

# EFFICIENT MULTISTEP METHODS FOR TEMPERED FRACTIONAL CALCULUS: ALGORITHMS AND SIMULATIONS\*

LING GUO<sup>†</sup>, FANHAI ZENG<sup>‡</sup>, IAN TURNER<sup>§</sup>,  
KEVIN BURRAGE<sup>¶</sup>, AND GEORGE EM KARNIADAKIS<sup>||</sup>

**Abstract.** In this work, we extend the fractional linear multistep methods in [C. Lubich, *SIAM J. Math. Anal.*, 17 (1986), pp. 704–719] to the tempered fractional integral and derivative operators in the sense that the tempered fractional derivative operator is interpreted in terms of the Hadamard finite-part integral. We develop two fast methods, Fast Method I and Fast Method II, with linear complexity to calculate the discrete convolution for the approximation of the (tempered) fractional operator. Fast Method I is based on a local approximation for the contour integral that represents the convolution weight. Fast Method II is based on a globally uniform approximation of the trapezoidal rule for the integral on the real line. Both methods are efficient, but numerical experimentation reveals that Fast Method II outperforms Fast Method I in terms of accuracy, efficiency, and coding simplicity for dealing with the fractional derivative operator. The memory requirement and computational cost of Fast Method II are  $O(Q)$  and  $O(Qn_T)$ , respectively, where  $n_T$  is the number of the final time steps and  $Q$  is the number of quadrature points used in the trapezoidal rule. The effectiveness of the fast methods is verified through a series of numerical examples for long-time integration, including a numerical study of a fractional reaction-diffusion model.

**Key words.** fractional linear multistep method, fast convolution, (tempered) fractional integral and derivative, fractional activator-inhibitor system, fractional Brusselator model

**AMS subject classifications.** 26A33, 65M06, 65M12, 65M15, 35R11

**DOI.** 10.1137/18M1230153

**1. Introduction.** Fractional calculus is emerging as a powerful tool to model various physical processes involving anomalous diffusion. Under the framework of the continuous time random walks (CTRWs) model, the fractional Fokker–Planck and Klein–Kramers equations [28] are derived with power law waiting time distribution, assuming the particles may exhibit long waiting time. However, for some practical physical processes, it is necessary to make the first moment of the waiting time measure finite. This leads to the time tempered Fokker–Planck equation corresponding to the CTRWs model with a tempered power law waiting time distribution [33, 8]. For

---

\*Submitted to the journal’s Methods and Algorithms for Scientific Computing section December 3, 2018; accepted for publication (in revised form) June 4, 2019; published electronically August 1, 2019.

<https://doi.org/10.1137/18M1230153>

**Funding:** This work was supported by the National Natural Science Foundation through grant 11671265, Science Challenge Project TZ2018001, ARC Discovery Project DP150103675, and the MURI/ARO on “Fractional PDEs for Conservation Laws and Beyond: Theory, Numerics and Applications” through W911NF-15-1-0562.

<sup>†</sup>Department of Mathematics, Shanghai Normal University, Shanghai, China (lguo@shnu.edu.cn).

<sup>‡</sup>Corresponding author. South China Research Center for Applied Mathematics and Interdisciplinary Studies, South China Normal University, Guangzhou, Guangdong 510631, China (fanhaiz@foxmail.com).

<sup>§</sup>School of Mathematical Sciences, Queensland University of Technology, Brisbane, QLD 4001, Australia, and Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers, Queensland University of Technology, Brisbane, QLD 4001, Australia (i.turner@qut.edu.au).

<sup>¶</sup>School of Mathematical Sciences, Queensland University of Technology, Brisbane, QLD 4001, Australia, and Visiting Professor, Department of Computer Science, University of Oxford, Oxford, OXI 3QD, UK (kevin.burrage@qut.edu.au).

<sup>||</sup>Division of Applied Mathematics, Brown University, Providence RI, 02912 (george\_karniadakis@brown.edu).

more applications of tempered fractional calculus and differential equations in poroelasticity, ground water hydrology, and geophysical flows, see [11, 4, 27, 26, 25, 15].

The aim of this paper is to develop fast and memory-saving methods for discretizing the (tempered) fractional integral of the following form:

$$(1.1) \quad \frac{1}{\Gamma(-\alpha)} \int_0^t (t-s)^{-\alpha-1} e^{-\sigma(t-s)} u(s) \, ds, \quad \alpha \in \mathbb{R}, \sigma \geq 0.$$

If  $\alpha > 0$ , then the above integral is interpreted in terms of the Hadamard finite-part integral, which is equivalent to the (tempered) fractional derivative of order  $\alpha$ ; see Lemma 2.7.

When  $\sigma = 0$ , (1.1) reduces to the Riemann–Liouville (RL) fractional integral of order  $-\alpha$  ( $\alpha < 0$ ) or the RL fractional derivative of order  $\alpha$  ( $\alpha > 0$ ). Thus, the method developed in the present paper is a general framework for (tempered) fractional calculus. Therefore, we will mainly focus on the fast computation of the tempered fractional integral (1.1) for  $\alpha > 0$ . Recently, some numerical methods have been developed to solve the tempered fractional differential equations (FDEs) via finite difference methods; see [4, 6, 15, 22]. However, fast and memory-saving methods for tempered FDEs are limited.

In this paper, we extend Lubich’s fractional linear multistep methods (FLMMs) (see [20]) to discretize the tempered fractional integral and derivative operators, which yields the discrete convolution as

$$(1.2) \quad \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k, \quad 0 \leq n \leq n_T,$$

where  $\tau$  is the time step size,  $n_T$  is a positive integer,  $\alpha$  is real,  $\sigma \geq 0$ ,  $\omega_k^{(\alpha, \sigma)}$  are the convolution quadrature weights, and  $u_k$  can be any number; see section 3 for details.

The discrete convolution (1.2) requires  $O(n_T)$  active memory and  $O(n_T^2)$  arithmetic operations by direct computation. Thus, the direct calculation of (1.2) becomes computationally expensive when it is applied to discretize time-fractional partial differential equations (PDEs). Recently, some progress has been made to reduce the memory requirement and computational cost of the discrete convolution for approximating the RL fractional operators [1, 2, 14, 17, 19, 23, 37, 39]. For the fast methods based on piecewise polynomial interpolation, the kernel function in the fractional operators is approximated by the sum-of-exponentials, that is to say, the quadrature weights  $\omega_n^{(\alpha, \sigma)}$  in (1.2) originate from interpolation; see [14, 17, 37, 38].

In this work, we develop two fast methods for calculating (1.2) with the quadrature weights  $\omega_n^{(\alpha, \sigma)}$  derived from generating functions, where the methods in [14, 17, 37, 38] cannot apply here. Although the discrete convolution (1.2) can be efficiently calculated by FFT with  $O(n_T \log n_T)$  operations, the storage requirement is  $O(n_T)$ . Hairer, Lubich, and Schlichte [10] employed FFTs to develop a fast method with  $O(n_T (\log n_T)^2)$  operations for calculating (1.2), which was applied to solve nonlinear Volterra convolution equations.

The basic idea of the present fast method is to reexpress the weight  $\omega_n^{(\alpha, \sigma)}$  as an integral form. In the *first method*, we express  $\omega_n^{(\alpha, \sigma)}$  as a contour integral of the form

$$(1.3) \quad \omega_n^{(\alpha, \sigma)} = \frac{\tau^{1+\alpha} e^{-n\sigma\tau}}{2\pi i} \int_{\mathcal{C}} \lambda^\alpha (1 - \lambda\tau)^{-1-n} F_\omega(\lambda) \, d\lambda,$$

then a suitable contour quadrature (such as Talbot, hyperbolic, or parabolic contour quadrature) is used to discretize (1.3). The case of  $\sigma = 0$  has been investigated in

[2, 19, 39]. The detailed derivation of (1.3) is illustrated in Appendix B (see also [19, 39]). In this work, we extend the method in [39] to the tempered fractional calculus ( $\sigma > 0$ ) to obtain Fast Method I, in which the Talbot contour quadrature used in [39] is also applied here.

The *second method* is inspired by [2], where a Hankel contour beginning and ending in the left half of the complex plane is applied to transform the contour integral into an integral on the half line, which is discretized by a multidomain Gauss quadrature, yielding a uniform approximation. We can also choose the same Hankel contour as in [2] to express the quadrature weight  $\omega_n^{(\alpha, \sigma)}$  defined by (1.3) as an integral on the half line

$$(1.4) \quad \omega_n^{(\alpha, \sigma)} = \tau^{1+\alpha} e^{-n\sigma\tau} \frac{\sin(\alpha\pi)}{\pi} \int_0^\infty \lambda^\alpha (1 + \lambda\tau)^{-1-n} F_\omega(-\lambda) d\lambda.$$

The above integral is further transformed into an integral on the real line by letting  $\lambda = \exp(x)$ ; see (4.11). Finally, the exponentially convergent trapezoidal rule [35] is applied to obtain a uniform approximation of (1.4), which leads to Fast Method II.

We list the main contributions of this work as follows:

- We extend the FLMMs proposed in [20] to both the tempered fractional integral and derivative operators, where the tempered fractional operators are interpreted in terms of the Hadamard finite-part integral, which significantly simplifies the results in [5].
- We develop two new fast methods, Fast Methods I and II, to calculate the discrete convolutions to the approximation of the (tempered) fractional integral and derivative operators. For  $\alpha > 0$ , Fast Method II outperforms Fast Method I in terms of accuracy, efficiency, and coding simplicity and has the following advantages:
  - (a) The time interval is not divided into exponentially increasing subintervals, which makes the implementation of Fast Method II much easier than Fast Method I and the existing fast methods in [3, 32, 39].
  - (b) Only real operations are performed and the recurrence relation (4.21) used in Fast Method II is stable.
  - (c) Using the same number of quadrature points, Fast Method II achieves higher accuracy than Fast Method I for  $\alpha > 0$ .

Theoretically speaking, Fast Method I may work for a wider range of  $\alpha \in \mathbb{R}$  by choosing a suitable contour quadrature rule but may achieve less accurate numerical results when  $\alpha > 1$  (see Figure 4(d)). Fast Method II exhibits less accuracy when  $-1 < \alpha < 0$  and  $\alpha \rightarrow -1$  (see Figure 4(a)) and cannot be applied for  $\alpha \leq -1$ . The obvious disadvantage of Fast Method I is that its implementation is more complicated than Fast Method II. We compare the two fast methods to show the superiority of Fast Method II over Fast Method I when  $\alpha \in (0, 2)$ . We focus on the use of Fast Method II to solve fractional models through numerical simulations.

This paper is organized as follows. In section 2, we prove that the tempered fractional derivative can be interpreted in terms of the Hadamard finite-part integral. This interpretation helps us to extend Lubich's FLMMs to both the tempered fractional integral and derivative operators directly; see section 3. In section 4, we propose two fast methods for approximating the discrete convolution in the considered FLMM for the tempered fractional operator, and we also make a comparison between these two methods. Fast Method II is applied to solve tempered fractional ordinary differential equations (ODEs) and a coupled system of nonlinear time-fractional activator-inhibitor equations in section 5 before the conclusion is given in the last section.

**2. Preliminaries.** In this section, we introduce definitions of fractional integrals and derivatives, and the properties that will be used in this paper.

DEFINITION 2.1 (RL fractional integral). *The RL fractional integral operator  $I_{0,t}^\alpha u(t)$  of order  $\alpha$  ( $\alpha \geq 0$ ) is defined by*

$$(2.1) \quad I_{0,t}^\alpha u(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} u(s) \, ds.$$

DEFINITION 2.2 (RL fractional derivative). *The RL fractional derivative operator  ${}_R L D_{0,t}^\alpha$  of order  $\alpha$  is defined by*

$$(2.2) \quad {}_R L D_{0,t}^\alpha u(t) = \frac{1}{\Gamma(m-\alpha)} \left[ \frac{d^m}{dt^m} \int_0^t (t-s)^{m-\alpha-1} u(s) \, ds \right],$$

where  $m-1 < \alpha \leq m$ ,  $m$  is a positive integer.

DEFINITION 2.3 (tempered fractional integral). *The tempered fractional integral operator  $I_{0,t}^{\alpha,\sigma}$  of order  $\alpha$  ( $\alpha, \sigma \geq 0$ ) is defined by*

$$(2.3) \quad I_{0,t}^{\alpha,\sigma} u(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} e^{-\sigma(t-s)} u(s) \, ds.$$

DEFINITION 2.4 (tempered fractional derivative). *The tempered fractional derivative operator  $D_{0,t}^{\sigma,\alpha}$  of order  $\alpha > 0$  is defined by*

$$(2.4) \quad \begin{aligned} D_{0,t}^{\sigma,\alpha} u(t) &= (\partial_t + \sigma)^m I_{0,t}^{\sigma, m-\alpha} u(t) \\ &= (\partial_t + \sigma)^m \left[ \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-s)^{m-\alpha-1} e^{-\sigma(t-s)} u(s) \, ds \right], \end{aligned}$$

where  $\sigma \geq 0$ ,  $(\partial_t + \sigma)^m = \sum_{k=0}^m \binom{m}{k} \partial_t^k \sigma^{m-k}$ ,  $m-1 < \alpha \leq m$ ,  $m$  is a positive integer.

Next, we introduce the Hadamard finite-part integral, which plays a crucial role in the numerical approximation of the (tempered) fractional derivative operator.

**2.1. Fractional derivatives in the Hadamard sense.** In [30, 31], the RL fractional derivative operator is proved to be equivalent to a Hadamard finite-part integral. In this section, we extend this proof to the tempered fractional calculus.

DEFINITION 2.5 (Hadamard finite-part integral; see [30, 31]). *Let a function  $f(x)$  be integrated on an interval  $(\epsilon, A)$  for any  $A > 0$  and  $0 < \epsilon < A$ . The function  $f(x)$  is said to possess the Hadamard property at the point  $x = 0$  if there exist constants  $a_k$ ,  $b_0$ , and  $\lambda_k > 0$  such that*

$$(2.5) \quad \int_\epsilon^A f(x) \, dx = \sum_{k=1}^N a_k \epsilon^{-\lambda_k} + b_0 \ln \frac{1}{\epsilon} + J_0(\epsilon),$$

where  $\lim_{\epsilon \rightarrow 0} J_0(\epsilon)$  exists and is finite, which is also denoted by

$$(2.6) \quad P.V. \int_0^A f(x) \, dx = \lim_{\epsilon \rightarrow 0} J_0(\epsilon).$$

LEMMA 2.6 (see [31, p. 112]). *The RL fractional derivative  ${}_R L D_{0,t}^\alpha u(t)$ ,  $\alpha > 0$ ,  $\alpha \neq 1, 2, \dots$ , is equivalent to the following integral in the Hadamard sense, that is,*

$$(2.7) \quad {}_R L D_{0,t}^\alpha u(t) = \frac{1}{\Gamma(-\alpha)} P.V. \int_0^t (t-s)^{-\alpha-1} u(s) \, ds.$$

LEMMA 2.7. *The tempered fractional derivative of order  $\alpha > 0$  is equivalent to the following Hadamard finite-part integral:*

$$(2.8) \quad D_{0,t}^{\alpha,\sigma} u(t) = \frac{1}{\Gamma(-\alpha)} P.V. \int_0^t (t-s)^{-\alpha-1} e^{-\sigma(t-s)} u(s) ds.$$

*Proof.* Let  $f(t) = e^{-\sigma t}$  and  $g(t) = \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-s)^{m-\alpha-1} e^{\sigma s} u(s) ds$ . Then

$$\begin{aligned} D_{0,t}^{\alpha,\sigma} u(t) &= (\partial_t + \sigma)^m (fg) = \sum_{k=0}^m \binom{m}{k} \sigma^{m-k} \partial_t^k (fg) \\ &= \sum_{k=0}^m \binom{m}{k} \sigma^{m-k} \sum_{j=0}^k \binom{k}{j} f^{(k-j)}(t) g^{(j)}(t) \\ &= f(t) \sum_{k=0}^m \sum_{j=0}^k \binom{m}{k} \binom{k}{j} \sigma^{m-j} (-1)^{k-j} g^{(j)}(t) \\ (2.9) \quad &= f(t) \sum_{j=0}^m \sigma^{m-j} g^{(j)}(t) \sum_{k=j}^m \binom{m}{k} \binom{k}{j} (-1)^{k-j}, \end{aligned}$$

where we have used  $f^{(k)}(t) = (-\sigma)^k e^{-\sigma t} = (-\sigma)^k f(t)$ .

In the following, we will prove that

$$(2.10) \quad \sum_{k=j}^m \binom{m}{k} \binom{k}{j} (-1)^{k-j} = \begin{cases} 0, & 0 \leq j \leq m-1, \\ 1, & j = m. \end{cases}$$

Obviously, one has  $\sum_{k=j}^m \binom{m}{k} \binom{k}{j} (-1)^{k-j} = 1$  for  $j = m$ . For  $0 \leq j \leq m-1$ , we have

$$\begin{aligned} \sum_{k=j}^m \binom{m}{k} \binom{k}{j} (-1)^{k-j} &= \sum_{k=j}^m (-1)^{k-j} \frac{m!}{k!(m-k)!} \frac{k!}{(k-j)!j!} \\ (2.11) \quad &= \frac{m(m-1) \cdots (m-j+1)}{j!} \sum_{k=j}^m \frac{(m-j)!(-1)^{k-j}}{(m-k)!(k-j)!} \\ &= \frac{m(m-1) \cdots (m-j+1)}{j!} (1-1)^{m-j} = 0, \quad j < m. \end{aligned}$$

Combining (2.9) and (2.10) yields

$$\begin{aligned} D_{0,t}^{\alpha,\sigma} u(t) &= f(t) g^{(m)}(t) = e^{-\sigma t} \frac{d^m}{dt^m} \left[ \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-s)^{m-\alpha-1} e^{\sigma s} u(s) ds \right] \\ (2.12) \quad &= \frac{e^{-\sigma t}}{\Gamma(-\alpha)} P.V. \int_0^t (t-s)^{-\alpha-1} e^{\sigma s} u(s) ds, \end{aligned}$$

where Lemma 2.6 is applied. The proof is complete.  $\square$

**3. Fractional linear multistep methods.** In this section, we extend Lubich's FLMMs (see [20]) to discretize the tempered fractional integral and derivative operators. For convenience, we introduce the following notation:

$$(3.1) \quad D_{\tau}^{\alpha,\sigma,\gamma,m,n} u = \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha,\sigma)} (u(t_k) - u_0) + \tau^{-\alpha} \sum_{k=1}^m w_{n,k}^{(\alpha,\sigma)} (u(t_k) - u_0),$$

where  $\tau$  is the step size,  $t_k = k\tau$  is the grid point,  $\gamma = (\gamma_1, \gamma_2, \dots)$ ,  $\gamma_{j+1} > \gamma_j > 0$ , and the quadrature weights  $\omega_k^{(\alpha, \sigma)}$  are chosen such that  $D_\tau^{\alpha, \sigma, \gamma, m, n} u$  is a stable approximation of  $[D_{0,t}^{\alpha, \sigma}(u(t) - u_0)]_{t=t_n}$ . When the quadrature weights  $\omega_k^{(\alpha, \sigma)}$  are determined, the starting weights  $w_{n,k}^{(\alpha, \sigma)}$  ( $1 \leq k \leq m$ ) will be chosen such that  $D_\tau^{\alpha, \sigma, \gamma, m, n} u = [D_{0,t}^{\alpha, \sigma} u(t)]_{t=t_n}$  for some  $u(t) = t^{\gamma_j}$ ,  $1 \leq j \leq m$ . The details on calculating the starting weights  $w_{n,k}^{(\alpha, \sigma)}$  are presented in Appendix A.

The convolution quadrature weights  $\omega_k^{(\alpha, \sigma)}$  in (3.1) can be given by the following generating functions; see [20].

- The fractional backward difference formula of order  $p$  (FBDF- $p$ ):

$$(3.2) \quad \omega^{(\alpha, \sigma)}(z) = \left( \sum_{k=1}^p \frac{1}{k} (1 - ze^{-\sigma\tau})^k \right)^\alpha = \sum_{k=0}^{\infty} \omega_k^{(\alpha, \sigma)} z^k.$$

- The generalized Newton–Gregory formula of order  $p$  (NGGF- $p$ )

$$(3.3) \quad \omega^{(\alpha, \sigma)}(z) = (1 - ze^{-\sigma\tau})^\alpha \sum_{k=1}^p g_{k-1} (1 - ze^{-\sigma\tau})^{k-1} = \sum_{k=0}^{\infty} \omega_k^{(\alpha, \sigma)} z^k,$$

where  $g_k$  are given by (see, e.g., [9])

$$\begin{aligned} g_0 &= 1, & g_1 &= \frac{\alpha}{2}, & g_2 &= \frac{\alpha^2}{8} + \frac{5\alpha}{24}, \\ g_3 &= \frac{\alpha^3}{48} + \frac{5\alpha^2}{48} + \frac{\alpha}{8}, & g_4 &= \frac{\alpha^4}{384} + \frac{5\alpha^3}{192} + \frac{97\alpha^2}{1152} + \frac{251\alpha}{2880}, \\ g_5 &= \frac{\alpha^5}{3840} + \frac{5\alpha^4}{1152} + \frac{61\alpha^3}{2304} + \frac{401\alpha^2}{5760} + \frac{19\alpha}{288}. \end{aligned}$$

- The fractional trapezoidal rule

$$(3.4) \quad \omega^{(\alpha, \sigma)}(z) = \left( \frac{(1 - ze^{-\sigma\tau})}{2(1 + ze^{-\sigma\tau})} \right)^\alpha = \sum_{k=0}^{\infty} \omega_k^{(\alpha, \sigma)} z^k.$$

See (3.6) for other choices of the coefficients  $\omega_k^{(\alpha, \sigma)}$ .

Under suitable conditions, (3.1) is a  $p$ th-order approximation of  $D_{0,t}^{\alpha, \sigma}(u(t) - u_0)$  if the generating function (3.2) or (3.3) is used, and a second-order approximation is derived if (3.4) is applied.

From [20], we immediately derive the following two theorems.

**THEOREM 3.1.** *Let  $\alpha \in \mathbb{R}$ ,  $\delta > 0$ . Then for  $u(t) = t^\delta$ , one has*

$$(3.5) \quad [D_{0,t}^{\alpha, \sigma} u(t)]_{t=t_n} = D_\tau^{\alpha, \sigma, \gamma, 0, n} u + O(t_n^{\alpha+\delta-p}\tau^p) + O(t_n^{\alpha-1}\tau^{\delta+1}).$$

**THEOREM 3.2.** *Let  $(\hat{\rho}, \hat{\sigma})$  denote an implicit linear multistep method (LMM) which is stable and consistent of order  $p$ , i.e.,  $\hat{\rho}(z)$  and  $\hat{\sigma}(z)$  are the characteristic polynomials of the LMM of order  $p$  for the first-order ODE. Assume that the zeros of  $\hat{\sigma}(z)$  have absolute value less than 1. Let  $\omega(z) = \frac{\hat{\sigma}(1/z)}{\hat{\rho}(1/z)}$  and*

$$(3.6) \quad \omega^{(\alpha, \sigma)}(z) = (\omega(ze^{-\sigma\tau}))^\alpha = \sum_{k=0}^{\infty} \omega_k^{(\alpha, \sigma)} z^k, \quad \alpha \in \mathbb{R}.$$

Then, we have

$$[D_{0,t}^{\sigma, \alpha}(u(t) - u(0))]_{t=t_n} = D_\tau^{\alpha, \sigma, \gamma, m, n} u + O(\tau^p).$$

Next, we discuss how to implement the fast computation of the convolution quadrature coefficients  $\omega_k^{(\alpha, \sigma)}$  defined by (3.2), (3.3), and (3.4). In fact, we need only to consider how to derive  $\omega_k^{(\alpha, \sigma)}$  defined by (3.2), since  $\omega_k^{(\alpha, \sigma)}$  given in (3.3) and (3.4) can be derived from the coefficients given in (3.2) for  $p = 1$ . For the FBDF- $p$  given in (3.2), the coefficients satisfy  $\omega_k^{(\alpha, \sigma)} = e^{-k\tau\sigma} \omega_k^{(\alpha, 0)}$ , where  $\omega_k^{(\alpha, 0)}$  can be efficiently calculated by a recurrence formula; see, e.g., [7, 16].

**4. Fast calculation.** In this section, we present fast calculations for the convolution  $\tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha, \sigma)} u(t_k)$  defined in (3.5), where the coefficients  $\omega_k^{(\alpha, \sigma)}$  can be derived from (3.2), (3.3), (3.4), or (3.6). The key idea is to represent the coefficients  $\omega_k^{(\alpha, \sigma)}$  using the integral formula and then approximate it using numerical quadrature. We first extend the fast method in [39] to calculate the discrete convolution  $\tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha, \sigma)} u(t_k)$  in (3.1), which is called Fast Method I in the following context. Then we propose the second fast method based on the approaches in [2, 21, 39], which is called Fast Method II.

**4.1. Fast Method I.** Following the approach developed in [39], the convolution quadrature weights  $\omega_n^{(\alpha, \sigma)}$  in (3.5) can be expressed as

$$(4.1) \quad \omega_n^{(\alpha, \sigma)} = e^{-n\sigma\tau} \omega_n^{(\alpha, 0)} = \frac{\tau^{1+\alpha} e^{-n\sigma\tau}}{2\pi i} \int_{\mathcal{C}_\ell} \lambda^\alpha (1 - \lambda\tau)^{-1-n} F_\omega(\lambda) d\lambda,$$

where  $\mathcal{C}_\ell$  is a contour that surrounds the poles of  $(1 - \lambda\tau)^{-1-n}$  and  $F_\omega(\lambda)$  (see also (38) in [39]) is related to the FLMM (3.1) defined by the generating functions, which is given by

$$(4.2) \quad F_\omega(\lambda) = (\tau\lambda)^{-\alpha} \omega^{(\alpha, 0)}(1 - \tau\lambda),$$

where  $\omega^{(\alpha, 0)}(z)$  is defined by (3.2), (3.3), (3.4), or (3.6).

To approximate the contour integral (4.1) with high accuracy, we apply the mid-point rule based on the Talbot contour (see, e.g., [19, 39]) to obtain

$$(4.3) \quad \omega_n^{(\alpha, \sigma)} \approx \tilde{\omega}_n^{(\alpha, \sigma)} = 2\tau^{1+\alpha} e^{-n\sigma\tau} \mathbf{Im} \left( \sum_{j=0}^{N-1} w_j^{(\ell)} (\lambda_j^{(\ell)})^\alpha (1 - \lambda_j^{(\ell)} \tau)^{-1-n} F_\omega(\lambda_j^{(\ell)}) \right),$$

where the quadrature points  $\lambda_j^{(\ell)}$  and weights  $w_j^{(\ell)}$  are given by (see, e.g., [36, 39])

$$(4.4) \quad \lambda_j^{(\ell)} = z(\theta_j, N/T_\ell), \quad w_j^{(\ell)} = \partial_\theta z(\theta_j, N/T_\ell), \quad \theta_j = (2j+1)\pi/(2N),$$

with  $z(\theta, \mu) = \mu(-0.4814 + 0.6443(\theta \cot(\theta) + i0.5653\theta))$ ,  $T_\ell = (2B^\ell - 2 + n_0)\tau$ , and  $B > 1$  is a positive integer.

According to the procedure in [21, 32, 39], we need to first find the smallest integer  $L$  satisfying  $n - n_0 + 1 \leq 2B^L$  for each  $n \geq n_0$ . Then for  $\ell = 1, 2, \dots, L$ , we obtain a unique integer  $q_\ell$  satisfying

$$(4.5) \quad b_\ell^{(n)} = q_\ell B^\ell \quad \text{with} \quad n - n_0 + 1 - b_\ell^{(n)} \in [B^{\ell-1}, 2B^\ell - 1].$$

Set  $b_0^{(n)} = n - n_0$  and  $b_L^{(n)} = 0$ . Readers can refer to [32] for the pseudocode for determining  $q_\ell$  and  $b_\ell^{(n)}$ .

To develop the fast method, the convolution  $u_n^{(\alpha, \sigma)} = \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k$  is decomposed as

$$(4.6) \quad u_n^{(\alpha, \sigma)} = \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k = \sum_{\ell=0}^L u_n^{(\ell)}$$

with  $u_n^{(0)} = \tau^{-\alpha} \sum_{k=n-n_0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k$  and  $u_n^{(\ell)} = \tau^{-\alpha} \sum_{k=b_{\ell}^{(n)}}^{b_{\ell-1}^{(n)}-1} \omega_{n-k}^{(\alpha, \sigma)} u_k$ . For each part  $u_n^{(\ell)}$ , we can use (4.3) to approximate the corresponding quadrature weights. The summary of Fast Method I is given in Algorithm 1.

---

**Algorithm 1** Fast Method I for approximating  $u_n^{(\alpha, \sigma)} = \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k$ , where  $\omega_k^{(\alpha, \sigma)}$  satisfies (4.1).

---

- 1: Input: the fractional order  $\alpha$  and  $\sigma \geq 0$ , a time step size  $\tau > 0$ , the suitable positive integers  $n_0$ ,  $N$ , and  $B \geq 2$ , the quadrature points  $\lambda_j^{(\ell)}$  and weights  $w_j^{(\ell)}$  (see (4.15)), the coefficients  $\omega_n^{(\alpha, \sigma)}$  ( $0 \leq n \leq n_0$ ) defined by (3.2), (3.3), (3.4), or (3.6), and the function  $F_\omega(\lambda)$  defined by (4.2).
- 2: Output: the fast approximation  ${}_F u_n^{(\alpha, \sigma)}$  of  $u_n^{(\alpha, \sigma)}$  (see (4.6)).
  - Step 1. Find the smallest integer  $L$  satisfying  $n - n_0 + 1 \leq 2B^L$  for each  $n \geq n_0$ .
  - Step 2. Determine  $q_\ell$  according to (4.5) for  $\ell = 1, 2, \dots, L-1$ .
  - Step 3. For every  $1 \leq \ell \leq L$ , approximate  $u_n^{(\ell)}$  by

$$(4.7) \quad \hat{u}_n^{(\ell)} = 2\mathbf{Im} \left\{ \sum_{j=0}^{N-1} w_j^{(\ell)} F_\omega(\lambda_j^{(\ell)}) (1 - \tau \lambda_j^{(\ell)})^{-[n - b_{\ell-1}^{(n)} - 1]} y_j^{(n, \ell)} \right\},$$

where

$$y_j^{(n, \ell)} = \tau \sum_{k=b_{\ell}^{(n)}}^{b_{\ell-1}^{(n)}-1} \left( 1 - \tau \lambda_j^{(\ell)} \right)^{-\left( b_{\ell-1}^{(n)} - 1 - k \right) - 1} u_k$$

is the backward Euler approximation to the solution at  $t = b_{\ell-1}^{(n)} \tau$  of the linear initial-value problem

$$(4.8) \quad y'(t) = \lambda_j^{(\ell)} y(t) + u(t), \quad y(b_{\ell}^{(n)} \tau) = 0.$$

- Step 4. Calculate

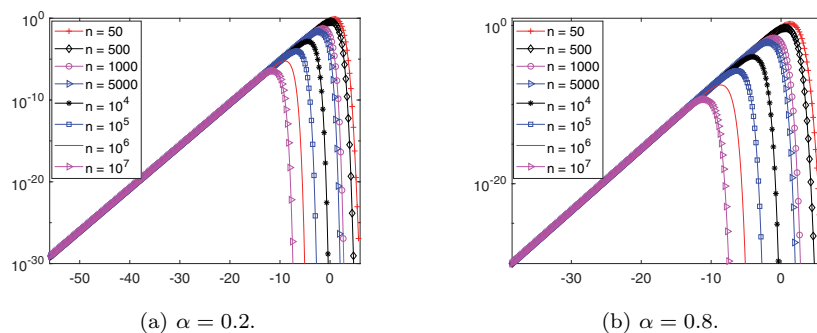
$$(4.9) \quad {}_F u_n^{(\alpha, \sigma)} = u_n^{(0)} + \hat{u}_n^{(1)} + \dots + \hat{u}_n^{(L)}.$$


---

*Remark 4.1.* Here we use the Talbot contour quadrature to approximate the contour integral (4.1). However, other contour quadratures can be used to discretize (4.1), such as the hyperbolic and parabolic contour quadratures. For more details, see [3, 18, 32, 39] and references therein.

*Remark 4.2.* It is shown in [39] that the memory requirement and computational cost of Fast Method I are about  $O(N \log n_T)$  and  $O(N n_T \log n_T)$ , respectively, when  $n_T$  is sufficiently large.



FIG. 1. The exponential decay of  $|\phi_n(x)|$  for second-order GNGF,  $\tau = 0.01$ .

**4.2. Fast Method II.** Instead of using (4.1) for expressing the convolution weights  $\omega_n^{(\alpha, \sigma)}$ , we extend [2, Lemma 9] to reexpress the contour integral (4.1) into the following form:

$$(4.10) \quad \omega_n^{(\alpha, \sigma)} = -\tau^{1+\alpha} e^{-n\sigma\tau} \frac{\sin(\alpha\pi)}{\pi} \int_0^\infty \lambda^\alpha (1 + \lambda\tau)^{-1-n} F_\omega(-\lambda) d\lambda, \quad \alpha > -1,$$

where  $F_\omega$  is given by (4.2). The key point is how to approximate (4.10) efficiently and accurately. Here we follow the idea in [24] and let  $\lambda = \exp(x)$ . Then the integral (4.10) becomes

$$(4.11) \quad \omega_n^{(\alpha, \sigma)} = e^{-n\sigma\tau} \omega_n^{(\alpha, 0)} = \tau^{1+\alpha} e^{-n\sigma\tau} \int_{-\infty}^\infty \phi_n(x) dx,$$

where

$$(4.12) \quad \phi_n(x) = (1 + e^x\tau)^{-1-n} \phi(x), \quad \phi(x) = -\frac{\sin(\alpha\pi)}{\pi} e^{(1+\alpha)x} F_\omega(-e^x).$$

We find that  $\phi_n(x)$  decays exponentially as  $|x| \rightarrow \infty$  for any  $n > n_0$ , where  $n_0$  is a suitable positive integer. Figure 1 shows the exponential decay of  $\phi_n(x)$  for  $\alpha = 0.2$  and  $0.8$  when the second-order GNGF (3.3) is applied,  $n_0 = 50$ . For the GNGF- $p$  and FBDF- $p$ , and any fractional order  $\alpha > 0$ , the corresponding  $\phi_n(x)$  decays exponentially for  $n > n_0$  as  $|x| \rightarrow \infty$  but these results are not shown here.

The exponential decay of  $\phi_n(x)$  inspires us to use the exponentially convergent trapezoidal rule (see [35]) to approximate the integral  $\int_{-\infty}^\infty \phi_n(x) dx$ . Thus, we have

$$(4.13) \quad \omega_n^{(\alpha, \sigma)} \approx \hat{\omega}_n^{(\alpha, \sigma)} = \tau^{1+\alpha} e^{-n\sigma\tau} \Delta x \sum_{j=-\infty}^\infty (1 + e^{j\Delta x}\tau)^{-1-n} \phi_n(j\Delta x),$$

where  $\Delta x > 0$  is a step size that determines the accuracy.

We have the following theorem for the error of (4.13); see [35].

**THEOREM 4.1** (Trefethen and Weideman [35]). *Suppose  $\phi_n(x)$  is analytic in the strip  $|Im(x)| < a$  for some  $a > 0$ . Suppose further that  $\phi_n(x) \rightarrow 0$  uniformly as  $|x| \rightarrow \infty$  in the strip, and for some  $M$ ,  $\int_{-\infty}^\infty |\phi_n(x + ib)| dx \leq M$  for all  $b \in (-a, a)$ . Then, for any  $\Delta x > 0$ ,  $\hat{\omega}_n^{(\alpha, \sigma)}$  as defined by (4.13) exists and satisfies*

$$(4.14) \quad |\omega_n^{(\alpha, \sigma)} - \hat{\omega}_n^{(\alpha, \sigma)}| \lesssim M \tau^{1+\alpha} e^{-n\sigma\tau} \left( e^{2\pi a/\Delta x} - 1 \right)^{-1},$$

where  $\lesssim$  means that there exists a positive constant  $C$  that is independent of  $n$ ,  $\tau$ , and  $\Delta x$ , such that  $|\omega_n^{(\alpha,\sigma)} - \hat{\omega}_n^{(\alpha,\sigma)}| \leq CM\tau^{1+\alpha}e^{-n\sigma\tau}(e^{2\pi a/\Delta x} - 1)^{-1}$ .

*Remark 4.3.* For the GNGF- $p$  (see (3.3)), the fractional trapezoidal rule (see (3.4)), or the FBDF- $p$  (see (3.2)), the corresponding  $a$  in Theorem 4.1 is given in Appendix D, and an upper bound of  $M$  in (4.14) is  $(n\tau)^{-\alpha-1}$ . Hence, (4.14) can be rewritten as

$$|\omega_n^{(\alpha,\sigma)} - \hat{\omega}_n^{(\alpha,\sigma)}| \lesssim e^{-n\sigma\tau} n^{-\alpha-1} \left(e^{2\pi a/\Delta x} - 1\right)^{-1},$$

where  $a$  can be chosen as  $\pi/2$  for the GNGF- $p$ .

In numerical simulations, we truncate (4.13) and derive the following modified version that is used in this paper:

$$\begin{aligned} \tilde{\omega}_n^{(\alpha,\sigma)} &= \tau^{1+\alpha} e^{-n\sigma\tau} \Delta x \sum_{j=Q_1}^{Q_2} (1 + e^{j\Delta x} \tau)^{-1-n} \phi_n(j\Delta x) \\ (4.15) \quad &= \tau^{1+\alpha} e^{-n\sigma\tau} \sum_{j=0}^{Q-1} \varpi_j (1 + \lambda_j \tau)^{-1-n}, \end{aligned}$$

where  $Q = Q_2 - Q_1$ ,  $x_j = (Q_1 + j)\Delta x$ ,  $\varpi_j = -\Delta x \frac{\sin(\alpha\pi)}{\pi} e^{(1+\alpha)x_j} F(-e^{x_j})$ , and  $\lambda_j = e^{x_j}$ . The truncated formula (4.15) is somewhat equivalent to applying the composite trapezoidal rule to the integral  $\tau^{1+\alpha} e^{-n\sigma\tau} \int_{x_{\min}}^{x_{\max}} \phi_n(x) dx$ , where

$$(4.16) \quad x_{\min} = \frac{\log(\epsilon)}{1+\alpha} - \log(n_T\tau), \quad x_{\max} = \log\left(\frac{-2\log(\epsilon) + 2(1+\alpha)\log(n_0\tau)}{n_0\tau}\right).$$

Here  $n_T$  is the largest number of time steps,  $n_0$  is a positive number that makes (4.15) work well for  $n \geq n_0$ , and  $\epsilon$  (we take  $\epsilon = 10^{-16}$  in this paper) is a given precision satisfying

$$\left|\int_{-\infty}^{x_{\min}} \phi_n(x) dx\right| \lesssim \epsilon(n\tau)^{-1-\alpha} \quad \text{and} \quad \left|\int_{x_{\max}}^{\infty} \phi_n(x) dx\right| \lesssim \epsilon(n\tau)^{-1-\alpha}.$$

In the numerical simulations performed in this work, the quadrature points  $x_j$  and weights  $\varpi_j$  in (4.15) are defined by  $\varpi_j = \Delta x \phi(x_j)$  (see (4.12) for the definition of  $\phi(x)$ ) and  $\lambda_j = \exp(x_j)$ , respectively, where  $\Delta x = (x_{\max} - x_{\min})/(Q-1)$ ,  $Q$  is a positive integer, and  $x_j = x_{\min} + j\Delta x$ . We show how to obtain (4.16) in Appendix C.

Based on (4.15), we give a detailed implementation of Fast Method II. We first decompose the discrete convolution  $u_n^{(\alpha,\sigma)} = \tau^{-\alpha} \sum_{j=0}^n \omega_{n-j}^{(\alpha,\sigma)} u_j$  into

$$(4.17) \quad u_n^{(\alpha,\sigma)} = {}_L u_{n,n_0}^{(\alpha,\sigma)} + {}_H u_{n,n_0}^{(\alpha,\sigma)} \equiv \tau^{-\alpha} \sum_{k=n-n_0}^n \omega_{n-k}^{(\alpha,\sigma)} u_k + \tau^{-\alpha} \sum_{k=0}^{n-n_0-1} \omega_{n-k}^{(\alpha,\sigma)} u_k.$$

Then, the local part  ${}_L u_{n,n_0}^{(\alpha,\sigma)}$  is calculated directly. In the following, we give a simple illustration on how to obtain  ${}_H u_{n,n_0}^{(\alpha,\sigma)}$ . Inserting  $\omega_n^{(\alpha,\sigma)}$  defined by (4.11) into  ${}_H u_{n,n_0}^{(\alpha,\sigma)}$ , we obtain

$$(4.18) \quad {}_H u_{n,n_0}^{(\alpha,\sigma)} = \tau \sum_{k=0}^{n-n_0-1} e^{-(n-k)\sigma\tau} u_k \int_{-\infty}^{\infty} (1 + e^x \tau)^{-1-(n-k)} \phi(x) dx.$$

Applying (4.15) to the above integral yields

$$\begin{aligned}
 {}_H u_{n,n_0}^{(\alpha,\sigma)} &\approx {}_F u_{n,n_0}^{(\alpha,\sigma)} = \tau \sum_{k=0}^{n-n_0-1} u_k e^{-(n-k)\sigma} \sum_{j=0}^{Q-1} \varpi_j (1 + \lambda_j \tau)^{-1-(n-k)} \\
 (4.19) \quad &= \sum_{j=0}^{Q-1} \varpi_j \tau \sum_{k=0}^{n-n_0-1} e^{-(n-k)\sigma\tau} (1 + \lambda_j \tau)^{-1-(n-k)} u_k \\
 &= e^{-n_0\tau\sigma} (1 + \lambda_j \tau)^{-(n_0+1)} \sum_{j=0}^{Q-1} \varpi_j y_{n-n_0}^{(j)},
 \end{aligned}$$

where  $y_{n-n_0}^{(j)} = \tau \sum_{k=0}^{n-n_0-1} (e^{\sigma\tau} (1 + \lambda_j \tau))^{-(n-n_0-k)} u_k$ , which satisfies (4.21).

A summary of the entire procedure of Fast Method II is given in Algorithm 2.

---

**Algorithm 2** Fast calculation of  $u_n^{(\alpha,\sigma)} = \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha,\sigma)} u_k$ , where  $\omega_k^{(\alpha,\sigma)}$  satisfies (3.2), (3.3), (3.4), or (3.6) (see also (4.10)).

---

1: Input: the fractional order  $\alpha$  and  $\sigma \geq 0$ , a time step size  $\tau > 0$ , a suitable positive integer  $n_0$ , the convolution weights  $\omega_n^{(\alpha,\sigma)}$  ( $0 \leq n \leq n_0$ ) defined by (3.2), (3.3), (3.4), or (3.6), the quadrature points  $\lambda_j$ , and weights  $\varpi_j$  (see (4.15) and its following paragraph).

2: Output:  ${}_F u_n^{(\alpha,\sigma)}$ .

- Step 1. Approximate the history part  ${}_H u_{n,n_0}^{(\alpha,\sigma)} = \tau^{-\alpha} \sum_{k=0}^{n-n_0-1} \omega_{n-k}^{(\alpha,\sigma)} u_k$  by

$$(4.20) \quad {}_F u_{n,n_0}^{(\alpha,\sigma)} = e^{-n_0\tau\sigma} (1 + \lambda_j \tau)^{-(n_0+1)\tau} \sum_{j=0}^{Q-1} \varpi_j y_{n-n_0}^{(j)},$$

where  $y_n^{(j)}$  is calculated by the following recurrence formula:

$$(4.21) \quad y_n^{(j)} = \frac{e^{-\tau\sigma}}{1 + \lambda_j \tau} \left( y_{n-1}^{(j)} + \tau u_{n-1} \right), \quad y_0^{(j)} = 0.$$

- Step 2. Calculate the local part  ${}_L u_{n,n_0}^{(\alpha,\sigma)}$  directly and let

$$(4.22) \quad {}_F u_n^{(\alpha,\sigma)} = {}_L u_{n,n_0}^{(\alpha,\sigma)} + {}_F u_{n,n_0}^{(\alpha,\sigma)}.$$


---

We now compare the computational performance and accuracy of the proposed fast methods against the direct method.

*Example 4.1.* Let  $u_n^{(\alpha,\sigma)} = \tau^{-\alpha} \sum_{k=0}^n \omega_{n-k}^{(\alpha,\sigma)} u_k$ , where  $u(t) = t + t^2$  and  $\omega_n^{(\alpha,\sigma)}$  satisfies (3.3). Compute  $u_n^{(\alpha,\sigma)}$  by the direct convolution method, Fast Method I, and Fast Method II.

Define the pointwise error  $e_n^{(r)}$  and the maximum pointwise error  $\|e^{(r)}\|_\infty$  by

$$e_n^{(r)} = |u_n^{(\alpha,\sigma)} - {}_F^{(r)} u_n^{(\alpha,\sigma)}| / |u_n^{(\alpha,\sigma)}|, \quad \|e^{(r)}\|_\infty = \max_{0 \leq n \leq T/\tau} e_n^{(r)}, \quad r = 1, 2,$$

where  ${}_F^{(1)} u_n^{(\alpha,\sigma)} = {}_F u_n^{(\alpha,\sigma)}$  is the fast solution from Fast Method I and  ${}_F^{(2)} u_n^{(\alpha,\sigma)}$  is the fast solution from Fast Method II.

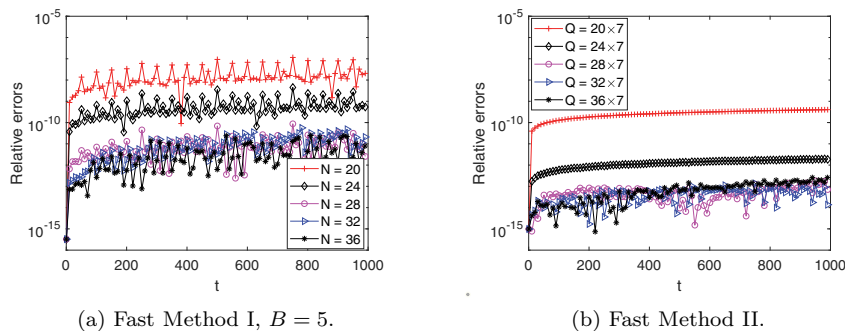


FIG. 2. The relative errors of Fast Method I and Fast Method II, Example 4.1:  $u(t) = t + t^2$ ,  $\tau = 0.01, \sigma = 0$ . The total number of quadrature points used for Fast Method I is  $N \times 7$  for  $n_0 = 50, \tau = 0.01$ , and  $T = 1000$ . The same number  $Q = N \times 7$  of quadrature points are used in Fast Method II for a fair comparison.

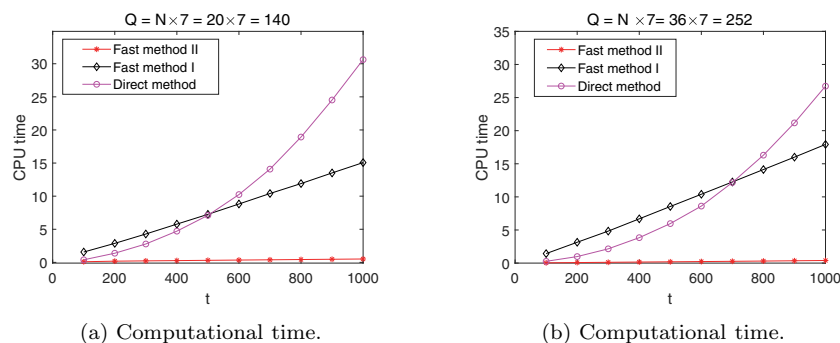


FIG. 3. The computational time of the direct method, Fast Method I, and Fast Method II, Example 4.1:  $u(t) = t + t^2$ ,  $\tau = 0.01, \sigma = 0$ . The total number of quadrature points: (a)  $Q = N \times 7 = 20 \times 7 = 140$ ; (b)  $Q = N \times 7 = 36 \times 7 = 252$ .

Figure 2 shows the relative errors of Fast Method I and Fast Method II for Example 4.1,  $\alpha = 0.5$ . We can see that Fast Method II shows better accuracy than Fast Method I when the same number of the quadrature points is used, which means Fast Method II saves memory and computational cost to achieve the same level of accuracy. Furthermore, Fast Method II requires only real arithmetic operations rather than complex arithmetic operations as in Fast Method I, which further reduces the computational cost.

Figure 3 depicts a comparison of the computational time of the direct method and the fast methods. We can see that both fast methods are more efficient than the direct method for long time computation, while Fast Method II is much faster than Fast Method I, since Fast Method II uses real arithmetic operations instead of complex arithmetic operations in Fast Method I.

Figure 4 further compares Fast Method II with Fast Method I. It is shown that the quadrature rule (4.15) is more accurate than the Talbot quadrature (4.3) for  $\alpha = 0.1, 0.9, 1.9$  but is less accurate for  $\alpha = -0.9$ . Sufficient numerical simulations show that (4.3) becomes more accurate as  $\alpha$  decreases, while (4.15) becomes more accurate as  $\alpha > -1$  increases.

In summary, Fast Method II is more efficient than Fast Method I for the fractional order  $\alpha \in (0, 2)$ , but these results are not displayed here.

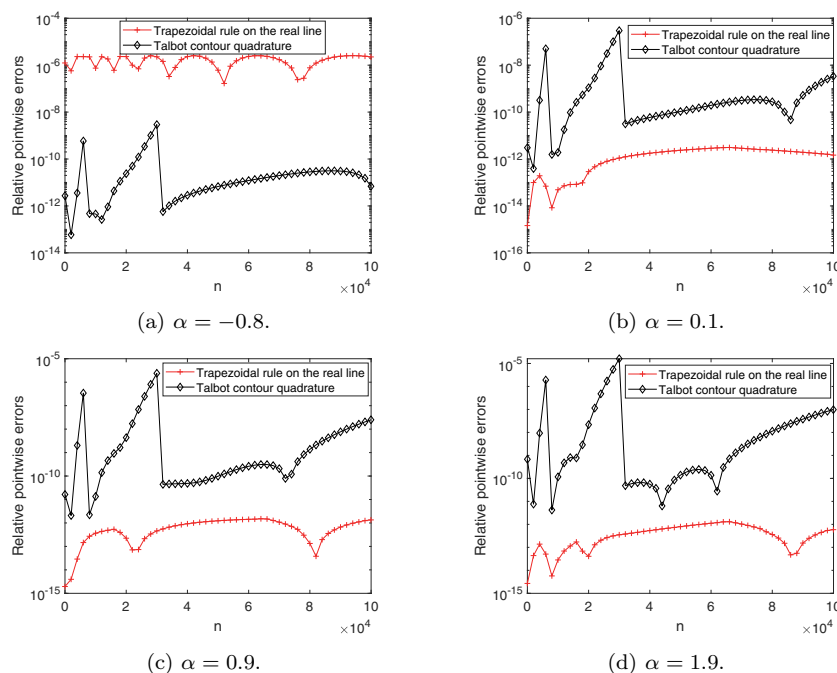


FIG. 4. The relative errors  $|\omega_n^{(\alpha,\sigma)} - \tilde{\omega}_n^{(\alpha,\sigma)}|/|\omega_n^{(\alpha,\sigma)}|$  (circles) and  $|\omega_n^{(\alpha,\sigma)} - \tilde{\omega}_n^{(\alpha,\sigma)}|/|\omega_n^{(\alpha,\sigma)}|$  (diamonds), Example 4.1,  $\sigma = 0.4$ ,  $B = 5$ ,  $\tau = 0.01$ . The number of quadrature points for discretizing each Talbot contour quadrature is  $N = 64$ ; the number of quadrature points for the trapezoidal rule is  $Q = N \times L = 64 \times 7 = 448$ .

In the following section, we apply Fast Method II to solve a number of time-fractional differential models.

**5. Numerical examples.** In this section, two examples are presented to verify the effectiveness of the present fast convolution. In the direct methods for solving FDEs in this section, the (tempered) fractional operators in the considered FDEs are always discretized by  $D_\tau^{\alpha,\sigma,\gamma,m,n}$  (see (3.1)) with the convolution quadrature weights defined by (3.3) with  $p = 2$ , i.e., GNGF-2 is applied. For convenience, we define

$$(5.1) \quad {}_F D_\tau^{\alpha,\sigma,\gamma,m,n} u = {}_F u_n^{(\alpha,\sigma)} + \tau^{-\alpha} \sum_{k=1}^m w_{n,k}^{(\alpha,\sigma)} (u_k - u_0) - b_n^{(\alpha,\sigma)} u_0,$$

where  $b_n^{(\alpha,\sigma)} = \tau^{-\alpha} \sum_{j=0}^n \omega_j^{(\alpha,\sigma)}$  and  ${}_F u_n^{(\alpha,\sigma)}$  is defined by (4.22).

All the algorithms are implemented using MATLAB 2017b, which were run in a 3.40 GHz PC having 16 GB RAM and Windows 7 operating system.

*Example 5.1.* Consider the following scalar fractional ODE:

$$(5.2) \quad D_{0,t}^{\alpha,\sigma} (u(t) - u(0)) = -u(t) + f(u, t), \quad u(0) = u_0, \quad t \in (0, T],$$

where  $0 < \alpha \leq 1$  and  $\sigma \geq 0$ .

Let  $U_n$  be the numerical solution of (5.2). The fully implicit fast method for solving (5.2) is given by

$$(5.3) \quad {}_F D_\tau^{\alpha,\sigma,\gamma,m,n} U = -U_n + f(U_n, t_n), \quad U_0 = u_0,$$

TABLE 1

The maximum error  $\|e\|_\infty$  for Example 5.1, Case I,  $\gamma_k = k\alpha$ ,  $\alpha = 0.5$ ,  $T = 10$ ,  $Q = 256$ , and  $\sigma = 0$ .

$\tau$	$m = 0$	Order	$m = 1$	Order	$m = 2$	Order	$m = 3$	Order
$2^{-5}$	4.8036e-2		4.7715e-4		2.5974e-5		1.9848e-5	
$2^{-6}$	3.5869e-2	0.4214	2.5331e-4	0.9135	1.2654e-5	1.0374	6.9865e-6	1.5064
$2^{-7}$	2.6373e-2	0.4437	1.3164e-4	0.9443	5.6593e-6	1.1609	2.3047e-6	1.6000
$2^{-8}$	1.9175e-2	0.4598	6.7484e-5	0.9640	2.3617e-6	1.2608	7.2179e-7	1.6749
$2^{-9}$	1.3830e-2	0.4714	3.4294e-5	0.9766	9.3879e-7	1.3309	2.1679e-7	1.7353

TABLE 2

The maximum error  $\|e\|_\infty$  for Example 5.1, Case I,  $\gamma_k = k\alpha$ ,  $\alpha = 0.5$ ,  $T = 10$ ,  $Q = 256$ , and  $\sigma = 0.5$ .

$\tau$	$m = 0$	Order	$m = 1$	Order	$m = 3$	Order	$m = 5$	Order
$2^{-5}$	4.8581e-2		2.1708e-4		4.1006e-5		3.6445e-5	
$2^{-6}$	3.6270e-2	0.4216	1.2066e-4	0.8473	1.1085e-5	1.8873	1.1950e-5	1.6087
$2^{-7}$	2.6622e-2	0.4462	6.4140e-5	0.9116	2.9336e-6	1.9178	3.5868e-6	1.7363
$2^{-8}$	1.9318e-2	0.4627	3.3264e-5	0.9473	7.6422e-7	1.9406	1.0161e-6	1.8196
$2^{-9}$	1.3908e-2	0.4740	1.7008e-5	0.9677	2.3723e-7	1.6877	2.7697e-7	1.8753

where  ${}_FD_{\tau}^{\alpha,\sigma,\gamma,m,n}$  is defined by (5.1), and  $m$  is the number of correction terms; see (3.1) and Appendix A for computing the correction weights.

We need to know  $U_k(1 \leq k \leq m)$  when (5.3) is applied. In this paper,  $U_k(1 \leq k \leq m)$  are obtained by solving (5.3) with a small step size  $2^{-7}\tau$  and  $m = 0$  or  $m = 1$  if there is at least one correction term. When we say the direct method is applied, we mean that  ${}_FD_{\tau}^{\alpha,\sigma,\gamma,m,n}$  in (5.3) is replaced by  $D_{\tau}^{\alpha,\sigma,\gamma,m,n}$ . The Newton method is applied to solve the nonlinear system (5.3) to obtain  $U_n$ .

The following two cases are considered in this example.

- Case I: For the linear case of  $f = 0$  and initial value  $u_0 = 1$ , the exact solution of (5.2) is

$$u(t) = E_{\alpha}(-t^{\alpha})e^{-\sigma t},$$

where  $E_{\alpha}(t)$  is the Mittag-Leffler function defined by  $E_{\alpha}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(k\alpha+1)}$ .

- Case II: Let  $f = u(1 - u^2)$  and  $u_0 = 1$ .

The maximum error is defined by

$$\|e\|_{\infty} = \max_{0 \leq n \leq T/\tau} |e_n|, \quad e_n = u(t_n) - U_n, \quad T = 10.$$

We first show that the use of the correction terms decreases the global error of the method significantly for Case I. From Tables 1–2, we can see that increasing the number of correction terms improves the accuracy significantly, and second-order accuracy is observed for some suitable  $m$ . Numerical simulations show that the inaccurate numerical solutions near the origin weakly affect the numerical solutions far from the origin. We show the numerical solutions at  $t = 10$  for  $\sigma = 0.2$  and  $0.5$  in Table 3. We can see that much better numerical solutions are obtained even if no correction term is added and second-order accuracy is observed using one or two correction terms for  $\sigma = 0.2$  and  $\sigma = 0.5$ .

For Case II, the explicit form of the analytical solution is unknown, and numerical solutions are shown in Figure 5. For a fixed fractional order  $\alpha = 0.3$ , the solution decays slower and attains a steady state as  $\sigma$  increases; see Figure 5(a). We observe similar behavior for  $\alpha = 0.8$ ; see Figure 5(b). For other fractional orders  $\alpha \in (0, 1)$ , we observe similar results, which are not shown here. Figures 6(a)–(c) show the difference

TABLE 3

The absolute error  $|e_n|$  at  $t = 10$ , Example 5.1, Case I,  $\gamma_k = k\alpha$ ,  $\alpha = 0.5$ ,  $Q = 256$ .

$\sigma = 0.2$						
$\tau$	$m = 0$	Order	$m = 1$	Order	$m = 2$	Order
$2^{-5}$	1.5240e-5		6.0747e-6		1.4693e-5	
$2^{-6}$	7.9796e-6	0.9335	1.5738e-6	1.9486	4.0865e-6	1.8462
$2^{-7}$	4.0758e-6	0.9692	4.0084e-7	1.9731	1.1023e-6	1.8903
$2^{-8}$	2.0580e-6	0.9859	1.0057e-7	1.9948	2.9090e-7	1.9220
$2^{-9}$	1.0335e-6	0.9937	2.4856e-8	2.0166	7.5573e-8	1.9446

$\sigma = 0.5$						
$\tau$	$m = 0$	Order	$m = 1$	Order	$m = 2$	Order
$2^{-5}$	2.0238e-5		6.0702e-5		4.1007e-5	
$2^{-6}$	5.0097e-6	2.0142	1.5645e-5	1.9560	1.0668e-5	1.9426
$2^{-7}$	1.2174e-6	2.0409	3.9910e-6	1.9709	2.7506e-6	1.9554
$2^{-8}$	2.8564e-7	2.0916	1.0114e-6	1.9804	7.0398e-7	1.9661
$2^{-9}$	6.1912e-8	2.2059	2.5521e-7	1.9866	1.7910e-7	1.9748

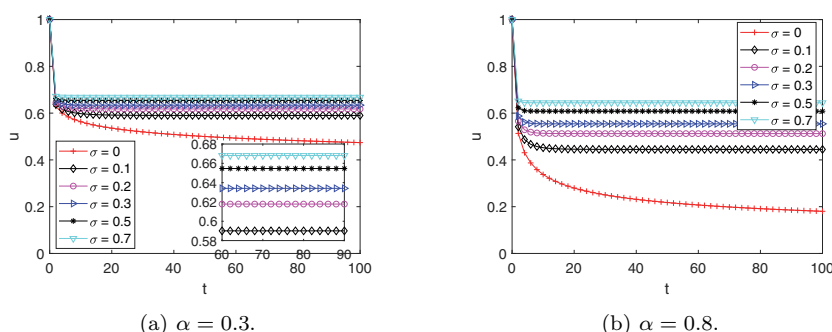


FIG. 5. Numerical solutions for Example 5.1, Case II,  $\tau = 0.001$ ,  $Q = 256$ .

between the fast solution and the direct solution. We can see that the two solutions are very close, which means the error caused from the trapezoidal (4.15) rule in Fast Method II is very small. Figure 6(d) shows the computational time of the fast method and the direct method, and we observe that the fast method really outperforms the direct method in efficiency and saves computational cost. The advantage of the fast method will be further displayed in the following example, solving a time-fractional activator-inhibitor system.

*Example 5.2.* Consider the fractional activator-inhibitor system [13]

$$(5.4) \quad \partial_t u(x, t) = \kappa f_1(u, v) + {}_{RL}D_{0,t}^{1-\alpha_1} \partial_x^2 u(x, t), \quad 0 \leq x \leq D,$$

$$(5.5) \quad \partial_t v(x, t) = \kappa f_2(u, v) + d {}_{RL}D_{0,t}^{1-\alpha_2} \partial_x^2 v(x, t), \quad 0 \leq x \leq D,$$

where  $u(x, t)$  and  $v(x, t)$  denote the concentrations of the activator and inhibitor, respectively,  $0 \leq \alpha_1 \leq 1$  is the anomalous diffusion exponent of the activator, and  $0 \leq \alpha_2 \leq 1$  is the anomalous diffusion exponent of the inhibitor,  $d$  is the ratio of the diffusion coefficients of inhibitor to activator, and  $\kappa > 0$  is a scaling variable that can be interpreted as the characteristic size of the spatial domain or as the relative strength of the reaction terms. The reaction kinetics is defined by the functions  $f_1(u, v)$  and  $f_2(u, v)$ .

In our following numerical test, we will consider the Turing pattern formation in the fractional activator-inhibitor model system described by system (5.4)–(5.5) with

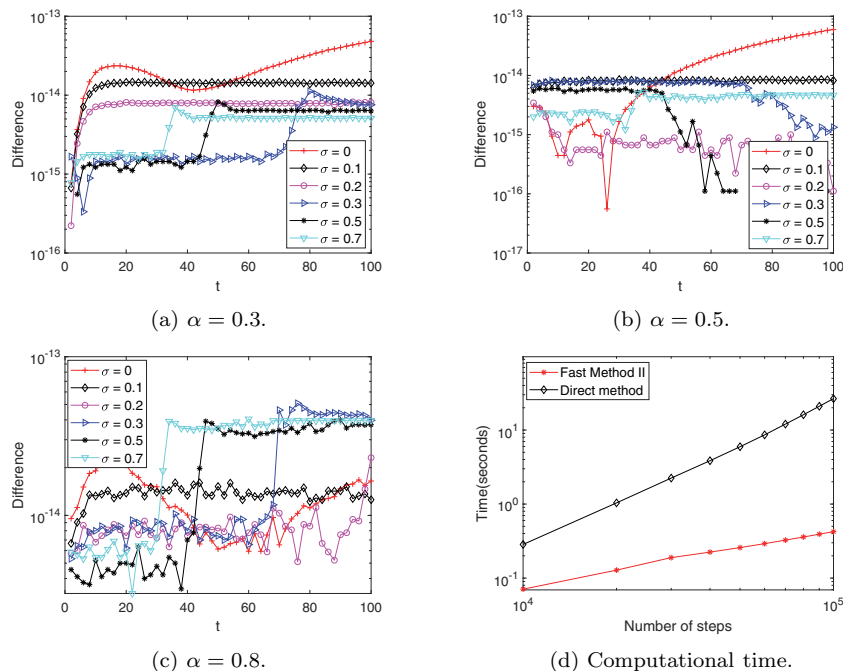


FIG. 6. (a)–(c) The difference between the numerical solutions of the direct method and the fast method. (d) The computational time of the fast method and the direct method; Example 5.1, Case II,  $\tau = 0.001$ ,  $Q = 256$ .

zero-flux boundary conditions at both ends of the spatial domain of length  $D$ , i.e.,

$$(5.6) \quad \partial_x u(0, t) = \partial_x v(0, t) = 0, \quad \partial_x u(D, t) = \partial_x v(D, t) = 0.$$

We apply a cubic finite element method to approximate the space of (5.4)–(5.5). For the time discretization, we apply a stabilized semi-implicit time-stepping method, i.e., the first-order time derivative is discretized by the second-order backward difference formula, the time-fractional derivative is discretized by the second-order GNGF, and the nonlinear term is approximated using a second-order extrapolation with a stabilization factor.

Let  $X_h$  be a cubic piecewise finite element space defined on the uniform grid  $\{x_j\}_{j=1}^{D/h}$ , where  $x_j = jh$ ,  $h$  is a space step size, and  $D/h$  is a positive integer. The numerical scheme for (5.4)–(5.6) is given by the following: For  $2 \leq n \leq n_T$ , find  $u_h^n, v_h^n \in X_h$ , such that

$$(5.7) \quad \begin{aligned} & (D_\tau^n u_h, w) + ({}_F D_\tau^{1-\alpha_1, 0, 0, n} \partial_x u_h, \partial_x w) + b_n^{(1-\alpha_1, 0)} (\partial_x u_h^0, \partial_x w) \\ & = \kappa(2F_1^{n-1} - F_1^{n-2}, w) - \kappa_1(u_h^n - 2u_h^{n-1} + u_h^{n-2}, w) \quad \forall w \in X_h, \end{aligned}$$

$$(5.8) \quad \begin{aligned} & (D_\tau^n v_h, w) + d({}_F D_\tau^{1-\alpha_2, 0, 0, n} \partial_x v_h, \partial_x w) + db_n^{(1-\alpha_2, 0)} (\partial_x v_h^0, \partial_x w) \\ & = \kappa(2F_2^{n-1} - F_2^{n-2}, w) - \kappa_2(v_h^n - 2v_h^{n-1} + v_h^{n-2}, w) \quad \forall w \in X_h, \end{aligned}$$

$$(5.9) \quad (u_h^0, w) = (u(0), w), \quad (u_h^1, w) = (u(0) + \tau \partial_t u(0), w) \quad \forall w \in X_h,$$

$$(5.10) \quad (v_h^0, w) = (v(0), w), \quad (v_h^1, w) = (v(0) + \tau \partial_t v(0), w) \quad \forall w \in X_h,$$

where  $F_1^n = f_1(u_h^n, v_h^n)$ ,  $F_2^n = f_2(u_h^n, v_h^n)$ ,  $u(t) = u(x, t)$ ,  $v(t) = v(x, t)$ ,  $\kappa_1$  and  $\kappa_2$  are



positive numbers that stabilize the time-stepping method,  $D_\tau^n u_h = (3u_h^n - 4u_h^{n-1} + u_h^{n-2})/(2\tau)$  and  ${}_FD_\tau^{\alpha,0,0,0,n}$  and  $b_n^{(\alpha,\sigma)}$  are defined in (5.1).

The Brusselator reaction kinetics will be considered for the fractional activator-inhibitor model system, where  $f_1$  and  $f_2$  are given by

$$(5.11) \quad f_1(u, v) = 2 - 3u + u^2v,$$

$$(5.12) \quad f_2(u, v) = 2u - u^2v.$$

The fractional activator-inhibitor model system defined by (5.4)–(5.6) and (5.11)–(5.12) has a homogeneous steady state of  $u^* = 2$  and  $v^* = 1$ . Standard linear stability analysis [29, 13, 12] reveals that in the case of standard diffusion  $\alpha_1 = \alpha_2 = 1$  nonhomogeneous steady states can occur if the value of  $d$  exceeds the critical value  $d^* \approx 23.31$ , while for  $d \leq d^*$  initial perturbations about the steady state decay to zero and no pattern results. The critical value of  $d^*$  for the fractional Brusselator reaction kinetics and the corresponding maximally excited modes over a range of  $\alpha$  are listed in [12].

We consider the same initial conditions as those in [12], which take the forms  $u(x, 0) = u^* + \epsilon r_1(x)$  and  $v(x, 0) = v^* + \epsilon r_2(x)$ . Three different types of perturbation are considered here: (i) random, where  $r_j(x)$  is a uniform random function on the interval  $[-1, 1]$ ; (ii) long-wavelength sinusoidal,  $r_1(x) = r_2(x) = \epsilon \sin(qx)$ , with  $q = 0.5$ ; (iii) short-wavelength sinusoidal,  $r_1(x) = r_2(x) = \epsilon \sin(qx)$ , with  $q = 5$ . We set  $\epsilon = 0.01$  in each case.

The parameters are taken as  $h = D/256$ ,  $D = 100$ ,  $\tau = 0.01$ ,  $\kappa_1 = \kappa_2 = 2$ , when the numerical method (5.7)–(5.10) is applied. We take the same values of  $\alpha_1, \alpha_2$  and  $d$  as in [12] in our simulations.

Figure 7 shows the full surface profiles for the concentrations of the activator  $u$  (left column) and inhibitor  $v$  (right column) with randomly perturbed initial conditions and  $\alpha_1 = \alpha_2 = \alpha$ , where the activator shows similar behavior as the inhibitor. We obtain similar results as those in [12]: (i) The concentrations of the activator and inhibitor both fluctuate about the homogeneous steady-state values. (ii) A spatiotemporal pattern develops on or before  $t = 500$ . (iii) The surface profiles become more spatially rough and/or less stationary as the fractional order  $\alpha$  decreases.

Figure 8 shows the surface density plots of  $u(x, t) \geq u^*$  (black) and  $u(x, t) < u^*$  (white) for the Brusselator model with sinusoidally perturbed initial conditions (ii): long-wavelength sinusoidally perturbations (left column) and short-wavelength sinusoidally perturbations (right column). We observe the same results as shown in [12], but we use finer spatial resolution to obtain more accurate solutions. For both long-wavelength sinusoidally perturbations and short-wavelength sinusoidally perturbations, similar patterns are observed after  $t = 500$  for the same parameters  $d$  and fractional orders  $\alpha_1 = \alpha_2$ .

Next, we choose different fractional orders  $\alpha_1 = 0.5$  (anomalous subdiffusion in the activator  $u(x, t)$ ) and  $\alpha_2 = 1$  (standard diffusion in the inhibitor  $v(x, t)$ ). In such a case, turning-instability-induced pattern formation might occur for any  $d > 0$  (see [12, 13]). We perform a number of numerical simulations of the fractional activator-inhibitor model with anomalous diffusion in the activator and standard diffusion in the inhibitor over a range of parameters. Figure 9 shows the surface profiles of the activator and inhibitor of the fractional Brusselator model with short-wavelength sinusoidally perturbed initial conditions (iii). Obviously, the activator and inhibitor display different fluctuations about the homogeneous steady-state solution. We obtain similar results as those reported in [12].

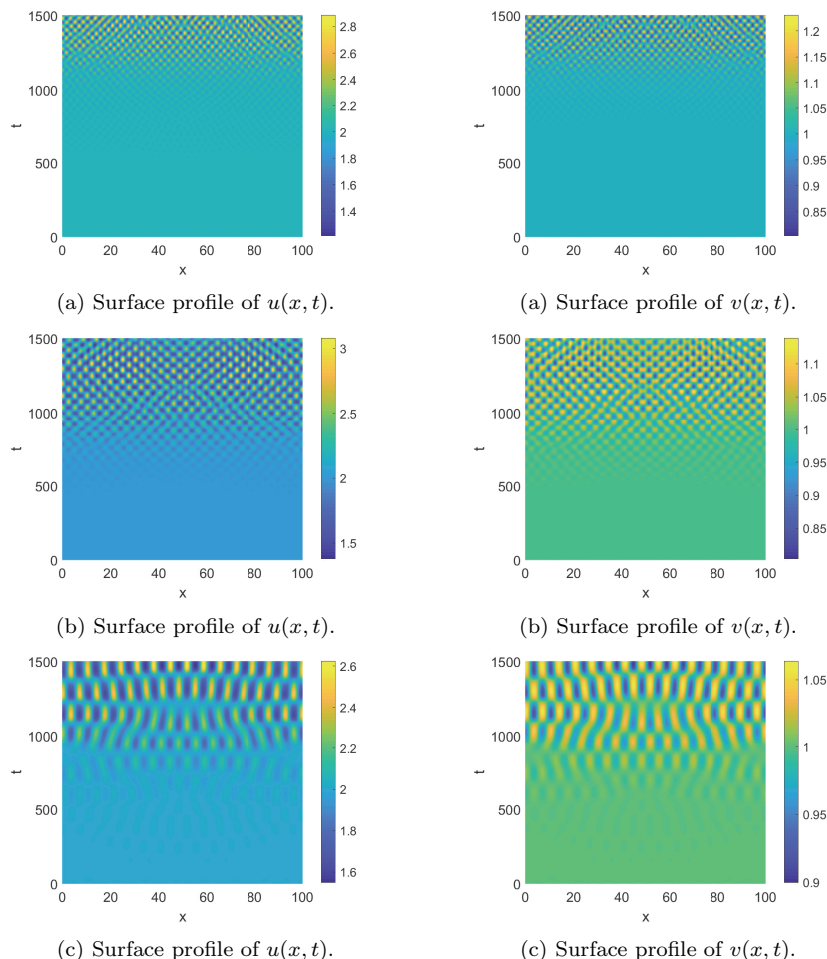


FIG. 7. Fractional Brusselator model with randomly perturbed initial conditions (i), Example 5.2: (a)  $\alpha_1 = \alpha_2 = 0.2$ ,  $d = 9$ ; (b)  $\alpha_1 = \alpha_2 = 0.5$ ,  $d = 17$ ; (c)  $\alpha_1 = \alpha_2 = 0.8$ ,  $d = 23$ .

Finally in this section, we show the efficiency and accuracy of the fast method. Figure 10(a) displays the difference  $\|Fu_h(t) - Du_h(t)\|_\infty$  of the fast solution and the direct solution of the fractional Brusselator model with long-wavelength sinusoidally perturbed initial conditions (ii), where  $Fu_h$  is the fast solution obtained from (5.7)–(5.10), and  $Du_h$  is the direct method solution that is obtained from (5.7)–(5.10) with  $FD_\tau^{1-\alpha,0,0,0,n}$  replaced by the direct calculation method  $D_\tau^{1-\alpha,0,0,0,n}$ . We choose 256 quadrature points in the trapezoidal rule used in the fast method, and an accuracy of  $10^{-9}$  is achieved (more accurate results can be obtained if we increase the number of quadrature points  $Q$ , but these results are not shown here). The most obvious observation is that the computational time of the fast method increases linearly, while the computational cost of the direct method increases quadratically; see Figure 10(b). For the case shown in Figure 10(b), the computational times of the fast method and the direct method are about 4000 seconds (about one hour) and 87000 seconds (about one day and two hours), respectively.

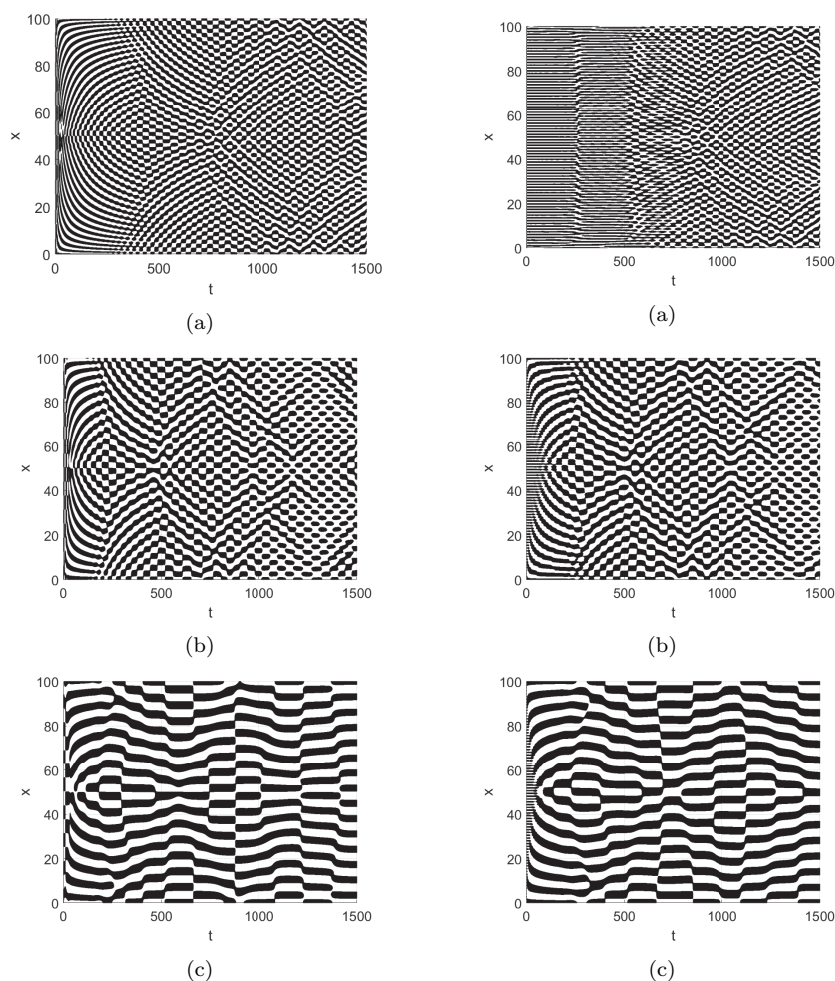


FIG. 8. Surface density plots of fractional Brusselator model for  $u(x, t)$  with long-wavelength (left column) and short-wavelength (right column) sinusoidally perturbed initial conditions (ii), Example 5.2: (a)  $\alpha_1 = \alpha_2 = 0.2$ ,  $d = 9$ ; (b)  $\alpha_1 = \alpha_2 = 0.5$ ,  $d = 17$ ; (c)  $\alpha_1 = \alpha_2 = 0.8$ ,  $d = 23$ .

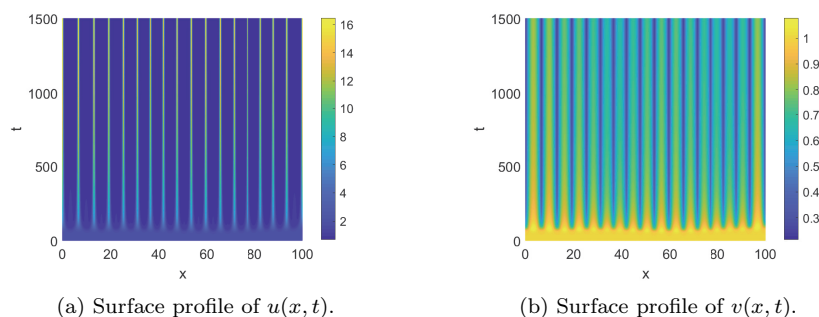


FIG. 9. Fractional Brusselator model with short-wavelength sinusoidally perturbed initial conditions (iii), Example 5.2:  $\alpha_1 = 0.5$ ,  $\alpha_2 = 1.0$ ,  $d = 10$ .

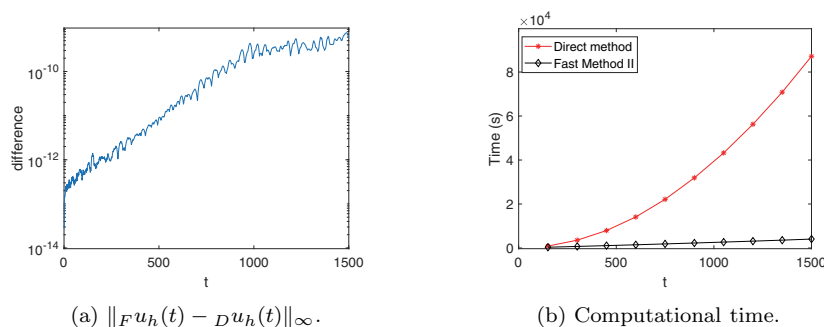


FIG. 10. (a) The maximum difference  $\|Fu_h(t) - Du_h(t)\|_\infty$  between the direct solution and the fast solution of the fractional Brusselator model with long-wavelength sinusoidally perturbed initial conditions (ii); (b) the computational time of the fast method and the direct method; Example 5.2,  $\alpha_1 = \alpha_2 = 0.5$ ,  $d = 17$ .

**6. Conclusion and discussion.** In this work, we first prove the equivalence between the tempered fractional derivative operator and the Hadamard finite-part integral. The interpretation of the tempered fractional derivative in terms of the finite-part integral makes a direct and obvious extension of Lubich's FLMMs to both the tempered fractional integral and derivative operators, which greatly simplifies the method in [5].

We then propose two fast methods, Fast Method I and Fast Method II, to approximate the discrete convolution  $\sum_{j=0}^n \omega_{n-j}^{(\alpha, \sigma)} u_j$  in the considered FLMM. Both methods are effective and efficient. Fast Method I can be seen as a direct extension of the fast method in [39] ( $\sigma = 0$ ) to the tempered fractional operator ( $\sigma > 0$ ). In Fast Method I, the convolution weight  $\omega_n^{(\alpha, \sigma)}$  is represented by a contour integral, which is approximated by a local contour quadrature for different  $n$ . The use of the local approximation for  $\omega_n^{(\alpha, \sigma)}$  makes the implementation of Fast Method I complicated. Furthermore, complex arithmetic operations are performed in Fast Method I, which leads to slightly larger roundoff errors; see Figure 2(a).

In order to overcome the drawbacks of Fast Method I, we propose Fast Method II, which has the following advantages:

- In Fast Method II, the convolution weight  $\omega_n^{(\alpha, \sigma)}$  is expressed by an integral on the real line instead of the contour integral in the complex plane in Fast Method I. A uniform approximation is derived to approximate this integral on the real line, which makes the implementation of Fast Method II much easier and simpler than that of Fast Method I.
- Only real arithmetic operations are performed in Fast Method II.
- In Fast Method I, an ODE of the form  $y'(t) = \lambda y(t) + u(t)$  is solved by the backward Euler method (see also (4.8)). However, the coefficient  $\lambda$  may have positive real part if the Talbot or hyperbolic contour quadrature (see, e.g., [32, 39]) is applied, which may affect the stability of Fast Method I. We always perform a stable recurrence relation (4.21) in Fast Method II, which avoids a possible negative effect caused by the positive real part of  $\lambda$  in Fast Method I.

In summary, Fast Method II outperforms Fast Method I in terms of both accuracy and efficiency when  $\alpha > 0$  and yields easier implementation, which is also verified by numerical simulations in this work. The disadvantage of Fast Method II is that it shows less accurate numerical results when  $-1 < \alpha < 0$  and  $\alpha \rightarrow -1$ , and it cannot work when  $\alpha \geq -1$  in the present framework. Fast Method I still works well,

but the most obvious disadvantage is its complicated implementation. Finally, we remark that the accuracy of the present fast method can be made independent of the time step size  $\tau$  when it is applied to solve time-fractional PDEs. We just need to rewrite  $\tau^{-\alpha} \sum_{j=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k$  into  $(\tau/\tau_0)^{-\alpha} [(\tau_0)^{-\alpha} \sum_{j=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k]$  and perform the fast calculation on  $(\tau_0)^{-\alpha} \sum_{j=0}^n \omega_{n-k}^{(\alpha, \sigma)} u_k$ , where  $\tau_0 > 0$  can be a fixed number.

The code for numerical simulations in this paper can be found at <https://github.com/fanhaizeng/fast-method-for-fractional-operators-generating-functions>.

**Appendix A. Calculation of the starting weights.** We show how to efficiently calculate the starting weights  $w_{n,k}^{(\alpha, \sigma)}$  in (3.1). For each  $n > 0$ , the starting weights satisfy the following system of equations:

$$(A.1) \quad \sum_{k=1}^m w_{n,k}^{(\alpha, \sigma)} t_k^{\gamma_j} = P.V.(k_{-\alpha, \sigma} * g_j)(t_n) - \sum_{k=0}^n \omega_k^{(\alpha, \sigma)} t_{n-k}^{\gamma_j}, \quad 1 \leq j \leq m,$$

where  $0 < \gamma_j < \gamma_{j+1}$  and

$$k_{\alpha, \sigma}(t) = \frac{t^{\alpha-1} e^{-\sigma t}}{\Gamma(\alpha)}, \quad g_j(t) = t^{\gamma_j}.$$

Clearly,  $\sum_{k=0}^n \omega_k^{(\alpha, \sigma)} t_{n-k}^{\gamma_j}$  in (A.1) can be efficiently and accurately approximated by Fast Method II in section 4, and the computational cost is  $O(Qn_T)$ . It remains to calculate  $P.V.(k_{-\alpha, \sigma} * g_j)(t_n)$ , which is equal to  $\frac{\Gamma(\gamma_j+1)}{\Gamma(\gamma_j+1-\alpha)} t_n^{\gamma_j-\alpha}$  for  $\sigma = 0$ . Next, we show how to calculate  $P.V.(k_{-\alpha, \sigma} * g_j)(t_n)$  for  $\sigma > 0$ .

Denote  $\mathcal{L}[g](\lambda)$  as the Laplace transform of  $g(t)$ . Using the property

$$\mathcal{L}[k_{-\alpha, \sigma} * g_j](\lambda) = \mathcal{L}[k_{-\alpha, \sigma}](\lambda) \cdot \mathcal{L}[g_j](\lambda) = (\lambda + \sigma)^\alpha \cdot \Gamma(\gamma_j + 1) \lambda^{-\gamma_j-1}$$

and the inverse Laplace transform, we can rewrite  $P.V.(k_{-\alpha, \sigma} * g_j)(t)$  as

$$(A.2) \quad \begin{aligned} P.V.(k_{-\alpha, \sigma} * g_j)(t) &= \frac{1}{2\pi i} \int_{\mathcal{C}} \mathcal{L}[k_{-\alpha, \sigma} * g_j](\lambda) e^{\lambda t} d\lambda \\ &= \frac{\Gamma(\gamma_j + 1)}{2\pi i} \int_{\mathcal{C}} \frac{(\lambda + \sigma)^\alpha}{\lambda^{\gamma_j+1}} e^{\lambda t} d\lambda. \end{aligned}$$

The above contour integral can be approximated efficiently and accurately by the numerical inverse Laplace transform. In this work, we apply the optimal Talbot contour quadrature [36], which is given by

$$(A.3) \quad \frac{1}{2\pi i} \int_{\mathcal{C}} \mathcal{L}[k_{-\alpha, \sigma} * g_j](\lambda) e^{\lambda t} d\lambda \approx 2\text{Im} \left( \sum_{j=0}^{N-1} w_j \mathcal{L}[k_{-\alpha, \sigma} * g_j](\lambda_j) e^{\lambda_j t} \right),$$

where

$$(A.4) \quad \lambda_j = z(\theta_j, N/t), \quad w_j = \partial_\theta z(\theta_j, N/t), \quad \theta_j = (2j+1)\pi/(2N),$$

with  $z(\theta, \mu) = \mu(-0.4814 + 0.6443(\theta \cot(\theta) + i0.5653\theta))$ ; see also (4.4). Theoretically, the convergence of (A.3) is about  $O(e^{-1.36N})$ , that is to say, we can use  $N = 27$  to achieve machine precision, which is also applied in this paper. Clearly, the computational cost of (A.3) for  $t = t_n, 1 < n \leq n_T$ , is  $O(Nn_T)$ . Once the right-hand side of (A.1) is known, the computational cost for solving the linear system to get  $w_{n,k}^{(\alpha)}$  is

$O(m^3 n_T)$  by QR factorization. In summary, the computational cost for calculating the starting weights is  $O(Qn_T) + O(Nn_T) + O(m^3 n_T)$ .

One computational difficulty is that the linear system (A.1) is ill-conditioned when  $m$  is suitably large [7, 20], which may yield inaccurate starting quadrature weights  $w_{n,k}^{(\alpha,\sigma)}$ . However, we find that a small number of correction terms (e.g.,  $m \leq 5$ ) can achieve highly accurate solutions in numerical simulations, which was illustrated in detail in [40].

**Appendix B. Derivation of (4.1).** We recall the derivation of (4.1); see also [32, 39]. For a stable FLMM, the corresponding generating function  $\omega^{(\alpha,\sigma)}(z)$  is analytical for  $|z| < 1$ , which is important to yield (4.1). For the generating function defined by (3.3),  $F_\omega(\lambda)$  defined by (4.2) is analytical on the whole plane for  $\alpha \in \mathbb{R}$ . For (3.2) or (3.4),  $F_\omega(\lambda)$  is analytical on the whole plane except for some weak singular points when  $\alpha > -1$ ; the number of the weak singular points is at most  $p$ .

Next, we show how to derive (4.1).

- Given a  $\delta(z)$ , define  $e_n(z)$  as

$$(B.1) \quad (\delta(\xi) - z)^{-1} = \sum_{n=0}^{\infty} e_n(z) \xi^n.$$

- Define  $\tilde{F}_\omega(\lambda)$  such that

$$(B.2) \quad \tilde{F}_\omega(\delta(\xi)/\tau) = \frac{1}{\tau^\alpha} \sum_{n=0}^{\infty} \omega_n^{(\alpha,0)} \xi^n = \frac{1}{\tau^\alpha} \omega^{(\alpha,0)}(\xi).$$

- By Cauchy's integral formula and (B.1), we obtain

$$(B.3) \quad \tilde{F}(\delta(\xi)/\tau) = \frac{1}{2\pi i} \int_{\mathcal{C}} \frac{\tilde{F}_\omega(\lambda)}{\delta(\xi)/\tau - \lambda} d\lambda = \sum_{n=0}^{\infty} \left( \frac{\tau}{2\pi i} \int_{\mathcal{C}} e_n(\tau\lambda) \tilde{F}_\omega(\lambda) d\lambda \right) \xi^n,$$

where  $\mathcal{C}$  is a suitable contour.

- Comparing (B.2) and (B.3), we derive

$$(B.4) \quad \omega_n^{(\alpha,0)} = \frac{\tau^{1+\alpha}}{2\pi i} \int_{\mathcal{C}} e_n(\lambda\tau) \tilde{F}_\omega(\lambda) d\lambda.$$

We are in a position to choose  $\delta(z)$  to obtain the specific form of (B.4). The choice of  $\delta(z)$  is very important and may determine the complexity of the algorithm. For example, in [32], for the FBDF of order  $p$ ,  $\delta(z) = \sum_{k=1}^p \frac{1}{k} (1-z)^k$ ,  $\tilde{F}_\omega(\lambda) = \lambda^\alpha$  is very simple, but  $e_n(z)$  seems difficult to obtain for  $p > 2$ . What we find is that we can always choose  $\delta(z) = 1 - z$ , which yields  $e_n(z) = (1 - z)^{-1-n}$  and

$$(B.5) \quad \tilde{F}_\omega(\lambda) = \frac{1}{\tau^\alpha} \omega^{(\alpha,0)}(\delta^{-1}(\tau\lambda)) = \frac{1}{\tau^\alpha} \omega^{(\alpha,0)}(1 - \tau\lambda).$$

Equation (4.1) follows immediately from the above equation, which simplifies the method in the previous work; see, e.g., [3, 32].

**Appendix C. Determine  $x_{\min}$  and  $x_{\max}$  in (4.16).** The goal in this section is to find  $x_{\min}$  and  $x_{\max}$  such that

$$(C.1) \quad \left| \omega_n^{(\alpha,0)} \tau^{-1-\alpha} - \int_{x_{\min}}^{x_{\max}} \phi_n(x) dx \right| = \left| \int_{-\infty}^{x_{\min}} \phi_n(x) dx + \int_{x_{\max}}^{\infty} \phi_n(x) dx \right| \lesssim \epsilon |\omega_n^{(\alpha,0)} \tau^{-1-\alpha}| \lesssim \epsilon (n\tau)^{-1-\alpha},$$

where  $\epsilon$  is a given precision and  $\omega_n^{(\alpha,0)} = O(n^{-\alpha-1})$  is used.

The asymptotic expansion of  $(1 + \tau e^x)^{-n}$  for  $|x| \rightarrow \infty$  is given by  $\exp(-n\tau e^x)$  when  $n \geq n_0$ . For  $x \leq 0$ , we can easily obtain

$$(C.2) \quad |\phi_n(x)| \lesssim \exp((1 + \alpha)x - (n + 1)\tau e^x) \lesssim \exp((1 + \alpha)x),$$

which leads to

$$\left| \int_{-\infty}^x \phi_n(s) ds \right| \lesssim \int_{-\infty}^x \exp((1 + \alpha)s) ds = \frac{\exp((1 + \alpha)x)}{1 + \alpha}.$$

Letting  $\exp((1 + \alpha)x) \leq \epsilon(n\tau)^{-1-\alpha}$  yields

$$x \leq (1 + \alpha)^{-1} \log(\epsilon(n\tau)^{-\alpha-1}) = (1 + \alpha)^{-1} \log(\epsilon) - \log(n\tau).$$

Hence,  $x_{\min}$  in (4.16) can be derived by letting  $n = n_T$  in the right-hand side of the above inequality.

For  $x > 0$ , we can similarly obtain  $|\phi_n(x)| \lesssim \exp((p + \alpha)x - (n + 1)\tau e^x)$  and

$$(C.3) \quad \left| \int_x^\infty \phi_n(s) ds \right| \lesssim \int_x^\infty \exp((p + \alpha)s - (n + 1)\tau e^s) ds \\ \lesssim \exp((p + \alpha)x - (n + 1)\tau e^x),$$

where we have used the asymptotic expansions of the incomplete Gamma function [34]. Now we need to find  $x_{\max}$  such that  $\exp((p + \alpha)x - (n + 1)\tau e^x) \lesssim \epsilon(n\tau)^{-1-\alpha}$  for all  $x \geq x_{\max}$  and  $n \geq n_0$ . For  $0 < \theta < 1$  and  $n \geq n_0$ , we also have

$$(C.4) \quad \exp((p + \alpha)x - (n + 1)\tau e^x) \leq \exp((p + \alpha)x - (1 - \theta)n\tau e^x) \exp(-\theta n\tau e^x) \\ \lesssim \exp(-\theta n\tau e^x).$$

Therefore, we can let  $\exp(-\theta n\tau e^x) \leq \epsilon(n\tau)^{-1-\alpha}$  and obtain

$$(C.5) \quad x \geq \log\left(\frac{-\log(\epsilon) + (1 + \alpha)\log(n\tau)}{\theta n\tau}\right).$$

For  $n \geq n_0$  and  $n\tau \geq \epsilon\epsilon^{\frac{1}{1+\alpha}}$ , we find that the maximum value of the right-hand side of (C.5) is

$$(C.6) \quad \log\left(\frac{-\log(\epsilon) + (1 + \alpha)\log(n_0\tau)}{\theta n_0\tau}\right).$$

Letting  $\theta = 1/2$  in (C.6) yields  $x_{\max}$  in (4.16). Numerical simulations show that (C.6) works very well for  $\theta = 1/2$ ,  $\epsilon = 10^{-16}$ , and  $0 < \alpha < 2$ .

**Appendix D. Determine  $a$  in Theorem 4.1.** We discuss how to determine  $a$  in Theorem 4.1. It is very easy to verify that  $\phi_n(x) \rightarrow 0$  as  $|x| \rightarrow \infty$  if the generating functions (3.2)–(3.4) are used for  $n > n_0$ . We need to find  $a > 0$  such that  $\phi_n(x)$  is analytical for  $|x| < a$  and  $\int_{-\infty}^\infty |\phi_n(x + ib)| dx \leq M$  for  $b \in (-a, a)$ . It is easy to obtain

$$(D.1) \quad \phi_n(x) = c_\alpha e^{(1+\alpha)x} F_\omega(-e^x) (1 + e^x \tau)^{-1-n}, \\ |\phi_n(x + ib)| = |c_\alpha| e^{(1+\alpha)x} |F_\omega(-e^{x+ib})| (1 + \tau^2 e^{2x} + 2\tau \cos(b)e^x)^{-\frac{n+1}{2}} \\ (D.2) \quad \leq |c_\alpha| e^{(1+\alpha)x} |F_\omega(-e^{x+ib})| (1 + \tau^2 e^{2x})^{-\frac{n+1}{2}}, \quad b \in (-\pi/2, \pi/2),$$

where  $c_\alpha = -\frac{\sin(\alpha\pi)}{\pi}$ . The following two facts can be easily verified:

TABLE 4  
The possible poles of  $F_\omega(-e^x)$  for the FBDF- $p$ ,  $p > 1$ .

$p$	Possible poles	$a$
2	$-\log \tau + 0.6931$	0
3	$-\log \tau + 0.5493 \pm 1.1230i$	1.1230
4	$-\log \tau + 0.4515 \pm 1.6626i$ $-\log \tau + 0.4833$	0
5	$-\log \tau + 0.3838 \pm 1.9784i$ $-\log \tau + 0.4209 \pm 0.6564i$	0.6564
6	$-\log \tau + 0.3344 \pm 2.1850i$ $-\log \tau + 0.3715 \pm 1.0874i$ $-\log \tau + 0.38$	0

(i) For  $\alpha > -1$ ,  $\int_{-\infty}^{\infty} |\phi_n(x + ib)| dx \leq M$  exists for  $n > n_0$ .

(ii)  $\phi_n(x)$  has poles  $x = \log(-1/\tau) = \log(1/\tau) + i(2k+1)\pi$ .

Next, we determine  $a$  by analyzing the property of  $F_\omega(-e^x)$ .

- For the GNGF- $p$ ,  $F_\omega(-e^x)$  is analytical for any  $x$ . Combining (D.2), we can choose  $a = \pi/2$ .
- For the FBDF- $p$ ,  $p > 1$ ,  $F_\omega(-e^x)$  may have poles for a noninteger  $\alpha$ , which can be derived by solving the following equation:

$$F_\omega(-e^x) = \left(1 + \frac{1}{2}(-\tau e^x) + \frac{1}{3}(-\tau e^x)^2 + \cdots + \frac{1}{p}(-\tau e^x)^{p-1}\right)^\alpha = 0.$$

Table 4 shows the possible poles of  $F_\omega(-e^x)$  for  $2 \leq p \leq 6$  and the corresponding  $a$ .

- For the fractional trapezoidal rule, the poles of  $F_\omega(-e^x) = 2^{-\alpha}(2 + \tau e^x)^{-\alpha}$  are  $x = \log(2/\tau) + i(2k+1)\pi$ , and  $a$  can be chosen as  $a = \pi/2$ .

From Table 4, we find that  $a = 0$  for the FBDF of order 2, 4, and 6. However, (4.15) still works, which can be explained from the asymptotic error expansion of the composite trapezoidal rule (see [35])

$$I_Q(v) = \frac{\Delta x}{2} \sum_{j=0}^{Q-1} [v(x_j) + v(x_{j+1})], \quad x_j = x_0 + j\Delta x, \Delta x = (x_Q - x_0)/Q$$

for the approximation of the integral  $I(v) = \int_{x_0}^{x_Q} v(x)dx$ . The asymptotic error expansion of the above composite trapezoidal rule is given by (see, e.g., [35])

$$(D.3) \quad I_Q(v) - I(v) \sim \Delta x^2 \frac{B_2}{2!} [v'(x_Q) - v'(x_0)] + \Delta x^4 \frac{B_4}{4!} [v'''(x_Q) - v'''(x_0)] + \cdots,$$

where  $v(x)$  is sufficiently smooth in  $[x_0, x_Q]$  and  $\{B_k\}$  are the Bernoulli numbers ( $B_2 = 1/6, B_4 = -1/30, B_6 = 1/42, \dots$ ).

For our case, (4.15) can be derived from  $I_Q(\phi_n) + \frac{\Delta x}{2}(\phi_n(x_0) - \phi_n(x_Q))$ , where  $\phi_n(x)$  is smooth for  $x \in [x_0, x_Q]$ ,  $x_0 = x_{\min}$ , and  $x_Q = x_{\max}$ . It is not difficult to find that  $|\frac{d^k}{dx^k} \phi_n(x)|$  decays exponentially to zero as  $|x| \rightarrow \infty$ . Our numerical simulations show that  $|\frac{d^k}{dx^k} \phi_n(x_{\min})|$  and  $|\frac{d^k}{dx^k} \phi_n(x_{\max})|$  are very small for  $0 \leq k \leq r$ , which yields high accuracy of (4.15) by (D.3).

**Acknowledgment.** The authors wish to thank the anonymous referees for their constructive comments and suggestions, which have greatly improved the quality of this paper.



## REFERENCES

- [1] D. BAFFET AND J. S. HESTHAVEN, *High-order accurate adaptive kernel compression time-stepping schemes for fractional differential equations*, J. Sci. Comput., 72 (2017), pp. 1169–1195.
- [2] L. BANJAI AND M. LÓPEZ-FERNÁNDEZ, *Efficient high order algorithms for the fractional integral and associated FDEs*, Numer. Math., 141 (2019), pp. 289–317.
- [3] L. BANJAI, M. LÓPEZ-FERNÁNDEZ, AND A. SCHÄDLE, *Fast and oblivious algorithms for dissipative and two-dimensional wave equations*, SIAM J. Numer. Math., 55 (2017), pp. 621–639.
- [4] A. CARTEA AND D. DEL CASTILLO-NEGRET, *Fractional diffusion models of option prices in markets with jumps*, Phys. A, 374 (2007), pp. 749–763.
- [5] M. CHEN AND W. DENG, *Discretized fractional substantial calculus*, ESAIM Math. Model. Numer. Anal., 49 (2015), pp. 373–394.
- [6] W. DENG, L. ZHAO, AND Y. WU, *Fast predictor-corrector approach for the tempered fractional ordinary differential equations*, Numer. Algorithms, 74 (2017), pp. 717–754.
- [7] K. DIETHELM, J. M. FORD, N. J. FORD, AND M. WEILBEER, *Pitfalls in fast numerical solvers for fractional differential equations*, J. Comput. Appl. Math., 186 (2006), pp. 482–503.
- [8] J. GAJDA AND M. MAGDZIARZ, *Fractional Fokker-Planck equation with tempered  $\alpha$ -stable waiting times: Langevin picture and computer simulation*, Phys. Rev. E, 82 (2010), 011117.
- [9] L. GALEONE AND R. GARRAPPA, *Fractional Adams-Moulton methods*, Math. Comput. Simulation, 79 (2008), pp. 1358–1367.
- [10] E. HAIRER, CH. LUBICH, AND M. SCHLICHT, *Fast numerical solution of nonlinear Volterra convolution equations*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 532–541.
- [11] A. HANYGA, *Wave propagation in media with singular memory*, Math. Comput. Model., 34 (2001), pp. 1399–1421.
- [12] B. HENRY, T. LANGLANDS, AND S. WEARNE, *Turing pattern formation in fractional activator-inhibitor systems*, Phys. Rev. E (3), 72 (2005), 026101.
- [13] B. I. HENRY AND S. L. WEARNE, *Existence of Turing instabilities in a two-species fractional reaction-diffusion system*, SIAM J. Appl. Math., 62 (2002), pp. 870–887.
- [14] 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., 21 (2017), pp. 650–678.
- [15] C. LI, W. DENG, AND L. ZHAO, *Well-posedness and numerical algorithm for the tempered fractional ordinary differential equations*, Discrete Contin. Dyn. Syst. Ser. B, 24 (2019), pp. 1989–2015.
- [16] C. LI AND F. ZENG, *The finite difference methods for fractional ordinary differential equations*, Numer. Funct. Anal. Optim., 34 (2013), pp. 149–179.
- [17] J.-R. LI, *A fast time stepping method for evaluating fractional integrals*, SIAM J. Sci. Comput., 31 (2010), pp. 4696–4714.
- [18] M. LÓPEZ-FERNÁNDEZ, C. LUBICH, C. PALENCIA, AND A. SCHÄDLE, *Fast Runge-Kutta approximation of inhomogeneous parabolic equations*, Numer. Math., 102 (2005), pp. 277–291.
- [19] M. LÓPEZ-FERNÁNDEZ, C. LUBICH, AND A. SCHÄDLE, *Adaptive, fast, and oblivious convolution in evolution equations with memory*, SIAM J. Sci. Comput., 30 (2008), pp. 1015–1037.
- [20] C. LUBICH, *Discretized fractional calculus*, SIAM J. Math. Anal., 17 (1986), pp. 704–719.
- [21] C. LUBICH AND A. SCHÄDLE, *Fast convolution for nonreflecting boundary conditions*, SIAM J. Sci. Comput., 24 (2002), pp. 161–182.
- [22] O. MAROM AND E. MOMONIAT, *A comparison of numerical solutions of fractional diffusion models in finance*, Nonlinear Anal. Real World Appl., 10 (2009), pp. 3435–3442.
- [23] W. MCLEAN, *Fast summation by interval clustering for an evolution equation with memory*, SIAM J. Sci. Comput., 34 (2012), pp. A3039–A3056.
- [24] W. MCLEAN, *Exponential sum approximations for  $t^{-\beta}$* , in Contemporary Computational Mathematics—A Celebration of the 80th Birthday of Ian Sloan, Springer, New York, 2018, pp. 911–930.
- [25] M. MEERSCHAERT, F. SABZIKAR, M. PHANIKUMAR, AND A. ZELEKE, *Tempered fractional time series model for turbulence in geophysical flows*, J. Stat. Mech. Theory Exp., 14 (2014), pp. 1742–5468.
- [26] M. MEERSCHAERT AND A. SIKORSKII, *Stochastic Models for Fractional Calculus*, De Gruyter Stud. Math. 43, De Gruyter, Berlin, 2012.
- [27] M. MEERSCHAERT, Y. ZHANG, AND B. BAEUMER, *Tempered anomalous diffusion in heterogeneous systems*, Geophys. Res. Lett., 35 (2008), L17403.
- [28] R. METZLER AND J. KLAFTER, *The random walk's guide to anomalous diffusion: A fractional dynamics approach*, Phys. Rep., 339 (2000), pp. 1–77.

- [29] J. D. MURRAY, *Mathematical Biology II: Spatial Models and Biomedical Applications*, 3rd ed., Springer, New York, 2003.
- [30] I. PODLUBNY, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solution and Some of their Applications*, Math. Sci. Eng. 198, Academic Press, San Diego, CA, 1999.
- [31] S. G. SAMKO, A. A. KILBAS, AND O. I. MARICHEV, *Fractional Integrals and Derivatives: Theory and Applications*, Gordon and Breach, Yverdon, Switzerland, 1993.
- [32] A. SCHÄDLE, M. LÓPEZ-FERNÁNDEZ, AND C. LUBICH, *Fast and oblivious convolution quadrature*, SIAM J. Sci. Comput., 28 (2006), pp. 421–438.
- [33] M. SCHMIDT, F. SAGUES, AND I. SOKOLOV, *Mesoscopic description of reactions for anomalous diffusion: A case study*, J. Phys. Condens. Matter, 19 (2007), 065118.
- [34] N. M. TEMME, *Uniform asymptotic expansions of the incomplete Gamma functions and the incomplete Beta function*, Math. Comp., 29 (1975), pp. 1109–1114.
- [35] L. TREFETHEN AND J. WEIDEMAN, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458.
- [36] J. A. C. WEIDEMAN, *Optimizing Talbot’s contours for the inversion of the Laplace transform*, SIAM J. Numer. Anal., 44 (2006), pp. 2342–2362.
- [37] Y. YU, P. PERDIKARIS, AND G. E. KARNIADAKIS, *Fractional modeling of viscoelasticity in 3D cerebral arteries and aneurysms*, J. Comput. Phys., 323 (2016), pp. 219–242.
- [38] F. ZENG, I. TURNER, AND K. BURRAGE, *A stable fast time-stepping method for fractional integral and derivative operators*, J. Sci. Comput., 77 (2018), pp. 283–307.
- [39] F. ZENG, I. TURNER, K. BURRAGE, AND G. E. KARNIADAKIS, *A new class of semi-implicit methods with linear complexity for nonlinear fractional differential equations*, SIAM J. Sci. Comput., 40 (2018), pp. A2986–A3011.
- [40] F. ZENG, Z. ZHANG, AND G. E. KARNIADAKIS, *Second-order numerical methods for multi-term fractional differential equations: Smooth and non-smooth solutions*, Comput. Methods Appl. Mech. Engrg., 327 (2017), pp. 478–502.