Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2022-0012s

Vol. **15**, No. 4, pp. 876-902
November 2022

Efficient and Accurate Numerical Methods Using the Accelerated Spectral Deferred Correction for Solving Fractional Differential Equations

Xuejuan Chen[1], Zhiping Mao[2,*] and George Em Karniadakis[3]

[1] *School of Science, Jimei University, Xiamen 361021, China*
[2] *School of Mathematical Sciences, Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, China*
[3] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA*

Received 14 January 2021; Accepted (in revised version) 18 August 2021

Abstract. We develop an efficient and accurate spectral deferred correction (SDC) method for fractional differential equations (FDEs) by extending the algorithm in [14] for classical ordinary differential equations (ODEs). Specifically, we discretize the resulted Picard integral equation by the SDC method and accelerate the convergence of the SDC iteration by using the generalized minimal residual algorithm (GMRES). We first derive the correction matrix of the SDC method for FDEs and analyze the convergence region of the SDC method. We then present several numerical examples for stiff and non-stiff FDEs including fractional linear and nonlinear ODEs as well as fractional phase field models, demonstrating that the accelerated SDC method is much more efficient than the original SDC method, especially for stiff problems. Furthermore, we resolve the issue of low accuracy arising from the singularity of the solutions by using a geometric mesh, leading to highly accurate solutions compared to uniform mesh solutions at almost the same computational cost. Moreover, for long-time integration of FDEs, using the geometric mesh leads to great computational savings as the total number of degrees of freedom required is relatively small.

AMS subject classifications: 65N35, 65E05, 65M70, 41A05, 41A10, 41A25

Key words: Stiff problem, generalized minimal residual, geometric mesh refinement, long time evolution, fractional phase field models.

*Corresponding author. *Email addresses:* xue_105@jmu.edu.cn (X. Chen), zpmao@xmu.edu.cn (Z. Mao), george_karniadakis@brown.edu (G. E. Karniadakis)

1. Introduction

Fractional differential equations (FDEs) have been effective in modeling anomalous diffusion as well as capturing long-range spatio-temporal interactions [11, 24, 26]. Analytic solutions of some FDEs are typically obtained by using special functions (e.g., Wright functions) for simple linear problems [22]. However, it is usually difficult to obtain analytic solutions for more complex FDEs, especially for nonlinear problems. For time-fractional differential equations, high-order numerical methods including the finite difference method [4,10,18,19,29] and the polynomial based spectral method [17] have been developed for smooth solutions. However, due to the singular kernel of the fractional operator, the solutions of FDEs are usually of low regularity in the usual Sobolev space. To resolve this issue, Jin *et al.* used corrections to restore the theoretical high-order convergence rate [15] (see also [20, 21]). A different approach developed by Zayernouri and Karniadakis [34] was to use the poly-fractonomials as basis functions, which match the singularity of the kernel, giving spectral accuracy for a smooth source term; a rigorous error analysis was established in [6]. Nevertheless, this method cannot be extended to more general FDEs. Moreover, none of the aforementioned methods can easily handle the long-time evolution since it is computationally expensive and has high memory requirement due to the non-locality of the fractional operator. A number of papers using the finite difference method have been published to address this issue by using the FFT based discrete convolution or representing the discrete weights into integral forms [12, 13, 32, 33, 35]. However, spectral methods, especially multi-domain spectral methods are more favorable for fractional problems, giving much higher accuracy than local methods [5, 16].

In this work, we aim at developing an efficient and accurate numerical scheme using the spectral deferred correction (SDC) method for FDEs. The SDC method was first introduced by Dutt *et al.* in [9] to construct high-order stable methods for solving ordinary differential equations (ODEs). Some early work on using the SDC method for nonlocal equations or FDEs can be found in [3, 23, 25, 31]. In these works, again, the low regularity of the solutions was not addressed explicitly. It was shown that the convergence rate is $\mathcal{O}(\Delta T^{(2-\alpha)(k+1)})$ (or $\mathcal{O}(\Delta T^{(2-\alpha)+k})$) for the uniform mesh (or the Gauss-Lobatto mesh) in [23] while the convergence rate is $\mathcal{O}(\Delta T^{\min(p+1+\alpha,\alpha(k+1)+\delta)}), \delta = 1$ or 2 in [3], where ΔT is the length of the subdomain, α is the order of the time fractional operator, k is the number of SDC iterations and p is the degree of the polynomial used for the SDC scheme. More details about these parameters will be given in the next section. This means that there is an increase of the order of $2-\alpha, 1$ or α for each iteration for the global error. However, this fails and the so called order reduction occurs when solving stiff problems. Moreover, it is not computationally efficient to solve large systems by using the SDC methods developed in these works.

In the current work, we extend the techniques in [14] (see also [27]) for classical ODEs to fractional ODEs. In particular, we use the SDC method to discretize the time fractional operator for the fractional initial and/or boundary problems, and accelerate the convergence of the SDC iteration with the generalized minimal residual (GMRES)

algorithm with restart. Furthermore, we address the issue of the singularity by employing a non-uniform mesh, i.e., the geometric mesh, near the origin to obtain high accuracy.

The main contributions of the present work are:

- We derive the correction matrix of the SDC method for the FDEs and consequently present the convergence theory and analyze the convergence region of the SDC method.

- We demonstrate the effectiveness of the accelerated SDC method by several numerical examples for both stiff and non-stiff FDEs including linear/nonlinear fractional ODEs as well as fractional phase field models. We show the efficiency of the accelerated SDC method by studying the convergence behavior of the accelerated SDC method with different restart numbers for all considered problems, especially for the stiff problems.

- To overcome the issue of the singularity, we obtain high accuracy by employing geometric mesh refinement in the first subdomain of the original SDC scheme, demonstrating that the SDC method with geometric mesh delivers much higher accuracy than that based on uniform mesh at almost the same computational cost.

The rest of the paper is organized as follows: We introduce the SDC method in the next section. Then, we employ the present accelerated SDC method to linear/nonlinear fractional ODEs in Section 3 and fractional phase field models in Section 4. We conclude in Section 5.

2. Spectral deferred correction method

In this section we formulate the SDC method for FDEs. In particular, we begin by considering the following fractional ODE:

$$ {}_0^C D_t^\alpha u(t) = F\big(t, u(t)\big), \quad t \in [0, T], \quad u(0) = u_0, \tag{2.1} $$

where ${}_0^C D_t^\alpha$ is the Caputo fractional derivative of order $\alpha \in (0, 1]$ defined by

$$ {}_0^C D_t^\alpha u(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{u'(s)}{(t-s)^\alpha} ds. $$

Here $u(t), u_0 \in \mathbb{R}^N$ and $F : \mathbb{R} \times \mathbb{R}^N \longrightarrow \mathbb{R}^N$. We assume that F is continuous, bounded and fulfills a Lipschitz condition with respect to the second variable such that the problem (2.1) is well-posed, i.e., for the problem (2.1) there exists a unique solution $u(t) : [0, T] \to \mathbb{R}^N$ for $T > 0$ (see [8, Theorems 2.1 and 2.2]).

2.1. SDC method based on Euler schemes

The first step of the SDC scheme is to reformulate the above problem (2.1) as a Picard integral equation, which is then discretized on a sequence of time subintervals by the Gauss-type quadrature. We then solve the resulted system by using a low-order numerical method (such as the Euler method) and correct the solution to a higher-order accuracy by solving a sequence of correction equations on the same Gauss-type grid. To this end, we divide the time interval $[0,T]$ into K non-overlapping equidistant subintervals, i.e., $[0,T]=\cup_{j=0}^{K-1}[t_j,t_{j+1}], \Delta T=\Delta T_j=(t_{j+1}-t_j)$ with $0=t_0<t_1<\cdots<t_K=T$. Then, we advance the solution from t_j to t_{j+1} with the SDC method. To simplify the notation, we denote the interval $[t_j,t_{j+1}]$ by the generic interval $[a,b]$.

To obtain the Picard integral equation, we first apply the fractional integral on both sides of Eq. (2.1) to obtain the analytic solution (see [8, Lemma 2.1])

$$u(t)=u_0+\frac{1}{\Gamma(\alpha)}\int_0^t\frac{F\big(\tau,u(\tau)\big)}{(t-\tau)^{1-\alpha}}d\tau,\quad t\in[0,T]. \tag{2.2}$$

By considering the generic interval $[a,b]$, the above equation reduces to

$$u(t)=u_a(t)+\frac{1}{\Gamma(\alpha)}\int_a^t\frac{F\big(\tau,u(\tau)\big)}{(t-\tau)^{1-\alpha}}d\tau, \tag{2.3}$$

where

$$u_a(t)=u_0+\frac{1}{\Gamma(\alpha)}\int_0^a\frac{F\big(\tau,u(\tau)\big)}{(t-\tau)^{1-\alpha}}d\tau$$

representing the non-local history term, which will be computed by the hybrid method described in Appendix A. Suppose that we now compute the numerical solution at the subinterval $[t_j,t_{j+1}]$ by using the SDC algorithm, then we set $a=t_j$.

We now derive the correction equation, which will be used to update the SDC approximation. Assume we have an approximation $u^0(t)$ to (2.3), we then define the error $\delta(t):=u(t)-u^0(t)$ and substitute it into (2.3) leading to

$$u^0(t)+\delta(t)=u_a(t)+\frac{1}{\Gamma(\alpha)}\int_a^t(t-\tau)^{\alpha-1}F\big(\tau,u^0(\tau)+\delta(\tau)\big)d\tau.$$

On the other hand, we can also define the residual function

$$\epsilon\big(t,u^0(t)\big)=u_a(t)+\frac{1}{\Gamma(\alpha)}\int_a^t(t-\tau)^{\alpha-1}F\big(\tau,u^0(\tau)\big)d\tau-u^0(t). \tag{2.4}$$

By combining the above two equations, we derive the so-called correction equation

$$\delta(t)=\frac{1}{\Gamma(\alpha)}\int_a^t(t-\tau)^{\alpha-1}\left[F\big(\tau,u^0(\tau)+\delta(\tau)\big)-F\big(\tau,u^0(\tau)\big)\right]d\tau+\epsilon\big(t,u^0(t)\big). \tag{2.5}$$

A deferred correction method proceeds on a Gauss-type grid in the interval $[a,b]$ by iteratively solving the above correction equation with a low-order method to improve

the provisional solution $\vec{u}^0$, which can be obtained by using either the forward Euler method or the backward Euler method. In this work, we use the $p+1$ Legendre-Gauss-Lobatto points $a = s_0 < s_1 < s_2 < \cdots < s_p = b$ given by the formula $s_i = (b-a)r_i/2 + (b+a)/2, i = 0, 1, \ldots, p$, where $r_0, r_1, r_2, \ldots, r_p$ are the Legendre-Gauss-Lobatto points defined on the interval $[-1, 1]$.

Let u_i^k denote the numerical approximation to $u^k(s_i)$ (likewise for δ_i^k and ϵ_i^k), where the superscript k means the k-th iteration. To compute a sequence of corrections δ_i^k, we have from (2.5) that we need to compute the residual ϵ_i^k by discretizing (2.4). To do this, let $F(t, u^k(t))$ be approximated by Gauss-Legendre-Lobotto interpolation

$$F_p\big(t, u^k(t)\big) = \mathbb{I}_p F\big(t, u^k(t)\big) = \sum_{m=0}^{p} F_m^k \hat{h}_m(t), \tag{2.6}$$

where $F_m^k := F(s_m, u^k(s_m)), \mathbb{I}_p v(t_j) = v(t_j)$ is the interpolation operator and $\hat{h}_j(t)$ is the Lagrange interpolation polynomial based on the $p+1$ Legendre-Gauss-Lobatto points in the interval $[a, b]$. Then, the residual function $\epsilon(s, u^k(s))$ is approximated as follows:

$$\vec{\epsilon}^k = \vec{u}_a + \Delta T^\alpha \boldsymbol{A}\vec{F}^k - \vec{u}^k, \tag{2.7}$$

where

$$\vec{\epsilon}^k = \left[\epsilon_0^k, \epsilon_1^k, \ldots, \epsilon_p^k\right]^T, \qquad \vec{u}_a = \left[u_a(s_0), u_a(s_1), \ldots, u_a(s_p)\right]^T,$$
$$\vec{F}^k = \left[F_0^k, F_1^k, \ldots, F_p^k\right]^T, \quad \vec{u}^k = \left[u_0^k, u_1^k, \ldots, u_p^k\right]^T,$$

$\boldsymbol{A} = I_N \otimes A$ is a $N(p+1) \times N(p+1)$ block diagonal matrix, here I_N is the $N \times N$ identity matrix, A is the fractional spectral integration matrix given by

$$A_{ij} = \frac{1}{2^\alpha \Gamma(\alpha)} \int_{-1}^{r_i} (r_i - r)^{\alpha-1} h_j(r) dr, \tag{2.8}$$

and $h_j(r)$ is the Gauss Lagrange interpolation polynomial based on the $p+1$ Legendre-Gauss-Lobatto points in the interval $[-1, 1]$.

We now construct a sequence of corrections $\{\delta^k\}$ and new approximations $\{u^k\}$. Let

$$G(t) = F\big(t, u(t) + \delta(t)\big) - F\big(t, u(t)\big). \tag{2.9}$$

Using the correction equation (2.5), at the k-th iteration, we can obtain the following equation:

$$\begin{aligned}\delta^k(s_{i+1}) = \delta^k(s_i) &+ \frac{1}{\Gamma(\alpha)} \int_a^{s_i} \left[(s_{i+1} - \tau)^{\alpha-1} - (s_i - \tau)^{\alpha-1}\right] G^k(\tau) d\tau \\ &+ \frac{1}{\Gamma(\alpha)} \int_{s_i}^{s_{i+1}} (s_{i+1} - \tau)^{\alpha-1} G^k(\tau) d\tau + \epsilon_{i+1}(u^k) - \epsilon_i(u^k).\end{aligned} \tag{2.10}$$

Using the backward Euler scheme, we compute the two integrals of the above equation as follows:

$$\frac{1}{\Gamma(\alpha)}\int_a^{s_i}\left[(s_{i+1}-\tau)^{\alpha-1}-(s_i-\tau)^{\alpha-1}\right]G^k(\tau)d\tau \approx \frac{1}{\Gamma(1+\alpha)}\sum_{l=1}^{i}\left(\tilde{s}_{i+1,l}^{\alpha}-\tilde{s}_{i,l}^{\alpha}\right)G_l^k,$$
$$\frac{1}{\Gamma(\alpha)}\int_{s_i}^{s_{i+1}}(s_{i+1}-\tau)^{\alpha-1}G^k(\tau)d\tau \approx \frac{1}{\Gamma(1+\alpha)}\tilde{s}_{i+1,i+1}^{\alpha}G_{i+1}^k,$$

where G_l^k is the numerical approximation of $G^k(s_l)$ and

$$\tilde{s}_{i,l}^{\alpha}=(s_i-s_{l-1})^{\alpha}-(s_i-s_l)^{\alpha},\quad i\geq l\geq 1. \tag{2.11}$$

To simplify the notation, we let $\tilde{s}_{i,i+1}=0$. Then, the correction equation (2.10) together with the above implicit scheme leads to the following scheme:

$$\delta_{i+1}^k=\delta_i^k+\frac{1}{\Gamma(1+\alpha)}\sum_{l=1}^{i+1}\left(\tilde{s}_{i+1,l}^{\alpha}-\tilde{s}_{i,l}^{\alpha}\right)G_l^k+\epsilon_{i+1}(u^k)-\epsilon_i(u^k). \tag{2.12}$$

Let us denote

$$I_i^{i+1}(u^k)=\frac{1}{\Gamma(\alpha)}\Bigg\{\int_0^{s_i}\left[(s_{i+1}-\tau)^{\alpha-1}-(s_i-\tau)^{\alpha-1}\right]F\big(\tau,u^k(\tau)\big)d\tau + \int_{s_i}^{s_{i+1}}(s_{i+1}-\tau)^{\alpha-1}F\big(\tau,u^k(\tau)\big)d\tau\Bigg\}.$$

We then derive from (2.4) that

$$I_i^{i+1}(u^k)=\epsilon_{i+1}(u^k)-\epsilon_i(u^k)+u_{i+1}^k-u_i^k. \tag{2.13}$$

Since $u_l^k+\delta_l^k=u_l^{k+1}$, $l=0,1,\ldots,p$. From (2.12) and (2.13), we obtain

$$u_{i+1}^{k+1}=u_i^{k+1}+I_i^{i+1}(u^k)+\frac{1}{\Gamma(1+\alpha)}\sum_{l=1}^{i+1}\left(\tilde{s}_{i+1,l}^{\alpha}-\tilde{s}_{i,l}^{\alpha}\right)G_l^k. \tag{2.14}$$

Now, we can use the system (2.14) to compute a new approximation u^{k+1} from u^k. Note that $I_i^{i+1}(u^k)$ can be calculated from (2.13). Similarly, by using the forward Euler scheme, we deduce that

$$u_{i+1}^{k+1}=u_i^{k+1}+I_i^{i+1}(u^k)+\frac{1}{\Gamma(1+\alpha)}\sum_{l=1}^{i+1}\left(\tilde{s}_{i+1,l}^{\alpha}-\tilde{s}_{i,l}^{\alpha}\right)G_{l-1}^k. \tag{2.15}$$

2.2. Error estimate

Next we obtain the error estimate with respect to the degree of polynomial p for the solution of the SDC method at each subinterval. Let $u_p(t)$ be the limit approximation solution of the SDC procedure. Then, $u_p(t)$ satisfies

$$u_p(t) = u_0(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \mathbb{I}_p F(\tau) d\tau, \quad t \in [a,b]. \tag{2.16}$$

Using Eqs. (2.1) and (2.16), we obtain

$$\begin{aligned} |u - u_p| &= \frac{1}{\Gamma(\alpha)} \left| \int_0^t (t-\tau)^{\alpha-1} (F - \mathbb{I}_p F)(\tau) d\tau \right| \\ &\leq \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} \left| F - \mathbb{I}_p F \right| d\tau \\ &\leq \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} d\tau \cdot \left\| F - \mathbb{I}_p F \right\|_\infty \\ &\leq \frac{T^\alpha}{\Gamma(1+\alpha)} \left\| F - \mathbb{I}_p F \right\|_\infty . \end{aligned} \tag{2.17}$$

Then, with the help of [7, Eq. (4.11)], we have the following estimate:

Theorem 2.1. *Suppose $F(t) \in H^m([a,b])$, let $u(t)$ and $u_p(t)$ be the solutions of* (2.1) *and* (2.16)*, respectively, it holds that*

$$\|u(t) - u_p(t)\|_\infty \leq C T^\alpha p^{\frac{1}{2}-m} \|F(t)\|_m. \tag{2.18}$$

Remark 2.1. The estimate (2.18) indicates that the convergence depends only on the regularity of F with respect to t.

2.3. SDC scheme in the matrix form

In this subsection, we first derive an explicit representation of the iteration in matrix form for the linear problem, i.e., $F(t,u(t)) = Lu(t) + f(t)$, where $L \in \mathbb{R}^N$ with each of the components is a constant. Thus, the correction equation (2.5) becomes

$$\delta(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} L\delta(\tau) d\tau + \epsilon(t). \tag{2.19}$$

For the k-th correction step, by discretizing the above equation with the backward Euler scheme and noting that $\delta_0^k = \epsilon_0^k = 0$, we obtain

$$\delta_i^k = \frac{1}{\Gamma(\alpha+1)} \sum_{l=1}^{i} L\tilde{s}_{i,l}^\alpha \, \delta_l^k + \epsilon_i^k, \quad i = 1, \ldots, p,$$

where $\tilde{s}_{i,l}^{\alpha}$ is given in (2.11). Writing the above equation into the matrix form yields

$$(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L})\vec{\delta}^{k}=\vec{\epsilon}^{k}, \tag{2.20}$$

where

$$\vec{\delta}^{k}=\left[\delta_0^k,\delta_1^k,\ldots,\delta_p^k\right]^T,\quad \vec{\epsilon}^{k}=\left[\epsilon_0^k,\epsilon_1^k,\ldots,\epsilon_p^k\right]^T,\quad \boldsymbol{L}=\mathrm{diag}(L)\otimes I_{p+1},\quad \widetilde{\boldsymbol{A}}=I_N\otimes\widetilde{A}$$

and the matrix $\Delta T^{\alpha}\widetilde{A}$ is given by

$$\Delta T^{\alpha}\widetilde{A}=\frac{1}{\Gamma(\alpha+1)}\begin{pmatrix}0&0&0&\cdots&0&0\\0&\tilde{s}_{1,1}^{\alpha}&0&\cdots&0&0\\0&\tilde{s}_{2,1}^{\alpha}&\tilde{s}_{2,2}^{\alpha}&\cdots&0&0\\\vdots&\vdots&\vdots&\ddots&\vdots&\vdots\\0&\tilde{s}_{p-1,1}^{\alpha}&\tilde{s}_{p-1,2}^{\alpha}&\cdots&\tilde{s}_{p-1,p-1}^{\alpha}&0\\0&\tilde{s}_{p,1}^{\alpha}&\tilde{s}_{p,2}^{\alpha}&\cdots&\tilde{s}_{p,p-1}^{\alpha}&\tilde{s}_{p,p}^{\alpha}\end{pmatrix}.$$

Similarly, for the explicit Euler scheme, the matrix $\Delta T^{\alpha}\widetilde{A}$ takes the form

$$\Delta T^{\alpha}\widetilde{A}=\frac{1}{\Gamma(\alpha+1)}\begin{pmatrix}0&0&\cdots&0&0&0\\\tilde{s}_{1,1}^{\alpha}&0&\cdots&0&0&0\\\tilde{s}_{2,1}^{\alpha}&\tilde{s}_{2,2}^{\alpha}&\cdots&0&0&0\\\vdots&\vdots&\ddots&\vdots&\vdots&\vdots\\\tilde{s}_{p-1,1}^{\alpha}&\tilde{s}_{p-1,2}^{\alpha}&\cdots&\tilde{s}_{p-1,p-1}^{\alpha}&0&0\\\tilde{s}_{p,1}^{\alpha}&\tilde{s}_{p,2}^{\alpha}&\cdots&\tilde{s}_{p,p-1}^{\alpha}&\tilde{s}_{p,p}^{\alpha}&0\end{pmatrix}.$$

Assuming we have the provisional solution $\vec{u}^{k}$ obtained from the previous correction step, then using Eq. (2.20), we have

$$\vec{u}^{k+1}=\vec{u}^{k}+\vec{\delta}^{k}=\vec{u}^{k}+\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\vec{\epsilon}^{k}.$$

By the above equation and (2.7), we derive

$$\begin{aligned}\vec{u}^{k+1}&=\vec{u}^{k}+\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\left(\vec{u}_a+\Delta T^{\alpha}\boldsymbol{A}\left(\boldsymbol{L}\vec{u}^{k}+\vec{f}\right)-\vec{u}^{k}\right)\\&=\vec{u}^{k}+\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\left(\vec{u}_a-\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)\vec{u}^{k}+\Delta T^{\alpha}\left(\boldsymbol{A}-\widetilde{\boldsymbol{A}}\right)\boldsymbol{L}\vec{u}^{k}+\Delta T^{\alpha}\boldsymbol{A}\vec{f}\right)\\&=\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\vec{u}_a+\boldsymbol{C}\vec{u}^{k}+\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\Delta T^{\alpha}\boldsymbol{A}\vec{f},\end{aligned} \tag{2.21}$$

where $\boldsymbol{C}$ is the so called "correction matrix" given by

$$\boldsymbol{C}=\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\Delta T^{\alpha}\left(\boldsymbol{A}-\widetilde{\boldsymbol{A}}\right)\boldsymbol{L}. \tag{2.22}$$

Consequently, by replacing $k+1$ by k, we have

$$\vec{u}^{k}=\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\vec{u}_a+\boldsymbol{C}\vec{u}^{k-1}+\left(\boldsymbol{I}-\Delta T^{\alpha}\widetilde{\boldsymbol{A}}\boldsymbol{L}\right)^{-1}\Delta T^{\alpha}\boldsymbol{A}\vec{f}.$$

Subtracting the above equation from (2.21) yields the recursive relationship

$$\vec{\delta}^k = \boldsymbol{C}\vec{\delta}^{k-1} = \boldsymbol{C}^k\vec{\delta}^0.$$

Consequently, we have the solution after k corrections given by the Neumann series expansion

$$\vec{u}^k = \vec{u}^0 + \sum_{i=0}^{k-1} \boldsymbol{C}^i\vec{\delta}^0. \tag{2.23}$$

We can also obtain the above Neumann series expansion by discretizing equation (2.19) at the Gauss-type collocation points. In particular, we have

$$\big(\boldsymbol{I} - \Delta T^\alpha \boldsymbol{A}\boldsymbol{L}\big)\vec{\delta} = \vec{\epsilon}, \tag{2.24}$$

where A can be found in (2.7). The SDC procedure is to iteratively approximate the above system with the low order approximations $\vec{\delta}^k$ for $k = 1, 2, \ldots$ obtained by solving the correction system (2.20). Applying the low-order preconditioner $(\boldsymbol{I} - \Delta T^\alpha \widetilde{\boldsymbol{A}}\boldsymbol{L})^{-1}$ to the above system, we obtain the following preconditioned linear system:

$$\big(\boldsymbol{I} - \Delta T^\alpha \widetilde{\boldsymbol{A}}\boldsymbol{L}\big)^{-1}\big(\boldsymbol{I} - \Delta T^\alpha \boldsymbol{A}\boldsymbol{L}\big)\vec{\delta} = \big(\boldsymbol{I} - \Delta T^\alpha \widetilde{\boldsymbol{A}}\boldsymbol{L}\big)^{-1}\vec{\epsilon} = \vec{\delta}^0, \tag{2.25}$$

which gives

$$(\boldsymbol{I} - \boldsymbol{C})\vec{\delta} = \vec{\delta}^0, \tag{2.26}$$

where $\boldsymbol{C}$ is given by (2.22). Note that $\widetilde{\boldsymbol{A}}$ is a low-order approximation of $\boldsymbol{A}$ when $\lambda\Delta T$ is small, then the solution to the above linear system is given by the Neumann series expansion

$$\begin{aligned}\vec{\delta} &= \vec{\delta}^0 + \boldsymbol{C}\vec{\delta} = \vec{\delta}^0 + \boldsymbol{C}\big(\vec{\delta}^0 + \boldsymbol{C}\vec{\delta}\big)\\ &= \vec{\delta}^0 + \boldsymbol{C}\vec{\delta}^0 + \boldsymbol{C}^2\vec{\delta} = \cdots = \vec{\delta}^0 + \boldsymbol{C}\vec{\delta}^0 + \boldsymbol{C}^2\vec{\delta}^0 + \cdots,\end{aligned}$$

which is equivalent to Eq. (2.23). This means that the SDC method is equivalent to solving the spectral collocation system (2.24) preconditioned by the low-order approximation system (2.20). Consequently, we have the following convergence result:

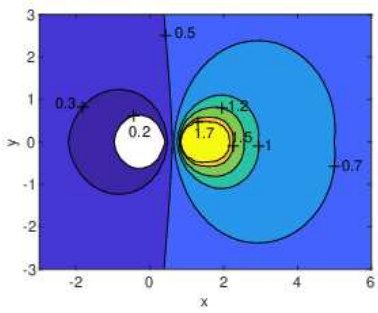
Theorem 2.2. *For linear fractional ODEs, the SDC iteration is convergent if and only if the spectral radius of the correction matrix $\boldsymbol{C}$ given in* (2.22) *is less than 1, i.e., $\rho(\boldsymbol{C}) < 1$.*

2.4. Convergence region

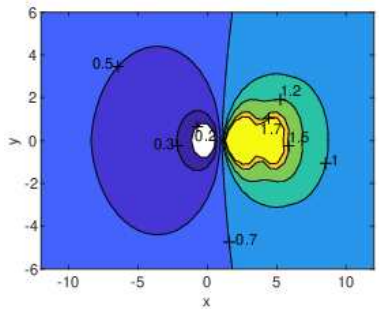
We determine the "convergence region" to quantify when the SDC method is convergent for linear problems. For linear fractional ODEs, we define the "convergence region" Ω of the SDC method as $\Omega = \{\boldsymbol{L} : \rho(\boldsymbol{C}(\boldsymbol{L})) < 1, \boldsymbol{L} \in \mathbb{C}\}$.

Consider the following scalar problem:

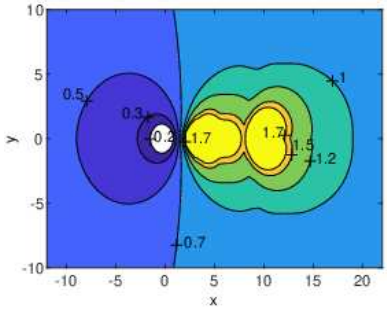
$${}_0^C D_t^\alpha u(t) = \lambda u(t), \quad t \in [0, T]. \tag{2.27}$$



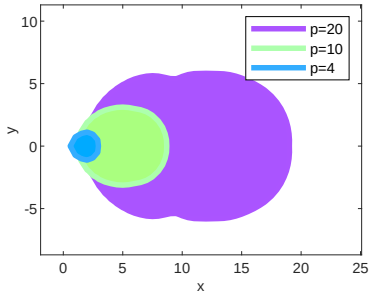
(a) $\alpha = 0.6, p = 4$



(b) $\alpha = 0.6, p = 10$

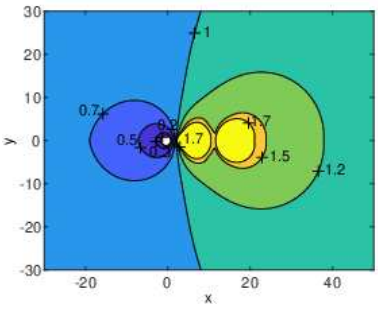


(c) $\alpha = 0.6, p = 20$

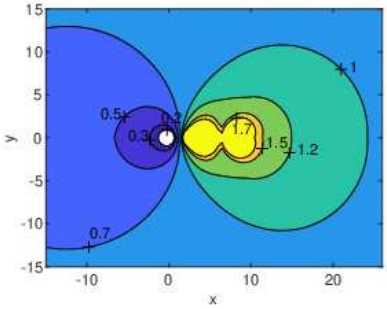


(d) $\rho(\boldsymbol{C}(\lambda)) = 1, \alpha = 0.6$

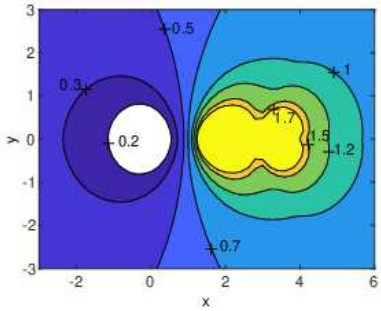
Figure 1: (a)-(c): Contours of $\rho(\boldsymbol{C}(\lambda))$ for different values of p: $\alpha = 0.6, T = 2, \Delta T = 2, \lambda = x + y\mathrm{i}$. (d): Contours of $\rho(\boldsymbol{C}(\lambda)) = 1$.



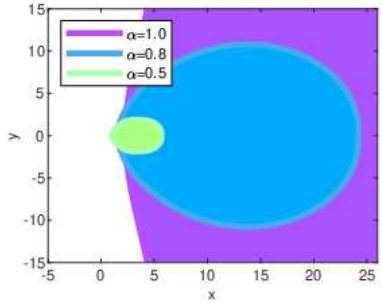
(a) $\alpha = 1.0, p = 10$



(b) $\alpha = 0.8, p = 10$



(c) $\alpha = 0.5, p = 10$



(d) $\rho(\boldsymbol{C}(\lambda)) = 1, p = 10$

Figure 2: (a)-(c): Contours of $\rho(\boldsymbol{C}(\lambda))$ for different values of the fractional order α, $p = 10, T = 2, \Delta T = 2, \lambda = x + y\mathrm{i}$. (d): Contours of $\rho(\boldsymbol{C}(\lambda)) = 1$.

Table 1: The spectral radius $\rho(\boldsymbol{C}(\lambda))$ for different values of p and $\alpha, \lambda = -10000$.

p \ α	1.0	0.9	0.8	0.7	0.6	0.5	0.3
7	0.8595	0.8007	0.7398	0.6758	0.6076	0.5338	0.3621
11	0.9677	0.8971	0.8255	0.7516	0.6738	0.5906	0.3992
15	1.0221	0.9454	0.8684	0.7894	0.7069	0.6189	0.4178
19	1.0544	0.9741	0.8938	0.8120	0.7266	0.6360	0.4290
25	1.0838	1.0003	0.9172	0.8327	0.7449	0.6517	0.4395

First, we study how the degree of polynomial p and the fractional order α affect the convergence region. For the sake of simplicity, we only consider the case of implicit Euler scheme. Let $\Delta T = T = 2$, we show the contours of the spectral radii of the correction matrix $\boldsymbol{C}$ for the backward Euler SDC method for different values of p and α in Figs. 1 and 2. We observe from Fig. 1 that the convergence region gets smaller as the value of p increases. Also, we observe from Fig. 2 that the convergence region gets smaller as the value of fractional order α increases.

We now turn to the case when $|\lambda|$ is very large, namely, the stiff problem. In particular, for the above scalar problem, we show the spectral radii $\rho(\boldsymbol{C}(\lambda))$ of the correction matrix C with different values of p and α for the stiff case, i.e., $\lambda = -10000$, in Table 1. We observe that the spectral radius becomes larger as the number of Gaussian nodes increases or the value of the fractional order α increases. Moreover, we have from the table that there are some radii either close to or greater than 1.

We have from the analysis in the above subsection that if the spectral radius of the correction matrix $\boldsymbol{C}$ is greater than 1, i.e., $\rho(\boldsymbol{C}(\boldsymbol{L})) > 1$, the Neumann series expansion (2.23) is divergent, consequently, we fail to obtain a convergent numerical solution. Also, if $\rho(\boldsymbol{C}(\boldsymbol{L})) < 1$ but close to 1, then the SDC iterations still converges but very slowly. This is the so called order reduction phenomenon [14], which usually happens for stiff problems.

2.5. Accelerating the convergence of SDC method using GMRES

It has been shown in Subsection 2.3 that the SDC method is equivalent to solving the preconditioned system (2.25), and the numerical solution after k-th SDC iteration can be represented by the Neumann series expansion, i.e.,

$$\vec{u}^k - \vec{u}^0 = \vec{\delta}^0 + \boldsymbol{C}\vec{\delta}^0 + \boldsymbol{C}^2\vec{\delta}^0 + \cdots + \boldsymbol{C}^k\vec{\delta}^0.$$

The above equation encourages us to search for the optimal solution in the Krylov subspace $\boldsymbol{K}(\boldsymbol{C}, \vec{\delta}^0) = \mathrm{span}\{\vec{\delta}^0, \boldsymbol{C}\vec{\delta}^0, \ldots, \boldsymbol{C}^k\vec{\delta}^0\}$ by using the GMRES or other Krylov subspace based iterative methods for the linear system (2.25). In this work, we use the GMRES algorithm with restart, denoted by Re, to accelerate the convergence of the original SDC method. More details can be found in [14]. For the nonlinear problem, we use the Newton iteration method by using the implicit scheme based SDC method.

3. SDC method for fractional ODEs

In this section, we solve fractional ODEs by using the SDC or the GMRES-SDC method. We recall and consider the following fractional ODE:

$$ {}_0^C D_t^\alpha u(t) = F(t, u(t)), \quad t \in [0, T], \quad u(0) = u_0. \tag{3.1}$$

3.1. Linear problem

We begin by considering linear fractional ODEs, namely,

$$ F\big(t, u(t)\big) = \lambda u(t) + f(t). \tag{3.2}$$

We first present accuracy tests for both $\lambda = -1$ (non-stiff case) and $\lambda = -10000$ (stiff case) with a smooth right hand function (RHF) F (with respect to t). Note that in this case, the solution u is non-smooth in the classical sense.

Example 3.1. We consider the linear problem (3.1)-(3.2) with $F(t, u(t)) = \cos(t)$. In this case, we have $f(t) = \cos(t) - \lambda\, {}_0I_t^\alpha \cos(t), u_0 = 0$ and the exact solution is $u(t) = {}_0I_t^\alpha \cos(t)$.

Let $T = 2, \Delta T = 2$, and the tolerence and the total maximum number of iterations be 1.0×10^{-15} and 1000, respectively. We point out here that for Example 3.1, if not specified, we use the implicit Euler based scheme. We show the p convergence of the L^∞-error for both the non-stiff ($\lambda = -1$) and stiff ($\lambda = -10000$) cases with different values of α in Fig. 3(a). Here we use the SDC method for the non-stiff case and the GMRES-SDC method for the stiff case. Observe that spectral accuracy is obtained. This is in agreement with the estimate (2.18) in Theorem 2.1 since the RHF is smooth.

We also present the number of the SDC iterations for the stiff case ($\lambda = -10000$) in Table 2. We see that for $\alpha = 0.9$, as discussed in the above section, the spectral radii of the SDC method are close to 1 for large values of p resulting to a very large number of SDC iterations. Furthermore, we show the convergence procedure of the GMRES-SDC method with $p = 25$ and different values of the restart number in Figs. 3(b)-3(d) and different values of $\alpha = 0.3, 0.5, 0.7, 0.9, 1.0$. Observe that the convergence is faster when using larger values of the restart number. Moreover, the convergence is faster for smaller values of the fractional order. This is coincide with the observation in Table 1 showing that the smaller value of the fractional order is, the smaller of the spectral radius $\rho(\boldsymbol{C}(\lambda))$ is. Note that the original SDC procedures diverges for $\alpha = 0.9, 1.0$

Table 2: Example 3.1: Number of SDC iterations for the stiff case ($\lambda = -10000$).

α \ p	7	9	11	13	15	17	19	21
0.7	81	95	107	116	125	133	139	145
0.9	141	200	276	376	518	732	1000	1000

Table 3: Example 3.1: The L^∞-error and the relative residual of the GMRES-SDC iteration for $\lambda = -10000$ by using the implicit GMRES-SDC method with different pairs of (α, p). For each case, we fix the total number of iterations to be the value of p.

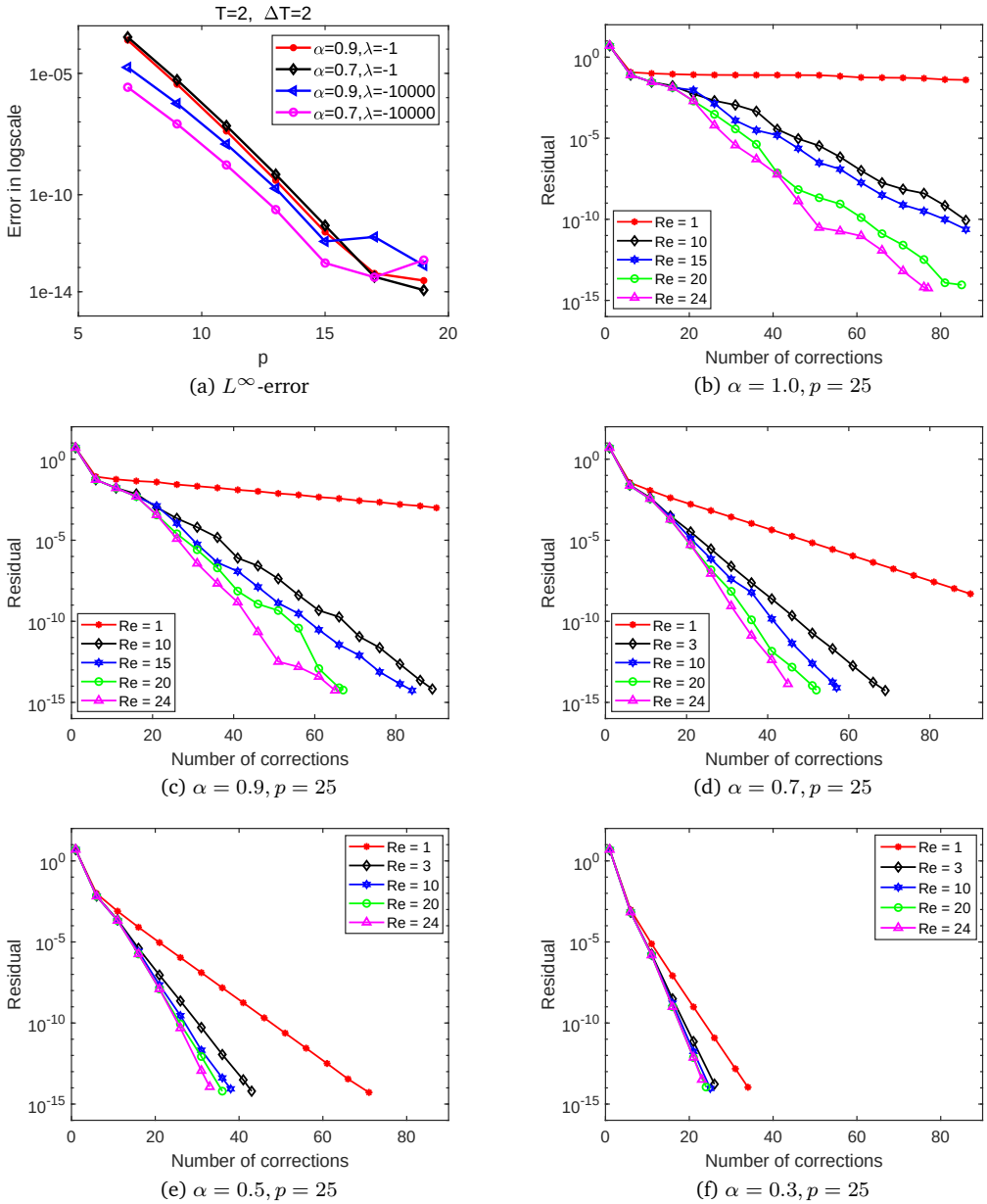
(α,p)	Re	Residual	Error	(α,p)	Re	Residual	Error
	0	6.0463e-03	1.2314e+00		0	9.6922e-04	1.3474e-01
	1	1.5866e-02	5.4090e-01		1	5.4012e-03	6.6295e-02
$(1.0, 25)$	3	2.8770e-04	2.9179e-03	$(0.9, 25)$	3	8.0014e-05	6.2747e-04
	5	2.4815e-05	2.6619e-04		15	1.1092e-06	6.4619e-06
	15	1.1860e-15	1.0035e-13		25	2.0989e-15	2.7537e-13
	0	1.3605e-04	1.2695e-02		0	1.5443e-05	8.9345e-04
	1	1.1096e-03	9.0454e-03		1	1.3367e-04	8.2341e-04
$(0.8, 25)$	3	9.8377e-06	8.8143e-05	$(0.7, 25)$	3	8.5468e-07	5.1033e-06
	15	5.9486e-08	6.0377e-07		15	3.2271e-09	2.4187e-08
	25	2.1556e-15	1.9052e-13		25	1.3452e-15	2.3663e-14

Table 4: Example 3.1: The L^∞-error and the relative residual of the GMRES-SDC iteration for $\lambda = -10$ by using the explicit GMRES-SDC with different pairs of (α, p). For each case, we fix the total number of iteration to be the value of p.

(α,p)	Re	Residual	Error	(α,p)	Re	Residual	Error
	0	2.3355e+09	1.8228e+11		0	7.2100e+16	4.6715e+18
	1	2.6250e-03	1.1541e-02		1	1.0753e-03	4.4384e-03
$(1.0, 15)$	3	9.7863e-05	3.9872e-04	$(0.9, 20)$	5	2.7418e-05	1.2931e-04
	5	7.1572e-05	3.8762e-04		10	3.9943e-07	2.1853e-06
	15	1.3716e-15	3.1455e-11		20	4.5961e-15	2.0969e-14
	0	5.7286e+23	3.4670e+25		0	4.5091e+59	2.2398e+61
	1	3.8503e-09	1.5734e-08		1	7.2448e-04	2.3136e-03
$(0.8, 40)$	4	1.2968e-12	6.5541e-12	$(0.7, 50)$	5	1.6102e-08	6.4783e-08
	10	4.1219e-14	2.6659e-13		10	1.4925e-10	9.3083e-10
	40	2.3042e-15	2.5420e-14		50	6.0183e-15	2.9521e-14

since the spectral radii are greater than 1. However, this issue is resolved by using the GMRES-SDC with restart number greater than 1. In addition, we present the residuals and L^∞ errors for different combinations of (α, p) with the same number of iteration for each combination in Tables 3 and 4, where we use in Table 3 the implicit scheme for $\lambda = -10000$ while we use in Table 4 the explicit scheme for $\lambda = -10$, respectively. We observe that even for a small value of λ, the original SDE diverges using the explicit scheme with some large values of p, so we can use the GMRES-SDC to resolve this issue.

We now consider problems with singular solutions, specifically, we consider the Mittag-Leffler solution problem. In this case, both the solution u and the RHF F have low regularity in the classical sense.



(a) L^∞-error (b) $\alpha = 1.0, p = 25$

(c) $\alpha = 0.9, p = 25$ (d) $\alpha = 0.7, p = 25$

(e) $\alpha = 0.5, p = 25$ (f) $\alpha = 0.3, p = 25$

Figure 3: Example 3.1: (a): Convergence of the L_∞ errors against the polynomial degree p with $T = 2$, $\Delta T = 2$ and different values of α for both non-stiff ($\lambda = -1$) and stiff ($\lambda = -10000$) cases. (b)-(f): Convergence of the residual of the implicit GMRES-SDC method with $p = 25$ and different values of α for $\lambda = -10000$ and different values of the restart number Re.

Example 3.2. Consider the problem (3.1)-(3.2) with $f(t) = 0$ and $\lambda \neq 0$. In this case, the exact solution is given by the Mittag-Leffler function [8]

$$u(t) = E_\alpha(-\lambda t^\alpha),$$

where

$$E_\alpha(t) = \sum_{l=0}^{\infty} \frac{t^l}{\Gamma(\alpha l + 1)}.$$

Here, we set the tolerance and the maximum iteration number for the SDC or the GMRES-SDC method to be 1.0×10^{-14} and 100, respectively, and we use the implicit Euler based scheme for this example. Again, we first test the accuracy. In particular, we show the convergence of the L^∞-error with $\alpha = 0.9$ at time $T = 200$ by using h- or p-refinement in Fig. 4 for both the non-stiff ($\lambda = -10000$) and the stiff ($\lambda = -1$) cases. Similar as in the previous example, we use the SDC scheme for the non-stiff case while we use the GMRES-SDC scheme with $Re = p - 1$ for the stiff case. Here, we let $p = 8$ for the h-refinement while we let $K = 200$ for the p-refinement. We observe that we only obtain algebraic convergence by using ether h- or p-refinement. Moreover, the convergence is slower when using the h-refinement. For the p-refinement, we have from Theorem 2.1 that we cannot obtain fast convergence since the regularity of F is very low. The convergence of the h-refinement also depends on the regularity of u or F.

To resolve this issue, we introduce a geometric mesh near $t = 0$ since we know that we only have the low regularity at the origin. In particular, we re-divide the first subdomain $[0, \Delta T] = [t_0, t_1]$ into a geometric mesh given by

$$t_{1,0} = t_0, \quad t_{1,k} = t_0 + \Delta T \cdot r^{K1-k}, \quad k = 1, 2, \ldots, K1. \tag{3.3}$$

To this end, we have total $K + K1 - 1$ subdomains, and we employ the same SDC or GMRES-SDC procedure on these subdomains. Let $r = 0.3$, the L^∞ convergence results at time $T = 200$ for the non-stiff case with $K1 = 12$ and for the stiff case with $K1 = 10$ are shown in Fig. 4. We have that the SDC method with geometric mesh delivers superior accuracy. This is a big advantage for the long time evolution since the total number of degrees of freedom increases only slightly.

Next, we show convergence of the GMRES-SDC method for the stiff case by using different values of restart number in Fig. 5; again, we observe that the convergence

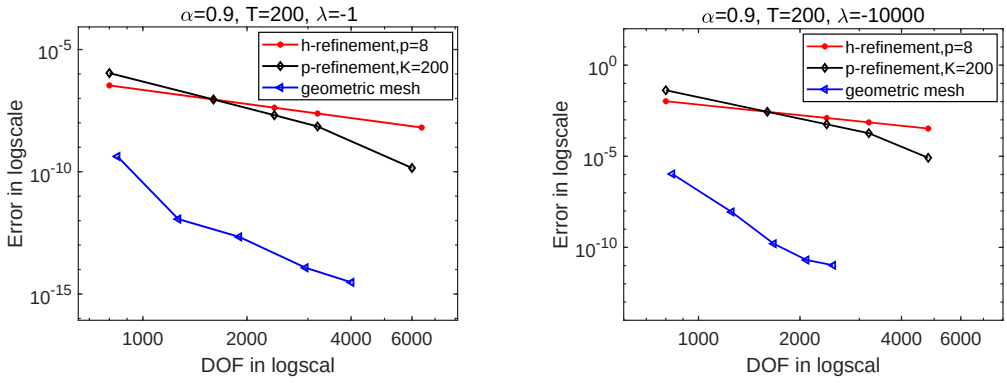


Figure 4: Example 3.2: Convergence of the L^∞-error by using $h - p$ refinement. We fix $K = 100$ when using the geometric mesh. $\Delta T = 1$. Left: $\lambda = -1, K1 = 12$, right: $\lambda = -10000, K1 = 10$.

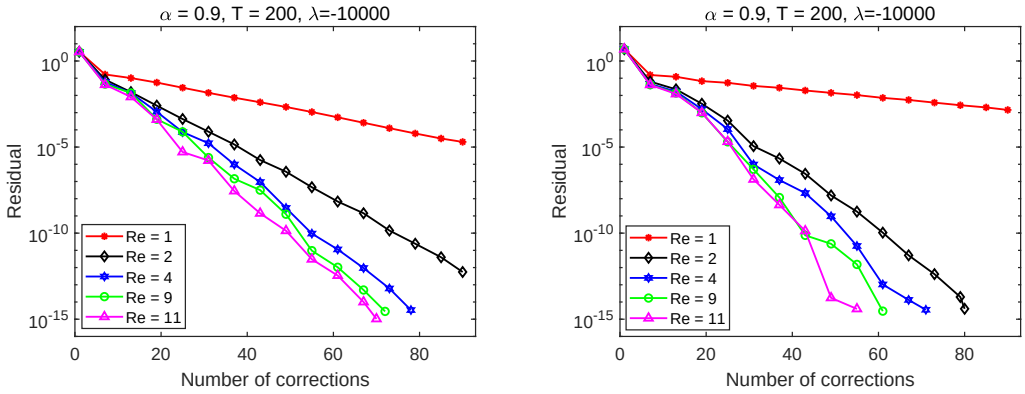


Figure 5: Example 3.2: Convergence of the residual of the GMRES-SDC iteration for $\lambda = -10000$ with different values of the restart number Re and $\alpha = 0.9$. $\Delta T = 1$. Left: $p = 12$, right: $p = 24$.

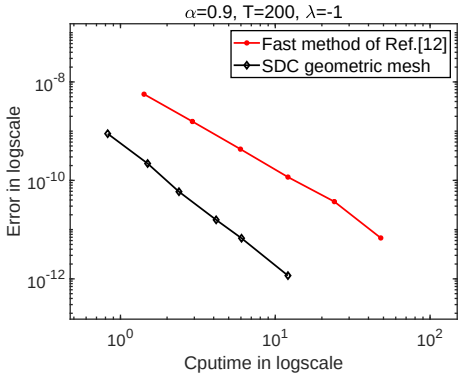


Figure 6: Example 3.2: Comparison of the efficiency, i.e., L^{∞}-error vs. CPU time (in second), between the one using the SDC method and the one using the fast method II developed in [12]. The code is implemented by using Matlab 2018b on the Thinkpad laptop with an Intel(R) Core(TM) i7-10710U CPU.

is accelerated by using GMRES with restart. To further illustrate the efficiency of the GMRES-SDC method, in Fig. 6 we compare the efficiency between the result obtained by using the GMRES-SDC method and the one obtained by using the fast method II developed in reference [12]. We briefly introduce the fast method II in Appendix B, and we take $m = 1$ and $\tau = [1/16, 1/32, 1/64, 1/128, 1/256, 1/512]$ (τ is the step of time) in the fast method II. Observe that to obtain the same accuracy, the CPU time by using the GMRES-SDC method is much less than the one by using the fast method II in [12]. This is due to two facts, the first one is that the requirement of the number of degrees of freedom (DOF) is significantly reduced by using the GMRES-SDC method. The second one is that the convergence of the GMRES-SDC method is very fast. We also mention here that from the point-view of the storage, the complexity of the storage of the fast method II in [12] is almost independent of the step-size while it is $\mathcal{O}(DOF)$ in the present algorithm. This means that the present method is more accurate and efficient than the fast method II given in [12].

3.2. Nonlinear problems

We now consider the nonlinear fractional differential equations. Let us first consider the following example (see also [12, Example 5.1])

Example 3.3. We consider the problem (3.1) with $F(t,u(t)) = -u(t)^3$. In this example, we fix $\alpha = 0.8, T = 100$ and $u_0 = 1$. The tolerance and the maximum number for the GMRES-SDC iteration are set to 10^{-3} and 100, respectively. The tolerance and the maximum number for the Newton iteration are set to 10^{-14} and 100, respectively. Again, we compare the accuracy between the result obtained by using the uniform mesh and the one obtained by using the geometric mesh, and show the results of the L^∞ relative errors for each moment t_j, $j = 1,\ldots,K$ in Fig. 7 for both h ($p = 25$) and p ($K = 100$) refinements, showing that we have much higher accuracy by using the geometric mesh. Here, we use the numerical solution with geometric mesh and

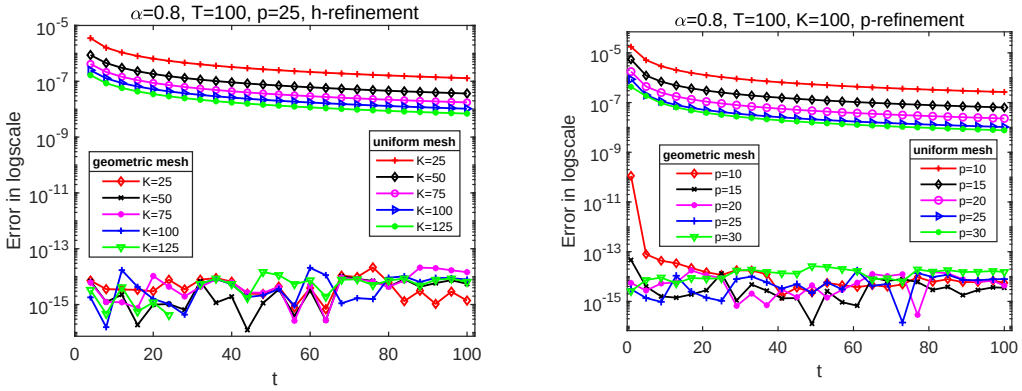


Figure 7: Example 3.3: The relative L^∞ errors at each point t_j using the GMRES-SDC method, $Re = p-1$. The upper curves are obtained by using the uniform mesh while the lower curves are obtained by using the geometric mesh. Left: h-refinement, $p = 25$. Right: p-refinement, $K = 100$.

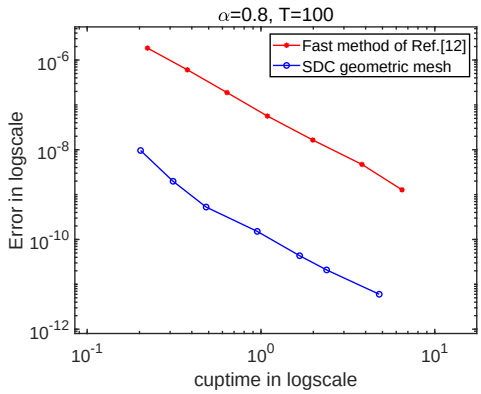


Figure 8: Example 3.3: Comparison of the efficiency, i.e., L^∞-error vs. CPU time (in second), between the one using the SDC method and the one using the fast method II developed in [12]. The code is implemented by using Matlab 2018b on the Thinkpad laptop with an Intel(R) Core(TM) i7-10710U CPU.

$p = 25, K = 1000$ as the reference solution. For the parameters used for the geometric mesh, we set $r = 0.3, K1 = 12$. Here, we use the implicit Euler based GMRES-SDC method and always take the value of restart number $Re = p-1$. Similarly, we compare the efficiency between the result obtained by using the GMRES-SDC method and the one obtained by using the fast method II given in reference [12] in Fig. 8, showing again that our method is more efficient than the fast method II in [12]. We take $m = 1$ and $\tau = [1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024]$ (τ is the step of time) for the fast method II.

Example 3.4. Consider in this example the following fractional nonlinear equation considered in [3]:

$$\begin{aligned} &\left({}_0^C D_t^\alpha\right)^2 x(t) - \epsilon\left(1 - x^2(t)\right) {}_0^C D_t^\alpha x(t) + x(t) = 0, \quad t \in [0, T], \\ &x(0) = x_0, \quad {}_0^C D_t^\alpha x(0) = y_0. \end{aligned}$$

Here ϵ is a non-negative constant and $x_0, y_0 \in \mathbb{R}$. For $\alpha = 1$; this equation reduces to the classical Van der Pol equation, which has a stable periodic solution. To solve the above equation, we first rewrite the above equation as a nonlinear fractional ODE system

$$\begin{aligned} &{}_0^C D_t^\alpha x(t) = y(t), \quad {}_0^C D_t^\alpha y(t) = \epsilon\big(1 - x^2(t)\big) y(t) - x(t), \\ &x(0) = x_0, \qquad\qquad y(0) = y_0. \end{aligned} \tag{3.4}$$

In the numerical experiments, we fix $\alpha = 0.8, \epsilon = 4, x_0 = 2, y_0 = 0$ and $T = 25$. The tolerance and the maximum number for the GMRES-SDC iteration are set to 10^{-3} and 100, respectively. The tolerance and the maximum number for the Newton iteration are set to 10^{-14} and 1000, respectively. Similarly as that in Example 3.2, we first show the convergences of the L^∞ errors for the $h - p$ refinement based on the uniform or geometric mesh in the left plot of Fig. 9. When we use the geometric mesh, we always take $r = 0.3, K1 = 12$. Since we do not have the exact solutions, we use the numerical

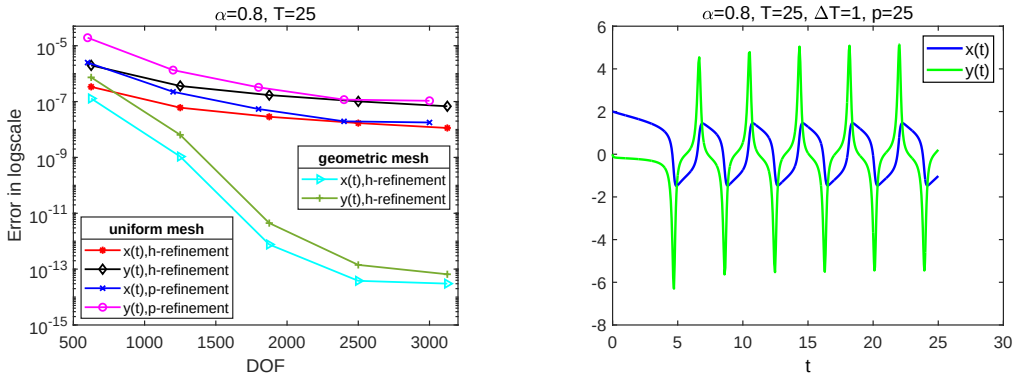


Figure 9: Example 3.4: Left: Convergence of the L^∞ error at time $T = 25$ with h-refinement ($p = 25$) and p-refinement ($K = 100$) based on uniform mesh, and h-refinement based on geometric mesh ($r = 0.3, K1 = 12, p = 25$). Right: Numerical solutions of $x(t)$ and $y(t)$.

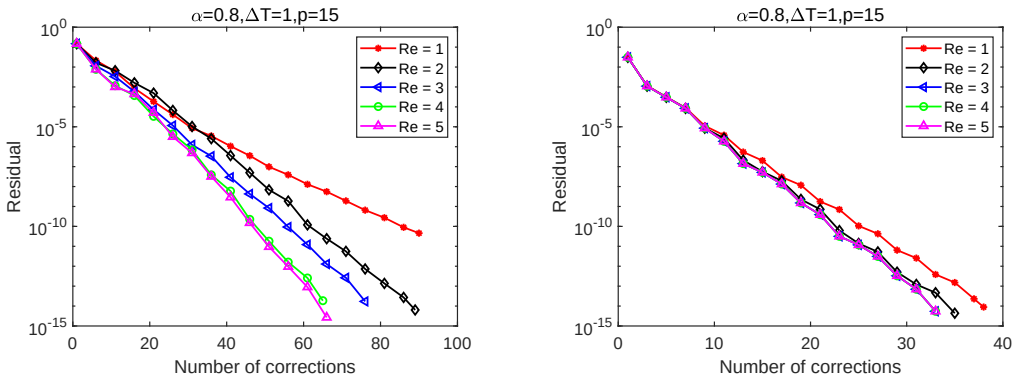


Figure 10: Example 3.4: Convergence of residual of the GMRES-SDC iteration for different values of the restart number Re. $\alpha = 0.8, p = 15$. Left: explicit GMRES-SDC. Right: implicit GMRES-SDC.

solution based on the geometric mesh with $p = 25, K = 300$ as the reference solution. Again, we see that the accuracy based on the geometric mesh is much higher than that based on the uniform mesh. The numerical solutions of $x(t)$ and $y(t)$ are shown in the right plot of Fig. 9. Then, in Fig. 10, we show the convergence of the residual with different values of the restart number for both the explicit and the implicit Euler based GMRES-SDC schemes. Here, we set $p = 15$, and we observe that increasing the value of the restart number does not help too much to accelerate the convergence of the implicit Euler based SDC method. This is due to the fact that the spectral radius of the implicit SDC is not close to 1. However, for the explicit Euler based SDC method, by using GMRES we accelerate the convergence of the iteration procedure.

4. SDC method for the space-time fractional phase field model

In this section, we employ the present GMRES-SDC method to solve the space-time fractional partial differential equations. In particular, let $\Omega = [0, 2\pi]^2$, we consider the following two-dimensional fractional phase field model:

$$\begin{cases} {}_0^C D_t^\alpha u(x,t) + \gamma(-\Delta)^\beta \left(-\varepsilon \Delta u(x,t) + \dfrac{1}{\varepsilon} f\big(u(x,t)\big) \right) = 0, \quad (x,t) \in \Omega \times (0,T], \\ u(x,0) = u_0(x), \quad x \in \Omega \end{cases} \tag{4.1}$$

with periodic boundary conditions, ε is a positive constant, $x = (x_1, x_2), u_0 \in L_\infty(\Omega) \cap H^1_{per}(\Omega)$, and $f(u) = u^3 - u$. The choice of $\alpha = 1, \beta = 1$ corresponds to the classical Cahn-Hilliard (CH) equation while the choice of $\alpha = 1, \beta = 0$ reduces to the Allen-Cahn (AC) equation. We define the fractional Laplacian by using the spectral definition based on the Fourier decomposition; see references [1, 2] for more details.

We use the Fourier-Galerkin approximation for the space discretization of the problem (4.1). Let $C^\infty_{per}(\Omega)$ be the set of all restrictions onto $\Omega = (0, 2\pi)^2$ of all real-valued, 2π-periodic, C^∞-functions on $\mathbb{R}^2$. For each $s \geq 0$, let $H^s_{per}(\Omega)$ be the closure of $C^\infty_{per}(\Omega)$

in the usual Sobolev norm $\|\cdot\|_s$. The weak form of (4.1) is to find $u \in H_{per}^{1+\beta}(\Omega)$, such that

$$\begin{aligned}&\left({}_0^C D_t^\alpha u(t), v\right) + \left(\varepsilon^2(-\Delta)^{\frac{(1+\beta)}{2}} u(t), (-\Delta)^{\frac{(1+\beta)}{2}} v\right)\\ &\quad + \left(f\big(u(t)\big), (-\Delta)^\beta v\right) = 0, \quad \forall v \in H_{per}^{1+\beta}(\Omega). \end{aligned} \tag{4.2}$$

Let

$$\mathbb{X}_N = \operatorname{span}\left\{e^{ikx_1+ilx_2},\ -N \le k,\, l \le N\right\}.$$

The Fourier-Galerkin approximation for (4.2) consists of finding $u_N \in \mathbb{X}_N$ such that

$$\begin{aligned}&\left({}_0^C D_t^\alpha u_N(t), v\right) + \left(\varepsilon^2(-\Delta)^{\frac{(1+\beta)}{2}} u_N(t), (-\Delta)^{\frac{(1+\beta)}{2}} v\right)\\ &\quad + \left(f\big(u_N(t)\big), (-\Delta)^\beta v\right) = 0, \quad \forall v \in \mathbb{X}_N \end{aligned} \tag{4.3}$$

with initial condition $u_N(0) = \Pi_N u_0$, where Π_N is the usual L^2-projection operator in $\mathbb{X}_N$.

Let

$$u_N(x,t) = \sum_{k,l=-\frac{N}{2}}^{\frac{N}{2}} \hat{u}_{kl}(t) e^{ikx_1+ilx_2},$$

and take $v = e^{ipx_1+iqx_2}, p, q = -\frac{N}{2}, \ldots, \frac{N}{2}$. According to the definition of the fractional Laplacian, we arrive at the following fractional ordinary differential system:

$${}_0^C D_t^\alpha \hat{u}_{kl}(t) + \varepsilon^2 \left(k^2+l^2\right)^{1+\beta} \hat{u}_{kl}(t) + \left(k^2+l^2\right)^\beta \hat{f}_{kl}(t) = 0, \quad k,l = -\frac{N}{2}, \ldots, \frac{N}{2}, \tag{4.4}$$

where $\hat{f}_{kl}(t), k, l = -\frac{N}{2}, \ldots, \frac{N}{2}$, are the Fourier coefficients of $f(u_N)$. We then employ the GMRES-SDC method to solve the above ODE system.

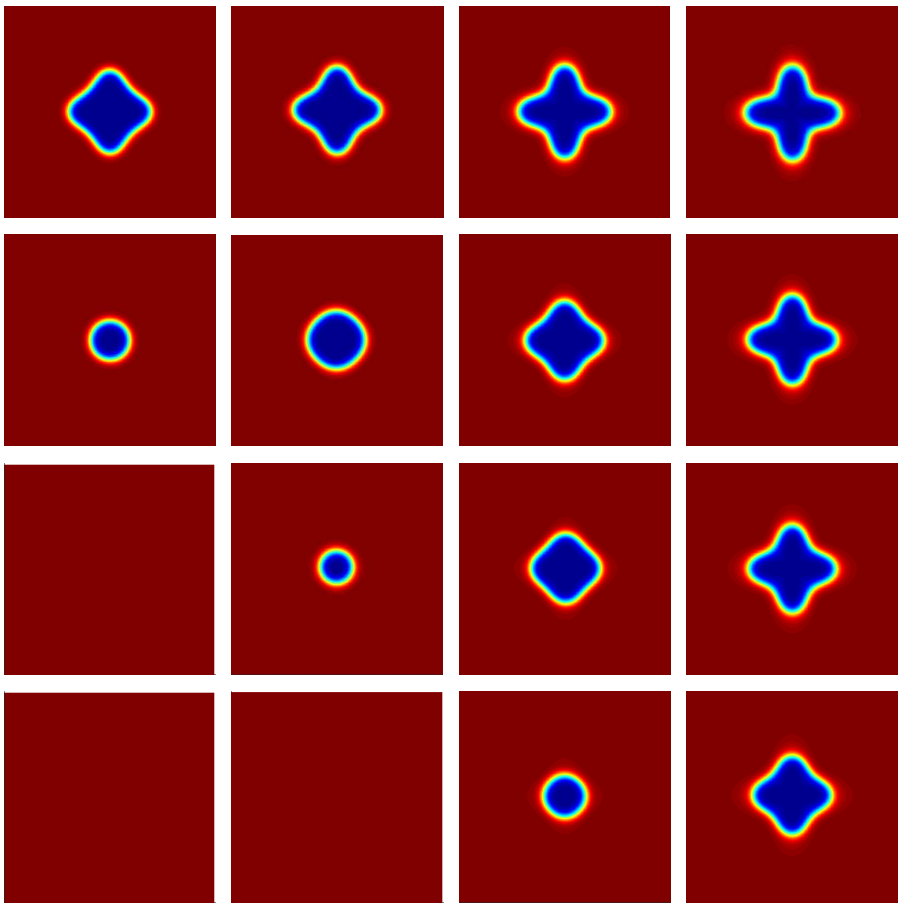
We now consider the following initial condition given in [30, Section 5.1].

Example 4.1. Consider the fractional phase field model (4.1) with the following initial condition:

$$u_0(x_1,x_2) = \tanh\left(\frac{1}{\sqrt{2}\varepsilon}\left(\sqrt{x_1^2+x_2^2} - \frac{1}{4} + \frac{1-\cos\left(4\arctan(x_2/x_1)\right)}{16}\right)\right). \tag{4.5}$$

We set $\varepsilon = 0.05, \Delta T = 1, Re = 30, tol = 10^{-9}, maxit = 1000, N = 128, p = 30$. The tolerance and the maximum number for the GMRES-SDC iteration are set to 10^{-3} and 100, respectively. The tolerance and the maximum number for the Newton iteration are set to 10^{-9} and 1000, respectively. We mainly consider two cases:

- Case I: Fractional Allen-Cahn equation (FACE), i.e., $\beta = 0$. In this case, we set $\gamma = 0.05$.



(a) $(\alpha, \beta) = (1.0, 0.0)$ (b) $(\alpha, \beta) = (0.8, 0.0)$ (c) $(\alpha, \beta) = (0.5, 0.0)$ (d) $(\alpha, \beta) = (0.3, 0.0)$

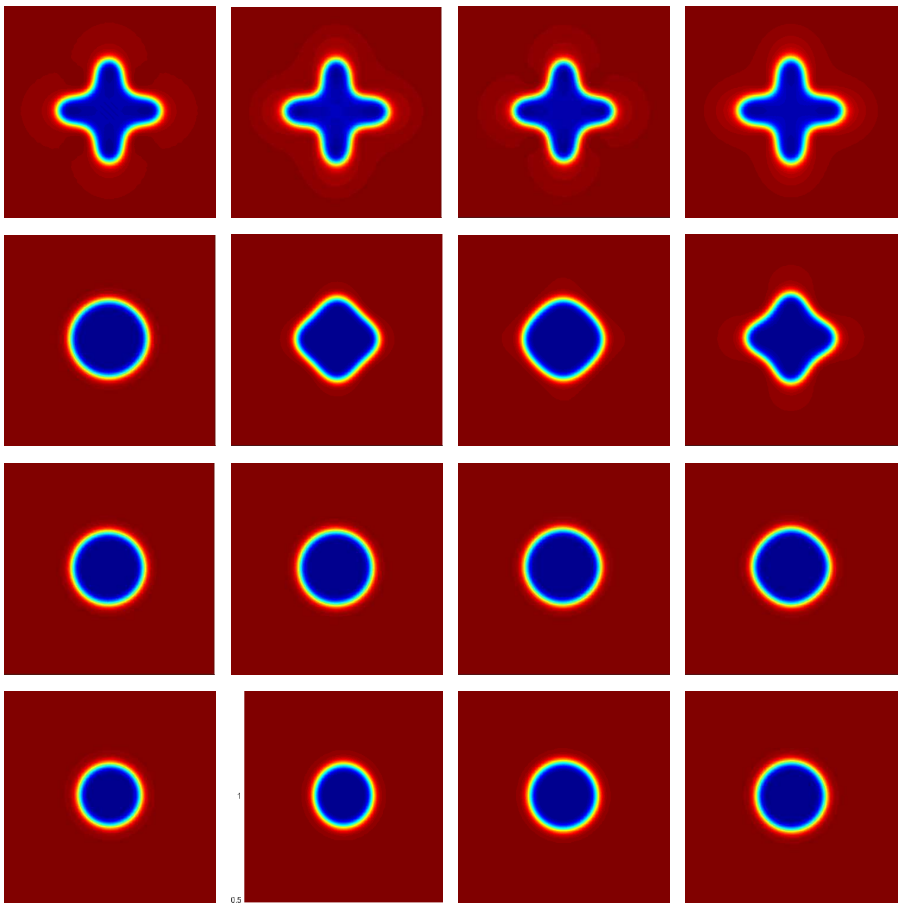
Figure 11: Example 4.1, Case I: Evolution of solutions of FACE with initial condition (4.5) for different values of fractional order α. From top to bottom, $t = 10, 50, 100, 400$.

- Case II: Fractional Cahn-Hilliard equation (FCHE), i.e., $0 < \beta \leq 1$. In this case, we set $\gamma = 0.0025$.

We show the evolution of solutions for different values of fractional order α for both FACE ($\beta = 0$) and FCHE ($\beta = 0.8, 1.0$) in Figs. 11 and 12, respectively. We observe that the solutions are in agreement with the ones in [30]. We also show the residual at time $T = 1$ with different values of the restart number for FACE and FCHE in Figs. 13 and 14, respectively. We see that the iteration procedure is accelerated by using GMRES.

5. Summary

In this work, we have extended the idea of [14] for classical ordinary differential equations (ODEs) to fractional differential equations (FDEs). We first reformulated the fractional ODE into the Picard integral form divided into two terms: a non-local history



(a) $(\alpha, \beta) = (1.0, 1.0)$ (b) $(\alpha, \beta) = (1.0, 0.8)$ (c) $(\alpha, \beta) = (0.8, 1.0)$ (d) $(\alpha, \beta) = (0.8, 0.8)$

Figure 12: Example 4.1, Case II: Evolution of solutions of FCHE with initial condition (4.5) for different values of fractional order (α, β). From top to bottom, $t = 1, 10, 50, 400$.

term and a local term. The non-local history term is computed by using the hybrid method while a spectral deferred correction (SDC) method is developed for the local term and the correction matrix of the SDC method is derived. The SDC iteration has slow convergence or diverges if the spectral radius of the correction matrix is close to or greater than 1. We overcame this issue by accelerating the convergence of the SDC method with the generalized minimal residual algorithm. We numerically analyzed the accelerated SDC method by considering both stiff and non-stiff linear problems showing that the accelerated SDC method is more efficient than the original SDC method. Moreover, we obtained high accuracy by applying the geometric mesh near the origin for problems with singular solutions. Furthermore, the use of geometric mesh is advantageous for the long time evolution, since we only need a slight increase in the number of degrees of freedom. We employed the present accelerated SDC method to nonlinear fractional ODEs and fractional phase field models and demonstrated the effectiveness

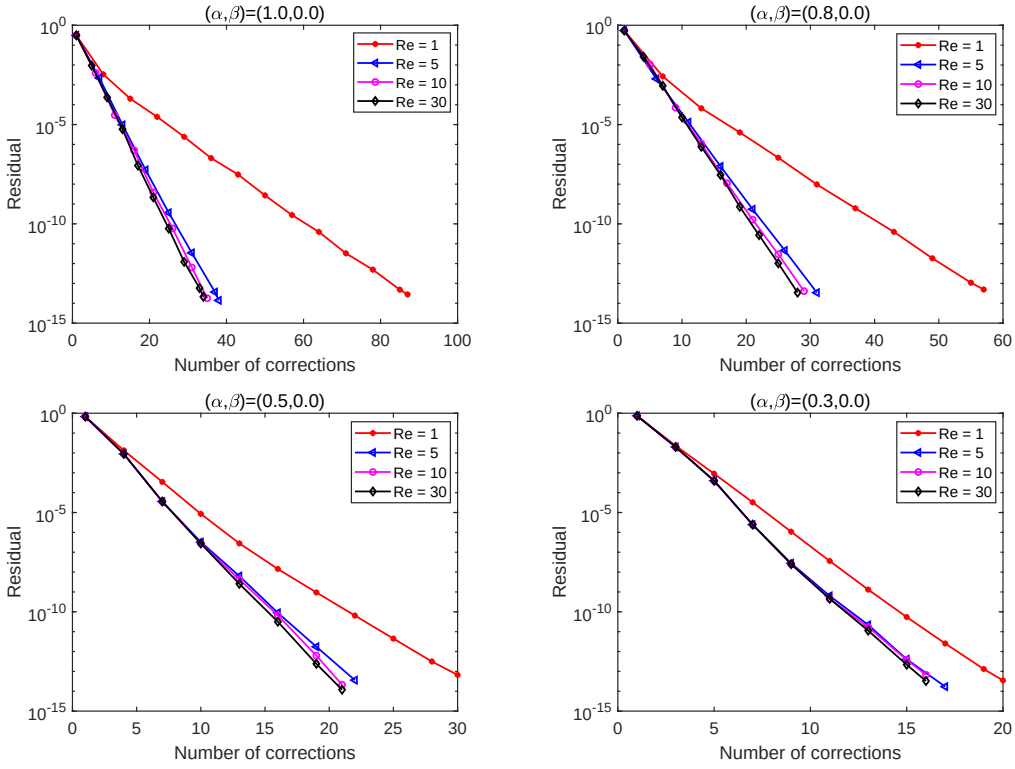


Figure 13: Example 4.1, case I: Convergence of the residual of the GMRES-SDC iteration for different values of the restart numberRe for FACE.

of the accelerated SDC method and the use of the geometric mesh. We also compared the results obtained by the present method with the one obtained by existing methods, i.e., the fast method II in [12]. We observed that to retain high accuracy requires much less degree of freedom and the CPU time is significantly reduced using our proposed method.

Appendix A. The computation of the non-local history term $u_a(t)$

In this section, we introduce the computation of the history term $u_a(t)$ given in (2.3). Assume the current interval is $[t_j, t_{j+1}]$, i.e., $a = t_j, b = t_{j+1}$, by using (2.6), we approximate $u_a(t)$ by the following:

$$u_a(t) \approx u_0 + \frac{1}{\Gamma(\alpha)} \int_0^{t_j} \frac{F_p\big(\tau, u(\tau)\big)}{(t-\tau)^{1-\alpha}} d\tau = u_0 + \frac{1}{\Gamma(\alpha)} \sum_{l=0}^{j-1} \int_{t_l}^{t_{l+1}} \frac{F_p\big(\tau, u(\tau)\big)}{(t-\tau)^{1-\alpha}} d\tau$$

$$= u_0 + \frac{1}{\Gamma(\alpha)} \sum_{l=0}^{j-1} \int_{t_l}^{t_{l+1}} (t-\tau)^{\alpha-1} \sum_{m=0}^{p} F_m^k \hat{h}_m(\tau) d\tau$$

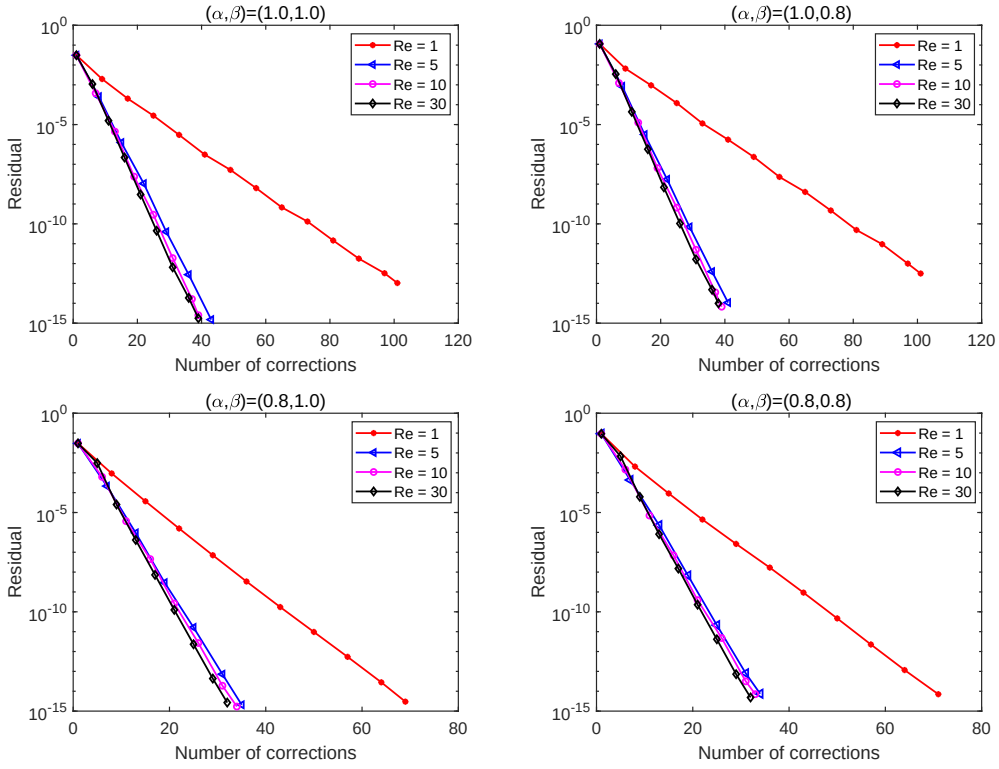


Figure 14: Example 4.1, case II: Convergence of the residual of the GMRES-SDC iteration for different values of the restart number Re for FCHE.

$$= u_0 + \frac{\Delta T}{2\Gamma(\alpha)} \sum_{l=0}^{j-1} \int_{-1}^{1} \left(t - t_l - \frac{\Delta T}{2}(1+r) \right)^{\alpha-1} \sum_{m=0}^{p} F_m^k h_m(r) dr, \qquad \text{(A.1)}$$

where $h_m(r)$ is the Gauss Lagrange interpolation polynomial based on the $p+1$ Legendre-Gauss-Lobatto points. We expand F_p in terms of Jacobi polynomials

$$\sum_{m=0}^{p} F_m^k h_m\big(\tau(r)\big) = \sum_{n=0}^{p} \hat{F}_n^k J_n^{\tilde{a},\tilde{b}}(r),$$

and obtain the coefficients $\{\hat{F}_n^k\}_{n=0}^p$ by using the forward discrete transform (cf. [28, Theorem 3.28]), here $J_n^{\tilde{a},\tilde{b}}(\tau)$ is the n-th order Jacobi polynomial of index $(\tilde{a},\tilde{b})$. Therefore, (A.1) becomes

$$u_a(t) = u_0 + \frac{\Delta T}{2\Gamma(\alpha)} \sum_{l=0}^{j-1} \sum_{n=0}^{p} \left[\int_{-1}^{1} \left(t - t_l - \frac{\Delta T}{2}(1+r) \right)^{\alpha-1} J_n^{\tilde{a},\tilde{b}}(r) dr \right] \hat{F}_n^k.$$

The integrals in the above equations can be computed by a high accurate hybrid approach originally developed in [5]. In particular, we use the three-term-recurrence relation when $j-l$ is small while use the Gauss quadrature when $j-l$ is large.

Appendix B. The fast method II

In this section, we briefly introduce the algorithm of the fast method II in reference [12] for solving the fractional ODE (3.1). In particular, it extends the idea of Lubich's FLMMs (see [20]) for integral operators to fractional derivative operators. Let U_n be the numerical solution of (3.1) at $t = t_n$, and the fully implicit fast method of (3.1) is given as follows:

$$ {}_F D_\tau^{\alpha,\gamma,m,n} U = F(t_n, U_n), \quad U_0 = u_0, \tag{B.1} $$

where ${}_F D_\tau^{\alpha,\gamma,m,n}$ is defined as

$$ {}_F D_\tau^{\alpha,\gamma,m,n} U = \tau^{-\alpha} \sum_{k=0}^{n} \omega_{n-k}^{\alpha} (U_k - U_0) + \tau^{-\alpha} \sum_{k=1}^{m} \omega_{n,k}^{\alpha} (U_k - U_0), $$

where τ is the step size, $t_k = k\tau$ is the grid point, $\gamma = (\gamma_1, \gamma_2, \ldots)$, $\gamma_{j+1} > \gamma_j > 0$, and the quadrature weights ω_k^α are chosen such that ${}_F D_\tau^{\alpha,\gamma,m,n} U$ is a stable approximation of $[{}_0^C D_t^\alpha u(t)]_{t=t_n}$. Once the quadrature weights ω_k^α are given, the starting weights $\omega_{n,k}^\alpha (1 \le k \le m)$ are chosen such that ${}_F D_\tau^{\alpha,\gamma,m,n} U = [{}_0^C D_t^\alpha u(t)]_{t=t_n}$ for $u(t) = t^{\gamma_j}, 1 \le j \le m$. We can obtain the convolution quadrature weights ω_k^α by the generating functions, see [20, Table 1]. A fast calculation for calculating the convolution quadrature coefficients ω_k^α is derived based on a globally uniform approximation of the trapezoidal rule for the integral on the real line. Please see the reference [12] for more details.

Acknowledgments

Z. Mao was supported by the Fundamental Research Funds for the Central Universities (Grant 20720210037), G. E. Karniadakis was supported by the MURI/ARO on Fractional PDEs for Conservation Laws and Beyond: Theory, Numerics and Applications (Grant W911NF-15-1-0562), and X. Chen was supported by the Fujian Provincial Natural Science Foundation of China (Grants 2022J01338, 2020J01703).

References

[1] M. AINSWORTH AND Z. MAO, *Analysis and approximation of a fractional Cahn-Hilliard equation*, SIAM J. Numer. Anal. 55 (2017), 1689–1718.

[2] M. AINSWORTH AND Z. MAO, *Analysis and approximation of gradient flows associated with a fractional order Gross-Pitaevskii free energy*, Commun. Appl. Math. Comput. 1 (2019), 5–19.

[3] D. BAFFET, *A Gauss-Jacobi kernel compression scheme for fractional differential equations*, J. Sci. Comput. 79 (2019), 227–248.

[4] J. CAO AND C. XU, *A high order schema for the numerical solution of the fractional ordinary differential equations*, J. Comput. Phys. 238 (2013), 154–168.

[5] F. CHEN, Q. XU, AND J. S. HESTHAVEN, *A multi-domain spectral method for time-fractional differential equations*, J. Comput. Phys. 293 (2015), 157–172.

[6] S. CHEN, J. SHEN, AND L.-L. WANG, *Generalized Jacobi functions and their applications to fractional differential equations*, Math. Comp. 85 (2016), 1603–1638.
[7] Y. CHEN, X. LI, T. TANG, *A note on Jacobi spectral-collocation methods for weakly singular Volterra integral equations with smooth solutions*, J. Comput. Math. (2013), 47–56.
[8] K. DIETHELM AND N. J. FORD, *Analysis of fractional differential equations*, J. Math. Anal. Appl. 265 (2002), 229–248.
[9] A. DUTT, L. GREENGARD, AND V. ROKHLIN, *Spectral deferred correction methods for ordinary differential equations*, BIT 40 (2000), 241–266.
[10] G.-H. GAO, Z.-Z. SUN, AND H.-W. ZHANG, *A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications*, J. Comput. Phys. 259 (2014), 33–50.
[11] R. GORENFLO AND F. MAINARDI, *Fractional Diffusion Processes: Probability Distributions and Continuous Time Random Walk*, in: *Processes with Long-Range Correlations*, Springer, 2003, 148–166.
[12] L. GUO, F. ZENG, I. TURNER, K. BURRAGE, AND G. E. KARNIADAKIS, *Efficient multistep methods for tempered fractional calculus: Algorithms and simulations*, SIAM J. Sci. Comput. 41 (2019), A2510–A2535.
[13] E. HAIRER, C. LUBICH, AND M. SCHLICHTE, *Fast numerical solution of nonlinear Volterra convolution equations*, SIAM Journal on Scientific and Statistical Computing, 6 (1985), 532–541.
[14] J. HUANG, J. JIA, AND M. MINION, *Accelerating the convergence of spectral deferred correction methods*, J. Comput. Phys. 214 (2006), 633–656.
[15] B. JIN, B. LI, AND Z. ZHOU, *Correction of high-order BDF convolution quadrature for fractional evolution equations*, SIAM J. Sci. Comput. 39 (2017), A3129–A3152.
[16] X. LI, Z. MAO, N. WANG, F. SONG, H. WANG, AND G. E. KARNIADAKIS, *A fast solver for spectral elements applied to fractional differential equations using hierarchical matrix approximation*, Comput. Methods Appl. Mech. Engrg. 366 (2020), 113053.
[17] X. LI AND C. XU, *A space-time spectral method for the time fractional diffusion equation*, SIAM J. Numer. Anal. 47 (2009), 2108–2131.
[18] Y. LIN, X. LI, AND C. XU, *Finite difference/spectral approximations for the fractional cable equation*, Math. Comp. 80 (2011), 1369–1396.
[19] Y. LIN AND C. XU, *Finite difference/spectral approximations for the time-fractional diffusion equation*, J. Comput. Phys. 225 (2007), 1533–1552.
[20] C. LUBICH, *Discretized fractional calculus*, SIAM J. Math. Anal. 17 (1986), 704–719.
[21] C. LUBICH, I. SLOAN, AND V. THOMÉE, *Nonsmooth data error estimates for approximations of an evolution equation with a positive-type memory term*, Math. Comp. 65 (1996), 1–17.
[22] Y. LUCHKO, *The wright function and its applications*, A. Kochubei, Yu. Luchko (Eds.), Handbook of Fractional Calculus with Applications, 1 (2019), 241–268.
[23] C. LV, M. AZAIEZ, AND C. XU, *Spectral deferred correction methods for fractional differential equations*, Numer. Math. Theor. Meth. Appl. 11 (2018), 729–751.
[24] Z. MAO, Z. LI, AND G. E. KARNIADAKIS, *Nonlocal flocking dynamics: Learning the fractional order of pdes from particle simulations*, Commun. Appl. Math. Comput. 1 (2019), 597–619.
[25] Z. MAO AND J. SHEN, *A semi-implicit spectral deferred correction method for two water wave models with nonlocal viscous term and numerical study of their decay rates*, Sci. China Math. 8 (2015), 1153–1168.
[26] R. METZLER AND J. KLAFTER, *The random walk's guide to anomalous diffusion: a fractional dynamics approach*, Phys. Rep. 339 (2000), 1–77.

[27] W. Qu, N. Brandon, D. Chen, J. Huang, and T. Kress, *A numerical framework for integrating deferred correction methods to solve high order collocation formulations of ODEs*, J. Sci. Comput. 68 (2016), 484–520.
[28] J. Shen, T. Tang, and L. Wang, *Spectral Methods: Algorithms, Analysis and Applications*, Springer, 2011.
[29] Z.-Z. Sun and X. Wu, *A fully discrete difference scheme for a diffusion-wave system*, Appl. Numer. Math. 56 (2006), 193–209.
[30] T. Tang, H. Yu, and T. Zhou, *On energy dissipation theory and numerical stability for time-fractional phase-field equations*, SIAM J. Sci. Comput. 41 (2019), A3757–A3778.
[31] J. Xin, J. Huang, W. Zhao, and J. Zhu, *A spectral deferred correction method for fractional differential equations*, in: *Abstract and Applied Analysis*, Vol. 2013, Hindawi, 2013.
[32] Y. Yan, Z.-Z. Sun, and J. Zhang, *Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations: a second-order scheme*, Commun. Comput. Phys. 22 (2017), 1028–1048.
[33] Y. Yu, P. Perdikaris, and G. E. Karniadakis, *Fractional modeling of viscoelasticity in 3d cerebral arteries and aneurysms*, J. Comput. Phys. 323 (2016), 219–242.
[34] M. Zayernouri, and G. E. Karniadakis, *Fractional Sturm–Liouville eigen-problems: theory and numerical approximation*, J. Comput. Phys. 252 (2013), 495–517.
[35] F. Zeng, I. Turner, and K. Burrage, *A stable fast time-stepping method for fractional integral and derivative operators*, J. Sci. Comput. 77 (2018), 283–307.