SPECTRAL METHODS FOR FRACTIONAL LAPLANCIAN

A Major Qualifying Project Report:

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

In Partial Fulfillment of the Requirements for the

Degree of Bachelor of Science

By

Shuyang Sun

Zitai Huang

Advisor:

Prof. Zhongqiang Zhang

March 25, 2016

Abstract

This project is to numerically solve fractional Laplace equations and related equations. The major numerical methods we use are Jacobi-spectral methods. Our ultimate goal is to numerically solve a turbulence testbed problem. To achieve our goal, we first establish fractional Laplacian of Jacobi polynomials and choose proper test functions for our spectral methods. We start from linear steady and unsteady equations and then solve nonlinear unsteady equations. For unsteady equations, we have to take a different test function than that for steady equations to avoid singularity of the resulting linear systems. For each problem we consider, we perform numerical tests with MATLAB and present numerical results. We also discuss how to solve our ultimate goal problem based on our numerical results.

Acknowledgment

We would like to thank our professor, Professor Zhongqiang Zhang, for his patience, advice, and guidance during the project, especially for leading us to the world of fractional Laplacian.

Executive Summary

This project is to numerically solve fractional Laplace equations and related equations. The major numerical methods we use are finite difference in time and spectral methods in space. We consider the following situations to determine solutions of the following form:

$$u_N = \sum_{n=0}^{N} a_n h_n(x) \in (1-x^2)^{\alpha/2}\mathbb{P}_N, \quad h_n = (1-x^2)^{\alpha/2} P_n^{\alpha/2,\alpha/2}(x).$$

The fractional Laplacian is defined as $(-\Delta)^{\alpha/2}u(x) = c_{d,\alpha}\int_{\mathbb{R}^d} \frac{u(x)-u(y)}{|x-y|^{d+\alpha}}\,dy$.

Determine u_N for a Linear Equation We solve two linear fractional differential equations with fractional Laplacian: $(-\Delta)^{\alpha/2}u + \lambda u = g$ with $\lambda = 0$ and $\lambda \neq 0$. We give two numerical examples and discuss the convergence order of our method, when $g(x)$=$\sin(x)$ and $g(x)$=$|\sin(x)|$. We find that the solution when $\lambda = 0$ can be much smoother than that when $\lambda \neq 0$.

Determine u_N for a Nonlinear Equation We solve one nonlinear fractional differential equation with fractional Laplacian: $(-\Delta)^{\alpha/2}u = f(u) + g(x)$. Then, we give two examples and discuss the convergence order of our method, when $g(x)$=$\sin(x)$ and $g(x)$=$|\sin(x)|$ and $f(u) = u - u^3$. We apply fixed-point iteration methods solve the resulting nonlinear systems.

Determine u_N for Time Dependent Equations We solve two fractional differential equation in time dependent with fractional Laplacian: $(-\Delta)^{\alpha/2}u + \frac{\partial u}{\partial t} = f(u) + \sin(x)$. We consider two equations: the first equation is a linear equation where $f(u) = u$ and the second one is a nonlinear equation where $f(u) = u - u^3$.

Determine u_N for a turbulence testbed problem We discuss how to solve our ultimate goal problem which is a MMT model [2] based on our numerical results. The equation reads u_t=-$i((-\Delta)^{\alpha/2}u$+$\lambda\left|u^2\right|u)$+$\Delta u$.

In conclusion, we find that to have solvable linear or nonlinear algebraic system, it is required to choose a proper form of approximation basis as well as test functions. In numerical examples, we show that the convergence order is low when the force $g(x)$ is smooth or has only bounded first derivatives. Finally, for the MMT model, we discuss the difficulty of the problem and some potential solution which we believe has provided enough knowledge for future students to solve interesting fractional equations.

Matlab Code for examples in this report is available upon request.

Contents

List of Figures

List of Tables

1 Introduction

Fractional calculus means calculus with fractional orders. For instance, x^2, we can easily find its first order or any integer order of derivatives and integrals. However, it is more complicate with the fractinonal order. If we want 2.3th order of x,then we need to use fractional calculus. Fractional calculus not only apply to the math field, but also apply to some physics, sciences and technology field.

To use fractional calculus, essential knowledge in mathematical fields should be used. Such as the basic functions Euler's gamma function and Euler's beta function. Some famous fractional operators to apply are Grunewald Letnikov Fractional Deravative, Riemann Lowville Fractional Deravative, and Caputo Fractional Derivative. The major numerical methods are finite difference methods, spectral methods and finite element methods.

The goal of our project is to numerically solve a turbulence testbed problem – Majda-McLaughlin-Tabak(MMT) [2] model which is a nonlinear time dependent fractional differential equation. To achieve our goal, the first objective is establishing fractional Jacobi Polynomial; the second objective is developing numerical methods for some fractional differential equations; the third objective is measuring errors and convergence orders and infer the smoothness of solutions. For each problem we consider, we perform numerical tests with MATLAB and present numerical results and include figures and tables and make discussion based on results. We also discuss how to solve our ultimate goal problem based on our numerical results.

The rest of the paper is organized as follows. We present preliminary knowledge in chapter 2, solve fractional Laplace equations in linear system in chapter 3. We also show that how to solve fractional Laplace equations in non-linear system in chapter 4. In addition, we provide the equation with time dependent in chapter 5. Moreover, we explain how to solve the MMT model in chapter 6. Finally, we make a conclusion in chapter 7.

2 Preliminaries

2.1 Fractional Laplacian

We consider the fractional Laplacian, which is defined as

$$(-\Delta)^{\alpha/2}u(x) = c_{d,\alpha}\int_{\mathbb{R}^d}\frac{u(x)-u(y)}{|x-y|^{d+\alpha}}\,dy, \tag{2.1}$$

where $c_{d,\alpha}$ is a normalization constant

$$c_{d,\alpha} = \frac{2^{\alpha}\Gamma(\frac{\alpha+d}{2})}{\pi^{d/2}\,|\Gamma(-\alpha/2)|}.$$

The integral in (2.1) is understood in the sense of principle value:

$$(-\Delta)^{\alpha/2}u(x) = c_{d,\alpha}\lim_{\varepsilon\to 0^+}\int_{\mathbb{R}^d\cap\{|y-x|>\varepsilon\}}\frac{u(x)-u(y)}{|x-y|^{d+\alpha}}\,dy.$$

When $d=1$, we have the following conclusion:

Lemma 2.1 ([3]) *Let $u_p = (1-x^2)^p$, $|x|\le 1$, $p>-1$ and $u_p(x)=0$ when $|x|>1$, then for $x\in(-1,1)$*

$$(-\Delta)^{\alpha/2}u_p(x) = c_{1,\alpha}B(-\alpha/2,\,p+1)\,{}_2F_1(\frac{\alpha+1}{2},-p+\frac{\alpha}{2};\frac{1}{2};x^2). \tag{2.2}$$

Let $v_p(x) = (1-x^2)^p x$, $|x|\le 1$ and $v_p(x)=0$ when $|x|>1$, $p>-1$. Then for $x\in(-1,1)$

$$(-\Delta)^{\alpha/2}v_p(x) = (\alpha+1)c_{1,\alpha}B(-\alpha/2,\,p+1)\,{}_2F_1(\frac{\alpha+3}{2},-p+\frac{\alpha}{2};\frac{3}{2};x^2)x. \tag{2.3}$$

Here the hypergeometric function ${}_2F_1(a,b;c;z)$ is defined for $|z|<1$ by the power series

$${}_2F_1(a,b;c;z) = \sum_{n=0}^{\infty}\frac{(a)_n(b)_n}{(c)_n}\frac{z^n}{n!}.$$

Here $(q)_n$ is the (rising) Pochhammer symbol, which is defined by:

$$(q)_n = \begin{cases} 1 & n=0 \\ q(q+1)\cdots(q+n-1) & n>0 \end{cases}.$$

2.2 Jacobi Polynomials

The Jacobi polynomials are mutually orthogonal: for $\gamma,\ \theta > -1$,

$$\int_{-1}^{1}(1-x)^{\gamma}(1+x)^{\theta}P_m^{\gamma,\theta}(x)P_n^{\gamma,\theta}(x)\,dx = e_n(\gamma,\theta)\delta_{nm}. \tag{2.4}$$

Here $\delta_{nm} = 1$ if $n = m$ and is zero otherwise and

$$e_n(\gamma,\theta) = \frac{2^{\gamma+\theta+1}}{2n+\gamma+\theta+1}\frac{\Gamma(n+\gamma+1)\Gamma(n+\theta+1)}{\Gamma(n+\gamma+\theta+1)n!}.$$

When $\gamma = \theta$, we write $e_n(\gamma) =: e_n(\gamma,\theta)$.

From Lemma 2.1, we introduce the following conclusion.

Lemma 2.2 *For $P_n^{\alpha/2,\,\alpha/2}(x)$,the n-th order Jacobi Polynomial with weight function $(1-x^2)^{\alpha/2}$, it holds that*

$$(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}P_n^{\alpha/2,\,\alpha/2}(x)] = A_{n,\alpha}P_n^{\alpha/2,\,\alpha/2}(x).$$

where $A_{n,\alpha} = \frac{\Gamma(\alpha+n+1)}{n!}$ and we assume that $(1-x^2)^{\alpha/2}P_n^{\alpha/2,\,\alpha/2}(x)$ is defined to be zero when $|x| \geq 1$.

This lemma can be proved from Lemma 2.1. The proof is due to Professor Zhang and the proof can be found in the appendix B.

2.3 Gauss-Jacobi quadrature rule

The Gaussian Quadrature Rule [11] is to approximate the following integration with a continuous function $f(x)$ using a finite sum

$$\int_{-1}^{1}(1-x)^{\gamma}(1+x)^{\theta}f(x)\,dx \approx \sum_{i=0}^{n} w_i f(x_i), \tag{2.5}$$

where x_i's are the zeros of the Jacobi polynomials $P_{n+1}^{\gamma,\theta}$ and w_i are the corresponding quadrature weights. The quadrature rule (2.5) is exact when $f(x)$ is an algebraic polynomial of degree $2n+1$ or less.

2.4 Famous Fractional Derivatives

Riemann-Liouville fractional derivative [8]

If $f(x) \in C(a{,}b)$ and $x \in (a{,}b)$ then

$$ {}_aD_t^{\alpha}f(t) = \frac{d^n}{dt^n}{}_aD_t^{-(n-\alpha)}f(t) = \frac{d^n}{dt^n}{}_aI_t^{n-\alpha}f(t) $$

where

$$ {}_aI_t^{n-\alpha}f(t) = \frac{1}{\Gamma(n-\alpha)}\int_a^t \frac{f(\tau)}{(t-\tau)^{\alpha+1-n}}d\tau. $$

Caputo fractional derivative [1]

Caputo's definition is illustrated as follows:

$$ {}_a^CD_t^{\alpha}f(t) = \frac{1}{\Gamma(n-\alpha)}\int_a^t \frac{f^{(n)}(\tau)d\tau}{(t-\tau)^{\alpha+1-n}}. $$

Riesz fractional potential

Spectral relationship for the classical Riesz potential [10]. If $0 < v < 1$ and r and k are integer numbers such that $r > -\frac{v+1}{2}$, $k > -\frac{v+3}{2}$, then for $-1 < x < 1$ the following holds

$$ \int_{-1}^{1} \frac{Q_m^{\frac{v-1}{2}+r,\frac{v+1}{2}+k}(t)}{|x-t|^v}dt = \frac{\pi(-1)^r 2^{r+k+1}\Gamma(m+v)}{m!\Gamma(v)\cos(v\pi/2)}P_{m+r+k+1}^{\frac{v-1}{2}-r,\frac{v-3}{2}-k}(x), $$

where $m+k+r+1 > 0$ and

$$ Q_m^{\alpha,\beta}(t) = (1-t)^{\alpha}(1+t)^{\beta}P_m^{\alpha,\beta}(t), \quad -1 < t < 1, \quad \alpha,\beta > -1. $$

Specifically, when $r = k = 0$, we have if $0 < v < 1$, then for $-1 < x < 1$ the following holds

$$ \int_{-1}^{1} \frac{Q_m^{\frac{v-1}{2},\frac{v+1}{2}}(t)}{|x-t|^v}dt = \frac{\pi\Gamma(m+v)}{m!\Gamma(v)\cos(v\pi/2)}P_{m+r+k+1}^{\frac{v-1}{2},\frac{v-1}{2}}(x), \quad m = 0,1,2,\ldots $$

Through Riesz fractional potential, one can define Riesz derivative [12].

2.5 Computation of convergence rate

Suppose that g_n is a good approximation of f, say, $\|f-g_n\| \sim Cn^{-r}$, where C does not depend on n. To determine the convergence rate of methods, we can use the following formula

$$ \frac{\log(\|f-g_{n_2}\| / \|f-g_{n_1}\|)}{\log(n_2/n_1)}. $$

Denote that $E_n = \|f-g_n\|$. Suppose that $E_n \sim Cn^{-r}$. We then have

$$ \frac{E_{n_2}}{E_{n_1}} \sim (\frac{n_2}{n_1})^{-r}. $$

Taking the logarithm over both sides leads to the formula above.

When f is not known, we can replace f with some f_N obtained with a numerical method where N is sufficiently large so that $f - f_N$ is much smaller than $f_N - g_n$

$$\|f - g_n\| = \|(f - f_N) + f_N - g_n\| \approx \|f_N - g_n\| \,.$$

We call this f_N as a reference solution and measure the rate by

$$\frac{\log(\|f_N - g_{n_2}\| \,/\, \|f_N - g_{n_1}\|)}{\log(n_2/n_1)}, \quad n_1,\, n_2 \ll N.$$

3 How to Determine u_N for a Linear Equation

In this chapter, we solve two linear fractional differential equations with fractional Laplacian. We give two numerical examples and discuss the convergence order of our method, when $f(x)$=sin(x) and $f(x)$=|sin(x)|.

3.1 u_N for Fractional Equation $(-\Delta)^{\alpha/2}u = f$

We first consider the following Fractional Poisson Equation in one dimension for $0 < \alpha < 2$.

$$\begin{aligned} (-\Delta)^{\alpha/2}u &= f, \quad x \in (-1,1), \\ u &= 0, \quad x \in (-1,\infty) \cup [1,\infty). \end{aligned} \tag{3.1}$$

We approximate u by u_N where u_N is a truncated Fourier-Jacobi expansion:

$$u_N(x) = \sum_{n=0}^{N} a_n h_n(x) \in (1-x^2)^{\alpha/2}\mathbb{P}_N, \quad h_n = (1-x^2)^{\alpha/2} P_n^{\alpha/2,\,\alpha/2}(x). \tag{3.2}$$

Here $\mathbb{P}_N$ is the set of algebraic polynomials of order less than $N+1$ and for any $v \in (1-x^2)^{\alpha/2}\mathbb{P}_N$, there exists a polynomial p_n of order less than $N+1$ such that $v = 1-x^2)^{\alpha/2}p_n$

Then we can formulate the numerical solution as follows: to find $u_N \in (1-x^2)^{\alpha/2}\mathbb{P}_N$ such that for all test functions $v \in (1-x^2)^{\alpha/2}\mathbb{P}_N$ it holds

$$\int_{-1}^{1} (-\Delta)^{\alpha/2} u_N(x) v(x)\, dx = \int_{-1}^{1} f(x)v(x)\, dx. \tag{3.3}$$

Plugging u_N for (3.2) into (3.3) and by **Lemma 2.2**, we have

$$\sum_{n=0}^{N} a_n A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x) v(x)\, dx = \int_{-1}^{1} f(x)v(x)\, dx.$$

Taking $v = h_k = (1-x^2)^{\alpha/2} P_k^{\alpha/2,\,\alpha/2}(x)$, $k = 0, 1, \ldots, N$, we obtain

$$\sum_{n=0}^{N} a_n A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} P_k^{\alpha/2,\,\alpha/2}(x)\, dx = \int_{-1}^{1} f(x) P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\, dx.$$

Recall the orthogonality of Jacobi polynomials (2.4) and we then have

$$\boxed{a_k A_{k,\alpha} e_k(\alpha/2) = f_k, \quad k = 0, 1, 2, \ldots, N.} \tag{3.4}$$

where $e_k(\alpha/2)$ is from (2.2) and we denote

$$\boxed{f_k = \int_{-1}^{1} f(x) P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} dx, \quad k = 0, 1, 2, \ldots, N.} \tag{3.5}$$

By (3.4), it only requires to find f_k. Here we find f_k by numerical integration using Gauss-Jacobi quadrature rule. Specifically, when $f(x)$ have high-order derivatives, we use

$$\boxed{f_k \approx \sum_{n=0}^{m} f(x_i) P_k^{\alpha/2,\,\alpha/2}(x_i) w_i.} \tag{3.6}$$

Here, x_i's are the nodes of Jacobi Polynomial $P_{m+1}^{\alpha/2,\,\alpha/2}(x)$, w_i's are the corresponding quadrature weights.

Once we find f_k, we have a_k and thus have the numerical solution u_N from (3.4) and (3.2).

3.1.1 Numerical Results

Consider the two cases, when $f(x) = \sin(x)$ and $f(x) = |\sin(x)|$, to check if solutions are smooth or non-smooth by measuring the computation errors.

We introduce the method we used for measuring error. For error in weighted L_2

$$E_{WL_2} = \frac{\left|\left|(1-x^2)^{-\alpha/2}(u_N - u_{\text{ref}})\right|\right|_{L^2}}{\left|\left|(1-x^2)^{-\alpha/2}u_{\text{ref}}\right|\right|_{L^2}} \tag{3.7}$$

Here u_{ref} is a reference solution using the same method but with $N = 256$.

For error in weighted L_∞

$$E_{WL_\infty} = \frac{\left|\left|(1-x^2)^{-\alpha/2}(u_N - u_{\text{ref}})\right|\right|_{L^\infty}}{\left|\left|(1-x^2)^{-\alpha/2}u_{\text{ref}}\right|\right|_{L^\infty}} \tag{3.8}$$

Here the norms are defined as

$$||v||_{L^2} = (\int_{-1}^{1} v^2(x)dx)^{1/2}, \quad ||v||_{L^\infty} = \max_{0\le j\le M} |v(x_j)|. \tag{3.9}$$

Here we use $M = 1000$ points between -1 and 1 from the Gauss-Jacobi rule with $\alpha = 0$ (the zeros of $P_{M+1}^{0,0}(x)$).

Example 3.1 *Let $f(x) = \sin(x)$.*

The numerical solutions are plotted in Figure 3.1 where $\alpha = 0.4$ and $N = 256$ and in Figure 3.2 where $\alpha = 1.4$ and $N = 256$.

The convergence order and errors of numerical solutions are calculated in Table 3.1 where $\alpha = 0.4$ and $N = 256$ and Table 3.2 where $\alpha = 1.4$ and $N = 256$. We used $N = 256$ to obtain a reference solution, i.e., $u_{\text{ref}} = u_{256}$.

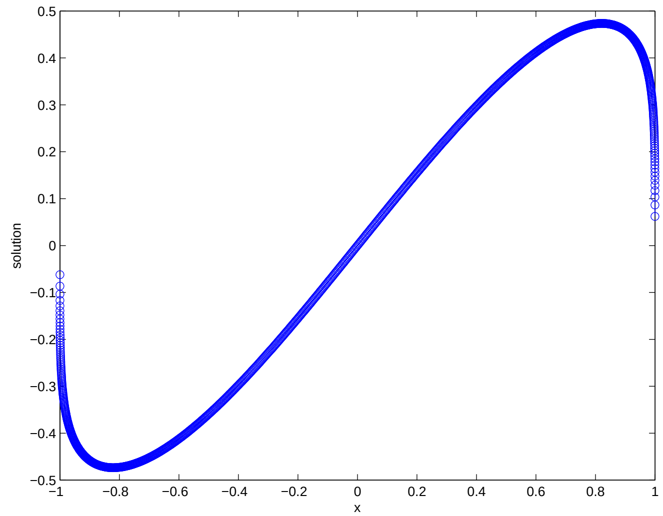

Figure 3.1: Numerical Solution for $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	3.5051e-04	5.1207e-04	-	-
8	5.9788e-09	1.0272e-08	-15.8393	-15.6053
16	2.3450e-14	2.2614e-13	-17.9599	-15.4712
32	2.1719e-14	1.8523e-13	-0.1106	-0.2879
64	2.0031e-14	1.5064e-13	-0.1167	-0.2982
128	1.8159e-14	1.0627e-13	-0.1416	-0.5034

Table 3.1: Error table for $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

We observe that when $N = 16$, the accuracy is close to machine accuracy in Table 3.1 and is slightly improved when $N = 32, 64, 128$. The numerical results (convergence order) suggests the smoothness of the solution, to be more precise, $u(x)/(1 - x^2)^{\alpha/2}$.

Compared to Table 3.1, similar effects are observed in Table 3.2 when $\alpha = 1.4$. But the accuracy for $N = 16$ is closer to machine accuracy compared to the case when $\alpha = 0.4$.Also, when $\alpha = 1.4$, the solution has smaller magnitude when than that when $\alpha = 0.4$. The maximum is around 0.5 while the maximum for $\alpha = 0.4$ is around 0.15.

Example 3.2 *Let* $f(x) = |\sin(x)|$.

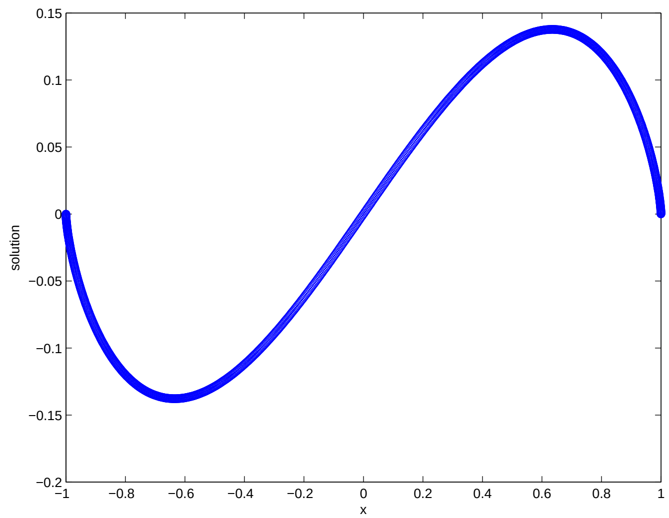

Figure 3.2: Numerical Solution for $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	1.1655e-04	1.2598e-04	-	-
8	1.1993e-09	1.3023e-09	-16.5684	-16.5618
16	5.5248-16	1.3099e-15	-21.0498	-19.9232
32	3.2690e-16	6.0458e-16	-0.7571	-1.1155
64	1.2384e-16	3.0229e-16	-1.4004	-1
128	6.0900e-17	2.0153e-16	-1.0239	-0.5850

Table 3.2: Error table for $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$

The numerical solutions are plotted in Figure 3.3 where $f(x) = |\sin(x)|$ with $\alpha = 0.4$ and $N = 256$ and Figure 3.4 where $f(x) = |\sin(x)|$ with $\alpha = 1.4$ and $N = 256$.

The convergence order and errors of numerical solutions are calculated in Table 3.3 when $f(x) = |\sin(x)|$ with $\alpha = 0.4$ and $N = 256$, and Table 3.4 when $f(x) = |\sin(x)|$ with $\alpha = 1.4$ and $N = 256$. We used $N = 256$ to obtain a reference solution, i.e., $u_{\text{ref}} = u_{256}$.

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0334	0.0711	-	-
8	0.0120	0.0314	-1.4753	-1.1778
16	0.0038	0.0120	-1.6486	-1.3918
32	0.0011	0.0036	-1.7924	-1.7195
64	2.5902e-04	6.4433e-04	-2.0949	-2.4963
128	5.4686e-05	2.0706e-04	-2.2439	-1.6377

Table 3.3: Error table when $f(x) = |\sin(x)|$ with $\alpha = 0.4$ and $N = 256$

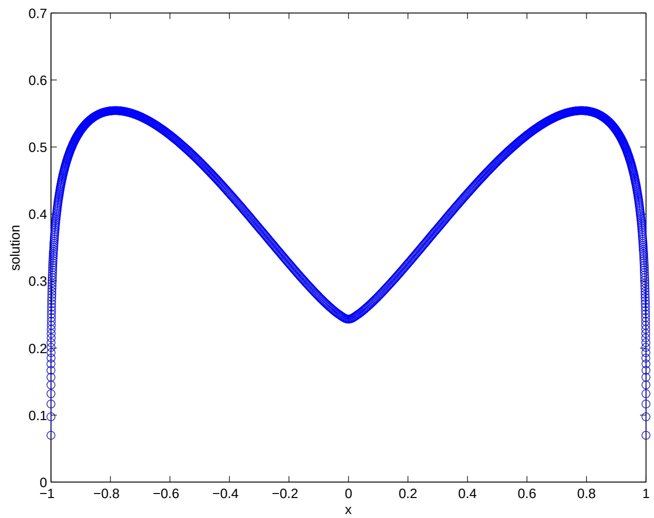

Figure 3.3: Numerical Solution of u_N for $f(x) = |\sin(x)|$, $\alpha = 0.4$ and $N = 256$

The figure suggests that there is a sharp change at 0 while $f(x) = |\sin(x)|$ has no second-order derivative at 0. The error table shows that with N getting larger, the error is getting smaller. However, the convergence order is around 2, which is much smaller than that in Example 3.1 when $\alpha = 0.4$. The numerical results suggests that the solution in Example 3.1 is much smoother than that in here.

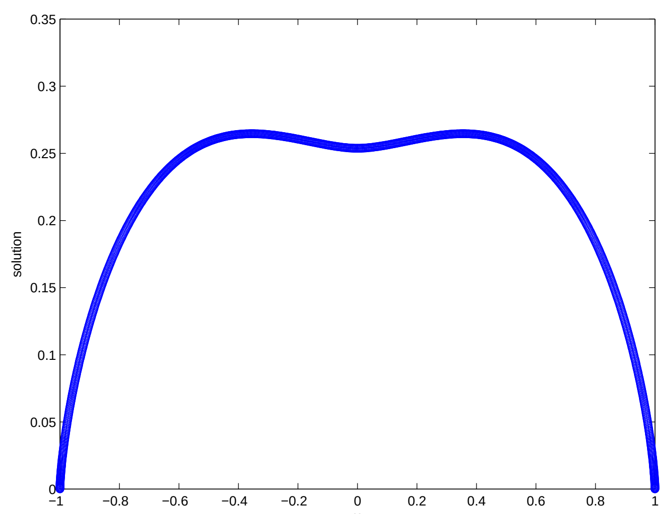

Figure 3.4: Numerical Solution of u_N for $f(x) = |\sin(x)|$, $\alpha = 1.4$ and $N = 256$

Compared to the case when $\alpha = 0.4$, we do not observe a sharp change at 0 even though $f(x) = |\sin(x)|$ has no second-order derivative at 0. The figure suggests that the solution is much smoother than that when $\alpha = 0.4$.

The error table shows that with N getting larger, the error is getting smaller. However, the

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0066	0.0131	-	-
8	0.0016	0.0037	-2.0659	-1.8086
16	3.0064e-04	8.6304e-04	-2.3889	-2.1177
32	4.8634e-05	1.5387e-04	-2.6280	-2.4877
64	6.5219e-06	1.5399e-05	-2.8986	-3.3208
128	6.6519e-07	2.4062e-06	-3.2935	-2.6780

Table 3.4: Error table when $f(x) = |\sin(x)|$ with $\alpha = 1.4$ and $N = 256$

convergence order is around 3, which is larger than that for $\alpha = 0.4$. The convergence order again suggests that the solution here is much smoother than that for $\alpha = 0.4$.

Also, we observe that the solution has a smaller magnitude than that for $\alpha = 0.4$. The maximum is around 0.28 while the maximum for $\alpha = 0.4$ is around 0.56.

3.2 u_N for Fractional Equation $(-\Delta)^{\alpha/2}u{+}\lambda u{=}f$

We consider the following equation in one dimension for $0 < \alpha < 2$ and $\lambda > 0$:

$$\begin{aligned} (-\Delta)^{\alpha/2}u + \lambda u &= f, \quad x \in (-1,1), \\ u &= 0, \quad x \in (-1,\infty) \cup [1,\infty) \end{aligned} \tag{3.10}$$

We again approximate u by u_N from (3.2) and formulate the numerical solution as follows: to find $u_N \in (1-x^2)^{\alpha/2}\mathbb{P}_N$ such that for all $v \in (1-x^2)^{\alpha/2}\mathbb{P}_N$, it holds

$$\int_{-1}^{1} (-\Delta)^{\alpha/2} u_N(x) v(x)\, dx + \lambda \int_{-1}^{1} u_N(x) v(x)\, dx = \int_{-1}^{1} f(x) v(x)\, dx. \tag{3.11}$$

Plugging u_N (3.2) into (3.11) and by **Lemma 2.2**, we have

$$\sum_{n=0}^{N} a_n A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x) v(x)\, dx + \lambda \int_{-1}^{1} a_n h_n(x) v(x)\, dx = \int_{-1}^{1} f(x) v(x)\, dx.$$

Taking $v = h_k = (1-x^2)^{\alpha/2} P_k^{\alpha/2,\,\alpha/2}(x)$, $k = 0, 1, \ldots, N$, we obtain

$$\begin{aligned} &\sum_{n=0}^{N} a_n A_{n,\alpha} \int_{-1}^{1} (1-x^2)^{\alpha/2} P_n^{\alpha/2,\,\alpha/2}(x) P_k^{\alpha/2,\,\alpha/2}(x)\, dx \\ &\quad + \lambda \sum_{n=0}^{N} a_n \int_{-1}^{1} (1-x^2)^{\alpha} P_n^{\alpha/2,\,\alpha/2}(x) P_k^{\alpha/2,\,\alpha/2}(x)\, dx = \int_{-1}^{1} f(x) P_k^{\alpha/2,\,\alpha/2}(x) (1-x^2)^{\alpha/2}\, dx. \end{aligned}$$

Recall the orthogonality of Jacobi polynomials (2.4), we then have

$$\boxed{a_k A_{k,\alpha} e_n(\alpha/2) + \lambda \sum_{n=0}^{N} M_{n,k} a_k = f_k, \quad k = 0, 1, 2, \ldots, N} \tag{3.12}$$

where f_k is the same as in (3.5) and

$$M_{n,k} = \int_{-1}^{1} (1-x^2)^{\alpha} P_n^{\alpha/2,\,\alpha/2}(x) P_k^{\alpha/2,\,\alpha/2}(x)\, dx. \tag{3.13}$$

To find f_k, we again apply numerical integration using Gauss-Jacobi quadrature rule when $f(x)$ have high-order derivatives. Specifically, we use

$$\boxed{f_k \approx \sum_{n=0}^{m} f(x_i) P_k^{\alpha/2,\,\alpha/2}(x_i) w_i.}$$

Here, x_i's are the nodes of Jacobi Polynomial $P_{m+1}^{\alpha/2,\,\alpha/2}(x)$, w_i's are the corresponding quadrature weights.

To find $M_{n,k}$, we also apply numerical integration using Gauss-Jacobi quadrature rule (2.5)

$$\boxed{M_{n,k} = \int_{-1}^{1} (1-x^2)^{\alpha} P_n^{\alpha/2,\,\alpha/2}(x) P_k^{\alpha/2,\,\alpha/2}(x)\, dx = \sum_{j=0}^{N} P_n^{\alpha/2,\,\alpha/2}(\mathsf{x}_j) P_k^{\alpha/2,\,\alpha/2}(\mathsf{x}_k) \mathsf{w}_j.} \tag{3.14}$$

Here x_i's are the zeros of Jacobi Polynomial $P_{N+1}^{\alpha,\alpha}(x)$, w_i's are the corresponding quadrature weights. The quadrature rule here is exact since $n + k \le 2N$ while the quadrature rule here is exact for all $2N + 1$-th order polynomials.

Remark 3.3 *We can reduce the amount of operations in obtaining* (3.14)*. Note that* $M_{n,k} = M_{k,n}$*. Moreover, when* $n + k$ *is odd,* $M_{n,k} = 0$*. In fact,* $P_n^{\alpha/2,\alpha/2}$*'s are odd functions when* n *is odd and are even functions* n *is even.*

Let

$$S_k = A_{k,\alpha} e_k(\alpha/2). \tag{3.15}$$

Plugging in (3.15) and (3.14) we have

$$\boxed{a_k S_k + \lambda \sum_{n=0}^{N} a_k M_{n,k} = f_k, \quad k = 0, 1, 2, \ldots, N.} \tag{3.16}$$

We rewrite (3.16) in a matrix format. Here, denoting S as the matrix of S_k, M as the matrix of M_k.

$$S = \begin{bmatrix} S_0 & 0 & 0 & \dots & 0 \\ 0 & S_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & S_N \end{bmatrix}, \quad M = \begin{bmatrix} M_{0,0} & M_{1,0} & M_{2,0} & \dots & M_{N,0} \\ M_{0,1} & M_{1,1} & M_{2,1} & \dots & M_{N,1} \\ \dots & \dots & \dots & \dots & \dots \\ M_{0,N} & M_{1,N} & M_{2,N} & \dots & M_{N,N} \end{bmatrix}.$$

Denote B that is

$$B = S + \lambda M = \begin{bmatrix} S_0 + \lambda M_{0,0} & \lambda M_{1,0} & \lambda M_{2,0} & \dots & \lambda M_{N,0} \\ \lambda M_{0,1} & S_1 + \lambda M_{1,1} & \lambda M_{2,1} & \dots & \lambda M_{N,1} \\ \dots & \dots & \dots & \dots & \dots \\ \lambda M_{0,N} & \lambda M_{1,N} & \lambda M_{2,N} & \dots & S_N + \lambda M_{N,N} \end{bmatrix}.$$

Denote $\vec{a} = (a_0, a_1, a_2, ...a_N)^\top$ and $\vec{f} = (f_0, f_1, f_2, ...f_N)^\top$. Then we have the resulting linear system

$$\boxed{B\vec{a} = \vec{f}.}$$

Thus, we get $\vec{a} = B^{-1}\vec{f}$ and obtain the numerical solution u_N.

3.2.1 Numerical Results

Take λ=1. Consider two cases: when $f(x) = \sin(x)$ and $f(x) = |\sin(x)|$.

Example 3.4 *Let* $f(x) = \sin(x)$.

The numerical solutions of two situation are plotted in Figure 3.5 with $\alpha = 0.4$ and $N = 256$ and Figure 3.6 with $\alpha = 1.4$ and $N = 256$.

The convergence order and errors of numerical solutions are calculated in Table 3.5 where $\alpha = 0.4$ and $N = 256$, and Table 3.6 when $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$. We used $N = 256$ to obtain a reference solution, i.e., $u_{\text{ref}} = u_{256}$.

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0220	0.1190	-	-
8	0.0079	0.0699	-1.4824	-0.7661
16	0.0026	0.0362	-1.6122	-0.9501
32	7.7253e-04	0.0149	-1.7378	-1.2763
64	1.8538e-04	0.0033	-2.0591	-2.1675
128	4.0150e-05	7.0276e-04	-2.2070	-2.2431

Table 3.5: Error table when $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

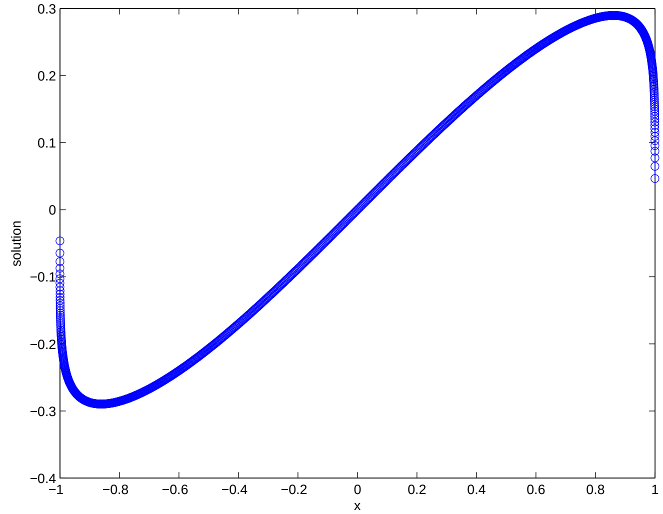

Figure 3.5: Numerical solution when $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

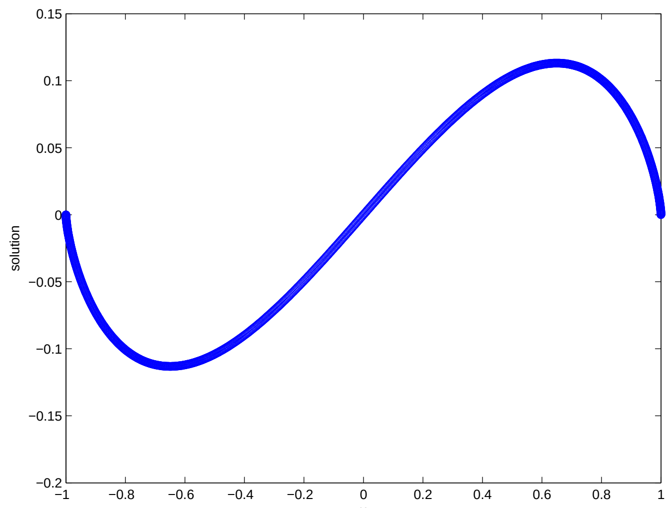

Figure 3.6: Numerical solution when $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0028	0.0089	-	-
8	2.3131e-04	0.0013	-3.5955	-2.7347
16	1.9453e-05	2.0654e-04	-3.5718	-2.6911
32	1.5523e-06	2.8718e-05	-3.6475	-2.8464
64	1.1639e-07	2.9474e-06	-3.7374	-3.2845
128	2.8217e-09	7.4302e-08	-5.3663	-5.3099

Table 3.6: Error table when $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$

The error table shows that with N getting larger, the error is getting smaller. When α=0.4, the convergence order is around 2 which indicates that the solution is not smooth. When $alpha$=1.4, the convergence order is around 4 which means that the solution is smoother than α=0.4.However, the convergence order is still low, around 4, which indicates gain that the solution is not smooth. Also, the solution has smaller maximum than the previous one, see Figure 3.6 and Figure 3.5.

Example 3.5 *Let* $f(x) = |\sin(x)|$.

The numerical solutions of two cases are plotted in Figure 3.7 with $\alpha = 0.4$ and $N = 256$) and Figure 3.8($f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$).

The convergence order and errors of numerical solutions are calculated in Table 3.7 with $\alpha = 0.4$ and $N = 256$ and Table 3.8 with $\alpha = 1.4$ and $N = 256$.We used $N = 256$ to obtain a reference solution, i.e., $u_{\text{ref}} = u_{256}$.

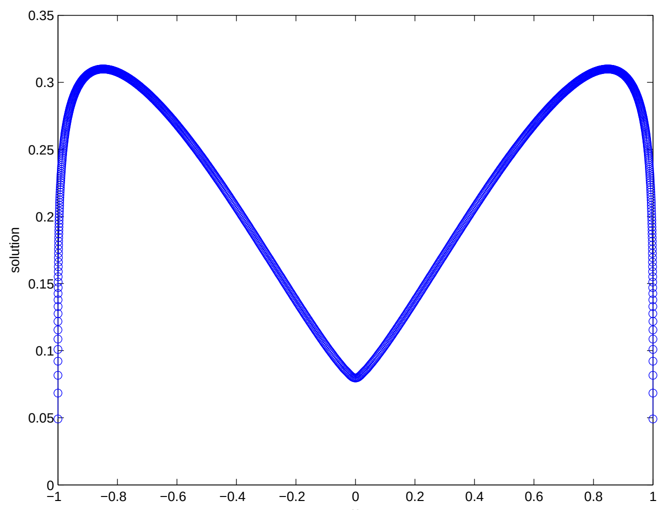

Figure 3.7: Numerical solution when $f(x) = |\sin(x)|$ with $\alpha = 0.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0543	0.0705	-	-
8	0.0185	0.0399	-1.5520	-0.8214
16	0.0060	0.0167	-1.6228	-1.2589
32	0.0018	0.0053	-1.7499	-1.6409
64	4.3186e-04	9.8859e-04	-2.0505	-2.4341
128	9.5353e-05	3.3140e-04	-2.1792	-1.5768

Table 3.7: Error table when $f(x) = |\sin(x)|$ with $\alpha = 0.4$ and $N = 256$

The figure 3.7 suggests that there is a sharp change at 0 while $f(x) = |\sin(x)|$ has no second-order derivative at 0. The error table 3.7 of α=0.4 shows that with N getting larger, the error is getting smaller. However, the convergence order is around 2, which is almost the same as in Example 3.2 when $\alpha = 0.4$. The numerical results suggests that the solution is not smooth.

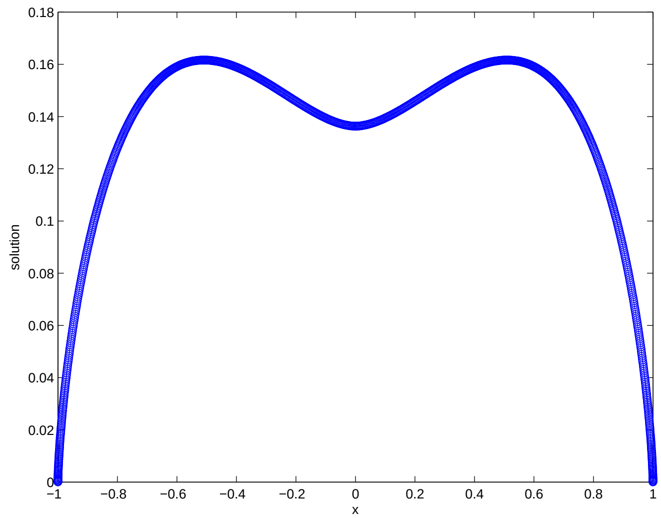

Figure 3.8: Numerical solution when $f(x) = |\sin(x)|$ with $\alpha = 1.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0101	0.0192	-	-
8	0.0025	0.0059	-2.0138	-1.6982
16	4.8709e-04	0.0014	-2.3646	-2.0846
32	7.9412e-05	2.5055e-04	-2.6168	-2.4760
64	1.0684e-05	2.5154e-05	-2.8939	-3.3163
128	1.0913e-06	3.9368e-06	-3.2913	-2.6757

Table 3.8: Error table when $f(x) = |\sin(x)|$ with $\alpha = 1.4$ and $N = 256$

Compared to the case when $\alpha = 0.4$, we do not observe a sharp change at 0 in Figure 3.8 . The figure suggests that the solution is much smoother than that when $\alpha = 0.4$. However, from Table 3.8 we can see that the convergence order is around 3, which is almost the same as in Example 3.2 when $\alpha = 1.4$. The convergence order again suggests that the solution here is much smoother than that for $\alpha = 0.4$.

Also, we observe from figure 3.8 that the solution has a smaller magnitude than that for $\alpha = 0.4$. From figure 3.7, the maximum is around 0.16 while the maximum for $\alpha = 0.4$ is around 0.32.

3.3 Summary and Discussion

In this chapter, we considered Jacobi spectral methods for two linear fractional differential equations:

$$(-\Delta)^{\alpha/2}u + \lambda u = f.$$

We found that

- When $\lambda = 0$ and f was smooth, the convergence order was high and the solution $(u(x)/(1-x^2)^{\alpha/2})$ was smooth, see Example 3.1.
- When $\lambda = 0$ and f was not smooth, the convergence order was high and the solution $(u(x)/(1-x^2)^{\alpha/2})$ was not smooth, see Example 3.2.
- When $\lambda = 1$ and $f = \sin(x)$ or $f = |\sin(x)|$, the convergence order was low and the solution $(u(x)/(1-x^2)^{\alpha/2})$ was not smooth, see Example 3.4 and Example 3.5.

When $\lambda = 0$, we considered two different f: f =$\sin(x)$ and f=$|\sin(x)|$. For $\alpha = 0.4$ and $N = 128$, the error in weighted L_2 for $\sin(x)$ was $1.8159e-14$ and the error in weighted L_2 for $|\sin(x)|$ was $5.4686e-05$. This was because that $\sin(x)$ was differentiable when x in $\mathbb{R}$ and the derivatives are always bounded by 1. But $|\sin(x)|$ is not differentiable at 0. We could see that $\sin(x)$ was smoother than $|\sin(x)|$. That's why they had such large difference. When $\lambda = 1$, we did not observe the same effects and we believed that the solution was not smooth because of the operator $(-\Delta)^{\alpha/2}$.

Moreover, from error tables with convergence order, we found that when α was getting larger, the convergence order was getting larger. This suggested that with a large α, the solution was smoother than that from a small α.

4 How to Determine u_N for a Nonlinear Equation

In this chapter, we solve a nonlinear fractional differential equation with fractional Laplacian. We give two numerical examples and discuss the convergence order of our method, when $g(x)$=sin(x) and $g(x)$=|sin(x)|.

4.1 u_N for Fractional Equation $(-\Delta)^{\alpha/2}u = f(u) + g(x)$

Consider the following equation in one dimension for $0 < \alpha < 2$ and $\lambda > 0$:

$$\int_{-1}^{1} (-\Delta)^{\alpha/2} uv \, dx \quad = \quad \int_{-1}^{1} f(u_N) v \, dx + \int_{-1}^{1} gv \, dx, \tag{4.1}$$

Plugging u_N (3.2) into (4.1) and by Lemma 2.2, we have

$$\sum_{n=0}^{N} a_n A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x) v(x) \, dx = \int_{-1}^{1} f(u_N) v(x) dx + \int_{-1}^{1} g(x) v(x) \, dx.$$

Taking $v = h_k = (1-x^2)^{\alpha/2} P_k^{\alpha/2,\,\alpha/2}(x)$, $k = 0, 1, \ldots, N$, we obtain

$$\begin{aligned} &\sum_{n=0}^{N} a_n A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} P_k^{\alpha/2,\,\alpha/2}(x) \, dx \\ &\qquad = \int_{-1}^{1} f(u_N) P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} \, dx + \int_{-1}^{1} g(x) P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} \, dx. \end{aligned}$$

Recall the orthogonality of Jacobi polynomials (2.4) , we then have equation of g_k :

$$a_k A_{k,\alpha} \int_{-1}^{1} (1-x^2)^{\alpha/2} P_k^{(\alpha,\beta)}(x) P_k^{(\alpha,\beta)}(x) \, dx + \int_{-1}^{1} f(u_N) P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} \, dx = g_k, \quad k = 0, 1, 2, \ldots, N.$$

where we denote

$$\boxed{g_k = \int_{-1}^{1} g(x) P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2} dx, \quad k = 0, 1, 2, \ldots, N.}$$

By numerical integration using Gauss-Jacobi quadrature rule (2.5) when $g(x)$ have high-order derivatives, we use

$$\boxed{g_k \approx \sum_{n=0}^{m} g(x_i) P_k^{\alpha/2,\,\alpha/2}(x_i) w_i.}$$

Here, x_i's are the nodes of Jacobi Polynomial $P_{m+1}^{\alpha/2,\,\alpha/2}(x)$, w_i's are the corresponding quadrature weights.

Assume that $f(u) = u - u^3$.

$$\int_{-1}^{1} f(u_N)P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx = \int_{-1}^{1} u_N P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx - \int_{-1}^{1} (u_N)^3 P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx.$$

We can rewrite the equation (4.1) as

$$\sum_{n=0}^{N} a_n A_{n,\alpha} P_n^{\alpha/2,\,\alpha/2}(x_i) P_k^{\alpha/2,\,\alpha/2}(x_i) w_i - \int_{-1}^{1} u_N v(x)dx + \int_{-1}^{1} u_N^3 v(x)dx - g_k = 0 \tag{4.2}$$

Denote that
$F(\vec{a}) = (\int_{-1}^{1}(u_N)^3 P_0^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx, \ldots, \int_{-1}^{1}(u_N)^3 P_N^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx)^T$. To find $F(\vec{a})_k = \int_{-1}^{1}(u_N)^3 P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx$, we also apply Gauss-quadrature rule (2.5). Specifically, we have from (3.2) that

$$\int_{-1}^{1} (u_N)^3 P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx = \int_{-1}^{1} (1-x^2)^{2\alpha} \big(\sum_{j=0}^{N} a_j^n P_i^{\alpha/2}(x)\big)^3, P_k^{\alpha/2}(x)\,dx.$$

Note that the integrand $\big(\sum_{j=0}^{N} a_j^n P_i^{\alpha/2}(x)\big)^3, P_k^{\alpha/2}(x)$ is a polynomial of order no more than $4N$. Then we apply Gaussian quadrature rule (2.5) and we have

$$\boxed{\int_{-1}^{1} (u_N)^3 P_k^{\alpha/2,\,\alpha/2}(x)(1-x^2)^{\alpha/2}\,dx = \sum_{i=0}^{2N} \big(\sum_{j=0}^{N} a_j^n P_j^{\alpha/2}(\mathsf{y}_i)\big)^3 P_k^{\alpha/2,\,\alpha/2} \mathsf{w}_i.} \tag{4.3}$$

Here the y_i's ($i = 0, 1, 2, \ldots, 2N$) are the zeros of $P_{2N+1}^{2\alpha,2\alpha}$ and w_i's are the corresponding weights.

Denote also that $\vec{a} = (a_0, a_1, a_2, \ldots.a_N)^T$ and $\vec{g} = (g_0, g_1, g_2, \ldots.g_N)^T$. In a matrix form, we write (4.2) as

$$\boxed{G(\vec{a}) = S\vec{a} + F(\vec{a}) - \vec{g}(x) - M\vec{a} = 0.} \tag{4.4}$$

where M is the matrix with elements (3.14) and S is the diagonal matrix (3.15).

Now, by Newton's method, we obtain

$$\nabla_{\vec{a}} G(\vec{a})(\vec{a}^{r-1} - \vec{a}^r) = G(\vec{a}^r), \quad \text{where } \nabla_{\vec{a}} G(\vec{a}) = S - \nabla_{\vec{a}} F(\vec{a}) - M.$$

To use Newton's method, we need to find the initial value $\vec{a}^0$. Here we guess the value of $\vec{a}^0$ from the following equation

$$\nabla_{\vec{a}^0} G(\vec{a}^0) = S - M.$$

Thus, we can get $\vec{a}$ and then plug in equation (3.2) to get the numerical solution u_N.

4.1.1 Numerical Results

We check convergence order and measure errors of our method.

Example 4.1 *Assume* $f(u) = u^3$, $g(x) = \sin(x)$.

According to Newton's method, plugging in the matrix S, $\nabla_{\vec{a}} F(\vec{a})$, we get $\nabla_{\vec{a}} G(\vec{a}) =$

$$\begin{bmatrix} S_0 & 0 & 0 & \dots & 0 \\ 0 & S_1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & S_N \end{bmatrix} - \vec{g}(x) -$$

$$\begin{bmatrix} (3u_N^2 h_0(x), h_0(x)) & (3u_N^2 h_1(x), h_0(x)) & (3u_N^2 h_2(x), h_0(x)) & \dots & (3u_N^2 h_N(x), h_0(x)) \\ (3u_N^2 h_0(x), h_1(x)) & (3u_N^2 h_1(x), h_1(x)) & (3u_N^2 h_2(x), h_1(x)) & \dots & (3u_N^2 h_N(x), h_1(x)) \\ \dots & \dots & \dots & \dots & \dots \\ (3u_N^2 h_0(x), h_N(x)) & (3u_N^2 h_1(x), h_N(x)) & (3u_N^2 h_2(x), h_N(x)) & \dots & (3u_N^2 h_N(x), h_N(x)) \end{bmatrix}.$$

However, we find the solution is oscillatory which suggests that the Newton's method is not working well. We believe the failure of Newton's method is due to a bad initial guess $\vec{a}^0$ given above. Instead, we use a **fixed point iteration** method with tolerance $1e - 12$ to solve the equation.

The numerical solutions are plotted in Figure 4.1 where $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$, and in Figure 4.2 where $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$.

The convergence order and errors of numerical solutions are calculated in Table 4.1 where $\alpha = 0.4$ and $N = 256$, and Table 4.2 where $f(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$. We used $N = 256$ to obtain a reference solution, i.e., $u_{\text{ref}} = u_{256}$.

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0565	0.0785	-	-
8	0.0083	0.0135	-2.7703	-2.5417
16	7.2560e-04	0.0090	-3.5136	-0.5773
32	2.0721e-04	0.0045	-1.8081	-1.0128
64	5.0052e-05	0.0019	-2.0496	-1.2400
128	1.1760e-05	6.1751e-04	-2.0896	-1.6127

Table 4.1: Error table when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

The error table shows that with larger N, the error is smaller. We compare two error tables and find that when α=0.4, the convergence order is around 2. When α=1.4, the convergence

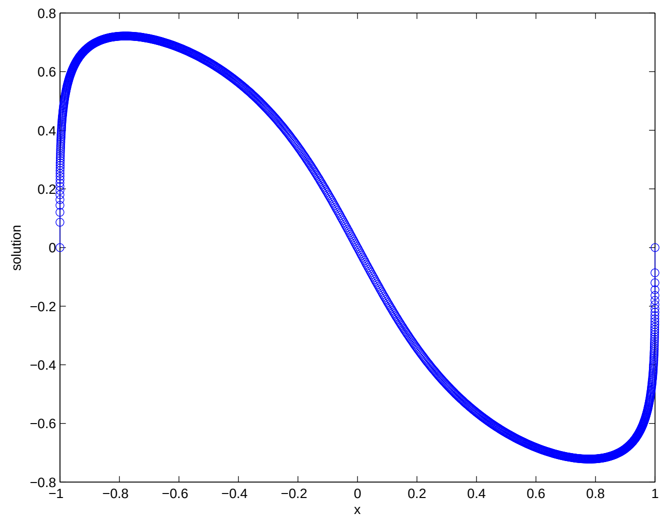

Figure 4.1: Numerical solution when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

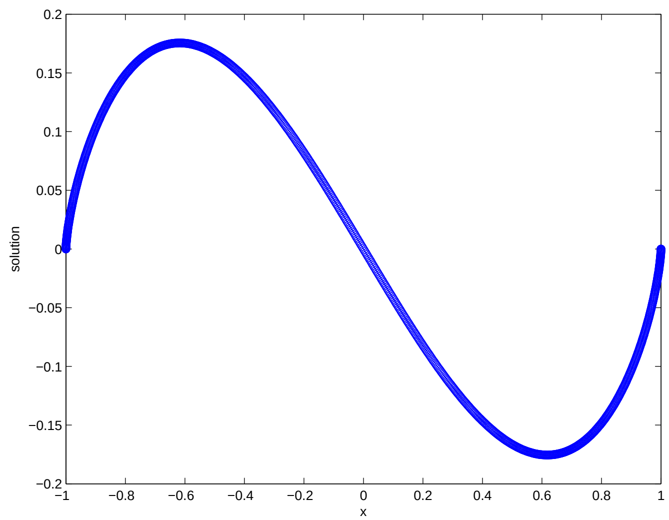

Figure 4.2: Numerical solution when $g(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$.

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	3.5444e-04	5.7414e-04	-	-
8	4.9591e-05	7.8128e-05	-2.8374	-2.8775
16	3.2080e-06	6.3001e-06	-3.9504	-3.6324
32	1.3194e-07	3.9494e-07	-4.6037	-3.9957
64	4.8471e-09	2.3120e-08	-4.7666	-4.0944
128	1.6717e-10	1.3178e-09	-4.8577	-4.1329

Table 4.2: Error table when $g(x) = \sin(x)$ with $\alpha = 1.4$ and $N = 256$.

.

order is around 4. It means that the solution with α=0.4 is smoother than with α=1.4. In addition, we find from the two figures that the maximum magnitude of Figure 4.1 is 0.75 and the maximum magnitude of Figure 4.2 is 0.15.

4.2 Summary and Discussion

In this chapter, we considered Newton's method and fixed point iteration method for this nonlinear equation:

$$(-\Delta)^{\alpha/2}u = f(u) + g(x).$$

We chose a fixed point method instead of Newton's method since Newton's method leads to solutions with high oscillation. We didn't find a good initial guess for Newton's method. Instead, fixed point iteration did not have this problem and worked for our nonlinear system.

We found that when $g = \sin(x)$ was smooth,the convergence order was low and thus the solution was not smooth. However, when α was lager, the numerical results suggested that solutions were smoother.

5 How to Determine u_N for Time Dependent Equations

In chapter 5, we solve both linear system and nonlinear system fractional equations in time dependent equations. Then, we give numerical examples and discuss the convergence order of our method, when $g(x)$=sin(x).

5.1 u_N for Fractional Equation $(-\Delta)^{\alpha/2}u+\frac{\partial u}{\partial t}=u+g$

We first consider the following equation in one dimension linear system for $0 < \alpha < 2$:

$$\int_{-1}^{1}(-\Delta)^{\alpha/2}u_N v\,dx + \int_{-1}^{1}\frac{\partial u_N}{\partial t}v\,dx \quad = \quad \int_{-1}^{1}u_N v\,dx + \int_{-1}^{1}gv\,dx. \tag{5.1}$$

We use the following approximation

$$u_N(t,x) = \sum_{n=0}^{N} a_n(t)h_n(x) \in (1-x^2)^{\alpha/2}\mathbb{P}_N, \quad h_n = (1-x^2)^{\alpha/2}P_n^{\alpha/2,\,\alpha/2}(x). \tag{5.2}$$

Plugging (5.2) into (5.1), we have

$$\begin{aligned} &\sum_{n=0}^{N} a_n A_{n,\alpha}\int_{-1}^{1}P_n^{\alpha/2,\,\alpha/2}(x)v(x)\,dx + \frac{d}{dt}\int_{-1}^{1}\sum_{n=0}^{N}a_n(t)h_n(x)v(x)\,dx \\ = \quad &\int_{-1}^{1}\sum_{n=0}^{N}a_n(t)h_n(x)v(x)\,dx + \int_{-1}^{1}g(x)v(x)\,dx. \end{aligned}$$

Taking $v = h_k(x)$ is not working because it makes the matrix M of $M_{n,k} = \int_{-1}^{1}(1-x^2)^{\alpha/2}h_k h_n\,dx$ a singular matrix. We find that when $N = 128$ and $\alpha = 1.4$, the determinant of M is $4.1054e-217$. Even when $N = 16$ and $\alpha = 1.4$, the determinant of M is $1.2863e-14$.

Instead of taking $v = h_k(x)$, we take $v = P_k^{\alpha/2,\,\alpha/2}(x)$, $k = 0, 1, \ldots, N$ and obtain

$$\begin{aligned} \sum_{n=0}^{N} a_n A_{n,\alpha}\int_{-1}^{1}P_n^{\alpha/2,\,\alpha/2}(x)P_k^{\alpha/2,\,\alpha/2}(x)\,dx + \frac{d}{dt}\sum_{n=0}^{N}a_n(t)\int_{-1}^{1}(1-x^2)^{\alpha/2}P_n^{\alpha/2,\,\alpha/2}(x)P_k^{\alpha/2,\,\alpha/2}(x)\,dx \\ = \sum_{n=0}^{N}a_n(t)\int_{-1}^{1}(1-x^2)^{\alpha/2}P_n^{\alpha/2,\,\alpha/2}(x)P_k^{\alpha/2,\,\alpha/2}(x)\,dx + \int_{-1}^{1}g(x)P_k^{\alpha/2,\,\alpha/2}(x)\,dx. \end{aligned}$$

By the orthogonality of (2.4) and (2.2), we can write the resulting equation as follows:

$$\boxed{\frac{d}{dt}a_k(t)e_k(\alpha/2) + \sum_{n=0}^{N} S_{n,k}a_k(t) = a_k(t)e_k(\alpha/2) + g_k, \quad k = 0, 1, \ldots, N.} \tag{5.3}$$

Here we denote

$$S_{n,k} = A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x) P_k^{\alpha/2,\,\alpha/2}(x)\,dx \text{ and } g_k = \int_{-1}^{1} g(x) P_k^{\alpha/2,\,\alpha/2}(x)\,dx.$$

To find $S_{n,k}$, we also apply numerical integration using Gauss-Jacobi quadrature rule (2.5)

$$S_{n,k} = A_{n,\alpha} \int_{-1}^{1} P_n^{\alpha/2,\,\alpha/2}(x) P_k^{\alpha/2,\,\alpha/2}(x)\,dx = A_{n,\alpha} \sum_{j=0}^{N} P_k^{\alpha/2,\,\alpha/2}(\mathbf{x}_i) P_k^{\alpha/2,\,\alpha/2}(\mathbf{x}_i)\mathbf{w}_i. \tag{5.4}$$

Here $\mathbf{x}_i$'s are the zeros of Jacobi Polynomial $P_{N+1}^{0,0}(x)$, $\mathbf{w}_i$'s are the corresponding quadrature weights. The value of g_k can be approximated similarly.

$$\boxed{g_k = \int_{-1}^{1} g(x) P_k^{\alpha/2,\,\alpha/2}(x)\,dx \approx \sum_{j=0}^{N} g(\mathbf{x}_i) P_k^{\alpha/2,\,\alpha/2}(\mathbf{x}_i)\mathbf{w}_i.}$$

Remark 5.1 *We can reduce the amount of operations as in obtaining* (3.14)*. Note that* $M_{n,k} = M_{k,n}$*. Moreover, when* $n+k$ *is odd,* $M_{n,k} = 0$*. In fact,* $P_n^{\alpha/2,\alpha/2}$*'s are odd functions when* n *is odd and are even functions* n *is even.*

To solve the ordinary differential equation (5.3), we first write (5.3) in a matrix form. Denote $\vec{a}(t) = (a_0(t), a_1(t), a_2(t), \ldots, a_N(t))^T$ and $\vec{g} = (g_0, g_1, g_2, \ldots. g_N)^T$. Denote also the diagonal matrix of $e_k(\alpha/2)$ by E and the matrix of $S_{n,k}$ by S'. The resulting linear system is

$$\boxed{S'\vec{a}(t) + \frac{d}{dt}\, E\vec{a}(t) = \lambda E\vec{a}(t) + \vec{g},} \tag{5.5}$$

where

$$E = \begin{bmatrix} e_0 & 0 & 0 & \ldots & 0 \\ 0 & e_1 & 0 & \ldots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \ldots & e_N \end{bmatrix}, \quad S' = \begin{bmatrix} S'_{0,0} & S'_{1,0} & S'_{2,0} & \ldots & S'_{N,0} \\ S'_{0,1} & S'_{1,1} & S'_{2,1} & \ldots & S'_{N,1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ S'_{0,N} & S'_{1,N} & S'_{2,N} & \ldots & S'_{N,N} \end{bmatrix}.$$

Let $y(t)$=$E\vec{a}$ and we have

$$\frac{d}{dt} y(t) = f(t, y), \quad y(0) = E\vec{a}(0),$$

where

$$f(t, y) = E\vec{a} + \vec{g} - S'\vec{a} = (E - S')E^{-1}y(t) + \vec{g}. \tag{5.6}$$

Now we explain here how $y(0)$ is computed. According to the equation (5.2), we find the value of $a_k(0)$,

$$a_k(0) = \frac{\int_{-1}^{1} u(0, x) P_k^{\alpha/2,\,\alpha/2}(x)\,dx}{\int_{-1}^{1} (P_k^{\alpha/2,\alpha/2}(x))^2 (1 - x^2)^{\alpha/2}},$$

where the integration can be approximated by a proper Gauss-Jacobi quadrature rule. Then we get the initial value y_0.

5.1.1 Discretization in time

We use forward and backward Euler method, midpoint methods, and RK4 method in time to find approximation of $y(t)$ at t_{n+1} and compare the numerical results. We provide the following discrete schemes of forward Euler method, backward Euler method, midpoint method and RK4 method.

Forward Euler method

Let $t_n = t_0 + h$. Now, one step of the Euler method from t_n to $t_{n+1} = t_n + h$ is

$$y_{n+1} = y_n + hf(t_n, y_n).$$

Specifically, we have

$$\begin{aligned}
y_0 &= Ea_k(0) \quad , t_0 = 0, \\
y_1 &= y_0 + hf(t_0, y_0) = y_0 + h((E - S')E^{-1}y(0) + \vec{g}), \\
y_2 &= y_0 + hf(t_1, y_1) = y_1 + h((E - S')E^{-1}y(1) + \vec{g}), \\
... &\quad ... \\
y_{n+1} &= y_n + hf(t_n, y_n) = y_n + h((E - S')E^{-1}y(n) + \vec{g}).
\end{aligned}$$

Backward Euler method

Let $t_n = t_0 + h$. Now, one step of the backward Euler method from t_n to $t_{n+1} = t_n + h$ is

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) = y_n + h(\vec{g} - SE^{-1}y_{n+1}) + y_{n+1}).$$

Simplifying it, we have

$$y_{n+1} = (I + ShE^{-1} - h)^{-1}(y_n + h\vec{g}), \quad y_0 = Ea_k(0).$$

Here I is the identity matrix.

Midpoint method

Let $t_n = t_0 + h$. Now, one step of the midpoint Euler method from t_n to $t_{n+1} = t_n + h$ is

$$y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, \frac{1}{2}(y_n + y_{n+1})\right) = y_n + h(g - SE^{-1}\frac{1}{2}(y_n + y_{n+1}) + \frac{1}{2}(y_n + y_{n+1})).$$

Simplifying it, we have

$$y_{n+1} = (I + \frac{1}{2}hSE^{-1} - \frac{h}{2})^{-1}((I - \frac{1}{2}hSE^{-1} + \frac{h}{2})y_n + h\vec{g}).$$

RK4 method

Let $t_n = t_0 + h$. Now, four steps of the RK4 method from $y_{n+1}= y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ $t_{n+1}= t_n + h$ for n=0,1,2,3,....., using:

$$k_1 = f(t_n, y_n),$$
$$k_2 = f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1),$$
$$k_3 = f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_2),$$
$$k_4 = f(t_n + h, y_n + hk_3).$$

5.1.2 Numerical Results

Example 5.2 (Forward Euler) *Let* $g(x) = \sin(x)$.

The numerical solution at $t = 0.1$ is plotted in Figure 5.1 where $\alpha = 0.4$ and $N = 256$. The convergence order and errors of numerical solutions are calculated in Table 5.1 where $\alpha = 0.4$ and $N = 256$.

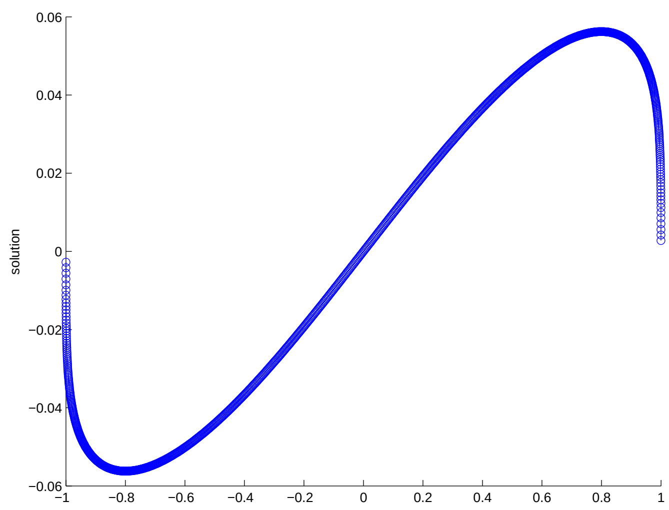

Figure 5.1: Forward Eulerwhen $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0234	0.6159	-	-
8	0.0167	0.5669	-0.4859	-0.1195
16	0.0108	0.4913	-0.6298	-0.2065
32	0.0062	0.3854	-0.7941	-0.3505
64	0.0031	0.2536	-1.0073	-0.6037
128	0.0013	0.1162	-1.3021	-1.1259

Table 5.1: Error table - forward Euler when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

Example 5.3 (Backward Euler) *Let* $g(x) = \sin(x)$.

The numerical solution at $t = 0.1$ is plotted in Figure 5.2 where $\alpha = 0.4$ and $N = 256$. The convergence order and errors of numerical solutions are calculated in Table 5.2 where $\alpha = 0.4$ and $N = 256$.

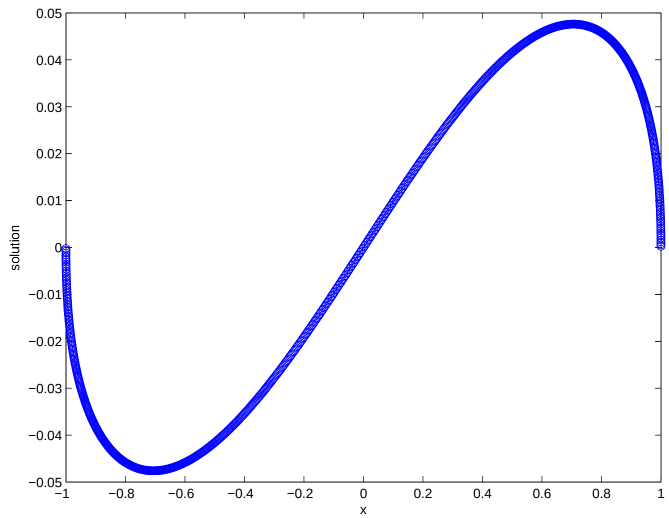

Figure 5.2: Backward Euler when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0113	0.1086	-	-
8	0.0071	0.0.0978	-0.6771	-0.1504
16	0.0039	0.0826	-0.8456	-0.2442
32	0.0020	0.0614	-0.9920	-0.4278
64	8.7975e-04	0.0377	-1.1686	-0.7041
128	3.4847e-04	0.0157	-1.3361	-1.2623

Table 5.2: Error table -backward Euler when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

Example 5.4 (Midpoint) *Let* $g(x) = \sin(x)$.

The numerical solution at $t = 0.1$ by Midpoint Method is plotted in Figure 5.3. The convergence order and errors of numerical solutions are calculated in Table 5.3 where $\alpha = 0.4$ and $N = 256$.

Example 5.5 (RK4) *Let* $g(x) = \sin(x)$.

The numerical solution by RK4 Method is plotted in Figure 5.4. The convergence order and errors of numerical solutions are calculated in Table 5.4 where $\alpha = 0.4$ and $N = 256$.

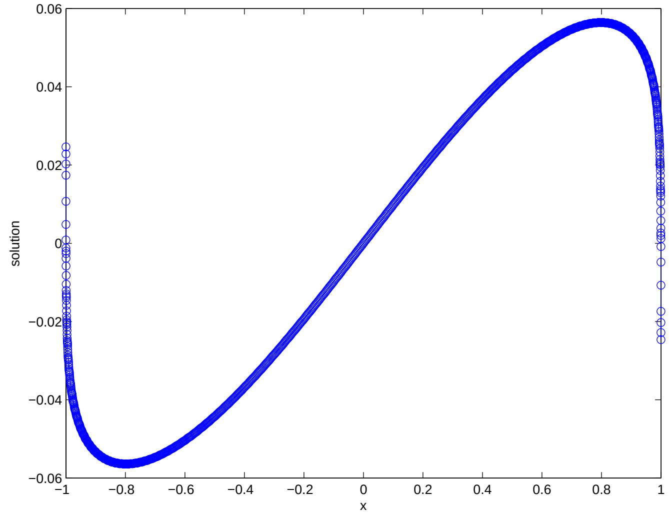

Figure 5.3: Midpoint when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0698	1.3390	-	-
8	0.0617	1.3086	-0.1782	-0.0331
16	0.0517	1.2455	-0.2561	-0.0713
32	0.0400	1.1197	-0.3686	-0.1536
64	0.0276	0.8882	-0.5349	-0.3342
128	0.0156	0.5148	-0.8200	-0.7869

Table 5.3: Error table -Midpoint method when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

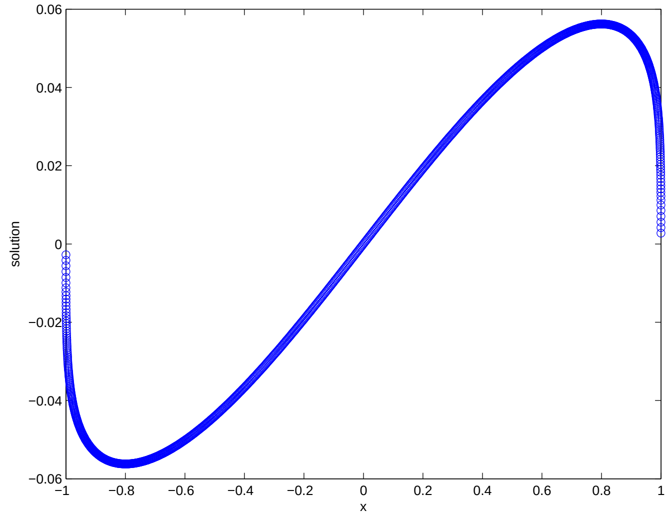

Figure 5.4: RK4 when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0234	0.6159	-	-
8	0.0167	0.5669	-0.4861	-0.1196
16	0.0108	0.4913	-0.6301	-0.2067
32	0.0062	0.3854	-0.7944	-0.3507
64	0.0031	0.2536	-1.0076	-0.6038
128	0.0013	0.1162	-1.3021	-1.1259

Table 5.4: Error table -RK4 when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

Table 5.1 shows that the convergence order is around 1.1. Table 5.2 shows that the convergence order is around 1.2. Table 5.3 shows that the convergence order is around 0.8. Table 5.4 shows that the convergence order is around 1.1. All the convergence order is low and thus the solution is non smooth. But the error tables show that the error is getting smaller with N getting larger.

5.2 u_N for Fractional Equation $(-\Delta)^{\alpha/2}u+\frac{\partial u}{\partial t}=f(u)+g$

Then we consider the following equation in one dimension nonlinear system.

$$(-\Delta)^{\alpha/2}u_N + \frac{\partial u_N}{\partial t} = f(u) + g$$

We again use the approximation (5.2).

From section 5.1 and equation 5.3, we can get the following equation

$$\boxed{\frac{d}{dt}a_k(t)e_k(\alpha/2) + \sum_{n=0}^{N} S_{n,k}a_k(t) = a_k(t)e_k(\alpha/2) + g_k + M_f(t), k = 0, 1, \ldots, N.} \tag{5.7}$$

Here $M_f(t) = \int_{-1}^{1}(1-x^2)^{3\alpha/2}(\sum_{j=0}^{N} a_j(t)P_i^{\alpha/2})^3 P_k^{\alpha/2}\, dx$. Using the forward Euler method, we get

$$\frac{a_k^{n+1} - a_k^n}{\Delta t}e_k(\alpha/2)) + \sum_{n=0}^{N} S_{n,k}a_k^n = a_k^n e_k(\alpha/2) + g_k + M_f^n,$$

where

$$M_f^n = \int_{-1}^{1}(1-x^2)^{3\alpha/2}(\sum_{j=0}^{N} a_j^n P_i^{\alpha/2})^3 P_k^{\alpha/2}\, dx.$$

5.2.1 Numerical Results

Example 5.6 *Assume* $f(u) = u - u^3$*,* $g(x) = \sin(x)$*.*

The errors are plotted in Figure 5.5 where $\alpha = 0.4$ and the reference solution is generated by $N = 256$ and we use $M = 256$ in (3.9). The convergence order and errors of numerical solutions are presented in Table 5.5 where $\alpha = 0.4$ and N=256.

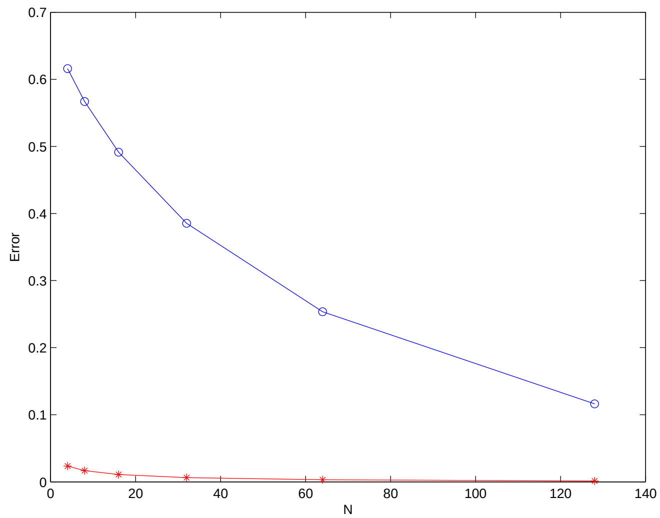

Figure 5.5: Numerical error when $f(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$. The red line represents errors in L_2 and the blue line represents errors in L_∞.

N	Errors in L_2	Errors in L_∞	Convergence Order of L_2	Convergence Order of L_∞
4	0.0234	0.6159	-	-
8	0.0167	0.5669	-0.4865	-0.1196
16	0.0108	0.4913	-0.6299	-0.2066
32	0.0062	0.3853	-0.7941	-0.3505
64	0.0031	0.2536	-1.0074	-0.6037
128	0.0013	0.1162	-1.3021	-1.1259

Table 5.5: Error table when $g(x) = \sin(x)$ with $\alpha = 0.4$ and $N = 256$

From Table 5.5, the accuracy is very low but we can still observe that some convergence order is around 1. The accuracy is slightly improved when $N = 64, 128$.

6 u_N for Fractional Equation u_t=-$i((-\Delta)^{\alpha/2}u+\lambda\left|u^2\right|u)+\Delta u$

In this chapter, we discuss how to solve this MMT equation. More information can be found in reference [2].

We consider the following equation in one dimension for $0 < \alpha < 2$ and $\lambda > 0$:

$$u_t = -i((-\Delta)^{\alpha/2}u + \lambda\left|u^2\right|u) + \Delta u. \tag{6.1}$$

Let $u(x)$=$v(x)$+$iw(x)$, we have:

$$\begin{aligned}
\partial_t(v + iw) &= -i((-\Delta)^{\alpha/2}u + \lambda\left|u^2\right|u) + \Delta u \\
&= -i((-\Delta)^{\alpha/2}(v + iw) + \lambda\left|(v + iw)^2\right|(v + iw))v + \Delta(v + iw) \\
&= -i(-(\Delta)^{\alpha/2})v + (-\Delta)^{\alpha/2}w - i\lambda(v^2 + w^2)v + \lambda(v^2 + w^2)w + \Delta v + i\Delta w.
\end{aligned}$$

Now we have

$$\begin{aligned}
v_t &= (-\Delta)^{\alpha/2}w + \lambda(v^2 + w^2)w + \Delta v, \\
w_t &= -(-\Delta)^{\alpha/2}v - \lambda(v^2 + w^2)v + \Delta w.
\end{aligned}$$

Similar to the approximation in (5.2), we let

$$v_n(t, x) = \sum_{n=0}^{N} z_n(t)h_n(x), \quad w_N(t, x) = \sum_{n=0}^{N} b_n(t)h_n(x).$$

But we are in trouble to calculate of $\Delta(h_n(x))$. Here we take the test function as $h_k(x)$

$$\int_{-1}^{1} \Delta h_n(x)h_k(x)\,dx = -\int_{-1}^{1} \partial_x h_n(x)\partial_x h_k(x)\,dx.$$

But the term $\int_{-1}^{1} h_n(x)h_k(x)dx$ lead to a singular matrix as stated in Chapter 5. It seems that we need to take the test function as $P_k^{\alpha/2,\alpha/2}(x)$. Then we have

$$\int_{-1}^{1} \Delta h_n(x)P_k^{\alpha/2,\alpha/2}(x)dx = \partial_x h_n(x)P_k^{\alpha/2,\alpha/2}(x)\mid_{-1}^{1} - \int_{-1}^{1} \partial_x h_n(x)\partial_x P_k^{\alpha/2,\alpha/2}(x)\,dx.$$

Notice that $\partial_x h_n(x)P_k^{\alpha/2,\alpha/2}(x)$ is not well defined at $x = \pm 1$ when $0 < \alpha/2 < 1$. We are in a position to choose a working test function. But we don't have a solution yet at the time of writing this report.

7 Conclusion

In conclusion, this project was to numerically solve fractional Laplace equations and related equations, where the fractional Laplacian was defined as

$$(-\Delta)^{\alpha/2}u(x) = c_{d,\alpha}\int_{\mathbb{R}^d}\frac{u(x)-u(y)}{|x-y|^{d+\alpha}}\,dy.$$

The major numerical methods we used were finite difference in time and spectral methods in space. We considered the following situations to determine solutions of the form u_N which read

$$u_N = \sum_{n=0}^{N} a_n h_n(x) \in (1-x^2)^{\alpha/2}\mathbb{P}_N, \quad h_n = (1-x^2)^{\alpha/2}P_n^{\alpha/2,\,\alpha/2}(x).$$

We solved two linear fractional differential equations with fractional Laplacian in Chapter 3: $(-\Delta)^{\alpha/2}u + \lambda u = g$ with $\lambda = 0$ and $\lambda \neq 0$. We gave two numerical examples and discuss the convergence order of our method, when $g(x)$=$\sin(x)$ and $g(x)$=$|\sin(x)|$. We found the relationship between λ, $g(x)$and convergence order: We found that

- When $\lambda = 0$ and g was smooth, the convergence order was high and the solution $(u(x)/(1-x^2)^{\alpha/2})$ was smooth.
- When $\lambda = 0$ and g was not smooth, the convergence order was high and the solution $(u(x)/(1-x^2)^{\alpha/2})$ was not smooth.
- When $\lambda = 1$ and $g = \sin(x)$ or $f = |\sin(x)|$, the convergence order was low and the solution $(u(x)/(1-x^2)^{\alpha/2})$ was not smooth.

Moreover, from error tables with convergence order, we found that when α was getting larger, the convergence order was getting larger. This suggested that with a large α, the solution was smoother than that from a small α.

We solved one nonlinear fractional differential equation with fractional Laplacian in chapter 4: $(-\Delta)^{\alpha/2}u = f(u) + g(x)$. Then, we gave two examples and discussed the convergence order of our method, when $g(x)$=$\sin(x)$ and $g(x)$=$|\sin(x)|$ and $f(u) = u - u^3$. We applied a fixed-point iteration method instead of Newton's method to solve the resulting nonlinear systems since Newton's method leads to solutions with high oscillation. We were able to find a good initial guess for Newton's method.

We found that when g was smooth,the convergence order was very low and thus the solution was not smooth. However, when α was lager, the numerical results suggested that solutions were smoother.

We solved two fractional differential equation in time dependent with fractional Laplacian in Chapter 5: $(-\Delta)^{\alpha/2}u + \frac{\partial u}{\partial t} = f(u) + \sin(x)$, where $f(u) = u$ or $f(u) = u - u^3$. We used forward and backward Euler methods, midpoint method and RK4 method for the linear equation and forward Euler method for the nonlinear equation. All numerical results showed a very low convergence order which indicated the solution was not smooth enough.

We discussed how to solve our ultimate goal problem which is a MMT model [2] based on our previous numerical results. We discussed the difficulty of the problem and some potential solution. We believed this report had provided enough knowledge for future students to solve MMT model and other fractional models.

Appendices

A Basic Functions

Gamma Function [10]

Some essential knowledge is required to enter the world of fractional calculus. One of the basic functions is the Gamma function. It is a special function which is the factorial n! and allows n to be real or even complex number except non-positive integer.It is defined by the integral

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x}\, dx. \tag{A.1}$$

Here is a useful properties of gamma function.

$$\Gamma(t+1) = t\Gamma(t). \tag{A.2}$$

(A.2) can be proved simply using integration by parts,

$$\Gamma(n) = (n-1)! \tag{A.3}$$

for all positive integer n.

Beta Function [10]

Euler's Beta function is like the family of Gamma function and also essential to the fractional calculus. It is also a spectial function defined by

$$B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}\, dt. \tag{A.4}$$

It also can be represented by gamma function.

$$B(x,y) = \frac{\Gamma(x)\,\Gamma(y)}{\Gamma(x+y)} \tag{A.5}$$

We can find out from (A.5) that Beta function is symmetric

$$B(x,y) = B(y,x). \tag{A.6}$$

B Proof of Lemma 2.2

Proof. The conclusion can be shown by induction. By Lemma 2.1, for $n = 0$,

$$(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}] = -\Gamma(\alpha+1),$$

and for $n = 1$,

$$(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}(\alpha+1)x] = -\Gamma(\alpha+2)(\alpha+1)x.$$

Now suppose that $k \le n$ the relation holds then when $k = n+1$, we want

$$(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}P_{n+1}^{\alpha/2,\,\alpha/2}(x)] = A_{n+1,\alpha}P_{n+1}^{\alpha/2,\,\alpha/2}(x).$$

By integration-by-parts formula [4], for any $j \le n$, we have

$$\begin{aligned}
& \int_{-1}^{1}(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}P_{n+1}^{\alpha/2,\,\alpha/2}(x)]P_j^{\alpha/2,\,\alpha/2}(1-x^2)^{\alpha/2})\,dx \\
= & \int_{-1}^{1}(1-x^2)^{\alpha/2}P_{n+1}^{\alpha/2,\,\alpha/2}(x)(-\Delta)^{\alpha/2}[P_j^{\alpha/2,\,\alpha/2}(1-x^2)^{\alpha/2}])\,dx \\
= & A_{j,\alpha}((1-x^2)^{\alpha/2}\int_{-1}^{1}P_{n+1}^{\alpha/2,\,\alpha/2},P_j^{\alpha/2,\,\alpha/2}\,dx = 0.
\end{aligned}$$

Here we have used the induction assumption.

By Lemma 2.1, $(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}P_{n+1}^{\alpha/2,\,\alpha/2}(x)$ is a polynomial of order $n+1$, then we have

$$(-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}P_{n+1}^{\alpha/2,\,\alpha/2}(x)] = C_{n+1,\alpha}P_{n+1}^{\alpha/2,\,\alpha/2}(x). \tag{B.1}$$

where $C_{n+1,\alpha}$ is a constant depending on $n+1$ and α. Then we compare the coefficients of leading-order term over both sides and obtain the conclusion at $k = n+1$.

The constant can be found by the following procedure.

Suppose that $m = n+1 = 2k$. Then we can compare the coefficients of the leading order term in a power series representation of $P_m^{\alpha/2,\,\alpha/2}(x)$. The coefficient can be obtained by

$$\partial_x^n P_m^{\alpha/2,\,\alpha/2}(x) = \frac{\Gamma(\alpha+2m+1)}{2^m\Gamma(\alpha+m+1)}P_0^{\alpha/2+m,\alpha/2+m}(x) = \frac{\Gamma(\alpha+2m+1)}{2^n\Gamma(\alpha+m+1)}. \tag{B.2}$$

While the coefficient of leading-order tem of LHS in (B.1) is the coefficient of leading order term of

$$(-1)^k\frac{\Gamma(\alpha+2m+1)}{2^n\Gamma(\alpha+m+1)} \times (-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}(1-x^2)^k],$$

which is ($p = \alpha/2 + k$)

$$(-1)^k\frac{\Gamma(\alpha+2m+1)}{2^m\Gamma(\alpha+m+1)}c_{1,\alpha}B(-\alpha/2,\,p+1)_2F_1(\frac{\alpha+1}{2},-p+\frac{\alpha}{2};\frac{1}{2};x^2)$$

So it requires to check the coefficient of leading-order term of

$c_{1,\alpha}B(-\alpha/2,\, p+1)_2F_1(\frac{\alpha+1}{2}, -p+\frac{\alpha}{2}; \frac{1}{2}; x^2)$.

$$
\begin{aligned}
& \frac{2^\alpha \Gamma(\frac{\alpha+d}{2})}{\pi^{d/2}\,|\Gamma(-\alpha/2)|} \frac{\Gamma(-\alpha/2)\Gamma(p+1)}{\Gamma(p-\alpha/2+1)} \frac{(\frac{(\alpha+1)}{2})_k(-p+\alpha/2)_k}{(1/2)_k k!} \\
& \frac{2^\alpha \Gamma(\frac{\alpha+d}{2})}{\pi^{d/2}\,|\Gamma(-\alpha/2)|} \frac{\Gamma(-\alpha/2)\Gamma(p+1)}{\Gamma(p-\alpha/2+1)} \frac{(\frac{(\alpha+1)}{2})_k(-k)_k}{(1/2)_k k!} \\
= \quad & \frac{2^\alpha \Gamma(\frac{\alpha+d}{2})}{\pi^{d/2}\,|\Gamma(-\alpha/2)|} \frac{\Gamma(-\alpha/2)\Gamma(p+1)}{\Gamma(k+1)} \frac{\Gamma(\frac{\alpha+1}{2}+k)\Gamma(\frac{1}{2})}{\Gamma(\frac{\alpha+1}{2})\Gamma(\frac{1}{2}+k)} \frac{(-k)_k}{k!} \\
= \quad & \frac{2^\alpha}{\pi^{d/2}\,|\Gamma(-\alpha/2)|} \frac{\Gamma(-\alpha/2)\Gamma(p+1)}{\Gamma(k+1)} \frac{\Gamma(\frac{\alpha+1}{2}+k)\Gamma(\frac{1}{2})}{\Gamma(\frac{1}{2}+k)} (-1)^k \\
= \quad & \frac{(-1)^{k+1} 2^\alpha \Gamma(\frac{1}{2})}{\pi^{d/2}} \frac{\Gamma(p+1)\Gamma(\frac{\alpha+1}{2}+k)}{\Gamma(k+1)\Gamma(\frac{1}{2}+k)}.
\end{aligned}
$$

Then by the following *the duplication formula*

$$\Gamma(z)\Gamma\left(z+\tfrac{1}{2}\right) = 2^{1-2z}\,\sqrt{\pi}\,\Gamma(2z),$$

we have

$$
\begin{aligned}
& \frac{2^\alpha \Gamma(\frac{\alpha+d}{2})}{\pi^{d/2}\,|\Gamma(-\alpha/2)|} \frac{\Gamma(-\alpha/2)\Gamma(p+1)}{\Gamma(p-\alpha/2+1)} \frac{(\frac{(\alpha+1)}{2})_k(-p+\alpha/2)_k}{(1/2)_k k!} \\
= \quad & \frac{(-1)^{k+1} 2^\alpha \Gamma(\frac{1}{2})}{\pi^{d/2}} \frac{\Gamma(p+1)\Gamma(\frac{\alpha+1}{2}+k)}{\Gamma(k+1)\Gamma(\frac{1}{2}+k)} \\
= \quad & \frac{(-1)^{k+1} 2^\alpha \Gamma(\frac{1}{2})}{\pi^{d/2}} \frac{\sqrt{\pi} 2^{-\alpha-2k}\Gamma(\alpha+1+2k)}{2^{-2k}\sqrt{\pi}\Gamma(2k+1)} \\
= \quad & (-1)^{k+1} \frac{\Gamma(\alpha+1+m)}{m!}.
\end{aligned}
$$

Comparing the coefficients of leading order terms of

$$(-1)^k \frac{\Gamma(\alpha+2m+1)}{2^n\Gamma(\alpha+m+1)} \times (-\Delta)^{\alpha/2}[(1-x^2)^{\alpha/2}(1-x^2)_+^k] \text{ and } C_{n+1,\alpha}.$$

Similarly, we can have the same conclusion when $m = n+1 = 2k+1$.

C Numerical Methods

Forward Euler Method [7]

Suppose that we want to approximate the solution of the initial value problem $y'(t)=f(t, y(t))$, $y(t_0)=y_0$. Choose a value h for the size of every step and set $t_n = t_0 + nh$. Now, one step of the Euler method from t_n to $t_{n+1} = t_n + h$ is $y_{n+1} = y_n + hf(t_n, y_n)$.

Backward Euler Method [7]

Consider the ordinary differential equation $\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y)$ with initial value $y(t_0) = y_0$. Choose a value h for the size of every step and set $t_n = t_0 + nh$. The backward Euler method computes the approximations using $y_{k+1} = y_k + hf(t_{k+1}, y_{k+1})$. This differs from the (forward) Euler method in that the latter uses $f(t_k, y_k)$ in place of $f(t_{k+1}, y_{k+1})$.

Midpoint Method [9]

The midpoint method is a one-step method for solving the ordinary differential equation: $y'(t) = f(t, y(t)), \quad y(t_0) = y_0$. The method by: $y_{n+1} = y_n + hf\left(t_n + \frac{h}{2}, \frac{1}{2}(y_n + y_{n+1})\right)$, for $n = 0, 1, 2, \ldots$. Here, h is the 'step size' - a small positive number, $t_n = t_0 + nh$, and y_n is the computed approximate value of $y(t_n)$.

Runge - Kutta Method [13]

The most famous one is "RK4" method. Let an initial value problem be specified as follows.$\dot{y} = f(t, y), \quad y(t_0) = y_0$. The initial time t_0 and initial value y_0 is given. A step-size h_0 is picked and define

$$y_{n+1} = y_n + \tfrac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right),$$

for $n = 0, 1, 2, 3, \ldots,$ using

$$\begin{aligned}
k_1 &= f(t_n, y_n),\\
k_2 &= f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_1),\\
k_3 &= f(t_n + \tfrac{h}{2}, y_n + \tfrac{h}{2}k_2),\\
k_4 &= f(t_n + h, y_n + hk_3).
\end{aligned}$$

Newton's Method (Newton-Raphson method) [5]

Suppose $f(x^*) = 0 \iff 0 = f(x) + f'(x)(x - x^*)$ if $|x - x^*|$ is small. Then there is the following scheme when $f'(x) \neq 0$,

$$0 = f(x_k) + f'(x_k)(x_k - x_{k+1}) \iff x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Fixed Point Iteration [6]

Let g be a continuous function on the interval $[a, b]$. If $g(x) \in [a, b]$ for each $x \in [a, b]$, then g has a fixed point in [a, b]. Furthermore, if g is differentiable on (a, b) and there exists a constant $k \leq 1$ such that $|g'(x)| \leq k$, $x \in (a, b)$, then g has exactly one fixed point in $[a, b]$.

The following algorithm computes a number $x^* \in (a, b)$ that is a solution to the equation $g(x)$ $= x$. Choose an initial guess $x_0 \in$ [a, b]. For $k = 0, 1, \ldots 2, \cdots$, do x_{k+1} $=g(x_k)$ iteration until $|x_{k+1} - x_k| < \epsilon$. Then let $x^* = x_{k+1}$.

References

[1] Michele Caputo. Linear models of dissipation whose q is almost frequency independent ii. *Geophysical Journal International*, 13(5):529–539, 1967.

[2] Will Cousins and Themistoklis P Sapsis. Quantification and prediction of extreme events in a one-dimensional nonlinear dispersive wave model. *Physica D: Nonlinear Phenomena*, 280:48–58, 2014.

[3] Bartłomiej Dyda. Fractional calculus for power functions and eigenvalues of the fractional Laplacian. *Fract. Calc. Appl. Anal.*, 15(4):536–555, 2012.

[4] Qing-Yang Guan. Integration by parts formula for regional fractional Laplacian. *Comm. Math. Phys.*, 266(2):289–329, 2006.

[5] E Eric Kalu et al. *Numerical Methods with Applications: Abridged.* Lulu. com, 2008.

[6] Jim Lambers. Fixed point iteration. *University of Southern Mississippi, Fall*, 2009.

[7] John Denholm Lambert. *Numerical Methods for Ordinary Differential.* Wiley, 1991.

[8] JD Munkhammar. Riemann-liouville fractional derivatives and the taylor-riemann series. *UUDM project report*, 7:1–18, 2004.

[9] Richard S Palais and Robert Andrew Palais. *Differential equations, mechanics, and computation*, volume 51. American Mathematical Soc., 2009.

[10] Igor Podlubny. *Fractional differential equations.* Academic Press, Inc., San Diego, CA, 1999.

[11] Anthony Ralston and Philip Rabinowitz. *A first course in numerical analysis.* Courier Corporation, 2012.

[12] Stefan G. Samko, Anatoly A. Kilbas, and Oleg I. Marichev. *Fractional integrals and derivatives.* Gordon and Breach Science Publishers, Yverdon, 1993. Theory and applications, Edited and with a foreword by S. M. Nikol′skiĭ, Translated from the 1987 Russian original, Revised by the authors.

[13] Endre Süli and David F Mayers. *An introduction to numerical analysis.* Cambridge university press, 2003.