

A PARALLEL-IN-TIME ALGORITHM FOR HIGH-ORDER BDF METHODS FOR DIFFUSION AND SUBDIFFUSION EQUATIONS*

SHUONAN WU[†] AND ZHI ZHOU[‡]

Abstract. In this paper, we propose a parallel-in-time algorithm for approximately solving parabolic equations. In particular, we apply the k -step backward differentiation formula and then develop an iterative solver by using the waveform relaxation technique. Each resulting iteration represents a periodic-like system, which could be further solved in parallel by using the diagonalization technique. The convergence of the waveform relaxation iteration is theoretically examined by using the generating function method. The argument could be further applied to the time-fractional subdiffusion equation, whose discretization shares common properties of the standard BDF methods due to the nonlocality of the fractional differential operator. Illustrative numerical results are presented to complement the theoretical analysis.

Key words. parabolic equation, subdiffusion equation, backward differentiation formula, parallel-in-time algorithm, convergence analysis, convolution quadrature

AMS subject classifications. 65Y05, 65M15, 65M12

DOI. 10.1137/20M1355690

1. Introduction. The aim of this paper is to develop a parallel-in-time (PinT) solver for high-order time stepping schemes of diffusion models. We begin with the normal diffusion, which is described by parabolic equations. Let $V \subset H = H' \subset V'$ be a Gelfand triple of complex Hilbert spaces. Namely, the embedding $V \hookrightarrow H$ is continuous and dense, and

$$\langle u, v \rangle = (u, v) \quad \text{for all } u \in H \hookrightarrow V' \text{ and } v \in V \hookrightarrow H,$$

where $\langle \cdot, \cdot \rangle$ is the duality pairing between V' and V , and (\cdot, \cdot) is the inner product on H . Throughout, we let $\|\cdot\|$ and $\|\cdot\|_V$ denote the norms of the space H and V , respectively.

Let $T > 0$, $v \in H$, $f \in L^2(0, T; V')$, and consider the initial value problem of seeking $u \in L^2(0, T; V) \cap H^1(0, T; V') \hookrightarrow C([0, T]; H)$ such that

$$(1.1) \quad \begin{cases} \partial_t u(t) + Au(t) = f(t) & \text{for all } t \in (0, T], \\ u(0) = v, \end{cases}$$

where $A : V \rightarrow V'$ is a linear, self-adjoint, positive definite operator with a compact inverse. Meanwhile, we assume the following elliptic property:

$$(1.2) \quad \beta_0 \|u\|_V^2 \leq \langle Au, u \rangle \leq \beta_1 \|u\|_V^2 \quad \text{for all } u \in V,$$

*Submitted to the journal's Methods and Algorithms for Scientific Computing section July 27, 2020; accepted for publication (in revised form) July 28, 2021; published electronically November 2, 2021.

<https://doi.org/10.1137/20M1355690>

Funding: The research of the first author was partially supported by the National Natural Science Foundation of China (grant 11901016) and a startup grant from Peking University. The research of the second author was partially supported by a Hong Kong RGC grant (15304420).

[†]School of Mathematical Sciences, Peking University, Beijing, 100871, China (snwu@math.pku.edu.cn).

[‡]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (zhizhou@polyu.edu.hk).

with constants $\beta_1 > \beta_0 > 0$. For example, if we consider a heat equation on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$ and A denotes negative Laplacian $-\Delta$ with homogeneous Dirichlet boundary conditions, then $H = L^2(\Omega)$ and $V = H_0^1(\Omega)$.

In recent years, the development and analysis of parallel algorithms for solving evolution problems have attracted a lot of attention. The first group of parallel schemes is based on the inverse Laplace transform which represents the solution as a contour integral in the complex plane, as well as a carefully designed quadrature rule [36, 48, 55, 60]. Such a method is directly parallelizable and accurate even for nonsmooth problem data. However, this strategy is not directly applicable for the nonlinear problem or anomalous diffusion problems with time-dependent diffusion coefficients. To the second group belongs the widely used parareal algorithm [4, 13, 15, 45, 62], which could be derived as a multigrid-in-time method or a multiple shooting method in the time direction. See also the space-time multigrid method [22, 24, 42, 61, 58]. We refer the interested reader to the survey papers [11, 53] and references therein.

Very recently, in [16], Gander and Wu developed a novel PinT algorithm by applying the waveform relaxation [49, 51] and a diagonalization technique [44, 14, 18]. In particular, they proposed a simple iteration: for given $u_{m-1}(T) \in H$, look for u_m such that

$$(1.3) \quad \begin{cases} \partial_t u_m(t) + Au_m(t) = f(t) & \text{for all } t \in (0, T], \\ u_m(0) = v + \kappa(u_m - u_{m-1})(T). \end{cases}$$

Here κ denotes a relaxation parameter. Note that the exact solution u is a fixed point of the iteration (1.3). It was proved in [16, Theorem 3.1] that by selecting a proper $\kappa \in (0, 1)$, the iteration (1.3) converges with the convergence factor $\kappa e^{-cT}/(1 - \kappa e^{-cT})$, with a constant c depending on the smallest eigenvalue of A .

Then a direct discretization of (1.3) by the backward Euler method immediately results in a periodic-like discrete system, and therefore the diagonalization technique is applicable here to carry out a direct parallel computation. The diagonalization technique was first proposed by Maday and Rønquist for solving evolution models [44]. The basic idea is to reformulate the time stepping system into a space-time all-at-once system, then diagonalize the time stepping matrix and solve all time steps in parallel. The computational cost of each iteration is proved to be $O([MN \log(N) + \widetilde{M}_f N]/p)$ for each processor, where M, N are the number of degrees of freedom in space and time, respectively, \widetilde{M}_f is the computational cost for solving a Poisson-like problem obtained by diagonalization, and p is the number of used processors. In particular, Gander and Wu considered single-step θ -methods for solving (1.3) with uniform step size [16, section 3.2]. The convergence analysis of the discrete system was also established, where the proof is similar to the argument for the continuous problem.

With ε being the desired error tolerance, a k th-order time stepping scheme requires $N = O(\varepsilon^{-1/k})$ time steps. Therefore, the computational complexity for each processor turns out to be $O([\widetilde{M}_f + M] \log(\varepsilon)/k \varepsilon^{-1/k}/p)$. Besides, the roundoff error of the algorithm is proved to be $O(\varepsilon \kappa^{-2} N^2) = O(\varepsilon \kappa^{-2} \varepsilon^{-2/k})$, where ε is the machine precision; see more details in section 3.2. Those facts motivate us to develop and analyze PinT schemes for (1.1) by using some high-order time stepping schemes, such as the k -step backward differentiation formula (BDF k) with $k = 2, 3, \dots, 6$, and the aforementioned waveform relaxation technique. This is beyond the scope of all existing references [12, 16], which only focus on θ -methods, and represents the main theoretical achievements of the work.

Instead of directly discretizing (1.3), we start with the time stepping schemes of (1.1) using BDF k with uniform step size $\tau = T/N$. Then, by perturbing the discrete problem, we obtain a periodic-like system in each iteration, which can be solved parallelly by using $O([MN \log(N) + \widetilde{M}_f N]/p)$ operations (for each processor). We prove that the resulting iteration linearly converges to the exact solution (with a proper choice of the relaxation parameter κ) by using the generating function technique as well as the decay property of the discrete solution operator. Specifically, let U_m^n be the solution of the m th iteration of the perturbed iterative system with the initial guess $U_0^n = v$ for all $0 \leq n \leq N$, and let u be the exact solution to (1.1). Provided certain data regularity exists, we show the error estimate for all $1 \leq n \leq N$ (Theorem 2.7):

$$\|U_m^n - u(t_n)\| \leq c(\gamma(\kappa))^m + \tau^k t_n^{-k}, \quad \text{with } \kappa \in (0, 1) \text{ independent of } \tau,$$

where the positive constant c and the convergence factor (2.18)

$$\gamma(\kappa) = \frac{c\kappa e^{-c_1 T}}{1 - c\kappa e^{-c_1 T}} \in (0, 1)$$

might depend on k, κ, β_0, T, v , and f , but it is independent of τ, n, m , and u . Therefore, to attain the discretization error $O(\tau^k)$ or $O(N^{-k})$, the computational complexity for each processor is $O(\log(N)[MN \log(N) + \widetilde{M}_f N]/p)$.

The above argument could be extended to the subdiffusion model, which involves a time-fractional derivative of order $\alpha \in (0, 1)$. Let $T > 0, v \in H, f \in L^p(0, T; V')$ with $p \in (2/\alpha, \infty)$, and consider the initial value problem of seeking $u \in L^p(0, T; V) \cap H^\alpha(0, T; V')$ such that

$$(1.4) \quad \begin{cases} \partial_t^\alpha u(t) + Au(t) = f(t) & \text{for all } t \in (0, T], \\ u(0) = v. \end{cases}$$

Here $\partial_t^\alpha u$ denotes the left-sided Caputo fractional derivative of order α , defined by

$$(1.5) \quad \partial_t^\alpha u(t) := \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} u'(s) ds.$$

Interest in (1.4) is motivated by its excellent capability of modeling anomalously slow diffusion, such as protein diffusion within cells [19], thermal diffusion in media with fractal geometry [52], and contaminant transport in groundwater [34], to name but a few. The literature on the numerical approximation for the subdiffusion equation (1.4) is vast. The most popular methods include convolution quadrature [5, 8, 10, 28, 31], the collocation-type method [67, 56, 38, 35], the discontinuous Galerkin method [46, 50, 47], and the spectral method [7, 25, 66]; see [29] for an overview of existing schemes. See also [3, 27, 39, 17, 63] for some fast algorithms.

Our argument for linear multistep schemes could be easily applied to many popular time stepping schemes for the subdiffusion problem (1.4). As an example, we consider the convolution quadrature generated by the BDF k method, which was established by Lubich's series of works [40, 41]. Note that the fractional derivative is nonlocal-in-time, and hence its discretization inherits the nonlocality and behaves like a multistep discretization with an infinitely wide stencil. By perturbing the time stepping scheme, we develop an iterative algorithm that requires a periodic-like system to be solved in each iteration, which could be parallelly solved by a diagonalization

technique with $O([MN \log(N) + \widetilde{M}_f N]/p)$ operations for each processor. Moreover, error estimates of the resulting numerical schemes are established by using the decay property of the (discrete) solution operator. We prove the following error estimate (Theorem 3.7):

$$\|U_m^N - u(t_N)\| \leq c(\gamma(\kappa))^m + \tau^k t_n^{-k}, \quad \text{with } \kappa = O(1/\log(N)),$$

where U_m^n is the solution to the iterative algorithm (3.10) with the initial guess $U_0^n = v$ for all $0 \leq n \leq N$, and u is the exact solution to the subdiffusion problem (1.4). In the estimate, the positive constant c and the convergence factor

$$\gamma(\kappa) = \frac{c\kappa \log(N)}{1 - c\kappa \log(N)} \in (0, 1)$$

might depend on α , k , κ , β_0 , T , v , and f , but they are always independent of τ , n , m , and u . The analysis is promising for some nonlinear evolution equations as well as (sub)diffusion problems involving time-dependent diffusion coefficients. See a very recent work of Gu and Wu [21] for a parallel algorithm by using the diagonalization technique, where the analysis only works for BDF2 with $\alpha < 5/8$.

The rest of the paper is organized as follows. In section 2, we develop a high-order time parallel scheme for solving the parabolic problem and analyze its convergence by generating function technique. In section 3, we extend our discussion to the nonlocal-in-time subdiffusion problem. Finally, in section 4, we present some numerical results to illustrate and complement the theoretical analysis.

2. Parallel algorithms for normal diffusion equations. The aim of this section is to propose high-order multistep PinT schemes, with rigorous convergence analysis, for approximately solving the parabolic equation (1.1).

2.1. BDF k scheme for normal diffusion equations. We consider the BDF k scheme, $k = 1, 2, \dots, 6$, with uniform step size. Let $\{t_n = \tau n\}$ be a uniform partition of the interval $[0, T]$, with a time step size $\tau = T/N$. For $n \geq 1$, the k -step BDF scheme seeks $U^n \in V$ such that [37]

$$(2.1) \quad \begin{aligned} \bar{\partial}_\tau U^n + AU^n &= f(t_n) + a_n^{(k)}(f(0) - Av) + \sum_{\ell=1}^{k-2} b_{\ell,n}^{(k)} \tau^\ell \partial_t^\ell f(0) =: \bar{f}_n, \\ U^{-(k-1)} &= \dots = U^{-1} = U^0 = v. \end{aligned}$$

In (2.1) we use the BDF k to approximate the first-order derivative by

$$\bar{\partial}_t U^n := \frac{1}{\tau} \sum_{j=0}^k \omega_j U^{n-j},$$

where constants $\{\omega_j\}$ are coefficients of the polynomials

$$(2.2) \quad \delta_k(\zeta) := \sum_{\ell=1}^k \frac{1}{\ell} (1 - \zeta)^\ell = \sum_{j=0}^k \omega_j \zeta^j.$$

For $\alpha = 1$, the BDF k scheme is known to be $A(\vartheta_k)$ -stable with angle $\vartheta_k = 90^\circ, 90^\circ, 86.03^\circ, 73.35^\circ, 51.84^\circ, 17.84^\circ$ for $k = 1, 2, 3, 4, 5, 6$, respectively [23, p. 251].

To apply the BDF k for parabolic problem, it is well known that one needs starting data $U^j = u(t_j) + O(\tau^k)$ for $0 \leq j \leq k - 1$. Then the error bound of the time stepping scheme is $O(\tau^k)$. However, for nonlocal-in-time subdiffusion models (which will be discussed in section 3), the knowledge of U^j for $0 \leq j \leq k - 1$ does not guarantee an error bound $O(\tau^k)$. It is due to the lack of compatibility of problem data. Fortunately, in the preceding work of the second author and his colleagues, it was proved that one can recover the optimal error bound $O(\tau^k)$ by modifying the starting $k - 1$ steps [31]. The strategy also works for the BDF k for classical parabolic equations [37]. In order to keep consistency with numerical schemes for subdiffusion models, we decide to apply the modified formulation (2.1) of the BDF k .

TABLE 1
The coefficients $a_n^{(k)}$ and $b_{\ell,n}^{(k)}$.

k	$a_1^{(k)}$	$a_2^{(k)}$	$a_3^{(k)}$	$a_4^{(k)}$	$a_5^{(k)}$	ℓ	$b_{\ell,1}^{(k)}$	$b_{\ell,2}^{(k)}$	$b_{\ell,3}^{(k)}$	$b_{\ell,4}^{(k)}$	$b_{\ell,5}^{(k)}$
$k = 2$	$\frac{1}{2}$										
$k = 3$	$\frac{11}{12}$	$-\frac{5}{12}$				$\ell = 1$	$\frac{1}{12}$	0			
$k = 4$	$\frac{31}{24}$	$-\frac{7}{6}$	$\frac{3}{8}$			$\ell = 1$	$\frac{1}{6}$	$-\frac{1}{12}$	0		
						$\ell = 2$	0	0	0		
$k = 5$	$\frac{1181}{720}$	$-\frac{177}{80}$	$\frac{341}{240}$	$-\frac{251}{720}$		$\ell = 1$	$\frac{59}{240}$	$-\frac{29}{120}$	$\frac{19}{240}$	0	
						$\ell = 2$	$\frac{1}{240}$	$-\frac{1}{240}$	0	0	
						$\ell = 3$	$\frac{1}{720}$	0	0	0	
$k = 6$	$\frac{2837}{1440}$	$-\frac{2543}{720}$	$\frac{17}{5}$	$-\frac{1201}{720}$	$\frac{95}{288}$	$\ell = 1$	$\frac{77}{240}$	$-\frac{7}{15}$	$\frac{73}{240}$	$-\frac{3}{40}$	0
						$\ell = 2$	$\frac{1}{96}$	$-\frac{1}{60}$	$\frac{1}{160}$	0	0
						$\ell = 3$	$-\frac{1}{360}$	$\frac{1}{720}$	0	0	0
						$\ell = 4$	0	0	0	0	0

The next lemma provides some properties of the generating function. The proof has been provided in [31, Theorem A.1] and hence is omitted here.

LEMMA 2.1. For any ε , there exists $\theta_\varepsilon \in (\pi/2, \pi)$ such that for any $\theta \in (\pi/2, \theta_\varepsilon)$, there holds $\delta_k(e^{-z}) \in \Sigma_{\pi-\vartheta_k+\varepsilon}$ for any $z \in \Gamma_\theta^{\pi/\sin\theta} = \{z = re^{\pm i\theta}, 0 \leq r \leq \pi/\sin\theta\}$, where $\Sigma_\psi := \{z \in \mathbb{C} : |\arg(z)| \leq \psi\}$. Meanwhile, there exist positive constants c_0 and c'_0 such that

$$c_0|z| \leq |\delta_k(e^{-z})| \leq c'_0|z| \quad \text{for all } z \in \Gamma_\theta^{\pi/\sin\theta}.$$

For $n \geq k$, we choose $a_n^{(k)}$ and $b_{\ell,n}^{(k)}$ to be zero. Then $\bar{\partial}_\tau u(t_n)$ is the standard approximation of $\partial_t u(t_n)$ by BDF k . For $1 \leq n \leq k - 1$, these constants have been determined in [31, 37]; cf. Table 1. In particular, if

$$(2.3) \quad v \in H \quad \text{and} \quad f \in W^{k,1}(0, T; H) \cap C^{k-1}([0, T]; H),$$

the numerical solution to (2.1) satisfies the following error estimate. We omit the proof here and refer interested readers to [37, Theorem 1.1] and [31, Theorem 2.1] for error analysis of (non-self-adjoint) parabolic systems and fractional subdiffusion equations, respectively.

LEMMA 2.2. Assume that the problem data v and f satisfy (2.3). Then

$$(2.4) \quad \|U^n - u(t_n)\| \leq c\tau^k \left(t_n^{-k} \|v\| + \sum_{\ell=0}^{k-1} t_n^{\ell-k+1} \|\partial_t^\ell f(0)\| + \int_0^{t_n} \|\partial_s^k f(s)\| ds \right),$$

where the constant c is independent of τ and n .

2.2. Development of parallel-in-time algorithm. Next, we develop a PinT algorithm for (2.1): for given U_{m-1}^n , $N - k + 1 \leq n \leq N$, we compute U_m^n by

$$(2.5) \quad \begin{aligned} \bar{\partial}_\tau U_m^n + AU_m^n &= \bar{f}_n, & n &= 1, 2, \dots, N, \\ U_m^{-j} &= v + \kappa(U_m^{N-j} - U_{m-1}^{N-j}), & j &= 0, 1, \dots, k-1, \end{aligned}$$

where the revised source term \bar{f}_n is given in (2.1). Note that the exact time stepping solution $\{U^n\}_{n=1}^N$ is a fixed point of this iteration. In section 2.4, we provide a systematic framework to study the iterative algorithm (2.5), which also works for the time-fractional subdiffusion problem discussed later.

We may rewrite the BDF k scheme (2.5) in the following matrix form:

$$(2.6) \quad \frac{1}{\tau}(B_k(\kappa) \otimes I_x)\mathbf{U}_m + (I_t \otimes A)\mathbf{U}_m = \mathbf{F}_{m-1},$$

where $\mathbf{U}_m = (U_m^1, U_m^2, \dots, U_m^N)^T$, $\mathbf{F}_{m-1} = (F_1, F_2, \dots, F_N)^T$ with

$$(2.7) \quad F_n := \bar{f}_n + \frac{\kappa}{\tau} \sum_{j=1}^n \omega_j U_{m-1}^{N-j+1} + \frac{1}{\tau} \sum_{j=0}^{n-1} \omega_j v,$$

and

$$B_k(\kappa) = \begin{bmatrix} \omega_0 & \cdots & 0 & \kappa\omega_k & \cdots & \kappa\omega_2 & \kappa\omega_1 \\ \omega_1 & \omega_0 & & 0 & \kappa\omega_k & \cdots & \kappa\omega_2 \\ \vdots & \vdots & & & & \ddots & \vdots \\ \vdots & \vdots & & \ddots & & 0 & \kappa\omega_k \\ \omega_k & \omega_{k-1} & \cdots & & & & 0 \\ 0 & \omega_k & \omega_{k-1} & \cdots & & & \vdots \\ & & & \ddots & \ddots & & \vdots \\ & & 0 & \omega_k & \cdots & \omega_1 & \omega_0 \end{bmatrix}.$$

Here, we recall that $\omega_j = 0$ if $j > k$ and $\sum_{j=0}^k \omega_j = 0$ for the normal diffusion equations. The following lemma is crucial for the design of PinT algorithm.

LEMMA 2.3 (diagonalization). Let $\Lambda(\kappa) = \text{diag}(1, \kappa^{-\frac{1}{N}}, \dots, \kappa^{-\frac{N-1}{N}})$. Then

$$B_k(\kappa) = \Lambda(\kappa)\tilde{B}_k(\kappa)\Lambda(\kappa)^{-1},$$

where the circular matrix $\tilde{B}_k(\kappa)$ has the form

$$\tilde{B}_k(\kappa) = \begin{bmatrix} \omega_0 & 0 & \kappa^{\frac{k}{N}} \omega_k & \cdots & \kappa^{\frac{2}{N}} \omega_2 & \kappa^{\frac{1}{N}} \omega_1 \\ \kappa^{\frac{1}{N}} \omega_1 & \omega_0 & 0 & \kappa^{\frac{k}{N}} \omega_k & \cdots & \kappa^{\frac{2}{N}} \omega_2 \\ \vdots & \vdots & & & \ddots & \vdots \\ \vdots & \vdots & \ddots & & 0 & \kappa^{\frac{k}{N}} \omega_k \\ \kappa^{\frac{k}{N}} \omega_k & \kappa^{\frac{k-1}{N}} \omega_{k-1} & \cdots & & & 0 \\ 0 & \kappa^{\frac{k}{N}} \omega_k & \kappa^{\frac{k-1}{N}} \omega_{k-1} & \cdots & & \vdots \\ & & & \ddots & \ddots & \vdots \\ & 0 & \kappa^{\frac{k}{N}} \omega_k & \cdots & \kappa^{\frac{1}{N}} \omega_1 & \omega_0 \end{bmatrix}.$$

As a consequence, $B_k(\kappa)$ can be diagonalized by

$$B_k(\kappa) = S(\kappa)D_k(\kappa)S(\kappa)^{-1}, \quad S(\kappa) := \Lambda(\kappa)V,$$

where the Fourier matrix

$$(2.8) \quad V = [v_1, v_2, \dots, v_N], \quad \text{with } v_n = [1, e^{i\frac{2(n-1)\pi}{N}}, \dots, e^{i\frac{2(n-1)(N-1)\pi}{N}}]^T,$$

and $D_k(k)$ is a diagonal matrix.

Using the above lemma, we can solve the system (2.5) in a parallel-in-time manner.

Algorithm 2.1 PinT algorithm by diagonalization technique for diffusion equation.

- 1: Solve $(S(\kappa) \otimes I_x)\mathbf{H} = \mathbf{F}_{m-1}$.
 - 2: Solve $(D_k(\kappa) \otimes I_x + \tau I_t \otimes A)\mathbf{Q} = \tau\mathbf{H}$.
 - 3: Solve $(S(\kappa)^{-1} \otimes I_x)\mathbf{U}_m = \mathbf{Q}$.
-

It is known that the circulant matrix can be diagonalized by the FFT with $\mathcal{O}(N \log N)$ operations [20, Chapter 4.7.7]. Then in each iteration, it turns out to be N independent Poisson-like equations, which can be efficiently solved by, for instance, the multigrid method.

Speedup analysis of Algorithm 2.1. Let M_f be the total real floating point operations for solving the (elliptic) Poisson-like equations. Then the cost for the serial computation is $\mathcal{O}(M_f N)$. It is known that with (optimal) multigrid method M_f is proportional to M , the number of degrees of freedom in space.

Consider the parallelization of Algorithm 2.1 with p processors. Steps 1 and 3 can be finished with total computational cost $\mathcal{O}([MN \log(N)]/p)$ by using the bulk synchronous parallel FFT algorithm [26]; see also [16, section 4.1] for the detailed analysis. We note here that the \mathbf{F} in (2.7) needs to be updated via \mathbf{U}_{m-1} , whose computational cost is $\mathcal{O}(M)$ since $\omega_j = 0$ when $j > k$. The PinT algorithm of \mathbf{F} will be more delicate for the time-fractional subdiffusion problem discussed later.

For step 2, the total computational cost is $\mathcal{O}(\tilde{M}_f N/p)$, where \tilde{M}_f denotes total real floating point operations for solving the Poisson-like equations obtained by diagonalization, and hence the cost for the parallel computation is $\mathcal{O}([MN \log(N) + \tilde{M}_f N]/p)$. In the case $p = \mathcal{O}(N)$, then the parallel computational cost reduces to $\mathcal{O}(M \log(N) + \tilde{M}_f)$. In some cases, \tilde{M}_f could be (almost) linear to M even in higher

dimensions. For example, if we consider the heat equation with periodic boundary conditions, we can apply the FFT to solve the Poisson-like equations with computational complexity $\widetilde{M}_f = O(M \log M)$.

With ε being the desired error tolerance, a k th-order time stepping scheme requires $N = O(\varepsilon^{-1/k})$ time steps. Then, in order to attain sufficient accuracy $O(\varepsilon)$, the number of iterations should be $O(\log N)$, because the convergence factor $\gamma(\kappa) \in (0, 1)$ is independent of N (cf. Theorem 2.7). Therefore, the total computational cost is $O([MN(\log N)^2 + \widetilde{M}_f N \log N]/p)$.

REMARK 2.1. *The proposed algorithm uses the FFT to diagonalize over time. It transforms the parabolic PDE into a decoupled set of elliptic PDEs. Such a technique is quite common for solving time-periodic problems, but relatively new for solving initial value problems.*

In this paper, we only discuss the parallelism in the time direction. Nevertheless, combining the proposed method with some parallel-in-space algorithms, one may obtain an algorithm with polylog parallel complexity if $O(MN)$ processors were available. For example, for parabolic equations with periodic boundary conditions, one may apply the FFT to transform the parabolic PDE into a decoupled set of ODEs:

$$u'_j(t) + \lambda_j u_j(t) = f_j(t), \quad \text{with } u_j(0) = v_j,$$

where $1 \leq j \leq M$ and M denotes the number of degrees of freedom in space. Then these decoupled ODEs could be efficiently solved by using the PinT algorithm proposed in this paper.

Roundoff error of Algorithm 2.1. Let \mathbf{U}_m be the exact solution of (2.6), and let $\widehat{\mathbf{U}}_m$ be the solution of Algorithm 2.1. We assume that step 2 of Algorithm 2.1 is solved in a direct manner (for example, the LU factorization). Then, for simplicity, we consider an arbitrary eigenvalue of the matrix A and analyze the relative roundoff error. To this end, we replace matrices I_x and A by scalars 1 and μ . Then the system (2.6) reduces to

$$B\mathbf{U}_m = F, \quad \text{with } B = \frac{1}{\tau} B_k(\kappa) + \mu I_t.$$

Then by Lemma 2.3 we define

$$D = S(\kappa)^{-1} B S(\kappa), \quad \text{and hence } D = \frac{1}{\tau} D_k(\kappa) + \mu I_t.$$

Note that to solve $B\mathbf{U}_m = F$ by diagonalization is equivalent to solving $(B + \delta B)\widehat{\mathbf{U}}_m = F$ with some perturbation δB , which can be easily bounded by [16, p. 496]

$$\|\delta B\|_2 \leq \epsilon(2N + 1)\|S(\kappa)\|_2\|S(\kappa)^{-1}\|_2\|D\|_2 + O(\epsilon^2),$$

where ϵ denotes the machine precision ($\epsilon = 2.2204 \times 10^{-16}$ for a 32-bit computer). Then the roundoff error satisfies

$$\begin{aligned} \frac{\|\mathbf{U}_m - \widehat{\mathbf{U}}_m\|_2}{\|\mathbf{U}_m\|_2} &\leq \text{cond}_2(B) \frac{\|\delta B\|_2}{\|B\|_2} \leq \epsilon(2N + 1)\|S(\kappa)\|_2\|S(\kappa)^{-1}\|_2\|D\|_2\|B^{-1}\|_2 \\ &\leq \epsilon(2N + 1) \text{cond}_2(S(\kappa))^2 \text{cond}_2(D). \end{aligned}$$

Here we note that for $\kappa \in (0, 1]$

$$\|S(\kappa)\|_2 \leq \|\Lambda(\kappa)\|_2\|V\|_2 \leq \kappa^{-\frac{N-1}{N}}\sqrt{N}$$

and

$$\|S(\kappa)^{-1}\|_2 \leq \|\Lambda(\kappa)^{-1}\|_2 \|V^{-1}\|_2 \leq \frac{1}{\sqrt{N}}.$$

Therefore, we arrive at

$$\text{cond}_2(S(\kappa)) = \|S(\kappa)\|_2 \|S(\kappa)^{-1}\|_2 \leq \kappa^{-\frac{N-1}{N}} \leq \kappa^{-1}.$$

For the diagonal matrix D

$$\|D^{-1}\|_2 = \max_{1 \leq n \leq N} \left| \frac{1}{\tau} \delta_k(\kappa^{\frac{1}{N}} e^{-i\frac{2(n-1)\pi}{N}}) + \mu \right|^{-1} \leq (\sin \theta_k \mu)^{-1}.$$

Here we apply $A(\theta_k)$ stability of the BDF k scheme with $\theta_k \in (0, \pi/2)$. Similarly, using the definition of generating function (2.2), we derive for any $\kappa \in (0, 1)$ and $N > 1$

$$(2.9) \quad \|D\|_2 = \max_{1 \leq n \leq N} \left| \frac{1}{\tau} \delta_k(\kappa^{\frac{1}{N}} e^{-i\frac{2(n-1)\pi}{N}}) + \mu \right| \leq \frac{1}{\tau} \delta_k(-\kappa^{\frac{1}{N}}) + \mu \leq \frac{1}{\tau} \delta_k(-\kappa^{\frac{1}{N}}) + \mu.$$

Hence, we obtain

$$\text{cond}_2(D) \leq \left(\frac{\delta_k(-1)}{\tau} + \mu \right) (\mu \sin \theta_k)^{-1}.$$

As a result, we have for $\mu \in [\mu_0, \infty)$ (where the positive number μ_0 depends on β_0 in (1.2))

$$(2.10) \quad \frac{\|\mathbf{U}_m - \widehat{\mathbf{U}}_m\|_2}{\|\mathbf{U}_m\|_2} \leq \epsilon(2N + 1)\kappa^{-2} \left(\frac{\delta_k(-1)}{\tau} + \mu \right) (\mu \sin \theta_k)^{-1} \leq C_k \epsilon \kappa^{-2} N^2,$$

where the constant C_k can be written as

$$C_k = 3 \left(1 + \left[\frac{\delta_k(-1)}{\tau} \right] / \mu_0 \right) (\sin \theta_k)^{-1}.$$

It only depends on the order of the BDF method and β_0 in (1.2). Note that the bound of roundoff error is uniform for $\mu \rightarrow \infty$, and therefore it holds for all self-adjoint operators A satisfying (1.2).

REMARK 2.2. *The above analysis shows a reasonable estimate that the roundoff error is $O(\epsilon \kappa^{-2} N^2)$. See a similar estimate for some A -stable single-step methods like the backward Euler scheme or Crank–Nicolson scheme [16]. The difference is that the BDF k scheme (with $k > 2$) is no longer A -stable. In our numerical experiments, we indeed observe that the roundoff error increases as $N \rightarrow \infty$.*

2.3. Representation of numerical solution. The aim of this section is to develop the representation of the numerical solution of the k -step BDF schemes through a contour integral in the complex domain, and to establish decaying properties of the solution operators.

By letting $W^n = U^n - v$, we can reformulate the time stepping scheme (2.1) as

$$(2.11) \quad \begin{aligned} \bar{\partial}_\tau W^n + AW^n &= -Av + \bar{f}_n, \quad n = 1, 2, \dots, N, \\ W^{-(k-1)} &= \dots = W^{-1} = W^0 = 0. \end{aligned}$$

By multiplying ξ^n on (2.11) and taking summation over n (we extend n in (2.11) to infinity in the sense that $\bar{f}_n = 0$ with $n \geq N$), we have

$$\sum_{n=1}^{\infty} \xi^n \bar{\partial}_t W^n + \sum_{n=1}^{\infty} \xi^n A W^n = - \sum_{n=1}^{\infty} \xi^n A v + \bar{f}_n \xi^n.$$

For any given sequence $\{V^n\}_{n=0}^{\infty}$, let $\tilde{V}(\xi) := \sum_{n=0}^{\infty} V^n \xi^n$ denote its generating function. Since $W^{-(k-1)} = \dots = W^{-1} = W^0 = 0$, according to properties of discrete convolution, we have the identity

$$\sum_{n=1}^{\infty} \xi^n \bar{\partial}_\tau V_n = \frac{\delta_k(\xi)}{\tau} \tilde{V}(\xi),$$

where $\delta_k(\xi)$ denotes the generating function of the k -step BDF method (2.2). Therefore we conclude that

$$\left(\frac{\delta_k(\xi)}{\tau} + A\right) \tilde{W}(\xi) = -\left(\frac{\xi}{1-\xi}\right) A v + \tilde{f}_n(\xi),$$

which implies that

$$\tilde{W}(\xi) = \left(\frac{\delta_k(\xi)}{\tau} + A\right)^{-1} \left[-\left(\frac{\xi}{1-\xi}\right) A v + \tilde{f}_n(\xi)\right].$$

It is easy to see that $\tilde{W}(\xi)$ is analytic with respect to ξ in the circle $|\xi| = \rho$, for $\rho > 0$ small, on the complex plane; then with Cauchy's integral formula, we have the following expression:

$$\begin{aligned} W^n &= \frac{1}{2\pi i} \int_{|\xi|=\rho} \xi^{-n-1} \tilde{W}(\xi) d\xi \\ &= \frac{\tau}{2\pi i} \int_{|\xi|=\rho} \xi^{-n-1} \left(\delta_k(\xi) + \tau A\right)^{-1} \left[-\left(\frac{\xi}{1-\xi}\right) A v + \tilde{f}_n(\xi)\right] d\xi. \end{aligned}$$

Therefore we obtain the solution representation

$$(2.12) \quad U^n = (I + F_\tau^n) v + \tau \sum_{j=1}^n E_\tau^{n-j} \bar{f}_j,$$

where the discrete operators F_τ^n and E_τ^n are, respectively, defined by

$$(2.13) \quad \begin{aligned} F_\tau^n &= -\frac{1}{2\pi i} \int_{|\xi|=\rho} \frac{1}{\xi^n (1-\xi)} \left(\delta_k(\xi)/\tau + A\right)^{-1} A d\xi, \\ E_\tau^n &= \frac{1}{2\pi i} \int_{|\xi|=\rho} \xi^{-n-1} \left(\delta_k(\xi)/\tau + A\right)^{-1} d\xi. \end{aligned}$$

Now we recall a useful estimate (cf. [57, Lemma 10.3]). For $k = 1, \dots, 6$, there are positive constants c, C , and λ_0 (depending only on the BDF k method) such that

$$(2.14) \quad \left| \frac{1}{2\pi i} \int_{|\xi|=\rho} \xi^{-n-1} (\delta_k(\xi) + \lambda)^{-1} d\xi \right| \leq \begin{cases} C e^{-cn\lambda}, & 0 < \lambda \leq \lambda_0, \\ C \lambda^{-1} e^{-cn}, & \lambda > \lambda_0. \end{cases}$$

This together with the coercivity property (1.2) immediately implies the following lemma.

LEMMA 2.4. Let E_τ^n be the discrete operator defined in (2.13). Then

$$\|E_\tau^n\|_{H \rightarrow H} \leq c_2 e^{-c_1 t_n}.$$

Here the generic positive constants c_1 and c_2 are independent of n and τ .

2.4. Convergence analysis. In this section, we analyze the convergence of the iterative scheme (2.5), or, equivalently,

$$(2.15) \quad \begin{aligned} \bar{\partial}_\tau U_m^n + AU_m^n &= \bar{f}_n - \frac{\kappa}{\tau} G_m^n, & n = 1, 2, \dots, N, \\ U_m^{-j} &= v, & j = 0, 1, \dots, k-1, \end{aligned}$$

where the term G_m^n is given by

$$(2.16) \quad G_m^n = \sum_{j=n}^k \omega_j \left(U_m^{N+n-j} - U_{m-1}^{N+n-j} \right).$$

Here, the summation is assumed to vanish if the lower bound is greater than the upper bound. We aim to show that U_m^N converges to U^N , the solution of the time stepping scheme (2.1), as $m \rightarrow \infty$.

LEMMA 2.5. Let U_m^n be the solution to the iterative scheme (2.5) with $v = 0$ and $\bar{f}_n = 0$ for all $n = 1, 2, \dots, N$. Then we can choose a proper parameter $\kappa > 0$ in (2.5), which is independent of step size τ , such that the following estimate is valid:

$$\sum_{j=0}^{k-1} \|U_m^{N-j}\| \leq \gamma(\kappa) \sum_{j=0}^{k-1} \|U_{m-1}^{N-j}\|.$$

Here $\gamma(\kappa) \in (0, 1)$ is constant depending on κ , β_0 , and T , but independent of τ .

Proof. Following the preceding argument in section 2.3, U_m^n could be represented by

$$(2.17) \quad \begin{aligned} U_m^n &= -\kappa \sum_{i=1}^n E_\tau^{n-i} G_m^i \\ &= -\kappa \sum_{i=1}^{\min\{k,n\}} E_\tau^{n-i} \sum_{j=i}^k \omega_j (U_m^{N+i-j} - U_{m-1}^{N+i-j}), & n = 1, 2, \dots, N. \end{aligned}$$

Now we take the H norm in (2.17) and apply Lemma 2.4 to obtain

$$\sum_{j=0}^{k-1} \|U_m^{N-j}\| \leq c\kappa e^{-c_1 T} \sum_{j=0}^{k-1} \|U_m^{N-j} - U_{m-1}^{N-j}\|,$$

where c is a generic constant and c_1 is the constant in Lemma 2.4. Then we apply the triangle inequality by choosing κ small enough such that $c\kappa e^{-c_1 T} < 1$, and hence we derive that

$$\sum_{j=0}^{k-1} \|U_m^{N-j}\| \leq \frac{c\kappa e^{-c_1 T}}{1 - c\kappa e^{-c_1 T}} \sum_{j=0}^{k-1} \|U_{m-1}^{N-j}\|.$$

Finally, we define the convergence factor

$$(2.18) \quad \gamma(\kappa) := \frac{c\kappa e^{-c_1 T}}{1 - c\kappa e^{-c_1 T}}.$$

Then by choosing κ sufficiently small such that $c\kappa e^{-c_1 T} \in (0, 1/2)$, we have $\gamma(\kappa) \in (0, 1)$. This completes the proof of the desired assertion. \square

COROLLARY 2.6. *Let U_m^n be the solution to the iterative scheme (2.5), and let U^n be the solution to the k -step BDF scheme (2.1). Then we can choose a proper parameter $\kappa > 0$ in (2.5), which is independent of step size τ , such that the following estimate is valid:*

$$\sum_{j=0}^{k-1} \|U^{N-j} - U_m^{N-j}\| \leq \gamma(\kappa) \sum_{j=0}^{k-1} \|U^{N-j} - U_{m-1}^{N-j}\| \quad \text{for all } m \geq 1.$$

Here $\gamma(\kappa) \in (0, 1)$, is constant depending on κ , β_0 , and T , but independent of τ .

Proof. To this end, we let $e_m^n = (U_m^n - U^n)$ and note that the time stepping solution $\{U^n\}_{n=1}^N$ is the fixed point of the iteration (2.5). Therefore e_m^n satisfies

$$\begin{aligned} \bar{\partial}_\tau e_m^n + A e_m^n &= -\frac{\kappa}{\tau} K_m^n, \quad n = 1, 2, \dots, N, \\ e_m^{-j} &= 0, \quad j = 0, 1, \dots, k-1, \end{aligned}$$

where the term K_m^n is given by

$$K_m^n = \sum_{j=n}^k \omega_j (e_m^{N+n-j} - e_{m-1}^{N+n-j}).$$

Then the convergence estimate follows immediately from Theorem 2.7. \square

Combining Corollary 2.6 with the estimate (2.4), we have the following error estimate of the iterative scheme (2.5).

THEOREM 2.7. *Suppose that the assumptions (1.2) and (2.3) are valid. Let U_m^n be the solution to the iterative scheme (2.5) with the initial guess $U_0^n = v$ for all $0 \leq n \leq N$, and let u be the exact solution to the parabolic equation (1.1). Then by choosing proper relaxation parameter $\kappa \in (0, 1)$ which is independent of step size τ , the following estimate is valid:*

$$\|U_m^n - u(t_n)\| \leq c(\gamma(\kappa))^m + \tau^k t_n^{-k} \quad \text{for all } n = 1, 2, \dots, N.$$

Here constants $\gamma(\kappa) \in (0, 1)$ and $c > 0$ might depend on k , κ , β_0 , T , v , and f , but they are independent of m , n , τ , and u .

Proof. We split the error into two parts:

$$U_m^n - u(t_n) = (U_m^n - U^n) + (U^n - u(t_n)),$$

where U^n is the solution to the k -step BDF scheme (2.1). Note that the second term has the error bound (2.4). Meanwhile, via (2.6), we have the estimate

$$\sum_{j=0}^{k-1} \|U^{N-j} - U_m^{N-j}\| \leq \gamma(\kappa)^m \sum_{j=0}^{k-1} \|U^{N-j} - v\| \leq \gamma(\kappa)^m \|v\| + \gamma(\kappa)^m \sum_{j=0}^{k-1} \|U^{N-j}\|.$$

By the error estimate (2.4) and the assumption of data regularity (2.3), we obtain that

$$\sum_{j=0}^{k-1} \|U^{N-j}\| \leq c_T.$$

This, (2.17), and Lemma 2.4 lead to the estimate that

$$\|U^n - U_m^n\| \leq c\gamma(\kappa) \left(\sum_{j=0}^{k-1} \|U^{N-j} - U_m^{N-j}\| + \sum_{j=0}^{k-1} \|U^{N-j} - U_{m-1}^{N-j}\| \right) \leq c\gamma(\kappa)^m.$$

Then we obtain the desired result. □

REMARK 2.3. *For the backward Euler method (BDF1), the convergence rate was proved to be [16]*

$$\gamma(\kappa) = \frac{\kappa e^{-c_1 T}}{1 - \kappa e^{-c_1 T}} < \frac{\kappa}{1 - \kappa}.$$

So the iterative algorithm converges linearly by choosing $\kappa < 1/2$, and the smaller parameter κ leads to the faster convergence. However, in section 2.2, we have shown that the roundoff error is proportional to $O(\kappa^{-2})$, so a tiny κ may lead to a disastrous roundoff error. Therefore one needs to choose a proper $\kappa \in (0, 1/2)$ in order to balance the roundoff error and the convergence rate.

For k -step BDF methods with $1 < k \leq 6$, we obtain a similar result,

$$\gamma(\kappa) = \frac{c\kappa e^{-c_1 T}}{1 - c\kappa e^{-c_1 T}} \leq \frac{c\kappa}{1 - c\kappa},$$

with an extra factor $c > 1$. This is due to the different stability estimate in Lemma 2.4 of linear multistep methods. Even though it is hard to derive an explicit bound of the generic constant c for k -step BDF methods, our empirical experiments show that the choice $\kappa \approx 0.1$ leads to an acceptable roundoff error ($\approx 10^{-12}$), and meanwhile the convergence is very fast (see Figure 1(b)). Note that the convergence rate is independent of N , so the increase in the total number of steps will not affect the robust convergence.

3. Parallel algorithms for nonlocal-in-time subdiffusion equations. In the section, we consider the subdiffusion equations (1.4), which involve a fractional-in-time derivative of order $\alpha \in (0, 1)$. The fractional-order differential operator is nonlocal, and its discretization inherits this nonlocality and looks like a multistep discretization with an infinitely wide stencil. This motivates us to extend the argument established in section 2 to the subdiffusion equations (1.4).

3.1. BDF k scheme for subdiffusion equations. To begin with, we discuss the development of a PinT algorithm for (1.4). We apply the convolution quadrature (CQ) to discretize the fractional derivative on uniform grids. Following the same setting in section 2.2, let $\{t_n = n\tau\}_{n=0}^N$ be a uniform partition of the time interval $[0, T]$, with a time step size $\tau = T/N$.

CQ was first proposed by Lubich [40, 41] for discretizing Volterra integral equations. This approach provides a systematic framework to construct high-order numerical methods to discretize fractional derivatives and has been the foundation of many early works. Specifically, CQ approximates the Riemann–Liouville derivative

${}^R\partial_t^\alpha \varphi(t_n)$ with $\alpha \in (0, 1)$, which is defined by

$${}^R\partial_t^\alpha \varphi := \frac{d}{dt} \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} \varphi(s) ds$$

(with $\varphi(0) = 0$) by a discrete convolution (with the shorthand notation $\varphi^n = \varphi(t_n)$)

$$(3.1) \quad \hat{\partial}_\tau^\alpha \varphi^n := \frac{1}{\tau^\alpha} \sum_{j=0}^n \omega_j^{(\alpha)} \varphi^{n-j}.$$

Here we consider the BDF k method, for example; then the weights $\{\omega_j^{(\alpha)}\}_{j=0}^\infty$ are the coefficients in the power series expansion

$$(3.2) \quad \delta_k(\xi)^\alpha = \sum_{j=0}^\infty \omega_j^{(\alpha)} \xi^j,$$

where $\delta_k(\xi)$ is given by (2.2). Generally, the weights $\{\omega_j^{(\alpha)}\}_{j=0}^\infty$ can be computed either by the FFT or recursion [54].

The next lemma provides a useful bound of the coefficients $\omega_j^{(\alpha)}$.

LEMMA 3.1. *The weights $\omega_n^{(\alpha)}$ satisfy the estimate that $|\omega_n^{(\alpha)}| \leq c(n+1)^{-\alpha-1}$, where the constant c only depends on α and k .*

Proof. The case of $k = 1$ has been proved in [32, Lemma 12] by using the expression of the coefficients: $\omega_n^{(\alpha)} = -\prod_{j=1}^n (1 - \frac{1+\alpha}{j})$. However, the closed forms of coefficients of high-order schemes are not available. Here we provide a systematic proof for all BDF k methods, $k = 1, 2, \dots, 6$.

By the definition of $\{\omega_j^{(\alpha)}\}$ and Cauchy’s integral formula, we obtain that

$$\omega_j^{(\alpha)} = \frac{1}{2\pi i} \int_{|\xi|=1} \delta_k(\xi)^\alpha \xi^{-n-1} d\xi = \frac{\tau^{1+\alpha}}{2\pi i} \int_{\Gamma_\tau} e^{zt_n} (\delta_k(e^{-z\tau})/\tau)^\alpha dz,$$

where $\Gamma^\tau := \{z = iy : y \in \mathbb{R}, |y| \leq \pi/\tau\}$. The analyticity together with the periodicity of the integrand allows the deformation of the contour to

$$\Gamma_\psi^\tau = \{z = re^{\pm i\psi} : 0 \leq r \leq \pi/\tau \sin \psi\},$$

with $\psi \in (\pi/2, \pi)$. Then Lemma 2.1 implies for $n \geq 1$

$$\begin{aligned} |\omega_j^{(\alpha)}| &\leq c\tau^{1+\alpha} \int_{\Gamma_\psi^\tau} |e^{zt_n}| |\delta_k(e^{z\tau})|^\alpha |dz| \leq c\tau^{1+\alpha} \int_0^{\frac{\pi}{\tau \sin \psi}} e^{r t_n \cos \psi} r^\alpha dr \\ &\leq c\tau^{1+\alpha} t_n^{-\alpha-1} \leq cn^{-\alpha-1} \leq 4c(n+1)^{-\alpha-1}. \end{aligned}$$

This together with the uniform bound of $\omega_0^{(\alpha)}$ leads to the desired result. □

Using the relation $\partial_t^\alpha \varphi(t) = {}^R\partial_t^\alpha (\varphi - \varphi(0))$ (see, e.g., [33, p. 91]), the subdiffusion problem could be rewritten in the form

$${}^R\partial_t^\alpha (u - v) + Au = f.$$

Then the time stepping scheme based on the CQ for problem (1.4) is to seek approximations U^n to the exact solution $u(t_n)$ by

$$(3.3) \quad \hat{\partial}_\tau^\alpha(U^n - v) + AU^n = f(t_n), \quad n = 1, \dots, N.$$

By the definition of the discretized operator $\hat{\partial}_\tau^\alpha$ in (3.1), we have

$$(3.4) \quad \hat{\partial}_\tau^\alpha(U^n - v) = \frac{1}{\tau^\alpha} \sum_{j=0}^n \omega_j^{(\alpha)}(U^{n-j} - v) = \frac{1}{\tau^\alpha} \sum_{j=0}^\infty \omega_j^{(\alpha)}U^{n-j} =: \bar{\partial}_\tau^\alpha U^n$$

by setting the historical initial data

$$(3.5) \quad U^n = v \quad \text{for all } n \leq 0.$$

Then we reformulate the time stepping scheme (3.3)–(3.5) by

$$(3.6) \quad \begin{aligned} \bar{\partial}_\tau^\alpha U^n + AU^n &= f(t_n), \quad n = 1, 2, \dots, N, \\ U^n &= v, \quad n \leq 0. \end{aligned}$$

If the exact solution u is smooth and has sufficiently many vanishing derivatives at $t = 0$, then the approximation U^n converges at a rate of $O(\tau^k)$ uniformly in time t [41, Theorem 3.1]. However, it generally only exhibits a first-order accuracy when solving fractional evolution equations even for smooth v and f [8, 31], because the requisite compatibility conditions

$$Av + f(0) = 0, \quad \text{and} \quad \partial_t^\ell f(0) = 0 \quad \text{for all } \ell = 1, 2, \dots, k$$

are usually not satisfied. This loss of accuracy is one distinct feature for most time stepping schemes deriving under the assumption that the solution u is sufficiently smooth.

In order to restore the high-order convergence rate, we simply modify the starting steps [8, 31, 43, 64]. In particular, for $n \geq 1$, the CQ-BDF k scheme seeks $U^n \in V$ such that

$$(3.7) \quad \begin{aligned} \bar{\partial}_\tau^\alpha U^n + AU^n &= f(t_n) + a_n^{(k)}(f(0) - Av) + \sum_{\ell=1}^{k-2} b_{\ell,n}^{(k)} \tau^\ell \partial_t^\ell f(0) =: \bar{f}_n, \\ U^n &= v, \quad n \leq 0. \end{aligned}$$

The coefficients $a_n^{(k)}$ and $b_{\ell,n}^{(k)}$ have to be chosen appropriately (cf. Table 1). For $n \geq k$, $a_n^{(k)}$ and $b_{\ell,n}^{(k)}$ are zero; then $\bar{\partial}_\tau^\alpha u(t_n)$ is the standard CQ-BDF k scheme that approximates $\partial_t^\alpha u(t_n)$. Then there holds the following error estimate [31, Theorem 2.1].

LEMMA 3.2. *If the initial data v and forcing data f satisfy*

$$(3.8) \quad v \in H \quad \text{and} \quad f \in W^{k, \frac{1}{\alpha} + \epsilon}(0, T; H) \quad \text{with some } \epsilon > 0,$$

then the time stepping solution U^n to (3.7) satisfies the following error estimate:

$$(3.9) \quad \|U^n - u(t_n)\| \leq c \tau^k \left(t_n^{-k} \|v\| + \sum_{\ell=0}^{k-1} t_n^{\alpha + \ell - k} \|\partial_t^\ell f(0)\| + \int_0^{t_n} (t-s)^{\alpha-1} \|\partial_s^k f(s)\| ds \right),$$

where the constant c is independent of τ and t_n .

3.2. Development of parallel-in-time scheme. In order to develop a parallel solver for the time stepping method (3.7), we apply the strategy developed in section 2. For given U_{m-1}^n , with $1 \leq n \leq N$, we compute U_m^n by

$$(3.10) \quad \begin{aligned} \bar{\partial}_\tau^\alpha U_m^n + AU_m^n &= \bar{f}_n, & n = 1, 2, \dots, N, \\ U_m^{-n} &= v + \kappa(U_m^{N-n} - U_{m-1}^{N-n}), & n = 0, 1, \dots, N-1, \\ U_m^n &= v, & n \leq -N, \end{aligned}$$

where the revised source term \bar{f}_n is given in (3.7). Note that $\{U^n\}_{n=1}^N$, the exact time stepping solution to (3.7), is a fixed point of this iteration. We shall examine convergence in section 3.3.

Now we may rewrite the perturbed BDF k scheme (3.10) in the following matrix form:

$$(3.11) \quad \frac{1}{\tau^\alpha} (B_k(\kappa) \otimes I_x) \mathbf{U}_m + (I_t \otimes A) \mathbf{U}_m = \mathbf{F}_{m-1},$$

where $\mathbf{U}_m = (U_m^1, U_m^2, \dots, U_m^N)^T$, $\mathbf{F}_{m-1} = (F_1, F_2, \dots, F_N)^T$ with

$$(3.12) \quad F_n = \bar{f}_n + \frac{\kappa}{\tau^\alpha} \sum_{j=n}^{N-1} \omega_j U_{m-1}^{N+n-j} + \frac{1}{\tau^\alpha} \sum_{j=0}^{n-1} \omega_j v,$$

and

$$B_k(\kappa) = \begin{bmatrix} \omega_0 & \kappa\omega_{N-1} & \cdots & \kappa\omega_2 & \kappa\omega_1 \\ \omega_1 & \omega_0 & \cdots & \kappa\omega_3 & \kappa\omega_2 \\ \omega_2 & \omega_1 & \ddots & & \vdots \\ \vdots & \vdots & & \omega_0 & \kappa\omega_{N-1} \\ \omega_{N-1} & \omega_{N-2} & \cdots & \omega_1 & \omega_0 \end{bmatrix}.$$

Similarly to Lemma 2.3, we have the following result.

LEMMA 3.3 (diagonalization). *Let $\Lambda(\kappa) = \text{diag}(1, \kappa^{-\frac{1}{N}}, \dots, \kappa^{-\frac{N-1}{N}})$. Then*

$$B_k(\kappa) = S(\kappa) D_k(\kappa) S(\kappa)^{-1}, \quad S(\kappa) = \Lambda(\kappa) V,$$

where V is the Fourier matrix defined in (2.8).

The above lemma implies the parallel solver for (3.10).

Algorithm 3.1 PinT BDF k scheme for subdiffusion.

- 1: Solve $(S(\kappa) \otimes I_x) \mathbf{H} = \mathbf{F}_{m-1}$.
 - 2: Solve $(D_k(\kappa) \otimes I_x + \tau^\alpha I_t \otimes A) \mathbf{Q} = \tau^\alpha \mathbf{H}$.
 - 3: Solve $(S(\kappa)^{-1} \otimes I_x) \mathbf{U}_m = \mathbf{Q}$.
-

Speedup analysis of Algorithm 3.1. Due to the nonlocality of the fractional-order differential operator, the discretized operator (3.4) requires the information of all the previous steps. In particular, in the n th step of CQ-BDF k scheme, we need to solve a Poisson-like problem:

$$(\omega_0^{(\alpha)} I + \tau^\alpha A) U^n = \sum_{j=1}^n \omega_j^{(\alpha)} (U^0 - U^{n-j}) + \omega_0^{(\alpha)} U^0 + \bar{f}^n.$$

The computation cost of this step is $O(nM + M_f)$. Then, taking summation over n from 1 to N , we derive that the total computational cost of the direct implementation of the CQ-BDF k scheme is $O(MN^2 + M_f N)$.

Consider the parallelization of Algorithm 3.1 with p used processors. Similar to the discussion on Algorithm 2.1, the cost of the parallel FFT in steps 1 and 3 is $O([MN \log(N)]/p)$. We check the computational cost of \mathbf{F}_{m-1} in step 1, which contains the following three components:

1. The source term \bar{f}_n defined in (3.7): The correction is taken at the first few steps and hence the computational cost is $O((MN)/p)$.
2. The convolution term $\sum_{j=n}^{N-1} \omega_j U_{m-1}^{N+n-j}$ can be rewritten as the n th entry of

$$\begin{bmatrix} \omega_N & \omega_{N-1} & \cdots & \omega_2 & \omega_1 \\ 0 & \omega_N & \cdots & \omega_3 & \omega_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \omega_N & \omega_{N-1} \\ 0 & 0 & \cdots & 0 & \omega_N \end{bmatrix} \begin{bmatrix} U_{m-1}^1 \\ U_{m-1}^2 \\ \vdots \\ U_{m-1}^{N-1} \\ U_{m-1}^N \end{bmatrix} := \mathbf{W} \mathbf{U}_{m-1}.$$

Although \mathbf{W} is not circulant, the above matrix can be extended to be circulant for the purpose of using the FFT algorithm. More precisely, consider

$$\begin{bmatrix} \mathbf{Z} & \mathbf{W} \\ \mathbf{W} & \mathbf{Z} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{U}_{m-1} \end{bmatrix}, \quad \text{with } \mathbf{Z} := \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \omega_1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \omega_{N-2} & \omega_{N-3} & \cdots & 0 & 0 \\ \omega_{N-1} & \omega_{N-2} & \cdots & \omega_1 & 0 \end{bmatrix}.$$

It can be easily seen that the extended matrix is circulant. Thanks again to the bulk synchronous parallel FFT algorithm [26], the FFT of the extended system and $[\mathbf{0}, \mathbf{U}_{m-1}]^T$ lead to the computational costs $O([N \log(N)]/p)$ and $O([MN \log(N)]/p)$, respectively. By using the inverse FFT, the computational cost of the convolution term $\sum_{j=n}^{N-1} \omega_j U_{m-1}^{N+n-j}$ turns out to be $O([MN \log(N)]/p)$.

3. The convolution term $\sum_{j=0}^{n-1} \omega_j v$ can be computed with the cost $O([N \log(N) + MN]/p)$.

For step 2, the total computational cost is $O([\widetilde{M}_f N]/p)$. To sum up, the overall cost for the parallel computation is $O([MN \log(N) + \widetilde{M}_f N]/p)$, which becomes $O(M \log N + \widetilde{M}_f)$ if $p = O(N)$.

Similar to the discussion on Algorithm 2.1, in order to attain the desired accuracy, the total computational cost is $O([MN(\log N)^2 + \widetilde{M}_f N \log N]/p)$ for each processor.

Roundoff error of Algorithm 3.1. In case that $I_x = 1$ and $A = \mu$, by using the same argument as in section 2.2, the roundoff error can be bounded by

$$\frac{\|\mathbf{U}_m - \widehat{\mathbf{U}}_m\|_2}{\|\mathbf{U}_m\|_2} \leq \epsilon(2N + 1)\kappa^{-2} \left(\left[\frac{\delta_k(-1)}{\tau} \right]^\alpha + \mu \right) (\mu \sin(\max(\alpha(\pi - \theta_k), \pi/2)))^{-1}.$$

For $\mu \in [\mu_0, \infty)$ and $\alpha \in (0, 1)$, we have the uniform estimate

$$(3.13) \quad \begin{aligned} \frac{\|\mathbf{U}_m - \widehat{\mathbf{U}}_m\|_2}{\|\mathbf{U}_m\|_2} &\leq \epsilon(2N + 1)\kappa^{-2} \left(1 + \left[\frac{\delta_k(-1)}{\tau}\right]^\alpha / \mu_0\right) (\sin(\max(\alpha(\pi - \theta_k), \pi/2)))^{-1} \\ &\leq C_k \epsilon \kappa^{-2} N^{1+\alpha}, \end{aligned}$$

where the constant C_k is

$$C_k = 3 \left(1 + \left[\frac{\delta_k(-1)}{T}\right]^\alpha / \mu_0\right) (\sin(\max(\alpha(\pi - \theta_k), \pi/2)))^{-1}.$$

Note that the constant C_k depends on the order of BDF method, β_0 in (1.2), the terminal time T , and the fractional order α .

3.3. Representation of numerical solution and convergence analysis.

Next, we represent the solution of the time stepping scheme (3.7) as a contour integral in the complex domain. Following the argument in section 2.3, the solution to the time stepping scheme (3.7) can be written as

$$(3.14) \quad U^n = (I + F_\tau^n)v + \tau \sum_{j=1}^n E_\tau^{n-j} \bar{f}_j,$$

where the discrete operators F_τ^n and E_τ^n are, respectively, defined by

$$(3.15) \quad \begin{aligned} F_\tau^n &= -\frac{1}{2\pi i} \int_{|\xi|=\rho} \frac{1}{\xi^n(1-\xi)} \left((\delta_k(\xi)/\tau)^\alpha + A \right)^{-1} A \, d\xi, \\ E_\tau^n &= \frac{1}{2\pi i} \int_{|\xi|=\rho} \xi^{-n-1} \left((\delta_k(\xi)/\tau)^\alpha + A \right)^{-1} \, d\xi. \end{aligned}$$

The following lemma provides the decay properties of the discrete solution operator. The proof is standard (see, for example, [43, 31]), and hence it is omitted.

LEMMA 3.4. *For the solution operators E_τ^n defined by (3.15), it holds that*

$$\|E_\tau^n\|_{H \rightarrow H} \leq c \tau_{n+1}^{\alpha-1} \quad \text{for all } n = 1, 2, \dots, N,$$

where the constant c is independent of τ and n .

In this section, we aim to show the convergence of the iterative method (3.10) by choosing an appropriate parameter κ . Equivalently, the scheme (2.5) could be reformulated as

$$(3.16) \quad \begin{aligned} \bar{\partial}_\tau^\alpha U_m^n + AU_m^n &= \bar{f}_n - \frac{\kappa}{\tau^\alpha} G_m^n, \quad n = 1, 2, \dots, N, \\ U_m^n &= v, \quad n \leq 0, \end{aligned}$$

where the term G_m^n is given by

$$(3.17) \quad G_m^n = \sum_{j=0}^{N-1} \omega_{n+j}^{(\alpha)} \left(U_m^{N-j} - U_{m-1}^{N-j} \right).$$

We aim to show that U_m^N converges to U^N , the solution of the CQ-BDF k scheme (3.7), as $m \rightarrow \infty$.

LEMMA 3.5. Let U_m^n be the solution to the iterative algorithm (3.10) with $v = 0$ and $\bar{f}_n = 0$ for all $n = 1, 2, \dots, N$. Then we can choose a proper parameter $\kappa = O(1/\log(N))$ in (2.5), such that the following estimate holds:

$$\|U_m^n\| \leq ct_n^{\alpha-1} \gamma(\kappa)^m \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U_0^{N-j+1}\| \right).$$

Here $\gamma(\kappa) \in (0, 1)$ might depend on α, κ, β_0 , and T , but independent of τ, n , and m .

Proof. By the equivalent formula (3.16) and the expression (3.14), we have

$$U_m^n = -\kappa\tau^{1-\alpha} \sum_{i=1}^n E_\tau^{n-i} G_m^i = -\kappa\tau^{1-\alpha} \sum_{i=1}^n E_\tau^{n-i} \sum_{j=0}^{N-1} \omega_{j+i}^{(\alpha)} (U_m^{N-j} - U_{m-1}^{N-j}).$$

Now we take the H norm in the above equality and apply Lemma 3.4 to obtain that

$$\begin{aligned} \|U_m^n\| &\leq c\kappa\tau^{1-\alpha} \sum_{i=1}^n t_{n-i+1}^{\alpha-1} \sum_{j=0}^{N-1} |\omega_{j+i}^{(\alpha)}| \|U_m^{N-j} - U_{m-1}^{N-j}\| \\ &\leq c\kappa \sum_{i=1}^n (n-i+1)^{\alpha-1} \sum_{j=0}^{N-1} |\omega_{j+i}^{(\alpha)}| \|U_m^{N-j} - U_{m-1}^{N-j}\|. \end{aligned}$$

Then Lemma 3.1 indicates that

$$\begin{aligned} \|U_m^n\| &\leq c\kappa \sum_{i=1}^n (n-i+1)^{\alpha-1} \sum_{j=0}^{N-1} (j+i)^{-\alpha-1} \|U_m^{N-j} - U_{m-1}^{N-j}\| \\ (3.18) \quad &= c\kappa \sum_{j=0}^{N-1} \|U_m^{N-j} - U_{m-1}^{N-j}\| \sum_{i=1}^n (n-i+1)^{\alpha-1} (j+i)^{-\alpha-1} \\ &\leq c\kappa\tau^\alpha \sum_{j=0}^{N-1} t_{j+1}^{-\alpha} \|U_m^{N-j} - U_{m-1}^{N-j}\| \sum_{i=1}^n (n-i+1)^{\alpha-1} (j+i)^{-1} \\ &\leq c\kappa t_n^{\alpha-1} \ln(n+1) \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U_m^{N-j+1} - U_{m-1}^{N-j+1}\| \right). \end{aligned}$$

The last inequality follows from the estimate that [32, Lemma 11]

$$\sum_{i=1}^n (n-i+1)^{\alpha-1} (j+i)^{-1} \leq cn^{\alpha-1} \ln(n+1).$$

Multiplying $\tau t_{N-n+1}^{-\alpha}$ on (3.18) and summing over n , we derive that for $\alpha \in (0, 1)$

$$\begin{aligned} \tau \sum_{n=1}^N t_n^{-\alpha} \|U_m^{N-n+1}\| &\leq c\kappa \left(\tau \sum_{n=1}^N t_{N-n+1}^{-\alpha} t_n^{\alpha-1} \ln(n+1) \right) \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U_m^{N-j+1} - U_{m-1}^{N-j+1}\| \right) \\ &\leq c\kappa \log(N) \left(\sum_{n=1}^N (N-n+1)^{-\alpha} n^{\alpha-1} \right) \left(\tau \sum_{n=1}^N t_n^{-\alpha} \|U_m^{N-n+1} - U_{m-1}^{N-n+1}\| \right) \\ &\leq c\kappa \log(N) \left(\tau \sum_{n=1}^N t_n^{-\alpha} \|U_m^{N-n+1} - U_{m-1}^{N-n+1}\| \right), \end{aligned}$$

where the constant c in the second inequality depends on α . In the last inequality, we use the fact that

$$\begin{aligned} \sum_{n=1}^N (N-n+1)^{-\alpha} n^{\alpha-1} &\leq \sum_{n=1}^N \int_{n-1}^n (N-n+1)^{-\alpha} n^{\alpha-1} ds \leq \sum_{n=1}^N \int_{n-1}^n (N-s)^{-\alpha} s^{\alpha-1} ds \\ &= \int_0^N (N-s)^{-\alpha} s^{\alpha-1} ds = \int_0^1 (1-s)^{-\alpha} s^{\alpha-1} ds = B(\alpha, 1-\alpha), \end{aligned}$$

where $B(\cdot, \cdot)$ denotes the Beta function.

Next, we apply the triangle inequality, chose κ small enough such that $c\kappa \log(N) < 1$, and hence derive that

$$(3.19) \quad \tau \sum_{n=1}^N t_n^{-\alpha} \|U_m^{N-n+1}\| \leq \frac{c\kappa \log(N)}{1 - c\kappa \log(N)} \left(\tau \sum_{n=1}^N t_n^{-\alpha} \|U_{m-1}^{N-n+1}\| \right).$$

Finally, we define

$$(3.20) \quad \gamma(\kappa) = \frac{c\kappa \log(N)}{1 - c\kappa \log(N)}.$$

By choosing κ such that $c\kappa \log(N) \in (0, 1/2)$, we have $\gamma(\kappa) \in (0, 1)$. Therefore, by (3.18), we obtain that

$$\begin{aligned} \|U_m^n\| &\leq c\kappa t_n^{\alpha-1} \log(N) \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U_m^{N-j+1} - U_{m-1}^{N-j+1}\| \right) \\ &\leq c\kappa t_n^{\alpha-1} \log(N) \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U_m^{N-j+1}\| + \tau \sum_{j=1}^N t_j^{-\alpha} \|U_{m-1}^{N-j+1}\| \right) \\ &\leq c\kappa t_n^{\alpha-1} \log(N) (1 + \gamma(\kappa)) \tau \sum_{j=1}^N t_j^{-\alpha} \|U_{m-1}^{N-j+1}\| \\ &\leq c t_n^{\alpha-1} \gamma(\kappa) \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U_{m-1}^{N-j+1}\| \right). \end{aligned}$$

Then repeating the estimate (3.19) leads to the desired result. \square

The stability result in Lemma 3.5 then leads to the convergence.

COROLLARY 3.6. *Let U_m^n be the solution to the iterative scheme (3.10), and let U^n be the solution to the k -step BDF scheme (3.7). Then we can choose a proper parameter $\kappa = O(1/\log(N))$ in (3.10), such that the following estimate holds:*

$$\|U^n - U_m^n\| \leq c t_n^{\alpha-1} \gamma(\kappa)^m \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U^{N-j+1} - U_0^{N-j+1}\| \right) \quad \text{for all } m \geq 1.$$

Here the convergence factor $\gamma(\kappa)$, given by (3.20), might depend on κ , β_0 , and T , but independent of τ , n , and m .

Proof. To this end, we let $e_m^n = (U_m^n - U^n)$ with $1 \leq n \leq N$ and note the fact that $\{U^n\}_{n=1}^N$ is the fixed point of the iteration (2.5). Therefore e_m^n satisfies

$$(3.21) \quad \begin{aligned} \bar{\partial}_\tau^\alpha e_m^n + Ae_m^n &= -\frac{\kappa}{\tau} K_m^n, \quad n = 1, 2, \dots, N, \\ e_m^{-j} &= 0, \quad j = 0, 1, \dots, k-1, \end{aligned}$$

where the term K_m^n is given by

$$K_m^n = \sum_{j=0}^{N-1} \omega_{n+j} (e_m^{N-j} - e_{m-1}^{N-j}).$$

Then the convergence estimate follows immediately from Lemma 3.5. □

Combining Corollary 3.6 with the estimate (3.9), we have the following error estimate of the iterative scheme (3.10).

THEOREM 3.7. *Suppose that the condition (1.2) and the assumption of data regularity (3.8) hold true. Let U_m^n be the solution to the iterative algorithm (3.10) with initial guess $U_0^n = v$ for all $0 \leq n \leq N$, and let u be the exact solution to the subdiffusion equation (1.4). Then for $1 \leq n \leq N$, we have*

$$\|U_m^n - u(t_n)\| \leq c(\gamma(\kappa)^m t_n^{\alpha-1} + \tau^k t_n^{-k}), \quad \text{with } \kappa = O(1/\log(N)).$$

Here constant c and the convergence factor $\gamma(\kappa)$ given by (3.20) might depend on k, κ, β_0, T, v , and f , but they are independent of τ, n, m , and u .

Proof. We split the error into two parts:

$$U_m^n - u(t_n) = (U_m^n - U^n) + (U^n - u(t_n)).$$

The second term has the error bound by (3.9). Meanwhile, via Corollary 3.6, the first component converges to zero as $m \rightarrow \infty$, and we have the estimate

$$\begin{aligned} \|U^n - U_m^n\| &\leq c t_n^{\alpha-1} \gamma(\kappa)^m \left(\tau \sum_{j=1}^N t_j^{-\alpha} \|U^{N-j+1} - v\| \right) \\ &\leq c t_n^{\alpha-1} \gamma(\kappa)^m \left(\|v\| + \tau \sum_{j=1}^N t_j^{-\alpha} \|U^{N-j+1}\| \right). \end{aligned}$$

Noting that the estimate (3.9) and the assumption of data regularity (3.8) imply the uniform bound of U^n for all $n = 1, 2, \dots, N$, we obtain that

$$\|U^N - U_m^N\| \leq c_T t_n^{\alpha-1} \gamma(\kappa)^m.$$

Then the desired result follows immediately. □

REMARK 3.1. *By the expression of the convergence factor $\gamma(\kappa)$ in (3.20), we expect that the iteration converges linearly when $c\kappa \log(N) \in (0, 1/2)$, i.e., $\kappa < 1/(2c \log N)$. Besides, it implies that the convergence rate might deteriorate slightly for a large N and a fixed κ . Surprisingly, our numerical results indicate that the iteration converges robustly even for relatively large κ (cf. Figure 2(b)), and the step number N seems not affect the convergence rate (cf. Figure 2(a)).*

Downloaded 09/25/22 to 158.132.161.68 . Redistribution subject to SIAM license or copyright; see https://pubs.siam.org/terms-privacy

4. Numerical tests. In this section, we present some numerical results to illustrate and complement our theoretical findings. The computational domain is the unit interval $\Omega = (0, 1)$ for Examples 4.1 and 4.2, and the unit square $\Omega = (0, 1)^2$ for Example 4.3. In space, it is discretized with the piecewise linear Galerkin finite element method on a uniform mesh with mesh size h for one-dimensional problems. For two-dimensional problems, we compute numerical solutions on a uniform triangulation with mesh size h . We focus on the convergence behavior of the iterative solver to the BDF k solution, since the temporal convergence of the BDF k scheme has been theoretically studied and numerically examined in [31]. That is, with the fixed time step size $\tau = T/N$, we measure the error in the m th iteration

$$e_m^N := \|U_m^N - U^N\|_{L^2(\Omega)},$$

where we take the BDF k solution U^N as the reference solution.

4.1. Numerical results for normal diffusion.

Example 4.1 (1D diffusion equation). We begin with the following one-dimensional normal diffusion equation:

$$(4.1) \quad \begin{cases} \partial_t u - \partial_{xx} u = f(x, t) & \text{in } \Omega \times (0, T], \\ u = 0 & \text{on } \partial\Omega \times (0, T], \\ u(0) = v & \text{in } \Omega, \end{cases}$$

where $\Omega = (0, 1)$ and $T = 0.5$. We consider the problem data

$$v(x) = \chi_{(0, \frac{1}{2})}(x) \quad \text{and} \quad f(x, t) = e^t \cos(x),$$

where χ denotes the characteristic function.

First, we check the performance of the algorithm for different PinT BDF k schemes. Taking $\kappa = 0.5$ and $\tau = T/100$, the numerical results using Algorithm 2.1 with different orders of BDF schemes are presented in Table 2. It can be seen that all the PinT BDF k schemes ($k \leq 6$) converge fast in a similar manner. In what follows, we take the PinT BDF3 scheme to check the influence of different N (or τ) and κ .

TABLE 2
Example 4.1: e_m^N for $T = 0.5$, $\tau = T/100$, $h = 1/1000$, and $\kappa = 0.5$.

$m \setminus$ BDF k	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
0	1.20e-01	1.20e-01	1.20e-01	1.20e-01	1.20e-01	1.20e-01
1	4.88e-04	4.43e-04	4.44e-04	4.44e-04	4.44e-04	4.44e-04
2	1.98e-06	1.59e-06	1.60e-06	1.60e-06	1.60e-06	1.60e-06
3	8.05e-09	5.72e-09	5.79e-09	5.79e-09	5.79e-09	5.78e-09
4	2.81e-11	2.37e-11	1.98e-11	1.74e-11	1.92e-11	1.99e-11
5	4.76e-12	3.09e-12	1.11e-12	3.54e-12	1.81e-12	1.10e-12

Taking $\kappa = 0.5$, we report the convergence histories with different time step sizes in Figure 1(a). It is seen that the convergence rate is independent of τ , which agrees well with Corollary 2.6. In Figure 1(b) we plot the convergence histories with different κ . It can be seen that, with the decrease of κ , the convergence becomes faster, which is in agreement with the convergence rate in theory (2.18). On the other hand, the smaller κ will lead to larger roundoff errors, as we proved in section 3.2. Hence, one needs to choose κ properly to balance the convergence rate and roundoff error.

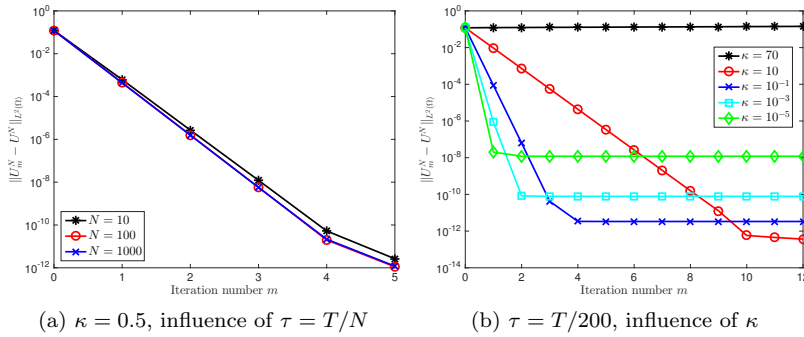


FIG. 1. *PinT BDF3 for Example 4.1: $T = 0.5$ and $h = 1/1000$.*

4.2. Numerical results for subdiffusion. In this subsection, we test the performance of the algorithm for the subdiffusion problem in both 1D and 2D:

$$(4.2) \quad \begin{cases} \partial_t^\alpha u - \Delta u = f(x, t) & \text{in } \Omega \times (0, T], \\ u = 0 & \text{on } \partial\Omega \times (0, T], \\ u(0) = v & \text{in } \Omega. \end{cases}$$

Example 4.2 (1D subdiffusion equation). In the one-dimensional problem, the computational domain is $\Omega = (0, 1)$ with equally spaced mesh. The mesh size is set to be $h = 1/M$ with $M = 1000$. We consider the problem data

$$v = \delta_{1/2}(x) \quad \text{and} \quad f(x, t) = 0.$$

Here the initial data is the Dirac-delta measure concentrated at $x = \frac{1}{2}$, which only belongs to $H^{-\frac{1}{2}-\epsilon}(\Omega)$ for any $\epsilon > 0$. In the computation, the initial value is set to be the L^2 -projection of delta function; see some details in [30].

Even though the initial condition is very weak, the inverse inequality and the analysis in [30, 31] imply the error estimate

$$\|U_n - u(t_n)\| \leq c(\tau^k h^{-\frac{1}{2}-\epsilon} t_n^{-k} + h^{\frac{3}{2}-\epsilon} t_n^{-\alpha})$$

and also the convergence result

$$\|U_n - U_n^m\| \leq ct_n^{\alpha-1} h^{-\frac{1}{2}-\epsilon} \gamma(\kappa)^m$$

with the same $\gamma(\kappa)$ as defined in (3.20).

Similar to the normal diffusion, the performance of all the *PinT BDFk* ($k \leq 6$) have the same convergence profile; see Table 3. Moreover, we check the influence of different N (or τ) and κ by using the *PinT BDF3* scheme. Our theoretical result (3.20) indicates that the convergence factor is

$$\gamma(\kappa) = \frac{c\kappa \log(N)}{1 - c\kappa \log(N)},$$

with some generic constant $c > 1$. So we expect that the iteration converges linearly when $c\kappa \log(N) \in (0, 1/2)$, i.e., $\kappa < 1/(2c \log N)$. Besides, it implies that the convergence rate might deteriorate slightly for a large N and a fixed κ . Surprisingly,

our numerical results indicate that the iteration converges robustly even for relatively large κ (cf. Figure 2(b)), and the step number N seems not to affect the convergence rate (cf. Figure 2(a)). From Figure 2(b), we observe that the influence of κ is similar to the normal diffusion case: the smaller κ will lead to a faster convergence rate but worse roundoff error. In practice, the choice $\kappa \approx 10^{-1}$ leads to an acceptable roundoff error ($\approx 10^{-11}$), and meanwhile the convergence is fast.

TABLE 3
 Example 4.2: e_m^N for $T = 0.1$, $\alpha = 0.5$, $\tau = T/100$, $h = 1/1000$, and $\kappa = 0.1$.

$m \setminus$ BDFk	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
0	2.46e-01	2.46e-01	2.46e-01	2.46e-01	2.46e-01	2.46e-01
1	6.31e-04	6.28e-04	6.28e-04	6.28e-04	6.28e-04	6.28e-04
2	2.88e-06	2.85e-06	2.85e-06	2.84e-06	2.85e-06	2.85e-06
3	1.34e-08	1.32e-08	1.32e-08	1.32e-08	1.33e-08	1.31e-08
4	8.12e-11	4.79e-11	4.98e-11	8.31e-11	4.27e-11	1.37e-10
5	1.88e-11	8.47e-12	1.86e-11	1.61e-11	3.44e-11	1.50e-10

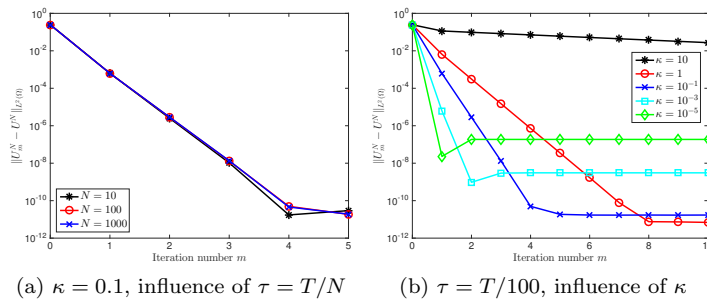


FIG. 2. $PinT$ BDF3 for Example 4.2: $\alpha = 0.5$, $T = 0.1$, and $h = 1/1000$.

Example 4.3 (2D subdiffusion equation). In this example, the spatial discretization is taken on the uniform triangulation of $\Omega = (0, 1)^2$. We consider the problem data

$$v(x) = \chi_{(0, \frac{1}{2}) \times (0, \frac{1}{2})}(x) \quad \text{and} \quad f(x, t) = \cos(t)\chi_{(\frac{1}{2}, 1) \times (\frac{1}{2}, 1)}(x).$$

The numerical solutions are computed on a uniform triangular mesh with $h = 10^{-2}$. We also observe that κ needs to be properly chosen to balance the convergence rate and roundoff error; see Figure 3.

Recall that, in the analysis, there is a generic constant c in the convergence rate (3.20) which depends on the fractional order α and T . We numerically check these dependencies and present the results in Figures 4 and 5, respectively. Taking $\kappa = 1/\log(N)$, we observe the faster convergence rate with smaller α when T is small; see Figure 4(a). With the increase of T , this difference is getting smaller. Further, we observe the faster convergence rate with greater T for various α in Figure 5, which shows the significant advantage of the proposed method for long-time simulation.

4.3. Extension to nonlinear problems. In this part, we shall briefly discuss a possible application of the time-parallel algorithm to the semilinear (sub)diffusion

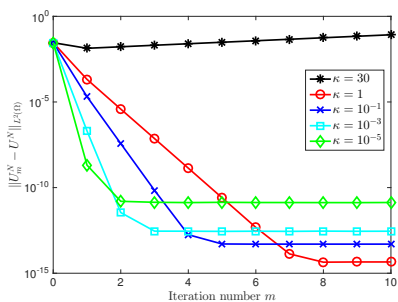


FIG. 3. *PinT BDF3 for Example 4.3: convergence histories for $\alpha = 0.5$, $\tau = 10^{-3}$, and $h = 10^{-2}$.*

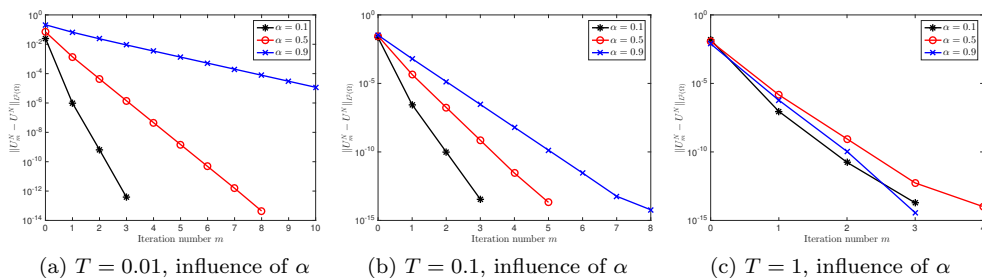


FIG. 4. *PinT BDF3 for Example 4.3: $\kappa = 1/\log(N)$, $\tau = 10^{-3}$, and $h = 10^{-2}$.*

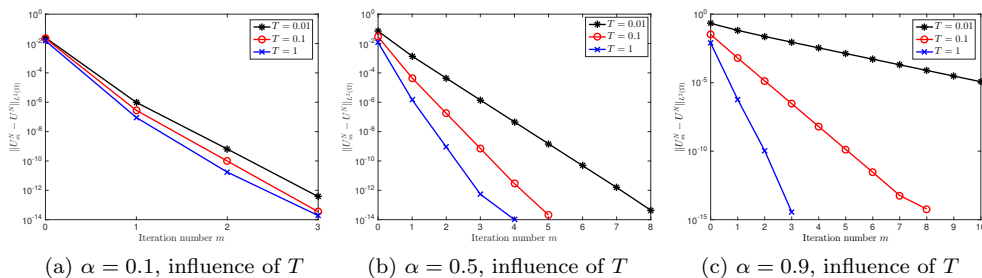


FIG. 5. *PinT BDF3 for Example 4.3: influence of T for the convergence rate, $\kappa = 1/\log(N)$, $\tau = 10^{-3}$, and $h = 10^{-2}$.*

problem (with $\alpha \in (0, 1]$):

$$(4.3) \quad \begin{cases} \partial_t^\alpha u(t) + Au(t) + g(u(t)) = f(t) & \text{for all } t \in (0, T], \\ u(0) = v. \end{cases}$$

To numerically solve (4.3), we follow an idea similar to that introduced in section 3 and consider the modified CQ-BDF k scheme:

$$(4.4) \quad \begin{aligned} \bar{\partial}_\tau^\alpha U^n - \Delta U^n + g(U^n) &= \bar{f}_n, & 1 \leq n \leq N, \\ U^n &= v, & n \leq 0, \end{aligned}$$

where $\bar{f}_n := f(x, t_n) + a_n^{(k)}(f(x, 0) + \Delta v - g(v)) + \sum_{\ell=1}^{k-2} b_{\ell,n}^{(k)} \tau^\ell \partial_t^\ell f(x, 0)$. If $\alpha = 1$, it reduces to a modified BDF k scheme for the classical semilinear parabolic equations.

It was proved in [59, Theorem 3.4] that

$$(4.5) \quad \|U^n - u(t_n)\|_H \leq c_T t_n^{\alpha - \min(k, 1 + 2\alpha - \epsilon)} \tau^{\min(k, 1 + 2\alpha - \epsilon)}$$

for arbitrarily small ϵ .

In order to solve the numerical solution in a time-parallel manner, we consider a modified Newton's iteration to linearize the problem: for integer $\ell \geq 1$, we compute $U_\ell^n = U_{\ell-1}^n + W_\ell^n$, where W_ℓ^n satisfies homogeneous Dirichlet boundary condition and

$$(4.6) \quad \begin{aligned} (\bar{\partial}_\tau^\alpha - \Delta)W_\ell^n - g'(\bar{U}_{\ell-1})W_\ell^n &= \bar{f}_n - (\bar{\partial}_\tau^\alpha - \Delta)U_{\ell-1}^n - g(U_{\ell-1}^n), & 1 \leq n \leq N, \\ W_\ell^n &= 0, & n \leq 0. \end{aligned}$$

Here $\bar{U}_{\ell-1}$ denotes an average of $U_{\ell-1}^n$ in all levels, defined as

$$\bar{U}_{\ell-1} := \frac{1}{N} \sum_{n=1}^N U_{\ell-1}^n.$$

Then, for each iteration, we shall solve the linear system (4.6) with a time-independent coefficient. Therefore, we can apply the strategy in sections 2 and 3, i.e., applying waveform relaxation to derive an iterative solver: with $W_{\ell,0}^n = 0$ and $m = 1, 2, \dots$, for given $W_{\ell,m-1}^n$, we compute $W_{\ell,m}^n$ such that

$$(4.7) \quad \begin{aligned} (\bar{\partial}_\tau^\alpha - \Delta)W_{\ell,m}^n - g'(\bar{U}_{\ell-1})W_{\ell,m}^n &= \bar{f}_n - (\bar{\partial}_\tau^\alpha - \Delta)U_{\ell-1}^n - g(U_{\ell-1}^n), & n = 1, 2, \dots, N, \\ W_{\ell,m}^{-n} &= \kappa(W_{\ell,m}^{N-n} - W_{\ell,m-1}^{N-n}), & n = 0, 1, \dots, N+1, \\ W_{\ell,m}^n &= 0, & n \leq -N. \end{aligned}$$

This is a periodic-like system and hence could be solved in parallel by a diagonalization technique. We describe the complete iterative algorithm in Algorithm 4.1.

Algorithm 4.1 PinT BDFk scheme for nonlinear models.

- 1: Initialize $U_0^n = v$ for all $n \in \mathbb{Z}$, and set $\ell = 0$.
 - 2: **for** $\ell = 1, \dots, L$ **do**
 - 3: Initialize $W_{\ell,0}^n = 0$ for all $n \in \mathbb{Z}$, and set $m = 0$.
 - 4: **for** $m = 1, \dots, m_\ell$ **do**
 - 5: Solve $W_{\ell,m}^n$ satisfying (4.7) in a time parallel manner using Algorithm 2.1 or 3.1.
 - 6: Check the stopping criterion of waveform relaxation.
 - 7: **end for**
 - 8: Update $U_\ell^n = U_{\ell-1}^n + W_{\ell,m}^n$.
 - 9: Check the stopping criterion of Newton's iteration.
 - 10: **end for**
-

Example 4.4 (Allen-Cahn equations). Taking $A = -\Delta$, $H = L^2(\Omega)$, $V = H_0^1(\Omega)$, and $g(u) = \frac{1}{\varepsilon^2}(u^3 - u)$ in (4.3), we obtain the nonlinear problem with $\alpha \in (0, 1]$:

$$(4.8) \quad \begin{cases} \partial_t^\alpha u - \Delta u + \frac{1}{\varepsilon^2}(u^3 - u) = f(x, t) & \text{in } \Omega \times (0, T), \\ u = 0 & \text{on } \partial\Omega \times (0, T), \\ u(0) = v & \text{in } \Omega. \end{cases}$$

In the case that $\alpha = 1$ and $f(x, t) = 0$, the model is called the Allen–Cahn equation, which is a popular phase-field model, introduced in [1] to describe the motion of antiphase boundaries in crystalline solids. In the context, u represents the concentration of one of the two metallic components of the alloy, and the parameter ε involved in the nonlinear term represents the interfacial width, which is small compared to the characteristic length of the laboratory scale; see also [2, 6, 65] for some applications and [9] for some discussion for fractional models with $\alpha \in (0, 1)$.

We shall investigate the numerical performance of Algorithm 4.1 on the domain $\Omega = (0, 1)$ with equally spaced mesh. The exterior force $f(x, t)$ is chosen such that the exact solution yields $u = \frac{t^2}{\Gamma(3)} \sin(2\pi x)$.

First of all, we test the nonlinear problem (4.8) with $\varepsilon = 1$ (mild nonlinearity) and report the error of Newton’s iteration, i.e., $e_\ell^N = \|U_\ell^N - U^N\|_{L^2(\Omega)}$. In the computation, we choose the stopping criteria of inner iteration (waveform relaxation) as

$$\|W_{\ell,m}^n - W_{\ell,m-1}^n\|_\infty < 1 \times 10^{-12} \quad \text{for all } \ell \in \mathbb{N}^+.$$

The numbers of inner iterations are listed in the parentheses. Invoking the error estimate (4.5), we report the numerical results for $\alpha = 0.25$ (with BDF1 and BDF2) and $\alpha = 0.75$ (with BDF1, BDF2, and BDF3) in Table 4. Numerical results in Table 4 indicate that the inner iteration (waveform relaxation) converges robustly and quickly for the linearized system (4.6), and the modified Newton’s iteration converges fast for both cases ($\alpha = 0.25$ and $\alpha = 0.75$), as does Algorithm 4.1.

TABLE 4

Example 4.4: e_ℓ^N with $T = 0.4$, $\tau = T/100$, $h = 1/1000$, $\kappa = 0.1$, and $\varepsilon = 1$.

$\ell \setminus \text{BDF}k$	$k = 1$	$k = 2$
0	6.30e-02	6.30e-02
1	9.27e-06(5)	9.27e-06(5)
2	3.64e-09(4)	3.63e-09(4)
3	1.42e-12(3)	1.42e-12(3)

(a) $\alpha = 0.25$

$\ell \setminus \text{BDF}k$	$k = 1$	$k = 2$	$k = 3$
0	5.94e-02	5.94e-02	5.94e-02
1	6.83e-06(5)	6.80e-06(5)	6.80e-06(5)
2	1.95e-09(4)	1.92e-09(4)	1.92e-09(4)
3	5.23e-13(3)	5.06e-13(3)	5.03e-13(3)

(b) $\alpha = 0.75$

Next, we investigate the influence of the strength of nonlinearity for both subdiffusion and normal diffusion cases in Figure 6. As can be seen from Table 4, the convergence behaviors of Algorithm 4.1 are insensitive with various BDF k schemes, and hence the results of the BDF2 scheme are presented. We observe that strong nonlinearity will lower the convergence rate, not only due to the strong nonlinearity itself to the Newton’s iteration, but also possibly to the more inaccurate average $\bar{U}_{\ell-1}$ in (4.6). As the nonlinearity gets stronger, for example, $\varepsilon = 0.05$ with $\alpha = 0.75$ or $\alpha = 1$, the modified Newton’s iteration does not converge. It is reasonable that the accuracy of average $\bar{U}_{\ell-1}$ hinges on the variation of the solutions on a certain time interval. Practically, a windowing technique could be used in this algorithm: after a certain number of time steps computed in parallel in the current time window, the

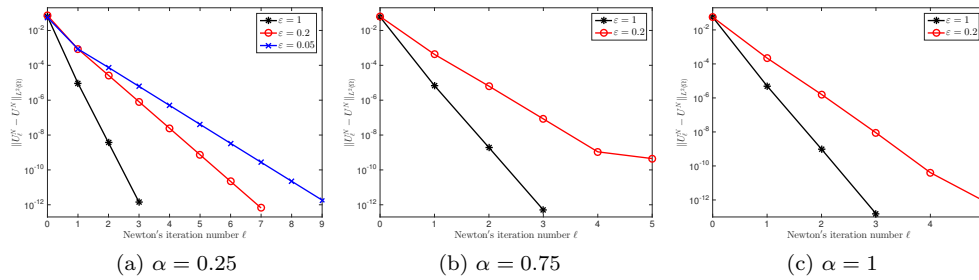


FIG. 6. *PinT* BDF3 for Example 4.4: influence of the strength of nonlinearity.

computation can be restarted for the next time window in a sequential way. This is beyond the scope of current paper and can be considered in the future.

REFERENCES

- [1] S. M. ALLEN AND J. W. CAHN, *A microscopic theory for anti-phase boundary motion and its application to anti-phase domain coarsening*, *Acta Metall.*, 27 (1979), pp. 1085–1095.
- [2] D. M. ANDERSON, G. B. MCFADDEN, AND A. A. WHEELER, *Diffuse-interface methods in fluid mechanics*, *Annu. Rev. Fluid Mech.*, 30 (1998), pp. 139–165.
- [3] D. BAFFET AND J. S. HESTHAVEN, *A kernel compression scheme for fractional differential equations*, *SIAM J. Numer. Anal.*, 55 (2017), pp. 496–520, <https://doi.org/10.1137/15M1043960>.
- [4] G. BAL, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, in *Domain Decomposition Methods in Science and Engineering*, Lect. Notes Comput. Sci. Eng. 40, Springer, Berlin, 2005, pp. 425–432.
- [5] L. BANJAI AND M. LÓPEZ-FERNÁNDEZ, *Efficient high order algorithms for fractional integrals and fractional differential equations*, *Numer. Math.*, 141 (2019), pp. 289–317.
- [6] L.-Q. CHEN, *Phase-field models for microstructure evolution*, *Annu. Rev. Materials Res.*, 32 (2002), pp. 113–140.
- [7] S. CHEN, J. SHEN, Z. ZHANG, AND Z. ZHOU, *A spectrally accurate approximation to subdiffusion equations using the log orthogonal functions*, *SIAM J. Sci. Comput.*, 42 (2020), pp. A849–A877, <https://doi.org/10.1137/19M1281927>.
- [8] E. CUESTA, C. LUBICH, AND C. PALENCIA, *Convolution quadrature time discretization of fractional diffusion-wave equations*, *Math. Comp.*, 75 (2006), pp. 673–696.
- [9] Q. DU, J. YANG, AND Z. ZHOU, *Time-fractional Allen-Cahn equations: Analysis and numerical methods*, *J. Sci. Comput.*, 85 (2020), 42.
- [10] M. FISCHER, *Fast and parallel Runge–Kutta approximation of fractional evolution equations*, *SIAM J. Sci. Comput.*, 41 (2019), pp. A927–A947, <https://doi.org/10.1137/18M1175616>.
- [11] M. J. GANDER, *50 years of time parallel time integration*, in *Multiple Shooting and Time Domain Decomposition Methods*, *Contrib. Math. Comput. Sci.* 9, Springer, Cham, 2015, pp. 69–113.
- [12] M. J. GANDER, L. HALPERN, J. RANNOU, AND J. RYAN, *A direct time parallel solver by diagonalization for the wave equation*, *SIAM J. Sci. Comput.*, 41 (2019), pp. A220–A245, <https://doi.org/10.1137/17M1148347>.
- [13] M. J. GANDER, Y.-L. JIANG, B. SONG, AND H. ZHANG, *Analysis of two parareal algorithms for time-periodic problems*, *SIAM J. Sci. Comput.*, 35 (2013), pp. A2393–A2415, <https://doi.org/10.1137/130909172>.
- [14] M. J. GANDER, J. LIU, S.-L. WU, X. YUE, AND T. ZHOU, *ParaDiag: Parallel-in-Time Algorithms Based on the Diagonalization Technique*, preprint, <https://arxiv.org/abs/2005.09158>, 2020.
- [15] M. J. GANDER AND S. VANDEWALLE, *Analysis of the parareal time-parallel time-integration method*, *SIAM J. Sci. Comput.*, 29 (2007), pp. 556–578, <https://doi.org/10.1137/05064607X>.
- [16] M. J. GANDER AND S.-L. WU, *Convergence analysis of a periodic-like waveform relaxation method for initial-value problems via the diagonalization technique*, *Numer. Math.*, 143 (2019), pp. 489–527.

- [17] F. J. GASPAR AND C. RODRIGO, *Multigrid waveform relaxation for the time-fractional heat equation*, SIAM J. Sci. Comput., 39 (2017), pp. A1201–A1224, <https://doi.org/10.1137/16M1090193>.
- [18] A. GODDARD AND A. WATHEN, *A note on parallel preconditioning for all-at-once evolutionary PDEs*, Electron. Trans. Numer. Anal., 51, 2019, pp. 135–150.
- [19] I. GOLDING AND E. C. COX, *Physical nature of bacterial cytoplasm*, Phys. Rev. Lett., 96 (2006), 098102.
- [20] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins Stud. Math. Sci., Johns Hopkins University Press, Baltimore, MD, 1996.
- [21] X.-M. GU AND S.-L. WU, *A parallel-in-time iterative algorithm for Volterra partial integro-differential problems with weakly singular kernel*, J. Comput. Phys., 417 (2020), 109576.
- [22] W. HACKBUSCH, *Parabolic multigrid methods*, in Computing Methods in Applied Sciences and Engineering, VI (Versailles, 1983), North-Holland, Amsterdam, 1984, pp. 189–197.
- [23] E. HAIRER AND G. WANNER, *Solving Ordinary Differential Equations. II. Stiff and Differential-Algebraic Problems*, 2nd ed., Springer-Verlag, Berlin, 1996.
- [24] G. HORTON AND S. VANDEWALLE, *A space-time multigrid method for parabolic partial differential equations*, SIAM J. Sci. Comput., 16 (1995), pp. 848–864, <https://doi.org/10.1137/0916050>.
- [25] D. HOU AND C. XU, *A fractional spectral method with applications to some singular problems*, Adv. Comput. Math., 43 (2017), pp. 911–944.
- [26] M. A. INDA AND R. H. BISSELING, *A simple and efficient parallel FFT algorithm using the BSP model*, Parallel Comput., 27 (2001), pp. 1847–1878.
- [27] S. JIANG, J. ZHANG, Q. ZHANG, AND Z. ZHANG, *Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations*, Commun. Comput. Phys., 21 (2017), pp. 650–678.
- [28] B. JIN, R. LAZAROV, AND Z. ZHOU, *Two fully discrete schemes for fractional diffusion and diffusion-wave equations with nonsmooth data*, SIAM J. Sci. Comput., 38 (2016), pp. A146–A170, <https://doi.org/10.1137/140979563>.
- [29] B. JIN, R. LAZAROV, AND Z. ZHOU, *Numerical methods for time-fractional evolution equations with nonsmooth data: A concise overview*, Comput. Methods Appl. Mech. Engrg., 346 (2019), pp. 332–358.
- [30] B. JIN, R. LAZAROV, J. PASCIAK, AND Z. ZHOU, *Galerkin FEM for fractional order parabolic equations with initial data in H^{-s} , $0 \leq s \leq 1$* , in Numerical Analysis and Its Applications, Lecture Notes in Comput. Sci. 8236, Springer, Heidelberg, 2013, pp. 24–37.
- [31] B. JIN, B. LI, AND Z. ZHOU, *Correction of high-order BDF convolution quadrature for fractional evolution equations*, SIAM J. Sci. Comput., 39 (2017), pp. A3129–A3152, <https://doi.org/10.1137/17M1118816>.
- [32] B. JIN AND Z. ZHOU, *Incomplete iterative solution of the subdiffusion problem*, Numer. Math., 145 (2020), pp. 693–725.
- [33] A. A. KILBAS, H. M. SRIVASTAVA, AND J. J. TRUJILLO, *Theory and Applications of Fractional Differential Equations*, Elsevier Science B.V., Amsterdam, 2006.
- [34] J. W. KIRCHNER, X. FENG, AND C. NEAL, *Fractal stream chemistry and its implications for contaminant transport in catchments*, Nature, 403 (2000), pp. 524–527.
- [35] N. KOPTEVA, *Error analysis of the L1 method on graded and uniform meshes for a fractional-derivative problem in two and three dimensions*, Math. Comp., 88 (2019), pp. 2135–2155.
- [36] H. LEE, J. LEE, AND D. SHEEN, *Laplace transform method for parabolic problems with time-dependent coefficients*, SIAM J. Numer. Anal., 51 (2013), pp. 112–125, <https://doi.org/10.1137/110824000>.
- [37] B. LI, K. WANG, AND Z. ZHOU, *Long-time accurate symmetrized implicit-explicit BDF methods for a class of parabolic equations with non-self-adjoint operators*, SIAM J. Numer. Anal., 58 (2020), pp. 189–210, <https://doi.org/10.1137/18M1227536>.
- [38] H.-L. LIAO, D. LI, AND J. ZHANG, *Sharp error estimate of the nonuniform L1 formula for linear reaction-subdiffusion equations*, SIAM J. Numer. Anal., 56 (2018), pp. 1112–1133, <https://doi.org/10.1137/17M1131829>.
- [39] M. LÓPEZ-FERNÁNDEZ, C. LUBICH, AND A. SCHÄDLE, *Adaptive, fast, and oblivious convolution in evolution equations with memory*, SIAM J. Sci. Comput., 30 (2008), pp. 1015–1037, <https://doi.org/10.1137/060674168>.
- [40] C. LUBICH, *Discretized fractional calculus*, SIAM J. Math. Anal., 17 (1986), pp. 704–719, <https://doi.org/10.1137/0517050>.
- [41] C. LUBICH, *Convolution quadrature and discretized operational calculus. I*, Numer. Math., 52 (1988), pp. 129–145.

- [42] C. LUBICH AND A. OSTERMANN, *Multigrid dynamic iteration for parabolic equations*, BIT, 27 (1987), pp. 216–234.
- [43] C. LUBICH, I. H. SLOAN, AND V. THOMÉE, *Nonsmooth data error estimates for approximations of an evolution equation with a positive-type memory term*, Math. Comp., 65 (1996), pp. 1–17.
- [44] Y. MADAY AND E. M. RÖNQVIST, *Parallelization in time through tensor-product space-time solvers*, C. R. Math. Acad. Sci. Paris, 346 (2008), pp. 113–118.
- [45] Y. MADAY AND G. TURINICI, *A parareal in time procedure for the control of partial differential equations*, C. R. Math. Acad. Sci. Paris, 335 (2002), pp. 387–392.
- [46] W. MCLEAN AND K. MUSTAPHA, *Convergence analysis of a discontinuous Galerkin method for a sub-diffusion equation*, Numer. Algorithms, 52 (2009), pp. 69–88.
- [47] W. MCLEAN AND K. MUSTAPHA, *Time-stepping error bounds for fractional diffusion problems with non-smooth initial data*, J. Comput. Phys., 293 (2015), pp. 201–217.
- [48] W. MCLEAN, I. H. SLOAN, AND V. THOMÉE, *Time discretization via Laplace transformation of an integro-differential equation of parabolic type*, Numer. Math., 102 (2006), pp. 497–522.
- [49] U. MIEKKALA AND O. NEVANLINNA, *Convergence of dynamic iteration methods for initial value problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 459–482, <https://doi.org/10.1137/0908046>.
- [50] K. MUSTAPHA, B. ABDALLAH, AND K. M. FURATI, *A discontinuous Petrov–Galerkin method for time-fractional diffusion equations*, SIAM J. Numer. Anal., 52 (2014), pp. 2512–2529, <https://doi.org/10.1137/140952107>.
- [51] O. NEVANLINNA, *Remarks on Picard-Lindelöf iteration*. I, BIT, 29 (1989), pp. 328–346.
- [52] R. R. NIGMATULIN, *The realization of the generalized transfer equation in a medium with fractal geometry*, Phys. Stat. Sol. B, 133 (1986), pp. 425–430.
- [53] B. ONG AND J. SCHRODER, *Applications of time parallelization*, Comput. Vis. Sci., 23 (2020), 11.
- [54] I. PODLUBNY, *Fractional Differential Equations*, Academic Press, San Diego, CA, 1999.
- [55] D. SHEEN, I. H. SLOAN, AND V. THOMÉE, *A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature*, Math. Comp., 69 (2000), pp. 177–195.
- [56] M. STYNES, E. O’RIORDAN, AND J. L. GRACIA, *Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation*, SIAM J. Numer. Anal., 55 (2017), pp. 1057–1079, <https://doi.org/10.1137/16M1082329>.
- [57] V. THOMÉE, *Galerkin Finite Element Methods for Parabolic Problems*, 2nd ed., Springer-Verlag, Berlin, 2006.
- [58] S. VANDEWALLE AND R. PIESSENS, *Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1330–1346, <https://doi.org/10.1137/0913075>.
- [59] K. WANG AND Z. ZHOU, *High-order time stepping schemes for semilinear subdiffusion equations*, SIAM J. Numer. Anal., 58 (2020), pp. 3226–3250, <https://doi.org/10.1137/19M1261225>.
- [60] J. A. C. WEIDEMAN AND L. N. TREFETHEN, *Parabolic and hyperbolic contours for computing the Bromwich integral*, Math. Comp., 76 (2007), pp. 1341–1356.
- [61] T. WEINZIERL AND T. KÖPPL, *A geometric space-time multigrid algorithm for the heat equation*, Numer. Math. Theory Methods Appl., 5 (2012), pp. 110–130.
- [62] S.-L. WU AND T. ZHOU, *Convergence analysis for three parareal solvers*, SIAM J. Sci. Comput., 37 (2015), pp. A970–A992, <https://doi.org/10.1137/140970756>.
- [63] Q. XU, J. S. HESTHAVEN, AND F. CHEN, *A parareal method for time-fractional differential equations*, J. Comput. Phys., 293 (2015), pp. 173–183.
- [64] Y. YAN, M. KHAN, AND N. J. FORD, *An analysis of the modified L1 scheme for time-fractional partial differential equations with nonsmooth data*, SIAM J. Numer. Anal., 56 (2018), pp. 210–227, <https://doi.org/10.1137/16M1094257>.
- [65] P. YUE, J. J. FENG, C. LIU, AND J. SHEN, *A diffuse-interface method for simulating two-phase flows of complex fluids*, J. Fluid Mech., 515 (2004), pp. 293–317.
- [66] M. ZAYERNOURI AND G. E. KARNIADAKIS, *Fractional Sturm-Liouville eigen-problems: Theory and numerical approximation*, J. Comput. Phys., 252 (2013), pp. 495–517.
- [67] H. ZHU AND C. XU, *A fast high order method for the time-fractional diffusion equation*, SIAM J. Numer. Anal., 57 (2019), pp. 2829–2849, <https://doi.org/10.1137/18M1231225>.