

University of Chester



**This work has been submitted to ChesterRep – the University of Chester’s
online research repository**

<http://chesterrep.openrepository.com>

Author(s): Kizito Ugochukwu Nwajeri

Title: Stability regions of numerical methods for solving fractional differential equations

Date: September 2012

Originally published as: University of Chester MSc dissertation

Example citation: Nwajeri, K. U. (2012). Stability regions of numerical methods for solving fractional differential equations. (Unpublished master’s thesis). University of Chester, United Kingdom.

Version of item: Submitted version

Available at: <http://hdl.handle.net/10034/295582>

Stability regions of numerical methods for solving fractional differential equations

Kizito Ugochukwu Nwajeri



**A dissertation submitted in partial requirement of the
University of Chester for the degree of Master of Science in
Applied Mathematics.**

September, 2012

Abstract

This dissertation deals with proper consideration of stability regions of well known numerical methods for solving fractional differential equations. It is based on the algorithm by Diethelm [15], predictor-corrector algorithm by Garrappa [31] and the convolution quadrature proposed by Lubich [3]. Initially, we considered the stability regions of numerical methods for solving ordinary differential equation using boundary locus method as a stepping stone of understanding the subject matter in Chapter 4. We extend the idea to the fractional differential equation in the following chapter and conclude that each stability regions of the numerical methods differs because of their differences in weights. They are illustrated by a number of graphs.

Key words. Fractional differential equations, ordinary differential equations, finite difference method, stability regions, Mittag-Leffler function, Riemann-Liouville fractional derivatives, Caputo fractional derivatives.

Declaration. I declare that this work is original and has not been previously submitted for any academic purposes.

Signed:

Date:

Acknowledgements

I am sincerely and heartily grateful to almighty God for His protection and guidance in my everyday activities. I am sure all my achievements so far would have not been possible without His wonderful love and care.

Above all, this dissertation would have not been possible without the constant help, support and ever patience of my principal supervisor, Dr. Yubin Yan, not to mention his dynamic advice and unsurpassed knowledge of mathematics. The good advice, support and friendship of my programme leader, Dr. Jason Roberts and Dr. Charles Simpson have been invaluable on both an academic and a personal level for which I am extremely grateful.

I would like to show my profound appreciation to Prof. Neville J. Ford for his wonderful support and friendship to the mathematics department. Unfortunately, we have not met in person but his contribution on the materials I used for my dissertation has not gone unnoticed.

Furthermore, I would like to thank specially my brother Mr. & Mrs. Stanley Nwajeri for his personal support and great patience at all times. My parents, brothers, sisters, girl friend, Miss Alliance and little angel, Brithany have given me their unequivocal support throughout, as always for which my mere expression of thanks likewise does not suffice.

I would also like to extend my acknowledgements to all staffs of department of mathematics for their wonderful contribution during my MSc. Programme. I also acknowledge the academic and technical support of University of Chester and its staff. The library and computer facilities of this great campus have been indispensable.

Lastly, but not the least, I would like thank all and sundry who have contributed on one way or the other in this dissertation.

Contents

1	Introduction	6
2	Preliminaries and fundamentals	9
2.1	Gamma function	9
2.2	Beta function	10
2.3	The Mittag-Leffler function	10
2.4	Definitions of fractional order derivative	11
2.5	Existence and Uniqueness of the solution	13
2.6	Analytical solutions for FDEs	17
3	Numerical methods for FDEs	19
3.1	Diethelms method	20
3.2	Predictor-Corrector method	27
3.3	Lubich's method	31
4	Stability regions of numerical methods for solving ODEs.	34
4.1	One-step methods	34
4.1.1	Explicit second order Runge-Kutta method	35
4.1.2	Implicit second order-Runge-Kutta method	39
4.1.3	Explicit third-stage Runge-Kutta method	42
4.1.4	Explicit fourth-stage Runge-Kutta method	43
4.2	Multistep methods	45
4.2.1	Explicit Adams-Bashforth methods	46
4.2.2	Implicit Adams-Moulton methods	46
4.2.3	Stability regions of multistep methods	48

5	Stability regions of numerical methods for solving FDEs	53
5.1	Lubich's convolution quadrature method [3].	53
5.2	Garrappa predictor-corrector algorithm [31].	58
5.3	Discrete stability polynomial method	60
5.3.1	Diethelm's method	61
5.3.2	Grünwald method	62
5.3.3	Lubich's method	64
6	Further works	65
7	Conclusion	68
8	Appendix	70
8.1	Figure 5.1 MATLAB programme	70
8.2	Figure 5.2 MATLAB programme	72
8.3	Figure 5.3 MATLAB programme	74
8.4	Figure 5.4 MATLAB programme	77

List of Figures

4.1	Stability region (shaded) of second order Runge-Kutta method	39
4.2	Stability region (shaded) of implicit second order Runge-Kutta method . .	41
4.3	Stability region (shaded) of third order Runge-Kutta method	43
4.4	Stability region (shaded) of fourth order Runge-Kutta method	45
4.5	Stability region (shaded) of second order Adams-Bashforth method.	51
5.1	Stability region of Garrappa algorithm	60
5.2	Stability region of Diethelm method	62
5.3	Stability region of Grünwald method	63
5.4	Stability region of Lubich method	64

Chapter 1

Introduction

The main purpose of this dissertation is to investigate the stability regions of numerical methods for solving fractional differential equations (FDEs).

Fractional derivatives and fractional integrals may not be new in the household subject area of mathematics, but have drawn a huge interest in recent years because of its vast areas of application. As mentioned in [32], the list of great mathematicians who have made significant contribution to FDEs includes P.S. Laplace (1812), J.B.J. Fourier (1822), N.H. Abel (1823-1826), J. Liouville (1832-1873), B. Riemann (1847), H. Holmgren (1865-1867), A.K. Grunward (1867-1872), A.V. Letnikov (1868-1872), H. Laurent (1884), P.A. Nekrassov (1888), A. Krug (1890), J. Hadamard (1892), O. Heaviside (1892-1912), S. Pincherle (1902), G.H. Hardy and J.E. Littlewood (1917-1928), H. Weyl (1917), P. Levy (1923), A. Marchaud (1927), H.T. Davis (1924-1936), A. Zygmund (1935-1945), E.R. Love (1938-1996), A Erdelyi (1939-1965), H. Kober (1940), D.V. Widder (1941), M. Riesz (1949).

However, FDEs have been used successfully to model frequency dependent damping behaviour of many viscoelastic materials [30], electrochemical process [27], a radial flow problem [28]. Many papers have also been involved in illustrating the application of FDEs in dielectric polarization [9], control of viscoelastic structures [35] etc..

Several analytical methods have been proposed to solve FDEs, for example Laplace transform, Mellin transform, Fourier transform, modal synthesis, eigenvector expansion

etc.. Most of these methods can only be applicable to solve linear FDEs but cannot be applied in non-linear FDEs.

Recent developments have seen a tremendous interest in approximating numerical solution for FDEs which can be effectively applied to both linear and non-linear FDEs (see Diethelm [15, 18], Lubich [5]). As pointed by Diethelm [18], most of the techniques of solving initial value problems (IVPs) of FDEs are equivalent to Volterra integral equation. Therefore the numerical schemes for Volterra integral equations can be applied to FDEs. Lubich [3, 4] took the advantage for the fact FDEs can be converted into Volterra integral equations to produce a stability analysis of convolution quadratures and fractional linear multistep methods for Abel-Volterra equations of the second kind. Diethelm et al. [19] presented extrapolation method for numerical solution of FDEs. These was based on the algorithm of [15] where the application of extrapolation were justified. The algorithm used the Hadamard finite-part integral stated in [20] to determine the weights of the numerical solution. Diethelm et al. [18] presented a predictor-corrector method for solution of numerical FDEs. It was demonstrated that Adam-Moulton predictor-corrector method of ODEs can be extended to predictor-corrector method of FDEs and produced a detailed error analysis for fractional Adams method.

Garrappa [31] considered the linear stability of predictor-corrector algorithms for FDEs. He took the advantage that FDEs can be converted into Volterra integral equations and used the idea in [4] to establish the stability analysis for convolution quadrature. Similar approach were used by Galeone and Garrappa [26] to produce the stability for an explicit methods for FDEs. There are other papers which deals with the numerical methods for FDEs, for example, multi-order FDEs and their numerical solution [22], the numerical solution of linear multi-term FDEs: systems of equations [12], numerical analysis for distributed-order differential equations [21], numerical treatment of differential equations of fractional order [25], some numerical methods of fractional calculus [33], stability and convergence of the difference methods for the space-time fractional advection-diffusion equation [8] etc..

In this dissertation, we will consider the stability regions of some numerical methods for solving FDEs. As with ordinary differential equations (ODEs), numerical method is stable if small changes or perturbation in the initial conditions produce correspondingly small changes in a numerical approximations. In other words, a stable method is one whose results depend continuously on the initial values.

This dissertation is organised as follows. In Chapter 2, some preliminaries and fundamentals for FDEs are introduced. Some numerical methods of linear FDE in Chapter 3 are investigated. The stability regions of numerical methods for ODEs are introduced in Chapter 4. In Chapter 5, we investigate the stability regions of numerical methods for FDEs. In Chapters 6 and 7, we introduce some future works and draw some conclusion respectively. Finally, we present all the MATLAB codes for our Chapter 5 experiments in appendix.

Chapter 2

Preliminaries and fundamentals

In this chapter we introduce some basic knowledge for fractional differential equations.

2.1 Gamma function

According to Podlubny [11], Euler's gamma function $\Gamma(z)$ is one of the main functions of the FDEs which generalizes the factorial $n!$ and allows n to take non-integer and complex values.

Definition 2.1.1. The gamma function $\Gamma(z)$ is defined by the integral

$$\Gamma(z) = \int_0^{\infty} e^{-t} t^{z-1} dt, \quad \operatorname{Re}(z) > 0.$$

Furthermore, in order to give a reasonable account of this function, we state the properties of this function from Podlubny [11] and Diethelm [16]. It is shown that gamma function satisfies the following functional equation

$$\Gamma(z+1) = z\Gamma(z) \tag{2.1}$$

Here $\Gamma(1) = 1$ and therefore by applying (2.1) gives for $z = 1, 2, 3, \dots$ i.e. $\Gamma(2) = 1!, \Gamma(3) = 2!, \Gamma(4) = 3!, \dots, \Gamma(n+1) = n!$.

Another important properties of gamma function is that it has simple poles at the points $z = -n$, where $n = 0, 1, 2, \dots$

2.2 Beta function

Definition 2.2.1. The beta function which has an upper limit one and lower limit zero is defined by [11, 14] the following expression:

$$B(z, w) = \int_0^1 t^{z-1}(1-t)^{w-1} dt, \quad \operatorname{Re}(z) > 0, \quad \operatorname{Re}(w) > 0.$$

From this definition, we can use the convolution theorem of Laplace transforms to establish that the Beta function can be expressed in terms of Gamma functions. This is possible by replacing the constant one in the upper limit of integration in definition (2.2.1) and the one in the second term with v . It is effectively possible now to interpret the Beta function as a convolution integral involving two power functions v^{z-1} and v^{w-1} such that

$$B(z, w) = \int_0^v t^{z-1}(v-t)^{w-1} dt, \quad \operatorname{Re}(z) > 0, \quad \operatorname{Re}(w) > 0.$$

Then taking the Laplace transform of the integral to obtain

$$\begin{aligned} \ell\{B(z, w)(v)\} &= \ell\{v^{z-1}\}\ell\{v^{w-1}\} \\ &= \frac{\Gamma(z)}{s^z} \frac{\Gamma(w)}{s^w} = \frac{\Gamma(z)\Gamma(w)}{s^{z+w}} \end{aligned}$$

Now taking the inverse Laplace transform gives

$$\ell^{-1}\{B(z, w)(v)\} = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)} v^{z+w-1}$$

and setting $v = 1$, we obtain following result

$$B(z, w) = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)}$$

which gives the relation between gamma and beta functions.

2.3 The Mittag-Leffler function

Definition 2.3.1. The Mittag-Leffler function is defined by [10]

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + 1)}, \quad \alpha \in \mathbb{C}, \quad z \in \mathbb{C}, \quad \operatorname{Re}(\alpha) > 0. \quad (2.2)$$

The general Mittag-Leffler function is defined by

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}, \quad \alpha, \beta \in \mathbb{C}, \quad z \in \mathbb{C}, \quad \operatorname{Re}(\alpha) > 0, \quad \operatorname{Re}(\beta) > 0. \quad (2.3)$$

where \mathbb{C} is the set of complex numbers. Prabhaker (1971) defined a generalized Mittag-Leffler function by

$$E_{\alpha,\beta}^v(z) = \sum_{k=0}^{\infty} \frac{(v)z^k}{\Gamma(\alpha k + \beta)k!}$$

where $\alpha, \beta, v \in \mathbb{C}$, $\operatorname{Re}(\alpha) > 0$ and $E_{\alpha,\beta}^v(z)$ is an entire function of order $\operatorname{Re}(\alpha)$.

2.4 Definitions of fractional order derivative

In this section, we will introduce definitions of fractional derivatives and integrals.

Definition 2.4.1. Gorenflo and Manardi [32] defined the Cauchy formula by

$$J^n f(t) = \frac{1}{(n-1)!} \int_0^t (t-\tau)^{n-1} f(\tau) d\tau, \quad t > 0, \quad n \in \mathbb{N}.$$

where \mathbb{N} is the set of positive integers. Extending from positive integers values of index to any positive real values leads to the definition of *Riemann-Liouville integral of fractional order*, with $\alpha > 0$:

$$J^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau, \quad t > 0, \quad \alpha \in \mathbb{R}. \quad (2.4)$$

where \mathbb{R} is the set of positive real numbers. In particular where $\alpha = 1$, we have

$$Jf(t) = \frac{1}{\Gamma(\alpha)} \int_0^t f(\tau) d\tau, \quad t > 0$$

assuming that function f is well behaved sufficiently.

Moreover, we can write the integral semi-group property as follows according to [32]

$$J^\alpha J^\beta = J^{\alpha+\beta}, \quad \alpha, \beta \geq 0$$

which gives the commutative property, $J^\beta J^\alpha = J^\alpha J^\beta$, and the effect of the operator J^α on the power functions

$$J^\alpha t^\beta = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1+\alpha)} t^{\beta+\alpha}, \quad \alpha > 0, \quad \beta > -1, \quad t > 0.$$

Also, the formula $\alpha \geq 0$,

$$D^\alpha J^\alpha = I$$

which implies that

$$D^\alpha J^\alpha [f(t) - f(0)] = [f(t) - f(0)]$$

Definition 2.4.2. ([32]) The Riemann-Liouville fractional derivative of order $\alpha > 0$ with lower limit zero for a function f is defined as

$${}^R D_t^\alpha f(t) = D^n J^{n-\alpha} f(t)$$

namely

$${}^R D_t^\alpha f(t) = \begin{cases} \frac{d^n}{dt^n} \left[\frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau \right], & n-1 < \alpha < n, \quad n \in \mathbb{N} \\ \frac{d^n}{dt^n} f(t), & \alpha = n \end{cases}$$

Note that $D^\alpha J^\alpha = D^\alpha D^{-\alpha} = I$, $J^\alpha D^\alpha \neq I$ where I is the identity for all $\alpha > 0$ and

$$D^\alpha t^\beta = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)} t^{\beta-\alpha}, \quad \alpha > 0, \quad \beta > -1, \quad t > 0. \quad (2.5)$$

Definition 2.4.3. ([32]) The Caputo fractional derivative of order $\alpha > 0$ for a function f is defined as

$${}^C D_t^\alpha f(t) = J^{n-\alpha} D^n f(t)$$

namely

$${}^C D_t^\alpha f(t) = \begin{cases} \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\alpha+1-n}} d\tau, & n-1 < \alpha < n, \quad n \in \mathbb{N} \\ \frac{d^n}{dt^n} f(t), & \alpha = n \end{cases}$$

Note also in general that

$${}^R D_t^\alpha f(t) = D^n J^{n-\alpha} f(t) \neq J^{n-\alpha} D^n f(t) = {}^C D_t^\alpha f(t)$$

for all $n-1 < \alpha < n$ and $t > 0$, one observe that the relation between Riemann-Liouville fractional derivatives and Caputo fractional derivatives is given by:

$${}^R D_t^\alpha f(t) = {}^C D_t^\alpha f(t) + \sum_{k=0}^{n-1} \frac{f^{(k)}(0)}{\Gamma(k-\alpha+1)} t^{k-\alpha} \quad (2.6)$$

and

$${}^C D_t^\alpha f(t) = {}^R D_t^\alpha f(t) - \sum_{k=0}^{n-1} \frac{f^{(k)}(0)}{k!} t^k \quad (2.7)$$

where n is the positive integer uniquely defined by $n - 1 < \alpha \leq n$ which provides the initial value $f^{(k)}(0) = p_k$, $k = 0, 1, 2, \dots, m - 1$.

Definition 2.4.4. ([15]) Hadamard finite-part integral is defined as follows:

$$\oint_a^b (x-a)^{-\alpha-1} f(x) dx = \sum_{\ell=0}^{[1+\alpha]-1} \frac{f^{(\ell)}(a)(b-a)^{\ell+1-\alpha-1}}{(\ell+1-p)\ell!} + \int_a^b (x-a)^{-\alpha-1} R_\mu(x, a) dx$$

Here

$$R_\mu(x, a) := \frac{1}{\mu!} \int_a^x (x-y)^\mu f^{\mu+1}(y) dy$$

$[1 + \alpha]$ is the largest integer not exceeding $1 + \alpha$. For example, in the case of $0 < \alpha < 1$, $[1 + \alpha] = 1$, we have

$$\oint_a^b (x-a)^{-\alpha-1} f(x) dx = \frac{f(a)(b-a)^{-\alpha}}{-\alpha} + \int_a^b (x-a)^{-1-\alpha} R_0(x, a) dx$$

where $R_0(x, a) dx = \int_a^x f^{(1)}(y) dy$ and \oint is the Hadamard part finite integral symbol.

2.5 Existence and Uniqueness of the solution

Diethelm and Ford [17] described the existence and uniqueness of the single term equations with following theorems.

Theorem 2.5.1 (Theorem 2.1 [17] **Existence**). Assume that $\mathbb{D} := [0, \chi^*] \times [y_0^{(0)} - \alpha, y_0^{(0)} + \alpha]$ with some $\chi^* > 0$ and some $\alpha > 0$, and let the function $f : \mathbb{D} \rightarrow \mathbb{R}$ be continuous. Furthermore, define $\chi := \min\{\chi^*, (\alpha\Gamma(q+1)/\|f\|_\infty)^{1/q}\}$. Then there exists a function $y : [0, \chi] \rightarrow \mathbb{R}$ solving the initial value problem single term equations.

Theorem 2.5.2 (Theorem 2.2 [17] **Uniqueness**). Assume that $\mathbb{D} := [0, \chi^*] \times [y_0^{(0)} - \alpha, y_0^{(0)} + \alpha]$ with some $\chi^* > 0$ and some $\alpha > 0$. Furthermore let the function $f : \mathbb{D} \rightarrow \mathbb{R}$ be bounded on \mathbb{D} and satisfy a Lipschitz condition with respect to the second variable; i.e.,

$$|f(x, y) - f(x, z)| \leq L|y - z|$$

with some constant $L > 0$ independent of x, y and z . Then denoting χ as in Theorem 2.5.1, there exist at most one function $y : [0, \chi] \rightarrow R$ solving the initial value problem for single term equations..

Proof of Theorem 2.5.2. The details of the proof can be found in Diethelm and Ford [17]. For the later reference, we extract and summarize the ideas of the proof here. We have the following Volterra equation,

$$y(t) = y_0^{(0)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau.$$

Let the set

$$U = \{y \in C[0, \chi] : \|y - y^{(0)}\|_\infty \leq \alpha\}, \quad (2.8)$$

where (2.8) is a closed subset of the Banach space of every continuous function on $[0, \chi]$, equispaced with the Chebyshev norm. We define operator A on U by

$$(Ay)(t) = y_0^{(0)} + \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau. \quad (2.9)$$

From this operator, we consider

$$y = Ay,$$

such that we can show that A has a unique fixed point.

Next we consider the properties of operator A . For $0 \leq t_1 \leq t_2 \leq \chi$,

$$\begin{aligned} |(Ay)(t_1) - (Ay)(t_2)| &= \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_1} (t_1 - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau \right. \\ &\quad \left. - \frac{1}{\Gamma(\alpha)} \int_0^{t_2} (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau \right| \\ &= \frac{1}{\Gamma(\alpha)} \left| \int_0^{t_1} (t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau \right. \\ &\quad \left. + \int_{t_1}^{t_2} (t_2 - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau \right| \\ &\leq \frac{\|f\|_\infty}{\Gamma(\alpha)} \left| \int_0^{t_1} ((t_1 - \tau)^{\alpha-1} - (t_2 - \tau)^{\alpha-1}) d\tau \right. \\ &\quad \left. + \int_{t_1}^{t_2} (t_2 - \tau)^{\alpha-1} d\tau \right| \\ &= \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} (2(t_1 - t_2)^\alpha + t_1^\alpha - t_2^\alpha), \end{aligned}$$

showing that Ay is continuous. For $y \in U$ and $t \in [0, \chi]$, we get

$$\begin{aligned} |(Ay)(t) - y_0^{(0)}| &= \frac{1}{\Gamma(\alpha)} \left| \int_0^t (t - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau \right| \leq \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} t^\alpha \\ &\leq \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} t^\alpha \leq \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} \frac{\beta \Gamma(\alpha + 1)}{\|f\|_\infty} = \beta. \end{aligned}$$

This proves that $Ay \in U$ if $y \in U$.

Now we need to show that for every $n \in \mathbb{N}_0$ and every $t \in [0, \chi]$, we have

$$\|A^n - A^n \hat{y}\|_{L_\infty[0,t]} \leq \frac{(Lt^\alpha)^n}{\Gamma(\alpha n + 1)} \|y - \hat{y}\|_{\infty[0,t]}. \quad (2.10)$$

By induction, for $n = 0$, (2.10) is trivially true. For $n - 1 \mapsto n$, we have

$$\begin{aligned} \|A^n - A^n \hat{y}\|_{L_\infty[0,t]} &= \|A(A^{n-1}y) - A(A^{n-1}\hat{y})\|_{L_\infty[0,t]} \\ &= \frac{1}{\Gamma(\alpha)} \sup_{0 \leq \omega \leq t} \left| \int_0^\omega (\omega - \tau)^{\alpha-1} [f(\tau, A^{n-1}y(\tau)) - f(\tau, A^{n-1}\hat{y}(\tau))] d\tau \right|. \end{aligned}$$

Using Lipschitz assumption on f and induction hypothesis, we get

$$\begin{aligned} \|A^n y - A^n \hat{y}\|_{L_\infty[0,t]} &\leq \frac{L}{\Gamma(\alpha)} \sup_{0 \leq \omega \leq t} \int_0^\omega (\omega - \tau)^{\alpha-1} |A^{n-1}y(\tau) - A^{n-1}\hat{y}(\tau)| d\tau \\ &\leq \frac{L}{\Gamma(\alpha)} \int_0^\omega (\omega - \tau)^{\alpha-1} \sup_{0 \leq \omega \leq t} |A^{n-1}y(\omega) - A^{n-1}\hat{y}(\omega)| d\tau \\ &\leq \frac{L^n}{\Gamma(\alpha)\Gamma(1 + \alpha(n-1))} \int_0^t (t - \tau)^{\alpha-1} \tau^{\alpha(n-1)} \sup_{0 \leq \omega \leq \tau} |y(\omega) - \hat{y}(\omega)| d\tau \\ &\leq \frac{L^n}{\Gamma(\alpha)\Gamma(1 + \alpha(n-1))} \sup_{0 \leq \omega \leq t} |y(\omega) - \hat{y}(\omega)| \int_0^t (t - \tau)^{\alpha-1} \tau^{\alpha(n-1)} d\tau \\ &= \frac{L^n}{\Gamma(\alpha)\Gamma(1 + \alpha(n-1))} \|y - \hat{y}\|_{L_\infty[0,t]} \frac{\Gamma(\alpha)\Gamma(1 + \alpha(n-1))}{\Gamma(1 + \alpha n)} t^{\alpha n}, \end{aligned}$$

which is the result of (2.10). By taking Chebyshev norms on the interval $[0, \chi]$,

$$\|A^n y - A^n \hat{y}\|_\infty \leq \frac{(L\chi^\alpha)^n}{\Gamma(1 + \alpha n)} \|y - \hat{y}\|_\infty.$$

It is clear to see that operator A satisfies the assumptions of Theorem 2.3 in [17] with $\beta_n = (L\chi^\alpha)^n / \Gamma(1 + \alpha n)$. To apply that theorem, we need to show that the series $\sum_{n=0}^\infty \beta_n$ converges. In fact with this series, we can say that our solution is unique. \square

Proof of Theorem 2.5.1. The comprehensive details of the proof can be found in Diethelm and Ford [17]. Using operator A in (2.9) which maps the nonempty, convex and closed set

$$U = \{y \in C[0, \chi] : \|y - y_0^{(0)}\|_\infty \leq \beta\}.$$

to itself. Assume that f is continuous on the compact set \mathbb{D} , then for $\epsilon > 0$, we get $\delta > 0$ such that

$$|f(t, y) - f(t, \tau)| < \frac{\epsilon}{\chi^\alpha} \Gamma(\alpha + 1), \quad |y - \tau| < \delta. \quad (2.11)$$

Let $y, \hat{y} \in U$ such that $\|y - \hat{y}\| < \delta$. Then in view of (2.11),

$$|f(t, y(t)) - f(t, \hat{y}(t))| < \frac{\epsilon}{\chi^\alpha} \Gamma(\alpha + 1) \quad (2.12)$$

for all $t \in [0, \chi]$. Hence

$$\begin{aligned} |(Ay)(t) - (A\hat{y})(t)| &= \frac{1}{\Gamma(\alpha)} \left| \int_0^t (t - \tau)^{\alpha-1} (f(\tau, y(\tau)) - f(\tau, \hat{y}(\tau))) d\tau \right| \\ &\leq \frac{\Gamma(\alpha + 1)\epsilon}{\chi^\alpha \Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} d\tau = \frac{\epsilon t^\alpha}{\chi^\alpha} \leq \epsilon, \end{aligned}$$

which shows the continuity of operator A . Next we consider the set of functions

$$A(U) = \{Ay : y \in U\}.$$

For $\tau \in A(U)$ we have for all $t \in [0, \chi]$,

$$\begin{aligned} |\tau(t)| &= |(Ay)(t)| \leq |y_0^{(0)}| + \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} |f(\tau, y(\tau))| d\tau \\ &\leq |y_0^{(0)}| + \frac{1}{\Gamma(\alpha + 1)} \|f\|_\infty \chi^\alpha, \end{aligned}$$

which means that $A(U)$ is bounded. For $0 \leq t_1 \leq t_2 \leq \chi$ we get in the proof of Theorem 2.5.2 that

$$\begin{aligned} |(Ay)(t_1) - (Ay)(t_2)| &= \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} (2(t_2 - t_1)^\alpha + t_1^\alpha - t_2^\alpha) \\ &\leq 2 \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} (t_2 - t_1)^\alpha. \end{aligned}$$

Thus, if $|t_2 - t_1| \leq \delta$, then

$$|(Ay)(t_1) - (Ay)(t_2)| \leq 2 \frac{\|f\|_\infty}{\Gamma(\alpha + 1)} \delta^\alpha.$$

□

2.6 Analytical solutions for FDEs

Definition 2.6.1. ([11] **Laplace transform**) Suppose $f(t)$ is a function of $t \in [0, \infty]$, then we define Laplace transform of $f(t)$ by

$$F(s) = \ell\{F\}(s) = \int_0^{\infty} e^{-st} f(t) dt$$

where s is a complex number, $s \in \mathbb{C}$. To ensure that (2.13) makes sense, we require that $f(t)$ must be of exponential order $a \in \mathbb{R}$ and $Re(s) > a$ i.e.

$$|f(t)| \leq Me^{at}, \quad a \in \mathbb{R}.$$

Thus, for $s = x + iy$, $x = Re(s)$ then

$$\begin{aligned} F(s) &= \int_0^{\infty} e^{-st} f(t) dt = \int_0^{\infty} e^{-(x+iy)t} f(t) dt = \int_0^{\infty} e^{-xt} \cdot e^{-iyt} f(t) dt \\ |F(s)| &\leq \left| \int_0^{\infty} e^{-xt} \cdot e^{-iyt} f(t) dt \right| \leq \int_0^{\infty} e^{-xt} \cdot |f(t)| dt \\ &\leq M \int_0^{\infty} e^{-xt} \cdot e^{at} dt = M \int_0^{\infty} e^{-(x-a)t} dt < \infty \end{aligned}$$

if $x - a > 0$ or $a < x$.

In other words, for any complex number $s \in \mathbb{C}$, if $Re(s) > a$, then we guarantee that $|F(s)| < \infty$, where the range of s depends on the properties of f .

Definition 2.6.2. The Laplace transform of an n^{th} derivatives is defined by the formula Doetsch (1961)

$$\ell\{D^n f\} = s^n \hat{f}(s) - \sum_{k=0}^{n-1} D^k f^{(n)} s^{n-k-1}$$

where $\ell(f) = \hat{f}$.

The theorems below gives the definitions of Laplace transform for Mittag-Leffler functions.

Theorem 2.6.3. The Mittag-Leffler function $E_{\alpha}(\mp(z t)^{\alpha})$ is the inverse Laplace transform of the function $\frac{s^{\alpha-1}}{s^{\alpha} \pm z^{\alpha}}$.

Theorem 2.6.4 (Generalized). The Mittag-Leffler function $t^{\beta-1} E_{\alpha, \beta}(-(z t)^{\alpha})$ is the inverse Laplace transform of the function $\frac{s^{\alpha-\beta}}{s^{\alpha} + z^{\beta}}$.

Next, we will solve the following FDEs analytically.

$${}_0^C D_t^\alpha y(t) = \beta y(t), \quad y(0) = y_0, \quad 0 < \alpha < 1$$

From Definition 2.6.2, the Laplace transform of Caputo derivatives gives

$$s^\alpha \widehat{y}(s) - s^{\alpha-1} y_0 = \beta \widehat{y}(s)$$

which implies that

$$\widehat{y}(s) = \frac{s^{\alpha-1} y_0}{s^\alpha - \beta}$$

Using Theorem 2.6.3 the inverse Laplace transform gives

$$y(t) = y_0 E_\alpha(\beta t^\alpha) \tag{2.13}$$

where function $E_\alpha(z)$ is defined by (2.2).

Let us consider one more example with $1 < \alpha < 2$,

$${}_0^C D_t^\alpha y(t) = -\beta^\alpha y(t) + t^2, \quad y(0) = y_0, \quad y'(0) = y_0, \quad 1 < \alpha < 2$$

From Definition 2.6.2, the Laplace transform gives

$$s^\alpha \widehat{y}(s) - (s^{\alpha-1} y_0 + s^{\alpha-2} y_0') = -\beta^\alpha \widehat{y}(s) + \frac{\Gamma(3)}{s^3}$$

which implies that

$$\widehat{y}(s) = \frac{s^{\alpha-1} y_0}{s^\alpha + \beta^\alpha} + \frac{s^{\alpha-2} y_0'}{s^\alpha + \beta^\alpha} + \frac{\Gamma(3)}{s^3 (s^\alpha + \beta^\alpha)}$$

Using Theorem 2.6.4 and the convolution of inverse Laplace transform we have

$$y(t) = y_0 E_\alpha(-(\beta t)^\alpha) + y_0' t E_{\alpha,2}(-(\beta t)^\alpha) + t^2 * (t^{\alpha-1} E_{\alpha,\alpha}(-(\beta t)^\alpha))$$

where $E_{\alpha,\beta}(z)$ is defined by (2.3) and the symbol ‘*’ denotes the convolution of two functions.

Chapter 3

Numerical methods for FDEs

It is very difficult to find the exact solutions of FDEs according to Diethelm [16]. This is why numerical method is essential to illustrate the behaviour of the solutions of FDEs. Let us consider the non-linear FDEs given by

$${}_0^C D_t^\alpha y(t) = f(t, y(t)), \quad 0 < \alpha \leq 1, \quad (3.1)$$

$$y(0) = y_0 \quad (3.2)$$

where f is a given function on the interval $[0, 1]$ and $0 < \alpha \leq 1$ is the order of the derivative. The existence and uniqueness of the solution for the linear FDEs have been discussed in Diethelm and Ford [17], Diethelm [16] and Podlubny (1992). Many authors have considered the numerical methods for solving FDEs. Lubich [3] applied the convolution quadrature method to approximate fractional integral and obtained the approximate solutions since FDEs can be replaced by an Abel-Volterra equations. Diethelm [20] wrote the fractional Riemann-Liouville derivative by the finite-part Hadamard integral, then approximated the integral by using quadrature formula approach and obtained an implicit numerical method for solving FDEs. Diethelm, Ford and Freed [23] introduced an algorithm to solve FDEs by using predictor-corrector argument. Diethelm and Luchko used the observation that FDEs has an exact solution which can be expressed as a Mittag-Leffler function. Both authors used convolution quadrature and discretized operational calculus to produce approximation to this Mittag-Leffler function. Blank [25] used collocation method to approximate FDEs. Podlubny [11] and Gorenflo [33] used the finite difference method to approximate the fractional derivative and obtained the

approximation scheme of FDEs.

The convergence orders of the different numerical methods are worth mentioning. Podlubny used Grünwald-Letnikov to approximate fractional derivatives of Caputo type, defined an implicit finite difference method for solving the initial value problems and proved the convergence order is $O(h)$. Gorenflo [33] introduced a second order $O(h^2)$ difference method for solving (3.1)- (3.2) but the condition to achieve the desired accuracy are restrictive. Diethelm (1997) proved the convergence order is $O(h^{2-\alpha})$ for the backward difference method of solving (3.1)- (3.2). The author also pointed that the convergent is also the same for $1 < \alpha \leq 2$.

In this chapter, we will construct a numerical method of solving (3.1)- (3.2) based on the methods of Lubich convolution quadrature and Diethelm algorithm. We will also extend our approach to predictor-corrector methods based on Diethelms [18] and Garrappa [31].

3.1 Diethelms method

In this section we review Diethelm's method (1997) for solving fractional differential equations where the finite-part Hadamard integral is approximated by piecewise linear interpolation polynomials.

Consider

$${}_0^C D_t^\alpha y(t) = \beta y(t) + f(t), \quad (3.3)$$

$$y(0) = y_0 \quad (3.4)$$

such that systems (3.3)- (3.4) can be replaced by

$${}_0^R D_t^\alpha [y(t) - y_0] = \beta y(t) + f(t), \quad 0 < \alpha \leq 1, \quad 0 \leq t \leq 1 \quad (3.5)$$

where α is the order of the derivative, f is a given function on the interval $[0,1]$, $\beta \leq 0$ and y is the unknown function. From the definition of Riemann-Liouville fractional derivative in Chapter 2, for $0 < \alpha < 1$ we get

$${}_0^R D_t^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_0^t (t-\tau)^{-\alpha} y(\tau) d\tau \quad (3.6)$$

Kumar and Agrawal [30] mentioned that Riemann-Liouville derivatives is more commonly used in pure mathematics than the Caputo derivatives. Let us consider the following lemmas and their proofs from Diethelm [15]. These lemmas will help us to understand the algorithm of numerical method for solving FDEs.

Lemma 3.1.1 (Elliot 1993 [6]). The Hadamard finite part integral for the Riemann-Liouville derivative (3.6) can be written as

$${}_0^R D_t^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \oint_0^t (t-\tau)^{-1-\alpha} y(\tau) d\tau$$

where $0 < \alpha \leq 1$, \oint represents the symbol of Hadamard integral.

Lemma 3.1.2 (Diethelm 1997). Assume that $0 = t_0 < t_1 < t_2 < \dots < t_k < \dots < t_N = 1$ is the partition on the interval $[0, 1]$ and $0 < \alpha < 1$, then at $t = t_j$,

$${}_0^R D_t^\alpha [y(t_j)] = h^{-\alpha} \sum_{k=0}^j \omega_{kj} y(t_j - t_k) + \frac{t_j^{-\alpha}}{\Gamma(\alpha)} R_j, \quad j = 1, 2, 3, \dots, N.$$

where R_j is the remainder term given by

$$|R_j| \leq C \cdot j^{\alpha-2} \|x''(t_j - t_j \omega)\|_\infty, \quad 0 < \omega \leq 1,$$

where h is the time-step size and ω_{kj} satisfy

$$\Gamma(2-\alpha)\omega_{kj} = \begin{cases} 1, & k=0 \\ -2k^{1-\alpha} + (k-1)^{1-\alpha} + (k+1)^{1-\alpha}, & k=1, 2, \dots, j-1 \\ -(\alpha-1)k^{-\alpha} + (k-1)^{1-\alpha} - k^{1-\alpha}, & k=j \end{cases}$$

The proof for this Lemma 3.1.2 is straight forward and requires a piecewise linear Lagrange interpolation polynomial.

Proof. We have

$$\begin{aligned} {}_0^R D_t^\alpha y(t_j) &= \frac{1}{\Gamma(-\alpha)} \oint_0^{t_j} \frac{y(\tau)}{(t-\tau)^{\alpha+1}} d\tau \\ &= \frac{t_j^{-\alpha}}{\Gamma(-\alpha)} \oint_0^1 \frac{x(t_j - t_j \omega)}{\omega^{\alpha+1}} d\omega \end{aligned}$$

By substitution i.e. $g(\omega) = x(t_j - t_j \omega)$, we have

$${}_0^R D_t^\alpha y(t_j) = \frac{t_j^{-\alpha}}{\Gamma(-\alpha)} \oint_0^1 g(\omega) \omega^{-(\alpha+1)} d\omega$$

Let $0 = t_0 < t_1 < t_2 < \dots < t_j < \dots < t_N = T$ be a partition of $[0, T]$, let N be a fixed positive integer and $h = T/N$ the step size. We have at $t_j = jh$ where $j = 0, 1, 2, \dots, N$,

$${}^R D_t^\alpha D_t^\alpha [y(t_j) - y_0] = \beta y(t_j) + f(t_j) \quad (3.7)$$

Using Lemma 3.1.1 we get

$$\frac{1}{\Gamma(-\alpha)} \int_0^{t_j} (t_k - \tau)^{-1-\alpha} [y(\tau) - y_0] d\tau = \beta y(t_j) + g(t_j), \quad j = 1, 2, \dots, N. \quad (3.8)$$

and by change of variables we get

$$\int_0^{t_j} (t_j - \tau)^{-1-\alpha} [y(\tau) - y_0] d\tau = \frac{t_j^{-\alpha}}{\Gamma(-\alpha)} \int_0^1 g(\omega) \omega^{-(\alpha+1)} d\omega$$

where $g(\omega) = y(t_j - t_j\omega)$.

For every j , we replace the integral by a piecewise interpolation polynomials with equispaced nodes $0, 1/j, 2/j, 3/j, \dots, j/j$. That is,

$$\int_0^1 g(\omega) \omega^{-\alpha-1} d\omega = \int_0^1 g_1(\omega) \omega^{-\alpha-1} d\omega + R_j$$

where $g_1(\omega)$ is the piecewise linear interpolation polynomial of $g(\omega)$ with the equispaced nodes and R_j is the remainder term.

Note that,

$$g_1(\omega) = \frac{\omega - \frac{k}{j}}{\frac{k-1}{j} - \frac{k}{j}} g\left(\frac{k-1}{j}\right) + \frac{\omega - \frac{k-1}{j}}{\frac{k}{j} - \frac{k-1}{j}} g\left(\frac{k}{j}\right), \quad \omega \in \left[\frac{k-1}{j}, \frac{k}{j}\right].$$

Thus,

$$\int_0^1 g(\omega) \omega^{-(1+\alpha)} d\omega \approx \int_0^1 g_1(\omega) \omega^{-(1+\alpha)} d\omega = Q_j(g)$$

Here we observe generally that

$$Q_j(g) = \int_0^1 g_1(\omega) \omega^{-(1+\alpha)} d\omega = \int_0^{\frac{1}{j}} g_1(\omega) \omega^{-(1+\alpha)} d\omega + \sum_{k=2}^j \int_{\frac{k-1}{j}}^{\frac{k}{j}} g_1(\omega) \omega^{-(1+\alpha)} d\omega$$

and in particular

$$Q_j(g) = \int_0^{\frac{1}{j}} g(\omega) \omega^{-(1+\alpha)} d\omega + \int_{\frac{1}{j}}^{\frac{2}{j}} g(\omega) \omega^{-(1+\alpha)} d\omega + \dots + \int_{\frac{j-1}{j}}^1 g(\omega) \omega^{-(1+\alpha)} d\omega \quad (3.9)$$

Applying the Lagrange interpolation polynomial on each integral on the right hand side of (3.9) gives

$$\begin{aligned} \oint_0^{\frac{1}{j}} g(\omega)\omega^{-(1+\alpha)}d\omega &= \oint_0^{\frac{1}{j}} \left[\frac{\omega - \frac{1}{j}}{0 - \frac{1}{j}}g(0) + \frac{\omega - 0}{\frac{1}{j} - 0}g\left(\frac{1}{j}\right) \right] \omega^{-(\alpha+1)} \\ \oint_{\frac{1}{j}}^{\frac{2}{j}} g(\omega)\omega^{-(1+\alpha)}d\omega &= \oint_{\frac{1}{j}}^{\frac{2}{j}} \left[\frac{\omega - \frac{2}{j}}{\frac{1}{j} - \frac{2}{j}}g\left(\frac{1}{j}\right) + \frac{\omega - \frac{1}{j}}{\frac{2}{j} - \frac{1}{j}}g\left(\frac{2}{j}\right) \right] \omega^{-(\alpha+1)} \\ \oint_{\frac{j}{j}}^{\frac{j-1}{j}} g(\omega)\omega^{-(1+\alpha)}d\omega &= \oint_{\frac{j}{j}}^{\frac{j-1}{j}} \left[\frac{\omega - \frac{j}{j}}{\frac{j-1}{j} - \frac{j}{j}}g\left(\frac{j-1}{j}\right) + \frac{\omega - \frac{j-1}{j}}{\frac{j}{j} - \frac{j-1}{j}}g\left(\frac{j}{j}\right) \right] \omega^{-(\alpha+1)} \end{aligned}$$

By Definition 2.4.4, we can deduce that

$$\begin{aligned} \oint_0^{\frac{1}{j}} g(\omega)\omega^{-(1+\alpha)}d\omega &= \frac{g_1(0)\left(\frac{1}{j}\right)^{1-(1+\alpha)}}{(0+1-(1+\alpha))0!} + \int_0^{\frac{1}{j}} \omega^{-(1+\alpha)}\frac{1}{0!} \left[\int_0^\omega (\omega-y)^0 g_1^{(1)}(y)dy \right] d\omega \\ &= \frac{1}{(-\alpha)j^{-\alpha}}g(0) + \int_0^{\frac{1}{j}} \omega^{-(1+\alpha)} \left[\int_0^\omega \left(jg\left(\frac{1}{j}\right) - jg(0) \right) dy \right] d\omega \\ &= \frac{1}{(-\alpha)j^{-\alpha}}g(0) + \left[jg\left(\frac{1}{j}\right) - jg(0) \right] \cdot \int_0^{\frac{1}{j}} \omega^{-\alpha}d\omega \\ &= \left[\frac{1}{(-\alpha)j^{-\alpha}} - \frac{1}{(1-\alpha)j^{-\alpha}} \right] g(0) + \frac{1}{(1-\alpha)j^{-\alpha}}g\left(\frac{1}{j}\right) \\ &= \frac{1}{(1-\alpha)j^{-\alpha}}g\left(\frac{1}{j}\right) - \frac{1}{\alpha(1-\alpha)j^{-\alpha}}g(0) \end{aligned}$$

Now we consider in general:

$$\begin{aligned} \int_{\frac{k-1}{j}}^{\frac{k}{j}} g_1(\omega)\omega^{-(1+\alpha)}d\omega &= \int_{\frac{k-1}{j}}^{\frac{k}{j}} \left[\frac{\omega - \frac{k}{j}}{-\frac{1}{j}}g\left(\frac{k-1}{j}\right) + \frac{\omega - \frac{k-1}{j}}{\frac{1}{j}}g\left(\frac{k}{j}\right) \right] \omega^{-(1+\alpha)}d\omega \\ &= g\left(\frac{k-1}{j}\right) \int_{\frac{k-1}{j}}^{\frac{k}{j}} j\left(\frac{k}{j} - \omega\right) \omega^{-(1+\alpha)}d\omega + g\left(\frac{k}{j}\right) \int_{\frac{k-1}{j}}^{\frac{k}{j}} j\left(\omega - \frac{k-1}{j}\right) \omega^{-(1+\alpha)}d\omega \\ &= g\left(\frac{k-1}{j}\right) \int_{\frac{k-1}{j}}^{\frac{k}{j}} (k\omega^{-(1+\alpha)}d\omega - j\omega^{-\alpha}) d\omega + g\left(\frac{k}{j}\right) \int_{\frac{k-1}{j}}^{\frac{k}{j}} (j\omega^{-\alpha} - (k-1)\omega^{-(1+\alpha)}) d\omega \\ &= g\left(\frac{k-1}{j}\right) \left[\frac{k}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} - \frac{j}{1-\alpha} \left(\frac{k}{j}\right)^{1-\alpha} - \frac{k}{-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} + \frac{j}{1-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} \right] \\ &+ g\left(\frac{k}{j}\right) \left[\frac{j}{1-\alpha} \left(\frac{k}{j}\right)^{1-\alpha} - \frac{k-1}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} - \frac{j}{1-\alpha} \left(\frac{k-1}{j}\right)^{1-\alpha} + \frac{k-1}{-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} \right] \end{aligned}$$

Thus we have

$$\begin{aligned}
Q_j(g) &= \frac{1}{(1-\alpha)j^{-\alpha}}g\left(\frac{1}{j}\right) - \frac{1}{\alpha(1-\alpha)j^{-\alpha}}g(0) + \sum_{k=2}^j \left(\frac{k-1}{j}\right) \left[\frac{k}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} - \frac{j}{1-\alpha} \left(\frac{k}{j}\right)^{1-\alpha} \right. \\
&\quad \left. - \frac{k}{-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} + \frac{j}{1-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} \right] + g\left(\frac{k}{j}\right) \left[\frac{j}{1-\alpha} \left(\frac{k}{j}\right)^{1-\alpha} - \frac{k-1}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} \right. \\
&\quad \left. - \frac{j}{1-\alpha} \left(\frac{k-1}{j}\right)^{1-\alpha} + \frac{k-1}{-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} \right] = \sum_{k=0}^j \alpha_{kj} g\left(\frac{k}{j}\right) \\
&= \sum_{k=0}^j \alpha_{kj} y(t_j - t_k),
\end{aligned} \tag{3.10}$$

where α_{kj} satisfy the following:

when $k = 0$,

$$\alpha_{0j} = \frac{1}{\alpha(1-\alpha)j^{-\alpha}},$$

and when $k = j$,

$$\begin{aligned}
\alpha_{jj} &= \left[\frac{j}{1-\alpha} \left(\frac{j}{j}\right)^{1-\alpha} - \frac{j-1}{-\alpha} \left(\frac{j}{j}\right)^{-\alpha} - \frac{j}{1-\alpha} \left(\frac{j-1}{j}\right)^{1-\alpha} + \frac{j-1}{-\alpha} \left(\frac{j-1}{j}\right)^{-\alpha} \right] \\
&= \frac{j}{1-\alpha} - \frac{j-1}{-\alpha} - \frac{(j-1)^{1-\alpha}}{(1-\alpha)j^{-\alpha}} + \frac{(j-1)^{1-\alpha}}{(-\alpha)j^{-\alpha}} \\
&= \frac{\alpha j^{1-\alpha} + (1-\alpha)(j-1)j^{-\alpha} - \alpha(j-1)^{1-\alpha} + (\alpha-1)(j-1)^{1-\alpha}}{\alpha(1-\alpha)j^{-\alpha}} \\
&= \frac{\alpha j^{1-\alpha} + (1-\alpha)j^{1-\alpha} - (1-\alpha)^{-\alpha} - (j-1)^{1-\alpha}}{\alpha(1-\alpha)j^{-\alpha}} \\
&= \frac{(\alpha-1)j^{-\alpha} - (j-1)^{1-\alpha} + j^{1-\alpha}}{\alpha(1-\alpha)j^{-\alpha}}.
\end{aligned}$$

For $k = 1, 2, 3, 4, \dots, j-1$, we have

$$\begin{aligned}
\alpha_{kj} &= \frac{k+1}{-\alpha} \left(\frac{k+1}{j}\right)^{-\alpha} - \frac{j}{1-\alpha} \left(\frac{k+1}{j}\right)^{-\alpha} - \frac{k+1}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} \\
&\quad + \frac{j}{1-\alpha} \left(\frac{k}{j}\right)^{1-\alpha} + \frac{j}{1-\alpha} \left(\frac{k}{j}\right)^{1-\alpha} - \frac{k-1}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} - \frac{j}{1-\alpha} \left(\frac{k-1}{j}\right)^{1-\alpha} \\
&\quad + \frac{k-1}{-\alpha} \left(\frac{k-1}{j}\right)^{-\alpha} \\
&= \left[\frac{1}{-\alpha j^{-\alpha}} - \frac{1}{(1-\alpha)j^{-\alpha}} \right] (k+1)^{1-\alpha} + \left[-\frac{1}{(1-\alpha)j^{-\alpha}} + \frac{1}{(-\alpha)j^{-\alpha}} \right] (k-1)^{1-\alpha}
\end{aligned}$$

$$\begin{aligned}
& + \left[-\frac{k+1}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} - \frac{k-1}{-\alpha} \left(\frac{k}{j}\right)^{-\alpha} \right] + \left[\frac{1}{(1-\alpha)j^{-\alpha}} + \frac{1}{(1-\alpha)j^{-\alpha}} \right] k^{1-\alpha} \\
& = \frac{\alpha-1-\alpha}{\alpha(1-\alpha)j^{-\alpha}}(k+1)^{1-\alpha} + \frac{-\alpha+(\alpha-1)}{\alpha(1-\alpha)j^{-\alpha}}(k-1)^{1-\alpha} \\
& + \frac{2k}{\alpha} \left(\frac{k}{j}\right)^{-\alpha} + \frac{2}{(1-\alpha)j^{-\alpha}} k^{1-\alpha} = \frac{(k+1)^{1-\alpha}}{\alpha(1-\alpha)j^{-\alpha}} - \frac{(k-1)^{1-\alpha}}{\alpha(1-\alpha)j^{-\alpha}} + \frac{2k^{1-\alpha}}{\alpha(1-\alpha)j^{-\alpha}} \\
& = \frac{1}{\alpha(1-\alpha)j^{-\alpha}} [2k^{1-\alpha} - (k-1)^{1-\alpha} - (k+1)^{1-\alpha}].
\end{aligned}$$

Further we write

$$Q_j(g) = \sum_{k=0}^j \alpha_{kj} y(t_j - t_k) = h^{-\alpha} \sum_{k=0}^j \omega_{kj} y(t_j - t_k),$$

where

$$\Gamma(2-\alpha)\omega_{kj} = \begin{cases} 1, & k=0 \\ -2k^{1-\alpha} + (k-1)^{1-\alpha} + (k+1)^{1-\alpha}, & k=1, 2, \dots, j-1 \\ -(\alpha-1)k^{-\alpha} + (k-1)^{1-\alpha} - k^{1-\alpha}, & k=j \end{cases}$$

Together, these estimates completes the proof of Lemma 3.1.2. \square

It was shown in [15] that the remainder term $R_j(g)$ satisfies

$$|R_j(g)| = \left| \oint_0^1 g(\omega)\omega^{-(1+\alpha)}d\omega - \sum_{k=0}^j \alpha_{kj} g\left(\frac{k}{j}\right) \right| \leq c j^{\alpha-2}$$

Thus we can write (3.8) into

$$y(t_j) = \frac{1}{\alpha_{0j} - t_j^\alpha \Gamma(-\alpha)\beta} \left[t_j^\alpha \Gamma(-\alpha) g(t_j) \sum_{k=1}^j \alpha_{kj} y(t_{j-k}) + y_0 \sum_{k=0}^j \alpha_{kj} - R_j \right] \quad (3.11)$$

Let $y_j \approx y(t_j)$ denote the approximate solution of $y(t_j)$, $j = 1, 2, 3, \dots, N$, then based on (3.11) we can define the following numerical method for solving (3.5) as

$$y_j = \frac{1}{\alpha_{0j} - t_j^\alpha \Gamma(-\alpha)\beta} \left[t_j^\alpha \Gamma(-\alpha) g_j \sum_{k=1}^j \alpha_{kj} y_{j-k} + y_0 \sum_{k=0}^j \alpha_{kj} \right] \quad (3.12)$$

We remark that Lemma 3.1.2 for $0 < \alpha < 1$ can be extended to the case for $1 < \alpha < 2$ to yield the following weights,

$$\alpha(1-\alpha)j^{-\alpha}\alpha_{kj} = \begin{cases} -1 & k=0 \\ \alpha & k=1, j=0 \\ 2-2^{1-\alpha} & k=1, j>1 \\ 2k^{1-\alpha} - (k-1)^{1-\alpha} - (k+1)^{1-\alpha} & k=1, 2, \dots, j-1, j \geq 3 \\ (\alpha-1)k^{-\alpha} - (k-1)^{1-\alpha} + k^{1-\alpha} & k=j, j \geq 2 \end{cases}$$

These weights are obtained by following the same process from Lemmas 3.1.1 and 3.1.2. The only difference lies from the Hadamard finite part integral. An error estimate for the scheme (3.12) can be performed by means of the following result whose proofs are similar to those in [15].

Theorem 3.1.3. Assume $y(t_j)$ and y_j are the exact and approximate solutions of (3.11) and (3.12) respectively. Also, assume that the function involved is sufficiently smooth, then there exists a constant $C = C(\alpha, g, \beta)$, such that

$$|y(t_j) - y_j| \leq Ch^{2-\alpha} \|y''\|_{\infty}. \quad j = 1, 2, \dots, m.$$

Proof. Assume

$$e_j = y(t_j) - y_j$$

then we have the error equation, subtracting (3.11) from (3.12),

$$e_j = \frac{1}{\alpha_{0j} - t_j^{\alpha} \Gamma(-\alpha) \beta} \left[- \sum_{k=1}^j \alpha_{kj} e_{j-k} - R_j \right]$$

Note that

$$\alpha_{0j} = \frac{1}{-\alpha(1-\alpha)j^{-\alpha}} < 0, \quad \Gamma(-\alpha) < 0, \quad \beta < 0, \quad \alpha_{kj} > 0.$$

then we have

$$\begin{aligned} |e_j| &\leq \frac{1}{-\alpha_{0j}} \left(\sum_{k=1}^j \alpha_{kj} |e_{j-k}| + |R_j| \right) \\ &\leq \alpha(1-\alpha)j^{-\alpha} \left(\sum_{k=1}^j \alpha_{kj} |e_{j-k}| + j^{\alpha-2} t_j^2 \|y''\|_{\infty} \right) \\ &\leq \alpha(1-\alpha)h^2 \|y''\|_{\infty} + \alpha(1-\alpha)j^{-\alpha} \sum_{k=1}^j \alpha_{kj} |e_{j-k}| \end{aligned}$$

By denoting $a = \alpha(1 - \alpha)h^2\|y''\|_\infty$ and assume for simplicity that $e_0 = 0$ then we get

$$|e_j| \leq a + \alpha(1 - \alpha)j^{-\alpha} \sum_{k=1}^j \alpha_{kj} |e_{j-k}|, \quad j = 1, 2, \dots, m$$

which implies that

$$|e_j| \leq ad_j, \quad j = 1, 2, \dots, m$$

where

$$\begin{cases} d_1 = 1 \\ d_j = 1 + \alpha(1 - \alpha)j^{-\alpha} \sum_{k=1}^j \alpha_{kj} d_{j-k} \end{cases}$$

Lemma 3.1.4 (Gronwall). Let $0 < \alpha < 1$ be the order of derivative and the sequence (d_j) satisfy

$$\begin{cases} d_1 = 1 \\ d_j = 1 + \alpha(1 - \alpha)j^{-\alpha} \sum_{k=1}^j \alpha_{kj} d_{j-k}. \end{cases}$$

Then, we have

$$1 \leq d_j \leq C_\alpha j^\alpha, \quad j = 1, 2, \dots, N.$$

Applying Lemma 3.1.4, we have

$$|e_j| \leq ad_j = \alpha(1 - \alpha)n^{-2} |y''(t)|_\infty j^\alpha h^\alpha h^{-\alpha} \leq Ch^{2-\alpha}.$$

The proof of Theorem 3.1.3 is complete. \square

3.2 Predictor-Corrector method

It is proposed in [18, 23, 31] that a more accurate solution can be calculated using the predictor-corrector method. To make our discussion more consistence, it is worth understanding what we meant by predictor-corrector method. In simplicity, a predictor-corrector method is an algorithm that proceeds in two steps. First, the predicton step calculates a rough approximation of the desired equation. Second, the correction step refines the approximation from the prediction step using another means. The use of suitable combination of an explicit and an implicit technique is the idea behind predictor-corrector method. Therefore, one can say the combination of an explicit and an implicit technique is called a predictor-corrector method where the explicit method predicts an approximation and the implicit method corrects this prediction.

Alfeld [7] noted that these methods have been successful because they occur in naturally arising families covering a range of orders. They have a better convergence characteristics and a good stability properties. They also allow for easy error control via proper step size changing techniques.

In most papers like [23], the combination of forward Euler (FE) and the classical one-step Adams-Moulton (AM2) or Trapezoidal method of ODEs is employed. The FE serves as the predictor approximation denoted by y_{k+1}^p and subsequently AM2 serves as a corrector approximation to get the final computed solution, y_{k+1} . However, we can refer this method as Euler-Trapezoidal method given by

$$y_{k+1}^p = y_k + f(t_k, y_k) \quad (3.13)$$

$$y_{k+1} = y_k + \frac{h}{2} [f(t_k, y_k) + f(y_{k+1}^p, t_{k+1})] \quad (3.14)$$

where (3.13) and (3.14) are the predictor and corrector equations respectively. We note that (3.14), the implicit term for the AM2, $f(y_{k+1}, t_{k+1})$ is replaced with $f(y_{k+1}^p, t_{k+1})$ which is the value of f evaluated at the predicted y_{k+1}^p is used. It is also observed in [18] that this method is of PECE type where P stands for Predict, Es stands for Evaluate, C stands for Correct. This is because we would start by calculating the predictor in (3.13), then evaluate $f(y_{k+1}^p, t_{k+1})$, use this evaluation to calculate the corrector in (3.14), and conclude by evaluating $f(y_{k+1}, t_{k+1})$. The result is stored for more iterations.

Next, let us consider the following example.

$$\begin{aligned} y'(t) &= -\lambda y(t), \quad \alpha > 0, \quad \lambda > 0, \\ y(0) &= 1. \end{aligned}$$

The predictor gives

$$\begin{aligned} y_{k+1}^p &= y_k + hf(t_k, y_k) \\ y_k^p &= (1 - \lambda h)y_{k-1} \end{aligned}$$

The corrector gives

$$\begin{aligned} y_k &= y_{k-1} + \frac{h}{2} [f(t_{k-1}, y_{k-1}) + f(t_k, y_k^p)] \\ &= \left(1 - \frac{\lambda h}{2}\right) y_{k-1} - \frac{\lambda h}{2} y_k^p \end{aligned}$$

The predictor-corrector method iterations can be computed for $k = 1, 2, 3, \dots, N$.

Having explained the predictor-corrector method in an ODEs sense, we now carry the basic ideas to fractional order problems. A recent and thorough discussion on predictor-corrector method for FDEs can be seen in Diethelm, Ford and Freed [18, 23] and Garrappa [31]. We will extract some main points from these papers.

Let us consider

$${}_0^C D_t^\alpha y(t) = f(t, y(t)), \quad 0 < \alpha \leq 1 \quad (3.15)$$

$$y(0) = y_0 \quad (3.16)$$

where $0 < \alpha \leq 1$ is the order of derivative. We assume that f satisfies the Lipschitz condition with respect to the second variable stated in Theorems 2.5.1 and 2.5.2. It is noted in [18] that (3.15)- (3.16) can be written as a Volterra integration equation,

$$y(t) = y_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} f(\tau, y(\tau)) d\tau \quad (3.17)$$

in the sense that the continuous function is a solution of (3.15)- (3.16) if and only if it is a solution of (3.17). In [18, 23], product trapezoidal quadrature (PTQ) was introduced to point these straightforward rules that generalizes the Adams method to fractional differential equations.

Let $0 = t_0 < t_1 < t_2 < \dots < t_j < \dots < t_N = T$ be the partition on $t \in [0, T]$ with the equispaced grid $t_j = jh$, h denotes the time step size and $f_j = f(t_j, y_j)$ where y_j is the numerical approximation to $y(t_j)$. The *PTQ* algorithm is given [18] by

$$y_k = y_0 + \sum_{j=0}^{k+1} \mu_{j,k+1} f_j \quad (3.18)$$

where

$$\mu_{j,k+1} = \frac{h^\alpha}{\alpha(\alpha+1)} \begin{cases} 1 & j = k+1, \\ (k^{\alpha+1} - (k-\alpha)(k+1)^\alpha) & j = 0, \\ ((k-j+2)^{\alpha+1} + (k-j)^{\alpha+1} \\ \quad - 2(k-j+1)^{\alpha+1}) & 1 \leq j \leq k, \end{cases}$$

such that the corrector formula [23] is given by

$$y_{k+1} = y_0 + \frac{1}{\Gamma(\alpha)} \left(\sum_{j=0}^{k+1} \mu_{j,k+1} f_j + \mu_{k+1,k+1} f(y_{k+1}^p, t_{k+1}) \right). \quad (3.19)$$

Next, the product rectangle rule algorithm is given [18] by,

$$y_k = y_0 + \sum_{j=0}^{k+1} \zeta_{j,k+1} f_j \quad (3.20)$$

where

$$\zeta_{j,k+1} = \frac{h^\alpha}{\alpha} ((k+1-j)^\alpha - (k-j)^\alpha) \quad (3.21)$$

such that the predictor y_{k+1}^p which is calculated by fractional forward Euler method is given [18] by

$$y_{k+1}^p = y_0 + \frac{1}{\Gamma(\alpha)} \left(\sum_{j=0}^k \zeta_{j,k+1} f_j \right). \quad (3.22)$$

The algorithm of this method have been studied by other authors. Garrappa [31], states the product rectangle rule as

$$y_k = y_0 + h^\alpha \sum_{j=0}^{k-1} \mu_{k-j-1} f_j \quad (3.23)$$

where

$$\mu_k = (k+1)^\alpha - \frac{k^\alpha}{\Gamma(\alpha+1)} \quad (3.24)$$

while the product trapezoidal rule is given by

$$y_k = y_0 + h^\alpha \left(\zeta_{k,0} f_0 + \sum_{j=1}^k \beta_{k-j} f_j \right) \quad (3.25)$$

where

$$\zeta_{k,0} = (k-1)^{\alpha+1} - \frac{k^\alpha(k-\alpha-1)}{\Gamma(\alpha+2)} \quad (3.26)$$

and

$$\beta_k = \begin{cases} \frac{1}{\Gamma(\alpha+2)} & k = 0 \\ \frac{(k-1)^{\alpha+1} - 2k^{\alpha+1} + (k+1)^{\alpha+1}}{\Gamma(\alpha+2)} & k = 1, 2, 3, \dots \end{cases}$$

It is also stated in [31] that predictor-corrector method algorithm is given by

$$\begin{cases} y_k^p = y_0 + h^\alpha \sum_{j=0}^{k-1} \mu_{k-j-1} f_j \\ y_k = y_0 + h^\alpha \left(\zeta_{k,0} f_0 + \sum_{j=1}^{k-1} \beta_{k-j} f_j + \beta_0 f(t_k, y_k^p) \right), \end{cases} \quad (3.27)$$

and that the k -step implicit Adams product trapezoidal rule converges like h^{k+1} as long as the exact solution is sufficiently smooth while that of the k -step explicit product rectangle rule converges like h^k .

3.3 Lubich's method

In this section we will consider fractional linear multistep methods for Abel-Volterra integral equations. The idea of this method has been presented by Lubich [4]. The method is said to be convergent of the order of the underlying multistep method and the stability properties are related to linear multistep methods. We begin our investigation of this method by extracting some main points from [4] and mostly use the same notations.

Let us consider the Abel-Volterra equation of the second kind of the form

$$y(t) = f(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} K(\tau, y(\tau)) d\tau, \quad t \in [0, T], \quad \alpha > 0. \quad (3.28)$$

where the forcing term f is a known function of t , K (the kernel) is a known function of two variables and y is the unknown function to be evaluated. We define Lubich's method by the convolution quadratures

$$y_n = f(t_n) + h^\alpha \sum_{j=0}^n \omega_{n-j} K(t_j, y_j) + h^\alpha \sum_{j=0}^m \omega_{nj} K(t_j, y_j) \quad (3.29)$$

where ω_n are the convolution weights, ω_{nj} are the starting weights and whose errors satisfies

$$\max_{0 \leq m \leq N} |y_n - y(t_n)| = O(h^{p-\epsilon})$$

where $\epsilon \geq 0$.

The convolution weights ω_n in (3.29) are given by the generating function

$$\omega^\alpha(\xi) = \left(\frac{\sigma(1/\xi)}{\rho(1/\xi)} \right)^\alpha \quad (3.30)$$

where (ρ, σ) are the well known linear multi-step methods characteristic polynomial and starting weights are shown by [29] (since starting weights do not play any role in stability regions, which is our main purpose in this dissertation, therefore we will not consider them). We remark that (ρ, σ) are practically multi-step methods of ODEs

$$y'(t) = f(t, y(t))$$

given by

$$\sum_{j=0}^p \alpha_j y_{n-j} = h \sum_{j=0}^p \beta_j f(t_{n-j}, y_{n-j})$$

where $n = p+1, p+2, \dots$, β_j is the attached coefficient to generate a better accuracy and

$$\rho(\xi) = \sum_{j=0}^p \alpha_j \xi^{p-j}, \quad \sigma(\xi) = \sum_{j=0}^p \beta_j \xi^{p-j}$$

For convenience reasons the starting weights ω_{nj} are given [29] by

$$\sum_{j=0}^m \omega_{nj} j^\gamma = \frac{\Gamma(1+\gamma)}{\Gamma(1+\gamma+\alpha)} n^{\alpha+\gamma} - \sum_{j=1}^k \omega_{n-j} j^\gamma, \quad \gamma \in U \quad (3.31)$$

with

$$U = \{ \gamma = k + j\alpha; k, j \in \mathbb{N}_0, \gamma \leq p-1 \}.$$

Let us consider the fractional differential equation. Suppose $\alpha > 0$ and $s = [\alpha]$, we have

$${}_0^C D_t^\alpha y(t) = f(t, y(t)), \quad D^k y(0) = b_k, \quad k = 0, 1, 2, \dots, s-1 \quad (3.32)$$

which can be written as Abel-Volterra integral equation

$$y(t) = y_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} K(\tau, y(\tau)) d\tau, \quad t \in [0, T].$$

Thus we can define the following numerical method for solving (3.32), with $y_n = y(t_n)$

$$y_n = y_0 + h^\alpha \sum_{j=0}^n \omega_{n-j} K(t_j, y_j) + h^\alpha \sum_{j=0}^m \omega_{nj} K(t_j, y_j) \quad (3.33)$$

where $n = 1, 2, \dots, N$ and the convolution weights ω_n are given by the generating function

$$\omega(\xi) = \left(\sum_{j=1}^p \frac{1}{j} (1 - \xi)^j \right)^{-\alpha}, \quad \omega(\xi) = \sum_{j=0}^{+\infty} \omega_j \xi^j,$$

and the starting weights ω_{nj} are given by the solution of the linear equation (3.31).

Chapter 4

Stability regions of numerical methods for solving ODEs.

4.1 One-step methods

In this chapter, we will study the stability regions of numerical methods for solving ODEs. Runge-Kutta method is one of the many types of one-step methods for solving ODEs numerically. These methods are classified into explicit and implicit methods which we will only state their definitions and show how to determine their stability regions. Some other definitions we deemed necessary on this chapter includes consistence, convergence, truncation error and global error for the numerical methods.

Consider the following non-linear equation

$$y'(t) = f(t, y(t)), \quad t > 0 \tag{4.1}$$

$$y(0) = y_0, \tag{4.2}$$

Let N be a fixed positive integer. Let $0 = t_0 < t_1 < t_2 < \dots < t_k < \dots < t_N = 1$ be a partition of $[0, 1]$ and h the step size. At the $t_k = k/N, j = 1, 2, \dots, N$, we can now proceed to summarize the Runge-Kutta methods in the following sections.

4.1.1 Explicit second order Runge-Kutta method

The explicit second order Runge-Kutta method is given by

$$y_{n+1} = y_n + h(w_1k_1 + w_2k_2), \quad (4.3)$$

$$k_1 = f(t_n, y_n)$$

$$, k_2 = f(t_n + \alpha_2h, y_n + \beta_{21}hk_1), \quad (4.4)$$

or

$$y_{n+1} = y_n + h\phi(x_n, y_n; h), \quad (4.5)$$

$$y(0) = y_0, \quad (4.6)$$

where

$$\phi(t_n, y_n; h) = w_1f(t_n, y_n) + w_2f(x_n + \alpha_2h, y_n + \beta_{21}hf(t_n, y_n)) \quad (4.7)$$

Now let us study the *truncation error*, *consistence*, *error estimates*, *convergence* and *stability* of the numerical methods.

Definition 4.1.1 ([34] **Truncation error**). The truncation error of (4.5)- (4.6) is defined by

$$\tau_n(h) = \frac{y(t_{n+1}) - y(t_n) - h\phi(t_n, y(t_n); h)}{h}$$

where $\phi(t_n, y_n; h)$ is given by (4.7).

In theory, truncation error is the residual that is obtained by inserting the solution of differential equation into the numerical methods. These errors are also the errors committed when a limiting process are broken off before one has come to the limiting value. It occurs for example when a finite series is cut-off after a finite number of terms or when a derivatives is approximated with a different quotient.

Remark 1. Replacing y_n by the exact solution $y(t_n)$ in the numerical method (4.5)- (4.6), we obtain the truncation error.

Definition 4.1.2 ([34]). The numerical method (4.5)- (4.6) is consistent if, for fixed $t_n = nh$,

$$\lim_{h \rightarrow 0} |\tau_n(h)| \rightarrow 0$$

Definition 4.1.3 ([34]). The global error is defined by

$$e_n = y(t_n) - y_n, \quad n = 0, 1, 2, \dots$$

where $y(t_n)$ is the exact solution, y_n is the approximate solution.

Definition 4.1.4 ([34]). The numerical method (4.5)- (4.6) is convergent if the global error goes to zero as h goes to zero, i.e.

$$|e_n| = |y(t_n) - y_n| \rightarrow 0, \quad \text{as } h \rightarrow 0$$

Definition 4.1.5 ([34] zero-stability). The numerical method (4.5)- (4.6) is zero-stable if there exist a constant $M > 0$ such that for two sequences y_n and z_n generated by different initial values y_0 and z_0 , we have

$$|y_n - z_n| \leq M|y_0 - z_0|.$$

To prove zero-stability of a numerical method is quite easy as we need to apply the root condition.

Theorem 4.1.6 ([34]). For a consistent numerical method, the numerical method is zero-stable if and only if the numerical method is convergent.

Definition 4.1.7 ([34]). The characteristic polynomial of the one-step explicit second order Runge-Kutta method is given by

$$\rho(z) = z - 1$$

Here the root of the characteristic polynomial is $z = 1$ which is a simple root and lies on the unit circle. By root condition, the second order Runge-Kutta method is zero-stable. Assume $f(t, y)$ satisfies the Lipschitz condition with respect to y , i.e.

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2|.$$

The exact solution of (4.1)- (4.2) satisfies

$$\begin{aligned} y(t_{n+1}) - y(t_n) &= h\phi(t_n, y(t_n); h) + \tau_n(h) \\ y_{n+1} - y_n &= h\phi(t_n, y_n; h) \end{aligned} \tag{4.8}$$

Assume that $e_n = y(t_n) - y_n$, then we have

$$e_{n+1} = e_n + h(\phi(y(t_n), t_n; h) - \phi(y_n, t_n; h)) + h\tau_n$$

Thus

$$\begin{aligned} |e_{n+1}| &\leq |e_n| + hL|e_n| + h|\tau_n| \\ &= (1 + hL)|e_n| + h|\tau_n| \end{aligned}$$

that is

$$\begin{aligned} |e_n| &\leq (1 + hL)^n |e_0| + [1 + (1 + hL) + \dots + (1 + hL)^{n-1}] T \\ &= (1 + hL)^n |e_0| + \frac{(1 + hL)^n - 1}{hL} T, \\ &\leq (1 + hL)^n |e_0| + \frac{e^{L(t_n - t_0)} - 1}{hL} T. \end{aligned}$$

Suppose $t_n = nh$, then we get the error estimate,

$$|e_n| \leq (1 + hL)^n |e_0| + \frac{T}{hL} [e^{L(t_n - t_0)} - 1],$$

Next, we need to study the stable regions for the numerical methods and plot the stability regions by using boundary-locus method. To understand the idea, let us consider the test equation,

$$y'(t) = \beta y(t), \quad t > 0, \tag{4.9}$$

$$y(0) = y_0. \tag{4.10}$$

where $\beta \in \mathbb{C}$ is a complex number. The exact solution of (4.9)- (4.10) is $y(t) = e^{\beta t} y_0$. It is easy to see that, for $\beta < 0$ or $Re(\beta) < 0$

$$|y| = |e^{\beta t} y_0| \rightarrow 0$$

as $t \rightarrow \infty$.

Assume that $Re(\beta) < 0$ and the parameters in (4.3)- (4.4) are given by $w_1 = w_2 = 1/2$ and $\alpha_2 = \beta_{21} = 1$. Let us consider the stability region of the second order Runge-Kutta method. Applying the second order Runge-Kutta method into the test equation (4.9)-(4.10), we get

$$y_{n+1} = \left(1 + \lambda h + \frac{(\lambda h)^2}{2}\right) y_n. \quad (4.11)$$

Assume that $z = \lambda h$ and $Q(z) = 1 + z + \frac{z^2}{2}$, then the absolute stability region of the numerical method (4.3)- (4.4) is

$$|Q(z)| = \left|1 + z + \frac{z^2}{2}\right| < 1.$$

Let us use the boundary locus method to find the stability region of (4.11). Assume that (4.11) has the solution $y_n = \xi^n$, then we have

$$\xi^{n+1} = \left(1 + z + \frac{z^2}{2}\right) \xi^n$$

or

$$\xi = 1 + z + \frac{z^2}{2}.$$

The stability region of second order Runge-Kutta method satisfies $|\xi| \leq 1$. Thus the boundary of the stability region is

$$\left\{z \in \mathbb{C} : 1 + z + \frac{z^2}{2} = e^{i\theta}, 0 \leq \theta \leq 2\pi\right\}. \quad (4.12)$$

Thus in order to obtain the stability region shown in Fig.(4.1) below, we need to plot the roots of the quadratic equation $z^2 + 2z + 2 - 2e^{i\theta} = 0$.

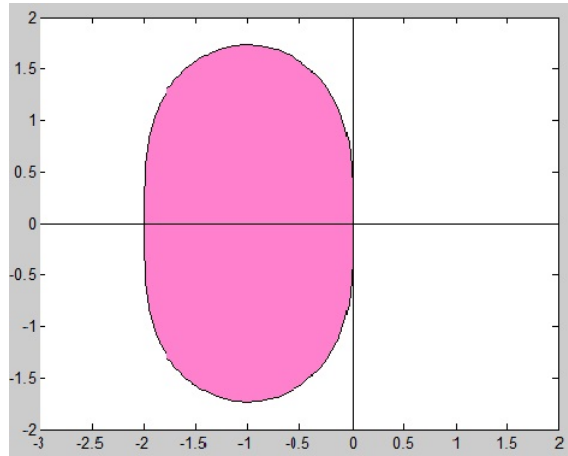


Figure 4.1: Stability region (shaded) of second order Runge-Kutta method

4.1.2 Implicit second order-Runge-Kutta method

Here we define the implicit second order Runge-Kutta method

$$y_{n+1} = y_n + h(w_1k_1 + w_2k_2) \quad (4.13)$$

$$k_1 = f(x_n + h\alpha_1, y_n + h(\lambda_{11}k_1 + \lambda_{12}k_2)) \quad (4.14)$$

$$k_2 = f(x_n + \alpha_2h, y_n + h(\lambda_{21}k_1 + \lambda_{22}k_2)) \quad (4.15)$$

Example 2. Applying the general two-stage implicit Runge-Kutta method into the test equation gives

$$k_1 = [1 + \beta h(\lambda_{12} - \lambda_{22})]\beta y_n / \Delta$$

$$k_2 = [1 + \beta h(\lambda_{21} - \lambda_{11})]\beta y_n / \Delta$$

where Δ is the determinant of the matrix $I - \lambda hA$ and with

$$A = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}$$

Using test equation into (4.14) and (4.15), we have

$$k_1 = \beta(y_n + h(\lambda_{11}k_1 + \lambda_{12}k_2))$$

$$k_2 = \beta(y_n + h(\lambda_{21}k_1 + \lambda_{22}k_2))$$

which gives

$$\begin{aligned} (1 - \beta h \lambda_{11})k_1 - \beta h \lambda_{12}k_2 &= \beta y_n \\ (1 - \beta h \lambda_{22})k_2 - \beta h \lambda_{21}k_1 &= \beta y_n \end{aligned} \tag{4.16}$$

Here we write the form of $Ak = b$,

$$\begin{bmatrix} 1 - \beta h \lambda_{11} & -\beta h \lambda_{12} \\ -\beta h \lambda_{21} & 1 - \beta h \lambda_{22} \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \beta y_n \\ \beta y_n \end{bmatrix}$$

Inverse matrix gives

$$A^{-1} = \frac{1}{\Delta} \begin{bmatrix} 1 - \beta h \lambda_{22} & \beta h \lambda_{12} \\ \beta h \lambda_{21} & 1 - \beta h \lambda_{11} \end{bmatrix}$$

where

$$\Delta = 1 - \frac{1}{2}\beta h + \frac{1}{12}(\beta h)^2 \tag{4.17}$$

Now $A^{-1}b = k$ gives

$$\frac{1}{\Delta} \begin{bmatrix} 1 - \beta h \lambda_{22} & \beta h \lambda_{12} \\ \beta h \lambda_{21} & 1 - \beta h \lambda_{11} \end{bmatrix} \begin{bmatrix} \beta y_n \\ \beta y_n \end{bmatrix} = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

Thus,

$$k_1 = [1 + \beta h(\lambda_{12} - \lambda_{22})]\beta y_n / \Delta \tag{4.18}$$

$$k_2 = [1 + \beta h(\lambda_{21} - \lambda_{11})]\beta y_n / \Delta \tag{4.19}$$

Example 3. For the method defined by the Butcher tableau,

$\frac{1}{6}(3 - \sqrt{3})$	$\frac{1}{4}$	$\frac{1}{12}(3 - 2\sqrt{3})$
$\frac{1}{6}(3 + \sqrt{3})$	$\frac{1}{12}(3 + 2\sqrt{3})$	$\frac{1}{4}$
	$\frac{1}{2}$	$\frac{1}{2}$

It is possible to deduce that $y_{n+1} = R(\lambda h)y_n$, where

$$R(\lambda h) = \frac{1 + \frac{1}{2}\lambda h + \frac{1}{12}\lambda^2 h^2}{1 - \frac{1}{2}\lambda h + \frac{1}{12}\lambda^2 h^2}$$

The solution to Example 3 is straight-forward and is left as an exercise.

Definition 4.1.8 (absolute stability). A linear one-step method is said to be A -stable if the absolute stability region contains the negative (left) complex half-plane.

Lemma 4.1.9. The implicit second-stage Runge-Kutta method is A -stable.

Proof. By writing $R(z)$ in the factorized form, we have

$$|R(z)| = \left| \frac{(z+p)(z+q)}{(z-p)(z-q)} \right| = \left| \frac{z^2 + z(p+q) + pq}{z^2 - z(p+q) + pq} \right|$$

Let $z = x + iy$, $p + q = p$ and $pq = q$, then

$$\begin{aligned} |R(z)| &= \left| \frac{(x+iy)^2 + p(x+iy) + q}{(x+iy)^2 - p(x+iy) + q} \right| \\ &= \frac{\sqrt{((x+p)x - y^2 + q)^2 + (y(2x+p))^2}}{\sqrt{((x-p)x - y^2 + q)^2 + (y(2x-p))^2}} < 1 \end{aligned}$$

for any $Re(z) < 0$. □

The implicit second order Runge-Kutta method is A -stable. The stable region of the second order implicit Runge-Kutta method is the left half plane as shown in Fig.(4.2) below. Notice that the absolute stable region of the explicit second-stage Runge-Kutta

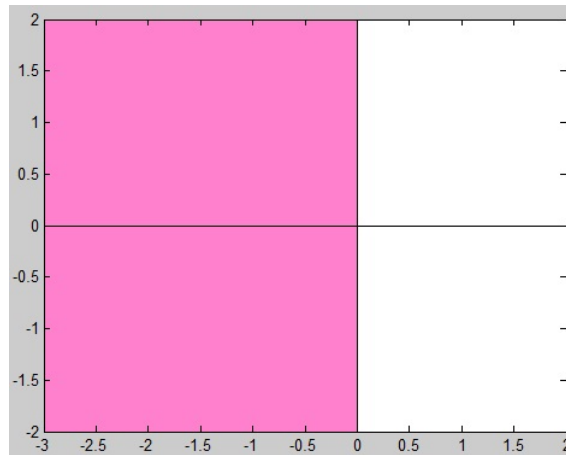


Figure 4.2: Stability region (shaded) of implicit second order Runge-Kutta method

method is $\left| 1 + z + \frac{z^2}{2} \right| < 1$.

4.1.3 Explicit third-stage Runge-Kutta method

We define the third order Runge-Kutta method

$$\begin{aligned}
 y_{n+1} &= y_n + h(w_1k_1 + w_2k_2 + w_3k_3) \\
 k_1 &= f(x_n, y_n) \\
 k_2 &= f(x_n + \alpha_2h, y_n + \beta_{21}hk_1) \\
 k_3 &= f(x_n + \alpha_3h, y_n + h(\beta_{31}k_1 + \beta_{32}k_2))
 \end{aligned} \tag{4.20}$$

where (4.20) is given as (4.5)- (4.6) and

$$\begin{aligned}
 \phi(x_n, y_n; h) &= w_1f(x_n, y_n) + w_2f(x_n + \alpha_2h, y_n + \beta_{21}hf(x_n, y_n)) \\
 &+ w_3f(x_n + \alpha_3h, y_n + h(\beta_{31}f(x_n, y_n) + \beta_{32}f(x_n + \alpha_2h, \\
 &+ y_n + \beta_{21}hf(x_n, y_n))))
 \end{aligned}$$

The parameters are given in the Butcher tableau below,

$\frac{1}{2}$	$\frac{1}{2}$	
1	-1	2
	$\frac{1}{6}$	$\frac{2}{3}$
	$\frac{1}{6}$	$\frac{1}{6}$

Applying (4.20) into the test equation we have

$$y_{n+1} = \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6}\right) y_n. \tag{4.21}$$

For y_0 given,

$$\begin{aligned}
 y_1 &= \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6}\right) y_0 \\
 &\vdots \\
 y_n &= \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6}\right)^n y_0 \\
 &\vdots \\
 y_{n+1} &= \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6}\right)^{n+1} y_0
 \end{aligned}$$

where $Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6}$, then $|y_{n+1}| \rightarrow 0$. Otherwise $|y_{n+1}| \rightarrow \infty$. The absolute stability region of the explicit third order Runge-Kutta method which is shown in Fig.(4.3) below, is given by

$$Q(z) = \left|1 + z + \frac{z^2}{2} + \frac{z^3}{6}\right| < 1.$$

Again, we consider the use of boundary locus method to determine the stability region of (4.21). Assume that (4.21) has the solution $y_n = \xi^n$, then we have

$$\xi^{n+1} = \left(1 + z + \frac{z^2}{2} + \frac{z^3}{6}\right) \xi^n$$

or

$$\xi = \left(1 + z + \frac{z^2}{2} + \frac{z^3}{6}\right)$$

The stability region of third order Runge-Kutta method satisfies $|\xi| \leq 1$. Thus the boundary of the stability region is

$$\left\{z \in \mathbb{C} : 1 + z + \frac{z^2}{2} + \frac{z^3}{6} = e^{i\theta}, 0 \leq \theta \leq 2\pi\right\}. \quad (4.22)$$

Thus in order to obtain the stability region shown in Fig.(4.3) below, we need to plot the roots of the polynomial $z^3 + 3z^2 + 6z + 6 - 6e^{i\theta} = 0$.

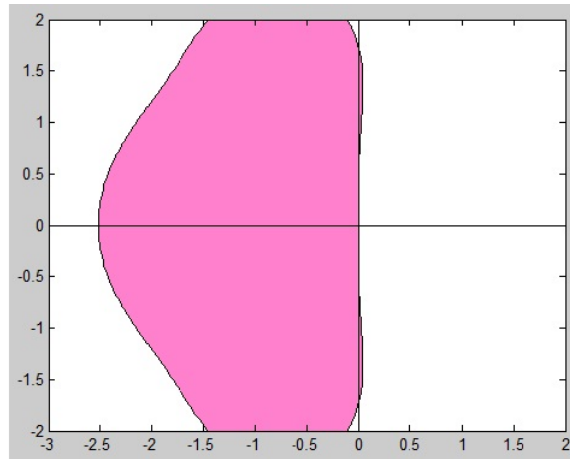


Figure 4.3: Stability region (shaded) of third order Runge-Kutta method

4.1.4 Explicit fourth-stage Runge-Kutta method

Here we define the fourth order Runge-Kutta method as.

$$y_{n+1} = y_n + h(w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4) \quad (4.23)$$

$$k_1 = f(x_n, y_n)$$

$$k_2 = f(x_n + \alpha_2h, y_n + \beta_{21}hk_1)$$

$$k_3 = f(x_n + \alpha_3h, y_n + h(\beta_{31}k_1 + \beta_{32}k_2))$$

$$k_4 = f(x_n + \alpha_4h, y_n + h(\beta_{41}k_1 + \beta_{42}k_2 + \beta_{43}k_3))$$

where (4.23) is given as (4.5)- (4.6) and

$$\begin{aligned} \phi(x_n, y_n; h) &= w_1 f(x_n, y_n) + w_2 f(x_n + \alpha_2 h, y_n + \beta_{21} h f(x_n, y_n)) \\ &+ w_3 f((x_n + \alpha_3 h, y_n + h(\beta_{31} k_1 + \beta_{32} k_2)) \\ &+ w_4 f(x_n + \alpha_4 h, y_n + h(\beta_{41} f(x_n, y_n) + \beta_{42} f(x_n + \alpha_2 h, y_n + \\ &+ \beta_{21} h f(x_n, y_n)) + \beta_{43} f(x_n + \alpha_3 h, y_n + h(\beta_{31} f(x_n, y_n) \\ &+ \beta_{32} f(x_n + \alpha_2 h, y_n + \beta_{21} h f(x_n, y_n)))))) \end{aligned}$$

This particular method is also called the Classical fourth-order Runge-Kutta method. It is popularly used compared to other Runge-Kutta methods because it gives a better convergence. The parameters are given in the Butcher tableau below

1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Similarly we begin by applying the general form into the test equation such that

$$y_{n+1} = \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24} \right) y_n \tag{4.24}$$

Thus, for y_0 given we have

$$\begin{aligned} y_1 &= \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24} \right) y_0 \\ &\vdots \\ y_n &= \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24} \right)^n y_0 \\ &\vdots \\ y_{n+1} &= \left(1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24} \right)^{n+1} y_0. \end{aligned}$$

Let $z = \lambda h$, we get

$$Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}.$$

The absolute stability region for this numerical method is defined by

$$|Q(z)| = \left| 1 + \lambda h + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \right| < 1.$$

We consider the use of boundary locus method to determine the stability region of (4.24). Assume that (4.24) has the solution $y_n = \xi^n$, then we have

$$\xi = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}$$

The stability region of the fourth order Runge-Kutta method satisfies $|\xi| \leq 1$. Thus the boundary of the stability region is

$$\left\{ z \in \mathbb{C} : 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} = e^{i\theta}, 0 \leq \theta \leq 2\pi \right\}. \quad (4.25)$$

Thus in order to obtain the stability region shown in Fig.(4.4) below, we need to plot the roots of the polynomial $z^4 + 4z^3 + 12z^2 + 24z + 24 - 24e^{i\theta} = 0$.

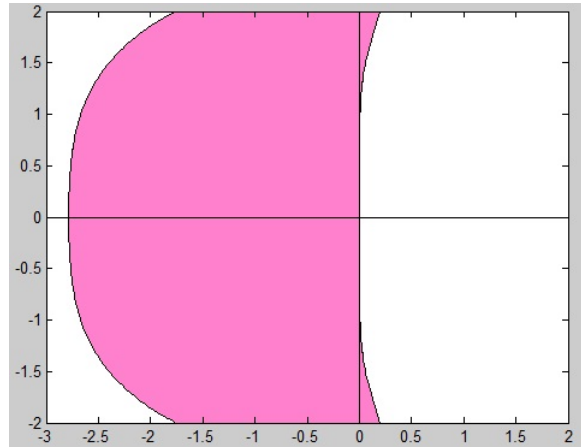


Figure 4.4: Stability region (shaded) of fourth order Runge-Kutta method

4.2 Multistep methods

In this section, we will consider the multistep method for solving initial value problem

$$y'(t) = f(t, y(t)), \quad t > 0 \quad (4.26)$$

$$y(0) = y_0, \quad (4.27)$$

Let $0 = t_0 < t_1 < t_2 < \dots < t_k < \dots < t_N = T$ be a partition of $[0, T]$. Let h be the step size. We consider the linear k -step method

$$\sum_{j=0}^k \alpha_j y_{n+j} = h \sum_{j=0}^k \beta_j f(t_{n+j}, y_{n+j}), \quad (4.28)$$

$$\text{where } y_0, y_1, y_2, \dots, y_{s-1} \text{ are given.} \quad (4.29)$$

Here $\alpha_0, \alpha_1, \dots, \alpha_k$ and $\beta_0, \beta_1, \dots, \beta_k$ are real constants. We assume that $\alpha_k \neq 0, \alpha_0^2 + \beta_0^2 \neq 0$. If $\beta_k = 0$, then the k -step method is then said to be explicit. If $\beta_k \neq 0$, the k -step method is then said to be implicit.

4.2.1 Explicit Adams-Bashforth methods

This explicit numerical method with $k = 1, 2, 3, 4$ are defined as follows:

For $k = 1$, the method is forward Euler methods,

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{4.30}$$

For $k = 2$, we have

$$y_{n+2} = y_{n+1} + h \left(\frac{3}{2}f(t_{n+1}, y_{n+1}) - \frac{1}{2}f(t_n, y_n) \right) \tag{4.31}$$

For $k = 3$, we have

$$y_{n+3} = y_{n+2} + h \left(\frac{23}{12}f(t_{n+2}, y_{n+2}) - \frac{16}{12}f(t_{n+1}, y_{n+1}) + \frac{5}{12}f(t_n, y_n) \right) \tag{4.32}$$

For $k = 4$, we have

$$y_{n+4} = y_{n+3} + h \left(\frac{55}{24}f(t_{n+3}, y_{n+3}) - \frac{59}{24}f(t_{n+2}, y_{n+2}) + \frac{55}{24}f(t_{n+1}, y_{n+1}) - \frac{5}{12}f(t_n, y_n) \right) \tag{4.33}$$

4.2.2 Implicit Adams-Moulton methods

This particular numerical methods are similar to Adams-Bashforth method in the sense they have the same α 's coefficients. As we said in the begining of our discussion, if $\beta_k \neq 0$, then the numerical method is implicit. This effectively means that the k -stage Adams-Moulton can attain order $k + 1$. In contrast, k -stage Adams-Bashforth method can only attain order k .

The Adams-Moulton methods with $k = 0, 1, 2, 3, 4$ are defined as follows: [34]

For $k = 1$, we have

$$y_n = y_{n-1} + hf(t_n, y_n) \quad (4.34)$$

For $k = 2$, we have

$$y_{n+1} = y_n + h \left(\frac{1}{2}f(t_{n+1}, y_{n+1}) - \frac{1}{2}f(t_n, y_n) \right) \quad (4.35)$$

For $k = 3$, we have

$$y_{n+2} = y_{n+1} + h \left(\frac{5}{12}f(t_{n+2}, y_{n+2}) + \frac{2}{3}f(t_{n+1}, y_{n+1}) - \frac{1}{12}f(t_n, y_n) \right) \quad (4.36)$$

For $k = 4$, we have

$$\begin{aligned} y_{n+3} = & y_{n+2} + h \left(\frac{3}{8}f(t_{n+3}, y_{n+3}) + \frac{19}{24}f(t_{n+2}, y_{n+2}) - \frac{5}{24}f(t_{n+1}, y_{n+1}) \right. \\ & \left. + \frac{1}{24}f(t_n, y_n) \right) \end{aligned} \quad (4.37)$$

Remark 4. Numerical methods with $k = 1, 2$ are one-step methods and those with $k > 2$ are $k - 1$ -methods. The methods (4.34) and (4.35) are called the backward Euler and Trapezoidal methods, respectively.

Having both explicit and implicit numerical methods to our disposal, it is worth defining the truncation error, and stability of the multistep methods. The definition of consistence, error estimates is the same as in single-step methods.

Definition 4.2.1. The truncation error of the multi-step methods (4.28) is defined by

$$\tau_n(h) = \frac{\sum_{j=0}^k \alpha_j y(x_{n+j}) - h \sum_{j=0}^k \beta_j f(x_{n+j}, y(x_{n+j}))}{h}.$$

Note that in some books, the truncation error is defined by

$$\tau_n(h) = \frac{\sum_{j=0}^k \alpha_j y(x_{n+j}) - h \sum_{j=0}^k \beta_j f(x_{n+j}, y(x_{n+j}))}{h \sum_{j=0}^k \beta_j}.$$

where $\sum_{j=0}^k \beta_j$ is a constant.

Definition 4.2.2. The numerical method (4.28) is said to have order of accuracy $p \geq 1$, if

$$|\tau_n| \leq Ch^p, \quad 0 \leq h \leq h_0,$$

where

$$\tau_n(h) = \frac{\sum_{j=0}^k \alpha_j y(x_{n+j}) - h \sum_{j=0}^k \beta_j f(x_{n+j}, y(x_{n+j}))}{h}.$$

Before we could consider the stability regions of this method, let us recall that in the one-step methods, it is proved that consistence and convergence guarantees stability. Unfortunately, it is not always the case with linear multistep methods. The outcome is not really simple to determine. Though it is possible for a convergent multi-step method to be consistence, but consistency alone is not sufficient to guarantee convergence (et al [34]).

4.2.3 Stability regions of multistep methods

The solutions of one-step method only depend on the initial value y_0 . However the solutions of linear multistep methods depend on $y_0, y_1, y_2, \dots, y_{k-1}$. These values are mostly negotiated by the one-step methods.

Definition 4.2.3 ([34] Root condition). A linear multistep method (4.28) is zero-stable if and only if all the roots of the first characteristic polynomial

$$\rho(z) = \sum_{j=0}^k \alpha_j z^j,$$

are inside the closed unit disc in the complex plane, with any which lie on the unit circle being simple. Infact, linear multi-step methods (LMMs) are zero-stable if and only if the root conditions above are satisfied.

Applying (4.28) into the test equation (4.9)- (4.10) we have

$$\sum_{j=0}^k (\alpha_j - z\beta_j) y_{n+j} = 0 \tag{4.38}$$

where $z = \lambda h$. Assume that (4.38) has solution $y_n = \xi^n$, then we have

$$\sum_{j=0}^k (\alpha_j - z\beta_j)\xi^{n+j} = 0$$

or

$$\sum_{j=0}^k (\alpha_j - z\beta_j)\xi^j = 0.$$

The region of absolute stability is the set of all $z \in \mathbb{C}$ for which the numerical method satisfies $|\xi| \leq 1$. If this condition contains left-hand complex plane, the multistep method is said to be A -stable.

Definition 4.2.4 ([34] Absolute stability). A linear multistep method is said to be absolutely stable for a given value z if each root $\omega_r = \omega_r(z)$ of the associated stability polynomial $Q(\omega; z)$ satisfies $|\omega_r(z)| < 1$.

Theorem 4.2.5 ([34] First Dahlquist's Barrier theorem). The zero-stable and linear s -step multistep method cannot exceed an order of convergence greater than $k + 1$ if k is odd and greater than $k + 2$ if k is even. If the method is explicit, it cannot exceed an order greater than k .

Theorem 4.2.6 ([34] Second Dahlquist's Barrier theorem). There are no explicit A -stable and linear multistep methods. The implicit ones have order of convergences at most 2. The trapezoidal rule has the smallest error constant amongst the A -stable linear multi-step methods of order 2.

Theorem 4.2.7 ([34] Dahlquist equivalence). Suppose that a consistent linear multistep method is applied to a sufficiently smooth differential equation and that the starting values y_1, y_2, \dots, y_{k-1} all converge to the initial value y_0 as $h \rightarrow 0$. Then the numerical solution converges to the exact solution as $h \rightarrow 0$ if and only if the method is zero-stable.

Definition 4.2.8 ([34] absolute stability). The region of absolute stability of a linear multi-step method is the set of all points λh in the complex plane for which the method is absolutely stable.

The region of the absolute stability of a numerical method must admit the values of λ , $Re(\lambda) < 0$, so as to ensure that there is no limitation on the step sizes, h for any large $|\lambda|$.

Example 5. Consider the explicit second-stage Adams-Bashforth method

$$y_{n+2} = y_{n+1} + h \left(\frac{3}{2}f(x_{n+1}, y_{n+1}) - \frac{1}{2}f(x_n, y_n) \right)$$

Applying test equation (4.9)-(4.10) gives

$$y_{n+2} = y_{n+1} + \frac{3}{2}h\lambda y_{n+1} - \frac{1}{2}h\lambda y_n$$

which can be simplified to

$$y_{n+2} - \left(1 + \frac{3}{2}z\right)y_{n+1} + \frac{1}{2}zy_n = 0 \tag{4.39}$$

where $z = \lambda h$. Using the same process as previous section, we assume that (4.39) has the solution $y_n = \xi^n$. Then we have

$$\xi^{n+2} - \left(1 + \frac{3}{2}z\right)\xi^{n+1} + \frac{1}{2}z\xi_n = 0$$

or

$$\xi = \frac{1}{2} \left(1 + \frac{3}{2}z \pm \sqrt{\frac{9}{4}z^2 + z + 1} \right) \tag{4.40}$$

The stability region of explicit second stage Adams-Bashforth method satisfies $|\xi| \leq 1$.

Thus the boundary of the stability region where $\xi = e^{i\theta}$ is

$$\left\{ z \in \mathbb{C} : \frac{1}{2} \left(1 + \frac{3}{2}z \pm \sqrt{\frac{9}{4}z^2 + z + 1} \right) = e^{i\theta}, 0 \leq \theta \leq 2\pi \right\}.$$

Thus in order to obtain the stability region shown in Fig.(4.5) below, we need to plot the roots of the equation $\left(1 + \frac{3}{2}z \pm \sqrt{\frac{9}{4}z^2 + z + 1}\right) - 2e^{i\theta} = 0$.

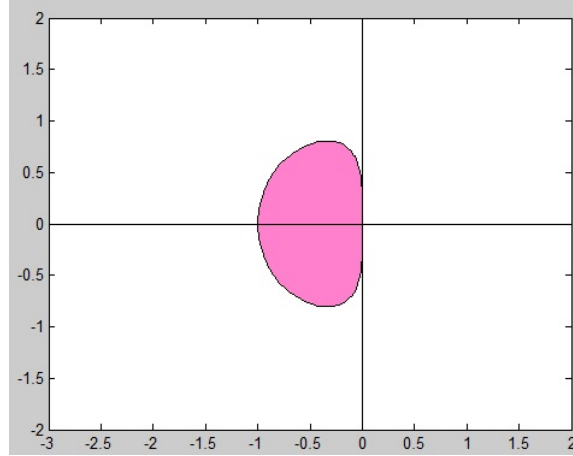


Figure 4.5: Stability region (shaded) of second order Adams-Bashforth method.

Example 6. Consider the second-stage Adams-Moulton method which is also known as the trapezoidal rule.

Applying test equation we have

$$y_{n+1} = y_n + \frac{\lambda h}{2} (y_{n+1} + y_n), \quad \lambda < 0$$

Simplifying, where $z = \lambda h$, we have

$$\left(1 - \frac{1}{2}z\right) y_{n+1} - \left(1 + \frac{1}{2}z\right) y_n = 0 \tag{4.41}$$

We assume that (4.41) has the solution $y_n = \xi^n$. Then we have

$$\left(1 - \frac{1}{2}z\right) \xi - \left(1 + \frac{1}{2}z\right) = 0$$

or

$$\xi = \left(\frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}\right) \tag{4.42}$$

The stability region for this numerical method satisfies $|\xi| \leq 1$. Thus, the boundary for the stability region where $\xi = e^{i\theta}$ is

$$\left\{ z \in \mathbb{C} : \left(\frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}\right) = e^{i\theta}, 0 \leq \theta \leq 2\pi \right\}.$$

We remark that this numerical method is absolutely stable and A -stable as shown in Fig.(4.2).

Example 7. Consider the third-stage Adams-Moulton method defined by (4.36)

Applying test equation we have

$$y_{n+2} = y_{n+1} + \frac{5}{12}h\lambda y_{n+2} + \frac{2}{3}h\lambda y_{n+1} - \frac{1}{12}h\lambda y_n$$

which can be simplified to

$$\left(1 - \frac{5}{12}z\right)y_{n+2} - \left(1 + \frac{2}{3}z\right)y_{n+1} + \frac{1}{12}zy_n = 0$$

where $z = \lambda h$. Applying the same process in Example 6, we define the boundary stability region for this numerical method by

$$\left\{ z \in \mathbb{C} : \frac{6}{12 - 5z} \left(1 + \frac{2}{3}z \pm \sqrt{1 + z + \frac{7}{12}z^2} \right) = e^{i\theta}, 0 \leq \theta \leq 2\pi \right\}$$

Remark 8. Explicit multistep methods can never be A -stable, just like the explicit Runge-Kutta method we have discussed in the previous chapter (see Theorem 4.2.6). Also in our previous section, we see that implicit Runge-Kutta methods are A -stable but not all implicit multi-step methods are A -stable. Implicit multistep methods can only be A -stable if their order is at most order 2. An example of a second-order A -stable method is the trapezoidal method, which is also known as implicit Adam-Moulton second-stage method. Note also that trapezoidal method is an implicit one-step method.

Chapter 5

Stability regions of numerical methods for solving FDEs

There are different ways to find the stability regions of the numerical methods for solving fractional differential equations. In this chapter, we will consider three ways: (1) Lubich convolution quadrature method. (2) Garrappa predictor-corrector algorithm. (3) Discrete stability polynomial method.

5.1 Lubich's convolution quadrature method [3].

Let us consider

$${}^C D_t^\alpha y(t) = g(y(t)), \quad 0 < \alpha < 1, \quad t > 0 \quad (5.1)$$

$$y(0) = y_0 \quad (5.2)$$

It is well known that (5.1)-(5.2) is equivalent to the Volterra integral equation

$$y(t) = y_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} g(y(s)) ds. \quad (5.3)$$

In [3], Lubich studied the stability region of the numerical method for solving the general Volterra integral equation

$$y(t) = f(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} g(y(s)) ds, \quad 0 < \alpha < 1, \quad t > 0. \quad (5.4)$$

Let $0 = t_0 < t_1 < t_2 < \dots < t_k < \dots < t_N = T$ be the partition on $[0, T]$. Let h be the step size. Then we have

$$y(t_k) = f(t_k) + \frac{1}{\Gamma(\alpha)} \int_0^{t_k} (t_k - s)^{\alpha-1} g(y(s)) ds. \quad (5.5)$$

The integral can be discretized by a product quadrature rule,

$$\frac{1}{\Gamma(\alpha)} \int_0^{t_k} (t_k - s)^{\alpha-1} g(y(s)) ds \approx h^\alpha \left(\sum_{j=-m}^{-1} \omega_{kj} g(y(t_j)) + \sum_{j=0}^k \omega_{k-j} g(y(t_j)) \right) \quad (5.6)$$

where m is fixed and $y(t_{-m}), \dots, y(t_{-1})$ are given starting values which are usually computed by a difference method.

Assume that $y_k \approx y(t_k)$ denotes the approximate value of $y(t_k)$, then we define the finite difference method of (5.3) by

$$y_k = f_k + h^\alpha \sum_{j=0}^k \omega_{k-j} g(y_j), \quad k \geq 0 \quad (5.7)$$

with

$$f_k = f(t_k) + h^\alpha \sum_{j=-m}^{-1} \omega_{kj} g(y_j) \quad (5.8)$$

where $t_k = mh + kh$, $t_{-m} = 0$, $t_0 = mh, \dots, k \geq 0$. It is noted that there are also many ways to determine convolution and starting weights. As we are concentrating on the stability regions, we will not pay much attention on the starting weights.

Let us consider the test equation,

$$y(t) = f(t) + \frac{1}{\Gamma(\alpha)} \int_0^t (t - s)^{\alpha-1} \beta y(s) ds \quad (5.9)$$

and when applied to (5.7), we have

$$y_k = f_k + h^\alpha \beta \sum_{j=0}^k \omega_{k-j} y_j, \quad (5.10)$$

Assume that $z = h^\alpha \beta$, then we have

$$y_k = f_k + z \sum_{j=0}^k \omega_{k-j} y_j, \quad (5.11)$$

Next we consider the stability region such that $y_k \rightarrow 0$ as $k \rightarrow \infty$. To accomplish this, we introduce some lemmas and definitons.

Lemma 5.1.1 (Lubich [3]). Assume that $f \in C[0, \infty]$ and suppose $f(t)$ has a finite limit as $t \rightarrow \infty$, then the solution of (5.9) satisfies $y(t) \rightarrow 0$ as $t \rightarrow \infty$ where

$$\left\{ \beta : |\arg(\beta) - \pi| < \left(1 - \frac{1}{2}\alpha\right) \pi \right\}$$

Definition 5.1.2 (Definition 1 [3]). The numerical method (5.11) is A -stable if $y_k \rightarrow 0$ as $k \rightarrow \infty$ and for every $h > 0$ and

$$\beta \in \left\{ \beta : |\arg(\beta) - \pi| < \left(1 - \frac{1}{2}\alpha\right) \pi \right\}$$

the analytical stability region of (5.9).

Definition 5.1.3 (Definition 2 [3]). The stability region S of fractional difference methods (FDMs) (5.11) is

$$S = \{z = h^\alpha \beta : y_k \rightarrow 0, \quad k \rightarrow \infty\}$$

The method is called $A(\theta)$ -stable if S contains the sector

$$\{z : |\arg(z) - \pi| < \theta\}.$$

Theorem 5.1.4 (Lubich [3]). Assume that

$$\omega_k = (-1)^k \binom{-\alpha}{k} + v_k, \quad k \geq 0$$

where $(v_k) \in \ell^1$, $\ell^1 = \{(v_k), \sum_{k=1}^{\infty} |v_k| < \infty\}$, then the stability region of (5.11) is

$$S = \{z \in \mathbb{C} : 1 - z\omega(\xi) \neq 0, \quad |\xi| \leq 1\}$$

and

$$\omega(\xi) = \omega_0 + \omega_1\xi + \omega_2\xi^2 + \omega_3\xi^3 + \dots = \sum_{j=0}^{+\infty} \omega_j \xi^j$$

Proof. Suppose $z = h^\alpha \beta$, $z \neq 0$ then (5.11) can be written into

$$y(\xi) = f(\xi) + z\omega(\xi).y(\xi)$$

or

$$y(\xi) = \frac{f(\xi)}{1 - z\omega(\xi)} = \frac{(1 - \xi)^\alpha f(\xi)}{(1 - \xi)^\alpha [1 - z\omega(\xi)]}. \quad (5.12)$$

Here

$$y(\xi) = \sum_{j=0}^{+\infty} y_j \xi^j, \quad f(\xi) = \sum_{j=0}^{+\infty} f_j \xi^j, \quad \omega(\xi) = \sum_{j=0}^{+\infty} \omega_j \xi^j.$$

We first show that

$$S \supseteq \{z \in \mathbb{C} : 1 - z\omega(\xi) \neq 0, \quad |\xi| \leq 1\}.$$

Assume that

$$z \in \{z \in \mathbb{C} : 1 - z\omega(\xi) \neq 0, \quad |\xi| \leq 1\},$$

we will show that

$$(1 - \xi)^\alpha [1 - z\omega(\xi)] \neq 0, \quad |\xi| \leq 1.$$

If $\xi \neq 1$ and $|\xi| \leq 1$, then

$$g(\xi) = (1 - \xi)^\alpha [1 - z\omega(\xi)] \neq 0.$$

Note that

$$\omega_k = (-1)^k \binom{-\alpha}{k} + v_k, \quad k \geq 0, \quad v_k \in \ell^1$$

implies that $\omega(\xi) = (1 - \xi)^\alpha + v(\xi)$ is continuous on $\{\xi \in \mathbb{C} : |\xi| \leq 1, \xi \neq 1\}$ and $\omega(1) = \lim_{\xi \rightarrow 1-\infty} \omega(\xi) = \infty$. Suppose $\xi = 1$, then

$$g(\xi) = (1 - \xi)^\alpha [1 - z\omega(\xi)] = (1 - \xi)^\alpha [1 - zv(\xi)] - z = -z \neq 0,$$

since $v(\xi) = \sum_{j=0}^{+\infty} v_j \xi^j$ converges at $\xi = 1$ and $(v_n) \in \ell^1$. By Wiener's inversion theorem in [3], we have

$$\frac{1}{(1 - \xi)^\alpha [1 - z\omega(\xi)]} \in \ell^1, \quad |\xi| \leq 1.$$

Furthermore, we claim that $(1 - \xi)^\alpha f(\xi)$ converges to zero. Suppose $\hat{f}_k = f_k - f_\infty \rightarrow 0$, we have

$$\begin{aligned} (1 - \xi)^\alpha f(\xi) &= (1 - \xi)^\alpha \left[\frac{f_\infty}{1 - \xi} + \hat{f}(\xi) \right] \\ &= (1 - \xi)^{\alpha-1} f_\infty + (1 - \xi)^\alpha \hat{f}(\xi) \end{aligned}$$

We note that

$$(1 - \xi)^{\alpha-1} f(\xi) = \sum_{k=0}^{+\infty} (-1)^k \binom{-\alpha}{k} \xi^k$$

and

$$(-1)^k \binom{-\alpha}{k} = \frac{k^{\alpha-1}}{\Gamma(\alpha)} [1 + O(k^{-1})].$$

Here the coefficient sequence of $(1 - \xi)^{\alpha-1}$ tends to zero. We also note that $(1 - \xi)^\alpha$ is convergent for $|\xi| \leq 1$. Thus the coefficients sequence of $(1 - \xi)^\alpha$ is in ℓ^1 .

Lemma 5.1.5 (Lubich [3]). Let $(\ell_k) \in \ell^1$, let (c_n) be the space of the sequence convergent to zero, then

$$\lim_{k \rightarrow \infty} \sum_j \ell_j c_{k-j} = \sum_j \ell_j \lim_{k \rightarrow \infty} c_{k-j} = 0$$

By using Lemma 5.1.5 the coefficient of the sequence gives $(1 - \xi)^\alpha \hat{f}(\xi)$ which converges to zero. This effectively shows that $(1 - \xi)^\alpha f(\xi)$ holds. Therefore by (5.12), we can see that the coefficient sequence (y_k) of $y(\xi)$ tends to zero. Hence $z \in S$. Lastly, we prove that S is exhausted by

$$\{z \in \mathbb{C} : 1 - z\omega(\xi) \neq 0, \quad |\xi| \leq 1\}.$$

Assume that $1 - z\omega(\xi_0) = 0$ for some $|\xi_0| < 1$, then we will show that $z \notin S$. If we choose

$$y(\xi) = \frac{(1 - \xi)^\alpha}{\xi - \xi_0} = \frac{(1 - \xi)^\alpha - (1 - \xi_0)^\alpha}{\xi - \xi_0} + (1 - \xi_0)^\alpha \frac{1}{\xi - \xi_0}$$

Lemma 5.1.6 (Lubich [3]). Assume that the coefficient sequence of $a(\xi)$ is in ℓ^1 . Let $|\xi_0| \leq 1$, then the coefficient sequence of

$$b(\xi) = \frac{a(\xi) - a(\xi_0)}{\xi - \xi_0}$$

converges to zero.

Again by Lemma 5.1.6 we get the coefficient sequence of

$$\frac{(1 - \xi)^\alpha - (1 - \xi_0)^\alpha}{\xi - \xi_0} \rightarrow 0$$

while the coefficient sequence of

$$\frac{1}{\xi - \xi_0} = - \sum_{k=0}^{\infty} \xi_0^{-k-1} \xi^k \rightarrow \infty$$

since $|\xi_0^{-k-1}| \rightarrow 0$, $|\xi| \leq 1$. Hence y_k diverges. Again we chose

$$\begin{aligned} f(\xi) &= [1 - z\omega(\xi)]y(\xi) = (1 - \xi)^\alpha [1 - z\omega(\xi)](1 - \xi)^{-\alpha} y(\xi) \\ &= \frac{\{(1 - \xi)^\alpha [1 - z\omega(\xi)] - (1 - \xi_0)^\alpha [1 - z\omega(\xi_0)]\}}{(\xi - \xi_0)}. \end{aligned}$$

The coefficients of $f(\xi) \rightarrow 0$ by Lemma 5.1.6, y_n does not tend to zero. Hence $z \notin S$. \square

5.2 Garrappa predictor-corrector algorithm [31].

In [31], Garrappa studied the stability region of predictor-corrector algorithm for solving fractional differential equation.

Let us consider the following test equation

$${}_0^C D_t^\alpha y(t) = \lambda y(t), \quad \lambda \in \mathbb{C}, \quad 0 < \alpha < 1, \quad (5.13)$$

$$y(0) = y_0. \quad (5.14)$$

The exact solution of (5.13)- (5.14) discussed in [11] has the form

$$y(t) = y_0 E_\alpha(\lambda t^\alpha),$$

where

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

It is noted in [3] that $y(t) \rightarrow 0$ as $t \rightarrow \infty$ when λ lies in

$$S = \{z \in \mathbb{C} : |\arg(z) - \pi| < (1 - \alpha/2)\pi\}$$

This set is called the analytic stability region of (5.13)- (5.14).

Applying the predictor-corrector 1-step Adams product quadrature (APQ) method to (5.13)-(5.14), we get, [31],

$$y_k = f_k + \sum_{j=n}^k \omega_{k-j} y_j, \quad k \geq n \quad (5.15)$$

where $z = h^\alpha \lambda$ and

$$\begin{cases} f_k = (1 + z\zeta_{k,0} + z\beta_0 + z^2\beta_0\mu_{k-1})y_0, \\ \omega_0 = 0, \\ \omega_k = z\beta_k + z^2\beta_0\mu_{k-1}, \quad k \geq 1. \end{cases} \quad (5.16)$$

for some suitable coefficients $\beta_k, \zeta_{k,0}, \mu_{k-1}$.

Recall that in Chapter 4 the stability regions of those numerical methods is the set of all $z = \lambda h$ for which the numerical solution y_k of the test equation behaves as the exact solution and tends to zero as $k \rightarrow \infty$. In fact the same statement is applicable on the fractional order except that $z = h^\alpha \lambda$. In a nutshell, we are concentrating on the set of $z \in \mathbb{C}$ for which the zero solution of (5.15) is uniformly asymptotically stable [31]. Next let us consider the main result in Garrappa [31] about the stability region of the numerical method.

Theorem 5.2.1 (Garrappa [31]). Let the sequence f_k of starting terms be convergent and let the quadrature weights ω_k satisfy

$$\omega_k = \frac{k^{\alpha-1}}{\Gamma(\alpha)} + v_k, \quad k \geq n+1$$

with

$$\sum_{k=1}^{\infty} |v_k| < \infty.$$

The stability region of the convolution quadrature (5.15) is given by

$$S = \{z \in \mathbb{C} : 1 - \omega(\xi) \neq 0, |\xi| \leq 1\}$$

where $\omega(\xi) = \sum_{k=0}^{\infty} \omega_k \xi^k$ is the generating power series of ω_k .

It appears that with Theorem 5.2.1, $\mu(\xi) = \sum_{k=0}^{\infty} \mu_k \xi^k$ and $\beta(\xi) = \sum_{k=0}^{\infty} \beta_k \xi^k$ will serve as generating power series for μ_k and β_k respectively.

Proposition 9 (Garrappa [31]). The stability region of (5.15) shown in Fig.(5.1) is

$$S = \{z \in \mathbb{C} : 1 - z(\beta(\xi) - \beta_0) - z^2\beta_0\xi\mu(\xi) \neq 0, |\xi| \leq 1\}$$

where $\beta(\xi) = \sum_{k=0}^{\infty} \beta_k \xi^k$, $\mu(\xi) = \sum_{k=0}^{\infty} \mu_k \xi^k$.

Proof. We note that

$$\begin{aligned} \omega_k &= z\beta_k + z^2\beta_0\mu_{k-1} \\ &= z \left[\frac{1}{\Gamma(\alpha)}k^{\alpha-1} + O(k^{\alpha-3}) \right] + z^2\beta_0 \left[\frac{1}{\Gamma(\alpha)}(k-1)^{\alpha-1} + O(k^{\alpha-2}) \right] \\ &= \dots = \frac{1}{\Gamma(\alpha)}k^{\alpha-1} + v_k, \quad \sum_{k=1}^{\infty} |v_k| < \infty. \end{aligned}$$

We further note that $\omega(\xi) = z(\beta(\xi) - \beta_0) + z^2\beta_0\xi\mu(\xi)$. The proof is complete. □

Next, we use the boundary locus method to justify Proposition 9. Let $\xi = e^{i\theta}$, $0 \leq \theta \leq 2\pi$, we can then find the roots of

$$1 - z(\beta(\xi) - \beta_0) - z^2\beta_0\xi\mu(\xi) = 0.$$

In Fig.(5.1) below, we choose $\alpha = 0.7$, $\beta(\xi) = \sum_{k=0}^N \beta_k \xi^k$, $N = 2000$ and $\theta = 0 : h : 2\pi$. We plot the boundary of the stability region. The stability region of this numerical method is inside of the boundary.

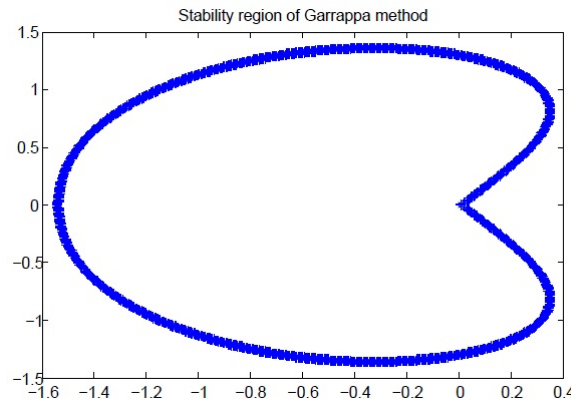


Figure 5.1: Stability region of Garrappa algorithm

5.3 Discrete stability polynomial method

In this chapter, we will consider the stability region of the finite difference method of fractional differential equation by using discrete stability polynomial method. The idea is similar to the method previously discussed in Chapter 4. We first find the discrete stability polynomial of the numerical method for solving FDE. Then we use the boundary locus method to determine the stability region.

5.3.1 Diethelm's method

Let us consider

$${}^C D_t^\alpha y(t) = \beta y(t), \quad y(0) = y_0. \quad (5.17)$$

The equivalent form of (5.17) is

$${}^R D_t^\alpha [y(t_j) - y_0] = \beta y(t_j)$$

The Diethelm's method is

$${}^R D_t^\alpha y(t_j) = \frac{1}{h^\alpha} \sum_{k=0}^j \omega_{kj} y(t_j - t_k) + O(h^{2-\alpha})$$

Assume that y_j is the approximate solution of $y(t_j)$, then

$$\frac{1}{h^\alpha} \sum_{k=0}^j \omega_{kj} y_{j-k} + D_t^\alpha y_0 = \beta y_j \quad (5.18)$$

We note that

$${}^R D_t^\alpha (y_0) = \frac{y_0}{\Gamma(1-\alpha)} t_j^{-\alpha},$$

such that (5.18) can be written by

$$\frac{1}{h^\alpha} \sum_{k=0}^j \omega_{kj} y_{j-k} - \frac{y_0}{\Gamma(1-\alpha)} t_j^{-\alpha} = \beta y_j$$

or

$$\sum_{k=0}^j \omega_{kj} y_{j-k} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} y_0 = h^\alpha \beta y_j.$$

Assume that $z = \beta h^\alpha$, then

$$\sum_{k=0}^j \omega_{kj} y_{j-k} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} y_0 = z y_j$$

or

$$\begin{aligned} (\omega_{0j} - z)y_j + \omega_{1j}y_{j-1} + \omega_{2j}y_{j-2} + \dots + \omega_{j-1,j}y_1 \\ + \left(\omega_{jj} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} \right) y_0 = 0 \end{aligned}$$

Let $y_j = \xi^j$, we obtain the discrete stability polynomial

$$\begin{aligned} (\omega_{0j} - z)\xi^j + \omega_{1j}\xi^{j-1} + \omega_{2j}\xi^{j-2} + \dots + \omega_{j-1,j}\xi \\ + \left(\omega_{jj} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} \right) = 0 \end{aligned}$$

If we let $\xi = e^{i\theta}$ with $0 \leq \theta \leq 2\pi$ we get

$$(\omega_{0j} - z) (e^{i\theta})^j + \omega_{1j} (e^{i\theta})^{j-1} + \omega_{2j} (e^{i\theta})^{j-2} + \dots + \left(\omega_{jj} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} \right) = 0,$$

then the stability region shown is

$$S = \left\{ z : z = \xi^j \omega_{0j} + \frac{\omega_{1j} \xi^{j-1} + \omega_{2j} \xi^{j-2} + \dots + \left(\omega_{jj} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} \right)}{\xi^j}, \xi = e^{i\theta}, 0 \leq \theta < 2\pi. \right\}$$

In Fig.(5.2), we choose $\alpha = 0.7$, $j = 160$ and $\theta = 0 : h : 2\pi$ with $h = 0.005$, and plot the stability region which lies outside of the boundary.

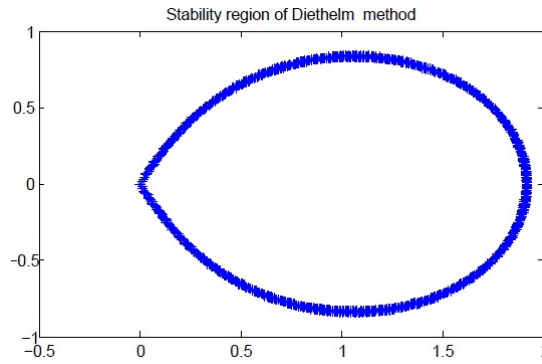


Figure 5.2: Stability region of Diethelm method

5.3.2 Grünwald method

We consider this particular method which is similar to Diethelms' approach.

$${}_0^C D_t^\alpha y(t) = \beta y(t), \quad y(0) = y_0. \quad (5.19)$$

The equivalent form of (5.19) is

$${}_0^R D_t^\alpha [y(t_j) - y_0] = \beta y(t_j)$$

Then the Grünward method is

$${}_0^R D_t^\alpha y(t_j) = \frac{1}{h^\alpha} \sum_{k=0}^j \omega_j y(t_j - t_k) + O(h) \quad (5.20)$$

where

$$\omega_j = (-1)^j \binom{\alpha}{j} = (-1)^j \frac{\alpha(\alpha-1)\dots(\alpha-j+1)}{j!}.$$

We denote that $y(t_j) \approx y_j$, then

$$\frac{1}{h^\alpha} \sum_{k=0}^j \omega_j y_{j-k} + D_t^\alpha y_0 = \beta y_j, \quad (5.21)$$

such that

$$\sum_{k=0}^j \omega_j y_{j-k} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} y_0 = h^\alpha \beta y_j.$$

Assume that $z = \beta h^\alpha$, then

$$\sum_{k=0}^j \omega_j y_{j-k} - \frac{j^{-\alpha}}{\Gamma(1-\alpha)} y_0 = z y_j$$

or

$$z = \frac{(y_j + y_{j-1} - y_{j-2})\omega_j + \dots + \left(\omega_j - \frac{j^{-\alpha}}{\Gamma(1-\alpha)}\right) y_0}{y_j}$$

Let $y_j = \xi^j$, we obtain the characteristic polynomial

$$z = \frac{(\xi^j + \xi^{j-1} - \xi^{j-2})\omega_j + \dots + \left(\omega_j - \frac{j^{-\alpha}}{\Gamma(1-\alpha)}\right) y_0}{\xi^j}$$

Assume that $\xi = e^{i\theta}$ with $0 \leq \theta \leq 2\pi$ we get the stability region

$$S = \left\{ z : z = \frac{((e^{i\theta})^j + (e^{i\theta})^{j-1} - (e^{i\theta})^{j-2})\omega_j + \dots + \left(\omega_j - \frac{j^{-\alpha}}{\Gamma(1-\alpha)}\right) y_0}{(e^{i\theta})^j} \right\}.$$

In Fig.(5.3), we use the same parameters as in Section 5.3.2, we obtain the boundary of the stability region. The stability region is outside of the boundary.

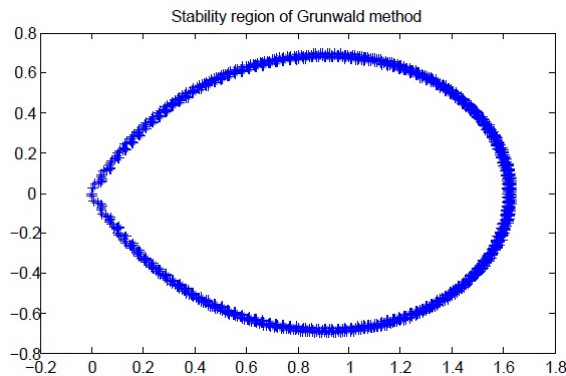


Figure 5.3: Stability region of Grünwald method

5.3.3 Lubich's method

Using equation (5.19) and its equivalent, we get

$$y(t) = y_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t - \tau)^{\alpha-1} \beta y(\tau) d\tau,$$

The Lubich's method applied to the test equation reads

$$y_k = y_0 + \beta h^\alpha \sum_{j=0}^k \omega_{k-j}^{(\alpha)} y_j + \beta h^\alpha \sum_{j=0}^l \omega_{kj}^{(\alpha)} y_j \quad (5.22)$$

Assume that $z = h^\alpha \beta$, then

$$y_k = y_0 + z \sum_{j=0}^k \omega_{k-j}^{(\alpha)} y_j + z \sum_{j=0}^l \omega_{kj}^{(\alpha)} y_j$$

$$z = \frac{y_k - y_0}{\sum_{j=0}^k \omega_{k-j}^{(\alpha)} y_j + \sum_{j=0}^l \omega_{kj}^{(\alpha)} y_j}.$$

Suppose $y_k = \xi^k$, then we obtain the discrete stability polynomial

$$z = \frac{\xi^k - 1}{\sum_{j=0}^k \omega_{k-j}^{(\alpha)} \xi^j + \sum_{j=0}^l \omega_{kj}^{(\alpha)} \xi^j}.$$

If we let $\xi = e^{i\theta}$ for $0 \leq \theta \leq 2\pi$ we get the stability region

$$S = \left\{ z : z = \frac{(e^{i\theta})^k - 1}{\sum_{j=0}^k \omega_{k-j}^{(\alpha)} (e^{i\theta})^j + \sum_{j=0}^l \omega_{kj}^{(\alpha)} (e^{i\theta})^j} \right\}.$$

Again, we obtained in Fig.(5.4) by choosing $\alpha = 0.7$, $j = 1000$ and $\theta = 0 : h : 2\pi$ with $h = 0.0001$, the boundary of the stability region. The stability region is outside of the boundary.

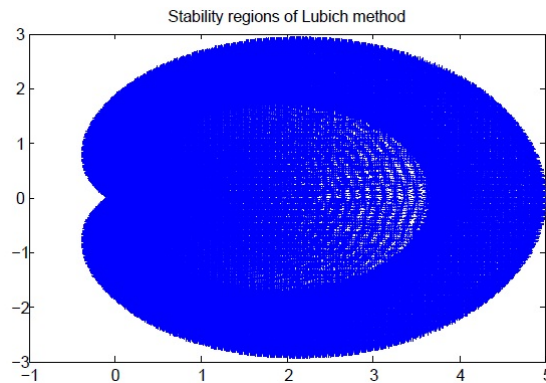


Figure 5.4: Stability region of Lubich method

Chapter 6

Further works

In Section 3.1 via Chapter 3, we reviewed Diethelm's method [15] where the finite-part Hadamard integral is approximated by using piecewise linear interpolation polynomials. We see that the first-degree compound quadrature formula was used to approximate the integral and the order of convergence of the proposed numerical method is $O(h^{2-\alpha})$. The stability regions of this numerical method was investigated and determined.

It is natural to consider the approximation of the finite-part Hadamard integral by a piecewise quadratic polynomial. Then we can define a numerical method for solving FDE and study the stability properties and convergence for such numerical method.

For a start, assume that $N = 2m$, where m denotes a fixed positive integer. Let $0 = t_0 < t_1 < \dots < t_{2j} < t_{2j+1} < \dots < t_{2m} = 1$ be a partition of $[0, 1]$ and h be the step size. At point $t_{2j} = 2j/2m$, the equation (3.5) can be written into

$${}_0^R D_t^\alpha [y(t_{2j}) - y_0] = \beta y(t_{2j}) + g(t_{2j}), \quad j = 1, 2, \dots, m, \quad (6.1)$$

and at point t_{2j+1} equation (3.5) can be written into

$${}_0^R D_t^\alpha [y(t_{2j+1}) - y_0] = \beta y(t_{2j+1}) + g(t_{2j+1}), \quad j = 1, 2, \dots, m - 1. \quad (6.2)$$

Firstly, let us consider the discretization of (6.1). Note that

$$\begin{aligned} {}_0^R D_t^\alpha y(t_{2j}) &= \frac{1}{\Gamma(-\alpha)} \int_0^{t_{2j}} (t_{2j} - \tau)^{-1-\alpha} y(\tau) d\tau \\ &= \frac{t_{2j}^{-\alpha}}{\Gamma(-\alpha)} \int_0^1 \omega^{-1-\alpha} y(t_{2j} - t_{2j}\omega) d\omega. \end{aligned}$$

For every $2j$, we replace the integral by a piecewise quadratic interpolation polynomial with the equispaced nodes $0, \frac{1}{2j}, \frac{2}{2j}, \dots, \frac{2j}{2j}$. We then have, for smooth function $g(\omega)$,

$$\oint_0^1 \omega^{-1-\alpha} g(\omega) d\omega = \oint_0^1 \omega^{-1-\alpha} g_2(\omega) d\omega + R_{2j}(g),$$

where $g_2(\omega)$ is the piecewise quadratic interpolation polynomial of $g(\omega)$ with the equispaced nodes $0, 1/2j, 2/2j, \dots, 2j/2j$ and $R_{2j}(g)$ is the remainder term.

Lemma 6.0.1. Let $0 < \alpha < 1$. We have

$$\oint_0^1 \omega^{-1-\alpha} g(\omega) d\omega = \sum_{k=0}^{2j} \alpha_{k,2j} g\left(\frac{k}{2j}\right) + R_{2j}(g),$$

where

$$(-\alpha)(1-\alpha)(2-\alpha)(2j)^{-\alpha} \alpha_{l,2j} = \begin{cases} 2^{-\alpha}(\alpha+2), & l=0, \\ (-\alpha)2^{2-\alpha}, & l=1, \\ (-\alpha)(-2^{-\alpha}\alpha) + \frac{1}{2}U_0(2), & l=2, \\ -U_1(k), & l=2k-1, \quad k=2,3,\dots,j, \\ \frac{1}{2}(U_2(k) + U_0(k+1)), & l=2k, \quad k=2,3,\dots,j-1, \\ \frac{1}{2}(U_2(k)), & l=2j, \end{cases}$$

and

$$\begin{aligned} U_0(k) &= (2k-1)(2k) \left((2k)^{-\alpha} - (2(k-1))^{-\alpha} \right) (1-\alpha)(-\alpha+2) \\ &\quad - \left((2k-1) + 2k \right) \left((2k)^{-\alpha+1} - (2(k-1))^{-\alpha+1} \right) (-\alpha)(-\alpha+2) \\ &\quad + \left((2k)^{-\alpha+2} - (2(k-1))^{-\alpha+2} \right) (-\alpha)(-\alpha+1), \end{aligned}$$

$$\begin{aligned} U_1(k) &= (2k-2)(2k) \left((2k)^{-\alpha} - (2k-2)^{-\alpha} \right) (1-\alpha)(-\alpha+2) \\ &\quad - \left((2k-2) + 2k \right) \left((2k)^{-\alpha+1} - (2k-2)^{-\alpha+1} \right) (-\alpha)(-\alpha+2) \\ &\quad + \left((2k)^{-\alpha+2} - (2k-2)^{-\alpha+2} \right) (-\alpha)(-\alpha+1), \end{aligned}$$

and

$$\begin{aligned} U_2(k) &= (2k-2)(2k-1) \left((2k)^{-\alpha} - (2k-2)^{-\alpha} \right) (1-\alpha)(-\alpha+2) \\ &\quad - \left((2k-2) + (2k-1) \right) \left((2k)^{-\alpha+1} - (2k-2)^{-\alpha+1} \right) (-\alpha)(-\alpha+2) \\ &\quad + \left((2k)^{-\alpha+2} - (2k-1)^{-\alpha+2} \right) (-\alpha)(-\alpha+1), \end{aligned}$$

Next we consider the discretization of (6.2) at $t_{2j+1} = \frac{2j+1}{2m}$, $j = 1, 2, \dots, m-1$. We have

$$\begin{aligned} {}_0^R D_t^\alpha y(t_{2j+1}) &= \frac{1}{\Gamma(-\alpha)} \oint_0^{t_{2j+1}} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau) d\tau \\ &= \frac{1}{\Gamma(-\alpha)} \oint_0^{t_1} (t_{2j+1} - \tau)^{-1-\alpha} y(\tau) d\tau \\ &\quad + \frac{t_{2j+1}^{-\alpha}}{\Gamma(-\alpha)} \oint_{t_1}^{t_{2j+1}} \omega^{-1-\alpha} y(t_{2j+1} - t_{2j+1}\omega) d\omega \end{aligned}$$

For every $2j+1$, $j = 1, 2, \dots, m-1$, we replace the integral by a piecewise quadratic interpolation polynomial with the equispaced nodes $0, \frac{1}{2j+1}, \frac{2}{2j+1}, \dots, \frac{2j}{2j+1}$. We then have, for smooth function $g(\omega)$,

$$\oint_0^{\frac{2j}{2j+1}} \omega^{-1-\alpha} g(\omega) d\omega = \oint_0^{\frac{2j}{2j+1}} \omega^{-1-\alpha} g_2(\omega) d\omega + R_{2j+1}(g),$$

where $g_2(\omega)$ is the piecewise quadratic interpolation polynomial of $g(\omega)$ with the equispaced nodes $0, \frac{1}{2j+1}, \frac{2}{2j+1}, \dots, \frac{2j}{2j+1}$ and $R_{2j+1}(g)$ is the remainder term.

Lemma 6.0.2. Let $0 < \alpha < 1$. We have

$$\oint_0^{\frac{2j}{2j+1}} \omega^{-1-\alpha} g(\omega) d\omega = \sum_{k=0}^{2j} \alpha_{k,2j+1} g\left(\frac{k}{2j}\right) + R_{2j+1}(g),$$

where $\alpha_{k,2j+1} = \alpha_{k,2j}$, $k = 1, 2, \dots, 2j$ and $\alpha_{k,2j}$ are given by Lemma 6.0.2.

Our future works will focus mainly on the stability regions of Diethelm's method by using quadratic interpolation polynomial. We will also consider the finite difference method for fractional partial differential equation and discuss the stability, convergence, error estimates for such numerical method.

Chapter 7

Conclusion

The purpose of the study was set out to explore the concept of stability regions of well known numerical methods for FDEs. The discussion has sought to understand whether numerical methods for FDEs can result in the same or different stability regions especially when a small sufficient step size is chosen. The study sought to answer these questions:

1. Do the stability regions of these numerical methods for solving FDEs differ because of their difference in weights?
2. Diethelm's method has a higher rate of convergence than the other numerical methods. Therefore, does this factor has any effect in determining their stability regions?

The actual findings for this project are specified in Chapters 3 and 5 respectively and were specified within the respective sections in each chapters. Here, we will synthesize the findings to answer the two study research questions.

1. Do the stability regions of these numerical methods for solving FDEs differ because of their difference in weights and the rate of convergence?
 - The experiments we presented in Fig.(5.1)- Fig.(5.4) speaks a great volume and have demonstrated that the stability of each numerical methods for FDEs differs because of the difference in weights and convergences. Though clear observation shows that Fig.(5.2) and Fig.(5.3) have a slight difference.
 - We also observe that Diethelm's method and Günward are A -stable. But Lubich's method are not A -stable.

- The stability region of Diethelm's method by using linear interpolation is larger than the Diethelm's method by using quadratic interpolation polynomial.

The study has offered an evaluative perspective on an important aspect of numerical methods. As a direct consequence of this methodology, we will not forget to mention that the study encountered a number of limitations, which need to be considered.

Among all the concepts of numerical methods, stability appears to be the brain box among all. The benefit of stability region of the numerical methods is that it determines which numerical methods may be suitable in application of sciences and engineering.

Chapter 8

Appendix

8.1 Figure 5.1 MATLAB programme

```
% Figure 5.1 in the dissertation

% Check the stability region of numerical methods for fractional
% differential equation by using lucus method
%
%
% Produce Figure 1 in the following paper based on Proposition 3.2:
% Roberto Garrappa, On linear stability of predictor-corrector algorithms
% for fractional differential equations, International Journal of Computer
% Mathematics, 87(2010), 2281-2290
%
%
% The stability region is:
%  $S_{\text{RT}} = \{ z \in \mathbb{C} \mid 1 - z (\alpha(\xi) - \alpha_{\{0\}}) - z^2 \alpha_{\{0\}} \xi$ 
%  $b(\xi) \neq 0 : |\xi| \leq 1 \}$ 
% The lucus method:
% Step 1. Let  $\xi = e^{i \theta}$ ,  $\theta \in [0, 2\pi]$ 
% Step 2. Find  $z$  such that
%  $1 - z (\alpha(\xi) - \alpha_{\{0\}}) - z^2 \alpha_{\{0\}} \xi b(\xi) = 0$ 
```

```

% by using MATLAB build in function 'roots'
% Step 3. plot all roots by using
%
%           plot(bdy, '*')
%

clear

N=2000;
n=0:1:N;
n=n';

bt=0.7;
alph=1/gamma(bt+2)*(n.^(bt+1) -2*(n+1).^(bt+1) +(n+2).^(bt +1));
alph=[1/gamma(bt+2);alph];
alph=alph(1:end-1);

a=1/gamma(bt+2)*(n.^(bt+1)-(((n+1).^(bt)).*(n+1-bt-1)));
%a=n.^(bt+1)-(((n+1).^(bt)).*(n+1-bt-1))/gamma(bt +2);
b=1/gamma(bt+1)*((n+1).^(bt)-n.^(bt));
%b=(n+1).^(bt)-n.^(bt)/gamma(bt+1);
M=2000;
h=2*pi/M;
theta=0:h:2*pi; theta=theta';
exp_theta = exp(i*n*theta');
b_xi=b'*exp_theta;
xi_b_xi=exp(i*theta').*b_xi;
alph_xi=alph'*exp(i*n*theta');
c=ones(size(alph_xi));

coeff_poly=[alph(1)*xi_b_xi; alph_xi-alph(1);-c];

```

```

%plot
bdy=[];
for i=1:length(theta)
    bdy=[bdy;roots(coeff_poly(:, i))];
end
plot(bdy, '*')
title('Stability region of Garrappa method')

```

8.2 Figure 5.2 MATLAB programme

```

%Figure 5.2 in the dissertation
%Check the stability region of the numerical method
%for fractional differential equations
%
%
%Consider
%  $D^{\alpha} y(t) = \lambda * y(t)$ 
%  $y(0) = 0$ 
%
%
%By using Diethelm's method, we have, with  $z = h * \lambda$ 
%  $(w_{0,n} - z) y_n + w_{1,n} y_{n-1} + w_{2,n} y_{n-2}$ 
%  $+ \dots + w_{n-1,n} y_1 + (w_{n,n} - n^{-\alpha} / \Gamma(1-\alpha)) y_0 = 0.$ 
%
%
% Let  $y_n = x_i^n$ . Then we get
%  $(w_{0,n} - z) x_i^n + w_{1,n} x_i^{n-1} + w_{2,n} x_i^{n-2}$ 
%  $+ \dots + x_i^{n-1} y_1 + (x_i^{n,n} - n^{-\alpha} / \Gamma(1-\alpha)) y_0 = 0.$ 
%
%

```

```

% The stability region is
%S=
% \{ z \in C: z= g(xi), |xi|=1 \}
%where
% g(z) = (1/xi^{n}) * (w_{0,n} xi^n + w_{1,n} xi^{(n-1)} +
% ... + w_{n-1,n} xi + (w_{n,n} - n^{-al})/gamma(1-al)).
%
%
% Let xi= 0:0.001:2*pi
% We can plot all z
%Then we get the stability region of the numerical method.

clear

n=1000;
al=0.7;

theta=0:0.005:2*pi;
xi=exp(i*theta);

z=0;

w_d=[];

for k=0:n
z=z+ w_alpha(k,n,al)*xi.^(n-k);

w_d=[w_d;w_alpha(k,n,al)];

end

z=z-n^{-al}/gamma(1-al);

```

```

z=z./(xi.^n);

x=real(z);
y=imag(z);

figure
plot(x,y,'b*')
%plot(w_d,'b*')
title('Stability region of Diethelm method')

%coefficients of w_{kj}
function [ y ] = w_alpha(k,j,al)
if k==0
    y=1/gamma(2-al);
else if k==j
    y=(-(al-1)*k^(-al) +(k-1)^(1-al)-k^(1-al))/gamma(2-al);
else
    y=(-2*k^(1-al) +(k-1)^(1-al) + (k+1)^(1-al))/gamma(2-al);
end
end
end

```

8.3 Figure 5.3 MATLAB programme

```

% Figure 5.3 in the dissertation
% Check the stability region of the numerical method
% for fractional differential equations
%
%
%Consider
%  $D^{\{al\}} y (t) = \lambda * y(t)$ 

```

```

% y(0) =0
%
%
%By using Diethelm's method, we have, with  $z = h * \lambda$ 
%  $(w_{\{0,n\}} - z) y_{\{n\}} + w_{\{1,n\}} y_{\{n-1\}} + w_{\{2, n\}} y_{\{n-2\}}$ 
%  $+ \dots + w_{\{n-1, n\}} y_{\{1\}} + (w_{\{n,n\}} - n^{-\alpha}/\gamma(1-\alpha)) y_{\{0\}} = 0.$ 
%
%
% Let  $y_{\{n\}} = x_i^{\{n\}}$ . Then we get
%  $(w_{\{0,n\}} - z) x_i^{\{n\}} + w_{\{1,n\}} x_i^{\{n-1\}} + w_{\{2, n\}} x_i^{\{n-2\}}$ 
%  $+ \dots + x_i^{\{n-1, n\}} y_{\{1\}} + (x_i^{\{n,n\}} - n^{-\alpha}/\gamma(1-\alpha)) y_{\{0\}} = 0.$ 
%
%
% The stability region is
%S=
%  $\{ z \in \mathbb{C} : z = g(x_i), |x_i|=1 \}$ 
%where
%  $g(z) = (1/x_i^{\{n\}}) * (w_{\{0,n\}} x_i^{\{n\}} + w_{\{1,n\}} x_i^{\{n-1\}} +$ 
%  $\dots + w_{\{n-1,n\}} x_i + (w_{\{n,n\}} - n^{-\alpha}/\gamma(1-\alpha)).$ 
%
%
% Let  $x_i = 0:0.001:2*\pi$ 
% We can plot all  $z$ 
%Then we get the stability region of the numerical method.

clear

n=160;
al=0.7;

theta=0:0.005:2*pi;
```

```
xi=exp(i*theta);

z1=0;

w_g=[];
for k=0:n

z1=z1+ w_grunwald(k,n,al)*xi.^(n-k);

w_g=[w_g;w_grunwald(k,n,al)];
end

z1=z1-n^(-al)/gamma(1-al);

z1=z1./(xi.^n);

x1=real(z1);
y1=imag(z1);

figure

plot(x1,y1,'b*')
%plot(w_g,'go')
title('Stability region of Grunwald method')

%coefficients of w_{kj}
function [ y ] = w_grunwald(k,j,al)

    if k==0
        y=1;
    else
```

```

    A=a1-([1:k]-1);
    y= (-1)^k * prod(A)/factorial(k);
end

```

8.4 Figure 5.4 MATLAB programme

```

% MATLAB program for Figure 5.4 in the dissertation.
%Check the stability region of the numerical method
%for fractional differential equations given by Lubich's method by using
%difference equation
% The program is only for p=2. But we can solve all the backward
% difference formula (BDFp) for order p=1,2,3,4,5,6.
%
%Consider
%  $D^{\{a1\}} y(t) = \lambda * y(t)$ 
%  $y(0) = 1$ 
%
%
%
% Let  $z = \lambda * h^{\{a1\}}$ . Then
%
%  $y_{\{m\}} = y_{\{0\}} = z \sum_{\{j=0\}}^{\{m\}} w_{\{m-j\}}^{\{(a1)\}} y_{\{j\}} + z \sum_{\{j=0\}}^{\{p\}}$ 
%  $w_{\{m, j\}}^{\{(a1)\}} y_{\{j\}}$ .
%
% Let  $y_{\{m\}} = \xi^{\{m\}}$ .
%
% The stability region is
%S=
%  $\{ z \in C: z = g(\xi), |\xi|=1 \}$ 
%where

```



```
% g(xi) = (xi^{m} - 1) / ( \sum_{j=0}^{m} w_{m-j}^{al} xi^{j} + \sum_{j=0}^{2}
% w_{m, j}^{al} xi^{j})
%
% Let xi= exp(i*theta), where theta=0:0.001:2*pi,
% We can plot all z
% Then we get the stability region of the numerical method.

clear

n=1000;
al=0.7;

theta=0:0.0001:2*pi;
xi=exp(i*theta);

cz=0; % convolution part

    cw=c_w(n+1,al);
for k=0:n
    cz=cz+ cw(n+1-k)*xi.^k;
end

sz=0; %starting part

sw=s_w(n,al);
for k=0:2
    sz=sz+sw(n,k+1)*xi.^k;
end

z=(xi.^n - 1) ./ (cz+sz);
```

```

x=real(z);
y=imag(z);

figure
plot(x,y,'*')
title('Stability regions of Lubich method')

%coefficients of  $w_{\{kj\}}$ 
function [cw] = c_w(N,al)

    p=2; % order of BDF
% Calculate aggregated Taylor polynomial coefficients
un=[3/2;-2;1/2]; % p=2;
%Calculate convolution weights for each interval node

    cw(1)=1/(un(1)^al); % Interval node 0
% Interval nodes 1 to p
    for k=1:p
        cw(k+1)=0;
        for j=0:(k-1)
            cw(k+1)=cw(k+1) +(-al*(k-j)-j)*cw(j+1)*un(k-j+1);
        end %j
        cw(k+1)=cw(k+1)/(k*(un(1)));
    end %k

%Interval nodes p+1 to n
    for k=(p+1):N
        cw(k+1)=0;
        for j=(k-p):(k-1)
            cw(k+1) =cw(k+1) + ( -al*(k-j)-j)*cw(j+1)*un(k-j+1);
        end
    end

```

```
        end    %j
        cw(k+1)=cw(k+1)/(k*un(1));
    end    %k

%coefficients of starting weights;
% Let p=2
% sw=[w_{10}, w_{11}, w_{12};
%     w_{20}, w_{21}, w_{22};
%     w_{30}, w_{31}, w_{32};
%     ... ...
%     w_{N0}, w_{N1}, w_{N2}]

function [sw] = s_w(N,alpha)

    p=2; % order of BDF
    alpha=alpha;

    i=1;
    if p>1
        j=0;
        k=0;
        gam=0;
        for r=1:p-1
            while gam<r
                g(i)=k+j*alpha;
                gam=k+(j+1)*alpha;
                j=j+1;
                i=i+1;
            end    %gam <r
            k=k+1;
```

```

        j=0;
    end %r
end %if p<>1
g(i)=p-1;

```

```

un=[3/2;-2;1/2];      % p=2;

```

```

%Calculate convolution weights for each interval node

```

```

cw(1)=1/(un(1)^alpha); % Interval node 0

```

```

% Interval nodes 1 to p

```

```

for k=1:p

```

```

    cw(k+1)=0;

```

```

    for j=0:(k-1)

```

```

        cw(k+1)=cw(k+1) +(-alpha*(k-j)-j)*cw(j+1)*un(k-j+1);

```

```

    end %j

```

```

    cw(k+1)=cw(k+1)/(k*(un(1)));

```

```

end %k

```

```

%Interval nodes p+1 to N

```

```

for k=(p+1):N

```

```

    cw(k+1)=0;

```

```

    for j=(k-p):(k-1)

```

```

        cw(k+1) =cw(k+1) + ( -alpha*(k-j)-j)*cw(j+1)*un(k-j+1);

```

```

    end %j

```

```

    cw(k+1)=cw(k+1)/(k*un(1));

```

```

end    %k

%Calculate starting weights for each interval
%Number of starting weights (s+1) at each interval node
s=p; % Choose ANY p+1 gamma values from the possible big set \mathcal{A}
%Starting weights for all interval nodes
sw=zeros(N, s+1);
%Starting weights for current interval nodes
w=zeros(1, s+1);
%Calculate (s+1) starting weights for each interval
%Calculate (s+1) terms for the convolution weights (for starting
%weights) and store in array

% Calculate (s+1) gamma terms for the interval node

%Set up the calculation arrays for each interval

rA=zeros(s+1,1);
RR=zeros(s+1,1);
%Calculate starting weights in turn for N interval nodes

for k=1:N
    %Calculate (s+1) starting weights for each interval node
    rB=zeros(s+1,1);
    for ja=1:(s+1)
        for j=1:k % original
            rB(ja, 1)= rB(ja,1) + cw(k-j+1)*j^(g(ja));%Calc convolution weight te
        end % j
        for jB=1:(s+1)
            lB(ja, jB)=(jB-1)^(g(ja)); %set up starting weight gamma power terms
        end % jB
    end
end

```

```
%set up interval gamma terms
rA(ja,1) = gamma(1+g(ja))*k^(g(ja) + alpha)/gamma(1+g(ja)+alpha);
RR(ja, 1) = rA (ja,1) - rB(ja, 1); % calculates RHS of equation
end %ja
w=lB\RR; %calculate starting weights for the current interval node
%Save starting weights for current interval node
for j=1:(s+1)
    sw(k, j) = w(j);
end %j
end %k
```

Bibliography

- [1] P. Alfeld, *Predictor-Corrector methods*, JWB 127, 801-581-6842 (2010), 1 - 5.
- [2] B. Bonilla, M. Rivero, L.Rodriguez, J.J. Trujillo, *Fractional differential equations as alternative models to non-linear differential equations*, Appl. Math. Compt., 187 (2007), 79 - 88.
- [3] CH. Lubich, *A stability analysis of convolution quadratures for Abel-Volterra integral equations*, IMA J. Numer. Anal. 6 (1986), 87 - 101.
- [4] CH. Lubich, *Fractional linear multi-step methods for Abel-Volterra integral equations of the second kind*, American Math. Society, Math. Comp. Vol. 45, No. 172 (1985), 463 - 469.
- [5] CH. Lubich, *Discretized fractional calculus*, SIAM Journal Mathematics Anal. Vol. 17 (1986), 704 - 719.
- [6] D. Elliot, *An asymptotic analysis of two algorithms for certain Hadamard finite-part integrals*, IMA J. Numerical Anal., 13 (1993) 445- 462.
- [7] E. Capeles de Oliveira, A. L. Soubhia, R.F. Camargo, Jayne Vaz Jr. *On a general Mittag-Leffler Theorem*
- [8] F. Liu, P. Zhuang, V. Anh, I. Turner, K. Burrage, *Stability and convergence of the difference methods for the space-time fractional advection-diffusion equation*, Appl. Math. and Comp., 191 (2007) 12- 20.
- [9] H.H. Sun, A.A. Abdelwahab, B. Onaral, *Linear approximation of transfer function with a pole of fractional order*, IEEE Trans. Automat. Control AC-29 (5) (1984) 441- 444.

-
- [10] H.J. Haubold, A.M. Mathai, R.K. Saxena, *Mittag-Leffler functions and their applications*, J. Math. Compt., 2009, pp. 1 - 51.
- [11] I. Podlubny, *Fractional differential equations, an introduction of fractional derivatives, fractional differential equations, to method of their solution and some of their applications*. Academic press (1999).
- [12] J.T. Edwards, N.J. Ford and A.C. Simpson, *The numerical solution of linear multi-term fractional differential equations: systems of equations*, Elsevier Science B.V., Journal of Comp. and Appl. Math. 148 (2002) 401 - 418.
- [13] J.A. Connolly, *The numerical solution of fractional and distributed order differential equations*, PhD thesis, University of Liverpool (University of Chester), Dec-2004.
- [14] J. Landy, *Fractional differential equations and numerical methods*, MSc. dissertation, University of Chester, 2009.
- [15] K. Diethelm, *An algorithm for the numerical solution of differential equation of fractional order*, Electron . Trans. Numer. Anal. 5 (1997) 1 - 6.
- [16] K. Diethelm, *The analysis of fractional differential equations*, Springer- Verlag Berlin Heidelberg, 2010.
- [17] K. Diethelm, N.J. Ford, *Analysis of fractional differential equations*, J.Math. Anal. Appl. 265 (2) (2002) 229 -248.
- [18] K. Diethelm, N.J. Ford and A.D. Freed, *Detailed error analysis for a fractional Adams method*, 1 -18.
- [19] K. Diethelm and G. Walz, *Numerical solution of fractional differential equations by extrapolation*, J. Numer. Algorithms 16 (1997) 231 - 253.
- [20] K. Diethelm, *Generalized compound quadrature formulae for finite part integral*, IMA J. Numer. Anal., 17 (1997) 479- 493.
- [21] K. Diethelm, N. J. Ford, *Numerical analysis for distributed-order differential equations*, Jour. Comp. Appl. Math., 225 (2009) 96- 104.

-
- [22] K. Diethelm, N. J. Ford, *Multi-order fractional differential equations and their numerical solution*, Appl. Math. of Comp., 154 (2004) 621- 640.
- [23] K. Diethelm, N. J. Ford, A.D. Freed *A predictor-corrector approach for the numerical solution of fractional differential equations*, Nonlinear Dynam., 29 (2002), 3-22.
- [24] K. Diethelm, A.D. Freed, *The FracPECE subroutine for numerical solution of differential equations of fractional order*, in Forschung und Wissenschaftliches Rechnen 1998, eds. S. Heinzl, and T. Plesser, Gessellschaft fur wissenschaftliches Datenverarbeitung, Gottingen, (1999) 57- 71.
- [25] L. Blank, *Numerical treatment of differential equation of fractional order*, Numerical analysis report 287, Manchester Centre for Comp. Mathematics, 1996.
- [26] L. Galeone and R. Garrappa, *Explicit methods for fractional differential equations and their stability properties*, Journal Comp. and Appl. Math. 228 (2009), 548 - 560.
- [27] M. Ichise, Y. Nagayanagi, T.Kojima, *An analog simulation of non-integer order transfer functions for analysis of electrode processes*, J. Electronical. Chem. Inetr-facial Electrochem. 33 (1971) 253- 263.
- [28] M. M. Meerschaert, C. Tadjeran, *Finite difference aproximation for fractional advection-dispersion flow equations*, Jour. Comp. Appl. Math. 172 (2004) 65- 77.
- [29] M. Weilbeer, *Efficient numerical methods for fractional differential equations and their analytical background*, Dr. rer. nat, Technischen Universitat Braunschweig.
- [30] P. Kumar and O.P. Agrawal, *An approximate method for numerical method of fractional differential equation*, Signal Proc., 86 (2006) 2602- 2610.
- [31] R. Garrappa, *On linear stability of predictor-corrector algorithm for fractional differential equations*, Journal Comp. and Appl. Math. 87:10 (2010), 2281 - 2290.
- [32] R. Gorenflo, F. Mainardi, *Fractional calculus: Integral and differential equations of fractional order*, Springer Verlag, Wien and New York 1997, pp. 223 - 276.

-
- [33] R. Gorenflo, *Fractional calculus: Some numerical methods*, Springer Verlag, Wien and New York 1997, pp. 277- 290.
- [34] R.J. LeVeque, *Finite difference methods for differential equations*, University of Washington, 2005.
- [35] R.L. Bagley, R.A. Calico, *Fractional order state equations for the control of viscoelastic structures*, J. Guid. Control Dynam. 14 (2) (1991) 304- 311.