



Waveform relaxation for fractional sub-diffusion equations

Jun Liu¹ · Yao-Lin Jiang² · Xiao-Long Wang³ · Yan Wang¹

Received: 23 August 2019 / Accepted: 13 September 2020 / Published online: 10 October 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

We report a new kind of waveform relaxation (WR) method for general semi-linear fractional sub-diffusion equations, and analyze the upper bound for the iteration errors. It indicates that the WR method is convergent superlinearly, and the convergence rate is dependent on the order of the time-fractional derivative and the length of the time interval. In order to accelerate the convergence, we present the windowing WR method. Then, we elaborate the parallelism based on the discrete windowing WR method, and present the corresponding fast evaluation formula. Numerical experiments are carried out to verify the effectiveness of the theoretic work.

Keywords Waveform relaxation · Fractional sub-diffusion equations · Superlinear convergence · Windowing technique · Parallelism

Mathematics Subject Classification (2010) 49M20 · 35K57 · 65M15

1 Introduction

In recent years, fractional differential equations have attracted more and more researchers' attention, since they have inherent advantages to describe the various processes and phenomenon with memory and hereditary properties [1, 2]. Fractional sub-diffusion equation is one of the most typical equations and it is often used to model the viscoelastic anomalous diffusion in disordered media [3], nanoprecipitate

✉ Jun Liu
liujun@upc.edu.cn

¹ College of Science, China University of Petroleum (East China), Qingdao, 266580, Shandong, China

² Department of Mathematical Sciences, Xi'an Jiaotong University, Xi'an, 710049, Shaanxi, China

³ School of Science, Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, China

growth in solid solutions [4], diffusion process in magnetic resonance imaging [5], and so on. Because of the nonlocal properties, the sub-diffusion equations are more complicated than the classical integer-order differential equations, and it is usually difficult to obtain the analytic solutions of the sub-diffusion equations. Various numerical schemes for such equations are emerging [6–8].

The waveform relaxation (WR) method, as an iterative method for time-dependent equations, produces a series of functions with respect to the time variable, to approximate the solution of the equations. The WR method is firstly proposed for the simulation of large circuits [9], and it is able to decouple a complicated large system into a series of weakly coupled small sub-systems, which can be solved simultaneously. It has been proved in [10, 11] that the WR method has superlinear convergence for general nonlinear ordinary differential equations (ODEs) on bounded domain, and is of linear convergence for linear ODEs on unbounded domain. The WR method has been widely used to solve differential algebra equations (DAEs) [12, 13], integral-differential-algebraic equations (IDAEs) [14], and delay-differential equations (DDEs) [15]. To apply the WR method to partial differential equations (PDEs), one popular method is to decompose spatial domain into several sub-domains, and solve a set of systems on sub-domains, which is also called Schwarz WR method [16–18]. Moreover, we have also proposed a kind of WR method for reaction diffusion equations [19], in which we preserve the diffusion term and relax the nonlinear reaction term. This kind of WR method converges quickly, and can also be implemented in parallel.

The WR method is also an alternative method for fractional differential equations. In ref. [20], the WR method is used to solve linear and nonlinear time-fractional ODEs, and proper convergent splitting is constructed. Then, the WR method is extended to handle fractional DDEs [21] and DAEs [22]. In ref. [23], a multigrid WR method is proposed for the time-fractional heat equation, and a fast computing technique is also supplied, based on the Toeplitz-like structure of the coefficient matrix. Furthermore, Schwarz WR methods for fractional PDEs are also reported recently [24, 25]. In this paper, we apply the WR method to a kind of fractional sub-diffusion equation with a nonlinear reaction term at the PDE level, which produces a series of linear time-fractional PDEs. The WR method has the same advantages as that for integer-order PDEs in [19], but the convergence is totally different, which is dependent on the order of the fractional derivative, see Theorem 2.1 for details. In order to accelerate the convergence, a windowing technique is combined, and the corresponding discrete windowing WR method is discussed.

The outline of the paper is as follows. In Section 2, we propose the new WR method for a kind of semi-linear fractional sub-diffusion equations, and analyze the convergence. Combing the windowing technique, we introduce the windowing WR method in Section 3. In Section 4, the discrete windowing WR method is constructed, and the convergence and the parallelism are presented, then the fast evaluation formula for the windowing WR method is provided. In Section 5, two numerical experiments are given to verify the effectiveness of the theoretic results. Finally, conclusions are attached in the last section.

2 Waveform relaxation method and convergence

We consider the following fractional-order initial value problem satisfying homogeneous boundary conditions

$$\begin{cases} {}_0^C\mathcal{D}_t^\alpha u(x, t) = Ku_{xx}(x, t) + f(x, t, u(x, t)), & 0 < x < L, \quad 0 < t < T, \\ u(x, 0) = \varphi(x), & 0 \leq x \leq L, \\ u(0, t) = u(L, t) = 0, & 0 \leq t \leq T, \end{cases} \tag{1}$$

where the positive constant K is the transport-related coefficient, $f(x, t, u(x, t))$ is a given nonlinear source/sink term, $\varphi(x)$ is a prescribed initial value, and ${}_0^C\mathcal{D}_t^\alpha u(x, t)$ is the Caputo fractional derivative of order α ($0 < \alpha < 1$) defined by [1]

$${}_0^C\mathcal{D}_t^\alpha u(x, t) := \frac{1}{\Gamma(1 - \alpha)} \int_0^t \frac{\partial u(x, s)}{\partial s} \frac{ds}{(t - s)^\alpha}, \tag{2}$$

where $\Gamma(\cdot)$ denotes the Gamma function. It can be seen that system (1) involves a nonlinear fractional sub-diffusion equation.

2.1 Construction of the WR method

A typical iterative scheme of the WR method for system (1) is

$$\begin{cases} {}_0^C\mathcal{D}_t^\alpha u^{(k+1)}(x, t) = Ku_{xx}^{(k+1)}(x, t) + F(x, t, u^{(k+1)}(x, t), u^{(k)}(x, t)), & 0 < x < L, \quad 0 < t < T, \\ u^{(k+1)}(x, 0) = \varphi(x), & 0 \leq x \leq L, \\ u^{(k+1)}(0, t) = u^{(k+1)}(L, t) = 0, & 0 \leq t \leq T, \end{cases}$$

for $k = 0, 1, \dots$, where the splitting function $F(x, t, u(x, t), v(x, t))$ determines the type of WR algorithm, and it always satisfies $F(x, t, u(x, t), u(x, t)) = f(x, t, u(x, t))$. In this paper, we choose $F(x, t, u^{(k+1)}(x, t), u^{(k)}(x, t)) = f(x, t, u^{(k)}(x, t))$ for simplicity, i.e., the iterative scheme can be written as

$$\begin{cases} {}_0^C\mathcal{D}_t^\alpha u^{(k+1)}(x, t) = Ku_{xx}^{(k+1)}(x, t) + f(x, t, u^{(k)}(x, t)), & 0 < x < L, \quad 0 < t < T, \\ u^{(k+1)}(x, 0) = \varphi(x), & 0 \leq x \leq L, \\ u^{(k+1)}(0, t) = u^{(k+1)}(L, t) = 0, & 0 \leq t \leq T, \end{cases} \tag{3}$$

for $k = 0, 1, \dots$. The initial guess of the iterations is often chosen as $u^{(0)}(x, t) \equiv \varphi(x)$, for $0 \leq t \leq T$. The special WR algorithm (3) is also called Picard relaxation algorithm. We can see that WR algorithm (3) leads to a series of linear systems.

For any fixed iteration index k , by the separation of variables, the solution of system (3) has the form

$$u^{(k)}(x, t) = \sum_{n=1}^\infty a_n^{(k)}(t) \sin \frac{n\pi x}{L}, \quad (x, t) \in [0, L] \times [0, T], \tag{4}$$

where the coefficients $a_n^{(k)}$, for $n = 1, 2, \dots$, are to be determined. We further suppose

$$f(x, t, u^{(k)}(x, t)) = \sum_{n=1}^{\infty} f_n(t) \sin \frac{n\pi x}{L}, \quad \varphi(x) = \sum_{n=1}^{\infty} \varphi_n \sin \frac{n\pi x}{L}, \quad (5)$$

with

$$f_n(t) = \frac{2}{L} \int_0^L f(x, t, u^{(k)}(x, t)) \sin \frac{n\pi x}{L} dx, \quad \varphi_n = \frac{2}{L} \int_0^L \varphi(x) \sin \frac{n\pi x}{L} dx.$$

Substituting expressions (4) and (5) into system (3), we have

$$\begin{cases} \sum_{n=1}^{\infty} \left[{}^C_0\mathcal{D}_t^\alpha a_n^{(k+1)}(t) + K \left(\frac{n\pi}{L} \right)^2 a_n^{(k+1)}(t) \right] \sin \frac{n\pi x}{L} = \sum_{n=1}^{\infty} f_n(t) \sin \frac{n\pi x}{L}, \\ \sum_{n=1}^{\infty} a_n^{(k+1)}(0) \sin \frac{n\pi x}{L} = \sum_{n=1}^{\infty} \varphi_n \sin \frac{n\pi x}{L}. \end{cases}$$

Comparing the coefficients of the basis functions $\{\sin \frac{n\pi x}{L}\}_{n=1}^{\infty}$, we can obtain

$$\begin{cases} {}^C_0\mathcal{D}_t^\alpha a_n^{(k+1)}(t) + K \left(\frac{n\pi}{L} \right)^2 a_n^{(k+1)}(t) = f_n(t), \\ a_n^{(k+1)}(0) = \varphi_n. \end{cases} \quad (6)$$

According to Theorem 5.15 of ref. [26], we get the solution of (6)

$$a_n^{(k+1)}(t) = \varphi_n E_\alpha \left[-K \left(\frac{n\pi}{L} \right)^2 t^\alpha \right] + \int_0^t (t-s)^{\alpha-1} E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right] \frac{2}{L} \int_0^L f(\xi, s, u^{(k)}(\xi, s)) \sin \frac{n\pi \xi}{L} d\xi ds,$$

where $E_{\alpha,\beta}(z)$ is the Mittag-Leffler function, defined by [26]

$$E_{\alpha,\beta}(z) := \sum_{i=0}^{\infty} \frac{z^i}{\Gamma(i\alpha + \beta)}, \quad (\alpha, \beta \in \mathbb{C}, \Re(\alpha) > 0),$$

and

$$E_\alpha(z) = E_{\alpha,1}(z) = \sum_{i=0}^{\infty} \frac{z^i}{\Gamma(i\alpha + 1)}.$$

Then, the solution of system (3) has the form

$$\begin{aligned}
 u^{(k+1)}(x, t) &= \sum_{n=1}^{\infty} a_n^{(k+1)}(t) \sin \frac{n\pi x}{L} \\
 &= \frac{2}{L} \sum_{n=1}^{\infty} \int_0^L \varphi(x) \sin \frac{n\pi x}{L} dx E_{\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 t^{\alpha} \right] \sin \frac{n\pi x}{L} \\
 &\quad + \int_0^t \left(\sum_{n=1}^{\infty} \frac{2}{L} \int_0^L f(\xi, s, u^{(k)}(\xi, s)) \sin \frac{n\pi \xi}{L} d\xi (t-s)^{\alpha-1} \right. \\
 &\quad \left. \times E_{\alpha, \alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^{\alpha} \right] \sin \frac{n\pi x}{L} \right) ds.
 \end{aligned} \tag{7}$$

Denote

$$\begin{aligned}
 a_n(t) &:= \frac{2}{L} \int_0^L \varphi(x) \sin \frac{n\pi x}{L} dx E_{\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 t^{\alpha} \right] \\
 &\quad + \int_0^t (t-s)^{\alpha-1} E_{\alpha, \alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^{\alpha} \right] \frac{2}{L} \int_0^L f(\xi, s, u(\xi, s)) \\
 &\quad \sin \frac{n\pi \xi}{L} d\xi ds,
 \end{aligned}$$

we can verify that the function $\sum_{n=1}^{\infty} a_n(t) \sin \frac{n\pi x}{L}$ satisfies system (1).

2.2 Convergence analysis

In this part, we present the convergence of the iterative sequence by WR algorithm (3). Before that, we still need a preliminary definition and lemmata as follows.

For $\varphi(t) \in C([0, T], \mathbb{R})$, we define an operator

$$(\mathcal{A}\varphi)(t) = \int_0^t \lambda(t-s)^{\alpha-1} \varphi(s) ds, \tag{8}$$

where λ is s positive constant. Then, we have

Lemma 2.1 For any positive integer k ,

$$|(\mathcal{A}^k \varphi)(t)| \leq \frac{(\lambda \Gamma(\alpha) t^{\alpha})^k}{\Gamma(k\alpha + 1)} \max_{0 \leq s \leq t} |\varphi(s)|, \quad t \in [0, T], \tag{9}$$

where $(\mathcal{A}^k \varphi)(t) = (\mathcal{A}(\mathcal{A}^{k-1} \varphi))(t)$.

Proof We prove the result by induction. For $k = 1$, we have

$$\begin{aligned}
 |(\mathcal{A}\varphi)(t)| &\leq \int_0^t \lambda(t-s)^{\alpha-1} ds \max_{0 \leq s \leq t} |\varphi(s)| = \frac{\lambda t^{\alpha}}{\alpha} \max_{0 \leq s \leq t} |\varphi(s)| \\
 &= \frac{\lambda \Gamma(\alpha) t^{\alpha}}{\Gamma(\alpha + 1)} \max_{0 \leq s \leq t} |\varphi(s)|.
 \end{aligned}$$

Suppose that inequality (9) holds for any fixed index k , then

$$\begin{aligned}
 |(\mathcal{A}^{k+1}\varphi)(t)| &\leq \int_0^t \lambda(t-s)^{\alpha-1}|(\mathcal{A}^k\varphi)(s)|ds \\
 &\leq \lambda \int_0^t (t-s)^{\alpha-1} \frac{(\lambda\Gamma(\alpha)s^\alpha)^k}{\Gamma(k\alpha+1)} ds \max_{0\leq s\leq t} |\varphi(s)| \\
 &= \frac{\lambda^{k+1}\Gamma^k(\alpha)t^{(k+1)\alpha}}{\Gamma(k\alpha+1)} B(\alpha, k\alpha+1) \max_{0\leq s\leq t} |\varphi(s)| \\
 &= \frac{\lambda^{k+1}\Gamma^k(\alpha)t^{(k+1)\alpha}}{\Gamma(k\alpha+1)} \frac{\Gamma(\alpha)\Gamma(k\alpha+1)}{\Gamma(k\alpha+\alpha+1)} \max_{0\leq s\leq t} |\varphi(s)| \\
 &= \frac{(\lambda\Gamma(\alpha)t^\alpha)^{k+1}}{\Gamma((k+1)\alpha+1)} \max_{0\leq s\leq t} |\varphi(s)|,
 \end{aligned}$$

where $B(\cdot, \cdot)$ denotes the Beta function. The conclusion results. □

Lemma 2.2 For any $t \in [0, T]$, $\alpha \in (0, 1)$, and $\lambda > 0$, the series

$$\sum_{n=1}^{\infty} E_{\alpha,\alpha}(-\lambda n^2 t^\alpha)$$

converges.

Proof Since function $E_{\alpha,\alpha}(-\lambda n^2 t^\alpha)$ is completely monotonous with respect to n [27], for any $\lambda, t > 0$, the series $\sum_{n=1}^{\infty} E_{\alpha,\alpha}(-\lambda n^2 t^\alpha)$ has the same convergence as the integration $\int_1^\infty E_{\alpha,\alpha}(-\lambda t^\alpha x^2) dx$.

To evaluate the integration, we consider $\int_1^y E_{\alpha,\alpha}(-\lambda t^\alpha x^2) dx$ for $y \geq 1$, then we have

$$\begin{aligned}
 \int_1^y E_{\alpha,\alpha}(-\lambda t^\alpha x^2) dx &\leq \int_1^y E_{\alpha,\alpha}(-\lambda t^\alpha x) dx = \int_1^y \sum_{i=0}^{\infty} \frac{(-\lambda t^\alpha)^i x^i}{\Gamma(i\alpha + \alpha)} dx \\
 &= \sum_{i=0}^{\infty} \frac{(-\lambda t^\alpha)^i (y^{i+1} - 1)}{(i+1)\Gamma(i\alpha + \alpha)} = \sum_{i=1}^{\infty} \frac{\alpha(-\lambda t^\alpha)^i (y^i - 1)}{(-\lambda t^\alpha) i \alpha \Gamma(i\alpha)} \\
 &= \frac{\alpha}{-\lambda t^\alpha} \sum_{i=0}^{\infty} \frac{(-\lambda t^\alpha)^i (y^i - 1)}{\Gamma(i\alpha + 1)} = \frac{\alpha}{\lambda t^\alpha} [E_\alpha(-\lambda t^\alpha) - E_\alpha(-\lambda t^\alpha y)].
 \end{aligned}$$

For any $y \geq 1$, $E_\alpha(-\lambda t^\alpha y)$ converges, and then $\int_1^\infty E_{\alpha,\alpha}(-\lambda t^\alpha x^2) dx$ converges. The conclusion results. □

The convergence of the WR algorithm (3) can be found in the following theorem.

Theorem 2.1 If the nonlinear function $f(x, t, u)$ has a continuous first-order derivative with respect to u , then the iterative sequence of functions $\{u^{(k)}(x, t)\}$ for $k =$

1, 2, . . . , generated by the WR algorithm (3) converges to the true solution $u(x, t)$ of system (1), and satisfies

$$\max_{0 \leq x \leq L, 0 \leq t \leq T} |u^{(k)}(x, t) - u(x, t)| \leq \frac{(\bar{M}\Gamma(\alpha)T^\alpha)^k}{\Gamma(k\alpha + 1)} \max_{0 \leq x \leq L, 0 \leq t \leq T} |u^{(0)}(x, t) - u(x, t)|, \tag{10}$$

where \bar{M} is a positive constant.

Proof We define $\varepsilon^{(k)}(x, t) = u^{(k)}(x, t) - u(x, t)$, $0 \leq x \leq L$, $0 \leq t \leq T$, which satisfies

$$\begin{cases} {}_0^C D_t^\alpha \varepsilon^{(k+1)}(x, t) = K \varepsilon_{xx}^{(k+1)}(x, t) + f(x, t, u^{(k)}(x, t)) - f(x, t, u(x, t)), & 0 < x < L, \quad 0 < t < T, \\ \varepsilon^{(k+1)}(x, 0) = 0, & 0 \leq x \leq L, \\ \varepsilon^{(k+1)}(0, t) = \varepsilon^{(k+1)}(L, t) = 0, & 0 \leq t \leq T, \end{cases} \tag{11}$$

By the separation of variables, the solution of system (11) is

$$\begin{aligned} \varepsilon^{(k+1)}(x, t) &= \int_0^t \left(\sum_{n=1}^\infty \frac{2}{L} \int_0^L [f(\xi, s, u^{(k)}(\xi, s)) - f(\xi, s, u(\xi, s))] \sin \frac{n\pi\xi}{L} d\xi \right. \\ &\quad \left. \times (t-s)^{\alpha-1} E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right] \sin \frac{n\pi x}{L} \right) ds \\ &= \int_0^t \left(\sum_{n=1}^\infty \frac{2}{L} \int_0^L f'(\xi, s, u_*^{(k)}(\xi, s)) \varepsilon^{(k)}(\xi, s) \sin \frac{n\pi\xi}{L} d\xi \right. \\ &\quad \left. \times (t-s)^{\alpha-1} E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right] \sin \frac{n\pi x}{L} \right) ds. \end{aligned}$$

where $u_*^{(k)} \in [\min(u^{(k)}(x, t), u(x, t)), \max(u^{(k)}(x, t), u(x, t))]$.

We notice that for any $0 \leq s \leq t \leq T$,

$$\begin{aligned} &\sum_{n=1}^\infty \frac{2}{L} \int_0^L f'(\xi, s, u_*^{(i)}(\xi, s)) \varepsilon^{(i)}(\xi, s) \sin \frac{n\pi\xi}{L} d\xi \sin \frac{n\pi x}{L} \\ &= f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s), \end{aligned} \tag{12}$$

and the function $E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right]$ is bounded and monotonous with respect to n [27]. According to Abel’s test for uniform convergence, the series of functions

$$\sum_{n=1}^\infty \frac{2}{L} \int_0^L f'(\xi, s, u_*^{(k)}(\xi, s)) \varepsilon^{(k)}(\xi, s) \sin \frac{n\pi\xi}{L} d\xi E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right] \sin \frac{n\pi x}{L}$$

converges uniformly for $x \in [0, l]$ and $s \in [0, t]$. Denote

$$\eta(x, t, s) = \sum_{n=1}^{\infty} \frac{2}{L} \int_0^L f'(\xi, s, u_*^{(k)}(\xi, s)) \varepsilon^{(k)}(\xi, s) \sin \frac{n\pi\xi}{L} d\xi \\ \times E_{\alpha, \alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t - s)^\alpha \right] \sin \frac{n\pi x}{L},$$

then we have

$$\varepsilon^{(k+1)}(x, t) = \int_0^t (t - s)^{\alpha-1} \eta(x, t, s) ds.$$

For any fixed $t \in (0, T]$, we suppose that there is a $s_0 \in (0, t)$, such that

$$\min_{s_0 \leq s \leq t} \max_{0 \leq x \leq L} |f'(\xi, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s)| > 0,$$

and denote $\varepsilon_0 = \frac{1}{\Gamma(\alpha)} \min_{s_0 \leq s \leq t} \max_{0 \leq x \leq L} |f'(\xi, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s)|$. From (12) we know that,

$$\eta(x, t, t) = \frac{1}{\Gamma(\alpha)} f'(x, t, u_*^{(i)}(x, t)) \varepsilon^{(i)}(x, t),$$

so there exist a constant $\delta_1 \in (0, t)$, such that for any $s \in (t - \delta_1, t)$,

$$\left| \eta(x, t, s) - \frac{1}{\Gamma(\alpha)} f'(x, t, u_*^{(i)}(x, t)) \varepsilon^{(i)}(x, t) \right| < \frac{\varepsilon_0}{2},$$

and there exist a constant $\delta_2 \in (0, t)$, such that for any $s \in (t - \delta_2, t)$,

$$\left| \frac{1}{\Gamma(\alpha)} f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s) - \frac{1}{\Gamma(\alpha)} f'(x, t, u_*^{(i)}(x, t)) \varepsilon^{(i)}(x, t) \right| < \frac{\varepsilon_0}{2}.$$

Let $s_1 = \max\{t - \delta_1, t - \delta_2, s_0\}$. For any $s \in (s_1, t)$, we have

$$\left| \eta(x, t, s) - \frac{1}{\Gamma(\alpha)} f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s) \right| \\ \leq \left| \eta(x, t, s) - \frac{1}{\Gamma(\alpha)} f'(x, t, u_*^{(i)}(x, t)) \varepsilon^{(i)}(x, t) \right| \\ + \left| \frac{1}{\Gamma(\alpha)} f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s) - \frac{1}{\Gamma(\alpha)} f'(x, t, u_*^{(i)}(x, t)) \varepsilon^{(i)}(x, t) \right| \\ \leq \varepsilon_0.$$

Therefore,

$$|\eta(x, t, s)| \leq \frac{1}{\Gamma(\alpha)} |f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s)| + \varepsilon_0,$$

and

$$\begin{aligned} \max_{0 \leq x \leq L} |\eta(x, t, s)| &\leq \frac{1}{\Gamma(\alpha)} \max_{0 \leq x \leq L} |f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s)| + \varepsilon_0 \\ &\leq \frac{2}{\Gamma(\alpha)} \max_{0 \leq x \leq L} |f'(x, s, u_*^{(i)}(x, s)) \varepsilon^{(i)}(x, s)| \quad (13) \\ &\leq \frac{2M}{\Gamma(\alpha)} \max_{0 \leq x \leq L} |\varepsilon^{(i)}(x, s)|, \end{aligned}$$

where M is a uniform upper bound of $|f'(x, t, u(x, t))|$, for $0 \leq x \leq L$ and $0 \leq t \leq T$.

For $s \in [0, s_1]$, we consider the following series of functions,

$$G(x, \xi, t - s) = \frac{2}{L} \sum_{n=1}^{\infty} E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t - s)^\alpha \right] \sin \frac{n\pi\xi}{L} \sin \frac{n\pi x}{L}. \tag{14}$$

For any $x, \xi \in [0, L]$, we have

$$\left| E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t - s)^\alpha \right] \sin \frac{n\pi\xi}{L} \sin \frac{n\pi x}{L} \right| \leq E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t - s)^\alpha \right].$$

Together with Lemma 2.2, series of functions (14) is convergent uniformly for $x, \xi \in [0, L]$, and $0 \leq s \leq t \leq s_1$. We denote by M_1 an upper bound of the series of functions (14), then

$$\eta(x, t, s) = \int_0^L G(x, \xi, t - s) f'(\xi, s, u_*^{(k)}(\xi, s)) \varepsilon^{(k)}(\xi, s) d\xi.$$

and

$$\max_{0 \leq x \leq L} |\eta(x, t, s)| \leq LM_1 M \max_{0 \leq x \leq L} |\varepsilon^{(k)}(x, s)|. \tag{15}$$

We choose $\bar{M} = \max\{\frac{2M}{\Gamma(\alpha)}, LM_1 M\}$. By the inequalities (13) and (15), we obtain

$$\max_{0 \leq x \leq L} |\eta(x, t, s)| \leq \bar{M} \max_{0 \leq x \leq L} |\varepsilon^{(k)}(x, s)|, \quad s \in [0, t],$$

and

$$\begin{aligned} \max_{0 \leq x \leq L} |\varepsilon^{(k+1)}(x, t)| &\leq \int_0^t (t - s)^{\alpha-1} \max_{0 \leq x \leq L} |\eta(x, t, s)| ds \\ &\leq \int_0^t \bar{M} (t - s)^{\alpha-1} \max_{0 \leq x \leq L} |\varepsilon^{(k)}(x, s)| ds. \end{aligned}$$

According to Lemma 2.1, we can obtain

$$\max_{0 \leq x \leq L} |\varepsilon^{(k)}(x, t)| \leq \frac{(\bar{M}\Gamma(\alpha)t^\alpha)^k}{\Gamma(k\alpha + 1)} \max_{0 \leq x \leq L, 0 \leq s \leq t} |\varepsilon^{(0)}(x, s)|, \quad t \in [0, T],$$

which leads to

$$\max_{0 \leq x \leq L, 0 \leq t \leq T} |\varepsilon^{(k)}(x, t)| \leq \frac{(\bar{M}\Gamma(\alpha)T^\alpha)^k}{\Gamma(k\alpha + 1)} \max_{0 \leq x \leq L, 0 \leq t \leq T} |\varepsilon^{(0)}(x, t)|.$$

This completes the proof. □

Remark For the convergence rate shown in estimation (10), we denote $v_k = \frac{(\bar{M}\Gamma(\alpha)T^\alpha)^k}{\Gamma(k\alpha+1)}$. Then, we compute

$$r = \lim_{k \rightarrow \infty} \frac{v_{k+1}}{v_k} = \bar{M}\Gamma(\alpha)T^\alpha \lim_{k \rightarrow \infty} \frac{\Gamma(k\alpha)}{\Gamma(k\alpha + \alpha)}.$$

Based on Stirling’s formula of the Gamma function, we get $r \sim \mathcal{O}(k^{-\alpha})$, which means that WR algorithm (3) is convergent superlinearly to the true solution of system (1).

3 Windowing WR method

We can see from Theorem 2.1 that the upper bound of the WR iteration is related to the length of the time interval. Therefore, the windowing technique is a proper way to reduce the iteration error [28]. In this section, we combine the windowing technique with the WR method introduced in the last section to establish an improved WR method, say the windowing WR method.

3.1 Construction of the windowing WR method

For the fractional sub-diffusion equation (1), we first divide the time interval $[0, T]$ into N windows, with

$$0 = T_0 < T_1 < \dots < T_N = T.$$

We denote the i -th window by $\Omega_i = [T_i, T_{i+1}]$, with the length $H_i = T_{i+1} - T_i$, for $i = 0, 1, \dots, N - 1$, and the maximum length of the windows is $H = \max_{0 \leq i \leq N-1} \{H_i\}$.

We denote by $u_i^{(k)}(x, t)$ the k -th waveform on the i -th window Ω_i , for $k = 0, 1, \dots, k_i$, where k_i is the maximum number of WR iterations on Ω_i . Then, the windowing WR method for system (1) on Ω_i is given by

$$\begin{cases} {}_T^C \mathcal{D}_t^\alpha u_i^{(k+1)}(x, t) + \sum_{j=0}^{i-1} \frac{1}{\Gamma(1-\alpha)} \int_{T_j}^{T_{j+1}} (t-s)^{-\alpha} \frac{\partial}{\partial s} u_j^{(k_j)}(x, s) ds \\ = K \frac{\partial^2}{\partial x^2} u_i^{(k+1)}(x, t) + f(x, t, u_i^{(k)}(x, t)), & 0 < x < L, \quad T_i < t < T_{i+1}, \\ u_i^{(k+1)}(x, T_i) = u_{i-1}^{(k_{i-1})}(x, T_i), & 0 \leq x \leq L, \\ u_i^{(k+1)}(0, t) = u_i^{(k+1)}(L, t) = 0, & T_i \leq t \leq T_{i+1}, \end{cases} \tag{16}$$

for $k = 0, 1, \dots, k_i - 1$, and $i = 0, 1, \dots, N - 1$. We define $u_{-1}^{(k-1)}(x, 0) = \varphi(x)$ and choose $u_i^{(0)}(x, t) \equiv u_{i-1}^{(k_{i-1})}(x, T_i)$ as the initial guess on Ω_i . We can observe that, when solving the waveforms $u_i^{(k)}$ on Ω_i , the approximations of the solution on the previous windows are already known.

For $t \in [T_i, T_{i+1}]$ and any fixed k , we denote

$$\bar{f}_i(x, t) = f(x, t, u_i^{(k)}(x, t)) - \sum_{j=0}^{i-1} \frac{1}{\Gamma(1-\alpha)} \int_{T_j}^{T_{j+1}} (t-s)^{-\alpha} \frac{\partial}{\partial s} u_j^{(k_j)}(x, s) ds.$$

Similar to the method shown in sub-section 2.1, we have the solution of system (16) with the form

$$\begin{aligned} u_i^{(k+1)}(x, t) &= \sum_{n=1}^{\infty} \frac{2}{L} \int_0^L u_{i-1}^{(k_{i-1})}(x, T_i) \sin \frac{n\pi x}{L} dx E_\alpha \left[-K \left(\frac{n\pi}{L} \right)^2 (t - T_i)^\alpha \right] \sin \frac{n\pi x}{L} \\ &+ \int_{T_i}^t \left(\sum_{n=1}^{\infty} \frac{2}{L} \int_0^L \bar{f}_i(\xi, s) \sin \frac{n\pi \xi}{L} d\xi (t-s)^{\alpha-1} E_{\alpha, \alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right] \sin \frac{n\pi x}{L} \right) ds. \end{aligned}$$

3.2 Convergence of the windowing WR method

Before giving the convergence of the windowing WR method (16), we first investigate an operator and its property.

For $\varphi(t) \in C([0, T], \mathbb{R})$, we define an operator

$$(\mathcal{B}\varphi)(t) = B_0 + \int_{T_i}^t \lambda(t-s)^{\alpha-1} \varphi(s) ds, \quad t \in [T_i, T_{i+1}], \tag{17}$$

where B_0 and λ are positive constants. Then, we have

Lemma 3.1 *For any positive integer k , we have*

$$|(\mathcal{B}^k \varphi)(t)| \leq B_0 \sum_{j=0}^{k-1} \frac{(\lambda \Gamma(\alpha)(t - T_i)^\alpha)^j}{\Gamma(j\alpha + 1)} + \frac{(\lambda \Gamma(\alpha)(t - T_i)^\alpha)^k}{\Gamma(k\alpha + 1)} \max_{T_i \leq s \leq t} |\varphi(s)|, \tag{18}$$

where $(\mathcal{B}^k \varphi)(t) = (\mathcal{B}(\mathcal{B}^{k-1} \varphi))(t)$.

Proof We prove the result by induction. For $k = 1$, we have

$$|(\mathcal{B}\varphi)(t)| \leq B_0 + \int_{T_i}^t \lambda(t-s)^{\alpha-1} ds \max_{T_i \leq s \leq t} |\varphi(s)| = B_0 + \frac{\lambda \Gamma(\alpha)(t - T_i)^\alpha}{\Gamma(\alpha + 1)} \max_{T_i \leq s \leq t} |\varphi(s)|.$$

Suppose that inequality (18) holds for any fixed index k , then

$$\begin{aligned} & |(\mathcal{B}^{k+1} \varphi)(t)| \\ & \leq B_0 + \lambda \int_{T_i}^t (t-s)^{\alpha-1} \left[B_0 \sum_{j=0}^{k-1} \frac{(\lambda \Gamma(\alpha)(s - T_i)^\alpha)^j}{\Gamma(j\alpha + 1)} + \frac{(\lambda \Gamma(\alpha)(s - T_i)^\alpha)^k}{\Gamma(k\alpha + 1)} \max_{T_i \leq \mu \leq s} |\varphi(\mu)| \right] ds \\ & = B_0 + B_0 \sum_{j=0}^{k-1} \frac{\lambda^{j+1} \Gamma^j(\alpha)}{\Gamma(j\alpha + 1)} (t - T_i)^{(j+1)\alpha} B(\alpha, j\alpha + 1) + \frac{\lambda^{k+1} \Gamma^k(\alpha)}{\Gamma(k\alpha + 1)} (t - T_i)^{(k+1)\alpha} B(\alpha, k\alpha + 1) \\ & \quad \max_{T_i \leq s \leq t} |\varphi(s)|, \end{aligned}$$

where we have used the equality

$$\int_{T_i}^t (t-s)^{\alpha-1} (s - T_i)^{j\alpha} ds = (t - T_i)^{(j+1)\alpha} B(\alpha, j\alpha + 1), \quad j = 0, 1, \dots, k.$$

By the properties of the Beta function $B(\alpha, j\alpha + 1)$, we further have

$$\begin{aligned} & |(\mathcal{B}^{k+1} \varphi)(t)| \\ & \leq B_0 + B_0 \sum_{j=0}^{k-1} \frac{\lambda^{j+1} \Gamma^{j+1}(\alpha)}{\Gamma((j+1)\alpha + 1)} (t - T_i)^{(j+1)\alpha} + \frac{\lambda^{k+1} \Gamma^{k+1}(\alpha)}{\Gamma((k+1)\alpha + 1)} (t - T_i)^{(k+1)\alpha} \\ & \quad \max_{T_i \leq s \leq t} |\varphi(s)| \\ & = B_0 \sum_{j=0}^k \frac{(\lambda \Gamma(\alpha)(t - T_i)^\alpha)^j}{\Gamma(j\alpha + 1)} + \frac{(\lambda \Gamma(\alpha)(t - T_i)^\alpha)^{k+1}}{\Gamma((k+1)\alpha + 1)} \max_{T_i \leq s \leq t} |\varphi(s)|. \end{aligned}$$

The conclusion results. □

Then, the convergence of the windowing WR method for system (1) can be found in the following theorem.

Theorem 3.1 *If the nonlinear function $f(x, t, u)$ has a continuous first-order derivative with respect to u , then the iteration error between the approximation $\{u_i^{(k_i)}(x, t), i = 0, 1, \dots, N - 1\}$, generated by the windowing WR method (16), and the true solution $\{u_i(x, t), i = 0, 1, \dots, N - 1\}$ of system (1) restricted in Ω_i , can be bounded by*

$$\begin{aligned} & \max_{T_i \leq t \leq T_{i+1}} \max_{0 \leq x \leq L} |u_i^{(k_i)}(x, t) - u_i(x, t)| \\ & \leq \sum_{j=0}^i \bar{\Phi}_j \left(\prod_{l=j+1}^i \Theta_l \tilde{\Upsilon}_l \right) \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |u_i^{(0)}(x, t) - u_i(x, t)|, \end{aligned} \tag{19}$$

where

$$\begin{aligned} \Theta_i &= M_i \left(1 + \frac{2}{\Gamma(1 - \alpha)} \frac{\pi}{\sin(\pi\alpha)} \right), \quad \tilde{\Upsilon}_i = \sum_{j=0}^{k_i-1} \frac{(MM_i\Gamma(\alpha)H_i^\alpha)^j}{\Gamma(j\alpha + 1)}, \\ \bar{\Phi}_i &= \frac{(MM_i\Gamma(\alpha)H_i^\alpha)^{k_i}}{\Gamma(k_i\alpha + 1)}, \end{aligned}$$

with M, \bar{M} positive constants, for $i = 0, 1, \dots, N - 1$.

Proof We define $\varepsilon_i^{(k)}(x, t) = u_i^{(k)}(x, t) - u_i(x, t), 0 \leq x \leq L, T_i \leq t \leq T_{i+1}$, which satisfies

$$\left\{ \begin{aligned} & C_{T_i} D_t^\alpha \varepsilon_i^{(k+1)}(x, t) + \sum_{j=0}^{i-1} \frac{1}{\Gamma(1 - \alpha)} \int_{T_j}^{T_{j+1}} (t - s)^{-\alpha} \frac{\partial}{\partial s} \varepsilon_j^{(k_j)}(x, s) ds \\ & = K \frac{\partial^2}{\partial x^2} \varepsilon_i^{(k+1)}(x, t) + f(x, t, u_i^{(k)}(x, t)) - f(x, t, u_i(x, t)), \quad 0 < x < L, \\ & T_i < t < T_{i+1}, \\ & \varepsilon_i^{(k+1)}(x, T_i) = \varepsilon_{i-1}^{(k_{i-1})}(x, T_i), \quad 0 \leq x \leq L, \\ & \varepsilon_i^{(k+1)}(0, t) = \varepsilon_i^{(k+1)}(L, t) = 0, \quad T_i \leq t \leq T_{i+1}. \end{aligned} \right. \tag{20}$$

The solution of system (20) has the form

$$\begin{aligned} \varepsilon_i^{(k+1)}(x, t) &= \sum_{n=1}^{\infty} \frac{2}{L} \int_0^L \varepsilon_{i-1}^{(k_{i-1})}(\xi, T_i) \sin \frac{n\pi\xi}{L} d\xi E_\alpha \left[-K \left(\frac{n\pi}{L} \right)^2 (t - T_i)^\alpha \right] \sin \frac{n\pi x}{L} \\ &+ \int_{T_i}^t \left(\sum_{n=1}^{\infty} \frac{2}{L} \int_0^L g_i(\xi, s) \sin \frac{n\pi\xi}{L} d\xi (t-s)^{\alpha-1} E_{\alpha,\alpha} \left[-K \left(\frac{n\pi}{L} \right)^2 (t-s)^\alpha \right] \sin \frac{n\pi x}{L} \right) ds. \end{aligned} \tag{21}$$

where

$$g_i(x, t) = f(x, t, u_i^{(k)}(x, t)) - f(x, t, u_i(x, t)) - \sum_{j=0}^{i-1} \frac{1}{\Gamma(1-\alpha)} \int_{T_j}^{T_{j+1}} (t-s)^{-\alpha} \frac{\partial}{\partial s} \varepsilon_j^{(k_j)}(x, s) ds.$$

Similar to Theorem 2.1, there exist constants M_i , for $i = 0, 1, \dots, N - 1$, such that

$$\max_{0 \leq x \leq L} |\varepsilon_i^{(k+1)}(x, t)| \leq M_i \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_i-1)}(x, T_i)| + M_i \int_{T_i}^t (t-s)^{\alpha-1} \max_{0 \leq x \leq L} |g_i(x, s)| ds, \quad t \in [T_i, T_{i+1}]. \tag{22}$$

We might as well suppose $M_i \geq 1$, for $i = 0, 1, \dots, N - 1$.

Using integration by parts and the property $\varepsilon_j^{(k_j)}(x, T_{j+1}) = \varepsilon_{j+1}^{(k_{j+1})}(x, T_{j+1})$, we have

$$\begin{aligned} g_i(x, t) &= f'(x, t, u_{i,*}^{(k)}(x, t)) \varepsilon_i^{(k)}(x, t) + \frac{\alpha}{\Gamma(1-\alpha)} \sum_{j=0}^{i-1} \int_{T_j}^{T_{j+1}} (t-s)^{-\alpha-1} \varepsilon_j^{(k_j)}(x, s) ds \\ &\quad - \frac{1}{\Gamma(1-\alpha)} \sum_{j=0}^{i-1} \left[(t - T_{j+1})^{-\alpha} \varepsilon_j^{(k_j)}(x, T_{j+1}) - (t - T_j)^{-\alpha} \varepsilon_j^{(k_j)}(x, T_j) \right] \\ &= f'(x, t, u_{i,*}^{(k)}(x, t)) \varepsilon_i^{(k)}(x, t) - \frac{1}{\Gamma(1-\alpha)} (t - T_i)^{-\alpha} \varepsilon_{i-1}^{(k_{i-1})}(x, T_i) \\ &\quad + \frac{\alpha}{\Gamma(1-\alpha)} \sum_{j=0}^{i-1} \int_{T_j}^{T_{j+1}} (t-s)^{-\alpha-1} \varepsilon_j^{(k_j)}(x, s) ds, \end{aligned}$$

where $u_{i,*}^{(k)} \in \left[\min(u_i^{(k)}(x, t), u_i(x, t)), \max(u_i^{(k)}(x, t), u_i(x, t)) \right]$. Then,

$$\begin{aligned} \max_{0 \leq x \leq L} |g_i(x, t)| &\leq M \max_{0 \leq x \leq L} |\varepsilon_i^{(k)}(x, t)| + \frac{1}{\Gamma(1-\alpha)} (t - T_i)^{-\alpha} \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_{i-1})}(x, T_i)| \\ &\quad + \frac{1}{\Gamma(1-\alpha)} (t - T_i)^{-\alpha} \max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)|, \end{aligned} \tag{23}$$

where M is an upper bound of $|f'(x, t, u_{i,*}^{(k)}(x, t))|$. Taking inequality (23) into inequality (22), we get

$$\begin{aligned} \max_{0 \leq x \leq L} |\varepsilon_i^{(k+1)}(x, t)| &\leq M_i \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_i-1)}(x, T_i)| + MM_i \int_{T_i}^t (t-s)^{\alpha-1} \max_{0 \leq x \leq L} |\varepsilon_i^{(k)}(x, s)| ds \\ &\quad + \frac{M_i}{\Gamma(1-\alpha)} \int_{T_i}^t (t-s)^{\alpha-1} (s-T_i)^{-\alpha} ds \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_i-1)}(x, T_i)| \\ &\quad + \frac{M_i}{\Gamma(1-\alpha)} \int_{T_i}^t (t-s)^{\alpha-1} (s-T_i)^{-\alpha} ds \max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)| \\ &\leq C_i \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_i-1)}(x, T_i)| + \bar{M}_i \max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)| \\ &\quad + MM_i \int_{T_i}^t (t-s)^{\alpha-1} \max_{0 \leq x \leq L} |\varepsilon_i^{(k)}(x, s)| ds, \end{aligned}$$

where $\bar{M}_i = \frac{M_i}{\Gamma(1-\alpha)} B(\alpha, 1-\alpha) = \frac{M_i}{\Gamma(1-\alpha)} \frac{\pi}{\sin(\pi\alpha)}$, and $C_i = M_i + \bar{M}_i$. After k_i iterations of WR on $[T_i, T_{i+1}]$, we have by Lemma 3.1 that

$$\begin{aligned} &\max_{0 \leq x \leq L} |\varepsilon_i^{(k_i)}(x, t)| \\ &\leq \left[C_i \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_i-1)}(x, T_i)| + \bar{M}_i \max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)| \right] \\ &\quad \sum_{j=0}^{k_i-1} \frac{(MM_i \Gamma(\alpha)(t-T_i)^\alpha)^j}{\Gamma(j\alpha+1)} \\ &\quad + \frac{(MM_i \Gamma(\alpha)(t-T_i)^\alpha)^{k_i}}{\Gamma(k_i\alpha+1)} \max_{T_i \leq s \leq t} \max_{0 \leq x \leq L} |\varepsilon_i^{(0)}(x, s)|, \end{aligned}$$

for $t \in [T_i, T_{i+1}]$. We notice that

$$\max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_i-1)}(x, T_i)| \leq \max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)|.$$

For simplicity, we further denote

$$\begin{aligned} \Upsilon_i(t) &= \sum_{j=0}^{k_i-1} \frac{(MM_i \Gamma(\alpha)(t-T_i)^\alpha)^j}{\Gamma(j\alpha+1)}, \\ \Phi_i(t) &= \frac{(MM_i \Gamma(\alpha)(t-T_i)^\alpha)^{k_i}}{\Gamma(k_i\alpha+1)}, \quad t \in [T_i, T_{i+1}]. \end{aligned}$$

Then, we have

$$\begin{aligned} \max_{0 \leq x \leq L} |\varepsilon_i^{(k_i)}(x, t)| &\leq \Theta_i \Upsilon_i(t) \max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)| \\ &\quad + \Phi_i(t) \max_{T_i \leq s \leq t} \max_{0 \leq x \leq L} |\varepsilon_i^{(0)}(x, s)|. \end{aligned} \tag{24}$$

Since $\Theta_i \geq 1$, and $\Upsilon_i(t) \geq 1$, we can deduce from inequality (24) that the iteration errors accumulate along the windows line, i.e.,

$$\max_{0 \leq j \leq i-1} \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(k_j)}(x, t)| \leq \max_{T_{i-1} \leq t \leq T_i} \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_{i-1})}(x, t)|,$$

which leads to

$$\begin{aligned} \max_{0 \leq x \leq L} |\varepsilon_i^{(k_i)}(x, t)| &\leq \Theta_i \Upsilon_i(t) \max_{T_{i-1} \leq t \leq T_i} \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_{i-1})}(x, t)| \\ &\quad + \Phi_i(t) \max_{T_i \leq s \leq t} \max_{0 \leq x \leq L} |\varepsilon_i^{(0)}(x, s)|, \end{aligned}$$

and further we have

$$\begin{aligned} \max_{T_i \leq t \leq T_{i+1}} \max_{0 \leq x \leq L} |\varepsilon_i^{(k_i)}(x, t)| &\leq \Theta_i \tilde{\Upsilon}_i \max_{T_{i-1} \leq t \leq T_i} \max_{0 \leq x \leq L} |\varepsilon_{i-1}^{(k_{i-1})}(x, t)| \\ &\quad + \tilde{\Phi}_i \max_{T_i \leq t \leq T_{i+1}} \max_{0 \leq x \leq L} |\varepsilon_i^{(0)}(x, t)|. \end{aligned}$$

By recurrence, we get

$$\max_{T_i \leq t \leq T_{i+1}} \max_{0 \leq x \leq L} |\varepsilon_i^{(k_i)}(x, t)| \leq \sum_{j=0}^i \tilde{\Phi}_j \left(\prod_{l=j+1}^i \Theta_l \tilde{\Upsilon}_l \right) \max_{T_j \leq t \leq T_{j+1}} \max_{0 \leq x \leq L} |\varepsilon_j^{(0)}(x, t)|,$$

The conclusion results. □

4 Discrete windowing WR method

In fact, the standard WR method introduced in Section 2 is a special case of the windowing WR method. In this section, we only consider the discrete version of the windowing WR method.

4.1 Construction of the discrete windowing WR method

For simplicity, we suppose that all the time windows have the same lengths, i.e., $H_i = H$, for $i = 0, 1, \dots, N-1$. Let N_0 and N_s be two positive integers. We partition the spatial domain $[0, L]$ by a uniform mesh, $x_l = lh$, for $l = 0, 1, \dots, N_s$, with $h = \frac{L}{N_s}$, and further partition the time window Ω_i by a uniform mesh $t_{i,n} = T_i + n\tau$, for $n = 0, 1, \dots, N_0$, with $\tau = \frac{H}{N_0}$, which satisfy $t_{i,N_0} = T_{i+1}$.

At any mesh point $(x_l, t_{i,n})$, the windowing WR method (16) is

$$\begin{aligned}
 & C_T \mathcal{D}_{t_{i,n}}^\alpha u_i^{(k+1)}(x_l, t_{i,n}) + \sum_{j=0}^{i-1} \frac{1}{\Gamma(1-\alpha)} \int_{T_j}^{T_{j+1}} (t_{i,n} - s)^{-\alpha} \frac{\partial}{\partial s} u_j^{(k)}(x_l, s) ds \\
 & = K \frac{\partial^2}{\partial x^2} u_i^{(k+1)}(x_l, t_{i,n}) + f(x_l, t_{i,n}, u_i^{(k)}(x_l, t_{i,n})),
 \end{aligned} \tag{25}$$

for $l = 1, 2, \dots, N_s - 1$, $i = 0, 1, \dots, N - 1$, and $n = 0, 1, \dots, N_0$. We denote by $[u_i^{(k)}]_l^p$ the approximation of $u_i^{(k)}(x_l, t_{i,p})$, and $[u_i]_l^p$ the approximation of $u_i(x_l, t_{i,p})$. We approximate the term $\frac{\partial^2}{\partial x^2} u_i^{(k+1)}(x_l, t_{i,n})$ in the right-hand side of (25) by the classical finite difference operation $\delta_x^2 [u_i^{(k+1)}]_l^n$, and adopt the idea of L1 scheme [29] to approximate the terms in the left-hand side of (25). Then, we have

$$\begin{aligned}
 C_T \mathcal{D}_{t_{i,n}}^\alpha u_i^{(k+1)}(x_l, t_{i,n}) & = \frac{1}{\Gamma(1-\alpha)} \sum_{p=0}^{n-1} \int_{t_{i,p}}^{t_{i,p+1}} (t_{i,n} - s)^{-\alpha} \frac{\partial}{\partial s} u_i^{(k+1)}(x_l, s) ds \\
 & \approx \frac{1}{\Gamma(1-\alpha)} \sum_{p=0}^{n-1} \frac{1}{\tau} \left([u_i^{(k+1)}]_l^{p+1} - [u_i^{(k+1)}]_l^p \right) \int_{t_{i,p}}^{t_{i,p+1}} (t_{i,n} - s)^{-\alpha} ds \\
 & = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \left(a_1 [u_i^{(k+1)}]_l^n - \sum_{p=1}^{n-1} (a_{n-p} - a_{n-p+1}) [u_i^{(k+1)}]_l^p - a_n [u_i^{(k+1)}]_l^0 \right) \\
 & := \mathbb{D}_\tau^\alpha [u_i^{(k+1)}]_l^n,
 \end{aligned}$$

where

$$a_q = q^{1-\alpha} - (q - 1)^{1-\alpha}, \quad q \geq 1.$$

For each term in the summation in the left side of (25), we have

$$\begin{aligned}
 & \frac{1}{\Gamma(1-\alpha)} \int_{T_j}^{T_{j+1}} (t_{i,n} - s)^{-\alpha} \frac{\partial}{\partial s} u_j^{(k)}(x_l, s) ds \\
 & \approx \frac{1}{\Gamma(1-\alpha)} \sum_{p=0}^{N_0-1} \frac{1}{\tau} \left([u_j^{(k)}]_l^{p+1} - [u_j^{(k)}]_l^p \right) \int_{t_{j,p}}^{t_{j,p+1}} (t_{i,n} - s)^{-\alpha} ds \\
 & = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \left(a_{(i-j-1)N_0+n+1} [u_j^{(k)}]_l^{N_0} - \sum_{p=1}^{N_0-1} (a_{(i-j)N_0+n-p} \right. \\
 & \quad \left. - a_{(i-j)N_0+n-p+1}) [u_j^{(k)}]_l^p - a_{(i-j)N_0+n} [u_j^{(k)}]_l^0 \right) := \mathbb{D}_\tau^\alpha [u_j^{(k)}]_l^n.
 \end{aligned}$$

The resulting numerical scheme for (25) can be written as

$$\mathbb{D}_\tau^\alpha [u_i^{(k+1)}]_l^n + \sum_{j=0}^{i-1} \mathbb{D}_\tau^\alpha [u_j^{(k)}]_l^n = K \delta_x^2 [u_i^{(k+1)}]_l^n + f(x_l, t_{i,n}, [u_i^{(k)}]_l^n). \tag{26}$$

Similarly, the $L1$ scheme for the original equation can be written as

$$\mathbb{D}_\tau^\alpha [u_i]_l^n + \sum_{j=0}^{i-1} \mathbb{D}_\tau^\alpha [u_j]_l^n = K \delta_x^2 [u_i]_l^n + f(x_l, t_{i,n}, [u_i]_l^n). \tag{27}$$

4.2 Convergence of the discrete windowing WR method

Let $[\varepsilon_i^{(k)}]_l^n = [u_i^{(k)}]_l^n - [u_i]_l^n$, for $l = 1, 2, \dots, N_s - 1, i = 0, 1, \dots, N - 1, n = 0, 1, \dots, N_0$ and $k = 0, 1, \dots, k_i$, which satisfies

$$\begin{aligned} \mathbb{D}_\tau^\alpha [\varepsilon_i^{(k+1)}]_l^n + \sum_{j=0}^{i-1} \mathbb{D}_\tau^\alpha [\varepsilon_j^{(k+1)}]_l^n &= K \delta_x^2 [\varepsilon_i^{(k+1)}]_l^n + f(x_l, t_{i,n}, [u_i^{(k)}]_l^n) \\ &- f(x_l, t_{i,n}, [u_i]_l^n). \end{aligned} \tag{28}$$

We denote the vectors

$$\begin{aligned} [\mathbf{u}_i^{(k)}]_l^n &= \left([u_i^{(k)}]_1^n, [u_i^{(k)}]_2^n, \dots, [u_i^{(k)}]_{N_s-1}^n \right)^\top \in \mathbb{R}^{N_s-1}, \\ [\mathbf{u}_i]_l^n &= ([u_i]_1^n, [u_i]_2^n, \dots, [u_i]_{N_s-1}^n)^\top \in \mathbb{R}^{N_s-1}, \\ [\boldsymbol{\varepsilon}_i^{(k)}]_l^n &= [\mathbf{u}_i^{(k)}]_l^n - [\mathbf{u}_i]_l^n, \end{aligned}$$

together with the fact $[\boldsymbol{\varepsilon}_j^{(k)}]_l^{N_0} = [\boldsymbol{\varepsilon}_{j+1}^{(k+1)}]_l^0$, for $j = 0, 1, \dots, i - 1$, and $[\boldsymbol{\varepsilon}_0^{(k_0)}]_l^0 = \mathbf{0}$, we have

$$\begin{aligned} &\frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \left(a_1 [\boldsymbol{\varepsilon}_i^{(k+1)}]_l^n - \sum_{p=1}^{n-1} (a_{n-p} - a_{n-p+1}) [\boldsymbol{\varepsilon}_i^{(k+1)}]_l^p \right. \\ &\quad \left. - \sum_{j=0}^{i-1} \sum_{p=1}^{N_0} (a_{(i-j)N_0+n-p} - a_{(i-j)N_0+n-p+1}) [\boldsymbol{\varepsilon}_j^{(k_j)}]_l^p \right) \\ &= \frac{K}{h^2} \mathbf{B} [\boldsymbol{\varepsilon}_i^{(k+1)}]_l^n + \mathbf{f}(t_{i,n}, [\mathbf{u}_i^{(k)}]_l^n) - \mathbf{f}(t_{i,n}, [\mathbf{u}_i]_l^n), \end{aligned} \tag{29}$$

where

$$\begin{aligned} \mathbf{f}(t_{i,n}, [\mathbf{u}_i^{(k)}]_l^n) &= \left(f(x_1, t_{i,n}, [u_i^{(k)}]_1^n), f(x_2, t_{i,n}, [u_i^{(k)}]_2^n), \dots, \right. \\ &\quad \left. f(x_{N_s-1}, t_{i,n}, [u_i^{(k)}]_{N_s-1}^n) \right)^\top, \\ \mathbf{f}(t_{i,n}, [\mathbf{u}_i]_l^n) &= \left(f(x_1, t_{i,n}, [u_i]_1^n), f(x_2, t_{i,n}, [u_i]_2^n), \dots, f(x_{N_s-1}, t_{i,n}, [u_i]_{N_s-1}^n) \right)^\top, \\ [\boldsymbol{\varepsilon}_i^{(k)}]_l^n &= [\mathbf{u}_i^{(k)}]_l^n - [\mathbf{u}_i]_l^n, \end{aligned}$$

and

$$B = \begin{pmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \dots & \\ 0 & 1 & \dots & \dots & 0 \\ \dots & \dots & \dots & \dots & 1 \\ 0 & \dots & 0 & 1 & -2 \end{pmatrix} \in \mathbb{R}^{(N_s-1) \times (N_s-1)}.$$

We further define the vector

$$\bar{\mathbf{e}}_i^{(k)} = \left(\left([\mathbf{e}_i^{(k)}]^1 \right)^\top, \left([\mathbf{e}_i^{(k)}]^2 \right)^\top, \dots, \left([\mathbf{e}_i^{(k)}]^{N_0} \right)^\top \right)^\top \in \mathbb{R}^{N_0(N_s-1)}.$$

Then, equality (29) can be rewritten in the following matrix form,

$$\begin{aligned} & (\mathbf{A} \otimes \mathbf{I}_s) \bar{\mathbf{e}}_i^{(k+1)} - \sum_{j=0}^{i-1} (\mathbf{G}_{ij} \otimes \mathbf{I}_s) \bar{\mathbf{e}}_j^{(k_j)} \\ &= \Gamma(2 - \alpha) \frac{K \tau^\alpha}{h^2} (\mathbf{I}_0 \otimes \mathbf{B}) \bar{\mathbf{e}}_i^{(k+1)} + \tau^\alpha \Gamma(2 - \alpha) \left(\bar{\mathbf{f}}(t_i, \bar{\mathbf{u}}_i^{(k)}) - \bar{\mathbf{f}}(t_i, \bar{\mathbf{u}}_i) \right), \end{aligned} \tag{30}$$

where \mathbf{I}_s and \mathbf{I}_0 are identity matrixes with scales $N_s - 1$ and N_0 , respectively, \mathbf{A} is a $N_0 \times N_0$ lower triangular Toeplitz matrix with

$$A(p, q) = \begin{cases} a_1, & p = q, \\ a_{p-q+1} - a_{p-q}, & p > q, \\ 0, & \text{else,} \end{cases}$$

\mathbf{G}_{ij} is a $N_0 \times N_0$ Toeplitz matrix with

$$\mathbf{G}_{ij}(p, q) = a_{(i-j)N_0+p-q} - a_{(i-j)N_0+p-q+1},$$

and

$$\begin{aligned} \bar{\mathbf{f}}(t_i, \bar{\mathbf{u}}_i^{(k)}) &= \left(\mathbf{f}^\top \left(t_{i,1}, [\mathbf{u}_i^{(k)}]^1 \right), \mathbf{f}^\top \left(t_{i,2}, [\mathbf{u}_i^{(k)}]^2 \right), \dots, \mathbf{f}^\top \left(t_{i,N_0}, [\mathbf{u}_i^{(k)}]^{N_0} \right) \right)^\top, \\ \bar{\mathbf{f}}(t_i, \bar{\mathbf{u}}_i) &= \left(\mathbf{f}^\top \left(t_{i,1}, [\mathbf{u}_i]^1 \right), \mathbf{f}^\top \left(t_{i,2}, [\mathbf{u}_i]^2 \right), \dots, \mathbf{f}^\top \left(t_{i,N_0}, [\mathbf{u}_i]^{N_0} \right) \right)^\top. \end{aligned}$$

We denote $r = \Gamma(2 - \alpha) \frac{K \tau^\alpha}{h^2}$ and $\mathbf{D} = \mathbf{A} \otimes \mathbf{I}_s - r(\mathbf{I}_0 \otimes \mathbf{B})$, the error (30) can be written as

$$\bar{\mathbf{e}}_i^{(k+1)} = \tau^\alpha \Gamma(2 - \alpha) \mathbf{D}^{-1} \mathbf{J}_f \bar{\mathbf{e}}_i^{(k)} + \sum_{j=0}^{i-1} \mathbf{D}^{-1} (\mathbf{G}_{ij} \otimes \mathbf{I}_s) \bar{\mathbf{e}}_j^{(k_j)}, \tag{31}$$

where \mathbf{J}_f is the Jacobian matrix of $\bar{\mathbf{f}}$, which is bounded by a positive M_0 . Then, we have

$$\|\bar{\mathbf{e}}_i^{(k+1)}\|_\infty \leq \tau^\alpha \Gamma(2 - \alpha) M_0 \|\mathbf{D}^{-1}\|_\infty \|\bar{\mathbf{e}}_i^{(k)}\|_\infty + \sum_{j=0}^{i-1} \|\mathbf{D}^{-1}\|_\infty \|\mathbf{G}_{ij}\|_\infty \|\bar{\mathbf{e}}_j^{(k_j)}\|_\infty.$$

By induction on the index k , it leads to

$$\|\bar{\mathbf{e}}_i^{(k_i)}\|_\infty \leq \theta^{k_i} \|\bar{\mathbf{e}}_i^{(0)}\|_\infty + \sum_{j=0}^{i-1} \psi_{ij} \|\bar{\mathbf{e}}_j^{(k_j)}\|_\infty, \tag{32}$$

where

$$\theta = \tau^\alpha \Gamma(2 - \alpha) M_0 \|\mathbf{D}^{-1}\|_\infty, \quad \psi_{ij} = \frac{1}{1 - \theta} \|\mathbf{D}^{-1}\|_\infty \|\mathbf{G}_{ij}\|_\infty. \tag{33}$$

We can choose τ small enough, such that $\theta < 1$, and $\psi_{ij} > 0$.

Define the real numbers η_{ij} , for $i = 0, 1, \dots, N - 1$, and $j = 0, 1, \dots, i - 1$ by

$$\eta_{ii} = 1, \quad \eta_{ij} = \sum_{l=j}^{i-1} \psi_{il} \eta_{lj}. \tag{34}$$

We have the following theorem.

Theorem 4.1 *If the nonlinear function $f(x, t, u)$ has a continuous first-order derivative with respect to u , then the iteration error $\bar{\mathbf{e}}_i^{(k_i)}$ of the windowing WR method (26) on the i -th time window Ω_i can be bounded by*

$$\|\bar{\mathbf{e}}_i^{(k_i)}\|_\infty \leq \sum_{j=0}^i \eta_{ij} \theta^{k_j} \|\bar{\mathbf{e}}_j^{(0)}\|_\infty, \quad i = 0, 1, \dots, N - 1, \tag{35}$$

where k_j is the maximum number of WR iterations on Ω_j .

Proof We prove the result by induction on the index i .

For $i = 0$, on the first window, we obtain that by (32)

$$\|\bar{\mathbf{e}}_0^{(k_0)}\|_\infty \leq \theta^{k_0} \|\bar{\mathbf{e}}_0^{(0)}\|_\infty = \eta_{00} \theta^{k_0} \|\bar{\mathbf{e}}_0^{(0)}\|_\infty.$$

Assume that inequality (35) holds for $i = 1, 2, \dots, n - 1$, then we have

$$\|\bar{\mathbf{e}}_n^{(k_n)}\|_\infty \leq \theta^{k_n} \|\bar{\mathbf{e}}_n^{(0)}\|_\infty + \sum_{j=0}^{n-1} \psi_{nj} \|\bar{\mathbf{e}}_j^{(k_j)}\|_\infty \leq \theta^{k_n} \|\bar{\mathbf{e}}_n^{(0)}\|_\infty + \sum_{j=0}^{n-1} \psi_{nj} \sum_{l=0}^j \eta_{jl} \theta^{k_l} \|\bar{\mathbf{e}}_l^{(0)}\|_\infty.$$

By exchanging the summation order, we can obtain

$$\begin{aligned} \|\bar{\mathbf{e}}_n^{(k_n)}\|_\infty &\leq \theta^{k_n} \|\bar{\mathbf{e}}_n^{(0)}\|_\infty + \sum_{l=0}^{n-1} \left(\sum_{j=l}^{n-1} \psi_{nj} \eta_{jl} \right) \theta^{k_l} \|\bar{\mathbf{e}}_l^{(0)}\|_\infty \\ &= \eta_{nn} \theta^{k_n} \|\bar{\mathbf{e}}_n^{(0)}\|_\infty + \sum_{l=0}^{n-1} \eta_{nl} \theta^{k_l} \|\bar{\mathbf{e}}_l^{(0)}\|_\infty \\ &= \sum_{l=0}^n \eta_{nl} \theta^{k_l} \|\bar{\mathbf{e}}_l^{(0)}\|_\infty. \end{aligned}$$

It means that the result also holds for $i = n$, and the theorem is proved. □

It can be seen that $\{\eta_{ij}\}$ are defined by recurrence, and it is not intuitive to bound $\|\bar{\mathbf{e}}_i^{(k_i)}\|_\infty$. Therefore, we present the following corollary.

Corollary 4.1 *The iteration error $\bar{\mathbf{e}}_i^{(k_i)}$ can be further bounded by*

$$\|\bar{\mathbf{e}}_i^{(k_i)}\|_\infty \leq \max \left\{ 1, \frac{\|\mathbf{D}^{-1}\|_\infty^i}{(1-\theta)^i} \right\} \theta^{k_{min}} \max_{0 \leq j \leq i-1} \|\bar{\mathbf{e}}_j^{(0)}\|_\infty, \quad i = 0, 1, \dots, N-1, \tag{36}$$

where $k_{min} = \min_{0 \leq j \leq i-1} \{k_j\}$, and θ is defined in (33) with $\theta < 1$.

Proof From estimation (35), we have

$$\|\bar{\mathbf{e}}_i^{(k_i)}\|_\infty \leq \left(\sum_{j=0}^i \eta_{ij} \right) \theta^{k_{min}} \max_{0 \leq j \leq i-1} \|\bar{\mathbf{e}}_j^{(0)}\|_\infty. \tag{37}$$

Denote $W_i = \sum_{j=0}^i \eta_{ij}$, then we have $W_0 = 1$, and for $i = 1, 2, \dots, N-1$,

$$\begin{aligned} W_i &= \sum_{j=0}^i \sum_{l=j}^{i-1} \psi_{il} \eta_{lj} = \sum_{l=0}^{i-1} \psi_{il} \sum_{j=0}^l \eta_{lj} = \sum_{l=0}^{i-1} \psi_{il} W_l \\ &\leq \left(\sum_{l=0}^{i-1} \psi_{il} \right) \max_{0 \leq l \leq i-1} W_l \\ &= \frac{1}{1-\theta} \|\mathbf{D}^{-1}\|_\infty \left(\sum_{j=0}^{i-1} \|\mathbf{G}_{ij}\|_\infty \right) \max_{0 \leq l \leq i-1} W_l. \end{aligned} \tag{38}$$

Since \mathbf{G}_{ij} is a Toeplitz matrix, and

$$\sum_{j=0}^{i-1} \|\mathbf{G}_{ij}\|_\infty = \sum_{j=0}^{i-1} (a_{(i-j-1)N_0+1} - a_{(i-j)N_0+1}) = 1 + (iN_0)^{1-\alpha} - (iN_0+1)^{1-\alpha} \leq 1.$$

We have

$$W_i \leq \frac{1}{1-\theta} \|\mathbf{D}^{-1}\|_\infty \max_{0 \leq l \leq i-1} W_l.$$

By the induction on the index i , we get

$$W_i \leq \max \left\{ 1, \frac{\|\mathbf{D}^{-1}\|_\infty^i}{(1-\theta)^i} \right\}.$$

Together with inequality (37), we can obtain the result (36). □

Remark The quantity $\|\mathbf{D}^{-1}\|_\infty$ in estimation (36) is usually a moderate number. In fact, a number of numerical tests are carried out to support this observation. Therefore, we can take a proper WR iteration number k_i on each time window to control the iteration errors.

4.3 Parallelism of the discrete windowing WR method

The discrete windowing WR method (26) can be written in the following form,

$$\begin{aligned} & \left(\frac{\tau^{-\alpha} a_1}{\Gamma(2-\alpha)} \mathbf{I}_s - \frac{K}{h^2} \mathbf{B} \right) \left[\mathbf{u}_i^{(k+1)} \right]^n = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \\ & \times \left(\sum_{p=1}^{n-1} (a_{n-p} - a_{n-p+1}) \left[\mathbf{u}_i^{(k+1)} \right]^p + \sum_{j=0}^{i-1} \sum_{p=1}^{N_0} (a_{(i-j)N_0+n-p} \right. \\ & \left. - a_{(i-j)N_0+n-p+1}) \left[\mathbf{u}_j^{(k_j)} \right]^p \right) \\ & + \mathbf{f} \left(t_{i,n}, \left[\mathbf{u}_i^{(k)} \right]^n \right). \end{aligned} \tag{39}$$

For any fixed $n = 1, 2, \dots, N_0$, in order to compute $\left[\mathbf{u}_i^{(k+1)} \right]^n$, we need $\left[\mathbf{u}_i^{(k+1)} \right]^p$ on the present waveform for $p = 1, 2, \dots, n - 1$, and $\left[\mathbf{u}_i^{(k)} \right]^n$, which is the approximation at $t = t_i + n\tau$ on the last waveform, as well as $\left[\mathbf{u}_j^{(k_j)} \right]^p$, for $p = 1, 2, \dots, N_0$ and $j = 0, 1, \dots, i - 1$, which are the approximations of the waveforms on all the previous time windows. It is unnecessary to wait for all the approximations on the k -th waveform before computing the approximations on the $(k + 1)$ -th waveform.

For simplicity, we assume that the WR methods on all the time windows have the same number of iterations, say k_0 . The parallelism of the discrete windowing WR method (39) is shown in Fig. 1, where “dt” denotes the computational cost at each time step. In fact, for the computation of the time-fractional PDE, the computational costs at later time steps are usually bigger than that at former time steps, since more matrix-vector multiplications are involved in later time steps, though “dt” is used uniformly for clarity.

Assume that k_0 processors are available, and each processor is in charge of the computation of a waveform. We can see from Fig. 1 that k_0 processors are running almost simultaneously on each time window, such that the theoretical total running time is $N(N_0 + k_0)dt$, rather than Nk_0N_0dt in serial manner.

The steps of the parallel procedure are shown in the following algorithm,

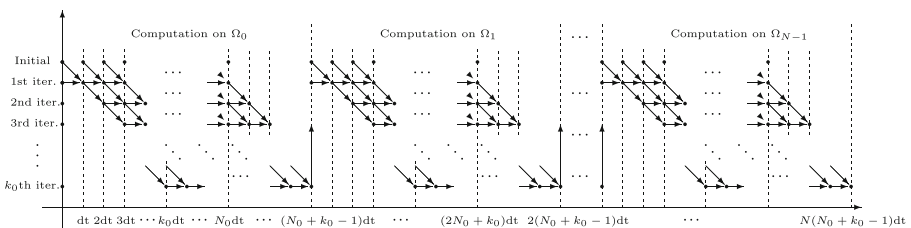


Fig. 1 The parallelism of the discrete windowing WR scheme (39)

Algorithm 1

- (1) The No. k processor obtain initial values $\left[\mathbf{u}_0^{(k)} \right]^0$, for $k = 1, 2, \dots, k_0$.
 - (2) **do** $i = 0, 1, \dots, N - 1$,
on the time window Ω_i
 - (2.1) The No.1 processor receives the initial guess $\left[\mathbf{u}_i^{(0)} \right]^n$, for $n = 0, 1, \dots, N_0$.
 - (2.2) [Parallel] For all the processors from the No.1 to the No. k_0 , simultaneously,
do $n = 1, 2, \dots, N_0$,
 - (2.2.1) The No. k processor (except No.1 processor) waits for $\left[\mathbf{u}_i^{(k-1)} \right]^n$.
 - (2.2.2) The No. k processor computes $\left[\mathbf{u}_i^{(k)} \right]^n$ by scheme (39), as long as it receives $\left[\mathbf{u}_i^{(k-1)} \right]^n$.
 - (2.2.3) The No. k processor (except No. k_0 processor) sends $\left[\mathbf{u}_i^{(k)} \right]^n$ to the No. $(k + 1)$ processor.
 - end do**
 - (2.3) The No. k_0 processor broadcasts $\left[\mathbf{u}_i^{(k_0)} \right]^n$ for $n = 0, 1, \dots, N_0$ to the other processors as the necessary data for the computation on the later time windows.
- end do**
- (3) Collect the results of the No. k_0 processor as the final approximations to the original system.
-

4.4 Efficient implementation of the discrete windowing WR method

We can see from (26) that the discrete windowing WR method is based on the idea of $L1$ scheme, which is computationally expensive on long time interval. In this subsection, we employ the sum-of-exponentials (SOE) method [30, 31] to further reduce the memory and the computational cost. The fast evaluation formula based on SOE method for the Caputo fractional derivative in (25) can be expressed as

$$\begin{aligned}
 & \frac{\left[\mathbf{u}_i^{(k+1)} \right]_l^n - \left[\mathbf{u}_i^{(k+1)} \right]_l^{n-1}}{\tau^\alpha \Gamma(2 - \alpha)} \\
 & + \frac{1}{\Gamma(1 - \alpha)} \left[\frac{\left[\mathbf{u}_i^{(k+1)} \right]_l^{n-1}}{\tau^\alpha} - \frac{\left[\mathbf{u}_0^{(0)} \right]_l^0}{t_{i,n}^\alpha} - \alpha \sum_{j=1}^{N_{exp}} \omega_j \left[U_{hist,j}^{(k+1)}(t_{i,n}) \right]_l \right] \quad (40) \\
 & = K \delta_x^2 \left[\mathbf{u}_i^{(k+1)} \right]_l^n + f(x_l, t_{i,n}, \left[\mathbf{u}_i^{(k)} \right]_l^n).
 \end{aligned}$$

At the initial point, we have

$$\begin{cases} [u_0^{(k)}]_l^0 = \varphi(x_l), \\ [U_{hist,j}^{(k)}(t_{0,1})]_l = 0, \quad \text{for } k = 1, 2, \dots, k_0, \end{cases}$$

and at the boundaries of any two time windows, we have

$$\begin{cases} [u_{i+1}^{(k)}]_l^0 = [u_i^{(k_i)}]_l^{N_0}, \\ U_{hist,j}^{(k)}(t_{i+1,0}) = U_{hist,j}^{(k_i)}(t_i, N_0), \quad \text{for } k = 1, 2, \dots, k_0. \end{cases}$$

It means that the discrete windowing WR method is implemented on one time window after another, and the WR iterations are performed k_i times on the i -th window.

In fact, the fast evaluation formula based on SOE method transforms the discrete windowing WR method as a nonlocal numerical integrator to be a local evaluation, and the computational cost at each time point looks more balanced. Suppose that we have enough processors and denote by $\tilde{\tau}$ the computational cost at each time point, then the theoretical computational cost of the fast windowing WR method is $N(N_0 + k_0 - 1)\tilde{\tau}$.

For the choice of the parameters, we have two considerations. First, k_0 should be as small as possible to avoid too much computational cost. Second, if the time window is too short, the number of time steps on each window, N_0 , will be very small, then the degree of parallelism will be reduced. In particular, if only one time step is contained in each window, i.e., $N_0 = 1$, the WR method will deteriorate to the classical semi-implicit scheme, which cannot be implemented in parallel at all. In order to balance the two aspects, we can take moderate values for N_0 , such that N_0 is much larger than k_0 . Therefore, the theoretical computational cost with enough processors seems slightly more than the cost of a general numerical scheme for a linear problem.

5 Numerical experiments

In this section, we use two examples to verify the effectiveness of the windowing WR methods, and the methods are coded using Matlab R2017b on a Lenovo desktop with Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz.

Example 5.1 We first consider the following sub-diffusion equation in one spatial dimension,

$$\begin{cases} {}^C_0D_t^\alpha u(x, t) = 0.2u_{xx}(x, t) + u(x, t) + u^2(x, t), & 0 < x < 1, \quad 0 < t < 2, \\ u(x, 0) = \sin(\pi x), & 0 \leq x \leq 1, \\ u(0, t) = u(1, t) = 0, & 0 \leq t \leq 2. \end{cases} \tag{41}$$

To discretize the x and t variables, we use a uniform space-time mesh with time step $\tau = \frac{1}{2048}$ and spatial grid size $h = \frac{1}{256}$. The time interval $[0, 2]$ is divided into N time windows uniformly, and each window includes $\frac{4096}{N}$ time steps. For simplicity, we perform the same number of WR iterations on all time windows, i.e., $k_i = k_0$, for $i = 0, 1, \dots, N - 1$.

The discrete windowing WR method for (41) can be written as

$$\mathbb{D}_\tau^\alpha \left[u_i^{(k+1)} \right]_l^n + \sum_{j=0}^{i-1} \mathbb{D}_\tau^\alpha \left[u_j^{(k_0)} \right]_l^n = 0.2 \delta_x^2 \left[u_i^{(k+1)} \right]_l^n + \left[u_i^{(k)} \right]_l^{n-1} + \left(\left[u_i^{(k)} \right]_l^{n-1} \right)^2, \tag{42}$$

for $i = 0, 1, \dots, N - 1, l = 1, 2, \dots, 255, n = 1, 2, \dots, \frac{4096}{N}, k = 0, 1, \dots, k_0 - 1$, and the reference solution can be obtained by the scheme

$$\frac{\tau^{-\alpha}}{\Gamma(2 - \alpha)} \left(a_1 u_l^n - \sum_{p=1}^{n-1} (a_{n-p} - a_{n-p+1}) u_l^p - a_n u_l^0 \right) = 0.2 \delta_x^2 u_l^n + u_l^{n-1} + \left(u_l^{n-1} \right)^2, \tag{43}$$

for $l = 1, 2, \dots, 256$, and $n = 1, 2, \dots, 4096$. We notice from scheme (42) that the nonlinear term is approximated using the values at $t = t_{iN_0+n-1}$, in order to match with scheme (43). In fact, if we employ a fully implicit scheme in the discrete windowing WR method for (41), the resulting numerical scheme, i.e., the nonlinear term in (42) is replaced by $\left[u_i^{(k)} \right]_l^n + \left(\left[u_i^{(k)} \right]_l^n \right)^2$, will still be a linear equation, since the values are from the last waveform, while the corresponding reference equation is a nonlinear equation.

The iteration error of the windowing WR method is measured by

$$Error = \max_{1 \leq l \leq 256} \left| \left[u_{N-1}^{(k_0)} \right]_l^{N_0} - u_l^{NN_0} \right|.$$

The iteration errors for various values of k_0 and N are shown in Tables 1, 2, and 3, for $\alpha = 0.2, 0.5, 0.8$, respectively. In fact, the windowing WR method with only one time window, i.e., $N = 1$, is indeed the standard WR method. We can see from Tables 1, 2, and 3 that the windowing technique can reduce the number of iterations effectively.

To show the convergence of WR clearly, we plot the relationship between the iteration error and the iteration number k_0 with various time windows in the first three figures of Fig. 2, for $\alpha = 0.2, \alpha = 0.5$, and $\alpha = 0.8$, respectively. Furthermore, We also compare the convergence behaviors of WR with 64 time windows for various values of α in the last figure in Fig. 2. We can see that the convergence rates of the windowing WR method are related to the values of α , and such an observation fits very well with the theoretical results introduced in Sections 2–3.

We also test the efficiency of the fast evaluation formula of the WR method. We fix $NN_0 = 4096$, the observed errors, and the consumed CPU time for the discrete windowing WR method, and its fast evaluation version with tolerance $\epsilon=1.0e-10$ is shown in Table 4. We can see that the fast windowing WR method has almost the same errors as the windowing WR method which is based on $L1$ scheme, while the

Table 1 Errors of the windowing WR method (42) for Example 5.1 with $\alpha = 0.2$

k_0	$N = 1024$	$N = 256$	$N = 64$	$N = 16$	$N = 1$
1	3.83e-04	2.07e-03	9.54e-03	4.74e-02	2.90e-01
2	5.51e-05	5.35e-04	2.95e-03	1.53e-02	2.48e-01
3	6.39e-06	1.81e-04	1.24e-03	7.12e-03	2.07e-01
4	0	6.59e-05	5.82e-04	3.77e-03	1.70e-01
10	0	1.02e-07	1.07e-05	1.75e-04	3.99e-02
15	0	1.42e-11	3.87e-07	1.62e-05	9.92e-03
30	0	0	5.62e-12	1.03e-08	1.15e-04
40	0	0	1.11e-15	5.45e-11	5.01e-06
60	0	0	0	1.11e-15	6.74e-09
80	0	0	0	0	5.86e-12
100	0	0	0	0	3.66e-15

Table 2 Errors of the windowing WR method (42) for Example 5.1 with $\alpha = 0.5$

k_0	$N = 1024$	$N = 256$	$N = 64$	$N = 16$	$N = 1$
1	7.29e-04	3.84e-03	1.76e-02	7.74e-02	3.60e-01
2	1.85e-05	3.23e-04	2.75e-03	1.97e-02	3.07e-01
3	2.39e-07	2.76e-05	4.99e-04	5.99e-03	2.53e-01
4	0	2.12e-06	8.97e-05	1.92e-03	2.02e-01
10	0	2.52e-14	1.16e-09	1.48e-06	3.06e-02
16	0	0	3.44e-15	4.71e-10	2.64e-03
24	0	0	0	3.44e-15	5.58e-05
40	0	0	0	0	4.09e-09
60	0	0	0	0	1.89e-15

Table 3 Errors of the windowing WR method (42) for Example 5.1 with $\alpha = 0.8$

k_0	$N = 1024$	$N = 256$	$N = 64$	$N = 16$	$N = 1$
1	9.55e-04	4.80e-03	2.03e-02	8.09e-02	3.81e-01
2	3.01e-06	8.29e-05	1.13e-03	1.23e-02	3.16e-01
3	3.93e-09	1.13e-06	5.50e-05	1.75e-03	2.47e-01
4	0	1.24e-08	2.30e-06	2.23e-04	1.81e-01
7	0	6.00e-15	8.07e-11	2.52e-07	4.98e-02
12	0	0	1.89e-15	6.71e-13	2.27e-03
14	0	0	0	2.33e-15	5.05e-04
25	0	0	0	0	1.21e-08
37	0	0	0	0	3.00e-15

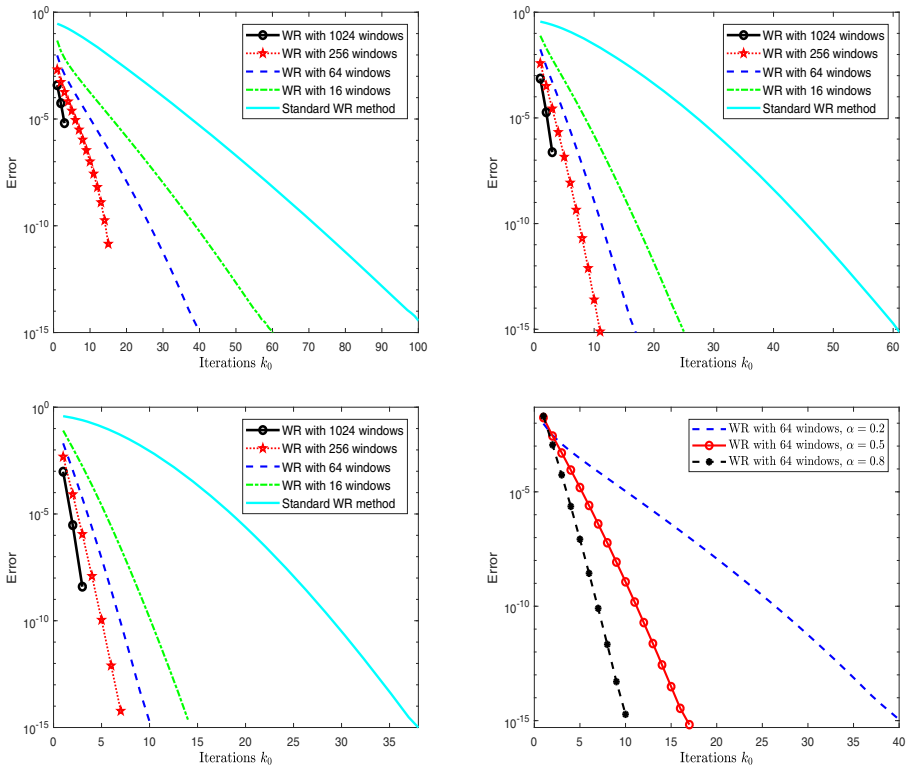


Fig. 2 Errors of windowing WR for Example 5.1 with $\alpha = 0.2$ (top left), $\alpha = 0.5$ (top right), and $\alpha = 0.8$ (bottom left), and errors of WR with 64 windows for various values of α (bottom right)

Table 4 Comparisons of the discrete WR method and its fast version for Example 5.1

	N	$\alpha = 0.2, k_0 = 9$		$\alpha = 0.5, k_0 = 7$		$\alpha = 0.8, k_0 = 4$	
		Error	Time (s)	Error	Time (s)	Error	Time (s)
WR	16	2.83e-04	63.31	5.95e-05	43.68	2.23e-04	27.82
	32	8.18e-05	62.93	5.55e-06	44.31	2.41e-05	27.83
	64	2.05e-05	62.10	3.97e-07	44.03	2.30e-06	28.06
	128	3.90e-06	62.21	1.95e-08	43.77	1.91e-07	27.69
	256	3.44e-07	62.30	4.47e-10	45.03	1.24e-08	27.90
Fast WR	16	2.83e-04	6.96	5.95e-05	5.47	2.23e-04	3.16
	32	8.18e-05	6.85	5.55e-06	5.59	2.41e-05	3.20
	64	2.05e-05	6.88	3.97e-07	5.41	2.30e-06	3.45
	128	3.90e-06	6.91	1.95e-08	5.55	1.91e-07	3.16
	256	3.44e-07	6.76	4.47e-10	6.00	1.24e-08	3.06

fast version costs much less CPU time. For example, when $N = 16$, $\alpha = 0.2$, and $k_0 = 9$, the two methods have the same observed error $2.83\text{e-}04$, but the consumed CPU time of the fast windowing WR method has reduced from 63 to 7 s.

Based on the fast evaluation formula of the WR method, we also investigate the relationship between the number of windows N and the number of iterations k_0 . We fix the number of time steps $NN_0 = 4800$ on the whole time interval, and denote k_0 as the minimum uniform number of WR iterations on each time window, such that the iteration error below $2.0\text{e-}10$ and the relationships between N and k_0 for different values of α can be found in Table 5, which are also shown in Fig. 3. It can be observed that, with more time windows, less WR iterations are needed for a desired accuracy.

Example 5.2 We then consider the following sub-diffusion equation in two spatial dimensions,

$$\begin{cases} {}_0^C D_t^\alpha u(x, y, t) = 0.2 u_{xx}(x, y, t) + 0.3 u_{yy}(x, y, t) \\ \quad + u(x, y, t) + u^2(x, y, t), & (x, y) \in \Omega, \quad 0 < t < 2, \\ u(x, y, 0) = \sin(\pi x) \sin(\pi y), & (x, y) \in \bar{\Omega}, \\ u(x, y, t) = 0, & (x, y) \in \partial\Omega, \quad 0 \leq t \leq 2, \end{cases} \quad (44)$$

where $\Omega = [0, 1] \times [0, 1]$.

We still use a uniform space-time mesh with time step $\tau = \frac{1}{256}$ and spatial grid size $h_x = h_y = \frac{1}{64}$. N uniform time windows are employed, and each of them has $\frac{512}{N}$ time steps. The discrete windowing WR method for (44) can be written as

$$\begin{aligned} \mathbb{D}_\tau^\alpha [u_i^{(k+1)}]_{lq}^n + \sum_{j=0}^{i-1} \mathbb{D}_\tau^\alpha [u_j^{(k_0)}]_{lq}^n &= 0.2 \delta_x^2 [u_i^{(k+1)}]_{lq}^n + 0.3 \delta_y^2 [u_i^{(k+1)}]_{lq}^n \\ &+ [u_i^{(k)}]_{lq}^{n-1} + \left([u_i^{(k)}]_{lq}^{n-1} \right)^2, \end{aligned} \quad (45)$$

for $i = 0, 1, \dots, N - 1, l, q = 1, 2, \dots, 63, n = 1, 2, \dots, \frac{512}{N}, k = 0, 1, \dots, k_0 - 1$, and the reference solution can be obtained by the scheme

$$\begin{aligned} &\frac{\tau^{-\alpha}}{\Gamma(2 - \alpha)} \left(a_1 u_{lq}^n - \sum_{p=1}^{n-1} (a_{n-p} - a_{n-p+1}) u_{lq}^p - a_n u_{lq}^0 \right) \\ &= 0.2 \delta_x^2 u_{lq}^n + 0.3 \delta_y^2 u_{lq}^n + u_{lq}^{n-1} + \left(u_{lq}^{n-1} \right)^2, \end{aligned} \quad (46)$$

for $l, q = 1, 2, \dots, 63$, and $n = 1, 2, \dots, 512$. The iteration error of the windowing WR method is measured by

$$Error = \max_{1 \leq l, q \leq 63} \left| [u_{N-1}^{(k_0)}]_{lq}^{N_0} - u_{lq}^{NN_0} \right|.$$

Table 5 Relationships between N and k_0 for different values of α in Example 5.1

N	960	800	600	480	400	320	240	160	80	40	20	10	5	1
$k_0(\alpha = 0.2)$	–	–	–	–	11	13	16	19	24	30	35	42	48	70
$k_0(\alpha = 0.5)$	–	5	6	6	7	7	8	9	10	13	15	19	25	44
$k_0(\alpha = 0.8)$	4	4	4	5	5	5	5	6	7	8	9	12	15	28

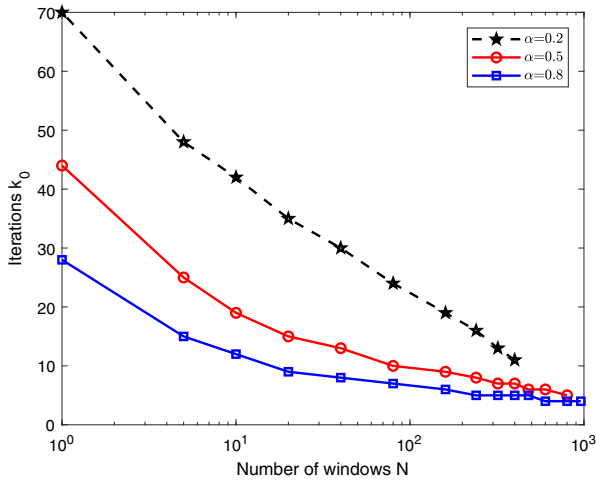


Fig. 3 Relationships between N and k_0 for different values of α in Example 5.1

Table 6 Errors of the windowing WR method (45) for Example 5.2 with $\alpha = 0.2$

k_0	$N = 128$	$N = 64$	$N = 32$	$N = 8$	$N = 1$
1	6.98e-04	1.71e-03	3.90e-03	2.39e-02	2.84e-01
2	1.08e-04	3.81e-04	1.00e-03	5.88e-03	1.57e-01
3	1.38e-05	1.04e-04	3.39e-04	2.28e-03	9.30e-02
4	0	2.81e-05	1.23e-04	9.98e-04	5.51e-02
7	0	1.55e-07	5.67e-06	1.01e-04	1.06e-02
15	0	0	3.00e-11	2.15e-07	1.03e-04
36	0	0	0	2.28e-15	2.84e-10
55	0	0	0	0	1.50e-15

Table 7 Errors of the windowing WR method (45) for Example 5.2 with $\alpha = 0.5$

k_0	$N = 128$	$N = 64$	$N = 32$	$N = 8$	$N = 1$
1	1.51e-03	3.73e-03	8.61e-03	4.64e-02	3.44e-01
2	7.82e-05	3.76e-04	1.26e-03	1.03e-02	1.99e-01
3	2.30e-06	3.71e-05	2.08e-04	2.96e-03	1.19e-01
4	0	2.99e-06	3.28e-05	9.05e-04	6.95e-02
7	0	1.46e-10	7.04e-08	2.39e-05	1.15e-02
13	0	0	4.83e-15	6.83e-09	1.98e-04
22	0	0	0	3.05e-15	1.81e-07
42	0	0	0	0	1.28e-15

Table 8 Errors of the windowing WR method (45) for Example 5.2 with $\alpha = 0.8$

k_0	$N = 128$	$N = 64$	$N = 32$	$N = 8$	$N = 1$
1	2.44e-03	5.84e-03	1.31e-02	6.31e-02	3.98e-01
2	3.08e-05	1.91e-04	8.17e-04	1.02e-02	2.33e-01
3	1.72e-07	4.74e-06	4.55e-05	1.73e-03	1.35e-01
4	0	8.40e-08	2.11e-06	2.74e-04	7.28e-02
7	0	2.60e-14	7.05e-11	6.18e-07	7.84e-03
10	0	0	6.10e-16	6.36e-10	5.41e-04
15	0	0	0	1.61e-15	2.64e-06
30	0	0	0	0	1.61e-15

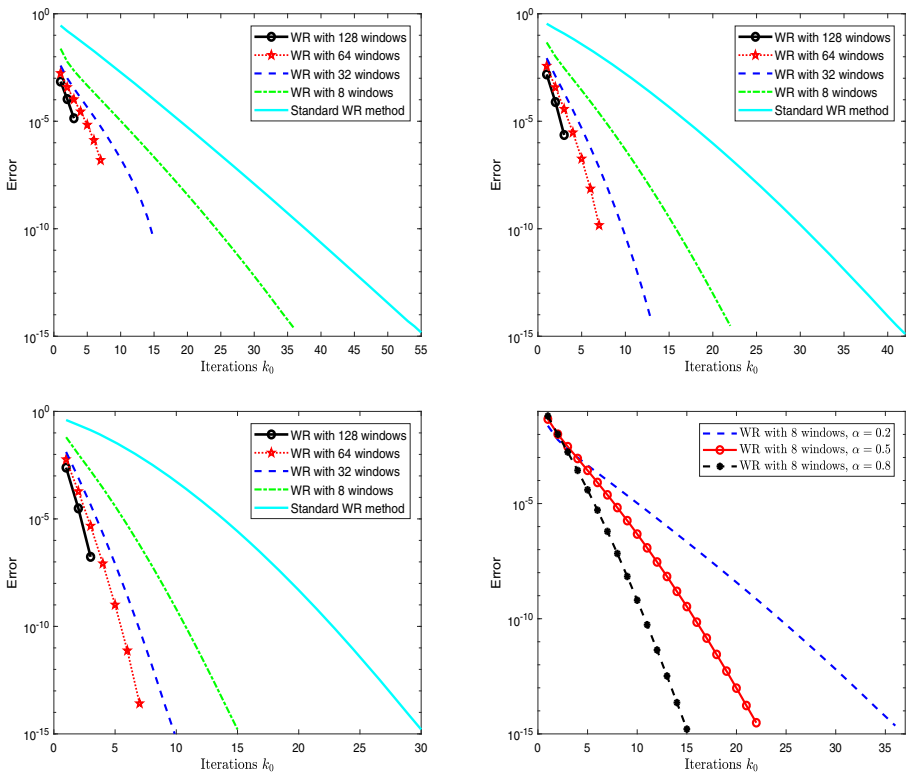


Fig. 4 Errors of windowing WR for Example 5.2 with $\alpha = 0.2$ (top left), $\alpha = 0.5$ (top right), and $\alpha = 0.8$ (bottom left), and errors of WR with 8 windows for various values of α (bottom right)

Table 9 Comparisons of the discrete WR method and its fast version for Example 5.2

	<i>N</i>	$\alpha = 0.2, k_0 = 7$		$\alpha = 0.5, k_0 = 6$		$\alpha = 0.8, k_0 = 4$	
		Error	Time (s)	Error	Time (s)	Error	Time (s)
WR	16	4.09e-05	355.77	1.61e-05	280.08	3.86e-05	194.73
	32	1.36e-05	353.67	2.10e-06	278.31	4.15e-06	194.92
	64	4.16e-06	355.38	2.25e-07	279.96	3.95e-07	194.50
	128	1.08e-06	355.94	1.84e-08	282.42	3.23e-08	194.72
	256	1.84e-07	356.06	8.84e-10	280.02	2.07e-09	191.91
Fast WR	16	4.09e-05	123.51	1.61e-05	104.57	3.86e-05	68.06
	32	1.36e-05	125.11	2.10e-06	104.96	4.15e-06	70.91
	64	4.16e-06	124.97	2.25e-07	104.72	3.95e-07	71.42
	128	1.08e-06	123.45	1.84e-08	104.20	3.23e-08	70.19
	256	1.84e-07	125.05	8.84e-10	104.53	2.07e-09	70.42

The iteration error for various values of k_0 and N are shown in Tables 6, 7, and 8, for $\alpha = 0.2, 0.5, 0.8$, respectively. We can also see that the windowing technique can accelerate the convergence of the WR method.

We plot the relationship between the iteration error and the iteration number k_0 with various time windows for $\alpha = 0.2, \alpha = 0.5$, and $\alpha = 0.8$, as well as the convergence behaviors of WR with 8 time windows for various values of α , in Fig. 4. The same observation can be found as that in Example 5.1.

We also use the fast evaluation formula of the WR method for Example 5.2. We fix $NN_0 = 4096$, and the observed errors and the consumed CPU times for the discrete windowing WR method and its fast evaluation version with different values of N and tolerance $\epsilon=1.0e-10$ are shown in Table 9. We can also see that the two versions with the same parameters have the same accuracy, while the fast windowing WR method needs much less running time.

Moreover, we fix the number of time steps $NN_0 = 2400$ on $[0, T]$, and denote k_0 as the minimum uniform number of WR iterations on each time window, such that the iteration error below $2.0e-10$ and the relationships between N and k_0 for different values of α are shown in Table 10 and Fig. 5, from which the same observation as that in Example 5.1 can be found.

Table 10 Relationships between N and k_0 for different values of α in Example 5.2

<i>N</i>	400	300	200	100	50	25	10	5	1
$k_0(\alpha = 0.2)$	-	-	11	15	18	21	25	27	37
$k_0(\alpha = 0.5)$	5	6	7	8	10	12	16	19	30
$k_0(\alpha = 0.8)$	4	5	5	6	7	8	10	13	23

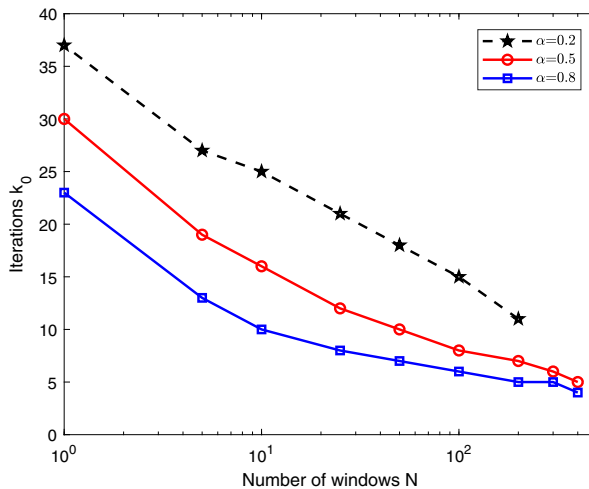


Fig. 5 Relationships between N and k_0 for different values of α in Example 5.2

6 Conclusions

A new WR method for semi-linear fractional sub-diffusion equations has been presented, and the error estimation has been given. Then, the windowing WR method and its discrete version have also been presented. We can see from the numerical experiments that the (windowing) WR method is convergent superlinearly, and the convergence rate will be slow with the decrease of the order of the time-fractional derivative. Furthermore, we have also combined the fast evaluation formula based on SOE approximation with the windowing WR method, and verified the efficiency of the fast windowing WR method by numerical tests.

Moreover, the WR method used in this paper is one of the simplest WR methods. Proposing some other effective WR methods to further improve the convergence rate will be considered in the future.

Acknowledgments The authors would like to thank the anonymous referees for their helpful comments and valuable suggestions to improve the paper significantly.

Funding This work was supported by the National Natural Science Foundation of China (Nos. 11401589, 11871393, and 11871400), the Fundamental Research Funds for the Central Universities (Nos. 18CX02049A and 17CX02066), and the International Science and Technology Cooperation Program of Shaanxi Research and Development Plan (2019KWZ-08).

References

1. Podlubny, I.: Fractional Differential Equations. New York: Academic Press (1999)
2. Sun, H., Zhang, Y., Baleanu, D., Chen, W., Chen, Y.: A new collection of real world applications of fractional calculus in science and engineering. *Commun Nonlinear Sci Numer Simulat* **64**, 213–231 (2018)

3. Metler, R., Jeon, J.H., Cherstvy, A.G., Barkai, E.: Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking. *Phys Chem Chem Phys* **16**, 24128–24164 (2014)
4. Svetukhin, V.V., Sibatov, R.T.: Kinetics of subdiffusive growth of new phase particles in supersaturated solid solutions. *J Exp Theor Phys* **120**, 678–686 (2015)
5. Bennett, K.M., Schmaidna, K.M., Bennett, R., Rowe, D.B., Lu, H., Hyde, J.S.: Characterization of continuously distributed cortical water diffusion rates with a stretched-exponential model. *Magn Reson Med* **50**, 727–734 (2003)
6. Alikhanov, A.A.: A new difference scheme for the time fractional diffusion equation. *J Comput Phys* **280**, 424–438 (2015)
7. Lin, Y., Xu, C.: Finite difference/spectral approximations for the time-fractional diffusion equation. *J Comput Phys* **225**, 1533–1552 (2007)
8. Sun, Z., Wu, X.: A fully discrete difference scheme for a diffusion-wave system. *Appl Numer Math* **56**, 193–209 (2006)
9. Lelarsmee, E., Ruehli, A.E., Sangiovanni-Vincentelli, A.L.: The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE T Comput Aid D* **1**, 131–145 (1982)
10. Vandewalle, S.: Parallel multigrid waveform relaxation for parabolic problems. Stuttgart: B. G. Teubner (1993)
11. Miekkala, U., Nevanlinna, O.: Convergence of dynamic iteration methods for initial value problems. *SIAM J Sci Stat Comput* **8**, 459–482 (1987)
12. Jiang, Y.L.: On time-domain simulation of lossless transmission lines with nonlinear terminations. *SIAM J Numer Anal* **42**, 1018–1031 (2004)
13. Jiang, Y.L., Wing, O.: On monotone waveform relaxation for systems of nonlinear differential-algebraic equations. *SIAM J Numer Anal* **38**, 170–185 (2000)
14. Jiang, Y.L., Chen, R.M.M.: Multisplitting waveform relaxation for systems of linear integral-differential-algebraic equations in circuit simulation. *J Circuit Syst Comp* **10**, 205–218 (2000)
15. Bartoszewski, Z., Kwapisz, M.: On error estimates for waveform relaxation methods for delay-differential equations. *SIAM J Numer Anal* **38**, 639–659 (2000)
16. Gander, M.J.: A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations. *Numer Linear Algebr* **6**, 125–145 (1999)
17. Gander, M.J.: Optimized schwarz methods. *SIAM J Numer Anal* **44**, 699–731 (2006)
18. Gander, M.J., Zhao, H.K.: Overlapping Schwarz waveform relaxation for the heat equation in n -dimensions. *BIT* **42**, 779–795 (2002)
19. Liu, J., Jiang, Y.L.: Waveform relaxation for reaction diffusion equations. *J Comput Appl Math* **235**, 5040–5055 (2011)
20. Jiang, Y.L., Ding, X.L.: On waveform relaxation methods for fractional differential equations with the Caputo derivatives. *J Comput Appl Math* **238**, 51–67 (2013)
21. Ding, X.L., Jiang, Y.L.: Waveform relaxation method for fractional functional differential equations. *Fract Calc Appl Anal* **16**, 573–594 (2013)
22. Ding, X.L., Jiang, Y.L.: Waveform relaxation method for fractional differential-algebraic equations. *Fract Calc Appl Anal* **17**, 585–604 (2014)
23. Gaspar, F.J., Rodrigo, C.: Multigrid waveform relaxation for the time-fractional heat equation. *SIAM J Sci Comput* **39**, A1201–A1224 (2017)
24. Wu, S.L.: Optimized overlapping Schwarz waveform relaxation for a class of time-fractional diffusion problems. *J Sci Comput* **72**, 842–862 (2017)
25. Wu, S.L., Huang, C.M.: Asymptotic results of Schwarz waveform relaxation algorithm for time-fractional cable equations. *Commun Comput Phys* **25**, 390–415 (2019)
26. Kilbas, A.A., Srivastava, H.M., Trujillo, J.J.: Theory and applications of fractional differential equations. North-holland mathematics studies, 204. Amsterdam: Elsevier Science B. V. (2006)
27. Schneider, W.R.: Completely monotone generalized Mittag-Leffler functions. *Expo Math* **14**, 3–16 (1996)
28. Jiang, Y.L.: On Windowing waveform relaxation of initial value problems. *Acta Math Appl Sin-E* **22**, 543–556 (2006)
29. Gao, G., Sun, Z., Zhang, H.: A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications. *J Comput Phys* **259**, 33–50 (2014)
30. Jiang, S., Zhang, J., Zhang, Q., Zhang, Z.: Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations. *Commun Comput Phys* **21**, 650–678 (2017)

31. Liao, H., Yan, Y., Zhang, J.: Unconditional convergence of a fast two-level linearized algorithm for semilinear subdiffusion equations. *J Sci Comput* **80**, 1–25 (2019)

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