) can be rewritten as follows:
\begin{equation}
v(t) \leq a(t) + I_t^{1-\alpha} v(t) = a(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} v(\tau) d\tau.
\end{equation}
Obviously \( a(t) \) is non-decreasing. A direct application of the lemma 3.2 to (3.7) with \( b(t) = \frac{1}{\Gamma(\alpha)} \) leads to
\begin{equation}
v(t) \leq a(t) E_{\alpha}(t^\alpha).
\end{equation}
Applying the operator \( I_t^{\alpha} \) on both sides of (3.8) leads to the desired result (3.3). \( \square \)

**Lemma 3.6.** For any \( T > 0 \), suppose \( f \) and \( g \) are smooth functions with \( f(0) = g(0) = 0 \). Then
\[ I_t^{1}(2f_0^C D_t^\alpha f - (\mathcal{C} D_t^\alpha f))^2|_{t=T} \geq 0, \]
\[ I_t^{1}(2\mathcal{C} D_t^\alpha g - (\mathcal{C} D_t^\alpha f))^2|_{t=T} \leq 0. \]

**Proof.** Let \( F(t) = f(t)1_{[0,T]}, G(t) = g(t)1_{[0,T]}(t) \). Then we have
\[ I_t^{1}(2f_0^C D_t^\alpha f - (\mathcal{C} D_t^\alpha f))^2|_{t=T} = \int_0^{+\infty} (2f_0^C D_t^\alpha f - (\mathcal{C} D_t^\alpha f))^2 dt. \]
Applying the Plancherel theorem yields
\[
\int_0^{+\infty} (2f_0^C D_t^\alpha f - (\mathcal{C} D_t^\alpha f))^2 dt
\leq \frac{1}{2\pi} \int_{-\infty}^{+\infty} (2(i\xi)^{\alpha} - |\xi|^{\alpha}) \left| \hat{F}(i\xi) \right|^2 d\xi
\leq \frac{1}{2\pi} \int_{0}^{+\infty} 2(2\cos \frac{\pi \alpha}{\beta} - 1) |\xi|^{\alpha} \left| \hat{F}(i\xi) \right|^2 d\xi \geq 0.
\]
In addition, we have
\[ I_t^{1}(2\mathcal{C} D_t^\alpha g - (\mathcal{C} D_t^\alpha f))^2|_{t=T}
= \int_0^{+\infty} \left[ 2(\mathcal{C} D_t^\alpha G - (\mathcal{C} D_t^\alpha f) F - G^2 \right] dt
\leq \int_0^{+\infty} \left[ 2f_0^C D_t^\alpha G - (\mathcal{C} D_t^\alpha f)^2 - G^2 \right] dt
\]
For a given integer $N$, let $\{ t_k \}_{k=0}^{N}$ be a partition of $[0,T]$ with $t_k = k\Delta t$ and $\Delta t_k = t_k - t_{k-1} = \frac{T}{N}$. For a mesh function $U = \{ U^k = u(t_k) \}_{0 \leq k \leq N}$, we approximate $u(t)$ via a piecewise linear polynomial with $\{ t_k \}_{k=0}^{N}$ as the interpolation nodes and obtain the following $L^1$ discretization for approximating the Caputo fractional derivative:

$$ D_t^\alpha U^k = \frac{\Delta t^{-\alpha}}{\Gamma(2-\alpha)} \left[ a_0^{(\alpha)} U^k - \sum_{l=1}^{k-1} (a_{k-l-1}^{(\alpha)} - a_{k-l}^{(\alpha)}) U^l - a_{k-1}^{(\alpha)} U^0 \right] $$

for $1 \leq k \leq N$. Here $a_i^{(\alpha)} = (l+1)^{1-\alpha} - l^{1-\alpha}$. The accuracy of the $L^1$ discretization is given as follows.
Lemma 4.1 (see [28]). Suppose that \( u(t) \in C^2[0,T] \). Let
\[
R(u(t_k)) := \frac{C}{\alpha} D_t^\alpha u(t) |_{t=t_k} - D_t^\alpha U^k, \quad 1 \leq k \leq N.
\]
Then
\[
|R^k u| \leq \frac{1}{1(2-\alpha)} \left[ \frac{1}{12} + \frac{2^{2-\alpha}}{2-\alpha} - (1 + 2^{-\alpha}) \right] \max_{0 \leq t \leq t_k} |u''(t)| \Delta t^{2-\alpha}.
\]

Next, we discuss the discretization of spatial derivatives and boundary conditions. For a given integer \( M \), let \( \{x_i\}_{i=0}^M \) be an equispaced partition of \((x_l,x_r)\) with \( x_i = x_l + ih \) and \( h = (x_r - x_l)/M \). Denote \( U_i^n = u(x_i, t_n) \), \( f_i^n = f(x_i, t_n) \), and
\[
(4.8) \quad \delta_x U_i^k = \frac{1}{h} (U_{i+1}^k - U_i^k), \quad U_{i+\frac{1}{2}}^k = \frac{1}{2} (U_i^k + U_{i+1}^k), \quad 0 \leq i \leq M - 1,
\]
\[
(4.9) \quad \delta_x^2 U_i^k = \frac{1}{h^2} (\delta_x U_{i+\frac{1}{2}}^k - \delta_x U_{i-\frac{1}{2}}^k) = \frac{1}{h^2} (U_{i+1}^k - 2U_i^k + U_{i-1}^k), \quad 1 \leq i \leq M - 1.
\]
We also denote \( V_i^n = v(x_i, t_n) \) and define \( \delta_x V_i^k \), \( V_{i+\frac{1}{2}}^k \), and \( \delta_x^2 V_i^k \) similarly as in (4.8) and (4.9). We apply these second-order central differences to approximate the spatial derivatives in (4.1) and (4.2) at interior nodes. As for the boundary nodes, we note that the Taylor theorem leads to the following lemma.

Lemma 4.2 (see [10]). Suppose that \( y(x) \in C^3(\Omega_e) \). Then
\[
y''(x_0) - \frac{2}{h} \left[ \frac{y(x_1) - y(x_0)}{h} - y'(x_0) \right] = -\frac{h}{3} y'''(x_0 + \theta_1 h), \quad \theta_1 \in (0,1),
\]
\[
y''(x_M) - \frac{2}{h} \left[ \frac{y'(x_M) - y(x_M-1)}{h} \right] = -\frac{h}{3} y'''(x_M - \theta_2 h), \quad \theta_2 \in (0,1).
\]

We now discuss the discretization at the right end point \( x_M \). The boundary condition (4.5) can be discretized via the \( L^1 \) discretization of the Caputo derivative directly since it does not contain any spatial derivative. To obtain another discrete condition at \( x_M \), we discretize the Caputo derivatives in the PDE (4.1) and the boundary condition (4.4) to obtain
\[
\begin{align*}
D_t^\alpha U_M^k + v_{xx}(x_M, t_k) &\approx f_M^k, \\
v_x(x_M, t_k) &\approx D_{t+\frac{1}{2}}^\alpha U_M^k.
\end{align*}
\]
The second equation in Lemma 4.2 leads to
\[
v_{xx}(x_M, t_k) \approx \frac{2}{h} \left[ v_x(x_M, t_k) - \delta_x V_{M-\frac{1}{2}}^k \right].
\]
Combining the above three equations to eliminate \( v_x \) and \( v_{xx} \), we have
\[
(4.10) \quad D_t^\alpha U_M^k + \frac{2}{h} \left( D_{t+\frac{1}{2}}^\alpha U_M^k - \delta_x V_{M-\frac{1}{2}}^k \right) \approx f_M^k.
\]
Similarly, we can derive the discrete condition at \( x_0 \). To summarize, we have the following approximation for the IBVP (4.1)–(4.6).
\[
\begin{align*}
(4.11) \quad &\quad D_t^\alpha U_i^k + \delta_x^2 V_i^k = f_i^k + T_i^k, \quad 1 \leq i \leq M - 1, \quad 1 \leq k \leq N, \\
(4.12) \quad &\quad V_{i+\frac{1}{2}}^k = \delta_x U_{i+\frac{1}{2}}^k + Q_{i+\frac{1}{2}}^k, \quad 0 \leq i \leq M - 1, \quad 1 \leq k \leq N, \\
(4.13) \quad &\quad D_{t+\frac{1}{2}}^\alpha U_M^k = V_M^k + R_M^k, \quad 1 \leq k \leq N, \\
(4.14) \quad &\quad D_t^\alpha U_M^k + \frac{2}{h} \left( D_{t+\frac{1}{2}}^\alpha U_M^k - \delta_x V_{M-\frac{1}{2}}^k \right) = f_M^k + T_M^k, \quad 1 \leq k \leq N,
\end{align*}
\]
Lemma 4.2. combines the PDE (4.1), the boundary condition (4.6), and the first equation in (4.10); and (4.15) follows from a similar derivation for the left end point (4.19).

That is, (4.11) follows from the L\textsuperscript{1} scheme for solving the IBVP (2.25)–(2.29): central difference discretization of \( v \) have

\[
\begin{align*}
T_1^k &= \left[ D_t^\alpha U_0^k - \frac{\alpha}{\Gamma(\alpha)} \frac{D_0^\alpha u(x_i, t_k)}{\alpha} \right] + \left[ \frac{\alpha}{2} V_i^k - v(x_i, t_k) \right], \quad 1 \leq i \leq M - 1, 1 \leq k \leq N, \\
T_0^k &= \left[ D_t^\alpha U_0^k - \frac{\alpha}{\Gamma(\alpha)} \frac{D_0^\alpha u(x_0, t_k)}{\alpha} \right] + \left[ \frac{\alpha}{2} V_0^k - v(x_0, t_k) \right], \\
T_k^k &= \left[ D_t^\alpha U_M^k - \frac{\alpha}{\Gamma(\alpha)} \frac{D_M^\alpha u(x_M, t_k)}{\alpha} \right] + \left[ \frac{\alpha}{2} V_M^k - v(x_M, t_k) \right], \quad 1 \leq k \leq N, \\
R_k^k &= \left[ D_t^\alpha U_M^k - \frac{\alpha}{\Gamma(\alpha)} \frac{D_M^\alpha u(x_M, t_k)}{\alpha} \right], \quad 1 \leq k \leq N. \\
Q_k^k &= u_v(x_{i+\frac{1}{2}}, t_k) - \delta_x U_k^k + V_k^k - v(x_{i+\frac{1}{2}}, t_k), \quad 0 \leq i \leq M - 1, 1 \leq k \leq N.
\end{align*}
\]

By Lemma 4.1 and the Taylor theorem, for \( u \in C^{5,2}_x(\Omega \times [0, t_k]) \) (1 \( \leq k \leq N \) we have

\[
\begin{align*}
|T_1^k| &\leq c(\Delta t^{2-\alpha}) \Delta x^2 + \Delta t^{2-\alpha} / \Delta x^2, \quad 1 \leq i \leq M - 1, 1 \leq k \leq N, \\
|T_0^k| &\leq c(\Delta t^{2-\alpha}) \Delta x^2 + \Delta t^{2-\alpha} / \Delta x^2, \quad 1 \leq k \leq N, \\
|T_k^k| &\leq c(\Delta t^{2-\alpha}) \Delta x^2 + \Delta t^{2-\alpha} / \Delta x^2, \quad 1 \leq k \leq N, \\
|R_k^k| &\leq c(\Delta t^{2-\alpha}) \Delta x^2, \quad 1 \leq k \leq N, \\
|Q_k^k| &\leq c\Delta t^2 \Delta x^2, \quad 0 \leq i \leq M - 1, 1 \leq k \leq N.
\end{align*}
\]

Finally, omitting the truncation terms \( T_k^k(0 \leq i \leq M, 1 \leq k \leq N), R_k^k(1 \leq k \leq N) \) and \( Q_k^k(1 \leq i \leq M - 1, 1 \leq k \leq N) \), we obtain the following finite difference scheme for solving the IBVP (2.25)–(2.29):

\[
\begin{align*}
D_t^\alpha U_k^k + \delta_x^2 V_i^k &= f_i^k, \quad 1 \leq i \leq M - 1, 1 \leq k \leq N, \\
D_t^\alpha U_0^k + \delta_x^2 V_0^k &= f_0^k, \quad 1 \leq k \leq N, \\
D_t^\alpha U_M^k + \delta_x^2 V_M^k &= f_M^k, \quad 1 \leq k \leq N, \\
V_k^k &= \delta_x U_k^k, \quad 0 \leq i \leq M - 1, 1 \leq k \leq N.
\end{align*}
\]

Remark 4.3. As compared with the direct FD scheme mentioned at the beginning of this subsection, our FD scheme has second order accuracy for \( v \) at interior points.
and first order accuracy at boundary points. Since \( v = u_x \), this implies third order accuracy for \( u \) at interior points and second order accuracy at boundary points. Thus, our FD scheme has higher order of accuracy and better stability property.

4.2. Stability Analysis of the Finite Difference Scheme. We now show that the FD scheme (4.21)–(4.25) is unconditionally stable. We first recall that the Riemann-Liouville fractional derivative is defined by the formula (see, for example, [24])

\[
\frac{\mathcal{D}^\alpha_t}{\mathcal{T}^{\alpha}} u(x, t) = \frac{1}{\Gamma(1 - \alpha)} \frac{d}{dt} \int_0^t \frac{u(x, s)}{(t-s)^\alpha} \, ds, \quad 0 < \alpha < 1.
\]

For any \( 1 \leq n \leq N \), suppose that the piecewise constant function of a mesh vector \((0, u^1, u^2, \cdots, u^n)\) with zero initial value is defined by the formula

\[
u(t) := \begin{cases} u^k, & t \in [t_{k-1}, t_k), \quad 1 \leq k \leq n, \\ 0, & t \notin [t_0, t_n). \end{cases}
\]

The following lemma connects the Riemann-Liouville fractional derivative of a piecewise constant function with the \( L^1 \) discretization (4.7) of the corresponding mesh vector with zero initial data.

**Lemma 4.4.** Let \( u(t) \) and \( v(t) \) be the piecewise constant functions of the mesh vectors \((0, u^1, u^2, \cdots, u^n)\) and \((0, v^1, v^2, \cdots, v^n)\) with zero initial values, respectively. Then \( u, v \in H^\alpha_{\text{loc}}(\mathbb{R}^+) \). Furthermore,

\[
\mathcal{I}_t^1 \left( u(t) \mathcal{D}^\alpha_t v(t) \right) \big|_{t=t_n} = \Delta t \sum_{k=1}^{n} u^k \mathcal{D}^\alpha_t v^k,
\]

and

\[
\mathcal{I}_t^1 \left( u(t) \mathcal{D}^\alpha_t v(t) \right) \big|_{t=t_n} \geq \Delta t \sum_{k=1}^{n} (\mathcal{D}^\alpha_t u^k)^2.
\]

**Proof.** The identity (4.26) can be proved via straightforward calculation. The details are presented in [29].

As for the inequality (4.27), we first apply Jensen’s inequality to obtain

\[
\mathcal{I}_t^1 \left( \mathcal{D}^\alpha_t u(t) \right)^2 \big|_{t=t_n} = \frac{1}{\Gamma(1 - \alpha)^2} \sum_{k=1}^{n} \int_{t_{k-1}}^{t_k} \left( \frac{d}{dt} \int_0^t \frac{u(s)}{(t-s)^\alpha} \, ds \right)^2 \, dt \geq \frac{\Delta t^{-1}}{\Gamma(1 - \alpha)^2} \sum_{k=1}^{n} \left( \int_{t_{k-1}}^{t_k} \frac{d}{dt} \int_0^t \frac{u(s)}{(t-s)^\alpha} \, ds \right)^2 dt.
\]

We then observe that

\[
\frac{1}{\Gamma(1 - \alpha)} \int_{t_{k-1}}^{t_k} \frac{d}{dt} \int_0^t \frac{u(s)}{(t-s)^\alpha} \, ds \, dt = \mathcal{I}_t^1 \left( \mathcal{D}^\alpha_t u(t) \right) \big|_{t=t_k} - \mathcal{I}_t^1 \left( \mathcal{D}^\alpha_t u(t) \right) \big|_{t=t_{k-1}} = \Delta t \sum_{l=1}^{k} \mathcal{D}^\alpha_t u^l - \Delta t \sum_{l=1}^{k-1} \mathcal{D}^\alpha_t u^l = \Delta t \sum_{l=1}^{k} \mathcal{D}^\alpha_t u^l,
\]

and

\[
\Delta t \sum_{l=1}^{k} \mathcal{D}^\alpha_t u^l = \Delta t \sum_{l=1}^{k} \mathcal{D}^\alpha_t u^l.
\]
For any mesh functions $f, g \in H^{\infty}(0, T)$. Then

\begin{align}
\mathcal{I}_t^4(2f \mathcal{D}^\alpha_t f - (\mathcal{D}^\alpha_t f)^2)|_{t=0} \geq 0, \\
\mathcal{I}_t^4(2(\mathcal{D}^\alpha_t g - \mathcal{D}^\alpha_t f)f - g^2)|_{t=0} \leq 0.
\end{align}

The next lemma can be found in [10].

**Lemma 4.6.** For any mesh functions $g = \{g_k|0 \leq k \leq N\}$ defined on $\Omega_t = \{t_k|0 \leq k \leq N\}$, the following inequality holds:

\begin{align}
\Delta t \sum_{k=1}^{n} (\mathcal{D}^\alpha_t g_k)^2 \geq \frac{t_n^{1-\alpha}}{2\Gamma(1-\alpha)} \Delta t \sum_{k=1}^{n} (g_k^2) - \frac{t_n^{1-\alpha}}{2\Gamma(2-\alpha)} (g_0^2), \quad 1 \leq n \leq N.
\end{align}

Proof. We multiply both sides of (4.21) by $\frac{1}{2}U_0^k$, and then sum up for $i$ from 1 to $M - 1$; we then multiply both sides of (4.22) and (4.23) by $\frac{1}{2}U_0^k$ and $\frac{1}{2}U_M^k$, respectively; finally we add up these equations to obtain

\begin{align}
(D_t^\alpha U_k, U_k) + \sum_{i=1}^{M-1} h\delta_x^2 V_i^k \cdot U_i^k + \delta_x V_i^k \cdot U_0^k - \delta_x V_{M-\frac{1}{2}} \cdot U_M^k \\
\quad - D_t^\frac{2+\alpha}{2} U_0^k \cdot U_0^k + D_t^\frac{2+\alpha}{2} U_M^k \cdot U_M^k = (f_k, U_k).
\end{align}

We then use (4.25) to simplify the second term to the fourth term of the left side in (4.34),

\begin{align}
\sum_{i=1}^{M-1} h\delta_x^2 V_i^k \cdot U_i^k + \delta_x V_i^k \cdot U_0^k - \delta_x V_{M-\frac{1}{2}} \cdot U_M^k \\
= \sum_{i=1}^{M-1} \left( \delta_x V_i^k \cdot U_i^k \right) = \sum_{i=1}^{M-1} (V_i^k - V_i^k) \delta_x U_i^k.
\end{align}
Substituting (4.35) and (4.24) into (4.34), we obtain
\begin{equation}
(D_t^\alpha U^k, U^k) + \frac{1}{2}(V_0^k)^2 - D_t^\alpha V_0^k \cdot U_0^k + D_t^\alpha U_0^k \cdot U_M^k
- \frac{1}{2} \left( D_t^\alpha U_M^k \right)^2 + D_t^\alpha U_M^k \cdot U_M^k = (f^k, U^k).
\end{equation}
Multiplying \(\Delta t\) on both sides of the (4.36), summing up for \(k\) from 1 to \(n\), and applying the identity (4.26), we obtain
\begin{equation}
\sum_{k=1}^n \Delta t (D_t^\alpha U^k, U^k) + \mathcal{I}_t^1 \left( \frac{1}{2} (V_0^k)^2 - (R_t^\alpha V_0^k) U_0^k + (R_t^\alpha U_0^k) U_0^k \right) |_{t=t_n}
+ \mathcal{I}_t^1 \left( \frac{1}{2} \left( R_t^\alpha U_M^k \right)^2 \right) |_{t=t_n} = \sum_{k=1}^n \Delta t(f^k, U^k).
\end{equation}
Here \(U_0, V_0, U_M\) are piecewise constant functions of the mesh vectors \((U_0^0, U_0^1, \ldots, U_0^n), (V_0^0, V_0^1, \ldots, V_0^n), (U_M^0, U_M^1, \ldots, U_M^n)\), respectively. And we have used the fact that \(U_M^0 = U_0^0 = V_0^0 = 0\) by the assumption that the initial data are compactly supported.

By the inequality (4.31), the second term on the left side of (4.37) is nonnegative; by the inequality (4.30), the third term on the left side of (4.37) is nonnegative. Thus,
\begin{equation}
\sum_{k=1}^n \Delta t (D_t^\alpha U^k, U^k) \leq \sum_{k=1}^n \Delta t(f^k, U^k).
\end{equation}
Applying the inequality (4.32) to estimate the left side of (4.38), we have
\begin{equation}
\frac{t_n^\alpha}{2\Gamma(1-\alpha)} \Delta t \sum_{k=1}^n \|U^k\|^2 \leq \frac{t_n^{\alpha-\alpha}}{2\Gamma(2-\alpha)} \|U^0\|^2 + \sum_{k=1}^n \Delta t(f^k, U^k).
\end{equation}
Now the Cauchy-Schwarz inequality yields
\begin{equation}
(f^k, U^k) \leq \|f^k\| \cdot \|U^k\|
\leq \frac{t_n^\alpha}{4\Gamma(1-\alpha)} \|f^k\|^2 + \frac{t_n^{\alpha-\alpha}}{4\Gamma(1-\alpha)} \|U^k\|^2.
\end{equation}
Finally, combining (4.39) with (4.40) and simplifying the resulting expression, we obtain (4.33).

4.3. Acceleration of the Finite Difference Scheme. The direct implementation of the FD scheme (4.21)–(4.25) is very expensive in both the computational and storage cost. This is because the Caputo derivative is a nonlocal operator and its \(L1\) discretization (or any consistent discretization) contains a summation that involves all values of the solution up to the current time. Hence, one needs to store all previous solution values and the total cost of evaluating the \(L1\) discretization at each spatial point is \(O(N^2)\) with \(N\) the total number of time steps.

We apply the fast algorithm for the evaluation of the Caputo fractional derivative in [15] to reduce the computational and storage cost. In order to make the paper more or less self-contained, we present a short summary of the algorithm in [15].
here. We first split the fractional Caputo derivative into a sum of local part and history part

\[
\frac{D^\alpha}{dt^\alpha} u(t)|_{t=t_k} = \frac{1}{\Gamma(1-\alpha)} \int_{t_{k-1}}^{t_k} u'(s)ds + \frac{1}{\Gamma(1-\alpha)} \int_0^{t_k-1} u'(s)ds (t_k-s)^\alpha
\]

\[
:= C_l(t_k) + C_h(t_k).
\]

(4.41)

For the local part \(C_l(t_k)\), we apply the \(L1\) approximation in the form

\[
C_l(t_k) \approx \frac{u(t_k) - u(t_{k-1})}{\Delta t_k \Gamma(1-\alpha)} \int_{t_{k-1}}^{t_k} \frac{1}{(t_{k}-s)\alpha}ds = \frac{u(t_k) - u(t_{k-1})}{\Delta t_k \Gamma(2-\alpha)}.
\]

(4.42)

For the history part \(C_h(t_k)\), we first perform integration by parts to obtain

\[
C_h(t_k) = \frac{1}{\Gamma(1-\alpha)} \left[ \frac{u(t_{k-1})}{\Delta t_k^\alpha} - \frac{u(t_0)}{t_0^\alpha} - \alpha \int_0^{t_{k-1}} \frac{u(s)ds}{(t_{k}-s)^{1+\alpha}} \right].
\]

(4.43)

Obviously, the most work is on the evaluation of the convolution integral with the kernel \(\frac{1}{t_{k}-s}\) on the right side of (4.43). It has been shown in [15] that there exist a sufficient sum-of-exponentials approximation for \(\frac{1}{t_{k}-s}\), for any given time interval \([\Delta t_k, T]\) with a prescribed absolute error \(\varepsilon\). To be more precise, there exist positive real numbers \(s_i\) and \(w_i\) \((i = 1, \ldots, N_{\text{exp}})\) such that

\[
\left| \frac{1}{t_{k}+\alpha} - \sum_{i=1}^{N_{\text{exp}}} \omega_i e^{-s_i t} \right| \leq \varepsilon, \quad t \in [\Delta t, T];
\]

(4.44)

and the number of exponentials \(N_{\text{exp}}\) needed is of the order (see Table 1 for an example of the actual number of exponentials needed)

\[
O \left( \log \frac{1}{\varepsilon} \left( \log \log \frac{1}{\varepsilon} + \log \frac{T}{\Delta t} \right) \right) + O \left( \log \frac{1}{\Delta t} \left( \log \log \frac{1}{\varepsilon} + \log \frac{1}{\Delta t} \right) \right).
\]

That is, for fixed precision \(\varepsilon\), we have \(N_{\text{exp}} = O(\log N)\) for \(T \geq 1\) or \(N_{\text{exp}} = O(\log^2 N)\) for \(T \approx 1\) assuming that \(N \approx \frac{T}{\Delta t}\).

**Table 1.** Number of exponentials \(N_{\text{exp}}\) needed to approximate \(t^{-1-\alpha}\) with fixed \(\Delta t = 10^{-3}\) for \(\alpha = 0.5\).

<table>
<thead>
<tr>
<th>(\varepsilon)</th>
<th>(\Delta t)</th>
<th>(10^3)</th>
<th>(10^4)</th>
<th>(10^5)</th>
<th>(10^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10^{-4})</td>
<td>28</td>
<td>30</td>
<td>35</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>(10^{-6})</td>
<td>42</td>
<td>47</td>
<td>47</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>(10^{-5})</td>
<td>49</td>
<td>55</td>
<td>64</td>
<td>67</td>
<td></td>
</tr>
</tbody>
</table>

Replacing the convolution kernel \(\frac{1}{t_{k}-s}\) in (4.43) by its sum-of-exponentials approximation in (4.44), we obtain

\[
C_h(t_k) \approx \frac{1}{\Gamma(1-\alpha)} \left[ \frac{u(t_{k-1})}{\Delta t_k^\alpha} - \frac{u(t_0)}{t_0^\alpha} - \alpha \sum_{i=1}^{N_{\text{exp}}} \omega_i U_{\text{hist},i}(t_k) \right],
\]

(4.45)

where \(U_{\text{hist},i}(t_k)\) is defined by

\[
U_{\text{hist},i}(t_k) = \int_0^{t_{k-1}} e^{-(t_{k-1})s_i} u(\tau) d\tau.
\]

(4.46)
The reduction of the computational and storage cost relies on the fact that $U_{\text{hist},i}(t_k)$ satisfies a simple recurrence relation

\begin{equation}
U_{\text{hist},i}(t_k) = e^{-s_i \Delta t_k} U_{\text{hist},i}(t_{k-1}) + \int_{t_{k-2}}^{t_{k-1}} e^{-s_i (t_k - \tau)} u(\tau) d\tau
\end{equation}

with $U_{\text{hist},i}(t_0) = 0$ for $i = 1, \cdots, N_{\exp}$. The integral on the right hand side of (4.47) can be calculated by a linear approximation of $u$ followed with a product integration. That is,

\begin{equation}
\int_{t_{k-2}}^{t_{k-1}} e^{-s_i (t_k - \tau)} u(\tau) d\tau \approx \frac{e^{-s_i \Delta t_k}}{s_i^2 \Delta t_{k-1}} \left[ (e^{-s_i \Delta t_{k-1}} - 1 + s_i \Delta t_{k-1}) U_{k-1}^i - (1 - e^{-s_i \Delta t_{k-1}} - e^{-s_i \Delta t_{k-1}} s_i \Delta t_{k-1}) U_{k-2}^i \right].
\end{equation}

Thus, one only needs $O(1)$ cost to compute $U_{\text{hist},i}(t_k)$ at each step as $U_{\text{hist},i}(t_{k-1})$ is already computed and stored. As there are $N_{\exp}$ history modes altogether, the cost of evaluating the Caputo derivative at each time step is $O(N_{\exp})$. That is, a reduction from $O(N)$ to $O(\log N)$ or $O(\log^2 N)$.

To summarize, the fast evaluation of the Caputo derivative can be implemented by the formula

\begin{equation}
C_0^{\alpha} D_t^\alpha U^k \approx \frac{U^k - U^{k-1}}{\Delta t_k^\alpha (2 - \alpha)} + \frac{1}{\Gamma(1 - \alpha)} \left[ \frac{U^{k-1}}{\Delta t_k^\alpha} - \frac{U_0}{\Gamma(1 - \alpha)} - \alpha \sum_{i=1}^{N_{\exp}} \omega_i U_{\text{hist},i}(t_k) \right]
= \tilde{D}_t^\alpha U^k, \quad \text{for } k > 0,
\end{equation}

where $U_{\text{hist},i}(t_k)$ is evaluated via (4.47) and (4.48). Finally, replacing the $L1$ discretization $D_t^\alpha U^k$ by its fast version $\tilde{D}_t^\alpha U^k$ in the FD scheme (4.21)–(4.25), we obtain an accelerated FD scheme as follows.

\begin{align}
FD_t^\alpha U_i^k + s_i^2 V_i^k &= f_i^k, \quad 1 \leq i \leq M - 1, \ 1 \leq k \leq N, \\
FD_t^\alpha U_0^k + \frac{2}{h} \left[ \delta_x V_0^k - FD_t^\alpha V_0^k + FD_t^\alpha U_0^k \right] &= f_0^k, \quad 1 \leq k \leq N, \\
FD_t^\alpha U_M^k + \frac{2}{h} \left( FD_t^\alpha U_M^k - \delta_x V_M^k \right) &= f_M^k, \quad 1 \leq k \leq N, \\
FD_t^\alpha U_M^{k+1} &= V_M^k, \quad 1 \leq k \leq N, \\
V_{i+\frac{1}{2}}^k &= \delta_x U_{i+\frac{1}{2}}^k, \quad 0 \leq i \leq M - 1, \ 1 \leq k \leq N.
\end{align}

Remark 4.8. The fast algorithm relies on the fact that the convolution with the exponential kernel can be evaluated in linear time since it is equivalent to solving an ordinary differential equation (ODE). Thus it can be applied to adaptive time marching scheme as well. The only penalty here is that the number of exponentials will increase slightly due to the decrease of the smallest time step.

Remark 4.9. Another fast algorithm for the evaluation of the fractional derivative has been proposed in [4], where the compression is carried out in the Laplace domain, and the convolution with an exponential kernel is computed by solving the equivalent ODE with some one-step A-stable scheme.

Remark 4.10. Both the $L1$ discretization $D_t^\alpha U^k$ and the fast algorithm $\tilde{D}_t^\alpha U^k$ approximate $u(t)$ by a piecewise polynomial and then apply the product integration. Therefore, the difference is very minor and can in fact be made arbitrarily small if
the prescribed precision $\varepsilon$ is set to a small number, say, close to machine precision. In our numerical experiments, we have not observed any significant differences between the FD scheme (4.21)–(4.25) and its fast version (4.50)–(4.54) in both accuracy and stability.

5. Numerical examples

We now report results of numerical experiments which demonstrate the effectiveness of our exact ABCs and offer quantitative features of the finite difference orders, we define the error norm and convergence order by the formulas

$$E(h, \Delta t) = \frac{\sum_{k=1}^{m} \Delta t \|e^k\|^2}{2}, \quad r_t = \log_2 \frac{E(h, \Delta t)}{E(h, \Delta t/2)}, \quad r_s = \log_2 \frac{E(h, \Delta t)}{E(h/2, \Delta t)}.$$ 

In order to obtain smooth reference solutions in example 5.1, we set the right-hand side of the first equation in (1.1) based on the given exact smooth solutions. As we will see, both schemes (4.21)-(4.25) and (4.50)-(4.54) have the same convergence order $h^4 + \Delta t^{2-\alpha}$ due to the negligible influence of the SOE approximation error $\varepsilon$ on the accuracy of the fast algorithm. But the fast algorithm is much faster than the direct method even when $N$ is of moderate size. We will illustrate the effectiveness of ABCs. On the one hand, no reflection wave from artificial boundaries can be seen from the evolution graph of the solutions. On the other hand, we also check the energy defined by $E(t) = \int_{\Omega} u^2(x, t)dx$ on the bounded domain to verify the effectiveness of our ABCs, which can be numerically calculated via the trapezoidal rule.

**Example 5.1.**

We consider the exact solution of the form $u(x, t) = \exp(-x^2)t^2$ for the problem (1.1), which leads to the source function $f(x, t) = \frac{2}{(3-\alpha)t^{2-\alpha}} \exp(-x^2) + t^2 \exp(-x^2)(12x - 8x^3)$. In our computation, we set the computational domain $\Omega = (-6, 6)$, final time $T = 1$ and the tolerance precision $\varepsilon = 10^{-12}$ for the SOE approximation.

Fixing $h = 10^{-3}$ and varying $\Delta t$ from $\frac{1}{50}$ to $\frac{1}{150}$. Table 2 illustrates the errors $E(h, \Delta t)$ and the convergence order of $\Delta t^{2-\alpha}$ for both schemes (4.21)-(4.25) and (4.50)-(4.54) in cases of $\alpha = 0.2, 0.4, 0.6$ and 0.8. Next, fixing $\Delta t = 10^{-4}$ and varying $h$ from $\frac{1}{50}$ to $\frac{1}{150}$, Table 3 illustrates the errors $E(h, \Delta t)$ and the convergence order of $h^4$ in space for both methods in cases of $\alpha = 0.2, 0.4, 0.6$ and 0.8.

Tables 2 and 3 show that both the fast algorithm and the direct method produce the same overall convergence order of $h^4 + \Delta t^{2-\alpha}$. We plot the difference of solutions of these two schemes for $\varepsilon = 10^{-7}, 10^{-8}, 10^{-9}$, and $10^{-10}$, respectively. As we can see in Fig. 1, the difference in the solutions of the two schemes becomes smaller and smaller when $\varepsilon$ gets smaller and smaller.

We now check the computational cost of both methods. For this, we calculate the CPU time for $N = 2 \times 10^4, 4 \times 10^4, \cdots, 1.2 \times 10^5$ with $\alpha = 0.5, M = 30$ fixed. Fig. 2 clearly shows the $O(N^2)$ complexity of the direct method and the $O(N)$ complexity of the fast method.
Table 2. Example 1: The error and convergence order in time for fast and direct schemes, the spatial mesh size is fixed at $h = 10^{-3}$.

<table>
<thead>
<tr>
<th>Δt</th>
<th>Fast scheme</th>
<th>Direct scheme</th>
<th>Fast scheme</th>
<th>Direct scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E(h, \Delta t)$</td>
<td>$r_1$</td>
<td>$E(h, \Delta t)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>$\alpha = 0.2$</td>
<td>6.5385109e-04</td>
<td>1.71</td>
<td>6.5385109e-04</td>
<td>1.71</td>
</tr>
<tr>
<td>1/20</td>
<td>2.0084305e-04</td>
<td>1.72</td>
<td>2.0084305e-04</td>
<td>1.72</td>
</tr>
<tr>
<td>1/40</td>
<td>6.0979782e-05</td>
<td>1.73</td>
<td>6.0979782e-05</td>
<td>1.73</td>
</tr>
<tr>
<td>1/80</td>
<td>1.8360664e-05</td>
<td>1.74</td>
<td>1.8360664e-05</td>
<td>1.74</td>
</tr>
</tbody>
</table>

Table 3. Example 1: The error, convergence order in space and CPU time for fast and direct schemes, the time step size is fixed at $\Delta t = 10^{-4}$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>Fast scheme</th>
<th>Direct scheme</th>
<th>Fast scheme</th>
<th>Direct scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E(h, \Delta t)$</td>
<td>$r_1$</td>
<td>$E(h, \Delta t)$</td>
<td>$r_1$</td>
</tr>
<tr>
<td>1/36</td>
<td>8.6965895e-05</td>
<td>1.19</td>
<td>8.6965895e-05</td>
<td>1.19</td>
</tr>
<tr>
<td>1/48</td>
<td>2.6950162e-05</td>
<td>1.14</td>
<td>2.6950162e-05</td>
<td>1.14</td>
</tr>
<tr>
<td>1/64</td>
<td>2.6950162e-05</td>
<td>1.14</td>
<td>2.6950162e-05</td>
<td>1.14</td>
</tr>
<tr>
<td>CPU(s)</td>
<td>34.53</td>
<td>3.34e+02</td>
<td>34.53</td>
<td>3.34e+02</td>
</tr>
</tbody>
</table>

In summary, this example provides a detailed comparison between the fast algorithm and the direct method. Specifically, we have been able to understand the superiority of our fast algorithm when $N$ is large enough. We can use the fast evaluation to obtain almost the same accuracy as the direct method, and yet, the fast algorithm is much faster and saves the computational memory for long time simulation.

Example 5.2.

We now use our fast algorithm to analyze some features of the solution to (1.1) and verify the effectiveness of our ABCs. We choose the source term $f(x, t) = 0$ and initial value $u_0(x) = \exp(-8(x-5)^2)$. In our computation, we set $\varepsilon = 10^{-10}$ and $\Omega_c = (-8, 8)$. Note that using Fourier transform, we can compute the exact solution $u_{\text{exact}}(x, t)$. In details, using Fourier transform in space variable to the first equation in Eq.(1.1), we get

$$\frac{C_0}{\alpha} D_t^\alpha \hat{u}(\xi, t) - i\xi^3 \hat{u}(\xi, t) = 0,$$

where $\xi$ is the Fourier variable. Then the solution to above equation is

$$\hat{u}_{\text{exact}}(\xi, t) = \hat{u}_0 E_\alpha(i\xi^3 t^\alpha),$$
Figure 1. Example 1: The evolution of the difference (i.e. $\text{Error} = F_{n}^{\varepsilon} - D_{n}^{\varepsilon}$) in the solutions to the direct scheme (4.21)-(4.24) (denote by $D_{n}^{\varepsilon}$) and to the fast scheme (4.50)-(4.53) (denote by $F_{n}^{\varepsilon}$) for $\varepsilon = 1e-7, 1e-8, 1e-9, 1e-10$, respectively. In the calculation, we set $M = 120$, $N = 1000$, and $\alpha = 0.5$.

Figure 2. Example 1: The CPU time of different $N$ with $M = 30$.

where $E_{\alpha}$ is the Mittag-Leffler function defined in Lemma 3.2. Then we have the exact solution of the problem using inverse Fourier transform. We compute the reference solution in a larger domain using 30000 points in space.
From Example 5.1, we find the convergence rate in space is $O(h^4)$. To confirm this result, we test the spatial convergence rate in this case. From Table 4, we can observe that the convergence rate is still $O(h^4)$.

Table 4. Example 2: The errors and convergence orders in space with $\alpha = 0.99$, $T = 0.001$ and $\Delta t = h^4$.  

<table>
<thead>
<tr>
<th>$h$</th>
<th>$E(h, \Delta t)$</th>
<th>$r_s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>3.7373804042e-05</td>
<td>4.0810</td>
</tr>
<tr>
<td>1/20</td>
<td>2.1970772051e-06</td>
<td>4.0800</td>
</tr>
<tr>
<td>1/30</td>
<td>4.0840637549e-07</td>
<td>4.2186</td>
</tr>
<tr>
<td>1/40</td>
<td>1.2188775601e-07</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3 shows the evolution of the solutions by comparing with the reference solutions for different values of $\alpha$. As can be seen in Fig. 3, there are no artificial reflections from the boundaries, which demonstrates the accuracy of our ABCs.

We also plot energies in the computational domain in Fig. 4 for different values of $\alpha$. We observe that the energy decays as the wave moves away from the computational domain, which indicates again the stability of our ABCs for long-time simulations.

**Example 5.3.**

We now check the effectiveness of our ABCs in the high-frequency region. We consider the high-frequency wave generated by $f(x, t) = 0$ and $u_0(x) = \exp(-8(x-5)^2)\sin(50x\pi/4)$. We calculate the reference solution using finer mesh sizes in a larger domain to minimize the influence of ABCs, and calculate the numerical solutions via the fast algorithm. In the calculation, we set $\epsilon = 10^{-12}$ and choose different lengths of the computational interval $\Omega$ and different final times $T$ for various values of $\alpha$. Fig. 5 shows the evolutions of the numerical solutions, which move away from the center of this wave package.

To further investigate the effectiveness of our ABCs, Fig. 6 shows the reference solutions and numerical solutions as well as their errors, and Fig. 7 shows the energy remained in the computational domains for different values of $\alpha$. We observe that the errors are very small in Fig. 6 and the energy decays in Fig. 7. We remark again that no instability has been observed in any of our computations.

**Example 5.4.**

Our fast algorithm can be easily applied to the case of non-uniform time step size. We have observed from the first example that the convergence order in time is $2 - \alpha$ when the solution is sufficiently smooth. But the rate of the convergence deteriorates significantly when an initial layer is present near $t = 0$, as shown in the third column of Table 5. In this example, we will use graded time steps to capture the singular behaviour of the solution near $t = 0$. We consider the exact solution $u(x, t) = E_\alpha(-t^\alpha)e^{-x^2}$ for the problem (1.1). The source function is $f(x, t) = E_\alpha(-t^\alpha)e^{-x^2}(-1 + 12x - 8x^3)$ and the initial data is $u_0(x) = e^{-x^2}$. In the calculation, we take the computational domain $\Omega_c = (-5, 5)$, final time $T = 1$, and...
and the tolerance precision $\varepsilon = 10^{-10}$ for the SOE approximation. And the grid in time is taken to be

$$t_k = \left(\frac{k}{N}\right)^\gamma T,$$

where $\gamma$ is a parameter to be chosen. When $\gamma = 1$, the grid is uniform; when $\gamma > 1$, the grid is non-uniform with finer mesh near $t = 0$. Table 5 shows the errors and convergence order in time for $\gamma = 1, \gamma = 2$ and $\gamma = 5/2$. We observe that while the uniform temporal grid (i.e., $\gamma = 1$) does not resolve the initial layer and thus has
Figure 4. Example 2: The evolution of the energy of the wave remained in the bounded computational domain for different values of $\alpha$. In this calculation, $M = 2000$, $N = 5000$, $x_l = -8$, $x_r = 8$, and $T = 10$.

poor convergence rate, the graded mesh steps with $\gamma = 2$ and $\gamma = 5/2$ capture the initial layer very well and the convergence rate of the overall scheme is close to the theoretical value.

Table 5. Example 4: The errors and convergence orders with $\alpha = 0.5$ and $M = 10000$ in time for various values of $\gamma$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\gamma = 1$</th>
<th></th>
<th>$\gamma = 2$</th>
<th></th>
<th>$\gamma = 5/2$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>4.1815408e-02</td>
<td></td>
<td>1.8695288e-02</td>
<td></td>
<td>1.5216478e-02</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2.4212692e-02</td>
<td>0.79</td>
<td>8.0986602e-03</td>
<td>1.21</td>
<td>6.1581981e-03</td>
<td>1.31</td>
</tr>
<tr>
<td>40</td>
<td>1.3641058e-02</td>
<td>0.83</td>
<td>3.3676129e-03</td>
<td>1.27</td>
<td>2.3893399e-03</td>
<td>1.37</td>
</tr>
<tr>
<td>80</td>
<td>7.5412148e-03</td>
<td>0.86</td>
<td>1.3592995e-03</td>
<td>1.31</td>
<td>9.0128419e-04</td>
<td>1.41</td>
</tr>
<tr>
<td>160</td>
<td>3.3672356e-03</td>
<td>0.88</td>
<td>5.3672356e-04</td>
<td>1.34</td>
<td>3.358692e-04</td>
<td>1.43</td>
</tr>
<tr>
<td>time(s)</td>
<td>8.4671958e+00</td>
<td>9.1750619e+00</td>
<td>1.2134220e+01</td>
<td>1.2134220e+01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Conclusions

In this paper, we have presented an efficient and stable numerical method for solving the linearized time-fractional KdV equation. We have derived the exact artificial boundary conditions and showed that resulting initial-boundary value problem is $L^2$ stable. An unconditionally stable finite difference is then constructed for the discretization of the initial-boundary value problem. Finally, the fast algorithm for the evaluation of the Caputo fractional derivative is incorporated into the stable finite difference scheme, which leads to a robust numerical algorithm with nearly optimal complexity. To be more precise, our fast algorithm reduces the storage requirement from $O(MN)$ to $O(MN_{\text{exp}})$ and the overall computational cost from $O(MN^2)$ to $O(MNN_{\text{exp}})$ as compared with the direct method, where $N_{\text{exp}}$ is $\log^d N$ with either $d = 1$ or $d = 2$. This provides a practical way for the long-time simulation of the linearized time-fractional KdV equation.
Figure 5. Example 3: The evolution of the reference solution (left) and the numerical solution (right). In this calculation, $x_l = -6$, $x_r = 6$, $h = 10^{-3}$, $N = 1000$. $T = 1 \times 10^{-8}$ for $\alpha = 0.5$, $T = 1 \times 10^{-5}$ for $\alpha = 0.8$, $T = 1 \times 10^{-4}$ for $\alpha = 0.9$, $T = 4 \times 10^{-4}$ for $\alpha = 0.99$. $\Delta t = T/N$. 

(a) $\alpha = 0.5$

(b) $\alpha = 0.8$

(c) $\alpha = 0.9$

(d) $\alpha = 0.99$
Figure 6. Example 3: Reference solutions (left), numerical solutions (center) and the errors between them (right) at $T = 1 \times 10^{-6}$. In the calculation of numerical solutions, we take $M = 1000$, $N = 1000$ to ensure $\Delta t$ and $h$ are same with the ones in the calculation of reference solutions.

Acknowledgements

The authors would like to thank Prof. Zhizhong Sun and the anonymous referees for many helpful suggestions and comments.
Figure 7. Example 3: The evolution of the energy of the wave remained in the bounded computational domain for different values of $\alpha$. In this calculation, we take $M = 1000$, $N = 1000$, $x_l = 0$, $x_r = 10$ and $T = 4 \times 10^{-4}$.

References


Beijing Computational Science Research Center, Beijing 100193, China
E-mail address: qianzhang@csrc.ac.cn
Beijing Computational Science Research Center, Beijing 100193, China
E-mail address: jwzhang@csrc.ac.cn

Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ 07102, USA
E-mail address: shidong.jiang@njit.edu

Beijing Computational Science Research Center, Beijing 100193, China; Department of Mathematics, Wayne State University, Detroit, MI 48202, USA
E-mail address: zzhang@math.wayne.edu