Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

A parallel algorithm for space-time-fractional partial differential equations

E. Lorin[1,2*]

[*]Correspondence:
elorin@math.carleton.ca
[1]Centre de Recherches Mathématiques, Université de Montréal, Montréal, Canada
[2]School of Mathematics and Statistics, Carleton University, Ottawa, Canada

Abstract

This paper is dedicated to the derivation of a simple parallel in space and time algorithm for space and time fractional evolution partial differential equations. We report the stability, the order of the method and provide some illustrating numerical experiments.

Keywords: Fractional differential equations; Parallel-in-time algorithm; Pseudospectral method; Approximate solver

1 Introduction

This paper is devoted to the derivation of a parallel algorithm for solving fractional *in space and time* fractional partial differential equations. We will first focus on the parallelization in-time using a *parareal algorithm*, which is the most standard parallel-in-time algorithm for solving differential equations [20, 29]. In this goal, we focus in the first part of the paper on the derivation and analysis of a parallel-in-time algorithm for fractional ordinary differential equations (FODEs) [40]. More specifically, a parareal-Gorenflo's scheme [40] is derived and analyzed for FODEs. We notice that the analysis of convergence of the parareal method for solving ODEs remains partially valid for fractional equations [22, 32]. Then, we provide a parallel *in space and time* algorithm for fractional in space and time partial differential equations. Basically, the spatial approximation is provided thanks to (i) a Fourier-based pseudospectral method, which will benefit from the scalability of standard parallel FFT algorithms [18] and (ii) an efficient combination with the parallel-in-time algorithm.

A key element of the proposed overall algorithm is the parareal method. Let us recall its principle for solving differential equations. It consists in

1. first solving *sequentially* the differential equation over $[0;T]$ using a large time-step $\Delta T = T/N$ for some $N \in \mathbb{N}^*$, that is, on a coarse grid in time;
2. solving (using the same or more accurate scheme as the one used on the coarse mesh) *in parallel on say q processors* (typically such that $N \in q\mathbb{N}^*$ for an efficient workload) with a time-step $\Delta t = \Delta T/R$ for some $R \gg 1$, the equation on q subdomains $\{[T_{qi}; T_{q(i+1)}]\}_{0\leq i\leq R-1}$, where $T_n = n\Delta T$ for $0 \leq n \leq N$, $T_N = T$, and with

initial conditions (at $\{T_n\}_{0\leq n\leq N-1}$) on each subdomain, given by the solution initially computed on the coarse grid (Step 1).

The parareal method allows for an accurate parallel computation of ODEs, and we refer to [16, 21, 29] for details about this celebrated method for solving ordinary/partial differential equations. It is in particular successfully combined with traditional domain decomposition methods in space. Indeed it allows us to use a very large number of processors going beyond the usual limits of efficiency of domain decomposition methods with a too large number of spatial subdomains. In this paper, we are interested in the extension of the parareal method to FODEs, which have become very popular over this past decade. The recent progress in fractional differential equation solvers [6, 7, 25, 27, 30, 33, 39] is largely motivated by the development of fractional differential models in physics, mechanics, epidemiology, and applied probability allowing to take into account nonlocal in space or time effects [9–11, 15, 26, 30, 35, 37]. The main purpose of this paper is then to study efficient parallel algorithms for fractional ordinary and partial differential equations. In particular, we propose an original algorithm combining a parallelization in space and time. Very few works actually exist on parallel-in-time methods for FODEs; however, let us cite [38] where a parareal method along with collocation and Fourier-based FODE solvers is developed. At this stage, we do not consider realist models from the literature, but focus on toy-scalar linear equations, for which it is possible to provide a relatively precise analysis and to exhibit the strong convergence and efficiency properties. In this paper, we first propose a parareal version of standard Gorenflo's scheme for approximating FODE [40]. In this goal, we consider

$$D_t^\alpha y(t) = f\bigl(t, y(t)\bigr), \qquad y(0) = y_0, \quad t \in [0;T] \tag{1}$$

for $0 < \alpha < 1$, $T > 0$, and where f is a non-highly oscillatory Lipschitz continuous function. Lipschitz continuity allows for existence of a unique solution, while the non-highly oscillatory condition will allow for an efficient and accurate use standard quadratures. In particular the existence of unique solutions can be proved in some weighted subsets of C^α, see [19] for details. The analysis will be presented for the parareal-Gorenflo scheme approximating a linear FODE:

$$D_t^\alpha y(t) = -\lambda y(t) + g(t), \qquad y(0) = y_0, \tag{2}$$

where $\lambda > 0$, $0 < \alpha < 1$, and $g \in C^0([0;T])$. The operator $D_t^\alpha = {}^{\mathrm{C}}D_t^\alpha$ is here defined as a Caputo's derivative [40], that is,

$$^{\mathrm{C}}D_t^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{-\alpha} D_\tau y(\tau)\, d\tau,$$

where the special gamma function Γ, for $\mathrm{Re}(z) > 0$, is defined as

$$\Gamma(z) := \int_0^\infty x^{z-1} e^x \, dx,$$

and $\Gamma(p) = (p-1)!$ for $p \in \mathbb{N}^*$. We also recall the definition of Caputo's derivative ${}^{\mathrm{C}}I_t^\alpha y(t)$

$$^{\mathrm{C}}I_t^\alpha y(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} y(\tau)\, d\tau. \tag{3}$$

Hence, (1) and (say) for $y(0) = 0$, and y solution to (2) we have

$$y(t) = -\frac{\lambda}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1}D_\tau^\alpha y(\tau)\,d\tau + \frac{1}{\Gamma(\alpha)}\int_0^t (t-\tau)^{\alpha-1}g(\tau)\,d\tau.$$

The well-posedness (existence and uniqueness) of this equation is analyzed in [14] for Lipschitz functions f. Let us also refer to [28, 31, 34] for other types of fractional differential equations. The main difficulty in parallelizing a FODE comes from the fact that the fractional derivative is a nonlocal integro-differential operator. As a consequence, it is no more possible to *directly and efficiently* compute in parallel the differential equation on fine grids, as usually proposed in the parareal method. In this paper, we propose a natural strategy which consists in computing in parallel the fractional integrals by decomposing them into a local (containing the "singularity" and computed with a fine resolution) and a history part (computed with a coarse resolution).

In the second part of the paper, we are then interested in a parallel algorithm for *space-time* fractional differential equations of the form

$$D_t^\alpha u(t,x) = -\lambda(x)D_x^\beta u(t,x), \qquad u(0,x) = u_0 \in L^2(\mathbb{R}),$$

on $[0;T] \times \mathbb{R}$, with λ in the set of continuous and bounded real function $C_b^0(\mathbb{R})$, $0 < \alpha < 1$ and $\beta > 0$. We recall that, denoting $m = \lceil\beta\rceil$, Caputo's derivative $D_x^\beta = {}^{\mathrm{C}}D_x^\beta$ in this case is given by

$${}^{\mathrm{C}}D_x^\beta u(t,x) = \frac{1}{\Gamma(m-\beta)}\int_{-\infty}^x (x-y)^{m-1-\beta}\partial_x^m u(t,y)\,dy.$$

Notice that several alternative definitions of fractional derivatives exist [12] such as the Riemann–Liouville (RL) fractional derivative of order β over the interval $]-\infty;x]$, which is defined by

$${}^{\mathrm{RL}}D_x^\beta u(t,x) = \frac{1}{\Gamma(m-\beta)}\frac{\partial^m}{\partial x^m}\int_{-\infty}^x \frac{u(t,y)}{(x-y)^{\beta-m+1}}\,dy, \tag{4}$$

while the right RL fractional derivative is given by

$${}^{\mathrm{RL}}{}_xD^\beta u(t,x) = \frac{(-1)^m}{\Gamma(m-\beta)}\frac{\partial^m}{\partial x^m}\int_x^{+\infty} \frac{u(t,y)}{(y-x)^{\beta-m+1}}\,dy.$$

Similarly, we introduce the left fractional Riemann–Liouville integral operator of order β as follows:

$${}^{\mathrm{RL}}I_x^\beta u(x) = \frac{1}{\Gamma(\beta)}\int_{-\infty}^x \frac{u(t,y)}{(x-y)^{1-\beta}}\,dy. \tag{5}$$

Through these notations, we have the relation

$${}^{\mathrm{RL}}D_x^\beta u(t,x) = \frac{\partial^m}{\partial x^m}{}^{\mathrm{RL}}I_x^{m-\beta}u(t,x). \tag{6}$$

Let us also recall that the two derivatives are linked by the simple relation

$$^{\mathrm{RL}}D_x^{\alpha}u(x) = {}^{\mathrm{C}}D_x^{\alpha}u(x) + \frac{1}{\Gamma(1-\alpha)} \times \frac{u(a)}{(x-a)^{\alpha}}$$

for some real a. These definitions can naturally be used to define spatial fractional derivatives. However, to define the spatial fractional derivative, we rather use the Fourier-based definition (Riesz's derivative denoted by $^{\mathrm{R}}D_x^{\beta}$) which will be more convenient and which could still be implemented on a bounded spatial domain. Denoting by ξ the co-variable to x in Fourier space, and by $\mathcal{F}$ the corresponding Fourier transform, we define

$$^{\mathrm{R}}D_x^{\beta}u(t,x) = \mathcal{F}^{-1}\bigl((\mathtt{i}\xi)^{\beta}\mathcal{F}\bigl(u(t,\xi)\bigr)\bigr).$$

Based on this definition, we will use a pseudospectral method based on discrete Fourier transform for spatial approximation. We again refer to [6, 7, 13, 30, 36] for some discussions about fractional operators and fractional differential equations. Finally, we discuss the combination of the parallel in time and space algorithms along with some numerical experiments.

This paper is organized as follows. Section 2 is dedicated to the derivation of the parareal method based on Gorenflo's scheme for solving FODE. Using in particular the parareal method and standard parallel FFTs, we then derive in Sect. 3 a parallel algorithm for space-time fractional differential equations. We finally conclude in Sect. 4.

2 Parareal-Gorenflo algorithm for fractional differential equations

We denote by ΔT a coarse grid time-step and by Δt a fine grid time-step such that $\Delta T = R\Delta t$. The coefficient $R \in \mathbb{N}\backslash\{0\}$ corresponds to the number of subiterations in-time for each large time-step ΔT. We denote $T_n = n\Delta T$ and $t_{n;i} := n\Delta T + i\delta t$ for $n = 0,\ldots,N-1$ and for $i = 0,\ldots,R$ such that $t_{0,0} = 0$ and $t_{N-1;R} = T$. For a total of N coarse grid time iterations, we proceed to

- sequential computations on $[0;T_N]$;
- parallel computations on each $[T_n;T_{n+1}]$ requiring computations from $[0;T_n]$ and on the grids $[T_n;T_{n+1}] = \bigcup_{i=1}^{R}[t_{n;i-1};t_{n;i}]$, with $n = 0,\ldots,N-1$.

We now derive the algorithm and provide some analytical details.

2.1 Gorenflo's scheme

We approximate (2) with Gorenflo's scheme [17], which reads (on the coarse grid, using the notation $\{y_{\Delta T}^n\}_n$ approximating $\{y(T_n)\}_n$) as follows:

$$y_{\Delta T}^{n+1} = -\lambda\Delta T^{\alpha}y_{\Delta T}^{n} - \sum_{i=1}^{n+1} w_i^{(\alpha)}y_{\Delta T}^{n+1-i} + \Delta T^{\alpha}g(T_{n+1}), \tag{7}$$

where $y_{\Delta T}^n$ approximates $y(T_n)$, and the weights read

$$w_i^{(\alpha)} = \sum_{l=1}^{i}\frac{\Gamma(l-\alpha)}{\Gamma(-\alpha)\Gamma(l+1)}. \tag{8}$$

Gorenflo's scheme, which extends Grünwald–Letnikov's idea of a finite difference approximation of the fractional integral. In theory any other FODE solver could be combined

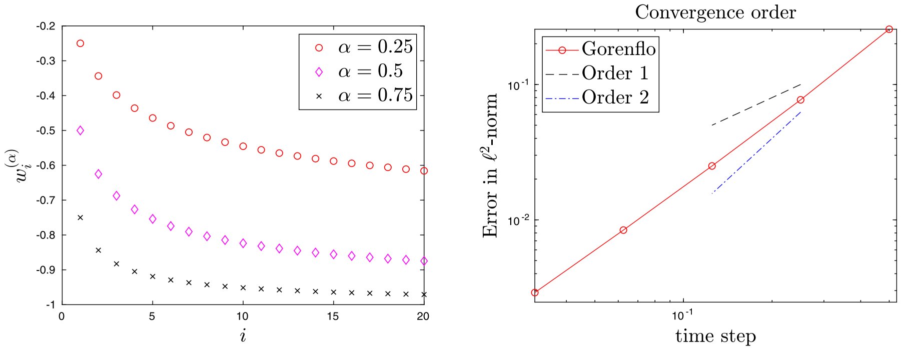


Figure 1 Numerical experiment 1. (Left) Coefficients $\{w_i^{(\alpha)}\}_{1\leq i\leq 20}$ for $\alpha = 0.25, 0.5, 0.75$. (Right) Convergence graph for Gorenflo's scheme

with the parareal method such as those presented in [23] or the one used in [38]. The coefficients $w_i^{(\alpha)}$ for $\alpha = 0.25, 0.5, 0.75$ are reported for $1 \leq i \leq 20$ in Fig. 1. The choice of this method is motivated by its simplicity and the fact that it is possible to easily increase its order of convergence. Other methods, such as the ones used in [23], can naturally be explored.

Numerical experiment 1. We present a simple experiment from [17]: for $0 < \alpha < 1$,

$$D_t^\alpha y(t) = -y(t) + t^2 + \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)}, \qquad y(0) = 0, \quad t \in [0,1],$$

for which an explicit solution $y_{\text{exact}}(t) = t^2$ exists. We report the convergence graph in Fig. 1 (Right), that is, the ℓ^2-norm error as a function of the time-steps; we numerically estimate the order of convergence of the method which on this specific example ($\alpha = 0.5$) leads to $p \approx 1.5$. We refer to [25] on the derivation and convergence analysis of this scheme. In a compact form (7) reads

$$\mathbf{Y}_{\Delta T}^{n+1} = \mathcal{C}_{\Delta T}\big(\mathbf{Y}_{\Delta T}^{n}\big),$$

where $\mathbf{Y}_{\Delta T}^{n} = [y_{\Delta T}^{0}, \ldots, y_{\Delta T}^{n}] \in \mathbb{R}^{n+1}$, where we use the convenient notation from [38]. The algebraic operator $\mathcal{C}_{\Delta T}$ (from $\mathbb{R}^{n+1}$ to $\mathbb{R}^{n+2}$) basically constructs the solution at time t_{n+1} from the solutions at previous times from (7).

We first discuss the absolute stability of Gorenflo's scheme on a (coarse) grid:

$$y_{\Delta T}^{n+1} = -\lambda\Delta T^\alpha y_{\Delta T}^{n} - \sum_{i=1}^{n+1} w_i^{(\alpha)} y_{\Delta T}^{n+1-i} + \Delta T^\alpha g(T_{n+1}),$$

which can also be rewritten

$$\mathbf{Y}_{\Delta T}^{n+1} = \mathcal{M}_{\Delta T}\mathbf{Y}_{\Delta T}^{n} + \mathcal{G}_{\Delta T}^{n+1},$$

where $\mathcal{G}_{\Delta T}^{n+1} = [g(T_1),\ldots,g(T_{n+1})]$ and the matrix $\mathcal{M}_{\Delta T} = \{m_{\Delta T;ij}\}_{i,j}$ is defined as

$$\begin{cases} m_{\Delta T;ij} = 0, & i > j, \\ m_{\Delta T;ij} = -w_1^{(\alpha)} - \lambda\Delta t^{\alpha}, & i = j-1, \\ m_{\Delta T;ij} = -w_i^{(\alpha)}, & i > j+1, \end{cases}$$

where $\{w_i^{(\alpha)}\}_i$ is defined in (8). We are interested in the conditional stability of Gorenflo's scheme. We then have the following.

Proposition 2.1 *For* $1 > \alpha > 0$, $\lambda > 0$, *Gorenflo's scheme is conditionally absolutely stable in the sense that*

$$\lim_{n\to\infty} \bigl|y_{\Delta T}^n\bigr| = 0, \quad \textit{for any } \Delta T \leq (\alpha/\lambda)^{1/\alpha}.$$

Proof We aim to study the limit at infinity of the numerical solution to

$$D_t^{\alpha} y(t) = -\lambda y(t), \qquad y(0) = 1, \tag{9}$$

when t goes to infinity. As Gorenflo's method requires a priori $y(0) = 0$, we reformulate (9) as

$$D_t^{\alpha} y(t) = -\lambda y(t) - \lambda, \qquad y(0) = 0. \tag{10}$$

The solution to (9) is hence deduced from (10) by adding 1: $y \leftarrow y + 1$. Then Gorenflo's scheme with $n \geq 0$

$$y_{\Delta T}^{n+1} = -\lambda\Delta T^{\alpha} y_{\Delta T}^n - \sum_{i=1}^{n+1} w_i^{(\alpha)} y_{\Delta T}^{n+1-i} - \lambda\Delta T^{\alpha}.$$

First we notice that $w_1^{(\alpha)} = -\alpha \in (-1,0)$ and more generally $\alpha < -w_i^{(\alpha)}$. As a consequence, we can rewrite the scheme as

$$y_{\Delta T}^{n+1} = \bigl(\alpha - \lambda\Delta T^{\alpha}\bigr) y_{\Delta T}^n - \sum_{i=2}^{n+1} w_i^{(\alpha)} y_{\Delta T}^{n+1-i} + \lambda\Delta T^{\alpha}. \tag{11}$$

We prove by induction that $\{y_{\Delta T}^n\}_n$ is decreasing and bounded from below by -1:

1. We assume that $y_{\Delta T}^n \leq y_{\Delta T}^{n-1} \leq \cdots \leq y^0 = 0$. We have

$$y_{\Delta T}^n = \bigl(\alpha - \lambda\Delta T^{\alpha}\bigr) y_{\Delta T}^{n-1} - \sum_{i=2}^{n} w_i^{(\alpha)} y_{\Delta T}^{n-i} - \lambda\Delta T^{\alpha}. \tag{12}$$

Thus

$$y_{\Delta T}^{n+1} - y_{\Delta T}^n = \bigl(\alpha - \lambda\Delta T^{\alpha}\bigr)\bigl(y_{\Delta T}^n - y_{\Delta T}^{n-1}\bigr) - \sum_{i=2}^{n+1} w_i^{(\alpha)} y_{\Delta T}^{n-i+1} + \sum_{i=2}^{n} w_i^{(\alpha)} y_{\Delta T}^{n-i}$$

$$= \big(\alpha - \lambda\Delta T^{\alpha}\big)\big(y_{\Delta T}^{n} - y_{\Delta T}^{n-1}\big) - \sum_{i=2}^{n} w_i^{(\alpha)}\big(y_{\Delta T}^{n-i+1} - y_{\Delta T}^{n-i}\big) - w_{n+1}^{(\alpha)} y^0$$
$$= \big(\alpha - \lambda\Delta T^{\alpha}\big)\big(y_{\Delta T}^{n} - y_{\Delta T}^{n-1}\big) - \sum_{i=2}^{n} w_i^{(\alpha)}\big(y_{\Delta T}^{n-i+1} - y_{\Delta T}^{n-i}\big).$$

Hence, as (i) the coefficients $w_i^{(\alpha)}$ are negative, (ii) less than 1 in absolute value, and (iii) $y^0 = 0$, we get

$$y_{\Delta T}^{n+1} - y_{\Delta T}^{n} = \big(\alpha - \lambda\Delta T^{\alpha}\big)\big(y_{\Delta T}^{n} - y_{\Delta T}^{n-1}\big) + \sum_{i=2}^{n}\big|w_i^{(\alpha)}\big|\big(y_{\Delta T}^{n-i+1} - y_{\Delta T}^{n-i}\big) \leq 0,$$

and conclude that at least for $\Delta T \leq (\alpha/\lambda)^{1/\alpha}$ we get $y_{\Delta T}^{n+1} \leq y_{\Delta T}^{n}$. Hence, the sequence $\{y_{\Delta T}^{n}\}_n$ is increasing. Notice that applying a standard discrete Gronwall inequality would lead to a similar conclusion and can be used for proving the convergence of the method.

$$y_{\Delta T}^{n+1} - y_{\Delta T}^{n} \leq \big(y_{\Delta T}^{1} - y_{\Delta T}^{0}\big)\exp\Bigg(\big(\alpha - \lambda\Delta T^{\alpha}\big)| + \sum_{i=2}^{n}\big|w_i^{(\alpha)}\big|\Bigg).$$

2. We still assume by induction that $y_{\Delta T}^{n} \geq -1$ for all $n \geq 0$. As

$$y_{\Delta T}^{n+1} = \big(\alpha - \lambda\Delta T^{\alpha}\big)y_{\Delta T}^{n} - \sum_{i=1}^{n} w_{i+1}^{(\alpha)} y_{\Delta T}^{n-i} - \lambda\Delta T^{\alpha} \tag{13}$$

and

$$y_{\Delta T}^{n} = \big(\alpha - \lambda\Delta T^{\alpha}\big)y_{\Delta T}^{n-1} - \sum_{i=2}^{n} w_{i}^{(\alpha)} y_{\Delta T}^{n-i} - \lambda\Delta T^{\alpha}.$$

Hence, for $\alpha - \lambda\Delta T^{\alpha} \geq 0$,

$$y_{\Delta T}^{n} \leq \alpha y_{\Delta T}^{n-1} - \sum_{i=2}^{n} w_{i}^{(\alpha)} y_{\Delta T}^{n-i}$$

and $-\sum_{i=2}^{n} w_i^{(\alpha)} y_{\Delta T}^{n-i} \geq (\alpha - 1)$. Thus, yet under the assumption $\alpha - \lambda\Delta T^{\alpha} \geq 0$ and using $y_{\Delta T}^{n} \geq -1$, we have

$$\begin{aligned} y_{\Delta T}^{n+1} &= \big(\alpha - \lambda\Delta T^{\alpha}\big)y_{\Delta T}^{n} - \sum_{i=1}^{n} w_{i+1}^{(\alpha)} y_{\Delta T}^{n-i} - \lambda\Delta T^{\alpha} \\ &\geq \big(\alpha - \lambda\Delta T^{\alpha}\big)y_{\Delta T}^{n} + (\alpha - 1) - \lambda\Delta T^{\alpha} \\ &\geq \big(\alpha - \lambda\Delta T^{\alpha}\big)\big(1 + y_{\Delta T}^{n}\big) - 1 \\ &\geq -1. \end{aligned}$$

Thus $\{y^n_{\Delta T}\}_n$ is decreasing and bounded then convergent to say y^*. Taking the limit in n, we get

$$-\lim_{n\to\infty}\sum_{i=2}^{n} w_i^{(\alpha)} y^{n-i}_{\Delta T} = \bigl(1-\alpha-\lambda\Delta T^{\alpha}\bigr)y^*. \tag{14}$$

Taking for instance $\Delta T = (\alpha/\lambda)^{1/\alpha}$ in Gorenflo's scheme gives

$$y^n_{\Delta T} = -\sum_{i=2}^{n} w_i^{(\alpha)} y^{n-i}_{\Delta T} - \alpha,$$

and using (14), we get

$$y^* = (1-\alpha)y^* - \alpha.$$

We then deduce that the solution is convergent to $y^* = -1$. □

2.2 Parareal algorithm for fractional differential equations

We now explicitly derive a parallel algorithm for FODE. Each interval $[T_n; T_{n+1}]$ for $0 \leq n \leq N-1$ is then decomposed in R time-subdomains. For $0 \leq \ell \leq NR-1$ (corresponding to time $(n-1)\Delta T + i\Delta t$ with $0 \leq i \leq R$), Gorenflo's scheme on the fine grid reads

$$y^{\ell+1}_{\Delta t} = -\lambda\Delta t^{\alpha} y^{\ell}_{\Delta t} - \sum_{j=1}^{\ell+1} w_j^{(\alpha)} y^{\ell+1-j}_{\Delta t} + \Delta t^{\alpha} g(t_{n;i}).$$

It would be naturally highly inefficient to solve (even in parallel) the FODE on a "full" fine grid. In order to derive the parareal algorithm, we need to justify that the solution to (2) can be approximated thanks to a sum of "local" FODEs.

If we decompose $[0;T] = \bigcup_{n=0}^{N-1}[T_n;T_{n+1}]$, and for $1 \leq n \leq N-1$, we denote by y_n the solution to

$$\frac{1}{\Gamma(1-\alpha)}\int_{T_n}^{t}(t-\tau)^{-\alpha}D_{\tau}y_n(\tau)\,d\tau = -\lambda y_n(t) \quad \text{for } t\in[T_n;T_{n+1}], \qquad y_n(T_n) = y_{n-1}(T_n).$$

We naturally have the following.

Lemma 1 *If* $\alpha \in (0,1)$, *for any* $1 \leq n \leq N-1$, *we a priori have* $y_{|[T_n;T_{n+1}]} \neq y_n$, *where* y *is a solution to* (2).

This is a simple consequence on the fact that fractional derivatives and integrals are nonlocal operators. The integral decomposition must be carefully designed.

Proof Recall that y_1 and y_n are the respective solutions to

$$\frac{1}{\Gamma(1-\alpha)}\int_{0}^{t}(t-\tau)^{-\alpha}D_{\tau}y_1(\tau)\,d\tau = -\lambda y_1(t) \quad \text{for } t\in[0;T_1], \qquad y_1(0) = y_0,$$

and for $n > 1$,

$$\frac{1}{\Gamma(1-\alpha)}\int_{T_n}^{t}(t-\tau)^{-\alpha}D_\tau y_n(\tau)\,d\tau = -\lambda y_n(t) \quad \text{for } t \in [T_n;T_{n+1}], \qquad y_n(T_n) = y_{n-1}(T_n).$$

Trivially we have $y_{|[0;T_1]} = y_1$. Next, on any interval $[T_n;T_{n+1}]$, we have $y_n(T_n) = y_{n-1}(T_n)$. *If* $y_{|[T_n;T_{n+1}]} = y_n$, then for $t \in [T_n;T_{n+1}]$

$$\begin{aligned}-\lambda\bigl(y_{|[T_n;T_{n+1}]}(t) - y_n(t)\bigr) = {}& \frac{1}{\Gamma(1-\alpha)}\biggl[\int_0^{T_n}(t-\tau)^{-\alpha}D_\tau y(\tau)\,d\tau \\ &+ \int_{T_n}^{T_{n+1}}(t-\tau)^{-\alpha}\bigl(D_\tau y(\tau) - D_\tau y_n(\tau)\bigr)\,d\tau\biggr]_{|[T_n;T_{n+1}]}\end{aligned}$$

is equivalent to

$$\frac{1}{\Gamma(1-\alpha)}\int_0^{T_n}(t-\tau)^{-\alpha}D_\tau y(\tau)\,d\tau_{|[T_n;T_{n+1}]} = 0,$$

which is in general not true. □

The proposition shows that in order to derive a consistent parareal method, it is necessary on each $[T_n;T_{n+1}]$ to include a nonlocal contribution from $[0;T_n]$, corresponding to

$$\frac{1}{\Gamma(1-\alpha)}\int_0^{T_n}(t-\tau)^{-\alpha}D_\tau y(\tau)\,d\tau_{|[T_n;T_{n+1}]}.$$

Let us now detail the corresponding parareal method. Recall that different schemes can be used on the coarse and fine levels. Generally speaking, computations on fine grids will be only performed in parallel, while the coarse grid ones will be performed either in parallel or sequentially. We now detail the algorithms.

Algorithm The overall parareal algorithm reads using *usual* compact notations [29, 38]

$$\mathbf{Y}_{\Delta T}^{n+1;k} = \mathcal{C}_{\Delta T}\bigl(\mathbf{Y}_{\Delta T}^{n;k}\bigr) - \mathcal{C}_{\Delta T}\bigl(\mathbf{Y}_{\Delta T}^{n;k-1}\bigr) + \mathcal{F}_{\Delta T}\bigl(\mathbf{Y}_{\Delta t}^{n;k-1}\bigr), \tag{15}$$

where k denotes the domain decomposition in-time index/iteration, and where $\mathbf{Y}_{\Delta T}^{n;k} \in \mathbb{R}^{n+1}$ represents $[y_{\Delta T}^{0;k},\dots,y_{\Delta T}^{n;k}]$, that is, the approximate solution on the *coarse grid (but eventually computed by combining fine and coarse grid computations)* at iteration k using the scheme (7). However, the computation of this quantity differs from the parareal method for ODEs due to the nonlocality of fractional operators. We proceed as follows:

- for $k \geq 1$, we compute *in parallel*

$$\mathcal{F}_{\Delta T}\bigl(\mathbf{Y}_{\Delta T}^{n;k}\bigr) = \bigl[\mathcal{F}_{\Delta T}\bigl(y_{\Delta T}^{0;k}\bigr),\dots,\mathcal{F}_{\Delta T}\bigl(y_{\Delta T}^{n;k}\bigr)\bigr],$$

  and where $\mathcal{F}_{\Delta T}(y_{\Delta T}^{n;k})$ denotes the approximate solution to

$$D_t^\alpha y(t) = -\lambda y(t) + g(t), \quad t \in [T_n;T_{n+1}],$$

which is equivalent, thanks to (3), to

$$y(t) = y(T_n) - \frac{\lambda}{\Gamma(\alpha)} \int_{T_n}^{t} (t-\tau)^{\alpha-1} y(\tau)\,d\tau + \frac{1}{\Gamma(\alpha)} \int_{T_n}^{t} (t-\tau)^{\alpha-1} g(\tau)\,d\tau,$$

with initial data given at T_n, and computed *partially* on a fine grid $\{t_{n;0}, \dots, t_{n;R}\} \in [T_n; T_{n+1}]^{R+1}$. Unlike the differential case, the nonlocality in time requires a special care. Indeed, on any interval $[T_n; T_{n+1}]$, we still need to include a contribution of former times ($t < T_n$). In other words, this leads to a potentially very computationally complex method, even if it is solved in parallel. More specifically, for $t \in [T_n; T_{n+1}]$, we rewrite

$$\begin{aligned} I_t^{\alpha} y(t) &= \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} y(\tau)\,d\tau \\ &= \frac{1}{\Gamma(\alpha)} \int_0^{T_n} (t-\tau)^{\alpha-1} y(\tau)\,d\tau + \frac{1}{\Gamma(\alpha)} \int_{T_n}^{t} (t-\tau)^{\alpha-1} y(\tau)\,d\tau. \end{aligned}$$

In particular, for any $T_n \leq t \leq T_{n+1}$, we also have

$$\begin{aligned} y(t) = y(0) &- \frac{\lambda}{\Gamma(\alpha)} \left[\sum_{j=0}^{n-1} \int_{T_j}^{T_{j+1}} (t-\tau)^{\alpha-1} y(\tau)\,d\tau + \int_{T_n}^{t} (t-\tau)^{\alpha-1} y(\tau)\,d\tau \right] \\ &+ \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} g(\tau)\,d\tau. \end{aligned}$$

We propose to approximate the first integral (history part) using a coarse grid approximation at iteration k, and the second one on a fine grid. More specifically, we approximate on the *coarse* grid

$$H_n(t) = -\frac{\lambda}{\Gamma(\alpha)} \int_0^{T_n} (t-\tau)^{\alpha-1} y(\tau)\,d\tau.$$

Notice that we do not have to deal with singularities on this integral, any (higher order) quadrature rule can be utilized. We approximate this term at $t_{n;i} = T_n + i\Delta t$, say by (using a rectangle rule for simplicity)

$$\mathrm{H}_{n;i} = -\Delta T^{\alpha} \lambda \sum_{j=1}^{n} \frac{(n-j+i/R)^{\alpha} - (n-j-1+i/R)^{\alpha}}{\Gamma(\alpha+1)} y_{\Delta T}^{j}, \tag{16}$$

while the local part

$$L_n(t) = \frac{1}{\Gamma(\alpha)} \int_{T_n}^{t} (t-\tau)^{\alpha-1} y(\tau)\,d\tau$$

is approximated using the *fine grid* on $[T_n; T_{n+1}]$. The local part can simply be rewritten as

$$L_n(t) = \frac{1}{\Gamma(\alpha)} \int_0^{t-T_n} (t - T_n - \tau)^{\alpha-1} y(\tau + T_n)\,d\tau,$$

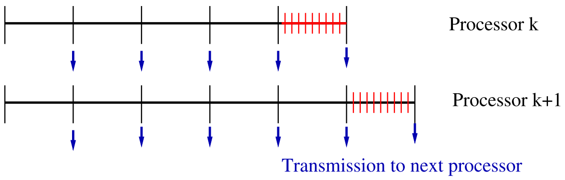


Figure 2 Parareal algorithm with fine/coarse grid in $\mathcal{F}_{\Delta T}$-step

with $y(T_n)$ given. For instance,

$$L_n(T_{n+1}) = \frac{1}{\Gamma(\alpha)} \int_0^{\Delta T} (\Delta T - \tau)^{\alpha-1} y(\tau + T_n)\, d\tau.$$

Over $[T_n; T_{n+1}]$, that is, $Rn \leq i + Rn \leq (R+1)n$ with $0 \leq n \leq N-1$ and $0 \leq i \leq R$, rather than computing on the fine grid a complete Gorenflo scheme

$$y_{\Delta t}^{nR+i+1;k} = -\lambda \Delta t^\alpha y_{\Delta t}^{nR+i;k} - \sum_{j=1}^{nR+i+1} w_j^{(\alpha)} y_{\Delta t}^{nR+i+1-j;k} + \Delta t^\alpha g(t_{n;i}),$$

which would be, as already mentioned above, highly inefficient from the computational point of view. We compute in parallel (indexed here by n)

$$y_{\Delta t}^{nR+i+1;k} = -\lambda \Delta t^\alpha y_{\Delta t}^{nR+i;k} - \sum_{j=1}^{i+1} w_j^{(\alpha)} y_{\Delta t}^{nR+i+1-j;k} + \mathrm{H}_{n;i} + \Delta t^\alpha g(t_{n;i}),$$

where $\mathrm{H}_{n;i}$ is defined in (16), that is, the history contribution (from $[0; T_n]$) is computed on the coarse grid and is consistent with $D_t^\alpha y$, see Fig. 2. In other words, the contribution, the fine-grid $\sum_{j=1}^{nR+i+1} w_j^{(\alpha)} y_{\Delta t}^{nR+i+1-j;k}$ on $[0; T_n]$, is replaced by a coarse grid approximation $H_{n;i} + \sum_{j=1}^{i+1} w_j^{(\alpha)} y_{\Delta t}^{nR+i+1-j;k}$. These algebraic equations are explicit and are solved with linear complexity. It is important to notice that at this stage, and say at time t_n, each processor should have access to the solution on the coarse grid at any time prior to t_n, which is however cheap from the computational and memory usage points of view. The fine grid (local part) contribution could also be approximated using a rectangle rule; recall that the parareal method does not require that the same method be used on the fine-grid and coarse-grid.

– for $k \geq 1$, the coarse grid contribution (prediction) reads formally

$$\mathcal{C}_{\Delta T}\big(\mathbf{Y}_{\Delta T}^{n;k}\big) = \big[\mathcal{C}_{\Delta T}\big(y_{\Delta T}^{0;k}\big), \ldots, \mathcal{C}_{\Delta T}\big(y_{\Delta T}^{n;k}\big)\big],$$

and where $\mathcal{C}_{\Delta T}(y_{\Delta T}^{n;k})$ denotes the approximate solution to

$$D_t^\alpha y(t) = -\lambda y(t) + g(t), \quad t \in [T_n; T_{n+1}],$$

computed on the coarse grid using Gorenflo's scheme (7).

The convergence criterion is as follows. We repeat the iterations for $1 \leq k \leq k_\infty$, until convergence (say in ℓ^∞-norm on $\mathbb{R}^{n+1}$)

$$\max_{1\leq n\leq N} \left\| \mathbf{Y}^{n;k+1}_{\Delta T} - \mathbf{Y}^{n;k}_{\Delta T} \right\|_\infty \leq \delta,$$

where δ is a small fixed parameter, and the corresponding k is denoted by $k_\infty \in \mathbb{N}^*$. The converged parareal-Gorenflo solution at final time is then $\mathbf{Y}^{N;k_\infty}_{\Delta T}$.

We propose an analysis of the order of the parareal method for the fractional equation

$$D^\alpha_t y = -\lambda y,$$

approximated by the parareal-Gorenflo scheme. We propose to follow and then extend the analysis presented in [29].

Proposition 2.2 *Assume that the algorithm on the fine grid is exact. Then, at iteration k, the parareal scheme has order k, that is, there exists $c_k(T)$ such that*

$$\left| y^{n;k}_{\Delta T} - y(T_n) \right| + \max_{t\in[T_n;T_{n+1}]} \left| y^{n;k}_{\Delta t}(t) - y(t) \right| \leq c_k(T)\Delta T^k \quad \textit{for all } 0 \leq n \leq N-1,$$

where y denotes the exact solution.

Notice that the extension to the case where the fine grid resolution is obtained by an approximate numerical method (as described in this paper) is more technical and is not presented here (see discussion below), but as in the classical parareal method for ODEs a similar conclusion is expected. We also refer to [38], where the authors arrive at the same conclusion with a different method. In order to prove the above proposition, we follow the same steps as [29].

Proof The proof relies on the estimate of the jumps defined as follows:

$$\delta^{n+1;k} = -\lambda \Delta T^\alpha \delta^{n;k} - \sum_{i=1}^{n+1} w^{(\alpha)}_i \delta^{n+1-i;k} + S^{n;k},$$

and we define

$$\Delta^{n;k} := \left(\delta^{1;k}, \ldots, \delta^{n;k}\right)^T, \qquad \boldsymbol{S}^{n;k} := \left(0, S^{1;k}, \ldots, S^{n-1;k}\right)^T, \qquad \boldsymbol{Y}^{n;k} := \left(y^{1;k}_{\Delta T}, \ldots, y^{n;k}_{\Delta T}\right)^T,$$

where

$$S^{n;k} = y^{n-1;k}_k(T_n) - y^{n;k}_{\Delta T} = G(\lambda, \Delta T) y^{n-1;k}_{\Delta T} - y^{n;k}_{\Delta T},$$

where we have denoted by G the exact semi-group for the FODE. We next introduce the lower triangular matrix $\boldsymbol{M}_n$ in $\mathbb{R}^{n\times n}$

$$\boldsymbol{M}_n \Delta^{n;k} = \boldsymbol{S}^{n;k},$$

where the pth line $\boldsymbol{M}_n^{(p)}$ of $\boldsymbol{M}_n$ is defined as

$$\boldsymbol{M}_n^{(p)} = \bigl[-w_{p-1}^{(\alpha)}, \ldots, -w_1^{(\alpha)} - \lambda \Delta T^{\alpha}, 1, 0, \ldots, 0\bigr],$$

and $\boldsymbol{Y}^{n;k+1} = \boldsymbol{y}^{n;k} + \Delta^{n;k} = \boldsymbol{y}^{n;k} + \boldsymbol{M}_n^{-1}\boldsymbol{S}^{n;k}$. In particular

$$y_{\Delta T}^{n;k+1} = y_{\Delta T}^{n-1;k}(T_n) + \delta^{n;k} = G(\lambda, \Delta T) y_{\Delta T}^{n-1;k} + \delta^{n;k}.$$

We denote $\varepsilon^{n;k+1} = y_{\Delta T}^{n;k+1} - y(T_n)$, hence

$$\begin{aligned}\varepsilon^{n;k+1} &= G(\lambda, \Delta T) y_{\Delta T}^{n-1;k} - G(\lambda, \Delta T) y(T_{n-1}) \\ &\quad + \bigl(\boldsymbol{M}_n^{-1}\bigl(0, G(\lambda, \Delta T) y_{\Delta T}^{0;0} - y_{\Delta T}^{1;k}, \ldots, G(\lambda, \Delta T) y_{\Delta T}^{n-2;k} - y_{\Delta T}^{n-1;k}\bigr)^T\bigr)_n.\end{aligned}$$

Then

$$\varepsilon^{n;k+1} = G(\lambda, \Delta T)\varepsilon^{n-1;k} + \bigl(\boldsymbol{M}_n^{-1}\bigl(0, G(\lambda, \Delta T) y_{\Delta T}^{0} - y_{\Delta T}^{1;k}, \ldots, G(\lambda, \Delta T) y_{\Delta T}^{n-2;k} - y_{\Delta T}^{n-1;k}\bigr)^T\bigr)_n,$$

where we assume by *induction assumption* that $|\varepsilon^{n;k}| \leq c_k(T)\Delta T^k$. Then we rewrite

$$\begin{aligned}&G(\lambda, \Delta T) y_{\Delta T}^{p-1;k} - y_{\Delta T}^{p;k} \\ &\quad = G(\lambda, \Delta T)\bigl(y_{\Delta T}^{p-1;k} - G\bigl(\lambda, (p-1)\Delta T\bigr) y_{\Delta T}^{0}\bigr) - \bigl(y_{\Delta T}^{p;k} - G(\lambda, p\Delta T) y_{\Delta T}^{0}\bigr) \\ &\quad = G(\lambda, \Delta T)\varepsilon^{p-1;k} - \varepsilon^{p;k}.\end{aligned}$$

Next, we denote the triangular matrix $\boldsymbol{N}_n = \{N_{ij}\}_{1 \leq i,j \leq n} := \boldsymbol{M}_n^{-1}$

$$\varepsilon^{n;k+1} = G(\lambda, \Delta T)\varepsilon^{n-1;k} + \sum_{j=2}^{n} N_{nj}\bigl(G(\lambda, \Delta T)\varepsilon^{j-2;k} - \varepsilon^{j-1;k}\bigr).$$

As $N_{nn} = 1$ and $N_{n-1n} = 0$, and using exactly similar arguments as in [29] (although it is algebraically a bit more heavy), we get that there exists $c > 0$ such that

$$\bigl|\varepsilon^{n;k+1}\bigr| \leq c\Delta T^2 \sum_{j=0}^{n-1} \bigl|\varepsilon^{j;k}\bigr|.$$

That is, as $n\Delta T \leq T$, there exists $c_{k+1}(T) > 0$ such that

$$\varepsilon^{n;k+1} \leq c_{k+1}(T)\Delta T^{k+1}.$$

This concludes the proof. □

Denoting by $y_{\Delta T}^{n;k}$ (resp. $y_{\Delta t}^{n;k}$) the solution on the coarse (resp. fine) grid, we get

$$\bigl|y_{\Delta T}^{n;k} - y(T_n)\bigr| \leq \bigl|y_{\Delta T}^{n;k} - y_{\Delta t}^{n;k}\bigr| + \bigl|y_{\Delta t}^{n;k} - y(T_n)\bigr|.$$

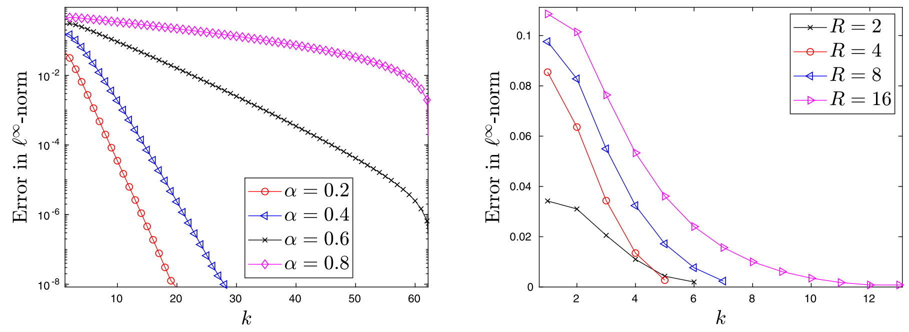


Figure 3 Numerical experiment 2. $\|\mathbf{Y}^n_{\text{exact}} - \mathbf{Y}^n_{\Delta T}\|_2$ as function of k with (Left) $R = 4$, $\Delta T = 2^{-6}$ and $\alpha = 0.2, 0.4, 0.6, 0.8$. (Right) $\Delta T = 2^{-3}$ and $R = 2, 4, 8, 16$

Then from $y^{1;k}_{\Delta T} - G(\lambda, \Delta T)y^0_{\Delta T}$ we have

$$\left|y^{1;k}_{\Delta T} - G(\lambda, \Delta T)y^0_{\Delta T}\right| \leq \left|y^{1;k}_{\Delta T} - y^{1;k}_{\Delta t}\right| + \left|y^{1;k}_{\Delta t} - G(\lambda, \Delta T)y^0_{\Delta T}\right|.$$

On the fine grid $|y^{1;k}_{\Delta t} - G(\lambda, \Delta T)y^0_{\Delta T}| \leq c\Delta t\Delta T$. This comes from the fact that the length of the interval is ΔT and Δt is the fine grid time step. By induction, we have the following.

Corollary 2.1 *At iteration k, the parareal algorithm such that Gorenflo's scheme is used on both the coarse and fine grids is an order k method, that is,*

$$\left|y^{n;k}_{\Delta T} - y(T_n)\right| + \max_{t\in[T_n;T_{n+1}]}\left|y^{n;k}_{\Delta t}(t) - y(t)\right| \leq c_k(T)\Delta T^k \quad \textit{for all } 0 \leq n \leq N-1.$$

A detailed proof can actually be deduced from the theorem of convergence presented in [38].

Numerical experiment 2. We present some simple experiments to illustrate the method developed above. We still consider a benchmark presented in [17]: for $0 < \alpha < 1$,

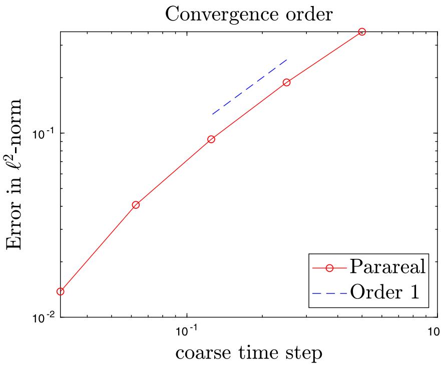
$$D^\alpha_t y(t) = -y(t) + t^2 + \frac{2t^{2-\alpha}}{\Gamma(3-\alpha)}, \qquad y(0) = 0, \quad t \in [0,1],$$

for which an explicit solution $y_{\text{exact}}(t) = t^2$ exists. We denote by $\mathbf{Y}^{n;k}_{\Delta T}$ the parareal/Gorenflo solution projected on the coarse grid at parareal iteration k. For different values of α, we then report the error $\max_{1\leq n\leq N}\|\mathbf{Y}^n_{\text{exact}} - \mathbf{Y}^{n;k}_{\Delta T}\|_2$ in logscale in Fig. 3 (Left) for different values of α, with $\Delta t = \Delta T/R$ and $R = 8$, $\Delta T = 2^{-6}$. The test shows in particular the strong dependence of the convergence rate as a function of the fractional derivative order. We also report in Fig. 3 (Right) for $\alpha = 0.5$, $\Delta T = 2^{-3}$ and respectively with $R = 2, 4, 8, 16$ subdomains the ℓ^∞-norm error as function of k, that is,

$$\max_{1\leq n\leq N}\left\|\mathbf{Y}^{n;k+1}_{\Delta T} - \mathbf{Y}^{n;k}_{\Delta T}\right\|_\infty.$$

In Fig. 4, we finally report the graph of convergence, calculated from $\max_{1\leq n\leq N}\|\mathbf{Y}^{n;k_\infty}_{\Delta T} - y(T_n)\|_\infty$ as a function of coarse time-steps $\Delta T = 1/2^i$ for $i = 3, 4, 5, 6$ and $R = 4$ subdomains. All these experiments illustrate the convergence and efficiency of the parareal method combined with Gorenflo's scheme.

Figure 4 Numerical experiment 2. Convergence graph for $R = 4$



2.3 Computational complexity

We discuss the computational complexity and data storage of Gorenflo's scheme for N time iterations on a given time grid. At iteration $1 \leq n \leq N$, the number of operations for updating the solution is linear in n, that is, the overall computational complexity is $O(N^2)$. Moreover, at iteration N, $O(N)$ data must be stored. Regarding the computational complexity of the parareal approach, we denote by N the number of iterations on the *coarse* grid and NR on the *fine* grid. At each parareal iteration k, (i) $O(N^2)$ operations are performed sequentially on the coarse grid, (ii) $O(NR^2)$ operations on each fine grid covering $[T_n; T_{n+1}]$. The total number of operations is hence $O(k_\infty N(N + R^2))$. On p processors, the complexity is $O(k_\infty N(N + R^2)/p)$ per processor. Notice that a sequential computation on a fine grid requires $O(N^2R^2)$ operations.

3 Space-time parallel algorithm for linear fractional in space-time differential equations

The approach which was presented above can be extended to fractional in space and time equation:

$$D_t^\alpha u(t,x) = -\lambda(x)D_x^\beta u(t,x), \qquad u(0,x) = u_0 \in L^2(\mathbb{R}), \tag{17}$$

with $0 < \alpha < 1$, $\beta > 0$, and $\lambda \in C_b^0(\mathbb{R})$ on $[0;T] \times \mathbb{R}$ and where D_t^α denotes Caputo's derivative and D_x^β denotes Riesz' derivative. In this case it is possible to implement a coupled pseudospectral parareal method. Formally, denoting by $\widehat{u}(t,\cdot) = \mathcal{F}_x u(t,\cdot)$ the Fourier transform in space (where ξ denotes the co-variable associated with x), we first assume that λ is a real constant. We directly have

$$D_t^\alpha \widehat{u}(t,\xi) = -\lambda(\mathtt{i}\xi)^\beta \widehat{u}(t,\xi).$$

We can then directly apply the parareal-Gorenflo's method derived above. We set $y_\xi(t) := (\mathtt{i}\xi)^\beta \widehat{u}(t,\xi)$, hence

$$D_t^\alpha y_\xi(t) = -\lambda y_\xi(t).$$

In this case the Fourier transform can easily be implemented in parallel, while the time-fractional derivative can be parallelized using the parareal method. However, whenever λ is no more constant, even the sequential algorithm is not that simple anymore.

3.1 Sequential algorithm

When λ is space-dependent, it is no more possible to efficiently directly apply a Fourier transform in space. In the latter case, we detail the proposed algorithm and we will then focus on the parallelization aspects. We denote the spatial grid points and indices as follows:

$$\mathcal{D}_{N_x} = \{x_k\}_{k\in\mathcal{O}_{N_x}}, \quad \mathcal{O}_{N_x} = \{k \in \mathbb{N}/k = 0,\dots,N_x - 1\},$$

and the uniform mesh size by $\Delta x := x_{k+1} - x_k = 2L_x/N_x$ for the entire domain $\mathcal{D} := [-L_x, L_x]$. The corresponding discrete wavenumbers are defined by $\xi_p = p\pi/L_x$ for $p \in \mathcal{P}_{N_x} := \{-N_x/2,\dots,N_x/2-1\}$. In practice, (17) is hence solved on a bounded spatial domain $[-L_x, L_x]$ with periodic boundary conditions. Regarding the pseudospectral approximations, we use the following notation:

$$\widehat{u}_p(t) = \sum_{k=0}^{N_x-1} u(t,x_k)e^{-\mathtt{i}\xi_p(x_k+L_x)}, \qquad \widetilde{u}_k(t) = \frac{1}{N_x}\sum_{p=-N_x/2}^{N_x/2-1} \widehat{u}_p(t)e^{\mathtt{i}\xi_p(x_k+L_x)}. \tag{18}$$

That is, $\widehat{u}_p(t)$ is an approximate discrete Fourier transform with and $\widetilde{u}_k(t)$ is an approximation to $u(t,x_k)$ obtained through an approximate inverse discrete Fourier transform with, for some $c > 0$,

$$\max_{k\in\mathcal{O}_{N_x}} \bigl|\widetilde{u}_k(t) - \mathcal{F}^{-1}(u)(t,x_k)\bigr| \leq cN_x^{1/2-s}\bigl\|u(t,\cdot)\bigr\|_{H^s},$$

for $s > 1/2$ (in 1d) and $u(t,\cdot) \in L^1 \cap H^s$ periodic. In general, we do not have $u(t,x_k) \neq \widetilde{u}_k(t)$. Such pseudospectral projection is for instance studied in [8]. Typically, the high modes which are neglected in the above approximation lead to the following aliasing error estimates for $u(t,\cdot) \in H^r$: there exists $c > 0$ such that

$$\bigl\|\widetilde{\widehat{u}}(t,\cdot) - u(t,\cdot)\bigr\|_{H^s} \leq cN_x^{s-r}\bigl\|u(t,\cdot)\bigr\|_{H^r},$$

for some $r > s > 1/2$ (in 1d) and $u(t,\cdot) \in L^1 \cap H^r$ periodic. We yet refer to [8] for details.

We then introduce a discrete operator $[[D_x^\beta]]$ approximating $\mathcal{F}^{-1}((\mathtt{i}\xi)^\beta\mathcal{F})$, that is, $D_x^\beta u(t,x_k)$ is approximated by

$$\bigl[[D_x^\beta]\bigr]u_k(t) := \frac{1}{N_x}\sum_{p=-N_x/2}^{N_x/2-1} (\mathtt{i}\xi_p)^\beta\widehat{\widetilde{u}}_p(t)e^{\mathtt{i}\xi_p((x_k+L_x)+\beta\pi)}. \tag{19}$$

This approximation was analyzed in [6] for approximating space fractional partial differential equations or the Dirac equation in [5]. Interestingly, when solving the equation as an initial boundary value problem (on a bounded domain $[-L_x, L_x]$), the above spectral approach is still applicable. Indeed, imposing periodic boundary conditions $u(t,L_x) = u(t,-L_x)$, it is in principle possible to include absorbing layers $[-L_x, -L_x^*] \cup [L_x^*, L_x]$ in (17) in order to avoid (i) potential artificial wave reflections and to absorb (ii) periodic waves traveling from one boundary to the next. Basically, denoting by S an absorbing function

$$S(x) = \begin{cases} 1 & \text{if } |x| < L_x^*, \\ 1 + e^{\mathtt{i}\theta}\widetilde{\sigma}(x) & \text{if } L_x^* \leq |x| < L_x, \end{cases}$$

where the absorbing function $\widetilde{\sigma} : \mathcal{D} \to \mathbb{R}$ is defined [3] as ($\alpha \in \mathbb{N}^*$)

$$\widetilde{\sigma}(x) = \begin{cases} \sigma(|x| - L_x), & L_x^* \leq |x| < L_x, \\ 0, & |x| < L_x^* \end{cases} \tag{20}$$

with traditional absorbing function σ absorbing functions. The rotation angle θ is usually fixed by the problem under study. Hence the equation is for instance transformed into

$$D_t^\alpha u(t,x) = -\frac{\lambda(x)}{S^\beta(x)} D_x^\beta u(t,x), \qquad u(0,x) = u_0.$$

From a practical point of view, the inclusion of the absorbing function in (17) does not complexify the approximation or analysis. In the following, we can simply consider that S is included in λ.

Hence, the parallel-in-time algorithm is applied to any $k \in \mathcal{O}_{N_x}$ with $\lambda_k = \lambda(x_k)$ to

$$D_t^\alpha u_k(t) = -\lambda_k \big[\big[D_x^\beta\big]\big] u_k(t).$$

It was shown and numerically observed in [2, 5] that applied to fractional in-space (only) partial differential equations or to partial differential equations this (i) matrix-free, (ii) Fourier-based pseudospectral approximation allows for spectral convergence, while avoiding explicit convolution product computations. We then rewrite

$$u_k(t) = u_k(0) - \frac{\lambda_k}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \big[\big[D_x^\beta\big]\big] D_\tau u_k(\tau)\, d\tau.$$

This pseudospectral-type approach was also used for instance for fractional and partial differential equations in [1, 3, 4, 6]. The pseudospectral-Gorenflo method reads on a (coarse) time-grid, $T_n = n\Delta T$ with $n = 0, \dots, N_x$ and $k \in \mathcal{O}_{N_x}$

$$u_k^{n+1} = -\lambda \Delta T^\alpha u_k^n - \lambda_k \sum_{i=1}^{n+1} w_i^{(\alpha)} \big[\big[D_x^\beta\big]\big] u_k^{n+1-i}, \tag{21}$$

where $\mathbf{u}^n = \{u_k^n\}_{k \in \mathcal{O}_{N_x}}$ approximates $\{u(T_n, x_k)\}_{k \in \mathcal{O}_{N_x}}$ and

$$\big[\big[D_x^\beta\big]\big] u_k^n := \frac{1}{N_x} \sum_{p=-N_x/2}^{N_x/2-1} (\mathtt{i}\xi_p)^\beta \widehat{u}_p^n e^{\mathtt{i}\xi_p((x_k+L_x)+\beta\pi)}. \tag{22}$$

Denoting $\Lambda := [\lambda_1, \dots, \lambda_{N_x}]^T \otimes \mathbf{1} \in \mathbb{R}^{N_x \times N_x}$, and $\mathbf{1}$ the unit vector in $\mathbb{R}^{N_x}$, the numerical scheme reads overall

$$\mathbf{u}^{n+1} = -\Delta T^\alpha \Lambda \mathbf{u}^n - \Lambda \sum_{i=1}^{n+1} w_i^{(\alpha)} \big[\big[D_x^\beta\big]\big] \mathbf{u}^{n+1-i}. \tag{23}$$

From traditional numerical analysis (see [8, 10, 11, 24, 26]), we expect that (i) for $u_0 \in H^s(\mathbb{R})$, and (ii) denoting $\mathbf{u}^n$ the pseudospectral parareal approximation of the $u(T_n, \cdot)$, there

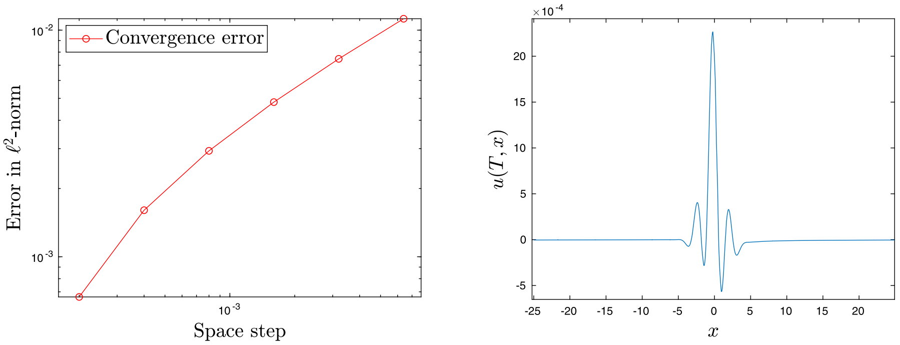


Figure 5 Numerical experiment 3. (Left) Graph of convergence $\|\mathbf{u}_{\Delta T}^{N_T} - u_{\text{exact}}(T,\cdot)\|_2$ as function of ΔT in logscale. (Right) Solution to (24) at $t = T$

exists $c(T;u_0;k) > 0$ such that after k parareal iterations

$$\sup_{1\leq n\leq N_T} \left\|\mathbf{u}^n - u(T_n,\cdot)\right\|_2 \leq c(k;T;u_0)\big(\Delta T^k + N_x^{-s}\big),$$

where $\|\cdot\|_2$ is the $\ell^2(\Delta x\mathbb{Z})$.

Numerical experiment 3. In order to illustrate the spectral convergence in space, we propose a simple test over $[0;T] \times [-L_x, L_x]$ with periodic boundary conditions at $\pm L_x$

$$D_t^{1/2}u(t,x) = -\lambda(x)D_x^{1/2}u(t,x), \qquad u(0,x) = u_0, \tag{24}$$

with $u_0(x) = \exp(-x^2 + \mathrm{i}k_0x)\cos(\pi x/2)/\mathcal{N}$, where $k_0 = -1$, $T = 0.1$, $L_x = 25$, $\lambda(x) = \exp(-x^2/100)$, and $\mathcal{N}$ is a normalization constant such that $\|u_0\|_{L^2} = 1$. As far as we know, there is no explicit solution to this equation. We choose $\Delta T = 10^{-2}$ and make Δx vary. We report in logscale the ℓ^2-norm error with a solution of reference at time $t = T$ (computed on a 5×10^5-point grid) as function of the space step from to 64×10^{-4} to 4×10^{-4} in Fig. 5 (Left), and the solution at final time T (Right). The experiment again shows the nice convergence properties of the proposed parallel methodology.

3.2 Space and time parallelization

In the following, we assume that p nodes/processors are used to solve in parallel a fractional space-time differential equations.

Parallelization in space. The parallelization in space is relatively straightforward and is two-fold. We assume in the following that we have access to p processors.

1. It is first based on the parallelization of the discrete Fourier transform (or fast Fourier transform), namely of $[[D_x^\beta]]u_k^n$, which approximates $\mathcal{F}^{-1}((\mathrm{i}\xi)^\beta\mathcal{F})u(t,x_k)$. The parallelization in space is hence first based on the parallel computation of finite sums involved in the discrete Fourier transforms. This is typically implemented thanks to FFTW [18] on $\mathcal{O}_{N_x}$ and $\mathcal{P}_{N_x}$.
2. Once $[[D_x^\beta]]\mathbf{u}^n$ is computed, in order to update (in time) the approximate solution, we decompose

$$\mathbf{u}_\ell^n = \big\{u_{k+(\ell-1)N_x^{(p)}}^n\big\}_{1\leq k\leq N_x^{(p)}}, \qquad \mathbf{u}^n = \big\{u_{k+(\ell-1)N_x^{(p)}}^n\big\}_{1\leq k\leq N_x^{(p)};1\leq\ell\leq p},$$

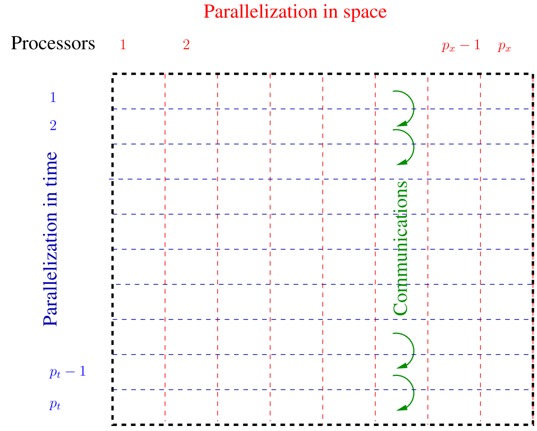


Figure 6 Parallelization in space and time

where $N_x^{(p)} = N_x/p$ and $1 \leq \ell \leq p$. The following algorithm is *embarrassingly* parallel and does not require any communication: for $\ell = 1, \ldots, p$,

$$\mathbf{u}_\ell^{n+1} = -\Delta T^\alpha \Lambda \mathbf{u}_\ell^n - \Lambda \sum_{i=1}^{n+1} w_i^{(\alpha)} \bigl[\bigl[D_x^\beta\bigr]\bigr] \mathbf{u}_\ell^{n+1-i}. \tag{25}$$

For a very large number of processors $p \in \mathbb{N}\backslash\{0\}$, the first step of this parallelization in space, more specifically the FFT parallelization, can be become inefficient (corresponding to the discrete Fourier transform parallelization), in particular of course if $p = O(N_x)$. The latter condition can indeed occur with high performance computers which can possess hundreds of thousands of processors. A coupling with parallelization in time becomes relevant and is thus discussed in the following paragraph.

Parallelization in space and time. For a given number of processors p, we denote by p_x and p_t two integers such that $p = p_x + p_t$. The key element in the simultaneous parallelization in space and time is the commutation of D_t^α and $\lambda(x)D_x^\beta$, that is, $[D_t^\alpha, \lambda(x)D_x^\beta] = 0$, where $[\cdot,\cdot]$ is the operator commutator. Whenever p is very large, we propose a parallelization in space and time using p_x (resp. p_t) processors for the parallelization in space (resp. time). More specifically, we decompose $\mathcal{O}_{N_x}$ in p_x disjoint subdomains $\mathcal{O}_{N_x} = \bigcup_{\ell=1,\ldots,p_x} \mathcal{O}_{N_x}^{(\ell)}$ and p_t processors are used for the parallelization in time. That is, for $r_\ell \in \mathcal{O}_{N_x}^{(\ell)}$ with $\ell \in \{1, \ldots, p_x\}$, we perform a parallelization in time, as described in Sect. 2. The parallelization in space requires standard FFT parallelizations (Step 1) to compute on p_x processors $[[D_x^\beta]]\mathbf{u}^n$ and Algorithm (25) (Step 2) which is embarrassingly parallel and

$$\mathbf{u}_{\Delta T;\ell}^{n+1} = \mathcal{C}_{\Delta T}\bigl(\mathbf{u}_{\Delta T;\ell}^n\bigr) - \mathcal{C}_{\Delta T}\bigl(\mathbf{u}_{\Delta T;\ell}^n\bigr) + \mathcal{F}_{\Delta T}\bigl(\mathbf{u}_{\Delta t;\ell}^{n;}\bigr), \tag{26}$$

where we have denoted by $\mathbf{u}_{\Delta T;\ell}^n = \{\mathbf{u}_{\Delta T;\ell}^{n;k}\}_k$ the coarse grid in-time approximation of u (i) at time T_n, (ii) iteration k, (iii) and $r_\ell \in \mathcal{O}_{N_x}^{(\ell)}$. The parallelization in time requires communications, see Fig. 6.

4 Conclusion

We have proposed a simple extension of the parareal method combined with a fractional equation solver, namely Gorenflo's scheme. The same strategy, based on the parareal method, can naturally be adapted to other FODE solvers and still benefit from the outstanding convergence properties of the parareal method. Some mathematical properties, such as stability, accuracy, etc., were also proposed along with numerical experiments (Figs. 1, 4). The latter allowed to illustrate the analytical properties proven in the paper and to show the feasibility of the approach from a practical point of view. Extension to parallel algorithms to space-time fractional PDE solvers was also proposed. The spatial parallelization was relying on (i) the Riesz derivative and (ii) the parallelization of the fast Fourier transform, and was successfully combined with the parareal-based FODE solver, see Fig. 5.

The extension of the proposed strategy to nonlinear scalar equations is currently investigated as well as its application to fractional physical models.

Acknowledgements
Not applicable.

Funding
No funding.

Availability of data and materials
Not applicable.

Competing interests
The author declares that he has no competing interests.

Authors' contributions
All authors read and approved the final manuscript.

Publisher's Note

Received: 5 February 2020 Accepted: 31 May 2020 Published online: 11 June 2020

References

1. Antoine, X., Besse, C., Rispoli, V.: High-order IMEX-spectral schemes for computing the dynamics of systems of nonlinear Schrödinger/Gross–Pitaevskii equations. J. Comput. Phys. **327**, 252–269 (2016)
2. Antoine, X., Fillion-Gourdeau, F., Lorin, E., MacLean, S.: Pseudospectral computational methods for the time-dependent Dirac equation in static curved spaces. J. Comput. Phys. **411**, 109412 (2020)
3. Antoine, X., Geuzaine, C., Tang, Q.: Coupling spectral methods and perfectly matched layer for simulating the dynamics of nonlinear Schrödinger equations. Application to rotating Bose–Einstein condensates (2019, submitted)
4. Antoine, X., Lorin, E.: Computational performance of simple and efficient sequential and parallel Dirac equation solvers. Comput. Phys. Commun. **220**, 150–172 (2017)
5. Antoine, X., Lorin, E.: A simple pseudospectral method for the computation of the time-dependent Dirac equation with perfectly matched layers. J. Comput. Phys. **395**, 583–601 (2019)
6. Antoine, X., Lorin, E.: Towards perfectly matched layers for time-dependent space fractional PDEs. J. Comput. Phys. **391**, 59–90 (2019)
7. Antoine, X., Lorin, E., Zhang, Y.: Derivation and analysis of computational methods for fractional Laplacian equations with absorbing layers (2020, submitted)
8. Bardos, C., Tadmor, E.: Stability and spectral convergence of Fourier method for nonlinear problems: on the shortcomings of the 2/3 de-aliasing method. Numer. Math. **129**(4), 749–782 (2015)
9. Baudron, A.-M., Lautard, J.-J., Maday, Y., Mula, O.: The parareal in time algorithm applied to the kinetic neutron diffusion equation. Lect. Notes Comput. Sci. Eng. **98**, 437–444 (2014)
10. Chechkin, A.V., Gorenflo, R., Sokolov, I.M.: Retarding subdiffusion and accelerating superdiffusion governed by distributed-order fractional diffusion equations. Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Topics **66**(4), 7 (2002)
11. Chechkin, A.V., Gorenflo, R., Sokolov, I.M.: Fractional diffusion in inhomogeneous media. J. Phys. A, Math. Gen. **38**(42), L679–L684 (2005)
12. De Oliveira, E.C., Tenreiro Machado, J.A.: A review of definitions for fractional derivatives and integral. Math. Probl. Eng. **2014**, Article ID 238459 (2014)
13. Di Nezza, E., Palatucci, G., Valdinoci, E.: Hitchhiker's guide to the fractional Sobolev spaces. Bull. Sci. Math. **136**(5), 521–573 (2012)
14. Diethelm, K., Ford, N.J.: Analysis of fractional differential equations. J. Math. Anal. Appl. **265**(2), 229–248 (2002)
15. Ezzat, M.A., El-Karamany, A.S., El-Bary, A.A.: Thermo-viscoelastic materials with fractional relaxation operators. Appl. Math. Model. **39**(23–24), 7499–7512 (2015)

16. Fischer, P.F., Hecht, F., Maday, Y.: A parareal in time semi-implicit approximation of the Navier–Stokes equations. Lect. Notes Comput. Sci. Eng. **40**, 433–440 (2005)
17. Ford, N.J., Connolly, J.A.: Comparison of numerical methods for fractional differential equations. Commun. Pure Appl. Anal. **5**(2), 289–307 (2006)
18. Frigo, M., Johnson, S.G.: The design and implementation of FFTW3. Proc. IEEE **93**(2), 216–231 (2005)
19. Gambo, Y.Y., Ameen, R., Jarad, F., Abdeljawad, T.: Existence and uniqueness of solutions to fractional differential equations in the frame of generalized Caputo fractional derivatives. Adv. Differ. Equ. **13**, 134 (2018)
20. Gander, M., Petcu, M.: Analysis of a modified parareal algorithm for second-order ordinary differential equations. In: AIP Conference Proceedings, vol. 936, pp. 233–236 (2007)
21. Gander, M.J., Jiang, Y.-L., Li, R.-J.: Parareal Schwarz waveform relaxation methods. Lect. Notes Comput. Sci. Eng. **91**, 451–458 (2013)
22. Gander, M.J., Vandewalle, S.: Analysis of the parareal time-parallel time-integration method. SIAM J. Sci. Comput. **29**(2), 556–578 (2007)
23. Garrappa, R.: Numerical solution of fractional differential equations: a survey and a software tutorial. Mathematics **6**(2), 16 (2018)
24. Goodman, J., Hou, T., Tadmor, E.: On the stability of the unsmoothed Fourier method for hyperbolic equations. Numer. Math. **67**(1), 93–129 (1994)
25. Gorenflo, R.: Fractional calculus: some numerical methods. In: CISM Courses and Lectures, vol. 378. Springer, Berlin (1997)
26. Gorenflo, R., Mainardi, F., Moretti, D., Paradisi, P.: Time fractional diffusion: a discrete random walk approach. Nonlinear Dyn. **29**(1–4), 129–143 (2002)
27. Kumar, S., Kumar, A., Momani, S., Aldhaifallah, M., Nisar, K.S.: Numerical solutions of nonlinear fractional model arising in the appearance of the stripe patterns in two-dimensional systems. Adv. Differ. Equ. **2019**, 413 (2019)
28. Li, X., Xu, C.: Existence and uniqueness of the weak solution of the space-time fractional diffusion equation and a spectral method approximation. Commun. Comput. Phys. **8**(5), 1016–1051 (2010)
29. Lions, J.-L., Maday, Y., Turinici, G.: Résolution d'EDP par un schéma en temps "pararéel". C. R. Acad. Sci., Sér. 1 Math. **332**(7), 661–668 (2001)
30. Lischke, A., Pang, G., Gulian, M., Song, F., Glusa, C., Zheng, X., Mao, Z., Cai, W., Meerschaert, M., Ainsworth, M., Karniadakis, G.E.: What is the fractional Laplacian? (2018) arXiv:1801.09767v2
31. Logeswari, K., Ravichandran, C.: A new exploration on existence of fractional neutral integro-differential equations in the concept of Atangana–Baleanu derivative. Phys. A, Stat. Mech. Appl. **544**, 123454 (2020)
32. Maday, Y.: Symposium: Recent Advances on the Parareal in Time Algorithms. In: AIP Conference Proceedings, vol. 1168, pp. 1515–1516 (2009)
33. Mainardi, F., Gorenflo, R.: On Mittag-Leffler-type functions in fractional evolution processes. J. Comput. Appl. Math. **118**(1–2), 283–299 (2000)
34. Ravichandran, C., Logeswari, K., Jarad, F.: New results on existence in the framework of Atangana–Baleanu derivative for fractional integro-differential equations. Chaos Solitons Fractals **125**, 194–200 (2019)
35. Scalas, E., Gorenflo, R., Mainardi, F.: Fractional calculus and continuous-time finance. Phys. A, Stat. Mech. Appl. **284**(1), 376–384 (2000)
36. Tarasov, V.E.: On chain rule for fractional derivatives. Commun. Nonlinear Sci. Numer. Simul. **30**(1–3), 1–4 (2016)
37. Veeresha, P., Baskonus, H.M., Prakasha, D.G., Gao, W., Yel, G.: Regarding new numerical solution of fractional schistosomiasis disease arising in biological phenomena. Chaos Solitons Fractals **2020**, 133 (2020)
38. Xu, Q., Hesthaven, J.S., Chen, F.: A parareal method for time-fractional differential equations. J. Comput. Phys. **293**, 173–183 (2015)
39. Zeid, S.S.: Approximation methods for solving fractional equations. Chaos Solitons Fractals **125**, 171–193 (2019)
40. Zhou, Y.: Basic Theory of Fractional Differential Equations. World Scientific, Singapore (2014)