A Parareal Method for Time-fractional Differential Equations

Qinwu Xu, Jan S Hesthaven, Feng Chen

Significance and Novelty

The paper provides the first parallel-in-time method for the time-fractional (partial) differential equations, as the major novelty of our work. To match the global feature of fractional derivatives, the new method has in the correction step embraced the history part of the solution. The second contribution is a systematic study of the convergence behavior in terms of the coarse resolution and the nature of the differential operators. Thirdly, it is the first time that a multi-domain spectral integrator together with a parareal method has been used for integral equations, which makes the numerical scheme very efficient. Last but not least, the paper provides a convergence analysis under the assumption of Lipschitz stability conditions.

A Parareal Method for Time-fractional Differential Equations

Qinwu Xu [*]   Jan S Hesthaven [†]   Feng Chen [‡]

Abstract

In this paper, a parareal method is proposed for the parallel-in-time integration of time-fractional differential equations (TFDEs). It is a generalization of the original parareal method, proposed for classic differential equations. To match the global feature of fractional derivatives, the new method has in the correction step embraced the history part of the solution. We provide a convergence analysis under the assumption of Lipschitz stability conditions. We use a multi-domain spectral integrator to build the serial solvers and numerical results demonstrate the feasibility of the new approach and confirm the convergence analysis. Studies also show that both the coarse resolution and the nature of the differential operators can affect the performance.

Keywords. fractional calculus, time-fractional, parareal, parallel-in-time, multi-domain spectral.

1   Introduction

Fractional calculus is an old concept that can be dated back to G.W. Leibniz and L. Euler in 1660s. Recently it has attracted considerable attention within the natural and social sciences due to its ability to model phenomena dominated by memory effects [11, 10] and problems exhibiting non-Markovian behavior in time. The speed by which a system relaxes in time can be related to the order of time derivatives in the constitutive equation, leading to the differential equation of fractional order in time. In this paper, we consider numerical methods for the following equation:

$$\begin{cases} {}_0D_t^\alpha u(t) = \lambda u(t) + f(t), & t \in \Omega, \\ u(0) = u_0, \end{cases} \tag{1.1}$$

where $\Omega = (0, T]$ and ${}_0D_t^\alpha$ is the fractional differential operator in the Caputo sense.

We focus on the time integration of the TFDE as (1.1), where the time-fractional derivative emerges as an integro-differential operator, defined by the convolution of the classical derivative of the function and a

---

[*]Qinwu Xu (qw_xu@hotmail.com), School of Mathematics and Statistics, Central South University, China.

[†]Jan S Hesthaven (jan.hesthaven@epfl.ch), Mathematics Section-MATHICSE, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland.

[‡]Feng Chen (feng_chen_1@brown.edu), Division of Applied Mathematics, Brown University, Providence, RI 02912.

singular kernel of fractional power-law type. Hence, the solution to a TFDE at a certain time depends on the whole history of the solution at previous times. Considering the design of numerical schemes, an immediate consequence is that the entire past trajectory of the numerical solution must be carried forward and used in the computation at the current time step. This impacts both the storage and the cost of the numerical method, both of which may substantially increase over time. These issues will be even more prominent if partial integro-differential equations are considered. Therefore, it is natural to adopt parallelism in terms of both storage and computation.

In this paper, we will develop parallel methods based following the idea of the parareal method proposed by Lion, Maday and Turinici in 2001 [9]. Relying on a computationally inexpensive but inaccurate solver and an accurate but expensive solver, the parareal method utilizes an iterative, predictor-corrector procedure that allows the expensive solver to run across many processors in parallel. Under suitable conditions, the parareal iteration is expected to converge after a small number of iterations to the serial solution [9]. During the last decade, there has been an explosion of research work on the analysis and applications of the parareal method to diverse scientific fields, including quantum control [13, 12], plasma simulation [16, 15], fluid dynamics [6, 5], finance [1], Volterra integral equations [8], and stochastic calculus [17], demonstrating its potential, accuracy, and robustness. However, there is currently no development of the parallel-in-time method to fractional differential equations.

This paper is an effort to close the gap between the high computational demand of the TFDE and the utilization of parallel computing. The original parareal formula for classic ODEs must be be modified to take into account the memory effect of the TFDE and we propose to include the history part of the solution in the correction step. On one hand, this generalized parareal method maintains the nice properties of the original parareal method, i.e., it is guaranteed to recover the fine solution u_k after k iterations and the fine solution can be parallelized across a large number of processors. On the other hand, it has particular advantages for the TFDE in that the numerical solution on each interval is stored on each processor, thus the memory restriction is relaxed. In this paper, we provide a convergence analysis of the new method and perform computational studies of the convergence behavior to further illuminate the importance of various factors such as the coarse solver and the different orders of derivatives. In general, the parareal method works well for dissipative problems but instability appears for problems with wave nature [7, 3]. In [8], a similar modification has been considered for Volterra integral equations with smooth kernels.

To implement the parareal method, one also needs to be cognizant of the challenge in the serial time stepping scheme for the TFDE. In this paper, we adopt a recent proposed spectral element approach [4] to build both the coarse and the fine solvers, needed in the parareal algorithm. The whole time domain will be decomposed into intervals, on which spectral approximations with low and high order polynomials are established. As such, the interpolation between two Gaussian grids are needed. For a desired accuracy, this method requires very low storage since it possesses the hp-convergence.

The paper is organized as follows. In the next section, a brief background on the parareal method and the TFDE will be provided. In Section 3, we describe the new parareal method and perform a convergence analysis. Numerical results will be shown in Section 4. We end the paper with a few of concluding remarks in Section 5.

2

2 Preliminaries

In this section, we provide background material on the parareal method for classic ODEs and introduce the fractional differential equation.

2.1 The Parareal Method

Consider the following scalar ODE:

$$\begin{cases} u_t = F(u,t), & t \in \Omega := (0,T], \\ u(0) = u_0, \end{cases} \tag{2.1}$$

where u is the unknown solution, and $F(u,t)$ is a given function in u and t. In the following, we describe the basic settings of the original parareal method (cf. [9]), aiming to solve (2.1) parallel-in-time.

First, the time domain is decomposed into N elements:

$$0 = T_0 < \cdots < T_n < \cdots < T_N = T, \quad T_n = n\Delta T, \quad \Delta T = \frac{T}{N}, \quad \Omega^n := (T_{n-1}, T_n). \tag{2.2}$$

Next, we introduce a fine integrator, $\mathcal{F}$, which can integrate the problem forward at the desired accuracy but possibly at a high computational expense. We also introduce a coarse integrator, $\mathcal{G}$, which is assumed to have a lower accuracy but providing results at a lower computational cost as compared to the fine solver. As an example, one can use the same classic fourth-order Runge-Kutta scheme for both $\mathcal{F}$ and $\mathcal{G}$ but integrate $\mathcal{F}$ using a small step size δt, while $\mathcal{G}$ is utilizing a bigger step size Δt ($\Delta t \gg \delta t$).

Denote $\mathcal{F}_{\Delta T}(u; T_n, T_{n+1})$ as the numerical solution obtained by integrating u from T_n to T_{n+1} with $\mathcal{F}_{\delta t}$ and $\mathcal{G}_{\Delta T}(u; T_n, T_{n+1})$ denoting the similar integration forward in time using the coarse solver. Figure 1 provides an illustration. Without ambiguity, we denote them as $\mathcal{F}$ and $\mathcal{G}$ for short.
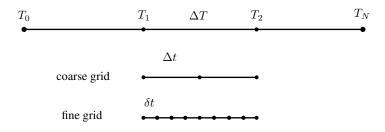

Figure 1: Decomposition of the time domain in the parareal method for classic differential equations.

In the parareal method, one starts the iterations from an initial guess, $\{u_n^0\}_{n=0}^N$, obtained from the coarse integrator $\mathcal{G}$. Assume that $\{u_n^k\}_{n=0}^N$ are available. The solution at the next iteration is obtained from

$$u_{n+1}^{k+1} = \mathcal{G}u_n^{k+1} + \mathcal{F}u_n^k - \mathcal{G}u_n^k, \quad 0 \leqslant n \leqslant N-1, \quad k = 0, 1, 2, \cdots \tag{2.3}$$

It is easy to see that the different $\mathcal{F}u_n^k$'s can be computed in parallel. Therefore, N in (2.2) represents the number of processors. Assume that $\mathcal{F}$ and $\mathcal{G}$ are continuous and $\lim_{k\to\infty} u_n^k$ exists. By taking the limit of $k \to \infty$ in (2.3), we obtain:

$$u_{n+1}^{k+1} \to u_{n+1} = \mathcal{F}_{\Delta T}(u_n; T_n, T_{n+1}). \tag{2.4}$$

In other words, the parareal solution converges to the solution obtained from the sequential fine integrator. One terminates the parareal algorithm if $k = n_{\max}$, i.e., the maximum number of iterations is reached, or, if

$$|u_n^k - u_n^{k-1}| \leqslant \tau, \tag{2.5}$$

where τ is a given tolerance. Clearly, the smaller the number of iterations, the more efficient the parareal method. It is not hard to see that the parareal method is general and can be applied to time-dependent partial differential equations.

2.2 Fractional Differential Equations

Let us denote the time domain $(0, T)$ as Ω, and let $t \in \Omega$. The fractional integral of order α of a given function $u(t)$ is defined as

$$_0D_t^{-\alpha}u(t) \triangleq \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1}u(s)\mathrm{d}s, \tag{2.6}$$

where $\Gamma(x)$ is the Gamma function. This allows us to define the Caputo fractional derivative of order $\alpha, (0 < \alpha \leq 1)$ as

$$_0D_t^{\alpha}u(t) \triangleq \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-s)^{n-1-\alpha}\frac{\mathrm{d}^n u(s)}{\mathrm{d}s^n}\mathrm{d}s. \tag{2.7}$$

which is preferred over alternative definitions to deal with standard initial conditions. This definition can naturally be extended to higher order as

$$_0D_t^{\alpha}u(t) \triangleq {_0D_t^{\alpha-n}}\frac{\mathrm{d}^n u(t)}{\mathrm{d}t^n} = \frac{1}{\Gamma(n-\alpha)} \int_0^t (t-s)^{n-1-\alpha}\frac{\mathrm{d}^n u(s)}{\mathrm{d}s^n}\mathrm{d}s, \tag{2.8}$$

where n is an integer such that $\alpha \in (n-1, n]$.

With this notation we express the multi-term fractional differential equation as:

$$\begin{cases} \sum_{k=1}^m a_k(t)\,_0D_t^{\alpha_k}u(t) = \mathcal{L}u(t) + f(t), & t \in \Omega, \\ u^{(k)}(0) = u_k, & 0 \leqslant k \leqslant n-1. \end{cases} \tag{2.9}$$

Here we assume that $0 \leqslant \alpha_1 < \alpha_2 < \cdots < \alpha_m$ and $\alpha_m \in (n-1, n]$. On the right hand side of the equation, $\mathcal{L}$ can be either a scalar function for the time-fractional differential equation (TFDE) case or an operator in a spatial domain for the time-fractional partial differential equation (TFPDE) case. We refer to [14] for an introduction to fractional calculus.

3 The Parareal Method for TFDEs

In this section, we first describe how to generalize the original parareal method to fractional differential equations and then prove that the parareal solution converges to the fine solution.

3.1 Description of the Method

Without the loss of generality, consider the same domain decomposition in (2.2) and Figure 1. Denote the fine grid on Ω^n as $\{t_1^n, t_2^n, \cdots, t_M^n\}$. Let $\boldsymbol{u}_n$ be a solution vector of length M, with each component the numerical approximation to $u(t_i^n)$. Furthermore, $\boldsymbol{u}_{1:n}$ indicate a concatenation of $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n\}$. To recover $\boldsymbol{u}_{n+1}$, the time integrators $\mathcal{G}$ and $\mathcal{F}$ of the TFDE require the entire history part, $\boldsymbol{u}_{1:n}$, as the input. Hence, the parareal method for the TFDE is given as

$$\boldsymbol{u}_{n+1}^{k+1} = \mathcal{G}\boldsymbol{u}_{1:n}^{k+1} + \mathcal{F}\boldsymbol{u}_{1:n}^k - \mathcal{G}\boldsymbol{u}_{1:n}^k, \quad 1 \leqslant n \leqslant N-1, \quad k = 0, 1, 2, \cdots \tag{3.1}$$

It can be easily verified that $\boldsymbol{u}_{n+1}^{k+1} \to \mathcal{F}\boldsymbol{u}_{1:n}$ as $k \to \infty$, if $\lim_{k\to\infty} \boldsymbol{u}_{n+1}^{k+1}$ exists. Since $\mathcal{G}$ and $\mathcal{F}$ are defined on different grids, interpolations between the coarse and fine solutions may be needed to perform (3.1). Like the parareal method for classic ODEs in (2.3), the new parareal method in (3.1) goes forward in a sequential way. Unlike the classic case, however, the new parareal method uses the history part of the solution at each step (Figure 2). Another difference between the original parareal method in (2.3) and the new parareal
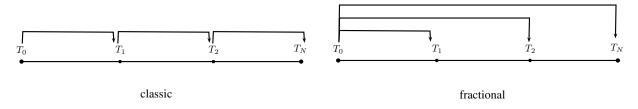

Figure 2: At each iteration, both two methods go sequentially element by element. In the fractional case, the history part is needed.

method in (3.1) is that (2.3) updates only the parareal solution at the right end point of the current element, i.e., the solution at T_n. while (3.1) updates a solution vector $\boldsymbol{u}_n^k$ containing solutions at two end points as well as interior points on the fine grid of Ω^n. This is natural since the solution at the current time depends on solutions at all previous times in the TFDE case.

3.2 Analysis of the Method

Suppose $u(t)$ is the exact solution to (1.1) and denote $\{\boldsymbol{u}_n\}_{n=1}^N$ as the numerical solution obtained from the fine propagator $\mathcal{F}$, i.e., $\boldsymbol{u}_{n+1} = \mathcal{F}\boldsymbol{u}_{1:n}$. Introduce the following two kinds of errors:

$$\boldsymbol{e}_n^k = \boldsymbol{u}_n^k - \boldsymbol{u}(t_{\Omega^n}), \quad \boldsymbol{\theta}_n^k = \boldsymbol{u}_n^k - \boldsymbol{u}_n, \tag{3.2}$$

5

where $\boldsymbol{u}(t_{\Omega^n}) = (u(t_1^n), u(t_2^n), \cdots, u(t_M^n))^T$, a vector of exact solutions on the fine grid. We denote a generic, nonnegative constant c independent of n or k. An important tool is the discrete Gronwall lemme suitable for integral equations (cf. [2]):

Lemma 3.1 (Discrete Gronwall Lemma). Let $a, b \geqslant 0$. Assume that $\{\phi_n\}$ is a nonnegative sequence satisfying

$$\phi_n \leqslant b + a \sum_{s=0}^{n-1} \phi_s, \quad n = 1, 2, \cdots. \tag{3.3}$$

Then it holds that

$$\phi_n \leqslant b \exp(na). \tag{3.4}$$

Assumption 3.2 (Convergence of $\mathcal{F}$ and $\mathcal{G}$). Let $\mathcal{F}$ and $\mathcal{G}$ denote the fine and coarse numerical propagators, respectively, and let $\boldsymbol{u}(t_{\Omega^n})$ be the vector of exact solutions defined in Section 3.1. Assume that

$$\begin{aligned} \|\mathcal{F}\boldsymbol{u}_{1:n} - \boldsymbol{u}(t_{\Omega^{n+1}})\| &\leqslant \epsilon_F, \\ \|\mathcal{G}\boldsymbol{u}_{1:n} - \boldsymbol{u}(t_{\Omega^{n+1}})\| &\leqslant \epsilon_G, \end{aligned} \tag{3.5}$$

where ϵ_F and ϵ_G decay with the order of the scheme.

Assumption 3.3 (Lipschitz condition of $\mathcal{F}$ and $\mathcal{G}$). Let $\mathcal{F}$ and $\mathcal{G}$ denote the fine and coarse numerical propagators, respectively, and let $\{\boldsymbol{\phi}_n, \boldsymbol{\psi}_n\}$ be two discrete sequences defined on Ω. We assume that

$$\begin{aligned} \|\mathcal{F}\boldsymbol{\phi}_{1:n} - \mathcal{F}\boldsymbol{\psi}_{1:n}\| &\leqslant c\Delta T \sum_{j=1}^{n} \|\boldsymbol{\phi}_j - \boldsymbol{\psi}_j\|, \\ \|\mathcal{G}\boldsymbol{\phi}_{1:n} - \mathcal{G}\boldsymbol{\psi}_{1:n}\| &\leqslant c\Delta T \sum_{j=1}^{n} \|\boldsymbol{\phi}_j - \boldsymbol{\psi}_j\|. \end{aligned} \tag{3.6}$$

Denote $\mathcal{E}$ as the exact propagator corresponding to (1.1). We define $\delta\mathcal{F} = \mathcal{E} - \mathcal{F}$ and similarly for $\delta\mathcal{G}$.

Assumption 3.4 (Lipschitz condition of $\delta\mathcal{F}$ and $\delta\mathcal{G}$). Let $\mathcal{F}$ and $\mathcal{G}$ be the fine and coarse numerical propagators, respectively, and let $\{\boldsymbol{\phi}_n, \boldsymbol{\psi}_n\}$ be two discrete sequences defined on Ω. We assume that

$$\begin{aligned} \|\delta\mathcal{F}\boldsymbol{\phi}_{1:n} - \delta\mathcal{F}\boldsymbol{\psi}_{1:n}\| &\leqslant c\Delta T\epsilon_F \sum_{j=1}^{n} \|\boldsymbol{\phi}_j - \boldsymbol{\psi}_j\|, \\ \|\delta\mathcal{G}\boldsymbol{\phi}_{1:n} - \delta\mathcal{G}\boldsymbol{\psi}_{1:n}\| &\leqslant c\Delta T\epsilon_G \sum_{j=1}^{n} \|\boldsymbol{\phi}_j - \boldsymbol{\psi}_j\|. \end{aligned} \tag{3.7}$$

Theorem 3.5 (Convergence of the Parareal Method). The error of the parareal method in (3.1) for the TFDE in (1.1) satisfies

$$\|\boldsymbol{e}_n^k\| \leqslant (cT_n)^k \exp(cT_n k)\epsilon_G^{k+1} + \epsilon_F, \tag{3.8}$$

where ϵ_G and ϵ_F were defined in Assumption 3.2.

Proof. First, use the triangle inequality and Assumption 3.2 to recover

$$\|\boldsymbol{e}_n^k\| \leqslant \|\boldsymbol{\theta}_n^k\| + \|\boldsymbol{u}_n - \boldsymbol{u}(t_{\Omega^n})\| \leqslant \|\boldsymbol{\theta}_n^k\| + \epsilon_F. \tag{3.9}$$

Step 1. Using the parareal formula, we have

$$\begin{aligned}
\boldsymbol{\theta}_n^k &= (\mathcal{G}\boldsymbol{u}_{1:n-1}^k - \mathcal{G}\boldsymbol{u}_{1:n-1}) + (\mathcal{G}\boldsymbol{u}_{1:n-1} - \mathcal{G}\boldsymbol{u}_{1:n-1}^{k-1}) + (\mathcal{F}\boldsymbol{u}_{1:n-1}^{k-1} - \mathcal{F}\boldsymbol{u}_{1:n-1}) \\
&= (\mathcal{G}\boldsymbol{u}_{1:n-1}^k - \mathcal{G}\boldsymbol{u}_{1:n-1}) + (\delta\mathcal{G}\boldsymbol{u}_{1:n-1} - \delta\mathcal{G}\boldsymbol{u}_{1:n-1}^{k-1}) + (\delta\mathcal{F}\boldsymbol{u}_{1:n-1}^{k-1} - \delta\mathcal{F}\boldsymbol{u}_{1:n-1})
\end{aligned} \tag{3.10}$$

Apply the triangle inequality, Assumption 3.3, and Assumption 3.4, to obtain

$$\begin{aligned}
\|\boldsymbol{\theta}_n^k\| &\leqslant \|\mathcal{G}\boldsymbol{u}_{1:n-1}^k - \mathcal{G}\boldsymbol{u}_{1:n-1}\| + \|\delta\mathcal{G}\boldsymbol{u}_{1:n-1} - \delta\mathcal{G}\boldsymbol{u}_{1:n-1}^{k-1}\| + \|\delta\mathcal{F}\boldsymbol{u}_{1:n-1}^{k-1} - \delta\mathcal{F}\boldsymbol{u}_{1:n-1}\| \\
&\leqslant c\Delta T \sum_{j=1}^{n-1} \|\boldsymbol{\theta}_j^k\| + c\epsilon_G \Delta T \sum_{j=1}^{n-1} \|\boldsymbol{\theta}_j^{k-1}\|.
\end{aligned} \tag{3.11}$$

Step 2. Apply the discrete Gronwall Lemma (Lemma 3.1) to (3.11) in n, to obtain

$$\|\boldsymbol{\theta}_n^{k+1}\| \leqslant c\Delta T \epsilon_G \exp(c(n-1)\Delta T) \sum_{j=1}^{n-1} \|\boldsymbol{\theta}_j^k\|. \tag{3.12}$$

By noticing $n\Delta T = T_n$, we have

$$\|\boldsymbol{\theta}_n^{k+1}\| \leqslant cT_n \epsilon_G \exp(cT_n) \max_j \|\boldsymbol{\theta}_j^k\|. \tag{3.13}$$

Step 3. A simple induction in k to (3.13) provides

$$\|\boldsymbol{\theta}_n^{k+1}\| \leqslant (cT_n)^{k+1} \exp(cT_n(k+1)) \epsilon_G^{k+1} \max_j \|\boldsymbol{\theta}_j^0\|. \tag{3.14}$$

Now notice that $\|\boldsymbol{\theta}_j^0\| \sim \epsilon_G$ and use (3.9) to complete the proof. $\square$

Remark 3.6. In (3.8), the growth factor $(cT_n)^k \exp(cT_n k)$ can be controlled and the parareal method converges. This is the case for most of the diffusion dominated problems, because c is typically negative and real. For wave problems, in which case c can have dominating imaginary component, the non-normal behavior of $(cT_n)^k \exp(cT_n k)$ becomes significant and may lead to an instability of the parareal method.

4   Numerical Examples

Wr first introduce a robust and efficient spectral method for time-fractional differential equations as the serial numerical propagator although we emphasize that one can use any type of serial solvers in the parareal method. In the second subsection, we provide the main numerical results.

4.1 Description of the Serial Solver

Before providing numerical examples for the parareal method, we describe the fine and coarse solvers used subsequently. More details can be found in [4].

First, decompose the time domain Ω into:

$$0 = t_0 < t_1 < \cdots < t_K = T, \quad \Omega^i = [t_{i-1}, t_i], \quad h_i = t_i - t_{i-1}, \quad 1 \leqslant i \leqslant K. \tag{4.1}$$

The model equation (1.1) takes the form:

$$\begin{cases} {}_{t_{i-1}}D_t^\alpha u(t) + \sum_{k=1}^{i-1} {}_{t_{k-1}}\tilde{D}_{t_k}^\alpha u(t) = \lambda u(t) + f(t), \quad t \in \Omega^i, \quad 1 \leqslant i \leqslant K, \\ u(0) = u_0, \end{cases} \tag{4.2}$$

where $\tilde{D}^\alpha$ is defined as

$${}_a\tilde{D}_b^\alpha u(t) := \frac{1}{\Gamma(n-\alpha)} \int_a^b (t-s)^{n-\alpha-1} \frac{\mathrm{d}^n}{\mathrm{d}s^n} u(s)\mathrm{d}s, \quad t > b. \tag{4.3}$$

Next, define the approximating space:

$$\mathbb{X}_N = \{u \in C(\Omega); u|_{\Omega^i} \in \mathbb{P}_N, 1 \leqslant i \leqslant K\}, \tag{4.4}$$

where $\mathbb{P}_N$ is the polynomial space of degree N defined on Ω^i, i.e., we represent the global solution as piecewise polynomials. For simplicity, we consider a uniform decomposition ($h_i \equiv h$) and the same number of degrees of freedom on each element. The numerical solution can be expressed as

$$u_h(t) = \sum_{l=0}^{N} \hat{u}_l^i P_l(\tilde{t}), \quad t \in \Omega^i, \tag{4.5}$$

where

$$t = \frac{1-\tilde{t}}{2} t_{i-1} + \frac{1+\tilde{t}}{2} t_i, \quad \tilde{t} \in [-1, 1], \tag{4.6}$$

and $P_l(\tilde{t})$ is the l-th order Legendre polynomial defined on $[-1, 1]$.

Insert (4.5) into (4.2), and enforce the equation to hold at a set of collocation points ξ_j^i, to recover

$${}_{t_{i-1}}D_t^\alpha u_h(\xi_j^i) + \sum_{k=1}^{i-1} {}_{t_{k-1}}\tilde{D}_{t_k}^\alpha u_h(\xi_j^i) = \lambda u_h(\xi_j^i) + f(\xi_j^i), \quad 0 \leqslant j \leqslant N, \quad 1 \leqslant i \leqslant K, \tag{4.7}$$

We assume that $\{\xi_j^i\}_{j=0}^N$ is the Legendre-Gauss-Lobatto points on Ω^i but this can be relaxed. Using the affine transformation in (4.6), we introduce the following matrices:

8

$$(\hat{\mathcal{M}}_0)_{jl} = (\frac{h}{2})^{-\alpha} \, _{-1}D_t^\alpha P_l(\xi_j), \quad 0 \leqslant j, l \leqslant N,$$

$$(\hat{\mathcal{M}}_{i-k})_{jl} = (\frac{h}{2})^{-\alpha} \, _{-1}\tilde{D}_1^\alpha P_l(\tilde{\xi}_j^i), \quad 0 \leqslant j, l \leqslant N, \quad 1 \leqslant k \leqslant i-1, \tag{4.8}$$

where

$$\xi_j^i = \frac{1 - \tilde{\xi}_j^i}{2} t_{k-1} + \frac{1 + \tilde{\xi}_j^i}{2} t_k. \tag{4.9}$$

The multi-domain spectral method may now be expressed as follows. Assume that $\{\hat{\boldsymbol{u}}^k\}_{k=1}^{i-1}$ were solved on previous elements, i.e., the history is known. Then (4.7) is equivalent to the following linear system in $\hat{\boldsymbol{u}}^i$:

$$\hat{\mathcal{M}}_0\hat{\boldsymbol{u}}^i + \sum_{k=1}^{i-1} \hat{\mathcal{M}}_{i-k}\hat{\boldsymbol{u}}^k = \lambda\mathcal{V}\hat{\boldsymbol{u}}^i + \boldsymbol{f}^i, \tag{4.10}$$

where $(\boldsymbol{f}^i)_j = f(t_j^i)$. The expression for $\{\hat{\mathcal{M}}_i\}$ is provided in [4].

In practice, it is preferred to work directly in physical space. Hence, we write (4.10) as

$$\mathcal{N}_0\boldsymbol{u}^i + \sum_{k=1}^{i-1} \mathcal{N}_{i-k}\boldsymbol{u}^k = \lambda(\frac{h}{2})^\alpha\boldsymbol{u}^i + (\frac{h}{2})^\alpha\boldsymbol{f}^i, \tag{4.11}$$

where

$$\mathcal{N}_k = (\frac{h}{2})^\alpha \hat{\mathcal{M}}_k \mathcal{V}^{-1}, \quad 0 \leqslant k \leqslant i-1. \tag{4.12}$$

To complete the method, one needs to enforce the initial condition on each element: the first row of (4.11) is replaced by $(1, 0, \cdots, 0)$ and the first element of the right hand side vector is replaced by $u_h(t_N^{i-1})$. If $\alpha \in (1, 2)$, the second row of (4.11) also needs to be replaced by the first row of the first-order differentiation matrix, and the right hand side vector is changed accordingly.

The above method yields the desired hp-convergence as a spectral element method provided the solution has sufficient smoothness. When $N = 1$, it is equivalent to a low-order finite difference method (linear piecewise basis in use).

Based on the multi-domain spectral method, we define the fine and coarse integrators that are used in the subsequent examples (see Figure 3 for an illustration). On each Ω^n, we use $N_G + 1$ quadrature points for the coarse grid, and $N_F + 1$ for the fine grid. For simplicity, the quadrature points are chosen as Legendre-Gauss-Lobatto points. Denote the numerical solution on the fine grid of each Ω^n as $\boldsymbol{u}_n$, which is a vector of length $N_F + 1$. The parareal method remains the form of (3.1) with interpolation between the fine and the coarse solutions needed.

4.2 Numerical results of the fractional parareal method

We use the parareal method in (3.1) and the multi-domain spectral method in (4.12) to solve the time-fractional differential equation, (1.1). Let $\alpha < 1$. All the errors and differences are calculated on the last element and measured in the maximum norm.
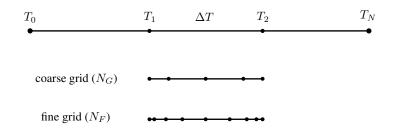
Figure 3: Decomposition of the time domain of serial solvers in the parareal method for time-fractional differential equations.

Example 4.1. In the first example, we illustrate the convergence rate and show that the rate depends on the coarse resolution determined by N_G. We choose the exact solution to be $u(t) = \sin(t)$, and the right hand side function $f(t)$ is calculated accordingly. In this example, the following parameters are used:

$$T = 4\pi, \quad \lambda = -1, \quad \alpha = 0.6, \quad N = 16, \quad N_F = 16. \tag{4.13}$$

First, we consider the difference between the parareal solution and the fine solution, as shown in Figure 4(a). The parareal solution converges to fine solution at a rate as $O(\epsilon_G^k)$, consistent with Theorem 3.5. The method reaches machine accuracy after a few of iteration steps. The error between the parareal solution and exact solution is shown in Figure 4(b). In all cases, $\epsilon_F \sim 10^{-10}$.

In the following examples (Example 4.2-4.5), we consider

$$\begin{cases} {}_0D_t^\alpha u(t) = \lambda u(t) + e^{\sin(t)}, & t \in (0, T], \\ u(0) = 1. \end{cases} \tag{4.14}$$

For this equation, it is hard to obtain the exact solution for long time integrations. We assume that a sufficiently good solution can be obtained using the fine solver and use this as the surrogate for the exact solution.

Example 4.2. Besides N_G, the convergence rate also depends on α. In Figure 5, we still use parameters in (4.13) with N_G fixed as 1. In general, the larger the α is, the faster the convergence rate is. It is caused by the fact that the solution decays exponentially as $\alpha \to 1$.

Example 4.3. We now show that the performance can be improved by increasing the number of processors as would be a clear advantage for the large scale parallelization. In the two plots of Figure 6, we solve (4.14) with the following parameters:

$$T = 15\pi, \quad \lambda = 1, \quad \alpha = 0.6, \quad N_G = 2, \quad N_F = 16. \tag{4.15}$$

As indicated in previous examples, the solution converges when N is fixed. In this example, it is shown that the error decreases algebraically as N increases, when k is fixed. This is a result of ΔT decreasing so the
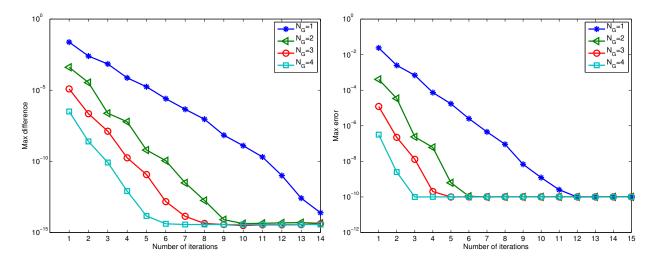
10

Figure 4: Convergence results of Example 4.1. Different coarse accuracies lead to different convergence rates. Left: difference between the parareal solution and the fine solution. Right: difference between the parareal solution and the exact solution.

coarse accuracy is improved. Therefore, we have three ways to improve the accuracy: increasing the number of iterations, improving the accuracy of the coarse solver, and increasing the number of processors.

Example 4.4. In this example, we focus on the impact of α on the convergence rate when $\Re(\lambda) = 0$. In Figure 7, the following parameters are used:

$$T = 15\pi, \quad \lambda = i, \quad N = 15, \quad N_G = 1, \quad N_F = 16, \quad \alpha = 0.1, 0.2, \cdots, 0.9, 1.0. \qquad (4.16)$$

When λ is a purely imaginary number, the TFDE tends to have wave nature. From Figure 7, the solution is seen to oscillate as $\alpha \to 1$. It coincides with the well-known instability of the parareal method for wave equations [7, 3]. As α decreases, the parareal solution recovers stability.

Example 4.5. Continuing Example 4.4, we demonstrate that the stronger the dissipation is, the better the parareal method works. In Figure 8, we consider the case where $\Re(\lambda) \leqslant 0$ and use following parameters:

$$T = 15\pi, \quad N = 15, \quad N_G = 1, \quad N_F = 16, \quad \alpha = 0.95, \quad \lambda = i, -0.5 + i, -1 + i, -2 + i. \quad (4.17)$$

Example 4.6. In the last example, we consider a TFPDE - a time-fractional diffusion equation:

$$\begin{cases} {}_0D_t^\alpha u(x,t) = \Delta u(x,t) + f(x,t), & (x,t) \in (0, 2\pi) \times (0, T], \\ u(x,0) = u_0(x), \end{cases} \qquad (4.18)$$
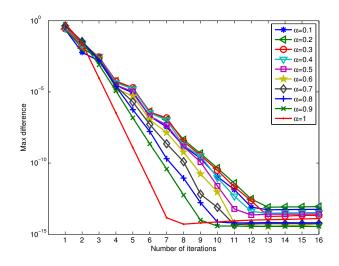
11

Figure 5: Convergence results of Example 4.2. Different α values lead to different convergence rates.

subject to 2π-periodic boundary condition. Here we choose $u(x,0) = 0$ and

$$f(x,t) = \exp(\cos(x) - 0.05t)\sin(0.2t).$$

Since u and f are periodic in the spatial domain, their numerical approximations are expressed as

$$u_h(x,t) = \sum_{|k| \leq M/2} \hat{u}_k(t)e^{ikx}, \quad f_h(x,t) = \sum_{|k| \leq M/2} \hat{f}_k(t)e^{ikx}, \tag{4.19}$$

where $i = \sqrt{-1}$ and $\hat{f}_k(t)$ can be obtained through the fast Fourier transform (FFT).

Inserting (4.19) into (4.18) and requiring the equation to be satisfied in a Fourier Galerkin sense yields a set of time-fractional differential equations:

$$\forall k: \quad {}_0D_t^\alpha \hat{u}_k(t) = -k^2 \hat{u}_k(t) + \hat{f}_k(t), \tag{4.20}$$

for which we apply the multi-domain spectral method and the new parareal method in the temporal direction. Once $\hat{u}_k$'s are recovered, u_h are obtained through the inverse fast Fourier transform. We fix the following parameters in the numerical experiments:

$$T = 100, \quad \alpha = 0.3, \quad N_G = 2, \quad N_F = 20, \quad N = 40, \quad M = 40. \tag{4.21}$$

The numerical solution and the convergence rate of the parareal method are shown in Figure 9.
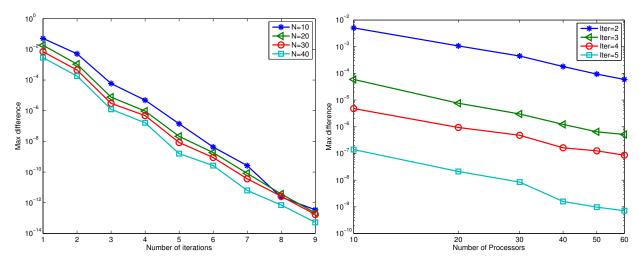
Figure 6: Convergence results with different numbers of processors in Example 4.3.

5 Conclusions

We have proposed and studied a new parareal method, suitable for time-fractional differential equations. We proved that the parareal method converges under mild assumptions and confirmed the analysis through a number of numerical experiments. This also exposed other factors influencing the overall performance. The parareal method works well for dissipative problems, but appears unstable for problems with wave nature, similar to what is known for the standard parareal method. Nevertheless, for a large class of problems, the robustness and versatility of the method has been demonstrated and sets the stage for the evaluation on more complex problems.

Acknowledgement

References

[1] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. *Lecture Notes in Computational Science and Engineering*, 23:189–202, 2002.
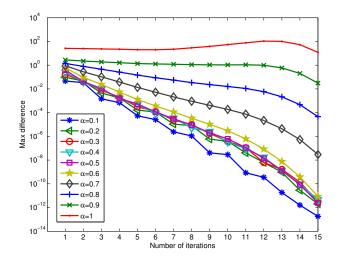
Figure 7: Convergence results for $\lambda = i$ and different α values in Example 4.4.

[2] Hermann Brunner. *Collocation methods for Volterra integral and related functional differential equations*, volume 15. Cambridge University Press, 2004.

[3] Feng Chen, Jan Hesthaven, and Xueyu Zhu. On the use of reduced basis methods to accelerate and stabilize the parareal method. In Alfio Quarteroni and Gianluigi Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *MS&A – Modeling, Simulation and Applications*. Springer, 2014.

[4] Feng Chen, Qinwu Xu, and Jan Hesthaven. A multi-domain spectral method for time-fractional differential equations. *preprint*, 2013.

[5] Franz Chouly and Miguel A. Fern-ndez. An enhanced parareal algorithm for partitioned Parabolic-Hyperbolic coupling. *AIP Conference Proceedings*, 1168(1):1517–1520, September 2009.

[6] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. *International Journal for Numerical Methods in Engineering*, 58(9):1397–1434, 2003.

[7] M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J Sci Comput*, 29:556–578, 2007.

[8] Xianjuan Li, Tao Tang, and Chuanju Xu. Parallel in time algorithm with spectral-subdomain enhancement for volterra integral equations. *SIAM Journal on Numerical Analysis*, 51(3):1735–1756, June 2013.
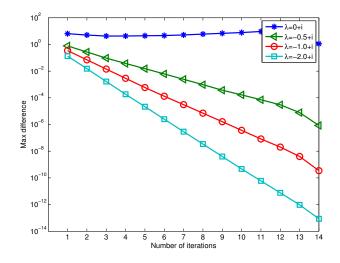
Figure 8: Convergence results for $\alpha = 0.95$ and different λ in Example 4.5.

[9] J. L Lions, Y. Maday, and G. Turinici. A "parareal" in time discretization of PDE's. *Comptes Rendus de lAcademie des Sciences Series I Mathematics*, 332(7):661–668, 2001.

[10] J. Tenreiro Machado, Alexandra M. Galhano, and Juan J. Trujillo. Science metrics on fractional calculus development since 1966. *Fractional Calculus and Applied Analysis*, 16(2):479–500, June 2013.

[11] J. Tenreiro Machado, Virginia Kiryakova, and Francesco Mainardi. Recent history of fractional calculus. *Communications in Nonlinear Science and Numerical Simulations*, 16:1140–1153, March 2011.

[12] Y. Maday, J. Salomon, and G. Turinici. Monotonic parareal control for quantum systems. *SIAM J. Numer. Anal*, 45(6):2468–2482, 2007.

[13] Yvon Maday and Gabriel Turinici. A parallel in time approach for quantum control: the parareal algorithm. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 62–66, 2002.

[14] Igor Podlubny. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*. Academic Press, October 1998.

[15] J.M. Reynolds-Barredo, D.E. Newman, R. Sanchez, D. Samaddar, L.A. Berry, and W.R. Elwasif. Mechanisms for the convergence of time-parallelized, parareal turbulent plasma simulations. *Journal of Computational Physics*, August 2012.
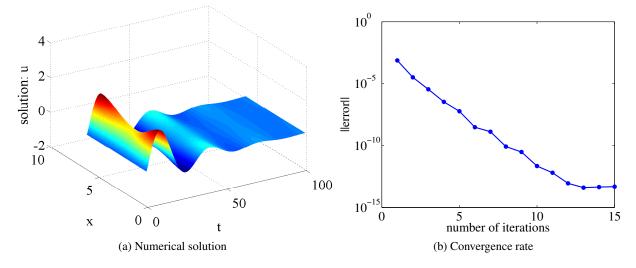
(a) Numerical solution

(b) Convergence rate

Figure 9: Numerical solution and convergence rate for Example 4.6.

[16] D. Samaddar, D. E Newman, and R. S-nchez. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. *Journal of Computational Physics*, 229(18):6558–6573, 2010.

[17] Wu Shulin, Wang Zhiyong, and Huang Chengming. Analysis of mean-square stability of the parareal algorithm. *Mathematica Numerica Sinica*, 2:002, 2011.