

A Self-Singularity-Capturing Scheme for Fractional Differential Equations

Jorge L. Suzuki^{a,b} and Mohsen Zayernouri^{a,c}

^aDepartment of Mechanical Engineering, Michigan State University, East Lansing, MI, USA

^bDepartment of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI, USA

^cDepartment of Probability and Statistics, Michigan State University, East Lansing, MI, USA

ARTICLE HISTORY

Compiled April 30, 2020

ABSTRACT

We develop a two-stage computational framework for robust and accurate time-integration of multi-term linear/nonlinear fractional differential equations. In the first stage, we formulate a self-singularity-capturing scheme, given available/observable data for diminutive time, experimentally obtained or sampled from an approximate numerical solution utilizing a fine grid nearby the initial time. The fractional differential equation provides the necessary knowledge/insight on how the hidden singularity can bridge between the initial and the subsequent short-time solution data. In the second stage, we utilize the multi-singular behavior of solution in a variety of numerical methods, without resorting to making any *ad-hoc*/uneducated guesses for the solution singularities. Particularly, we employed an implicit finite-difference method, where the captured singularities, in the first stage, are taken into account through some Lubich-like correction terms, leading to an accuracy of order $\mathcal{O}(\Delta t^{3-\alpha})$. We show that this novel framework can control the error even in the presence of strong multi-singularities.

KEYWORDS

multi-fractal singularities; random singularities; correction terms; short-time data; long-time integration; automated algorithms

1. Introduction

Fractional differential equations (FDEs) have been successfully applied in diverse problems that present the fingerprint of power-laws/heavy-tailed statistics, such as visco-elastic modeling of bio-tissues [7, 34–37], cell rheology behavior [12], food preservatives [20], complex fluids [21], visco-elasto-plastic modeling for power-law-dependent stresses/strains [17, 51], earth sciences [62], anomalous diffusion in SephadexTM gels [28], among others.

The increasing number of works involving fractional modeling in the past two decades is largely due to the development of efficient numerical schemes for fractional-order/partial differential equations (FDEs/FPDEs). Starting with Lubich's early works [31, 32] on finite-difference schemes, and followed by several developments, *e.g.*, on discretizing Burgers' equation [47], fractional Adams methods for nonlinear

problems [10, 11, 58], and fractional diffusion processes [15, 16, 29, 48]. Also, other classes of global methods were developed, such as spectral methods for FDEs/FPDEs [23, 25, 39, 42, 43, 49, 55–57], distributed-order differential equations (DODEs) [22, 24, 41], fractional subgrid-scale modeling [40] and also mesh-free schemes, such as the improved singular boundary method (SBM) developed by Chen and Gu [6]. Of particular interest, we highlight a family of fast convolution schemes for time-fractional integrals/derivatives, initially developed by Lubich [33]. In such schemes, instead of a direct discretization by splitting the fractional operator in N uniform time-integration intervals, it is split in exponentially increasing intervals, reducing the overall computational complexity from $\mathcal{O}(N^2)$ to $\approx \mathcal{O}(N \log N)$. Furthermore, the power-law kernels are computed through a convolution quadrature using complex integration contours [18, 27, 33, 45, 54]. The storage requirements are also reduced from $\mathcal{O}(N)$ to $\approx \mathcal{O}(\log N)$. Recently, Zeng *et al.* [59] developed an improved version of the scheme using a Gauss-Laguerre convolution quadrature, which utilizes real-valued integration contours. In the same work, the authors also addressed the short memory principle and applied Lubich’s correction terms [32], leading to a more accurate and stable fast time-stepping scheme. We also refer readers to other classes of fast schemes for time-fractional operators that make use of the resulting matrix structures [5], kernel compression [1], and divide-and-conquer fast finite-difference schemes [14].

There have also been works on ‘tunable accuracy’, spectral collocation methods for FDEs/FPDEs that utilize singular basis functions [30, 56, 60]. Zayernouri and Karniadakis [56] developed an exponentially accurate spectral collocation method utilizing fractional Lagrange basis functions, given by the product of a singular term with interpolation parameter μ and a smooth part given by Lagrange interpolants. Later, Zeng *et al.* [60] generalized the scheme for variable-order FDEs/FPDEs with endpoint singularities and demonstrated that a proper tuning of the μ parameter enhanced the accuracy of the numerical solutions. Lischke *et al.* [30] developed a Laguerre Petrov-Galerkin method for multi-term FDEs with tunable accuracy and linear computational complexity. Despite the high precision obtained by fine-tuning the bases in the developed works, such numerical accuracy is extremely sensitive to the ‘single’-singularity basis parameter μ and no self-capturing scheme was developed to find its correct, application-specific value.

The accuracy and efficiency of the aforementioned numerical schemes strongly depend on the regularity of the solution. To address such problem, a correction method was introduced by Lubich [32], which utilizes M correction terms with singular powers σ_k , with $k = 1, \dots, M$, given regularity assumptions for $u(t)$. The corresponding correction weights are obtained through the solution of a Vandermonde-type linear system of size M . Given a convenient choice of singular powers, some developed schemes were able to attain their theoretical accuracy with non-smooth solutions [59, 61]. However, such works utilize *ad-hoc* choices of σ_k , even without prior knowledge of the regularity of the solution. According to Zeng *et al.* [59], for such cases, a reasonable choice would be $\sigma_k = 0.1k$, which could improve the numerical accuracy when strong singularities are present, and without loss of accuracy when there is enough regularity for $u(t)$. However, such procedure could still lead to arbitrary choices of σ_k far from the true singularities of the solution. There are higher chances of approximating the singularities by increasing the number of correction terms, but using $M > 9$ terms induces a large condition number on the Vandermonde system (see Figure 2a), and consequently large residuals, leading to large errors when computing the correction weights, which are propagated to the discretized fractional operator [9].

Regarding the numerical solution of FDEs with correction terms, let N be the

total number of time-steps with size Δt . For the first M time-steps, prior information of the numerical solution for $u(t)$ is required *see (25)*. The procedure proposed in [59] involves the solution of a sub-problem with time domain $t \in (0, M\Delta t]$ using a time-step size $\tau = \Delta t^2$ and one correction term to obtain the numerical solution u_n^N for $n = 1, \dots, M$. However, the choice of smaller step size τ even for a short time might be too expensive, and the numerical solution for the M initial time-steps might experience total loss of accuracy in the presence of strong singularities, not captured by the single guess of correction term. Such deficiencies outline the need for **I)** Capturing the multi-singularities of $u(t)$ and **II)** Efficient and accurate schemes that incorporate all captured singularities for time-integration of the (most crucial) initial M time-steps.

The main contribution of the present work is to develop a two-stage data-infused computational framework for accurate time-integration of FDEs. In Stage-I, we formulate a self-singularity-capturing framework where:

- The scheme utilizes available data for initial diminutive time (application-oriented), and self-captures/determines multi-singular behaviors of the solution through the knowledge introduced by the FDE and its corresponding fractional operators. *No knowledge of the analytical solution is required a-priori*, and some of the numerical tests throughout this work sample from analytical solution points only for verification purposes.
- We develop a new (finite-difference based) algorithm for automatic determination of the underlying power-law singularities nearby the initial data, employing gradient descent optimization. The singularities are introduced through Lubich's correction method [32].
- We introduce the capturing scheme for M correction terms and construct a hierarchical, self-capturing framework. We test the framework for the particular case of up to $S = 3$ singularities and $M = 3$ correction terms.
- The self-capturing procedure makes use of two stopping criteria for the error minimization, namely ϵ (for solution error) and ϵ_1 (for gradient norm error). Here, numerical convergence with respect to ϵ_1 corresponds to the minimum error for a fixed number of correction terms. On the other hand, numerical convergence with tolerance ϵ defines our error control criterion, and determines if additional correction terms are needed to capture the true solution singularities.

In Stage-II we utilize the captured multi-singularities for full time-integration of the FDE in question, where a variety of the aforementioned numerical methods could be, in general, employed. At this stage, no knowledge of the solution data is required, but only the usual forcing terms. Our approach is explained as the following:

- To handle the numerical solution of the FDE using multiple correction terms, we develop an implicit finite-difference method, where we solve a linear system of size M for the first M time-steps. Therefore, we incorporate the captured singular behavior up to the desired precision ϵ , without the need of using a fine time-grid;
- We numerically demonstrate that the developed methodology is of order $\mathcal{O}(\Delta t^{3-\alpha})$;

We perform a set of numerical tests, where we demonstrate that the developed scheme is able to capture multiple singularities using a few number of time-steps, which can later be utilized for efficient time-integration of FDEs with relatively large time-step size Δt , being much more accurate than *ad-hoc* choices of singularities when the regularity of the solution is unknown. The successful capturing of singularities

motivates the development of kernel- and knowledge-based refinement of time-grids near $t = 0$ [46], as well as self-construction of basis function spaces using Müntz polynomials [13, 19] for spectral element methods for FDEs/FPDEs [23].

This paper is organized as follows: Section 2 introduces the definitions for fractional operators and the FDE in consideration. In section 3 we develop the two-stage framework for efficient time-integration of FDEs, where we start with the self-singularity-capturing scheme (Stage-I) in Section 3.1, followed by the finite-difference scheme for solution of FDEs (Stage-II) in Section 3.2 using the captured singularities. The numerical results with discussions for self-capturing up to three singularities are shown in Section 4, followed by the conclusions in Section 5.

2. Definitions

We start with some preliminary definitions of fractional calculus [38]. The left-sided Riemann-Liouville (RL) integral of order $\alpha \in \mathbb{R}$, with $0 < \alpha < 1$ and $t \in \mathbb{R}$, is defined as

$$({}^{RL}\mathcal{I}_a^\alpha v)(t) = \frac{1}{\Gamma(\alpha)} \int_a^t \frac{v(s)}{(t-s)^{1-\alpha}} ds, \quad t > a, \quad (1)$$

where Γ represents the Euler gamma function and a denotes the lower integration limit. The corresponding inverse operator, i.e., the left-sided RL fractional derivative of order α , is then defined based on (1) as

$$({}^{RL}\mathcal{D}_t^\alpha v)(t) = \frac{d}{dt}({}^{RL}\mathcal{I}_t^{1-\alpha} v)(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_a^t \frac{v(s)}{(t-s)^\alpha} ds, \quad t > a. \quad (2)$$

Furthermore, the corresponding left-sided Caputo derivative of order $\mu \in (0, 1)$ is obtained as

$$({}^C\mathcal{D}_t^\alpha v)(t) = ({}^{RL}\mathcal{I}_t^{1-\alpha} \frac{dv}{dx})(x) = \frac{1}{\Gamma(1-\alpha)} \int_a^t \frac{v'(s)}{(t-s)^\alpha} ds, \quad t > a. \quad (3)$$

The definitions of Riemann-Liouville and Caputo derivatives are linked by the following relationship, which can be derived by a direct calculation:

$$({}^{RL}\mathcal{D}_t^\alpha v)(t) = \frac{v(a)}{\Gamma(1-\alpha)(t-a)^\alpha} + ({}^C\mathcal{D}_t^\alpha v)(t), \quad (4)$$

which denotes that the definitions of the aforementioned derivatives coincide when dealing with homogeneous Dirichlet initial/boundary conditions.

We now introduce the Cauchy problem to be solved in this work and its corresponding well-posedness (see Theorem 3.25(i) [26] with $n = 1$). Let $C(\Omega)$ be the space of continuous functions $u(t)$ in $\Omega = [a, b]$ with the norm:

$$\|u\|_{C(\Omega)} = \max_{t \in \Omega} |u(t)| \quad (5)$$

Also, let $0 < \alpha < 1$ and $\gamma \in \mathbb{R}^+$, with $\gamma \leq \alpha$. We define $C_\gamma(\Omega)$ to be the following

weighted space of continuous functions:

$$C_\gamma(\Omega) = \{g(t) \mid (t-a)^\gamma g(t) \in C(\Omega), \|g\|_{C_\gamma} = \|(t-a)^\gamma g(t)\|_C\}, \quad (6)$$

Let G be an open set in \mathbb{R} , and let a function $f(t, u(t)) : (a, b] \times G \rightarrow \mathbb{R}$, such that $f(t, u(t)) \in C_\gamma(\Omega)$ and is Lipschitz continuous with respect to any $u(t) \in G$. Let the fractional Cauchy problem of interest given by:

$${}_a^C \mathcal{D}_t^\alpha u(t) = f(t, u(t)), \quad u(a) = u_0. \quad (7)$$

Given the aforementioned conditions, there exists a unique solution for (7) in the following space of functions:

$$\mathbf{C}_\gamma^\alpha(\Omega) = \{u(t) \in C(\Omega) \mid {}_a^C \mathcal{D}_t^\alpha u(t) \in C_\gamma(\Omega)\}. \quad (8)$$

The corresponding solution with respect to $u(t)$ is equivalent to the following Volterra integral equation of second kind: $u(t) = u_0 + \frac{1}{\Gamma(\alpha)} \int_a^t \frac{f(s, u(s))}{(t-s)^{1-\alpha}} ds$, with $a \leq t \leq b$. We remark that for other particular forms of (7), the corresponding Volterra equation of second kind contains Mittag-Leffler functions $E_\alpha(z)$, which are defined through infinite sums and thus are not computationally-friendly. Therefore, we choose to work with form (7), since the developed self-singularity-capturing framework can be extended for other FDEs in a straightforward way, without resorting to computationally expensive hyper-geometric functions. Furthermore, although the existence and uniqueness of Cauchy problems has been investigated [8, 10, 26, 44], yet, there is no comprehensive framework to understand the singular behavior of the solution given any general $f(t, u(t)) \in C_\gamma(\Omega)$. The developed formulation in this work is suitable for problems where a few initial discrete data points are known from $u(t)$ and $f(t, u(t))$.

Due to the employed finite-difference discretization over the RL derivative in this work (see Section 3.1.2), we choose to rewrite (7) using (4) to obtain:

$${}^{RL}_a \mathcal{D}_t^\alpha u(t) = f(t, u(t)) - \frac{u_0}{\Gamma(1-\alpha)(t-a)^\alpha}. \quad (9)$$

As will be shown in the next sections, the developed scheme is able to capture singularities with a minimal number of time-steps, *i.e.* with restricted data for a short period of time.

3. Two-Stage Time-Integration Framework

In this section, we develop the two-stage framework for efficient time-integration of FDEs. We start with Stage-I, where the self-singularity-capturing scheme is presented for S singularities, computed over a small number of initial data points denoted by \tilde{N} . Then, after the multi-singular solution behavior is learned, we present the developed solution for FDEs in Stage-II, where the subsequent full time-integration of the FDE is carried out over N time steps (see Figure 1).



Figure 1.: In Stage-I, the scheme captures the singularities for the short time-domain $\tilde{\Omega}$, given the short initial data. In Stage-II, the framework uses the singularities obtained in Stage-I for full time-integration of the FDE over Ω .

3.1. Stage-I: Self-Singularity-Capturing Stage

We develop the self-singularity-capturing framework, starting with M correction terms for the initial time-steps \tilde{N} , at which data is available. We then utilize the developed algorithm in a self-capturing approach, through a hierarchical and iterative fashion.

3.1.1. Minimization Method

Let $t \in \tilde{\Omega}$, with $\tilde{\Omega} = [t_0, t_{\tilde{N}}]$, where \tilde{N} denotes the number of initial short data points, such that $\tilde{N} \geq M$, and let $\boldsymbol{\sigma} \in \mathbb{R}^M$ be the following correction-power (singularity) vector:

$$\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \dots, \sigma_M]^T. \quad (10)$$

We define an error function $E : \mathbb{R}^M \rightarrow \mathbb{R}^+$, given by:

$$E(\boldsymbol{\sigma}) = \sum_{n=1}^{\tilde{N}} \left(u_n^{data} - u_n^N(\boldsymbol{\sigma}) \right)^2, \quad (11)$$

where $u_n^N(\boldsymbol{\sigma})$ denotes the $\boldsymbol{\sigma}$ -dependent numerical solution of $u(t)$ at $t = t_n$, and u_n^{data} represents the known initial short-time data. Figure 1 illustrates the integration domain $\tilde{\Omega}$ for Stage-I, where the error function (11) is evaluated for a short time. We apply an iterative gradient descent scheme in order to find $\boldsymbol{\sigma}^*$ that minimizes (11). Let $\boldsymbol{\sigma}^k$ be known at the k -th iteration. We compute the updated value $\boldsymbol{\sigma}^{k+1}$ for iteration $k+1$ in the following fashion [4]:

$$\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^k - \gamma^k \nabla E(\boldsymbol{\sigma}^k), \quad (12)$$

where γ^k denotes the following two-point step size for the k -th iteration, given by [3]:

$$\gamma^k = \frac{(\boldsymbol{\sigma}^k - \boldsymbol{\sigma}^{k-1})^T [\nabla E(\boldsymbol{\sigma}^k) - \nabla E(\boldsymbol{\sigma}^{k-1})]}{\|\nabla E(\boldsymbol{\sigma}^k) - \nabla E(\boldsymbol{\sigma}^{k-1})\|_{L^2(\mathbb{R}^M)}^2}. \quad (13)$$

The gradient $\nabla E : \mathbb{R}^+ \rightarrow \mathbb{R}^M$ of the error function is given by $\nabla E(\boldsymbol{\sigma}) = [\partial E / \partial \sigma_1, \partial E / \partial \sigma_2, \dots, \partial E / \partial \sigma_M]^T$, where, instead of obtaining a closed form for the derivatives (which would implicitly involve the differentiation of u_i^N), we utilize complex-step differentiation. Therefore, let $\mathbf{e}_j = [0, 0, \dots, 1, \dots, 0]^T$ be a vector in \mathbb{R}^M of zeros,

with a unit value in the j -th entry. Therefore, we have the following:

$$\nabla E(\boldsymbol{\sigma})_j \approx \frac{\text{Im}(E(\boldsymbol{\sigma} + i\Delta\sigma \mathbf{e}_j))}{\Delta\sigma}, \quad j = 1, 2, \dots, M, \quad i = \sqrt{-1}, \quad (14)$$

which only requires the additional evaluation of the error function perturbed by $\Delta\sigma$, which can be taken, *e.g.*, as 10^{-14} . Given two numerical tolerances ϵ and ϵ_1 , we iterate and find a new $\boldsymbol{\sigma}^{k+1}$ while both criteria $E(\boldsymbol{\sigma}^{k+1}) > \epsilon$ and $\|\nabla E(\boldsymbol{\sigma}^k)\| > \epsilon_1$ are true. The latter criterion is introduced to minimize our error function, while the former is used for error control in our self-capturing scheme. As will be shown in the next sections, the criterion $\|\nabla E(\boldsymbol{\sigma}^k)\| < \epsilon_1$ is satisfied for small ϵ_1 even when $M < S$, but the criterion $E(\boldsymbol{\sigma}^{k+1}) < \epsilon$ will only be satisfied when we use enough correction terms (see Section 3.1.3) to fully capture the S number of power-law singularities.

3.1.2. Numerical Scheme for Short-Time Integration

In the singularity-capturing scheme, given $\boldsymbol{\sigma}$, in order to compute the error in (11) and the gradient (due to the error perturbation) (14), we need to compute the numerical solution u_n^N for $n = 1, \dots, \tilde{N}$ initial data points over time-intervals Δt . For this purpose, we employ the finite-difference discretization with corrections presented in [32] for the fractional RL derivative. Including correction terms, the discretization of the left-sided RL fractional derivative of order α , evaluated at time $t = t_n$, with initial time $a = 0$, has the following form:

$${}^{RL}_0\mathcal{D}_t^\alpha u(t)|_{t=t_n} \approx {}^{RL}_0\mathcal{D}_t^{(\alpha, M)} u^N(t)|_{t=t_n}. \quad (15)$$

The above approximation is augmented by the *so-called* Lubich's correction terms, which is given by:

$${}^{RL}_0\mathcal{D}_t^{(\alpha, M)} u^N(t)|_{t=t_n} = {}^{RL}_0\mathcal{D}_t^\alpha u^N(t)|_{t=t_n} + \sum_{j=1}^M W_{j,n}(\boldsymbol{\sigma}) (u_j^N - u_0), \quad (16)$$

with $n = 1, \dots, \tilde{N}$. The term ${}^{RL}_0\mathcal{D}_t^\alpha u^N(t)|_{t=t_n}$ is usually denoted in the convolution quadrature form $\Delta t^{-\alpha} \sum_{j=0}^n w_{n-j} u_j^N$ with weights w_{n-j} . However we keep the former notation in this work to be consistent with the adopted discretization for fractional RL derivatives [59] presented later in this section. The term $W_{j,n}(\boldsymbol{\sigma})$ denotes the correction weights, which depend on the correction powers $\boldsymbol{\sigma} \in \mathbb{R}^M$. We assume a power-law solution singularity about $t = 0$, driven by the power-law kernel of the RL fractional derivative, *i.e.*, $u|_{\text{nearby}} \approx \sum_{k=1}^M c_k t^{\sigma_k}$. If $\boldsymbol{\sigma} = \{\sigma_k\}_{k=1}^M$ are known, the RL fractional derivative of each singular term, t^{σ_k} , can be obtained by:

$${}^{RL}_0\mathcal{D}_t^\alpha (t^{\sigma_k})|_{t=t_n} + \sum_{j=1}^M W_{j,n} t_j^{\sigma_k} = \frac{\Gamma(1 + \sigma_k)}{\Gamma(1 + \sigma_k - \alpha)} t_n^{\sigma_k - \alpha}, \quad k = 1, \dots, M, \quad (17)$$

where the first term on the left-hand side of (17) denotes the discretized RL fractional derivative of t^{σ_k} , while the right-hand side represents the exact fractional derivative

of t^{σ_k} , evaluated at $t = t_n$. Equation (17) can be written as:

$$\sum_{j=1}^M V_{k,j} W_{j,n} = \frac{\Gamma(1 + \sigma_k)}{\Gamma(1 + \sigma_k - \alpha) \Delta t^\alpha} n^{\sigma_k - \alpha} - \frac{1}{\Delta t^{\sigma_k}} {}^{RL}\mathcal{D}_t^\alpha(t^{\sigma_k})|_{t=t_n}, \quad (18)$$

with $k = 1, \dots, M$, where $V_{k,j} = j^{\sigma_k}$ denotes a Vandermonde matrix with size $M \times M$. Therefore, to obtain the starting weights $W_{j,n}$, the above linear system has to be solved for all $n = 1, \dots, \tilde{N}$.

Substituting (16) into (9), we obtain the following discrete form for the FDE:

$${}^{RL}\mathcal{D}_t^\alpha u^N(t)|_{t=t_n} + \sum_{j=1}^M W_{j,n}(\sigma) (u_j^N - u_0) + \frac{u_0}{\Gamma(1 - \alpha)t_n^\alpha} = f_n^{data}, \quad n = 1, \dots, \tilde{N}, \quad (19)$$

where f_n^{data} denotes the known short-time right-hand-side at time t_n associated with u^{data} , which can be obtained experimentally or sampled through a fine solution grid nearby $t = 0$.

In order to discretize ${}^{RL}\mathcal{D}_t^\alpha u^N(t)$, we follow the difference scheme developed in [59], which is based on a second-order interpolation of $u(t)$ for the fractional RL integral (1), similar to the L1-2 scheme for fractional Caputo derivatives developed in [15]. Therefore, when evaluating (1) at $t = t_n$, we split it into local \mathcal{L} and history \mathcal{H} parts as follows:

$${}^{RL}\mathcal{I}_t^\alpha u^N(t)|_{t=t_n} = \underbrace{\int_{t_{n-1}}^{t_n} k(t_n - s) u^N(s) ds}_{\mathcal{L}(u, t_n)} + \underbrace{\sum_{k=0}^{n-2} \int_{t_k}^{t_{k+1}} k(t_n - s) u^N(s) ds}_{\mathcal{H}(u, t_n)}, \quad (20)$$

with the kernel $k(t) = t^{\alpha-1}/\Gamma(\alpha)$. The function $u^N(t)$ is approximated in an implicit fashion using Lagrange interpolants $l_j^{(p)}$ of order p :

$$u^N(t_n) \approx I^{(p)} u_n^N = \sum_{j=0}^p l_j^{(p)} u_{n+j-p}^N. \quad (21)$$

Substituting (21) into (20), and evaluating the convolution integrals, we obtain the following approximations for the local and history parts, respectively, as

$$\mathcal{L}^{(n,\alpha)} u^N = \sum_{j=0}^p d_j^{(p)} u_{n+j-p}^N, \quad (22)$$

$$\mathcal{H}^{(n,\alpha)} u^N = \sum_{j=0}^{n-p} \left(b_{n-1-j}^{(1)} u_j^N + b_{n-1-j}^{(2)} u_{j+1}^N + b_{n-1-j}^{(3)} u_{j+2}^N \right), \quad (23)$$

where the corresponding α - and Δt -dependent coefficients $d_j^{(p)}$ and $b_j^{(1)}, b_j^{(2)}, b_j^{(3)}$ are presented in Appendix A. In our computations, we make use of $p = 1$ (piece-wise linear

approximation) for the local part $\mathcal{L}^{(1,\alpha)}u$, when $n = 1$ (first time-step), and $p = 2$ for $n > 1$ (subsequent time-steps). The fractional RL derivative can be obtained from the above discretization by setting $-1 < \alpha < 0$. Therefore:

$${}^{RL}_0\mathcal{D}^\alpha u^N(t)|_{t=t_n} \approx \mathcal{L}^{(n,-\alpha)}u^N + \mathcal{H}^{(n,-\alpha)}u^N, \quad (24)$$

which has accuracy of $\mathcal{O}(\Delta t^{3-\alpha})$ when $p = 2$. Finally, substituting (24) into (19), and recalling (22), we obtain the discretized form for our FDE:

$$\sum_{j=0}^p d_j^{(p)} u_{n+j-p}^N + \mathcal{H}^{(n,-\alpha)}u^N + \sum_{j=1}^M W_{j,n}(\sigma) (u_j^N - u_0) + \frac{u_0}{\Gamma(1-\alpha)t_n^\alpha} = f_n^{data}, \quad (25)$$

with $n = 1, \dots, \tilde{N}$.

Remark 1. Among a variety of available schemes, adopted the discretization of fractional operators introduced in [59] using Lubich's correction terms. However, in [59], the authors did not perform a fully implicit computation of the second term on the left-hand-side of (19) when $M > 1$. Instead, they obtained the solutions for $n = 1, \dots, M$ using a fine, uniform time-grid with time-step size $\tau = \Delta t^2$ and $M = 1$ correction term. We observe that using such a fine initial time-grid might be computationally expensive and ineffective when strong singularities are present (*e.g.* $u(t) = t^{\sigma^*}$, with $0 < \sigma^* < 1 - \alpha$). In our developed scheme, we treat such term in a fully implicit fashion, where we solve a small linear system with M unknowns to obtain u_n^N for $n = 1, \dots, M$. This ensures a proper inclusion of all M singularities in all time-steps near $t = 0$.

We present here the developed finite-difference scheme to solve the discretized FDE (25) for the particular case of $M = 3$. Here, we solve a small linear system for the fully-implicit computation at the first 3 time-steps, that is, $t = \{\Delta t, 2\Delta t, 3\Delta t\}$, which is obtained by expanding (25) for each of the time-steps, where we use $p = 1$ for the first one, with $\mathcal{H}^{(1,-\alpha)}u^N = 0$, and $p = 2$ for the subsequent ones. The expansions are presented as follows:

First time-step $t = \Delta t$:

$$\underbrace{(d_1^{(1)} + W_{1,1})}_{A_{11}} u_1^N + \underbrace{(W_{2,1})}_{A_{12}} u_2^N + \underbrace{(W_{3,1})}_{A_{13}} u_3^N = f_1^{data} - r_1 u_0, \quad (26)$$

with,

$$r_1 = d_0^{(1)} - \left(W_{1,1} + W_{2,1} + W_{3,1} + \frac{\Delta t^{-\alpha}}{\Gamma(1-\alpha)} \right). \quad (27)$$

Second time-step $t = 2\Delta t$:

$$\underbrace{(d_1^{(2)} + b_0^{(2)} + W_{1,2})}_{A_{21}} u_1^N + \underbrace{(d_2^{(2)} + b_0^{(3)} + W_{2,2})}_{A_{22}} u_2^N + \underbrace{(W_{3,2})}_{A_{23}} u_3^N = f_2^{data} - r_2 u_0, \quad (28)$$

with,

$$r_2 = d_0^{(2)} + b_0^{(1)} - \left(W_{1,2} + W_{2,2} + W_{3,2} + \frac{(2\Delta t)^{-\alpha}}{\Gamma(1-\alpha)} \right). \quad (29)$$

Third time-step $t = 3\Delta t$:

$$\begin{aligned} & \underbrace{\left(d_0^{(2)} + b_0^{(1)} + b_1^{(2)} + W_{1,3} \right)}_{A_{31}} u_1^N + \underbrace{\left(d_1^{(2)} + b_0^{(2)} + b_1^{(3)} + W_{2,3} \right)}_{A_{32}} u_2^N \\ & + \underbrace{\left(d_2^{(2)} + b_0^{(3)} + W_{3,3} \right)}_{A_{33}} u_3^N = f_3^{data} - r_3 u_0, \end{aligned} \quad (30)$$

with,

$$r_3 = b_1^{(1)} - \left(W_{1,3} + W_{2,3} + W_{3,3} + \frac{(3\Delta t)^{-\alpha}}{\Gamma(1-\alpha)} \right). \quad (31)$$

Therefore, from (26)-(31), we solve the following linear system for $n \leq 3$:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} u_1^N \\ u_2^N \\ u_3^N \end{bmatrix} = \begin{bmatrix} f_1^{data} \\ f_2^{data} \\ f_3^{data} \end{bmatrix} - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} u_0, \quad \text{if } n \leq 3. \quad (32)$$

Remark 2. To obtain the solutions for $M = 2$, we only need to remove the third row/column of the coefficient matrix \mathbf{A} , and set $W_{1,3} = W_{3,1} = W_{2,3} = W_{3,2} = 0$ in (26)-(31). Also, we observe that the correction terms lead to a full matrix of coefficients, which is reduced to a lower-triangular form for the uncorrected case.

For the remaining time-steps ($n > 3$), u_1^N, u_2^N, u_3^N are known and we solve for u_n^N in the usual (but still implicit) time-stepping fashion, as follows:

$$u_n^N = \left(f_n^{data} - \sum_{j=0}^1 d_j^{(2)} u_{n-2+j}^N - \mathcal{H}^{(n,-\alpha)} u^N - \sum_{j=1}^M W_{j,n} (u_j^N - u_0) - \frac{u_0 t_n^{-\alpha}}{\Gamma(1-\alpha)} \right) / d_2^{(2)}, \quad (33)$$

with if $3 < n \leq \tilde{N}$.

Remark 3. The current formulation can be extended for a larger number of correction terms; however, using $M > 9$ will incur in an ill-conditioned system for (17). This fact was first analyzed by Diethelm *et al.* [9] and later numerically shown by Zeng *et al.* [61], which would incur in significant errors when computing the weights $W_{n,j}$ and consequently propagate the errors to the operator discretization. In that sense, we choose $M = 3$ to capture the most critical singularities while keeping a small condition number and small residual for the linear system (17).

Algorithm I-1 Stage-I Self-Singularity-Capturing Scheme for FDEs.

```
1: Known data about  $u^{data}$ ,  $f^{data}$  for  $\tilde{N}$  time-steps of size  $\Delta t$ .
2: Set  $M = 1$ , initial guess  $\sigma^{(0)} = 0$  and numerical tolerances  $\epsilon$ ,  $\epsilon_1$ .
3: while  $M \leq 3$  do
4:   Estimate  $\sigma_i^{(k)}$ ,  $i = 1, \dots, M$  using Algorithm I-1 and obtain the error  $E(\sigma^{(k)})$ .
5:   if  $E(\sigma^{(k)}) < \epsilon$  then
6:     The singularities were captured within the specified tolerance  $\epsilon$ .
7:     break
8:   else
9:     Additional correction terms needed: Set  $M = M + 1$ .
10:    if  $M == 2$  then
11:      Set initial guess  $\sigma^{(0)} = [0, \sigma_1^{(k)}]^T$ .
12:    else if  $M == 3$  then
13:      Set initial guess  $\sigma^{(0)} = [\sigma_1^{(k)}, \sigma_2^{(k)}, (\sigma_1^{(k)} + \sigma_2^{(k)})/2]^T$ .
14:    end if
15:  end if
16: end while
17: return  $M, \sigma, E(\sigma)$ .
```

Algorithm I-2 Stage-I Singularity Capturing Algorithm (with M correction terms).

```
1: Initial guess  $\sigma^0$ ,  $\gamma^0$ , compute  $W_{n,j}$  using (17) and  $E(\sigma^0)$  using (11), (32), (33).
2: while  $E(\sigma^k) > \epsilon$  and  $\|\nabla E(\sigma^k)\| > \epsilon_1$  do
3:   Compute the perturbation  $\sigma^k + i\Delta\sigma e_j$  for  $j = 1, 2, \dots, M$ .
4:   Compute  $\nabla E(\sigma^k)$  using (14), (17), (11), (32), (33).
5:   Compute  $\gamma^k$  using (13) and update  $\sigma^{k+1} = \sigma^k - \gamma^k \nabla E(\sigma^k)$ .
6: end while
7: return  $\sigma, E(\sigma)$ .
```

3.1.3. Stage-I Algorithm

The Stage-I framework is described in Algorithm I-1, which learns about the singularities of the solution using small \tilde{N} . The M captured singularities σ are utilized to initialize an FDE solver in Stage-II (see Section 3.2).

Algorithm I-2 summarizes the main steps for the numerical scheme to capture the singularities of u^{data} , f^{data} for diminutive times at fixed M .

3.1.4. Computational Complexity of Stage-I

We recall that in the presented scheme, the error function (11) is evaluated N_{it} times, which is the number of iterations until convergence of the minimization scheme. The complexity of this error function is dominated by the time-integration of the numerical solution u^N over \tilde{N} time-steps. For the first M time-steps, we use (32) and solve the corresponding linear system, which costs $\mathcal{O}(M^3)$. For the remaining $\tilde{N} - M$ time-steps we use (33), where the dominant cost is due to the direct numerical evaluation of the fractional derivatives, that is, $\mathcal{O}((\tilde{N} - M)^2)$. Therefore, the computational complexity of the entire scheme is $\mathcal{O}((M^3 + (\tilde{N} - M)^2)N_{it})$. However, here we use $M \leq 3$, and therefore the asymptotic complexity becomes $\mathcal{O}(\tilde{N}^2 N_{it})$. Furthermore, \tilde{N} is assumed to be small due to the short-time data, and we show in Section 4 that we are able to

capture singularities with $\tilde{N} = \mathcal{O}(M)$, which makes the presented scheme numerically efficient, as long as a large number of iterations for convergence is not required.

Remark 4. We observe that since \tilde{N} is small, there is no need to use fast time-stepping schemes in Stage-I, since the break-even point between fast and direct schemes usually lies in a range of moderate number of time-steps (*e.g.* about $\mathcal{O}(10^4)$ for the fast scheme in [59]). Therefore, for small \tilde{N} , fast schemes would decrease the performance of Stage-I, and would only be beneficial for Stage-II.

3.2. Stage-II: Integration of FDEs with the Captured Singularities

Once the multi-singular behavior of the solution is captured through the framework presented in Section 3.1.3 and Algorithm I-1, a variety of numerical methods for FDEs can be implemented (*e.g.* other finite-difference and fast convolution schemes), incorporating the captured singularities. Here, to solve the FDE (9), we use the developed implicit finite-difference scheme presented in Section 3.1.2, and given by expressions (32) and (33) for $t \in \Omega = [t_0, t_N]$, using N time-steps of size Δt (not necessarily the same time-step size as Stage-I). The solution procedure is described in Algorithm II.

The computational complexity of Stage-II depends exclusively on the employed numerical discretization for the FDE. We utilize the same time-integration scheme as in Stage-I, for N time-steps. Therefore, recalling (32) and (33), we have a complexity of $\mathcal{O}(M^3 + (N - M)^2)$. However, since $M \ll N$, the asymptotic computational complexity of Stage-II becomes $\mathcal{O}(N^2)$.

4. Numerical Results

We start with Stage-I with a computational analysis of the correction weights (see Section 4.1), followed by the capturing scheme for the particular case of a single time-step and correction term (see Section 4.2). Then, we test the singularity-capturing Algorithm I-2 for $M = 1, 2$ and the self-capturing Algorithm I-1 for $M = 3$ (see Section 4.3). We also utilize the two-stage framework with random singularities and compare the accuracy of the entire time-integration framework with the captured singularities against *ad-hoc* choices through a comparison between the obtained error functions $E(\sigma)$ (see Section 4.4.1). In all aforementioned tests, if not stated otherwise, we use the method of fabricated solutions, that is, we take $u^{data} = u^{ext}$ and $f^{data} = f^{ext}$, where we assume the following solution:

$$u^{ext}(t) = \sum_{j=1}^S t^{\sigma_j^*}, \quad \sigma_j^* \in \mathbb{R}^+, \quad (34)$$

with homogeneous initial condition $u^{ext}(0) = 0$. The term S denotes the number singularities/terms in $u^{ext}(t)$, and σ_j^* represents prescribed singularities to be captured. From the defined analytical solution, we obtain the forcing term using a direct calculation, given by [38]:

$$f^{ext}(t) = \sum_{j=1}^S \frac{\Gamma(1 + \sigma_j^*)}{\Gamma(1 + \sigma_j^* - \alpha)} t^{\sigma_j^* - \alpha}. \quad (35)$$

Algorithm II Complete solution framework for FDEs.

- 1: **Stage-I: Self-Singularity-Capturing**
 - 2: Given data for u^{data} , f^{data} for \bar{N} time-steps, capture σ using Algorithm I-1.
 - 3: **Stage-II: Numerical solution of the FDE**
 - 4: Let N be the total number of time-steps. Compute $W_{n,j}$ using (18).
 - 5: Solve u_n^N for $n = 1, \dots, M$ using (32).
 - 6: **for** $n = 4$ **until** N **do**
 - 7: Solve for u_n^N using (33).
 - 8: **end for**
-

The numerical examples in this work are presented for verification of the developed framework, where u^{ext} and f^{ext} represent synthetic data for the Cauchy problem. In practice, only Stage-I of the framework utilizes experimental data for diminutive time to capture the multi-singularities, and Stage-II utilizes the captured singularities for the full time-integration of the FDE.

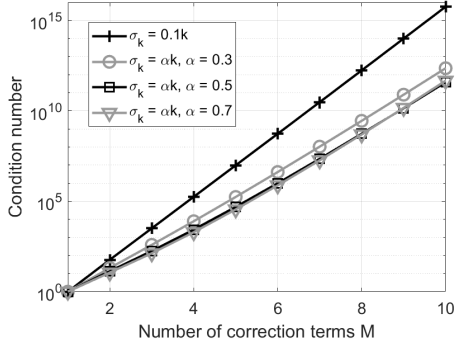
Furthermore, we test Stage-I of the framework for a multi-term FDE (see Section 4.4.2) and the two-stage framework for a one-term FDE with a *singular-oscillatory* solution (see Section 4.5). Finally, we test the developed framework against a nonlinear FDE and demonstrate its effectiveness when the solution data is sampled from a fine time-grid nearby the initial time (see Section 4.6). The developed algorithms were implemented in MATLAB R2018b, and were run in a laptop with Intel Core i7-8650U CPU with 1.90 GHz, 16 GB RAM and Windows 10 operating system.

4.1. Numerical Behavior of Correction Weights

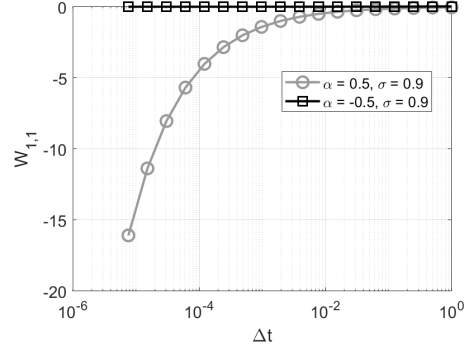
We investigate the behavior of the initial correction weights $W_{n,j}$ and the condition number for the Vandermonde matrix V presented in (18). Figure 2a illustrates the condition numbers for $V_{M \times M}$ using different choices for σ_k , namely $\sigma_k = \alpha k$ when the regularity of $u(t)$ is known, and $\sigma_k = 0.1k$ when it is unknown. We also illustrate the behavior of the correction weights using $M = 1$ with respect to the time-step size Δt and fractional order α (corresponding to fractional differentiation/integration) in Figures 2b and 3. We observe that for $\alpha = 0.5$ (fractional differentiation), $W_{1,1}$ increases in magnitude as Δt decreases. On the other hand, for $\alpha = 0.5$ (fractional integration), it starts with $W_{1,1} \approx 0.2$ for $\Delta t = 1$, and decreases with Δt . For fractional integration, only positive correction weights are observed, that decrease slower with respect to n , but with all values decreasing to zero as Δt decreases.

4.2. Single Time-Step and Correction Term

We present the numerical results for a particular case of Stage-I in Appendix B for $S = 1$ singularity and $M = 1$ correction term, with a closed-form for the correction weights (see (B6)). Let the time domain $\tilde{\Omega} = [0, \Delta t]$, with fractional-order and time-step size kept constant, respectively, at $\alpha = 0.5$ and $\Delta t = 0.01$ in this section. Figure 4 presents the convergence behavior for $\sigma_1^* = 0.5$, and Figure 5 illustrates the obtained results using $\sigma_1^* = 0.1$. We observe that in both cases, the scheme captures the power σ_1^* from the analytical solution, with a relatively small number of iterations. Furthermore, we observe an overshooting in the iterative procedure for initial guess $\sigma^{(0)} = 1.1$, but nevertheless, the scheme still converges within machine precision. Figures 6 and 7

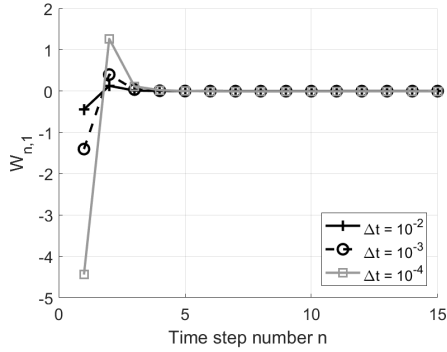


(a) Condition number of V vs M .

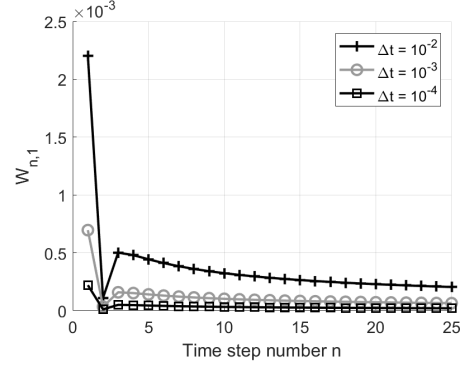


(b) First correction weight $W_{1,1}$ vs Δt .

Figure 2.: (a) Condition numbers for different choices of powers σ_k . We observe that the choice $\sigma_k = 0.1k$ for unknown regularity of $u(t)$ [59] leads to the highest condition number, where at most $M = 9$ correction terms can be used with double precision arithmetic. (b) First correction weight $W_{1,1}$ with respect to Δt .



(a) $\alpha = 0.5$, $\sigma_k = 0.9$.

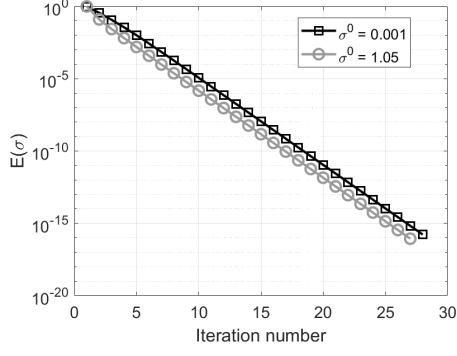


(b) $\alpha = -0.5$, $\sigma_k = 0.9$.

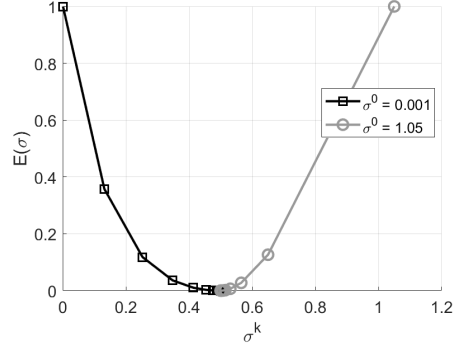
Figure 3.: Initial weights $W_{1,n}$ for different time-step sizes Δt , with corrections for fractional differentiation/integration (a) $\alpha = 0.5$ (fractional differentiation), where the initial weights have a larger magnitude that quickly decreases in the first time-steps, indicating that they are most relevant near $t = 0$. (b) $\alpha = -0.5$ (fractional integral).

show, respectively, the results for $S = 2$ with $\sigma_1^* = 0.1$, $\sigma_2^* = 0.2$ and $S = 3$ with $\sigma_1^* = 0.1$, $\sigma_2^* = 0.3$, $\sigma_3^* = 0.5$. We observe that the converged values for σ are closer to σ_1^* , which is the most critical singularity for the chosen $u^{ext}(t)$.

Since the converged values for σ captured intermediate values between the defined singularities $\sigma_1^*, \sigma_2^*, \sigma_3^*$ for the choice of $\Delta t = 0.01$, we analyze the convergence behavior of σ with respect to Δt . We present the obtained results in Figure 8 for two different sets of singular values, and we observe for the defined range of Δt , that σ lies between the singularities $\sigma_1^*, \sigma_2^*, \sigma_3^*$, and converge to the strongest singularity (in this case σ_1^*) as $\Delta t \rightarrow 0$.

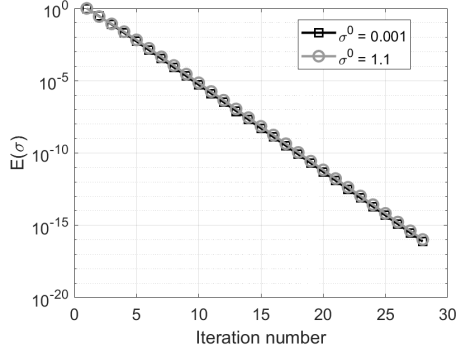


(a) Error *vs* iteration number.

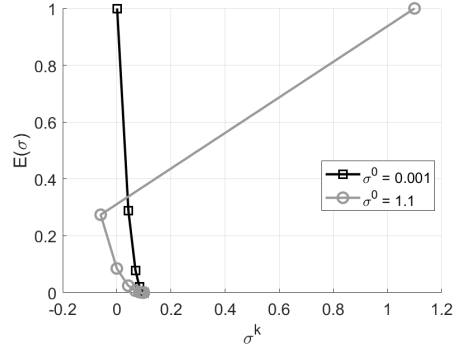


(b) Error *vs* iterated singularity

Figure 4.: Convergence behavior with $\sigma^* = 0.5$ and initial guesses $\sigma^{(0)} = \{0.0001, 1.05\}$ and time-step size $\Delta t = 0.01$.

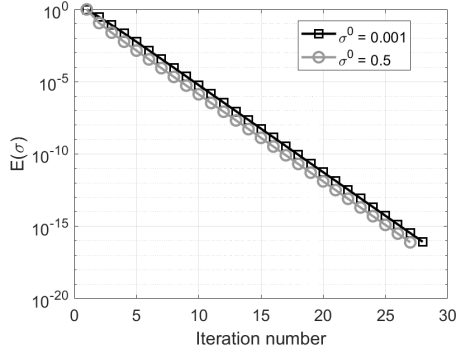


(a) Error *vs* iteration number

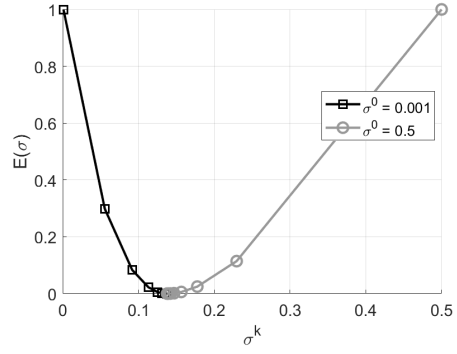


(b) Error *vs* iterated singularity

Figure 5.: Convergence behavior with $\sigma^* = 0.1$ and initial guesses $\sigma^0 = \{0.0001, 1.1\}$ and time-step size $\Delta t = 0.01$.



(a) Error *vs* iteration number



(b) Error *vs* iterated singularity

Figure 6.: Convergence behavior with $\sigma_1^* = 0.1$, $\sigma_2^* = 0.2$, initial guesses $\sigma^{(0)} = \{0.001, 0.5\}$ and time-step size $\Delta t = 0.01$. The converged value for the singularity $\sigma \approx 0.1377$ is between σ_1^*, σ_2^* .

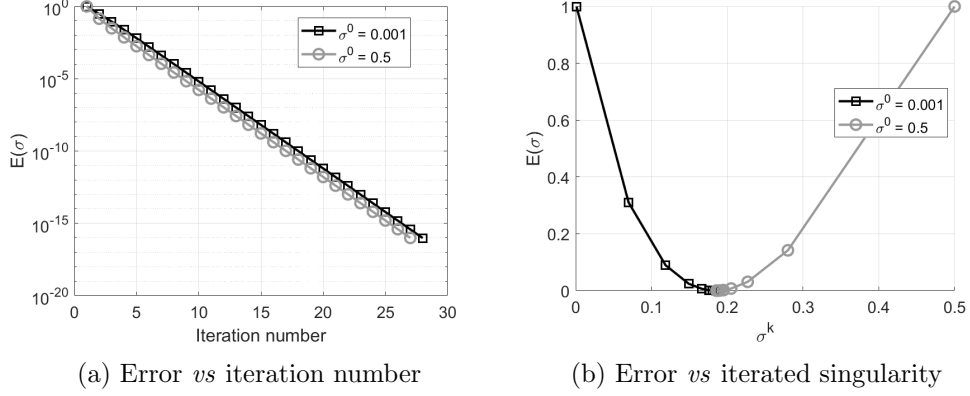


Figure 7.: Convergence behavior with $\sigma_1^* = 0.1$, $\sigma_2^* = 0.3$, $\sigma_3^* = 0.5$, initial guesses $\sigma^{(0)} = \{0.001, 0.5\}$ and time-step size $\Delta t = 0.01$. The converged value for the singularity is $\sigma \approx 0.1856$.

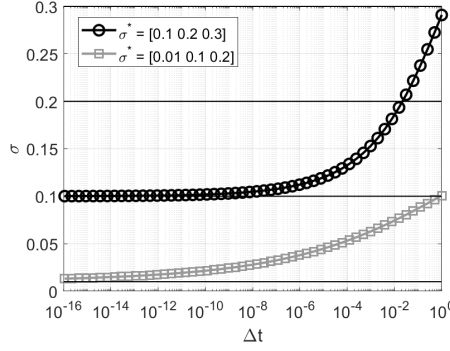


Figure 8.: Converged values for the singularity σ vs time-step size Δt with two choices of $\sigma^* = [\sigma_1^*, \sigma_2^*, \sigma_3^*]^T$.

4.3. Singularity Capturing for M Correction Terms

We systematically test the capturing Algorithm I-2 for multiple correction terms $M = 1, 2, 3$ and multiple defined singularities $S = 1, 2, 3$ in u^{ext} . The tests are presented in an incremental fashion for M , where we show the capabilities of Algorithm I-2 to capturing/approximating the singularities. We then demonstrate how the self-capturing framework defined in Algorithm I-1 successfully determines the singularities for $S = 3$ and $M = 3$, where we compare the obtained errors with *ad-hoc* choices for σ using random strong singularities. Throughout this section, we keep the fractional order fixed at $\alpha = 0.5$, as well as short time-domain $\tilde{\Omega} = [0, 1]$, and perturbation $\Delta\sigma = 10^{-14}$ for the complex step differentiation. For all cases, we define a tolerance $\epsilon = 10^{-15}$ for $E(\sigma)$ and unless stated otherwise, $\epsilon_1 = 10^{-14}$ for $\|\nabla E(\sigma)\|$. We choose a smaller tolerance for the norm of the error gradient (since the $E(\sigma)$ is defined with the norm $\|\cdot\|_2^2$) to make sure that $E(\sigma)$ is always minimized before we introduce additional correction terms in the self-capturing Stage-I. We also use $\gamma^0 = 10^{-3}$ for initialization in Algorithm I-2. For cases where $M = S$, we compute the component-wise relative

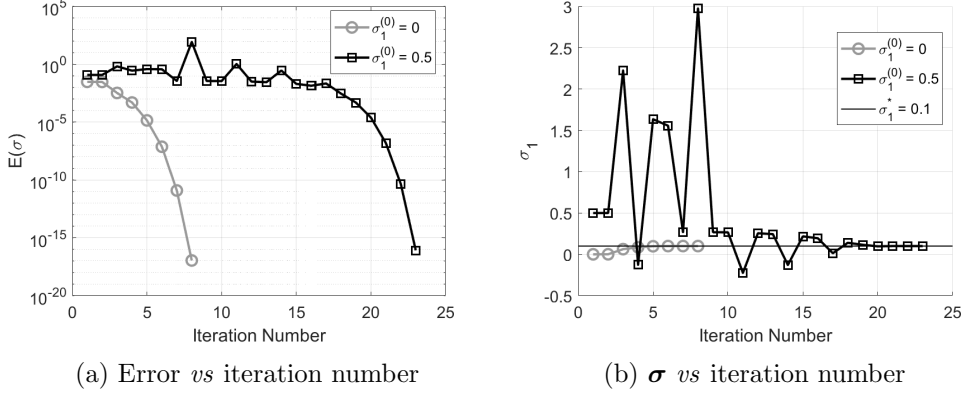


Figure 9.: Second-order asymptotic convergence behavior for $M = 1$ correction term, $S = 1$ singularity with value $\sigma_1^* = 0.1$. The obtained relative error for the converged σ_1 is $E_1^\sigma = 2.290 \times 10^{-8}$.

error of the converged σ^k , which we define as:

$$E_j^\sigma = \frac{|\sigma_j^* - \sigma_j^{(k)}|}{|\sigma_j^*|}, \quad j = 1, \dots, M, \quad (36)$$

where k denotes the iteration number when convergence is achieved, *i.e.*, $E(\sigma^{(k)}) < \epsilon$.

4.3.1. One Correction Term

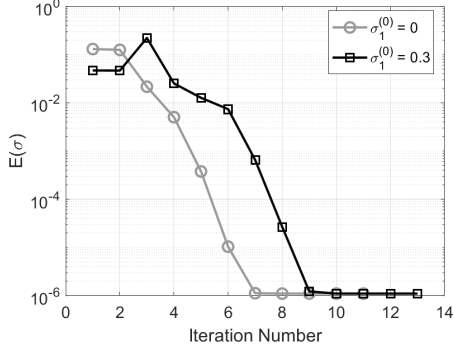
We consider $M = 1$ and $\tilde{N} = 100$ initial data-points. Figures 9, 10 and 11 present the obtained results, respectively, for $S = 1, 2, 3$ singularities, using Algorithm I-2. We observe that, when $S = M$, we can capture singularities within machine precision. When $M < S$, we are still able to find a minimum for $E(\sigma)$ with an intermediate value for σ_1 , similar to the results obtained for the simplified case in Section 4.2. Furthermore, when $S \geq 2$, or the initial guess $\sigma^{(0)}$ is far from the true singularity, the iterative procedure assumes the typical zig-zag behavior of the gradient descent scheme.

4.3.2. Two Correction Terms

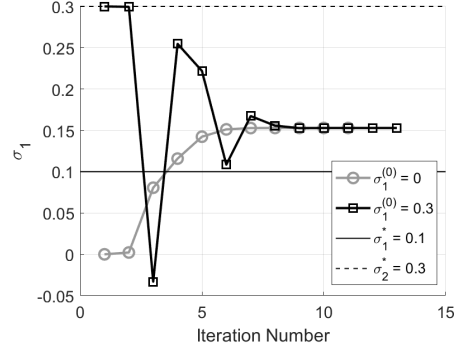
We consider $M = 2$, where we first analyze the behavior of the error function $E(\sigma)$ using $S = 2$ singularities (*see Figure 12*). Then, we use Algorithm I-2 to capture the singularities $\sigma^* = [0.1, 0.3]^T$ (*see Figure 13*), where we observe a more pronounced zig-zag behavior for convergence, compared to $M = 1$. We also obtain estimates of σ_1, σ_2 when dealing with $S = 3$ singularities, namely $\sigma^* = [0.1, 0.3, 0.5]^T$ (*see Figure 14*). Of particular interest, we observe that the obtained singularities lie in the range of the true singularities, which are useful observations to define initial guesses for cases with three correction terms and singularities.

4.3.3. Three Correction Terms

We test the case for $M = 3$, using $S = 1, 2, 3$. In this analysis, we use $\tilde{N} = 3$ initial data points. Figures 15 and 16 illustrate, respectively, the obtained results for $S = 1, 2$

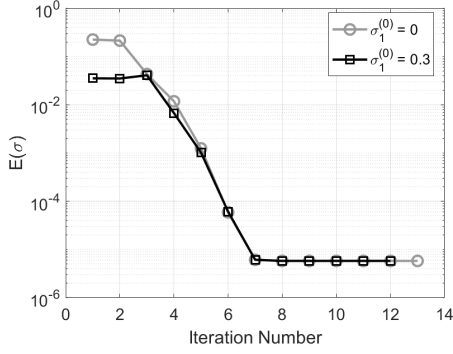


(a) Error *vs* iteration number.

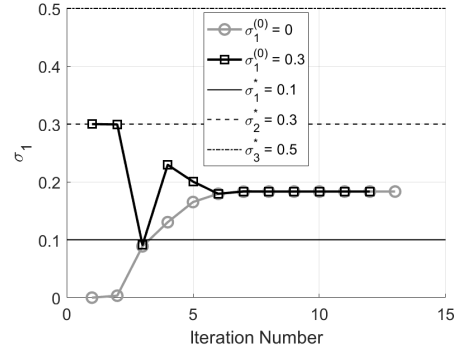


(b) σ *vs* iteration number.

Figure 10.: Convergence behavior for $M = 1$, $S = 2$ with $\sigma^* = [0.1, 0.3]^T$. With the plateau in error and σ due to gradient convergence within the tolerance ϵ_1 . The scheme converges to an “intermediate” value $\sigma \approx 0.153$, between the true singularities.



(a) σ *vs* iteration number.



(b) σ *vs* iteration number.

Figure 11.: Convergence behavior for $M = 1$, $S = 3$ with $\sigma^* = [0.1, 0.3, 0.5]^T$. In for both initial guesses, the scheme converges to an “intermediate” value $\sigma \approx 0.184$, which is closer to the strongest true singularity $\sigma_1^* = 0.1$.

using Algorithm I-2. We observe a quadratic convergence rate for $S = 1$, similar to Fig.9, but with a few more iterations. Furthermore, for $M = 2$, we observed a small change in the final value of σ_3 , but nevertheless, capturing the two singularities σ_1^* , σ_2^* is sufficient to minimize $E(\sigma)$ when $S = 2$.

The numerical convergence of Algorithm I-2 towards the correct singularities σ^* becomes much more difficult when using $S = M = 3$. Therefore, we make use of the self-capturing approach summarized in Algorithm I-1, which incrementally solves the minimization problem using $M = 1, 2, 3$ and use the respective output singularities as initial guesses for the subsequent number of correction terms $M = M + 1$. We present the obtained results in Fig.17. We observe that the scheme converges to the true singularities with errors for σ up to 0.024, but nevertheless, they yield an approximation error for u^N of 10^{-15} .

We also test the influence of the number of initial data points \tilde{N} in Stage-I over the number of iterations N_{it} and number of corrections terms M . For this purpose, we set $S = 3$ singularities with values $\sigma^* = [0.1, 0.3, 0.5]^T$ and we fix $\Delta t = 0.1$ [s], $\alpha = 0.5$,

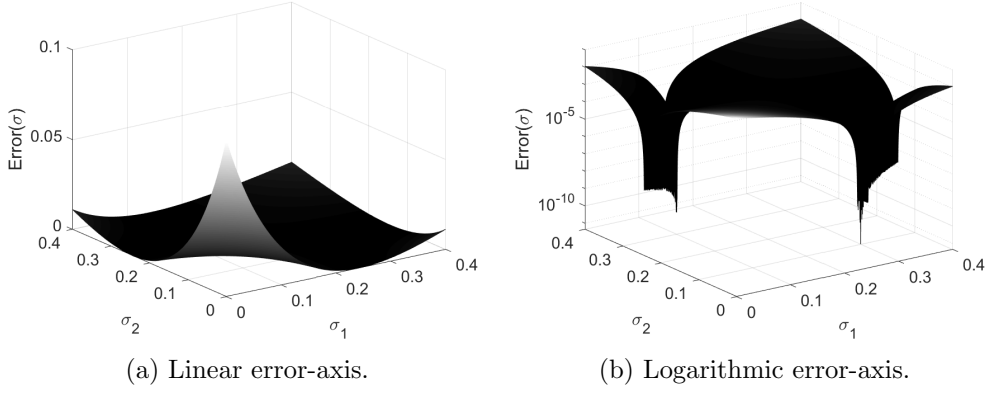


Figure 12.: Error function $E(\boldsymbol{\sigma})$ for $M = 2$, $S = 2$, with $\sigma_1^* = 0.1$, $\sigma_2^* = 0.3$. Two minima occur at $\boldsymbol{\sigma}^* = [0.1, 0.3]^T = [0.3, 0.1]^T$, corresponding to the true singularities. We set $\sigma_1 \neq \sigma_2$ to avoid a singular matrix in linear system (17).

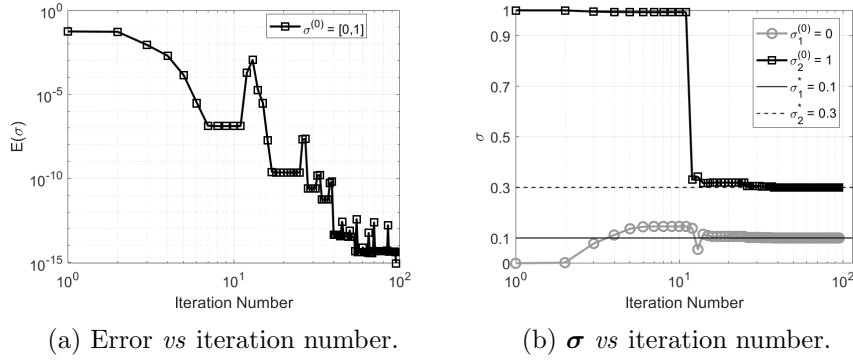
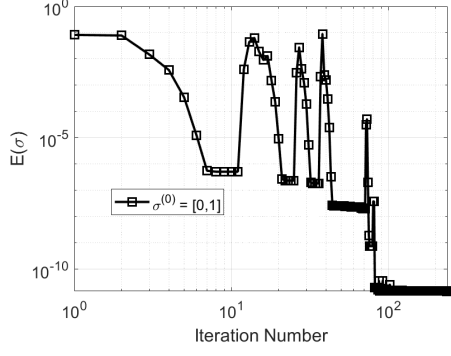
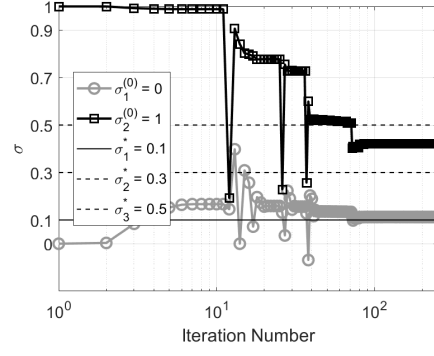


Figure 13.: Two correction terms $M = 2$ and two singularities $S = 2$, with $\boldsymbol{\sigma}^* = [0.1, 0.3]^T$. The obtained converged values are $\boldsymbol{\sigma} = [0.299974, 0.099990]^T$, which correspond to component-wise relative errors $E^\sigma = [8.61, 9.94]^T \times 10^{-5}$.

$\epsilon = 10^{-13}$, $\epsilon_1 = 10^{-10}$. We then vary \tilde{N} in Algorithm I-1 and analyze the convergence behavior and computational time, which are presented in Table 1. We observe that the variation of \tilde{N} did not affect the number of captured singularities M , but the choice $\tilde{N} = 3^7$ significantly increased the number of iterations. We also observe that despite the similar number of iterations N_{it} obtained for $3^1 \leq \tilde{N} \leq 3^6$, the computational time increases with \tilde{N} due to the computational complexity of Stage-I. Therefore, we consider $\tilde{N} = \mathcal{O}(M)$ an appropriate choice, as the singularities $\boldsymbol{\sigma}^*$ are captured with less computational cost, which is suitable when diminutive initial data is available.

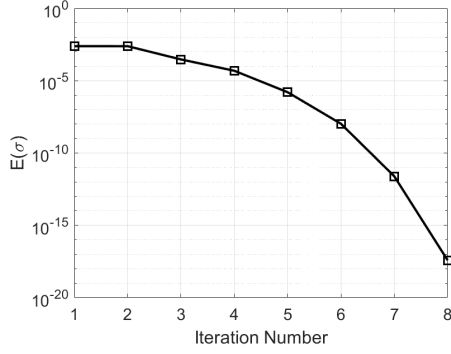


(a) Error *vs* iteration number.

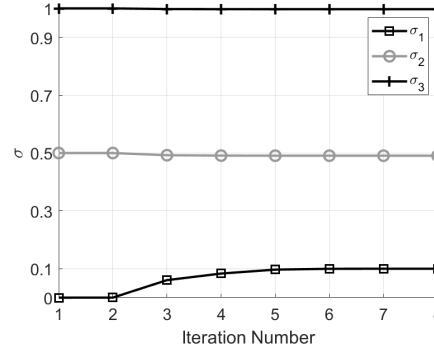


(b) σ *vs* iteration number.

Figure 14.: Two correction terms $M = 2$ and three singularities $S = 3$, with $\sigma^* = [0.1, 0.3]^T$ and $\epsilon_1 = 10^{-11}$. The converged values are $\sigma = [0.112635, 0.420495]^T$, which are closer to the stronger singularities $\sigma_1 = 0.1$ and $\sigma_3 = 0.5$. The approximate singularities still lead to an approximation error of 1.47×10^{-11} .



(a) Error *vs* iteration number.



(b) σ *vs* iteration number

Figure 15.: Results for $M = 3$, $S = 1$. The obtained relative error for σ_1 is $E_1^\sigma = 4.869 \times 10^{-8}$. The components σ_1, σ_2 remained practically constant.

Table 1.: Influence of \tilde{N} in Stage-I of the developed framework.

\tilde{N}	M	N_{it}	Computation Time [s]	$E(\sigma)$
3^1	3	74	0.217	9.863015×10^{-14}
3^2	3	44	0.293	2.101287×10^{-14}
3^3	3	51	0.617	9.433562×10^{-14}
3^4	3	62	2.251	9.037342×10^{-14}
3^5	3	98	11.346	9.101754×10^{-14}
3^6	3	52	180.719	1.550662×10^{-14}
3^7	3	445	1275.168	8.823108×10^{-14}

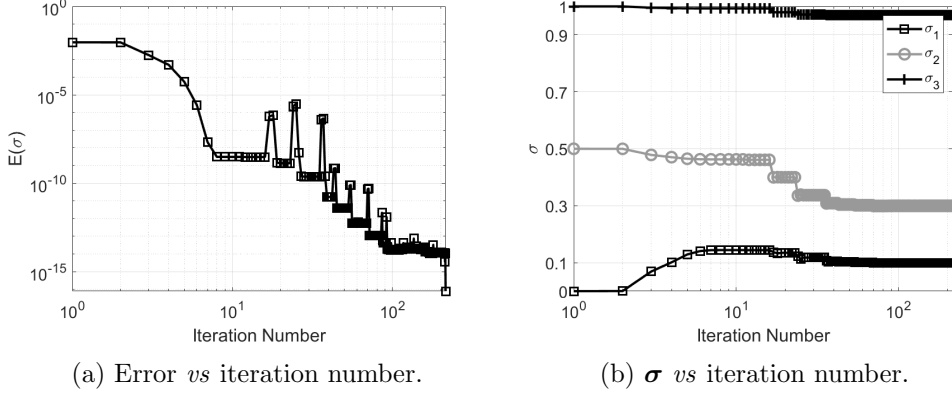


Figure 16.: Results for $M = 3$, $S = 2$. The obtained converged values are $\sigma = [0.099997, 0.299995, 0.939015]^T$, where σ_1, σ_2 captured the singularities, respectively with errors $E_1^\sigma = 3.07 \times 10^{-5}$ and $E_2^\sigma = 1.65 \times 10^{-5}$.

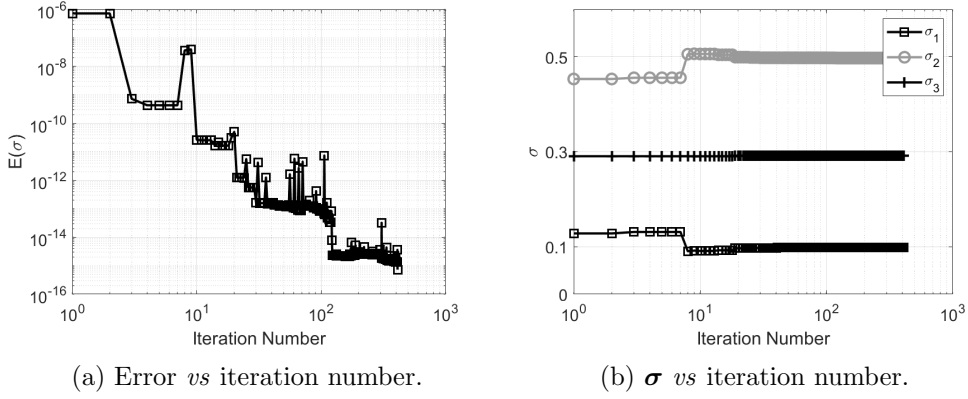


Figure 17.: Results for $M = 3$, $S = 3$ using $\epsilon = 10^{-15}$. The converged values are $\sigma = [0.099, 0.498, 0.293]^T$, which correspond to component-wise relative errors $E^\sigma = [0.014, 0.005, 0.024]^T$.

4.4. Random Singularities

We perform two numerical tests involving random power-law singularities, where we define three strong singularities $\sigma_1^*, \sigma_2^*, \sigma_3^*$ randomly sampled from a uniform distribution $\mathcal{U}(0, 1/2)$. We employ the self-singularity-capturing procedure in the context of single- and multi- term FDEs.

4.4.1. FDEs with Random Singularities

We test the two-stage framework for efficient time-integration of (9). We then compute $u^{ext}(t)$ and $f^{ext}(t)$, respectively using (34) and (35), and use the self-capturing framework presented in Algorithm I-1. We then compare the approximation error $E(\sigma)$ with the choice $\sigma_k = 0.1k$ defined in [59] when the singularities are unknown. Although our framework has no explicit information about the generated random singularities, we

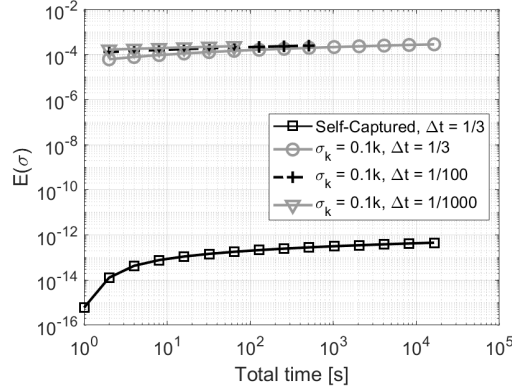


Figure 18.: Comparison of error *vs* time for long time-integration between the captured singularities and $\sigma_k = 0.1k$ for different values of time-step size Δt .

present them for verification purposes, which are given by:

$$\sigma^* = [0.0172230402514543, 0.219372179828199, 0.190779228546504]^T. \quad (37)$$

We set $\tilde{N} = 3$ with $\Delta t = 1/3$, and therefore, $\tilde{\Omega} = [0, 1]$, $\epsilon = 10^{-15}$, and $\epsilon_1 = 10^{-13}$ to Algorithm II. Stage-I of the framework (Algorithm I-1) converges with $E(\sigma^{(k)}) = 2.96 \times 10^{-16}$ with $M = 2$ correction terms, with obtained values:

$$\sigma = [0.0187990387914248, 0.206944449676742]^T, \quad (38)$$

with elapsed computational time of 0.169 [s]. After capturing the singularities within the desired precision ϵ , we enter Stage-II and compute the numerical solution $u^N(t)$ using the captured σ for multiple, longer time-integration domains Ω . We remark that when using an *ad-hoc* choice of fixed $\sigma_k = 0.1k$ with 4 correction terms, that is,

$$\sigma = [0.1, 0.2, 0.3, 0.4]^T, \quad (39)$$

we obtain an approximation error $E(\sigma) = 5.25 \times 10^{-5}$ over $N = 4$ time-steps. We use the captured powers (38) and the predefined ones (39) for longer time-integration. The obtained results are presented in Figure 18, where we observe how using 3 time-steps to capture the singularities leads to precise long time-integration with a large time-step size $\Delta t = 1/3$ (solid black curve). On the other hand, the errors for $\sigma_k = 0.1k$ are much larger, and also do not improve with smaller Δt due to the presence of very strong singularities.

4.4.2. Multi-Term FDEs with Random Singularities

In this section we show how the scheme in Stage-I can be applied for multi-term fractional differential equations. Let the following multi-term FDE:

$$\sum_{l=1}^{N_\alpha} {}^{RL}_0 \mathcal{D}_t^{\alpha_l} u(t) = f(t, u(t)), \quad u(0) = u_0, \quad 0 < \alpha_l < 1. \quad (40)$$

where α_l denotes the N_α multi-fractional orders with $l = 1, \dots, N_\alpha$. Using the approximation (16), and assuming the same powers σ for the correction weights for all fractional derivatives, we obtain:

$$\sum_{l=1}^{N_\alpha} {}^{RL}\mathcal{D}_t^{\alpha_l} u^N(t)|_{t=t_n} + \sum_{j=1}^M \left(\sum_{l=1}^{N_\alpha} W_{j,n}^{(\alpha_l)}(\sigma) \right) (u_j^N - u_0) = f_n^\delta, \quad (41)$$

where we use the superscript (α_l) in $W_{j,n}^{(\alpha_l)}$ to denote the distinct sets of correction weights due to the fractional order α_l through (18). The entire minimization scheme is identical, since it depends on the solution data for $u(t)$. In order to solve (41) with the developed scheme in Stage-I, due to the linearity of (40), we only need to replace the initial correction weights $W_{n,j}$, and α -dependent coefficients d_j , b_j in (32) and (33) with the following summations:

$$\tilde{W}_{n,j} = \sum_{l=1}^{N_\alpha} W_{n,j}^{(\alpha_l)}, \quad \tilde{d}_j^{(p)} = \sum_{l=1}^{N_\alpha} d_j^{(p,\alpha_l)}, \quad \tilde{b}_j^{(k)} = \sum_{l=1}^{N_\alpha} b_j^{(k,\alpha_l)}, \quad k = 1, 2, 3. \quad (42)$$

We utilize the fabricated solution (34) in (40), which yields the following force term:

$$f^{ext}(t) = \sum_{\alpha_l} \sum_{j=1}^S \frac{\Gamma(1 + \sigma_j^*)}{\Gamma(1 + \sigma_j^* - \alpha_l)} t^{\sigma_j^* - \alpha_l}. \quad (43)$$

We set $N_\alpha = 3$ fractional orders $\{\alpha_1, \alpha_2, \alpha_3\} = \{0.3, 0.5, 0.7\}$, and identically to Section 4.4.1, we sample the following random singularities from $\mathcal{U}(0, 1/2)$:

$$\sigma^* = [0.13924910943352420, 0.2734407596024919, 0.4787534177171488]^T. \quad (44)$$

Also, we set $\tilde{N} = 3$ with $\Delta t = 1/3$ and $\tilde{\Omega} = [0, 1]$, $\epsilon = 5 \times 10^{-15}$, and $\epsilon_1 = 10^{-14}$, with $\gamma^0 = 10^{-3}$ to Algorithm I-1, where the scheme converges with the following approximate singularities:

$$\sigma = [0.1469249923105880, 0.4869203803691072, 0.3066386453671829]^T. \quad (45)$$

Figure 19 presents the obtained results for each M in the self-capturing procedure, where we observe that the scheme does not fully capture the true singularities, but provides sufficiently good approximations to obtain errors as low as $E(\sigma) = 2.31 \times 10^{-15}$. The elapsed computational time is 10.65 [s], due to the larger number of iterations N_{it} required for convergence of Stage-I.

4.5. Singular-Oscillatory Solutions

We also test the two-stage framework sampling data points from a fabricated solution for (9), defined as the following multiplicative coupling between a power-law and oscillatory parts:

$$u^{ext}(t) = t^{\sigma^*} \cos(\omega t), \quad (46)$$

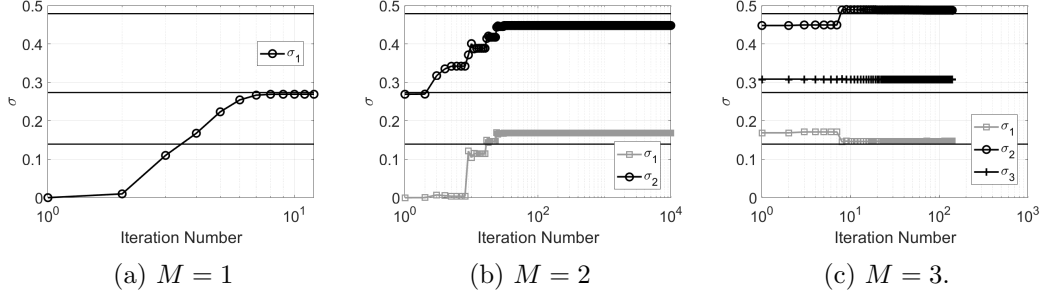


Figure 19.: Singularities σ vs iteration number for a multi-term FDE with $N_\alpha = 3$ in Stage-I for each M in Algorithm I-1. (a) $M = 1$, with error $E(\sigma) = 1.10 \times 10^{-5}$, (b) $M = 2$, with error $E(\sigma) = 3.83 \times 10^{-13}$, and (c) $M = 3$ with convergence at $E(\sigma) = 2.31 \times 10^{-15}$. The black horizontal lines correspond to the true random singularities in the solution.

with homogeneous initial condition $u^{ext}(0) = 0$. The corresponding right-hand-side is analytically obtained as:

$$f^{ext}(t) = c_1 t^{\sigma^* - \alpha} \left[{}_p\tilde{F}_q(a^{(1)}; b^{(1)}; -\omega^2 t^2/4) + c_3 \omega^2 t^2 {}_p\tilde{F}_q(a^{(2)}; b^{(2)}; -\omega^2 t^2/4) \right], \quad (47)$$

with $c_1 = -2^{\alpha - \sigma^* - 4} \pi \Gamma(1 + \sigma^*)$, $c_2 = 8(\alpha - \sigma^* - 1)$ and $c_3 = (1 + \sigma^*)(2 + \sigma^*)$. Also, ${}_p\tilde{F}_q$ represents the regularized, generalized hypergeometric function given by [53]:

$${}_p\tilde{F}_q(a_1, \dots, a_p; b_1, \dots, b_q; z) = \frac{{}_pF_q(a_1, \dots, a_p; b_1, \dots, b_q; z)}{\Gamma(b_1) \dots \Gamma(b_q)}, \quad (48)$$

where, for the particular case (47), we have:

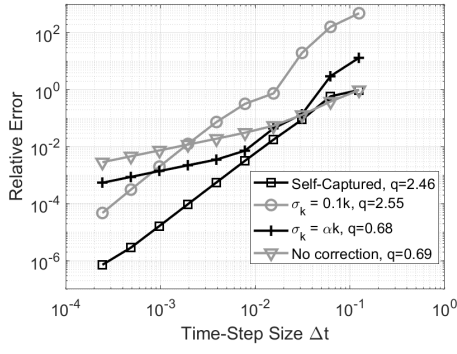
$$\begin{cases} a^{(1)} = \{(1 + \sigma^*)/2, (2 + \sigma^*)/2\}, & b^{(1)} = \{1/2, (2 - \alpha + \sigma^*)/2, (3 - \alpha + \sigma^*)/2\}, \\ a^{(2)} = \{(3 + \sigma^*)/2, (4 + \sigma^*)/2\}, & b^{(2)} = \{3/2, (4 - \alpha + \sigma^*)/2, (3 - \alpha + \sigma^*)/2\}. \end{cases}$$

We fix the fractional order $\alpha = 0.5$, frequency $\omega = 10\pi$ and randomly sample σ^* from a uniform distribution $\mathcal{U}(0, 1/3)$ and obtain a value $\sigma^* = 0.2426481954401539$. For Stage-I, we consider $\tilde{N} = 3$ data points with $\tilde{\Omega} = [0, 0.01]$, with $\Delta t = 3/100$ [s]. Furthermore, we set $\epsilon = 10^{-11}$, $\epsilon_1 = 10^{-14}$ and $\gamma^0 = 10^{-3}$. Using Algorithm I-1 with a slight initial guess modification for $\sigma^{(0)}$ when $M = 2$ in line 11 to $\sigma^{(0)} = [1, \sigma_1^{(k)}]^T$ (since the developed framework is mainly oriented to strong singularities), we capture the following two singularities:

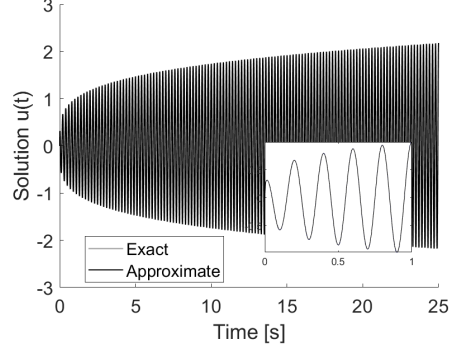
$$\sigma = [0.2427452349772425, 2.220682758797950]^T,$$

where the first converged singularity is an approximation to the randomly sampled value for σ^* . To understand the second captured singularity σ_2 , let the series expansion of (46) around ωt :

$$u^{ext}(t) = t^{\sigma^*} \cos(\omega t) = t^{\sigma^*} \left(1 - \frac{\omega^2}{2} t^2 + \mathcal{O}(t^4) \right) = t^{\sigma^*} - \frac{\omega^2}{2} t^{2+\sigma^*} + \mathcal{O}(t^{4+\sigma^*}). \quad (49)$$



(a) Relative error *vs* time-step size.



(b) Solutions for $u(t)$ *vs* time.

Figure 20.: (a) Convergence analysis comparing the accuracy of the captured singularities in Stage-I with *ad-hoc* choices for time-integration with $T = 1$ [s] in Stage-II. The term q denotes the convergence rate. The self-captured case uses $M = 2$ correction terms, while for $\sigma_k = 0.1k$ and $\sigma_k = \alpha k$ we use $M = 4$. (b) Exact and approximate solutions for $u(t)$ using the captured singularities for longer time-integration over $T = 25$ [s].

Therefore, the captured σ_2 is an approximation of the second term $2 + \sigma^*$ of the above series expansion. Although capturing the third singularity $4 + \sigma^*$ is possible, we would need to set higher values for the initial guesses in the self-capturing algorithm, and finding the corresponding minimum becomes more difficult, since such higher-order, weaker singularities have less influence in the accuracy of the solution.

After capturing the singularities in Stage-I, with elapsed computational time of 0.149 [s], we apply them in Stage-II for a convergence test with $T = 1$ [s], with $\Omega = [0, T]$, and compare the approximate solution $u^N(t)$ with the exact one (49) through the relative error defined as

$$E_{L^2} = \frac{\|\mathbf{u}^N - \mathbf{u}^{ext}\|_{L^2(\Omega)}}{\|\mathbf{u}^{ext}\|_{L^2(\Omega)}}. \quad (50)$$

between the two-stage framework other *ad-hoc* choices for σ , and also without corrections (see Figure 20a). We observe that capturing $M = 2$ singularities leads to the theoretical accuracy $\mathcal{O}(\Delta t^{3-\alpha})$ of the employed discretization, and also to errors about 2 orders of magnitude lower than the choice $\sigma_k = 0.1k$ with $M = 4$ for the singularities in the observe range for Δt . We also compare the exact and approximate solutions over $T = 25$ [s] with $N = 4096$ time-steps of size $\Delta t \approx 6.10 \times 10^{-3}$, and observe that we fully capture the singular and oscillatory behaviors of the solution, with elapsed computational time of 0.426 [s].

4.6. Nonlinear FDE

Finally, we apply the developed framework for a nonlinear right-hand-side of the FDE (9), now given by:

$${}_0^C \mathcal{D}_t^\alpha u(t) = C_1 u(t)^2 \exp(-C_2 t) + 1, \quad u(0) = 0, \quad (51)$$

where we set $C_1 = 1$ and $C_2 = 5$. Differently from the previous examples, where the diminutive data points were sampled from a known solution $u^{ext}(t)$, here, we sample $u^{data}(t)$ from a benchmark solution obtained through a fine-grid nearby $t = 0$. The right-hand-side is evaluated explicitly. Our procedure follows:

- **I)** Set the short time-domain $\tilde{\Omega} = [0, 0.01]$ and numerically obtain the approximate solution of (51) using $\Delta t = 10^{-5}$ [s].
- **II)** Sample $\tilde{N} = 100$ benchmark solution data points and utilize the singularity capturing algorithm (Stage-I) with $\Delta t = 10^{-4}$ and $\epsilon_1 = 10^{-14}$ over $\tilde{\Omega}$ to obtain σ . The value of ϵ is chosen according to the employed fractional order α .
- **III)** We utilize the captured singularities and the RHS of (51) evaluated explicitly for full time-integration (Stage-II) over $\Omega = [0, 1]$, using $\Delta t = 10^{-4}$ [s], and compare the results with the benchmark solution integrated over Ω .

We perform two tests: In the first, we set $\alpha = 0.2$, $\epsilon = 7.5 \times 10^{-5}$ in Stage-I, and capture $\sigma = 7.236455491751606 \times 10^{-2}$. In the second one, we set $\alpha = 0.25$, $\epsilon = 1.1 \times 10^{-5}$ and capture $\sigma = 7.602862102264105 \times 10^{-2}$. Figure 21 illustrates the obtained results for full time-integration. We observe that the self-captured solution matches the benchmark case for the entire time-integration domain, including nearby $t = 0$, demonstrating that the developed scheme is effective for nonlinear FDEs with unknown solutions. The obtained computational times for the self-capturing framework are: 0.07 [s] for fine grid solution and sampling, 0.15 [s] for Stage-I and 0.86 [s] for Stage-II, yielding a total computational time of 1.08 [s], compared to 48.58 [s] for the benchmark solution. Therefore, the fine grid sampling at diminutive time had minimal influence on the computational cost.

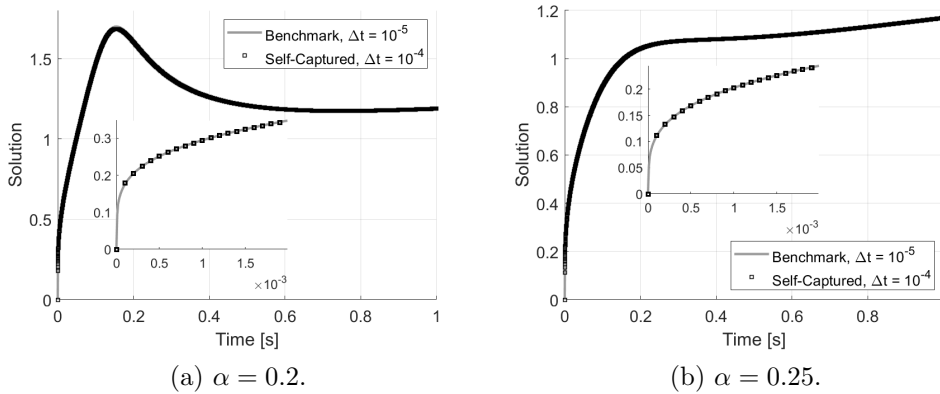


Figure 21.: Self-captured and benchmark solutions *versus* time for full time-integration of a nonlinear FDE.

5. Conclusions

We developed a two-stage framework for accurate and efficient solution of FDEs. It consisted of: **I)** A self-singularity-capturing scheme, where the inputs were limited data for diminutive time, and the output was the captured power-law singular values of the solution. The limited data can be either obtained experimentally or sampled from a fine time-grid nearby $t = 0$. We developed an implicit finite-difference algorithm for

the FDE solution and employed a gradient optimization scheme. **II)** The captured singularities were then inputs of an implicit finite-difference scheme for FDEs with accuracy $\mathcal{O}(\Delta t^{3-\alpha})$. Based on our numerical tests, we observed that:

- The scheme was able to fully capture S singularities using $M \geq S$ correction terms. When $S > M$, the scheme still obtained approximations of the M singularities, which was useful for error control.
- When $M = 1$, the most critical singularity was estimated by the scheme, regardless of S .
- The developed scheme was used in a self-singularity-capturing framework, which determined the singularity values up to the desired tolerance ϵ .
- Once the singularities are captured in the first stage, we can successfully employ any known numerical scheme for more accurate and efficient solution of the corresponding FDE, when compared to the *ad-hoc* singularity choices done in [59].
- The scheme was able to capture two singularities from a solution using a strong power-law singularity coupled with an oscillatory smooth part, which allowed us to obtain the theoretical accuracy $\mathcal{O}(\Delta t^{3-\alpha})$ of the employed numerical discretization for the fractional derivative. Furthermore, by solving a nonlinear FDE, we demonstrate that the developed framework does not require the knowledge of any analytical solution.

The main advantage of the developed scheme over other works on correction methods [32, 59, 61] is the self-capturing of strong multi-singularities σ of the solution data in Stage-I instead of *ad-hoc* choices. This led to smaller solution errors while achieving the theoretical convergence rate of the employed numerical integration schemes in Stage-II, with algorithmically minimal correction terms M , and consequently requiring less computational cost.

The total computational complexity of the two-stage framework is $\mathcal{O}(\tilde{N}^2 N_{it} + N^2)$, since the number of correction terms M is small. Among a variety of numerical schemes, Stage-II of the framework could incorporate fast convolution frameworks [33, 59] or a divide-and-conquer scheme [14]. This would dramatically increase the efficiency for long time-fractional integration of anomalous materials under complex loading conditions [2, 50–52] and also further improve the numerical accuracy for nonlinear problems using fractional multi-step/IMEX schemes [58, 63]. The developed scheme can also be used to capture the power-law attenuations and creep-ringing [20] in highly-oscillatory vibration of anomalous systems. Also, discovering the multi-singular behavior of the solution can give insight on knowledge- and kernel-based grid refinement near $t = 0$ [46]. Furthermore, the presented scheme can leverage the development of self-constructing function spaces using Müntz polynomials [13, 19], increasing the efficiency of spectral element methods [23] applied to anomalous transport.

Acknowledgement(s)

The authors would like to thank Dr. Ehsan Kharazmi and Dr. Yongtao Zhou for productive discussions along the development of this work.

Funding

This work was supported by the ARO Young Investigator Program Award (W911NF-19-1-0444), the MURI/ARO (W911NF-15-1-0562), and partially by the National Science Foundation Award (DMS-1923201) and AFOSR Young Investigator Program Award (FA9550-17-1-0150).

References

- [1] Baffet, D. and Hesthaven, J. (2017). A kernel compression scheme for fractional differential equations. *SIAM J. Numer. Anal.*, 55(2):496–520.
- [2] Barros de Moraes, E., Zayernouri, M., and Meerschaert, M. M. (2020). An integrated sensitivity-uncertainty quantification framework for stochastic phase-field modeling of material damage. *Int. J. Numer. Methods Eng. (accepted)*.
- [3] Barzilai, J. and Borwein, J. (1988). Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148.
- [4] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.
- [5] Breiten, T., Simoncini, V., and Stoll, M. (1983). Low-rank solvers for fractional differential equations. *Electron. Trans. Numer. Anal.*, 45:107–132.
- [6] Chen, W. and Gu, Y. (2012). An improved formulation of singular boundary method. *Adv. Appl. Math. Mech.*, 4:543–558.
- [7] Craiem, D., Rojo, F., Atienza, J., Armentano, R., and Guinea, G. (2008). Fractional-order viscoelasticity applied to describe uniaxial stress relaxation of human arteries. *Phys. Med. Biol.*, 53(17):4543.
- [8] Delbosco, D. and Rodino, L. (1996). Existence and uniqueness for a nonlinear fractional differential equation. *J. Math. Anal. Appl.*, 204(2):609–625.
- [9] Diethelm, K., Ford, J., Ford, N., and Weilbeer, M. (2006). Pitfalls in fast numerical solvers for fractional differential equations. *J. Comput. Appl. Math.*, 186:482–503.
- [10] Diethelm, K. and Ford, N. J. (2002). Analysis of fractional differential equations. *J. Math. Anal. Appl.*, 265(2):229–248.
- [11] Diethelm, K., Ford, N. J., and Freed, A. D. (2004). Detailed error analysis for a fractional adams method. *Numer. Algorithms*, 36(1):31–52.
- [12] Djordjević, V., Jarić, J., Fabry, B., Fredberg, J., and Stamenović, D. (2003). Fractional derivatives embody essential features of cell rheological behavior. *Ann. Biomed. Eng.*, 31(6):692–699.
- [13] Esmaeili, S., Shamsi, M., and Luchko, Y. (2011). Numerical solution of fractional differential equations with a collocation method based on müntz polynomials. *Comput. Math. Appl.*, 62(3):918–929.
- [14] Fu, H., Ng, M., and Wang, H. (2017). A divide-and-conquer fast finite difference method for space-time fractional partial differential equation. *Comput. Math. Appl.*, 73:1233–1242.
- [15] Gao, G.-h., Sun, Z.-z., and Zhang, H.-w. (2014). A new fractional numerical differentiation formula to approximate the caputo fractional derivative and its applications. *J. Comp. Phys.*, 259:33–50.
- [16] Gorenflo, R., Mainardi, F., Moretti, D., and Paradisi, P. (2002). Time fractional diffusion: a discrete random walk approach. *Nonlinear Dyn.*, 29(1-4):129–143.
- [17] Hei, X., Chen, W., Pang, G., Xiao, R., and Zhang, C. (2018). A new visco-elasto-plastic model via time-space derivative. *Mech. Time Depend. Mater.*, 22:129–141.
- [18] Hiptmair, R. and Schädle, A. (2003). Non-reflecting boundary conditions for maxwell’s equations. *Computing*, 71:265–292.
- [19] Hou, D. and Xu, C. (2017). A fractional spectral method with applications to some singular problems. *Adv. Comput. Math.*, 43(5):911–944.
- [20] Jaishankar, A. and McKinley, G. (2013). Power-law rheology in the bulk and at the inter-

- face: quasi-properties and fractional constitutive equations. *Proc R Soc A* 469: 20120284.
- [21] Jaishankar, A. and McKinley, G. (2014). A fractional k-bkz constitutive formulation for describing the nonlinear rheology of multiscale complex fluids. *J. Rheol.*, 58.
 - [22] Kharazmi, E. and Zayernouri, M. (2018). Fractional pseudo-spectral methods for distributed-order fractional PDEs. *Int. J. Comput. Math.*, pages 1–22.
 - [23] Kharazmi, E. and Zayernouri, M. (2019). Fractional sensitivity equation method: Applications to fractional model construction. *J. Sci. Comput.*, 80:110–140.
 - [24] Kharazmi, E., Zayernouri, M., and Karniadakis, G. E. (2017a). Petrov–Galerkin and spectral collocation methods for distributed order differential equations. *SIAM J. Sci. Comput.*, 39(3):A1003–A1037.
 - [25] Kharazmi, E., Zayernouri, M., and Karniadakis, G. E. (2017b). A Petrov–Galerkin spectral element method for fractional elliptic problems. *Comput. Methods Appl. Mech. Eng.*, 324:512–536.
 - [26] Kilbas, A. A., Srivastava, H. M., and Trujillo, J. J. (2006). *Theory and Applications of Fractional Differential Equations, Volume 204 (North-Holland Mathematics Studies)*. Elsevier Science Inc., New York, NY, USA.
 - [27] Li, J.-R. (2010). A fast time stepping method for evaluating fractional integrals. *SIAM J. Sci. Comput.*, 31(6):4696–4714.
 - [28] Liang, Y., Chen, W., Akpa, B., Neuberger, T., Webb, A., and Magin, R. (2017). Using spectral and cumulative spectral entropy to classify anomalous diffusion in sephadexTM gels. *Comput. Math. Appl.*, 73:765–774.
 - [29] Lin, Y. and Xu, C. (2007). Finite difference/spectral approximations for the time-fractional diffusion equation. *J. Comp. Phys.*, 225(2):1533–1552.
 - [30] Lischke, A., Zayernouri, M., and Karniadakis, G. (2017). A petrov–galerkin spectral method of linear complexity for fractional multiterm odes on the half line. *SIAM J. Sci. Comput.*, 39(3):A922–A946.
 - [31] Lubich, C. (1983). On the stability of linear multistep methods for Volterra convolution equations. *IMA J. Numer. Anal.*, 3(4):439–465.
 - [32] Lubich, C. (1986). Discretized fractional calculus. *SIAM J. Math. Anal.*, 17:704–719.
 - [33] Lubich, C. and Schädle, A. (2002). Fast convolution for nonreflecting boundary conditions. *SIAM J. Sci. Comput.*, 24:161–182.
 - [34] Meral, F., Royston, T., and Magin, R. (2010). Fractional calculus in viscoelasticity: An experimental study. *Commun. Nonlinear Sci. Numer. Simul.*, 15:939–945.
 - [35] Naghibolhosseini, M. (2015). *Estimation of outer-middle ear transmission using DPOAEs and fractional-order modeling of human middle ear*. PhD thesis, City University of New York, NY.
 - [36] Naghibolhosseini, M. and Long, G. (2018). Fractional-order modelling and simulation of human ear. *Int. J. Comput. Math.*, 95(6-7):1257–1273.
 - [37] Perdikaris, P. and Karniadakis, G. (2014). Fractional-order viscoelasticity in one-dimensional blood flow models. *Ann. Biomed. Eng.*, 42(5):1012–1023.
 - [38] Podlubny, I. (1999). *Fractional Differential Equations*. San Diego, CA, USA: Academic Press.
 - [39] Rodrigues, C., Suzuki, J., and Bittencourt, M. (2016). Construction of minimum energy high-order Helmholtz bases for structured elements. *J. Comp. Phys.*, 306:269–290.
 - [40] Samiee, M., Akhavan-Safaei, A., and Zayernouri, M. (2019a). A fractional subgrid-scale model for turbulent flows: Theoretical formulation and a priori study. *Phys. Fluids (accepted)*.
 - [41] Samiee, M., Kharazmi, E., Zayernouri, M., and Meerschaert, M. M. (2020). Petrov–Galerkin method for fully distributed-order fractional partial differential equations. *Commun. Appl. Math. Comput.* (accepted).
 - [42] Samiee, M., Zayernouri, M., and Meerschaert, M. M. (2019b). A unified spectral method for FPDEs with two-sided derivatives; part I: A fast solver. *J. Comp. Phys.*, 385:225–243.
 - [43] Samiee, M., Zayernouri, M., and Meerschaert, M. M. (2019c). A unified spectral method for FPDEs with two-sided derivatives; part II: Stability, and error analysis. *J. Comp. Phys.*,

- 385:244–261.
- [44] Samko, S. G., Kilbas, A. A., and Marichev, O. I. (1993). *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach Science Publishers, Switzerland.
 - [45] Schädle, A., López-Fernández, M., and Lubich, C. (2006). Fast and oblivious convolution quadrature. *SIAM J. Sci. Comput.*, 28(2):421–438.
 - [46] Stynes, M., O’Riordan, E., and Gracia, J. (2017). Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. *SIAM J. Numer. Anal.*, 55(2):1057–1079.
 - [47] Sugimoto, N. (1991). Burgers equation with a fractional derivative; hereditary effects on nonlinear acoustic waves. *J. Fluid Mech.*, 225(4):631–653.
 - [48] Sun, Z.-z. and Wu, X. (2006). A fully discrete difference scheme for a diffusion-wave system. *Appl. Numer. Math.*, 56(2):193–209.
 - [49] Suzuki, J. and Bittencourt, M. (2016). *Application of the hp-FEM for Hyperelastic Problems with Isotropic Damage*, pages 113–150. Springer International Publishing.
 - [50] Suzuki, J., Varghaei, P., Kharazmi, E., and Zayernouri, M. (2020a). Nonlinear vibration of fractional viscoelastic cantilever beam: Towards structural health monitoring. *Submitted to J. Vib. Acoust.*
 - [51] Suzuki, J., Zayernouri, M., Bittencourt, M., and Karniadakis, G. (2016). Fractional-order uniaxial visco-elasto-plastic models for structural analysis. *Comput. Methods Appl. Mech. Eng.*, 308:443 – 467.
 - [52] Suzuki, J., Zhou, Y., D’Elia, M., and Zayernouri, M. (2020b). A thermodynamically consistent fractional visco-elasto-plastic model with memory-dependent damage for anomalous materials. *Submitted to Comput. Methods Appl. Mech. Eng.*
 - [53] Weisstein, E.W. (2003). Regularized Hypergeometric Function.
 - [54] Yu, Y., Perdikaris, P., and Karniadakis, G. (2016). Fractional modeling of viscoelasticity in 3d cerebral arteries and aneurysms. *J. Comp. Phys.*, 323:219–242.
 - [55] Zayernouri, M., Cao, W., Zhang, Z., and Karniadakis, G. E. (2014). Spectral and discontinuous spectral element methods for fractional delay equations. *SIAM J. Sci. Comput.*, 36(6):B904–B929.
 - [56] Zayernouri, M. and Karniadakis, G. E. (2014). Exponentially accurate spectral and spectral element methods for fractional odes. *J. Comp. Phys.*, 257:460–480.
 - [57] Zayernouri, M. and Karniadakis, G. E. (2015). Fractional spectral collocation methods for linear and nonlinear variable order FPDEs. *J. Comp. Phys. (Special issue on FPDEs: Theory, Numerics, and Applications)*, 293:312–338.
 - [58] Zayernouri, M. and Matzavinos, A. (2016). Fractional adams-bashforth/moulton methods: An application to the fractional keller-segel chemotaxis system. *J. Comp. Phys.*, 317:1–14.
 - [59] Zeng, F., Turner, I., and Burrage, K. (2018). A stable fast time-stepping method for fractional integral and derivative operators. *J. Sci. Comput.*
 - [60] Zeng, F., Zhang, Z., and Karniadakis, G. (2015). A generalized spectral collocation method with tunable accuracy for variable-order fractional differential equations. *SIAM J. Sci. Comput.*, 37(6):A2710–A2732.
 - [61] Zeng, F., Zhang, Z., and Karniadakis, G. (2017). Second-order numerical methods for multi-term fractional differential equations: smooth and non-smooth solutions. *Comput. Methods Appl. Mech. Eng.*, 327:478–502.
 - [62] Zhang, Y., Sun, H., Stowell, H. H., Zayernouri, M., and Hansen, S. E. (2017). A review of applications of fractional calculus in earth system dynamics. *Chaos, Solitons & Fractals*, 102:29–46.
 - [63] Zhou, Y., Suzuki, J., Zhang, C., and Zayernouri, M. (2020). Implicit-explicit time integration of nonlinear fractional differential equations. *Appl. Numer. Math. (accepted)*.

Appendix A. Discretization Coefficients for Fractional Derivatives

We present the α - and Δt - dependent coefficients for the local (22) and history (23) approximations for the RL fractional derivative, which are given by [59]:

$$\begin{cases} d_0^{(1)} = -\frac{\alpha\Delta t^\alpha}{\Gamma(2+\alpha)}, & d_1^{(1)} = \frac{\Delta t^\alpha}{\Gamma(2+\alpha)}, & p = 1, \\ d_0^{(2)} = -\frac{\alpha\Delta t^\alpha}{2\Gamma(3+\alpha)}, & d_1^{(2)} = \frac{\alpha(3+\alpha)\Delta t^\alpha}{\Gamma(3+\alpha)}, & d_2^{(2)} = \frac{(4+\alpha)\Delta t^\alpha}{2\Gamma(3+\alpha)}, & p = 2. \end{cases} \quad (\text{A1})$$

$$b_j^{(1)} = \frac{\Delta t^\alpha}{2\Gamma(\alpha)} \left[a_j^{(\alpha+2)} - (2j-1) a_j^{(\alpha+1)} + j(j-1) a_j^{(\alpha)} \right], \quad (\text{A2})$$

$$b_j^{(2)} = -\frac{\Delta t^\alpha}{\Gamma(\alpha)} \left[a_j^{(\alpha+2)} - 2j a_j^{(\alpha+1)} + (j+1)(j-1) a_j^{(\alpha)} \right], \quad (\text{A3})$$

$$b_j^{(3)} = \frac{\Delta t^\alpha}{2\Gamma(\alpha)} \left[a_j^{(\alpha+2)} - (2j+1) a_j^{(\alpha+1)} + j(j+1) a_j^{(\alpha)} \right], \quad (\text{A4})$$

$$a_j^{(\alpha)} = \frac{1}{\alpha} [(j+1)^\alpha - j^\alpha]. \quad (\text{A5})$$

Appendix B. Singularity-Capturing Formulation for a Single Correction Term and Time-Step

Let the time domain $\tilde{\Omega} = [0, \Delta t]$, and let $E : \mathbb{R} \rightarrow \mathbb{R}^+$ be the following quadratic error function over σ :

$$E(\sigma) = \|u^{data}(t) - u^N(t; \sigma)\|_{L^2(\tilde{\Omega})}^2, \quad (\text{B1})$$

where we assume $u^{data}(t)$ to be known, and $u^N(t; \sigma)$ represents the numerical approximation of $u^{data}(t)$, with $\sigma \in \mathbb{R}$ to be determined, such that $E(\sigma)$ is minimized. Therefore, let the case for $M = 1$ correction term and for the first time-step $t_1 = \Delta t$. We obtain the following cost function:

$$E(\sigma) = \left(u_1^{data} - u_1^N(\sigma) \right)^2 = \left(\Delta t^{\sigma^*} - u_1^N(\sigma) \right)^2, \quad (\text{B2})$$

where $u_1^N(\sigma)$ denotes the numerical solution for the FDE (9) at $t = \Delta t$, obtained through (19) and (35) in the following fashion:

$${}^{RL}_0\mathcal{D}_t^\alpha u^N(t)|_{t=\Delta t} + W_{1,1} (u_1^N - u_0) = \sum_{j=1}^S \frac{\Gamma(1 + \sigma_j^*)}{\Gamma(1 + \sigma_j^* - \alpha)} \Delta t^{\sigma_j^* - \alpha}. \quad (\text{B3})$$

Recalling (17), we obtain the initial correction weight $W_{1,1}$ by the following equation:

$${}^{RL}_0\mathcal{D}_t^\alpha (t^{\sigma_1})|_{t=\Delta t} + W_{1,1} (t_1^\sigma) = \frac{\Gamma(1 + \sigma_1)}{\Gamma(1 + \sigma_1 - \alpha)} t_1^{\sigma_1 - \alpha}, \quad (\text{B4})$$

Recalling (22) with $p = 1$:

$$d_1^{(1)} \Delta t^{\sigma_1} + \Delta t^{\sigma_1} W_{1,1} = \frac{\Gamma(1 + \sigma_1)}{\Gamma(1 + \sigma_1 - \alpha)} \Delta t^{\sigma_1 - \alpha}, \quad (\text{B5})$$

therefore, we obtain the closed form for the correction weight:

$$W_{1,1} = \frac{\Gamma(1 + \sigma_1)}{\Gamma(1 + \sigma_1 - \alpha)} \Delta t^{-\alpha} - d_1^{(1)}, \quad (\text{B6})$$

Substituting (B6) into (B3), rewriting the RL derivative in its discretized form at $t = \Delta t$, and assuming homogeneous initial conditions $u_0 = 0$, we obtain:

$$d_1^{(1)} u_1^N + \left(\frac{\Gamma(1 + \sigma_1)}{\Gamma(1 + \sigma_1 - \alpha)} \Delta t^{-\alpha} - d_1^{(1)} \right) u_1^N = f_1^{data}, \quad (\text{B7})$$

hence,

$$u_1^N = \Delta t^\alpha \frac{\Gamma(1 + \sigma_1 - \alpha)}{\Gamma(1 + \sigma_1)} f_1^{data}. \quad (\text{B8})$$

Substituting (B8) into (B2), we obtain, for $t = \Delta t$:

$$E(\sigma_1) = \left(u_1^{data} - \Delta t^\alpha \frac{\Gamma(1 + \sigma_1 - \alpha)}{\Gamma(1 + \sigma_1)} f_1^{data} \right)^2. \quad (\text{B9})$$

We assume that the above equation has a root at $\sigma_1 = \sigma^*$, that is, $E(\sigma^*) = 0$. Therefore, given an initial guess $\sigma_1^{(0)}$, we can apply a Newton scheme to iteratively obtain σ_1^{k+1} in the following fashion:

$$\sigma_1^{k+1} = \sigma_1^k - \frac{E(\sigma_1)}{\frac{\partial E(\sigma_1)}{\partial \sigma_1}} \Big|_{\sigma_1 = \sigma_1^k}, \quad (\text{B10})$$

where $\frac{\partial E(\sigma_1)}{\partial \sigma_1}$ can be obtained analytically, and is given by:

$$\frac{\partial E(\sigma_1)}{\partial \sigma_1} = -2\psi^* \Delta t^\alpha \frac{\Gamma(1 + \sigma_1 - \alpha)}{\Gamma(1 + \sigma_1)} \left(u_1^{data} - \Delta t^\alpha \frac{\Gamma(1 + \sigma_1 - \alpha)}{\Gamma(1 + \sigma_1)} \right), \quad (\text{B11})$$

with $\psi^* = \psi_0(1 + \sigma_1 - \alpha) - \psi_0(1 + \sigma_1)$, where $\psi_0(z)$ denotes the Polygamma function. We observe that the computational cost of the above procedure per iteration k is minimal, since there is no history computation when evaluating the corresponding time-fractional derivatives of u_1^N .