Numerical Methods for Ordinary Differential Equations

A Survey of Some Standard Methods

Jane Hyo Jin Lee

A thesis submitted in partial fulfilment of the requirements for the degree of

Master of Science in Applied Mathematics,

The University of Auckland, 2004.

Abstract

Numerical methods have made enormous progress alongside the rapid development of computers. For finding solutions of ordinary differential equations, numerical methods are a valuable tool, since finding an analytic solution is often very difficult or impossible.

The earliest numerical method for ordinary differential equations is the famous Euler method which evaluates the driving function once in each step and uses an approximated solution from the previous step to update a solution. The early extensions of this methods are the two well-known and most commonly used methods, Runge-Kutta methods and linear multistep methods. A Runge-Kutta method uses a result given at the end of the previous step while evaluating functions at the one or more off-step points. A linear multistep method uses already approximated solutions and evaluated function values from several previous steps but only evaluates the function once in each step. The unifying framework for these two traditional methods is known as general linear methods which are multistage like Runge-Kutta methods and multivalue like linear multistep methods.

Mainly in this thesis, the important fundamental properties of these numerical methods for ordinary differential equations are investigated. This involves the formulation for these methods, order and stability conditions and other basic concepts that are required to understand the methods. Furthermore, some numerical experiments on general linear methods are presented. This includes comparions between the different methods and investigations on predictors.

Acknowledgements

First of all, I would like to express my deepest gratitude to my supervisor, Professor John Butcher for his enormous help and advice and for providing inspiration which cannot be expressed with words. I would not have accomplished this thesis without his patient care, understanding and encouragement.

I also wish to thank the other members of the numerical analysis group at the University of Auckland for their friendship, continuous support and help on mathematical knowledge that they freely discussed and shared with me. I especially mention Dr. Robert Chan, Nicolette Rattenbury, Shirley Huang, Angela Tsai and Dr. Allison Heard who has spent many hours on checking my work and finally Dr. Helmut Podhaisky in Germany who has shared lots of technical knowledge with me and helped me on computational experiment work during his one year visit to Auckland. I will never forget their care and kindness and I will always be proud and happy to have been a member of this group.

I have been very fortunate to meet and share the office with many fellow graduate students in the mathematics department during my degree. They provided a happy and enjoyable atmosphere and helped me to figure out various MATLAB and LaTeX difficulties. I would particularly like to acknowledge Kate Lee, Dion O'Neale, David Simpson and Priscilla Tse for their companionship.

Finally, I wish to thank my family, my mother and father, my sister Christine Lee and my friend Isaac Park from the bottom of my heart. I am very grateful to them for all the wonderful, never-ending and unlimited support and encouragement.

Contents

Abstract iii

Acknowledgements v

Contents vii

List of Figures xi

List Of Tables xiii

1 Introduction **1**

1.1 Ordinary Differential Equations 1

1.1.1 Existence and Uniqueness of Solutions 3

1.2 Euler Method 4

1.2.1 Order of Convergence 5

1.2.2 Stability and A-Stability 5

1.3 Stiff Problems 7

1.3.1 Examples of Stiff Problem 9

1.4 Framework of Thesis 16

2 Runge-Kutta Methods **19**

2.1 Formulations 19

2.1.1 Explicit Methods 21

2.1.2 Implicit Methods 22

2.2 Order Conditions 23

2.2.1 Trees 23

2.2.2 Elementary Differentials 26

2.2.3 Elementary Weights 28

2.2.4 Conditions for Order 30

2.3 Stability 31

3 Linear Multistep Methods 35

3.1 Formulation 35

3.2 Consistency, Stability and Convergence 36

3.3 Order Conditions 40

3.3.1 Adams-Bashforth Methods 42

3.3.2 Adams-Moulton Methods 43

3.3.3 Backward-Difference Methods 44

3.4 Errors and Accuracy 44

3.4.1 Truncation Error 44

3.4.2 Error Estimation and Correction 46

3.5 Stability 46

4 General Linear Methods 53

4.1 Formulation 53

4.1.1 Examples 55

4.2 Consistency, Stability and Convergence 56

4.3 Order of Accuracy 57

4.4 Order Conditions 58

4.5 Runge-Kutta Stability 59

4.6 Inherent Runge-Kutta Stability 60

5 Numerical Experiments 61

5.1 Comparison between IRKS Methods 61

5.2 Comparison between Predictors 65

5.3 Comparison with `ode15s` and `oode23s` 70

A Test Problems 77

A.1 VDPOL 77

A.2 OREGO 79

A.3 HIRES 81

B IRKS Methods 83

B.1 Order 1 methods 83

B.2 Order 2 methods 84

B.3 Order 3 methods 86

Bibliography 89

List of Figures

1.1 Stability Region for Explicit Euler method 6

1.2 Stability Region for Implicit Euler method 6

1.3 Analytic solution of *Example 1*, stiff linear problem 10

1.4 Analytic solution of *Example 2*, stiff nonlinear problem 11

1.5 Stiff linear example by explicit Euler method 12

1.6 Stiff linear example by implicit Euler method 13

1.7 Stiff nonlinear example by explicit Euler method 14

1.8 Stiff linear example by implicit Euler method 15

1.9 Types of numerical methods 16

2.1 Stability Regions for explicit Runge-Kutta methods 33

3.1 Stability Regions for Adams-Bashforth methods 49

3.2 Stability Regions for Adams-Moulton methods 50

3.3 Stability Regions for BDF methods 51

3.4 A(α)-stability regions for the order 3 and 4 BDF methods 52

4.1 Local truncation error 57

4.2 Global truncation error 58

5.1 Comparison between order 1 IRKS methods 62

5.2 Comparison between order 2 IRKS methods 63

5.3 Comparison between order 3 IRKS methods 64

5.4 Comparison of covergence of predictors on OREGO for order 1 71

5.5 Comparison of covergence of predictors on OREGO for order 2 72

5.6 Comparison of covergence of predictors on OREGO for order 3 73

5.7 Order 3 IRKS methods, `ode15s` and `ode23s` on VDPOL 74

5.8 Order 3 IRKS methods, `ode15s` and `ode23s` on OREGO 75

5.9 Order 3 IRKS methods, `ode15s` and `ode23s` on HIRES 76

A.1 Solution of VDPOL 78

A.2 Solution of OREGO 80

A.3 Solution of HIRES 82

List of Tables

2.1 Notation for trees and various functions on trees 24

2.2 Elementary differentials and elementary weights for trees 29

3.1 Error coefficients of the order 2 Adams-Bashforth method 45

3.2 Error coefficients of the order 2 Adams-Moulton method 46

Chapter 1

Introduction

Mathematical modelling has been used more and more in many areas such as in science, engineering, medicine, economics and social science. Differential equations are one of the important and widely used techniques in mathematical modelling. However, not many differential equations have an analytic solution and even if there is one, usually it is extremely difficult to obtain and it is not very practical. Thus, numerical methods are truly a crucial part of solving differential equations which cannot be neglected. Since the late 18^{th} century, numerical methods for solving differential equations have been developed continuously by many mathematicians. Later on in the 20^{th} century, this subject made great improvements in the context of modern computers.

In this chapter, an introduction to ordinary differential equations and a review of the earliest numerical approach to solving ordinary differential equations, Euler methods, are covered followed by a brief discussion of stiffness. Also, an outline of the rest of this thesis is included at the end.

1.1 Ordinary Differential Equations

Among the models using differential equations (DE), ordinary differential equations are frequently used to describe various physical problems, for example, motions of the planet in a gravity field like the Kepler problem, the simple pendulum, electrical circuits and chemical kinetics problems.

An ordinary differential equation (ODE) has the form

$$y'(x) = f\left(x, y(x)\right) \tag{1.1}$$

where x is the independent variable which often refers to time in a physical problem and the dependent variable, $y(x)$, is the solution. Moreover, since $y(x)$ could be an N dimensional vector valued function, the domain and range of the differential equation, f and the solution, y are given by

$$\begin{aligned} f &: \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}^N, \\ y &: \mathbb{R} \rightarrow \mathbb{R}^N. \end{aligned}$$

The above equation (1.1) where f is a function of both x and y is called 'non-autonomous'. However, by simply introducing an extra variable which is always exactly equal to x, it can be easily rewritten in an equivalent 'autonomous' form below where f is a function of y only.

$$y'(x) = f\left(y(x)\right). \tag{1.2}$$

Even though many problems are naturally expressed in the non-autonomous form, the autonomous form of differential equation (1.2) is preferred for most of the theoretical investigations. Furthermore, the autonomous form has some advantages in numerical analysis since it gives a greater possibility that a numerical methods can solve the differential equation exactly.

The differential equation by itself is not enough to find a unique solution. Hence, some other additional information is needed. If extra information is given at several values of x i.e. 'boundary conditions' then the differential equation is known as a 'boundary value problem (BVP)'. However, if all components of y are given at a certain value of x i.e. 'initial conditions' then the differential equation is called as an 'initial value problem (IVP)' which is closely and naturally involved with physical modelling.

An initial value problem with the given initial condition $y(x_0) = y_0$ has the structure

$$y'(x) = f\left(x, y(x)\right), \qquad y(x_0) = y_0 \tag{1.3}$$

in non-autonomous form and

$$y'(x) = f\left(y(x)\right), \qquad y(x_0) = y_0 \tag{1.4}$$

in autonomous form.

1.1.1 Existence and Uniqueness of Solutions

From a practical point of view of scientific modelling, it is very important to examine whether there exists a solution to an initial value problem and if it exists, whether it is unique. In addition, how sensitive the solution is to small perturbations to the initial conditions, given a unique solution exists, must be considered. These matters can be explained using a Lipschitz condition defined as follows.

Definition 1.1. Butcher [7] *The function $f : [a, b] \times \mathbb{R}^N \to \mathbb{R}^N$ is said to satisfy a 'Lipschitz condition' if there exists a 'Lipschitz constant' L, such that for all $x \in [a, b]$ and all $y, z \in \mathbb{R}^N$,*

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

This definition allows us to deduce the following theorem which ensures the existence and uniqueness of solutions to an initial value problem.

Theorem 1.2. Butcher [7] *Consider an initial value problem*

$$y'(x) = f\left(x, y(x)\right), \qquad y(a) = y_0$$

where $f : [a, b] \times \mathbb{R}^N \to \mathbb{R}^N$ is continuous in its first variable and satisfies a Lipschitz condition in its second variable. Then there exists a unique solution to this problem.

Furthermore, the existing and unique solution should not be too sensitive to the initial condition. If y and z each satisfy *Theorem 1.2* with the initial conditions $y(a) = y_0$ and $z(a) = z_0$, respectively then

$$\frac{d}{dx}\|y(x)-z(x)\| \le L\|y(x)-z(x)\|.$$

Multiply both sides by $\exp(-Lx)$ then,

$$\begin{aligned}
\exp(-Lx)\frac{d}{dx}\|y(x)-z(x)\| &\le L\exp(-Lx)\|y(x)-z(x)\| \\
\exp(-Lx)\frac{d}{dx}\|y(x)-z(x)\| - L\exp(-Lx)\|y(x)-z(x)\| &\le 0 \\
\frac{d}{dx}\left(\exp(-Lx)\|y(x)-z(x)\|\right) &\le 0.
\end{aligned}$$

This implies that $\exp(-Lx)\|y(x)-z(x)\|$ is non-increasing. Hence,

$$\begin{aligned}
\exp(-Lx)\|y(x)-z(x)\| &\le \exp(-La)\|y(a)-z(a)\| \\
\|y(x)-z(x)\| &\le \exp(L(x-a))\|y_0-z_0\|.
\end{aligned}$$

This limitation on the growth rate of initial perturbations could seem to be hopeless in some senses. However, it can be improved by the one-sided Lipschitz condition defined in the next section.

In this thesis, only an initial value problem in autonomous form which satisfies *Definition 1.1* and *Theorem 1.2* will be considered to make sure the solution exists and unique.

1.2 Euler Method

The famous Euler method was introduced by Euler in 1768. This simplest of all methods became the stepping-stone of numerical methods for solving ordinary differential equations.

Recall the initial value problem,

$$y'(x) = f\left(x, y(x)\right), \qquad y(x_0) = y_0.$$

Let x_N be an end point of a solution where N is the number of steps required to approximate the solution at x_N. Hence, the interval of integration is $[x_0, x_N]$ and the stepsize h is $(x_N - x_0)/N$. The Euler method generates the numerical approximation,

y_n, to the exact solution, $y(x_n)$, at $x_n = x_0 + hn$ $(n = 1, 2, ..., N)$ by the following scheme.

$$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1}), \quad n = 1, 2, ..., N. \tag{1.5}$$

This original formula of Euler which detemines y_n directly from y_{n-1} is an 'explicit' method. However, if y_n itself is involved in the approximation, then it is an 'implicit Euler' method. The implicit form of Euler method can be written as

$$y_n = y_{n-1} + hf(x_n, y_n), \quad n = 1, 2, ..., N. \tag{1.6}$$

1.2.1 Order of Convergence

The order of the Euler method can be determined by comparing the Taylor series expansion of the exact solution with the computed solution. If the exact solution $y(x_n) = y(x_{n-1} + h)$ at x_n, where $x_n = x_{n-1} + h$, is expanded using the Taylor series about x_{n-1}, then

$$y(x_n) = y(x_{n-1}) + hy'(x_{n-1}) + \frac{h^2}{2!}y''(x_{n-1}) + \frac{h^3}{3!}y'''(x_{n-1}) + \cdots . \tag{1.7}$$

The numerical solution y_n by the Euler method is given by

$$y_n = y_{n-1} + hf(x_{n-1}, y_{n-1}). \tag{1.8}$$

Substract equation (1.8) from (1.7) and obtain

$$y(x_n) - y_n = O(h^2). \tag{1.9}$$

Hence, the order of the Euler method is 1.

1.2.2 Stability and A-Stability

For an analysis of the stability of a numerical method, the standard linear test problem $y' = \lambda y$ proposed by Dahlquist [26] is used, where the solution is $y = \exp(\lambda y)$ and λ is a complex parameter. Using the Euler method for this test problem $y' = \lambda y$ gives,

$$\begin{aligned} y_n &= y_{n-1} + hf(x_{n-1}, y_{n-1}) \\ &= y_{n-1} + h\lambda y_{n-1}. \end{aligned}$$

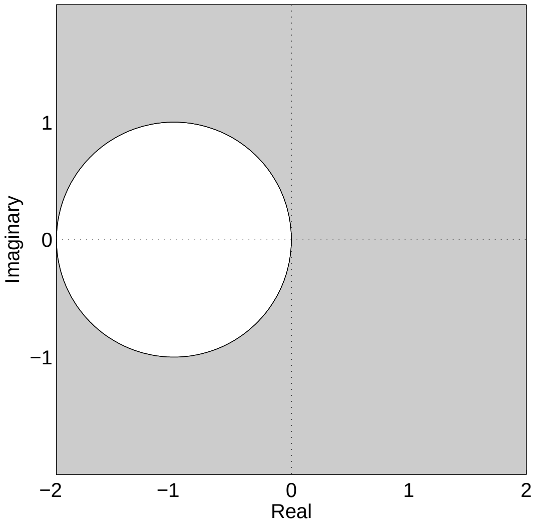


Figure 1.1: Stability Region for Explicit Euler method

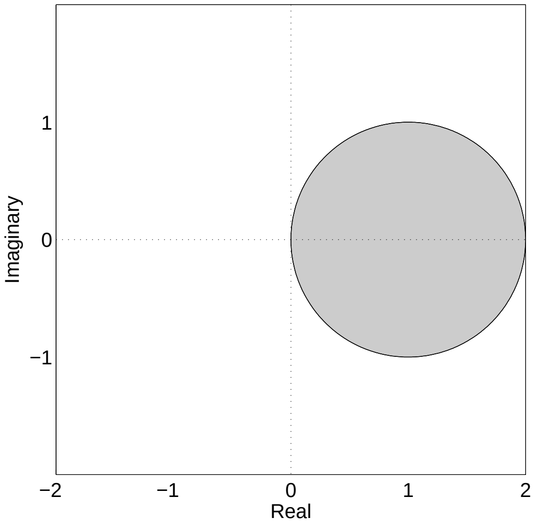


Figure 1.2: Stability Region for Implicit Euler method

Letting $z = h\lambda$ gives,

$$y_n = y_{n-1} + zy_{n-1}.$$

Rearrange to get

$$y_n = (1+z)y_{n-1}.$$

The method is said to be stable if and only if $|1+z| \leq 1$, where $|1+z| = 1$ is the unit disc with center -1 and radius 1 in the complex plane. For the explicit Euler method, the stability region is the bounded region by the unit disc. This is shown as the unshaded part of *Figure 1.1*.

The ratio y_n/y_{n-1} is known as 'stability function', $R(z)$, hence, the stability function $R(z)$ of the explicit Euler method is given by,

$$R(z) = \frac{y_n}{y_{n-1}} = (1+z). \tag{1.10}$$

Simliarly, the stability region and function of implicit Euler method can be deduced.

$$\begin{aligned} y_n &= y_{n-1} + hf(x_n, y_n) \\ \Rightarrow \qquad y_n &= y_{n-1} + h\lambda y_n \\ \Rightarrow \qquad y_n &= y_{n-1} + zy_n \\ \Rightarrow \quad (1-z)y_n &= y_{n-1}. \end{aligned}$$

Hence, the stability function $R(z)$ of the implicit Euler method is given by,

$$R(z) = \frac{y_n}{y_{n-1}} = \frac{1}{1-z}. \tag{1.11}$$

For the stability region, $|R(z)| \leq 1$ is required, that is $|1-z| \geq 1$, where $|1-z| = 1$ is the unit disc with center 1 and radius 1 in the complex plane. As clearly shown in *Figure 1.2*, the stability region, the unshaded part, for implicit method is unbounded and includes whole left half plane.

A numerical method for which the stability region includes all of the left half plan like the implicit Euler method is called an 'A-stable' method. An A-stable method ensures that the numerical approximation is bounded in cases when the exact solution is bounded.

Definition 1.3. *A numerical method is A-stable if and only if*

$$|R(z)| \leq 1$$

for all $z \in \mathbb{C}$ such that $Re(z) \leq 0$.

It is commonly known that A-stability is a desired property for a numerical method to be able to solve the special type of problem called a 'stiff problem' which is discussed in further detail in the next section.

1.3 Stiff Problems

Even if there exists the numerical solution to a differential equation, certain types of differential equations are difficult to solve, in fact, they need certain types of numerical methods. This phenomenon knowns as 'stiffness' was first recognised by Curtiss and Hirschfelder [24] in 1952. Stiffness occurs when some components of the solution decay much more rapidly than others. These problems have highly stable exact solutions but have highly unstable numerical solutions. There are several ways of characterising 'stiffness' and one way of understanding is looking at the Lipschitz constant. Stiff problems typically have a large Lipschitz constant; however, many of them have a more moderate size one-sided Lipschitz constant.

Definition 1.4. Butcher [7] *The function $f : [a, b] \times \mathbb{R}^N \to \mathbb{R}^N$ is said to satisfy a 'one-sided Lipschitz condition' if there exists a 'one-sided Lipschitz constant' l, such that for all $x \in [a, b]$ and all $y, z \in \mathbb{R}^N$,*

$$\langle f(x, y) - f(x, z), y - z\rangle \leq l\|y - z\|^2$$

where the norm is defined by $\|y\|^2 = \langle y, y\rangle$ assuming that there exists an inner-product on $\mathbb{R}^N$.

Therefore, the Lipschitz constant could be large while the one-sided Lipschitz constant could be small, or even negative. This theorem leads us to deduce the following result.

Theorem 1.5. Butcher [7] *If f satisfies a one-sided Lipschitz condition with one-sided Lipschitz constant l, and y and z are solutions of $y'(x) = f(x, y(x))$, then for all $x \geq x_0$,*

$$\|y(x) - z(x)\| \leq \exp(l(x - x_0))\|y(x_0) - z(x_0)\|.$$

Notice from this result that the distance between any two solutions will not increase rapidly or may even decrease if the equation has an adequate one-sided Lipschitz constant. Since stiffness is closely related to the behaviour of perturbations to a given solution, it is important to find out the effect of small perturbations with a one-sided Lipschitz condition.

Consider

$$y'(x) = f(x, y(x)) \tag{1.12}$$

with $y(x)$, a solution, and $\epsilon Y(x)$, a small perturbation to the given solution. Replace $y(x)$ in the equation (1.12) by $y(x) + \epsilon Y(x)$ and expand the solution in a series in powers of ϵ up to the second order, then get

$$y'(x) + \epsilon Y'(x) = f(x, y(x)) + \epsilon\frac{\partial f}{\partial y}Y(x). \tag{1.13}$$

Subtract the equation (1.12) from (1.13) and simplify it, then finally obtain the equation which controls the behaviour of the perturbation,

$$\begin{aligned} Y'(x) &= \frac{\partial f}{\partial y} Y(x) \\ &= J(x)Y(x) \end{aligned}$$

where $J(x)$ is the Jacobian matrix of $f(x, y(x))$. We can use the spectrum of eigenvalues of $J(x)$ to characterise stiffness. The eigenvalues of $J(x)$ determine the growth rate of the perturbation with a moderate change in the value of the solution and a very small change in $J(x)$ in a time interval Δx. The existence of one or more large and negative values of λ where $\lambda \in \sigma(J(x))$ where $x \in \Delta x$ indicates that stiffness is present.

1.3.1 Examples of Stiff Problem

Stiffness can be understood by the practical difficulty found in numerical calculation as well. The stiff problems are impossible or very difficult to solve by explicit methods, mainly because the small bounded stability region of explicit methods forces the numerical method to take very small stepsizes for the smooth solution. Two examples of stiff problems are given here to observe how explicit and implicit methods work for these problems.

Example 1. Stiff linear problem

Consider the stiff system of three linear ordinary diffrential equations with corresponding initial conditions.

$$\begin{bmatrix} y_1'(x) \\ y_2'(x) \\ y_3'(x) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ -L & 1 & L \end{bmatrix} \begin{bmatrix} y_1(x) \\ y_2(x) \\ y_3(x) \end{bmatrix}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \\ y_3(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \epsilon \end{bmatrix}$$

where $L = -25$ and $\epsilon = 2$.

The analytic solution is

$$\begin{bmatrix} y_1(x) \\ y_2(x) \\ y_3(x) \end{bmatrix} = \begin{bmatrix} \sin(x) \\ \cos(x) \\ \sin(x) + \epsilon \exp(Lx) \end{bmatrix},$$

which is drawn in *Figure 1.3*.

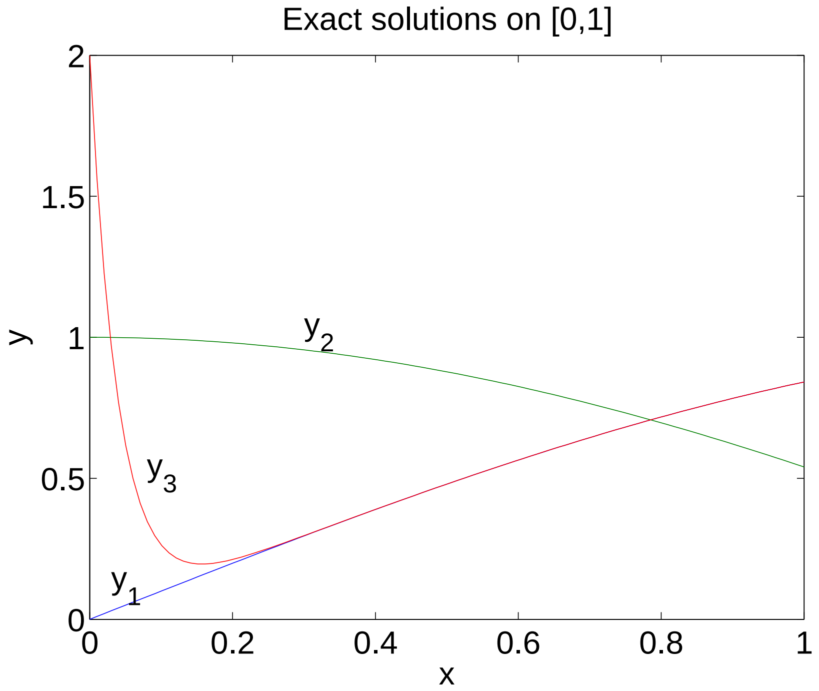


Figure 1.3: Analytic solution of *Example 1*, stiff linear problem

The results of using the explicit and implicit Euler methods for solving this stiff problem on the interval of $[0, 1]$ are presented in *Figure 1.5* and *Figure 1.6*. The first figure shows that the explicit method definitely seems to have difficulty approximating y_3 while y_1 and y_2 are computed without difficulties. Especially the approximations with $n = 10$ and $n = 15$ are hopeless. However, the implicit method performs perfectly well even for n as low as 4 as shown in *Figure 1.6*.

Example 2. Stiff nonlinear problem (The Kaps problem)

Consider the stiff system of two dimensional Kaps problem with corresponding initial conditions.

$$\begin{bmatrix} y_1'(x) \\ y_2'(x) \end{bmatrix} = \begin{bmatrix} -1002y_1(x) + 1000y_2(x)^2 \\ y_1(x) - y_2(x)(1 + y_2(x)) \end{bmatrix}, \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Analytic solution is

$$\begin{bmatrix} y_1(x) \\ y_2(x) \end{bmatrix} = \begin{bmatrix} \exp(-2x) \\ \exp(-x) \end{bmatrix},$$

which is drawn in *Figure 1.4*.

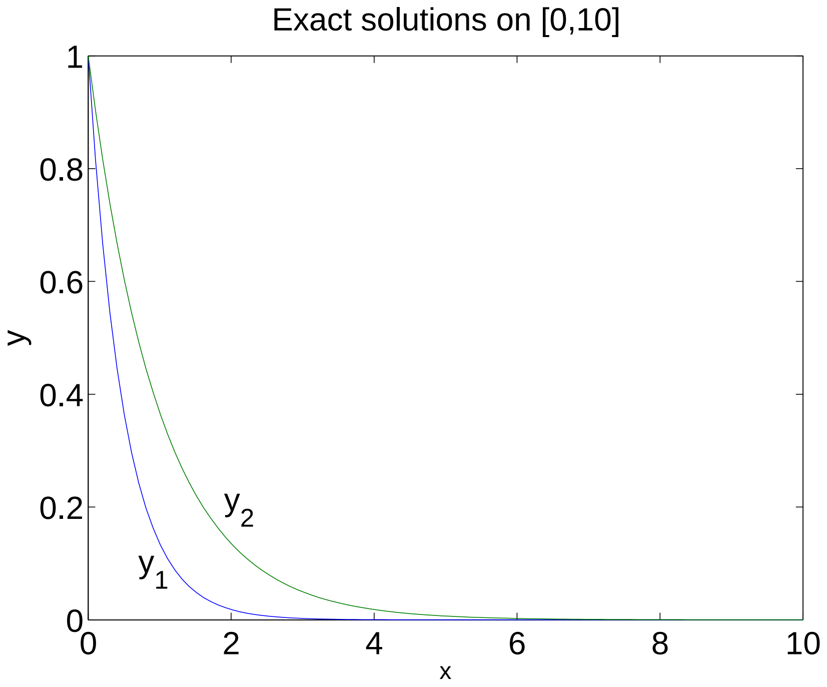


Figure 1.4: Analytic solution of *Example 2*, stiff nonlinear problem

In *Figure 1.7* and *Figure 1.8*, the computed solutions of this problem using the explicit and implicit Euler method on the interval of $[0, 10]$ are displayed. Even using a large number of steps, the explicit method performs poorly. However the implicit method easily gives a good approximation.

From these two examples, it is clearly confirmed that the explicit method is not suitable but the implicit method is appropriate for stiff problems.

Figure 1.5: Numerical solutions of *Example 1*, stiff linear problem, by explicit Euler method with various numbers of steps.

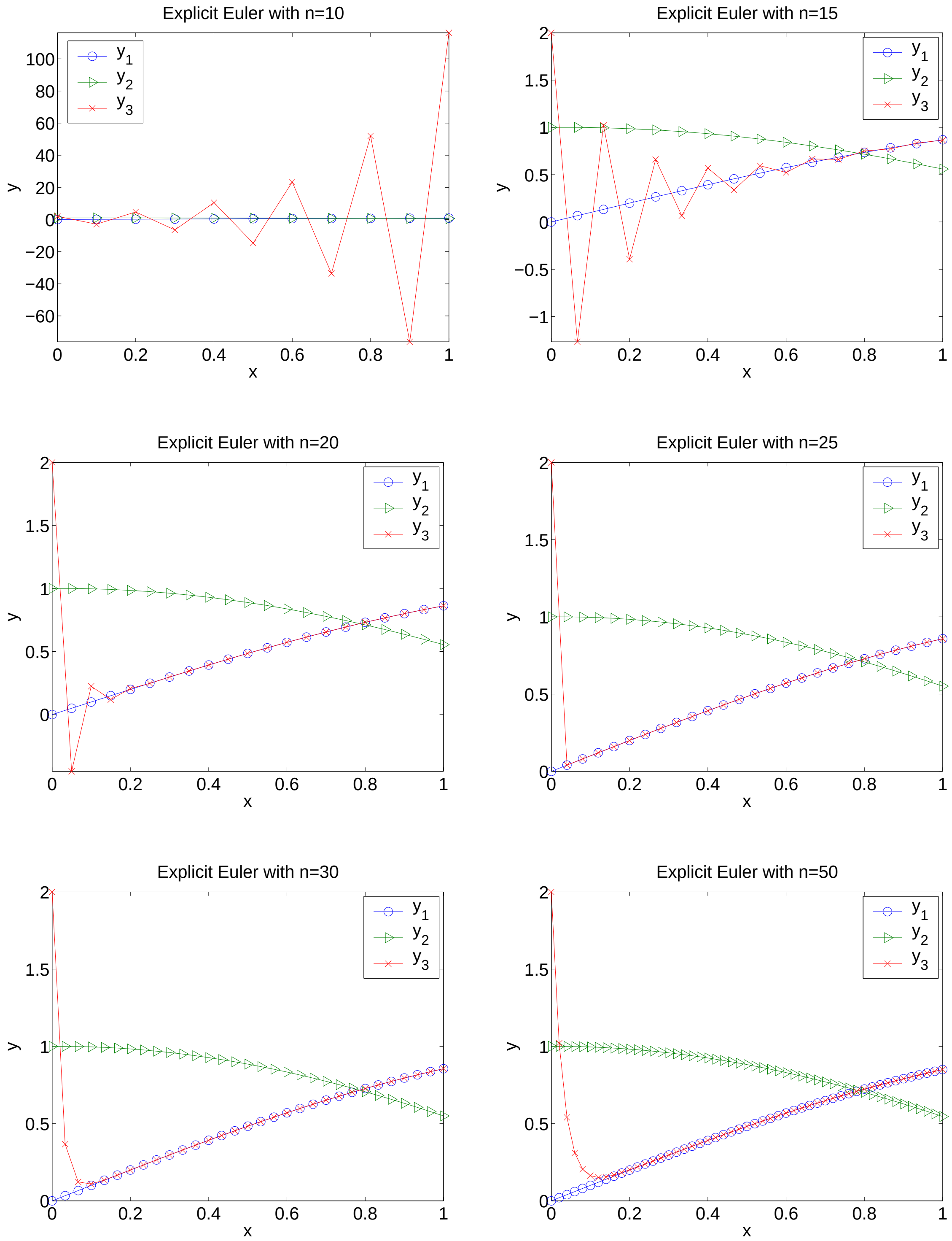
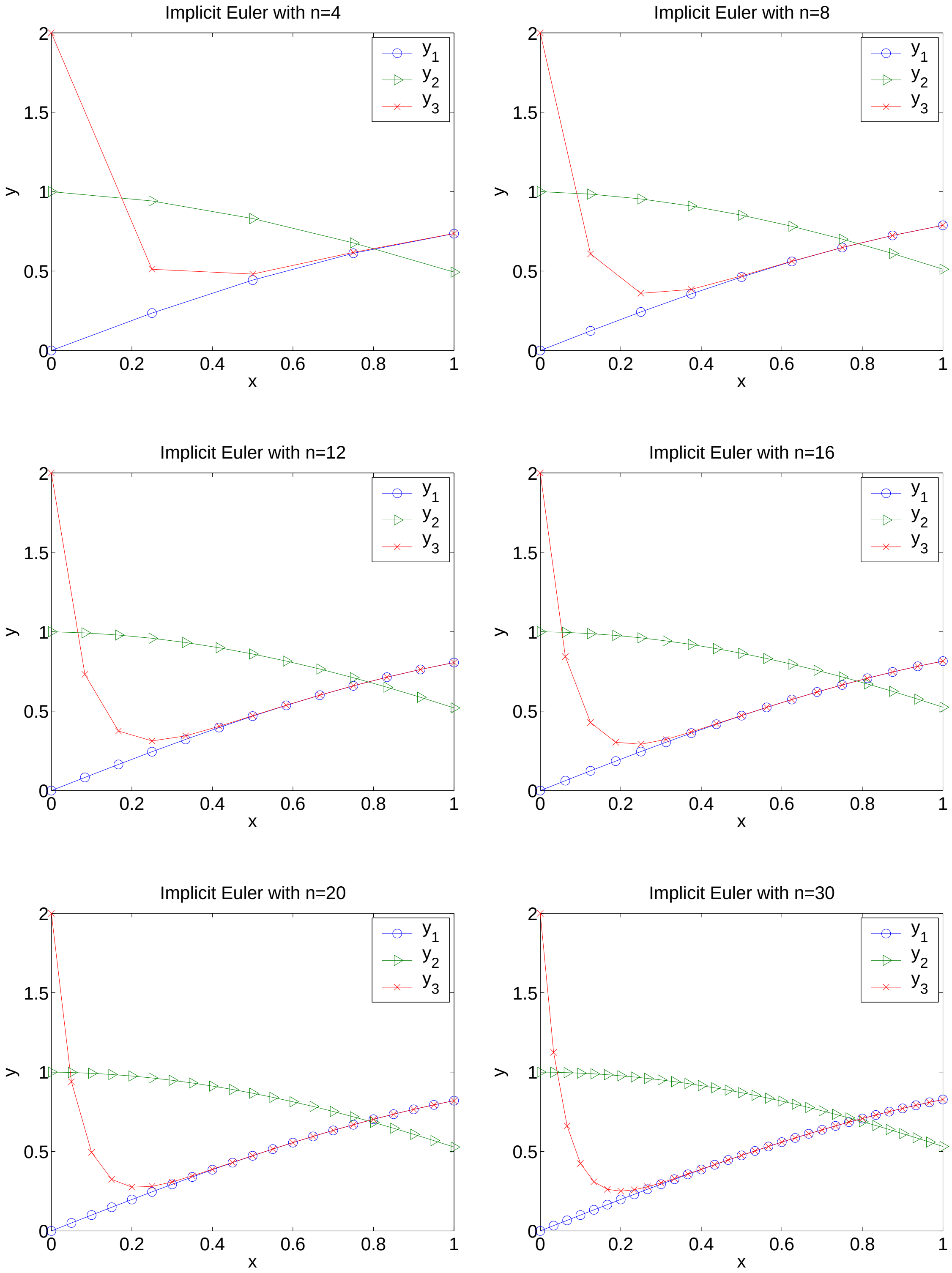


Figure 1.6: Numerical solutions of *Example 1*, stiff linear problem, by implicit Euler method with various numbers of steps.



Implicit Euler with n=4

Implicit Euler with n=8

Implicit Euler with n=12

Implicit Euler with n=16

Implicit Euler with n=20

Implicit Euler with n=30

Figure 1.7: Numerical solutions of *Example 2*, stiff nonlinear problem, by explicit Euler method with various numbers of steps.

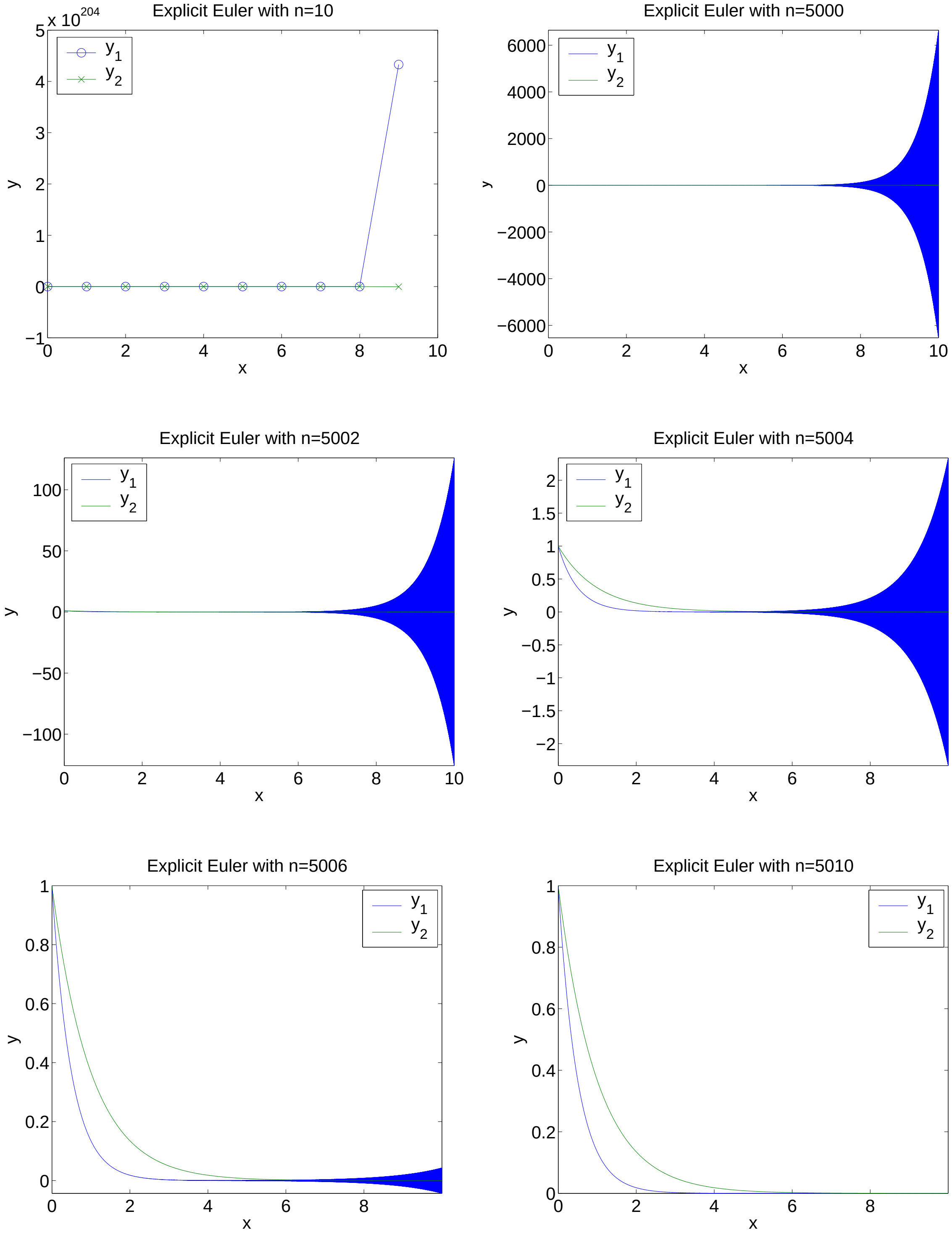
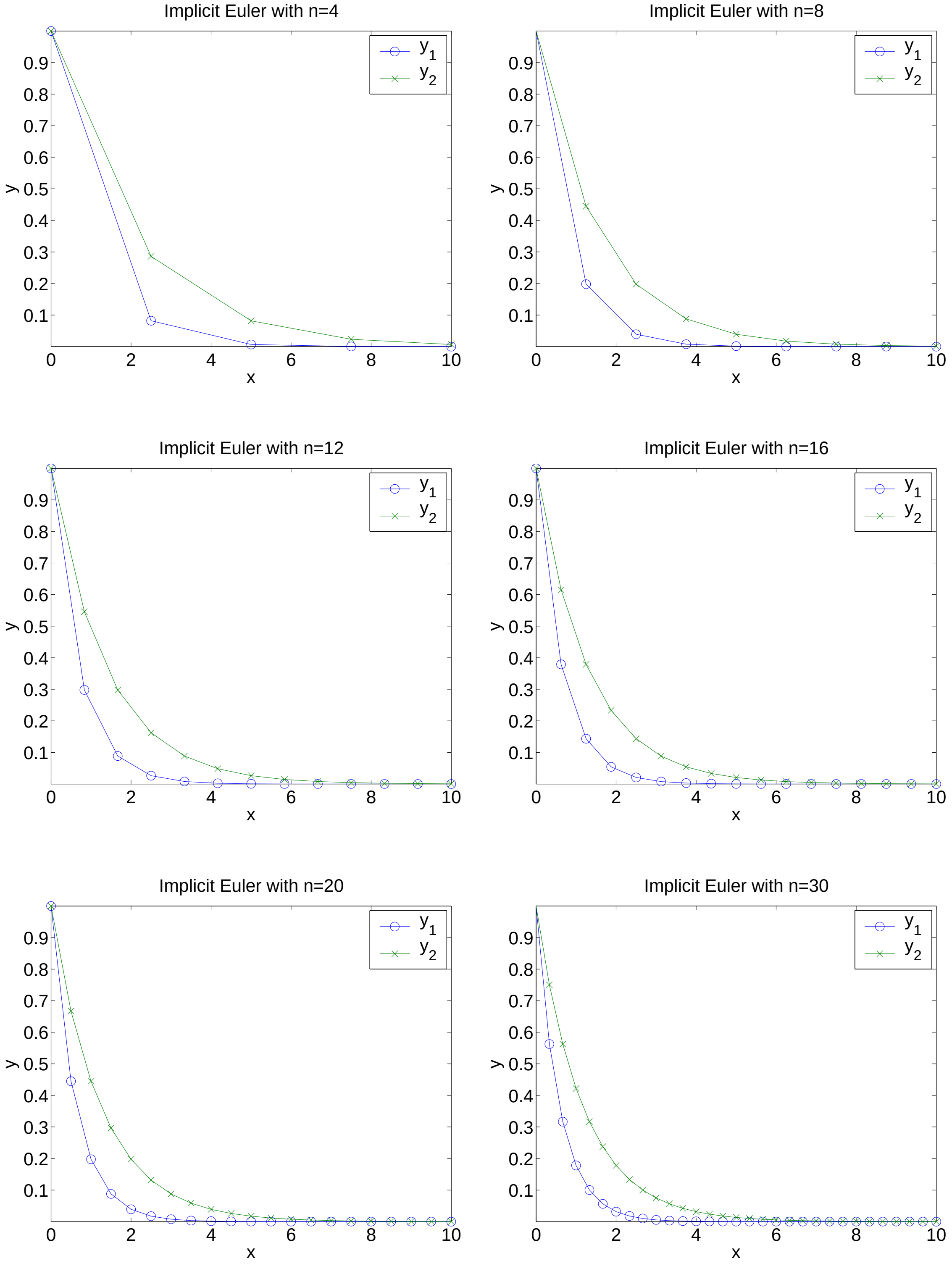


Figure 1.8: Numerical solutions of *Example 2*, stiff nonlinear problem, by implicit Euler method with various numbers of steps.



1.4 Framework of Thesis

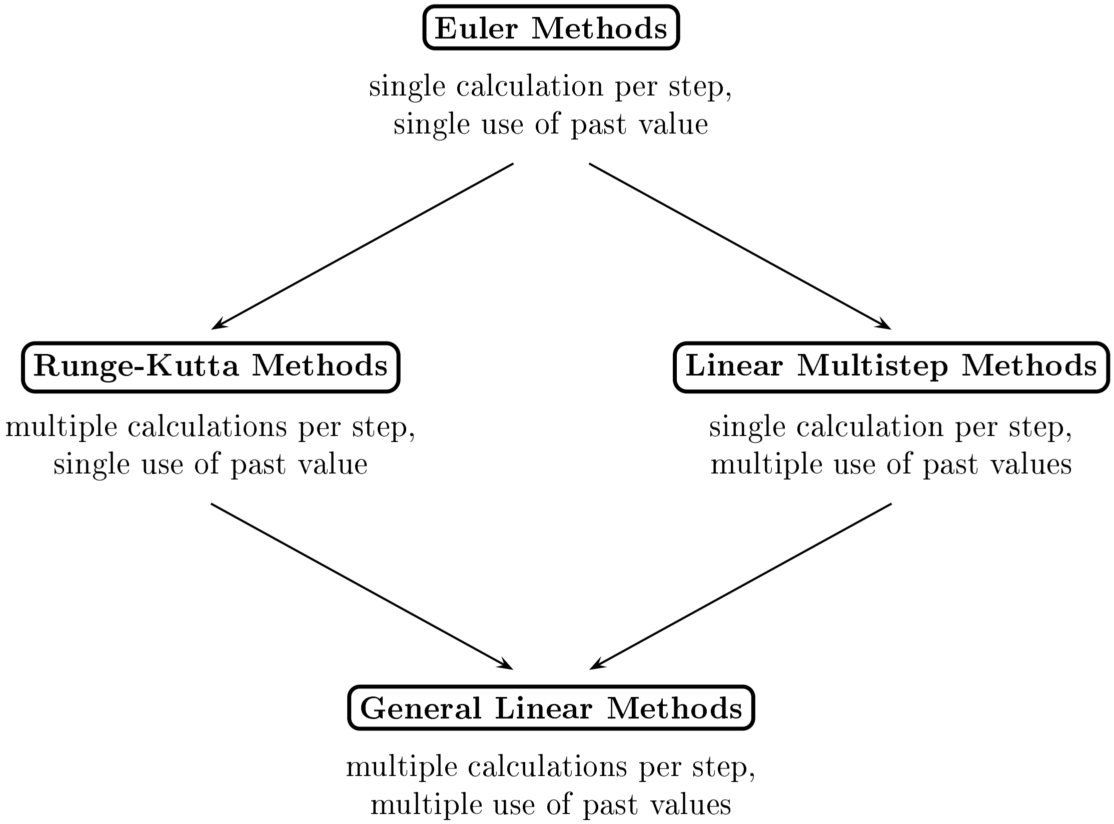

Euler Methods
single calculation per step,
single use of past value
Runge-Kutta Methods
multiple calculations per step,
single use of past value
Linear Multistep Methods
single calculation per step,
multiple use of past values
General Linear Methods
multiple calculations per step,
multiple use of past values

Figure 1.9: Types of numerical methods

The Euler method is simple. It uses only one piece of information from the past and evaluates the driving function only once per step. However, the Euler method is not very practical for computational purposes since considerable computational effort is required to improve accuracy. In spite of its limitaions, the Euler method is the fundamental building block for the higher accuracy methods which can be generalised to two main classes of the traditional methods, the multistage (Runge-Kutta) methods and the multistep (linear multistep) methods. Runge-Kutta methods allow more function evaluations per step while linear multistep methods use more information from past steps to obtain higher accuracy. These two methods provided a momentum for advancing numerical methods for ordinary differential equations. About thirty years ago, general linear methods were introduced to combine these two traditional classes of methods. In *Figure 1.9* the relation between the different types of methods is shown in a schematic diagram.

In the rest of this thesis, some of the important fundamental properties of the three numerical methods, extended from the Euler method, for ordinary differential equations are investigated. This includes the formulation for these methods, order and stability conditions and other basic concepts of methods. Chapter 2 covers Runge-Kutta methods with the order conditions using rooted trees. Chapters 3 and 4 are the discussion on linear multistep methods and general linear methods respectively. In Chapter 5, some numerical experiments on IRKS methods, which are one family of general linear methods, are presented. To begin with, the results of comparing the IRKS methods of order 1, 2 and 3 on the three test problems are presented. Then the predictors for the IRKS methods are investigated in detail to compare the performance of the predictors. Finally, a comparison between IRKS methods and two MATLAB built-in solvers, `ode15s` and `ode23s` are presented.

Chapter 2

Runge-Kutta Methods

Runge-Kutta methods compute the first derivative, f several times per step. This extension to the Euler method was first proposed by Runge [35] in 1895 and a little later, Heun [30] and Kutta [31] contributed to further early developement in this area.

Runge-Kutta methods have both advantages and disadvantages. Runge-Kutta methods are stable and easy to adapt for variable stepsize and order. However, they have difficulties in achieving high accuracy at reasonable cost.

In this chapter, a general introduction to Runge-Kutta methods is given. This includes the formulation of the methods and a discussion of different types of methods in this family. Then, order and stability conditions of Runge-Kutta methods are discussed.

2.1 Formulations

A Runge-Kutta method with s-stages is defined by

$$\begin{aligned} Y_i &= y_{n-1} + h \sum_{j=1}^{s} a_{ij} f\left(x_{n-1} + hc_j, Y_j\right), \quad i = 1, 2, ..., s \\ y_n &= y_{n-1} + h \sum_{j=1}^{s} b_j f\left(x_{n-1} + hc_j, Y_j\right). \end{aligned} \tag{2.1}$$

In this general formula, Y_i represnts the internal stage values and y_n is the update at the n^{th} step, that is the numerical approximation to the solution $y(x)$ at $x = x_n$. Naturally, h denotes the stepsize, $x_n - x_{n-1}$, and a_{ij}, b_j and c_j are the constant coefficients which can be constructed so that y_n is a good approximation to the solution $y(x_n) = y(x_{n-1} + h)$. For convenience, the stage derivatives $f\left(x_{n-1} + hc_j, Y_j\right)$ are often written as F_j.

A Runge-Kutta method needs to be consistent to be a suitable for solving problems. In other words, when a method is used to integrate an ordinary differential equation at x_{n-1}, the difference between the exact and the numerical solution at $x_{n-1} + h$ should tend to zero as h tends to zero. To ensure consistency, the following condition is required.

$$\sum_{j=1}^{s} b_j = 1. \tag{2.2}$$

Further to that, another condition,

$$\sum_{i=1}^{s} a_{ij} = c_j, \quad j = 1, 2, ..., s \tag{2.3}$$

is necessary to guarantee that the correct value is obtained at each of the stages.

The method can be characterised using the $s \times s$ matrix A and the $s \times 1$ vectors b and c,

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{ss} \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_s \end{bmatrix}, \qquad c = \begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix}$$

which can be conveniently presented using a Butcher tableau,

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \tag{2.4}$$

The components of the A matrix are the coefficients used to find the internal stages using linear combinations of the stage derivatives, b is the vector of weights indicating how the approximation to the solution depends on the derivatives of the internal stages and c is the vector of abscissae representing the position of the approximations in the step.

Runge-Kutta methods can be divided into two main types according to the style of the matrix A. If the A matrix is strictly lower triangular then the method is called ‘explicit’; otherwise, the method is called ‘implicit’. Implicit methods can be divided into several sub-categories, for example, ‘fully implicit’ if A is not lower triangular, ‘semi-implicit’ if A is lower triangular with at least one non-zero diagonal element, ‘diagonally implicit’ if A is lower triangular with all the diagonal elements equal and non-zero and ‘singly implicit’ if A is a non-singular matrix with a single eigenvalue.

2.1.1 Explicit Methods

Explicit methods are easy to implement and have cheap implementation costs because the internal stages can be calculated one after another without depending on later stages. However, explicit methods have poor stability, compared with implicit methods, which makes the methods unsuitable for stiff problems. The following is an example of an explicit method displayed in a Butcher tableau.

Example 1. Order 3 explicit method with 3 stages

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

Note that the upper triangular components of A are left blank indicating that their values are zero. This tableau gives the following scheme. The stage values at the n^{th} step are calculated as

$$\begin{aligned} Y_1 &= y_{n-1}, \\ Y_2 &= y_{n-1} + h(\tfrac{1}{2})F_1, \\ Y_3 &= y_{n-1} + h(-1)F_1 + h(2)F_2 \end{aligned}$$

with the stage derivatives,

$$\begin{aligned}
F_1 &= f(x_{n-1} + h(0), Y_1),\\
F_2 &= f(x_{n-1} + h(\tfrac{1}{2}), Y_2),\\
F_3 &= f(x_{n-1} + h(1), Y_3).
\end{aligned}$$

The approximation to the solution at the n^{th} step is

$$y_n = y_{n-1} + h(\tfrac{1}{6})F_1 + h(\tfrac{2}{3})F_2 + h(\tfrac{1}{6})F_3.$$

2.1.2 Implicit Methods

Implicit methods are expensive and difficult to implement since the stage-by-stage scheme is no longer available but a simultaneous computation is needed for the evaluation of stage values. However, the methods have some advantages compared with explicit methods, such as fewer stages for the same order and better stability which suits stiff problems. The following is the example of an implicit method displayed in a Butcher tableau.

Example 2. Order 4 implicit method with 2 stages (Gauss method)

$$\begin{array}{c|cc}
\frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6}\\
\frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4}\\
\hline
 & \frac{1}{2} & \frac{1}{2}
\end{array}$$

Note that the matrix A is not a lower triangular matrix, hence it is a fully implicit method. In addition, it has higer order than the explicit case, *Example 1*, with fewer stages. Here, the stage values Y_1 and Y_2 at the n^{th} step need to be evaluated simultaneously, for example, using a full Newton scheme.

$$\begin{aligned}
Y_1 &= y_{n-1} + h(\tfrac{1}{4})F_1 + h(\tfrac{1}{4} - \tfrac{\sqrt{3}}{6})F_2,\\
Y_2 &= y_{n-1} + h(\tfrac{1}{4} + \tfrac{\sqrt{3}}{6})F_1 + h(\tfrac{1}{4})F_2
\end{aligned}$$

where the stage derivatives are

$$\begin{aligned}
F_1 &= f(x_{n-1} + h(\tfrac{1}{2} - \tfrac{\sqrt{3}}{6}), Y_1),\\
F_2 &= f(x_{n-1} + h(\tfrac{1}{2} + \tfrac{\sqrt{3}}{6}), Y_2).
\end{aligned}$$

The approximation to the solution at the n^{th} step is

$$y_n = y_{n-1} + h(\tfrac{1}{2})F_1 + h(\tfrac{1}{2})F_2.$$

2.2 Order Conditions

In the investigation on numerical methods, it is very important to assess the quality of the numerical solutions obtained by the method from various points of view and one desirable property is good accuracy. One way of examining accuracy is to consider the rate at which the numerical solution converges to the exact solution, the order of the method. As the order gets higher, the requirements for a method to maintain a particular order become increasingly complicated. However, these order conditions of Runge-Kutta methods can be expressed using trees.

2.2.1 Trees

Let t denote a rooted tree and T denote the set of all rooted trees. Any $t \in T$ can be defined recursively by removing the root of t and denoting the subtrees trees as t_1, t_2, ..., t_m, where m is the number of disconnected trees from the root of t. Then, this relationship between t and t_1, t_2, ..., t_m is written as $t = [t_1 t_2 \cdots t_m]$. Recursive notations of the first 8 trees up to order 4 are shown in *Table 2.1*.

To be able to use these trees for constructing the order conditions, some special properties of rooted trees need to be discussed.

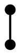
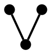
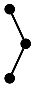
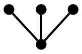
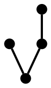
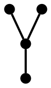
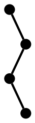
Definition 2.1. Butcher [3] *Let $r : T \to \mathbb{R}$ and then the order of the tree t, $r(t)$ is defined by*

$$r(t) = \begin{cases} 1 & \text{if } t = \tau, \\ 1 + r(t_1) + \cdots + r(t_m) & \text{if } t = [t_1 t_2 \cdots t_m]. \end{cases}$$

Note that the order $r(t)$ represents the number of vertices of the tree t.

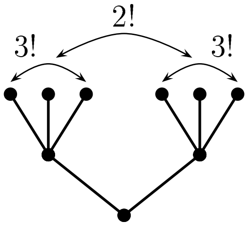
Definition 2.2. Butcher [3] *Let $\sigma : T \to \mathbb{R}$ and then the symmetry of the tree t, $\sigma(t)$ is defined by*

Table 2.1: Notation for trees and various functions on trees up to order 4

$r(t)$	t	Notation	$\sigma(t)$	$\gamma(t)$	$\alpha(t)$	$\beta(t)$
1	●	τ	1	1	1	1
2		$[\tau]$	1	2	1	2
3		$[\tau,\tau]=[\tau^2]$	2	3	1	3
3		$[[\tau]]$	1	6	1	6
4		$[\tau,\tau,\tau]=[\tau^3]$	6	4	1	4
4		$[\tau,[\tau]]$	1	8	3	24
4		$[[\tau,\tau]]=[[\tau^2]]$	2	12	1	12
4		$[[[\tau]]]$	1	24	1	24

$$\sigma(t) = \begin{cases} 1 & \text{if } t = \tau, \\ n_1!n_2!\cdots n_m!\sigma(t_1)^{n_1}\sigma(t_2)^{n_2}\cdots\sigma(t_m)^{n_m} & \text{if } t = [t_1^{n_1}t_2^{n_2}\cdots t_m^{n_m}]. \end{cases}$$

Note that the symmetry function $\sigma(t)$ is the order of the automorphism group of t and it is assumed that t_1, t_2, ..., t_m are all distinct for the tree $t = [t_1^{n_1}t_2^{n_2}\cdots t_m^{n_m}]$. Here is an example of finding the symmetry of the tree.

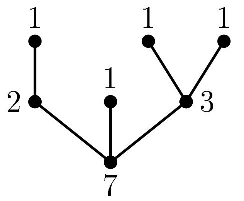


For this tree with nine vertices, the symmetry function $\sigma(t)$ is $\sigma(t) = 3!\times 3!\times 2! = 72$. All values of $\sigma(t)$ for the trees up to order 4 are given in *Table 2.1*.

Definition 2.3. Butcher [3] *Let* $\gamma : T \to \mathbb{R}$ *and then the density of the tree t,* $\gamma(t)$ *is defined by*

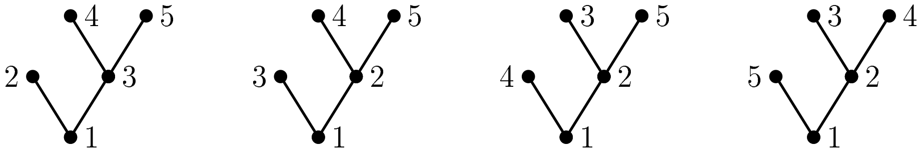
$$\gamma(t) = \begin{cases} 1 & \text{if } t = \tau, \\ r(t)\gamma(t_1)\gamma(t_2)\cdots\gamma(t)(t_m)^{n_m} & \text{if } t = [t_1^{n_1}t_2^{n_2}\cdots t_m^{n_m}]. \end{cases}$$

Note that the density function $\gamma(t)$ is the product of all the densities of the subtrees. For example,

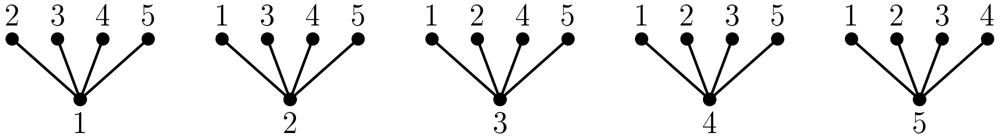


the density function $\gamma(t)$ of the tree with seven vertices is $\gamma(t) = 1\times 1\times 1\times 2\times 1\times 3\times 7 = 42$. See *Table 2.1* for $\gamma(t)$ of the trees up to order 4.

Furthermore, the labelling of a tree by numbers needs to be considered under three conditions. First, each vertex can have only one label, second, symmetry is not taken into account and third, label of each vertex must have a lower number than the other vertices which are connected to it. When all three conditions are applied, the number of distinct ways of labelling the given tree t is denoted by $\alpha(t)$. However, the third condition does not need to be applied for finding $\beta(t)$, the total number of ways of labelling. The values of $\alpha(t)$ and $\beta(t)$ for the trees up to order 4 is shown in *Table 2.1* as well. Here is an example of finding the function $\alpha(t)$. There are four acceptable labellings, that is $\alpha = 4$, for this tree with five vertices. Notice that the root is labelled with 1 only, since the third condition is applied.



Another example, the same number of vertices but with different shape, is given for finding the function $\beta(t)$. The third condition is not applied in this case. Hence, there exist five possible labellings.



The functions $\alpha(t)$ and $\beta(t)$ can be formulated using the functions $r(t)$, $\sigma(t)$ and $\gamma(t)$ by the following theorem.

Theorem 2.4. Butcher [7] *For all $t \in T$,*

$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)}, \qquad \beta(t) = \frac{r(t)!}{\sigma(t)}.$$

2.2.2 Elementary Differentials

The error in carrying out a single step of a Runge-Kutta method can be investigated by comparing the successive terms in the Taylor series expansions of the exact and the approximated solutions. In order to express as a Taylor series the exact solution, $y(x)$, of the autonomous differential equation,

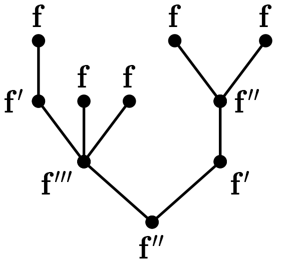
$$y'(x) = f\left(y(x)\right), \tag{2.5}$$

the derivatives of $y(x)$ are required. By introducing the new notation $\mathbf{f}$ called the 'elementary differential $F(t)(y(x))$' for $f(y(x))$, the first three derivatives $y'(x)$, $y''(x)$ and $y'''(x)$ can be represented as follows.

$$\begin{aligned}
y'(x) &= f(y(x)),\\
&= \mathbf{f},\\
y''(x) &= f'(y(x))(y'(x)),\\
&= f'(y(x))(f(y(x))),\\
&= \mathbf{f}'\mathbf{f},\\
y'''(x) &= f''(y(x))(y'(x), f(y(x))) + f'(y(x))(f'(y(x))(y'(x))),\\
&= f''(y(x))(f(y(x)), f(y(x))) + f'(y(x))(f'(y(x))(f(y(x)))),\\
&= \mathbf{f}''(\mathbf{f},\mathbf{f}) + \mathbf{f}'\mathbf{f}'\mathbf{f}'.
\end{aligned}$$

Note that the third derivative of $y(x)$ is the linear combination of two elementary differentials. In fact, any derivative of $y(x)$ can be defined by a linear combination of elementry differentials.

It seems that elementary differentials are complicated to derive. However, there is an excellent way to determine elementary differentials using rooted trees. All terminal vertices represent **f** and all other vertices which have n numbers of branches represent $\mathbf{f}^n$. For example, the following tree with ten vertices,



has the elementary differential $F(t)(y(x)) = \mathbf{f}''(\mathbf{f}'''(\mathbf{f}'\mathbf{f},\mathbf{f},\mathbf{f}),\mathbf{f}'\mathbf{f}''(\mathbf{f},\mathbf{f}))$ which appears in the Taylor series expansion of $y^{(10)}(x)$. The elementary differentials for trees with order up to 4 are given in *Table 2.2.* The Taylor series expansion for $y^{(4)}(x)$ consists of the linear combination of four elementary differentials obtained from order 4 trees shown in the table. In the following definition, the elementary differentials are defined more formally.

Definition 2.5. Butcher [3] *For all $y \in \mathbb{R}^N$ and a function $f : \mathbb{R}^N \to \mathbb{R}^N$, the elementary differential, $F(t)(y(x))$ corresponding to $t \in T$ is recursively defined by*

$$F(t)(y) = \begin{cases} f(y(x)), & \text{if } t = \tau, \\ f^{(m)}(y(x))(F(t_1)(y(x)), F(t_2)(y(x)), ..., F(t_m)(y(x))), & \text{if } t = [t_1 t_2 \cdots t_m]. \end{cases}$$

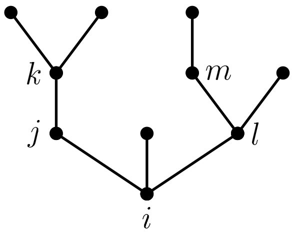
Furthermore, the derivative of y of a particular order can be represented by combining all the elementary differentials of this order.

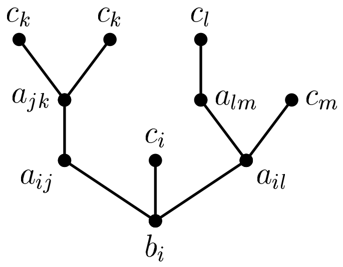
Theorem 2.6. Butcher [3] *If $f : \mathbb{R}^N \to \mathbb{R}^N$ is differentiable $k-1$ times at $y(x)$ and $y'(x) = f(y(x))$, then y is differentiable n times at x and*

$$y^{(k)} = \sum_{r(t)=k} \alpha(t) F(t)(y(x)).$$

2.2.3 Elementary Weights

Recall the general formula of a Runge-Kutta method shown in equation (2.1). A polynomial expression using the coefficients of the method with a particular order, called the 'elementary weight $\Phi(t)$', can be derived from each tree t for all $t \in T$ up to the particular order. Here is an example of obtaining elementary weights from a tree with ten vertices.



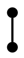
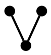
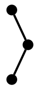
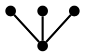
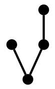
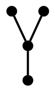
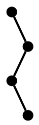


First of all, starting from the root, label vertices which have outward branches as i, j, ..., except terminal vertices. Then, name the root as b_i and other vertices, except terminal vertices, as a_{**} where the first subscript is the label attached to the parent vertex and the second subscript is the label of the current vertex. Finally, name all terminal vertices as c_* where $*$ is the label attached to the parent vertex. The elementary weight is the sum of all possible choices of their product. For this example, the elementary weight is

$$\Phi(t) = \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k c_k c_i a_{il} a_{lm} c_m c_l = \sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k^2 c_i a_{il} a_{lm} c_m c_l.$$

In a more formal context, the elementary weights can be defined recursively.

Table 2.2: Elementrary differentials and elementary weights for trees up to order 4

$r(t)$	t	$F(t)$	$\Phi(t)$	$=$	$\frac{1}{\gamma(t)}$
1		$\mathbf{f}$	$\sum_{i=1}^{s} b_i$	$=$	1
2		$\mathbf{f}'\mathbf{f}$	$\sum_{i=1}^{s} b_i c_i$	$=$	$\frac{1}{2}$
3		$\mathbf{f}''(\mathbf{f},\mathbf{f})$	$\sum_{i=1}^{s} b_i {c_i}^2$	$=$	$\frac{1}{3}$
3		$\mathbf{f}'\mathbf{f}'\mathbf{f}$	$\sum_{i,j=1}^{s} b_i a_{ij} c_j$	$=$	$\frac{1}{6}$
4		$\mathbf{f}'''(\mathbf{f},\mathbf{f},\mathbf{f})$	$\sum_{i=1}^{s} b_i {c_i}^3$	$=$	$\frac{1}{4}$
4		$\mathbf{f}''(\mathbf{f},\mathbf{f}'\mathbf{f})$	$\sum_{i,j=1}^{s} b_i c_i a_{ij} c_j$	$=$	$\frac{1}{12}$
4		$\mathbf{f}'\mathbf{f}''(\mathbf{f},\mathbf{f})$	$\sum_{i,j=1}^{s} b_i a_{ij} {c_j}^2$	$=$	$\frac{1}{8}$
4		$\mathbf{f}'\mathbf{f}'\mathbf{f}'\mathbf{f}$	$\sum_{i,j,k=1}^{s} b_i a_{ij} a_{jk} c_k$	$=$	$\frac{1}{24}$

Definition 2.7. Butcher [7] *For an s stage Runge-Kutta method, the elementary weights, $\Phi(t)$, are defined by*

$$\Phi(t) = \begin{cases} \sum\limits_{i=1}^{s} b_i, & \text{if } t = \tau, \\ \sum\limits_{i=1}^{s} b_i \Phi_i(t_1)\Phi_i(t_2)\ldots\Phi_i(t_m), & \text{if } t = [t_1 t_2 \cdots t_m], \end{cases}$$

where $\Phi_i(t)$ is elementary stage weight for the i^{th} stage and is defined by

$$\Phi_i(t) = \begin{cases} \sum\limits_{j=1}^{s} a_{ij} = c_i, & \text{if } t = \tau, \\ \sum\limits_{j=1}^{s} a_{ij} \Phi_j(t_1)\Phi_j(t_2)\ldots\Phi_j(t_m), & \text{if } t = [t_1 t_2 \cdots t_m]. \end{cases}$$

The elementary weights for trees with order up to 4 are shown in *Table 2.2.*

2.2.4 Conditions for Order

To derive conditions for a Runge-Kutta method to have a specific order p, the Taylor series expansion of the approximated solution needs to be compared with the exact solution. The exact solution at x_n, $y(x_n) = y(x_{n-1} + h)$, to order p is represented by the Taylor series expansion

$$\begin{aligned} y(x_{n-1} + h) &= y(x_{n-1}) + hy'(x_{n-1}) + \frac{h^2}{2!}y''(x_{n-1}) + \cdots + \frac{h^p}{p!}y^{(p)}(x_{n-1}) + O(h^{p+1}) \\ &= y(x_{n-1}) + \sum_{k=1}^{p} \frac{h^k}{k!} y^{(k)}(x_{n-1}) + O(h^{p+1}). \end{aligned}$$

This can be expressed using *Theorem 2.6* and *Theorem 2.4*,

$$\begin{aligned} y(x_{n-1} + h) &= y(x_{n-1}) + \sum_{k=1}^{p} \frac{h^k}{k!} \sum_{r(t)=k} \alpha(t) F(t)(y(x_{n-1})) + O(h^{p+1}) \\ &= y(x_{n-1}) + \sum_{k=1}^{p} \frac{h^k}{k!} \sum_{r(t)=k} \frac{k!}{\sigma(t)\gamma(t)} F(t)(y(x_{n-1})) + O(h^{p+1}) \\ &= y(x_{n-1}) + \sum_{k=1}^{p} h^k \sum_{r(t)=k} \frac{1}{\sigma(t)\gamma(t)} F(t)(y(x_{n-1})) + O(h^{p+1}). \end{aligned}$$

For all $t \in T$,

$$y(x_{n-1} + h) = y(x_{n-1}) + \sum_{t \in T} h^{r(t)} \frac{1}{\sigma(t)\gamma(t)} F(t)(y(x_{n-1})) + O(h^{p+1}). \tag{2.6}$$

The approximated solution at x_n produced after one step of a Runge-Kutta method, y_n, can be determined using the elementary weights over all $t \in T$.

$$y_n = y_{n-1} + \sum_{t \in T} h^{r(t)} \frac{1}{\sigma(t)} \Phi(t) F(t)(y(x_{n-1})) + O(h^{p+1}). \tag{2.7}$$

Compare the expressions, (2.6) and (2.7) and deduce the fact that the Taylor series expansion of the exact solution and the Runge-Kutta approximation are equal up to h^p term, if

$$\Phi(t) = \frac{1}{\gamma(t)} \tag{2.8}$$

for $r(t) \leq p$. The order conditions for a Runge-Kutta method corresponding to trees with order up to 4 are displayed in *Table 2.2*.

By applying the order conditions together with the two consistency conditions, (2.2) and (2.3), a Runge-Kutta method of order p can be constructed so that the computed solution y_n is a good approximation to the exact solution $y(x_n)$.

2.3 Stability

The Runge-Kutta tableau given below is for the order 3 explicit method with 3 stages.

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

Consider the standard test problem $y' = \lambda y$ where λ is a complex number. The exact solution of this linear test problem, $y' = \lambda y$ is bounded, if $Re(\lambda) \leq 0$. If the Runge-Kutta method is applied to this test problem, then the 'Stability Function', $R(z)$ is defined by

$$R(z) = 1 + zb^T(I - zA)^{-1}\mathbf{1} \tag{2.9}$$

However, to derive the stability function, let $z = h\lambda$. Then,

$$\begin{array}{rcl} Y_1 &=& y_0 \\ Y_2 &=& y_0 + \frac{h}{2}F_1 = (1+\frac{z}{2})y_0 \\ Y_3 &=& y_0 - hF_1 + 2hF_2 = (1-z+2z(1+\frac{z}{2}))y_0 = (1+z+z^2)y_0 \end{array}$$

where

$$\begin{array}{rcl} F_1 &=& \lambda y_0 \\ F_2 &=& \lambda(1+\frac{z}{2})y_0 \\ F_3 &=& \lambda(1+z+z^2)y_0 \end{array}$$

Hence,

$$\begin{array}{rcl} y_1 &=& y_0 + h(\frac{1}{6}F_1 + \frac{2}{3}F_2 + \frac{1}{6}F_3) \\ &=& y_0 + \frac{1}{6}zy_0 + \frac{2}{3}z(1+\frac{z}{2})y_0 + \frac{1}{6}z(1+z+z^2)y_0 \\ &=& y_0(1+\frac{1}{6}z + \frac{2}{3}z + \frac{1}{3}z^2 + \frac{1}{6}z + \frac{1}{6}z^2 + \frac{1}{6}z^3) \\ &=& y_0(1+z+\frac{1}{2}z^2 + \frac{1}{6}z^3) \end{array}$$

Note that $1+z+\frac{1}{2}z^2+\frac{1}{6}z^3$ is the stability function, $R(z)$, which is the first 4 terms of $\exp(z)$ series. This illustrates that the stability function for an explicit Runge-Kutta method with s-stage and order s is

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + ... + \frac{z^s}{s!}.$$

For example, if $y_0 = 1$, then

$$y(x_0+h) = \exp(h\lambda) = 1 + h\lambda + \frac{h^2\lambda^2}{2!} + \frac{h^3\lambda^3}{3!} + ... + \frac{h^s\lambda^s}{s!} + O(h^{s+1}).$$

The 'Stability Region' is the part of the complex plane such that

$$|R(z)| \leq 1.$$

These are the stability functions up to order 4.

$$\begin{array}{rcll} R(z) &=& 1+z, & s=p=1 \\ R(z) &=& 1+z+\frac{1}{2}z^2, & s=p=2 \\ R(z) &=& 1+z+\frac{1}{2}z^2+\frac{1}{6}z^3, & s=p=3 \\ R(z) &=& 1+z+\frac{1}{2}z^2+\frac{1}{6}z^3+\frac{1}{24}z^4, & s=p=4. \end{array}$$

The plots of the boundaries of the stability regions defined by above functions are shown in *Figure 2.1*, where the unshaded part represents the stable region. In each case, the stability region is the bounded set enclosed by the curves. To find the boundaries, that is find the values of z for which $|R(z) = 1|$, let $1 + z + \frac{z^2}{2!} + ... + \frac{z^s}{s!} = \exp(i2\pi\theta)$ and find the roots of the polynomial. For example, when $p = 3$

$$1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} = e^{i2\pi\theta}$$
$$(1 - e^{i2\pi\theta}) + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 = 0.$$

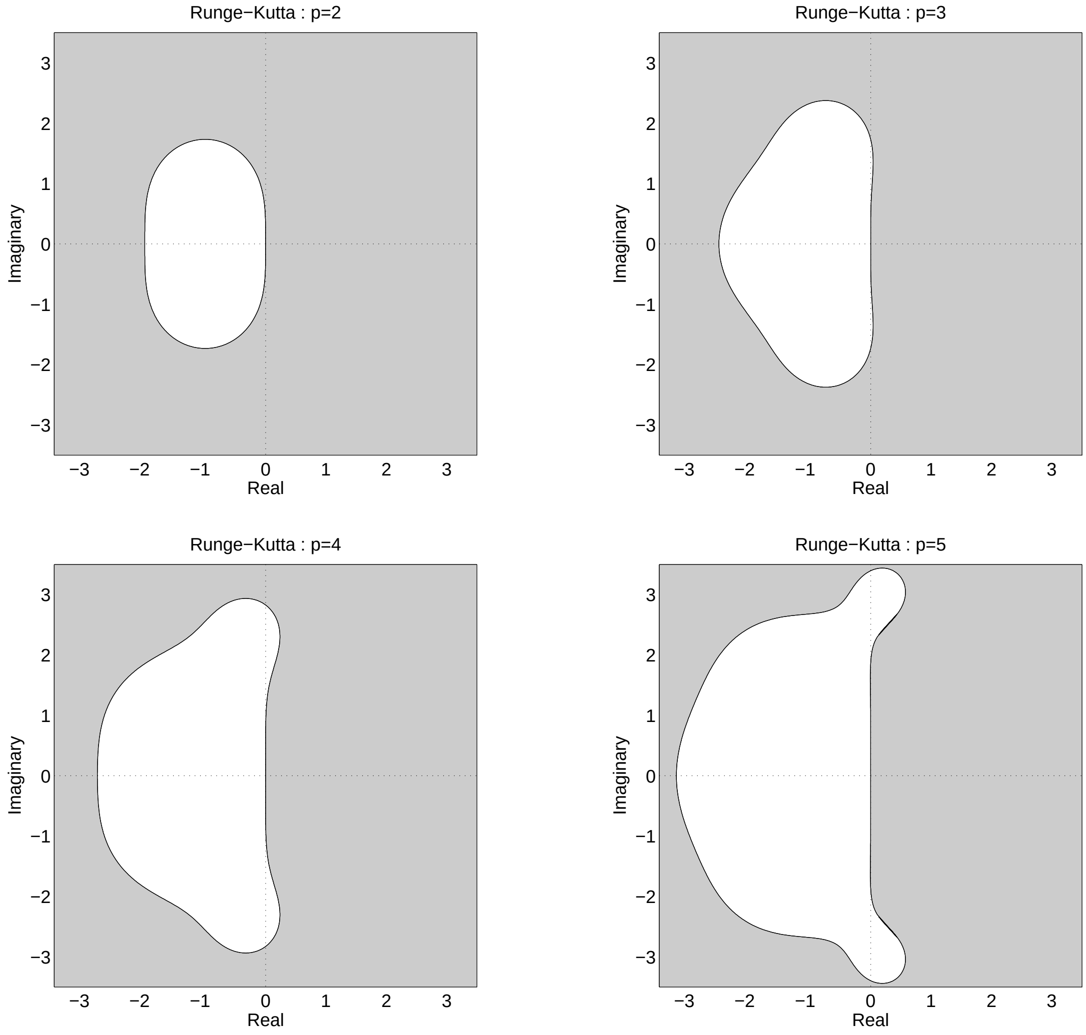


Figure 2.1: Stability Regions for explicit Runge-Kutta methods

Chapter 3

Linear Multistep Methods

Another extension to the Euler method is linear multistep methods. Unlike the Runge-Kutta methods, linear multistep methods use several values of the function and its derivative from previous steps. In 1883, Adams and Bashforth [1] introduced the explicit types of method, well-known as Adams-Bashforth methods. Later, Moulton [33] developed the idea of Adams and Bashforth further and proposed the implicit type of method, Adams-Moulton method.

Linear multistep methods, as for Runge-Kutta methods, have both advantages and disadvantages. Implementation cost is low compared with the Runge-Kutta method due to fewer function evaluations per step. However, adapting variable stepsize and order and the stability questions are more complicated and difficult than for Runge-Kutta methods.

A brief discussion of linear multistep methods are given in this chapter including the property of consistency, stability and convergence. Moreover, the order and stability conditions of three main types linear multistep methods are covered.

3.1 Formulation

A general form of a linear multistep method that depends on k previous steps is given by

$$y_n = \sum_{i=1}^{k} \alpha_i y_{n-i} + h \sum_{i=0}^{k} \beta_i f(y_{n-i}), \tag{3.1}$$

where y_n is the numerical approximation to the exact value, $y(x_n)$ at the point x_n. The method can be characterised by the values of α and β which can be written as $[\alpha, \beta]$. In addition, the polynomials $\alpha(x)$ and $\beta(x)$ are defined by $1 - \alpha_1 x - \alpha_2 x^2 - \alpha_3 x^3 - \cdots - \alpha_k x^k$, and $\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_k x^k$ respectively.

The method can be classified as two groups, explicit and implicit. The method is explicit if $\beta_0 = 0$, that is y_n depends only on past values. The first linear multistep methods, Adams-Bashforth methods are explicit. The order k Adams-Bashforth method has the form

$$y_n = y_{n-1} + h(\beta_1 f(x_{n-1}, y_{n_1}) + \beta_2 f(x_{n-2}, y_{n_2}) + \cdots + \beta_k f(x_{n-k}, y_{n_k})).$$

If $\beta_0 \neq 0$ then the method is called implicit. A typical example of an implicit method, the Adams-Moulton method with order $k+1$ has the general form

$$y_n = y_{n-1} + h(\beta_0 f(x_n, y_n) + \beta_1 f(x_{n-1}, y_{n_1}) + \cdots + \beta_k f(x_{n-k}, y_{n_k})).$$

Linear multistep methods are suitable for predictor-corrector schemes. The value of the function is first estimated by the predictor, Adams-Bashforth, and then used by the corrector, Adams-Moulton, to improve the calculated solution. This approach is known as 'Predict-Evaluate-Correct' (PEC) methods, where the derivative is evaluated only after the predictor step, or 'Predict-Evaluate-Correct-Evaluate' (PECE) methods, where the derivative is re-evaluated after the corrector step. Furthermore, the PECE methods can be extended to P(EC)^{m}E methods where a number of correction steps and derivative evaluations are used.

3.2 Consistency, Stability and Convergence

As with Runge-Kutta methods, linear multistep methods need to be consistent in order for the numerical solution to converge to the exact solution as the stepsize h tends to zero. Consider the numerical solution computed by the equation (3.1), a linear multistep method is consistent if it is able to solve exactly two simple differential equations, $y'(x) = 0$ with the initial condition $y(x_0) = 1$, and $y'(x) = 1$ with the initial condition $y(x_0) = 0$.

First, assume that the exact solutions at $x_{n-k}, x_{n-k+1}, ..., x_{n-1}$ are known so that $y_i = 1$ for $i = n-k, n-k+1, ..., n-1$. The approximated solution at x_n is therefore,

$$y_n = \alpha_1 + \alpha_2 + \cdots + \alpha_k, \tag{3.2}$$

which can give the correct solution if, and only if

$$1 = \alpha_1 + \alpha_2 + \cdots + \alpha_k. \tag{3.3}$$

A method which satisfies equation (3.3) is said to be 'preconsistent'.

The second equation is $y'(x) = 1$ with the initial condition $y(x_0) = 0$ which has the exact solution at each step given by $y_i = hi$. Again assume that the solution is known exactly for the previous k points, then the correct solution at y_n is given by

$$hn = \alpha_1(n-1)h + \alpha_2(n-2)h + \cdots + \alpha_k(n-k)h + h\left(\beta_0 + \beta_1 + \cdots + \beta_k\right). \tag{3.4}$$

If a factor of h from both sides is cancelled and the equation (3.3) multipled by n is subtracted, the following equation can be derived.

$$\alpha_1 + 2\alpha_2 + 3\alpha_3 + \cdots + k\alpha_k = \beta_0 + \beta_1 + \cdots + \beta_k \tag{3.5}$$

A linear multistep method is said to be consistent if it satisfies the conditions of equations (3.3) and (3.5).

Consistency condition ensures the local accuracy of the method. However, this condition is not sufficient to guarantee convergence. Because the error introduced in step s may accumulate over the integration process and affect the solution of later steps as these depend on the values of y_s and on $hy'_s = hf(x_s, y_s)$. Ignore the error propagation due to the derivatives to make the analysis of the growth of these errors simpler. Then, for a method to be stable, the difference between the exact and the approximated solutions given by the differential equation $y'(x) = 0$ with the initial condition $y(x_0) = 0$ need be bounded as h tends to zero. Hence, simpler corresponding difference equation is obtained.

$$y_n = \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \cdots + \alpha_k y_{n-k}. \tag{3.6}$$

Definition 3.1. Butcher [7] *A linear multistep method $[\alpha, \beta]$ is stable if the difference equation (3.6) has only bounded solutions.*

The following theorem characterises convergence of methods related to consistency and stability.

Theorem 3.2. Dahlquist [26] *A linear multistep method is convergent if and only if it is stable and consistent.*

Furthermore, three supporting theorems state that stability and consistency are neccessary for convergence using the same simple problem introduced in the beginning of the section.

Theorem 3.3. Butcher [7] *A convergent linear multistep method is stable.*

Proof. *For a stable method all solutions to $y_n = \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + \cdots + \alpha_k y_{n-k}$ are bounded, hence, an unstable method must have an unbounded solution to*

$$z_n = \alpha_1 z_{n-1} + \alpha_2 z_{n-2} + \cdots + \alpha_k z_{n-k}.$$

Define the sequence ζ by

$$\zeta_n = \max_{i=0}^{n} |z_i|,$$

then ζ converges monotonically to infinity. Consider the differential equation

$$y'(x) = 0, \qquad y(0) = 0$$

with the solution computed at $x_{end} = 1$. Take n steps of size $h = \frac{1}{n}$ then the starting values given by $y_i = \frac{z_i}{\zeta_n}$, $i = 0, 1, ..., k-1$ satisfy our condition that $y_i \to y(x_0)$ as $h \to 0$ since $\zeta_n \to \infty$ as $n \to \infty$. The numerical approximation computed for $y(1)$ is $\frac{z_n}{\zeta_n}$. Because ζ is unbounded, there are an infinite number of n for which

$$|\zeta_n| > \max_{i=0}^{n-1} |z_i|.$$

For such n, $z_n = \zeta_n$ and hence $\frac{z_n}{\zeta_n} = 1$ so the sequence $n \mapsto \frac{z_n}{\zeta_n}$ cannot converge to zero and hence an unstable method cannot be convergent. □

Theorem 3.4. Butcher [7] *A convergent linear multistep method is preconsistent.*

Proof. *Consider the fact that a convergent method is stable. Define z as the solution to the difference equation*

$$z_n = \alpha_1 z_{n-1} + \alpha_2 z_{n-2} + \cdots + \alpha_k z_{n-k} \tag{3.7}$$

with the initial values given by $z_i = 1$, $i = 0, 1, 2, \ldots, k-1$. Consider the differential equation

$$y'(x) = 0, \qquad y(0) = 1, \qquad x_{end} = 1.$$

The solution computed after n steps is given by $y_n = z_n$. Since the method is convergent, $y_n \to 1$ as $n \to \infty$ and it follows that there exists N large enough such that $\forall n > N$ $|y_i - 1| \leq \epsilon$, $\epsilon > 0$ for $i = n-k, n-k+1, \ldots, n$. Hence the equation (3.7) can be rearranged as follows

$$\begin{aligned}
0 &= z_n - \alpha_1 z_{n-1} - \alpha_2 z_{n-2} - \cdots - \alpha_k z_{n-k} \\
&= (z_n - 1) + 1 - \alpha_1(z_{n-1} - 1) - \alpha_1 \\
&\quad -\alpha_2(z_{n-2} - 1) - \alpha_2 - \cdots - \alpha_k(z_{n-k} - 1) - \alpha_k \\
\implies 1 - \alpha_1 - \alpha_2 - \cdots - \alpha_k &= (1 - z_n) + \alpha_1(z_{n-1} - 1) \\
&\quad +\alpha_2(z_{n-2} - 1) + \cdots + \alpha_k(z_{n-k} - 1) \\
\implies |1 - \alpha_1 - \alpha_2 - \cdots - \alpha_k| &\leq |1 - z_n| + |\alpha_1||(z_{n-1} - 1)| \\
&\quad +|\alpha_2||(z_{n-2} - 1)| + \cdots + |\alpha_k||(z_{n-k} - 1)| \\
&\leq \epsilon + |\alpha_1|\epsilon + \cdots + |\alpha_k|\epsilon \\
&\leq \epsilon\left(1 + \sum_{i=1}^{k} |\alpha_i|\right)
\end{aligned}$$

Because $1 - \alpha_1 - \alpha_2 - \cdots - \alpha_k \to 0$ as $\epsilon \to 0$ hence $\alpha_1 - \alpha_2 - \cdots - \alpha_k = 1$ and the method is preconsistent. □

Theorem 3.5. Butcher [7] *A convergent linear multistep method is consistent.*

Proof. *Consider the fact that a convergent method is stable. Hence, note that $\alpha_1 + 2\alpha_2 + \cdots + k\alpha_k \neq 0$ or the method would not be stable. Define the sequence*

$$z_p = \frac{\beta_0 + \beta_1 + \cdots + \beta_k}{\alpha_1 + 2\alpha_2 + \cdots + k\alpha_k} p = \frac{p\sum_{i=0}^{k} \beta_i}{\sum_{j=1}^{k} j\alpha_j}, \qquad = 0, 1, 2, ...$$

Consider the numerical solution to the differential equation

$$y'(x) = 1, \qquad y(0) = 0$$

with the solution computed at $x_{end} = 1$ using n steps of stepsize $h = \frac{1}{n}$. Choose starting approximations as

$$y_i = \frac{1}{n} z_i, \tag{3.8}$$

for $i = 0, 1, 2, ..., k-1$ so that these values converge to zero as $n \to \infty$. The computed solution for all $i = k, k+1, ..., n$ can be given by (3.8). It follows that the solution at $x = 1$ is

$$\frac{\beta_0 + \beta_1 + \cdots + \beta_k}{\alpha_1 + 2\alpha_2 + \cdots + k\alpha_k}$$

which is independent of n. The fact that the method is convergent means that we also have $y(1) = 1$ and hence we conclude that

$$\beta_0 + \beta_1 + \cdots + \beta_k = \alpha_1 + 2\alpha_2 + \cdots + k\alpha_k$$

and hence the method is consistent. □

3.3 Order Conditions

In order to find out the conditions on $[\alpha, \beta]$ which give a method with local error, $y(x_n) - y_n$ of $O\left(h^{p+1}\right)$,

$$y\left(x_n\right) - \sum_{i=1}^{k} \alpha_i y\left(x_{n-i}\right) - h\sum_{i=0}^{k} \beta_i y'\left(x_{n-i}\right) = O\left(h^{p+1}\right) \tag{3.9}$$

must be satisfied provided that $y_{n-k} = y\left(x_{n-k}\right) + O\left(h^{p+1}\right)$ for the k previous steps. Assume that the problem being solved, $y' = f(y)$, is C^{p+1}, that is the derivatives up to $p+1$ are continuous. Then the equation (3.9) can be expanded as a Taylor series about the point x_n with the general form

$$C_0 y(x_n) + C_1 h y'(x_n) + C_2 h^2 y''(x_n) + \cdots + C_p h^p y^{(p)}(x_n) + C_{p+1} h^{p+1} y^{(p+1)}(x_n) + \cdots .$$

Hence, the right hand side of equation (3.9) become

$$\begin{aligned} y(x_n) &- \sum_{i=1}^{k} \left[\alpha_i y(x_n) + (-ih)\alpha_i y'(x_n) + \frac{(-ih)^2\alpha_i}{2!} y''(x_n) + \frac{(-ih)^3\alpha_i}{3!} y'''(x_n) + \cdots\right] \\ &- \sum_{i=0}^{k} h \left[\beta_i y'(x_i) + \beta_i(-ih)y''(x_n) + \frac{\beta_i(-ih)^2}{2!} y'''(x_n) + \frac{\beta_i(-ih)^3}{3!} y^{(4)(x_n)} + \cdots\right] \end{aligned}$$

Therefore, the coefficient of the $y(x_n)$ term is $1-\sum_{i=1}^{k}\alpha_i$, and for higher order terms the coefficient C_j of $y^{(j)}$, $(j \geq 1)$, is

$$\begin{aligned} C_j h^j &= -\sum_{i=1}^{k} \frac{\alpha_i(-ih)^j}{j!} - \sum_{i=0}^{k} \frac{h\beta_i(-ih)^{j-1}}{(j-1)!} \\ \Longrightarrow\ C_j &= -\sum_{i=1}^{k} \frac{\alpha_i(-i)^j}{j!} i - \sum_{i=0}^{k} \frac{\beta_i(-i)^{j-1}}{(j-1)!}. \end{aligned} \tag{3.10}$$

Also, look at the Taylor expansion of $\alpha(\exp(-z)) - z\beta\left(\exp(-z)\right)$,

$$1 - \sum_{i=1}^{k} \alpha_i \sum_{j=0}^{\infty} \frac{(-i)^j}{j!} z^j - z \sum_{i=0}^{k} \beta_i \sum_{i=0}^{\infty} \frac{(-i)^j}{j!} z^j$$

where the first term of the expression is $1-\sum_{i=1}^{k}\alpha_i$ and that the coefficient of the term z^j for $j \geq 1$ is given by

$$-\sum_{i=1}^{k} \frac{\alpha_i(-i)^j}{j!} - \sum_{i=0}^{k} \frac{\beta_i(-i)^{j-1}}{(j-1)!}. \tag{3.11}$$

Since equations (3.10) and (3.11) are identical it is the case that equation (3.9) must hold when

$$\alpha(\exp(-z)) - z\beta\left(\exp(-z)\right) = O\left(|z|^{p+1}\right). \tag{3.12}$$

This leads to the following theorem.

Theorem 3.6. Butcher [7] *A multistep method* $[\alpha, \beta]$ *has at least order* p *if and only if*

$$\alpha(\exp(z)) + z\beta\left(\exp(z)\right) = O\left(|z|^{p+1}\right).$$

Substitute

$$\exp(z) = (1+z)^{-1} \implies z = -\ln(1+z)$$

into the equation (3.12) then it becomes

$$\alpha\left((1+z)^{-1}\right) - \ln(1+z)\beta\left((1+z)^{-1}\right) = O\left(|z|^{p+1}\right), \tag{3.13}$$

where $z \in \mathbb{C} : |z| < 1$ so $\ln(1+z)$ can be defined using the power series $z - \frac{z^2}{2} + \frac{z^3}{3} - \frac{z^4}{4} + \cdots$. Note that since $\alpha(1) = 0$, both $\alpha(1+z)$ and $\ln(1+z)$ vanish when $z = 0$. Rearrange equation (3.13) as follows.

$$\begin{aligned}
\alpha\left((1+z)^{-1}\right) - z\left(1 - \frac{z}{2} + \frac{z^2}{3} - \cdots\right)\beta\left((1+z)^{-1}\right) &= O\left(|z|^{p+1}\right) \\
\frac{\alpha\left((1+z)^{-1}\right)}{-\ln\left((1+z)\right)} + \beta\left((1+z)^{-1}\right) &= \frac{O\left(|z|^{p+1}\right)}{-z\left(1 - \frac{z}{2} + \frac{z^2}{3} - \cdots\right)} \\
\frac{\alpha\left((1+z)^{-1}\right)}{\ln\left((1+z)^{-1}\right)} + \beta\left((1+z)^{-1}\right) &= O\left(|z|^{p}\right).
\end{aligned}$$

The order of the method must be asymptotically correct, that is as $h \to 0$. Since z represents the quantity hy', $z^j = \frac{h^j y^{(j)}}{j!}$ and $(1+z)^{-1} \to (1-z)$ as $h \to 0$, the following equation which holds for $|z| \ll 1$, can be obtained.

$$\frac{z}{\ln(1+z)}\frac{\alpha(1+z)}{z} + \beta\left(1+z\right) = O\left(|z|^{p}\right). \tag{3.14}$$

Equation (3.14) shows the relationship between the coefficients of the polynomials α and β such that the method $[\alpha, \beta]$ has the required order. There are several free variables in determining the values of these coefficients which should be chosen such that the resulting method has additional properties.

3.3.1 Adams-Bashforth Methods

Adams-Bashforth, AB methods, are explicit methods and hence have $\beta_0 = 0$. To calculate the coefficients in this case, define $\widehat{\beta}(z)$ as $\beta(z) = z\widehat{\beta}(z)$, hence, when $\beta(z)$ is a polynomial of degree k, $\widehat{\beta}(z)$ is of degree $k-1$. This gives the following form for equation (3.14).

$$\frac{z}{(1+z)\ln(1+z)}\frac{\alpha(1+z)}{z} + \widehat{\beta}(1+z) = O\left(z^{p}\right).$$

Put $\alpha(z) = 1 - z \implies \alpha(1+z) = -z$, and obtain,

$$\widehat{\beta}(1+z) = \frac{z}{(1+z)\ln(1+z)} + O(z^p) = \frac{1}{(1+z)\left(1 - \frac{1}{2}z + \frac{1}{3}z^2 - \frac{1}{4}z^3 + \cdots\right)} + O(z^p)$$

for which the Taylor expansion about $z = 0$ is given

$$\widehat{\beta}(1+z) = 1 - \frac{1}{2}z + \frac{5}{12}z^2 - \frac{3}{8}z^3 + \frac{251}{720}z^4 - \cdots . \tag{3.15}$$

Substitue $z = (z-1)$ to get a method of order k with k stages and truncate equation (3.15) at the term z^{k-1}. Thus deduce the following expression where the the values of $\beta_1, \beta_2, \ldots, \beta_k$ are given by the coefficients of $z^0, z^1, z^2, \ldots, z^{k-1}$.

$$\widehat{\beta}(z) = 1 - \frac{1}{2}(z-1) + \frac{5}{12}(z-1)^2 - \frac{3}{8}(z-1)^3 + \frac{251}{720}(z-1)^4 - \cdots .$$

For example, the third order AB method has $\beta_1 = \frac{23}{12}$, $\beta_2 = -\frac{4}{3}$, and $\beta_3 = \frac{5}{12}$.

3.3.2 Adams-Moulton Methods

Adams-Moulton, AM methods, have $\alpha(z) = 1 - z$. Use this with equation (3.14) to obtain

$$\frac{1-(1+z)}{z\left(1 - \frac{z}{2} + \frac{z^2}{3} - \frac{z^3}{4} + \cdots\right)} + \beta_0 + \beta_1(1+z) + \beta_2(1+z)^2 + \beta_3(1+z)^3 + \cdots = O\left(z^p\right)$$

$$\implies \beta_0 + \beta_1(1+z) + \beta_2(1+z)^2 + \beta_3(1+z)^3 + \cdots = \frac{1}{\left(1 - \frac{z}{2} + \frac{z^2}{3} - \frac{z^3}{4} + \cdots\right)} + O(z^p)$$

Using the Taylor expansion of the right hand side about $z = 0$ gives

$$\beta(1+z) = 1 + \frac{1}{2}z - \frac{1}{12}z^2 + \frac{1}{24}z^3 - \frac{19}{720}z^4 + \cdots .$$

This can be rewritten by substituting $z = z - 1$.

$$\beta(z) = 1 + \frac{1}{2}(z-1) - \frac{1}{12}(z-1)^2 + \frac{1}{24}(z-1)^3 - \frac{19}{720}(z-1)^4 + \cdots$$

Expanding this expression and truncating after the term z^k will give the β coefficients for a method with an error of $O(z^{k+1})$. For example, the fourth order case with $k = 3$:

$$\begin{aligned} \beta(z) &= 1 + \frac{1}{2}(z-1) - \frac{1}{12}(z-1)^2 + \frac{1}{24}(z-1)^3 \\ \beta(z) &= \frac{3}{8} + \frac{19}{24}z - \frac{5}{24}z^2 + \frac{1}{24}z^3 \end{aligned}$$

This gives $\beta_0 = \frac{3}{8}$, $\beta_1 = \frac{19}{24}$, $\beta_2 = -\frac{5}{24}$ and $\beta_3 = \frac{1}{24}$.

3.3.3 Backward-Difference Methods

Backward-difference methods, BDF methods, use several previous values of y but only one evaluation of $y'(x_n)$ per step. They therefore have $\beta = \beta_0$ so equation (3.14) becomes $\alpha(1+z) + \beta_0 \ln(1+z) = O(z^{p+1})$. The following condition can be obtained by replacing z by $z-1$ and expanding the $\ln(1+z)$ term.

$$\alpha(z) = -\beta_0 \left[(z-1) - \frac{(z-1)^2}{2} + \frac{(z-1)^3}{3} - \cdots \right] + O(z^{p+1}).$$

We choose β_0 such that $\alpha(0) = 1$ as required for consistency. For example, when $p = k = 3$ we have

$$\begin{aligned} \alpha(z) &= -\beta_0 \left[(z-1) - \frac{(z-1)^2}{2} + \frac{(z-1)^3}{3} \right] \\ \alpha(z) &= -\beta_0 \left[\frac{11}{6} - 3z + \frac{3z^2}{2} - \frac{z^3}{3} \right]. \end{aligned}$$

Hence $\beta_0 = \frac{6}{11}$, $\alpha_1 = \frac{18}{11}$, $\alpha_2 = -\frac{9}{11}$ and $\alpha_3 = \frac{2}{11}$.

3.4 Errors and Accuracy

We have seen that we are able to develop a linear multistep method which is asymptotically correct up to an order of our choice. It is also possible to estimate the size of the asymptotically correct truncation error that occurs in a single step of a particular method. Knowing the estimated error in a single step allows us to either correct for that error or to use the error as a criterion for when we should change the stepsize or the order of a scheme. To confirm that the error, and hence the order, of a method is what we would expect we can employ numerical experiments where we can study the behaviour of a method with respect to a particular problem.

3.4.1 Truncation Error

Linear multistep methods seek to approximate the solution to a differential equation by its Taylor series which is constructed from estimates of previous function values and derivative values. This means that as each step depends on only a finite number of previous function evaluations we have an error due to the truncation of the approximating Taylor series. If we compare the Taylor series for the terms that make up our linear multistep method with the Taylor series for the solution at the point of interest, we are able to calculate the difference between the two expressions and hence to estimate the truncation error of the method. As an example we will calculate the truncation error of the second order AB method of order 2. This is given by $y_n = y_{n-1} + h\left(\frac{3}{2}y'_{n-1} - \frac{1}{2}y'_{n-2}\right)$ hence the error in the method is

$$y(h) - y(0) - \frac{3}{2}hy'(0) + \frac{h}{2}y'(-h). \tag{3.16}$$

We can then write down the Taylor series about zero for each term in equation (3.16) as follows.

$$\begin{aligned} y(h) &= y(0) + hy'(0) + \frac{1}{2}h^2y''(0) + \frac{1}{6}h^3y'''(0) + \cdots \\ -y(0) &= -y(0) \\ -\frac{3}{2}hy'(0) &= -\frac{3}{2}hy'(0) \\ \frac{1}{2}hy'(-h) &= \frac{1}{2}hy'(0) - \frac{1}{2}h^2y''(0) + \frac{1}{4}h^3y'''(0) + \cdots . \end{aligned}$$

This gives *Table 3.1.*

	$y(0)$	$hy'(0)$	$h^2y''(0)$	$h^3y'''(0)$
$y(h)$	1	1	$\frac{1}{2}$	$\frac{1}{6}$
$-y(0)$	-1	0	0	0
$-\frac{3}{2}hy'(0)$	0	$-\frac{3}{2}$	0	0
$\frac{1}{2}hy'(-h)$	0	$\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{4}$
	0	0	0	$\frac{5}{12}$

Table 3.1: Error coefficients of the order 2 Adams-Bashforth method

We are able to see that the error in the solution for y after one step is $\frac{5}{12}h^3y'''(x_{n-1})$. In this way it is possible to calculate an error term for any linear multistep method. Tables of the error coefficients corresponding to AB and AM methods of different orders are in [7].

3.4.2 Error Estimation and Correction

	$y(0)$	$hy'(0)$	$h^2y''(0)$	$h^3y'''(0)$
$y(h)$	1	1	$\frac{1}{2}$	$\frac{1}{6}$
$-y(0)$	-1	0	0	0
$-\frac{1}{2}hy'(h)$	0	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{4}$
$-\frac{1}{2}hy'(0)$	0	$-\frac{1}{2}$	0	0
	0	0	0	$-\frac{1}{12}$

Table 3.2: Error coefficients of the order 2 Adams-Moulton method

As a comparison to the AB method we will also look at the truncation error of the second order AM method. This has the form $y_n = y_{n-1} + \frac{h}{2}\left(y'_n + y'_{n-1}\right)$ and hence the Taylor expansion for each of its terms gives *Table 3.2*

Since we now know the truncation error for the second order AM method is $-\frac{1}{12}h^3y'''(x_{n-1})$ it is possible to estimate the actual error in the value of y_n. If y_n is evaluated using both the second order AB and AM with the same step size for each then we can estimate the value of $y'''(x_{n-1})$ from the difference between the results of the two methods. This technique is known as the 'Milne Device' [32]. The estimate of the truncation error can either be used to correct the calculated value of y_n or as the error estimate for a stepsize control scheme.

3.5 Stability

Consider the standard test problem $y' = \lambda y$ where λ is a constant parameter which need not be real. To be able to solve this linear test problem it is necessary that

$$y_n = \sum_{i=1}^{k} \alpha_i y_{n-i} + qh \sum_{i=0}^{k} \beta_i y_{n-i} \tag{3.17}$$

be bounded as $n \to \infty$. If we make the substitution $z = h\lambda$ then we can rearrange equation (3.17) to get the difference equation associated with the linear test problem.

$$(1 - z\beta_0)y_n - (\alpha_1 + z\beta_1)y_{n-1} - \cdots - (\alpha_k + z\beta_k)y_{n-k} = 0. \tag{3.18}$$

We therefore want to find the values of z such that solutions are bounded as $n \to \infty$, or we can require the slightly stronger condition, that the solutions converge as $n \to \infty$. These conditions are equivalent to requiring that the characteristic polynomial for the difference equation has all its roots in the open unit disc.

We can see that equation (3.18) has a characteristic equation given by:

$$(1 - z\beta_0)w^k - (\alpha_1 + z\beta_1)w^{k-1} - \cdots - (\alpha_k + z\beta_k) = 0. \tag{3.19}$$

We then rearrange equation (3.19) to get

$$z = \frac{w^k - \alpha_1 w^{k-1} - \alpha_2 w^{k-2} - \cdots - \alpha_k}{\beta_0 w^k + \beta_1 w^{k-1} + \beta_2 w^{k-2} + \cdots + \beta_k}. \tag{3.20}$$

For w on the boundary of the unit disc, w^{-1} also lies on the boundary of the unit disc so for such w we may re-write equation(3.20) as

$$z = \frac{\alpha(w)}{\beta(w)}. \tag{3.21}$$

We are interested in the values of z that satisfy equation (3.21) such that the corresponding values for w lies in the open unit disc. If we take $w = \exp(i\theta)$, $\theta \in [0, 2\pi]$ then equation (3.21) maps the boundary of the unit circle to a closed curve in the complex plane, known as the 'Boundary Locus Curve'. The region of z values that will give bounded solutions as $n \to \infty$ is that portion of the region enclosed by the boundary locus curve that lies to the left of the curve when beginning from the origin.

For example, the method AB with order 2 is given by

$$y_n = y_{n-1} + h\left(\frac{3}{2}y'_{n-1} - \frac{1}{2}y'_{n-2}\right)$$

with the characteristic equation

$$w^2 - w - \frac{3z}{2}w + \frac{z}{2}.$$

After substituting $w = \exp(i\theta)$ we have

$$z = \frac{\exp(2i\theta) - \exp(i\theta)}{\frac{3}{2}\exp(i\theta) - \frac{1}{2}} = \frac{2\exp(i\theta)\,(\exp(i\theta) - 1)}{3\exp(i\theta) - 1}.$$

The stability regions for Adams-Bashforth, Adams-Moluton and BDF methods are shown in *Figure 3.1*, *Figure 3.2* and *Figure 3.3*. The unshaded region of the complex plane corresponds to the region to the left of the boundary as the value of w moves around the unit circle from 0 to 2π, and represents the values of z which will give convergent solutions. By applying the same treatment to higher order methods it is possible to plot their stability regions too. In these we can see that the size of the stability regions of the methods decreases as order increases and that the implicit AM methods have larger stability regions than the explicit AB methods of comparable order. It is worth noting that for the fourth order AB method, the loops attached to the stability region are not part of the stability region.

The AM method of order two with $k = 1$ is given by

$$y_n = y_{n-1} + h(\frac{1}{2}y'_n + \frac{1}{2}y'_{n-1}).$$

The characteristic polynomial for this method is therefore $w - 1 - \frac{z}{2}(w + 1) = 0$ which gives $z = \frac{2(w-1)}{w+1}$. Putting $w = x + iy$ we get

$$z = \frac{2(x^2 - 1 + 2iy + y^2)}{x^2 + 2x + 1 + y^2}$$

Since we are interested in the values of z for w on the unit circle, $x^2 + y^2 = 1$. Hence, after changing to polar coordinates we have

$$z = \frac{2i\sin\theta}{\cos(\theta) + 1},$$

so $z \to \pm\infty$ as $\theta \to \pm\pi$. Hence the boundary of the stability region for the method order 2 AM is given by the imaginary axis and the method is therefore A-stable. However, we can notice from *Figure 3.2* that the stability region decreases as the order increases.

The backward difference methods have the general form

$$y_n = \sum_{i=1}^{k} \alpha_i y_{n-i} + h\beta_0 y'_n,$$

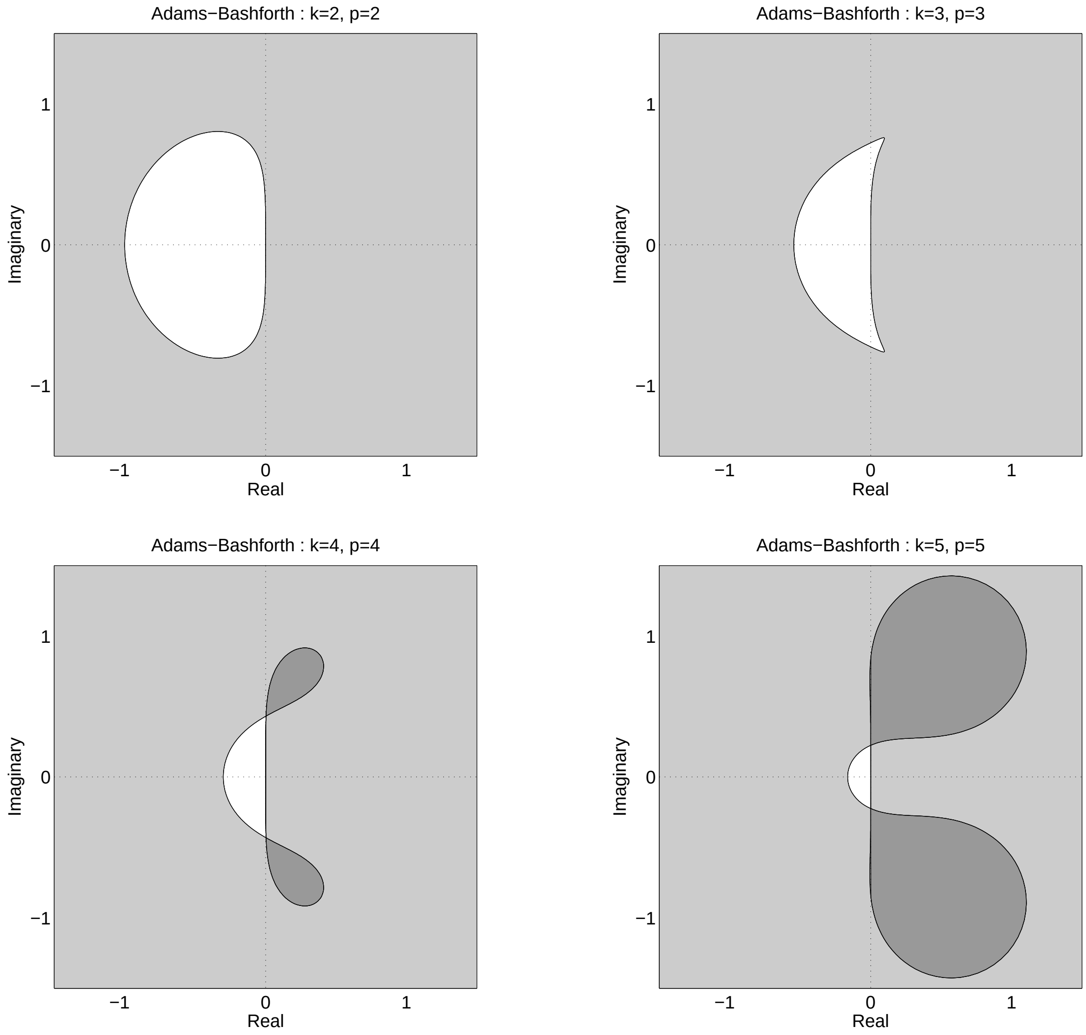


Figure 3.1: Stability Regions for Adams-Bashforth methods

hence the boundary of the stability region for such methods is given by

$$z = \frac{w^k - \sum_{i=1}^{k} \alpha_i w^{k-i}}{\beta_0 w^k}.$$

For the second order backward difference method with $\alpha_1 = \frac{4}{3}$, $\alpha_2 = -\frac{1}{3}$, $\beta_0 = \frac{2}{3}$ the boundary locus curve is given by

$$z = \frac{3w^2 - 4w + 1}{2w^2}.$$

The stability region for this method is the unbounded region outside the shaded area shown in *Figure 3.3*. If we make the substitution $w = \cos(\theta) + i\sin(\theta)$ it is

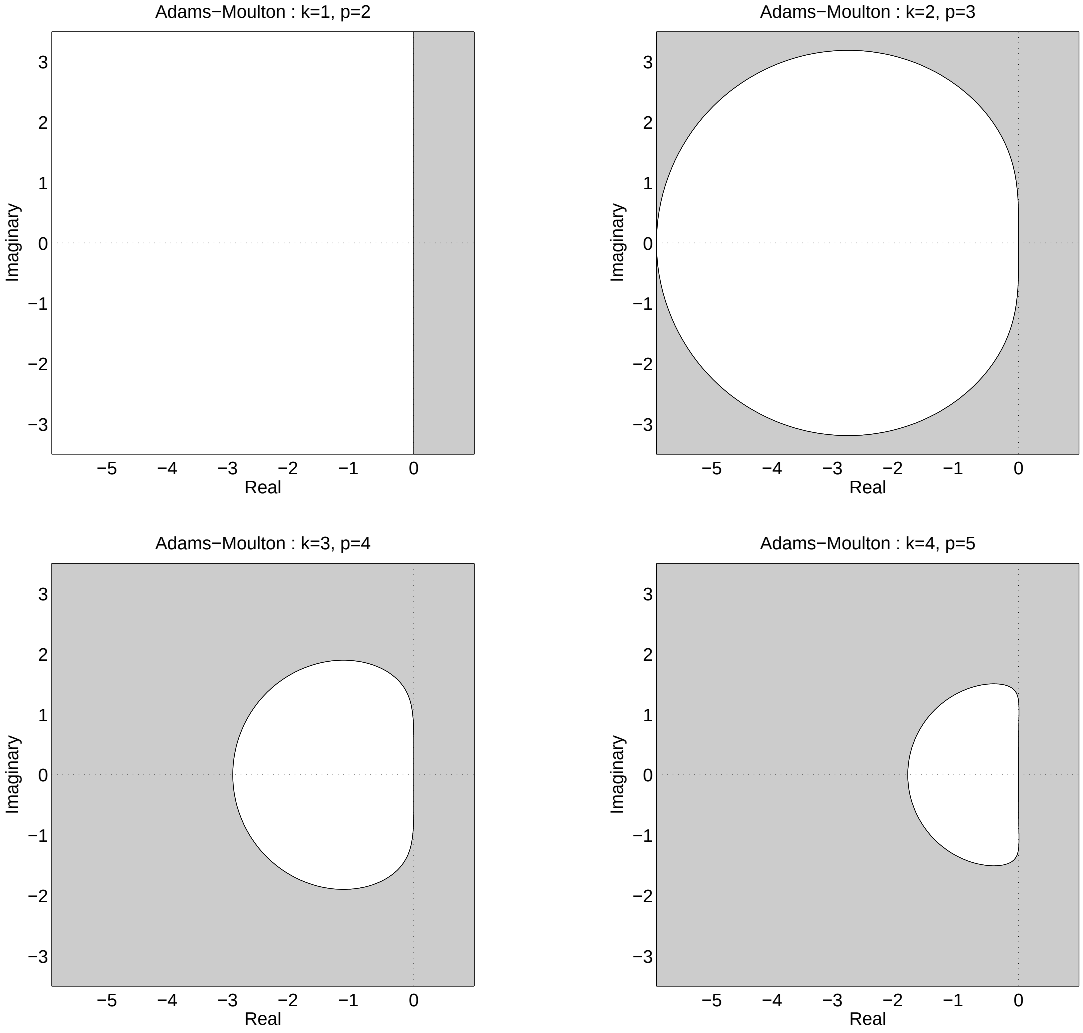


Figure 3.2: Stability Regions for Adams-Moulton methods

possible to show analytically that $Re(z) = \cos^2(\theta) - 2\cos(\theta) + 1 \geq 0$ for all θ when w is on the unit circle, hence we can see that this method is A-stable.

The third order backward difference method is $y_n = \frac{18}{11}y_{n-1} - \frac{9}{11}y_{n-2} + \frac{2}{11}y_{n-3} + h\frac{6}{11}y'_n$ and has a boundary locus given by

$$z = \frac{11w^3 - 18w^2 + 9w - 2}{6w^3}$$

The stability region corresponding to this method is given in *Figure 3.3* where we can see that although the method has a large unbounded stability region it does not cover the entire left half-plane and hence the method is not A-stable. This method

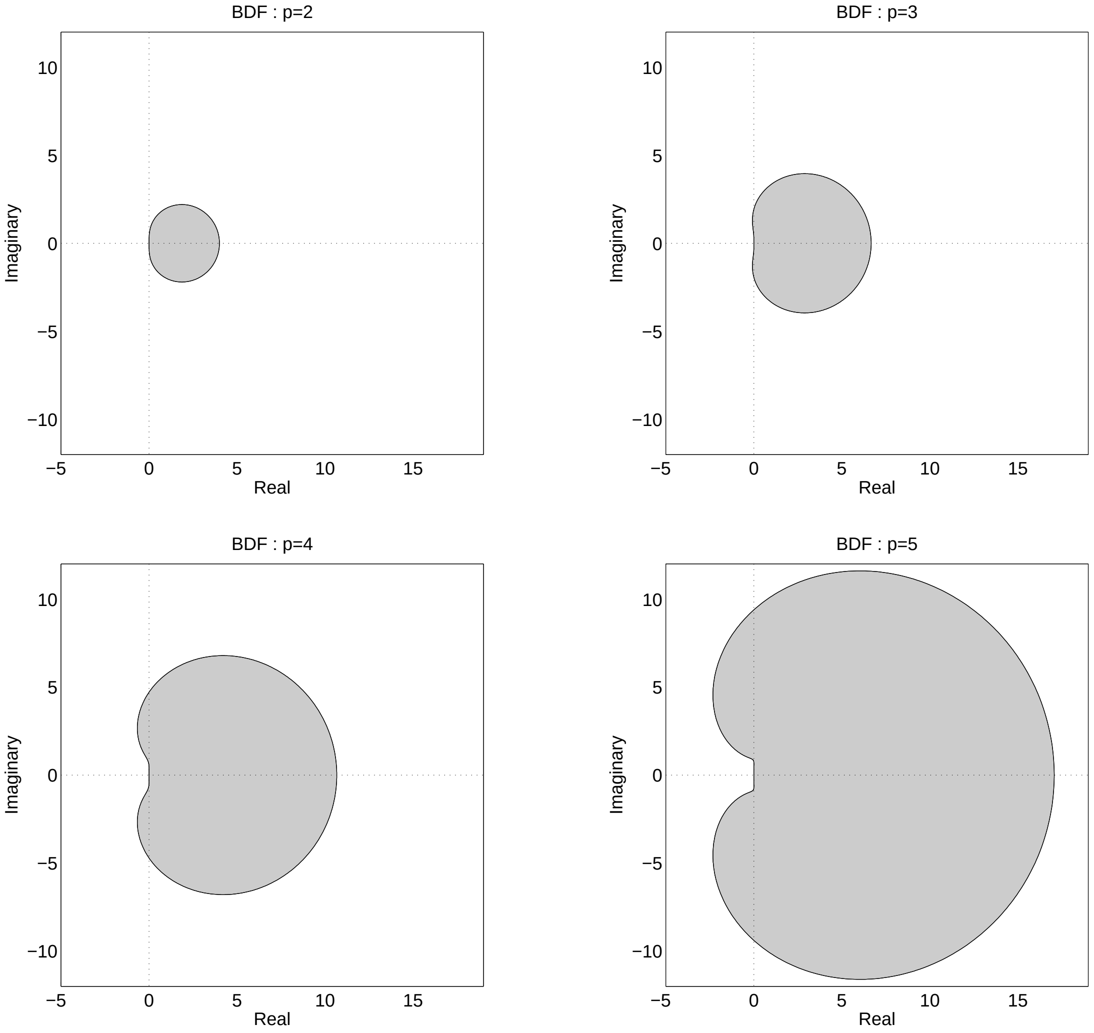


Figure 3.3: Stability Regions for BDF methods

does however have a property close to A-stability. A method is said to be A(α) stable if there exists $\alpha \in [0, \frac{\pi}{2})$ such that the method is stable for all $z = x + iy$, where $x < 0$ and $\tan(\theta) = \frac{|y|}{|x|} \leq \tan(\alpha)$. For practical purposes it may be desirable to use a method that has A(α)-stability for α close to $\frac{\pi}{2}$ so as to make available methods of higher order. A(α)-stability is also desirable for problems which we know to have eigenvalues with only small imaginary parts. Suitable methods for solving such problems are those with stability regions that include as much of the negative half-plane as possible. The third order backward difference method is A(α)-

stable with $\alpha \simeq 86.0324$ degrees while the BDF methods with orders four and five have similarly shaped stability regions and are A(α)-stable with α values of 73.3518. and 51.8410 degrees respectively. We can close look at the A(α)-stability region of BDF order 3 and 4 in *Figure 3.4*.

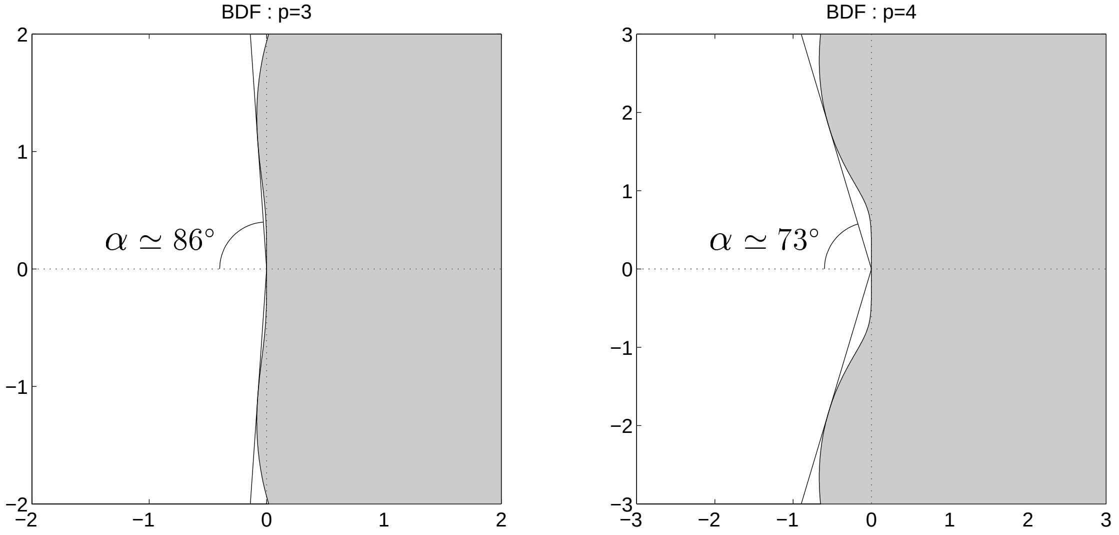


Figure 3.4: A(α)-stability regions for the order 3 and 4 BDF methods

Chapter 4

General Linear Methods

The two traditional methods, Runge-Kutta methods and linear multistep methods for ordinary differential equations have been studied separately. General linear methods were introduced to provide a unifying framework for both multistage and multivalue methods about thirty years ago. The notation was introduced in [2]. There is a hope that we can find some nicely balanced methods between accuracy and stability with low cost. Some methods showing clear advantages over the traditional methods have already been discovered.

In this chapter, a brief investigation on general linear methods is presented. This includes a formulation, consistency, stability and convergence conditions and order of accuracy. Furthermore, order and stability conditions are discussed.

4.1 Formulation

General linear methods (GLMs) are characterized by four matrices A, U, B and V which make up a partitioned $(s+r) \times (s+r)$ matrix.

$$\left[\begin{array}{c} Y_1^{[n]} \\ \vdots \\ Y_s^{[n]} \\ \hline y_1^{[n]} \\ \vdots \\ y_r^{[n]} \end{array}\right] = \left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] \left[\begin{array}{c} hf\left(Y_1^{[n]}\right) \\ \vdots \\ hf\left(Y_s^{[n]}\right) \\ \hline y_1^{[n-1]} \\ \vdots \\ y_r^{[n-1]} \end{array}\right]$$

Each notation represents the followings.

- $Y_1^{[n]}, ..., Y_s^{[n]}$ are the s internal stage values computed in the n^{th} step.
- $f\left(Y_1^{[n]}\right), ..., f\left(Y_s^{[n]}\right)$ are the s corresponding stage derivatives.
- $y_1^{[n-1]}, ..., y_r^{[n-1]}$ are the r input values for the n^{th} step.
- $y_1^{[n]}, ..., y_r^{[n]}$ are the r corresponding output approximations.

In addition, p is the order of the method and q is the stage order. General linear methods can be defined in terms of the elements of A, U, B and V by the following equations with the stepsize h.

$$\begin{aligned} Y_i^{[n]} &= \sum_{j=1}^{s} a_{ij} hf\left(Y_j^{[n]}\right) + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]}, \quad i = 1, 2, ..., s \\ y_i^{[n]} &= \sum_{j=1}^{s} b_{ij} hf\left(Y_j^{[n]}\right) + \sum_{j=1}^{r} v_{ij} y_j^{[n-1]}, \quad i = 1, 2, ..., r \end{aligned}$$

4.1.1 Examples

The order 2 Runge-Kutta method

$$\begin{aligned} y_{n-\frac{1}{2}} &= y_{n-1} + \frac{1}{2}hy'_{n-1} \\ y_n &= y_{n-1} + hy'_{n-\frac{1}{2}} \end{aligned}$$

can be rewritten as A, U, B, V formulation

$$\left[\begin{array}{c} y_{n-1} \\ y_{n-\frac{1}{2}} \\ \hline y_n \end{array}\right] = \left[\begin{array}{cc|c} 0 & 0 & 1 \\ \frac{1}{2} & 0 & 1 \\ \hline 0 & 1 & 1 \end{array}\right] \left[\begin{array}{c} hy'_{n-1} \\ hy'_{n-\frac{1}{2}} \\ \hline y_{n-1} \end{array}\right].$$

Similarly, the order 2 Adams-Bashforth method

$$y_n = y_{n-1} + \frac{3}{2}hy'_{n-1} - \frac{1}{2}hy'_{n-2}$$

can be rewritten as A, U, B, V formulation

$$\left[\begin{array}{c} y_{n-1} \\ \hline y_n \\ hy'_{n-1} \end{array}\right] = \left[\begin{array}{c|cc} 0 & 1 & 0 \\ \hline \frac{3}{2} & 1 & -\frac{1}{2} \\ 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} hy'_{n-1} \\ \hline y_{n-1} \\ hy'_{n-2} \end{array}\right].$$

This is an example of oder 3 GLM.

$$\begin{aligned} y_{n-\frac{1}{2}} &= y_{n-1} + \frac{3}{4}hy'_{n-1} - \frac{1}{4}hy'_{n-\frac{3}{2}} \\ y_n &= y_{n-1} + \frac{7}{6}hy'_{n-\frac{1}{2}} - \frac{1}{3}hy'_{n-1} + \frac{1}{6}hy'_{n-\frac{3}{2}} \end{aligned}$$

Rewrite this as A, U, B, V formulation then,

$$\left[\begin{array}{c} y_{n-1} \\ y_{n-\frac{1}{2}} \\ \hline y_n \\ hy'_{n-\frac{1}{2}} \end{array}\right] = \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ \frac{3}{4} & 0 & 1 & -\frac{1}{4} \\ \hline -\frac{1}{3} & \frac{7}{6} & 1 & \frac{1}{6} \\ 0 & 1 & 0 & 0 \end{array}\right] \left[\begin{array}{c} hy'_{n-1} \\ hy'_{n-\frac{1}{2}} \\ \hline y_{n-1} \\ hy'_{n-\frac{3}{2}} \end{array}\right]$$

4.2 Consistency, Stability and Convergence

Consistency, stability and convergence conditions for GLMs are a little more complicated than for traditional methods. However, the well-known classical result from Dahlquist [26] shown below can be generalised to apply to GLMs.

$$\text{Consistency, Stability} \quad \Longleftrightarrow \quad \text{Convergence}$$

The preconsistency and consistency conditions are determined by ensuring that the trivial one-dimensional differential equation $y'(x) = 0$ and the equation $y'(x) = 1$ can be solved exactly. If e denotes the s dimensional vector with each component equal to 1, then the preconsistency and consistency conditions for GLMs are given by

$$Vu = u, \quad Uu = e, \quad Be + Vu = u + v.$$

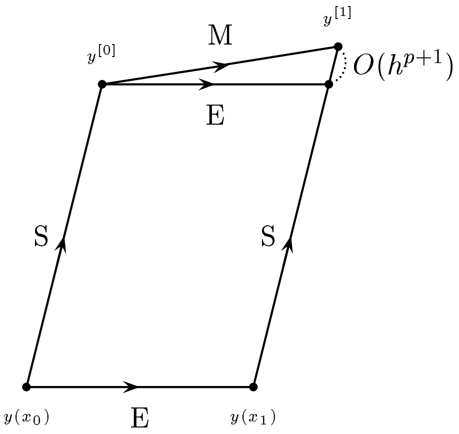
Stability is defined as a generalisation of the definition for linear multistep methods. The method is stable if the matrix V is power-bounded, that is, $\|V^n\| \leq k$ for all $n = 1, 2, ...$ where k is a constant.

Convergence is the ability of a numerical method to produce approximations which converge to the exact solution as the stepsize goes to zero. The equivalence of convergence to consistency plus stability can be established for GLM.

4.3 Order of Accuracy

Since GLMs are multivalue methods, we need some kind of starting procedure to obtain the initial input values before carrying out the first step of GLM. If a GLM has order p then each of the r components of the initial approximation is accurate to order p. Hence, a starting procedure should have order at least p to maintain this accuracy. The order of accuracy of the method, $O\left(h^{p+1}\right)$ can be defined relative to a starting procedure.

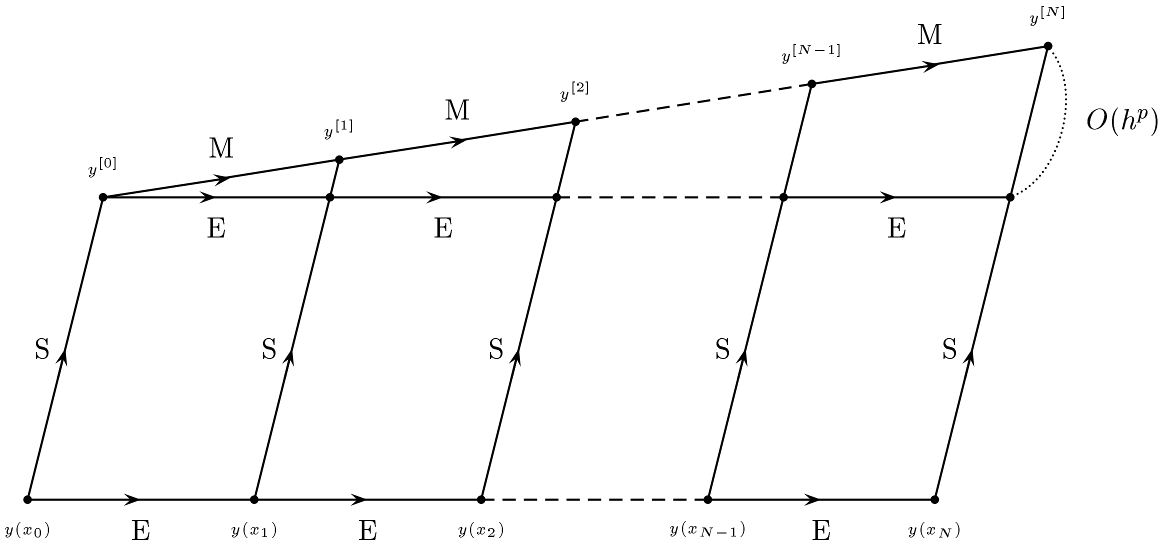
Figure 4.1: Local truncation error



We apply a starting procedure and get the initial input vector, $y^{[0]}$, then use the GLM to compute the first step and get the output, $y^{[1]}$, which we will use as the input vector for the next step. Assume that we know the exact solution at the first step. If we apply the starting procedure to this exact solution at the first step and compare the result with the computed output $y^{[1]}$ then there will be a difference of $O(h^{p+1})$. This is a local truncation error at each step as shown in *Figure 4.1.*

This concept of local truncation error guides us to an approximation of the global error shown in *Figure 4.2.* If we repeat the numerical approximation for n steps to the end then there will be a difference of $O(h^p)$ between the final approximation and the result from applying the starting procedure to the exact solution at the end point.

Figure 4.2: Global truncation error



4.4 Order Conditions

Construction and implementation of GLMs is difficult. In order to have any chance of constructing practical methods with simple implementation, two main simplifying assumptions will be required.

First, the stage order q should be equal to the order of the method p. Therefore, the stage values satisfy

$$Y^{[n]} \approx \begin{bmatrix} y(x_{n-1}+c_1h) \\ y(x_{n-1}+c_2h) \\ \vdots \\ y(x_{n-1}+c_sh) \end{bmatrix} + O\left(h^{p+1}\right)$$

Second, the quantities passed from step to step should have a simple form to overcome the difficulty of changing the stepsize. Therefore, the input and output quantities are in Nordsieck form.

$$y^{[n]} \approx \begin{bmatrix} y(x_n) \\ hy'(x_n) \\ \vdots \\ h^p y^{(p)}(x_n) \end{bmatrix} + O\left(h^{p+1}\right)$$

Recall the formula of a GLM for a single step,

$$\begin{aligned} Y^{[n]} &= Ahf(Y^{[n]}) + Uy^{[n-1]} \\ y^{[n]} &= Bhf(Y^{[n]}) + Vy^{[n-1]} \end{aligned}$$

To satisfy order conditions based on the simplifying assumptions, it is necessary and sufficient that

$$\begin{aligned} \exp(cz) &= zA\exp(cz) + UZ + O\left(h^{p+1}\right) \\ \exp(z)Z &= zB\exp(cz) + VZ + O\left(h^{p+1}\right) \end{aligned}$$

where

$$\exp(cz) = \begin{bmatrix} \exp(c_1 z) \\ \exp(c_2 z) \\ \vdots \\ \exp(c_s z) \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 \\ z \\ \vdots \\ z^p \end{bmatrix}.$$

4.5 Runge-Kutta Stability

The stability behaviour of GLMs is defined using the standard linear problem $y' = \lambda y$, where λ is a complex parameter. If the GLM is applied to this test problem, then the stability matrix $M(z)$ is defined by

$$M(z) = V + zB(I - zA)^{-1}U$$

where $z = h\lambda$.

If the characteristic polynomial of $M(z)$ has the special form below, then the method is said to have Runge-Kutta stability.

$$\det\left(wI - M(z)\right) = w^{r-1}\left(w - R(z)\right)$$

Note that $R(z) = \text{trace}\left(M(z)\right)$.

4.6 Inherent Runge-Kutta Stability

It is very hard to determine the conditions on the method in order to satisfy RK stability. However, it is possible to find the some relationships among the A, U, B and V matrices which will ensure the method has RK stability. These relationships are known as inherent Runge-Kutta stability, [7][38], which are sufficient but not necessary to ensure a GLM has RK stability.

Consider methods for which $Ve_1 = e_1$ and where V has its remaining p eigenvalues equal to 0. In this case, a GLM has inherent Runge-Kutta stability if

$$\begin{aligned} BA &= XB \\ BU &\equiv XV - VX \end{aligned}$$

for some matrix X where $\equiv$ denotes the left hand side and the right hand side are identical except for the first row.
The methods which have this inherent Runge-Kutta stability (IRKS) property will be used for the numerical experiments in next chapter.

Chapter 5

Numerical Experiments

The main purpose of this chapter is to describe some numerical experiments performed with a preliminary implementation of IRKS methods in MATLAB. For these experiments, a set of diagonally implicit methods with the IRKS property up to order 3 and three well known test problems [25], the van der Pol equation (VDPOL), the Oregonator equation (OREGO) and HIRES are used. The IRKS methods and a description of the test problems are given in the appendix.

In this chapter, the experiments on comparing the IRK methods are presented followed by a discussion of different predictors. Finally, we compare the order 3 IRKS methods with the MATLAB's built-in ODE solvers `ode15s` and `ode23s` based on BDF and Runge-Kutta methods respectively.

5.1 Comparison between IRKS Methods

In this section, performance of the IRKS methods are investigated. The integration scheme starts with the exact Nordsieck vector instead of using a starting method and involves a variable stepsize.

Figure 5.1 compares the four different order 1 methods, 's2o1a', 's2o1b', 's2o13' and 's2o1d', and *Figure 5.2* compares the five different order 2 methods, 's3o2a', 's3o2b', 's3o2', 's3o2d' and 's3o2e'. Finally, *Figure 5.3* compares the five different order 3

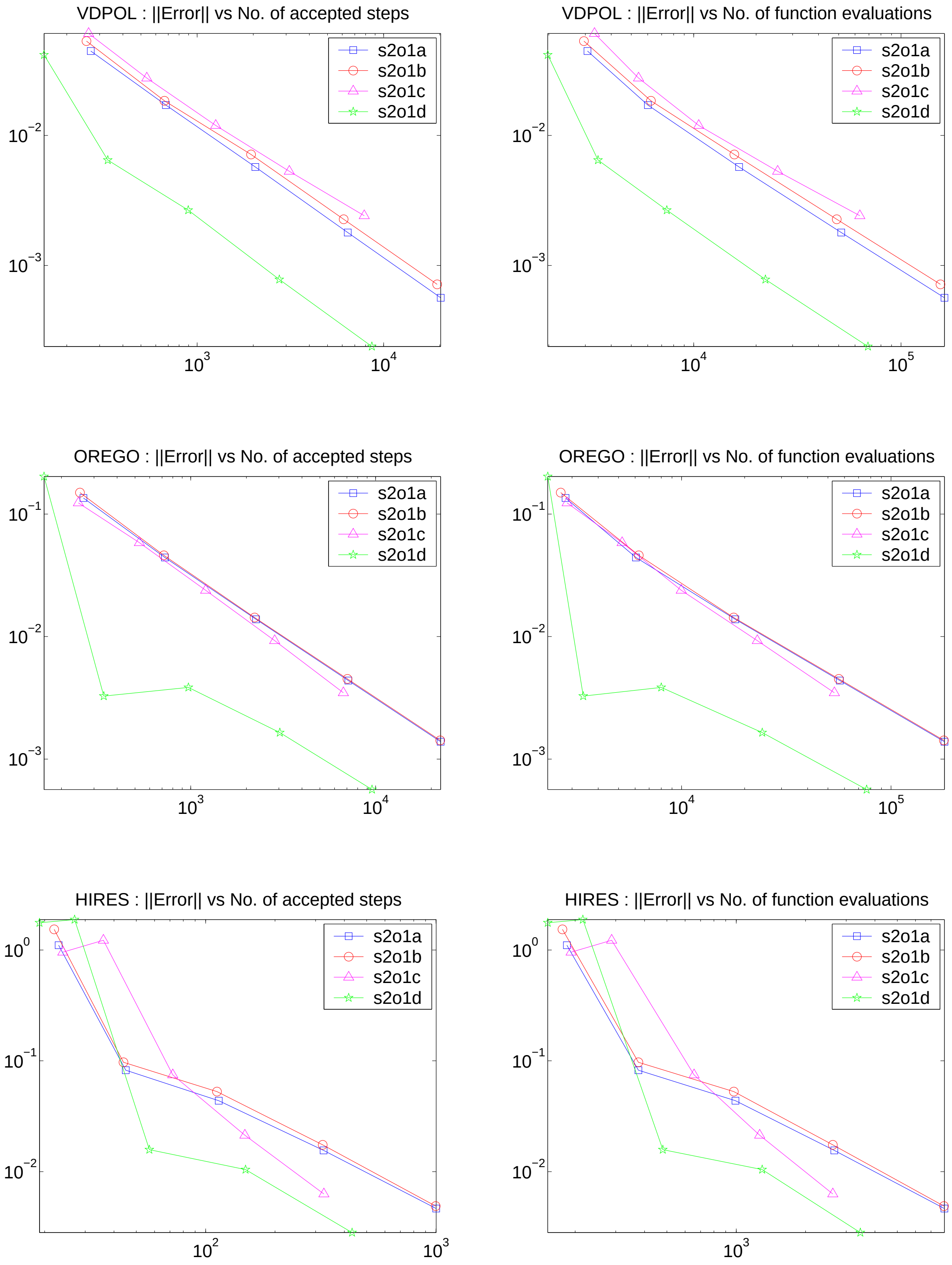


Figure 5.1: Comparing accuracy against no. of accepted steps and no. of function evaluations of order 1 IRKS methods on VDPOL, OREGO and HIRES problem.

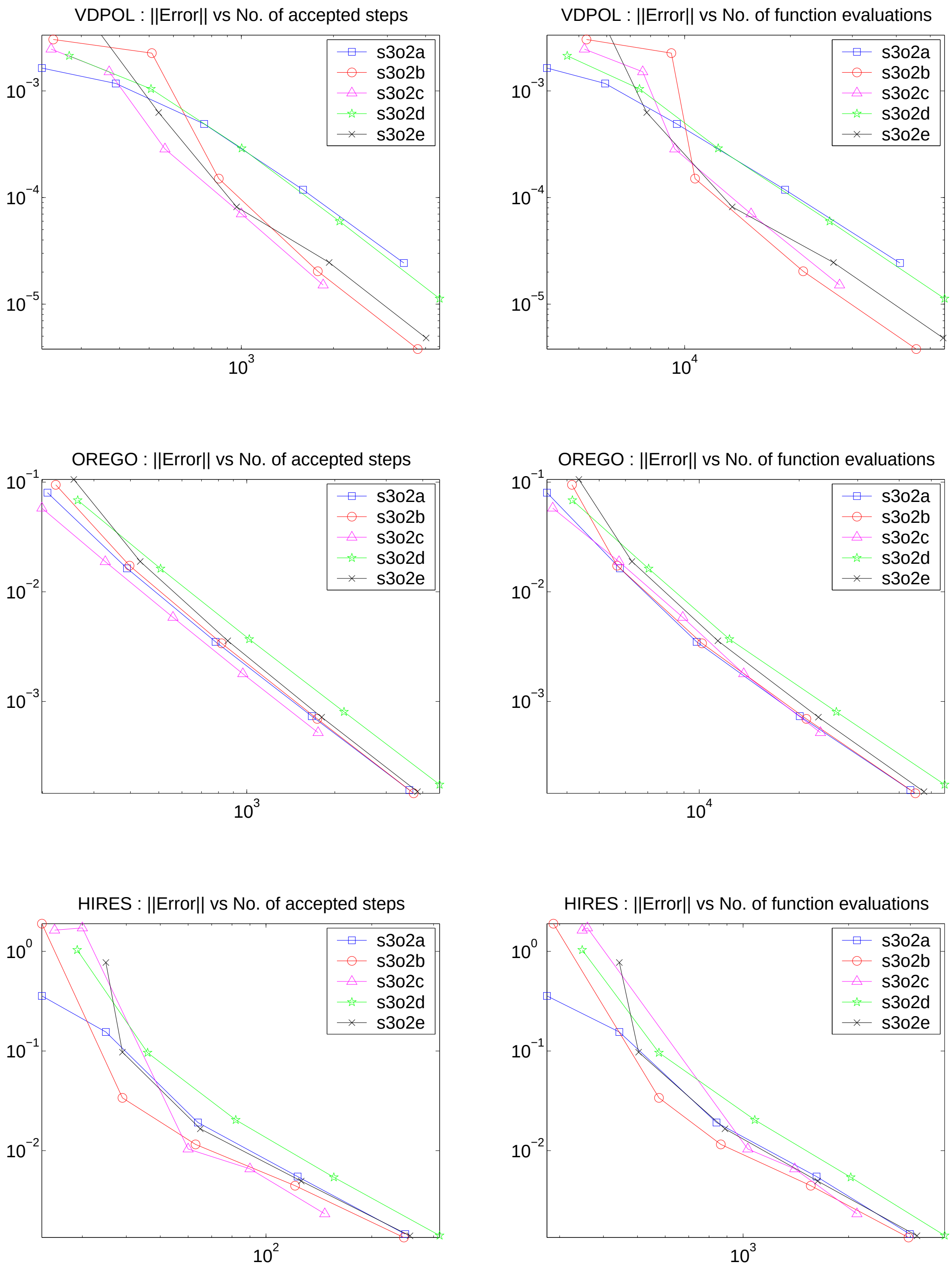


Figure 5.2: Comparing accuracy against no. of accepted steps and no. of function evaluations of order 2 IRKS methods on VDPOL, OREGO and HIRES problem.

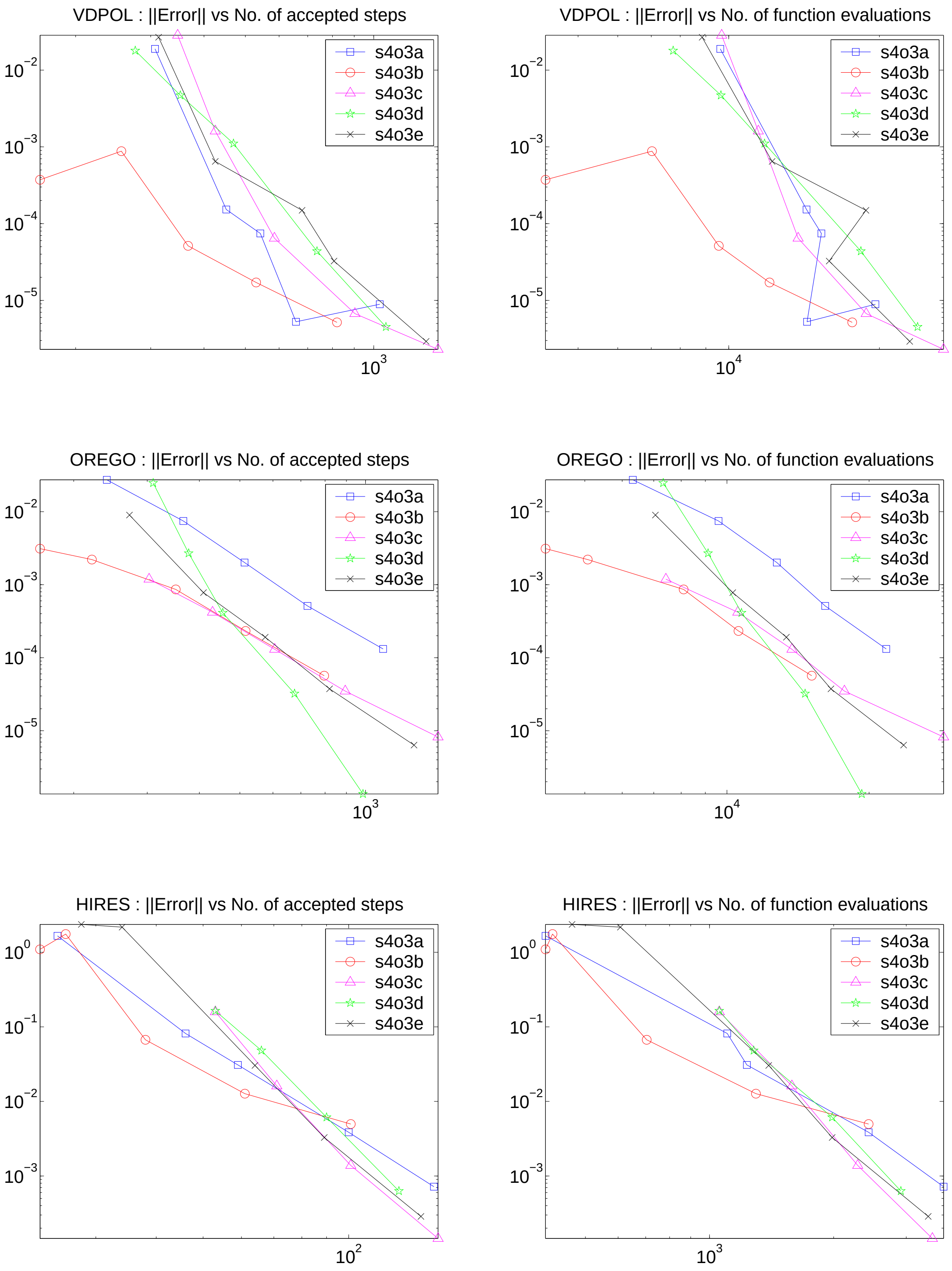


Figure 5.3: Comparing accuracy against no. of accepted steps and no. of function evaluations of order 3 IRKS methods on VDPOL, OREGO and HIRES problem.

methods, ‘s4o3a’, ‘s4o3b’, ‘s4o3’, ‘s4o3d’ and ‘s4o3e’. In each figure, there are six plots in $\log - \log$ scale. Three rows represent the three different test problems, VDPOL, OREGO and HIRES in order. The plots of the global error at the end of the integration step against the number of the accepted steps are shown in the left column and the plots of the global error against the number of function evaluations are in the right column. The global errors are calculated using the reference solutions given for the problem in [25].

Among the order 1 methods, the last method, ‘s2o1d’ performs well, although the comparison varies from problem to problem. Overall we do not observe any unexpected or critical behaviours from the results and we can see by looking at the slope that the methods are following their orders well.

5.2 Comparison between Predictors

When we implement a diagonally implicit general linear method,

$$M = \left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right]$$

with $a_{ii} = \lambda$, we have to find the solutions to the nonlinear stage equations,

$$Y_i^{[n]} = h\lambda f(Y_i^{[n]}) + w_i, \quad i = 1, \ldots, s, \tag{5.1}$$

where $w_i = \sum_{j=1}^{i-1} a_{ij} h f(Y_j^{[n]}) + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]}$ is known. It is not difficult to write down Newton’s method for (5.1) but we need to be cautious since f evaluations of perturbed values may have large errors due to the stiffness. In order to avoid this problem, it is customary to compute the derivative approximations $K_i \approx hF(\xi_i)$ of the iterates $\xi_i \approx Y_i^{[n]}$ not by evaluating the right hand side of the ordinary differential equations but by using the linear relation given by the method, that is $K_i = (\xi_i - w_i)/\lambda$. It is known that this small change can have a great effect on the numerical efficiency of the approximate process, despite the fact that either gives the same solutions if we iterate until convergence.

There are two questions connected to the numerical solution of the stages, (5.1). First, what kind of predictor represented by two matrices $\widehat{A}$ and $\widehat{U}$ with the relationship,

$$\widehat{Y}_i^{[n]} = \sum_{j=1}^{i-1} \widehat{a}_{ij} h f(Y_j) + \sum_{j=1}^{r} \widehat{u}_{ij} y_j^{[n]}, \tag{5.2}$$

shall we use, and second, how many Newton iterations shall we perform.

Before the detailed description of the specific predictors, some general remarks on the predictor $\widehat{A}$ and $\widehat{U}$ are made.

- Matrix $\widehat{A}$ must be strictly lower triangular to make the predictor explicit. But $\widehat{U}$ can be a full matrix.

- The order q of the predictor is defined similarly to the stage order, that is,

  $$\exp(cz) = z\widehat{A}\exp(cz) + \widehat{U}Z + O(z^{q+1}). \tag{5.3}$$

  If we have order $q = p$, our approximate method will always have order p, even if we perform only one Newton step. Conversely, if the predictor is of low order, we expect to need more than one Newton step to observe order p of the IRKS method.

- Provided that the predictor has full order $q = p$, $\widehat{U}$ can be obtained from (5.3) whatever we choose for $\widehat{A}$, since the equation is a linear relationship between $\widehat{A}$ and $\widehat{U}$.

Now, four different predictors used in the experiment are discussed in detail.

- The trivial predictor uses the approximation to the solution in the last output point as an initial value for the stages. It has order 0 only and corresponds to setting $\widehat{A} = 0$ and $\widehat{U}$ is zero except for the first column with ones. In our diagrams, we will denote this predictor with $\boxed{\text{P1o0}}$.

- A predictor of order 1 is obtained by using the explicit Euler method. We have again $\widehat{A} = 0$, and $\widehat{U}$ is of the form

$$\widehat{U} = \begin{bmatrix} 1 & c_1 & 0 & 0 \\ 1 & c_2 & 0 & 0 \\ 1 & c_3 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix},$$

where we use $s = 4$ as an example. We will use the symbol $\boxed{\text{P2o1}}$ to label this predictor in the figures.

- Continuing this construction, we can add more and more columns to $\widehat{U}$ until our predictor has the full order $q = p$ with

$$\widehat{U} = \begin{bmatrix} 1 & c_1 & \frac{c_1^2}{2} & \frac{c_1^3}{3!} \\ 1 & c_2 & \frac{c_2^2}{2} & \frac{c_2^3}{3!} \\ 1 & c_3 & \frac{c_3^2}{2} & \frac{c_3^3}{3!} \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

This predictor is denoted by $\boxed{\text{P3op}}$.

- To take advantage of the already computed stages Y_j, $j = 1, \ldots, i-1$, we shall use $\widehat{A} \neq 0$ in our predictor. This gives us some freedom even if we want to have full order $q = p$. Since it seems reasonable to minimize the influence from the past, we want to minimize $\widehat{U}$ in some sense. In the first row of $\widehat{U}$, we have no choice but using the Taylor series since no stages are at our disposal, but, on the other hand, for the last stage we can set $e_s^T \widehat{U} = e_1^T$ since we have $s-1$ stage values Y_j to fulfil the order condition with the last row of $\widehat{A}$ alone. To keep it simple, we propose to put zeros in the bottom right triangle of $\widehat{U}$, that is,

$$\widehat{U} = \begin{bmatrix} 1 & c_1 & \frac{c_1^2}{2!} & \frac{c_1^3}{3!} \\ 1 & * & * & 0 \\ 1 & * & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

and to solve the order condition for the stars and the elements in $\widehat{A}$. In the experiments, we denote this predictor by $\boxed{\text{P4op}}$. The values for the values '*' are computed together with $\widehat{A}$ from the linear relation (5.3), line by line. Note, that each line has the same number of unknowns, the elements of lower triangular matrix $\widehat{A}$ and the stars in $\widehat{U}$. The solution of this linear system corresponds exactly to the MATLAB code

```
case 'P4op'
      dU=zeros(s,s);
      rhs=C-dU;
      op=eye(s,s);
      for i=1:s
          ii=s+1-i;
          op(ii+1:s,:)=CK(1:i-1,:);
          u=rhs(i,:)/op;
          Uhat(i,1:ii)=u(1:ii);
          Ahat(i,1:i-1)=u(ii+1:s);
      end
```

in `predictors.m`. We can use the function to compute $\widehat{A}$ and $\widehat{U}$. The example of finding $\widehat{A}$ and $\widehat{U}$ for order 2 and 3 using the function is given.

```
[ahat,uhat]=predictors([1/3,2/3,1],'P4op')

ahat =
      0            0            0
     2/3           0            0
     1/2          1/2           0

uhat =
      1           1/3          1/18
      1            0            0
      1            0            0

[ahat,uhat]=predictors([1/4,1/2,3/4,1],'P4op')
```

```
ahat =
      0            0            0            0
     2/3           0            0            0
      0           9/16          0            0
     2/3         -1/3          2/3           0

uhat =
      1           1/4          1/32          1/384
      1          -1/6         -1/24          0
      1           3/16          0            0
      1            0            0            0
```

We now investigate how the various predictors perform by looking at how close the predictor is to the exact convergence after each Newton iteration. We present the results for one sample problem, since similar results are found for the other two test problems. Because the predictor might perform differently at different states of the solution, we have chosen two different time points, $t = 14$ and $t = 20$. At $t = 20$, the behaviour of three components of solution is more dynamic than at $t = 14$.

Figure 5.4, *Figure 5.5* and *Figure 5.6* correspond to orders 1, 2 and 3 respectively. The first column in the figure shows the convergence of predictor at $t = 14$ and the second column shows the convergence at $t = 20$. The plots on the first row represent the overall performance of the predictor, and the convergence at each stages is plotted below in increasing order. For the first row, vertical axis represents log of norm of all the stage errors and for the rest of rows, vertical axis is log of norm of idividual stage errors. The horizontal axis are the number of Newton iterations. Notice that only three predictors are compared for order 1 case since the predictor ‘P2o1’ and ‘P3op’ are the same in this case.

We can clearly observe that the higher order predictors perform better for order 1, 2 and 3. The rate of convergence is expected to be linear since the integration scheme uses modified Newton iteration. Although the convergence rate of the predictors is the same for each case, the higer order ones predict more accurately from the beginning, hence, reach the exact convergence with less iterations. We can also notice that the accuracies are much better at $t = 14$ than at $t = 20$ overall. Since the behaviour of the solutions at $t = 14$ is less dynamical, the predictors can give the better predictions. For example, in *Figure 5.6*, the predictor 'P4op' converges to even less than the machine accuracy after only 3 iterations. There certainly exists a good choice of predictor. Even if it is only a small effect on each stage and per step, a good choice of a predictor will certainly improve the performance of the integration.

5.3 Comparison with `ode15s` and `oode23s`

Finally, the five third order IRKS methods is compared with MATLAB built-in integrators, `ode15s` and `ode23s` which are fixed to be order 3 as well. *Figure 5.7* shows the results on VDPOL, *Figure 5.8* shows the results on OREGO and *Figure 5.9* shows the result on HIRES problem. As marked in the figures, the five rows represent the five different order 3 IRKS methods. The left column contains the plots of the global error versus the number of accepted steps and the right column displays the plots of the global error versus the number of function evaluations both in $\log - \log$ scale.

Overall the IRKS methods perform well for all three test problems. The acccuracy of the IRKS methods is definitely better per step. In terms of function evaluations which is directly related to the cost of implementation, the IRKS methods are not as successful as in terms of per step. However, for all three test problems, we clearly see that the IRKS methods have advantage for higher order.

In fact, it has been shown that some high order IRKS methods have advantages over the current popular integrators [37]. These results are very promissing for the IRKS methods and encourage further investigation. The IRKS methods certainly have the potential to be competitive solvers for stiff ordinary equations if further improvements are made to their implementation.

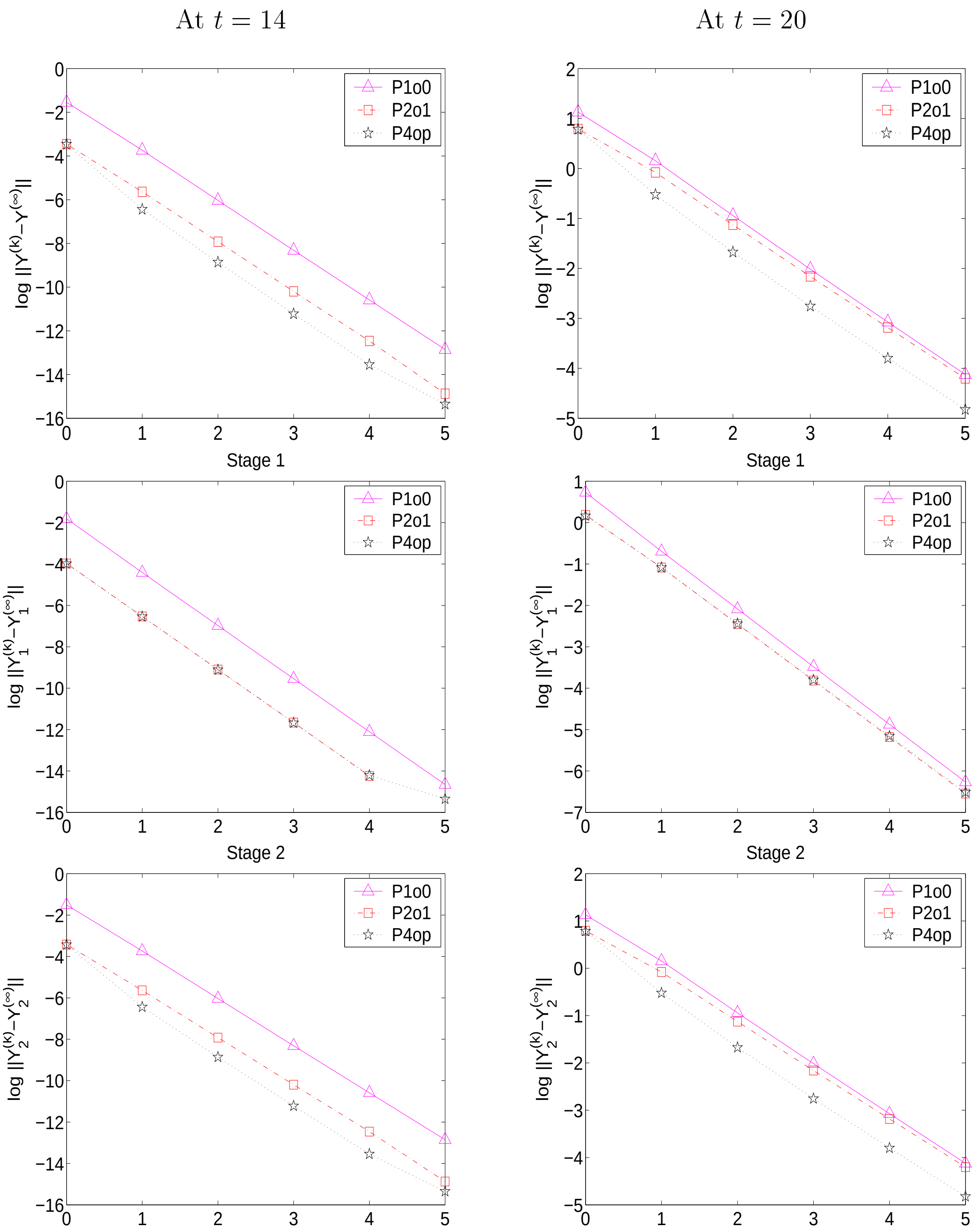


Figure 5.4: Comparing four different predictors on their convergence using the OREGO problem for the order 1 cases.

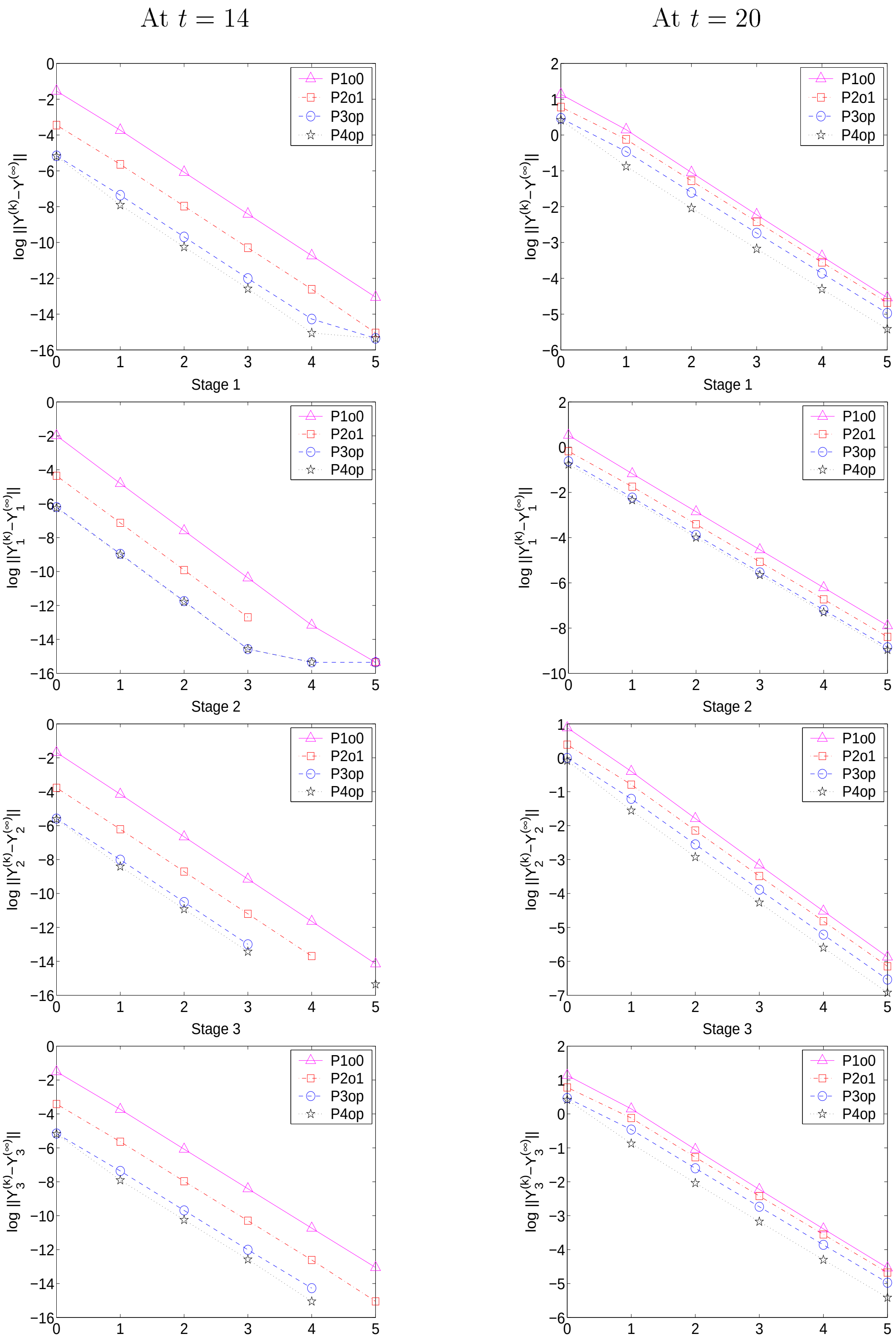


Figure 5.5: Comparing five different predictors on their convergence using the OREGO problem for the order 2 cases.

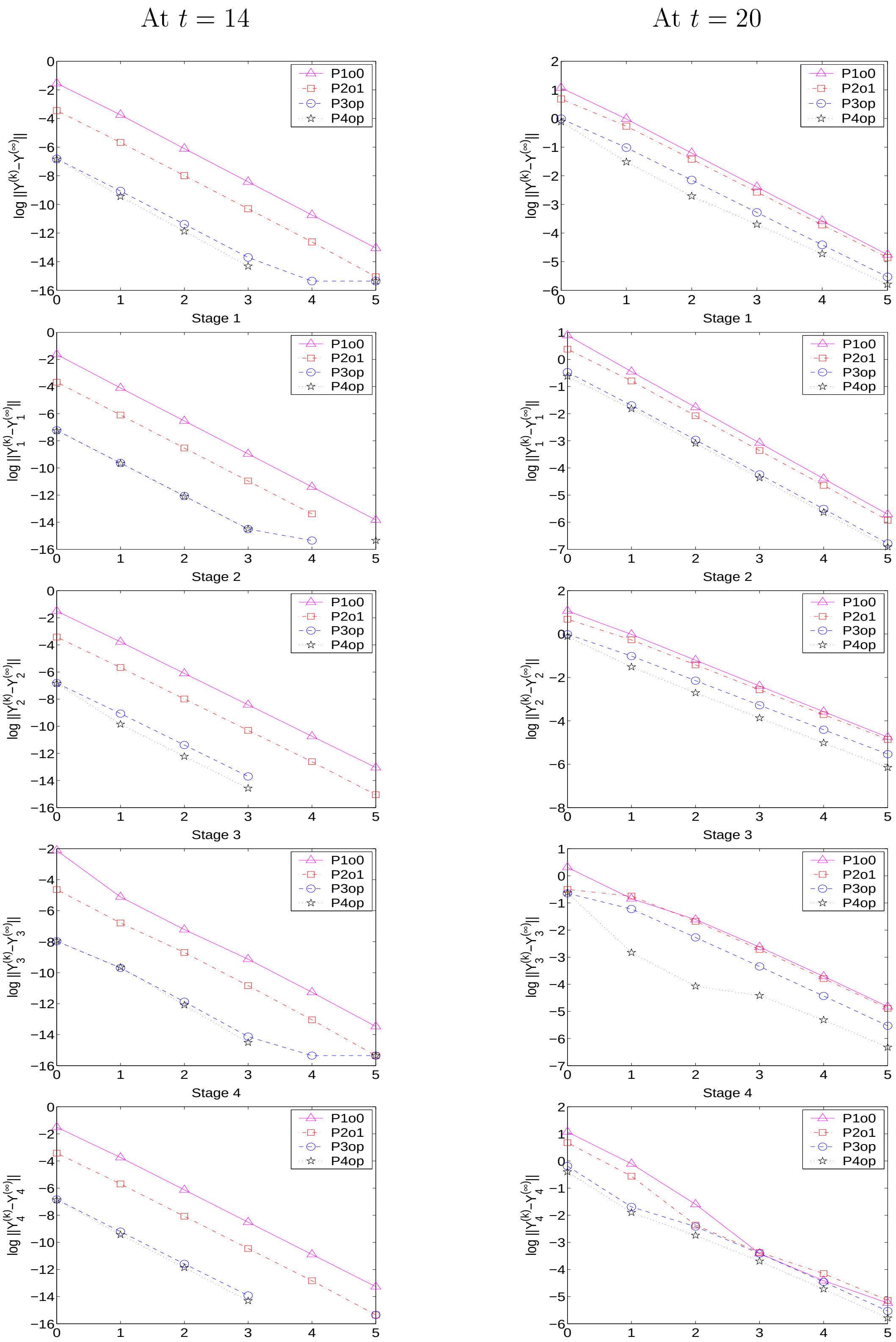


Figure 5.6: Comparing five different predictors on their convergence using the OREGO problem for the order 3 cases.

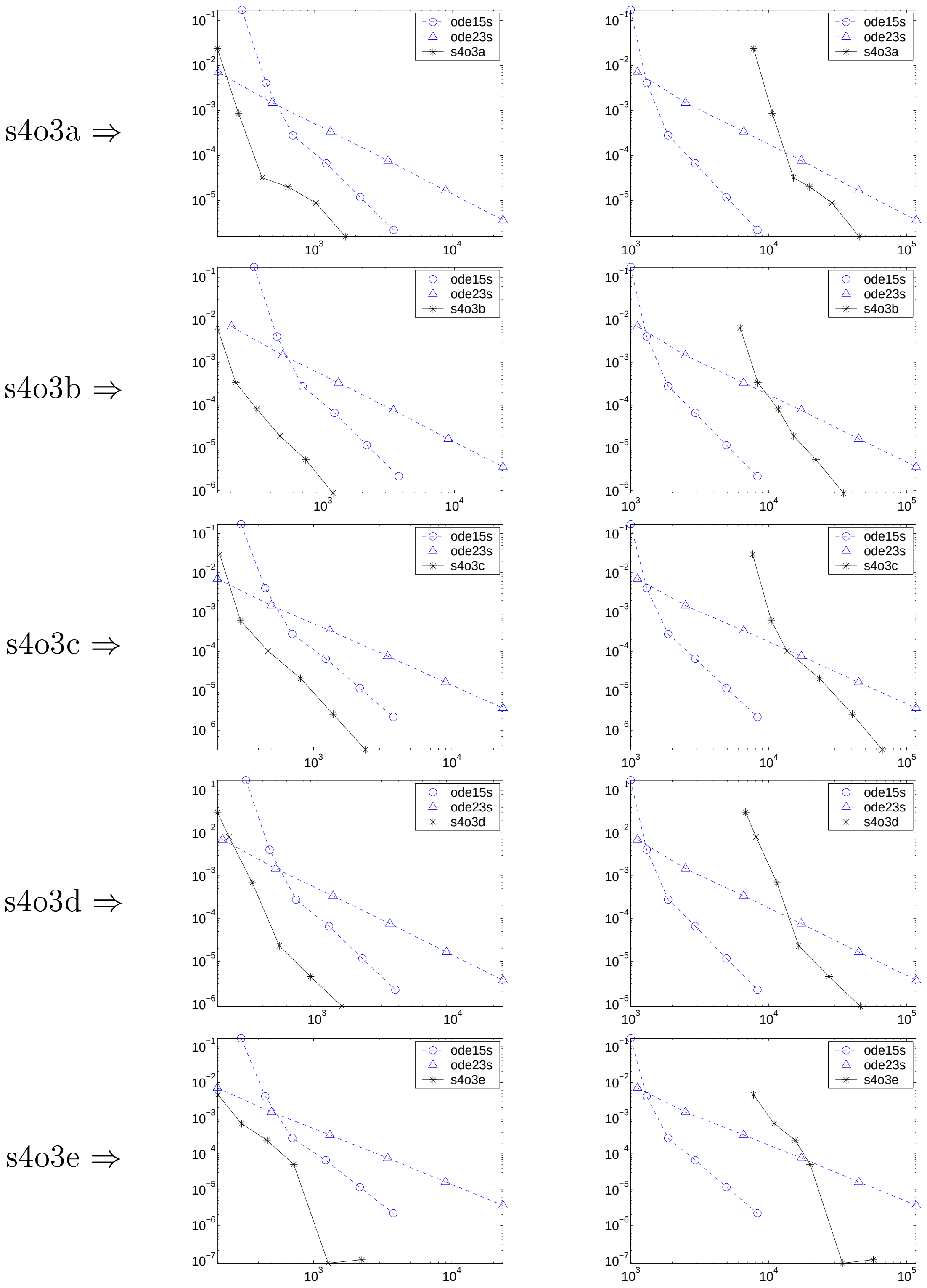


Figure 5.7: Comparing the order 3 IRKS methods with two MATLAB built-in stiff IVP solvers, `ode15s` and `ode23s` on the VDPOL problem.

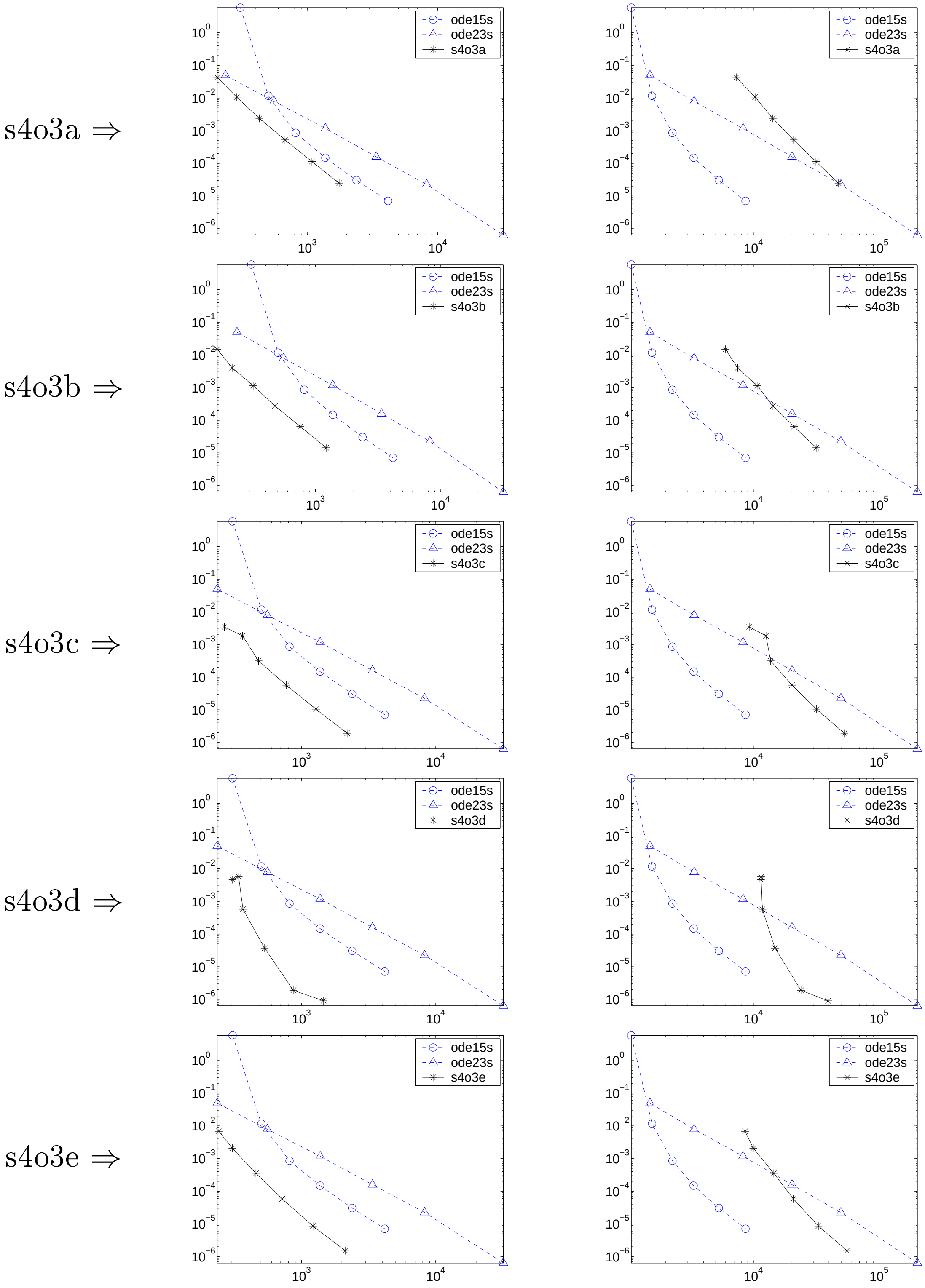


Figure 5.8: Comparing the order 3 IRKS methods with two MATLAB built-in stiff IVP solvers, `ode15s` and `ode23s` on the OREGO problem.

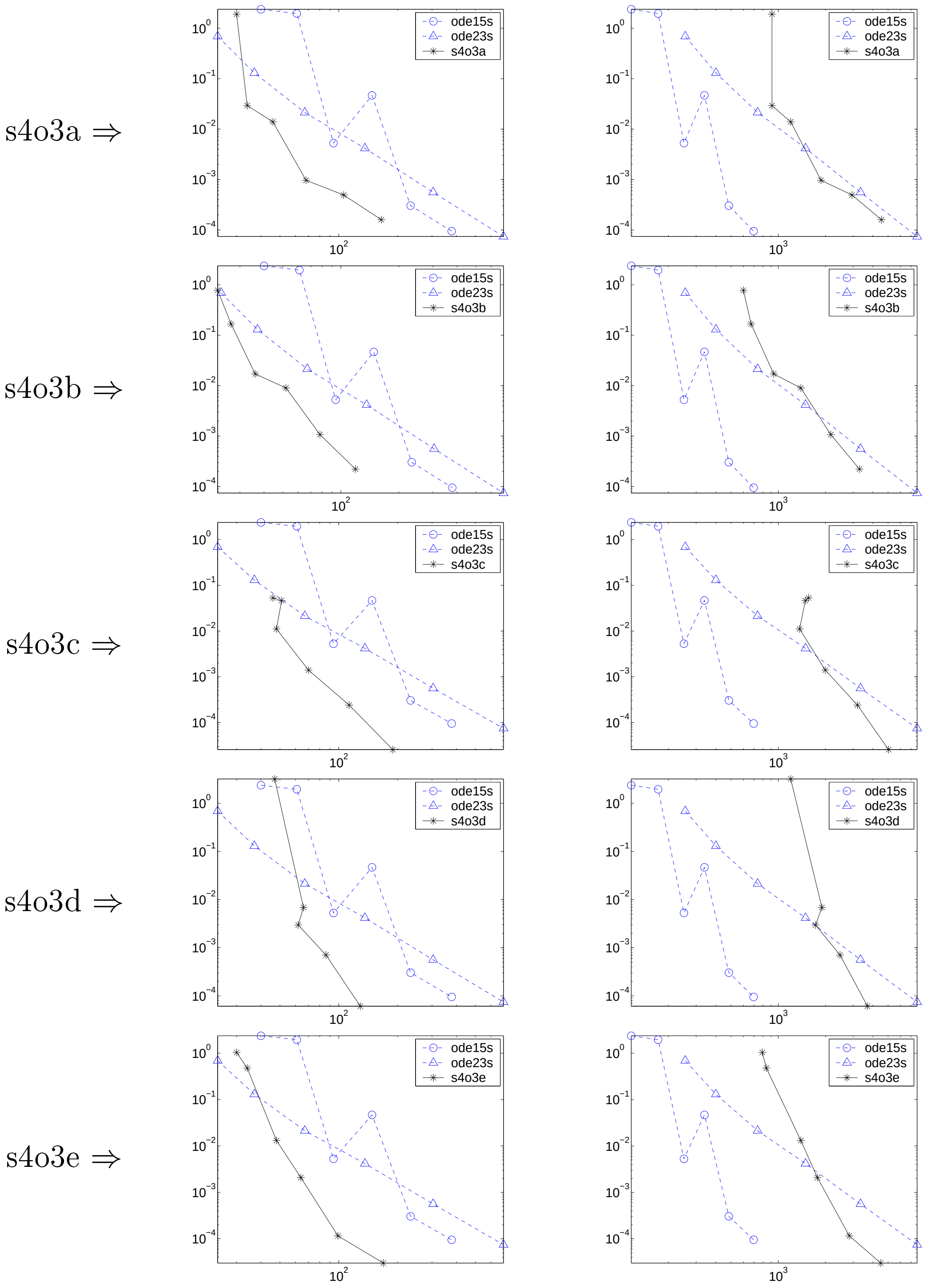


Figure 5.9: Comparing the order 3 IRKS methods with two MATLAB built-in stiff IVP solvers, `ode15s` and `ode23s` on the HIRES problem.

Appendix A

Test Problems

Three well known test problems, the van der Pol's equation (VDPOL), the Oregonator reaction (OREGO) and the High Irradiance Responses of photomorphogenesis (HIRES) are chosen as test problems.

A.1 VDPOL

The famous test problem, van der Pol's equation [25] describes the behaviour of oscillator circuits. Originally, the problem had the form of a second order differential equation,

$$y'' + \mu(y^2 - 1)y' + y = 0,$$

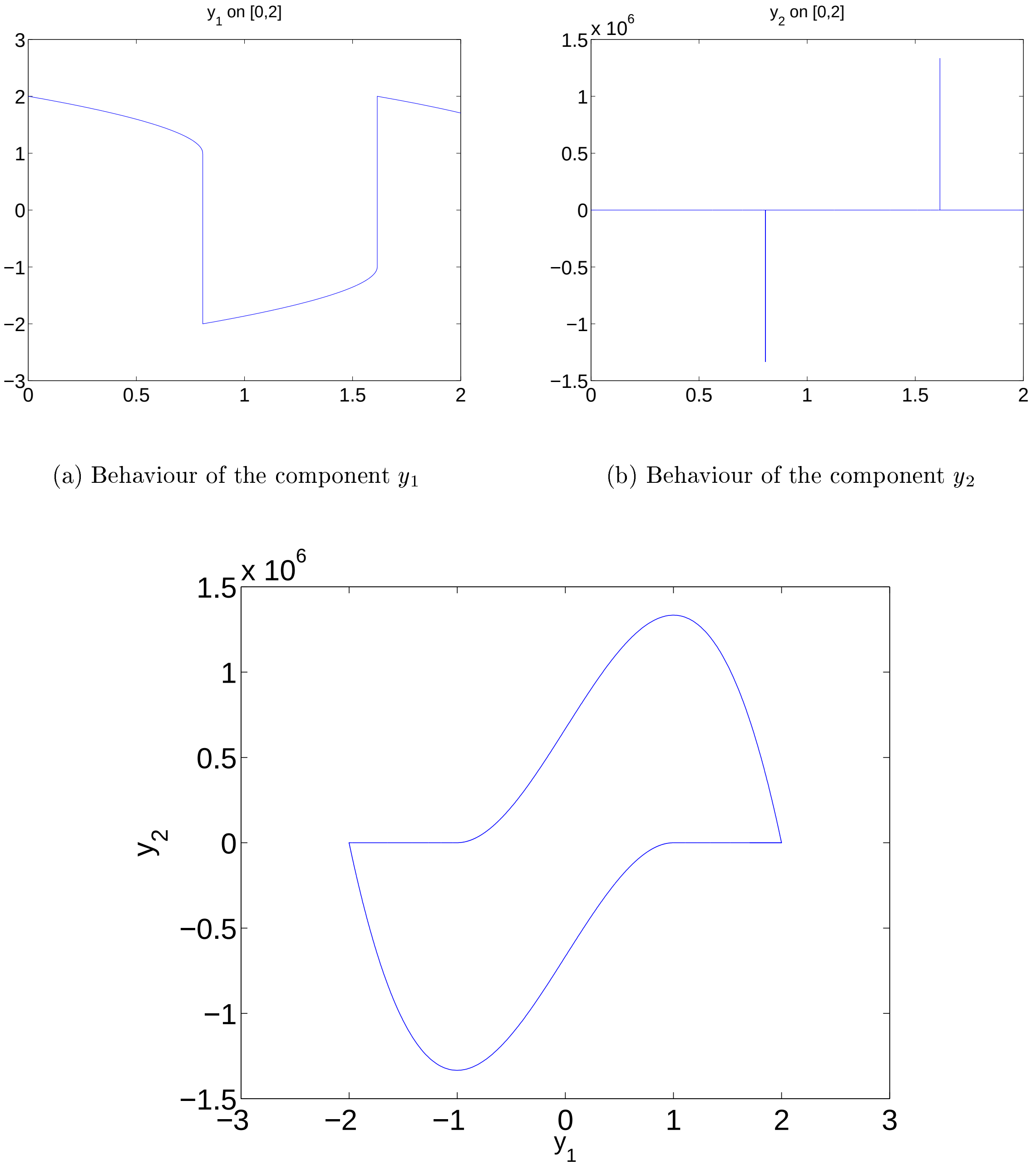
where the parameter μ is set to be a large constant, 10^3 in this case, to make the nonlinear part of the equation more interesting, that is stiff. This can be equivalently written as a coupled system of two first order ordinary differential equations,

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= \mu(1 - y_1^2)y_2 - y_1. \end{aligned}$$

Rescale it for easier tracking of the solution on a small interval and obtain the following with the initial conditions.

$$\begin{aligned} y_1' &= y_2, & y_1(0) &= 2 \\ y_2' &= ((1 - y_1^2)y_2 - y_1)/\epsilon, & y_2(0) &= 0 \end{aligned}$$

Figure A.1: Solution of VDPOL



(a) Behaviour of the component y_1

(b) Behaviour of the component y_2

(c) Limit cycle

where $\epsilon = 10^{-6}$.

The reference solutions at the end of the integration interval, $[0, 2]$ are

$$\begin{aligned} y_1 &= 0.1706167732170483 \times 10, \\ y_2 &= 0.8928097010247975. \end{aligned}$$

The van der Pol equation has two periodic solutions, the constant solution which is unstable and the nontrivial periodic solution called a 'limit cycle'. The behaviour of the two components, y_1 and y_2, and a plot of the limit cycle are displayed in *Figure A.1* at the end of this appendix.

A.2 OREGO

The Oregonator model [25] originates from the Belousov-Zhabotinskii reaction. It describes the chemical reaction of bromous acid, bromide ion and cerium ion which oscillates with changes in colour and structure. The chemical reaction can be modelled by a stiff system of 3 nonlinear ordinary differential equations with corresponding initial conditions,

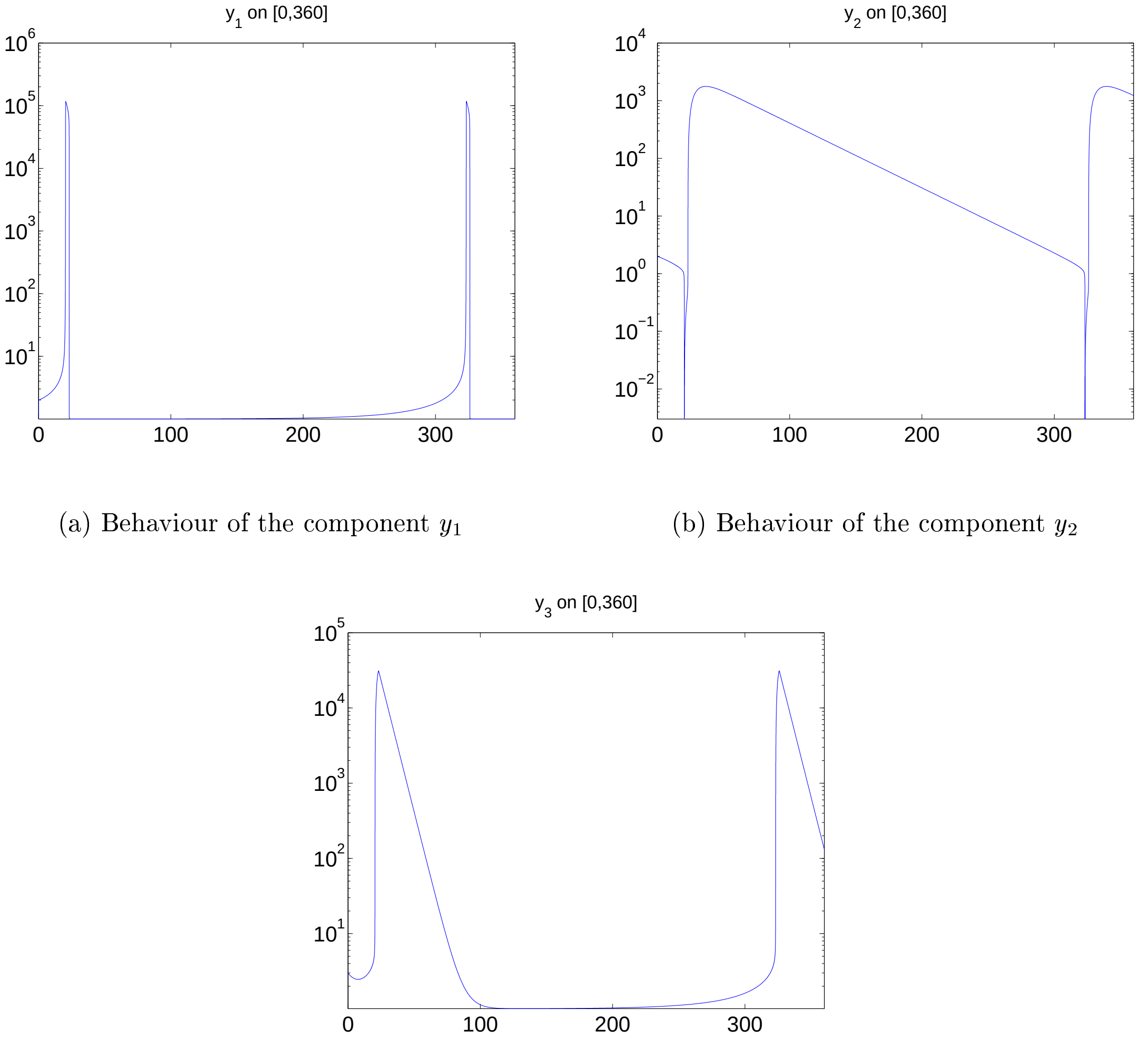
$$\begin{aligned} y_1' &= 77.27(y_2 + y_1(1 - 8.375 \times 10^{-6} y_1 - y_2)), & y_1(0) = 1 \\ y_2' &= \tfrac{1}{77.27}(y_3 - (1 + y_1)y_2), & y_2(0) = 2 \\ y_3' &= 0.161(y_1 - y_3), & y_3(0) = 3. \end{aligned}$$

The reference solutions at the end of the integration interval, $[0, 360]$ are

$$\begin{aligned} y_1 &= 0.1000814870318523 \times 10, \\ y_2 &= 0.1228178521549917 \times 10^4, \\ y_2 &= 0.1320554942846706 \times 10^3. \end{aligned}$$

The behaviour of the solutions of each component over the integration period, $[0, 360]$ are presented in *Figure A.2*.

Figure A.2: Solution of OREGO



(a) Behaviour of the component y_1

(b) Behaviour of the component y_2

(c) Behaviour of the component y_3

A.3 HIRES

The 'High Irradiance Responses' (HIRES) [25] problem consists of a stiff system of 8 nonlinear ordinary differential equations which originates from plant physiology and describes how light is involved in morphogenesis. It explains the high irradiance responses of photomorphogenesis on the basis of phytochrome, by means of a chemical reaction involving eight reactants. The problem is decribed below in the form of an initial value problem.

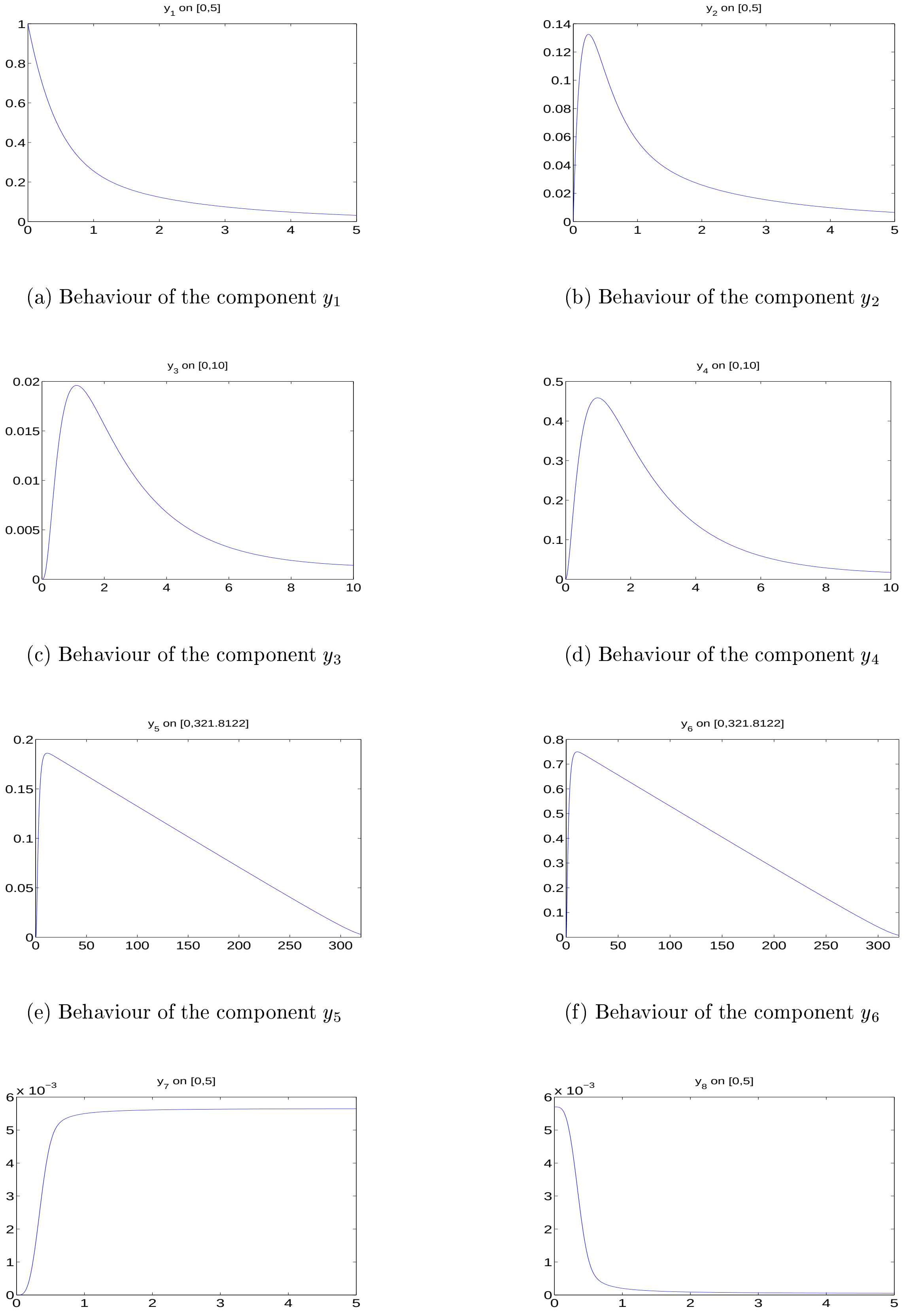
$$\begin{aligned}
y_1' &= -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007, & y_1(0) &= 1\\
y_2' &= 1.71y_1 - 8.75y_2, & y_2(0) &= 0\\
y_3' &= -10.03y_3 + 0.43y_4 + 0.035y_5, & y_3(0) &= 0\\
y_4' &= 8.32y_2 + 1.71y_3 - 1.12y_4, & y_4(0) &= 0\\
y_5' &= -1.745y_5 + 0.43y_6 + 0.43y_7, & y_5(0) &= 0\\
y_6' &= -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7, & y_6(0) &= 0\\
y_7' &= 280y_6y_8 - 1.81y_7, & y_7(0) &= 0\\
y_8' &= -y_7', & y_8(0) &= 0.0057.
\end{aligned}$$

The reference solutions at the end of the integration interval, $[0, 321.8122]$ are

$$\begin{aligned}
y_1 &= 0.7371312573325668 \times 10^{-3},\\
y_2 &= 0.1442485726316185 \times 10^{-3},\\
y_3 &= 0.5888729740967575 \times 10^{-4},\\
y_4 &= 0.1175651343283149 \times 10^{-2},\\
y_5 &= 0.2386356198831331 \times 10^{-2},\\
y_6 &= 0.6238968252742796 \times 10^{-2},\\
y_7 &= 0.2849998395185769 \times 10^{-2},\\
y_8 &= 0.2850001604814231 \times 10^{-2}.
\end{aligned}$$

The behaviour of each of the eight components is shown in *Figure A.3.* Note that, for some components only, the first short period of the solution is plotted for better observation of the behaviour.

Figure A.3: Solution of HIRES



(a) Behaviour of the component y_1

(b) Behaviour of the component y_2

(c) Behaviour of the component y_3

(d) Behaviour of the component y_4

(e) Behaviour of the component y_5

(f) Behaviour of the component y_6

(g) Behaviour of the component y_7

(h) Behaviour of the component y_8

Appendix B

IRKS Methods

B.1 Order 1 methods

s2o1a: $c = \{\frac{1}{2}, 1\}$

$$\left[\begin{array}{cc|cc} \frac{1}{3} & 0 & 1 & \frac{1}{6} \\ \frac{4}{9} & \frac{1}{3} & 1 & \frac{2}{9} \\ \hline \frac{4}{9} & \frac{1}{3} & 1 & \frac{2}{9} \\ 0 & 1 & 0 & 0 \end{array}\right]$$

s2o1b: $c = \{\frac{2}{3}, 1\}$

$$\left[\begin{array}{cc|cc} \frac{1}{3} & 0 & 1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} \\ \hline \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} \\ 0 & 1 & 0 & 0 \end{array}\right]$$

s2o1c: $c = \{1, 1\}$

$$\left[\begin{array}{cc|cc} \frac{1}{3} & 0 & 1 & \frac{2}{3} \\ \frac{2}{9} & \frac{1}{3} & 1 & \frac{4}{9} \\ \hline \frac{2}{9} & \frac{1}{3} & 1 & \frac{4}{9} \\ 0 & 1 & 0 & 0 \end{array}\right]$$

s2o1d: $c = \{\frac{1}{2}, 1\}$

$$\left[\begin{array}{cc|cc} \frac{3}{10} & 0 & 1 & \frac{1}{5} \\ \frac{21}{50} & \frac{3}{10} & 1 & \frac{7}{25} \\ \hline \frac{21}{50} & \frac{3}{10} & 1 & \frac{7}{25} \\ 0 & 1 & 0 & 0 \end{array}\right]$$

B.2 Order 2 methods

s3o2a: $c = \{\frac{1}{3}, \frac{2}{3}, 1\}$

$$\left[\begin{array}{ccc|ccc} \frac{1}{4} & 0 & 0 & 1 & \frac{1}{12} & -\frac{1}{36} \\ \frac{1}{4} & \frac{1}{4} & 0 & 1 & \frac{1}{6} & -\frac{1}{36} \\ \frac{9}{32} & \frac{9}{32} & \frac{1}{4} & 1 & \frac{3}{16} & -\frac{1}{32} \\ \hline \frac{9}{32} & \frac{9}{32} & \frac{1}{4} & 1 & \frac{3}{16} & -\frac{1}{32} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -\frac{9}{2} & 4 & 0 & \frac{1}{2} & 0 \end{array}\right]$$

s3o2b: $c = \{\frac{1}{3}, \frac{2}{3}, 1\}$

$$\left[\begin{array}{ccc|ccc} \frac{1}{3} & 0 & 0 & 1 & 0 & -\frac{1}{18} \\ -\frac{2}{3} & \frac{1}{3} & 0 & 1 & 1 & \frac{2}{9} \\ 0 & \frac{1}{6} & \frac{1}{3} & 1 & \frac{1}{2} & \frac{1}{18} \\ \hline 0 & \frac{1}{6} & \frac{1}{3} & 1 & \frac{1}{2} & \frac{1}{18} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -3 & -\frac{3}{2} & 3 & 0 & \frac{3}{2} & 0 \end{array}\right]$$

s3o2c: $c = \{\frac{3}{7}, 1, 1\}$

$$\left[\begin{array}{ccc|ccc}
\frac{1}{5} & 0 & 0 & 1 & \frac{8}{35} & \frac{3}{490} \\
\frac{1372}{2025} & \frac{1}{5} & 0 & 1 & \frac{248}{2025} & \frac{13}{1350} \\
\frac{2107}{4500} & \frac{9}{100} & \frac{1}{5} & 1 & \frac{272}{1125} & \frac{7}{750} \\
\hline
\frac{2107}{4500} & \frac{9}{100} & \frac{1}{5} & 1 & \frac{272}{1125} & \frac{7}{750} \\
0 & 0 & 1 & 0 & 0 & 0 \\
-\frac{49}{12} & -\frac{9}{4} & 5 & 0 & \frac{4}{3} & 0
\end{array}\right]$$

s3o2d: $c = \{\frac{1}{3}, \frac{2}{3}, 1\}$

$$\left[\begin{array}{ccc|ccc}
\frac{1}{2} & 0 & 0 & 1 & -\frac{1}{6} & -\frac{1}{9} \\
\frac{1}{6} & \frac{1}{2} & 0 & 1 & 0 & -\frac{1}{6} \\
\frac{5}{3} & -\frac{2}{3} & \frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{9} \\
\hline
\frac{5}{3} & -\frac{2}{3} & \frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{9} \\
0 & 0 & 1 & 0 & 0 & 0 \\
-\frac{3}{2} & 0 & \frac{3}{2} & 0 & 0 & 0
\end{array}\right]$$

s3o2e: $c = \{\frac{1}{3}, \frac{2}{3}, 1\}$

$$\left[\begin{array}{ccc|ccc}
\frac{3}{10} & 0 & 0 & 1 & \frac{1}{30} & -\frac{2}{45} \\
-\frac{9}{400} & \frac{3}{10} & 0 & 1 & \frac{467}{1200} & \frac{107}{3600} \\
-\frac{987}{2000} & \frac{12}{25} & \frac{3}{10} & 1 & \frac{1427}{2000} & \frac{89}{2000} \\
\hline
-\frac{987}{2000} & \frac{12}{25} & \frac{3}{10} & 1 & \frac{1427}{2000} & \frac{89}{2000} \\
0 & 0 & 1 & 0 & 0 & 0 \\
\frac{3}{4} & -\frac{3}{4} & \frac{5}{4} & 0 & -\frac{5}{4} & 0
\end{array}\right]$$

B.3 Order 3 methods

s4o3a: $c = \{\frac{1}{3}, \frac{2}{3}, 1, 1\}$

$$\left[\begin{array}{cccc|cccc}
\frac{1}{2} & 0 & 0 & 0 & 1 & -\frac{1}{6} & -\frac{1}{9} & -\frac{7}{324} \\
-\frac{3}{2} & \frac{1}{2} & 0 & 0 & 1 & \frac{5}{3} & \frac{7}{18} & \frac{7}{324} \\
-\frac{99}{14} & \frac{9}{14} & \frac{1}{2} & 0 & 1 & \frac{97}{14} & \frac{27}{14} & \frac{1}{6} \\
\frac{9}{2} & \frac{9}{8} & -\frac{7}{8} & \frac{1}{2} & 1 & -\frac{17}{4} & -\frac{11}{8} & -\frac{7}{48} \\
\hline
\frac{9}{2} & \frac{9}{8} & -\frac{7}{8} & \frac{1}{2} & 1 & -\frac{17}{4} & -\frac{11}{8} & -\frac{7}{48} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\frac{9}{2} & -9 & \frac{7}{2} & 2 & 0 & -1 & 0 & 0 \\
36 & -45 & 14 & 4 & 0 & -9 & 0 & 0
\end{array}\right]$$

s4o3b: $c = \{\frac{3}{4}, 1, \frac{1}{4}, 1\}$

$$\left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & \frac{1}{2} & \frac{3}{32} & 0 \\
\frac{49}{351} & \frac{1}{4} & 0 & 0 & 1 & \frac{857}{1404} & \frac{17}{117} & \frac{1}{416} \\
-\frac{2563}{4914} & \frac{39}{140} & \frac{1}{4} & 0 & 1 & \frac{11941}{49140} & \frac{1523}{18720} & \frac{1}{455} \\
\frac{1325}{1512} & -\frac{131}{336} & -\frac{5}{24} & \frac{1}{4} & 1 & \frac{1427}{3024} & \frac{5}{144} & -\frac{3}{896} \\
\hline
\frac{1325}{1512} & -\frac{131}{336} & -\frac{5}{24} & \frac{1}{4} & 1 & \frac{1427}{3024} & \frac{5}{144} & -\frac{3}{896} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-\frac{178}{21} & \frac{53}{21} & \frac{10}{3} & 4 & 0 & -\frac{29}{21} & 0 & \frac{1}{56} \\
-\frac{112}{3} & \frac{16}{3} & \frac{80}{3} & 16 & 0 & -\frac{32}{3} & 0 & 0
\end{array}\right]$$

s4o3c: $c = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$

$$\left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{28}{15} & \frac{1}{4} & 0 & 0 & 1 & -\frac{97}{60} & -\frac{7}{15} & -\frac{11}{160} \\
\frac{273}{80} & -\frac{45}{128} & \frac{1}{4} & 0 & 1 & -\frac{1639}{640} & -\frac{747}{1280} & -\frac{321}{5120} \\
0 & -\frac{21}{16} & \frac{1}{3} & \frac{1}{4} & 1 & \frac{83}{48} & \frac{21}{32} & \frac{43}{384} \\
\hline
0 & -\frac{21}{16} & \frac{1}{3} & \frac{1}{4} & 1 & \frac{83}{48} & \frac{21}{32} & \frac{43}{384} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-28 & \frac{27}{2} & -\frac{16}{3} & 4 & 0 & \frac{95}{6} & \frac{5}{4} & -\frac{5}{16} \\
-112 & 78 & -\frac{128}{3} & 16 & 0 & \frac{182}{3} & 5 & -\frac{5}{4}
\end{array}\right]$$

s4o3d: $c = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$

$$\left[\begin{array}{cccc|cccc}
\frac{9}{40} & 0 & 0 & 0 & 1 & \frac{1}{40} & -\frac{1}{40} & -\frac{17}{3840} \\
\frac{646797}{257800} & \frac{9}{40} & 0 & 0 & 1 & -\frac{287951}{128900} & -\frac{633907}{1031200} & -\frac{2120851}{24748800} \\
\frac{2037663381}{467391400} & -\frac{11601}{72520} & \frac{9}{40} & 0 & 1 & -\frac{1717514451}{467391400} & -\frac{1677800361}{1869565600} & -\frac{29168331}{267080800} \\
-\frac{4617}{92500} & -\frac{138537}{148000} & \frac{1323}{4000} & \frac{9}{40} & 1 & \frac{529183}{370000} & \frac{1502041}{2960000} & \frac{5662643}{71040000} \\
\hline
-\frac{4617}{92500} & -\frac{138537}{148000} & \frac{1323}{4000} & \frac{9}{40} & 1 & \frac{529183}{370000} & \frac{1502041}{2960000} & \frac{5662643}{71040000} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-\frac{8926}{333} & \frac{7684}{555} & -\frac{98}{15} & \frac{40}{9} & 0 & \frac{2784}{185} & \frac{137}{111} & -\frac{411}{1480} \\
-\frac{394336}{2997} & \frac{102320}{999} & -\frac{1568}{27} & \frac{1600}{81} & 0 & \frac{67408}{999} & \frac{5480}{999} & -\frac{137}{111}
\end{array}\right]$$

s4o3e: $c = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$

$$\left[\begin{array}{cccc|cccc}
\frac{1}{4} & 0 & 0 & 0 & 1 & 0 & -\frac{1}{32} & -\frac{1}{192} \\
\frac{49}{25} & \frac{1}{4} & 0 & 0 & 1 & -\frac{171}{100} & -\frac{49}{100} & -\frac{43}{600} \\
\frac{123}{1225} & -\frac{225}{392} & \frac{1}{4} & 0 & 1 & \frac{1363}{1400} & \frac{13941}{39200} & \frac{5379}{78400} \\
-\frac{95}{84} & -\frac{59}{84} & \frac{7}{36} & \frac{1}{4} & 1 & \frac{43}{18} & \frac{31}{42} & \frac{37}{336} \\
\hline
-\frac{95}{84} & -\frac{59}{84} & \frac{7}{36} & \frac{1}{4} & 1 & \frac{43}{18} & \frac{31}{42} & \frac{37}{336} \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-\frac{268}{21} & \frac{86}{21} & -\frac{28}{9} & 4 & 0 & \frac{70}{9} & \frac{10}{21} & -\frac{5}{21} \\
-\frac{32}{21} & \frac{88}{21} & -\frac{224}{9} & 16 & 0 & \frac{56}{9} & \frac{20}{21} & -\frac{10}{21}
\end{array}\right]$$

Bibliography

[1] F. Bashforth and J. C. Adams. *An Attempt to Test the Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops of Fluid, with an Explanation of the Method of Integration Employed in Constructing the Tables which Give the Theoretical Forms of Such Drops.* Cambridge University Press, 1883.

[2] K. Burrage and J. C. Butcher. Non-linear stability of a general class of differential equation methods. *BIT*, 20:185–203, 1980.

[3] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods.* John Wiley & Sons Ltd., 1987.

[4] J. C. Butcher. ARK methods up to order five. *Numerical Algorithms*, 17:193–221, 1998.

[5] J. C. Butcher. General linear methods for stiff differential equations. *BIT*, 41:240–264, 2001.

[6] J. C. Butcher. The A-stability of methods with Padé and generalized Padé stability functions. *Numerical Algorithms*, 31:47–58, 2002.

[7] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations.* John Wiley & Sons Ltd., 2003.

[8] J. C. Butcher and T. M. H. Chan. A new approach to the algebraic structures for integration methods. *BIT*, 42:477–489, 2002.

[9] J. C. Butcher and P. Chartier. Parallel general linear methods for stiff ordinary differential and differential algebraic equations. *Appl. Num. Math.*, 17:213–222, 1995.

[10] J. C. Butcher and P. Chartier. The effective order of singly-implict Runge-Kutta methods. *Numerical Algorithms*, 20:269–284, 1999.

[11] J. C. Butcher, P. Chartier, and Z. Jackiewicz. Nordsieck representation of DIMSIMs. *Numerical Algorithms*, 16:209–230, 1997.

[12] J. C. Butcher, P. Chartier, and Z. Jackiewicz. Experiments with a variable-order type 1 DIMSIM code. *Numerical Algorithms*, 22:237–261, 1999.

[13] J. C. Butcher and D. J. L. Chen. On the implementation of ESIRK methods for stiff IVPs. *Numerical Algorithms*, 26:201–218, 2001.

[14] J. C. Butcher and M. T. Diamantakis. DESIRE: diagonally extended singly implicit Runge-Kutta effective order methods. *Numerical Algorithms*, 17:121–145, 1998.

[15] J. C. Butcher and A. D. Heard. Stability of numerical methods for ordinary differential equations. *Numerical Algorithms*, 31:59–73, 2002.

[16] J. C. Butcher and Z. Jackiewicz. Construction of diagonally implicit general linear methods of type 1 and 2 for ordinary differential equations. *Appl. Num. Math.*, 21:385–415, 1996.

[17] J. C. Butcher and Z. Jackiewicz. Construction of high order diagonally implicit multistage integration methods for ordinary differential equations. *Appl. Num. Math.*, 27:1–12, 1998.

[18] J. C. Butcher and Z. Jackiewicz. A reliable error estimation for diagonally implicit multistage integration methods. *BIT*, 41:656–665, 2001.

[19] J. C. Butcher and Z. Jackiewicz. Error estimation for Nordsieck methods. *Numerical Algorithms*, 31:75–85, 2002.

[20] J. C. Butcher and N. Moir. Experiments with a new fifth order method. *Numerical Algorithms*, 33:137–151, 2003.

[21] J. C. Butcher and A. E. O'Sullivan. Nordsieck methods with an off-step point. *Numerical Algorithms*, 31:87–101, 2002.

[22] J. C. Butcher and A. D. Singh. The choice of parameters in parallel general linear methods for stiff problems . *Appl. Num. Math.*, 34:59–84, 2000.

[23] J. C. Butcher and W. M. Wright. A transformation relating explicit and diagonally-implicit general linear methods. *Appl. Nume. Math.*, 44:313–327, 2003.

[24] C. F. Curtiss and J. O. Hirschfelder. Integration of Stiff Equations. *Proc. Nat. Acad. Sci.*, 38:235–243, 1952.

[25] CWI Test Set for IVP Solvers Release 2.2, 1996-2003. http://pitagora.dm.uniba.it/ testset/.

[26] G. Dahlquist and Å. Björck. *Numerical Methods.* Prentice-Hall Inc., 1974.

[27] C. W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall Inc., 1971.

[28] E. Hairer, S. P. Nø rsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer-Verlag, 2nd edition, 1993.

[29] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems.* Springer-Verlag, 2nd edition, 1996.

[30] K. Heun. Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen. *Zeitschr. Math. Phys.*, 45:23–38, 1900.

[31] W. Kutta. Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Zeitschr. Math. Phys.*, 46:435–453, 1901.

[32] W. E. Milne. *Numerical Solution of Differential Equations.* Dover Publications Inc., 1970.

[33] F. R. Moulton. *New Methods in Exterior Ballistics.* University of Chicago, 1926.

[34] A. Prothero and A. Robinson. On the Stability and Accuracy of One-Step Methods for Solving Stiff Systems of Ordinary Differential Equations. *Math. Comp.*, 28(125):145–162, 1974.

[35] C. Runge. Über die numerische Auflösung von Differentialgleichungen. *Math. Ann.*, 46:167–178, 1895.

[36] W. M. Wright. The construction of order 4 DIMSIMs for ordinary differential equations. *Numerical Algorithms*, 26:123–130, 2001.

[37] W. M. Wright. Explicit general linear methods with inherent Runge-Kutta stability. *Numerical Algorithms*, 31:381–399, 2002.

[38] W. M. Wright. *General linear methods with inherent Runge-Kutta stability.* PhD thesis, The University of Auckland, 2002.