A robust A-stable collocation method for a class of stiff initial value problems

Amos Otieno Okech (amos@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Prof. Kailash C. Patidar
University of the Western Cape, South Africa

22 May 2014

Submitted in partial fulfillment of a structured masters degree at AIMS South Africa

Abstract

Stiff initial value problems for ordinary differential equations (ODEs) were well studied by a family of adaptive Runge-Kutta (RK) methods, such as, Implicit-Explicit (IMEX) methods, Implicit-Explicit Predictor Corrector (IMEX-PC) methods, Singly Diagonally Implicit Runge-Kutta (SDIRK) methods, etc. In this project, we discuss another class of these RK-methods which is based on collocation approximation. We discuss necessary theoretical details of this method. Then we discuss detailed implementation aspects along with the numerical simulations for typical stiff ODEs and/or their systems.

Declaration

I, the undersigned, hereby declare that the work contained in this research project is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Amos Otieno Okech, 22 May, 2014

Contents

List of Tables

List of Figures

1. Introduction

Mathematics is a field that has been integrated with other disciplines to explain the natural phenomena that occurs around us. We find that most of the time, modelling of certain events is needed to achieve a desired outcome such as prediction or understanding of the physical world. Most models that are developed always take the form of differential equations and this requires analytical solutions so as to get the necessary information from them.

Ordinary differential equations can either be classified as autonomous or non-autonomous. The non-autonomous form is written as

$$y'(x) = f(x, y(x)),$$

where x is the independent variable and $y(x)$ is the solution to the differential equation. Note that $y(x)$ can be a vector-valued function, that is, a function from $\mathbb{R} \to \mathbb{R}^m$, with m being the dimension of the differential equation.

The autonomous form is expressed as

$$y'(x) = f(y(x)),$$

where the derivative $y'(x)$ does not depend directly on x.

With the addition of initial condition(s) $y_0 = y(x_0)$ to the system of ordinary differential equations we get the initial value problem

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \tag{1.0.1}$$

for the non-autonomous form given above and for the autonomous form shown below

$$y'(x) = f(y(x)), \quad y(x_0) = y_0.$$

A lot of attention has been paid to the analytical solutions of ODEs over the last few centuries since the work of Newton and Leibniz in the 1670's. However, not all of these ODEs could be solved by the available analytical methods and hence researchers found a need for an accurate numerical approximation to such problems.

Several techniques for finding the numerical solution of ordinary differential equations have been developed over the years. There are well established methods such as the Runge-Kutta methods and their variants that form the basis for effective approximations. However, there are problems that cannot be approximated effectively with these well known classical methods. These numerical solutions are identified from their behaviour when approximated by the normal numerical methods. Even though the exact solutions of such problems exists and is differentiable, their numerical solutions may be extremely unstable, we refer to this problems as *stiff*. This has led to a new classification of ordinary differential equations as either *stiff* or *non-stiff*.

The term *stiff* was first used in the early 1950s by chemists, Curtiss (1952), when dealing with problems in chemical kinetics. They defined a problem to be stiff if the implicit Euler method performs better than the explicit Euler method when finding its numerical solution. They formulated a numerical method, the Backward Differentiation Formula (BDF), for the integration of such problems. This method has

withstood the test of time and it is currently used by some of the popular software packages in numerical analysis.

A lot of attention has been given to finding an acceptable numerical method for integrating stiff problems. In pursuit of the numerical solution of such problems, adoption of implicit Runge-Kutta methods have been the standard method due to their excellent convergence and stability properties as compared to their explicit counterparts. Motivated by this fact, we will discuss a special class of implicit Runge-Kutta methods called Runge-Kutta collocation (RKC) methods that produce excellent results when solving such problems.

In our attempt to explore the collocation methods, we will demonstrate the derivation of such methods and their implementation by the use of an example based on the Chebyshev-Gauss-Lobatto points which are the zeros of a certain Chebyshev polynomial of second kind. The choice of this method has been motivated by their popularity in numerical analysis in solving stiff problems and also because of the excellent convergence properties that they possess. We intend to discuss the order and stability of such methods in detail.

The rest of the project is organized as follows

In Chapter 2, we will discuss the necessary theoretical details of the stiff differential equations and give an example of one such problem. We demonstrate its solution using the explicit Euler and implicit Euler methods.

Chapter 3 is dedicated mainly to the Runge-Kutta methods in which we intend to give a brief introduction to the topic as well as present some Runge-Kutta methods that have been used to solve stiff problems.

Chapter 4 is devoted to the Runge-Kutta collocation method. In this chapter, we also mention some of the standard collocation methods that have been popularly used. To complete this chapter, we discuss the stability analysis which form a very important ingredient in the evaluation of numerical solution to the stiff problems.

In Chapter 5, we pick an example of the collocation methods and demonstrate the general technique of using the Runge-Kutta methods in solving differential equations. This will include construction of the Butcher tableau's and studying the stability properties of the method.

Finally, Chapter 6 is dedicated to demonstrate the effectiveness of the method as we intend to carry out experiments that will compare our approximation against the exact solution of well known stiff problems.

Some concluding remarks are presented in Chapter 7.

2. Stiff initial value problems and their systems

Many applications in such important areas as chemical reactions, control systems, and electronic networks, usually involve initial value problems whose solutions possess fast and slow components. Such systems are difficult to solve because different processes in the system behave on different time-scales.

We consider such an example from Spijker (1996) given by

$$U'(t) = -10^6 \left[U(t) - \frac{1}{3} \right], \quad 0 \leq t \leq T = 1, \quad U(0) = u_0$$

whose solution is given by

$$U(t) = \left(u_0 - \frac{1}{3} \right) \exp(-10^6 t) + \frac{1}{3}.$$

From the above example, the first term (*transient component* of $U(t)$) on the right hand side of the equality decays faster while the second component (*nontransient component* of $U(t)$) decays slowly. After a short duration, usually referred to as the *transient phase*, the first component will die out and the solution will therefore take the form $U(t) \approx \frac{1}{3}$. For this type of differential equations, we may take 'large' step sizes for good accuracy as opposed to the classical situation of 'small' step sizes. But at times we may be forced to use small step sizes if the *stability region* is limited because of the fast time scale in the differential equation.

There have been numerous attempts by many authors in the literature to provide a comprehensive definition of what constitutes a stiff problem but as it stand now, there is no universal agreement as to what should be the standard definition. We therefore provide a brief summary of what different authors term as a stiff problem.

- McDonough (2007) refers to a problem as stiff if some components of the exact solution decay much faster than the others.
- According to Iserles (1996), a problem is stiff if step-size of the solver is determined by the stability rather than accuracy.
- A problem is referred to as stiff if it cannot be solved by explicit methods, or they work only extremely slowly.
- A linear system is stiff if its eigenvalues have negative real parts and the ratio of the magnitudes of the real parts of the largest and smallest eigenvalues, referred to as *stiffness ratio*, is large. In other words, some researchers say that, we refer to a problem of the form $\mathbf{u}' = \mathbf{f}(\mathbf{u}, t)$ as stiff if the eigenvalues of the Jacobian of f differ greatly in magnitude.

Most differential equations encountered in real life are non-linear and there are methods devised to reduce them to linear forms, which are easy to deal with. To illustrate this, let us consider the autonomous non-linear ordinary differential equation given by

$$y'(x) = f(y(x)), \tag{2.0.1}$$

where $y \in \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^m$.

Linearising the above equation at a point y_0 results into the following linear system

$$y'(x) = f(y_0) + \frac{\delta f_i}{\delta y_j}(y_0)(y - y_0); \quad i, j = 1, 2, \ldots, m,$$

where $\frac{\delta f_i}{\delta y_j}(y_0)$ is the evaluated Jacobian matrix at the point y_0. This is now a reduced linear form of (2.0.1).

Methods for evaluating the numerical solution of ODEs usually involve finding $\{y_n\}$ to the solution of (2.0.1) with initial condition y_0 on a set of given time interval $\{t_n\}$ where $t_0 < t_1 < \cdots < t_N = T$.

Most methods for solving ODEs calculate the next approximations of y_n explicitly, that is, by using past values at each step. These methods can not be relied on when solving stiff differential equations as they prove to be unstable. An example of such a method is the Euler explicit method. Below is an example of a stiff problem obtained from Hairer and Wanner.

$$y' = -50(y - \cos(x)), \tag{2.0.2}$$

with initial values $y(0) = 0.15$ and its exact solution mimics the function given by $\cos(x)$.

Below is a plot showing the comparison of the solution to the stiff differential equation by the method of explicit and implicit Euler method.

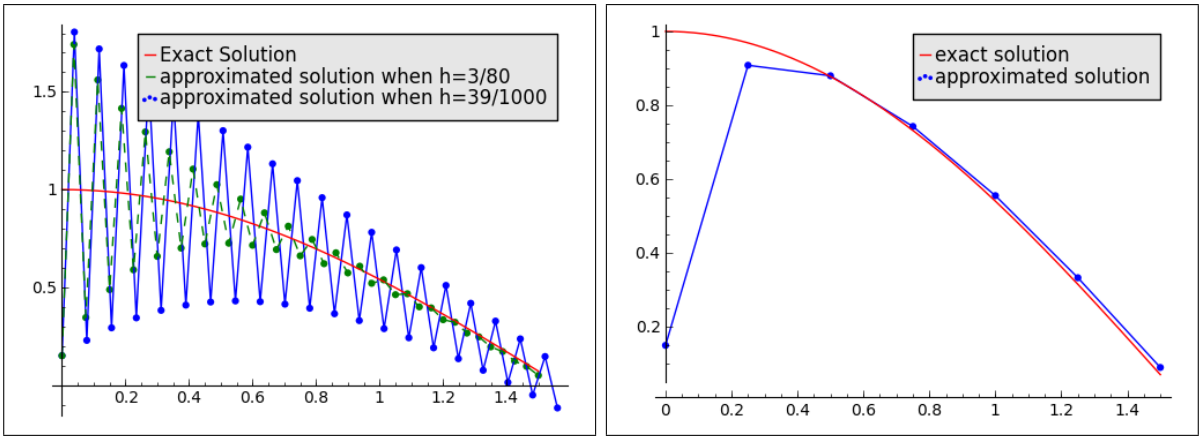

Figure 2.1: Exact and numerical solutions obtained via explicit and implicit Euler method for (2.0.2).

For the plot on the left, we have chosen the step-size $h = \frac{39}{1000}$ that ensures that the step-size for the explicit Euler method is far below the stability boundary, otherwise, a bigger step size larger than $\frac{1}{25}$ causes the numerical solution to be unstable and produces very inaccurate numerical approximations which diverge away from the exact solution as the iterations increase. Therefore, we see that explicit methods cannot be used to find the solution of this type of problem as the region of stability of these methods will require small step sizes even when the problem being solved is smooth.

The diagram on the right shows the numerical solution of the same problem being handled by an implicit Euler method of step-size 0.25 over the same interval with less steps. It takes only two steps before the implicit Euler starts tracing the exact solution.

Before we proceed, we should note that when finding the numerical solution of stiff ODEs, we opt to choose step-sizes according to the dominant eigenvalue of the solution. Most methods for solving

ODEs restrict the step-sizes according to the eigenvalues for stability purposes. Some of the factors that might be considered when choosing an appropriate step-size apart from stability reasons are accuracy and smoothness.

3. Runge-Kutta family and their use in solving initial value problems

Runge-Kutta family: The general set-up

In pursuit of a numerical solution of the initial value problem (1.0.1), many formulations have been proposed, the earliest of these methods being the simple Euler method given by

$$y_{n+1} = y_n + hf(y_n), \tag{3.0.1}$$

which approximates the solution at $t_n + h$ with h being a time-step.

However, the Euler method is unsymmetrical as it uses the derivative only at the beginning of the interval to find the approximate solution after time h, which means that the error in each step is only one power of h.

Euler's method of approximation is usually not recommended for practical purposes because

- Compared to other more advanced methods, they tend to be less accurate with the same step-size.
- They are not stable.

Some of the early methods that were devised to tackle this problem are the mid-point rule and the trapezoidal rule which are advancements of the Euler method. They give better accuracy and are more stable than the explicit Euler method. These methods are presented below.

$$y_{n+1} = y_n + hf(y_n + \frac{1}{2}hf(y_n)),$$
$$y_{n+1} = y_n + \frac{1}{2}hf(y_n) + \frac{1}{2}hf(y_n + hf(y_n)).$$

We can find various ways of approximating $f(t, y)$ such that they agree to first order but have different coefficients of high order error terms. By finding the right combinations of these we can eliminate the error terms order by order. This is basically the rationale behind the Runge-Kutta methods.

Note that, a Runge-Kutta method according to (Butcher, 1964) is a method of obtaining an approximate solution at $t_{n+1} = t_n + h$ for the system

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}(t)), \quad \mathbf{y}(t_0) = \mathbf{y}_0, \tag{3.0.2}$$

given that $\mathbf{y}$ is a vector of n elements and $\mathbf{f}(\mathbf{y})$ is a function of these elements.

Throughout this project, we take the standard assumption that $\mathbf{f}$ satisfies the Cauchy-Lipschitz condition with respect to $\mathbf{y}$ and is continuous with respect to t, so that a unique solution of the ordinary differential equation exists in a given interval, say $[t_n, t_{n+1}]$.

We can rewrite the differential equation (3.0.2) in integral form between t_n and t_{n+1} as

$$\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{f}(\tau, \mathbf{y}(\tau))\mathrm{d}\tau. \tag{3.0.3}$$

The approximation of the integral $\int_{t_n}^{t_{n+1}} \mathbf{f}(\tau, \mathbf{y}(\tau))\mathrm{d}\tau$ can be found using a suitable quadrature formula. If we choose the left rectangle rule to approximate the integral then we will have

$$\mathbf{y}(t_{n+1}) \approx \mathbf{y}(t_n) + h\mathbf{f}(t_n, \mathbf{y}(t_n)),$$

which gives us the explicit Euler formula that we discussed earlier at the beginning of this chapter. If we use the right rectangle rule then we get

$$\mathbf{y}(t_{n+1}) \approx \mathbf{y}(t_n) + h\mathbf{f}(t_{n+1}, \mathbf{y}(t_{n+1})),$$

which gives us the implicit Euler scheme that we used in the previous chapter to obtain the numerical solution to the stiff problem.

If we now look for a rule that combines the explicit and implicit Euler schemes we get

$$\mathbf{y}(t_{n+1}) \approx \mathbf{y}(t_n) + \alpha\mathbf{f}(t_n, \mathbf{y}(t_n)) + \theta\mathbf{f}(t_{n+1}, \mathbf{y}(t_{n+1})).$$

For the above rule to be of at least order 1, we will have $\alpha + \theta = h$ and this give rise to the theta method.

We therefore provide an algorithm for the construction of Runge-Kutta methods. Consider v knots c_j that belong in the interval $[0, 1]$ which are distinct and arranged in an ascending order.

To find the numerical approximation to the integral on the right hand side of (3.0.3), we may use well known quadrature rules such as the Newton-Cote's formula or the Gaussian quadrature. To use these formulas, the integral on the right hand side is transformed into the interval $[0, 1]$.

Let $h_n = t_{n+1} - t_n$ be the step-size and

$$x = \frac{t - t_n}{t_{n+1} - t_n} = \frac{t - t_n}{h_n}. \tag{3.0.4}$$

This therefore implies that $t = t_n + xh_n$ and so $\mathrm{dt} = h_n\mathrm{dx}$. Substitute this into the integral equation (3.0.3), to get

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \int_0^1 \mathbf{f}(t_n + h_n x, \mathbf{y}(t_n + h_n x))\mathrm{dx}. \tag{3.0.5}$$

If we insert the quadrature rule given by

$$\int_0^1 \mathbf{f}(t)\mathrm{dt} \approx \sum_{j=1}^{v} b_j \mathbf{f}(c_j), \tag{3.0.6}$$

then we get

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \sum_{j=1}^{v} b_j \mathbf{f}(t_n + c_j h_n, \mathbf{y}(t_n + c_j h_n)). \tag{3.0.7}$$

However, since the values of $\dot{\mathbf{y}}(t_n + c_j h_n) = (t_n + c_j h_n, \mathbf{y}(t_n + c_j h_n))$ are unknown when $c_j \neq 0$, we obtain the main theorem of differential calculus

$$\mathbf{y}(t_n + c_j h_n) = \mathbf{y}(t_n) + h_n \int_0^{c_j} \dot{\mathbf{y}}(t_n + h_n\gamma)\mathrm{d}\gamma.$$

If we use the quadrature rule again for the integral on the right with the nodes c_j, we get

$$\begin{aligned}\mathbf{y}(t_n + c_j h_n) &\approx \mathbf{y}(t_n) + h_n \sum_{j=1}^{v} a_{ij}\dot{\mathbf{y}}(t_n + h_n c_j),\\ &= \mathbf{y}(t_n) + h_n \sum_{j=1}^{v} a_{ij}\mathbf{f}(t_n + h_n c_j, \mathbf{y}(t_n + h_n c_j)).\end{aligned}$$

This leads us to the following summary of the $s - stage$ Runge-Kutta method which we can write as

$$Y_{n,i} = \mathbf{y}_n + h_n \sum_{j=1}^{v} a_{ij}\mathbf{f}(t_n + c_j h_n, Y_{n,j}), \quad 1 \leq i \leq s, \tag{3.0.8}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h_n \sum_{j=1}^{v} b_j\mathbf{f}(t_n + c_j h_n, Y_{n,j}), \tag{3.0.9}$$

where the given coefficients a_{ij} ,b_j ,c_j, $(i, j = 1, 2, \ldots, s)$ are numerical constants.

The coefficients of the above $s-stage$ Runge-Kutta method is represented in an array called the Butcher Tableau shown below

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \ldots & a_{1s} \\ c_2 & a_{21} & a_{22} & \ldots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \ldots & a_{ss} \\ \hline & b_1 & b_2 & \ldots & b_s \end{array} \tag{3.0.10}$$

where

$$c_i = \sum_{j=1}^{s} a_{ij}.$$

As examples, we present the Butcher tableau of the explicit Euler, implicit Euler and theta methods, respectively, as follows:

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}, \quad \begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}, \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1-\theta & \theta \\ \hline & 1-\theta & \theta \end{array}.$$

We refer to a Runge-Kutta method as semi-explicit if $a_{ij} = 0$ for $i < j$. If in addition $a_{ij} = 0$ for $i = j$, then we refer to that process as an explicit Runge-Kutta method. Otherwise, we refer to such a process an implicit Runge-Kutta process.

Definition 3.1. A method is said to be of order k if given past values $y(t_{n-i})$, the computed solution for the steps following satisfy

$$e_{n+i} = \mathcal{O}(h_{n+i}^{k+1}).$$

We present the Butcher tableau of the classical Runge-Kutta method of order 4 which is explicit in the table below

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}.$$

Use of Runge-Kutta methods in solving Stiff Differential Equations

Runge-Kutta methods can be used to solve stiff differential equations. However, we need implicit Runge-Kutta methods because of their excellent stability properties and high accuracy. Note that the implicit Euler method and the trapezoidal method that we mentioned in the previous section are implicit Runge-Kutta methods and are also BDF methods. However, higher-order BDF are not Runge-Kutta methods because they require y_{n-1} to compute y_{n+1}.

The implicit Runge-Kutta methods are costly and difficult to solve compared to Explicit ones, but due to their efficiency in terms of stability and accuracy, they are suitable for solving stiff differential equations.

We will consider some of the implicit Runge-Kutta methods that have been widely used to solve stiff differential equations. We have the following methods written down due to their popularity among numerical analysts when it come to solving stiff problems.

- Implicit Explicit (IMEX) methods.
- Singly Diagonally Implicit Runge-Kutta (SDIRK) methods.
- Implicit-Explicit Predictor-Corrector (IMEX-PC) methods.

Singly-Diagonal-Implicit Runge-Kutta (SDIRK) methods.

These methods are semi explicit methods with elements in the diagonal of the Butcher Tableau being constants, that is, $a_{ij} = \lambda$ for $i = j$. This property makes the process of computing the coefficients a_{ij} easier since the s equations can be solved in successive stages.

It is popular because it is less costly and easy to implement.

These methods are *stiffly accurate* and are L-*stable*.

Note that, a method is stiffly accurate if $b_j = a_{sj}$, $j = 1, 2, \ldots, s$.

SDIRK methods are efficient as they possess the required stability properties of implicit Runge-Kutta methods.

However, the main disadvantage of these methods is their low order accuracy especially when applied to stiff ODEs.

More details of this method can be found in Ferracina and Spijker (2008).

Implicit-Explicit (IMEX) Runge-Kutta method.

In this method, we assume the following system of ODEs

$$y'(t) = F(y(t)) + G(y(t)),$$

with F non-stiff, or mildly stiff, and G a stiff term.

The numerical solution to such problems involve solving the non-stiff term using explicit methods and the stiff parts using implicit methods. We therefore get the following equations

$$y_{n,i} = y_n + \sum_{j=1}^{i-1} \hat{\mathsf{a}}_{ij} \Delta t F(y_{n+1,j}) + \sum_{j=1}^{i} a_{ij} \Delta t G(y_{n+1,j}),$$
$$y_{n+1} = y_n + \sum_{j=1}^{s} \hat{\mathsf{b}}_j \Delta t F(y_{n+1,j}) + \sum_{j=1}^{s} b_j \Delta t G(y_{n+1,j}).$$

The stiff and non-stiff components are solved separately and then combined later.

For IMEX schemes, the following compatibility conditions need to be satisfied:

Solving the stiff part

- The implicit scheme being deployed should be stable for stiff systems and should possess good damping properties.

Solving the non-stiff part

- The stability region of the explicit scheme should be the largest possible.
- The ratio $\frac{|y_{n+1}|}{|y_n|}$ obtained as a result of applying the explicit scheme should be less than 1.

Although these methods are lower order accurate, they are very popular since they are easy to implement. It also gives users the freedom to choose different methods to solve the stiff and non-stiff parts of the differential equation.

In this case, we tend to have two Butcher tableau, one for the explicit scheme and the other for the implicit scheme. For the explicit case, $a_{i,j} = 0$ for $j \geq i$ whereas for the implicit case $a_{i,j} = 0$ for $j > i$. The implicit tableau is *stiffly accurate*. Derivation of these methods can be found in Ascher et al. (1997). These methods have undesirable time-step restrictions when applied to convection-diffusion problems.

There are specialised classes of implicit Runge-Kutta methods that give us desirable properties when applied to stiff differential equations and they will be the subject of our discussion in the next chapter.

4. Runge-Kutta collocation methods

Before we dive into collocation methods, it is necessary to review the topics on polynomial interpolation and numerical quadrature.

Polynomial interpolation

If we have $\mathbb{P}_v$ as the space of real polynomials of degree $\leq v$, such that we have a set of v distinct interpolation points $c_1 < \cdots < c_v$, $c_i \in \mathbb{R}$ and a set of their corresponding data given by $g_1, \ldots, g_v$, then the interpolating polynomial is the unique polynomial $P(x) \in \mathbb{P}_v$ satisfying $P(c_i) = g_i$, $i = 1, \ldots, v$.

One popular interpolating polynomial that we will use in this project is the Lagrange interpolating polynomial (Ascher et al., 1995) given by

$$\ell_j(x) = \prod_{k \neq j}^{v} \frac{x - c_k}{c_j - c_k}. \tag{4.0.1}$$

The basis for $\mathbb{P}_v$ is formed by these interpolating polynomials and they assume the following equations

$$P(x) = \sum_{j=1}^{v} g_j \, \ell_j(x). \tag{4.0.2}$$

Numerical quadrature

Given a smooth function $g \in \mathbb{R}$, powerful numerical methods have been developed to approximate the integral of g on the interval $[0, 1]$. Finding the approximate integral involves integrating the interpolating polynomial of order $v - 1$ from v quadrature points $0 \leq c_1 < \cdots < c_v \leq 1$ where g are the values at the quadrature points $g_j = g(c_j)$, $j = 1, \ldots, v$.

If we define the weights

$$b_j = \int_0^1 \ell_j(\tau) \mathrm{d}\tau, \tag{4.0.3}$$

then we get the quadrature formula as

$$\int_0^1 g(x) \approx \int_0^1 P(x) = \sum_{j=1}^{v} b_j g(c_j). \tag{4.0.4}$$

Now, to get the integral $\int_{t_0}^{t_0+h} g(t)\mathrm{d}t$, we repeat the process defined by (3.0.4)–(3.0.7).

Collocation methods

We adopt the definition of collocation methods from Brunner (2004).

Definition 4.1. A collocation solution u_h to a functional equation (for example an ordinary differential equation or a Volterra integral equation) on an interval I is an element from some finite-dimensional function space (the collocation space) which satisfies the equation on an appropriate finite subset of points in I (the collocation points) whose cardinality essentially matches the dimension of the collocation space. If initial conditions are present then u_h will usually be required to fulfil these conditions, too.

Therefore to find the solution of the ordinary differential equation (3.0.3) by a collocation method over the interval $[t_0, t_0 + h]$, we choose collocation points $0 \leq c_1 < c_2 < \cdots < c_v \leq 1$ and then find a collocation polynomial of degree $\leq v$, which satisfies

$$\begin{aligned} y(t_0) &= y_0, \\ y'(t_0 + c_i h) &= f(t_0 + c_i h, y(t_0 + c_i h)), \quad i = 1, \ldots, v, \end{aligned}$$

and its numerical solution is defined by $y_1 = y(t_0 + h)$.

The reason why collocation methods are considered a powerful tool in solving a system of ordinary differential equations is that it enables us not only to get a discrete set of solutions but also a continuous approximation to the solution.

The interpolation polynomial $y'(t_0+c_i h)$ can be found by the Lagrange interpolation as shown in (4.0.2). Therefore the interpolation polynomial becomes

$$y'_n(t_n + \tau h_n) = \sum_{j=1}^{v} \ell_j(\tau) Y_{n,j}; \quad Y_{n,j} = y'_n(t_n + c_j h_n), \tag{4.1.1}$$

where ℓ_j are the Lagrange polynomial corresponding to the collocation points c_i given by (4.0.1).

If we integrate equation (4.1.1) we have

$$y_n(t_n + c_i h_n) = y_0 + h \sum_{j=1}^{v} y'_n(t_n + c_j h_n) \int_0^{c_i} \ell_j(\tau) \mathrm{d}\tau,$$

and if $c_i = 1$, we get

$$y_n(t_n + h) = y_0 + h \sum_{j=1}^{v} y'_n(t_n + h) \int_0^{1} \ell_j(\tau) \mathrm{d}\tau = y_0 + h \sum_{j=1}^{v} b_j y'_n(t_n + h).$$

If we take

$$a_{ij} = \int_0^{c_i} \ell_j(\tau) \mathrm{d}\tau, \tag{4.1.2}$$

then we get equation (3.0.8) and equation (3.0.9).

Note that $A = (a_{ij})_{i,j=1}^{v}$, is the elements of the coefficient matrix, $b = (b_1, \ldots, b_v)^T$, is the weight vector and $c = (c_1, \ldots, c_v)^T$ the node vector are provided in the Butcher tableau.

The coefficients of a collocation method must satisfy the following properties

$$C(v) : A \cdot c^{q-1} = \frac{1}{q}c^q, \quad 1 \leq q \leq v, \tag{4.1.3}$$

$$B(v) : \sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, \quad 1 \leq q \leq v. \tag{4.1.4}$$

Most of the collocation methods usually use collocation points obtained from the normalized Legendre Polynomials of degree s given by the following equation

$$P_k^*(x) = \sum_{j=0}^{k} (-1)^{j+k} \binom{k}{j} \binom{k+j}{j} x^j, \quad k = 0, 1, \ldots, s. \tag{4.1.5}$$

This type of collocation methods can be found in Butcher (2004).

A brief summary of some special cases of these collocation methods found in Butcher (2004) is presented below.

- Methods based on Gauss points. The collocation points, c_i, are the roots of (4.1.5). The Gauss formula are exact upto order $2v - 1$.
- Methods based on Radau points. These ones are divided into two.

  The first of these is the Radau I method. In this method, the collocation points are the roots of the polynomial $P_v^*(x) + P_{v-1}^*(x)$. After evaluating the roots, we will get $c_1 = 0$.

  They are exact upto degree $2v - 2$.

  The second case of these methods is the Radau II methods where the collocation points are the roots of the polynomial $P_v^*(x) - P_{v-1}^*(x)$. Evaluating the roots yields $c_v = 1$.

  The Radau II formula is exact upto polyn degree $2v - 2$.

  A specical case of the Radau II methods, Radau IIA, plays a very important role in the integration of stiff differential equations.
- Methods based on Lobatto points. In these methods, the collocation are the roots of the polynomial $P_v^*(x) - P_{v-2}^*(x)$. Evaluation of the roots will result to $c_0 = 0$ and $c_v = 1$.

  They are stiffly accurate with an explicit first stage. Therefore the first stage do not require any computations because they coincide with the last stage of the previous step.

  The Lobatto formula is exact upto degree $2v - 3$ and therefore have order $2v - 2$. A special case of the Lobatto methods, Lobatto IIIA methods have the highest possible order.
- Strongly A-stable First Explicit Runge-Kutta (SAFERK) methods. These methods have been discussed extensively in Gonzalez-Pinto et al. (2014) and Gonzalez-Pinto et al. (2010) and their references therein. The SAFERK methods are a recently introduced method for solving stiff differential equations. They aim at achieving stability of the process.

  The collocation points of the SAFERK methods are obtained from the equation below

  $$\sqrt{2v+1}(P_v^*(x) - P_{v-2}^*(x)) + \alpha\sqrt{2v-1}(P_{v-1}^*(x) - P_{v-3}^*(x)),$$

  where α is a free parameter chosen to achieve A-stability.

These methods are exact upto degree $2v-4$ implying that they have a classical order of $2v-3$.

The v-stage of this method is equivalent to the $(v-1)$-stage RADAU IIA method since they have the same number of implicit steps with the exception being that the SAFERK methods have an initial explicit stage.

They are also stiffly accurate and strongly A-stable. It is easily observed that if we take $\alpha = 0$, then we obtain the Lobatto IIIA method.

The A-matrix of the SAFERK method will take the form

$$A = \left(\begin{array}{c|c} 0 & 0^T \\ \hline A_1 & \widetilde{A} \end{array}\right).$$

where $\widetilde{A}$ is a $(v-1)\times(v-1)$ non singular matrix and A_1 a column vector of dimension 1.

We also have collocation methods that use the Chebyshev-Gauss-Lobatto points which will be part of our discussion in chapter 6.

The reason the collocation methods are usually used in approximating stiff differential equations is because they have excellent stability properties.

A numerical method is stable for a step-size and a formula selection scheme if small perturbations in the initial condition do not cause the numerical approximation to diverge away from the true solution given that the true solution is bounded.

We therefore investigate the stability properties of the collocation methods in the next section.

Stability analysis for Runge-Kutta methods

When considering numerical methods for initial value problems, the concept of absolute stability is very important. To introduce this concept, we consider the test equation

$$y'(t) = \lambda y(t). \tag{4.1.6}$$

This equation might appear to be very simple but it is very important in predicting the stability behaviour of given numerical methods.

If we apply the numerical methods to the test equation (4.1.6) we get a first order difference equation

$$y_{n+1} = R(z)y_n,$$

where $z = \lambda h$.

We refer to the function $R(z)$ as the *stability function* of the method. We therefore define the stability domain of the method below that can be found in Butcher (2004).

Definition 4.2. The stability region, $\{z \in \mathbb{C} : |R(z)| < 1\}$, is the set of points that allow the solution to remain bounded even after many steps of calculation. We concentrate in the negative half plane because the solution is bounded and would therefore allow a good modelling of the differential equation.

Definition 4.3. According to Ferreira (2010), if $|R(z)| \leq 1$ for all z, then we say that the method is absolutely stable.

A-Stability. A numerical method is considered useful if it preserves stability. This numerical method will usually perform better within their stability domains but will give less accurate results outside this domains. Therefore a method that have large domains will always perform better than those that have smaller domains.

Definition 4.4. Hairer et al. (1996) defines a method, whose stability domain satisfies

$$S \supset \mathbb{C}^- = \{z; \mathrm{Re} z \leq 0\},$$

as A-stable.

Definition 4.5. According to Hairer et al. (1996), a method is called L-stable if it is A-stable and if in addition

$$\lim_{z\to\infty} R(z) = 0.$$

To find the stability of Runge-Kutta collocation method, we follow the procedure below.

The s-stage implicit Runge-Kutta methods given by (3.0.8) and (3.0.9) applied to $y' = \lambda y$ will yield

$$\mathbf{Y} = \mathbf{y_0} + h\lambda A\mathbf{Y} \quad \Rightarrow \mathbf{Y} = (1 - zA)^{-1}\mathbf{y_0},$$

where $z = h\lambda$.

Therefore,

$$\mathbf{y_1} = \mathbf{y_0} + zb^T(I - zA)^{-1}\mathbf{y_0} = (1 + zb^T(I - zA)^{-1}e)\mathbf{y_0},$$

where $e = (1, \ldots, 1)^T$.

Therefore in this case, the stability function is given by

$$R(z) = 1 + zb^T(I - zA)^{-1}e.$$

Similarly if we take equations (3.0.8) and (3.0.9) and apply them to $y' = \lambda y$ taking $Y = (Y_{n,i})_{n,i=1}^{v}$, we can rewrite (3.0.8) and (3.0.9) as

$$\begin{aligned} Y - zAY &= \mathbf{y_n}, \\ -zb^TY + y_{n+1} &= y_n, \end{aligned}$$

or in a matrix form as

$$\begin{pmatrix} I - zA & 0 \\ -zb^T & 1 \end{pmatrix} \begin{pmatrix} Y \\ y_{n+1} \end{pmatrix} = y_n \begin{pmatrix} e \\ 1 \end{pmatrix}.$$

Solving for y_{n+1} using Crammer's rule, we get

$$y_{n+1} = \frac{\det \begin{pmatrix} I - zA & e \\ -zb^T & 1 \end{pmatrix}}{\det \begin{pmatrix} I - zA & 0 \\ -zb^T & 1 \end{pmatrix}} y_n.$$

The numerator of the above equation is equivalent to

$$\det \begin{pmatrix} I - zA + zeb^T & 0 \\ -zb^T & 1 \end{pmatrix} = \det(I - z(A - eb^T)).$$

Similarly the denominator is equivalent to

$$\det(I - zA).$$

Therefore,

$$y_{n+1} = \frac{\det(I - z(A - eb^T))}{\det(I - zA)} y_n,$$

which implies that the stability function can also be written as

$$R(z) = \frac{\det(I - z(A - eb^T))}{\det(I - zA)} = \frac{P(z)}{Q(z)}. \tag{4.5.1}$$

Note that the solution of y_{n+1} given $y' = \lambda y$ is $\exp(z)$.

Therefore, if a Runge-Kutta collocation method is of order p, then

$$\exp(z) = R(z) + O(z^{p+1}).$$

We can approximate $\exp(z)$ using Padé's approximation which is written as

$$\exp(z) = \frac{P_{lm}(z)}{Q_{lm}(z)},$$

where

$$P_{lm}(z) = \frac{l!}{(l+m)!} \sum_{i=0}^{l} \frac{(l+m-i)!}{i!(l-i)!} z^i,$$

$$Q_{lm}(z) = \frac{m!}{(l+m)!} \sum_{i=0}^{l} \frac{(l+m-i)!}{i!(m-i)!} z^i.$$

From this formulations and upon calculations, we find that though the Lobatto IIIA have the highest possible order as mentioned earlier, their stability function is such that $|R(\infty)| = 1$, thus they are not good enough for solving stiff differential equations, that is, they are not *L-stable*.

5. A Robust A-stable collocation method for solving a class of stiff initial value problems

In this chapter, we intend to work out an example of a Runge-Kutta collocation method, explain its derivation and theoretical details of the specific method. Finally, we carry out numerical experiments to measure its effectiveness with respect to stability and accuracy when solving stiff differential equations.

Derivation of the family of collocation methods based on Chebyshev-Gauss-Lobatto points

Our main interest is to get a numerical approximation of a stiff differential equation of the form (3.0.2).

We start by taking a fixed step-size, h which means that the value of t_n is equal to $t_0 + nh$.

If we take an assumption that at step t_n we have a numerical approximation $y_n \simeq y(t_n)$ to the problem (3.0.2), then to obtain the numerical approximation y_{n+1} at time t_{n+1}, we shall consider a $(v+1)$-stage Runge-Kutta method defined by the butcher tableau given in (3.0.10) which can be written in the form (3.0.8) and (3.0.9). Note that this collocation method is stiffly accurate.

Evaluating the collocation points and the coefficients in the Butcher Tableau.

As seen in the other collocation methods we discussed in chapter 4, there is always a procedure followed when getting the collocation points and they are associated with names alluding to historical reasons. Similarly, in our case, the collocation points will be obtained from the Chebyshev-Gauss-Lobatto points given by

$$\alpha_j = \cos(\theta_j), \quad \theta_j = \frac{v-j}{v}\pi, \quad j = 0, 1, \ldots, v, \tag{5.0.1}$$

and therefore the collocation points will be given by

$$c_{j+1} = \epsilon_j = \frac{1}{2}(1 + \alpha_j). \tag{5.0.2}$$

To get the coefficients a_{ij}, we let $z(t)$ be a smooth differentiable function and we define the linear operator associated with the equations (3.0.8) and (3.0.9) by

$$L_i(z(t), h) = z(t + \epsilon_{i-1}h) - z(t) - h\sum_{j=1}^{v+1} a_{ij} z'(t + \epsilon_{j-1}h). \tag{5.0.3}$$

If we expand $z(t+\epsilon_{i-1}h)$ and $z'(t+\epsilon_{i-1}h)$ using the Taylor series expansion about t and collecting the powers in h we get

$$z(t + \epsilon_{i-1}h) = z(t) + \epsilon_{i-1}hz'(t) + \frac{(\epsilon_{i-1}h)^2}{2!}z''(t) + \cdots + \frac{(\epsilon_{i-1}h)^n}{n!}z^{(n)}(t) + \ldots,$$
$$z'(t + \epsilon_{i-1}h) = z'(t) + \epsilon_{i-1}hz''(t) + \frac{(\epsilon_{i-1}h)^2}{2!}z^3(t) + \cdots + \frac{(\epsilon_{i-1}h)^{n-1}}{(n-1)!}z^{(n)}(t) + \ldots.$$

Substituting the above equations into (5.0.3) we get

$$L_i(z(t),h)=\sum_{i=1}^{\infty}\left(\frac{\epsilon_{i-1}^{p}}{p!}-\frac{1}{(p-1)!}\sum_{j=1}^{v+1}a_{ij}\epsilon_{i-1}^{p-1}\right)h^p z^{(n)}(t)$$

and it is from this that we get an equation for evaluating a_{ij} given by

$$\sum_{j=1}^{v+1}a_{ij}\epsilon_{i-1}^{p-1}=\frac{\epsilon_{i-1}^{p}}{p},\quad j=1,\ldots,v+1. \tag{5.0.4}$$

From the above algebraic system, we can conclude that the process is a collocation method since it satisfies $C(v)$ given by (4.1.3).

We should take note that the above system have $(v+1)$ uncoupled systems each with $(v+1)$ equations that also have $(v+1)$ number of unknowns. Therefore at the end we will have $(v+1)(v+1)$ coefficients of a_{ij}.

Lets consider one of the coupled system below

$$\begin{aligned}
a_{j1}+a_{j2}+\cdots+a_{j(v+1)}&=\epsilon_{j-1},\\
a_{j1}\epsilon_0+a_{j2}\epsilon_1+\cdots+a_{j(v+1)}\epsilon_v&=\frac{\epsilon_{j-1}^2}{2},\\
&\vdots\\
a_{j1}\epsilon_0^v+a_{j2}\epsilon_1^v+\cdots+a_{j(v+1)}\epsilon_v^v&=\frac{\epsilon_{j-1}^{v+1}}{2},
\end{aligned}$$

for $j=1,\ldots,v+1$.

Theorem 5.1. *For $j=1,\ldots,v+1$, each system of equations given by*

$$A\mathbf{v_j}=\mathbf{B_j}$$

where

$$A=\begin{pmatrix}1&1&\ldots&1\\ \epsilon_0&\epsilon_1&\ldots&\epsilon_v\\ \epsilon_0^2&\epsilon_1^2&\ldots&\epsilon_v^2\\ \vdots&\vdots&\ddots&\vdots\\ \epsilon_0^v&\epsilon_1^v&\ldots&\epsilon_v^v\end{pmatrix},$$

$$\mathbf{v_j}=\begin{pmatrix}a_{j1}\\ a_{j2}\\ \vdots\\ a_{j(v+1)}\end{pmatrix},\quad \mathbf{B_j}=\begin{pmatrix}\epsilon_{j-1}\\ \frac{\epsilon_{j-1}^2}{2}\\ \vdots\\ \frac{\epsilon_{j-1}^{v+1}}{v+1}\end{pmatrix},$$

has a unique solution.

Proof. See Vigo-Aguiar and Ramos (2007). □

So far, we have shown the methodology for evaluating the coefficients c_i's and a_{ij}'s, we now seek to evaluate the coefficients b_j's.

We recall that this method is stiffly accurate and therefore $b_j=a_{v+1}j$.

Examples of the methods

Before we present the examples, we will consider the theorem below without proof.

Theorem 5.2. *Let* $M(t) = \prod_{i=1}^{p}(t - c_i)$ *and suppose that* M *is orthogonal to polynomials of degree* $(r-1)$*, i.e.*

$$\int_0^1 M(t)t^{q-1}\mathrm{dt} = 0, \quad q = 1, \ldots, r;$$

then the Runge-Kutta collocation method obtained from the c_i *has order* $(p + r)$.

From these, we present a corollary that will give us the order of the collocation method which we have described above if v is even.

Corollary 5.3. When v is even, the Runge-Kutta collocation methods we have described in section 2 will have order $(v + 2)$. Proof is presented in Vigo-Aguiar and Ramos (2007).

We now construct the Butcher Tableau for different values of v. We will manually do the simplest one which is when $v = 1$ and afterwards we will present some selected Butcher Tableau for different values of v and evaluate A-stability for the respective cases.

When $v = 1$**.** To get the collocation points, we follow the procedure (5.0.1) and (5.0.2). When $j = 0$, we get

$$\theta_0 = \frac{1}{1}\pi = \pi, \quad \alpha_0 = \cos(\pi) = -1,$$

and therefore

$$c_1 = \epsilon_0 = \frac{1}{2}(1 + (-1)) = 0.$$

For $j = 1$, the result is presented below

$$c_2 = \epsilon_1 = \frac{1}{2}(1 + \cos(0)) = 1.$$

We now use the system given by (5.0.4) to evaluate the values of a_{ij}. In this case we will have 4 a_{ij}'s. To start with, we evaluate a_{11} and a_{12} as follows

$$\begin{aligned} a_{11} + a_{12} &= 0, \\ a_{11} \times 0 + a_{12} \times 1 &= 0, \end{aligned}$$

which when solved simultaneously gives $a_{11} = 0$ and $a_{12} = 0$.

With similar arrangement, we solve for a_{21} and a_{22} which both evaluates to $\frac{1}{2}$. Since the method is stiffly accurate then the values of b_j are also equal to $\frac{1}{2}$. Therefore the Butcher tableau is given below

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}. \tag{5.3.1}$$

We recall that the Butcher Tableau above is equivalent to the one for the trapezoidal rule and the Lobatto IIIA method of order 2.

The stability of the above function which is obtained from (4.5.1) is given by

$$R(z) = \frac{2+z}{2-z}$$

and it is easily seen that on the left half complex plane, $|R(z)| \leq 1$ and therefore the method is A-stable.

When $v = 2$. In this case, the Butcher Tableau is given by

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & \frac{-1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} . \tag{5.3.2}$$

Again this is equivalent to the Lobatto IIIA method. Using the corollary we have defined above, then we can say that this method has order $v + 2 = 4$.

The stability function of this step is given by

$$R(z) = \frac{12 + 6z + z^2}{12 - 6z + z^2}.$$

Therefore, this method is A-stable since on the left half of the complex plane, $|R(z)| \leq 1$.

When $v = 3$. For this case the Butcher Tableau is presented below

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{4} & \frac{59}{576} & \frac{47}{288} & \frac{-7}{288} & \frac{5}{576} \\ \frac{3}{4} & \frac{3}{64} & \frac{15}{32} & \frac{9}{32} & \frac{-3}{64} \\ 1 & \frac{1}{18} & \frac{4}{9} & \frac{4}{9} & \frac{1}{18} \\ \hline & \frac{1}{18} & \frac{4}{9} & \frac{4}{9} & \frac{1}{18} \end{array} . \tag{5.3.3}$$

The stability function of this method is given by

$$R(z) = \frac{384 + 192z + 38z^2 + 3z^3}{384 - 192z + 38z^2 - 3z^3},$$

and so on the left hand side of the complex plane, $|R(z)| \leq 1$ implying that the method is A-stable. Not forgetting that it has order 4.

The Butcher Tableau and the Stability function for the cases when $v = 4$ and 5 are given in Vigo-Aguiar and Ramos (2007).

Therefore we take this opportunity to find the Butcher tableau and stability functions when $v = 6$ and 7.

When $v = 6$. The Butcher Tableau will be given by

$$
\begin{array}{c|ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{2-\sqrt{3}}{4} & \frac{53+9\sqrt{3}}{2520} & \frac{256-47\sqrt{3}}{4032} & \frac{2269-1368\sqrt{3}}{20160} & \frac{164-93\sqrt{3}}{1260} & \frac{2339-1368\sqrt{3}}{20160} & \frac{256-145\sqrt{3}}{4032} & \frac{-17+9\sqrt{3}}{2520} \\
\frac{1}{4} & \frac{1}{280} & \frac{189\sqrt{3}+296}{4032} & \frac{233}{2240} & \frac{-23}{1260} & \frac{23}{2240} & \frac{296-189\sqrt{3}}{4032} & \frac{1}{280} \\
\frac{1}{2} & \frac{53}{2520} & \frac{16+7\sqrt{3}}{252} & \frac{319}{1260} & \frac{41}{315} & \frac{-31}{1260} & \frac{16-7\sqrt{3}}{252} & \frac{-17}{2520} \\
\frac{3}{4} & \frac{3}{280} & \frac{24+21\sqrt{3}}{448} & \frac{489}{2240} & \frac{39}{140} & \frac{279}{2240} & \frac{24-21\sqrt{3}}{448} & \frac{3}{280} \\
\frac{2+\sqrt{3}}{4} & \frac{53-9\sqrt{3}}{2520} & \frac{256+145\sqrt{3}}{4032} & \frac{2269+1368\sqrt{3}}{20160} & \frac{164+93\sqrt{3}}{1260} & \frac{2339+1368\sqrt{3}}{20160} & \frac{256+47\sqrt{3}}{4032} & \frac{-17-9\sqrt{3}}{2520} \\
1 & \frac{1}{70} & \frac{8}{63} & \frac{8}{35} & \frac{82}{315} & \frac{8}{35} & \frac{8}{63} & \frac{1}{70} \\
\hline
 & \frac{1}{70} & \frac{8}{63} & \frac{8}{35} & \frac{82}{315} & \frac{8}{35} & \frac{8}{63} & \frac{1}{70}
\end{array},
$$

and the stability function is given by

$$R(z) = \frac{2580480 + 1290240z + 291840z^2 + 38400z^3 + 3108z^4 + 146z^5 + 3z^6}{2580480 - 1290240z + 291840z^2 - 38400z^3 + 3108z^4 - 146z^5 + 3z^6}.$$

In the left half complex plane, $|R(z)| \leq 1$ and therefore this method is A-stable.

We warn that putting the system of linear equations in a numerical software might give us decimal places instead of fractions as we have above and so we recommend the use of the equation (4.1.2) to evaluate the coefficients a_{ij}'s for this case and other cases where v is large. The method has order 8 as defined by the lemma we mentioned earlier in this section.

When $v = 7$. This is a complex case where the values of a_{ij} will be represented in 4 significant figures. We present this method to demonstrate the complexity of the process as v increases.

$$
\begin{array}{c|ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.04952 & 0.02010 & 0.03198 & -0.003601 & 0.001621 & -0.001007 & 0.0007512 & -0.0006388 & 0.0003032 \\
0.1883 & 0.001961 & 0.1164 & 0.07893 & -0.01331 & 0.007113 & -0.005011 & 0.004160 & -0.001961 \\
0.3887 & 0.01657 & 0.08068 & 0.1986 & 0.1054 & -0.01882 & 0.01081 & -0.008310 & 0.003837 \\
0.6113 & 0.006367 & 0.1034 & 0.1653 & 0.2374 & 0.1132 & -0.02244 & 0.01439 & -0.006367 \\
0.8117 & 0.01217 & 0.09091 & 0.1811 & 0.2115 & 0.2319 & 0.09719 & -0.02130 & 0.008243 \\
0.9505 & 0.009901 & 0.09571 & 0.1754 & 0.2196 & 0.2170 & 0.1797 & 0.06309 & -0.009901 \\
1 & 0.01020 & 0.09507 & 0.1761 & 0.2186 & 0.2186 & 0.1761 & 0.09507 & 0.01020 \\
\hline
 & 0.01020 & 0.09507 & 0.1761 & 0.2186 & 0.2186 & 0.1761 & 0.09507 & 0.01020
\end{array}.
$$

We obtain the following as the stability function

$$\frac{4.238 \times 10^{-8}z^7 + 2.797 \times 10^{-6}z^6 + 8.1380 \times 10^{-5}z^5 + 0.001409z^4 + 0.01581z^3 + 0.0049z^2 + 0.5z + 1}{-4.238 \times 10^{-8}z^7 + 2.797 \times 10^{-6}z^6 - 8.1380 \times 10^{-5}z^5 + 0.001409z^4 - 0.01581z^3 + 0.0049z^2 - 0.5z + 1}.$$

Note that in the above example, we have rounded everything to 4 significant figures. The method above is also A-stable and has order 8.

Characteristics of the methods

The first characteristic we observe is that the stage order of the internal stages of this method is atleast $\mathcal{O}(h^{v+2})$.

We have observed that the first stage of this methods coincides with the initial values while the last stage coincides with the final values. This implies that no further calculation is needed to find the value of y_{n+1}.

Another characteristic that we have stressed throughout the constructions of the Butcher tableau is that the method is A-stable.

In the examples that we have presented in the previous section and those that are in Vigo-Aguiar and Ramos (2007), we notice that the method is not L-stable since

$$\lim_{z\to\infty} R(z) = \pm 1.$$

A careful observation of the collocation points lead us to the following conclusion

$$c_i + c_{v+2-i} = 1, \quad i = 1, \ldots, v+1,$$

and therefore we conclude that the numerical approximation after one step backward will lead us to the initial value, that is, applying the step-size $-h$ to y_{n+1} will lead us to y_n.

The stability function of this method takes the form

$$R_v(z) = \frac{P_v(z)}{P_v(-z)},$$

where $P_v(z)$ is a polynomial of degree v that is obtained by

$$P_v(z) = M^{v+1}(1) + M^v(1)z + \cdots + M'(1)z^v,$$

and

$$M(z) = \prod_{i=0}^{v}(z - \epsilon_i),$$

with ϵ_i is defined by the equation (5.0.2).

Connection between Runge-Kutta Collocation method and linear multi-step methods

A numerical method is said to be an l step method if y_{n+1} depends on $y_{n+1-l}, \ldots, y_n$. Linear multistep methods are described by the relation below

$$\sum_{j=0}^{l} \alpha_j y_{n+j} = h_n \sum_{j=0}^{l} \beta_j F(t_{n+j}, y_{n+j}), n = 0, \ldots, N-l, \tag{5.3.4}$$

where α and β are constants to be determined.

Linear multisteps methods are divided into many classes and we will mention a few that we will encounter in this section.

1. **The Adams method.** In these methods, the values of α on the left hand side of the equal sign of equation (5.3.4) are given by

$$\alpha_0 = \alpha_1 = \cdots = \alpha_{l-2} = 0, \alpha_{l-1} = -1, \alpha_l = 1.$$

   These methods are given by the following expression

$$y_{n+l} = y_{n+l-1} + h_n \sum_{j=0}^{l} \beta_j F(t_{n+j}, y_{n+j}).$$

   If the value of $\beta_l = 0$ then this method is explicit and it has a special name, *Adams-Bashforth methods* otherwise if $\beta_l \neq 0$, we get the *Adams-Moulton methods* which are implicit.

2. **Milne-Simpson Methods.** These methods take the form

$$y_{n+l} = y_{n+l-2} + h_n \sum_{j=0}^{l} \beta_j F(t_{n+j}, y_{n+j}).$$

Computations for the vales of β_j in the multistep problems indicated above can be found in Hairer et al. (1993) and Butcher (2006).

A careful look at our collocation method and the linear multistep methods shows that there is a close link between them.

We start with the case when $v = 1$, as mentioned earlier, the Runge-Kutta collocation method is equivalent to the trapezoidal rule which is a linear multistep method.

Next, we observe the case when $v = 2$, and we can use the Butcher tableau to obtain the following system of equations

$$y_{n+\frac{1}{2}} = y_n + \frac{h}{24}\left(5f_n + 8f_{n+\frac{1}{2}} - f_{n+1}\right), \tag{5.3.5}$$

$$y_{n+1} = y_n + \frac{h}{6}\left(f_n + 4f_{n+\frac{1}{2}} + f_{n+1}\right). \tag{5.3.6}$$

It is easily observed that the first equation corresponds to the two-steps Adams Moulton formula for the equation of $y' = f(x)$ given by

$$y_{n+1} = y_n + \bar{h}\left(\frac{5}{12}f_{n+1} + \frac{8}{12}f_n - \frac{1}{12}f_{n-1}\right),$$

and to get the equality between the two equations then we realize that if we take $x_{n-1} = t_n + h$, $x_n = t_n + \frac{h}{2}$, $x_{n+1} = t_n$ and $\bar{h} = \frac{-h}{2}$.

By setting $x_{n-1} = t_n$, $x_n = t_n + \frac{h}{2}$, $x_{n+1} = t_n + h$ and $\bar{h} = \frac{h}{2}$, then the two-step Milne Simpson formula for the equation $y' = f(x, y)$ given by

$$y_{n+1} = y_{n-1} + \bar{h}\left(\frac{1}{3}f_{n+1} + \frac{4}{3}f_n + \frac{1}{3}f_{n-1}\right),$$

will lead us to the equation (5.3.6).

For the case when $v \geq 3$, then the computations become complicated but the method for solving them is just the same, so we will demonstrate the procedure for solving when $v = 3$ and the rest of the cases can be approached the same way.

From the Butcher tableau corresponding to $v = 3$, we get the following systems

$$\begin{aligned}
y_{n+\frac{1}{4}} &= y_n + \frac{h}{576}\left(59f_n + 94f_{n+\frac{1}{4}} - 14f_{n+\frac{3}{4}} + 5f_{n+1}\right),\\
y_{n+\frac{3}{4}} &= y_n + \frac{3h}{64}\left(f_n + 10f_{n+\frac{1}{4}} + 6f_{n+\frac{3}{4}} - f_{n+1}\right),\\
y_{n+1} &= y_n + \frac{h}{18}\left(f_n + 8f_{n+\frac{1}{4}} + 8f_{n+\frac{3}{4}} + f_{n+1}\right).
\end{aligned}$$

By adjusting the step-size as appropriate, the three equations above lead us to the three-step Adams-Bashforth formula, three-step Milne-Simpson formula and the variable-step Milne-Simpson formula of three steps, respectively, if we take the following adjustments $x_{n-2} = t_n + h$, $x_{n-1} = t_n + \frac{3h}{4}$, $x_n = t_n + \frac{h}{4}$ and $x_{n+1} = t_n$.

Implementation Strategy

Applying this collocation method to large systems of ODEs causes large systems of equations to be solved in each step. If we have m systems of differential equations then the dimensions of the system will be $(v + 1)m \times (v + 1)m$.

We can rewrite (3.0.8)-(3.0.9) as

$$\begin{aligned}
y_{n+c_1} &= y_n\\
y_{n+c_2} &= y_n + h_n \sum_{j=1}^{v} a_{2j} f(t_n + c_j h_n, Y_{n,j})\\
&\vdots\\
y_{n+1} &= y_n + h_n \sum_{j=1}^{v} a_{(v+1)j} f(t_n + c_j h_n, Y_{n,j}).
\end{aligned}$$

We introduce the notations

$$\begin{aligned}
Y_n &= (y_{n+c_1}, \ldots, y_{n+c_{v+1}})^T \in \mathbb{R}^{(v+1)m},\\
F(t_n, Y_n) &= (f(t + c_1 h_n, y_{n+c_1}), \ldots, f(t + c_{v+1} h_n, y_{n+c_{v+1}}))^T \in \mathbb{R}^{(v+1)m},\\
e &= (1, \ldots, 1)^T \in \mathbb{R}^n,
\end{aligned}$$

with n being the dimension of our ODE.

To write the system of ODEs, we use the kronecker product defined below

Definition 5.4. Let $A \in R^{(v+1),m}$ and $B \in \mathbb{R}^{p,q}$, the kronecker product, $A \otimes B$, is defined as the $((v+1)p, mq)$ matrix shown below

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots, & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{(v+1)1}B & \dots & a_{(v+1)m}B \end{pmatrix}.$$

Using this definition in writing down the system of ODEs we get

$$\begin{aligned} e \otimes \mathbf{y}_n &= (\mathbf{y}_n, \dots, \mathbf{y}_n)^T \in \mathbb{R}^{(v+1)m}, \\ \mathbf{A} \otimes I_n &= \begin{pmatrix} 0 & \dots & 0 \\ a_{21}I_n & \dots, & a_{2m}I_n \\ \vdots & \ddots & \vdots \\ a_{(v+1)1}I_n & \dots & a_{(v+1)m}I_n \end{pmatrix}, \\ b^T \otimes I_n &= (b_1 I_n, \dots, b_{v+1} I_n)^T \in \mathbf{R}^{m,m(v+1)} \end{aligned}$$

where $I_n \in \mathbb{R}^{n,n}$ denotes the identity matrix.

We can therefore rewrite the Runge-Kutta collocation method (3.0.8)-(3.0.9) in a more compact form as

$$Y_n = e \otimes \mathbf{y}_n + h_n(A \otimes I_n)\mathbf{F}(t_n, Yn).$$

Therefore, the system shown below is solved to find approximation to the solution.

$$\mathbf{G}(Y_n) = Y_n - e \otimes \mathbf{y}_n - h_n(A \otimes I_m)F(t_n, Y_n).$$

Note that in this project we have used a uniform step-size in our numerical simulations.

6. Numerical experiments and discussion

To test the effectiveness of this method in solving stiff differential equations, we have selected well known problems that have appeared in the literature. We will now put into practice what we have discussed so far.

6.0.1 A prothero-Robinson equation. Prothero and Robinson (1974). The Prothero-Robinson equation is given by the following equation

$$\begin{cases} y'(t) = \lambda(y(t) - g(t)) + g'(t), \quad t \in [0, 10], \\ y(0) = 0, \end{cases} \tag{6.0.1}$$

where $\lambda = -10^6$, $g(t) = \sin(t)$, and its exact solution is given by $y(t) = \sin(t)$.

We compare the plots of the numerical and exact solution in the same graph. For this plot, we have used the variant of the collocation method when $v = 2$. A table containing the error found at each step is also presented to measure the efficiency and consistency of this method when halving the step-sizes.

The plot is presented below

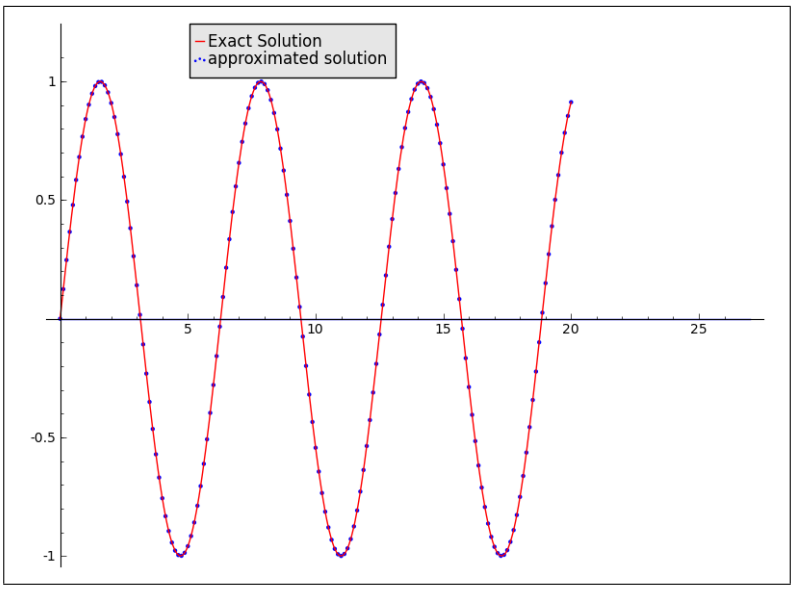

Figure 6.1: Exact and numerical solutions for the Prothero Robinson equation (6.0.1)

We present errors with different step-sizes in Table (6.1).

From the above experiment we observe that our numerical approximation is close to the exact solution even for large step-sizes and therefore the approximated solution mimics the exact solution as can be observed in the plot. Moreover, if small step-sizes are considered, then the numerical solution becomes more accurate as shown in Table (6.1). Therefore, methods with small step-sizes perform better than those with large step-sizes.

From Table (6.1), it can be seen that the order of convergence is 2.

From table (6.1), we note that as the value of t increases, the error remain fairly constant for the different values of h.

Table 6.1: Demonstration of dependence of error on step-sizes

t	$h = 2$	$h = 1$	$h = 0.5$	$h = 0.25$	$h = 0.125$
0	0.000	0.000	0.00	0.00	0.000
2	$1.31E{-7}$	$3.02E{-8}$	$7.42E{-9}$	$1.85E{-9}$	$4.60E{-10}$
4	$1.53E{-7}$	$3.53E{-8}$	$8.67E{-9}$	$2.16E{-9}$	$5.37E{-10}$
6	$3.68E{-9}$	$8.49E{-8}$	$2.07E{-10}$	$5.04E{-11}$	$1.13E{-11}$
8	$1.06E{-7}$	$2.45E{-8}$	$6.00E{-9}$	$1.49E{-9}$	$3.71E{-10}$
10	$1.70E{-7}$	$3.93E{-8}$	$9.64E{-9}$	$2.40E{-9}$	$5.96E{-10}$
12	$1.45E{-8}$	$3.33E{-9}$	$8.15E{-10}$	$2.00E{-10}$	$4.77E{-11}$
14	$7.99E{-8}$	$1.84E{-8}$	$4.52E{-9}$	$1.12E{-9}$	$2.78E{-10}$
16	$1.81E{-7}$	$4.18E{-8}$	$1.03E{-8}$	$2.55E{-9}$	$6.33E{-10}$
18	$3.15E{-8}$	$7.25E{-9}$	$1.78E{-9}$	$4.38E{-10}$	$1.06E{-10}$
20	$5.48E{-8}$	$1.26E{-8}$	$3.10E{-9}$	$7.67E{-10}$	$1.88E{-10}$

6.0.2 A mildly stiff linear system. Vigo-Aguiar and Ramos (2007). In this problem we consider the system of differential equations

$$\begin{cases} y_1'(t) = 998y_1(t) + 1998y_2(t), \\ y_2'(t) = -999y_1(t) - 1999y_2(t), \end{cases} \tag{6.0.2}$$

with initial values $y_1(0) = 1$ and $y_2(0) = 1$. Its exact solution is given by

$$\begin{cases} y_1(t) = 4e^{-t} - 3e^{-1000t}, \\ y_2(t) = -2e^{-t} + 3e^{-1000t}. \end{cases}$$

The stiffness ratio of this problem is $1 : 1000$. We find the numerical solution for the values of t between $[0, 0.04]$ with $h = 0.001$. As done in the previous example, we will present the errors in a table and plot the numerical solution and its exact solution over the same interval. There will be two plots, the first one for $y_1(t)$ and the other one for $y_2(t)$.

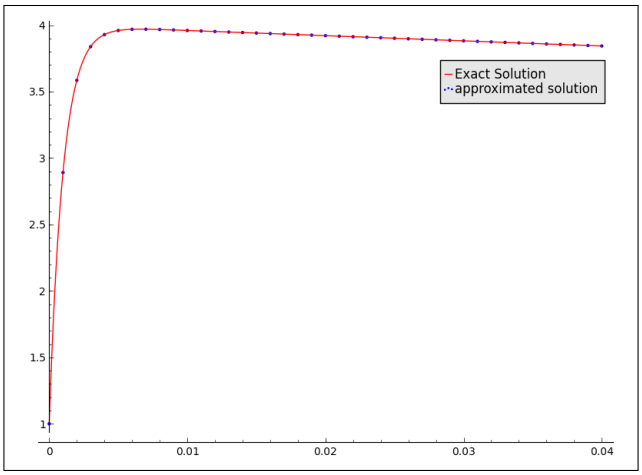

Figure 6.2: Exact and numerical solutions for $y_1(t)$ of the mildly stiff problem (6.0.2)

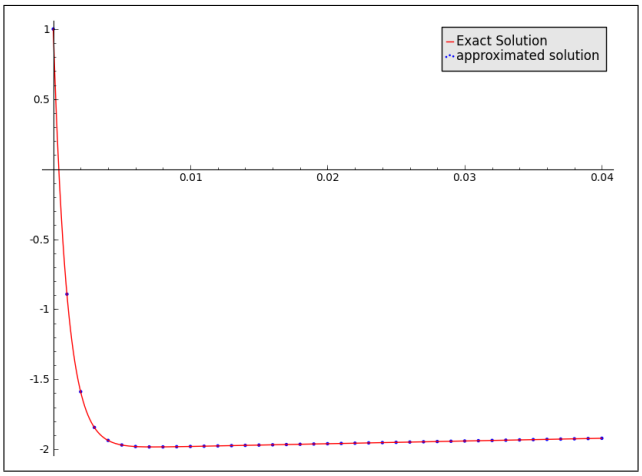

Figure 6.3: Exact and numerical solutions for $y_2(t)$ of the mildly stiff problem (6.0.2)

We present errors obtained when we use variants of an RKC method by using $v = 2$, $v = 4$, $v = 6$ and $v = 8$ and compare the four instances.

Table 6.2: Errors for y_1 using the classical Runge-Kutta method and the collocation method at different steps when $v = 2$, $v = 4$, $v = 6$ and $v = 8$.

t	Classical RK	RKC $(v = 2)$	RKC $(v = 4)$	RKC $(v = 6)$	RKC $(v = 8)$
0	0.00	0.00	0.00	0.00	0.00
0.005	0.78	$1.49E-4$	$1.26E-7$	$3.42E-11$	$4.44E-15$
0.010	0.77	$2.02E-6$	$1.70E-9$	$4.83E-13$	$1.95E-14$
0.015	0.75	$2.05E-8$	$1.72E-11$	$4.31E-14$	$5.33E-14$
0.020	0.73	$1.85E-10$	$1.98E-13$	$5.95E-14$	$8.17E-14$
0.025	0.72	$1.51E-12$	$5.60E-14$	$8.53E-14$	$1.21E-13$
0.030	0.700	$6.26E-14$	$7.64E-14$	$1.04E-13$	$1.41E-13$
0.035	0.68	$9.99E-14$	$9.55E-14$	$1.11E-13$	$1.65E-13$
0.040	0.67	$8.04E-14$	$1.30E-13$	$1.22E-13$	$1.64E-13$

Similarly for y_2, we present the errors in Table (6.3).

Table 6.3: Errors for y_2 using the classical Runge-Kutta method in the first column and the collocation method at different steps when $v = 2$, $v = 4$, $v = 6$ and $v = 8$.

t	Classical RK	RKC $(v = 2)$	RKC $(v = 4)$	RKC $(v = 6)$	RKC $(v = 8)$
0	0.00	0.00	0.00	0.00	0.00
0.005	0.39	$1.5E-4$	$1.26E-7$	$3.42E-11$	$9.33E-15$
0.010	0.38	$2.02E-6$	$6.87E-10$	$4.73E-13$	$5.11E-15$
0.015	0.38	$2.05E-8$	$1.72E-11$	$2.26E-14$	$2.46E-14$
0.020	0.37	$1.85E-10$	$1.75E-13$	$2.84E-14$	$3.55E-14$
0.025	0.35	$5.69E-13$	$2.84E-14$	$4.22E-14$	$5.35E-14$
0.030	0.35	$2.24E-14$	$3.80E-14$	$5.20E-14$	$6.51E-14$
0.035	0.34	$4.82E-14$	$4.60E-14$	$5.60E-14$	$7.93E-14$
0.040	0.33	$3.97E-14$	$6.24E-14$	$6.11E-14$	$8.22E-14$

One thing to note is that the cost of doing the experiment increases as the value of v increases due to the time it takes to generate the matrix A. However, even $v = 2$ gives us pretty good approximation at certain points.

Even though the cost increases, it is worth noting as demonstrated in the table above that the accuracy of the numerical approximation (especially for the first iterations) increases as the value of v increases.

The numerical approximation of the stiff problem with the classical Runge-Kutta method is not as accurate as the collocation method that we have chosen and this may be due to the fact that they are not A-stable.

The plots provided for this problem show that the numerical solution coincides with the exact solution and we can therefore draw the conclusion that the collocation method that we chose is well suited to solving this type of stiff problem.

7. Conclusion

Collocation methods are a special class of the implicit Runge-Kutta methods, but with some benefits that the other methods do not possess. The purpose of this project was to investigate A-stable collocation methods that can be used to solve a class of stiff initial value problems. We chose collocation methods based on Chebyshev-Gauss-Lobatto points. Theses methods have the advantage of being stiffly accurate with the first step being explicit. This ensures that the last internal stage will directly yield the next discrete approximation and the previous discrete approximation becomes the first internal stage of the next step. We carried out stability analysis for these collocation methods and established that they are A-stable. This shows that they can compete with the other well established collocation methods such as the Lobatto IIIA methods. The numerical approximations we have obtained from the two experiments we carried out are plausible. We have seen that the error of the method deteriorates as the step-size becomes smaller and therefore our method is consistent. We also compared our Runge-Kutta collocation approach with the classical Runge-Kutta method and found that the former outperforms the latter. This topic is still wide open for research as still many ideas have not been utilized to make it more satisfactory and complete. In future, we intend to use variable step-size selection to minimize the number of functions evaluated and thereby minimizing the computational cost of this method.

Acknowledgements

Foremost, I would like to thank my mother, Caroline Okech for the continuous encouraging remarks she gave me when I was young upto now. Without forgetting my family members for their support throughout my time at AIMS even when i felt like giving up due to pressure.

I would like also to thank my Supervisor, Prof. Kailash for his constructive criticism and patience with me when doing this project. His style of supervision made me get the best out of this topic.. Not forgetting the two people who have made it possible for us to accomplish whatever we have here in AIMS, that is Jan and Frances for the skills they planted in me which will not be forgotten in many ages to come.

I would also like to thank my tutor Adriaan for his support while doing this essay. His commitment even during the last day of submission was encouraging and heart touching. His contribution of ideas and counter checking of the grammar in the essay were of much help. I would like to appreciate my AIMS brothers and sisters for making this place a complete home especially Achamyelesh, Kujo, Belthasara, Fanuel, Fikre, Leakey, Evans Ochiaga and Rosemary.

I would also like to pay my respect to Professor Barry and Professor Jeff Sanders for their support doing the tough times I went running to them.

Finally I would like to thank one person who has given me wings during this period, that is Corazon Akinyi, what i feel about her can only be summarised by a quotation from Winnie the Pooh *"If you live to be 100, I would like to be 100 minus one day, so I never have to live without you"*.

References

U. M. Ascher, R. M. M. Mattheij, and R. D. Russell. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1995. ISBN 9780898713541.

U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit runge-kutta methods for time-dependent partial differential equations. *Appl. Numer. Math*, 25:151–167, 1997.

H. Brunner. *Collocation Methods for Volterra Integral and Related Functional Differential Equations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004. ISBN 9780521806152.

J. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2004. ISBN 9780470868263.

J. C. Butcher. Implicit runge-kutta processes. *Mathematics of Computation*, 18(85):50–64, 1964.

J. C. Butcher. General linear methods. *Acta Numerica*, 15:157–256, 5 2006. ISSN 1474-0508. doi: 10.1017/S0962492906220014.

J. O. Curtiss, C. F.; Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America*, 38:225–243, 1952.

L. Ferracina and M. Spijker. Strong stability of singly-diagonally-implicit runge–kutta methods. *Applied Numerical Mathematics*, 58(11):1675 – 1686, 2008. URL http://www.sciencedirect.com/science/article/pii/S0168927407001651.

J. A. Ferreira. *Computational Mathematics*. University of Coimbra, 2010. URL http://www.mat.uc.pt/~ferreira/CompMathematics.pdf.

S. Gonzalez-Pinto, D. Hernandez-Abreu, and J. I. Montijano. An efficient family of strong a-stable runge-kutta collocation methods for stiff systems and dae's. part i: stability and order results., j. comput. appl. math. 234:1105–1116, 2010.

S. Gonzalez-Pinto, D. Hernandez-Abreu, and S. B. Strongly a-stable first stage explicit collocation methods with stepsize control for stiff and differential-algebraic equations. 259:138–152, 2014.

E. Hairer and G. Wanner. Stiff differential equations solved by radau methods. *Journal of Computational and Applied Mathematics*, 111:93–111.

E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Solving Ordinary Differential Equations. Springer, 1993. ISBN 9783540566700.

E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Lecture Notes in Economic and Mathematical Systems. Springer, 1996. ISBN 9783540604525.

A. Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge texts in applied mathematics. Cambridge University Press, 1996. ISBN 9780521556552.

J. M. McDonough. Lectures in basic computational numerical analysis, 2007. URL http://www.engr.uky.edu/~acfd/egr537-lctrs.pdf.

A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Appl. Numer. Math.*, 28, 1974.

M. Schatzman. *Numerical analysis: a mathematical introduction*. Oxford University Press, 2002.

M. Spijker. Stiffness in numerical initial-value problems. *Journal of Computational and Applied Mathematics*, 72(2):393 – 406, 1996.

J. Vigo-Aguiar and H. Ramos. A family of a-stable runge-kutta collocation methods of higher order for initial-value problems. *IMA Journal of Numerical Analysis*, 1:1–20, 2007.