Numerical Methods for Solving Fractional Differential Equations

Keon Ho Kim

Department of Mathematical Sciences
Graduate School of UNIST

Numerical Methods for Solving Fractional Differential Equations

A thesis
submitted to the Graduate School of UNIST
in partial fulfillment of the
requirements for the degree of
Master of Science

Keon Ho Kim

06.15.2018
Approved by

Advisor
Bongsoo Jang

Numerical Methods for Solving Fractional Differential Equations

Keon Ho Kim

This certifies that the thesis of Keon Ho Kim is approved.

06.15.2018

Advisor: Bongsoo Jang

Chang-Yeol Jung: Thesis Committee Member #1

Bongsuk Kwon: Thesis Committee Member #2

I dedicate this thesis to my parents

Abstract

In this thesis, several efficient numerical methods are proposed to solve initial value problems and boundary value problems of fractional differential equations.

For fractional initial value problems, we propose a new type of the predictor-evaluate-corrector-evaluate method based on the Caputo fractional derivative operator. Furthermore, we propose a new type of the Caputo fractional derivative operator that does not have a differential form of a solution. However, with some fractional orders, there are problems that a solution blows up and the scheme has a low convergence. Thus, we identify new treatments for these values. Then, we can expect a significant improvement for all fractional orders. The advantages and improvements are shown by testing various numerical examples.

For fractional BVPs, we propose an explicit method that dramatically reduces the computational time for solving a dense matrix system. Moreover, by adopting high-order predictor-corrector methods which have uniform convergence rates $O(h^2)$ or $O(h^3)$ for all fractional orders [8], we propose a second-order method and a third-order method by using the Newton's method and the Halley method, respectively. We show its advantage by testing various numerical examples.

Contents

List of Figures

List of Tables

Chapter 1

Introducation

Fractional calculus has recently been considered as an important mathematical model to describe various phenomena in nature. The origins of fractional calculus date back to the end of the 17th century. It started with a question from L'Hôpital to Leibnitz: "What does $\frac{d^n}{dx^n}f(x)$ mean if $n = 1/2$?" [2]. From this, the field of fractional calculus was born and several well-known definitions of fractional integral and derivative were developed, such as the Riemann-Liouville operators, the Caputo operators, the Hadamard operatos, and the Grünwald-Letnikov derivative. In this paper, we only consider the Riemann-Liouville and Caputo operators to solve fractional differential equations. The main difference between the derivatives of integers and fractions is a non-local property. For example, the derivative of an integer is only defined by a current point. However, the Riemann-Liouville fractional derivative is a form of integral equation with a kernel function that contains all previous information from an initial point to a present point. By cause of the non-local property, fractional derivatives have a unique aspect – the so-called memory effect – differentiating them from regular derivatives. There exists many research that the fractional order can be physically explained as an index of memory [5]. This is why many scientists and engineers use mathematical models with fractional orders to illustrate variety of natural phenomena. However, the robust interpretation of fractional derivative is still an open problem [5]. Moreover, due to the non-local property of a fractional derivative, there still remains many improvements in the conventional numerical approaches for solving fractional differential equations in terms of computational algorithms.

In this paper, we primarily concentrate on constructing efficient numerical algorithms for solving fractional initial value problems (IVPs) and boundary value problems (BVPs). First of all, we propose a new type of numerical method for solving fractional IVPs: the Direct Method. In the conventional method, which is based on the Caputo fractional derivative, the finite difference method is adopted to approximate a derivative of a solution. However, we propose a new type of Caputo fractional derivative without a solution's derivative. It reduces the computational cost by using the same accuracy order. Furthermore, an explicit method for solving FDEs has several problems with some fractional orders. There is a problem of stability with a small fractional order and a problem of low convergence with a large fractional order. To overcome these defects, we propose the enhanced methods for each case. For a small fractional order, the problem of stability is highly improved by using the Newton's method with an initial value from our method. Moreover, we get a higher convergence by decomposing a FDE into a system of equations with small fractional orders compared to the original FDE. Several numerical examples are demonstrated to show the effectiveness for the proposed methods.

Next, the High-Order Method is introduced to solve two-point BVPs of FDEs. In general, we construct a matrix for imposing boundary conditions of FDEs. However, due to the non-local property, it takes many computational costs to solve a matrix equation at each time. For nonlinear fractional problems, we might need huge amounts of computational time to solve a nonlinear system of matrices. To reduce the computational cost, we propose the High-Order Method that changes a BVP to an IVP and use the nonlinear shooting method for updating an approximation of an IVP. Then, the new scheme achieves a uniform accuracy order regardless of the value of fractional order by adopting a modified PECE method [8], the Newton's Method, and the Halley's Method. Several numerical examples are demonstrated to show the effectiveness for the proposed methods.

Chapter 2

Fractional Differential Equations

In this section, we introduce basic definitions and properties in fractional calculus. There exists several well-known definitions which define fractional derivatives, such as the Caputo operators, the Riemann-Liouville operators, and the the Grünwald-Letnikov definition. In addition, we introduce several conventional numerical schemes for solving fractional differential equations.

2.1 Preliminaries

First of all, we begin with a special function which is well-known in fractional calculus.

Definition 2.1.1. *The function* $\Gamma : (0, \infty) \to \mathbb{R}$*, defined by*

$$\Gamma(x) := \int_0^\infty t^{x-1} e^{-t} dt, \tag{2.1}$$

is called Euler's Gamma function (or Euler's integral of the second kind) [2].

Now, we shall introduce fractional integral and derivative operators J^n and D^n, where $n \notin \mathbb{N}$.

Definition 2.1.2. *Let* $n \in \mathbb{R}^+$*. The operator* J_a^n*, defined on* $L_1[a, b]$ *by*

$$J^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-s)^{\alpha-1} f(s) ds \tag{2.2}$$

for $a \leq t < b$*, is called the Riemann-Liouville fractional integral operator of order* α*.*

For the Riemann-Liouville fractional integral operator, the following properties have been known:

Property 2.1.1.

1. *Identity, i.e.,* $J^0 f(t) = f(t)$

2. *Linearity, i.e.,* $J^\alpha(\omega_1 f(t) + \omega_2 g(t)) = \omega_1 J^\alpha f(t) + \omega_2 J^\alpha g(t), \quad \alpha \in \mathbb{R}^+,\ \omega_1, \omega_2 \in \mathbb{C}$

3. *If* $f(t)$ *is continuous for* $t \in \mathbb{R}_0^+$*, then*

   - $\lim\limits_{\alpha \to 0} J^\alpha f(t) = f(t),$
   - $J^\alpha(J^\beta f(t)) = J^\beta(J^\alpha f(t)) = J^{\alpha+\beta} f(t), \quad \alpha, \beta \in \mathbb{R}^+,\ \lambda \in \mathbb{C}$

There is the left-inverse operator of the fractional integral operator [6].

Definition 2.1.3. *Let* $n \in \mathbb{R}$ *and* $m = \lceil n \rceil$*. The operator* D_a^n*, defined by*

$$D_a^n f := D^m J_a^{m-n} f \tag{2.3}$$

is called the Riemann-Liouville fractional differential operator of order n*.*

Definition 2.1.4. *Suppose that* $\alpha > 0,\ t > a,\ \{\alpha, a, t\} \subset \mathbb{R}$*. Then,*

$$\tilde{D}_a^\alpha f(t) = \begin{cases} \frac{1}{\Gamma(m-\alpha)} \frac{d^m}{dt^m} \int_a^t (t-s)^{m-\alpha-1} f(s) ds, & m-1 < \alpha < m \in \mathbb{N}, \\ \frac{d^m}{dt^m} f(t), & \alpha = m \in \mathbb{N}, \end{cases} \tag{2.4}$$

is called the Riemann-Liouville fractional derivative or the Riemann-Liouville fractional differential operator of order α *[9].*

Theorem 2.1.2. *Let* $n > 0$*. If there exists some* $\phi \in L_1[a, b]$ *such that* $f = J_a^n \phi$*, then*

$$J_a^n D_a^n f = f \tag{2.5}$$

almost everywhere.

However, we are interested in the Caputo fractional operator, which is an alternative operator to the Riemann-Liouville fractional differential operator. The most significant point is that they do not coincide in general and the following is not the left-inverse operator of the Riemann-Liouville fractional integral operator.

Definition 2.1.5. *Suppose that $\alpha > 0,\ t > a,\ \{\alpha, a, t\} \subset \mathbb{R}$. The fractional operator*

$$D_a^\alpha f(t) = \begin{cases} \frac{1}{\Gamma(m-\alpha)} \int_a^t (t-s)^{m-\alpha-1} f^{(m)}(s) ds, & m-1 < \alpha < m \in \mathbb{N}, \\ \frac{d^m}{dt^m} f(t), & \alpha = m \in \mathbb{N} \end{cases} \tag{2.6}$$

is called the Caputo fractional derivative or Caputo fractional differential operator of order α [1].

The Caputo fractional operator is equivalent to $(m-\alpha)$-fold integration after m-th order differentiation [6]. It means the following lemma:

Lemma 2.1.3. *If $f(t)$ is a function such that $\exists\ D_a^\alpha f(t)$, then*

$$D_a^\alpha f(t) = J^{m-\alpha} D^m f(t),\ \text{where } m-1 < \alpha < m,\ m \in \mathbb{N},\ \alpha \in \mathbb{R}_0^+ \tag{2.7}$$

We will study linear fractional differential equations in the Caputo sense, and one of the useful formulations regarding the Caputo fractional derivative is the following [7, 4]:

Theorem 2.1.4. *The Caputo fractional derivative of the power function satisfies*

$$D_a^\alpha t^p = \begin{cases} \frac{\Gamma(p+1)}{\Gamma(p-\alpha+1)} t^{p-\alpha}, & m-1 < \alpha < m,\ p > m-1,\ p \in \mathbb{R}, \\ 0, & m-1 < \alpha < m,\ p \le m-1,\ p \in \mathbb{N}. \end{cases} \tag{2.8}$$

Definition 2.1.6. *Let $n > 0$. The function E_n defined by*

$$E_n(z) := \sum_{j=0}^{\infty} \frac{z^j}{\Gamma(jn+1)} \tag{2.9}$$

whenever the series converges is called the Mittag-Leffler function of order n [2].

Definition 2.1.7. *Let $n_1, n_2 > 0$. The function E_{n_1,n_2} defined by*

$$E_{n_1,n_2}(z) := \sum_{j=0}^{\infty} \frac{z^j}{\Gamma(jn_1+n_2)} \tag{2.10}$$

whenever the series converges is called the two-parameter Mittag-Leffler function with parameters n_1 and n_2 [2].

Lemma 2.1.5. *Let $n \in \mathbb{R}^+$, $m = \lceil n \rceil$, and $b > 0$. Assume that the function $f : G \to \mathbb{R}$ is continuous and bounded in G and that it fulfills a Lipschitz condition with respect to the second* *variable, i.e. there exist a constant* $L > 0$ *such that, for all* (x, y_1) *and* $(x, y_2) \in G$*, we have*

$$|f(x, y_1) - f(x, y_2)| < L\,|y_1 - y_2|\,. \tag{2.11}$$

Then, the function $y \in C(a, b)$ *is a solution of the fractional differential equation*

$$\begin{cases} D_a^n y(x) = f(x, y(x)), \\ D_a^k y(a) = y_k, \;\; k = 0, 1, \cdots, m-1, \end{cases} \tag{2.12}$$

if and only if it is a solution of the Volterra integral equation

$$y(x) = \sum_{k=0}^{m-1} y_k \frac{x^k}{k!} + \frac{1}{\Gamma(n)} \int_a^x (x-t)^{n-1} f(t, y(t)) dt \;\; [2]. \tag{2.13}$$

2.2 Conventional Numerical Methods for Solving FDEs

In this section, we introduce well-known conventional methods for solving fractional differential equations. To be precise, the fractional differential equations is

$$D^\alpha y(t) = f(t, y(t)), \tag{2.14}$$

where $\alpha \in \mathbb{R}^+$. Let $m = \lceil \alpha \rceil$, and a solution $y(t)$ is on the interval $[a, T]$, where $T > 0$. Then, there are two typical approaches to solve the equation; by using the Caputo fractional derivative operator or the Riemann-Liouville fractional integral operator.

Let $t \in \Omega := [a, T]$.

- *The Caputo Operator:*

$$\frac{1}{\Gamma(m-\alpha)} \int_a^t (t-s)^{m-\alpha-1} y^{(m)}(s) ds = f(t, y(t)). \tag{2.15}$$

- *The Riemann-Liouville Operator:*

$$y(t) = g(t) + \frac{1}{\Gamma(\alpha)} \int_a^t (t-s)^{\alpha-1} f(s, y(s)) ds, \tag{2.16}$$

where $g(t) = \sum\limits_{k=0}^{m-1} y_k \frac{x^k}{k!}$ is an initial condition.

Let us discretize the domain Ω to be

$$\Phi_N := \{t_j : a = t_1 < \ldots < t_j < \ldots < t_n < t_{n+1} < \ldots < t_N = T\}. \tag{2.17}$$

By (2.15), (2.16), and (2.17), we get discretized integral equations

$$\frac{1}{\Gamma(m-\alpha)}\left[\sum_{j=1}^{n-2}\int_{t_j}^{t_{j+1}}(t_n-s)^{m-\alpha-1}y^{(m)}(s)ds + \int_{t_{n-1}}^{t_n}(t_n-s)^{m-\alpha-1}y^{(m)}(s)ds\right] = f(t_n, y(t_n)), \tag{2.18}$$

$$y(t_n) = g(t_n) + \frac{1}{\Gamma(\alpha)}\left[\sum_{j=1}^{n-2}\int_{t_j}^{t_{j+1}}(t_n-s)^{\alpha-1}f(s,y(s))ds + \int_{t_{n-1}}^{t_n}(t_n-s)^{\alpha-1}f(s,y(s))ds\right]. \tag{2.19}$$

Define a memory term and a local term as follows.

- *The Caputo Operator:*

$$\begin{cases} \text{Memory term: } \sum_{j=1}^{n-2}\int_{t_j}^{t_{j+1}}(t_n-s)^{m-\alpha-1}y^{(m)}(s)ds \\ \text{Local term: } \int_{t_{n-1}}^{t_n}(t_n-s)^{m-\alpha-1}y^{(m)}(s)ds \end{cases} \tag{2.20}$$

- *The Riemann-Liouville Operator:*

$$\begin{cases} \text{Memory term: } \sum_{j=1}^{n-2}\int_{t_j}^{t_{j+1}}(t_n-s)^{\alpha-1}f(s,y(s))ds \\ \text{Local term: } \int_{t_{n-1}}^{t_n}(t_n-s)^{\alpha-1}f(s,y(s))ds \end{cases} \tag{2.21}$$

Then, we can solve the fractional differential equation by only updating a numerical solution on a current interval. There are several ways to update a numerical solution.

Now, we look at the popular algorithm that is so-called Predictor-Evaluate-Corrector-Evaluate (PECE) method [3]. The PECE method was first proposed by Kai Diethelm in 2001 and is still well-used to solve fractional differential equations. This approach is mainly based on the Volterra integral equation in Lemma 2.1.5

$$y(t_n) = g(t_n) + \frac{1}{\Gamma(\alpha)}\left[\sum_{j=1}^{n-1}\int_{t_j}^{t_{j+1}}(t_n-s)^{\alpha-1}f(s,y(s))ds\right]. \tag{2.22}$$

We apply the linear interpolation for f on each interval

$$\int_{t_j}^{t_{j+1}} (t_n - s)^{\alpha-1} f(s, y(s))ds \approx \int_{t_j}^{t_{j+1}} (t_n - s)^{\alpha-1} \tilde{f}_j(s, y(s))ds, \tag{2.23}$$

where $\tilde{f}_j$ is the piecewise linear interpolant for f on $[t_j, t_{j+1}]$ for $j = 1, \cdots, n-1$. We can rewrite the integral by using the standard quadrature theory as

$$\sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} (t_n - s)^{\alpha-1} \tilde{f}_j(s, y(s))ds = \sum_{j=1}^{n} C_j f(t_j, y(t_j)), \tag{2.24}$$

$$\sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} (t_n - s)^{\alpha-1} \tilde{f}_j(s, y(s))ds = \sum_{j=1}^{n-1} D_j f(t_j, y(t_j)), \tag{2.25}$$

where C_j and D_j are coefficients which are generated by the linear interpolation. Let $\tilde{y}$ be a numerical solution. Then, it gives us the PECE method, which is

$$\tilde{y}(t_n) = g(t_n) + \frac{1}{\Gamma(\alpha)} \sum_{j=1}^{n-1} C_j f(t_j, y(t_j)) + \frac{h^\alpha}{\Gamma(\alpha+2)} f(t_n, \tilde{y}^P(t_n)), \tag{2.26}$$

where

$$\tilde{y}^P(t_n) = g(t_n) + \frac{1}{\Gamma(\alpha)} \sum_{j=1}^{n-1} D_j f(t_j, y(t_j)). \tag{2.27}$$

The convergence analysis shows that the error is expected to behave as

$$\max_{j=1,\cdots,n} |y(t_j) - \tilde{y}_j| = O(h^p), \tag{2.28}$$

where $p = \min(2, 1+\alpha)$.

Chapter 3

Numerical Method for Solving Fractional IVPs

In this section, we consider the following ordinary differential equation with fractional order $\alpha \in \mathbb{R}^+$,

$$\begin{cases} D_a^\alpha y(t) = \tilde{f}(t, y(t)), \quad t \in [a, T], \\ y^{(k)}(a) = y_k, \quad k = 0, 1, \ldots, m-1, \end{cases} \tag{3.1}$$

where $m - 1 < \alpha \leq m \in \mathbb{Z}^+$.

3.1 Direct Method

3.1.1 Description of Direct Method

To solve the problem numerically, we adopt a fractional derivative in the Caputo sense (2.1.5), i.e.,

$$D_a^\alpha y(t) = \frac{1}{\Gamma(m-\alpha)} \int_a^t (t-s)^{m-1-\alpha} y^{(m)}(s) ds, \tag{3.2}$$

because it imposes the initial conditions with homogeneous conditions. Furthermore, let $0 < \alpha < 1$ for the convenience of computation. In conventional methods based on the Caputo fractional derivative, the derivative of a solution is approximated by the linear interpolation or quadratic interpolation. However, in this method, we eliminate the derivative by using the integration by parts. We can transform Eq. (3.2) into a form without the derivative of a solution as follows:

$$\int_a^t \left[\frac{y(t)-y(s)}{(t-s)^\alpha}\right]' ds = -\frac{y(t)-y(a)}{(t-a)^\alpha} = -\int_a^t \frac{y'(s)}{(t-s)^\alpha}ds + \alpha\int_a^t \frac{y(t)-y(s)}{(t-s)^{\alpha+1}}ds. \tag{3.3}$$

Then, we have the Caputo derivative without the derivative, i.e.,

$$\Gamma(1-\alpha)D_a^\alpha y(t) = \int_a^t \frac{y'(s)}{(t-s)^\alpha}ds = \frac{y(t)-y(a)}{(t-a)^\alpha} + \alpha\int_a^t \frac{y(t)-y(s)}{(t-s)^{\alpha+1}}ds. \tag{3.4}$$

In terms of numerical approaches, the main difficulty one has to tackle is the non-locality property of the solution $y(t)$ due to the kernel $(t-s)^{\alpha-1}$ under the integral equation on the right-hand side of Eq. (3.4). In order to illustrate this, let us first discretize the grid to be

$$\Phi_N := \{t_j : a = t_1 < \ldots < t_j < \ldots < t_n < t_{n+1} < \ldots < t_N = T\}. \tag{3.5}$$

For simplicity, we assume that the grid is uniform, i.e., $h = t_{j+1} - t_j$, $\forall j = 1, \ldots, N-1$. Let $\mathbb{D}_a^\alpha = \Gamma(1-\alpha)D_a^\alpha$ and $f(t, u(t)) = \Gamma(1-\alpha)\tilde{f}(t, y(t))$. By multiplying $\Gamma(1-\alpha)$ to the both sides of (3.1) and employing (3.4), we have

$$\frac{y(t)-y(t_1)}{(t-t_1)^\alpha} + \alpha\int_{t_1}^t \frac{y(t)-y(s)}{(t-s)^{\alpha+1}}ds = f(t, y(t)). \tag{3.6}$$

3.1.2 Direct Method with Linear Interpolation

Before proceeding, we first define some notations. We denote $y_j = y(t_j)$ the restriction of the exact solution at time t_j, $j = 1, \ldots, N$. Let $\tilde{y}_n$ be the approximations of y_n. Similarly, we also denote $f_j = f(t_j, y_j)$, $\tilde{f}_j = f(t_j, \tilde{y}_j)$. For clarification, we re-derive the ABM ([3]) as follows. The exact solution (3.4) is written as

$$\frac{y(t_n)-y(a)}{(t_n-a)^\alpha} + \alpha\int_{t_1}^{t_n} \frac{y(t_n)-y(s)}{(t_n-s)^{\alpha+1}}ds = f(t_n, y(t_n)). \tag{3.7}$$

On each interval $I_j = [t_j, t_{j+1}]$, we interpolate $y(t)$ by a linear Lagrange polynomial

$$y(t) \approx L_j^1 y(t) = \frac{t_{j+1}-t}{h}y(t_j) + \frac{t-t_j}{h}y(t_{j+1}). \tag{3.8}$$

The value of $f(t_n, u(t_n))$ can be approximated by using the linear interpolation of $f(t, u(t))$ with grid point t_{n-2} and t_{n-1},

$$f(t_n, y(t_n)) \approx L^1_{n-2} f_n = -f(t_{n-2}, y(t_{n-2})) + 2f(t_{n-1}, y(t_{n-1})). \tag{3.9}$$

Substituting the approximation (3.8) and (3.9) into (3.7), taking into account the approximated values $\tilde{y}_j$ and that the grid is uniform, we obtain that

$$\frac{\tilde{y}_n - y(a)}{(t_n - a)^\alpha} + \alpha \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{\tilde{y}_n - L^1_j \tilde{y}(s)}{(t_n - s)^{\alpha+1}} ds = L^1_{n-2} \tilde{f}_n, \tag{3.10}$$

where

$$L^1_j \tilde{y}(t) = \frac{t_{j+1} - t}{h} \tilde{y}_j + \frac{t - t_j}{h} \tilde{y}_{j+1}, \quad L^1_{n-2} \tilde{f}_n = -f(t_{n-2}, \tilde{y}_{n-2}) + 2f(t_{n-1}, \tilde{y}_{n-1}).$$

On (t_{n-1}, t_n) the integral can be simplified by

$$\int_{t_{n-1}}^{t_n} \frac{\tilde{y}_n - L^1_{n-1} \tilde{y}(s)}{(t_n - s)^{\alpha+1}} ds = \frac{1}{h} \int_{t_{n-1}}^{t_n} \frac{\tilde{y}_n - \tilde{y}_{n-1}}{(t_n - s)^\alpha} ds = \frac{1}{(1-\alpha) h^\alpha} (\tilde{y}_n - \tilde{y}_{n-1}). \tag{3.11}$$

Therefore, $\tilde{y}_n$ can be evaluated by solving the following equation

$$\left[\frac{1}{(t_n - a)^\alpha} + \alpha \sum_{j=1}^{n-2} A_{n,j} + \frac{\alpha}{(1-\alpha) h^\alpha} \right] \tilde{y}_n = \alpha \sum_{j=1}^{n-2} (B_{n,j} \tilde{y}_j + B_{n,j+1} \tilde{y}_{j+1}) + \frac{\alpha \tilde{y}_{n-1}}{(1-\alpha) h^\alpha} + \frac{y(a)}{(t_n - a)^\alpha} + L^1_{n-2} \tilde{f}_n, \tag{3.12}$$

where

$$A_{n,j} = \int_{t_j}^{t_{j+1}} \frac{1}{(t_n - s)^{\alpha+1}} ds, \quad B_{n,j} = \frac{1}{h} \int_{t_j}^{t_{j+1}} \frac{t_{j+1} - s}{(t_n - s)^{\alpha+1}} ds, \quad B_{n,j+1} = \frac{1}{h} \int_{t_j}^{t_{j+1}} \frac{s - t_j}{(t_n - s)^{\alpha+1}} ds,$$

which can be evaluated explicitly. Moreover, the first two terms of the left hand side in (3.12) can be simplified by

$$\frac{1}{(t_n - a)^\alpha} + \alpha \sum_{j=1}^{n-2} \int_{t_j}^{t_{j+1}} \frac{1}{(t_n - s)^{\alpha+1}} ds = h^{-\alpha}. \tag{3.13}$$

Hence, the approximated solution $\tilde{y}_n$ can be evaluated by solving the following explicit form

$$h^{-\alpha} \left[1 + \frac{\alpha}{(1-\alpha)} \right] \tilde{y}_n = \alpha \sum_{j=1}^{n-2} (B_{n,j} \tilde{y}_j + B_{n,j+1} \tilde{y}_{j+1}) + \frac{\alpha h^{-\alpha}}{(1-\alpha)} \tilde{y}_{n-1} + \frac{y(a)}{(t_n - a)^\alpha} + L^1_{n-2} \tilde{f}_n. \tag{3.14}$$

3.1.3 Direct Method with Quadratic Interpolation

In this section, we further improve our scheme by employing a quadratic interpolation of $y(t)$ over each interval $I_j = [t_j, t_{j+1}]$. For each $I_j, j \geq 2$, we interpolate $y(t)$ by a quadratic Lagrange polynomial

$$u(t) \approx L_j^2 y(t) = \sum_{k=j-1}^{j+1} y_k Q_k^j(t), \tag{3.15}$$

where

$$Q_k^j(t) = \prod_{\substack{m=j-1 \\ m\neq k}}^{j+1} \frac{t-t_m}{t_k - t_m},$$

On $I_1 = [t_1, t_2]$, $yt)$ is interpolated by using the grid $t_{3/2}$

$$y(t) \approx L_1^2 y(t) = y_1 Q_1^1(t) + y_{\frac{3}{2}} Q_{\frac{3}{2}}^1(t) + y_2 Q_2^1(t), \tag{3.16}$$

where

$$Q_1^1(t) = \frac{(t-t_{\frac{3}{2}})(t-t_2)}{(t_1 - t_{\frac{3}{2}})(t_1 - t_2)}, Q_{3/2}^1(t) = \frac{(t-t_1)(t-t_2)}{(t_{\frac{3}{2}} - t_1)(t_{\frac{3}{2}} - t_2)}, Q_2^1(t) = \frac{(t-t_1)(t-t_{\frac{3}{2}})}{(t_2 - t_1)(t_2 - t_{\frac{3}{2}})}.$$

Now, the value of $f(t_n, y(t_n))$ can be approximated by using the quadratic interpolation of $f(t, y(t))$ with grid points t_{n-3}, t_{n-2} and t_{n-1}

$$f(t_n, y(t_n)) \approx L_{n-2}^2 f(t_n, y(t_n)) = f(t_{n-3}, y(t_{n-2})) - 3f(t_{n-2}, y(t_{n-2})) + 3f(t_{n-1}, y(t_{n-1})). \tag{3.17}$$

Substituting (3.15), (3.16) and (3.17) into Eq. (3.7), taking the uniform property of the grid, we obtain that

$$\frac{\tilde{y}_n - y(a)}{(t_n - a)^\alpha} + \alpha \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{\tilde{y}_n - L_j^2 \tilde{y}(s)}{(t_n - s)^{1+\alpha}} ds = L_{n-2}^2 \tilde{f}_n. \tag{3.18}$$

Lemma 3.1.1.

$$\int_{t_{n-1}}^{t_n} \frac{\tilde{y}_n - L_{n-1}^2 \tilde{y}(s)}{(t_n - s)^{1+\alpha}} = h^{-\alpha}[D_1 \tilde{y}_n - D_2 \tilde{y}_{n-1} + D_3 \tilde{y}_{n-2}], \quad n \geq 3.$$

where

$$D_1 = \frac{(5-2\alpha)}{2(1-\alpha)(2-\alpha)}, \quad D_2 = \frac{(3-2\alpha)}{(1-\alpha)(2-\alpha)}, \quad D_3 = \frac{1}{2(1-\alpha)(2-\alpha)}.$$

Proof.

$$\begin{aligned}
&\tilde{y}_n - L_j^2\tilde{y}(t) = \frac{(t_n-t_{n-1})(t_n-t_{n-2})}{2h^2} - L_j^2\tilde{y}(t)\\
=&\frac{1}{2h^2}\left\{((t_n-t_{n-1})(t_n-t_{n-2})-(t-t_{n-1})(t-t_{n-2}))\tilde{y}_n + 2(t-t_n)(t-t_{n-2})\tilde{y}_{n-1} - (t-t_n)(t-t_{n-1})\tilde{y}_{n-2}\right\}\\
=&\frac{1}{2h^2}\left\{(t_n-t)(t+t_n-t_{n-1}-t_{n-2})\tilde{y}_n + 2(t-t_n)(t-t_{n-2})\tilde{y}_{n-1} - (t-t_n)(t-t_{n-1})\tilde{y}_{n-2}\right\}.
\end{aligned}$$

Then we have

$$\begin{aligned}
&\int_{t_{n-1}}^{t_n} \frac{\tilde{y}_n - L_{n-1}^2\tilde{y}(s)}{(t_n-s)^{1+\alpha}}\\
=&\frac{1}{2h^2}\int_{t_{n-1}}^{t_n}\frac{1}{(t_n-s)^{\alpha}}\left\{(s+t_n-t_{n-1}-t_{n-2})\tilde{y}_n - 2(s-t_{n-2})\tilde{y}_{n-1} + (s-t_{n-1})\tilde{y}_{n-2}\right\}. \qquad (3.19)
\end{aligned}$$

Using a simple change of variable, we can complete the proof. □

Lemma 3.1.1 gives a explicit form for (3.18) as follows

$$h^{-\alpha}\left[1+\alpha D_1\right]\tilde{y}_n = \alpha\tilde{y}_n^* + \alpha h^{-\alpha}[D_2\tilde{y}_{n-1} - D_3\tilde{y}_{n-2}] + \frac{y(a)}{(t_n-a)^{\alpha}} + L_{n-2}^2\tilde{f}_n, \quad n \geq 3. \qquad (3.20)$$

Here, the lag term is approximated as follows,

$$\tilde{y}_n^* = \left[C_n^{1,1}\tilde{y}_1 + C_n^{2,1}\tilde{y}_{3/2} + C_n^{3,1}\tilde{y}_2\right] + \sum_{j=2}^{n-2}\left[C_n^{1,j}\tilde{y}_{j-1} + C_n^{2,j}\tilde{y}_j + C_n^{3,j}\tilde{y}_{j+1}\right], \qquad (3.21)$$

where,

$$\begin{aligned}
C_n^{1,1} &= \int_{t_1}^{t_2}\frac{Q_1^1(\tau)}{(t_n-\tau)^{\alpha+1}}d\tau,\\
C_n^{2,1} &= \int_{t_1}^{t_2}\frac{Q_{\frac{3}{2}}^1(\tau)}{(t_n-\tau)^{\alpha+1}}d\tau,\\
C_n^{3,1} &= \int_{t_1}^{t_2}\frac{Q_2^1(\tau)}{(t_n-\tau)^{\alpha+1}}d\tau,
\end{aligned}$$

and, for $2 \le j \le n-2$,

$$C_n^{i,j} = \int_{t_j}^{t_{j+1}} \frac{Q_{i+j-2}^j(\tau)}{(t_n-\tau)^{\alpha+1}} d\tau, \quad i = 1,2,3.$$

3.1.4 Error Analysis

From here, we denote C a generic constant which is independent of all grid parameters and may change case by case. We need the following lemmas.

Lemma 3.1.2. *(*Interpolation Errors*) Let $f \in \mathcal{C}^{n+1}[a,b]$ and $p_n \in \mathbb{P}_n[a,b]$ interpolate the function f at the grid Φ_n in (3.5) with $a = t_1$ and $b = t_n$, then there exists $\xi \in (a,b)$ such that, for any $t \in [a,b]$,*

$$f(t) - p_n(t) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (t - t_j).$$

Let $e_n = y(t_n) - y_n$ be an error at time step t_n. Then, subtracting (3.10) from (3.7) we have

$$\frac{e_n}{(t_n-a)^\alpha} + \alpha \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{e_n - (y(s) - L_j^1 \tilde{y}(s))}{(t_n-s)^{\alpha+1}} ds = f(t_n, y(t_n)) - L_{n-2}^1 \tilde{f}_n, \tag{3.22}$$

Theorem 3.1.3. *(Truncation Error with Linear Iinterpolation) Let τ_n be a truncation error at t_n. Suppose that $y(\cdot)$ and $f(\cdot, y(\cdot)) \in \mathcal{C}^2[a,T]$, and furthermore is Lipschitz continuous in the second argument, i.e.,*

$$|f(t,u_1) - f(t,u_2)| \le L|u_1 - u_2|, \quad \forall u_1, u_2 \in \mathbb{R}, \tag{3.23}$$

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

$$|\tau_n| \le Ch^{2-\alpha}, \quad n \ge 3. \tag{3.24}$$

Proof. From the linear interpolation of $y(t)$ and $f(t, y(t))$, we obtain

$$\tau_n = \alpha \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{-(y(s) - L_j^1 y(s))}{(t_n-s)^{1+\alpha}} ds - \left[f(t_n, y(t_n)) - L_{n-2}^1 f_n \right].$$

By Lemma 3.1.2, we have the following; for some $\xi_j \in (t_j, t_{j+1})$ and $\eta_{n-2} \in (t_{n-2}, t_{n-1})$,

$$|\tau_n| \leq \alpha \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \left| \frac{y''(\xi_j)}{2} \frac{(s-t_j)(s-t_{j+1})}{(t_n-s)^{1+\alpha}} \right| ds + \left| \frac{f''(\eta_{n-2})}{2} (t_n - t_{n-2})(t_n - t_{n-1}) \right|.$$

Let $M = \max_{1 \leq j \leq n-1} |y''(\xi_j)|$ and $\mathcal{M} = |f''(\eta_{n-2})|$. Then, we have

$$\begin{aligned}
|\tau_n| &\leq \frac{\alpha M}{2} h^2 \sum_{j=1}^{n-2} \int_{t_j}^{t_{j+1}} \frac{1}{(t_n-s)^{1+\alpha}} ds + \frac{\alpha M}{2} h \int_{t_{n-1}}^{t_n} \frac{1}{(t_n-s)^{\alpha}} ds + \frac{\mathcal{M}}{2} h^2 \\
&= \frac{\alpha M}{2} h^2 \left[\frac{1}{\alpha} (t_n - s) \right]_{t_1}^{t_{n-1}} + \frac{\alpha M}{2} h \left[\frac{1}{\alpha - 1} (t_n - s)^{-\alpha+1} \right]_{t_{n-1}}^{t_n} + \frac{\mathcal{M}}{2} h^2 \\
&= \frac{M}{2} (h^{-\alpha} - (nh)^{-\alpha}) + \frac{\alpha M}{2(1-\alpha)} h^{2-\alpha} + \frac{\mathcal{M}}{2} h^2 \\
&= \frac{M}{2} (1 - n^{1-\alpha}) h^{2-\alpha} + \frac{\alpha M}{2(1-\alpha)} h^{2-\alpha} + \frac{\mathcal{M}}{2} h^2 = O(h^{2-\alpha}).
\end{aligned}$$

□

By a similar procedure in the theorem (3.1.3), we can obtain the analysis for the accuracy order of the new scheme using a quadratic interpolation.

$$\frac{e_n}{(t_n - a)^{\alpha}} + \alpha \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{e_n - (y(s) - L_j^2 y(s))}{(t_n - s)^{1+\alpha}} ds = f(t_n, y(t_n)) - L_{n-2}^2 f_n, \tag{3.25}$$

Theorem 3.1.4. *(Truncation Error with Quadratic Interpolation) Let τ_n be a truncation error at t_n. Suppose that $y(\cdot)$ and $f(\cdot, y(\cdot)) \in \mathcal{C}^3[a, T]$, and furthermore is Lipschitz continuous in the second argument, i.e.,*

$$|f(t, u_1) - f(t, u_2)| \leq L |u_1 - u_2|, \quad \forall u_1, u_2 \in \mathbb{R}, \tag{3.26}$$

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

$$|\tau_n| \leq C h^{3-\alpha}, \quad n \geq 4. \tag{3.27}$$

Proof. By following similar procedures in the Theorem 3.1.3, we obtain the truncation error for the quadratic interpolation. □

3.1.5 Numerical Examples

In this section, we illustrate the accuracy and efficiency of our new methods, both with the linear (DL) and quadratic interpolation (DQ). Also, in the sense of the PECE method [3], we use our numerical solutions $\tilde{y}^P$ as a predictor and update a corrector $\tilde{y}$ with both the linear (DL-PC) and quadratic interpolation (DQ-PC).

- *With Linear Interpolation:*

$$h^{-\alpha}\left[1+\frac{\alpha}{(1-\alpha)}\right]\tilde{y}_n^P = \alpha\sum_{j=1}^{n-2}(B_{n,j}\tilde{y}_j + B_{n,j+1}\tilde{y}_{j+1}) + \frac{\alpha h^{-\alpha}}{(1-\alpha)}\tilde{y}_{n-1} + \frac{y(a)}{(t_n-a)^\alpha} + L^1_{n-2}\tilde{f}_n, \quad (3.28)$$

$$h^{-\alpha}\left[1+\frac{\alpha}{(1-\alpha)}\right]\tilde{y}_n = \alpha\sum_{j=1}^{n-2}(B_{n,j}\tilde{y}_j + B_{n,j+1}\tilde{y}_{j+1}) + \frac{\alpha\tilde{y}_{n-1}}{(1-\alpha)h^\alpha} + \frac{y_1}{(t_n-t_1)^\alpha} + f(t_n,\tilde{y}_n^P). \quad (3.29)$$

- *With Quadratic Interpolation:*

$$h^{-\alpha}\left[1+\alpha D_1\right]\tilde{y}_n^P = \alpha\tilde{y}_n^* + \alpha h^{-\alpha}[D_2\tilde{y}_{n-1} - D_3\tilde{y}_{n-2}] + \frac{y(a)}{(t_n-a)^\alpha} + L^2_{n-2}\tilde{f}_n, \quad (3.30)$$

$$h^{-\alpha}\left[1+\alpha D_1\right]\tilde{y}_n = \alpha\tilde{y}_n^* + \alpha h^{-\alpha}[D_2\tilde{y}_{n-1} - D_3\tilde{y}_{n-2}] + \frac{y(a)}{(t_n-a)^\alpha} + f(t_n,\tilde{y}_n^P). \quad (3.31)$$

For all below tests, we measure the error $(y_n - y(t_n))$ by using the following error estimate:

$$E_{Max} = \max_j |y_j - \tilde{y}_j|.$$

Example 3.1.5.

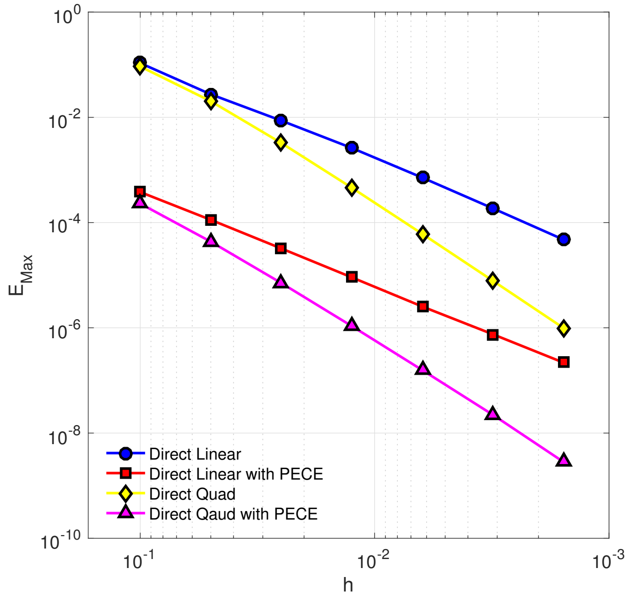
$$\begin{cases} D_0^\alpha y(t) = \dfrac{\Gamma(9)}{\Gamma(9-\alpha)}t^{8-\alpha} - 3\dfrac{\Gamma(5+\frac{\alpha}{2})}{\Gamma(5-\frac{\alpha}{2})}t^{4-\frac{\alpha}{2}}, \\ y(0) = 0, \end{cases}$$

whose exact solution is

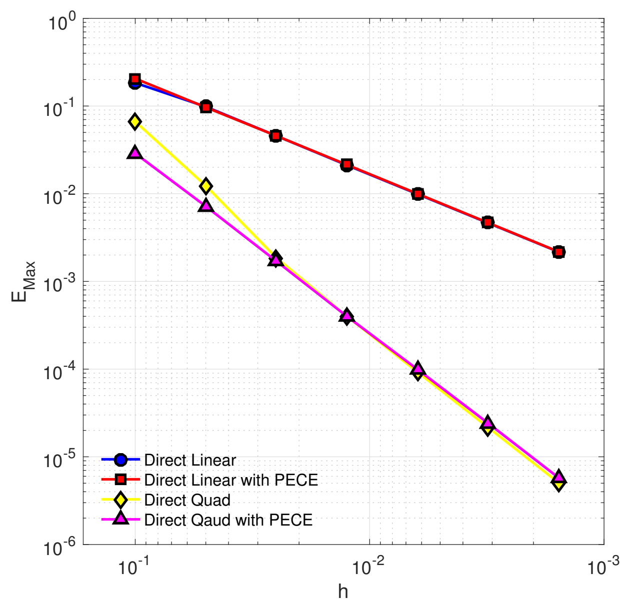
$$y(t) = t^8 - 3t^{4+\frac{\alpha}{2}}.$$

Table 3-1: Numerical comparisons of errors and orders by linear and quadratic interpolation in Example 3.1.5.

	$\alpha = 0.01$							
	DL		DL-PC		DQ		DQ-PC	
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	1.0749E-01	-	3.8052E-04	-	9.2576E-02	-	2.2939E-04	-
20	2.7388E-02	1.9725	1.1268E-04	1.7558	2.0240E-02	2.1934	4.3337E-05	2.4041
40	8.6430E-03	1.6640	3.2169E-05	1.8084	3.2485E-03	2.6394	7.0813E-06	2.6135
80	2.6081E-03	1.7285	9.0736E-06	1.8259	4.5743E-04	2.8281	1.0744E-06	2.7205
160	7.1169E-04	1.8737	2.5293E-06	1.8429	6.0602E-05	2.9161	1.5636E-07	2.7805
320	1.8558E-04	1.9392	7.5169E-07	1.7505	7.7952E-06	2.9587	2.2113E-08	2.8219
640	4.7354E-05	1.9705	2.1770E-07	1.7878	9.8848E-07	2.9793	2.8170E-09	2.9727
	$\alpha = 0.1$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	1.0454E-01	-	4.5666E-03	-	8.8051E-02	-	2.4067E-03	-
20	2.6512E-02	1.9794	1.3849E-03	1.7213	1.9426E-02	2.1804	4.8053E-04	2.3244
40	7.9392E-03	1.7396	4.1086E-04	1.7531	3.1217E-03	2.6376	8.2187E-05	2.5477
80	2.4003E-03	1.7258	1.2004E-04	1.7751	4.3879E-04	2.8307	1.2992E-05	2.6613
160	6.5298E-04	1.8781	3.4692E-05	1.7908	5.7935E-05	2.9210	1.9659E-06	2.7244
320	1.6930E-04	1.9475	9.9317E-06	1.8045	7.4192E-06	2.9651	2.8991E-07	2.7615
640	4.2875E-05	1.9813	2.8211E-06	1.8158	9.3416E-07	2.9895	4.3554E-08	2.7347
	$\alpha = 0.5$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	6.5888E-02	-	4.8710E-02	-	6.8290E-02	-	1.2247E-02	-
20	2.3036E-02	1.5161	1.8642E-02	1.3856	1.4812E-02	2.2049	3.3412E-03	1.8739
40	8.9031E-03	1.3715	6.8175E-03	1.4513	2.2180E-03	2.7394	7.3919E-04	2.1763
80	2.8217E-03	1.6577	2.4662E-03	1.4669	2.7183E-04	3.0285	1.4761E-04	2.3242
160	8.3546E-04	1.7559	8.8523E-04	1.4782	2.7807E-05	3.2892	2.7964E-05	2.4001
320	2.5803E-04	1.6951	3.1623E-04	1.4851	2.0375E-06	3.7705	5.1499E-06	2.4409
640	9.3744E-05	1.4607	1.1261E-04	1.4896	2.0087E-07	3.3425	9.3351E-07	2.4638
	$\alpha = 0.9$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	1.8389E-01	-	2.0583E-01	-	6.6342E-02	-	2.8864E-02	-
20	9.8235E-02	0.9045	9.6674E-02	1.0903	1.2306E-02	2.4306	7.1881E-03	2.0056
40	4.5732E-02	1.1030	4.5708E-02	1.0807	1.8537E-03	2.7309	1.7031E-03	2.0774
80	2.1169E-02	1.1113	2.1397E-02	1.0950	3.9965E-04	2.2136	3.9948E-04	2.0920
160	9.9166E-03	1.0941	1.0000E-02	1.0973	9.2479E-05	2.1115	9.9293E-05	2.0084
320	4.6454E-03	1.0940	4.6699E-03	1.0986	2.1625E-05	2.0964	2.4194E-05	2.0370
640	2.1731E-03	1.0960	2.1797E-03	1.0993	5.0593E-06	2.0957	5.7686E-06	2.0684



(a)



(b)

Figure 3-1: Maximum errors of Example 3.1.5 obtained by linear and quadratic interpolation with various h. (a) We set $\alpha = 0.01$. (b) We set $\alpha = 0.9$.

Example 3.1.6.

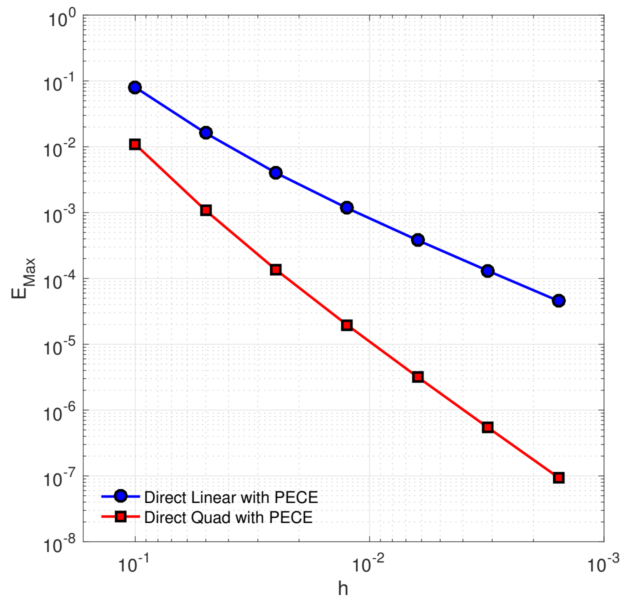
$$\begin{cases} D_0^\alpha y(t) = \dfrac{\Gamma(5)}{\Gamma(5-\alpha)} t^{4-\alpha} + \sin(t^4) + t^8 - \sin(y) - y^2, \\ y(0) = 0, \end{cases}$$

whose exact solution is

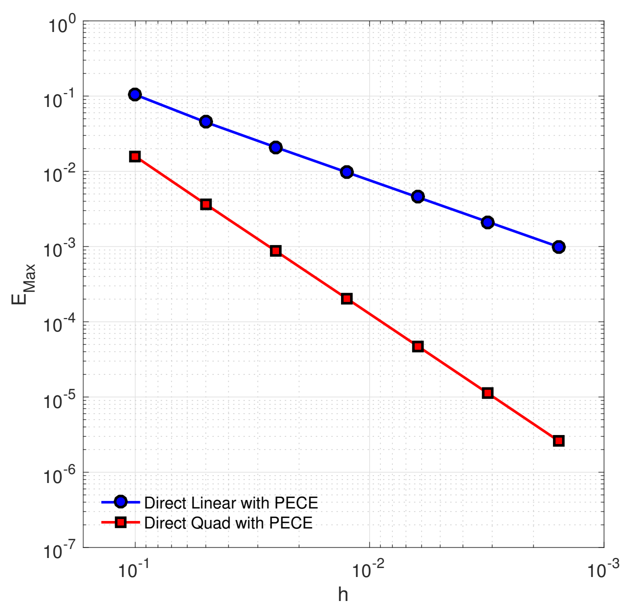
$$y(t) = t^4.$$

Table 3-2: Numerical comparisons of errors and orders by linear and quadratic interpolation in Example 3.1.6.

	$\alpha = 0.1$							
	DL		DL-PC		DQ		DQ-PC	
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	8.8913E-02	-	2.9958E+00	-	1.0741E+01	-	2.7049E+00	-
20	4.9812E+00	-	2.9660E+00	-	4.4773E+69	-	2.4529E+01	-
40	Inf	-	2.6055E+08	-	Inf	-	3.3461E+25	-
	$\alpha = 0.5$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	2.3398E-02	-	8.0027E-02	-	5.7667E-03	-	1.0903E-02	-
20	4.3325E-03	2.4331	1.6095E-02	2.3139	6.4081E-04	3.1698	1.0739E-03	3.3439
40	5.5369E-04	2.9680	3.9843E-03	2.0142	6.2650E-05	3.3545	1.3395E-04	3.0031
80	1.9840E-04	1.4807	1.1747E-03	1.7621	5.1008E-06	3.6185	1.9725E-05	2.7635
160	1.5414E-04	0.3642	3.8198E-04	1.6207	7.6452E-07	2.7381	3.1917E-06	2.6277
320	7.5947E-05	1.0212	1.3029E-04	1.5518	2.4976E-07	1.6140	5.4176E-07	2.5586
640	3.2268E-05	1.2349	4.5417E-05	1.5204	5.8548E-08	2.0928	9.4089E-08	2.5256
	$\alpha = 0.9$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	3.5795E-02	-	1.0444E-01	-	4.1974E-03	-	1.5799E-02	-
20	2.8239E-02	0.3421	4.4972E-02	1.2155	2.2256E-03	0.9153	3.6497E-03	2.1140
40	1.6560E-02	0.7699	2.0733E-02	1.1171	6.8612E-04	1.6977	8.6176E-04	2.0824
80	8.6506E-03	0.9369	9.6941E-03	1.0968	1.8137E-04	1.9195	2.0314E-04	2.0848
160	4.2775E-03	1.0160	4.5383E-03	1.0949	4.4986E-05	2.0114	4.7695E-05	2.0906
320	2.0573E-03	1.0560	2.1225E-03	1.0964	1.0828E-05	2.0547	1.1166E-05	2.0947
640	9.7538E-04	1.0767	9.9167E-04	1.0978	2.5676E-06	2.0763	2.6097E-06	2.0971



(a)



(b)

Figure 3-2: Maximum errors of Example 3.1.6 obtained by linear and quadratic interpolation with various h. (a) We set $\alpha = 0.5$. (b) We set $\alpha = 0.9$.

Example 3.1.7.

$$\begin{cases} D_t^{\alpha} y(x,t) = y_{xx} + e^x\left(\frac{\Gamma(4+\alpha)}{\Gamma(4)} t^3 - t^{3+\alpha}\right), \\ y(x,0) = 0, \\ y(0,t) = t^{3+\alpha}, y(1,t) = e t^{3+\alpha}, \end{cases}$$

whose exact solution is

$$y(t) = e^x t^{3+\alpha}.$$

We choose the final time as $T = 1$. Let $N(\tau)$ be a number of steps (a step-size) in the time and $M(h)$ be a number of steps (a step-size) in the space. For approximating the spatial derivative, we use the central difference method which has a second-order convergence. Then, we can expect a global convergence with $O(h^2)$ and $O(\tau^{3-\alpha})$ by controlling each step-size for linear and quadratic interpolation, respectively.

Table 3-3: Numerical comparisons of errors and orders with linear and quadratic interpolation in Example 3.1.7.

	$M = 12000$			
	$\alpha = 0.5$			
	DL		DQ	
N	E_{Max}	roc	E_{Max}	roc
10	1.0011E-02	-	9.6513E-04	-
20	3.7669E-03	1.4102	1.7872E-04	2.4330
40	1.3861E-03	1.4423	3.2479E-05	2.4602
80	5.0318E-04	1.4619	5.8541E-06	2.4720
160	1.8108E-04	1.4744	1.0424E-06	2.4895
320	6.4797E-05	1.4827	1.8560E-07	2.4897
640	2.3084E-05	1.4890	3.1310E-08	2.5675
	$\alpha = 0.9$			
N	E_{Max}	roc	E_{Max}	roc
10	6.4409E-02	-	8.5198E-03	-
20	3.1131E-02	1.0489	2.0738E-03	2.0386
40	1.4794E-02	1.0733	4.9394E-04	2.0699
80	6.9690E-03	1.0860	1.1639E-04	2.0854
160	3.2669E-03	1.0930	2.7284E-05	2.0928
320	1.5096E-03	1.1137	6.4172E-06	2.0880
640	6.7424E-04	1.1629	1.5641E-06	2.0366

Table 3-4: Numerical comparisons of spatial errors and orders by linear and quadratic interpolation in Example 3.1.7.

	$N = 10^3$			
	$\alpha = 0.5$			
	DL		DQ	
M	E_{Max}	roc	E_{Max}	roc
4	8.9880E-04	-	1.4515E-04	-
8	2.2693E-04	1.9857	3.6675E-05	1.9846
16	5.7628E-05	1.9774	9.1713E-06	1.9996

Table 3-5: Numerical comparisons of global errors and orders by linear and quadratic interplation with $N \approx h^{\frac{\alpha-2}{2}}$ and $M \approx \tau^{\frac{2}{\alpha-3}}$, respectively, in Example 3.1.7.

	$\alpha = 0.5$				
	DL			DQ	
N	E_{Max}	roc	M	E_{Max}	roc
10	3.3981E-03	-	10	1.0095E-03	-
20	9.3128E-04	1.8674	20	1.8703E-04	2.4324
40	2.3692E-04	1.9748	40	3.3927E-05	2.4627
80	6.0243E-05	1.9755	80	6.1019E-06	2.4751
160	1.5217E-05	1.9851	160	1.0905E-06	2.4842
320	3.8294E-06	1.9905	320	1.9427E-07	2.4889
640	9.6181E-07	1.9933	640	3.4448E-08	2.4955

Example 3.1.8.

$$\begin{cases} D_0^{\alpha(t)} y(t) = \dfrac{\Gamma(5)}{\Gamma(5-\alpha(t))} t^{4-\alpha(t)} + t^4 - y(t), \\ y(0) = 0, \ y'(0) = 0, \end{cases}$$

whose exact solution is

$$y(t) = t^4.$$

We test the example with several variable fractional orders,

$$\begin{aligned} \alpha_1(t) &= 0.7t + 0.3, \\ \alpha_2(t) &= \frac{t^2+1}{2}, \\ \alpha_3(t) &= e^{-t}, \\ \alpha_4(t) &= 0.1\sin(\frac{\pi}{2}t) + 0.9. \end{aligned}$$

Definition 3.1.1. *The variable order(VO) fractional ordinary differential equation (ODE) is defined by*

$$\begin{cases} \tilde{\mathbb{D}}_a^{\alpha(t)} y(t) = \tilde{f}(t, y(t)), \quad t \in [a, T], \ \tilde{f} \in C^m([a, T]), \ m \in \mathbb{Z}^+ \\ y^{(k)}(a) = y_k, \quad m-1 < \alpha(t) \leq m, \ 0 \leq k \leq m-1, \end{cases}$$

where the VO Caputo derivative is

$$\tilde{\mathbb{D}}_a^{\alpha(t)} y(t) = \frac{1}{\Gamma(m-\alpha(t))} \int_a^t (t-s)^{m-1-\alpha(t)} y^{(m)}(s) ds.$$

By definition 3.1.1, we can solve an IVP with a variable fractional order with our numerical methods.

Table 3-6: Numerical comparisons of errors and orders with Variable Fractional Order in Example 3.1.8.

	$\alpha_1(t)$			
	DQ		DQ-PC	
N	E_{Max}	roc	E_{Max}	roc
10	1.3552E-02	-	1.4757E-02	-
20	1.5241E-03	3.1525	3.5578E-03	2.0523
40	3.4621E-04	2.1382	8.6977E-04	2.0323
80	1.4479E-04	1.2577	2.1078E-04	2.0449
160	4.2231E-05	1.7776	5.0486E-05	2.0618
320	1.0970E-05	1.9447	1.2001E-05	2.0727
640	2.7139E-06	2.0152	2.8425E-06	2.0779
	$\alpha_2(t)$			
N	E_{Max}	roc	E_{Max}	roc
10	1.8407E-02	-	1.4135E-02	-
20	2.0330E-03	3.1786	3.2790E-03	2.1079
40	1.7189E-04	3.5640	7.8140E-04	2.0691
80	9.3326E-05	0.8812	1.8666E-04	2.0656
160	3.2368E-05	1.5277	4.4245E-05	2.0769
320	8.9226E-06	1.8591	1.0418E-05	2.0864
640	2.2577E-06	1.9826	2.4453E-06	2.0911
	$\alpha_3(t)$			
N	E_{Max}	roc	E_{Max}	roc
10	5.1526E-03	-	6.0493E-03	-
20	5.9180E-04	3.1221	8.1730E-04	2.8878
40	6.6548E-05	3.1526	1.2809E-04	2.6737
80	1.8965E-05	1.8111	2.6208E-05	2.2891
160	4.6379E-06	2.0318	5.4946E-06	2.2539
320	1.0638E-06	2.1242	1.1652E-06	2.2375
640	2.3740E-07	2.1638	2.4943E-07	2.2239
	$\alpha_4(t)$			
N	E_{Max}	roc	E_{Max}	roc
10	7.7424E-03	-	2.0578E-02	-
20	4.2517E-03	0.8647	5.9969E-03	1.7788
40	1.3844E-03	1.6188	1.6060E-03	1.9007
80	3.8589E-04	1.8430	4.1368E-04	1.9569
160	1.0105E-04	1.9331	1.0453E-04	1.9846
320	2.5727E-05	1.9738	2.6161E-05	1.9984
640	6.4622E-06	1.9932	6.5165E-06	2.0052

3.2 Enhanced Direct Method

3.2.1 Newton's Method for $\alpha \approx 0$

With small α, we face a problem that a numerical solution easily blows up because of a singularity under a small α in a nonlinear problem. Therefore, we suggest an improved scheme that use a numerical solution, which is obtained by our method, as the initial value of Newton Method.

Let C_1 be a coefficient of a numerical solution $\tilde{y}_n$ at time t_n, and C_2 be a memory term. Then, we can simplify the Direct Method with PECE as follows:

$$C_1 \tilde{y}_n = C_2 + f(t_n, \tilde{y}_n^P). \tag{3.32}$$

Define,

$$F(s) = s - \frac{C_2 + f(t_n, \tilde{y}_n^P)}{C_1}. \tag{3.33}$$

The general Newton's Method is

$$s_{k+1} = s_k - \frac{F(s_k)}{F'(s_k)}, \qquad k = 0, 1, 2, \dots\ . \tag{3.34}$$

Moreover, we can easily get a clear form of $\frac{dF}{ds}$.

3.2.2 Decomposition Method for $\alpha \approx 1$

With large α, our scheme has a low convergence rate compared to a smaller α. Therefore, we suggest a scheme that decomposes a large α into α_1 and α_2, such as $\alpha/2$, respectively. Suppose that $\alpha = \alpha_1 + \cdots + \alpha_k$. For simplicity, assume that $\alpha = \alpha_1 + \alpha_2$. We can decompose a FDE into a system of equations:

$$\begin{cases} \mathbb{D}^{\alpha} y(t) = \mathbb{D}^{\alpha_1}(\mathbb{D}^{\alpha_2} y(t)) = f(t, y(t)), \\ y(t_1) = y_0, \end{cases} \iff \begin{cases} \mathbb{D}^{\alpha_1} z(t) = f(t, y(t)),\ z(t_1) = 0, \\ \mathbb{D}^{\alpha_2} y(t) = z(t),\ y(t_1) = y_0. \end{cases} \tag{3.35}$$

Let $C_{1,1}$ and $C_{2,1}$ be coefficients of numerical solutions $\tilde{y}_n$ and $\tilde{z}_n$ at time t_n, and $C_{1,2}$ and $C_{2,2}$ be memory terms. Then, we have a predictor step as

$$\begin{cases} C_{2,1} \tilde{z}_n^P = C_{2,2} + L_{n-2} f_n, \\ C_{1,1} \tilde{y}_n^P = C_{1,2} + \tilde{z}_n^P, \end{cases} \tag{3.36}$$

and a corrector step as

$$\begin{cases} C_{2,1}\tilde{z}_n = C_{2,2} + f(t_n, \tilde{y}_n^P), \\ C_{1,1}\tilde{y}_n = C_{1,2} + \tilde{z}_n. \end{cases} \tag{3.37}$$

3.2.3 Numerical Examples

We illustrate the accuracy and efficiency of our improved methods for $\alpha \approx 0$. We have a stability problem in the previous method, but the problem is eliminated by combining with the Newton's Method.

Table 3-7: Numerical comparisons of errors and orders by quadratic interpolation and Newton's Method in Example 3.1.6.

	DQ-N					
	$\alpha = 0.01$		$\alpha = 0.05$		$\alpha = 0.1$	
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	1.2700E-05	-	7.0521E-05	-	1.6114E-04	-
20	1.8958E-06	2.7439	1.0694E-05	2.7213	2.4946E-05	2.6914
40	2.7368E-07	2.7922	1.5674E-06	2.7703	3.7324E-06	2.7406
80	3.8654E-08	2.8238	2.2480E-07	2.8016	5.4645E-07	2.7720
160	5.3790E-09	2.8452	3.1762E-08	2.8233	7.8828E-08	2.7933
320	7.4477E-10	2.8525	4.4503E-09	2.8353	1.1240E-08	2.8101

From now, we illustrate the accuracy and efficiency of our improved methods for $\alpha \approx 1$. Based on the error analysis and numerical results, the Direct Method has a low convergence rate with a large fractional order α. However, we transform an IVP into a system of IVPs to increase the convergence rate. Because of the direct updating of an auxiliary solution, the computational cost increases linearly. For simplicity, we assume $\alpha_1 = \alpha_2 = \alpha/2$. Then, we expect a global convergence as $O(h^{2-\alpha/2})$ and $O(h^{3-\alpha/2})$ for linear and quadratic interpolation, respectively.

Table 3-8: Numerical comparisons of errors and orders with quadratic interpolation and decomposition method in Example 3.1.5.

	$\alpha = 0.5$			
	D-DQ		D-DQ-PC	
N	E_{Max}	roc	E_{Max}	roc
10	6.8126E-02	-	1.1571E-02	-
20	1.5521E-02	2.1340	2.5973E-03	2.1555
40	2.4683E-03	2.6527	4.8785E-04	2.4125
80	3.3566E-04	2.8784	8.3754E-05	2.5422
160	4.2072E-05	2.9960	1.3697E-05	2.6123
320	5.0095E-06	3.0701	2.1779E-06	2.6529
640	5.7185E-07	3.1310	3.4032E-07	2.6780
	$\alpha = 0.7$			
N	E_{Max}	roc	E_{Max}	roc
10	5.8937E-02	-	1.5340E-02	-
20	1.3401E-02	2.1368	3.7609E-03	2.0282
40	2.0635E-03	2.6992	7.5773E-04	2.3113
80	2.6331E-04	2.9702	1.3836E-04	2.4533
160	2.9517E-05	3.1572	2.3984E-05	2.5283
320	2.8625E-06	3.3662	4.0381E-06	2.5703
640	2.0773E-07	3.7845	6.6840E-07	2.5949
	$\alpha = 0.9$			
N	E_{Max}	roc	E_{Max}	roc
10	5.1295E-02	-	1.7602E-02	-
20	1.1377E-02	2.1727	4.7897E-03	1.8777
40	1.6338E-03	2.7999	1.0450E-03	2.1965
80	1.7861E-04	3.1934	2.0426E-04	2.3550
160	1.3522E-05	3.7234	3.7723E-05	2.4369
320	1.5620E-06	3.1138	6.7565E-06	2.4811
640	3.5609E-07	2.1331	1.1939E-06	2.5006

Table 3-9: Numerical comparisons of errors and orders with quadratic interpolation and decomposition method in Example 3.1.6.

	$\alpha = 0.5$			
	D-DQ		D-DQ-PC	
N	E_{Max}	roc	E_{Max}	roc
10	6.5772E-03	-	9.2928E-03	-
20	8.4132E-04	2.9668	8.0372E-04	3.5313
40	1.0218E-04	3.0415	8.1622E-05	3.2997
80	1.1923E-05	3.0994	9.4832E-06	3.1055
160	1.3394E-06	3.1540	1.2151E-06	2.9643
320	1.4375E-07	3.2199	1.6623E-07	2.8698
640	1.4458E-08	3.3136	2.3692E-08	2.8107
	$\alpha = 0.7$			
N	E_{Max}	roc	E_{Max}	roc
10	6.2868E-03	-	5.8996E-03	-
20	7.3543E-04	3.0957	7.1293E-04	3.0488
40	7.7613E-05	3.2442	9.9899E-05	2.8352
80	7.2305E-06	3.4241	1.5136E-05	2.7225
160	5.8180E-07	3.6355	2.3777E-06	2.6704
320	7.4653E-08	2.9623	3.7927E-07	2.6483
640	2.2912E-08	1.7041	6.0770E-08	2.6418
	$\alpha = 0.9$			
N	E_{Max}	roc	E_{Max}	roc
10	5.4180E-03	-	6.7505E-03	-
20	5.1709E-04	3.3893	1.0730E-03	2.6533
40	4.1227E-05	3.6488	1.8135E-04	2.5648
80	9.4853E-06	2.1198	3.1213E-05	2.5386
160	2.6836E-06	1.8216	5.3894E-06	2.5340
320	5.9175E-07	2.1811	9.2924E-07	2.5360
640	1.1613E-07	2.3492	1.5827E-07	2.5537

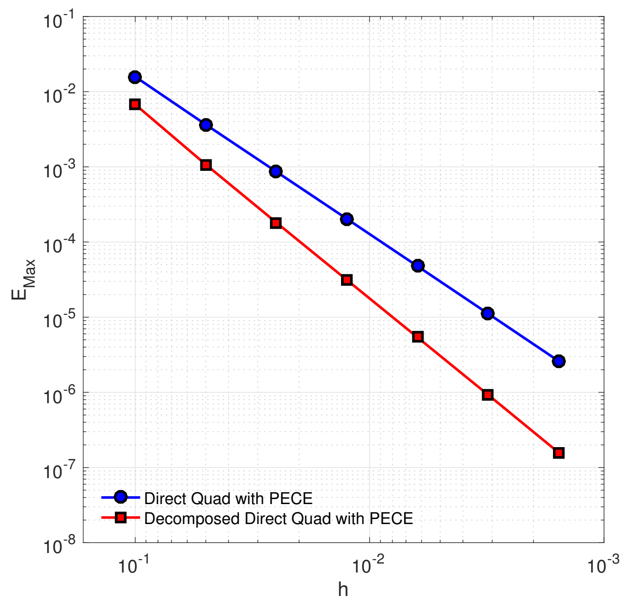


Figure 3-3: Maximum errors of Example 3.1.6 obtained by quadratic interpolation with various h. We set $\alpha = 0.9$.

Table 3-10: Numerical comparisons of errors and orders with Direct(M) and Decomposition(M/2) Methods in Example 3.1.7.

	$M = 12000$							
	$\alpha = 0.9$							
	DL		D-DL		DQ		D-DQ	
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	6.4409E-02	-	1.2475E-02	-	8.5198E-03	-	2.7754E-03	-
20	3.1131E-02	1.0489	4.4457E-03	1.4886	2.0738E-03	2.0386	4.9811E-04	2.4781
40	1.4794E-02	1.0733	1.5631E-03	1.5080	4.9394E-04	2.0699	8.7773E-05	2.5046
80	6.9690E-03	1.0860	5.4481E-04	1.5206	1.1639E-04	2.0854	1.5301E-05	2.5201
160	3.2669E-03	1.0930	1.8877E-04	1.5291	2.7284E-05	2.0928	2.6498E-06	2.5297
320	1.5096E-03	1.1137	6.5140E-05	1.5350	6.4172E-06	2.0880	4.5717E-07	2.5351
640	6.7424E-04	1.1629	2.2412E-05	1.5392	1.5641E-06	2.0366	7.6804E-08	2.5735
	$\alpha = 0.99$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	9.3585E-02	-	1.7218E-02	-	1.3271E-02	-	3.9521E-03	-
20	4.7921E-02	0.9656	6.3080E-03	1.4486	3.4444E-03	1.9460	7.2999E-04	2.4367
40	2.4166E-02	0.9877	2.2804E-03	1.4679	8.7373E-04	1.9790	1.3236E-04	2.4634
80	1.2093E-02	0.9988	8.1758E-04	1.4799	2.1925E-04	1.9946	2.3746E-05	2.4787
160	6.0267E-03	1.0048	2.9153E-04	1.4877	5.4706E-05	2.0028	4.2341E-06	2.4876
320	2.8503E-03	1.0802	1.0357E-04	1.4930	1.3609E-05	2.0071	7.5158E-07	2.4940
640	1.4163E-03	1.0091	3.6704E-05	1.4966	3.4035E-06	1.9995	1.3288E-07	2.4998

Table 3-11: Numerical comparisons of global errors and orders with Direct($N \approx h^{\frac{\alpha-2}{2}}$ or $M \approx \tau^{\frac{2}{\alpha-3}}$) and Decomposition($N \approx h^{\frac{\alpha/2-2}{2}}$ or $M \approx \tau^{\frac{\alpha/2-3}{2}}$) Methods in Example 3.1.7.

	$\alpha = 0.5$							
	DL		D-DL		DQ		D-DQ	
$N(M)$	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	3.3981E-03	-	3.5012E-04	-	1.0095E-03	-	5.2906E-04	-
20	9.3128E-04	1.8674	6.1499E-05	2.5092	1.8703E-04	2.4324	8.3939E-05	2.6560
40	2.3692E-04	1.9748	1.2177E-05	2.3364	3.3927E-05	2.4627	1.3108E-05	2.6789
80	6.0243E-05	1.9755	2.6475E-06	2.2015	6.1019E-06	2.4751	2.0244E-06	2.6949
160	1.5217E-05	1.9851	6.1504E-07	2.1059	1.0905E-06	2.4842	3.1019E-07	2.7062
320	3.8294E-06	1.9905	1.4814E-07	2.0537	1.9427E-07	2.4889	4.7335E-08	2.7122
640	9.6181E-07	1.9933	3.6522E-08	2.0201	3.4448E-08	2.4955	6.8873E-09	2.7809

Table 3-12: Numerical comparisons of errors and orders with variable fractional orders in Example 3.1.8.

	$\alpha_1(t)$							
	DQ		D-DQ		DQ-PC		D-DQ-PC	
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	1.3552E-02	-	2.5786E-02	-	1.4757E-02	-	5.0121E-03	-
20	1.5241E-03	3.1525	3.3538E-03	2.9427	3.5578E-03	2.0523	9.3424E-04	2.4236
40	3.4621E-04	2.1382	3.9400E-04	3.0895	8.6977E-04	2.0323	1.6842E-04	2.4718
80	1.4479E-04	1.2577	4.3035E-05	3.1946	2.1078E-04	2.0449	2.9821E-05	2.4977
160	4.2231E-05	1.7776	4.6344E-06	3.2150	5.0486E-05	2.0618	5.2273E-06	2.5122
320	1.0970E-05	1.9447	4.9610E-07	3.2237	1.2001E-05	2.0727	9.1099E-07	2.5205
640	2.7139E-06	2.0152	5.2906E-08	3.2291	2.8425E-06	2.0779	1.5824E-07	2.5253
	$\alpha_2(t)$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	1.8407E-02	-	3.1662E-02	-	1.4135E-02	-	4.8345E-03	-
20	2.0330E-03	3.1786	4.4320E-03	2.8367	3.2790E-03	2.1079	8.9632E-04	2.4313
40	1.7189E-04	3.5640	5.5536E-04	2.9965	7.8140E-04	2.0691	1.6086E-04	2.4782
80	9.3326E-05	0.8812	6.4353E-05	3.1093	1.8666E-04	2.0656	2.8360E-05	2.5038
160	3.2368E-05	1.5277	6.8492E-06	3.2320	4.4245E-05	2.0769	4.9503E-06	2.5183
320	8.9226E-06	1.8591	6.9430E-07	3.3023	1.0418E-05	2.0864	8.5911E-07	2.5266
640	2.2577E-06	1.9826	6.9164E-08	3.3275	2.4453E-06	2.0911	1.4841E-07	2.5332
	$\alpha_3(t)$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	5.1526E-03	-	1.5508E-02	-	6.0493E-03	-	1.3326E-03	-
20	5.9180E-04	3.1221	2.0941E-03	2.8886	8.1730E-04	2.8878	2.2151E-04	2.5887
40	6.6548E-05	3.1526	2.6779E-04	2.9672	1.2809E-04	2.6737	3.5859E-05	2.6270
80	1.8965E-05	1.8111	3.3259E-05	3.0093	2.6208E-05	2.2891	5.7274E-06	2.6464
160	4.6379E-06	2.0318	4.0559E-06	3.0357	5.4946E-06	2.2539	9.0903E-07	2.6555
320	1.0638E-06	2.1242	4.8760E-07	3.0562	1.1652E-06	2.2375	1.4397E-07	2.6586
640	2.3740E-07	2.1638	5.8007E-08	3.0714	2.4943E-07	2.2239	2.2823E-08	2.6572
	$\alpha_4(t)$							
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
10	7.7424E-03	-	1.4076E-02	-	2.0578E-02	-	6.0873E-03	-
20	4.2517E-03	0.8647	1.5285E-03	3.2030	5.9969E-03	1.7788	1.1754E-03	2.3726
40	1.3844E-03	1.6188	1.3811E-04	3.4683	1.6060E-03	1.9007	2.1813E-04	2.4299
80	3.8589E-04	1.8430	1.1535E-05	3.5818	4.1368E-04	1.9569	3.9650E-05	2.4598
160	1.0105E-04	1.9331	1.5413E-06	2.9038	1.0453E-04	1.9846	7.1236E-06	2.4766
320	2.5727E-05	1.9738	5.7067E-07	1.4334	2.6161E-05	1.9984	1.2711E-06	2.4865
640	6.4622E-06	1.9932	1.3821E-07	2.0458	6.5165E-06	2.0052	2.2593E-07	2.4921

Chapter 4

Numerical Method for Solving Fractional BVPs

In this section, we discuss a new numerical scheme for solving the following multi-term fractional differential equation with two point boundary values:

Let $0 < \alpha_1 < 1 < \alpha_2 < 2$ and $g(\cdot,\cdot)$, $h(\cdot,\cdot)$ be linear functions.

$$\begin{cases} {}_cD_{a,t}^{\alpha_2} y(t) = f\left(t, y(t), {}_c D_{a,t}^{\alpha_1} y(t)\right), \\ g(y(a), y'(a)) = \Gamma_a, h(y(T), y'(T)) = \Gamma_T. \end{cases} \tag{4.1}$$

4.1 High-Order Method

4.1.1 Description of High-Order Method

In conventional methods for solving fractional BVPs, many computational cost is required to solve a dense matrix and a multi-dimensional nonlinear solver. However, by changing a BVP to an IVP with the following theorem, we can explicitly solve a two-point BVP of fractional order.

Theorem 4.1.1. *Multi-term Fractional Differential Equations*

$$D^{\alpha_k} y(t) = f(t, y(t), D^{\alpha_1} y(t), \cdots, D^{\alpha_{k-1}} y(t)) \tag{4.2}$$

subject to the initial conditions

$$y^{(j)}(0) = y_0^{(j)},\ j = 0, 1, \cdots, \lceil \alpha_k \rceil - 1, \tag{4.3}$$

where $\alpha_k > \alpha_{k-1} > \cdots > \alpha_1 > 0$, $\alpha_k - \alpha_{k-1} \le 1$ *for all* $j = 2, 3, \cdots, k$ *and* $0 < \alpha_1 < 1$. *Then, given the equation* (4.2), *we may write* $\beta_1 := \alpha_1$, $\beta_j := \alpha_j - \alpha_{j-1}$, $y_1 := y$, *and* $y_j := D^{\alpha_{j-1}} y$. *Subject to the condition above, the multi-term equation* (4.2) *with initial conditions* (4.3) *is equivalent to the system*

$$\begin{aligned} D^{\beta_1} y_1(y) &= y_2(t), \\ D^{\beta_2} y_2(y) &= y_3(t), \\ \vdots \quad & \quad \vdots \\ D^{\beta_{k-1}} y_{k-1}(y) &= y_k(t), \\ D^{\beta_k} y_k(y) &= f(t, y_1(t), y_2(t), \cdots, y_k(t)) \end{aligned}$$

with the initial conditions

$$y_j(0) = \begin{cases} y_0^{(0)} & \text{if } j = 1, \\ y_0^{(l)} & \text{if } \alpha_{j-1} = l \in \mathbb{N}, \\ 0 & \text{else.} \end{cases}$$

Proof. □

For simplicity, we first consider the Dirichlet boundary conditions,

$$ {}_cD_{a,t}^{\alpha_2} y(t) = f\left(t, y(t), {}_cD_{a,t}^{\alpha_1} y(t)\right), \; y(a) = y_a, \; y(T) = y_T. \tag{4.4}$$

By Theorem 4.1.1, the two points BVP (4.4) can be rewritten in a system of equations with fractional orders as follows:

$$\begin{cases} {}_cD_{a,t}^{\alpha_1} y(t) & = w(t), \quad y(a) = y_a, \; y(T) = y_T, \\ {}_cD_{a,t}^{\lceil \alpha_1 \rceil - \alpha_1} w(t) & = z(t), \quad w(a) = 0, \\ {}_cD_{a,t}^{\alpha_2 - \lceil \alpha_1 \rceil} z(t) & = f(t, y(t), w(t)), \quad z(a) = y'(a). \end{cases} \tag{4.5}$$

Since we do not have any information about $y'(a)$, we apply the shooting method to this problem.

Then, we can change the BVP into the IVP as follows:

$$\begin{cases} {}_cD_{a,t}^{\alpha_1} y(t) & = w(t), \quad y(a) = y_a, \\ {}_cD_{a,t}^{\lceil \alpha_1 \rceil - \alpha_1} w(t) & = z(t), \quad w(a) = 0, \\ {}_cD_{a,t}^{\alpha_2 - \lceil \alpha_1 \rceil} z(t) & = f(t, y(t), w(t)), \quad z(a) = y'(a) = s. \end{cases} \tag{4.6}$$

By letting s be a variable and solving the IVP with numerical approaches for FDEs, we can get $y(s) := y(t,s)|_{t=T}$ when s differs. However, since Newton's Method is for an integer-order system, we have to adjust it for a fractional-order system to update an approximation.

4.1.2 Second-Order Scheme with Newton's Method

Define

$$F(s) = y(s) - y_T. \tag{4.7}$$

The general Newton's Method is

$$s_{k+1} = s_k - \frac{F(s_k)}{F'(s_k)}, \qquad k = 0, 1, 2, \ldots . \tag{4.8}$$

We need to get

$$F'(s) = \left. \frac{\partial y(t,s)}{\partial s} \right|_{t=T}. \tag{4.9}$$

So we apply the operator $\frac{\partial}{\partial s}$ to (4.6) that

$$\begin{cases} {}_cD_{a,t}^{\alpha_1} \frac{\partial y(t)}{\partial s} & = \frac{\partial w(t)}{\partial s}, & \frac{\partial y(t)}{\partial s}|_{t=a} = 0, \\ {}_cD_{a,t}^{\lceil \alpha_1 \rceil - \alpha_1} \frac{\partial w(t)}{\partial s} & = \frac{\partial z(t)}{\partial s}, & \frac{\partial w(t)}{\partial s}|_{t=a} = 0, \\ {}_cD_{a,t}^{\alpha_2 - \lceil \alpha_1 \rceil} \frac{\partial z(t)}{\partial s} & = \frac{\partial f(t,y(t),w(t))}{\partial s}, & \frac{\partial z(t)}{\partial s}|_{t=a} = 1. \end{cases} \tag{4.10}$$

Notice that t and s are independent, we can get

$$\frac{\partial f(t, y(t), w(t))}{\partial s} = \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial s}(t) + \frac{\partial f}{\partial w} \cdot \frac{\partial w}{\partial s}(t) \tag{4.11}$$

Define

$$\frac{\partial y}{\partial s} = \widehat{y}(t), \ \frac{\partial w}{\partial s} = \widehat{w}(t), \text{ and } \frac{\partial z}{\partial s} = \widehat{z}(t). \tag{4.12}$$

Then we can rewrite the auxiliary IVP (4.10) as

$$\begin{cases} {}_cD_{a,t}^{\alpha_1}\widehat{y}(t) = \widehat{w}(t), & \widehat{y}(a) = 0, \\ {}_cD_{a,t}^{\lceil\alpha_1\rceil-\alpha_1}\widehat{w}(t) = \widehat{z}(t), & \widehat{w}(a) = 0, \\ {}_cD_{a,t}^{\alpha_2-\lceil\alpha_1\rceil}\widehat{z}(t) = \frac{\partial f}{\partial y}\cdot\widehat{y}(t) + \frac{\partial f}{\partial w}\cdot\widehat{w}(t), & \widehat{z}(a) = 1. \end{cases} \tag{4.13}$$

We can get clear forms of $\frac{\partial f}{\partial y}$ and $\frac{\partial f}{\partial w}$. Hence, the solution of (4.13) can be solved by numerical approaches for fractional-order equations, such as PECE [3], Second-Order method [8], or Third-Order Method [8]. When updating an approximated solution $\widehat{y}(T)$, we can apply it to Newton's Method until $F(s)$ becomes small enough.

4.1.3 Third-Order Method with Halley's Method

By using the Newton's Method with high-order methods for solving FDEs, we have limitations to get an accurate approximated solution. The Newton's Method has a second-order convergence, and the initial condition should be sufficiently close to an exact value. So, even if we use a high-order method for solving FDEs which has a higher convergence than the second-order, we do not expect any improvement in terms of using a high-order method with the Newton's Method. It it clearly shown in the error analysis of our method with the Newton's Method. Thus, we apply a more efficient root-finding method, the Halley's Method, to our method for updating the approximated solution of the IVP.

The general Halley's Method is

$$s_{k+1} = s_k - \frac{2F(s_k)F'(s_k)}{2F'^2(s_k) - F(s_k)F''(s_k)}, \qquad k = 0, 1, 2, \dots . \tag{4.14}$$

To get $F''(s)$ distinctly, we apply operator $\frac{\partial^2}{\partial s^2}$ on (4.6) and define

$$\frac{\partial^2 y}{\partial s^2} = \widetilde{y}(t), \ \frac{\partial^2 w}{\partial s^2} = \widetilde{w}(t), \text{ and } \frac{\partial^2 z}{\partial s^2} = \widetilde{z}(t). \tag{4.15}$$

Then,

$$\begin{cases} {}_cD_{a,t}^{\alpha_1}\widetilde{y}(t) = \widetilde{w}(t), \\ {}_cD_{a,t}^{\lceil\alpha_1\rceil-\alpha_1}\widetilde{w}(t) = \widetilde{z}(t), \\ {}_cD_{a,t}^{\alpha_2-\lceil\alpha_1\rceil}\widetilde{z}(t) = \frac{\partial f}{\partial y}\cdot\widetilde{y}(t) + \frac{\partial f}{\partial w}\cdot\widetilde{w}(t) + \frac{\partial^2 f}{\partial y^2}\cdot\widehat{y}(t)^2 + \frac{\partial^2 f}{\partial w^2}\cdot\widehat{w}(t)^2 + \frac{2\partial^2 f}{\partial w\partial y}\cdot\widehat{w}(t)\,\widehat{y}(t), \\ \widetilde{y}(a) = 0, \ \widetilde{w}(a) = 0, \ \widetilde{z}(a) = 0. \end{cases} \tag{4.16}$$

We can either get clear forms of $\frac{\partial f}{\partial y}$, $\frac{\partial f}{\partial w}$, $\frac{\partial^2 f}{\partial y^2}$, $\frac{\partial^2 f}{\partial w^2}$, and $\frac{2\partial^2 f}{\partial w\partial y}$.

4.1.4 Error Analysis

Let y_h be a numerical solution which is obtained by an IVP solver and h be a step-size for an IVP solver. Suppose that we can get the value $y(s)$ at $t = T$ by using a shooting method with the Newton's Method in a fixed interval $[a, T]$ with an initial value s_0. Let s^* be a unique solution of $F(s, y(s)) = 0$ in the interval $[a, T]$. We may assume that $F(s, y(s))$ is sufficiently differentiable to s, and furthermore is Lipschitz continuous in the second argument, i.e., $y(s)$ is Lipschitz continuous.

Theorem 4.1.2. *Suppose that we have* $||y_h(s) - y(s)|| = O\left(h^{\delta_1}\right)$, *and there exists a* $\delta > 0$ *such that* $|s_0 - s^*| < \delta$, *for* $\delta < h$. *Then, the high-order method with the Newton's Method has a rate of convergence at least* δ_2*-th order, where* $\delta_2 = \min\{2, \delta_1\}$.

Proof. Let $e_n = s_n - s^*$. By expanding $F\left(s, y\left(s\right)\right)$ about s^*, we get

$$F\left(s, y\left(s\right)\right)|_{s=s_n} = \frac{\partial F}{\partial s}\left(s^*, y\left(s^*\right)\right)\left[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + ...\right]. \tag{4.17}$$

Moreover, by applying $\frac{\partial}{\partial s}$ to expanded $F\left(s, y\left(s\right)\right)$, we get

$$\frac{\partial F}{\partial s}\left(s, y\left(s\right)\right)|_{s=s_n} = \frac{\partial F}{\partial s}\left(s^*, y\left(s^*\right)\right)\left[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + ...\right], \tag{4.18}$$

where $c_k = \frac{1}{k!}\frac{\frac{\partial^k F}{\partial s^k}(s^*, y(s^*))}{\frac{\partial F}{\partial s}(s^*, y(s^*))}, k = 1, 2, 3, ...$.

From the Newton's Method, the recursive formula of the error e_n is

$$e_{n+1} = \left(e_n - \frac{F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}\right) - \left(\frac{F\left(s_n, y_h(s_n)\right)}{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right)} - \frac{F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}\right).$$

Then,

$$|e_{n+1}| \leqslant \left|e_n - \frac{F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}\right| + \left|\frac{F\left(s_n, y_h(s_n)\right)}{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right)} - \frac{F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}\right|.$$

Define

$$I_1 := \left|e_n - \frac{F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}\right|$$

and

$$I_2 := \left|\frac{F\left(s_n, y_h(s_n)\right)}{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right)} - \frac{F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}\right|.$$

By (4.17) and (4.18), we get

$$I_1 = \left| \frac{c_2 e_n^2 + 2c_3 e_n^3 + 3c_4 e_n^4 + ...}{1 + 2c_2 e_n + 3c_3 e_n^2 + ...} \right| = c_2 e_n^2 + O\left(e_n^3\right) \leqslant O\left(h^2\right),$$

$$\begin{aligned} I_2 &= \left| \frac{F\left(s_n, y_h(s_n)\right) \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right) - \frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right) F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right) \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)} \right|, \\ &\leqslant \left| \frac{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right) - \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right) \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)} \right| \cdot \left|F\left(s_n, y\left(s_n\right)\right)\right| \\ &+ \left| \frac{F\left(s_n, y_h(s_n)\right) - F\left(s_n, y\left(s_n\right)\right)}{\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right) \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)} \right| \cdot \left| \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right) \right|, \\ &\leqslant C\left|y_h(s_n) - y\left(s_n\right)\right| = O(h^{\delta_1}). \end{aligned}$$

Hence, it concludes as

$$|e_{n+1}| \leqslant I_1 + I_2 \leqslant O(h^{\delta_2}), \tag{4.19}$$

where $\delta_2 = \min\{2, \delta_1\}$. □

Theorem 4.1.3. *Suppose that we have $||y_h(s) - y(s)|| = O\left(h^{\delta_1}\right)$. Then, the high-order method with the Halley's Method has a rate of convergence at least δ_3-th order, where $\delta_3 = \min\{3, \delta_1\}$.*

Proof. By applying $\frac{\partial^2}{\partial s^2}$ to expanded $F\left(s, y\left(s\right)\right)$, we get

$$\frac{\partial^2 F}{\partial s^2}\left(s, y\left(s\right)\right)|_{s=s_n} = \frac{\partial F}{\partial s}\left(s^*, y\left(s^*\right)\right)\left[2c_2 + 6c_3 e_n + 12c_4 e_n^2 + 20c_5 e_n^3 + ...\right], \tag{4.20}$$

where $c_k = \frac{1}{k!} \frac{\frac{\partial^k F}{\partial s^k}(s^*, y(s^*))}{\frac{\partial F}{\partial s}(s^*, y(s^*))}, k = 2, 3, ...$.

From the Halley's Method, we can define

$$I_3 := \left| e_n - \frac{2F\left(s_n, y\left(s_n\right)\right) \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}{2\left(\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)\right)^2 - F\left(s_n, y\left(s_n\right)\right) \frac{\partial^2 F}{\partial s^2}\left(s_n, y\left(s_n\right)\right)} \right|,$$

$$I_4 := \left| \frac{2F\left(s_n, y\left(s_n\right)\right) \frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)}{2\left(\frac{\partial F}{\partial s}\left(s_n, y\left(s_n\right)\right)\right)^2 - F\left(s_n, y\left(s_n\right)\right) \frac{\partial^2 F}{\partial s^2}\left(s_n, y\left(s_n\right)\right)} - \frac{2F\left(s_n, y_h(s_n)\right) \frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right)}{2\left(\frac{\partial F}{\partial s}\left(s_n, y_h(s_n)\right)\right)^2 - F\left(s_n, y_h(s_n)\right) \frac{\partial^2 F}{\partial s^2}\left(s_n, y_h(s_n)\right)} \right|.$$

From (4.17), (4.18), and (4.20), we get

$$\begin{aligned}
I_3 &= \left| e_n - \frac{2e_n \left(\frac{\partial F}{\partial s}\right)^2 \left(s^*, y\left(s^*\right)\right) \left(1 + c_2 e_n + c_3 e_n^2 + c_4 e_n^3 + \ldots\right) \left(1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 \ldots\right)}{\left(\frac{\partial F}{\partial s}\right)^2 \left(s^*, y\left(s^*\right)\right) 2\left(1 + 2c_2 e_n + 3c_3 e_n^2 \ldots\right)^2 - \left(e_n + c_2 e_n^2 + \ldots\right)\left(2c_2 + 6c_3 e_n + 12c_4 e_n^2 + \ldots\right)} \right| \\
&= \left| e_n - \frac{2e_n + 6c_2 e_n^2 + \left(8c_3 + 4c_2^2\right) e_n^3 + O\left(e_n^4\right)}{2 + 6c_2 e_n + \left(6c_3 + 6c_2^2\right) e_n^2 + O\left(e_n^3\right)} \right| \\
&= \left| \frac{\left(2c_3 - 2c_2^2\right) e_n^3 + O\left(e_n^4\right)}{2 + 6c_2 e_n + \left(6c_3 + 6c_2^2\right) e_n^2 + O\left(e_n^3\right)} \right| \\
&= O\left(e_n^3\right) \leqslant O\left(h^3\right).
\end{aligned}$$

Define $f_1 := F\left(s_n, y_h(s_n)\right)$ and $f_2 := F\left(s_n, y\left(s_n\right)\right)$. Notice that the denominator of I_4 is bounded, we only consider its numerator and get

$$\begin{aligned}
I_4's \text{ numerator } &= \left| 4f_1 \cdot \frac{\partial f_1}{\partial s}\left(\frac{\partial f_2}{\partial s}\right)^2 - 2f_1 f_2 \frac{\partial f_1}{\partial s}\frac{\partial f_2}{\partial s} - 4f_2 \cdot \frac{\partial f_2}{\partial s}\left(\frac{\partial f_1}{\partial s}\right)^2 + 2\frac{\partial f_1}{\partial s}\frac{\partial f_2}{\partial s}\frac{\partial^2 f_1}{\partial s^2}\frac{\partial^2 f_2}{\partial s^2} \right| \\
&\leqslant 4\left| \frac{\partial f_1}{\partial s}\frac{\partial f_2}{\partial s}\left(f_1 \frac{\partial f_2}{\partial s} - f_2 \frac{\partial f_1}{\partial s}\right) \right| + 2\left| \frac{\partial f_1}{\partial s}\frac{\partial f_2}{\partial s}\left(\frac{\partial f_1}{\partial s}\frac{\partial^2 f_2}{\partial s^2} - \frac{\partial f_2}{\partial s}\frac{\partial^2 f_1}{\partial s^2}\right) \right| \\
&= 4\left| \frac{\partial f_1}{\partial s}\frac{\partial f_2}{\partial s}\left[\left(f_1 - f_2\right)\frac{\partial f_2}{\partial s} + f_2\left(\frac{\partial f_2}{\partial s} - \frac{\partial f_1}{\partial s}\right)\right] \right| \\
&\quad + 2\left| \frac{\partial f_1}{\partial s}\frac{\partial f_2}{\partial s}\left[\left(\frac{\partial f_1}{\partial s} - \frac{\partial f_2}{\partial s}\right)\frac{\partial^2 f_2}{\partial s^2} + \frac{\partial f_2}{\partial s}\left(\frac{\partial^2 f_2}{\partial s^2} - \frac{\partial^2 f_1}{\partial s^2}\right)\right] \right|
\end{aligned}$$

Moreover, f_2 and its derivatives are bounded in the interval $[a, b]$. Hence, we get

$$I_4 = C\left|y_h(s_n) - y(s_n)\right| = O\left(h^{\delta_1}\right).$$

Hence, it concludes as

$$\left|e_{n+1}\right| \leqslant I_3 + I_4 \leqslant O(h^{\delta_3}), \tag{4.21}$$

where $\delta_3 = \min\{3, \delta_1\}$. □

4.1.5 Numerical Examples

In this section, we illustrate the accuracy and efficiency of our new methods. We set $\alpha_1 = 0.5$ and $\alpha_2 = 1.5$.

Example 4.1.4. *Consider the two-point boundary value problem*

$$\begin{cases} {}_cD_{0,t}^{\alpha_2} y(t) = \frac{24}{\Gamma(5-\alpha_2)} t^{4-\alpha_2} - \frac{24}{\Gamma(5-\alpha_1)} t^{4-\alpha_1} + y^2 - t^8 + {}_c D_{0,t}^{\alpha_1} y(t), \\ y(0) = 0, \ y(1) = 1, \end{cases}$$

whose exact solution is

$$y(t) = t^4.$$

Table 4-1: Approximated errors in Example 4.1.4 by PECE and Newton's Method.

Error	s=0.2	s=0.4	s=0.6	s=0.8	s=1.0	s=1.2
m=1	1.7096E-01	5.5210E-01	9.5997E-01	1.3962	1.8624	2.3605
m=2	4.3260E-03	3.1847E-02	8.2967E-02	1.5647E-01	2.5151E-01	3.6758E-01
m=3	3.9053E-05	3.7634E-04	1.4013E-03	3.7464E-03	8.2173E-03	1.5722E-02
m=4	3.3544E-07	3.2455E-06	1.2232E-05	3.3600E-05	7.7445E-05	1.6018E-04
m=5	2.8799E-09	2.7865E-08	1.0503E-07	2.8858E-07	6.6551E-07	1.3778E-06
m=6	2.4725E-11	2.3923E-10	9.0170E-10	2.4776E-09	5.7136E-09	1.1829E-08
m=7	2.1227E-13	2.0537E-12	7.7411E-12	2.1271E-11	4.9053E-11	1.0156E-10
m=8	1.7764E-15	1.7986E-14	6.6613E-14	1.8274E-13	4.2100E-13	8.7197E-13
m=9	2.2204E-16	1.1102E-16	4.4409E-16	1.3323E-15	3.7748E-15	7.5495E-15
m=10	-	-	-	2.2204E-16	1.1102E-16	1.1102E-16

Table 4-2: Approximated errors in Example 4.1.4 by Second-Order Scheme and Newton's Method.

Error	s=0.2	s=0.4	s=0.6	s=0.8	s=1.0	s=1.2
m=1	4.0222E-01	8.4594E-01	1.3363	1.8789	2.4799	3.1462
m=2	1.9777E-02	7.4713E-02	1.6372E-01	2.8655E-01	4.4368E-01	6.3627E-01
m=3	1.3045E-04	9.9618E-04	3.8926E-03	1.0687E-02	2.3622E-02	4.5156E-02
m=4	5.3868E-07	4.2226E-06	1.7922E-05	5.8336E-05	1.6712E-04	4.3970E-04
m=5	2.2156E-09	1.7369E-08	7.3752E-08	2.4036E-07	6.9090E-07	1.8329E-06
m=6	9.1127E-12	7.1440E-11	3.0334E-10	9.8861E-10	2.8417E-09	7.5390E-09
m=7	3.7748E-14	2.9332E-13	1.2477E-12	4.0661E-12	1.1688E-11	3.1008E-11
m=8	2.2204E-16	1.3323E-15	5.1070E-15	1.6653E-14	4.8406E-14	1.2768E-13
m=9	-	1.1102E-16	2.2204E-16	2.2204E-16	3.3307E-16	4.4409E-16
m=10	-	-	-	-	-	-

Table 4-3: Approximated errors in Example 4.1.4 by Third-Order Scheme and Newton's Method.

Error	s=0.2	s=0.4	s=0.6	s=0.8	s=1.0	s=1.2
m=1	3.1529E-01	6.5121E-01	1.0109	1.3965	1.8104	2.2553
m=2	1.0199E-02	3.9874E-02	8.8836E-02	1.5715E-01	2.4510E-01	3.5323E-01
m=3	1.9808E-05	2.0339E-04	9.0724E-04	2.6940E-03	6.3089E-03	1.2658E-02
m=4	1.6793E-08	1.7645E-07	8.5589E-07	3.0599E-06	9.6181E-06	2.7918E-05
m=5	1.4201E-11	1.4921E-10	7.2385E-10	2.5885E-09	8.1433E-09	2.3693E-08
m=6	1.1546E-14	1.2612E-13	6.1240E-13	2.1889E-12	6.8863E-12	2.0035E-11
m=7	2.2204E-16	2.2204E-16	4.4409E-16	1.9984E-15	5.7732E-15	1.6875E-14
m=8	-	-	-	-	2.2204E-16	2.2204E-16
m=9	-	-	-	-	-	-
m=10	-	-	-	-	-	-

Table 4-4: Approximated errors in Example 4.1.4 by PECE and Halley's Method.

Error	s=0.2	s=0.4	s=0.6	s=0.8	s=1.0	s=1.2
m=1	1.2375E-01	4.9818E-01	8.9901E-01	1.3278	1.7864	2.2763
m=2	1.0327E-03	4.3187E-03	8.7785E-03	1.5272E-02	2.4588E-02	3.7447E-02
m=3	8.7129E-06	3.6421E-05	7.3984E-05	1.2860E-04	2.0680E-04	3.1447E-04
m=4	7.3523E-08	3.0733E-07	6.2430E-07	1.0852E-06	1.7450E-06	2.6535E-06
m=5	6.2042E-10	2.5934E-09	5.2681E-09	9.1570E-09	1.4725E-08	2.2391E-08
m=6	5.2356E-12	2.1884E-11	4.4454E-11	7.7270E-11	1.2425E-10	1.8895E-10
m=7	4.3965E-14	1.8430E-13	3.7526E-13	6.5215E-13	1.0483E-12	1.5945E-12
m=8	4.4409E-16	1.7764E-15	3.1086E-15	5.5511E-15	9.1038E-15	1.3323E-14
m=9	-	-	-	-	-	2.2204E-16
m=10	-	-	-	-	-	-

Table 4-5: Approximated errors in Example 4.1.4 by Second-Order Scheme and Halley's Method.

Error	s=0.2	s=0.4	s=0.6	s=0.8	s=1.0	s=1.2
m=1	3.7979E-01	8.1077E-01	1.2871	1.8140	2.3976	3.0447
m=2	1.9011E-03	4.9348E-03	1.0437E-02	1.9887E-02	3.4843E-02	5.6933E-02
m=3	9.4156E-06	2.4421E-05	5.1580E-05	9.8062E-05	1.7125E-04	2.7866E-04
m=4	4.6655E-08	1.2101E-07	2.5558E-07	4.8589E-07	8.4851E-07	1.3807E-06
m=5	2.3118E-10	5.9960E-10	1.2664E-09	2.4076E-09	4.2044E-09	6.8414E-09
m=6	1.1453E-12	2.9712E-12	6.2752E-12	1.1930E-11	2.0834E-11	3.3899E-11
m=7	5.5511E-15	1.4655E-14	3.1086E-14	5.9286E-14	1.0325E-13	1.6831E-13
m=8	-	2.2204E-16	2.2204E-16	2.2204E-16	4.4409E-16	4.4409E-16
m=9	-	-	-	-	-	-
m=10	-	-	-	-	-	-

Table 4-6: Approximated errors in Example 4.1.4 by Third-Order Scheme and Halley's Method.

Error	s=0.2	s=0.4	s=0.6	s=0.8	s=1.0	s=1.2
m=1	3.1144E-01	6.3934E-01	9.9026E-01	1.3663	1.7697	2.2032
m=2	5.1742E-04	1.2199E-03	2.3964E-03	4.3286E-03	7.2964E-03	1.1574E-02
m=3	8.8500E-07	2.0860E-06	4.0957E-06	7.3924E-06	1.2446E-05	1.9709E-05
m=4	1.5140E-09	3.5686E-09	7.0067E-09	1.2647E-08	2.1292E-08	3.3718E-08
m=5	2.5899E-12	6.1049E-12	1.1987E-11	2.1636E-11	3.6426E-11	5.7684E-11
m=6	4.6629E-15	1.0214E-14	2.0650E-14	3.6637E-14	6.2617E-14	9.9032E-14
m=7	-	-	-	2.2204E-16	1.1102E-16	1.1102E-16
m=8	-	-	-	-	-	-
m=9	-	-	-	-	-	-
m=10	-	-	-	-	-	-

Example 4.1.5. *Consider the two-point boundary value problem*

$$\begin{cases} {}_cD_{0,t}^{\alpha_2}y(t) + (2t+6)y'(t) + y(t) = f(t), \ \textit{for}\ 0 < t < 1, \\ y(0) - \frac{1}{\alpha_2 - 1}y'(0) = \gamma_0, \ y(1) + y'(1) = \gamma_1, \end{cases}$$

whose exact solution is

$$y(t) = t^4.$$

Table 4-7: Approximated errors in Example 4.1.5 by Newton's Method with $\alpha_1 = 0.5$ and $\alpha_2 = 1.5$.

	PECE		Second-Order		Third-Order	
N	E_{Max}	roc	E_{Max}	roc	E_{Max}	roc
64	4.2849E-02	-	2.9781E-04	-	8.2528E-06	-
128	6.9098E-03	2.6326	2.5989E-05	3.5184	1.5487E-08	9.0577
256	1.5955E-03	2.1146	4.0708E-06	2.6745	7.7000E-10	4.3300
512	4.2928E-04	1.8940	8.1725E-07	2.3165	3.8895E-11	4.3072
1024	1.2598E-04	1.7687	8.1725E-07	2.1469	1.7150E-12	4.5033
2048	3.9103E-05	1.6879	4.4007E-08	2.0681	3.9169E-13	2.1304

Chapter 5

Conclusion

In this paper, we introduced several numerical approaches for solving fractional differential equations. In the Direct Method, we proposed a new type of the Caputo differential operator without a derivative by using integration by parts. For a small fractional order, we proposed the enhanced Direct Method with the Newton's Method for a stable numerical solution. For a large fractional order, we proposed the enhanced Direct Method with a decomposition to increase a convergence rate. All numerical results support efficiency of the proposed methods for solving fractional IVPs.

In the High-Order Method, we change a BVP into an IVP instead of solving a matrix system which causes much computational time. Then, we can explicitly solve the equation with higher efficiency by using a high-order scheme [8]. For updating an approximation of IVP, we employ the nonlinear shooting methods that construct auxiliary IVPs. Even though we solve at least two systems of IVPs, the computational time is linearly increasing, whereas the conventional methods are exponentially increasing.

In the sense of computational mathematics, we can expect outstanding improvements of using our explicit methods. For example, when using the conventional PECE method, the computational cost to draw a bifurcation diagram of a fractional dynamical system when using the conventional PECE method is extremely high, approximately a month. However, when implementing our proposed methods, the expected computational time drastically reduces to approximately a few days.

References

[1] Michele Caputo, *Linear models of dissipation whose q is almost frequency independent?ii*, Geophysical Journal International **13** (1967), no. 5, 529–539. 5

[2] Kai Diethelm, *The analysis of fractional differential equations: An application-oriented exposition using differential operators of caputo type*, Springer, 2010. 1, 3, 5, 6

[3] Kai Diethelm, Neville J Ford, and Alan D Freed, *A predictor-corrector approach for the numerical solution of fractional differential equations*, Nonlinear Dynamics **29** (2002), no. 1-4, 3–22. 7, 10, 16, 35

[4] Kai Diethelm, Neville J Ford, Alan D Freed, and Yu Luchko, *Algorithms for the fractional calculus: a selection of numerical methods*, Computer methods in applied mechanics and engineering **194** (2005), no. 6-8, 743–773. 5

[5] Maolin Du, Zaihua Wang, and Haiyan Hu, *Measuring memory with the order of fractional derivative*, Scientific reports **3** (2013), 3431. 1

[6] Rudolf Gorenflo and Francesco Mainardi, *Essentials of fractional calculus*, 2000. 4, 5

[7] M Ishteva, *Properties and applications of the caputo fractional operator*, Department of Mathematics, University of Karlsruhe, Karlsruhe (2005). 5

[8] Thien Binh Nguyen and Bongsoo Jang, *A high-order predictor-corrector method for solving nonlinear differential equations of fractional order*, Fractional Calculus and Applied Analysis **20** (2017), no. 2, 447–476. v, 2, 35, 42

[9] Igor Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, vol. 198, Elsevier, 1998. 4

ACKNOWLEDGEMENTS

First of all, I am sincerely grateful to my advisor Dr. Bongsoo Jang for his guidance and care during my academic life at Ulsan National Institute of Science and Technology. Under his tutelage, I learned a great amount of knowledge in the field of computational mathematics. Moreover, he was not only my academic advisor but also a father figure. He has always guided and assisted me to continue in the right path both in academically and in personally. Without his dedicated support, I would not have come this far.

I would also like to thank many other professors, my laboratory fellows, and other members in the department of mathematical sciences. I especially appreciate Dr. Bong-suk Kwon and Dr. Chang-Yeol Jung for serving as thesis committee members and counselors for my future career in mathematics.

Last but not least, I would like to give my special thanks to my family, my mother Mikyung Lee, my father Jae Kyung Kim, and my sister Yerim Kim. Without their encouragement and faith in me, I would not have finished my degree. Even though I now start a new journey in the US, I will always remember all the support I have back in Korea.