fractal and fractional

MDPI

Article

A Novel Numerical Method for Solving Nonlinear Fractional-Order Differential Equations and Its Applications

Seyeon Lee [1], **Hyunju Kim** [2] **and Bongsoo Jang** [3,*]

[1] Division of Industrial Mathematics, National Institute for Mathematical Sciences (NIMS) ,
Daejeon 34047, Republic of Korea; sylee218@nims.re.kr

[2] Department of Energy Engineering, Korea Institute of Energy Technology (KENTECH),
Naju 58217, Republic of Korea; hjkim@kentech.ac.kr

[3] Department of Mathematical Sciences, Ulsan National Institute of Science and Technology (UNIST),
Ulsan 44919, Republic of Korea

* Correspondence: bsjang@unist.ac.kr; Tel.: +82-52-217-3136

Abstract: In this article, a considerably efficient predictor-corrector method (PCM) for solving Atangana–Baleanu Caputo (ABC) fractional differential equations (FDEs) is introduced. First, we propose a conventional PCM whose computational speed scales with quadratic time complexity $\mathcal{O}(N^2)$ as the number of time steps N grows. A fast algorithm to reduce the computational complexity of the memory term is investigated utilizing a sum-of-exponentials (SOEs) approximation. The conventional PCM is equipped with a fast algorithm, and it only requires linear time complexity $\mathcal{O}(N)$. Truncation and global error analyses are provided, achieving a uniform accuracy order $\mathcal{O}(h^2)$ regardless of the fractional order for both the conventional and fast PCMs. We demonstrate numerical examples for nonlinear initial value problems and linear and nonlinear reaction-diffusion fractional-order partial differential equations (FPDEs) to numerically verify the efficiency and error estimates. Finally, the fast PCM is applied to the fractional-order Rössler dynamical system, and the numerical results prove that the computational cost consumed to obtain the bifurcation diagram is significantly reduced using the proposed fast algorithm.

Keywords: Atangana–Baleanu fractional derivative; fractional differential equations; predictor-corrector methods; sum-of-exponentials approximation; sub-diffusion equation.

check for updates

Citation: Lee, S.; Kim, H.; Jang, B. A Novel Numerical Method for Solving Nonlinear Fractional-Order Differential Equations and Its Applications. *Fractal Fract.* **2024**, *8*, 65. https://doi.org/10.3390/fractalfract8010065

Academic Editor: Rekha Srivastava

Received: 26 December 2023
Revised: 12 January 2024
Accepted: 13 January 2024
Published: 17 January 2024

1. Introduction

Fractional-order differential equations (FDEs) have been successfully applied in physics, biology, applied sciences, and engineering where memory effects are essential. FDEs are considered a novel mathematical model and are used to describe various phenomena in nature, such as anomalous transport [1–3], the flow in porous media [4], and so on. Some works [5–9] have demonstrated further applications, including in electrical circuits, chemistry, finance, chaos, control theory, heat transfer, gas dynamics, etc. The aforementioned applications adopted the Grünwald–Letnikov, Riemann–Liouville, or Liouville–Caputo fractional operators [5,6,8]. The definition of the Liouville–Caputo fractional operator is as follows:

Definition 1. *The Liouville–Caputo fractional derivative (Liouville–Caputo FD) of order $\nu \in \mathbb{R}^+$, $n-1 \le \nu < n$, for $u(t)$, which was introduced in [6,10], is defined by*

$$ {}^{C}\mathcal{D}_a^{\nu} u(t) = \frac{1}{\Gamma(n-\nu)} \int_a^t (t-s)^{n-\nu-1} u^{(n)}(s) ds. \tag{1}$$

Recently, new fractional differential operators with a nonlocal and non-singular kernel, e.g., Caputo–Fabrizio and Atangana–Baleanu, were proposed in [9,11]. Also, the more general Riemann–Liouville and Liuoville–Caputo fractional derivatives were introduced in [12,13].

In [14], Atangana and Baleanu suggested a new fractional differential operator involving the Mittag–Leffler kernel and applied it to a heat transfer model. This operator has been applied in many research areas [15–17]. In this paper, we aim to solve the nonlinear fractional differential equation of order $\nu \in (0,1)$,

$$\begin{cases} {}^{ABC}\mathcal{D}_a^\nu u(t) = f(t, u(t)), & t \in [a, T], \\ u(a) = u_0, \end{cases} \tag{2}$$

where ${}^{ABC}\mathcal{D}_a^\nu$ is the Atangana–Baleanu fractional derivative of order ν in the Liouville–Caputo sense (ABC FD):

Definition 2. *The Atangana–Baleanu fractional derivative of order* $\nu \in (0,1)$ in the Caputo sense *for* $u(t)$*, which was introduced in [14], is defined by*

$$ {}^{ABC}\mathcal{D}_a^\nu u(t) = \frac{B(\nu)}{1-\nu} \int_a^t E_\nu \left[-\nu \frac{(t-s)^\nu}{1-\nu} \right] u'(s) ds, \quad u(t) \in H^1(a, T) \tag{3}$$

where $B(\nu)$ *is a normalization function with* $B(0) = 1$ *and* $B(1) = 1$*, and* $E_\nu(z)$ *is the one-parameter Mittag–Leffler function defined by*

$$E_\nu(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\nu k + 1)}, \quad \nu > 0.$$

By taking the Laplace transform of (2) and then performing the inverse Laplace transform, we obtain the solution $u(t)$ of the model problem (2) as the Volterra integral equation of the second kind [14], as follows:

$$u(t) = \phi(t, f(t)) + \frac{d_\nu}{\Gamma(\nu)} \int_a^t (t-s)^{\nu-1} f(s, u(s)) ds, \tag{4}$$

where

$$\phi(t, f(t)) = u(a) + \frac{1-\nu}{B(\nu)} f(t, u(t)), \quad d_\nu = \frac{\nu}{B(\nu)}.$$

Numerical methods for solving the ABC FDE (2) have been introduced by many researchers. The authors of [18] suggested utilizing the Crank–Nicholson schemes for solving the groundwater flow model with the ABC fractional derivative (3). The spectral collocation method, based on the Chebyshev approximations, was employed to handle a biological fractional model [19]. The authors of [20] developed the fractional Euler method and the predictor-corrector method (PCM), which is similar to the method in [21]. The PCM, using the product trapezoidal quadrature rule and the product rectangle rule for solving ABC fractional initial value problems, was introduced in [22]. The modified PCM was designed to be very accurate. Its performance stays the same regardless of the order of the fractional derivative, as explained in [23]. This method has worked well and has been used effectively in different research studies, for example, in [24,25].

Other numerical methods were suggested in [11,26–34].

This paper focuses on the following three topics:

1. The development of a fast algorithm that can be applied to numerical methods for solving ABC FDEs.
2. The development of a fast PCM and its application to ABC fractional-order PDEs (FPDEs) and fractional dynamical systems.
3. Error estimates for both conventional and fast PCMs.

We introduce the fast algorithm regarding the computation of memory terms resulting from the non-local kernel in the ABC fractional derivative. The fast algorithm is integrated with the conventional PCM for solving the Volterra integral equation (4). To illustrate the idea of the proposed fast algorithm, we assume $a = 0$ and discretize the uniform grid, as follows:

$$\Psi_N := \{t_j : 0 = t_0 < t_1 < \ldots < t_j < \ldots < t_{n+1} < \cdots < t_N = T\},$$

with $h = t_j - t_{j-1}$. Equation (4) can be rewritten at time t_{n+1}, as follows

$$u(t_{n+1}) = \phi(t_{n+1}) + u^{mem}(t_{n+1}) + u^{loc}(t_{n+1}), \tag{5}$$

where

$$u^{mem}(t_{n+1}) = \frac{d_\nu}{\Gamma(\nu)} \int_0^{t_n} (t_{n+1} - s)^{\nu-1} f(s, u(s)) ds, \tag{6}$$

$$u^{loc}(t_{n+1}) = \frac{d_\nu}{\Gamma(\nu)} \int_{t_n}^{t_{n+1}} (t_{n+1} - s)^{\nu-1} f(s, u(s)) ds. \tag{7}$$

In the numerical approximation of $u(t_{n+1})$, the memory term (6) has to be computed from t_0 to t_n for every n due to the non-local property of the kernel $(t_{n+1} - s)^{\nu-1}$. Thus, it is easy to see that a computational cost of $\mathcal{O}(n^2)$ is incurred if linear interpolation of $g(s, u(s))$ over $I_j = [t_j, t_{j+1}]$ is employed. To reduce the computational cost, we modify the sum-of-exponentials (SOEs) approximation for $t^{-\beta}$ $(0 < \beta < 2)$ [35] and apply the algorithm to the memory term $u^{mem}(t_{n+1})$:

$$u^{mem}(t_{n+1}) \approx \frac{d_\nu}{\Gamma(\nu)} \sum_{i=1}^{N_{\exp}} \zeta_i \int_0^{t_n} e^{-\eta_i (t_{n+1}-s)} f(s, u(s)) ds, \tag{8}$$

where η_i and ζ_i $(i = 1, ..., N_{\exp})$ are positive real numbers. Here, the integral term in (8) can be rewritten as the following recurrence relation:

$$\int_0^{t_n} e^{-\eta_i (t_{n+1}-s)} f(s, u(s)) ds = e^{-\eta_i h} \int_0^{t_{n-1}} e^{-\eta_i (t_n - s)} f(s, u(s)) ds + \int_{t_{n-1}}^{t_n} e^{-\eta_i (t_{n+1}-s)} f(s, u(s)) ds.$$

It is worth noting that there exists a fixed number $N_{\exp}$ for all $t \in [\delta, T]$ such that the error in (8) is given, where $\delta = h$. Using the equation in (9), the integral term on the left-hand side can be evaluated recursively. Because of these facts, the computational cost to approximate the memory term $u^{mem}(t_{n+1})$ is reduced by up to $\mathcal{O}(n)$, even if the total number of time steps N is large. The highlights of this research, including the advantages of the proposed fast PCM, are as follows:

1. The conventional PCM for solving ABC FDEs is suggested.
2. The fast algorithm for the computation of the memory term is proposed, and the fast PCM only requires a computational cost of $\mathcal{O}(n)$.
3. Truncation and global error analyses for both conventional and fast PCMs are provided, achieving a uniform accuracy of order $\mathcal{O}(h^2)$ regardless of the fractional order ν.
4. We apply the proposed fast PCM to sub-diffusion FPDEs to demonstrate the efficiency of the proposed method.
5. The proposed fast PCM is implemented to handle the fractional Rössler dynamical system, and the performance of the fast algorithm is verified.

Our paper consists of the following sections. In Section 2, we succinctly describe a conventional PCM for solving FDEs with the ABC FD. We propose a modified SOE approximation of the power function based on [35]. We introduce a fast PCM for solving FDEs with the ABC FD. The error analyses for the suggested methods are provided in Section 3. In Section 4, numerical results are provided to support our theoretical results. Applications to reaction-diffusion FPDEs and the fractional Rössler dynamical system are also introduced. Finally, the conclusions are presented in Section 5.

2. Fast Predictor-Corrector Scheme

2.1. Description of Predictor-Corrector Scheme

First, let us define some notations. For each $j = 0, \cdots, N$, we denote $u_j = u(t_j)$, $g_j = g(t_j, u_j)$. Let $\tilde{u}_j^P$, and $\tilde{u}_j$ be the predicted and corrected approximations of u_j, respectively. Similarly, we also denote $\tilde{f}_j^P = f(t_j, \tilde{u}_j^P)$ and $\tilde{f}_j = f(t_j, \tilde{u}_j)$. On each interval $I_j = [t_j, t_{j+1}]$, we employ the Lagrange interpolating polynomial of $f(t, u(t))$, defined by

$$f(t) = \Pi_j f(t) + \mathcal{E}_j f(t), \tag{9}$$

where

$$\begin{aligned}\Pi_j f(t) &= \frac{t_{j+1} - t}{h} f_j + \frac{t - t_j}{h} f_{j+1},\\ \mathcal{E}_j f(t) &= \frac{f^{(2)}(\eta_j)}{2!}(t - t_j)(t - t_{j+1}), \quad \eta_j \in (t_j, t_{j+1}).\end{aligned}$$

The value of $\phi(t_{n+1}, f(t_{n+1}))$ can be expressed by using the linear interpolation of f on $I_{n-1} = [t_{n-1}, t_n]$

$$\phi(t_{n+1}, f(t_{n+1})) = \Phi_{n+1} + \mathcal{G}_{n+1}, \tag{10}$$

where

$$\Phi_{n+1} = u(0) + \frac{1-\nu}{B(\nu)}(-f_{n-1} + 2f_n), \quad \mathcal{G}_{n+1} = \frac{1-\nu}{B(\nu)} f^{(2)}(\eta_n, u(\eta_n))h^2, \quad \eta_n \in (t_{n-1}, t_n).$$

By substituting (9) and (10) into (5), we have

$$u_{n+1} = \Phi_{n+1} + \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} [\Theta_{n+1}^{0,j} f_j + \Theta_{n+1}^{1,j} f_{j+1}] + \frac{d_\nu}{\Gamma(\nu)} [\Theta_{n+1}^{0,n} f_n + \Theta_{n+1}^{1,n} f_{n+1}] + \mathcal{T}_{n+1}, \tag{11}$$

where

$$\begin{aligned}\Theta_{n+1}^{0,j} &= \int_{t_j}^{t_{j+1}} (t_{n+1} - s)^{\nu-1} \frac{t_{j+1} - s}{h} ds,\\ \Theta_{n+1}^{1,j} &= \int_{t_j}^{t_{j+1}} (t_{n+1} - s)^{\nu-1} \frac{s - t_j}{h} ds,\\ \mathcal{T}_{n+1} &= \mathcal{G}_{n+1} + \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n} \int_{t_j}^{t_{j+1}} (t_{n+1} - s)^{\nu-1} \mathcal{E}_j f(s, u(s)) ds.\end{aligned}$$

Let us rewrite (11), as follows:

$$u_{n+1} = \Phi_{n+1} + u_{n+1}^{mem} + u_{n+1}^{loc} + \mathcal{T}_{n+1},$$

where

$$u_{n+1}^{mem} = \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} [\Theta_{n+1}^{0,j} f_j + \Theta_{n+1}^{1,j} f_{j+1}], \quad u_{n+1}^{loc} = \frac{d_\nu}{\Gamma(\nu)} [\Theta_{n+1}^{0,n} f_n + \Theta_{n+1}^{1,n} f_{n+1}]. \tag{12}$$

Here, we employ the modified predictor-corrector method from [23] in order to obtain the second order of convergence. Specifically, referring to Lemma 1 [23], we have

$$U_{n+1}^{loc} = \frac{d_\nu}{\Gamma(\nu)} \int_{t_n}^{t_{n+1}} (t_{n+1} - s)^{\nu-1} \Pi_{n-1} f(s, u(s)) ds = \frac{d_\nu}{\Gamma(\nu+2)} h^\nu [-f_{n-1} + (\nu+2) f_n]. \tag{13}$$

Then, the predictor-corrector scheme of (5) is given as follows:

$$\begin{aligned}\text{Corrector: } \tilde{u}_{n+1} &= \phi(t_{n+1}, \tilde{f}^P_{n+1}) + \tilde{u}^{mem}_{n+1} + \tilde{u}^{loc}_{n+1},\\ \text{Predictor: } \tilde{u}^P_{n+1} &= \tilde{\Phi}_{n+1} + \tilde{u}^{mem}_{n+1} + \widetilde{U}^{loc}_{n+1},\end{aligned} \tag{14}$$

where

$$\tilde{u}^{mem}_{n+1} = \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} [\Theta^{0,j}_{n+1}\tilde{f}_j + \Theta^{1,j}_{n+1}\tilde{f}_{j+1}], \quad \tilde{u}^{loc}_{n+1} = \frac{d_\nu}{\Gamma(\nu)} [\Theta^{0,n}_{n+1}\tilde{f}_n + \Theta^{1,n}_{n+1}\tilde{f}^P_{n+1}],$$

$$\tilde{\Phi}_{n+1} = u(0) + \frac{1-\nu}{B(\nu)}(-\tilde{f}_{n-1} + 2\tilde{f}_n), \quad \widetilde{U}^{loc}_{n+1} = \frac{d_\nu}{\Gamma(\nu+2)} h^\nu [-\tilde{f}_{n-1} + (\nu+2)\tilde{f}_n].$$

2.2. Sum-of-Exponentials Approximation of the Power Function

In this section, we describe the modified version of the approximation of the kernel function $t^{-\beta}$ based on [35]. First, we give a summary of the effective SOE approximation for $t^{-\beta}$ $(0 < \beta < 2)$.

Lemma 1 ([35]). *For any $\beta > 0$,*

$$\frac{1}{t^\beta} = \frac{1}{\Gamma(\beta)} \int_0^\infty e^{-ts} s^{\beta-1} ds$$

Lemma 2 ([35]). *For $t \geq \delta > 0$, there exists $p > 0$ such that*

$$\left| \frac{1}{t^\beta} - \frac{1}{\Gamma(\beta)} \int_0^p e^{-ts} s^{\beta-1} ds \right| \leq \begin{cases} e^{-\delta p} \dfrac{p^{\beta-1}}{\Gamma(\beta)} \dfrac{1}{\delta}, & 0 < \beta \leq 1, \\ e^{-\delta p} 2^{\beta-1} \left(\dfrac{p^\beta}{\Gamma(\beta)} + \dfrac{1}{\delta^\beta} \right), & 1 < \beta < 2. \end{cases} \tag{15}$$

In the original SOE approximation in [35], the upper limit of the definite integral p in (15) was $p = 2^{N+1}$. The interval $[0, p]$ was split into three dyadic intervals: $[0, 2^{-M}]$, $\bigcup_{j=-M}^{-1} [2^j, 2^{j+1}]$, and $\bigcup_{j=0}^{N} [2^j, 2^{j+1}]$. The Gauss–Jacobi and Gauss–Legendre quadrature rules were employed in the three intervals. Here, we normalize the interval $[0, p]$ into $[0, 1]$. That is,

$$\frac{1}{\Gamma(\beta)} \int_0^p e^{-ts} s^{\beta-1} ds = \frac{p^\beta}{\Gamma(\beta)} \int_0^1 e^{-tp\tau} \tau^{\beta-1} d\tau.$$

Recalling $M = \mathcal{O}(\log T)$ and $N = \mathcal{O}(\log\log \frac{1}{\epsilon} + \log \frac{1}{\delta})$ from Theorem 1 ([35]), we obtain following modified SOE approximation by setting $\hat{M} = M + N + 1$:

Theorem 1 (Modified SOE approximation of the power function). *thmmsoeapp Let $\epsilon > 0$ be the desired error, and choose $0 < \delta \leq t \leq T$. Let $n = \mathcal{O}(\log \frac{1}{\epsilon})$; $\hat{M} = \mathcal{O}(\log T + \log\log \frac{1}{\epsilon} + \log \frac{1}{\delta}) + 1$; η^J_i and ζ^J_i be n-point Gauss–Jacobi quadratures on the interval $[0, 2^{-\hat{M}}]$; and $\eta^L_{i,j}$ and $\zeta^L_{i,j}$ be n-point Gauss–Legendre quadrature points and weights on the intervals $[2^j, 2^{j+1}]$ for $j = -\hat{M}, \cdots, -1$. Then, for $t \in [\delta, T]$ and $\beta \in (0, 2)$,*

$$\left| \frac{1}{t^\beta} - \frac{p^\beta}{\Gamma(\beta)} \left[\sum_{i=1}^n \zeta^J_i e^{-tp\eta^J_i} + \sum_{j=-\hat{M}}^{-1} \sum_{i=1}^n \zeta^L_{i,j} e^{-tp\eta^L_{i,j}} (\eta^L_{i,j})^{\beta-1} \right] \right| \leq \epsilon. \tag{16}$$

Remark 1. *We use $n = \mathcal{O}(\log\frac{1}{\epsilon})$ Gauss–Legendre quadrature nodes and weights because all dyadic intervals $[2^i, 2^{j+1}], j = -\hat{M}, \cdots, -1$ are less than or equal to* 1*. Theorem 1 can be written as follows for some positive real numbers η_i and ζ_i*

$$\left| \frac{1}{t^\beta} - \sum_{i=1}^{N_{\exp}} \zeta_i e^{-\eta_i t} \right| \leq \epsilon, \quad t \in [\delta, T], \tag{17}$$

where $N_{\exp}$ is the number of exponentials to be determined later and $\delta = h$.

As mentioned in [35], the number of exponentials $N_{\exp}$ is required to approximate $1/t^\beta$ because many nodes and weights are computed in each dyadic interval. Here, we apply the reduction method based on the balanced truncation method with singular-value decomposition instead of Cholesky factorization (see the details in Appendix A). Tables 1 and 2 show comparisons between our method and the method in [35] regarding the number of exponentials $N_{\exp}$ for approximating $t^{-\nu-1}$ for a fixed $\delta = 10^{-3}$ (Table 1) and for a fixed $T = 1$ (Table 2). In most cases, the number of exponentials $N_{\exp}$ in our method is comparably smaller than the number of exponentials $N_{\exp}$ in [35]. For the comparisons, we set $\hat{M} = \lceil \log(\log(1/\epsilon)) + \log(1/\delta^2) + \log T \rceil + 1$ and used $n = \lceil \log(1/\epsilon) \rceil$ Gauss–Legendre nodes with corresponding weights.

Table 1. Comparison of the number of exponentials N_{exp} for approximating $t^{-1-\nu}$ with $\delta = 10^{-3}$.

T/δ	10^3	10^4	10^5	10^6	10^3	10^4	10^5	10^6
ϵ	$\nu = 0.2$				$\nu = 0.5$			
The number of exponentials $N_{\exp}$ in [35]								
10^{-3}	25	29	33	37	30	35	40	44
10^{-6}	34	40	45	51	39	46	52	59
10^{-9}	43	51	58	66	49	57	65	74
The number of exponentials $N_{\exp}$ in our method								
10^{-3}	18	19	21	23	19	21	23	25
10^{-6}	30	34	37	43	31	34	39	43
10^{-9}	45	50	53	60	45	50	54	60

Table 2. Comparison of the number of exponentials N_{exp} for approximating $t^{-1-\nu}$ with $T = 1$.

δ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
ϵ	$\nu = 0.2$				$\nu = 0.5$			
The number of exponentials $N_{\exp}$ in [35]								
10^{-3}	25	34	43	55	30	41	52	66
10^{-6}	34	45	58	69	39	51	65	81
10^{-8}	40	51	64	79	43	57	72	84
The number of exponentials $N_{\exp}$ in our method								
10^{-3}	18	21	25	30	19	21	27	30
10^{-6}	30	37	43	49	31	37	44	50
10^{-8}	40	48	56	82	41	48	58	82

2.3. Description of Fast Predictor-Corrector Scheme

In this section, we describe the fast PCM, which boosts the computational speed of the memory term u_{n+1}^{mem}. To do this, we use the SOE approximation of $t^{-\beta}$ $(0 < \beta < 2)$ introduced in Section 2.2. For simplicity, let $\epsilon > 0$ and $\delta > 0$ be given. Then, we have the following: for $t \in [\delta, T]$,

$$\frac{1}{t^\beta} - \sum_{i=1}^{N_{\exp}} \zeta_i e^{-\eta_i t} = \epsilon, \tag{18}$$

By applying (18) to u_{n+1}^{mem} with $\beta = 1-\nu \in (0,1)$, we have

$$\begin{aligned}
u_{n+1}^{mem} &= \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} (t_{n+1}-s)^{\nu-1} \Pi_j f(s,u(s))ds \\
&= \frac{d_\nu}{\Gamma(\nu)} \sum_{i=1}^{N_{\exp}} \zeta_i \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e^{-\eta_i(t_{n+1}-s)} \Pi_j f(s,u(s))ds + \epsilon \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} \Pi_j f(s,u(s))ds \\
&= \frac{d_\nu}{\Gamma(\nu)} \sum_{i=1}^{N_{\exp}} \zeta_i F_{n+1}^i + T_{n+1}^{\epsilon} \qquad (19)
\end{aligned}$$

where

$$\begin{aligned}
F_{n+1}^i &= \sum_{j=0}^{n-1} [\Psi_{i,n+1}^{0,j} f_j + \Psi_{i,n+1}^{1,j} f_{j+1}], \quad T_{n+1}^{\epsilon} = \epsilon \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} \Pi_j f(s,u(s))ds, \\
\Psi_{i,n+1}^{0,j} &= \int_{t_j}^{t_{j+1}} e^{-\eta_i(t_{n+1}-s)} \frac{t_{j+1}-s}{h} ds, \quad \Psi_{i,n+1}^{1,j} = \int_{t_j}^{t_{j+1}} e^{-\eta_i(t_{n+1}-s)} \frac{s-t_j}{h} ds.
\end{aligned}$$

Since $t_{n+1} = t_n + h$ and

$$F_{n+1}^i = \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e^{-\eta_i(t_{n+1}-s)} \Pi_j f(s,u(s))ds,$$

it is easy to determine that F_{n+1}^i has the following recurrence relation:

$$F_{n+1}^i = e^{-\eta_i h} F_n^i + \int_{t_{n-1}}^{t_n} e^{-\eta_i(t_{n+1}-s)} \Pi_{n-1} f(s)ds, \quad i = 1,2,\cdots,N_{\exp}. \qquad (20)$$

Remark 2. *It is easy to see that u_{n+1}^{mem} is approximated by the sum of piecewise interpolations over n for each interval. That is, the computational cost for approximating u_{n+1}^{mem} is about $\mathcal{O}(n^2)$. From the recurrence relation in (20), F_{n+1}^i can be obtained by computing the interpolation over the interval I_{n-1} combined with F_n^i, which was computed at step n. Then, u_{n+1}^{mem} can be determined as the sum of F_{n+1}^i for all i, $i = 1,\cdots,N_{exp}$. It costs $\mathcal{O}(nN_{\exp})$ to approximate u_{n+1}^{mem} using the recurrence relation. For $n \gg N_{\exp}$, the computational cost for approximating u_{n+1}^{mem} is about $\mathcal{O}(n)$.*

Let the approximate solutions of $u(t)$ and $f(t,u(t))$ at $t = t_{n+1}$, obtained using the proposed PCM (14), replacing the memory term $\tilde{u}_{n+1}^{mem}$ with the summation term in (19) using the recurrence relation (20), be denoted by $\hat{u}_{n+1}$ and $\hat{f}_{n+1}$, respectively. Then, the fast PCM of (4) is proposed as follows

$$\begin{aligned}
&\text{Corrector: } \hat{u}_{n+1} = \phi(t_{n+1}, \hat{f}_{n+1}^P) + \hat{u}_{n+1}^{fmem} + \hat{u}_{n+1}^{loc}, \\
&\text{Predictor: } \hat{u}_{n+1}^P = \hat{\phi}_{n+1} + \hat{u}_{n+1}^{fmem} + \widehat{U}_{n+1}^{loc},
\end{aligned} \qquad (21)$$

where

$$\begin{aligned}
\hat{u}_{n+1}^{fmem} &= \frac{d_\nu}{\Gamma(\nu)} \sum_{i=1}^{N_{\exp}} \zeta_i \widehat{F}_{n+1}^i, \quad \widehat{F}_{n+1}^i = e^{-\eta_i h} \widehat{F}_n^i + [\Psi_{i,n+1}^{0,n-1} \hat{f}_{n-1} + \Psi_{i,n+1}^{1,n-1} \hat{f}_n], \\
\hat{u}_{n+1}^{loc} &= \frac{d_\nu}{\Gamma(\nu)} [\Theta_{n+1}^{0,n} \hat{f}_n + \Theta_{n+1}^{1,n} \hat{f}_{n+1}^P], \\
\hat{\phi}_{n+1} &= u(0) + \frac{1-\nu}{B(\nu)}(-\hat{f}_{n-1} + 2\hat{f}_n), \quad \widehat{U}_{n+1}^{loc} = \frac{d_\nu}{\Gamma(\nu+2)} h^\nu [-\hat{f}_{n-1} + (\nu+2)\hat{f}_n].
\end{aligned}$$

3. Error Analysis

3.1. Truncation and Global Error Analyses for the Conventional PCM

In this section, we show that the proposed conventional PCM has a uniform convergence rate regardless of ν. First, we recall the error analysis of the second-order predictor-corrector scheme with linear interpolation for solving FDEs with the Liouville–Caputo FD in [23] and state the necessary lemmas for the error analysis. Herein, we denote by C a generic constant that is independent of all grid parameters and may change case by case.

Lemma 3 (Discrete Gronwall's inequality [36])**.** *Let $\{a_n\}_{n=0}^{N}$, $\{b_n\}_{n=0}^{N}$ be non-negative sequences with monotonically increasing b_n, satisfying*

$$a_n \leq b_n + Mh^{\gamma} \sum_{j=0}^{n-1} (n-j)^{\gamma-1} a_j, \quad 0 \leq n \leq N,$$

where $M > 0$ is bounded and independent of h, and $0 < \gamma \leq 1$. Then,

$$a_n \leq b_n E_{\gamma}(M\Gamma(\gamma)(nh)^{\gamma}).$$

Lemma 4. *For $0 < \nu < 1$,$k = 0, 1$ and $j = 0, 1, \cdots, n-1$,*

$$|\Theta_{n+1}^{k,j}| \leq (n-j)^{\nu-1} h^{\nu}.$$

It is worth noting that the Volterra integral equation for the model problem in (4) is similar to the form presented to (1.3) in [23], as is the PCM algorithm. Here, $\phi(t, u(t)) = u_0$ and $d_\nu = 1$. Therefore, it is straightforward to derive the following error analysis results.

Lemma 5 (Truncation error for the predictor [23])**.** *Suppose that $f \in C^2[0, T]$. Let the truncation error of the predictor at t_{n+1} be*

$$r_{n+1}^{P} = \Big| u_{n+1} - (\phi_{n+1} + u_{n+1}^{mem} + U_{n+1}^{loc}) \Big|.$$

Then, there exists a constant C independent of all grid parameters such that

$$r_{n+1}^{P} \leq Ch^2.$$

Lemma 6 (Global error for the predictor [23])**.** *Suppose that $f \in C^2[0, T]$ satisfies the Lipschitz continuity condition in the second argument,*

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|, \quad \forall y_1, y_2 \in \mathbb{R},\ L > 0,$$

then, the global error for the predictor, $e_{n+1}^{P} = |u_{n+1} - \tilde{u}_{n+1}^{P}|$, is given by

$$e_{n+1}^{P} \leq r_{n+1}^{P} + C\frac{Lh^{\nu}}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} e_j, \quad e_j = |u_j - \tilde{u}_j|.$$

Lemma 7 (Truncation error for the corrector [23])**.** *Under the same assumptions as those in Lemma 5, the truncation error of the corrector at time t_{n+1} is given by*

$$r_{n+1}^{C} = \Big| u_{n+1} - (\phi_{n+1} + u_{n+1}^{mem} + u_{n+1}^{loc}) \Big|,$$

and can be expressed as

$$r_{n+1}^{C} \leq \frac{MT^{\nu}}{2\Gamma(\nu+2)} h^2 + \frac{Lh^{\nu} e_{n+1}^{P}}{\Gamma(\nu+1)},$$

where $||f^{(2)}||_{\infty} \leq M$.

Lemma 8 (Global error for the corrector [23])**.** *Under the same assumptions as those in Lemma 6, the global error $e_{n+1} = |u_{n+1} - \tilde{u}_{n+1}|$ is given by*

$$e^C_{n+1} \leq Ch^2,$$

given that the starting error $e^C_1 \leq Ch^2$.

Now, using the aforementioned lemmas with the substitutions $\phi(t) = u(0) + \frac{1-\nu}{B(\nu)} f(t, u(t))$, $d_\nu = \frac{\nu}{B(\nu)}$, we prove the following truncation and global error analyses for the conventional PCM (14).

Theorem 2 (Truncation error for the predictor $\tilde{u}^P_{n+1}$)**.** *Let the truncation error of predictor $\tilde{r}^P_{n+1}$ be defined by*

$$\tilde{r}^P_{n+1} = |u_{n+1} - u^P_{n+1}|.$$

where $u^P_{n+1} = \phi_{n+1} + u^{men}_{n+1} + U^{loc}_{n+1}$. Under the same assumptions as those in Lemma 5, there exists a constant C independent of all grid parameters such that

$$\tilde{r}^P_{n+1} \leq Ch^2.$$

Proof. Let $||f^{(2)}||_\infty \leq M$. Then, from (10) and Lemma 5, we have the following inequality:

$$\tilde{r}^P_{n+1} \leq |\mathcal{G}_{n+1}| + |d_\nu| r^P_{n+1} \leq \frac{1-\nu}{|B(\nu)|} M h^2 + |d_\nu| r^P_{n+1} \leq Ch^2.$$

□

Theorem 3 (Global error for the predictor $\tilde{u}^P_{n+1}$)**.** *Under the same assumptions as those in Lemma 6, the global error for the predictor, $\tilde{e}^P_{n+1} = |u_{n+1} - \tilde{u}^P_{n+1}|$, can be estimated by*

$$\tilde{e}^P_{n+1} \leq \tilde{r}^P_{n+1} + \frac{L(1-\nu)}{|B(\nu)|}[2\tilde{e}_n + \tilde{e}_{n-1}] + C\frac{|d_\nu| L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \tilde{e}_j,$$

where $\tilde{e}_j = |u_j - \tilde{u}_j|$.

Proof. From (14) and the triangle inequality, we have

$$\begin{aligned} \tilde{e}^P_{n+1} \leq & |u_{n+1} - u^P_{n+1}| + |u^P_{n+1} - \tilde{u}^P_{n+1}| \\ \leq & \tilde{r}^P_{n+1} + \frac{1-\nu}{|B(\nu)|}[2|f_n - \tilde{f}_n| + |f_{n-1} - \tilde{f}_{n-1}|] \\ & + |d_\nu| \left(\frac{1}{\Gamma(\nu)} \sum_{j=0}^{n-1} [|\Theta^{0,j}_{n+1}||f_j - \tilde{f}_j| + |\Theta^{1,j}_{n+1}||f_{j+1} - \tilde{f}_{j+1}|] \right. \\ & \left. + \frac{h^\nu}{\Gamma(\nu+2)}[|f_{n-1} - \tilde{f}_{n-1}| + (\nu+2)|f_n - \tilde{f}_n|] \right) \end{aligned}$$

From the Lipschitz condition and Lemma 6 combined with the argument of R_n (see R_n in Theorem 2 [23]), we have

$$\tilde{e}^P_{n+1} \leq \tilde{r}^P_{n+1} + \frac{L(1-\nu)}{|B(\nu)|}[2\tilde{e}_n + \tilde{e}_{n-1}] + C\frac{|d_\nu| L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \tilde{e}_j.$$

□

Theorem 4 (Truncation error for the corrector $\tilde{u}_{n+1}$)**.** *Under the same assumptions as those in Theorem 3, let the truncation error at time $t = t_{n+1}$ be*

$$\tilde{r}^C_{n+1} = \left| u_{n+1} - (\phi(t_{n+1}, f^P_{n+1}) + u^{mem}_{n+1} + u^{loc}_{n+1}) \right|.$$

Then,

$$\tilde{r}^C_{n+1} \leq \frac{|d_\nu| M T^\nu}{2\Gamma(\nu+1)} h^2 + \left(\frac{L(1-\nu)}{|B(\nu)|} + \frac{|d_\nu| L h^\nu}{\Gamma(\nu+1)} \right) \tilde{e}^P_{n+1}.$$

Proof. From Lemma 7 and (14), we obtain the following:

$$\tilde{r}^C_{n+1} \leq \frac{L(1-\nu)}{|B(\nu)|} \tilde{e}^P_{n+1} + |d_\nu| \left(\frac{M T^\nu}{2\Gamma(\nu+1)} h^2 + \frac{L h^\nu}{\Gamma(\nu+1)} \tilde{e}^P_{n+1} \right).$$

□

Theorem 5 (Global error for the corrector $\tilde{u}_{n+1}$)**.** *Under the same assumptions as those in Theorem 3, the global error $\tilde{e}_{n+1} = |u_{n+1} - \tilde{u}_{n+1}|$ is*

$$\tilde{e}_{n+1} \leq C h^2 + C\epsilon,$$

given that the starting error $\tilde{e}_1 \leq C h^2$.

Proof. From Lemma 4 and Theorem 4,

$$\begin{aligned}
&\tilde{e}_{n+1} \\
\leq &\tilde{r}^C_{n+1} + \frac{L(1-\nu)}{|B(\nu)|} \tilde{e}^P_{n+1} + \frac{L|d_\nu|}{\Gamma(\nu)} (|\Theta^{0,n}_{n+1}| \tilde{e}_n + |\Theta^{1,n}_{n+1}| \tilde{e}^P_{n+1}) + \frac{L|d_\nu|}{\Gamma(\nu)} \sum_{j=0}^{n-1} (|\Theta^{0,j}_{n+1}| \tilde{e}_j + |\Theta^{1,j}_{n+1}| \tilde{e}_{j+1}) \\
\leq &\tilde{r}^C_{n+1} + \frac{L(1-\nu)}{|B(\nu)|} \tilde{e}^P_{n+1} + |d_\nu| \left(\frac{L h^\nu}{\Gamma(\nu+1)} (\tilde{e}_n + \tilde{e}^P_{n+1}) + \frac{L h^\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} (n-j)^{\nu-1} (\tilde{e}_j + \tilde{e}_{j+1}) \right) \\
\leq &\frac{|d_\nu| M T^\nu}{2\Gamma(\nu+1)} h^2 + 2\left(\frac{|d_\nu| L h^\nu}{\Gamma(\nu+1)} + \frac{L(1-\nu)}{|B(\nu)|} \right) \tilde{e}^P_{n+1} \\
&+ |d_\nu| \left(\frac{L h^\nu}{\Gamma(\nu+1)} \tilde{e}_n + \frac{L h^\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} (n-j)^{\nu-1} (\tilde{e}_j + \tilde{e}_{j+1}) \right).
\end{aligned}$$

By applying a similar argument to R_n in Theorem 3, we have

$$\frac{L h^\nu}{\Gamma(\nu+1)} \tilde{e}_n + \frac{L h^\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} (n-j)^{\nu-1} (\tilde{e}_j + \tilde{e}_{j+1}) \leq C \frac{L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \tilde{e}_j. \tag{22}$$

Using Theorem 3 and Equation (22) implies that

$$\begin{aligned}
\tilde{e}_{n+1} \leq &\frac{|d_\nu| M T^\nu}{2\Gamma(\nu+1)} h^2 + \Lambda^\nu_h \left(\tilde{r}^P_{n+1} + \frac{L(1-\nu)}{|B(\nu)|} [2\tilde{e}_n + \tilde{e}_{n-1}] \right) \\
&+ C(\Lambda^\nu_h + 1) \frac{|d_\nu| L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \tilde{e}_j,
\end{aligned}$$

where

$$\Lambda^\nu_h = 2\left(\frac{|d_\nu| L h^\nu}{\Gamma(\nu+1)} + \frac{L(1-\nu)}{|B(\nu)|} \right).$$

Then, the discrete Gronwall's inequality yields

$$\tilde{e}_{n+1} \leq (\mathcal{Q} + \Lambda_h^\nu \tilde{r}_{n+1}^P) E_\nu \Big(C|d_\nu| L T^\nu (\Lambda_h^\nu + 1) \Big),$$

where

$$\mathcal{Q} = \frac{|d_\nu| M T^\nu}{2\Gamma(\nu+1)} h^2 + \Lambda_h^\nu \Big(\frac{L(1-\nu)}{|B(\nu)|} [2\tilde{e}_n + \tilde{e}_{n-1}] \Big)$$

Since $\tilde{e}_1 \leq Ch^2$ and $\tilde{r}_{n+1}^P \leq Ch^2$ in Theorem 2, it is clear that

$$\mathcal{Q} + \Lambda_h^\nu \tilde{r}_{n+1}^P \leq Ch^2, \quad n \geq 1.$$

This completes the proof. □

3.2. Global Error Analysis for the Fast PCM

In this section, we discuss the error analysis for the fast PCM in (21). From (19), we have

$$\Big| u_{n+1}^{mem} - u_{n+1}^{fmem} \Big| = |T_{n+1}^\epsilon| = \Big| \epsilon \frac{d_\nu}{\Gamma(\nu)} \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} \Pi_j f(s, u(s)) ds \Big| \leq \epsilon \frac{|d_\nu|}{\Gamma(\nu)} (\max_{0 \leq j \leq n} |f_j|) nh.$$

Thus, we have the following:

Lemma 9. *For $\epsilon > 0$, suppose that Equation (18) is satisfied. Then,*

$$\Big| u_{n+1}^{mem} - u_{n+1}^{fmem} \Big| \leq D\epsilon, \quad \text{where } D = T \frac{|d_\nu|}{\Gamma(\nu)} \max_{0 \leq j \leq n} |f_j|.$$

Theorem 6 (Global error for the predictor $\hat{u}_{n+1}^P$)**.** *Under the same assumptions as those in Theorem 3, the global error $\hat{e}_{n+1}^P = |u_{n+1} - \hat{u}_{n+1}^P|$ has the following inequality:*

$$\hat{e}_{n+1}^P \leq Ch^2 + \frac{L(1-\nu)}{|B(\nu)|} [2\hat{e}_n + \hat{e}_{n-1}] + C \frac{|d_\nu| L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \hat{e}_j + \hat{D}\epsilon,$$

where $\hat{D} = T \frac{|d_\nu|}{\Gamma(\nu)} \max_{0 \leq j \leq n} |\hat{f}_j|$.

Proof. From Theorem 3 and Lemma 9 with $u_{n+1}^{mem} - u_{n+1}^{fmem} = T_{n+1}^\epsilon$, we obtain the following:

$$\begin{aligned} \hat{e}_{n+1}^P =& \Big| u_{n+1} - (\hat{\phi}_{n+1} + \hat{u}_{n+1}^{fmem} + \widehat{U}_{n+1}^{loc}) \Big| \leq \Big| u_{n+1} - (\hat{\phi}_{n+1} + \hat{u}_{n+1}^{mem} + \widehat{U}_{n+1}^{loc}) \Big| + |\hat{T}_{n+1}^\epsilon| \\ \leq& Ch^2 + \frac{L(1-\nu)}{|B(\nu)|} [2\hat{e}_n + \hat{e}_{n-1}] + C \frac{|d_\nu| L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \hat{e}_j + \hat{D}\epsilon. \end{aligned}$$

□

Theorem 7 (Global error for the corrector $\hat{u}_{n+1}$)**.** *Under the same assumptions as those in Theorem 3, the global error $\hat{e}_{n+1} = |u_{n+1} - \hat{u}_{n+1}|$ is*

$$\hat{e}_{n+1} \leq Ch^2 + C\epsilon,$$

given that the starting error $\hat{e}_1 \leq Ch^2$.

Proof. From Theorems 5 and 6 and Lemma 9 , we have

$$\begin{aligned}
\hat{e}_{n+1} \leq & \left| u_{n+1} - (\phi(t_{n+1}, \hat{f}^P_{n+1}) + \hat{u}^{mem}_{n+1} + \hat{u}^{loc}_{n+1}) \right| + |\hat{T}^{\epsilon}_{n+1}| \\
\leq & \frac{|d_\nu| M T^\nu}{2\Gamma(\nu+1)} h^2 + \Lambda_h^\nu \hat{e}^P_{n+1} + |d_\nu| \left(C \frac{Lh^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \hat{e}_j \right) + \hat{D}\epsilon. \\
\leq & \frac{|d_\nu| M T^\nu}{2\Gamma(\nu+1)} h^2 + \Lambda_h^\nu \left(Ch^2 + \frac{L(1-\nu)}{|B(\nu)|} [2\hat{e}_n + \hat{e}_{n-1}] \right) \\
& + C(\Lambda_h^\nu + 1) \frac{|d_\nu| L h^\nu}{\Gamma(\nu)} \sum_{j=1}^{n} (n+1-j)^{\nu-1} \hat{e}_j + \hat{D}(\Lambda_h^\nu + 1)\epsilon.
\end{aligned}$$

Then, from the discrete Gronwall's inequality, we have the following:
$$\hat{e}_{n+1} \leq \left(\mathcal{Q} + C\Lambda_h^\nu h^2 + \hat{D}(\Lambda_h^\nu + 1)\epsilon \right) E_\nu \left(C|d_\nu| L T^\nu (\Lambda_h^\nu + 1) \right).$$
□

4. Numerical Results

In this section, we conduct numerical experiments to demonstrate the effectiveness of our proposed methods and numerically verify the error estimates. Let us define the following two errors over $[0, T]$:

- Maximum norm error:
$$E_{\max} = \max_{0 \leq j \leq N} |u_j - u_j^h|.$$
- Discrete L^2 norm error:
$$E_{\ell^2} = \left(h \sum_{j=0}^{N} |u_j - u_j^h|^2 \right)^{\frac{1}{2}},$$

where u_j^h is an approximate solution at time t_j.

Throughout this section, ABC-PCM and ABC-FPCM are used to denote the conventional PCM (14) and the fast PCM (21), respectively. Several numerical demonstrations are performed to obtain approximate solutions of nonlinear ABC fractional-order initial value problems and nonlinear reaction-diffusion problems. Moreover, the efficiency of the fast algorithm is clearly demonstrated in its application to fractional dynamical systems. All numerical experiments are implemented using Matlab. For simplicity, in all our numerical experiments, we set the constant $B(\nu)$ to a value of 1. In Section 4.1, we present the numerical results for the nonlinear ABC fractional differential equations in (2). These results validate our theoretical findings, including the convergence rate, and confirm the efficiency of our proposed fast algorithms compared to the conventional PCM. We discuss the extension of our proposed method to fractional partial differential equations. All pertinent numerical results supporting the efficacy and applicability of our method in this context are included in Section 4.2. Finally, the numerical efficiency of our fast algorithm is showcased through its application to various fractional dynamical systems in Section 4.3.

4.1. Nonlinear ABC Fractional-Order Initial Value Problems

Example 1. *Consider the following nonlinear problem with $u(0) = 0$ [20]:*

$$\begin{aligned}
{}^{ABC}\mathcal{D}_0^\nu u(t) + u(t) - u^2(t) = t^3 - \Bigg[& \frac{6t^3}{B(\nu) + 1 - \nu} \left((1-\nu) E_{\nu,4}\left(-\frac{\nu t^\nu}{B(\nu) + 1 - \nu} \right) \right. \\
& \left. + \nu t^\nu E_{\nu,\nu+4}\left(-\frac{\nu t^\nu}{B(\nu) + 1 - \nu} \right) \right) \Bigg]^2,
\end{aligned}$$

where the solution is $u(t) = \frac{6t^3}{B(\nu)+1-\nu} \left[(1-\nu) E_{\nu,4}\left(-\frac{\nu t^\nu}{B(\nu)+1-\nu} \right) + \nu t^\nu E_{\nu,\nu+4}\left(-\frac{\nu t^\nu}{B(\nu)+1-\nu} \right) \right].$

Example 2. *Consider the following nonlinear problem with $u(0) = 1$ [20]:*

$$
{}^{ABC}\mathcal{D}_0^{\nu} u(t) + \cos(u(t)) = \frac{B(\nu)}{1-\nu}\bigg(2\Gamma(5)t^4 E_{\nu,5}\Big(-\frac{\nu t^\nu}{1-\nu}\Big) - 3\nu\Gamma(2\nu+3)t^{2\nu+2}E_{\nu,2\nu+3}\Big(-\frac{\nu t^\nu}{1-\nu}\Big)
+ \Gamma(\nu+3)t^{\nu+2}E_{\nu,\nu+3}\Big(-\frac{\nu t^\nu}{1-\nu}\Big)\bigg) + \cos(2t^4 - 3\nu t^{2(\nu+1)} + t^{\nu+2} + 1),
$$

where the solution is $u(t) = 2t^4 - 3\nu t^{2(\nu+1)} + t^{\nu+2} + 1$.

In Examples 1 and 2, we fix $B(\nu) = 1$, $T = 1$, and $\epsilon = 10^{-9}$. From Tables 3 and 4 and Figure 1, we can make the following observations:

1. The maximum norm errors, discrete L^2 norm errors, and computed convergence rates versus time steps $h = 1/10, 1/20, 1/40, \cdots, 1/640$ are shown in Tables 3 and 4 for $\nu = 0.2, 0.5, 0.8$.
2. The numerical results obtained with the ABC-FPCM show little difference from those obtained with the ABC-PCM.
3. The computed convergence profiles for the maximum norm errors shown in Tables 3 and 4 are approximately 2. Thus, the numerical results support Theorems 5 and 7.
4. The computational costs, obtained by measuring the CPU time (in seconds) executed by the conventional PCM and the fast PCM versus the total number of steps N on the log-log scale in Example 2, are depicted in Figure 1. The figure shows that the CPU consumption rate of the fast PCM is $\mathcal{O}(N)$, whereas that of the conventional PCM is $\mathcal{O}(N^2)$.

Table 3. Errors and rates of convergence versus h for Example 1 with $\epsilon = 10^{-9}$, $\nu = 0.2, 0.5, 0.8$.

	ABC-FPCM				ABC-PCM			
	$E_{\max}$	roc	E_{ℓ_2}	roc	$E_{\max}$	roc	E_{ℓ_2}	roc
h				$\nu = 0.2$				
1/10	3.37×10^{-2}	-	1.85×10^{-2}	-	3.37×10^{-2}	-	1.85×10^{-2}	-
1/20	9.03×10^{-3}	1.90	4.92×10^{-3}	1.91	9.03×10^{-3}	1.90	4.92×10^{-3}	1.91
1/40	1.71×10^{-3}	2.40	1.14×10^{-3}	2.11	1.71×10^{-3}	2.40	1.14×10^{-3}	2.11
1/80	3.98×10^{-4}	2.10	2.67×10^{-4}	2.09	3.98×10^{-4}	2.10	2.67×10^{-4}	2.09
1/160	9.35×10^{-5}	2.09	6.33×10^{-5}	2.08	9.35×10^{-5}	2.09	6.33×10^{-5}	2.08
1/320	2.21×10^{-5}	2.08	1.51×10^{-5}	2.07	2.21×10^{-5}	2.08	1.51×10^{-5}	2.07
1/640	5.28×10^{-6}	2.07	3.61×10^{-6}	2.06	5.28×10^{-6}	2.07	3.61×10^{-6}	2.06
h				$\nu = 0.5$				
1/10	7.35×10^{-3}	-	4.93×10^{-3}	-	7.35×10^{-3}	-	4.93×10^{-3}	-
1/20	1.55×10^{-3}	2.24	1.11×10^{-3}	2.15	1.55×10^{-3}	2.24	1.11×10^{-3}	2.15
1/40	3.53×10^{-4}	2.14	2.56×10^{-4}	2.11	3.53×10^{-4}	2.14	2.56×10^{-4}	2.11
1/80	8.24×10^{-5}	2.10	6.02×10^{-5}	2.09	8.24×10^{-5}	2.10	6.02×10^{-5}	2.09
1/160	1.95×10^{-5}	2.08	1.43×10^{-5}	2.07	1.95×10^{-5}	2.08	1.43×10^{-5}	2.07
1/320	4.70×10^{-6}	2.06	3.46×10^{-6}	2.05	4.70×10^{-6}	2.06	3.46×10^{-6}	2.05
1/640	1.14×10^{-6}	2.04	8.41×10^{-7}	2.04	1.14×10^{-6}	2.04	8.41×10^{-7}	2.04
h				$\nu = 0.8$				
1/10	1.82×10^{-3}	-	1.38×10^{-3}	-	1.82×10^{-3}	-	1.38×10^{-3}	-
1/20	3.95×10^{-4}	2.21	2.99×10^{-4}	2.21	3.95×10^{-4}	2.21	2.99×10^{-4}	2.21
1/40	9.17×10^{-5}	2.11	6.84×10^{-5}	2.13	9.17×10^{-5}	2.11	6.84×10^{-5}	2.13
1/80	2.21×10^{-5}	2.05	1.62×10^{-5}	2.08	2.21×10^{-5}	2.05	1.62×10^{-5}	2.08
1/160	5.41×10^{-6}	2.03	3.93×10^{-6}	2.04	5.42×10^{-6}	2.03	3.94×10^{-6}	2.04
1/320	1.35×10^{-6}	2.01	9.68×10^{-7}	2.02	1.34×10^{-6}	2.01	9.67×10^{-7}	2.03
1/640	3.39×10^{-7}	1.99	2.41×10^{-7}	2.01	3.33×10^{-7}	2.01	2.39×10^{-7}	2.01

Table 4. Errors and rates of convergence versus h for Example 2 with $\epsilon = 10^{-9}$, $\nu = 0.2, 0.5, 0.8$.

	ABC-FPCM				**ABC-PCM**			
	$E_{\max}$	**roc**	E_{ℓ_2}	**roc**	$E_{\max}$	**roc**	E_{ℓ_2}	**roc**
h				$\nu = 0.2$				
1/10	5.14×10^{-1}	-	2.47×10^{-1}	-	5.14×10^{-1}	-	2.47×10^{-1}	
1/20	2.00×10^{-1}	1.36	8.53×10^{-2}	1.54	2.00×10^{-1}	1.36	8.53×10^{-2}	1.54
1/40	5.59×10^{-2}	1.84	2.29×10^{-2}	1.90	5.59×10^{-2}	1.84	2.29×10^{-2}	1.90
1/80	1.34×10^{-2}	2.06	5.56×10^{-3}	2.04	1.34×10^{-2}	2.06	5.56×10^{-3}	2.04
1/160	3.20×10^{-3}	2.07	1.36×10^{-3}	2.03	3.20×10^{-3}	2.07	1.36×10^{-3}	2.03
1/320	7.81×10^{-4}	2.03	3.36×10^{-4}	2.02	7.81×10^{-4}	2.03	3.36×10^{-4}	2.02
1/640	1.93×10^{-4}	2.02	8.31×10^{-5}	2.02	1.93×10^{-4}	2.02	8.31×10^{-5}	2.02
h				$\nu = 0.5$				
1/10	1.09×10^{-1}	-	5.72×10^{-2}	-	1.09×10^{-1}	-	5.72×10^{-2}	-
1/20	2.88×10^{-2}	1.93	1.45×10^{-2}	1.98	2.88×10^{-2}	1.93	1.45×10^{-2}	1.98
1/40	6.89×10^{-3}	2.06	3.51×10^{-3}	2.04	6.89×10^{-3}	2.06	3.51×10^{-3}	2.04
1/80	1.66×10^{-3}	2.05	8.49×10^{-4}	2.05	1.66×10^{-3}	2.05	8.49×10^{-4}	2.05
1/160	4.03×10^{-4}	2.04	2.06×10^{-4}	2.04	4.03×10^{-4}	2.04	2.06×10^{-4}	2.04
1/320	9.84×10^{-5}	2.04	5.03×10^{-5}	2.04	9.84×10^{-5}	2.04	5.03×10^{-5}	2.04
1/640	2.41×10^{-5}	2.03	1.23×10^{-5}	2.03	2.41×10^{-5}	2.03	1.23×10^{-5}	2.03
h				$\nu = 0.8$				
1/10	4.37×10^{-3}	-	2.03×10^{-3}	-	4.37×10^{-3}	-	2.03×10^{-3}	-
1/20	8.14×10^{-4}	2.42	3.70×10^{-4}	2.46	8.14×10^{-4}	2.42	3.70×10^{-4}	2.46
1/40	1.51×10^{-4}	2.43	6.97×10^{-5}	2.41	1.51×10^{-4}	2.43	6.97×10^{-5}	2.41
1/80	2.94×10^{-5}	2.36	1.39×10^{-5}	2.32	2.94×10^{-5}	2.37	1.39×10^{-5}	2.32
1/160	6.05×10^{-6}	2.28	2.97×10^{-6}	2.23	6.04×10^{-6}	2.28	2.96×10^{-6}	2.23
1/320	1.29×10^{-6}	2.23	6.54×10^{-7}	2.18	1.31×10^{-6}	2.20	6.63×10^{-7}	2.16
1/640	2.74×10^{-7}	2.23	1.46×10^{-7}	2.17	2.99×10^{-7}	2.13	1.54×10^{-7}	2.10

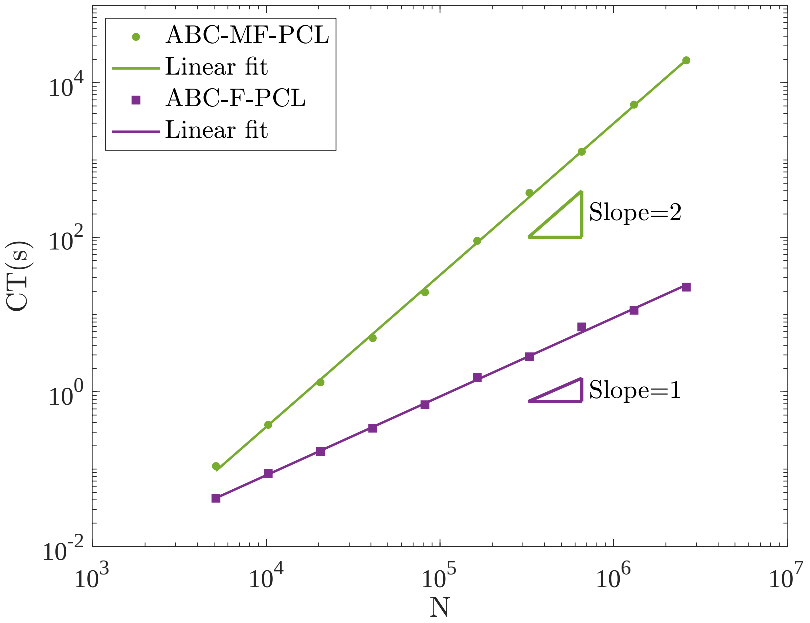


Figure 1. Log-log plot of CPU time (CT(s)) versus the number of steps N with $\nu = 0.5$ for Example 2.

4.2. Application to Fractional Order PDEs

In this subsection, we discuss the ABC fractional-order reaction-diffusion equation. The general form of this equation is given by

$$\begin{gathered} {}^{ABC}\mathcal{D}_0^{\nu} u(x,t) = c\frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t,u), \quad x \in [a_s, b_s],\ t \geq 0, \\ u(x,0) = u_0(x), \quad u(a_s,t) = \varphi(t), \quad u(b_s,t) = \psi(t). \end{gathered} \tag{23}$$

where $0 < \nu < 1$ is of fractional order, $c \in \mathbb{C}$ is the diffusion coefficient, and g is the nonlinear source term. The approximated solution U_n at time t_n is denoted as follows:

$$U_n = (u_1^n, u_1^n, ..., u_{M-1}^n)^T, \quad X = \{x_0 = a_s, ..., x_m = a + m\tau, ..., x_M = b_s\} \tag{24}$$

with the space step size $\tau = x_m - x_{m-1}$ for $m = 1, ..., M$, and the time step size $h = t_n - t_{n-1}$ for $n = 1, ..., N$, and $F_n = f(X, t_n, U_n)$. We employ the second-order central difference quotient to approximate $u_{xx}(x,t)$ at (x_m, t_n):

$$\frac{u(x_{m-1}, t_n) - 2u(x_m, t_n) + u(x_{m+1}, t_n)}{\tau^2} = u_{xx}(x,t)|_{(x,t)=(x_m,t_n)} + \frac{u_{xx}(x_m, t_n)}{12}\tau^2 + \mathcal{O}(\tau^4).$$

By applying the conventional PCM (14) combined with the second-order central difference to the sub-diffusion Equation (23), we obtain the following numerical scheme:

$$\begin{aligned} \left[\mathbf{I} + \frac{1-\nu}{B(\nu)}c\mathbf{L} + \frac{d_\nu}{\Gamma(\nu)}\Theta_{n+1}^{1,n}c\mathbf{L}\right]U_{n+1} &= \phi_{n+1}^P + U_{n+1}^{mem} + U_{n+1}^{loc,P}, \\ \left[\mathbf{I} + \frac{1-\nu}{B(\nu)}c\mathbf{L} + \frac{d_\nu}{\Gamma(\nu)}\Theta_{n+1}^{1,n}c\mathbf{L}\right]U_{n+1}^P &= \phi_{n+1} + U_{n+1}^{mem} + U_{n+1}^{loc} \end{aligned} \tag{25}$$

where $\mathbf{I}$ is an identity matrix, $\mathbf{L}$ is the finite-difference matrix for the space derivative, and

$$\begin{aligned} &\phi_{n+1}^P = U_0 + \frac{1-\nu}{B(\nu)}F_{n+1}^P, \; U_{n+1}^{loc,P} = \frac{d_\nu}{\Gamma(\nu)}\left[\Theta_{n+1}^{0,n}(c\mathbf{L}U_n + F_n) + \Theta_{n+1}^{1,n}F_{n+1}^P\right], \\ &U_{n+1}^{mem} = \frac{d_\nu}{\Gamma(\nu)}\sum_{j=0}^{n-1}\left[\Theta_{n+1}^{0,j}(c\mathbf{L}U_j + F_j) + \Theta_{n+1}^{1,j}(c\mathbf{L}U_{j+1} + F_{j+1})\right], \\ &\phi_{n+1} = U_0 + \frac{1+\nu}{B(\nu)}(-F_{n-1} + 2F_n), \; U_{n+1}^{loc} = \frac{d_\nu}{\Gamma(\nu)}\Theta_{n+1}^{0,n}c\mathbf{L}U_n + \frac{d^\nu h^\nu}{\Gamma(\nu)}[-F_{n-1} + (\nu+2)F_n]. \end{aligned}$$

Similarly, the fast PCM for the sub-diffusion equation can be expressed as follows:

$$\begin{aligned} \left[\mathbf{I} + \frac{1-\nu}{B(\nu)}c\mathbf{L} + \frac{d_\nu}{\Gamma(\nu)}\Theta_{n+1}^{1,n}c\mathbf{L}\right]U_{n+1} &= \phi_{n+1}^P + U_{n+1}^{Fmem} + U_{n+1}^{loc,P}, \\ \left[\mathbf{I} + \frac{1-\nu}{B(\nu)}c\mathbf{L} + \frac{d_\nu}{\Gamma(\nu)}\Theta_{n+1}^{1,n}c\mathbf{L}\right]U_{n+1}^P &= \phi_{n+1} + U_{n+1}^{Fmem} + U_{n+1}^{loc} \end{aligned} \tag{26}$$

where

$$\begin{aligned} U_{n+1}^{Fmem} &= \frac{d_\nu}{\Gamma(\nu)}\sum_{i=1}^{N_{\exp}}\omega_i F_{n+1}^i, \\ F_{n+1}^i &= \sum_{j=0}^{n-1}\left[\Phi_{i,n+1}^{0,j}(c\mathbf{L}U_j + F_j) + \Phi_{i,n+1}^{1,j}(c\mathbf{L}U_{j+1} + F_{j+1})\right], \\ &= e^{\xi_i h}F_n^i + \left[\Phi_{i,n+1}^{0,n-1}(c\mathbf{L}U_{n-1} + F_{n-1}) + \Phi_{i,n+1}^{1,n-1}(c\mathbf{L}U_n + F_n)\right] \end{aligned}$$

Now, we apply the PCMs combined with the second-order central difference schemes (25) and (26) for solving ABC fractional order sub-diffusion PDEs to the following examples. Although our proposed method is capable of handling an arbitrary diffusion coefficient c, for simplicity in our simulation, we have chosen to set the diffusion constant to 1.

Example 3. *Consider the linear sub-diffusion problem with $f(x,t)$, which is given by*

$$\begin{aligned} f(x,t) = &\frac{B(\nu)}{1-\nu}\Gamma(\nu+4)t^{\nu+3}E_{\nu,\nu+4}\left(-\frac{\nu t^{\nu}}{1-\nu}\right)e^{-x}x^4(x-\pi)^4 \\ &-(t^3e^{-x}+1)x^2(x-\pi)^2[12(x-\pi)^4+32x(x-\pi)+12x^2] \\ &-t^3e^{-x}x^3(x-\pi)^3[x(x-\pi)-8(x-\pi)-8x], \end{aligned}$$

and the corresponding initial boundary conditions are as follows:

$$u(x,0)=x^4(\pi-x)^4, \quad u(0,t)=0, \quad u(3,t)=3^4(\pi-3)^4[\exp(-3)t^{3+\nu}+1].$$

Then, $u(x,t)=x^4(\pi-x)^4[\exp(-x)t^{3+\nu}+1]$ is the exact solution.

Example 4. *Consider the nonlinear sub-diffusion problem with $f(x,t,u)$, which is given by*

$$\begin{aligned} f(x,t,u) = &\frac{B(\nu)}{1-\nu}\Gamma(5)t^4E_{\nu,5}\left(-\frac{\nu t^{\nu}}{1-\nu}\right)\cos(3\pi x)-9\pi^2t^4\cos(3\pi x) \\ &-u^2+t^8\cos^2(3\pi x), \end{aligned}$$

and the corresponding initial boundary conditions are as follows:

$$u(x,0)=0, \quad u(0,t)=t^4, \quad u(1,t)=-t^4.$$

Then, $u(x,t)=t^4\cos(3\pi x)$ is the exact solution.

From Tables 5 and 6, we can make the following observations:

1. Tables 5 and 6 show the maximum norm errors, discrete L^2 norm errors, and rates of convergence versus h and τ with $\tau = 1/5000$ and $h = 1/5000$ fixed, respectively, for $\nu = 0.2, 0.8$.
2. Theoretically, the rate of convergence for the second-order central difference quotient is $\mathcal{O}(\tau^2)$, and the convergence rates for the proposed methods are shown to be 2. Thus, the global order of convergence for both (25) and (26) is expected to be 2 when either τ or h is fixed. In the tables, one can see that the rates of convergence computed by the ABC-FPCM and ABC-PCM are approximately 2 in both cases where τ is fixed and h is fixed. This verifies that the global estimates of our proposed methods are valid in solving sub-diffusion FPDEs.
3. The gap between the CPU time executed by the ABC-PCM and that executed by the ABC-FPCM is evident in the tables. Particularly, the difference between them drastically increases as $h = 1/5000$ is fixed. This verifies that the proposed fast PCM is more efficient compared to the conventional PCM.

Table 5. Errors and rates of convergence versus h when $\tau = 1/5000$ and versus τ when $h = 1/5000$ for Example 3.

		$\nu = 0.2$						**$\nu = 0.8$**					
		ABC-FPCM			**ABC-PCM**			**ABC-FPCM**			**ABC-PCM**		
	h	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)
$\tau = 1/5000$	1/10	1.81×10^{-3}	-	1.017	1.81×10^{-3}	-	1.101	1.52×10^{-2}	-	1.005	1.52×10^{-2}	-	1.149
	1/20	4.96×10^{-4}	1.87	1.975	4.96×10^{-4}	1.87	2.639	3.82×10^{-3}	1.99	2.115	3.82×10^{-3}	1.99	2.508
	1/40	1.33×10^{-4}	1.89	3.866	1.33×10^{-4}	1.89	4.971	9.57×10^{-4}	2.00	3.935	9.57×10^{-4}	2.00	4.648
	1/80	3.56×10^{-5}	1.91	7.997	3.56×10^{-5}	1.91	10.120	2.40×10^{-5}	2.00	8.206	2.40×10^{-5}	2.00	10.139
	1/160	9.59×10^{-6}	1.89	15.985	9.59×10^{-6}	1.89	25.011	6.02×10^{-5}	1.99	15.744	6.02×10^{-5}	1.99	20.843
	τ	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)
$h = 1/5000$	1/10	9.67×10^{-2}	-	3.761	9.67×10^{-2}	-	62.772	1.00×10^{-1}	-	3.875	1.00×10^{-1}	-	61.249
	1/20	2.42×10^{-2}	2.00	7.065	2.42×10^{-2}	2.00	70.675	2.51×10^{-2}	2.00	7.113	2.51×10^{-2}	2.00	69.826
	1/40	6.05×10^{-3}	2.00	13.893	6.05×10^{-3}	2.00	90.210	6.27×10^{-3}	2.00	13.590	6.27×10^{-3}	2.00	86.519
	1/80	1.51×10^{-3}	2.00	26.930	1.51×10^{-3}	2.00	121.949	1.57×10^{-3}	2.00	26.398	1.57×10^{-3}	2.00	119.277
	1/160	3.78×10^{-4}	2.00	29.859	3.78×10^{-4}	2.00	153.570	3.92×10^{-4}	2.00	21.083	3.92×10^{-4}	2.00	155.221

Table 6. Errors and rates of convergence versus h when $\tau = 1/5000$ and versus τ when $h = 1/5000$ for Example 4.

		$\nu = 0.2$						$\nu = 0.8$					
		ABC-FPCM			**ABC-PCM**			**ABC-FPCM**			**ABC-PCM**		
	h	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)
	1/10	1.19×10^{-2}	-	0.101	1.19×10^{-2}	-	0.079	7.64×10^{-3}	-	0.101	7.64×10^{-3}	-	0.069
	1/20	3.44×10^{-3}	1.79	0.165	3.44×10^{-3}	1.79	0.123	1.96×10^{-3}	1.97	0.162	1.96×10^{-3}	1.97	0.131
$\tau = 1/5000$	1/40	9.07×10^{-4}	1.92	0.368	9.07×10^{-4}	1.92	0.324	4.79×10^{-4}	2.03	0.290	4.79×10^{-4}	2.03	0.312
	1/80	2.31×10^{-4}	1.98	0.628	2.31×10^{-4}	1.98	0.690	1.17×10^{-4}	2.03	0.608	1.17×10^{-4}	2.03	0.720
	1/160	5.79×10^{-5}	1.99	1.207	5.79×10^{-5}	1.99	1.807	2.89×10^{-5}	2.02	1.183	2.89×10^{-5}	2.02	1.797
	τ	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)	$E_{\max}$	roc	CT(s)
	1/10	1.12×10^{-1}	-	0.834	1.12×10^{-1}	-	48.032	1.08×10^{-1}	-	0.720	1.08×10^{-1}	-	53.056
	1/20	2.70×10^{-2}	2.06	0.642	2.70×10^{-2}	2.06	53.879	2.60×10^{-2}	2.05	0.779	2.60×10^{-2}	2.05	52.112
$h = 1/5000$	1/40	6.71×10^{-3}	2.01	0.959	6.71×10^{-3}	2.01	60.800	6.47×10^{-3}	2.01	0.971	6.47×10^{-3}	2.01	60.103
	1/80	1.68×10^{-3}	2.00	1.391	1.68×10^{-3}	2.00	69.584	1.62×10^{-3}	2.00	1.479	1.62×10^{-3}	2.00	64.850
	1/160	4.20×10^{-4}	2.00	2.493	4.20×10^{-4}	2.00	81.508	4.04×10^{-4}	2.00	2.515	4.04×10^{-4}	2.00	75.244

We apply the proposed methods to time-dependent PDEs in two dimensions. The authors of [37] discussed the finite-difference scheme for the predator–prey interaction in the two-dimensional space for the integer-order case. We aim to solve this model equipped with time-fractional derivatives, specifically the predator–prey reaction-diffusion systems with the Holling type II functional response and logistic growth of the prey. These systems are described below.

Example 5. *Consider the two-component reaction-diffusion equation system [37]:*

$$\begin{cases} \mathcal{D}_0^{\nu} u = \Delta u + u(1-|u|) - \dfrac{uv}{|u|+a}, \\ \mathcal{D}_0^{\nu} v = \delta \Delta v + \dfrac{buv}{|u|+a} - cv, \end{cases}$$

where $0 < \nu \leq 1$, and the parameters $a, b,$ and c are strictly positive. $\mathcal{D}_0^{\nu}$ denotes the Liouville–Caputo FD (1) or ABC FD. Its initial conditions are given by

$$u_0(x,y) = u^* - 2 \times 10^{-7}(x - 0.1y - 225)(x - 0.1y - 675),$$
$$v_0(x,y) = v^* - 3 \times 10^{-5}(x - 450) - 1.2 \times 10^{-4}(y - 150),$$

where the stationary solution (u^*, v^*) is $(ab/(b-c), (1-u^*)(u^*+a))$, and zero-flux boundary conditions are applied.

The suggested methods (25) and (26) can be extended to solve this example through the following setting:

$$U_n = (u_0^n, u_1^n, ..., u_m^n, ..., u_M^n)^T, \quad u_m^n = (u_{0m}^n, u_{1m}^n, ..., u_{Mm}^n)^T$$

and the matrix $\mathbf{L}$ and the parameters $a = 0.4$, $b = 2.0$, $c = 2.0$, and $\delta = 1$ are defined in [37].

We measure the CPU times of the suggested methods and investigate the effects of the types of fractional derivatives and fractional orders. We make the following observations:

1. Table 7 shows the CPU times for Scheme 2 in [37], the ABC-PCM, and the ABC-FPCM for Example 5. To measure the rate of the CPU time, the time T increases twice from T=125 to 1000, and we can observe that the rate of the CPU time for Scheme 2 and the ABC-FPCM is $\mathcal{O}(N)$, whereas for the ABC-PCM, it is $\mathcal{O}(N^2)$.
2. Furthermore, the ABC-FPCM is much more efficient in terms of memory management compared to the existing method because it requires storing all previous values (U_n, $\mathbf{L}U_n$, and G_n) to calculate U_{n+1} by using the ABC-PCM efficiently. On the other hand, the APC-FPCM requires only local values.
3. Figure 2 shows the effect of fractional order and types of fractional derivatives for $T = 250, 500, 1000$, $h = 1/3$, $L = 400$, and $\tau = 1$. The first row depicts the evolution

of the system for $\nu = 1$. We can see that the spiral pattern is broken when $T = 250$, and an irregular pattern appears when $T = 500, 1000$.

4. The second and third rows describe the evolution of the system for the ABC and Liouville–Caputo fractional derivatives of order $\nu = 0.98$, respectively. In the fractional-order system, similar to the case of $\nu = 1$, the spiral pattern is broken over time.
5. In addition, in the fractional systems, a spiral pattern exists at $T = 250, 1000$, but the spiral pattern disappears at $T = 1000$. This phenomenon is similar to the case where $\nu = 1$, $T = 250$, and has a wider pattern.
6. It can be seen that the solution of the problem equipped with the ABC derivative is slower and wider than the solution of the problem equipped with the Liouville–Caputo derivative.

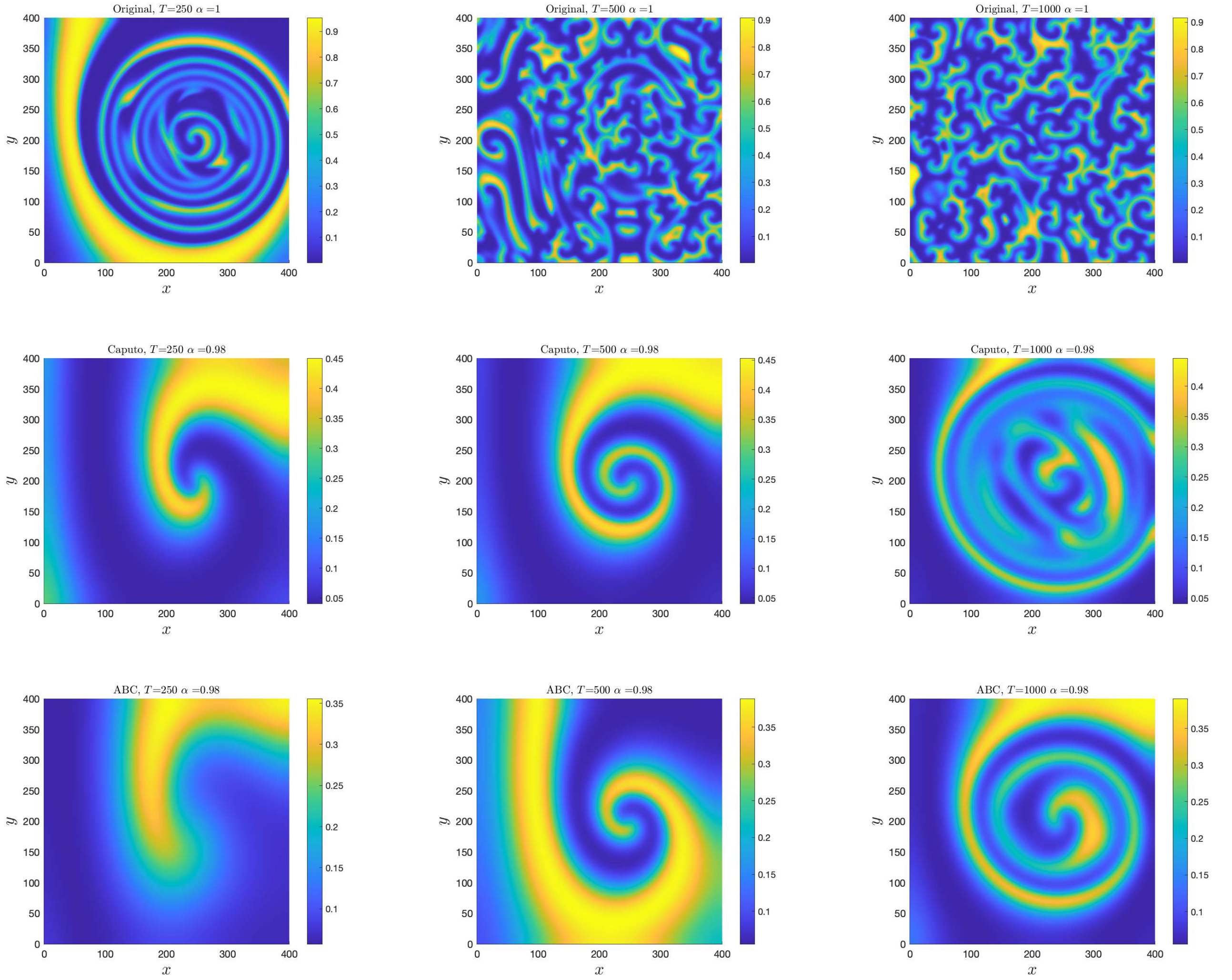


Figure 2. Comparison of numerical solution of Example 5 at $T = 250, 500, 1000$ with the parameters $a = 0.4$, $b = 2.0$, $c = 0.6$, $\delta = 1$, $L = 400$, $\tau = 1$, and $h = 1/3$. Each row represents $\nu = 1$, $\nu = 0.98$ with the Liouville–Caputo FD, and $\nu = .98$ with the ABC FD.

Table 7. CPU times of the fractional-order predator–prey interaction (5) with the parameters $a = 0.4$, $b = 2.0$, $c = 0.6$, $\delta = 1$, $L = 400$, $\tau = 1$, and $h = 1/3$.

T	**125**	**250**	**500**	**1000**
Scheme 2 [37]	17.03	35.22	74.13	155.21
ABC-PCM	52.41	149.55	495.41	1668.56
ABC-FPCM	55.23	109.69	223.01	547.09

4.3. Application to Fractional Dynamical Systems

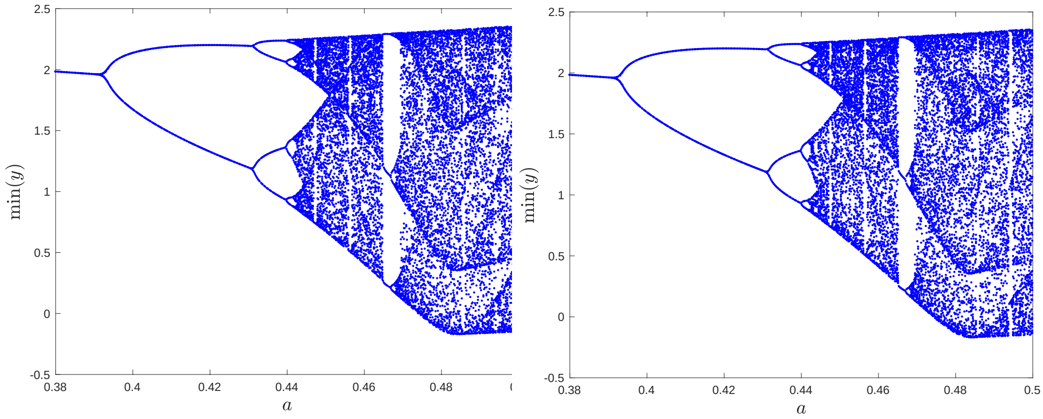
Here, we discuss the application to fractional dynamical systems. Recently, many researchers have studied fractional-order dynamical systems due to the memory effect. However, fractional-order dynamical systems consume excessive computational time compared to integer-order systems when drawing bifurcation diagrams with fixed points and extrema. We apply our fast PCM to plot bifurcation diagrams for the fractional-order Rössler dynamical system with the Liouville–Caputo FD and ABC FD. We aim to discuss the efficiency of the fast PCM and the influence of the definition of FD applied. First, we recall the fractional-order Rössler system:

Example 6. *The fractional-order Rössler system [38] is defined as*

$$\begin{cases} \mathcal{D}_0^{\nu} x &= -y - z, \\ \mathcal{D}_0^{\nu} y &= x + ay, \\ \mathcal{D}_0^{\nu} z &= b + z(x - c), \end{cases}$$

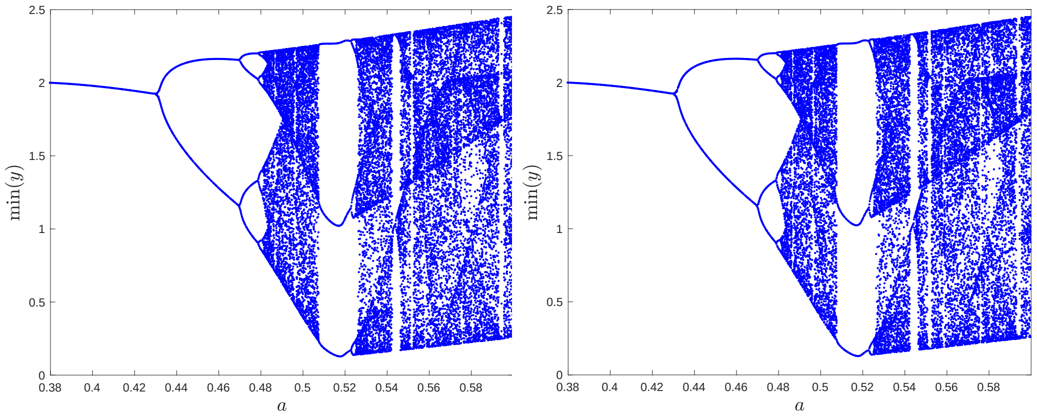
where $0 < \nu \le 1$ is the fractional order; x, y, z are the state variables; and a, b, c are the parameters. $\mathcal{D}_0^{\nu}$ denotes the Liouville–Caputo FD (1) or ABC FD. If $\nu = 1$, it becomes the integer-order Rössler system and is well known.

In this example, we show comparisons of the CPU times using the classical PCM and the fast PCM to obtain the bifurcation diagrams of the local maximum of y versus a for the Liouville–Caputo and ABC fractional Rössler systems, respectively. The parameters are set to $\nu = 0.98$, $b = 4$, $c = 4$, $h = 1/100$, and $T = 1200$ for the numerical simulations. The bifurcation diagrams for the Liouville–Caputo fractional Rössler system obtained by the ABC-FPCM (Figure 3a) and the ABC-PCM (Figure 3b) when parameter a varies from 0.38 to 0.5 are shown in Figure 3. Figure 4 shows the bifurcation diagrams for the ABC fractional system computed by the ABC-FPCM (Figure 4a) and the ABC-PCM (Figure 4b), as parameter a ranges from 0.38 to 0.6, where the increment in parameter a in both systems is set to $\Delta a = 0.0002$. It can be observed that the fast PCM significantly reduces the computational cost. For the Liouville–Caputo Rössler system, the CPU time for the ABC-FPCM is only 19 min, whereas that of the ABC-PCM is 2 days, 12 h, and 32 min. For the ABC system, the ABC-FPCM takes 1 h and 7 min, whereas the ABC-PCM takes four days, 21 h, and 2 min. This confirms that the proposed fast PCM has superior performance and efficiency compared to the conventional PCM.



(a) ***Fast method (19 min)*** **(b)** ***Conventional method (2 days, 12 h, 32 min)***

Figure 3. Bifurcation diagrams for Rössler system with the Liouville–Caputo FD, $0.38 \le a \le 0.5$.



(a) ***Fast method (1 h, 7 min)*** **(b)** ***Conventional method (4 days, 21 h, 2 min)***

Figure 4. Bifurcation diagrams for Rössler system with the ABC FD, $0.38 \leq a \leq 0.6$.

5. Conclusions

In this paper, we have introduced two distinct predictor-corrector methods (PCMs) for solving ABC fractional differential equations: the conventional PCM, based on linear interpolation, and the fast PCM, enhanced by the sum-of-exponentials (SOEs) approximation of $t^{-\beta}$.

Our findings demonstrate that although the fast PCM achieves linear time complexity $\mathcal{O}(N)$, the conventional PCM operates with quadratic time complexity $\mathcal{O}(N^2)$, as illustrated numerically by the CPU time in Figure 1.

We have established truncation and global error estimates, and the uniform accuracy order of $\mathcal{O}(h^2)$ was empirically verified across Examples 1 through 4. Our methods were applied to a range of problems, including nonlinear ABC fractional-order differential problems, ABC fractional-order sub-diffusion PDEs, and the fractional-order Rössler dynamical system with both the ABC FD and Liouville–Caputo FD. Of particular note is the application to the fractional Rössler dynamical system, where the fast PCM demonstrated significant potential in reducing computational costs. This efficiency is especially valuable in the analysis of bifurcation diagrams for various values of ν in fractional dynamical systems, a topic we plan to explore further in subsequent work.

While our methods have shown promising results, we acknowledge certain limitations. The scope of comparison with the existing literature was constrained due to the unique nature of our methods, particularly in the context of ABC PDEs. In future research, we aim to extend our comparative analysis and explore the full potential of our methods in a broader range of applications. Additionally, we plan to address the practical implications and applications of our fast PCM in analyzing bifurcation diagrams in fractional dynamical systems in more depth, which could open new avenues in the study of complex dynamical behaviors.

In conclusion, our work represents a significant step forward in the computational handling of ABC fractional differential equations. We have achieved our initial objectives of developing efficient computational methods and have laid a foundation for future explorations that could further enhance the understanding and application of these methods in complex systems.

Author Contributions: Conceptualization, S.L and B.J.; methodology, S.L., H.K. and B.J.; software, S.L. and H.K.; validation, H.K. and B.J.; formal analysis, S.L and B.J.; investigation, H.K. and B.J.; writing—original draft preparation, S.L and B.J.; writing—review and editing, S.L., H.K. and B.J.; visualization, S.L. and H.K.; supervision, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: S.Lee was supported by a National Institute for Mathematical Sciences (NIMS) grant, funded by the Korean government (MSIT) (No. B22810000). B. Jang and H. Kim acknowledge funding support from the National Research Foundation of Korea (NRF) grant, funded by the Korean government (MSIT) (2021R1A2C1011817, 2021R1A4A1032924 and 2022R1F1A1063605)

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data used in this study are included in the article. Further inquiries can be directed to the author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Additional Information on the Sum-of-Exponentials Approximation

In this section, we describe the method for reducing the number of exponentials. Here, we introduce the balanced truncation method (BTM). It is one of the model reduction methods used for linear time-invariant (LTI) systems. First, we explain the motivation for using this method. Our objective is to create a reduced model of the LTI system:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx(t), \tag{A1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times p}$, and $C \in \mathbb{R}^{q\times n}$. The vectors $x \in \mathbb{R}^n, u \in \mathbb{R}^p$, and $y \in \mathbb{R}^q$ are the state variables, input variables, and output variables, respectively. The transfer function is $G(s) = C(sI - A)^{-1}B$. We want to find the reduced system as (A^r, B^r, C^r), i.e.,

$$\dot{x^r}(t) = A^r x^r(t) + B^r u(t), \quad y(t) = Cx^r(t)$$

where $A^r \in \mathbb{R}^{k\times k}, B^r \in \mathbb{R}^{k\times p}$, and $C^r \in \mathbb{R}^{q\times k}$. Before describing this method, let us intuitively observe why it can be used to reduce the number of quadratures. We set $A = -\mathrm{diag}(\eta_1, ..., \eta_{N_{\mathrm{exp}}})$, $C = \sqrt{(\zeta_1, ..., \zeta_{N_{\mathrm{exp}}})}$, and $B = C^T$, where η_i's and ζ_i's are the quadrature nodes and weights, respectively. The transfer function is given by

$$G(s) = \sum_{i=1}^{N_{\mathrm{exp}}} \frac{\zeta_i}{s - (-\eta_i)} = \mathcal{L}\left\{\sum_{i=1}^{N_{\mathrm{exp}}} e^{-T\eta_i}\right\}(s) \tag{A2}$$

where $T = tp$. Note that this is the Laplace transform of our sum-of-exponentials approximation. So, we find the reduced form:

$$\hat{G}(s) = \sum_{i=1}^{k} \frac{\hat{\zeta}_i}{s - (-\hat{\eta}_i)}, \quad k \ll N_{\mathrm{exp}}, \tag{A3}$$

By applying the inverse Laplace transform, we obtain the reduced SOE approximation $\mathcal{L}^{-1}\{\hat{G}\}(T) = \sum_{i=1}^{k} \hat{\zeta}_i e^{-T\hat{\eta}_i}$. Therefore, we can use this method to reduce the number of quadratures. Now, we state some important definitions and theorems.

Definition A1 ([39]). *When determining the specific situation of the system, we use* Gramians. *As an example, Gramians are used to determine whether a system is controllable or reachable. The reachability Gramian and the observability Gramian of the system* (A1) *are defined as follows:*

- *The reachability Gramian*

$$P := \int_0^{t_f} e^{\tau A} B B^T e^{\tau A^T} d\tau. \tag{A4}$$

- *The observability Gramian*

$$Q := \int_0^{t_f} e^{\tau A} C^T C e^{\tau A^T} d\tau. \tag{A5}$$

where t_f is a fixed final time. Assume that A is asymptotically stable and $t_f = \infty$. In this case, P and Q are the solutions of the following Lyapunov equations:

$$AP + PA^T + BB^T = 0 \tag{A6}$$

$$A^T Q + QA + C^T C = 0. \tag{A7}$$

If the system (A1) is asymptotically stable and minimal, the two Lyapunov equations (A6) and (A7) have unique symmetric positive definite solutions.

Definition A2 ([40])**.** *The square roots of the eigenvalues (singular values) of the product PQ are the so-called* Hankel singular values *of the system (A1):*

$$\sigma_i = \sqrt{\lambda_i(PQ)}. \tag{A8}$$

There are some important properties of the Hankel singular values:

- *σ_i are basis independent.*
- *In a lot of instances, not only the eigenvalues of P and Q but also the Hankel singular values decrease very quickly.*
- *Note that one of properties of the balanced basis is that hard-to-reach states are hard to observe. Therefore, we obtain the reduced model by using the Hankel singular values, except for the small ones.*

Table A1. The number of exponentials and relative errors for the approximation of $t^{-\beta}$ with $T = 1$.

δ	$\mathbf{10^{-2}}$		$\mathbf{10^{-3}}$		$\mathbf{10^{-4}}$		$\mathbf{10^{-5}}$	
ϵ	N_{exp}	E_{rel}	N_{exp}	E_{rel}	N_{exp}	E_{rel}	N_{exp}	E_{rel}
				Theorem 1				
10^{-3}	84	2.85×10^{-10}	119	2.74×10^{-11}	147	2.87×10^{-11}	182	2.91×10^{-11}
10^{-6}	182	5.29×10^{-16}	238	6.15×10^{-16}	308	7.97×10^{-16}	378	9.39×10^{-16}
10^{-9}	273	8.01×10^{-16}	378	1.07×10^{-15}	462	1.10×10^{-15}	567	1.30×10^{-15}
				$\beta = 0.2$				
10^{-3}	15	1.01×10^{-4}	19	9.18×10^{-5}	23	1.11×10^{-4}	27	1.14×10^{-4}
10^{-6}	26	1.60×10^{-7}	32	1.50×10^{-7}	39	9.14×10^{-8}	46	1.21×10^{-7}
10^{-9}	37	1.10×10^{-10}	47	8.56×10^{-11}	54	7.89×10^{-11}	64	1.16×10^{-10}
				$\beta = 0.5$				
10^{-3}	14	1.62×10^{-4}	19	1.52×10^{-4}	22	1.72×10^{-4}	26	1.75×10^{-4}
10^{-6}	25	3.23×10^{-7}	31	2.99×10^{-7}	38	1.57×10^{-7}	45	2.20×10^{-7}
10^{-9}	36	2.12×10^{-10}	46	1.64×10^{-10}	53	1.44×10^{-10}	62	2.10×10^{-10}
				$\beta = 0.8$				
10^{-3}	14	3.19×10^{-4}	18	2.96×10^{-4}	21	3.35×10^{-4}	25	3.41×10^{-4}
10^{-6}	25	6.46×10^{-7}	30	5.62×10^{-7}	37	3.56×10^{-7}	43	4.48×10^{-7}
10^{-9}	35	5.44×10^{-10}	44	4.22×10^{-10}	52	4.47×10^{-10}	61	4.31×10^{-10}

Table A2. The number of exponentials and relative errors for the approximation of $t^{-\beta}$ with $\delta = 10^{-2}$.

T	**10**		$\mathbf{10^{2}}$		$\mathbf{10^{3}}$		$\mathbf{10^{4}}$	
ϵ	N_{exp}	E_{rel}	N_{exp}	E_{rel}	N_{exp}	E_{rel}	N_{exp}	E_{rel}
				Theorem 1				
10^{-3}	98	2.85×10^{-10}	119	2.74×10^{-11}	133	2.74×10^{-11}	147	2.74×10^{-11}
10^{-6}	210	6.09×10^{-16}	238	6.15×10^{-16}	280	6.15×10^{-16}	308	6.15×10^{-16}
10^{-9}	336	8.84×10^{-16}	378	1.07×10^{-15}	420	1.07×10^{-15}	462	1.07×10^{-15}
				$\beta = 0.2$				
10^{-3}	16	2.24×10^{-4}	19	1.48×10^{-4}	21	2.33×10^{-4}	23	1.76×10^{-4}
10^{-6}	29	1.15×10^{-7}	32	2.78×10^{-7}	36	2.51×10^{-7}	39	1.89×10^{-7}
10^{-9}	42	1.39×10^{-10}	47	1.14×10^{-10}	50	2.16×10^{-10}	54	2.20×10^{-10}
				$\beta = 0.5$				
10^{-3}	16	3.89×10^{-4}	19	3.37×10^{-4}	20	6.79×10^{-4}	22	5.16×10^{-4}
10^{-6}	29	2.54×10^{-7}	31	6.58×10^{-7}	35	8.26×10^{-7}	38	6.23×10^{-7}
10^{-9}	41	2.77×10^{-10}	46	2.56×10^{-10}	49	5.90×10^{-10}	53	7.82×10^{-10}
				$\beta = 0.8$				
10^{-3}	15	7.04×10^{-4}	18	4.08×10^{-4}	19	1.23×10^{-3}	21	9.38×10^{-4}
10^{-6}	28	5.32×10^{-7}	30	1.05×10^{-6}	34	1.66×10^{-6}	37	1.27×10^{-6}
10^{-9}	40	6.80×10^{-10}	44	6.08×10^{-10}	48	1.27×10^{-9}	52	1.70×10^{-9}

In order to obtain the reduced system, the following theorem is crucial:

Theorem A1 ([41])**.** *Assume that the asymptotically stable, minimal realization (A, B, C) of the full-order model transfer function, $G(s) = C(Is - A)^{-1}B$, is internally balanced, i.e.,*

$$A\Lambda + \Lambda A^T + BB^T = 0, \tag{A9}$$
$$A^T\Lambda + \Lambda A + C^T C = 0, \tag{A10}$$

where $\Lambda = \mathrm{diag}\{\sigma_i\}$*,* $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ *and* σ_i*s are the Hankel singular values of the system. If we obtain the reduced model with the following transfer function*

$$\hat{G}(s) = C_1(Is - A_{11})^{-1}B_1,$$

where $A_{11} \in \mathbb{R}^{k\times k}$*,* $B \in \mathbb{R}^{k\times p}$*, and* $C \in \mathbb{R}^{q\times k}$*, the model reduction error is bounded*

$$\|G(s) - \hat{G}(s)\|_{\mathcal{H}_\infty} \leq 2\sum_{i=k+1}^{n} \sigma_i. \tag{A11}$$

where $\|\cdot\|_{\mathcal{H}_\infty} = \sup_{z\in i\mathbb{R}} |\cdot|$ *is the* L_∞ *norm for the transfer function.*

We use the square-root method described in Algorithm A1. In our case, we use $S = U_P\Sigma_P^{1/2}$ and $L = U_Q\Sigma_Q^{1/2}$ instead of the Cholesky factors of P and Q because they are sometimes not positive-definite matrices in the numerical sense (i.e., they have some very small negative eigenvalues $-10^{-16} \lesssim \lambda_i < 0$). Moreover, the singular values of LS^T are also the Hankel singular values of the system (A1). Therefore, we can obtain the reduced system with the desired error ϵ using the balanced truncation method.

Tables A1 and A2 show the number of exponentials N_{exp} and the relative error E_{rel} for the approximations of $t^{-\beta}$ with some $\epsilon, T, \delta > 0$. To show the effectiveness of the balanced truncation method in Algorithm A1, we set $\hat{M} = \lceil \log(\log(1/\epsilon)) + \log(1/\delta^2) + \log T \rceil + 1$ and $n = \lceil \log(1/\epsilon) \rceil$ in Theorem 1. When the error bound ϵ becomes smaller, the number of exponentials N_{exp} also increases. It can be observed that the relative error E_{rel} is much smaller than the lowest value of ϵ using the method in Theorem 1. However, it is worth noting that the balanced truncation method in Algorithm A1 controls the number of exponentials N_{exp} with respect to the error bound ϵ. From the results in the tables, we can verify that the model reduction method significantly reduces the number of exponentials N_{exp}. It should be noted that the number of exponentials N_{exp} in Theorem 1 is almost independent of the value of β in approximating $t^{-\beta}$.

Algorithm A1: Balanced truncation method [42]

```
Data: The desired error $\epsilon$, quadrature points $\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_N)$, and weights
        $\boldsymbol{\zeta} = (\zeta_1, \cdots, \zeta_N)$.
Result: Reduced quadrature points $\hat{\boldsymbol{\eta}}$ and weights $\hat{\boldsymbol{\zeta}}$.
Set $A = -\mathrm{diag}(\boldsymbol{\eta})$, $C = \sqrt{\boldsymbol{\zeta}}$, and $B = C^T$;
Solve two Lyapunov equations $AP + PA^T + BB^T = 0$ and $AQ + QA^T + C^TC = 0$;
Compute two singular-value decompositions of $P = U_P\Sigma_P V_P$ and $Q = U_Q\Sigma_Q V_Q$ ;
Set $S = U_P\Sigma_P^{1/2}$ and $L = U_Q\Sigma_Q^{1/2}$;
Compute a singular-value decomposition of $LS^T = U\Sigma V$, where
  $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_N)$;
Find $k$ such that $2\sum_{j=k+1}^{N} \sigma_j \leq \epsilon$;
Form a $N \times k$ matrix $J$, where $J_{ii} = \sigma_i^{-1/2}$, $i = 1, \cdots, k$ and $J_{ij} = 0$ otherwise;
Set $T_l = L^TUJ$ and $T_r = S^TVJ$;
Set $\hat{A} = T_l^TAT_r$ , $\hat{B} = T_l^TB$, and $\hat{C} = CT_r$;
Compute the eigenvalue decomposition of $\hat{A} = X\Lambda X^{-1}$;
Set $\hat{\boldsymbol{\eta}} = (\Lambda_{11}, \cdots, \Lambda_{kk})$ ;
Form $\tilde{B} = X^{-1}\hat{B}^T$ and $\tilde{C} = \hat{C}X$;
Set $\hat{\boldsymbol{\zeta}} = (\tilde{B}_i\tilde{C}_i)_{i=1}^{k}$;
```

References

1. Goychuk, I.; Heinsalu, E.; Patriarca, M.; Schmid, G.; Hänggi, P. Current and universal scaling in anomalous transport. *Phys. Rev. E* **2006**, *73*, 020101. [CrossRef]
2. Klages, R.; Radons, G.; Sokolov, I.M. *Anomalous Transport: Foundations and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
3. Zaslavsky, G.M. Chaos, fractional kinetics, and anomalous transport. *Phys. Rep.* **2002**, *371*, 461–580. [CrossRef]
4. Caputo, M. Models of flux in porous media with memory. *Water Resour. Res.* **2000**, *36*, 693–705. [CrossRef]
5. Podlubny, I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*; Elsevier: Amsterdam, The Netherlands, 1998; Volume 198.
6. Diethelm, K. *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*; Springer Science & Business Media: Cham, Switzerland, 2010.
7. Sabatier, J.; Agrawal, O.P.; Machado, J.A.T. *Advances in Fractional Calculus: Theoretical Developments and Applications in Physics and Engineering*; Springer: Dordrecht, The Netherlands, 2007.
8. Petráš, I. *Fractional-Order Nonlinear Systems: Modeling, Analysis and Simulation*; Springer Science & Business Media: Cham, Switzerland, 2011.
9. Giusti, A. A comment on some new definitions of fractional derivative. *Nonlinear Dyn.* **2018**, *93*, 1757–1763. [CrossRef]
10. Caputo, M. Linear models of dissipation whose Q is almost frequency independent—II. *Geophys. J. Int.* **1967**, *13*, 529–539. [CrossRef]
11. Atangana, A.; Gómez-Aguilar, J. Fractional derivatives with no-index law property: Application to chaos and statistics. *Chaos, Solitons Fractals* **2018**, *114*, 516–535. [CrossRef]
12. Srivastava, H.M. Some parametric and argument variations of the operators of fractional calculus and related special functions and integral transformations. *J. Nonlinear Convex Anal.* **2021**, *22*, 1501–1520.
13. Srivastava, H.M. An introductory overview of fractional-calculus operators based upon the Fox-Wright and related higher transcendental functions,. *J. Adv. Eng. Comput.* **2021**, *5*, 135–166. [CrossRef]
14. Atangana, A.; Baleanu, D. New fractional derivatives with nonlocal and non-singular kernel: Theory and application to heat transfer model. *Therm. Sci.* **2016**, *20*, 763–769. [CrossRef]
15. Gómez, J.F. *Fractional Derivatives with Mittag-Leffler Kernel: Trends and Applications in Science and Engineering*; Springer: Berlin/Heidelberg, Germany, 2019.
16. Gómez-Aguilar, J.; Atangana, A. Fractional Derivatives with the Power-Law and the Mittag–Leffler Kernel Applied to the Nonlinear Baggs–Freedman Model. *Fractal Fract.* **2018**, *2*, 10. [CrossRef]
17. Owolabi, K.M. Modelling and simulation of a dynamical system with the Atangana-Baleanu fractional derivative. *Eur. Phys. J. Plus* **2018**, *133*, 15. [CrossRef]
18. Alqahtani, R.T. Atangana-Baleanu derivative with fractional order applied to the model of groundwater within an unconfined aquifer. *J. Nonlinear Sci. Appl.* **2016**, *9*, 3647–3654. [CrossRef]
19. Saad, K.M.; Khader, M.; Gómez-Aguilar, J.; Baleanu, D. Numerical solutions of the fractional Fisher's type equations with Atangana-Baleanu fractional derivative by using spectral collocation methods. *Chaos Interdiscip. J. Nonlinear Sci.* **2019**, *29*, 023116. [CrossRef] [PubMed]
20. Baleanu, D.; Jajarmi, A.; Hajipour, M. On the nonlinear dynamical systems within the generalized fractional derivatives with Mittag–Leffler kernel. *Nonlinear Dyn.* **2018**, *94*, 397–414. [CrossRef]
21. Diethelm, K.; Ford, N.J.; Freed, A.D. A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dyn.* **2002**, *29*, 3–22. [CrossRef]
22. Djida, J.; Atangana, A.; Area, I. Numerical computation of a fractional derivative with non-local and non-singular kernel. *Math. Model. Nat. Phenom.* **2017**, *12*, 4–13. [CrossRef]
23. Nguyen, T.B.; Jang, B. A high-order predictor-corrector method for solving nonlinear differential equations of fractional order. *Fract. Calc. Appl. Anal.* **2017**, *20*, 447–476. [CrossRef]
24. Kim, H.; Lee, J.; Jang, B. An efficient numerical approach for solving two-point fractional order nonlinear boundary value problems with Robin boundary conditions. *Adv. Differ. Equ.* **2021**, *193*. [CrossRef]
25. Lee, S.; Lee, J.; Kim, H.; Jang, B. A fast and high-order numerical method for nonlinear fractional-order differential equations with non-singular kernel. *Appl. Numer. Math.* **2021**, *163*, 57–76. [CrossRef]
26. Sun, Z.z.; Wu, X. A fully discrete difference scheme for a diffusion-wave system. *Appl. Numer. Math.* **2006**, *56*, 193–209. [CrossRef]
27. Cui, M. Compact finite difference method for the fractional diffusion equation. *J. Comput. Phys.* **2009**, *228*, 7792–7804. [CrossRef]
28. Brunner, H.; Hairer, E.; Nørsett, S. Runge-Kutta theory for Volterra integral equations of the second kind. *Math. Comput.* **1982**, *39*, 147–163. [CrossRef]
29. Capobianco, G.; Conte, D.; Del Prete, I.; Russo, E. Fast Runge–Kutta methods for nonlinear convolution systems of Volterra integral equations. *BIT Numer. Math.* **2007**, *47*, 259–275. [CrossRef]
30. Deng, J.; Zhao, L.; Wu, Y. Efficient algorithms for solving the fractional ordinary differential equations. *Appl. Math. Comput.* **2015**, *269*, 196–216. [CrossRef]
31. Kumar, P.; Agrawal, O.P. An approximate method for numerical solution of fractional differential equations. *Signal Process.* **2006**, *86*, 2602–2610. [CrossRef]

32. Li, C.; Chen, A.; Ye, J. Numerical approaches to fractional calculus and fractional ordinary differential equation. *J. Comput. Phys.* **2011**, *230*, 3352–3368. [CrossRef]
33. Owolabi, K.M.; Atangana, A. Chaotic behaviour in system of noninteger-order ordinary differential equations. *Chaos, Solitons Fractals* **2018**, *115*, 362–370. [CrossRef]
34. lknur Koca. Efficient numerical approach for solving fractional partial differential equations with non-singular kernel derivatives. *Chaos Solitons Fractals* **2018**, *116*, 278–286. [CrossRef]
35. Jiang, S.; Zhang, J.; Zhang, Q.; Zhang, Z. Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations. *Commun. Comput. Phys.* **2017**, *21*, 650–678. [CrossRef]
36. Cao, J.; Xu, C. A high order schema for the numerical solution of the fractional ordinary differential equations. *J. Comput. Phys.* **2013**, *238*, 154–168. [CrossRef]
37. Garvie, M.R. Finite-difference schemes for reaction–diffusion equations modeling predator–prey interactions in M ATLAB. *Bull. Math. Biol.* **2007**, *69*, 931–956. [CrossRef] [PubMed]
38. Li, Z.; Chen, D.; Ma, M.; Zhang, X.; Wu, Y. Feigenbaum's constants in reverse bifurcation of fractional-order Rössler system. *Chaos Solitons Fractals* **2017**, *99*, 116–123. [CrossRef]
39. Laub, A.; Heath, M.; Paige, C.; Ward, R. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Trans. Autom. Control* **1987**, *32*, 115–122. [CrossRef]
40. Gugercin, S.; Antoulas, A.C. A survey of model reduction by balanced truncation and some new results. *Int. J. Control* **2004**, *77*, 748–766. [CrossRef]
41. Enns, D.F. Model reduction with balanced realizations: An error bound and a frequency weighted generalization. In Proceedings of the 23rd IEEE Conference on Decision and Control, Las Vegas, NV, USA, 12–14 December 1984; IEEE: Manhattan, NY, USA, 1984; pp. 127–132.
42. Xu, K.; Jiang, S. A bootstrap method for sum-of-poles approximations. *J. Sci. Comput.* **2013**, *55*, 16–39. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.