Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author’s right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage.
http://researchspace.auckland.ac.nz/feedback

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form and Deposit Licence.

Note: Masters Theses

The digital copy of a masters thesis is as submitted for examination and contains no corrections. The print copy, usually available in the University Library, may contain corrections made by hand, which have been requested by the supervisor.
Extrapolation of symmetrized Runge-Kutta methods

Annie Gorgey

Department of Mathematics

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

Doctor of Philosophy in Applied Mathematics

June 2012
Smoothing was first introduced by Gragg in 1964 for ordinary differential equations and later used by others in one-step methods of order 1 or 2 for stiff equations. The generalization of smoothing to symmetric Runge-Kutta methods of arbitrary order is called symmetrization and was due to Chan. It involves an L-stable method, called a symmetrizer, constructed to preserve the $h^2$-asymptotic error expansion of an associated symmetric method. The resulting damping suppresses order reduction in solving stiff problems and thus increases the efficiency of extrapolation by improving the accuracy by two orders at a time. In this thesis we study symmetric methods of orders 4 and 6 from the Gauss and Lobatto IIIA families and investigate the accuracy and efficiency when applied with symmetrization and extrapolation.

The emphasis of this thesis is on implementation of the numerical methods and numerical experiments. In the constant stepsize setting, we study two ways of applying symmetrization, active and passive, and four ways of applying extrapolation with symmetrization. We give a complete theoretical analysis of the order behaviour for the Prothero-Robinson problem and verify the results by numerical experiments. We observe that the symmetrized Gauss methods are more efficient than the symmetrized Lobatto IIIA methods of the same order for stiff linear problems. However, for two dimensional nonlinear problems, the symmetrized 4-stage Lobatto IIIA method is more efficient. In all cases, we observe numerically that passive symmetrization with passive extrapolation is more efficient than active symmetrization with active extrapolation.

In the variable stepsize setting, we find that symmetrization can be used for error estimation. We find that this error estimation is more efficient than the use of the local extrapolation approach. There are two ways of applying extrapolation with symmetrization in the variable stepsize setting. In the numerical experiments on the STIFF DETEST problem set we observe that passive symmetrization with active extrapolation is more efficient than active symmetrization with active extrapolation. For linear problems the symmetrized 3-stage Gauss method with active extrapolation is more efficient that the symmetrized 4-stage Lobatto IIIA method with active extrapolation. For nonlinear problems, however, the opposite is observed.

This thesis provides definite evidence that symmetric methods of high order when applied with appropriate symmetrization/extrapolation can be effective practical methods for the numerical solution of stiff linear and nonlinear problems.
First and foremost, I would like to thank my Lord and Saviour Jesus Christ for teaching me through this process that He is to be feared and that He inspires awe. His hand and His forgiveness are the things that have sustained me to the end of this process. Thank you Lord for your wisdom and your mercy.

I would like to extend special thanks to my supervisor Dr. Robert Chan for his patience, guidance, motivation and diligent support. I have enjoyed learning from him. Dr. Chan is a great teacher and friend. Of course, special thanks to Prof. John Butcher for always being there to motivate and share ideas. His contributions scholastically and as a friend will be cherished as long as I know him.

I wish to thank the Numerical Runge-Kutta group in Auckland; Dr. Allison Heard, Dr. Shixiao Wang, Dr. Angela Tsai, Dr. Yousaf Habib, Dr. Pavla Sehnalová, Gulshad Imran, Saghir Ahmad and Noorhelyna Razali. Their insight and friendship are most appreciated.

I would also like to thank my special friends, who have been there for me when I needed them. They have given me support and motivation. Their time spent with me helped alleviate the stress, I truly thank the Lord for them. Thank you Every Nation church families: Sam Boychuk, Monique Knoetze, Su Moon Ting and Ian Thompson.

Not forgetting the Ministry of Higher Education Malaysia especially the “Universiti Pendidikan Sultan Idris, Malaysia” for providing me with the funds to study. I would not have been able to pursue my studies without their financial assistance. I would also like to thank the University of Auckland and the Mathematics department for the travel funding that has enabled me to attend conferences overseas. I would like to thank the administration staff in the Mathematics department; Adina Nagy, Olita Moala, Lynda Pitcaithly and Min-Ah for helping to sort out various administrative tasks.

And last but second only to Christ, my warm thanks to my family, my Mum and Dad, brothers and sister for teaching and showing me their never ending love.

“Trust in the LORD with all your heart and lean not on your own understanding; in all your ways submit to Him, and He will make your paths straight” (Proverbs 3:5-6).
Contents

Abstract i

Acknowledgements iii

List of Figures xiii

List of Tables xvi

1 Introduction and Motivation 1

1.1 Background and basic concepts ........................................ 1
  1.1.1 Ordinary differential equations .................................... 1
  1.1.2 Local and global errors ............................................. 2
  1.1.3 Stiff problems .................................................. 5

1.2 Numerical methods for IVPs .......................................... 6
  1.2.1 Runge-Kutta methods ............................................. 6
  1.2.2 Theory of Runge-Kutta methods .................................. 8
  1.2.3 Stability and stiffness ........................................... 12

1.3 Implicit methods .................................................. 16

1.4 Objectives and research contributions ................................ 21

1.5 Thesis outline .................................................. 23
## Contents

2 Extrapolation of symmetric Runge-Kutta methods

2.1 Symmetric Runge-Kutta methods

2.1.1 Symmetry of RK methods

2.1.2 S-Reducibility

2.1.3 Operations on RK methods

2.1.4 Asymptotic error expansion

2.1.5 Order reduction

2.1.6 Weak damping properties

2.2 Idea of extrapolation

2.2.1 Richardson extrapolation

2.2.2 Quadrature

2.2.3 Extrapolation in ODEs

2.3 Generalization of smoothing

3 Symmetrized Gauss and Lobatto IIIA methods

3.1 Symmetrization

3.2 Stability function of the symmetrizer

3.3 Order conditions

3.4 Construction of the symmetrizers

3.4.1 Symmetric methods with nonsingular $A$

3.4.2 Symmetric methods with singular $A$

4 Strategies for symmetrization and extrapolation

4.1 Choice of strategies for symmetrization

4.1.1 Active symmetrization

4.1.2 Passive symmetrization
4.2 Analysis of active and passive symmetrization ........................................ 77
  4.2.1 Active symmetrization by $\hat{R}_1 = \tilde{R}$ ........................................ 77
  4.2.2 Passive symmetrization after $n-1$ steps .......................................... 86
4.3 Choice of strategies for extrapolation ...................................................... 88
4.4 Numerical results for the test problems .................................................. 92
  4.4.1 Optimal choice of $m$ for active symmetrization ................................. 94
  4.4.2 Order behaviour of active and passive symmetrization ....................... 96
  4.4.3 Active and passive symmetrization on different intervals ................... 104
  4.4.4 Active and passive symmetrization with extrapolation ....................... 108
  4.4.5 Effect of perturbation on symmetrization with extrapolation ............. 110
4.5 Summary of results for the test problems .............................................. 113

5 Implementation issues ................................................................................. 117
  5.1 Implementation of implicit methods ....................................................... 118
  5.2 Variable stepsize and algorithms ........................................................... 122
    5.2.1 Error estimation by local extrapolation ......................................... 122
    5.2.2 Error estimation by the symmetrization ......................................... 123
    5.2.3 Error estimation by the local extrapolation with symmetrization ..... 124

6 Extended Numerical Experiments ............................................................... 131
  6.1 Results for constant stepsize ............................................................... 132
  6.2 Results for variable stepsize ............................................................... 140
  6.3 Performance of certain methods ........................................................... 144

7 Conclusions and Future Work ................................................................. 159
  7.1 Conclusions ......................................................................................... 159
List of Figures

1.1 Euler method at step $n$ with stepsize $h$ ........................................ 3
1.2 Error behaviour of the explicit and implicit Euler methods for stiff problem. 16

2.1 Phenomenon of order reduction of the 2-stage and 3-stage Gauss methods. 39
2.2 Error behaviour of IMR and ITR. ...................................................... 43
2.3 Error behaviour of EMR with and without smoothing for the Kepler problem. 50
2.4 Error behaviour of IMR and ITR with smoothing. ............................... 58

3.1 Symmetrization in the $m$-th step .................................................... 60
3.2 Stability functions and stability regions of the symmetrized Gauss and symmetrized Lobatto IIIA methods. ................................. 74

4.1 Passive symmetrization at step $n$. .............................................. 77
4.2 No symmetrization with passive extrapolation ................................. 89
4.3 No symmetrization with active extrapolation. .................................. 90
4.4 Passive symmetrization with passive extrapolation at $x_2$. .............. 90
4.5 Passive symmetrization with active extrapolation at $x_2$. ............... 91
4.6 Active symmetrization with passive extrapolation at $x_2$. ............... 91
4.7 Active symmetrization with active extrapolation at $x_2$. ............... 92
4.8 Efficiency diagrams of Gauss and Lobatto IIIA methods with active symmetrization on optimal $m$ values applied to the PR problem. . . . . . . . . . . . . 95
4.9 Efficiency diagrams of Gauss and Lobatto IIIA methods with active symmetrization on optimal $m$ values applied to the Kaps problem. . . . . . . . . . . . . 96
4.10 Accuracy diagrams of the 2-stage Gauss method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$. . . . . . . . . . . . . 97
4.11 Accuracy diagrams of the 3-stage Gauss method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$. . . . . . . . . . . . . 98
4.12 Accuracy diagrams of the 3-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$. . . . . . . . . . . . . 99
4.13 Accuracy diagrams of the 4-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$. . . . . . . . . . . . . 99
4.14 Accuracy diagrams of the 2-stage Gauss method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_n = 5$. . . . . . . . . . . . . 101
4.15 Accuracy diagrams of the 3-stage Gauss method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_n = 5$. . . . . . . . . . . . . 101
4.16 Accuracy diagrams of the 3-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_n = 5$. . . . . . . . . . . . . 102
4.17 Accuracy diagrams of the 4-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_n = 5$. . . . . . . . . . . . . 102
4.18 Efficiency diagrams of the 2-stage Gauss method with symmetrization at different intervals applied to the PR problem. . . . . . . . . . . . . . . . . . . 105
4.19 Efficiency diagrams of the 3-stage Gauss method with symmetrization at different intervals applied to the PR problem. We have used the order-3 symmetrizer in this experiment. . . . . . . . . . . . . . . . . . . 105
4.20 Efficiency diagrams of the 3-stage Lobatto IIIA method with symmetrization at different intervals applied to the PR problem. . . . . . . . . . . . . . . . . . . 106
4.21 Efficiency diagrams of the 4-stage Lobatto IIIA method with symmetrization at different intervals applied to the PR problem. . . . . . . . . . . . . . . . . . . 106
4.22 Efficiency diagrams of the Gauss methods with symmetrization at different intervals applied to the Kaps problem. 107

4.23 Efficiency diagrams of the Lobatto IIIA methods with symmetrization at different intervals applied to the Kaps problem. 107

4.24 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the PR problem. 109

4.25 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the Kaps problem. 109

4.26 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the perturbed PR problem with $\lambda = -10$. 110

4.27 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the perturbed PR problem with $\lambda = -10^6$. 111

4.28 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the linear matrix problem for $L = -10^6$ and $\epsilon = 0$. 112

4.29 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the linear matrix problem for $L = -10^6$ and $\epsilon = 2$. 112

5.1 Cheaper implementation of the 2-stage Gauss method for the Kaps and HIRES problems. 120

5.2 IMR with and without compensated summation for the PR problem. 121

5.3 The error estimation of the 2-stage Gauss method with symmetrization and extrapolation applied to the Van der Pol problem. 125

5.4 The error estimation of the 3-stage Gauss method with symmetrization and extrapolation applied to the Van der Pol problem. 125
List of Figures

6.1 Efficiency diagrams of the symmetrized 2-stage Gauss and symmetrized 3-stage Lobatto IIla methods with extrapolation applied to the linear test problems B.1.1 using constant stepsize. ................................. 136

6.2 Efficiency diagrams of the symmetrized 3-stage Gauss and symmetrized 4-stage Lobatto IIla methods with extrapolation applied to the linear test problems B.1.1 using constant stepsize. ................................. 137

6.3 Efficiency diagrams of the symmetrized 2-stage Gauss and symmetrized 3-stage Lobatto IIla methods with extrapolation applied to the nonlinear problems using constant stepsize. ................................. 138

6.4 Efficiency diagrams of the symmetrized 3-stage Gauss and symmetrized 4-stage Lobatto IIla methods with extrapolation applied to the nonlinear problems using constant stepsize. ................................. 138

6.5 Efficiency diagrams of order-4 methods with high level extrapolation applied to the nonlinear problems using constant stepsize. ................................. 139

6.6 Efficiency diagrams of order-6 methods with high level extrapolation applied to the nonlinear problems using constant stepsize. ................................. 140

6.7 Efficiency diagrams of the symmetrized Gauss and Lobatto IIla methods with extrapolation applied to the DETEST class A problems using variable stepsize. ....................................................... 141

6.8 Efficiency diagrams of the symmetrized Gauss and Lobatto IIla methods with extrapolation applied to the DETEST class B problems using variable stepsize. ....................................................... 142

6.9 Efficiency diagrams of the symmetrized Gauss and Lobatto IIla methods with extrapolation applied to the DETEST class C problems using variable stepsize. ....................................................... 142

6.10 Efficiency diagrams of the symmetrized Gauss and Lobatto IIla methods with extrapolation applied to the DETEST class D problems using variable stepsize. ....................................................... 143

6.11 Efficiency diagrams of the symmetrized Gauss and Lobatto IIla methods with extrapolation applied to the DETEST class E problems using variable stepsize. ....................................................... 143
A.1 Accuracy diagrams of the Gauss and Lobatto IIIA methods with symmetrization for different $\lambda$ values applied to the linear matrix problem at $x_n = 5$. ................................................................. 164

A.2 Efficiency diagrams of the 2-stage Gauss and the 3-stage Lobatto IIIA methods with symmetrization applied to the linear test problems B.1.1 using constant stepsize. ................................. 165

A.3 Efficiency diagrams of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization applied to the linear test problems B.1.1 using constant stepsize. ................................. 165

A.4 Efficiency diagrams of the 2-stage Gauss and 3-stage Lobatto IIIA methods with symmetrization applied to the nonlinear problems using constant stepsize. ......................................................... 166

A.5 Efficiency diagrams of the 3-stage Gauss and 4-stage Lobatto IIIA methods with symmetrization applied to the nonlinear problems using constant stepsize. ......................................................... 166

A.6 Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the test problem LP15 for $\alpha = 10^6$ using constant stepsize. ......................................................... 167
List of Tables

1.1 Some properties of trees up to order 5 .......................... 11
1.2 Gauss methods of order 2, 4 and 6. .............................. 18
1.3 Lobatto IIIA methods of order 2, 4 and 6. ....................... 20
1.4 Radau IIA methods of order 1, 3 and 5. ......................... 20
1.5 Lobatto IIIC methods of order 2, 4 and 6. ...................... 21
3.1 The order conditions for $\tilde{G}_3$ for trees of order 4 and 5 when $C(3)$ holds. . 65
3.2 Maximum attainable order of symmetrizers for Gauss methods ........ 69
4.1 Summary of order for Gauss and Lobatto IIIA methods with passive ($\tilde{p}$) and active ($\hat{p}$) symmetrization for linear and nonlinear scalar and matrix problems. ............................................. 103
4.2 Notation for the PR and Kaps problems on the choice of symmetrization with extrapolation. ................................................. 108
6.1 Notation for numerical experiments ............................... 134
6.2 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST A1 problem. .............................................. 146
6.3 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST A3 problem. .............................................. 147
6.4 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST B2 problem. .................................................. 148
6.5 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST B5 problem. .................................................. 149
6.6 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST C2 problem. .................................................. 150
6.7 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST C5 problem. .................................................. 151
6.8 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST D2 problem. .................................................. 152
6.9 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST D5 problem. .................................................. 153
6.10 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST D6 problem. .................................................. 154
6.11 Numerical results of the 3-stage Gauss and 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST E1 problem. .................................................. 155
6.12 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST E4 problem. .................................................. 156
6.13 Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST E5 problem. .................................................. 157
1

INTRODUCTION AND MOTIVATION

1.1 Background and basic concepts

1.1.1 Ordinary differential equations

To understand the behaviour of physical, chemical and biological systems, we often need to solve a mathematical model that includes one or more ordinary differential equations (ODEs). If an equation contains derivatives, we say that it is a differential equation abbreviated as DE.

In this thesis, we focus attention on a system of ordinary differential equations with given initial values, called an initial value problem (IVP). It has the form

\[
y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \quad f : [x_0, x_n] \times \mathbb{R}^N \to \mathbb{R}^N,
\]

(1.1)

where \( x \) is the independent or time variable and \( y \) is the dependent variable. In a vector
notation, (1.1) can be written as

\[
\begin{bmatrix}
y'_1 \\
y'_2 \\
\vdots \\
y'_N \\
\end{bmatrix}, \quad f(x,y) = \begin{bmatrix}
f_1(x,y_1, y_2, \ldots, y_N) \\
f_2(x,y_1, y_2, \ldots, y_N) \\
\vdots \\
f_N(x,y_1, y_2, \ldots, y_N) \\
\end{bmatrix}, \quad y_0 = \begin{bmatrix}
y_{10} \\
y_{20} \\
\vdots \\
y_{N0} \\
\end{bmatrix}.
\] (1.2)

If \( f \) depends explicitly on \( x \) it is said to be nonautonomous. On the other hand, if \( f \) is not a function of \( x \) but of \( y \) only it is said to be autonomous. However, it is always possible to write a nonautonomous equation in an equivalent autonomous form by introducing an additional component \( y_{N+1} \) into the \( y \) vector satisfying \( y'_{N+1} = 1 \).

We will assume that \( f \) satisfies the Lipschitz condition [64] to guarantee existence and uniqueness of the solutions.

**Definition 1.1 [15]** The function \( f : [x_0, x_n] \times \mathbb{R}^N \to \mathbb{R}^N \) is said to satisfy a Lipschitz condition in its second variable if there exists a constant \( L \), known as a ‘Lipschitz constant’, such that for any \( x \in [x_0, x_n] \) and \( y, z \in \mathbb{R}^N \),

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

**Theorem 1.1** For a given initial value problem (1.1), for which \( f : [x_0, x_n] \times \mathbb{R}^N \to \mathbb{R}^N \) is continuous in its first variable and satisfies a Lipschitz condition in its second variable, there exists a unique solution to this problem.

A proof of this is given in [15].

### 1.1.2 Local and global errors

To understand the concept of local and global errors, let (1.1) be the differential equations that we wish to solve with assumption that we have already computed the solution values \( y_1, y_2, \ldots, y_{n-1} \) at points \( x_1, x_2, \ldots, x_{n-1} \) respectively. To compute the solution at \( x_n \), the numerical method now attempts to solve

\[
y' = f(x, y), \quad y(x_{n-1}) = y_{n-1}.
\]
If the solution curve is \( u_n(x) \) and the exact solution curve is \( y(x) \), then the Local Error (LE) in the computed solution \( y_n \) can be written as

\[
\ell_n = u_n(x_n) - y_n.
\]

Figure 1.1 shows the numerical approximation of the Euler method at step \( n \). The error committed at each step is the local error. On the other hand, the global error (GE) of step \( n \) is the accumulation of local errors after \( n \) steps. It is however, not the sum of the local errors at each of the \( n \) steps but is bounded by the sum of the bounds on the local errors. This bound often over-estimates the actual error but does give an indication of the rate of error growth. The global error is given by

\[
\epsilon_n = y(x_n) - y_n, \\
= y(x_n) - u_n(x_n) + \ell_n, \quad (1.3)
\]

and is the actual error after \( n \) steps. Thus global error has two components, one due to the local error at the present step and the other due to local errors at the previous steps.

If the equations are very stable, then the errors in the early stages will have little effect. However, if they are unstable, the early errors will have a larger effect. It is therefore important to investigate the convergence of the numerical methods. For example, at the
Chapter 1. Introduction and Motivation

In the $n$-th step, the local error of the famous Euler method is given by

$$\ell_n = y(x_{n-1} + h) - y(x_{n-1}) - hf(x_{n-1}, y(x_{n-1})).$$

By the Taylor series expansion of $y(x_{n-1} + h)$ about $x_{n-1}$,

$$\ell_n = y(x_{n-1}) + hy'(x_{n-1}) + \frac{h^2}{2}y''(x_{n-1} + \theta h) - y(x_{n-1}) - hf(x_{n-1}, y(x_{n-1})),$$

$$= \frac{h^2}{2}y''(x_{n-1} + \theta h), \quad 0 < \theta < 1.$$

If the second derivative of the exact solution is bounded by some constant $K > 0$ throughout the interval $[x_0, x_n]$, then $||y''(x)|| \leq K$ for all $x \in [x_0, x_n]$ and therefore $||\ell_n|| \leq \frac{1}{2}Kh^2$ for all $n$. Then together with the global error (1.3) we have

$$\epsilon_n = y(x_{n-1} + h) - y_n = y(x_{n-1}) + hf(x_{n-1}, y(x_{n-1})) + \ell_n - y_{n-1} - hf(x_{n-1}, y_{n-1}),$$

$$= \epsilon_{n-1} + h(f(x_{n-1}, y(x_{n-1}))) - f(x_{n-1}, y_{n-1})) + \ell_n.$$

Taking the norm and applying the triangle inequality, the Lipschitz condition and the bound on the local error, we obtain

$$||\epsilon_n|| \leq ||\epsilon_{n-1}|| + h||f(x_{n-1}, y(x_{n-1}))) - f(x_{n-1}, y_{n-1})|| + ||\ell_n||,$$

$$\leq ||\epsilon_{n-1}|| + hL||y(x_{n-1}) - y_{n-1}|| + \frac{Kh^2}{2},$$

$$= (1 + hL)||\epsilon_{n-1}|| + \frac{Kh^2}{2}.$$

This is a first-order difference inequality in $||\epsilon_n||$ with constant coefficients and an inhomogeneous constant term. Since $\epsilon_0 = 0$, the inequality has solution given by

$$||\epsilon_n|| \leq (1 + hL)^n||\epsilon_0|| + \frac{Kh^2}{2} \left(1 + (1 + hL) + \cdots + (1 + hL)^{n-1}\right),$$

$$= \frac{Kh^2(1 + hL)^n - 1}{hL} = \frac{Kh}{2L} \left((1 + hL)^n - 1\right),$$

$$\leq \frac{Kh}{2L} \left(e^{L(x_n-x_0)} - 1\right).$$

Thus the numerical solution converges to the exact solution as $n \to \infty$ and $h \to 0$ such that $nh = x_n - x_0$ and we have $\epsilon_n \to 0$. In proving the convergence, we derived the error bound for the numerical solution. Although the derivation is given for the Euler method,
we can easily verify convergence for any high order method using a similar approach. In this case, the global error would be bounded by a higher power of $h$ with $K$ referring to a bound on a higher derivative of $x$.

To construct a method in the solution of ODEs, we need to consider A-stability, B-stability, B-convergence, order reduction and etc. We will discuss some of these properties in the following sections. Furthermore, different types of problems often require different numerical methods. The type of problem we are interested in are stiff problems.

1.1.3 Stiff problems

Stiff problems arise in practice whenever the solution components have widely different time scales. A detailed account of stiffness is given in Section 1.2.3. An example of a stiff problem is the Prothero-Robinson problem. It is a scalar problem often used as the test problem numerically and analytically since the exact solution is known.

Example 1.1 Prothero-Robinson problem

Prothero-Robinson (PR) is a linear problem used by Prothero and Robinson [66] in 1974 for the study of a phenomenon called order reduction. The problem is stiff when $|\lambda|$ is large but the exact solution will remain unaffected by the degree of stiffness.

$$y' = \lambda(y - g(x)) + g'(x), \quad y(0) = g(0), \quad (1.4)$$

with exact solution $y(x) = g(x)$ where $g(x)$ is a smooth function.

When solving a stiff problem, we cannot depend on the Lipschitz condition alone but instead we need to consider a one-sided Lipschitz condition.

Definition 1.2 The function $f$ satisfies a one-sided Lipschitz condition, with one-sided Lipschitz constant $l$, if for all $x \in [x_0, x_n]$ and all $u, v \in \mathbb{R}^N$,

$$\langle f(x, u) - f(x, v), u - v \rangle \leq l\|u - v\|^2,$$

where we assume that there exists an inner product on $\mathbb{R}^N$ denoted by $\langle u, v \rangle$ and the norm is defined by $\|u\|^2 = \langle u, u \rangle$. Therefore, even for stiff problems when $L$ becomes very large we can always manage with a one-sided Lipschitz constant. This is important in obtaining a realistic growth estimate for the effect of perturbations, which is the third
Chapter 1. Introduction and Motivation

requirement for being well-posed.

Most convergence concepts for discretizations of nonlinear stiff initial value problems are based on one-sided Lipschitz continuity. It was introduced by Dahlquist [31] in 1975 where he defined the concept of G-stability for multistep methods. In the case of Runge-Kutta methods, Butcher [13] was the first to develop the concept of B-stability. However, the Vienna group [5] showed that for many stiff problems, the assumption of moderately sized one-sided Lipschitz constants failed to be realised. The issue arises in the context of convergence results is the concept of B-convergence instead of the one-sided Lipschitz constant.

As mentioned above, in order to construct a practical method, we need to consider several criteria such as stability, accuracy and efficiency. For some problems, the stability of the method is important, while for some other problems accuracy is important. However, efficiency has become very important. We want to find a method that possess good properties as well as cheap implementation cost. For example, the Euler method requires only one function evaluation per-step and therefore is cheap to implement but this method fails for stiff problems (see Figure 1.2 on page 16).

In the following section, we consider the theory of Runge-Kutta methods and their properties particularly that of stability.

1.2 Numerical methods for IVPs

1.2.1 Runge-Kutta methods

An s-stage Runge-Kutta method for the step \((x_{n-1}, y_{n-1}) \rightarrow (x_n, y_n)\) with stepsize \(h\) is defined by

\[
Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f(x_{n-1} + c_j h, Y_j), \quad i = 1, \ldots, s,
\]

\[
y_n = y_{n-1} + h \sum_{i=1}^{s} b_i f(x_{n-1} + c_i h, Y_i).
\]  \hspace{1cm} (1.5)

It is a one step method and can be illustrated schematically in the following diagram:
1.2. Numerical methods for IVPs

\[
\begin{array}{cccccc}
& y_{n-1} & Y_1 & \ldots & Y_i & \ldots & y_n \\
& x_{n-1} & x_{n-1} + c_1h & \ldots & x_{n-1} + c_ih & \ldots & x_{n-1} + c_sh = x_n \\
\end{array}
\]

Here, the \( Y_i \) represents the internal stage value for the \( i \)-th stage and \( y_n \) is the update at the \( n \)-th step. We shall always assume the row-sum condition

\[
c_i = \sum_{j=1}^{s} a_{ij}, \quad i = 1, 2, \ldots, s. \tag{1.6}
\]

It is convenient to display (1.5) by a partitioned tableau known as the Butcher tableau

\[
\begin{array}{c|c}
\mathbf{c} & \mathbf{A} \\
\hline
\mathbf{b}^T \\
\end{array}
\]

where \( \mathbf{c} \) is the vector of abscissae which indicates the positions within the step of the stage values and the matrix \( \mathbf{A} \) are the coefficients used to find the internal stages using the linear combinations of the stage derivatives. The vector \( \mathbf{b} \) represents the quadrature weights which indicate how the approximation to the solution depends on the derivatives of the internal stages.

Runge-Kutta methods can be divided into two main types according to the structure of the matrix \( \mathbf{A} \). A Runge-Kutta method is called explicit if the \( \mathbf{A} \) matrix is strictly lower triangular such as the famous classical Runge-Kutta method of order 4 (RK4), or else the method is called implicit. Implicit methods can be divided into several other categories, for example, fully-implicit if matrix \( \mathbf{A} \) is not lower triangular, semi-implicit if \( \mathbf{A} \) is lower triangular with at least one non-zero diagonal element, diagonal implicit if \( \mathbf{A} \) is lower triangular with all the diagonal elements equal and non-zero (DIRK) and singly implicit if \( \mathbf{A} \) is matrix with a single nonzero eigenvalue (SIRK).

Explicit methods are easy to implement as the internal stages can be calculated sequentially without depending on later stages and are also easy to code. However, explicit methods cannot be used to solve stiff problems since they have poor stability behaviour (refer Subsection 1.2.3). On the other hand, implicit methods are suitable for solving stiff problems but are more costly to implement. Implementation of implicit methods is discussed in Chapter 5 on Section 5.1. Implicit methods are discussed in Section 1.3.
Examples of some methods in the form of (1.7) are

\[
\begin{array}{cccc|cccc|cc}
0 & & & & & & & & \\
\frac{1}{2} & \frac{1}{2} & & & \frac{1}{2} & -\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} & -\frac{\sqrt{3}}{6} & 0 \\
\frac{1}{2} & 0 & \frac{1}{2} & & \frac{1}{2} & +\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} & +\frac{\sqrt{3}}{6} & 0 \\
1 & 0 & 0 & 1 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0
\end{array}
\]

They are the RK4, 2-stage Gauss and DIRK method [2] respectively.

Next, we consider the theory of Runge-Kutta methods including the order conditions, elementary differentials, trees etc. A comprehensive theory of Runge-Kutta methods is given by Butcher in [17].

### 1.2.2 Theory of Runge-Kutta methods

Let \( T \) denoted the set of rooted trees. A rooted tree is simply a connected graph where each vertex has a single parent [15]. The order of the rooted tree \( t \) is the number of vertices of \( t \) and denoted by \( r(t) \).

The trees with up to four vertices are given by

\[
\text{(1.8)}
\]

The elementary differentials are derived using the chain rule and are given to the fourth derivative.

\[
\begin{align*}
y' &= f(y) = f, \\
y'' &= f'(y)y' = f'(y)f(y) = f'f, \\
y''' &= f''(y)(f(y), f(y)) + f'(y)f'(y)f(y) = f''(f, f) + f'f'f, \\
y^{(4)} &= f'''(y)(f(y), f(y), f(y)) + 3f''(y)(f(y), f'(y)f(y)) + f'(y)f''(y)(f(y)(f(y))) \\
&\quad + f'(y)f'(y)f'(y)f(y), \\
&= f'''(f, f, f) + 3f''(f, f'f) + f'f''(f, f) + f'f'f'.
\end{align*}
\]
The terms in this expression correspond exactly to the trees with 1, 2, 3 and 4 vertices as shown:

The expressions involving \( f, f', f'', \ldots \), are known as elementary differentials and are denoted by \( F(t) \).

In addition, we can also label a tree using the coefficients \( A, b^T \) and \( c \). The root is labelled with \( b^T \), terminal vertices with \( c \) and non-terminal vertices with \( A \). Corresponding to each \( t \) is a real number called the elementary weights and are denoted by \( \Phi(t) \).

Associated with the tree \( t \) are several real-valued functions. It can be shown that the Taylor expansion of the exact solution with \( y(x_0) = y_0 \) has the form

\[
y(x_0 + h) = y_0 + \sum_{t \in T} \frac{\alpha(t) h^{r(t)}}{r(t)!} F(t)(y_0),
\]

\[
= y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t) \gamma(t)} \Phi(t) F(t)(y_0), \tag{1.9}
\]

where \( \sigma(t) \) and \( \alpha(t) = \frac{r(t)!}{\sigma(t) \gamma(t)} \) denotes the symmetry of \( t \), and \( \gamma(t) \) is the density of \( t \).

The numerical solution computed using a Runge–Kutta method \( (A, b, c) \) is given by

\[
y_0 + \sum_{t \in T} \frac{h^{r(t)}}{\sigma(t)} \Phi(t) F(t)(y_0), \tag{1.10}
\]

Thus, comparing (1.9) with (1.10) gives the conditions for a method to be of order \( p \):

\[
\Phi(t) = \frac{1}{\gamma(t)}, \quad \text{for } t : r(t) \leq p.
\]
Chapter 1. Introduction and Motivation

The summary of trees up to order 5 is given in Table 1.1.

As the order increases, the number of order conditions increases rapidly and becomes unmanageable. For this reason, simplifying assumptions are introduced to simplify the order conditions. They are due to Butcher [10].

In matrix form, we have

\[
B(p) : \quad b^T c^{k-1} = \frac{1}{k}, \quad k = 1, \ldots, p, \\
C(q) : \quad A c^{k-1} = \frac{c^k}{k}, \quad k = 1, \ldots, q, \\
D(r) : \quad b^T C^{k-1} A = \frac{1}{k} [b^T - b^T C^k], \quad k = 1, \ldots, r, \quad (1.11)
\]

where \( C = \text{diag}(c_1, \ldots, c_s) \). The \( B(p) \) conditions refer to the bushy trees (solid, solid, solid, etc). The minimum of \( p \) and \( q \) when \( B(p) \) and \( C(q) \) hold is called the stage order.

To understand the simplifying assumptions better, consider the set of trees given in (1.8) of order up to 4.

**Example 1.2**  **Reduced order conditions when \( B(4) \) and \( C(2) \) hold**

The \( B(4) \) conditions reduce the order conditions to the set of trees to

\[
\begin{array}{c}
\begin{array}{c}
\text{\textbullet} \\
\text{\textbullet} \\
\text{\textbullet}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\text{\textbullet} \\
\text{\textbullet} \\
\text{\textbullet}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\text{\textbullet} \\
\text{\textbullet} \\
\text{\textbullet}
\end{array}
\end{array}

\text{\textbullet}
\end{array}
\]

The \( C(2) \) conditions further reduce the set to

\[
\begin{array}{c}
\begin{array}{c}
\text{\textbullet}
\end{array}
\end{array}
\]

Hence only one order condition is to be satisfied, \( b^T A c^2 = \frac{1}{12} \).

In the following section, we discuss some of the important properties of Runge-Kutta methods for solving stiff problems.
Table 1.1: Some properties of trees up to order 5

<table>
<thead>
<tr>
<th>Tree</th>
<th>Order Conditions</th>
<th>Order</th>
<th>Density</th>
<th>Order Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>$b^T e = 1$</td>
<td>$t_{11}$</td>
<td>1</td>
<td>$\Phi(t) = \frac{1}{20}$</td>
</tr>
<tr>
<td>I</td>
<td>$b^T c = 1/2$</td>
<td>$t_{21}$</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>( \vee )</td>
<td>$b^T c^2 = 1/3$</td>
<td>$t_{31}$</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( \vee )</td>
<td>$b^T Ac = 1/6$</td>
<td>$t_{32}$</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T c^3 = 1/4$</td>
<td>$t_{41}$</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T CAc = 1/8$</td>
<td>$t_{42}$</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T Ac^2 = 1/12$</td>
<td>$t_{43}$</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T A^2c = 1/24$</td>
<td>$t_{44}$</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T c^4 = 1/5$</td>
<td>$t_{51}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T C^2Ac = 1/10$</td>
<td>$t_{52}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T CAc^2 = 1/15$</td>
<td>$t_{53}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T CA^2c = 1/30$</td>
<td>$t_{54}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T (A(c))^2 = 1/20$</td>
<td>$t_{55}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T Ac^3 = 1/20$</td>
<td>$t_{56}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T ACAc = 1/40$</td>
<td>$t_{57}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T A^2c^2 = 1/60$</td>
<td>$t_{58}$</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>( \psi )</td>
<td>$b^T A^3c = 1/120$</td>
<td>$t_{59}$</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
1.2.3 Stability and stiffness

To understand the stability of Runge-Kutta methods we consider applying the method (1.5) to the Dahlquist test problem.

\[ y' = \lambda y, \quad y(x_0) = y_0, \quad y(x) = y_0 e^{\lambda x}, \quad (1.12) \]

where \( \lambda \) is a complex number with a negative real part. The exact solution decays when \( \lambda < 0 \) and grows exponentially if \( \lambda > 0 \).

Applying an \( s \)-stage Runge-Kutta method yields,

\[
\begin{align*}
Y &= e y_{n-1} + h A(\lambda Y) = (I - h \lambda A)^{-1} e y_{n-1}, \\
y_n &= y_{n-1} + h b^T(\lambda Y) = (1 + h \lambda b^T(I - h \lambda A)^{-1} e)y_{n-1}, \\
&= R(h \lambda) y_{n-1}.
\end{align*}
\]

The stability function is therefore defined by

\[ R(z) = 1 + z b^T (I - z A)^{-1} e. \quad (1.13) \]

Another approach is due to Stetter [81]. Applying (1.5) to (1.12) and solving for \( y_n \) by using Cramer’s rule yields \( y_n = (N/D)y_{n-1} \) where \( N = \det(I - zA + z e b^T) \) and \( D = \det(I - zA) \). Thus we have \( y_n = R(z) y_{n-1} \), where

\[
R(z) = \frac{\det(I - zA + z e b^T)}{\det(I - zA)} = \frac{N(z)}{D(z)}. \quad (1.14)
\]

Consider the more general problem of solving the linear system of ODEs,

\[ y' = \Omega y, \quad y(0) = y_0, \]

where \( \Omega \) is a \( n \times n \) constant matrix. Assume \( \Omega \) has distinct eigenvalues such that \( |\lambda_1| < |\lambda_2| < \ldots < |\lambda_n| \) with corresponding eigenvectors \( u_1, u_2, \ldots, u_n \), that is, \( \Omega u_i = \lambda_i u_i \) for \( i = 1, \ldots, n \). Then

\[ \Omega S = \Omega[u_1, u_2, \ldots, u_n] = [\lambda_1 u_1, \lambda_2 u_2, \ldots, \lambda_n u_n] = [u_1, u_2, \ldots, u_n] \Lambda = S \Lambda, \]

where \( \Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \) and hence \( \Omega = S \Lambda S^{-1} \), that is, \( \Omega \) is similar to the
diagonal matrix $\Lambda$. Now the system of ODEs becomes

$$S^{-1}y' = S^{-1}\Omega SS^{-1}y.$$ 

Now let $z = S^{-1}y$, then $z' = \Lambda z$. This decouples to $n$ scalar equations because the matrix $\Lambda$ is diagonal,

$$z_i = \lambda_i z_i, \quad z_i(0) = z_{i0}, \quad i = 1, \ldots, n,$$

with exact solution $z_i = z_{i0}e^{\lambda_i x}$. The components of $y(x)$ are linear combinations of these exponential functions. The behaviour of the system depends on the solutions to the individual scalar equations. The system is said to be decoupled. When $\text{Re}(\lambda_i)$ is small and negative, it corresponds to a slowly decaying component, whereas when $\text{Re}(\lambda_i)$ is large and negative it corresponds to a rapidly decaying component. If the negative eigenvalues of $\Omega$ include small as well as large magnitudes there would be rapidly decaying components in the presence of slowly decaying components. The slowly decaying components are usually the ones of interest, and ideally, we would like to be able to use a stepsize that is determined by the accuracy we desire. However, the presence of the rapidly decaying components has a detrimental effect even though they may have decayed to almost zero. The stepsize used by an explicit numerical method therefore has to be chosen much smaller to ensure the numerical solution of the rapidly decaying components do not grow unboundedly. A problem is said to be stiff if the presence of the large negative $\text{Re}(\lambda_i)$’s results in using a much smaller stepsize as requested by stability rather than by accuracy alone. Therefore, the test equation given in (1.12) by itself is not a stiff equation even with large negative $\lambda$ because we have no choice but to use a small stepsize to solve it to a given accuracy. The discussion given above describes the concept of stiffness when solving a linear system of ODEs.

Implicit methods are suitable for solving stiff problems because they can be $A$-stable. Hence we only need to adjust the stepsize of an $A$-stable method for reasons of accuracy rather than stability. We now discuss the concept of $A$-stability.

When solving stiff problems, we want the method to have good stability region. What does a good stability mean? Dahlquist [30] studied the stability behaviour of the trapezoidal formula and showed that it has the smallest truncation error among all multistep methods. He derived the error bounds under some requirements. The requirements are defined here:
Definition 1.3 The stability region $S$ is defined by

$$S = \{ z \in \mathbb{C} : |R(z)| \leq 1 \}.$$ 

Definition 1.4 A method whose stability region contains the whole left half-plane is called A-stable (Dahlquist 1963, [30]).

Not all Runge-Kutta methods are A-stable. Dahlquist proved that the classical 4-th order Runge-Kutta method is not A-stable and neither are all the other explicit methods. Explicit methods cannot be A-stable because their stability functions are polynomials. On the other hand, implicit methods have stability function that are rational functions and there is a possibility that $|R(z)| \leq 1$ holds for $z \in \mathbb{C}^-$. In addition to the stability concept given by Dahlquist, there is also A$(\alpha)$-stability which was proposed by Widlund [85] in 1967. The definition of A$(\alpha)$ stability given by Widlund is as follows:

Definition 1.5 A method is said to be A$(\alpha)$-stable, $\alpha \in (0, \pi/2)$ if

$$S_{\alpha} \supseteq \{ z \in \mathbb{C} : -\alpha < \pi - \arg(z) < \alpha \},$$

where $S_{\alpha}$ is the region of absolute stability of the method (Widlund, 1967).

A-stability can be determined by applying the maximum modulus principle. The maximum modulus principle given in [15] states that, if $R(z)$ is analytic on $\mathbb{C}^-$ then $R(z)$ attains its maximum modulus on the boundary of $\mathbb{C}^-$. In order to show $R(z)$ is bounded by 1 in $\mathbb{C}^-$, we need to show that all poles of $R(z)$ are in the right half-plane and that $|R(iy)| \leq 1, y \in \mathbb{R}$, that is $E(y) \geq 0$. The function $E(y)$ is known as the $E$-polynomial and is defined by

$$E(y) = |D(iy)|^2 - |N(iy)|^2,$$

where $R(z) = N(z)/D(z)$.

Example 1.3 The 3-stage Radau IIA method.

Consider the 3-stage Radau IIA method given in Table 1.4 (c). The stability function is given by

$$R(z) = \frac{1 + \frac{2z}{5} + \frac{z^2}{20}}{1 - \frac{3z}{5} + \frac{3z^2}{20} - \frac{z^3}{60}}.$$
The poles of $R(z)$ lie in the right half plane and so $R(z)$ is analytic in $\mathbb{C}^-$. The $E$-polynomial yields

$$E(y) = |D(iy)|^2 - |N(iy)|^2,$$

$$= \left(1 - \frac{3}{20}y^2\right)^2 + \left(-\frac{3}{5}y + \frac{1}{60}y^3\right)^2 - \left(1 - \frac{1}{20}y^2\right)^2 - \frac{4}{25}y^2 = \frac{1}{3600}y^6.$$ 

Since $|E(iy)| \geq 0$, for all $y \in \mathbb{R}$, it follows from the maximum modulus principle that the method is $A$-stable. □

In addition to the $A$-stability and $A(\alpha)$ stability, Ehle [37] has introduced a stronger stability property called $L$-stability.

**Definition 1.6** A method is said to be $L$-stable if it is $A$-stable and if, in addition,

$$\lim_{z \to \infty} R(z) = 0. \quad (1.16)$$

For implicit methods to be $L$-stable, the degree of the numerator must be less than the degree of the denominator. If the $A$-matrix is nonsingular then (1.16) is satisfied if $b^TA^{-1}e = 1$.

Next, we give an example of the behaviour of the explicit and implicit Euler methods applied to a stiff problem.

The concept of stiffness was introduced by Curtiss and Hirschfelder [29] in 1952. They realized that different types of methods work better on certain classes of problems. To understand the effect of stiffness, consider the Prothero-Robinson problem (1.4) with $g(x) = \sin(x)$ and $\lambda = -50$ at $x_n = 1$. Figure 1.2 shows the numerical behaviour of the explicit Euler and implicit Euler methods applied to the PR problem. For the explicit Euler method, we observe that as we increase the steps, the oscillations are slowly damped, whereas the implicit Euler method converges quickly to the exact solution even with a much larger stepsize. This is because the problem now depends on the stiffness parameter $\lambda$. Since $\lambda$ is a large negative value, the first terms decay very quickly, leaving $y(x) = g(x)$. Therefore, a problem is stiff if the ability to find reliable numerical approximations to its solution hinges on stability rather than accuracy alone, and when some components of the solution decay much more rapidly than others. For example, the explicit Euler is not $A$-stable or even $A(\alpha)$ stable for any $\alpha < \pi/2$. This is true for all explicit methods since none of the explicit can be $A(\alpha)$-stable. Therefore, none of the
explicit methods are suitable for solving stiff problems. A detailed discussion of stiffness is given by Shampine and Gear in [79]. In this paper, they discuss ways of determining whether equations are stiff and the importance of stiff equations. Furthermore they comment that if the system is known to be very stable, the governing equations are likely to be stiff if some variables are known to change on time scales very different from others. Further reading on A-stability can be found in Butcher’s paper “Forty-five years of A-stability” [16].

![Figure 1.2: Error behaviour of the explicit and implicit Euler methods for stiff problem.](image)

### 1.3 Implicit methods

Implicit methods were introduced by Butcher [11] in 1964, and play an important role in solving stiff problems and differential-algebraic equations. However, implicit methods are expensive and more difficult to implement as the stage value need to be determined by an iterative computation (refer Chapter 5, Section 5.1). Despite its difficulty in implementation [14], implicit methods give better stability and have fewer stages for the same order when compared to the explicit methods. The families of implicit methods are the Gauss methods, the Radau methods and the Lobatto methods.

#### The families of Gauss methods

An $s$-stage Gauss method, denoted by $\mathcal{G}_s$ satisfies $B(2s), C(s)$ and is of classical order $2s$. The abscissas are the zeros of the shifted Legendre polynomial $P_s(2x - 1)$, where $P_s(x)$
is the Legendre polynomial of degree $s$ defined on the interval $[-1, 1]$. The $A$ matrix can be constructed using the $C(s)$ condition defined in (1.11).

$$AS = A \begin{bmatrix} e & c & c^2 & \cdots & c^{s-1} \end{bmatrix},$$
$$= \begin{bmatrix} c & \frac{1}{2}c^2 & \frac{1}{3}c^3 & \cdots & \frac{1}{s}c^s \end{bmatrix} = CSD,$$
$$A = CSDS^{-1},$$

where $D = \text{diag}(1, \frac{1}{2}, \frac{1}{3}, \ldots, \frac{1}{s})$. Similarly, we construct the vector $b$ using the $B(s)$ conditions.

$$b^T S = b^T \begin{bmatrix} e & c & c^2 & \cdots & c^{s-1} \end{bmatrix} = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{s} \end{bmatrix} = (De)^T,$$
$$b^T = e^T DS^{-1},$$

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

Examples of Gauss methods are the implicit midpoint rule $G_1$, the 2-stage Gauss $G_2$ and the 3-stage Gauss $G_3$. Gauss methods are A-stable, but not L-stable.

Next, we give the construction of the $A$ matrix and the $b$ vector of the Gauss methods using the formulas given above.

**Example 1.4 2-stage Gauss method**

For $s = 2$, the Legendre polynomial gives $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ and $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$.

$$AS = CSD = \begin{bmatrix} \frac{1}{2} - \frac{\sqrt{3}}{6} & 0 & \frac{1}{2} - \frac{\sqrt{3}}{6} \\ 0 & \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{2} + \frac{\sqrt{3}}{6} \\ \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{2} + \frac{\sqrt{3}}{6} & 0 \end{bmatrix},$$

$$A = \begin{bmatrix} \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{12} (\sqrt{3} - 3)^2 \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{12} (\sqrt{3} + 3)^2 \\ \frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{2} + \frac{\sqrt{3}}{6} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \end{bmatrix},$$

$$b^T = DS^{-1} = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} + \frac{\sqrt{3}}{6} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix}. $$
The stability function is given by

\[ R = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2}. \]

\[ \Box \]

The construction of \( A \) and \( b^T \) can be accomplished using Maple.

**Example 1.5 3-stage Gauss method**

For \( s = 3 \), the Legendre polynomial gives \( c_1 = \frac{1}{2} - \frac{\sqrt{15}}{10} \), \( c_2 = \frac{1}{2} \) and \( c_3 = \frac{1}{2} + \frac{\sqrt{15}}{10} \):

\[
AS = CSD = \begin{bmatrix}
\frac{1}{2} - \frac{\sqrt{15}}{10} & 0 & 0 \\
0 & \frac{1}{2} & 0 \\
0 & 0 & \frac{1}{2} + \frac{\sqrt{15}}{10}
\end{bmatrix}
\begin{bmatrix}
1 & \frac{1}{2} - \frac{\sqrt{15}}{10} & (\frac{1}{2} - \frac{\sqrt{15}}{10})^2 \\
1 & \frac{1}{2} & \frac{3}{4} \\
1 & \frac{1}{2} + \frac{\sqrt{15}}{10} & (\frac{1}{2} + \frac{\sqrt{15}}{10})^2
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0
\end{bmatrix}.
\]

\[ A = \begin{bmatrix}
\frac{5}{36} & \frac{2}{9} & \frac{5}{36} & \frac{\sqrt{15}}{30} \\
\frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{30} \\
\frac{5}{36} & \frac{2}{9} & \frac{5}{36}
\end{bmatrix},
\]

\[ b^T = DS^{-1} = \begin{bmatrix}
1 & \frac{1}{2} & \frac{1}{3}
\end{bmatrix}
\begin{bmatrix}
1 & \frac{1}{2} - \frac{\sqrt{15}}{10} & (\frac{1}{2} - \frac{\sqrt{15}}{10})^2 \\
1 & \frac{1}{2} & \frac{3}{4} \\
1 & \frac{1}{2} + \frac{\sqrt{15}}{10} & (\frac{1}{2} + \frac{\sqrt{15}}{10})^2
\end{bmatrix}^{-1} = \begin{bmatrix}
\frac{5}{18} & \frac{5}{9} & \frac{5}{18}
\end{bmatrix}.
\]

The stability function is given by

\[ R = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2 + \frac{1}{20}z^3}{1 - \frac{1}{2}z + \frac{1}{12}z^2 - \frac{1}{20}z^3}. \]

\[ \Box \]

The Gauss methods are summarized in Table 1.2.

**Table 1.2: Gauss methods of order 2, 4 and 6.**

<table>
<thead>
<tr>
<th>( \frac{1}{2} )</th>
<th>( \frac{1}{2} - \frac{\sqrt{15}}{10} )</th>
<th>( \frac{1}{2} + \frac{\sqrt{15}}{10} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{3}{8} )</td>
<td>( \frac{5}{36} )</td>
<td>( \frac{5}{36} )</td>
</tr>
<tr>
<td>( \frac{1}{2} - \frac{\sqrt{15}}{10} )</td>
<td>( \frac{5}{36} + \frac{\sqrt{15}}{24} )</td>
<td>( \frac{5}{36} )</td>
</tr>
<tr>
<td>( \frac{3}{8} )</td>
<td>( \frac{5}{36} )</td>
<td>( \frac{5}{36} )</td>
</tr>
</tbody>
</table>

(a) IMR (b) 2-stage Gauss (c) 3-stage Gauss
The families of Lobatto IIIA methods

For Lobatto IIIA methods, the abscissas are the zeros of \( P_s(2x - 1) - P_{s-2}(2x - 1) \). An \( s \)-stage method satisfies \( B(2s - 2), C(s) \) and \( D(s - 2) \) and is of order \( 2s - 2 \). Like the Gauss methods, the \( A \) matrix and the \( b \) vector are constructed using the \( C(s) \) and \( B(s) \) conditions respectively. Examples of Lobatto IIIA methods are the implicit trapezoidal rule \( \mathcal{L}_2 \), the 3-stage Lobatto IIIA method \( \mathcal{L}_3 \) and the 4-stage Lobatto IIIA method \( \mathcal{L}_4 \). The Lobatto IIIA methods are A-stable, but not L-stable.

**Example 1.6 3-stage Lobatto IIIA method**
For \( s = 3 \), the Legendre polynomial gives \( c_1 = 0, c_2 = \frac{1}{2}, \) and \( c_3 = 1 \):

\[
A = CSDS^{-1} = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & \frac{1}{2} & \frac{1}{2} \\
0 & 0 & 1 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 \\
1 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\]

\[
b^T = DS^{-1} = \begin{bmatrix}
1 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{bmatrix}
\]

The stability function is the same as 2-stage Gauss method.

**Example 1.7 4-stage Lobatto IIIA method**
For \( s = 4 \), the Legendre polynomial gives \( c_1 = 0, c_2 = \frac{1}{2} - \frac{\sqrt{5}}{10}, c_3 = \frac{1}{2} + \frac{\sqrt{5}}{10} \) and \( c_4 = 1 \):

\[
A = CSDS^{-1} = \begin{bmatrix}
\frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{1}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{1}{12}
\end{bmatrix}
\]

\[
b^T = DS^{-1} = \begin{bmatrix}
\frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{1}{12} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{1}{12}
\end{bmatrix}
\]

where \( A \) and \( b^T \) are obtained using Maple. The stability function is the same as the
3-stage Gauss method.

The Lobatto IIIA methods are summarized in Table 1.3.

Table 1.3: Lobatto IIIA methods of order 2, 4 and 6.

\[
\begin{array}{cccc|cccc|cccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\end{array}
\]

(a) ITR (b) 3-stage Lobatto IIIA (c) 4-stage Lobatto IIIA

Gauss and Lobatto IIIA methods are of special interest because they are symmetric. Symmetry of Runge-Kutta methods will be discussed in Chapter 2.

The families of Radau IIA methods

The s-stage Radau IIA method satisfies \( B(2s-1), C(s), D(s-1) \) and is of order \( 2s-1 \). The abscissas are the zeros of \( P_s(2x-1) - P_{s-1}(2x-1) \). The \( A \)-matrix is determined by the \( C(s) \) conditions while the weight vector is determined by the \( B(s) \) conditions as for the Gauss and Lobatto IIIA methods. There is also Lobatto IIC methods of Chipman [26] satisfy the simplifying assumptions \( B(2s-2), C(s-1) \) and \( D(s-1) \) of order \( 2s-2 \). The Radau IIA and Lobatto IIC methods are defined in Table 1.4 and Table 1.5 respectively.

Table 1.4: Radau IIA methods of order 1, 3 and 5.

\[
\begin{array}{cccc|cccc|cccc}
1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
\frac{1}{4} & \frac{5}{12} & \frac{1}{4} & \frac{5}{12} & \frac{1}{4} & \frac{5}{12} & \frac{1}{4} & \frac{5}{12} & \frac{1}{4} & \frac{5}{12} \\
1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\end{array}
\]

(a) Implicit Euler (b) 2-stage Radau IIA (c) 3-stage Radau IIA
1.4 Objectives and research contributions

Richardson’s idea of accelerating the convergence of a sequence in 1927 has led to methods for numerical quadrature by Romberg [70] in 1955 and ODEs [49, 9, 62, 32]. This technique, which is known as extrapolation, has been successfully implemented by many researchers. However, extrapolation is more efficient if the method has an asymptotic error expansion in even powers of the stepsize $h$ since the order will be increased by two at a time. The first person to study this asymptotic error expansion was William Gragg [49]. He showed that the order-2, explicit midpoint rule has $h^2$-asymptotic error expansion and was a major achievement in the solution of ordinary differential equations. Another important contribution by Gragg was his use of smoothing. The explicit midpoint rule is a two-step method that uses the Euler method as a starter. It has a parasitic growth component in the solution that will eventually destroy the numerical solutions. To overcome this problem, Gragg introduced smoothing to dampen the errors due to the parasitic component in the solution. Gragg’s smoothing formula has been used by many researchers in the solution of nonstiff as well as stiff problems. We consider details of extrapolation and smoothing in Chapter 2, Section 2.2.

In the late 80’s, Chan [20] extended the theoretical study of extrapolation to arbitrary Runge-Kutta methods for stiff ODEs. He examined the class of symmetric Runge-Kutta methods which possess $h^2$-asymptotic error expansions and generalized the idea of symmetry to composite Runge-Kutta methods that preserve the $h^2$-error expansion and also provide the necessary damping. These methods enjoy the same advantage in increasing the order by two at a time on successive extrapolations. However, for stiff problems, the asymptotic error expansion has coefficients that depend on the stiffness parameter and as stiffness increases the coefficients get larger and the numerical solution

Table 1.5: Lobatto IIIC methods of order 2, 4 and 6.

<table>
<thead>
<tr>
<th></th>
<th>2-stage Lobatto IIIC</th>
<th>3-stage Lobatto IIIC</th>
<th>4-stage Lobatto IIIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
</tr>
<tr>
<td>1/6</td>
<td>2/3</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1/6</td>
<td>2/3</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1/6</td>
<td>2/3</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1/6</td>
<td>2/3</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1/6</td>
<td>2/3</td>
<td>1/16</td>
<td>1/16</td>
</tr>
<tr>
<td>1/6</td>
<td>2/3</td>
<td>1/16</td>
<td>1/16</td>
</tr>
</tbody>
</table>
may grow unboundedly and be destroyed eventually. Chan generalized the smoothing formula which is known as symmetrization for higher order symmetric methods giving a complete theory on symmetrization and extrapolation for the solution of stiff problems. Chan proposed a method known as symmetrizer that is L-stable. With this symmetrizer, he showed that certain extrapolations can be A-stable when performed with active extrapolation [24]. The advantages of using symmetrizers will be given in Chapter 2, Section 2.3.

The main idea of this thesis is to extend the theoretical study of extrapolation and symmetrization by Chan to the practical implementation and numerical study of higher order symmetric Runge-Kutta methods. We are particularly interested in the Gauss and Lobatto IIIA methods of orders 4 and 6. We wish to investigate the two modes of symmetrization in the constant and variable stepsize settings. In the active mode, the symmetrized value is propagated whenever symmetrization is applied. On the other hand, in the passive mode, symmetrization is applied without propagating the symmetrized value. The active and passive modes of applying symmetrization is shown to be efficient for linear and nonlinear problems.

This thesis focusses on the implementation of symmetrization and extrapolation in passive and active modes for various stiff problems. We have performed numerical experiments on the STIFF DETEST problems (see Appendix B.2). Although the idea of extrapolation is not new, we wish to investigate the extent of obtaining greater accuracy/efficiency when extrapolation is carried out with either passive or active symmetrization.

In this research, we attempt to address several questions of interest. Some of these are

1. How do symmetrizers behave in the passive and active mode for stiff problems?

2. Which mode of symmetrization is more efficient when applied with extrapolation?

3. What type of problems are suitable for solution with extrapolated symmetrized methods?

4. How is the symmetrized Runge-Kutta method implemented with stepsize control?

In Chapter 7, we present answers to the above questions.
1.5 Thesis outline

There are 7 chapters in this thesis.

Chapter 2 is divided into three main parts. In first part, we discuss the symmetry of RK methods such as the permutation and adjoint methods. They are used when we construct symmetrizers given in Chapter 3. Moreover, although symmetric methods enjoy some nice properties, there are disadvantages which we address. The main reason for studying symmetric methods is that their numerical solutions possess asymptotic error expansions in even powers of the stepsize which can therefore be exploited by the acceleration technique of extrapolation. Hence we will discuss Richardson extrapolation in the second part of this chapter. We give a brief history of applications to nonstiff and stiff problems. An example of a symmetric method that has been successfully used in solving nonstiff problems together with extrapolation is the explicit midpoint rule. This has lead to the development of the smoothing proposed by Gragg, which lead to the generalization of smoothing (symmetrization) for higher order symmetric methods by Chan. Smoothing of the IMR and the ITR also will be discussed in this chapter.

In Chapter 3, we give a detailed description of symmetrization, the construction and the order conditions of symmetrizers for the Gauss and Lobatto IIIA methods of orders 4 and 6.

Symmetrization can be applied in two modes; active and passive. In Chapter 4, we discuss the two modes. The active mode is interesting because symmetrization can be applied at every step, every two steps and so on. We discuss theoretically the order behaviour of active and passive symmetrization for the PR problem. In addition, we provide numerical experiments applied to the test problems, and investigate the performance of active and passive symmetrization for various degrees of stiffness, different intervals and optimal choice of applying active symmetrization.

Chapter 5 discusses implementation issues concerning constant and variable stepsize use. In all our numerical experiments, we solve the nonlinear equations using simplified Newton. In the case variable stepsize setting, details on estimating the local error is given and also suggest that symmetrization can be used as an alternative way of estimating the error. A variable stepsize code is implemented using the approach of Hairer and Wanner in [53].

Chapter 6 gives the numerical results of active and passive symmetrization with ex-
Chapter 1. Introduction and Motivation

Temporal interpolation for constant stepsize and variable stepsize settings. In the constant stepsize setting, numerical results are given for various linear and nonlinear problems. The problems are taken from the well known literature and are listed in Appendix B.1. In the variable stepsize setting, we discuss some implementation issues and choice of error estimation. Numerical results are presented for the STIFF DETEST problems given in [38].

Finally, Chapter 7 summarizes the numerical results and presents some conclusions and suggestions for future work.
In this chapter, we study the symmetry of RK methods. The concept of symmetry includes permutation of the stages and the definition of the adjoint method. A symmetric method is of special interest because its numerical solutions possesses an asymptotic error expansion in even powers of the stepsize. This property can therefore be exploited by acceleration techniques such as extrapolation to increase the order by two at a time. Although symmetric methods can have high classical order and can be A-stable, they suffer from the phenomenon of order reduction when applied to stiff problems. Besides the properties of symmetric Runge-Kutta methods, we study the order reduction phenomenon in detail. We also give a brief history of extrapolation and applications to nonstiff and stiff ODEs which has lead to the generalization of smoothing for the IMR and ITR known as symmetrization.
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

2.1 Symmetric Runge-Kutta methods

In the following subsection, we derive the conditions of symmetry for RK methods that will be used in Chapter 3.

2.1.1 Symmetry of RK methods

The symmetry of RK methods is defined in terms of permutation of the stages and the adjoint method. They are important for the construction of symmetric method and its symmetrizer, as discussed in Chapter 3.

Theorem 2.1 If the coefficients of a RK method satisfy the conditions,

\[(a)\quad a_{s+1-i,s+1-j} = b_{s+1-j} - a_{ij}, \quad i, j = 1, \ldots, s, \quad \Leftrightarrow \quad PAP = cb^T - A;\]
\[(b)\quad b_{s+1-j} = b_j, \quad j = 1, \ldots, s, \quad \Leftrightarrow \quad Pb = b,\]

where \(P\) is the permutation matrix with elements \(p_{ij} = \delta_{i,s+1-j}\), then the method is symmetric or self-adjoint.

Proof:
For the forward step, \((x_{n-1}, y_{n-1}) \mapsto (x_n, y_n)\) with stepsize \(h\), we have

\[Y_i = y_{n-1} + h \sum_j a_{ij} f(x_{n-1} + c_j h, Y_j), \quad i = 1, \ldots, s,\]
\[y_n = y_{n-1} + h \sum_j b_j f(x_{n-1} + c_j h, Y_j).\]

Now consider the step \((x_n, y_n) \mapsto (x_{n-1}, \overline{y}_{n-1})\) applied with stepsize \(-h\) having stage values given by

\[\overline{Y}_i = y_n - h \sum_j a_{ij} f(x_n - c_j h, \overline{Y}_j), \quad i = 1, \ldots, s,\]
\[= y_{n-1} + h \sum_j \left( b_{s+1-j} f(x_{n-1} + c_j h, Y_{s+1-j}) - a_{ij} f(x_n - c_j h, \overline{Y}_j) \right),\]

where the update from the forward step is used to substitute for \(y_n\) in terms of \(y_{n-1}\) and reordering the stages by \(j \leftrightarrow s + 1 - j\) in the summation over \(j\). The reordering in the
forward step leads to the equations
\[ Y_{s+1-i} = y_{n-1} + h \sum_j a_{s+1-i,s+1-j} f(x_{n-1} + c_{s+1-j}h, Y_{s+1-j}), \quad i = 1, \ldots, s, \]
\[ y_n = y_{n-1} + h \sum_j b_{s+1-j} f(x_{n-1} + c_{s+1-j}h, Y_{s+1-j}). \]

Taking the difference between the stage values and using the conditions (a) and (b), we obtain
\[ Y_i - Y_{s+1-i} = h \sum_j a_{ij} \left( f(x_{n-1} + c_{s+1-j}h, Y_{s+1-j}) - f(x_n - c_jh, \bar{Y}_j) \right). \]

Taking norms with the triangular inequality and assuming the Lipschitz condition for \( f \) then shows \( Y_i = Y_{s+1-i} \) for all \( i = 1, \ldots, s \).

Making the same substitution for \( y_n \) in terms of \( y_{n-1} \) with reordering for the update, we have
\[ \bar{Y}_{n-1} = y_n - h \sum_j b_j f(x_n - c_jh, \bar{Y}_j) \]
\[ = y_{n-1} + h \sum_j \left( b_{s+1-j} f(x_{n-1} + c_{s+1-j}h, Y_{s+1-j}) - b_j f(x_n - c_jh, \bar{Y}_j) \right). \]

The conditions (a) and (b) together with the result \( \bar{Y}_i = Y_{s+1-i} \) for all \( i = 1, \ldots, s \) then shows \( \bar{Y}_{n-1} = y_{n-1} \). Thus the method is self-adjoint or symmetric. ■

The adjoint method

If \( \mathcal{R} = (A, b, c) \) is the method mapping \( (x_{n-1}, y_{n-1}) \to (x_n, y_n) \) with stepsize \( h \), then the adjoint of \( \mathcal{R} \) is the method \( \mathcal{R}^* \) mapping \( (x_n, y_n) \to (x_{n-1}, y_{n-1}) \) with stepsize \( -h \).
\[ \bar{Y}_i = y_n - h \sum_j a_{ij} f(x_n - c_jh, \bar{Y}_j), \quad i = 1, \ldots, s, \]
\[ y_{n-1} = y_n - h \sum_j b_j f(x_n - c_jh, \bar{Y}_j). \]
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

Rearranging the equations, we have

\[ \overline{Y}_i = y_{n-1} + h \sum_j (b_j - a_{ij}) f(x_{n-1} + (1 - c_j)h, \overline{Y}_j), \quad i = 1, \ldots, s, \]

\[ y_n = y_{n-1} + h \sum_j b_j f(x_{n-1} + (1 - c_j)h, \overline{Y}_j). \]

We can interpret the adjoint method as the method that maps \((x_{n-1}, y_{n-1}) \rightarrow (x_n, y_n)\) with stepsize \(h\) in the forward direction.

It is convenient to cast the definition of a RK method in matrix form. From (1.5), we have

\[ Y = e \otimes y_{n-1} + h(A \otimes I_N)F(Y), \]

\[ y_n = y_{n-1} + h(b^T \otimes I_N)F(Y), \quad (2.1) \]

where \(Y\) and \(F\) are the \(sN \times 1\) vectors of the internal stage values and the stage derivatives respectively, \(e = (1, \ldots, 1)^T\), and \(\otimes\) denotes the Kronecker product of matrices \(A \otimes B = (a_{ij}B)\).

Now from the coefficients of the method we see that \(R^* = (eb^T - A, b, e - c)\). Hence, a self-adjoint method is one when \(R^* = R\). If \(P\) is the permutation matrix which interchanges the stages first with last, second with second-last, etc., then \((A, b, c) = (PAP, Pb, Pc)\) and therefore, \((eb^T - A, b, e - c) = (PAP, Pb, Pc)\) yields the conditions of symmetry,

\[ PA + AP = eb^T, \quad Pb = b, \quad Pc = e - c. \quad (2.2) \]

This adjoint method also satisfies the same simplifying assumptions (1.11) and thus have the same order \(2s\). Rudolf Scherer [74] has given a family of Radau and Lobatto type of Runge-Kutta methods that have this property (adjoint method) which is known as reflected methods in his context. Examples of methods that have this property are Radau IA (reflection of Radau II), Radau IIA (reflection of Radau I) and Lobatto IIIC (reflection of Lobatto III).

In the following section, we wish to study a useful technical property of Runge-Kutta methods \(R\) with \(s\)-stages which allows possible reduction to an equivalent method \(\overline{R}\) with a smaller number of stages \(r(< s)\). The method is reducible if there exists a method...
with \( r(< s) \) stages such that any solution for the \( r \) stage method is also a solution for the original method \([27]\). There are two types of reduction, \( DJ \)-reducible (Dahlquist and Jeltsch, 1979 \([33]\)) and \( S \)-reducible, due to Hundsdorfer and Spijker \([57]\). In this thesis, we only need \( S \)-reducibility.

### 2.1.2 \( S \)-Reducibility

**Definition 2.1** A Runge-Kutta method is \( S \)-reducible, if for some partition \((S_1, \ldots, S_r)\) of \( \{1, \ldots, s\} \) with \( r < s \) we have for all \( l \) and \( m \)

\[
\sum_{k \in S_m} a_{ik} = \sum_{k \in S_m} a_{jk}, \text{ if } i, j \in S_l.
\]

Otherwise it is called \( S \)-irreducible (Hundsdorfer and Spijker \([57]\)).

The definition above simply tells us that if the \( i \)-th and the \( j \)-th rows of \( A \) are identical that is \( a_{ik} = a_{jk} \) for all \( k = 1, 2, \ldots, s \), the \( j \)-th stage may be removed. The number of stages is then reduced to \( s - 1 \) with \( a_{ki} \) replaced by \( a_{ki} + a_{kj} \) for all \( k \neq j \) and \( b_i \) replaced by \( b_i + b_j \).

**Example 2.1 \( S \)-reducibility**

Consider a method that has the Butcher tableau given below:

\[
\begin{array}{c|cccccc}
0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\
1 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
\end{array}
\]

We can see that the 2nd and 3rd rows are identical. Therefore, removing the third row,
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

adding \(a_{42} + a_{43}\) and \(b_2 + b_3\) gives

\[
\begin{array}{ccc}
0 & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
1 & \frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\
\end{array}
\]

(2.3)

Thus by \(S\)-reducibility, the method reduces to 3 stages.

The method given above is the composition of two ITRs. Next, we introduce the operations of scalar multiplication, composition and inverse method.

2.1.3 Operations on RK methods

In this section, we introduce several concepts: scalar multiplication when a method is applied over a multiple of the stepsize \(h\); composition when one method is applied after another. In what follows we denote \(y_{n-1}\) by \(y_0\) and \(y_n\) by \(y_1\) for simplicity.

Definition 2.2 If \(\gamma \in \mathbb{R}\) and \(\mathcal{R} = [(A, b, c)]\) is a RK method then scalar multiplication of \(\mathcal{R}\) by \(\gamma\), denoted by \(\gamma \mathcal{R}\), is defined to be the method \([(\gamma A, \gamma b, \gamma c)]\).

Definition 2.3 If \(\mathcal{R}_1 = (A_1, b_1, c_1)_{s_1}\) and \(\mathcal{R}_2 = (A_2, b_2, c_2)_{s_2}\) are two RK methods applied with the same stepsize in succession, the composite method \(\mathcal{R}_1 \circ \mathcal{R}_2\) is defined by the Butcher tableau

\[
\begin{array}{c|cc}
  c_1 & A_1 & O \\
  c_2 + e_2 & e_2b_1^T & A_2 \\
\hline
  & b_1^T & b_2^T \\
\end{array}
\]

(2.4)

where \(e_i\) is the \(s_i \times 1\) vector of units for \(i = 1, 2\).
2.1. Symmetric Runge-Kutta methods

Example 2.2 Two IMRs
The composite of two IMRs is given by

\[
\mathcal{R} = \frac{1}{2}(\mathcal{G}_1 \circ \mathcal{G}_1) = \begin{bmatrix}
\frac{1}{4} & \frac{1}{4} & 0 \\
\frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\
\frac{1}{2} & \frac{1}{2}
\end{bmatrix}.
\] (2.5)

This method satisfies \( B(2) \) and \( C(1) \) and is of order 2. It is also symmetric and has the stability function given by

\[
R(z) = \left(\frac{1 + \frac{z}{4}}{1 - \frac{z}{4}}\right)^2. \quad \Box
\] (2.6)

Example 2.3 Two \( \mathcal{G}_2 \)

\[
\mathcal{R} = \frac{1}{2}(\mathcal{G}_2 \circ \mathcal{G}_2) = \begin{bmatrix}
\frac{1}{4} - \frac{\sqrt{3}}{12} & \frac{1}{8} & \frac{1}{8} - \frac{\sqrt{3}}{12} & 0 & 0 \\
\frac{1}{4} + \frac{\sqrt{3}}{12} & \frac{1}{8} + \frac{\sqrt{3}}{12} & \frac{1}{8} & 0 & 0 \\
\frac{3}{4} - \frac{\sqrt{3}}{12} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} - \frac{\sqrt{3}}{12} \\
\frac{3}{4} + \frac{\sqrt{3}}{12} & \frac{1}{4} & \frac{1}{4} & \frac{1}{8} + \frac{\sqrt{3}}{12} & \frac{1}{8}
\end{bmatrix}
\] (2.7)

This method satisfies \( B(4) \) and \( C(2) \) and is of order 4. It is symmetric and has the stability function

\[
R(z) = \left(\frac{1 + \frac{z}{4} + \frac{z^2}{48}}{1 - \frac{z}{4} + \frac{z^2}{48}}\right)^2. \quad \Box
\] (2.8)

The use of composition in Runge-Kutta methods was first introduced by Butcher [12] in the study of effective order. Stetter [81] studied the algebraic structure and properties of a set \( \mathcal{R} \) of Runge-Kutta methods under two operations, composition and scalar multiplication. This composite method has led to the more general concept of B-series for the ODE's case (see [15], page 280 or [52] page 264).
CHAPTER 2. EXTRAPOLATION OF SYMMETRIC RUNGE-KUTTA METHODS

Inverse method

If the method $\mathcal{R} = (A, b, c)$ maps $(x_{n-1}, y_{n-1}) \rightarrow (x_n, y_n)$ with stepsize $h$, the inverse method $\mathcal{R}^{-1}$ is defined as the method that maps $(x_n, y_n) \rightarrow (x_{n-1}, y_{n-1})$ applied with stepsize $-h$. In the forward step, we have

$$Y_i = y_{n-1} + h \sum_j a_{ij} f(x_{n-1} + c_j h, Y_j), \quad i = 1, \ldots, s,$$

$$y_n = y_{n-1} + h \sum_j b_j f(x_{n-1} + c_j h, Y_j).$$

Solving for $y_{n-1}$ in terms of $y_n$, we have

$$Y_i = y_n + h \sum_j (a_{ij} - b_j) f(x_n + (c_j - 1) h, Y_j), \quad i = 1, \ldots, s,$$

$$y_{n-1} = y_n + h \sum_j (-b_j) f(x_n + (c_j - 1) h, Y_j),$$

or, equivalently,

$$Y_{s+1-i} = y_n + h \sum_j (a_{s+1-i, s+1-j} - b_{s+1-j}) f(x_n + (c_{s+1-j} - 1) h, Y_{s+1-j}), \quad i = 1, \ldots, s,$$

$$y_{n-1} = y_n + h \sum_j (-b_{s+1-j}) f(x_n + (c_{s+1-j} - 1) h, Y_{s+1-j}).$$

Hence the method starts at $(x_n, y_n)$, and computes the internal stages $Y_1, Y_2, \ldots, Y_s$ at $x_n - (1 - c_s) h, x_n - (1 - c_{s-1}) h, \ldots, x_n - (1 - c_1) h$ respectively to finally give $(x_{n-1}, y_{n-1})$. If $P$ is the permutation matrix that reverses the stages then, in matrix form, we have

$$PY = ey_n + h \left( PAP - e(Pb)^T \right) F(x_n e + (Pc - e) h, PY),$$

$$y_{n-1} = y_n + h(-Pb)^T F(x_n e + (Pc - e) h, PY).$$

The inverse method $\mathcal{R}^{-1}$ is therefore generated by $\left( PAP - e(Pb)^T, -Pb, Pc - e \right)$ which is equivalent to the method $(A - eb^T, -b, c - e)$. It hence follows that the adjoint method $\mathcal{R}^* = (eb^T - A, b, e - c)$ is equivalent to $-\mathcal{R}^{-1}$.

We have defined some concepts of symmetric methods that will be used in Chapter 3.

Next, we wish to discuss the advantages and disadvantages of using symmetric methods. Some advantages of symmetric methods are:
• Symmetric methods are $A$ stable. The stability function of a symmetric method can be shown to satisfy $R(z)R(-z) = 1$ which implies the method is $A$-stable. However, $R(\infty) = \pm 1$. So the symmetric method is not $L$-stable.

• Symmetric methods are suitable for solving Hamiltonian systems [72].

• Symmetric methods have asymptotic error expansions in even powers of $h$.

2.1.4 Asymptotic error expansion

A Runge-Kutta method has asymptotic error expansion given by

$$y_n(x) = y(x) + \tau_1(x)h^p + \tau_2(x)h^{p+1} + \ldots + \tau_{p+k}(x)h^{p+k} + O(h^{p+k+1}),$$

where the coefficients $\tau_i(x)$ are independent of $h$ [52].

If the method is symmetric [84], then the asymptotic error expansion is in even powers.

$$y_n(x) = y(x) + \tau_1(x)h^p + \tau_2(x)h^{p+2} + \ldots + \tau_{p+k}(x)h^{p+2k} + O(h^{p+2k+1}).$$

This asymptotic global error expansion in even powers of $h$ enables Richardson extrapolation (see Section 2.2) to increase the order by two at a time.

Stetter [81] proved the existence of such expansions for a very general class of discretization algorithms for non-linear functional equations (e.g. initial and boundary value problems for both ordinary and partial differential equations, integral equations and integro-differential equations). A theoretical analysis on the asymptotic error expansion for the IMR and ITR is given by Auzinger and Frank in [4].

Although symmetric methods enjoy some of these benefits, there are also some disadvantages. Symmetric methods suffer from the phenomenon of order reduction and have weak damping properties. We will study them in detail in the following subsection.

2.1.5 Order reduction

This phenomenon was studied by Prothero and Robinson [66, 53]. They have investigated the order behaviour for the family of linear model problems (1.4) and found that the
classical order of Gauss, Radau and Lobatto methods is not attained when the problem is very stiff \(\text{i.e } Re(\lambda) \ll 0\).

Consider applying a RK method to the PR problem (1.4). Let \(\lambda \in \mathbb{C}^-\) and write (1.4) as

\[ y'(x) = \lambda y(x) + \phi(x), \quad \phi(x) = g'(x) - \lambda g(x), \quad y(x_0) = y_0, \quad (2.9) \]

where \(g(x)\) is a smooth function and \(\lambda\) is the complex stiffness parameter with negative real part. The exact solution is given by

\[ y(x) = g(x) + (y_0 - g(x_0))e^{\lambda(x-x_0)}. \]

In the special case where \(y_0 = g(x_0)\), the exact solution is \(y(x) = g(x)\).

Applying an \(s\)-stage Runge-Kutta method (2.1) yields

\[ Y = ey_{n-1} + hF(Y) = ey_{n-1} + hA(\lambda Y + \Phi), \]

\[ = (I - zA)^{-1}ey_{n-1} + h(I - zA)^{-1}A\Phi, \]

where \(z = h\lambda\) and

\[
\Phi = \begin{bmatrix}
\phi(x_{n-1} + c_1h) \\
\phi(x_{n-1} + c_2h) \\
\vdots \\
\phi(x_{n-1} + c_sh)
\end{bmatrix}, \quad Y = \begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_s
\end{bmatrix}, \quad F(Y) = \begin{bmatrix}
f(x_{n-1} + c_1h, Y_1) \\
f(x_{n-1} + c_2h, Y_2) \\
\vdots \\
f(x_{n-1} + c_sh, Y_s)
\end{bmatrix} \quad \text{and} \quad e = \begin{bmatrix}
1 \\
1 \\
\vdots \\
1
\end{bmatrix}.
\]

The update is now given by

\[
y_n = y_{n-1} + hb^T F(Y), \\
= y_{n-1} + hb^T (\lambda Y + \Phi), \\
= y_{n-1} + zb^T ((I - zA)^{-1}ey_{n-1} + h(I - zA)^{-1}A\Phi) + hb^T \Phi, \\
= R(z)y_{n-1} + hb^T (I - zA)^{-1} \Phi. \quad (2.10)
\]

Replacing the numerical values \(y_{n-1}, Y\) and \(y_n\) by the exact values \(y(x_{n-1}), y(x_{n-1} + ch)\) and \(y(x_n)\) respectively, and introducing the corresponding residuals yields
Similarly, the residual for the update is also given by the Taylor series expansions

\[ y(x_{n-1} e + ch) = ey(x_{n-1}) + hA(\lambda y(x_{n-1} e + ch) + \Phi) + L_n, \]
\[ y(x_n) = y(x_{n-1}) + hb^T (\lambda y(x_{n-1} e + ch) + \Phi) + \ell_n, \]
\[ = y(x_{n-1}) + zb^T (I - zA)^{-1} ey(x_{n-1}) + h(I - zA)^{-1} A\Phi + (I - zA)^{-1} L_n + \ell_n + zb^T (I - zA)^{-1} L_n. \]  

(2.11)

The residual vector for the internal stages is given by the Taylor series expansions about \( x_{n-1} : \)

\[ L_n = y(x_{n-1} e + ch) - ey(x_{n-1}) - hAy'(x_{n-1} e + ch), \]
\[ = \left( ey(x_{n-1}) + chy'(x_{n-1}) + \frac{e^2 h^2}{2} y''(x_{n-1}) + \ldots + \frac{e^k h^k}{k!} y^{(k)}(x_{n-1}) + \ldots \right) \]
\[ - ey(x_{n-1}) - hA \left( ey'(x_{n-1}) + chy''(x_{n-1}) + \ldots + \frac{e^{k-1} h^{k-1}}{(k-1)!} y^{(k)}(x_{n-1}) + \ldots \right). \]

Simplifying using \( Ae = c \) gives

\[ L_n = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{n-1}) \left( c^k - kA c^{k-1} \right). \]  

(2.12)

Similarly, the residual for the update is also given by the Taylor series expansions

\[ \ell_n = y(x_{n-1} + h) - y(x_{n-1}) - hb^T y'(x_{n-1} e + ch), \]
\[ = \left( y(x_{n-1}) + hy'(x_{n-1}) + \frac{h^2}{2} y''(x_{n-1}) + \ldots + \frac{h^k}{k!} y^{(k)}(x_{n-1}) + \ldots \right) \]
\[ - y(x_{n-1}) + hb^T \left( ey'(x_{n-1}) + chy''(x_{n-1}) + \ldots + \frac{e^{k-1} h^{k-1}}{(k-1)!} y^{(k)}(x_{n-1}) \right), \]
\[ = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{n-1}) \left( 1 - kb^T c^{k-1} \right) \text{ since } b^T e = 1. \]  

(2.13)

The global error is \( \epsilon_n = y(x_n) - y_n \) and subtracting (2.10) from (2.11) gives

\[ \epsilon_n = R(z) \epsilon_{n-1} + \psi_n, \]  

(2.14)

where \( \psi_n(z) = \ell_n + zb^T (I - zA)^{-1} L_n. \)
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

Solving the inhomogeneous difference equation yields

$$
\epsilon_n = R(z)^n \epsilon_0 + \psi_n + R(z) \psi_{n-1} + R(z)^2 \psi_{n-2} + \ldots + R(z)^{n-1} \psi_1,
$$

$$
= \sum_{i=1}^{n} R(z)^{n-i} \psi_i(z) + R(z)^n \epsilon_0. \quad (2.15)
$$

The local error at step $i$ is magnified by the stability function $R(z)^{n-i}$ and contributes to the global error at step $n$.

The local error is given by

$$
\psi_i(z) = \ell_i + zb^T (I - zA)^{-1} L_i,
$$

$$
= \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{i-1}) \left(1 - kb^T c^{k-1} + zb^T (I - zA)^{-1} (c^k - kAc^{k-1}) \right). \quad (2.16)
$$

In (2.16), the term $1 - kb^T c^{k-1}$ vanishes for $k = 1, \ldots, p$ if $B(p)$ holds, while $c^k - kAc^{k-1}$ vanishes for $k = 1, \ldots, q$ if $C(q)$ holds.

Now, we wish to analyse the order behaviour of the Gauss and Lobatto IIIA methods.

The family of Gauss methods

The Gauss methods with $s$-stages satisfy $B(2s)$ and $C(s)$. These methods are given in Table 1.2 (see Chapter 1, Section 1.3).

In the nonstiff case when $|\lambda| = O(1)$ or $|z| = O(h)$,

$$
zb^T (I - zA)^{-1} (c^{(s+1)} - (s + 1)Ac^{s}) = zb^T (I + zA + z^2 A^2 + \ldots) (c^{(s+1)} - (s + 1)Ac^s),
$$

$$
= zb^T (c^{(s+1)} - (s + 1)Ac^s) + O(z^2) \text{ as } h \to 0.
$$

In the stiff case (that is, $|\lambda| \sim O(1/h^2)$ or $\frac{1}{|z|} = O(h)$),

$$
zb^T (I - zA)^{-1} (c^{(s+1)} - (s + 1)Ac^s) = -b^T A^{-1} \left( I - \frac{1}{z} A^{-1} \right)^{-1} (c^{(s+1)} - (s + 1)Ac^s),
$$

$$
= -b^T A^{-1} \left( I + \frac{1}{z} A^{-1} + \frac{1}{z^2} A^{-2} + \ldots \right) (c^{(s+1)} - (s + 1)Ac^s),
$$

$$
= -b^T A^{-1} (c^{(s+1)} - (s + 1)Ac^s) + O(\frac{1}{z}) \text{ as } h \to 0.
$$
We will analyse the implications for the local and global errors in the following examples.

**Example 2.4 1-stage Gauss method**

$\mathcal{G}_1$ is also known as the implicit midpoint rule (IMR). It satisfies $B(2), C(1)$ and is of order 2. The method has tableau given in Table 1.2 (a) (refer Chapter 1, Section 1.3).

In the nonstiff case, the local error is given by

$$\psi_i(z) = \frac{h^3}{24} y^{(3)}(x_{i-1}) - \frac{zh^2}{8} y^{(2)}(x_{i-1}) + O(h^4) \quad \text{as } h \to 0$$

In the stiff case, the local error is given by

$$\psi_i(z) = \frac{h^2}{4} y^{(2)}(x_{i-1}) - \frac{h^2}{2z} y^{(2)}(x_{i-1}) + O(h^3) \quad \text{as } h \to 0.$$ 

Therefore,

$$\psi_i(z) = \begin{cases} 
O(h^3) & \text{if nonstiff,} \\
O(h^2) & \text{if strongly stiff}, 
\end{cases}$$

$$R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \to \begin{cases} 
1 & \text{if nonstiff,} \\
-1 & \text{if strongly stiff}, 
\end{cases} \quad \text{as } h \to 0.$$ 

The stability function $R(z) \to -1$ for large $|z|$, results in the cancellation of consecutive terms. On the other hand, for small $|z|$, $R(z) \to 1$, the sum over the local error terms in the global error introduce a factor $n$ which results in the cancellation of one power of $h$ in the nonstiff case.

Hence, by (2.15) with $\epsilon_0 = 0$,

$$\epsilon_n = O(h^2) \quad \text{as } h \to 0. \quad (2.17)$$

in both cases and there is no order reduction. □

**Example 2.5 2-stage Gauss method**

$\mathcal{G}_2$ has classical order 4 and stage order 2 since $B(4)$ and $C(2)$ hold. The method has tableau given in Table 1.2 (b) (refer Chapter 1, Section 1.3).
In the nonstiff case, the local error is given by
\[
\psi_i(z) = \frac{h^5}{5!36} y^{(5)}(x_{i-1}) - \frac{zh^4}{4!36} y^{(4)}(x_{i-1}) + O(h^6).
\]

In the stiff case, the local error is given by
\[
\psi_i(z) = \frac{h^3}{36} y''(x_{i-1}) + \frac{h^3}{6z} y''(x_{i-1}) + O(h^4).
\]

Therefore,
\[
\psi_i(z) = \begin{cases} 
O(h^5) & \text{if nonstiff,} \\
O(h^3) & \text{if strongly stiff,}
\end{cases}
\]

\(R(z) = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2} \rightarrow \begin{cases} 
1 & \text{if nonstiff,} \\
1 & \text{if strongly stiff,}
\end{cases} \text{ as } h \to 0.
\]

Hence, by (2.15) with \(\epsilon_0 = 0\),
\[
\epsilon_n = \begin{cases} 
O(h^4) & \text{if nonstiff,} \\
O(h^2) & \text{if strongly stiff,}
\end{cases} \text{ as } h \to 0, \tag{2.18}
\]

and order reduction from the classical order 4 to the stage order 2 is exhibited illustrating the phenomenon of order reduction where the error behaviour of the method for stiff problems is governed by the stage order (see Figure 2.1a).

\(\square\)

**Example 2.6** 3-stage Gauss method

\(G_3\) has classical order 6 and stage order 3 since \(B(6)\) and \(C(3)\) hold. The method has tableau given in Table 1.2 (c) (refer Chapter 1, Section 1.3).

In the nonstiff case, the local error is given by
\[
\psi_i(z) = \frac{h^7}{7!400} y^{(7)}(x_{i-1}) - \frac{zh^6}{6!400} y^{(6)}(x_{i-1}) + O(h^8).
\]

In the stiff case, the local error is given by
\[
\psi_i(z) = \frac{h^4}{4!20} y^{(4)}(x_{i-1}) + \frac{3h^4}{4!5z} y^{(4)}(x_{i-1}) + \frac{h^5}{5!8} y^{(5)}(x_{i-1}) + O(h^6).
\]
Therefore,
\[
\psi_{i}(z) = \begin{cases} 
O(h^7) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases}
\]
\[
R(z) = \frac{1 + \frac{1}{2}z + \frac{1}{10}z^2 + \frac{1}{120}z^3}{1 - \frac{1}{2}z + \frac{1}{10}z^2 - \frac{1}{120}z^3} \rightarrow \begin{cases} 1 & \text{if nonstiff,} \\
-1 & \text{if strongly stiff,}
\end{cases}
as h \to 0.
\]

Hence, by (2.15) with \( \epsilon_0 = 0 \),
\[
\epsilon_n = \begin{cases} 
O(h^6) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0,
\tag{2.19}
\]
and there is order reduction from 6 to 4 (see Figure 2.1b).

![Figure 2.1](image)

Figure 2.1: Phenomenon of order reduction of the 2-stage and 3-stage Gauss methods.

The family of Lobatto IIIA methods

The Lobatto IIIA methods with \( s \)-stages satisfy \( B(2s - 2) \) and \( C(s) \). These methods are given in Table 1.3 (see Chapter 1, Section 1.3). They are special in that the first row of \( A \) are zeros and the last row of \( A \) is equal to the weights. We define
\[
A = \begin{bmatrix} 0 & 0 \\
\pi & \overline{A} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\
\overline{b} \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\
\overline{c} \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\
0 & \overline{T} \end{bmatrix},
\tag{2.20}
\]

39
where \( \overline{a} \), \( \overline{b} \) and \( \overline{c} \) are \((s - 1) \times 1\) subvectors and \( \overline{A} \) is an \((s - 1) \times (s - 1)\) submatrix, assumed nonsingular. We have
\[
\overline{e}^T_{s-1} \overline{a} = b_1, \quad \overline{e}^T_{s-1} \overline{A} = \overline{b}^T, \quad \overline{e}^T_{s-1} \overline{c} = 1.
\]
The stability function has formula given by
\[
R(z) = 1 + z \left( b_1 + \overline{b}^T \left( I - z \overline{A} \right)^{-1} (\overline{c} + z \overline{a}) \right).
\]
In the nonstiff case, we have
\[
z \overline{b}^T (I - z A)^{-1} (c^{s+1} - (s + 1) A c^s) = z \overline{b}^T \left( I + z A^2 + \cdots \right) (\overline{e}^{s+1} - (s + 1) A e^s),
\]
\[
= z \overline{b}^T (\overline{e}^{s+1} - (s + 1) A e^s) + O(z^2).
\]
However in the stiff case,
\[
z \overline{b}^T (I - z A)^{-1} (c^{s+1} - (s + 1) A c^s) = -\overline{b}^T A^{-1} \left( I - \frac{1}{2} A^{-1} \right)^{-1} (\overline{e}^{s+1} - (s + 1) A e^s),
\]
\[
= -\overline{b}^T A^{-1} \left( I + \frac{1}{2} A^{-1} + \cdots \right) (\overline{e}^{s+1} - (s + 1) A e^s),
\]
\[
= (s + 1) \overline{b}^T \overline{e}^s - 1 - \frac{1}{z} \overline{b}^T A^{-2} (\overline{e}^{s+1} - (s + 1) A e^s) + O(1/z^2),
\]
if \( B(s + 1) \) holds and by the stiffly accurate property since the last row of the \( A \) matrix has the same coefficients as the \( b \) vector (i.e \( a_{sj} = b_j \) for \( j = 1, \ldots, s \)), and also the last component of \( c \) is 1.

In the following examples, we analyse the local and global errors of the Lobatto IIIA methods.

**Example 2.7 2-stage Lobatto IIIA method**
The method is also known as the implicit trapezoidal rule. It satisfies \( B(2), C(2) \) and is of order 2. The method has tableau given in Table 1.3 (a) (refer Chapter 1, Section 1.3).

In the nonstiff case, the local error gives
\[
\psi_i(z) = -\frac{h^3}{12} y'''(x_{i-1}) - \frac{z h^3}{24} y'''(x_{i-1}) + O(h^4).
\]
In the stiff case, the local error gives

\[ \psi_i(z) = \frac{h^3}{6z} y''(x_{i-1}) + O(h^4/z) + O(1/z^2). \]

Therefore, since \( z = h\lambda \), and as \( h \to 0 \),

\[ \psi_i(z) = \begin{cases} 
O(h^3) & \text{if nonstiff}, \\
O(h^3/z) = O(h^2/\lambda) & \text{if strongly stiff}, 
\end{cases} \]

The stability function is the same to the 1-stage Gauss method (see Example 2.4).

The local error \( \psi_i(z) \) behaves like \( O(h^2/\lambda) \) in the strongly stiff case because of cancellation in the \( h^3 \) terms. Hence, by (2.15) with \( \epsilon_0 = 0 \),

\[ \epsilon_n = \begin{cases} 
O(h^2) & \text{if nonstiff}, \\
O(h^2/\lambda) = O(h^4) & \text{if strongly stiff since } |\lambda| \sim O(1/h^2), 
\end{cases} \quad \text{as } h \to 0. \quad (2.23) \]

Thus we expect to observe superconvergence in the strongly stiff case. However, we did not observe the \( O(h^4) \) behaviour numerically. This may be due to the \( \lambda \) being very large and therefore, reasonable small stepsize is required to observe the order behaviour. \( \square \)

**Example 2.8 3-stage Lobatto IIA method**

\( \mathcal{L}_3 \) satisfies \( B(4), C(3) \). The method has tableau given in Table 1.3 (b) (refer Chapter 1, Section 1.3).

In the nonstiff case, the local error gives

\[ \psi_i(z) = \frac{z h^4}{576} y^{(4)}(x_{i-1}) - \frac{h^5}{2880} y^{(5)}(x_{i-1}) + O(h^6). \]

In the stiff case, the local error gives

\[ \psi_i(z) = \frac{h^4}{48z} y^{(4)}(x_{i-1}) + \frac{h^5}{96z} y^{(5)}(x_{i-1}) + O(h^6). \]

Therefore, since \( z = h\lambda \), and as \( h \to 0 \),

\[ \psi_i(z) = \begin{cases} 
O(h^5) & \text{if nonstiff}, \\
O(h^4/z) = O(h^3/\lambda) & \text{if strongly stiff}, 
\end{cases} \]
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

The stability function is the same to the 2-stage Gauss method (see Example 2.5).

Hence, by (2.15) with \( \epsilon_0 = 0 \),

\[
\epsilon_n = \begin{cases} 
O(h^4) & \text{if nonstiff,} \\
O(h^2/\lambda) = O(h^4) & \text{if strongly stiff since } |\lambda| \sim O(1/h^2),
\end{cases} \quad \text{as } h \to 0. \quad (2.24)
\]

No order reduction is observed in both cases. In the numerical experiments, we observe that the methods suffer order reduction 4 to 2 (see Subsection 4.4.2, Figure 4.12). However, the \( O(h^4) \) behaviour is seen for the Kaps problem (see Figure 4.16).

Example 2.9 4-stage Lobatto IIIA method

\( \mathcal{L}_4 \) satisfies \( B(6), C(4) \). The method has tableau given in Table 1.3 (c) (refer Chapter 1, Section 1.3).

In the nonstiff case, local error yields

\[
\psi_i(z) = -\frac{h^7}{7!300} y^{(7)}(x_{i-1}) + \frac{zh^6}{6!300} y^{(6)}(x_{i-1}) + O(h^8).
\]

In the stiff case, the local error is given by

\[
\psi_i(z) = \frac{h^5}{600z} y^{(5)}(x_{i-1}) + \frac{3zh^6}{6!5z} y^{(6)}(x_{i-1}) + O(h^7).
\]

Therefore, since \( z = h\lambda \) and as \( h \to 0 \),

\[
\psi_i(z) = \begin{cases} 
O(h^7) & \text{if nonstiff,} \\
O(h^5/z) & \text{if strongly stiff,}
\end{cases}
\]

The stability function is the same to the 3-stage Gauss method (see Example 2.6).

Hence, by (2.15) with \( \epsilon_0 = 0 \),

\[
\epsilon_n = \begin{cases} 
O(h^6) & \text{if nonstiff,} \\
O(h^4/\lambda) = O(h^6) & \text{if strongly stiff since } |\lambda| \sim O(1/h^2),
\end{cases} \quad \text{as } h \to 0. \quad (2.25)
\]

No order reduction is expected in the strongly stiff case. In the numerical experiments, we observe that the methods suffer order reduction 6 to 4 (see Subsection 4.4.2, Figure 4.13). The \( O(h^6) \) behaviour is seen for the Kaps problem (see Figure 4.17).
Since methods such as Gauss are not L-stable nor stiffly accurate, the local errors of the methods are not dampened out quickly when $|z|$ becomes large. The global error therefore, will depend on the local error. On the other hand, for L-stable methods, as $z \to \infty$, the global error is dominated by the error in the final step. If the local error behaves like $O(h^{p+1})$, then so does the global error. This is another reason why the damping property of L-stable methods is advantageous when solving stiff problems.

Besides showing order reduction, Gauss methods have weak damping properties. We wish to discuss this further in the next subsection.

### 2.1.6 Weak damping properties

Methods like Gauss especially the IMR and $G_3$ have odd stages. When these methods are used to solve stiff problems, global errors show oscillatory behaviour and can sometimes destroy the solutions. In Figure 2.2, oscillations are due to the property $R(\infty) = -1$ of the stability function. To explain this behaviour, consider applying IMR and ITR to the PR problem (1.4) for $g(x) = e^{-x}$ and $\lambda = -10^6$.

![Figure 2.2: Error behaviour of IMR and ITR.](image)

The global error after $n$ steps with stepsize $h$ is given by (2.15). For the IMR,

$$r = R(z) = \frac{1 + \frac{z}{2}}{1 - \frac{z}{2}} = -\left(1 + \frac{z}{2}\right) \left(1 + \frac{2}{z} + \frac{4}{z^2} + \cdots\right),$$

$$= -\left(1 + \frac{1}{z} + \frac{8}{z^2} + \cdots\right) = -1, \text{ as } z \to \infty.$$
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

Since the IMR satisfies $B(2)$ and $C(1)$, local error at the $i$-th step as given in Example 2.4 yields,

$$\psi_i(z) = \frac{h^2}{4} \left( y''(x_{i-1}) + \frac{h}{2} y'''(x_{i-1}) + \frac{2}{z} y''(x_{i-1}) + \frac{2h}{3z} y'''(x_{i-1}) \right) + O(1/z^2).$$

Hence as $z \to \infty$ and with the exact solution $g(x) = y(x) = e^{-x}$ and $x_0 = 0$, we have

$$\epsilon_1 = \psi_1(z),$$

$$= \frac{h^2}{4} \left( y''(x_0) + \frac{h}{2} y'''(x_0) + \frac{2}{z} y''(x_0) + \frac{2h}{3z} y'''(x_0) \right),$$

$$= \frac{h^2}{4} \left( 1 - \frac{h}{2} + \frac{2}{z} \left( 1 - \frac{h}{3} \right) \right) + O(1/z^2),$$

$$\epsilon_2 = r \psi_1(z) + \psi_2(z),$$

$$= -\frac{h^2}{4} \left( 1 + \frac{4}{z} \right) \left( y''(x_0) + \frac{h}{2} y'''(x_0) + \frac{2}{z} y''(x_0) + \frac{2h}{3z} y'''(x_0) \right) + \cdots,$$

$$= -\frac{h^2}{4} \left( -hy'''(x_0) + \frac{4}{z} y''(x_0) \right) = -\frac{h^2}{4} \left( h + \frac{4}{z} \right) + O(1/z^2),$$

$$\epsilon_3 = r^2 \psi_1(z) + r \psi_2(z) + \psi_3(z),$$

$$= \frac{h^2}{4} \left( 1 + \frac{8}{z} \right) \left( y''(x_0) + \frac{h}{2} y'''(x_0) + \frac{2}{z} y''(x_0) + \frac{2h}{3z} y'''(x_0) \right) + \cdots,$$

$$= \frac{h^2}{4} \left( y''(x_0) + \frac{3h}{2} y'''(x_0) + \frac{2}{z} \left( 3y''(x_0) + \frac{h}{3} y'''(x_0) \right) \right) + \cdots,$$

$$= \frac{h^2}{4} \left( 1 - \frac{3h}{2} + \frac{2}{z} \left( 3 - \frac{h}{3} \right) \right) + O(1/z^2),$$

$$\epsilon_4 = r^3 \psi_1(z) + r^2 \psi_2(z) + r \psi_3(z) + \psi_4(z),$$

$$= -\frac{h^2}{4} \left( 1 + \frac{12}{z} \right) \left( y''(x_0) + \frac{h}{2} y'''(x_0) + \frac{2}{z} y''(x_0) + \frac{2h}{3z} y'''(x_0) \right) + \cdots,$$

$$= -\frac{h^2}{2} \left( -hy'''(x_0) + \frac{2}{z} \left( 2y''(x_0) + hy'''(x_0) \right) \right) + \cdots,$$

$$= -\frac{h^2}{2} \left( h + \frac{2}{z} \left( 2 - h \right) \right) + O(1/z^2). \quad (2.26)$$

From this analysis, we observed that there are two factors contributing to the oscillations shown in Figure 2.2(a). They are the stability function and the derivatives $y''(x_0)$ or $y'''(x_0)$. As $z \to \infty$, the stability function tends to $-1$ and the sign changes according as $n$ is odd or even. In addition, for odd and even values of $n$, the global error depends on the second $y''(x_0)$ and third $y'''(x_0)$ derivatives respectively. When the exact solution is $g(x) = y(x) = e^{-x}$, the sign of the global error changes for the second and third.
derivatives explaining the oscillatory error behaviour observed in Figure 2.2(a).

However, this observation is not true for other smooth functions. If \( g(x) \) is chosen to be \( \sin(x) \) for example, the oscillatory behaviour does not occur. We can see this easily by verifying the global errors for \( g(x) = y(x) = \sin(x) \).

The global errors can be shown to be

\[
\begin{align*}
\epsilon_1 &= \frac{h^2}{4} \left( y''(x_0) + \frac{h}{2} y'''(x_0) \right) + O(1/z) = -\frac{3h^3}{8} + O(1/z), \\
\epsilon_2 &= -\frac{h^2}{4} \left( -hy'''(x_0) \right) + O(1/z) = -\frac{h^3}{4} + O(1/z), \\
\epsilon_3 &= \frac{h^2}{4} \left( y''(x_0) + \frac{3h}{2} y'''(x_0) \right) + O(1/z) = -\frac{5h^3}{8} + O(1/z), \\
\epsilon_4 &= -\frac{h^2}{2} \left( -hy'''(x_0) \right) + O(1/z) = -\frac{h^3}{2} + O(1/z).
\end{align*}
\]

Based on the analysis, there is no change in the sign and oscillatory error behaviour is not observed.

On the other hand, for the ITR, the method satisfies \( B(2) \) and \( C(2) \), local error at the \( i \)-th step as given in Example 2.7 yields,

\[
\psi_i(z) = \frac{h^2}{12\lambda} \left( 2y'''(x_{i-1}) + hy^4(x_{i-1}) \right) + O(1/z^2), \text{ as } z \to \infty.
\]

The ITR has the stability function similar to the IMR. Hence as \( z \to \infty \) and with the exact solution \( g(x) = y(x) = e^{-x} \) and \( x_0 = 0 \), the global errors can be shown to be

\[
\begin{align*}
\epsilon_1 &= \frac{h^2}{6\lambda} y'''(x_0) + \frac{h^3}{12\lambda} y^4(x_0) + O(1/z^2) = -\frac{h^2}{6\lambda} + \frac{h^3}{4\lambda} + O(1/z^2), \\
\epsilon_2 &= \frac{h^3}{6\lambda} y^4(x_0) + O(1/z^2) = \frac{h^3}{6\lambda} + O(1/z^2), \\
\epsilon_3 &= \frac{h^2}{6\lambda} y'''(x_0) + \frac{h^3}{4\lambda} y^4(x_0) + O(1/z^2) = -\frac{h^2}{6\lambda} + \frac{5h^3}{12\lambda} + O(1/z^2), \\
\epsilon_4 &= \frac{h^3}{3\lambda} y^4(x_0) + O(1/z^2) = \frac{h^3}{3\lambda} + O(1/z^2).
\end{align*}
\]

Thus the sign changes give the oscillatory error behaviour seen in Figure 2.2b. A similar behaviour is observed for the 3-stage Gauss method since its stability function will also tend to \(-1\) as \( z \to \infty \). Gauss methods with odd stages and Lobatto IIIA
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

methods with even stages are expected to show similar behaviour.

Additionally, Figure 2.2 also shows that the error is much smaller for the ITR than for the IMR when \(|\lambda|\) is large. This is easily seen by taking the ratio of the first term of the global errors for the ITR and IMR. The ratio of the ITR error to the IMR error for large \(|\lambda|\) is given by

\[
\frac{\tau_{n}^{\text{ITR}}}{\tau_{n}^{\text{IMR}}} = \frac{h^2}{h^2} = \frac{2}{3\lambda}.
\]

Figure 2.2 is given for the PR problem with \(g(x) = e^{-x}\) and \(\lambda = -10^6\). The difference between the log errors of 6.1761 tells us that the errors differ by the factor of \(10^6.1761 = 1500000\) and corresponds with the ratio above, since when \(\lambda = -10^6\), the ratio is approximately \(\frac{1}{1500000}\).

We have introduced symmetric methods and some properties that will be used in the construction of symmetrizers in Chapter 3. Moreover, we have also pointed out some of the advantages and disadvantages of symmetric methods in the solution of stiff problems. In the following section, we introduce a technique to improve the convergence of the symmetric Runge-Kutta methods.

2.2 Idea of extrapolation

2.2.1 Richardson extrapolation

Approximation in the numerical solution of differential equations was first observed by Richardson in 1910 [67]. Richardson showed that the approximations arising from finite differences can be expressed in the form of

\[ T(h) = \tau_0 + \tau_2 h^2 + \tau_4 h^4 + \tau_6 h^6 + \ldots + O(h^8), \]  

where \(T(0) = \tau_0\), \(h\) is an adjustable parameter and \(\tau_2, \tau_4, \tau_6\) are independent of \(h\). In his paper, Richardson showed that when using several values of \(h\) such that \(h, \frac{h}{2}, \frac{h}{4}\) and etc, the approximations will approach close to \(\tau_0\) and when \(h\) is small enough, the error is simply proportional to \(h^2\).

Similarly, using an appropriate linear combinations of three equations with \(h_1, h_2\) and
2.2. Idea of extrapolation

$h_3$, we can eliminate $\tau_2$ and $\tau_4$ to give a higher order approximations. The process can be continued to obtain more and more accurate approximations to $\tau_0$. Richardson [68] called this process as "the deferred approached to the limit". The term extrapolation is also used to describe the process.

The technique of accelerating the convergence of the sequence was successfully implemented in the study of numerical quadrature by Romberg [70] (see Subsection 2.2.2). A beautiful survey of extrapolation is given by Joyce [58].

2.2.2 Quadrature

Quadrature refers to the numerical approximations to a definite integral

$$ I = \int_a^b f(x)dx, $$

by using linear combinations of the integrand at points within the interval $[a, b]$. The aim is to attain a given level of precision with the fewest function evaluations.

Werner Romberg [70] was the first to apply extrapolation successfully in numerical quadrature for improving the accuracy of the trapezoidal rule. Theoretical investigations on various aspects in numerical quadrature is given by F. L. Bauer and H. Rutishauser and E. L. Stiefel [7]. Romberg formulated an approximation based on the trapezoidal rule. Romberg obtained higher order formulas by means of extrapolation of the trapezoidal rule. The trapezoidal sums are formed by dividing the original interval according to a predetermined sequence of subdivisions. A composite trapezoidal rule of approximate integration to the definite integral is given by

$$ T_n = \frac{h}{2} (f_0 + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n), $$

where $h = (b - a) / n$ and $f_k = f(a + kh / n)$ for $k = 0, 1, \ldots, n$. The asymptotic expansion of $T(h)$ has the form

$$ T(h) = \tau_0 + \tau_1 h^2 + \tau_2 h^4 + \tau_3 h^6 + \ldots + \tau_q h^{2q} + O(h^{2q+1}). \quad (2.28) $$

This expansion is the classical summation formula of Euler and Maclaurin [1]. The Euler-
Maclaurin summation formula for $T(h)$ is given by

\[ T(h) = I + \frac{B_2}{2!} (f'(b) - f'(a)) h^2 + \frac{B_4}{4!} (f^{(3)}(b) - f^{(3)}(a)) h^4 + \ldots \]
\[ + \frac{B_{2k}}{2k!} (f^{(2k-1)}(b) - f^{(2k-1)}(a)) h^{2k} + \ldots, \] as $h \to 0,$

where the $B_j$ are the classical Bernoulli numbers

\[ B_2 = \frac{1}{6}, \quad B_4 = -\frac{1}{30}, \quad B_6 = \frac{1}{42}, \quad B_8 = -\frac{1}{30}. \]

Romberg derived the well-known formula

\[ T_m^{(n)} = \frac{4^m T_{m-1}^{(n+1)} - T_{m-1}^{(n)}}{4^m - 1}, \quad m = 1, 2, \ldots, \]

where every entry $T_m^{(n)}$ can be displayed as $T$-table

\[ T_0^{(0)} \]
\[ T_0^{(1)} \quad T_1^{(0)} \]
\[ T_0^{(2)} \quad T_1^{(1)} \quad T_2^{(0)} \]
\[ T_0^{(3)} \quad T_1^{(2)} \quad T_2^{(1)} \quad T_3^{(0)} \]
\[ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \cdots \]

and $T_0^{(n)}$ obtained by the trapezoidal rule with stepsize $\frac{h_0}{2}$. 

From the $T$-table we can see that $T_1^{(n)}$ are the Simpson’s values and in general $T_2^{(n)}$ are values order 6 and the rest of the entries in that table are the higher order Romberg values.

Next, we discuss the applications of extrapolation in the solution of ODEs.
2.2.3 Extrapolation in ODEs

Smoothing technique

The theoretical foundation of the extrapolation technique used in solving ODEs was first investigated by William Gragg [49, 50]. Gragg introduced a smoothing formula to dampen the effects of the oscillatory parasitic component in the numerical solution of the two-step explicit midpoint rule (EMR). Using the Euler method as the starter, Gragg’s method for the EMR is given by

\[
\begin{align*}
y_1 &= y_0 + hf(x_0, y_0), \\
y_{n+1} &= y_{n-1} + 2hf(x_n, y_n), \\
\hat{y}_n(x) &= \frac{1}{4}(y_{n-1} + 2y_n + y_{n+1}),
\end{align*}
\]

(2.29)

where \( x_n = x_0 + nh \).

Gragg applied the EMR to the Kepler problem:

\[
\begin{align*}
y_1' &= y_2, \quad y_1(0) = 1, \\
y_2' &= -\frac{y_1}{r^3}, \quad y_2(0) = 0, \\
y_3' &= y_4, \quad y_3(0) = 0, \\
y_4' &= -\frac{y_3}{r^3}, \quad y_4(0) = \sqrt{1+e}. 
\end{align*}
\]

For \( 0 \leq e < 1 \), the components \( y_1(t) \), and \( y_3(t) \) describe an elliptical orbit with eccentricity \( e \) and \( r = \sqrt{y_1^2 + y_3^2} \). The results of the numerical experiments shown in Figure 2.3 are for \( e = 0.15, h = \frac{\pi}{10} \) and \( n = 200 \).

The oscillatory behaviour in Figure 2.3a is caused by the parasitic component in the numerical solution. To understand this parasitic behaviour, consider applying the method (2.29) to the Dahlquist test problem \( y'(x) = \lambda y \) and \( y(0) = 1 \) with exact solution \( y(x) = e^{\lambda x} \).

The numerical solution is given by

\[
y_n = e^{2z} e^{\lambda x_n \left(1-\frac{1}{6} z^2-\frac{1}{8} z^4\right)} + (-1)^n e^{-z} e^{-\lambda x_n \left(1-\frac{1}{6} z^2-\frac{1}{8} z^4\right)}, \quad \text{since } z = h\lambda.
\]

(2.30)
As $z \to 0$, the first term gives the exact solution, but the second term containing $(-1)^n e^{-\lambda x_n}$ is oscillatory and can grow unboundedly with increasing $x_n$ and will eventually destroy the solution. This component is called parasitic. However, Gragg showed that the parasitic behaviour can be dampened using the smoothing formula (2.29c). Substituting (2.30) into (2.29c) gives

$$\hat{y}_n = \frac{1}{2} e^{\frac{2}{3} z^2} e^{\lambda x_n} \left(1 - \frac{1}{6} z^2 - \frac{1}{8} z^4\right) \left(2 - \frac{2}{3} z^2 + O(z^4)\right) + (-1)^n e^{-z} e^{-\lambda x_n} \left(1 - \frac{1}{6} z^2 - \frac{1}{8} z^4\right) \left(\frac{1}{4} z^2 + O(z^4)\right).$$

Upon simplifying, we obtain

$$\hat{y}_n = e^{\lambda x_n} \left(1 + O(z^2)\right) + (-1)^n e^{-\lambda x_n} \left(\frac{1}{4} z^2 + O(z^4)\right).$$

The effect on the first term of $\hat{y}_n$ is to multiply the exact solution by a factor $(1 + O(z^2))$. However, the effect on the oscillatory component is multiplication by a factor $z^2/4$ which dampens this parasitic component. Dampening of the oscillatory component in $\hat{y}_n$ is called smoothing. Gragg was also the first to prove the existence of the asymptotic error expansion for the EMR. This therefore allows for the order of accuracy to be increased by eliminating the leading error term.

Gragg’s idea became popular and was tested by Bulirsch and Stoer for the solution of nonstiff problems. They developed an algorithm known as the *Gragg-Bulirsch-Stoer* (GBS) method using rational extrapolation and implemented with an automatic step-size selection mechanism. This was the first code (ODEX) developed by Bulirsch and
Stoer and their students in 1966 [9, 82]. There are also other codes like DIFEX1 by Deuflhard [36] that uses polynomial extrapolation and EULEX by Hairer and Wanner [53] that uses the Euler method. For DIFEX1, Deuflhard showed that polynomial extrapolation is more efficient than rational extrapolation.

Following Bulirsch and Stoer’s success, Dahlquist [32], Lindberg [62], Bader and Deuflhard [6] extended the application of extrapolation to the solution of stiff ordinary differential equations (ODEs). Dahlquist discussed the use of extrapolation by the trapezoidal rule. He showed that although the trapezoidal rule is A-stable, if active extrapolation is carried out then the resultant method will no longer be A-stable. Lindberg then studied the trapezoidal rule in a deeper context. When solving stiff problems using extrapolation by the trapezoidal rule, the numerical solutions show oscillatory behaviour. Lindberg suggested the use of smoothing formula given by Gragg to eliminate these oscillations. He developed a code named IMPEX2. Meanwhile, Bader and Deuflhard developed an extrapolation code for stiff ODEs known as METAN1 using the semi-implicit (linearly implicit) midpoint rule. This is analogous to the explicit midpoint rule. It starts with the backward Euler method and then \( n \) steps are taken using the linearly implicit midpoint rule. They compared the performance of METAN1 with the implicit multistep code GEAR [44] and the Rosebrock-Wanner code GRKF [60] on the stiff DETEST [38] problems. METAN1 performed better than the other two codes. Shampine [77] then modified METAN1 to produce a more robust code.

In general, since we are considering only symmetric methods, the modified \( T \)-table for polynomial extrapolation can be displayed as

\[
\begin{array}{c|cccccc}
\frac{H}{m_1} & T_{1,1} \\
\frac{H}{m_2} & T_{2,1} & T_{2,2} \\
\frac{H}{m_3} & T_{3,1} & T_{3,2} & T_{3,3} \\
\frac{H}{m_4} & T_{4,1} & T_{4,2} & T_{4,3} & T_{4,4} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\frac{H}{m_n} & T_{n,1} & T_{n,2} & T_{n,3} & T_{n,4} & \cdots & T_{n,n} \\
\hline
\text{order} & p & p+2 & p+4 & p+6 & \cdots & p+2(n+1) \\
\text{level} & 1^{st} & 2^{nd} & 3^{rd} & \cdots & n^{th}
\end{array}
\]
where

\[ p = \text{order of the methods}, \]
\[ n = \text{positive integer}, \]
\[ H = \text{the stepsizes at length } n, \]
\[ h_i = H/m_i, i = 1, 2, \ldots, n, \]
\[ m_i = \text{increasing sequence, } i = 1, 2, \ldots, n, \]
\[ T_{i,1} = \text{approximations using stepsizes } h_i = H/m_i. \]

The extrapolation formula by the Aitken-Neville formula is given as

\[ T_{i,j} = T_{i,j-1} + \frac{T_{i,j-1} - T_{i-1,j-1}}{\left(\frac{m_i}{m_{i-j+1}}\right)^p - 1}, \quad (2.31) \]

where \( i = j = 2, \ldots, n. \)

Early extrapolation codes are based on rational extrapolation [9]. Polynomial extrapolation is preferable, as in some systems, the numerical solution is more stable and it also corresponds to forming a highly structured family of Runge-Kutta formulas. One popular polynomial extrapolation code is based on the midpoint rule studied by Shampine and Baca [75].

Another important criterion when applying extrapolation is the choice of stepsize sequence. Some stepsize sequences that have been suggested are:

- Romberg [70] sequence

\[ m = 1, 2, 4, 8, 16, 32, 64, \ldots, \quad (2.32) \]

where \( m = 2^i, i = 0, 2, \ldots. \)

- Bulirsch [9] sequence

\[ m = 1, 2, 3, 4, 6, 8, 12, \ldots, \quad (2.33) \]

where \( m_0 = 1, m_{2i-1} = 2^i, m_{2i} = 3(2^{i-1}), i = 1, 2, \ldots. \)

This sequence became famous through the success of the GBS algorithm.
2.2. Idea of extrapolation

- Bader and Deuflhard [6] sequence

\[ m = 2, 6, 10, 14, 22, 50, 70, 98, 138. \]  \hspace{1cm} (2.34)

This sequence has been implemented in METAN1 using rational extrapolation.

- Bauer et al. [7] sequence

\[ m = 1, 2, 3, 6, 9, 18, 27, \ldots, \]  \hspace{1cm} (2.35)

where \( m_{2i} = 3^i, m_{2i+1} = 2(3^i), i = 1, 2, \ldots. \)

- Harmonic [6] sequence

\[ m = 1, 2, 3, 4, 5, 6, 7, \ldots, \]  \hspace{1cm} (2.36)

where \( m = i, i = 1, 2, \ldots. \)

This is the most economic sequence.

- Double harmonic [52] sequence

\[ m = 2, 4, 6, 8, 10, 12, 14, \ldots, \]  \hspace{1cm} (2.37)

where \( m = i + 2, i = 0, 2, 4, 6, \ldots. \)

Comparisons has been given in [52] on page 226 for the step sequences (2.32), (2.33) and (2.36) using explicit Euler. It is suggested that the harmonic sequence (2.36) is the best but they suffer from the rounding error (see Subsection 5.1). Since we are only using second level extrapolation, we have used the stepsize sequence \( \{m = 1, 2, 3\} \) in our numerical experiments.

A theoretical research on extrapolation which allows high order implicit Runge-Kutta methods to be used for stiff problems is studied by Chan in his PhD thesis [20] on page 65. Further analysis on extrapolation can be found in Chan’s paper “A-stability of implicit Runge-Kutta extrapolation” [24]. In that paper, Chan showed although symmetric methods are A-stable, their extrapolations are not A-stable. It is shown that neither the first nor the second extrapolation of the Gauss, Lobatto IIIA and Lobatto IIIB methods can be A-stable for any choice of stepsize sequence. For lower order implicit methods, Hairer, Bader and Lubich pointed out in [51] that when the linearly implicit midpoint rule
of Rosenbrock type is applied with smoothing, the first and second level of extrapolation can be A-stable for stepsize sequences (2.34) and (2.37).

As a result, Chan constructed a method that can overcome the stability barriers. Such a method (symmetrizer) is obtained by composing symmetric methods. The method itself is not symmetric but yet has some important features for the solution of stiff problems. With symmetrization, the 2-stage Gauss and the 3-stage Lobatto IIIA methods are shown to admit L-stable first and second extrapolation for the step sequence \( \{m = 1, 2, 3\} \). The symmetrization of the 2-stage Gauss and high order symmetric methods are given in Chapter 3.

### 2.3 Generalization of smoothing

In Section 2.1, we have addressed some of the issues that arise in using a symmetric method for solving stiff problems. We know that methods such as the Gauss and Lobatto IIIA of high order suffer from order reduction and they also have weak damping properties. The order reduction phenomenon will weaken the advantage of performing extrapolation because of the uncertainty in choosing the correct extrapolation formula. In addition, we know that the Gauss and Lobatto IIIA methods fail to preserve A-stability when applied with active extrapolation (first and second level). Therefore, although symmetric methods possesses \( h^2 \)-asymptotic error expansions, there are not efficient in performing extrapolation on the order-reduced solution for stiff problems. To overcome the above issues, Chan generalized the concept of smoothing introduced by Gragg. Chan’s generalized smoothing is called symmetrization and is constructed by taking the composition of two symmetric Runge-Kutta methods but with different weights \([21, 22, 23]\). The symmetrization generalizes the smoothing formulas used by Dahlquist and Lindberg \([32]\) for the IMR and ITR to higher order symmetric methods.

Symmetrizers for symmetric methods used in solving stiff problems are constructed to have the following properties:

1. Preserve the asymptotic error expansion of symmetric methods in even powers of the stepsize;

2. L-stability \( (R(\infty) = 0) \).

It is also desirable that symmetrization will result in some important features such as:
2.3. Generalization of smoothing

1. Damping of the oscillatory and stiff error components in the numerical solution;

2. Suppression of order reduction effects that occur for higher order symmetric methods.

3. Enabling first and second level extrapolations of the symmetrized 2-stage Gauss and the symmetrized 3-stage Lobatto IIIA methods to be A-stable with the stepsizes sequence \{1, 2, 3\}. However, Chan concluded that no higher level extrapolation of the symmetrized 2-stage Gauss or the symmetrized 3-stage Lobatto IIIA or IIIB can be A-stable for any choice of stepsizes sequence \[24\].

All these features are achievable with a method called a symmetrizer. The process of applying this symmetrizer is known as symmetrization. Construction of symmetrizers for high order symmetric methods is given in Chapter 3.

The symmetrizer \( \mathcal{R} \) for an \( s \)-stage symmetric method is generated by

\[
\begin{bmatrix}
  c & A & 0 \\
  e + c & e b^T & A \\
  b^T - u^T P & u^T & \end{bmatrix},
\]

where \( P \) is the \( s \times s \) permutation matrix that occurs in the symmetry conditions (2.2).

Next, we will give examples of symmetrizers for the IMR and ITR constructed using (2.38) and show that they are equivalent to the smoothing formula (see formula (2.29c)).

**Example 2.10 Symmetrized IMR**

The symmetrizer for the IMR, \( \mathcal{G}_1 = \left( \frac{1}{2}, 1, \frac{1}{2} \right) \) is given by

\[
\begin{bmatrix}
  \frac{1}{2} & 1 & 0 \\
  \frac{3}{2} & 1 & \frac{1}{2} \\
  1 - u & u & \end{bmatrix}
\]

\( u \) is determined by the damping property \( R(\infty) = 0 \). This method is of order 1 if \( u \neq 0 \).
and its stability function is given by

\[ R(z) = 1 + zb^T(I-zA)^{-1}e, \]

\[ = 1 + z \left[ \begin{array}{cc} 1 - u & u \\ -z & 1 + \frac{1}{2}z \end{array} \right]^{-1} \left[ \begin{array}{c} 1 \\ 1 \end{array} \right], \]

\[ = 1 + \frac{z}{(1 - \frac{1}{2}z)^2} \left[ \begin{array}{cc} 1 - u & u \\ -z & 1 + \frac{1}{2}z \end{array} \right]^{-1} \left[ \begin{array}{c} 1 \\ 1 \end{array} \right], \]

\[ = 1 + \frac{z(1 - \frac{1}{4}(1 - 2u)z)}{(1 - \frac{1}{2}z)^2}, \]

\[ = 1 - z^2(\frac{1}{4} - u), \]
Example 2.11 *Symmetrized ITR*

The symmetrizer for ITR $\tilde{L}_2$ is given by

$$
\tilde{L}_2 =
\begin{pmatrix}
0 & 0 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} & 0 \\
2 & \frac{1}{2} & 1 & \frac{1}{2} \\
\frac{1}{2} - u & \frac{1}{2} & u
\end{pmatrix}.
$$

The stability function for $\tilde{L}_2$ can be shown to be equivalent to Example 2.10. We obtain

$$
R(z) = \frac{1 + 2z^2(u - \frac{1}{8})}{(1 - \frac{1}{4}z^2)^2}.
$$

The choice of $u = \frac{1}{8}$ gives the same $L$-stable symmetrizer of order 1 as for the IMR. In the same way as (2.40), we can show the symmetrizer value is equivalent to the smoothing formula.

We understand that this smoothing formula is important for dampening the oscillatory and stiff error components in the numerical solution. We have shown in Figure 2.2 that the global error of the IMR is oscillatory when applied to a stiff PR problem. Similar oscillatory behavior has been observed by Gragg for the explicit midpoint rule. However, in the case of the EMR the oscillations arise from the parasitic component of the numerical solution as explained in Subsection 2.2.3 whereas for the IMR, the oscillations are due to the resulting sign changes of the stability function (see Subsection 2.1.6).

Figure 2.4 shows the error behaviour of IMR and ITR with smoothing applied to the PR problem. We observe that with smoothing, the oscillatory error has been dampened. To understand this behaviour, we carry out a similar analysis as that given in Subsection 2.1.6. The effect of smoothing on the IMR solution for the PR problem is
Chapter 2. Extrapolation of symmetric Runge-Kutta methods

Figure 2.4: Error behaviour of IMR and ITR with smoothing.

given by

\[ \tilde{e}_1 = \frac{\epsilon_0 + 2\epsilon_1 + \epsilon_2}{4} = \frac{h^2}{8} \left( 1 - h \right) - \frac{h}{12z} + O\left(1/z^2\right), \]

\[ \tilde{e}_2 = \frac{\epsilon_1 + 2\epsilon_2 + \epsilon_3}{4} = \frac{h^2}{8} \left( 1 - 2h \right) + \frac{h^2}{4z} \left( \frac{3}{2} - \frac{h}{2} \right) + O\left(1/z^2\right), \]

\[ \tilde{e}_3 = \frac{\epsilon_2 + 2\epsilon_3 + \epsilon_4}{4} = \frac{h^2}{8} \left( 1 - 3h \right) + \frac{h^2}{z} \left( \frac{3}{4} + \frac{h}{12} \right) + O\left(1/z^2\right), \]

where \( \epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3 \) and \( \epsilon_4 \) are the errors of the IMR given in (2.26). The explanation for the oscillatory error behaviour is given in Subsection 2.1.6, in which the contribution to these errors are due to the resulting sign changes of the global error accordingly as \( n \) is odd or even. However, in the case of the smoothed IMR, the magnitude of the resulting sign change is dampened.
In this section, we construct symmetrizers for order 4 and order 6 methods. For a method of order \( p \geq 4 \), symmetrizers are constructed to satisfy the order conditions to as high an order as possible and to achieve damping for stiff problems. In the case of an \( s \)-stage symmetric method of order \( p \) with nonsingular \( A \), the weight vector \( u \) has \( s \) components and these can be used to satisfy the damping condition \( \tilde{R}(\infty) = 0 \) and conditions for order \( p - 1 \). On the other hand, if \( A \) is singular, the weight vector \( u \) has \( s - 1 \) components. The symmetrizer is constructed so as to preserve the asymptotic error expansion in even powers of \( h \). In addition, we shall show that symmetrizers suppress order reduction which is important in solving stiff problems (see Section 4).
3.1 Symmetrization

Let $\mathcal{R}_m = \frac{1}{m} R^m$ denote the composition of $m$ steps of an arbitrary symmetric Runge-Kutta method $R \equiv (A, b, c)$ each with stepsize $h = H/m$. If the last step of $\mathcal{R}_m$ is replaced by a method $\tilde{R}$, the resulting method is then denoted by $\tilde{\mathcal{R}}_m = \frac{1}{m} (\mathcal{R}^{m-1} \circ \mathcal{R})$ as shown in Figure 3.1. When $m = 1$, $\mathcal{R}^{m-1} = I$, the identity method which leaves the starting value unchanged for all step sizes and problems. Since the symmetry property $-\mathcal{R}^{-1} = \mathcal{R}$ results in $\mathcal{R}_{-m} = \mathcal{R}_m$ for all $m$, where $\mathcal{R}_{-m}$ is the composition of $m$ steps of the adjoint method $-\mathcal{R}^{-1}$ with stepsize $h$, the method $\tilde{R}$ is constructed so that $\tilde{\mathcal{R}}_{-m} = \tilde{\mathcal{R}}_m$ for all $m$. The method $\tilde{R}$, called a (one-step) symmetrizer, must now satisfy $\tilde{R} \circ (-\mathcal{R}^{-1}) = \mathcal{R}^2$.

![Figure 3.1: Symmetrization in the m-th step](image)

The symmetrizer constructed by Chan [20] has Butcher tableau given below.

$$
\bar{R} =
\begin{bmatrix}
  c & A & 0 \\
  e + c & e b^T & A \\
  b^T - u^T P & u^T
\end{bmatrix},
$$

where $P$ is the permutation matrix whose $(i, j)$-th element is the Kronecker delta $\delta_{i,s+1-j}$.

The weight vector $u$ is chosen to satisfy the damping and the order conditions. We remark that the method $\tilde{R}$ is the composition of two steps of $R$ except that the weights are different and gives an update over one step. It has $2s$ stages in general if the base method has $s$ stages. A generalization to a composition involving four steps of $\mathcal{R}$ is also possible (called a two-step symmetrizer). However, two-step symmetrizers will not be considered in this thesis.
3.2 Stability function of the symmetrizer

An important result obtained by Chan in [20] is \( \hat{\mathcal{R}}^{-m} = \hat{\mathcal{R}}_m \).

\[
\hat{\mathcal{R}}^{-m} = \frac{1}{m} (\mathcal{R}^{-m} \circ \hat{\mathcal{R}}) = \frac{1}{m} (-\mathcal{R}^{-m} \circ (-\hat{\mathcal{R}})) = \frac{1}{m} (\mathcal{R}^{m+1} \circ -\hat{\mathcal{R}})
\]

\[
= \frac{1}{m} (\mathcal{R}^{m-1} \circ \mathcal{R}^2 \circ -\hat{\mathcal{R}}) = \frac{1}{m} (\mathcal{R}^{m-1} \circ \hat{\mathcal{R}}) = \hat{\mathcal{R}}_m,
\]

provided \( \hat{\mathcal{R}} \circ (-\hat{\mathcal{R}}^{-1}) = \mathcal{R}^2 \) since \( \mathcal{R} \) is symmetric and satisfies \( \mathcal{R} = -\mathcal{R}^{-1} \). This relation is true for arbitrary symmetric methods and thus the result leads to the \( h^2 \)-asymptotic error expansion.

In the method \( \hat{\mathcal{R}}_m = \frac{1}{m} (\mathcal{R}^{m-1} \circ \hat{\mathcal{R}}) \), the \( m \) refers to the number of symmetric method steps taken before applying symmetrization. For example, if we choose \( m = 1 \), we have \( \hat{\mathcal{R}} = \hat{\mathcal{R}} \) which means symmetrization is applied at every step. For \( m = 2 \), we have \( \hat{\mathcal{R}}_2 = \frac{1}{2} (\mathcal{R} \circ \hat{\mathcal{R}}) \). It means that symmetrization is applied at every alternate step. Active symmetrization and the preferred choice of \( m \) will be discussed in Chapter 4.

3.2 Stability function of the symmetrizer

For simplicity, we denote the symmetrizer (3.1) as

\[
\hat{\mathcal{R}} = \frac{\tilde{c}}{\tilde{b}} \tilde{\mathcal{A}},
\]

where

\[
\tilde{\mathcal{A}} = \begin{bmatrix} A & 0 \\ eb^T & A \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} b - Pu \\ u \end{bmatrix}, \quad \tilde{c} = \begin{bmatrix} c \\ e + c \end{bmatrix}, \quad \tilde{I} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad \tilde{e} = \begin{bmatrix} e \\ e \end{bmatrix},
\]

and \((A, b, c)\) refers to the symmetric method \( \mathcal{R} \).
The stability function of the symmetrizer (3.2) is given by

\[ \tilde{R}(z) = 1 + z \tilde{b}^T (\tilde{I} - z\tilde{A})^{-1} \tilde{e}, \]

\[ = 1 + z \begin{bmatrix} b^T - u^T \bar{P} & u^T \end{bmatrix} \begin{bmatrix} I - zA & 0 \\ -zeb^T & I - zA \end{bmatrix}^{-1} \begin{bmatrix} e \\ e \end{bmatrix}, \]

\[ = 1 + z \begin{bmatrix} b^T - u^T \bar{P} & u^T \end{bmatrix} \begin{bmatrix} (I - zA)^{-1}e \\ (I - zA)^{-1}e(1 + zb^T(I - zA)^{-1}e) \end{bmatrix}, \]

\[ = 1 + z \begin{bmatrix} b^T - u^T \bar{P} & u^T \end{bmatrix} \begin{bmatrix} (I - zA)^{-1}e \\ R(z)(I - zA)^{-1}e \end{bmatrix}, \]

\[ = R(z)(1 + zu^T(I - zA)^{-1}e) - zu^TP(I - zA)^{-1}e. \] (3.3)

Using the symmetry condition (2.2), \( P(I - zA)^{-1}e \) simplifies to

\[ (I + zA)P - P(I - zA) = z(AP + PA) = zeb^T, \]

\[ P(I - zA)^{-1} - (I + zA)^{-1}P = z(I + zA)^{-1}eb^T(I - zA)^{-1}, \]

\[ P(I - zA)^{-1} = z(I + zA)^{-1}eb^T(I - zA)^{-1} + (I + zA)^{-1}P, \]

\[ P(I - zA)^{-1}e = z(I + zA)^{-1}eb^T(I - zA)^{-1}e + (I + zA)^{-1}e, \]

\[ = R(z)(I + zA)^{-1}e. \] (3.4)

Now (3.3) becomes

\[ \tilde{R}(z) = R(z)(1 + zu^T(I - zA)^{-1}e) - zu^TP(I + zA)^{-1}e), \]

\[ = R(z)(1 + 2zu^T(I - z^2A^2)^{-1}e), \] (3.5)

since \( Ae = c \). Hence \( \tilde{R}(\infty) = R(\infty)(1 - 2u^TA^{-1}e) \) if \( A \) is nonsingular. Thus \( \tilde{R}(\infty) = 0 \) gives the damping condition

\[ u^T A^{-1}e = \frac{1}{2}, \] (3.6)

In addition, we can obtain stronger damping if we expand the inverse series given in equation (3.5) to a higher degree in \( \frac{1}{z^2} \) as \( z \to \infty \). The stronger damping condition is
3.2. Stability function of the symmetrizer

given by

\[ u^T A^{-1} c A = 0. \]  (3.7)

For Lobatto IIIA methods where \( A \) is singular as defined in (2.20) (see Chapter 2, Subsection 2.1.5), the stability function is given by

\[ R(z) = 1 + z \left( b_1 + b^T (I - z^2 A)^{-1} (\varpi + z a) \right). \]

Then from (3.5), the stability function of the symmetrizer is given by

\[
\tilde{R}(z) = R(z) \left( 1 + 2 z^2 u^T (I - z^2 A^2)^{-1} c \right),
\]

\[
= R(z) \left( 1 + 2 z^2 \begin{bmatrix} u_1 & \bar{u}^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -z^2 A \varpi & T - z^2 A^2 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \varpi \end{bmatrix} \right),
\]

\[
= R(z) \left( 1 + 2 z^2 \begin{bmatrix} u_1 & \bar{u}^T \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^2 A \varpi & \frac{1}{T - z^2 A^2} \end{bmatrix} \begin{bmatrix} 0 \\ \varpi \end{bmatrix} \right),
\]

\[
= R(z) \left( 1 + 2 z^2 \begin{bmatrix} u_1 & \bar{u}^T \end{bmatrix} \begin{bmatrix} 0 \\ \frac{\varpi}{T - z^2 A^2} \end{bmatrix} \right),
\]

\[
= R(z) \left( 1 + 2 z^2 \bar{u}^T \right) \left( T - z^2 A^2 \right)^{-1} \varpi, \]  (3.8)

with \( u^T = [u_1, \bar{u}] \).

As \( z \to \infty \), we have \( \tilde{R}(\infty) = R(\infty) \left( 1 - 2 \pi^T A^{-2} \varpi \right) \), which gives the damping condition

\[ \pi^T A^{-2} \varpi = \pi^T A^{-1} \varpi + \pi^T A^{-2} \varpi = \frac{1}{2} \]  (3.9)
Chapter 3. Symmetrized Gauss and Lobatto IIIA methods

3.3 Order conditions

The symmetrizer (3.2) satisfies $C(q)$ if the method $(A, b, c)$ satisfies $C(q)$ and $B(q)$ since

$$\bar{A}c^{k-1} = \begin{bmatrix} A & 0 \\ eb^T & A \end{bmatrix} \begin{bmatrix} c^{k-1} \\ (e+c)^{k-1} \end{bmatrix} = \begin{bmatrix} Ac^{k-1} \\ eb^T c^{k-1} + A(e+c)^{k-1} \end{bmatrix},$$

$$= \begin{bmatrix} \frac{c^k}{k} \\ \frac{c^k}{k} + A \sum_{j=1}^{k} \binom{k}{j-1} c^{j-1} \end{bmatrix} = \begin{bmatrix} \frac{c^k}{k} \\ \frac{c^k}{k} + A \sum_{j=1}^{k} \binom{k}{j-1} \frac{c^j}{j} \end{bmatrix},$$

$$= \begin{bmatrix} \frac{c^k}{k} \\ \frac{c^k}{k} + \sum_{j=1}^{k} \binom{k}{j} \frac{c^j}{j} \end{bmatrix} = \begin{bmatrix} \frac{c^k}{k} \\ \frac{(e+c)^k}{k} \end{bmatrix} = \bar{c}^k, \quad k = 1, \ldots, q.$$

One of the free parameters carried by the $s$ components of $u$ is used to satisfy the damping condition as given by (3.6), and the remaining $s-1$ parameters will be used to satisfy the order conditions to as high an order as possible.

Example 3.1 Order conditions for trees up to order 3 for (3.2)

$$\bar{b}^T \bar{c} = b^T e - u^T Pe + u^T e = b^T e = 1,$$

$$\bar{b}^T \tilde{c} = b^T c - u^T Pc + u^T (e+c) = \frac{1}{2} - u^T (e-c) + u^T (e+c) = \frac{1}{2} + 2u^T c,$$

$$\bar{c}^T \bar{c}^2 = b^T c^2 - u^T Pc^2 + u^T (e+c)^2 = \frac{1}{3} - u^T (e-c)^2 + u^T (e+c)^2 = \frac{1}{3} + 4u^T c,$$

$$\bar{b}^T \bar{A}c = b^T Ac - u^T (eb^T - A)(e-c) + \frac{1}{2} u^T e + u^T c + u^T Ac = b^T Ac + 2u^T c. \quad \square$$

If $C(2)$ holds, then we have

$$\bar{b}^T \bar{A}c = \bar{b}^T \left( \frac{\bar{c}^2}{2} \right) + 2u^T c = \frac{1}{6} + 2u^T c.$$

Thus for $\bar{G}_2$, the condition for order 3 is $u^T c = 0$. 

64
Similarly for the tree $\tilde{Y}$, if $C(3)$ holds then we have

$$
\tilde{b}^T \tilde{A} \tilde{c}^2 = \frac{1}{3} \tilde{b}^T \tilde{c}^3 = \frac{1}{3} b^T c^3 - \frac{1}{3} u^T (e - c)^3 + \frac{1}{3} u^T (e + c)^3 = \frac{1}{3} b^T c^3 + 2u^T c + \frac{2}{3} u^T c^3.
$$

We note that when $C(3)$ holds the order conditions for the order-4 trees reduce to that for the tree $\Psi$ while the conditions for the order-5 trees reduce to that for the trees $\Psi$ and $\tilde{Y}$ (see Table 3.1).

The summary of the order conditions for $\tilde{G}_3$ using (3.1) is given in Table 3.1. The order conditions are needed for the calculation of the weight vector $u$. Table 3.1 shows that to construct an order 5 symmetrizer for $\tilde{G}_3$, we need the two conditions $u^T c = 0$ and $u^T c^3 = 0$.

Table 3.1: The order conditions for $\tilde{G}_3$ for trees of order 4 and 5 when $C(3)$ holds.

<table>
<thead>
<tr>
<th>$r(t)$</th>
<th>$t$</th>
<th>Reduced order conditions when $C(3)$ holds</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{c}^3$</td>
</tr>
<tr>
<td>4</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T (\tilde{A} \tilde{c})$</td>
</tr>
<tr>
<td>4</td>
<td>$\tilde{Y}$</td>
<td>$\tilde{b}^T \tilde{A} \tilde{c}^2$</td>
</tr>
<tr>
<td>4</td>
<td>$\tilde{Y}$</td>
<td>$\tilde{b}^T \tilde{A}^2 \tilde{c}$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{c}^4$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{c}^2 (\tilde{A} \tilde{c})$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{c} \tilde{A} \tilde{c}^2$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{c} \tilde{A}^2 \tilde{c}$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T (\tilde{A} \tilde{c})^2$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{A} \tilde{c}^3$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{A} (\tilde{A} \tilde{c})$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{A} (\tilde{A}^2 \tilde{c})$</td>
</tr>
<tr>
<td>5</td>
<td>$\Psi$</td>
<td>$\tilde{b}^T \tilde{A} \tilde{A}^2 \tilde{c}$</td>
</tr>
</tbody>
</table>

In the case of $\tilde{K}_3$, the order condition is the same as that for $\tilde{K}_2$. On the other hand, for
\( \tilde{\mathcal{Z}}_4 \) of order-5, since \( C(4) \) holds, the order condition for the tree \( \Psi \) reduces to
\[
\tilde{b}^T \tilde{A} \tilde{c}^3 = \frac{1}{4} \tilde{b}^T \tilde{c}^4 = \frac{1}{4} \tilde{b}^T \tilde{c} - \frac{1}{4} u^T (e - c)^4 + \frac{1}{4} u^T (e + c)^4 = \frac{1}{20} + 2(u^T c + u^T c^3),
\]
and in the same way, the order conditions for this is satisfied if \( u^T c = u^T c^3 = 0 \).

### 3.4 Construction of the symmetrizers

#### 3.4.1 Symmetric methods with nonsingular \( A \)

For a method where the matrix \( A \) is not singular, for example \( \tilde{G}_2 \) and \( \tilde{G}_3 \), the defining equation is given by
\[
\tilde{Y}^{[n]} = \tilde{c} y_{n-1} + h \tilde{A} \tilde{F}(x_{n-1} + \tilde{c} h, \tilde{Y}^{[n]}),
\]
\[
\tilde{y} = y_{n-1} + \tilde{b}^T \tilde{A}^{-1} (\tilde{Y}^{[n]} - \tilde{c} y_{n-1}),
\]
\[
= (1 - \tilde{b}^T \tilde{A}^{-1} \tilde{c}) y_{n-1} + \tilde{b}^T \tilde{A}^{-1} \tilde{Y}^{[n]},
\]
\[
= \tilde{R}(\infty) y_{n-1} + \tilde{b}^T \tilde{A}^{-1} Y^{[n]} = \tilde{b}^T \tilde{A}^{-1} Y^{[n]},
\]

By (3.2), we have
\[
\tilde{y} = \begin{bmatrix} b^T - u^T P & u^T \end{bmatrix} \begin{bmatrix} A & O \\ eb^T & A \end{bmatrix}^{-1} \begin{bmatrix} Y^{[n]} \\ Y^{[n+1]} \end{bmatrix},
\]
\[
= \begin{bmatrix} b^T - u^T P & u^T \end{bmatrix} \begin{bmatrix} A^{-1} & O \\ -A^{-1} eb^T A^{-1} & A^{-1} \end{bmatrix} \begin{bmatrix} Y^{[n]} \\ Y^{[n+1]} \end{bmatrix},
\]
\[
= \begin{bmatrix} b^T - u^T P & u^T \end{bmatrix} \begin{bmatrix} A^{-1} Y^{[n]} \\ -A^{-1} eb^T A^{-1} Y^{[n]} + A^{-1} Y^{[n+1]} \end{bmatrix},
\]
\[
= b^T A^{-1} Y^{[n]} - u^T PA^{-1} Y^{[n]} - u^T A^{-1} eb^T A^{-1} Y^{[n]} + u^T A^{-1} Y^{[n+1]}. \tag{3.10}
\]
3.4. Construction of the symmetrizers

Using the symmetry condition (2.2), we have $PA^{-1} = A^{-1}eb^TA^{-1} - A^{-1}P$ and with the damping condition $u^TA^{-1}e = \frac{1}{2}$, (3.10) simplifies to

$$\tilde{y}_n = u^T A^{-1} (PY[n] + Y[n+1]).$$

(3.11)

Equation (3.11) means that the symmetrized value at step $n$ depends on $Y[n]$ and $Y[n+1]$ which are the internal stage vectors of the Gauss method at steps $n$ and $n+1$ respectively [18].

We are now ready to determine the weight vector $u$ of the symmetrizers for $G_2$ and $G_3$.

**Example 3.2** Order-3 symmetrizer for $G_2$

Since $G_2$ has 2 stages, the symmetrized 2-stage Gauss $\tilde{G}_2$ will have two parameters $u_1$ and $u_2$ to be determined.

The condition for order 3 is

$$u^T c = 0.$$

(3.12)

Substituting the $c$ values yields

$$3(u_1 + u_2) + \sqrt{3}(u_2 - u_1) = 0.$$

(3.13)

The damping condition (3.6) is

$$u_1 - u_2 = \frac{\sqrt{3}}{12}.$$

(3.14)

Solving (3.13) and (3.14) gives

$$u_1 = \frac{1}{24} + \frac{\sqrt{3}}{24}, \text{ and } u_2 = \frac{1}{24} - \frac{\sqrt{3}}{24}.$$

Hence using (3.11) the symmetrizer for $G_2$ is given by

$$\tilde{y}_n = \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right) (Y_1^{[n]} + Y_2^{[n]}) + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right) (Y_1^{[n]} + Y_2^{[n+1]}).$$

(3.15)
From (3.5), the stability function for \( \tilde{G}_2 \) found to be

\[
\tilde{R}(z) = \frac{1 - \frac{1}{17} z^2}{(1 - \frac{1}{2} z + \frac{1}{17} z^2)^2},
\]

(3.16)

This can be shown to be A-stable and hence L-stable since \( \tilde{R}(\infty) = 0 \). It can be easily verified by checking the E-polynomial as given in Chapter 1 (equation (1.15)). □

**Example 3.3  Order-5 symmetrizer for \( \tilde{G}_3 \)**

For \( \tilde{G}_3 \) the fulfillment of (3.6) and (3.12) leaves one further parameter to be determined. The conditions for order-5 are

\[
u^T c^3 = 0 \quad \text{and} \quad u^T A c^2 = 0.
\]

Since \( C(3) \) holds for \( G_3 \) these two conditions are equivalent and \( \tilde{G}_3 \) will be of order-5 if we choose

\[
u^T c^3 = 0.
\]

(3.17)

Solving for \( u_1, u_2 \) and \( u_3 \) using (3.6), (3.12) and (3.17) give 3 linear equations,

\[
\begin{align*}
\frac{1}{2} (u_1 + u_2 + u_3) + \frac{\sqrt{15}}{10} (u_3 - u_1) &= 0, \\
\frac{7}{20} (u_1 + u_3) + \frac{9 \sqrt{15}}{100} (u_3 - u_1) + \frac{1}{8} u_2 &= 0, \\
6(u_1 + u_3) - 3u_2 - \frac{1}{2} &= 0.
\end{align*}
\]

The solutions are \( u_1 = \frac{13 + 3 \sqrt{15}}{360}, u_2 = -\frac{1}{45} \) and \( u_3 = \frac{13 - 3 \sqrt{15}}{360} \).

The symmetrized value is given by

\[
\tilde{y}_n = \left( \frac{1}{4} + \frac{\sqrt{15}}{15} \right) \left( Y_1^{[n+1]} + Y_3^{[n]} \right) + \left( \frac{1}{4} - \frac{\sqrt{15}}{15} \right) \left( Y_1^{[n]} + Y_3^{[n+1]} \right).
\]

(3.18)

and its stability function

\[
\tilde{R}(z) = \frac{1 - \frac{1}{47} z^2 + \frac{1}{405} z^4}{(1 - \frac{1}{2} z + \frac{1}{47} z^2 - \frac{1}{120} z^3)^2}.
\]

(3.19)

This can be shown to be A-stable and hence L-stable since \( \tilde{R}(\infty) = 0 \). □

68
3.4. Construction of the symmetrizers

**Example 3.4** Order-3 symmetrizer for $G_3$

A second symmetrizer of order 3 can be derived if, instead of (3.17), we choose the stronger damping condition given in (3.7).

Solving for the $u$'s gives $u = \left[ \frac{43 + 9 \sqrt{15}}{1224}, \frac{-4}{153}, \frac{43 - 9 \sqrt{15}}{1224} \right]^T$.

The symmetrized value is therefore given by

$$\tilde{y}_n = \left( \frac{55}{204} - \frac{7 \sqrt{15}}{102} \right) \left( Y_1^{[n]} + Y_3^{[n+1]} \right) + \left( \frac{55}{204} + \frac{7 \sqrt{15}}{102} \right) \left( Y_1^{[n+1]} + Y_3^{[n]} \right) - \frac{2}{51} \left( Y_2^{[n]} + Y_2^{[n+1]} \right),$$

and the stability function

$$\tilde{R}(z) = \frac{1 - \frac{1}{20} z^2 + \frac{111}{5100} z^4}{\left( 1 - \frac{1}{2} z + \frac{1}{10} z^2 - \frac{1}{120} z^3 \right)^2}.$$ (3.21)

The order-3 symmetrizer is not A-stable for very small values of $|z|$. This can be verified using the $E$-polynomial. However, it will not be a problem when applied passively and in fact we show that it works well for stiff problems.

In this thesis we study, in particular, the Gauss methods with 2 and 3 stages, $G_2$ and $G_3$, of classical order 4 and 6, respectively, for stiff problems. Butcher and Chan [18] have given a complete set of maximum attainable order for the symmetrizer $\tilde{G}_s$ for the $s$-stage Gauss method, $G_s$ (see Table 3.2). In general, a symmetrizer $\tilde{R}$ has odd order $2p - 1$ whenever the even order conditions up to order $2p - 2$ are satisfied. The local error of $\tilde{R}$ of order $2p - 1$ behaves like $O(h^{2p})$. We will show in our analysis that the symmetrizer can achieve order 4 for $G_2$ and order 6 for $G_3$ with order-3 symmetrizer when solving stiff linear problems. The analysis of the order behaviour of the symmetrizers will be given in Chapter 4.

<table>
<thead>
<tr>
<th>Stage number $s$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classical order of $G_s$</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
<td>10</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>Maximum attainable order of $\tilde{G}_s$</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>7</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>
3.4.2 Symmetric methods with singular $A$

We consider the special case of Lobatto IIIA methods [23]. For this method, the symmetrizer is reduced by one stage and therefore the $L_s$ symmetrizer carries only $s - 1$ parameters in general. Although the symmetrized Lobatto IIIA methods are constructed using (3.1), the matrices $A, b$ and $c$ are also partitioned by

$$A = \begin{bmatrix} 0 & 0 \\ \pi & A \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ \pi \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 \\ 0 & I \end{bmatrix},$$

(3.22)

as defined in Chapter 2 (see Subsection (2.1.5)).

**Example 3.5** Order-3 symmetrizer for $L_3$

The symmetrizer for the 3-stage Lobatto IIIA method is generated by

$$\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} & 0 & 0 & 0 & 0 \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
\frac{3}{2} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} & 0 \\
2 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\
\frac{1}{6} - u_3 & \frac{2}{3} - u_2 & \frac{1}{6} - u_1 & u_1 & u_2 & u_3
\end{array} \equiv
$$

$$\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} & 0 & 0 & 0 & 0 \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
\frac{3}{2} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} & 0 \\
2 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\
\frac{1}{6} - u_3 & \frac{2}{3} - u_2 & \frac{1}{6} - u_1 & u_1 & u_2 & u_3
\end{array} \equiv
$$

As a result of reducibility, only two parameters $u_2$ and $u_3$ are available, and they allow for the construction of an $L$-stable symmetrizer of order 3. Using $\pi^Tc = 0$ gives

$$\frac{1}{2}u_2 + u_3 = 0,$$

while the damping condition (3.9) gives

$$u_2 - 4u_3 = \frac{1}{6}.$$
Solving for \( u_2 \) and \( u_3 \) yields \( u_2 = \frac{1}{18} \) and \( u_3 = -\frac{1}{36} \).

The symmetrizer can be easily derived by taking the linear combination of the internal stage values \( Y^n \), \( Y^{n+1} \) as well as the updates \( y_{n-1} \) and \( y_{n+1} \). For simplicity, we define

\[
\begin{align*}
F_1 &= hf(x_{n-1}, y_{n-1}), \\
F_2 &= hf(x_{n-1} + \frac{1}{2}h, Y^n), \\
F_3 &= hf(x_n, y_n), \\
F_4 &= hf(x_n + \frac{1}{2}h, Y^{n+1}), \\
F_5 &= hf(x_{n+1}, y_{n+1}).
\end{align*}
\]

The defining equations are

\[
\begin{align*}
Y^n &= y_{n-1} + \frac{1}{24}(5F_1 + 8F_2 - F_3), \\
y_n &= y_{n-1} + \frac{1}{6}(F_1 + 4F_2 + F_3), \\
Y^{n+1} &= y_n + \frac{1}{24}(5F_3 + 8F_4 - F_5), \\
y_{n+1} &= y_n + \frac{1}{6}(F_3 + 4F_4 + F_5), \\
\tilde{y}_n &= y_{n-1} + \frac{1}{36}(7F_1 + 22F_2 + 6F_3 - F_5) + \frac{1}{8}F_4.
\end{align*}
\]

Now solving for \( F_1, F_2, F_3, F_4 \) and \( F_5 \) in terms of the left hand side yields

\[
\tilde{y}_n = \frac{1}{12}\left( -y_{n-1} + 4Y^n + 6y_n + 4Y^{n+1} - y_{n+1} \right). \quad (3.23)
\]

The stability function is the same to the order-3 symmetrizer for the 2-stage Gauss method. \( \square \)
Example 3.6 Order-5 symmetrizer for $L_4$

With reducibility, the tableau for the symmetrizer for the 4-stage Lobatto IIIA method is given by

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{2} - \frac{\sqrt{5}}{10} & \frac{11}{120} + \frac{\sqrt{5}}{120} & \frac{5}{24} - \frac{\sqrt{5}}{120} & \frac{5}{24} - \frac{13\sqrt{5}}{120} & -\frac{1}{120} + \frac{\sqrt{5}}{120} & 0 & 0 & 0 \\
\frac{1}{2} + \frac{\sqrt{5}}{10} & \frac{11}{120} - \frac{\sqrt{5}}{120} & \frac{5}{24} + \frac{13\sqrt{5}}{120} & \frac{5}{24} + \frac{\sqrt{5}}{120} & -\frac{1}{120} - \frac{\sqrt{5}}{120} & 0 & 0 & 0 \\
1 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & 0 & 0 & 0 \\
\frac{3}{2} - \frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{21}{120} + \frac{\sqrt{5}}{120} & \frac{5}{24} - \frac{\sqrt{5}}{120} & \frac{5}{24} - \frac{13\sqrt{5}}{120} & -\frac{1}{120} + \frac{\sqrt{5}}{120} \\
\frac{3}{2} + \frac{\sqrt{5}}{10} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{21}{120} - \frac{\sqrt{5}}{120} & \frac{5}{24} + \frac{13\sqrt{5}}{120} & \frac{5}{24} + \frac{\sqrt{5}}{120} & -\frac{1}{120} - \frac{\sqrt{5}}{120} \\
2 & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{5}{12} & \frac{1}{12} \\
\frac{1}{12} - u_4 & \frac{5}{12} - u_3 & \frac{5}{12} - u_2 & \frac{1}{12} & u_2 & u_3 & u_4 \\
\end{array}
\]

In this case, we need to solve for $u_2, u_3$ and $u_4$. Since we require order 5, the conditions are $\mathbf{u}^T \mathbf{c} = \mathbf{u}^T \mathbf{c}^3 = 0$ in addition to the damping condition $\mathbf{u}^T \mathbf{A}^{-2} \mathbf{c} = \frac{1}{2}$.

Solving

\[
\frac{1}{2}(u_2 + u_3) + \frac{\sqrt{5}}{10}(u_3 - u_2) + u_4 = 0,
\]

\[
\frac{1}{5}(u_2 + u_3) + \frac{2\sqrt{5}}{25}(u_3 - u_2) + u_4 = 0,
\]

\[
\frac{12\sqrt{5}}{5}(u_2 - u_3) + 24u_4 - \frac{1}{2} = 0,
\]

yields $u_2 = \frac{1}{240} + \frac{\sqrt{5}}{80}$, $u_3 = \frac{1}{240} - \frac{\sqrt{5}}{80}$ and $u_4 = \frac{1}{120}$.

The symmetrizer for $L_4$ can be derived by writing the symmetrized value $\tilde{y}_n$ similarly to the one given for $L_3$. 

72
We define the derivative functions by

\[ F_1 = h f(x_{n-1}, y_{n-1}), \]
\[ F_2 = h f\left(x_{n-1} + \left(\frac{1}{4} - \frac{\sqrt{5}}{120}\right)h, Y_1^{[n]}\right), \]
\[ F_3 = h f\left(x_{n-1} + \left(\frac{1}{4} + \frac{\sqrt{5}}{120}\right)h, Y_2^{[n]}\right), \]
\[ F_4 = h f(x_n, y_n), \]
\[ F_5 = h f\left(x_n + \left(\frac{1}{4} - \frac{\sqrt{5}}{120}\right)h, Y_1^{[n+1]}\right), \]
\[ F_6 = h f\left(x_n + \left(\frac{1}{4} + \frac{\sqrt{5}}{120}\right)h, Y_2^{[n+1]}\right), \]
\[ F_7 = h f(x_{n+1}, y_{n+1}). \]

The defining equations for the symmetrizer are then given by

\[
Y_1^{[n]} = y_{n-1} + \frac{1}{120} \left( (11 + \sqrt{5})F_1 + (25 - \sqrt{5})F_2 + (25 - 13\sqrt{5})F_3 - (1 - \sqrt{5})F_4 \right), \\
Y_2^{[n]} = y_{n-1} + \frac{1}{120} \left( (11 - \sqrt{5})F_1 + (25 + 13\sqrt{5})F_2 + (25 + \sqrt{5})F_3 - (1 + \sqrt{5})F_4 \right), \\
y_n = y_{n-1} + \frac{1}{12} (F_1 + F_4) + \frac{5}{12} (F_2 + F_3), \\
Y_1^{[n+1]} = y_n + \frac{1}{120} \left( (11 + \sqrt{5})F_1 + (25 - \sqrt{5})F_5 + (25 - 13\sqrt{5})F_6 - (1 - \sqrt{5})F_7 \right), \\
Y_2^{[n+1]} = y_n + \frac{1}{120} \left( (11 - \sqrt{5})F_1 + (25 + 13\sqrt{5})F_5 + (25 + \sqrt{5})F_6 - (1 + \sqrt{5})F_7 \right), \\
y_{n+1} = y_n + \frac{1}{12} (F_4 + F_7) + \frac{5}{12} (F_5 + F_6), \\
\tilde{y}_n = y_{n-1} + \frac{1}{240} \left( 18F_1 + 3(33 + \sqrt{5})F_2 + 3(33 - \sqrt{5})F_3 + (1 + 3\sqrt{5})F_5 + (1 - 3\sqrt{5})F_6 \right) \\
+ \frac{1}{120} \left( 10F_4 + F_7 \right).
\]

Solving for \( F_1, F_2, F_3, F_4, F_5, F_6 \) and \( F_7 \), in terms of the left hand side then yields

\[
\tilde{y}_n = \frac{1}{40} \left( y_{n-1} + (5 - 4\sqrt{5})(Y_1^{[n]} + Y_2^{[n+1]}) + 18y_n + (5 + 4\sqrt{5})(Y_2^{[n]} + Y_1^{[n+1]}) + y_{n+1} \right). \quad (3.24)
\]

The stability function is the same as that for the order-5 symmetrizer for the 3-stage Gauss method. □
Figure 3.2 shows the stability region of the symmetrized Gauss and the symmetrized Lobatto IIIA methods. The unshaded stability region indicates that the method is A-stable.

\[ \tilde{R}(z) = \frac{1 - \frac{1}{12}z^2}{(1 - \frac{1}{2}z + \frac{1}{12}z^2)^2}. \]

\[ \tilde{R}(z) = \frac{1 - \frac{1}{20}z^2 + \frac{1}{600}z^4}{(1 - \frac{1}{2}z + \frac{1}{10}z^2 - \frac{1}{120}z^3)^2}. \]

Figure 3.2: Stability functions and stability regions of the symmetrized Gauss and symmetrized Lobatto IIIA methods.
Strategies for symmetrization and extrapolation

There are two ways of applying symmetrization and we wish to investigate them in the constant and variable stepsize settings. In Chapter 2, we studied the order reduction phenomena for the symmetric methods when applied to the Prothero-Robinson problem. In this chapter, we give the analysis for active and passive symmetrization and show that they can suppress order reduction for stiff linear problem. Numerical results are given that verifies the analysis. In addition to this analysis, we also discuss an optimal choice of applying active symmetrization and present results for the performance of active and passive symmetrization numerically on some test problems.
4.1 Choice of strategies for symmetrization

4.1.1 Active symmetrization

In the active mode, the symmetrized value is propagated whenever symmetrization is applied. In this mode, there are different ways of applying symmetrization. When symmetrization is applied at every step, the method is equivalent to $\tilde{R}_1 = \tilde{R}$ as defined by (4.1). We can also consider methods where symmetrization occurs at every alternate step or after every second step, etc. In these cases, the basic method is either $\tilde{R}$, or $R \circ \tilde{R}$, or $R \circ R \circ \tilde{R}$, etc. having the following Butcher tableaux:

$$\tilde{R}_1 = R = (\tilde{A}, \tilde{b}, \tilde{c}) = \begin{array}{ccc}
  c & A & 0 \\
  e + c & e b^T & A \\
  b^T - u^T P & u^T & 0
\end{array} \quad (4.1)$$

If $\tilde{R}_m = \frac{1}{m}(R^{m-1} \circ \tilde{R})$, for $m = 1$, we are applying $\tilde{R}$ at every step.

For $m = 2$, $\tilde{R}_2 = \frac{1}{2}(R \circ \tilde{R})$, symmetrization is applied at every alternative step,

$$\tilde{R} \circ \tilde{R} = \begin{array}{ccc}
  c & A & 0 \\
  e + c & e b^T & A \\
  e + 2c & e b^T & e b^T & A \\
  b^T & b^T - u^T P & u^T & 0
\end{array} \quad (4.2)$$

For $m = 3$, $\tilde{R}_3 = \frac{1}{3}(R^2 \circ \tilde{R})$, symmetrization is applied after every third step. The Butcher tableau has 4-blocks and the symmetrizer will have 4s-stages.

In the active case, applying symmetrization simply means applying the method $\tilde{R}_m$. We will investigate which choice of $m$ is optimal. Numerical results on efficiencies are given in subsection 4.4.1.
4.2 Analysis of active and passive symmetrization

4.1.2 Passive symmetrization

In the passive mode, symmetrization is applied whenever greater accuracy is required but the symmetrized values are not propagated. That is, a sequence \( y_h(x_0), y_h(x_1), \ldots, y_h(x_n) \) of numerical solutions of the symmetric method is computed with stepsize \( h \), and these values are stored together with the internal stage values \( Y^{[n]} \) and \( Y^{[n+1]} \). When improved accuracy is required at the point \( x_n = x_0 + nh \), we then apply symmetrization according to equation (4.1).

Schematically, we have

\[
\begin{array}{ccccccc}
\text{symmetrized values} & \rightarrow & \tilde{y}_n \\
\text{symmetric values} & \rightarrow & y_n(x_0) & \cdots & y_h(x_n) & y_h(x_{n+1})
\end{array}
\]

Figure 4.1: Passive symmetrization at step \( n \).

4.2 Analysis of active and passive symmetrization

In this section, we present the analysis of symmetrization for the Prothero-Robinson problem (1.4). We show that symmetrizers can suppress order reduction and therefore allows extrapolation to be performed to improve accuracy.

4.2.1 Active symmetrization by \( \widehat{R}_1 = \tilde{R} \)

The global error \( \tilde{\epsilon}_n = y(x_n) - \tilde{y}_n \) at \( x_n \) for a symmetrized Runge-Kutta method \( (\tilde{A}, \tilde{b}, \tilde{c}) \) applied with a constant stepsize \( h \) is given by

\[
\tilde{\epsilon}_n = \sum_{i=1}^{n} \tilde{R}(z)^{n-i} \tilde{\psi}_i(z),
\]

where \( \tilde{\psi}_i \) is the local error for the \( i \)-th step and is given by

\[
\tilde{\psi}_i = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{i-1}) \left( 1 - k\tilde{b}^T \tilde{c}^{k-1} + z\tilde{b}^T (\tilde{I} - z\tilde{A})^{-1}(\tilde{c}^k - k\tilde{A}\tilde{c}^{k-1}) \right).
\]
This is similar to the analysis given in Chapter 2 (see Subsection 2.1.5) except we are analysing the method \((\tilde{A}, \tilde{b}, \tilde{c})\).

**Symmetrized Gauss methods**

In the **nonstiff** case, \(|\lambda| = O(h)| as \(h \to 0\), (4.4) yields

\[
\tilde{\psi}_i(z) = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{i-1}) (1 - k\tilde{b}^T \tilde{c}^{k-1} + z\tilde{b}^T (\tilde{c}^k - k\tilde{A}\tilde{c}^{k-1}) + O(z^2)).
\]  

(4.5)

In the **stiff** case (that is, \(|\lambda| \sim O(1/h^2)|), then \(|z| = |\lambda|h \sim O(1/h) \to \infty\) as \(h \to 0\), the local error (4.4) becomes

\[
\tilde{\psi}_i = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{i-1}) \left(1 - \tilde{b}^T \tilde{A}^{-1} \tilde{c}^k - \frac{1}{z} \tilde{b}^T \tilde{A}^{-2} (\tilde{c}^k - k\tilde{A}\tilde{c}^{k-1}) + O(1/z^2)\right),
\]  

(4.6)

provided \(\tilde{A}\) is nonsingular.

**Symmetrized Lobatto IIIA methods**

If \(\tilde{A}\) is singular, we partitioning the coefficients of the method (4.1) according to (3.22).

In the **nonstiff** case, \(|\lambda| = O(h)| as \(h \to 0\), using (2.21), (4.4) yields

\[
\tilde{\psi}_i(z) = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{i-1}) (1 - k\tilde{b}^T \tilde{c}^{k-1} + z\tilde{b}^T (\tilde{c}^k - k\tilde{A}\tilde{c}^{k-1}) + O(z^2)).
\]  

(4.7)

In the **stiff** case (that is, \(|\lambda| \sim O(1/h^2)|), then \(|z| = |\lambda|h \sim O(1/h) \to \infty\) as \(h \to 0\), using (2.22), (4.4) gives

\[
\tilde{\psi}_i = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{i-1}) \left(1 - \tilde{b}^T \tilde{A}^{-1} \tilde{c}^k - \frac{1}{z} \tilde{b}^T \tilde{A}^{-2} (\tilde{c}^k - k\tilde{A}\tilde{c}^{k-1}) + O(1/z^2)\right).\]

(4.8)
The stability function of the symmetrizer for nonstiff and stiff cases is given by

\[
\tilde{R}(z) = R(z) \left( 1 + 2z^2 u^T (I - z^2 A^2)^{-1} c \right),
\]

\[
\tilde{R}(z) \to \begin{cases} 
O(1) & \text{if nonstiff,} \\
O(1/z^2) = O(h^2) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0. \tag{4.9}
\]

In the nonstiff case, the sum in (4.3) contributes the factor \( n \) which results in cancellation of one power of \( h \). On the other hand, in the strongly stiff case, \( \tilde{R}(z) = O(h^2) \) as \( h \to 0 \), the contributions to the global error at step \( n \) due to the local errors at all steps except the last are dampened by the stability function. The global error at step \( n \) is therefore essentially determined by the local error at step \( n \).

**Example 4.1 1-stage Gauss method**

The symmetrized IMR satisfies \( B(1) \) and \( C(1) \). The method is of order-1 and the Butcher tableau is given by

\[
\tilde{G}_1 = \frac{1}{2} 1 0 \quad \frac{1}{2} \frac{3}{2} 1 \frac{1}{2} \frac{3}{4} 1 \frac{1}{4},
\]

where the method is constructed using (4.1) and the weights are given in Example 2.10.

In the nonstiff case, from (4.5) the local error is given by

\[
\tilde{\psi}_i(z) = -\frac{h^2}{4} y''(x_i-1) - \frac{zh^2}{8} y''(x_i-1) + O(h^3) + O(zh^3).
\]

In the stiff case, from (4.6) the local error is given by

\[
\tilde{\psi}_i(z) = -\frac{3h^2}{16} y''(x_i-1) + O(h^3) + O(h^3/z).
\]

Therefore,

\[
\tilde{\psi}_i(z) = \begin{cases} 
O(h^2) & \text{if nonstiff,} \\
O(h^2) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0.
\]
Hence the global error (4.3) is given by

$$
\tilde{\epsilon}_n = \begin{cases} 
O(h) & \text{if nonstiff,} \\
O(h^2) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0. \quad \square
$$

**Example 4.2 2-stage Gauss method**

$\tilde{\mathcal{G}}_2$ satisfies $B(3)$ and $C(2)$. The method is of order-3. The Butcher tableau is given by

$$
\begin{array}{c|cccc}
\frac{1}{2} - \frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{2} - \frac{\sqrt{3}}{6} & 0 & 0 \\
\frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} & 0 & 0 \\
\frac{3}{2} - \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} & , \\
\frac{3}{2} + \frac{\sqrt{3}}{6} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{4} + \frac{\sqrt{3}}{6} & 0 \\
\frac{1}{21} + \frac{\sqrt{3}}{21} & \frac{1}{21} - \frac{\sqrt{3}}{21} & \frac{1}{21} + \frac{\sqrt{3}}{21} & \frac{1}{24} - \frac{\sqrt{3}}{24} \\
\end{array}
$$

where the weights are obtained in Example 3.2 (see Chapter 3, Subsection 3.4.1).

In the nonstiff case, from (4.5) the local error is given by

$$
\tilde{\psi}_i(z) = \frac{zh^3}{432} y'''(x_{i-1}) + \frac{h^4}{216} y^{(4)}(x_{i-1}) + O(zh^4).
$$

In the stiff case, from (4.6) the local error is given by

$$
\tilde{\psi}_i(z) = -\frac{h^3}{36z} y'''(x_{i-1}) + \frac{h^4}{864} y^{(4)}(x_{i-1}) + O(h^4/z),
$$

where the term $1 - \tilde{b}^T \tilde{A}^{-1} \tilde{c}^3 = 0$ while $1 - \tilde{b}^T \tilde{A}^{-1} \tilde{c}^4 = -2u^T A^{-1} c^4 = \frac{1}{36}$. Since $\tilde{b}^T \tilde{A}^{-2}(\tilde{c}^3 - 3\tilde{A} \tilde{c}^2) = \frac{1}{6}$, the leading error term in $\tilde{\psi}_i(z)$ is $O(h^4)$.

Thus we have,

$$
\tilde{\psi}_i(z) = \begin{cases} 
O(h^4) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0,
$$

$$
\tilde{\epsilon}_n = \begin{cases} 
O(h^3) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0. \quad \square
$$
The analysis shows that the order reduction is suppressed by active symmetrization.

**Example 4.3** 3-stage Gauss method
\( \tilde{G}_3 \) satisfies \( B(5) \) and \( C(3) \). The method is of order-5 if we use order-5 symmetrizer as given in Example 3.3 (see Chapter 3, Subsection 3.4.1). On the other hand, \( \tilde{G}_3 \) satisfies \( B(3) \) and \( C(3) \) and the method is of order-3 if we use order-3 symmetrizer (see Example 3.4).

**Order-5 symmetrizer**

The Butcher tableau is given by

\[ \tilde{G}_3 = \begin{array}{c|cccccc}
\frac{1}{2} - \frac{\sqrt{15}}{10} & \frac{5}{36} & \frac{2}{9} - \frac{\sqrt{15}}{15} & \frac{5}{36} & \frac{\sqrt{15}}{30} & 0 & 0 & 0 \\
\frac{1}{2} & \frac{5}{36} + \frac{\sqrt{15}}{24} & \frac{2}{9} & \frac{5}{36} & \frac{\sqrt{15}}{24} & 0 & 0 & 0 \\
\frac{1}{2} + \frac{\sqrt{15}}{10} & \frac{5}{36} + \frac{\sqrt{15}}{20} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} & 0 & 0 & 0 & 0 \\
\frac{3}{2} & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} & \frac{5}{36} + \frac{\sqrt{15}}{20} & \frac{2}{9} & \frac{5}{36} - \frac{\sqrt{15}}{24} & , \\
\frac{3}{2} + \frac{\sqrt{15}}{10} & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} & \frac{5}{36} + \frac{\sqrt{15}}{20} & \frac{2}{9} + \frac{\sqrt{15}}{15} & \frac{5}{36} & \\
\frac{3}{2} + \frac{\sqrt{15}}{20} & \frac{7}{15} & \frac{30}{180} & \frac{13}{300} + \frac{3\sqrt{15}}{360} & -\frac{1}{45} & \frac{13}{360} - \frac{3\sqrt{15}}{360} & \\
\end{array} \]

where the weights are given in Example 3.3.

In the nonstiff case, the local error is given by

\[ \tilde{\psi}_i(z) = -\frac{17zh^5}{5!4000}y^{(5)}(x_{i-1}) - \frac{17h^6}{6!10000}y^{(6)}(x_{i-1}) + O(zh^6). \]

In the stiff case, the local error is given by

\[ \tilde{\psi}_i(z) = \frac{h^4}{4!100}y^{(4)}(x_{i-1}) + O(h^4/z), \]

since the term \( \tilde{b}^T(\tilde{c}^4 - 4\tilde{A}\tilde{c}^3) = 0 \) by conditions for order 5.
Thus we have,
\[
\tilde{\psi}_i(z) = \begin{cases} 
O(h^6) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0,
\]
\[
\tilde{\epsilon}_n = \begin{cases} 
O(h^5) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0. \quad \square
\]

**Order-3 symmetrizer**

The order-3 symmetrizer has the same Butcher tableau like the order-5 symmetrizer except for the weights as given in Example 3.4 (see Chapter 3, Subsection 3.4.1):
\[
\tilde{b} = \begin{bmatrix}
\frac{33}{136} + \sqrt{15} & \frac{8}{17} & \frac{33}{136} - \sqrt{15} & \frac{43}{1224} + \sqrt{15} & -\frac{4}{153} & \frac{43}{1224} - \sqrt{15}
\end{bmatrix}.
\]

In the nonstiff case, the local error is given by
\[
\tilde{\psi}_i(z) = -\frac{h^4}{4!85} y^{(4)}(x_{i-1}) + O(z h^4).
\]

In the stiff case, the local error is given by
\[
\tilde{\psi}_i(z) = -\frac{h^5}{1600z} y^{(5)}(x_{i-1}) - \frac{h^6}{6!400} y^{(6)}(x_{i-1}) + O(h^6/z),
\]
since \(1 - \tilde{b}^T \tilde{A}^{-1} \tilde{c}^4 = -2u^T A^{-1} c^4 = 0\) by the choice of the stronger damping condition given in equation (3.7).

Therefore we have,
\[
\tilde{\psi}_i(z) = \begin{cases} 
O(h^4) & \text{if nonstiff,} \\
O(h^6) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0,
\]
\[
\tilde{\epsilon}_n = \begin{cases} 
O(h^3) & \text{if nonstiff,} \\
O(h^6) & \text{if strongly stiff,}
\end{cases} \quad \text{as } h \to 0. \quad \square
\]

The analysis shows that the order reduction is suppressed by the order-3 symmetrizer.
4.2. Analysis of active and passive symmetrization

Next, we give the analysis for the symmetrized Lobatto IIIA methods defined in (3.22).

**Example 4.4 2-stage Lobatto IIIA method**

The symmetrized ITR satisfies $B(1)$ and $C(2)$ and is of order-1. The reduced tableau is given by

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 \\
1 & \frac{1}{2} & \frac{1}{2} & 0 \\
2 & \frac{1}{2} & 1 & \frac{1}{2} \\
\hline
\frac{3}{8} & \frac{1}{2} & \frac{1}{8} \\
\end{array}
$$

where

$$
\bar{A} = \begin{bmatrix}
\frac{1}{2} & 0 \\
1 & \frac{1}{2} \\
\end{bmatrix}, \quad \bar{b} = \begin{bmatrix}
\frac{1}{2} \\
\frac{1}{8} \\
\end{bmatrix}, \quad \bar{c} = \begin{bmatrix}
1 \\
2 \\
\end{bmatrix}.
$$

The method is constructed using (4.1) and the weights are given in Example 2.11.

In the nonstiff case, from (4.7) the local error is given by

$$
\tilde{\psi}_i(z) = -\frac{h^2}{4} y''(x_{i-1}) + O(h^3) + O(zh^3).
$$

In the stiff case, from (4.8) the local error is given by

$$
\tilde{\psi}_i(z) = -\frac{h^2}{4} y''(x_{i-1}) + O(h^3).
$$

Hence,

$$
\tilde{\psi}_i(z) = \begin{cases} 
O(h^2) & \text{if nonstiff,} \\
O(h^2) & \text{if strongly stiff,} 
\end{cases} \quad \text{as } h \to 0,
$$

$$
\tilde{\epsilon}_n = \begin{cases} 
O(h) & \text{if nonstiff,} \\
O(h^2) & \text{if strongly stiff,} 
\end{cases} \quad \text{as } h \to 0. \quad \square
$$
Example 4.5 3-stage Lobatto IIIA method

$\mathcal{L}_3$ satisfies $B(3)$ and $C(3)$ and is of order-3. The reduced tableau is given in Example 3.5 (see Chapter 3, Subsection 3.4.2) where

$$\begin{bmatrix}
\frac{1}{3} & -\frac{1}{24} & 0 & 0 \\
\frac{2}{3} & \frac{1}{6} & 0 & 0 \\
\frac{2}{3} & \frac{3}{8} & \frac{1}{3} & -\frac{1}{24} \\
\frac{2}{3} & \frac{3}{8} & \frac{2}{3} & \frac{1}{6}
\end{bmatrix}, \quad \begin{bmatrix}
\frac{11}{18} \\
\frac{1}{6} \\
\frac{1}{18} \\
-\frac{1}{36}
\end{bmatrix}, \quad \begin{bmatrix}
1 \\
\frac{1}{24} \\
\frac{1}{36} \\
2
\end{bmatrix},$$

where the weights are obtained in Example 3.5.

In the nonstiff case, from (4.7) the local error is given by

$$\tilde{\psi}_i(z) = \frac{h^4}{144}y^{(4)}(x_{i-1}) + \frac{zh^4}{576}y^{(4)}(x_{i-1}) + O(h^5).$$

In the stiff case, from (4.8) the local error is given by

$$\tilde{\psi}_i(z) = \frac{h^4}{192}y^{(4)}(x_{i-1}) + O(h^5) + O(h^5/z).$$

Thus we have,

$$\tilde{\psi}_i(z) = \begin{cases} O(h^4) & \text{if nonstiff, as } h \to 0, \\ O(h^4) & \text{if strongly stiff,} \end{cases}$$

$$\tilde{\epsilon}_n = \begin{cases} O(h^3) & \text{if nonstiff, as } h \to 0, \\ O(h^4) & \text{if strongly stiff,} \end{cases}$$

Active symmetrization of the 3-stage Lobatto IIIA method has no effect on the order behaviour since we observed that $\mathcal{L}_3$ without symmetrization gives $O(h^4)$ behaviour in the stiff case (see Subsection 2.1.5, Example 2.8). However, numerical results on nonlinear problems show that the active and passive symmetrization of the Lobatto IIIA methods are more efficient than $\mathcal{L}_3$ without symmetrization (see Section 6.1 and 6.2).
4.2. Analysis of active and passive symmetrization

Example 4.6 4-stage Lobatto IIIA method
\( \mathcal{L}_4 \) satisfies \( B(5) \) and \( C(4) \) and is of order-5. The reduced method is given in Example 3.6 (see Chapter 3, Subsection 3.4.2) where

\[
\mathbf{A} = \begin{bmatrix}
\frac{5}{24} - \frac{\sqrt{5}}{120} & \frac{5}{24} - \frac{13\sqrt{5}}{120} & -\frac{1}{120} \pm \frac{\sqrt{5}}{120} & 0 & 0 & 0 \\
\frac{5}{24} + \frac{13\sqrt{5}}{120} & \frac{5}{24} + \frac{\sqrt{5}}{120} & -\frac{1}{120} - \frac{\sqrt{5}}{120} & 0 & 0 & 0 \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & 0 & 0 & 0 \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & -\frac{1}{120} - \frac{\sqrt{5}}{120} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & -\frac{1}{120} - \frac{\sqrt{5}}{120} \\
\frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} & \frac{5}{12} \\
\end{bmatrix},
\]

\[
\mathbf{b} = \begin{bmatrix}
\frac{33}{80} + \frac{\sqrt{5}}{80} \frac{33}{80} - \frac{\sqrt{5}}{80} \frac{1}{120} \frac{1}{240} + \frac{\sqrt{5}}{240} \frac{1}{80} \frac{1}{120} \\
\end{bmatrix}
\]

\[
\mathbf{c} = \begin{bmatrix}
\frac{1}{2} - \frac{\sqrt{5}}{10}, \frac{1}{2} + \frac{\sqrt{5}}{10}, 1, \frac{3}{2} - \frac{\sqrt{5}}{10}, \frac{3}{2} + \frac{\sqrt{5}}{10}, 2 \\
\end{bmatrix}
\]

In the nonstiff case, from (4.7) the local error is given by

\[
\tilde{\psi}_i(z) = -\frac{z h^5}{5!1000} y^{(5)}(x_{i-1}) - \frac{11 h^6}{6!250} y^{(4)}(x_{i-1}) + O(z h^6).
\]

In the stiff case, from (4.8) the local error is given by

\[
\tilde{\psi}_i(z) = -\frac{11 h^6}{6!500} y^6(x_{i-1}) + \frac{h^5}{5!25z} y^{(5)}(x_{i-1}) + O(h^6/z).
\]

Thus we have,

\[
\tilde{\psi}_i(z) = \begin{cases}
O(h^6) & \text{if nonstiff,} \\
O(h^6) & \text{if strongly stiff,}
\end{cases}
\]

as \( h \to 0 \).

Therefore, the global error is given by

\[
\tilde{\epsilon}_n = \begin{cases}
O(h^5) & \text{if nonstiff,} \\
O(h^6) & \text{if strongly stiff,}
\end{cases}
\]

as \( h \to 0 \).

Again, active symmetrization has no effect on the order behaviour since \( \mathcal{L}_4 \) gives \( O(h^6) \) behaviour in the stiff case (see Subsection 2.1.5, Example 2.9). Nevertheless, numerical results on nonlinear problems show that symmetrization in both modes are more efficient than \( \mathcal{L}_4 \) without symmetrization (see Section 6.1 and 6.2). □
Active symmetrization \( \tilde{R}_2 \) versus \( \tilde{R}_1 \)

In the case of active symmetrization at every alternate step using \( \tilde{R}_2 = \frac{1}{2}(\mathcal{R} \circ \tilde{\mathcal{R}}) \) as given by (4.2), we have for \( k \leq 4 \),

\[
1 - \tilde{b}^T \tilde{A}^{-1}c^k = (1 - 2u^T A^{-1}e) \left( 1 - \frac{1}{2k} R(\infty) b^T A^{-1} c^k - \frac{1}{2k} b^T A^{-1} (e + c)^k \right) - \left( \frac{k}{2} \right) u^T c - \frac{1}{8} \left( \frac{k}{4} \right) u^T A^{-1} c^4.
\]

For \( \tilde{R}_2 \), again by (3.6) and (3.12), \( 1 - \tilde{b}^T \tilde{A}^{-1}c^3 = 0 \), while \( 1 - \tilde{b}^T \tilde{A}^{-1}c^4 = -\frac{1}{576} \). Taking into account that each step of \( \tilde{R}_2 \) involves \( \frac{3}{2} \) times the computations compared with \( \tilde{R}_1 \), the coefficient of \( h^4 \) is smaller by a factor \( \left( \frac{3}{4} \right)^4 \). Similar analysis can be done for \( m = 3, 4 \) and etc. Thus as \( m \) increases, the factor reduces to a smaller value.

The optimal choice of \( m \) is discussed in Subsection 4.4.1.

In the next section, we analyse the effect of passive symmetrization on the Gaussian and Lobatto IIIA solutions of the PR problem (1.4).

### 4.2.2 Passive symmetrization after \( n - 1 \) steps

In passive symmetrization at step \( n \), the global error becomes

\[
\tilde{\epsilon}_n = \tilde{R}(z) \epsilon_{n-1} + \tilde{\psi}_n(z),
\]

where \( \epsilon_{n-1} \) is the global error after \( n - 1 \) steps of the symmetric method defined by

\[
\epsilon_{n-1} = \sum_{i=1}^{n-1} R(z)^{n-1-i} \psi_i(z).
\]

\( \tilde{R}(z) \) is the stability function of the symmetrizer given in (4.9) and \( \tilde{\psi}_n(z) \) is the local error of the symmetrizer at the \( n \)-th step defined by

\[
\tilde{\psi}_n(z) = \sum_{k=2}^{\infty} \frac{h^k}{k!} y^{(k)}(x_{n-1}) \left( 1 - k \tilde{b}^T \tilde{c}^{k-1} + \tilde{z}b^T (\tilde{I} - \tilde{z}\tilde{A})^{-1} (\tilde{c}^k - k\tilde{A}\tilde{c}^{k-1}) \right). \quad (4.11)
\]

Equation (4.10) shows that there is no contribution from \( \epsilon_{n-1} \) because of the damping by \( \tilde{R}(z) \). Therefore the global error will behave like the local error \( \tilde{\psi}_n(z) \) at the \( n \)-th step.
Example 4.7 1-stage Gauss method
As given in Example 4.1, \( \tilde{\psi}_n(z) = O(h^2) \) as \( h \to 0 \) in both cases, hence by (4.10),
\[
\tilde{\epsilon}_n = O(h^2) \quad \text{as} \quad h \to 0.
\]
\( \square \)

Example 4.8 2-stage Gauss method
As given in Example 4.2, \( \tilde{\psi}_n(z) = O(h^4) \) as \( h \to 0 \) in both cases and hence, by (4.10),
\[
\tilde{\epsilon}_n = O(h^4) \quad \text{as} \quad h \to 0.
\]
We therefore see that order reduction is also suppressed by passive symmetrization. \( \square \)

Example 4.9 3-stage Gauss method
For the order-5 symmetrizer, we have
\[
\tilde{\psi}_n(z) = \begin{cases} 
O(h^6) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as} \quad h \to 0,
\]
as given in given in Example 4.3.
Hence
\[
\tilde{\epsilon}_n = \begin{cases} 
O(h^6) & \text{if nonstiff,} \\
O(h^4) & \text{if strongly stiff,}
\end{cases} \quad \text{as} \quad h \to 0.
\]
For the order-3 symmetrizer, the local error behaves like
\[
\tilde{\psi}_n(z) = \begin{cases} 
O(h^4) & \text{if nonstiff,} \\
O(h^6) & \text{if strongly stiff,}
\end{cases} \quad \text{as} \quad h \to 0,
\]
Therefore,
\[
\tilde{\epsilon}_n = \begin{cases} 
O(h^4) & \text{if nonstiff,} \\
O(h^6) & \text{if strongly stiff,}
\end{cases} \quad \text{as} \quad h \to 0.
\]
The order-5 symmetrizer has no effect on the order behaviour as order reduction from 6 to 4 still occurs. On the other hand, the order-3 symmetrizer restores the order 6 in the
strongly stiff situation.

**Example 4.10 2-stage Lobatto IIIA method**

As given in Example 4.4, $\tilde{\psi}_n(z) = O(h^2)$ as $h \to 0$ in both cases, hence by (4.10),

$$\tilde{\epsilon}_n = O(h^2) \text{ as } h \to 0.$$  

**Example 4.11 3-stage Lobatto IIIA method**

In Example 4.5, we observed that $\tilde{\psi}_n(z) = O(h^4)$ as $h \to 0$ in both cases and hence, by (4.10),

$$\tilde{\epsilon}_n = O(h^4) \text{ as } h \to 0.$$  

Similar to the order behaviour obtained for the active mode, passive symmetrization of the 3-stage Lobatto IIIA method has no effect on the order behaviour. Nevertheless, for nonlinear problems, symmetrization is seen to be more efficient than the 3-stage Lobatto without symmetrization. □

**Example 4.12 4-stage Lobatto IIIA method**

In Example 4.6, we observed that $\tilde{\psi}_n(z) = O(h^6)$ as $h \to 0$ in both cases and hence, by (4.10),

$$\tilde{\epsilon}_n = O(h^6) \text{ as } h \to 0.$$  

The numerical results for the order behaviour is given in Subsection 4.4.2.

### 4.3 Choice of strategies for extrapolation

Extrapolation can be applied in two different forms; local (active) extrapolation [36] and global (passive) [54] extrapolation. In the active mode, the extrapolated value is used to propagate the numerical solution at the next step, while in the passive mode extrapolation is carried out only at points where greater accuracy is desired and where the unextrapolated solutions are propagated.

To apply extrapolation, the base symmetric method should have an asymptotic error expansion in even powers of $h$ so that we can obtain an increment of the order by two at a time. Since in the linear case, symmetrization can restore the classical order of the Gauss and Lobatto IIIA methods, we can therefore apply extrapolation with the correct formula to eliminate the leading error term and hence increase the accuracy.
4.3. Choice of strategies for extrapolation

With symmetrization, extrapolation can also be carried out in two modes: active and passive. It is of interest to know which mode is more efficient with active and passive symmetrization.

There are six possible ways of applying extrapolation with symmetrization as illustrated in Figure 4.2 – Figure 4.7. White circles represent the symmetric values of the base method, gray circles represent the symmetrized values and the black circles represent the extrapolated values. Solutions of the symmetric methods are denoted by $y(x_1), y(x_2), \ldots, y(x_n)$, extrapolated values are denoted by $\overline{y}(x_1), \overline{y}(x_2), \ldots, \overline{y}(x_n)$, passive symmetrized values are denoted by $\tilde{y}(x_1), \tilde{y}(x_2), \ldots, \tilde{y}(x_n)$ and active symmetrized values are denoted by $\hat{y}(x_1), \hat{y}(x_2), \ldots, \hat{y}(x_n)$. Extrapolation is carried out using stepsizes $h$ and $h/2$.

1. **No symmetrization with passive extrapolation.**

![Figure 4.2: No symmetrization with passive extrapolation](image)

Figure 4.2 shows passive extrapolation at $x_3$ and $x_n$ of the symmetric solutions. The solid lines represent applying the symmetric solutions of the base method until the desired time $x_3$ and $x_n$ while the vertical dashed lines show the linear combinations of two solutions with stepsizes $h$ and $h/2$. Extrapolation is carried out whenever greater accuracy is required without propagating the extrapolated solutions.
Figure 4.3: No symmetrization with active extrapolation.

Figure 4.3 shows active extrapolation at $x_3$ and $x_n$ of the symmetric solutions. The solid lines represent the symmetric solutions of the base method while the vertical dashed lines represent the linear combinations of two solutions with stepsizes $h$ and $h/2$. Extrapolation is carried out at each step and the extrapolated values are used to propagate the next solution.

(3) Passive symmetrization with passive extrapolation.

Figure 4.4: Passive symmetrization with passive extrapolation at $x_2$.

Figure 4.4 shows passive extrapolation at $x_2$ with passive symmetrization. The solid lines represent the symmetric solutions of the base method and the gray vertical lines are the symmetrized solutions. The vertical dashed lines represent the linear combinations of two symmetrized values with stepsizes $h$ and $h/2$. Extrapolation is carried out using the symmetrized values whenever greater accuracy is required without propagating the symmetrized extrapolated solution.
4.3. Choice of strategies for extrapolation

(4) Passive symmetrization with active extrapolation.

Figure 4.5: Passive symmetrization with active extrapolation at $x_2$.

Figure 4.5 shows active extrapolation at $x_2$ with passive symmetrization. Passive symmetrization is performed using the symmetric values $y_h(x_2)$ and $y_\frac{h}{2}(x_2)$ with stepsizes $h$ and $h/2$. The gray circles are the symmetrized solutions. Extrapolation is then carried out using the symmetrized solutions with stepsizes $h$ and $h/2$ and the symmetrized extrapolated solution is propagated.

(5) Active symmetrization with passive extrapolation.

Figure 4.6: Active symmetrization with passive extrapolation at $x_2$.

Figure 4.6 shows passive extrapolation at $x_2$ with active symmetrization. The solid lines represent active symmetrization at every step while the vertical dashed lines represent the linear combinations of two symmetrized values with stepsizes $h$ and $h/2$. Extrapolation is carried out whenever greater accuracy is required without propagating the symmetrized extrapolated solution.
Chapter 4. Strategies for symmetrization and extrapolation

(6) Active symmetrization with active extrapolation.

Figure 4.7: Active symmetrization with active extrapolation at $x_2$.

Figure 4.7 shows active extrapolation at $x_2$ with active symmetrization. The solid lines represent the symmetrized solutions while the vertical dashed lines denote applying active extrapolation using the two symmetrized values with stepsizes $h$ and $h/2$. The symmetrized extrapolated value is used to propagate the solution.

The numerical results on the performance of symmetrization with extrapolation are given in Subsection 4.4.4.

4.4 Numerical results for the test problems

In this section, we present numerical results on the accuracy and efficiency of active and passive symmetrization for linear and nonlinear problems. The efficiency plots are measured in CPU time. The results are given for the Gauss and Lobatto IIIA methods of orders 4 and 6. The experimental conditions are the same so that they can be compared but are given on different plots.

In these numerical experiment, we consider on active and passive symmetrization the following:

- Optimal choice of active symmetrization.
- Order behaviour of active and passive symmetrization.
- Performance of active and passive symmetrization on different intervals.
4.4. Numerical results for the test problems

- Choice of strategy in applying active and passive symmetrization with extrapolation.

- Effect of perturbation on active and passive symmetrization with extrapolation.

In Section 4.1.1, we have shown that there are several ways of applying active symmetrization. Now we wish to find out which one of them gives the optimal choice. The numerical results are given in Subsection 4.4.1.

The second item concerns the order behaviour of symmetrization for different values of the stiffness parameter. The order behaviour when solving stiff and nonstiff problems are important especially when we are using extrapolation to improve the accuracy. Greater accuracy with extrapolation can only be observed if we use the correct extrapolation formula. The numerical results are given in Subsection 4.4.2.

The third item concerns the performance of symmetrization on different intervals. We investigate the behaviour of the symmetrizers when we are integrating on a larger interval. We have observed for the ITR that the method itself is more efficient than the method with smoothing as given in Figure 2.2b and Figure 2.4b in Chapter 2. Therefore, similar observation is expected for the 3-stage and 4-stage Lobatto IIIA methods. Nevertheless, we want to investigate whether symmetrization wins over the base method over a large interval. The numerical results are given in Subsection 4.4.3.

The fourth item concerns the efficiency of symmetrization with extrapolation. It is interesting to know which of passive or active modes of extrapolation with symmetrization is more efficient. Numerical results are given in Subsection 4.4.4.

Lastly, we examine the effects of symmetrization when there is a perturbation. Perturbation is important since in most practical problems the starting values are unknown or not known precisely. The numerical results are given in Subsection 4.4.5.

For linear problems we have chosen the PR problem (1.4). In most of the plots, we have integrated to \(x_n = 5\) with \(g(x) = \sin(x), \lambda = -10^8\) using stepsize \(h = 0.5\) except for the plots given in Figures 4.18 – 4.22. In these plots, since we are solving at different intervals, larger stepsizes are used so that we can avoid the roundoff errors as much as possible. We have also considered the perturbed PR problem given by

\[
y'(x) = \lambda(y(x) - g(x)) + g'(x), \ y(x_0) = 1. \tag{4.12}
\]
Chapter 4. Strategies for symmetrization and extrapolation

The exact solution is given by \( y(x) = g(x) + (y(x_0) - g(x_0))e^{\lambda x} \) with \( g(x) = \sin(x) \). The numerical results are given in Figure 4.26 and Figure 4.27 for \( \lambda = -10 \) and \( -\lambda = -10^6 \) respectively.

For the nonlinear problem, we consider the Kaps problem defined by

\[
\begin{align*}
y'_1 &= (\lambda - 2)y_1 - \lambda y_2^2, \quad y_1(x) = e^{-2x}, \\
y'_2 &= y_1 - y_2 - y_2^2, \quad y_2(x) = e^{-x},
\end{align*}
\]

with \( y_1(0) = 1 \) and \( y_2(0) = 1 \). It can be written in standard singular perturbation form, and for such problems, at least along a smooth solution trajectory, the stiff eigendirections do not vary along the solutions, but are more or less fixed for \( |\lambda| \ll 0 \). This means that stiff and non-stiff components are only loosely coupled. In this problem, we integrated to \( x_n = 10 \), with \( \lambda = -10^8 \) and \( h = 0.5 \). Different stepsizes are used for the plots given in Figures 4.18 – 4.22.

For the matrix linear problem, we consider

\[
\begin{align*}
y'_1 &= y_2, \quad y_1(x) = \sin(x), \\
y'_2 &= -y_1, \quad y_2(x) = \cos(x), \\
y'_3 &= \lambda(y_3 - y_1) + y_2, \quad y_3(x) = \sin(x) + \epsilon \exp(\lambda x),
\end{align*}
\]

where \( \lambda \) is the stiffness parameter and \( \epsilon \) is the perturbation added to the third initial value. The initial values are \( y_1(0) = 0, y_2(0) = 1 \) and \( y_3(0) = \epsilon \). If \( \epsilon = 0 \), the exact value of the third component becomes \( y_3(x) = \sin(x) \). The problem becomes stiff when \( L \) is a large negative number. The numerical results are given in Figure 4.28 and Figure 4.29 for \( \lambda = -10^6, \epsilon = 0 \) and \( \lambda = -10^6, \epsilon = 2 \) respectively.

4.4.1 Optimal choice of \( m \) for active symmetrization

Here, we want to discuss the optimal choice of \( m \) for active symmetrization for the method \( \hat{R}_m = \frac{1}{m}(R^{m-1} \circ \hat{R}) \).

Figure 4.8 and Figure 4.9 show the efficiency of the Gauss and Lobatto IIIA methods with active symmetrization for different choices of \( m \) applied to the PR and Kaps problems. The base methods are plotted in red dashed lines (---) while the active symmetrizations are plotted in blue. We have AS1 (---), AS2 (----), AS3 (-----) and AS4 (------) that
4.4. Numerical results for the test problems

Figure 4.8: Efficiency diagrams of Gauss and Lobatto IIIA methods with active symmetrization on optimal $m$ values applied to the PR problem.

represent applying active symmetrization at every step ($m = 1$), every alternative step ($m = 2$), every third step ($m = 3$) and every fourth step ($m = 4$) respectively.

For the PR problem, in the case of $G_2$ and $G_3$, it is shown that AS2, AS3 and AS4 are more efficient than AS1 and the base method (see Figure 4.8). We have showed analytically that applying active symmetrization at every alternative step involves $3/2$ times the computations compared with applying symmetrization at every step (see Subsection 4.2.1). Therefore, it is expected that the methods will become more efficient with increasing $m$. On the other hand, in the case of $L_3$ and $L_4$, we observe that the base methods themselves are more efficient than the symmetrization. These results are not surprising as we have seen in the order analysis for the $L_3$ and $L_4$ without symmetrizer (Example 2.8 and Example 2.9) and with symmetrizer (Example 4.5 and Example 4.6) respectively that the errors for the methods without symmetrizer contain the factor $|\lambda|$ in the denominator and will be small when $|\lambda|$ is large. Therefore, the error for the base methods will be smaller when compared to the symmetrized methods.

For the Kaps problem, similar results to the PR problem are observed (see Figure 4.9). Therefore, the numerical results given for these two problems shown tells us that as $m$ increases, the computational cost also increases. Although these experiments were only
Chapter 4. Strategies for symmetrization and extrapolation

Figure 4.9: Efficiency diagrams of Gauss and Lobatto IIIA methods with active symmetrization on optimal \( m \) values applied to the Kaps problem.

carried out up to \( m = 4 \), we can expect that the computational cost will increase for higher \( m \). Moreover, the efficiencies of AS2, AS3 and AS4 are only marginally different. Therefore, we suggest that applying active symmetrization at every two steps \( m = 2 \) is a better choice at least for the Gauss methods.

4.4.2 Order behaviour of active and passive symmetrization

Figures 4.10 – 4.13 show the numerical results observed for the PR problem of the Gauss and Lobatto IIIA methods with active and passive symmetrization for different values of the stiffness parameter. The base methods are plotted in dashed lines (---) while the passive and active symmetrization are plotted in green (--) and blue (---) respectively. In these experiments, we study the order behaviour of the active and passive symmetrization of these base methods as \(|\lambda|\) increases.

In the case of \( G_2 \), when \( \lambda = -1, -10 \) and \( \lambda = -100 \), Figure 4.10 shows order-3 behaviour for active symmetrization and order-4 behaviour for passive symmetrization. This is true for the active case because we are applying (4.1) where the method itself is of order 3. In the passive case, this is also true because we are applying an order-
4.4. Numerical results for the test problems

Figure 4.10: Accuracy diagrams of the 2-stage Gauss method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$.

3 symmetrizer locally at the end of many steps and thus the global error will show order-4 behaviour. As stiffness increases (e.g. $\lambda = -10^3$ and $\lambda = -10^4$) we observe that $G_2$ will start to show order reduction from 4 to 2 and will recover its classical order as the stepsize is further reduced. We can also observe that the symmetrizer will start to have an effect here. When the problem is very stiff (e.g. $\lambda = -10^5, -10^6$ and $\lambda = -10^8$), the symmetrizer is more accurate compared to the base method. Another interesting observation is that the active symmetrization using $\tilde{R}_2$ will behave like passive symmetrization and give order-4 behaviour.

Similar behaviour is also observed for $G_3$ in Figure 4.11. Since we have two symmetrizers (order-3 and order-5) for $\tilde{G}_3$ (see Chapter 3, Subsection 3.3.1), it is important to know which symmetrizer gives greater accuracy. When $|\lambda|$ is small (nonstiff), the base method will behave like order-6 and with order-3 symmetrizer, active and passive modes behave like order 3 and order 4 respectively. As $|\lambda|$ increases (stiff), the base method will show order reduction from 6 to 4, while active and passive modes give order-6 behaviour. Therefore, when applied with extrapolation, the global error of $\tilde{G}_2$ and $\tilde{G}_3$ can be expected to behave like $O(h^6)$ and $O(h^8)$ respectively. However, if we use the order-5 symmetrizer, active symmetrizer gives order-5 behaviour and passive symmetrizer gives order-6 behaviour in the nonstiff case. On the other hand, in the stiff case, both ac-
Figure 4.11: Accuracy diagrams of the 3-stage Gauss method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$.

tive and passive symmetrizers give order-4 behaviour. This observation is important so that we can know which symmetrizer to use for $L_3$ when solving stiff and nonstiff linear problems.

For $L_3$ the order behaviour observed is similar to $L_2$ (refer to Figure 4.12) whereas for $L_4$, when $|\lambda|$ is small (nonstiff), the base method and the passive mode behaves like order 6 while the active mode behaves like order 5. As $|\lambda|$ increases (stiff), the active and passive modes behave like $O(h^6)$ while the base method shows order reduction from 6 to 4 (refer to Figure 4.13). We also observe for the Gauss methods that as $|\lambda|$ increases, the active and passive modes of symmetrization are more accurate than the base method. In contrast to the Gauss methods, although the base methods suffer order reduction, the Lobatto IIIA methods are still more accurate than symmetrization in either mode. Similar observation is observed for the ITR with and without smoothing as shown in Figure 2.2b and Figure 2.4b respectively. In these two figures, it is shown that although smoothing dampens the oscillatory error behavior, the global error without smoothing is smaller than with smoothing.
4.4. Numerical results for the test problems

Figure 4.12: Accuracy diagrams of the 3-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$.

Figure 4.13: Accuracy diagrams of the 4-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the PR problem at $x_n = 5$. 
Figures 4.14 – 4.17 show the numerical results observed for the Kaps problem. For $\tilde{G}_2$ and $\tilde{G}_3$, when the problem is nonstiff (e.g. $\lambda = -1$), the order behaviour of active and passive symmetrization is similar to that for the PR problem. In the mildly stiff case (e.g. $\lambda = -100, -10^3$ and $\lambda = -10^4$), $\tilde{G}_2$ and $\tilde{G}_3$ exhibit order reduction from 4 to 2 and from 6 to 4 respectively, while passive and active symmetrization of $\tilde{G}_2$ and $\tilde{G}_3$ show order behaviour similar to the PR case. When the problem is stiff (e.g. $\lambda = -10^5, -10^6$ and $\lambda = -10^8$), active symmetrization of $\tilde{G}_2$ shows order-3 behaviour, while passive symmetrization still gives order-4 behaviour (see Figure 4.14). Moreover for $\tilde{G}_3$, active symmetrization gives order-4 behaviour with the order-5 symmetrizer (see Figure 4.15). We did not show the behaviour of the order-3 symmetrizer since the order is expected to be less than the order-5 symmetrizer as concluded in Table 4.1. Furthermore, numerical results can also be found in [25]. Similarly, passive symmetrization gives order-4 behaviour with the order-5 symmetrizer. This is completely different from the PR case where the classical order is restored. For $L_3$, when the problem is nonstiff and mildly stiff, the order behaviour observed is similar to $\tilde{G}_2$. However, as the problem becomes stiff ($\lambda = -10^6$ and $\lambda = -10^8$), we observe that $L_3$ gives superconvergence, where we observe order-4 behaviour (see Figure 4.16). Similar behaviour is observed for the $L_4$ (refer to Figure 4.17). This is in contrast to the observation for $\tilde{G}_2$ and $\tilde{G}_3$ in which the base method shows order reduction.

For matrix linear problems, the numerical results are given in Figure A.1 in Appendix A. It is shown that none of the base methods exhibit order reduction. Analysis for this is given in [42] where they concluded that for linear stiff matrix problems in the form of $y' = Ay$ the classical order prevails.

The order behaviour results are important so that we can know which extrapolation formula to use with linear and nonlinear stiff problems. For $\tilde{G}_3$, when solving linear stiff problems, the order-3 symmetrizer is preferred since it gives order-6 behaviour so that with extrapolation, we should expect to observe order-8 behaviour. However, when solving nonlinear problems, the order-5 symmetrizer is preferred. A summary of order behaviour is given in Table 4.1.
4.4. Numerical results for the test problems

Figure 4.14: Accuracy diagrams of the 2-stage Gauss method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_0 = 5$.

Figure 4.15: Accuracy diagrams of the 3-stage Gauss method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_0 = 5$. 
Figure 4.16: Accuracy diagrams of the 3-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_n = 5$.

Figure 4.17: Accuracy diagrams of the 4-stage Lobatto IIIA method with symmetrization for different $\lambda$ values applied to the Kaps problem at $x_n = 5$.
### 4.4. Numerical results for the test problems

#### Linear scalar problems

<table>
<thead>
<tr>
<th></th>
<th>Nonstiff case, $\lambda = -10$</th>
<th>Stiff case, $\lambda = -10^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_2$</td>
<td>$p$</td>
<td>$\tilde{p}$</td>
</tr>
<tr>
<td>$L_3$</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$G_3$</td>
<td>equation (3.18)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>equation (3.20)</td>
<td>6</td>
</tr>
<tr>
<td>$L_4$</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

#### Linear matrix problems

<table>
<thead>
<tr>
<th></th>
<th>Nonstiff case, $\lambda = -10$</th>
<th>Stiff case, $\lambda = -10^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_2$</td>
<td>$p$</td>
<td>$\tilde{p}$</td>
</tr>
<tr>
<td>$L_3$</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$G_3$</td>
<td>equation (3.18)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>equation (3.20)</td>
<td>6</td>
</tr>
<tr>
<td>$L_4$</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

#### Nonlinear Kaps problem

<table>
<thead>
<tr>
<th></th>
<th>Nonstiff case, $\lambda = -10$</th>
<th>Stiff case, $\lambda = -10^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_2$</td>
<td>$p$</td>
<td>$\tilde{p}$</td>
</tr>
<tr>
<td>$L_3$</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$G_3$</td>
<td>equation (3.18)</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>equation (3.20)</td>
<td>6</td>
</tr>
<tr>
<td>$L_4$</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.1: Summary of order for Gauss and Lobatto IIIA methods with passive ($\hat{p}$) and active ($\tilde{p}$) symmetrization for linear and nonlinear scalar and matrix problems.
Chapter 4. Strategies for symmetrization and extrapolation

4.4.3 Active and passive symmetrization on different intervals

Now we study the effects of symmetrization on different intervals. We investigate what happens when we integrate over a larger interval?

Numerical results are given in Figures 4.18 – 4.22 for the PR and Kaps problems. For the PR problem, we integrated over several intervals up to $x_n = 10, 20, 40, 60, 80$ and $x_n = 100$. However, for the nonlinear Kaps problem, we present results for intervals up to $x_n = 20$ only because the results for larger intervals are similar. Numerical results are given for $\lambda = -10^6$.

In the case of the PR problem, we observed that the selection of intervals did not affect the efficiency of the methods. We still observe for Gauss methods that symmetrization in either modes is more efficient than the base method. Moreover, as the intervals get larger, the performance of active and passive modes is only marginally more efficient (refer to Figure 4.18 and Figure 4.19). On the contrary, for the Lobatto IIIA methods, both active and passive symmetrizations are not as efficient as the base methods. It is not surprising, since similar results are seen for the stiff order behaviour results given in Subsection 4.4.2 (see Figure 4.12 and Figure 4.13). However, for larger intervals, we observed that there is a possibility that symmetrization to be superior to the base method for greater computational costs (see Figure 4.20 and Figure 4.21). Both passive and active modes of symmetrization are marginally more efficient. In addition, we also observe that at $x_n = 10$, the error for the 4-stage Lobatto IIIA method is small enough to be affected by roundoff error.

The numerical results for the Kaps problem are given in Figure 4.22 for the Gauss methods, and in Figure 4.23 for the Lobatto IIIA methods. Symmetrization of Gauss methods in both modes is more efficient than the base methods. Moreover, we also observe that the active mode is more efficient than the passive mode at $x_n = 20$ (see Figure 4.22). This is completely different from the case of the PR problem where the passive mode is shown to be superior to the active mode. Similar observation is seen for the Lobatto IIIA methods where the active and passive modes of symmetrization are more efficient than the base method. Again, this observation is contrary to the one obtained for the PR problem where the base method itself is seen to be more efficient than the symmetrization. However, for Lobatto IIIA methods, the passive mode is shown to be more efficient that the active mode (see Figure 4.23). These results are important since for practical problems, the integration over longer interval is often required.
4.4. Numerical results for the test problems

Figure 4.18: Efficiency diagrams of the 2-stage Gauss method with symmetrization at different intervals applied to the PR problem.

Figure 4.19: Efficiency diagrams of the 3-stage Gauss method with symmetrization at different intervals applied to the PR problem. We have used the order-3 symmetrizer in this experiment.
Figure 4.20: Efficiency diagrams of the 3-stage Lobatto IIIA method with symmetrization at different intervals applied to the PR problem.

Figure 4.21: Efficiency diagrams of the 4-stage Lobatto IIIA method with symmetrization at different intervals applied to the PR problem.
4.4. Numerical results for the test problems

Figure 4.22: Efficiency diagrams of the Gauss methods with symmetrization at different intervals applied to the Kaps problem.

Figure 4.23: Efficiency diagrams of the Lobatto IIIA methods with symmetrization at different intervals applied to the Kaps problem.
4.4.4 Active and passive symmetrization with extrapolation

In Section 4.3, we discussed the possibilities in applying extrapolation with symmetrization. Numerical results are given in Figure 4.24 and Figure 4.25 for the PR and Kaps problems respectively. The notation used in our experiments is given in Table 4.2.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PX</td>
<td>Passive extrapolation</td>
</tr>
<tr>
<td>AX</td>
<td>Active extrapolation</td>
</tr>
<tr>
<td>PSPX</td>
<td>Passive Symmetrization with Passive extrapolation</td>
</tr>
<tr>
<td>ASPX</td>
<td>Active Symmetrization with Passive extrapolation</td>
</tr>
<tr>
<td>PSAX</td>
<td>Passive Symmetrization with Active extrapolation</td>
</tr>
<tr>
<td>ASAX</td>
<td>Active Symmetrization with Active extrapolation</td>
</tr>
</tbody>
</table>

Table 4.2: Notation for the PR and Kaps problems on the choice of symmetrization with extrapolation.

In the case of the PR problem, for Gauss methods, we observed that active symmetrization in both modes of extrapolation (ASPX) and (ASAX) are more efficient than the passive symmetrization with active and passive extrapolation (PSAX) and (PSPX). On the other hand, for Lobatto IIIA methods, active and passive modes of the base methods with extrapolation themselves are more efficient than the symmetrization with extrapolation (see Figure 4.24). This result is not surprising since we have seen in the earlier results in Subsection 4.4.3 for the PR problem that the Lobatto IIIA methods are themselves more efficient without symmetrization.

In the case of the Kaps problem, we observed that passive symmetrization in either mode of extrapolation (PSPX) and (PSAX) is more efficient than active symmetrization with extrapolation (ASPX) and (ASAX) for all the methods (see Figure 4.25). This is because, as seen in Table 4.1, active symmetrization gives order-5 behaviour (for $L_4$), therefore with extrapolation the order is only increased by one. Conversely, passive symmetrization is superior since it gives order-6 behaviour (for $L_4$) and with extrapolation, the order is increased by two (see Figure 4.17).
4.4. Numerical results for the test problems

Figure 4.24: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the PR problem.

Figure 4.25: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the Kaps problem.
Figure 4.26: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the perturbed PR problem with $\lambda = -10$.

### 4.4.5 Effect of perturbation on symmetrization with extrapolation

In this section, we investigate the effect of perturbation on symmetrization with extrapolation. We have considered two linear problems. One is the scalar PR problem given in (4.12) and the other one is the linear matrix problem given in (4.14). For the scalar PR problem, when $\lambda = -10$, we observe that passive symmetrization with passive extrapolation (PSPX) of the Gauss methods are most efficient compared to the other methods. However, for Lobatto IIIA methods, the base method with extrapolation itself is marginally as efficient as with symmetrization in either mode (see Figure 4.26). On the other hand, when $\lambda = -10^6$, we observe that all the errors of the base methods are not decreasing (horizontal line). In this case, all the symmetrizations gave excellent efficiency. The most efficient one is the active symmetrization with passive extrapolation (ASPX) (see Figure 4.27).

For the linear matrix problem without perturbation $\epsilon = 0$, we observe that all the base methods with active and passive extrapolation are more efficient than applied with either mode of symmetrization (see Figure 4.28). This is not surprising, since we know that for nonstiff and stiff matrix problems the base methods have similar order behaviour as the passive and active symmetrization (see Table 4.1). Therefore, we should at least
4.4. Numerical results for the test problems

Figure 4.27: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the perturbed PR problem with $\lambda = -10^6$.

expect the base methods as efficient as the symmetrized methods. Likewise, when we introduce perturbation to the linear matrix problem ($\epsilon = 2$), we observe that the errors of the base methods again not decreasing. In this case, passive symmetrization with passive extrapolation (PSPX) and (PSAX) is more efficient than active symmetrization with passive and active extrapolation (ASPX) and (ASAX) (see Figure 4.29). However, it is seen that for smaller stepsizes the error of the passive symmetrization begin to increase gradually. This is because we are using the solutions of the base methods in the passive mode (see Subsection 4.1.2) but not for the active mode since we are applying the method itself (see Subsection 4.1.1).

To understand the perturbation effects on the base methods, we look at the analysis given in Chapter 2 (see Subsection 2.1.5) for the PR problem. We have the global error defined by

$$
\epsilon_n = \sum_{i=1}^{n} R(z)^{n-i} \psi_i(z) + R(z)^n \epsilon_0,
$$

In the non-perturbed case ($\epsilon_0 = 0$), there is no contribution from the last term. However with $\epsilon_0 \neq 0$, the A-stable methods such as the Gauss and Lobatto IIIA methods have stability function behaviour $|R(z)| \to 1$, as $z \to \infty$, but for L-stable methods such as the symmetrized Gauss and symmetrized Lobatto IIIA methods, $R(z) = O(1/z^2)$ as $z \to \infty$. 

111
Figure 4.28: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the linear matrix problem for $L = -10^6$ and $\epsilon = 0$.

Figure 4.29: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the linear matrix problem for $L = -10^6$ and $\epsilon = 2$. 

112
4.5 Summary of results for the test problems

This means that when the problem is stiff, L-stable methods such as the symmetrized Gauss and symmetrized Lobatto IIIA methods can dampen the effects of $\epsilon_0$ faster than the A-stable methods (see Figure 4.27 and Figure 4.29).

We have given some numerical results for the active and passive symmetrization by considering some important features when solving stiff linear and nonlinear problems. The numerical results obtained from these test problems enable us to be more careful when carrying out experiments on practical problems using constant stepsize. We now summarize the results obtained from the numerical experiments.

4.5 Summary of results for the test problems

On the optimal choice of applying active symmetrization, applying symmetrization at every alternative step requires less computational time and is marginally as efficient as applying symmetrization at every three or four steps. Therefore, in Chapter 5, applying active symmetrization at every alternative step ($m = 2$) will be considered in all the numerical experiments.

In terms of the stiffness parameter, numerical results on the PR problem shows that the Gauss and Lobatto IIIA methods exhibit order reduction but that symmetrization can restore the classical order for stiff linear problems. On the other hand, for the Kaps problem, $G_3$ with symmetrization did not restore the classical order even with an order-5 symmetrizer. Furthermore, $L_3$ shows order-2 behaviour for large stepsize value but as the stepsize decreases it gives order-4 behaviour. When the problem is extremely stiff (e.g $\lambda = -10^7$ and $\lambda = -10^8$), the method shows order-4 behaviour. Similar observation is seen for $L_4$. However, this is contrary to the results observed for the 2-stage and 3-stage Gauss methods in which we observed order reduction from 4 to 2 and 6 to 4 respectively. Therefore, the order behaviour results are important for the Gauss and Lobatto IIIA methods when we are solving nonlinear stiff problems so that we can use the correct extrapolation formula.

On the selection of intervals, for the PR problem, the behaviour of active and passive symmetrization did not have much effect as the interval gets larger. We observe that the Gauss methods are more efficient when applied with either active or passive symmetrization. However, it is difficult to decide which mode of symmetrization is more efficient.
since their behaviour are almost identical. On the other hand, for the Lobatto IIIA methods, the base methods themselves are more efficient than used with symmetrization. However, when integrating to a larger interval, symmetrization can be more effective than the base method. In the case of the Kaps problem, the choice of intervals does play an important role in the performance of symmetrization especially for the Lobatto IIIA methods. For the Gauss methods, we observe that the passive mode is superior to the active mode when \( x_{\text{end}} = 10 \). However, the active mode is more efficient when integrated to \( x_{\text{end}} = 20 \). This is in contrast to the 3-stage and 4-stage Lobatto IIIA methods where the passive mode is more efficient than the active mode. In addition, we also observe that symmetrization in both modes is more efficient than the base method.

Lastly, for the choice of applying symmetrization with extrapolation, we observe completely different results for the PR and the Kaps problems. In the case of the PR problem, for Gauss methods, we observe that active symmetrization in either mode of extrapolation is more efficient than passive symmetrization with extrapolation. However, for Lobatto IIIA methods, we observe that symmetrization with extrapolation still does not improve the efficiency of the base methods with extrapolation. This is because the base method itself is more efficient than the symmetrization. On the other hand, in the case of the Kaps problem, we observe that passive symmetrization with active and passive modes of extrapolation is more efficient than active symmetrization with either mode of extrapolation. Symmetrization in either modes of extrapolation is shown to be more efficient than the base methods in this case. Thus the choice of strategy in applying symmetrization with extrapolation is important when solving different types of linear and nonlinear problems.

The numerical results from the PR and the Kaps problems summarized the following:

For the PR problem

- The Gauss methods with symmetrization is more efficient than the Lobatto IIIA methods.

- Active and passive symmetrization are only marginally efficient.

- Active symmetrization in both modes of extrapolation are more efficient than the passive symmetrization with either mode of extrapolation.
For the Kaps problem

- The Lobatto IIIA methods with symmetrization is more efficient than the Gauss methods.

- Passive symmetrization is more efficient than active symmetrization of the Gauss and Lobatto IIIA methods. However, when integrating to a larger interval, active symmetrization is seen to be more efficient than the passive symmetrization of the Gauss methods.

- Passive symmetrization in both modes of extrapolation are more efficient than the active symmetrization with either mode of extrapolation.

Additionally, the numerical results obtained from these test problems suggested the following:

- Applying active symmetrization at every alternative step is recommended.

- Order-5 symmetrizer for the 3-stage Gauss method is preferred when solving non-stiff linear, matrix and stiff nonlinear problems while the order-3 symmetrizer is preferred when solving linear problems.

- For Lobatto IIIA methods, symmetrization in both modes give greater accuracy when integrating over a large interval.

- Symmetrization in either mode works well on a perturbed problem although the base method itself failed to converge.
To construct a good method for solving stiff problems, we need to consider three important criteria such as high accuracy, good stability and low implementation cost. Gauss methods have good stability as well as high accuracy, but are very expensive because of the fully implicit structure of their coefficients matrix. We consider why implicit methods are difficult to implement and what cost means in terms of numerical methods. This chapter is divided into two main sections. The first section discusses the implementing techniques that reduces the cost in the constant stepsize setting. The second section discusses the implementation using variable stepsize.
5.1 Implementation of implicit methods

Let us consider an $N$ dimensional stiff differential equation system

$$y'(x) = f(y(x)). \quad (5.1)$$

If the solution at $x_n = x_{n-1} + h$ is computed using an $s$-stage implicit Runge-Kutta method, we have

$$y_n = y_{n-1} + h \sum_{i=1}^{s} b_i f(Y_i), \quad (5.2)$$

where

$$Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f(Y_j), \quad i = 1, 2, \ldots, s. \quad (5.3)$$

For simplicity we shall rewrite (5.2) and (5.3) as

$$Y = e \otimes y_{n-1} + h (A \otimes I_N) F(Y), \quad (5.4a)$$

$$y_n = y_{n-1} + h (b^T \otimes I_N) F(Y), \quad (5.4b)$$

where $e = (1, \ldots, 1)^T$ and $\otimes$ denotes the Kronecker product of matrices $A \otimes B = (a_{ij} B)$ and $Y, F(Y)$ are $sN \times 1$ vectors.

If $A$ is nonsingular, we can write (5.4b) as

$$y_n = (1 - b^T A^{-1} e) y_{n-1} + (b^T A^{-1} \otimes I_N) Y, \quad (5.5)$$

as suggested by Hairer and Wanner [53]. In this case, we avoid re-evaluating the function $F(Y)$. This was seconded by the González, Montijano and Rández [46] who showed theoretically that (5.5) is expected to be more efficient than (5.4b) when solving highly stiff problems due to better stability properties. Therefore, we have used (5.5) in all our numerical computations for the update.

The costly part in implementing implicit methods is in evaluating $Y_1, Y_2, \ldots, Y_s$ due to the nonlinear terms involved. The paper of Liniger and Willoughby [63] showed that Newton’s method is advantageous in solving the nonlinear equations. For an $s$-stage
5.1. Implementation of implicit methods

method with an $N$-dimensional problem, there are $sN$ unknowns to solve in $sN$ equations. If $f$ is nonlinear, then we need to solve a large system of nonlinear equations. Thus we need to solve $sN$ nonlinear equations. There are several approaches to solving the Newton iterations. Full Newton requires evaluating the Jacobian at each iteration whereas simplified Newton, requires evaluating Jacobian only on the first approximation and may be re-evaluated later. Evaluating the Jacobian at each iteration is wasteful and expensive because of the need to update and LU-factorize the Jacobian matrix. In this section, we discuss a cheaper way of implementing implicit methods for solving stiff problems.

It is suggested in [53] as well as in [15] that the influence of round-off errors can be reduced by using smaller quantities $Z = Y - e \otimes y_{n-1}$. Then (5.4a) becomes

$$G(Z) = Z - h(A \otimes I_N)F(Z + e \otimes y_{n-1}) = 0.$$  \hspace{1cm} (5.6)

Using the Newton method to solve this nonlinear system yields

$$W \Delta Z = -G(Z),$$ \hspace{1cm} (5.7)

where

$$W = (I_s \otimes I_N) - h(A \otimes J),$$ \hspace{1cm} (5.8)

$J$ is an approximation to the Jacobian matrix $\frac{df}{dy}(x_0, y_0)$ and $\Delta Z$ is the Newton increment such that $Z^{[n]} = Z^{[n-1]} + \Delta Z$. Here $Z^{[n]}$ is the value of the $n$-th iteration. The nonlinear Jacobian matrix on the left side of (5.7) has $sN \times sN$ equations and it will take about $s^3N^3 + O(N^2)$ operations for the LU factorization and $s^2N^2 + O(N)$ operations for back substitutions. The total computational cost of solving this system of nonlinear equations involves the following:

- the evaluation of the $F$ and $G$,
- evaluation of the Jacobian $J$,
- evaluation of $W$,
- LU factorization of the iteration matrix $W$,
- back substitution to get the Newton increment vector, $\Delta Z$. 

119
Figure 5.1: Cheaper implementation of the 2-stage Gauss method for the Kaps and HIRES problems.

However, a more complicated way of reducing the computational cost have been suggested by various people [28, 45, 47]. The first paper on implementation of implicit methods was given by Butcher [14] in 1976, which describes the transformation of the linear system as well as the operation counts. The computational cost can be reduced by using a linear transformation $A = PTP^{-1}$ that leads $A$ to a triangular form. By doing this, the computational cost is reduced at each iteration to $s$ systems of dimension $N$. This similarity transformation was used by Hairer and Wanner [53] in the RADAU5 code.

This approach is very useful when the matrix $A$ has a unique eigenvalue such as the SIRK and SDIRK methods because the $s$ systems have the same matrix of dimension $N$ that must be factorized only once. However, when the matrix $A$ has different eigenvalues we do not gain so much. For example, methods like the Gauss, Lobatto and Radau have different eigenvalues, some of them are even complex.

Since we are interested in solving stiff problems, it is advisable to use either simplified Newton or full Newton. We have carried out experiments to find which of these two schemes is more efficient. Figure 5.1 shows that when $\mathcal{G}_2$ is solved using simplified Newton, it is more efficient then when solved with full Newton. The numerical results are given for Kaps problem (4.13) with $\lambda = -10^6$ at $x_n = 5$ and for the HIRES problem (6.1), integrated at $x_n = 2$. We remark that the error at every step is reduced by $h/2$.

In all our numerical experiments, we have solved the nonlinear equations using simplified Newton.
5.1. Implementation of implicit methods

Roundoff error

In solving stiff problems, the errors sometimes get close to precision and therefore we may obtain inaccurate results. The error occurring in situations like this is known as the rounding or roundoff errors. Roundoff error is error created by approximate representation of numbers. In order to minimize this, we need to modify the numerical code in such way that it uses a smaller quantity. This is called compensated summation and is due to Kahan [59]. The aim of compensated summation is not only to capture the round-off error in each individual step but also to minimize the effect of round-off error. A detailed explanation of this can be found in [15] on page 80 or [52] on page 472.

Compensated summation works when the quantity to be summed is small compared with the total being added to. It is necessary that the corrected $y$ at any stage be smaller than the corresponding term. When we are using a very small stepsize, roundoff error dominates whereas with large stepsizes, it is normally accumulated truncation errors that dominate.

Figure 5.2 shows that when solving a stiff problem, as the stepsizes decreases, the roundoff errors will accumulate and therefore the last computations of the errors may be inaccurate. However, with compensated summation, we will no longer observe this behaviour. The numerical experiment is given for the PR test problem (1.4) with $g(x) = \sin(x)$ and $\lambda = -10^6$.

Since we are dealing mainly with stiff problems we have applied compensated summation in all the numerical codes.
5.2 Variable stepsize and algorithms

When solving most problems using a constant stepsize, a very small stepsize is usually required to obtain an accurate solution. For a Runge-Kutta method, using constant stepsize, we are more likely to deliver local errors which vary from step to step according to the variability of the error functions. Thus, the approach using constant stepsize is not recommended. Variable stepsize control is important for the solution of ordinary differential equations.

5.2.1 Error estimation by local extrapolation

A classical approach to estimate the error is to integrate once with step $h$ and to integrate once again along the same interval with stepsize $h/2$ using a fixed Runge-Kutta method of order $p$. The numerical results then gives $y_1(h)$ and $y_2(h/2)$. The difference between $y_1$ and $y_2$ yields an estimate of the error. This way of obtaining an estimate of the local error is known as step halving or step doubling (see Shampine [78]). If we want to obtain a higher order approximation, then we can also perform local extrapolation. The error estimation by local extrapolation is done as follows:

- Let $\overline{y}$ be the improved result that is accurate to order $p + 1$. We then have $y_2 = \overline{y} - \epsilon$ and $y_1 = \overline{y} - 2^p \epsilon$ where $\epsilon$ is the error.

- Subtracting these two values of $y_1$ and $y_2$ yields

$$\epsilon = \frac{y_2 - y_1}{2^p - 1}. \quad (5.9)$$

- We then approximate the improved result $\overline{y}$ with

$$\overline{y} = y_2 + \epsilon. \quad (5.10)$$

In the case of explicit methods, the local error is estimated using an embedding technique where two or more methods are embedded within one step. This is due to Merson [65]. According to Enright and Hull [39], the error estimation using embedded pairs is much more efficient than the error estimation using step doubling. They have carried out experiments using embedded pairs of order four and five due to Fehlberg and the classical
four stage method of order-4 with step doubling to the DETEST problems. They found that the error estimation using embedded pair is much more efficient than the error estimation using step doubling. Error estimation approaches are also given by Ceschino and Kuntzmann [19] based on quadrature formulas and the two-step error estimators given by Montijano et al. [48].

5.2.2 Error estimation by the symmetrization

In our variable stepsize setting, the error estimation is obtained by using symmetrization. We evaluate \( y_n \) by applying \( n - 1 \) applications of the symmetric method \( \mathcal{R} \) with stepsize \( h = H/n \). Then, replace the last step by the symmetrizer \( \tilde{\mathcal{R}} \) of length \( h \), and this gives \( \hat{y}_n \).

**Example 5.1 2-stage Gauss**

The update for \( \mathcal{G}_2 \) is given by

\[
y_n = y_{n-1} - \sqrt{3} \left( Y_2^n + Y_1^n \right),
\]

where \( Y_1^n \) and \( Y_2^n \) are the internal stage values for \( \mathcal{G}_2 \) at the \( n \)-th step. The symmetrizer for \( \mathcal{G}_2 \) (3.15), is given by

\[
\hat{y}_n = \left( \frac{1}{4} + \frac{\sqrt{3}}{6} \right) (Y_1^{n+1} + Y_2^n) + \left( \frac{1}{4} - \frac{\sqrt{3}}{6} \right) (Y_1^n + Y_2^{n+1})
\]  (5.11)

Thus, the local error is given by

\[
\hat{\epsilon}_n = \hat{y}_n - y_n. \quad \Box
\]  (5.12)

**Example 5.2 3-stage Gauss**

The update for \( \mathcal{G}_3 \) is given by

\[
y_n = -y_{n-1} - \frac{1}{3} (5Y_1^n - 4Y_2^n + 5Y_3^n),
\]

where \( Y_1 \), \( Y_2 \) and \( Y_3 \) are the internal stage values. The order-5 symmetrizer of \( \mathcal{G}_3 \) (3.18) is therefore given by

\[
\hat{y}_n = \left( \frac{1}{4} + \frac{\sqrt{15}}{15} \right) (Y_1^{n+1} + Y_3^n) + \left( \frac{1}{4} - \frac{\sqrt{15}}{15} \right) (Y_1^n + Y_3^{n+1}).
\]
Thus, the local error is also given by (5.12).

Now the error estimation by the local extrapolation with symmetrization is given next.

5.2.3 Error estimation by the local extrapolation with symmetrization

- Let $\hat{y}_n$ be the improved result then we have $\hat{y}_n(h/2) = \hat{y} - \epsilon$ and $\hat{y}_n(h) = \hat{y} - 2p\epsilon$.
- Subtracting these two values of $\hat{y}_n(h)$ and $\hat{y}_n(h/2)$ yields
  \[ \hat{\epsilon}_n = \frac{\hat{y}_n(h/2) - \hat{y}_n(h)}{2p - 1}. \]  
  (5.13)
- We then approximate the improved result $\hat{y}$ with
  \[ \hat{y} = \hat{y}_n(h/2) + \epsilon. \]  
  (5.14)

Figure 5.3 and Figure 5.4 show the error estimation using the symmetrization approach for the Van der Pol problem (see Example 5.3) by the 2-stage Gauss and 3-stage Gauss methods. This experiment is carried out to show the efficiencies of the symmetrization approach to error estimation for variable stepsize.

**Example 5.3 Van der Pol (VDPOL) problem**

VDPOL is a system of ODEs of dimension 2, proposed by B. Van der Pol in the 1920’s [53]. This problem describes the behaviour of nonlinear vacuum tube circuits. The parameter $\epsilon$ is a stiffness parameter. The stiffness of the problem increases with $\epsilon$. The problem is defined by

\[
\begin{align*}
  y_1' &= y_2, \\
  y_2' &= \frac{1}{\epsilon} \left( (1 - y_1^2) y_2 - y_1 \right),
\end{align*}
\]

with $y_1(0) = 2$ and $y_2(0) = -\frac{2}{3} + \frac{10}{81} \epsilon - \frac{292}{2187} \epsilon^2 + O(\epsilon^3)$ where $\epsilon = 10^{-6}$. We have integrated to $x_n = 3$ using initial stepsize, $h = 10^{-6}$. 

124
In Figure 5.3a, we use stepsize halving and symmetrization for error estimation for \( \mathcal{B}_2 \). It is seen that, the symmetrization approach is more efficient. On the other hand, In Figure 5.3b, we have used local extrapolation to estimate the error as given by (5.9) and (5.10). It is seen that the error estimation using local extrapolation with symmetrization as given in (5.13) and (5.14) is more efficient than without symmetrization. Similar results are obtained for the 3-stage Gauss method (see Figure 5.4a and Figure 5.4b).

Figure 5.3: The error estimation of the 2-stage Gauss method with symmetrization and extrapolation applied to the Van der Pol problem.

Figure 5.4: The error estimation of the 3-stage Gauss method with symmetrization and extrapolation applied to the Van der Pol problem.

We have shown that symmetrization can also give good error estimation in the variable stepsize setting. Therefore, in our numerical experiments for the DETEST problems given in Chapter 6, the local error for the Gauss and Lobatto IIIA methods are estimated
using symmetrization. The symmetrization approach is only applicable for the symmetric Gauss and Lobatto IIIA methods. For Radau IIA methods, the error estimation is computed using local extrapolation as given in (5.9) and (5.10). For comparison purposes, the efficiency is calculated using CPU time.

In this thesis, we have used the standard variable stepsize given by Hairer and Wanner in [53]. Since the error estimation using symmetrization is new, we have summarized the details of the stepsize control in the format of an algorithm.

In our Matlab implementation of variable stepsize, there are three script files. The first script file solves the nonlinear part of the method (see Algorithm 5.2.1), the second script file computes the $n$ steps of the base method and symmetrizer using constant step size (see Algorithm 5.2.2) and the third script file is the variable code that estimates the local error using the symmetrization approach (see Algorithm 5.2.3).

Algorithm 5.2.1 is important because it considers factors like the solution of the nonlinear equations, starting values and the stopping criterion. We have used simplified Newton to solve the nonlinear equations. For starting values, it is given in [53] that one may use interpolation polynomials to obtain estimates for the starting values. However, in our numerical experiments, the starting values are chosen to be zero. For the stopping criterion, we have used the approach given by Hairer and Wanner. The convergence rate $\Theta$ can be any value less than one.

Algorithm 5.2.2 is the constant stepsize algorithm that computes $n$ steps of the base method. In this algorithm, we also compute symmetrization at the $n$-th step. Now if the $\Theta$ value given in Algorithm 5.2.1 is greater than one, the method will diverge. When this happens, the stepsize will be halved.

Algorithm 5.2.3 is the variable stepsize algorithm. In this code, the local error is estimated using symmetrization.

The algorithm is given for the 2-stage Gauss method with symmetrization. When extrapolation is performed with symmetrization, we need to modify Algorithm 5.2.3, where we need to compute one step of $\hat{y}$ with stepsize $h$ and $h/2$ (see Chapter 4, Section 4.3, item [4] and [6]) and the local error is estimated using (5.13) (see Subsection 5.2.3).
Algorithm 5.2.1: NEWTON ITERATION(ΔZ)

Set TRACE=0, Y = 0 and ΔZ = 0.

Evaluate once the Jacobian (5.8) and compute ΔZ as in (5.7).

Evaluate \( \sigma = ||\Delta Z||_\infty \).

Evaluate \( \eta = \Theta / (1 - \Theta) \).

if \( \eta \sigma \leq \kappa \cdot \text{Tol} \), where \( \kappa = 10^{-1} \)

\[
\begin{align*}
Z_{NEW} & \leftarrow Y + \Delta Z \\
\text{else} & \quad Z \leftarrow Z + \Delta Z
\end{align*}
\]

for \( i \leftarrow 1 \) to 10

\[
\begin{align*}
& \text{Evaluate (5.7) using the same Jacobian.} \\
& \text{Recalculate } \beta = ||\Delta Z||_\infty \text{ and } \Theta = \beta / \sigma \text{.}
\end{align*}
\]

if \( \Theta \geq 1 \)

\[
\begin{align*}
& \text{TRACE} = 1 \\
& \eta = 1.0
\end{align*}
\]

else if \( \Theta^{(10-i)} / (1 - \Theta) \cdot \beta > \kappa \cdot \text{Tol} \)

\[
\begin{align*}
& \text{TRACE} = 1 \\
& \text{Evaluate } \eta \leftarrow \Theta / (1 - \Theta) \text{.} \\
& \text{Evaluate } \Theta \leftarrow \max(10^{-16}, \theta)^{0.8} \text{.}
\end{align*}
\]

if \( \eta \beta \leq \kappa \cdot \text{Tol} \)

\[
\begin{align*}
& \sigma \leftarrow \beta \\
& Z \leftarrow Z + \Delta Z
\end{align*}
\]

\( Z_{NEW} \leftarrow Z \)

return \((\text{TRACE})\)
Algorithm 5.2.2: CONSTANT STEPSIZE($y$)

Set $\text{TRACE}=1$ and $\Theta = 0.2$.

\[
\text{while } \text{TRACE} \\
\hspace{1em} \text{TRACE} = 0 \\
\hspace{1em} \text{for } i \leftarrow 1 \text{ to } n \\
\hspace{2em} \text{do } \\
\hspace{3em} \text{Compute } n \text{ step of } y \text{ using Algorithm 5.2.1.} \\
\hspace{3em} \text{Store the values of } Y_1 \text{ and } Y_2 \text{ at the } n\text{-th step} \\
\hspace{3em} \text{if } (\text{TRACE} = 1) \\
\hspace{4em} \{ h \leftarrow h/2 \} \\
\hspace{1em} \text{while } \text{TRACE} \\
\hspace{2em} \text{Compute } n+1 \text{ step of } y \text{ using Algorithm 5.2.1.} \\
\hspace{2em} \text{Store the values of } Y_1 \text{ and } Y_2 \text{ at the } n+1\text{-th step} \\
\hspace{2em} \text{if } (\text{TRACE} = 1) \\
\hspace{3em} \{ h \leftarrow h/2 \} \\
\hspace{2em} \text{Compute symmetrization } \hat{y}_n \text{ at the } n\text{-th step using (5.11).} \\
\hspace{1em} \text{if } h < h_0 \\
\hspace{2em} \{ \text{TRACE} = 1 \} \\
\hspace{1em} \text{return } y \\
\hspace{1em} \text{return } \hat{y}_n \\
\hspace{1em} \text{return } TR \\
\hspace{1em} \text{return } (h = h_{\text{out}}) 
\]
5.2. Variable stepsize and algorithms

Algorithm 5.2.3: VARIABLE STEPSIZE\((h)\)

Set \(x = x_0, p = 4\) if we are using order 4 method.

Set \(h_{\text{max}} = (x_n - x)/5\).

Set \(h_{\text{min}} = (x_n - x)/2^{-8}\).

Set \(h = \max([h_0, (x_n - x)/10^7])\).

\[
\text{while } (x < x_n) \text{ and } (h \geq h_{\text{min}})
\]

\[
\text{if } (x + h > x_n)
\]

\[
\left\{ \begin{array}{l}
\quad h \leftarrow x_n - x
\end{array} \right.
\]

Compute one step of \(y\) and \(\hat{y}\) using Algorithm 5.2.2.

Obtain \(h_{\text{out}}\) from Algorithm 5.2.2.

\[
\text{if TRACE } \quad h = h_{\text{out}}
\]

\[
\left\{ \begin{array}{l}
\quad \delta \leftarrow ||\hat{e}||_{\infty}
\quad \tau \leftarrow \text{Tol.max}(||y||_{\infty}, 1.0)
\end{array} \right.
\]

\[
\text{if } (\delta \leq \tau)
\]

\[
\left\{ \begin{array}{l}
\quad x \leftarrow x + h
\quad y \leftarrow \hat{y}
\end{array} \right.
\]

\[
\text{if } (\delta = 0)
\]

\[
\left\{ \begin{array}{l}
\quad h \leftarrow \min([h_{\text{max}}, 4.h, 0.9.h.(\tau/\delta)^{p+1}])
\quad \text{Accept } h.
\end{array} \right.
\]

\[
\text{else if}
\]

\[
\left\{ \begin{array}{l}
\quad \text{Reject } h.
\quad h \leftarrow h_{\text{max}}([0.25, 0.9.(\tau/\delta)^{p+1}])
\end{array} \right.
\]
In this chapter, we focus on the efficiency of the symmetrized Gauss and Lobatto IIIA methods with extrapolation for more practical problems using constant stepsize and variable stepsize. We draw conclusions on the practical performance of the symmetrized methods for different classes of problems. Although a symmetrizer is constructed so that it is L-stable, we also wish to study the performance of symmetrizers for nonstiff and mildly stiff problems. In the constant stepsize setting, numerical experiments were carried out for 15 linear problems and 3 nonlinear problems taken from the literature (see Appendix B). In addition to these nonlinear problems, we also gave numerical results for the HIRES, Brusselator and Oregonator problems. In the variable stepsize setting, numerical experiments are carried out on the DETEST problems given in [38].
6.1 Results for constant stepsize

In addition to the nonlinear problems given in Appendix B, the numerical results are also given for the following problems:

- **HIRES**
  It is a chemical reaction of 8 reactants proposed by Schäfer [73] in 1975. He studied the photomorphogenesis of a plant by using a high-frequency-controlled light source to grow the plant. The mathematical model was given by Hairer and Wanner [53] where the name HIRES comes from ‘High Irradiance RESponse’. The detailed description of this problem is given in [83]. HIRES is a nonlinear system of 8 dimensions. It is a moderately stiff problem. The problem is of the form

\[
\begin{align*}
y_1' &= -1.71y_1 + 0.43y_2 + 8.32y_3 + 0.0007, \\
y_2' &= 1.71y_1 - 8.75y_2, \\
y_3' &= -10.03y_3 + 0.43y_4 + 0.035y_5, \\
y_4' &= 8.32y_2 + 1.71y_3 - 1.12y_4, \\
y_5' &= -1.745y_5 + 0.43y_6 + 0.43y_7, \\
y_6' &= -280y_6y_8 + 0.69y_4 + 1.71y_5 - 0.43y_6 + 0.69y_7, \\
y_7' &= 280y_6y_8 - 1.81y_7, \\
y_8' &= -y_7',
\end{align*}
\]

with \(0 \leq x \leq 321.8122\). The initial value is given by \(y_0 = [1, 0, 0, 0, 0, 0, 0, 0.0057]^T\). We integrated to \(x_n = 10\) using \(h = 0.5\).

- **Oregonator (OREGO) problem**
  In the early 1970’s, Field, Körös, and Noyes [41] proposed the Oregonator model for the Belousov-Zhabotinskii (BZ) reaction while at the University of Oregon. The BZ reaction is one of a class of reactions that serve as a classical example of non-equilibrium thermodynamics, resulting in the establishment of a nonlinear chemical
oscillator. OREGO is a nonlinear stiff system of 3 dimensions.

\begin{align*}
  y'_1 &= 77.27(y_2 + y_1(1 - 8.375 \times 10^{-6} y_1 - y_2)), \\
  y'_2 &= \frac{1}{77.27}(y_3 - (1 + y_1)y_2), \\
  y'_3 &= 0.161(y_1 - y_3), \quad x \in [0, 3600],
\end{align*}

(6.2)

with \( y_1(0) = 1, y_2(0) = 2 \) and \( y_3(0) = 3 \). We integrated to \( x_n = 5 \) using \( h = 0.5 \).

- **Brusselator (BRUSS) problem**

  Brusselator is an autocatalytic oscillating chemical reaction problem modelled by Ilya Prigogine and his collaborators at the Free University of Brussels [53]. It is a system of two ODE’s given by

\begin{align*}
  y'_1 &= 1 + y_1^2y_2 - 4y_1, \\
  y'_2 &= 3y_1 - y_1^2y_2.
\end{align*}

(6.3)

with \( y_1(0) = 1.5 \) and \( y_2(0) = 3 \). We integrated to \( x_n = 1 \) using \( h = 0.05 \).

The notation for the numerical experiments is given in Table 6.1. In order to observe the efficiency of the symmetrized methods, we made comparison with the 2-stage and 3-stage Radau IIA methods of order 3 and order 5 and the 3-stage and 4-stage Lobatto IIIC methods of order 4 and order 6 respectively.

A symmetrizer is constructed so that it preserves the asymptotic error expansion of the associated symmetric method in even powers of the stepsize. The extrapolation technique can then be used to increase its efficiency. Extrapolation involves taking a linear combination of solutions with stepsizes \( h \) and \( h/2 \) to eliminate the leading error term (refer Section 2.2). By doing so, it requires additional computational time. The question is whether with extrapolation, the base methods can still be as efficient as methods such as Radau IIA and Lobatto IIIC. Extrapolation is not recommended for non-symmetric methods because the order can only be increased by one at a time. But it is still interesting to know when extrapolation is applied to the Radau IIA and the Lobatto IIIC methods, how they compare with the symmetric methods where the order can be increased by two at a time with extrapolation. Comparison can still be carried out since we are measuring the efficiency using CPU time. Out of the 15 problems listed in Appendix B, we chose 6 problems since the rest of the problems give similar results. The problems are classified according to the stiffness. We have considered three different
### Table 6.1: Notation for numerical experiments

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
<td>2-stage Gauss method</td>
</tr>
<tr>
<td>G3</td>
<td>3-stage Gauss method</td>
</tr>
<tr>
<td>L3</td>
<td>3-stage Lobatto IIIA method</td>
</tr>
<tr>
<td>L4</td>
<td>4-stage Lobatto IIIA method</td>
</tr>
<tr>
<td>R2</td>
<td>2-stage Radau IIA method</td>
</tr>
<tr>
<td>R3</td>
<td>3-stage Radau IIA method</td>
</tr>
<tr>
<td>L3C3</td>
<td>3-stage Lobatto IIIC method</td>
</tr>
<tr>
<td>L3C4</td>
<td>4-stage Lobatto IIIC method</td>
</tr>
<tr>
<td>PS</td>
<td>Passive Symmetrization</td>
</tr>
<tr>
<td>AS1</td>
<td>Active Symmetrization at every step</td>
</tr>
<tr>
<td>AS2</td>
<td>Active Symmetrization at every alternate step</td>
</tr>
<tr>
<td>AX</td>
<td>Active eXtrapolation</td>
</tr>
<tr>
<td>PX</td>
<td>Passive eXtrapolation</td>
</tr>
<tr>
<td>PSPX</td>
<td>Passive Symmetrization with Passive eXtrapolation</td>
</tr>
<tr>
<td>PSAX</td>
<td>Passive Symmetrization with Active eXtrapolation</td>
</tr>
<tr>
<td>ASPX</td>
<td>Active Symmetrization at every step with Passive eXtrapolation</td>
</tr>
<tr>
<td>ASAX1</td>
<td>Active Symmetrization at every step with Active eXtrapolation</td>
</tr>
</tbody>
</table>
classes of problems; nonstiff, mildly stiff and stiff problems. In addition, we have studied the performance of symmetrization on high dimension problems.

In the constant stepsize settings, we wish to know the answer to several questions such as

- Can the symmetrized methods still be as efficient as the base methods when they have the same order behaviour for nonstiff linear and matrix problems?
- Which type of problems are suitable for extrapolation of symmetrized methods?
- Which mode of symmetrization is more efficient when applied with extrapolation?
- Which mode of extrapolation is more efficient for certain types of problems?

The numerical results given for the constant stepsize settings show the performance of symmetrization with extrapolation for linear and nonlinear problems given in Appendix B.1. The numerical results for the 2-stage Gauss and the 3-stage Lobatto IIIA methods are given in one graph while the numerical results for the 3-stage Gauss and the 4-stage Lobatto IIIA methods are given in another graph. Although we have plotted them separately, we can still compare them since we have run the code for the same interval using the same stepsizes. Additional graphs for the performance of symmetrization without extrapolation are given in Figures A.2 – Figure A.5 (see Appendix A).

For linear problems, the numerical results are given in Figures 6.1 and Figure 6.2. LP1 is a scalar moderate stiff problem given by Curtiss and Hirschfelder in [29]. For this problem, we observed that \( L_3 \) and \( L_4 \) with passive symmetrization and passive extrapolation (L3 PSPX and L4 PSPX) are more efficient than the symmetrized Gauss and the Radau methods with extrapolation. Similar results are observed for the LP4 problem. LP9 is a two dimensional linear nonstiff problem. For this type of problem, it is shown that, the Gauss and Lobatto IIIA methods with passive symmetrization and passive extrapolation (PSPX) are the most efficient. However, the base methods themselves with extrapolation can be as efficient as the symmetrized methods. This is not surprising since the base methods are of order-4 and order-6 respectively for nonstiff matrix problems hence we will get increase accuracy with extrapolation (see Table 4.1). Although they are of the same order as the symmetrized methods, we still observe that the symmetrized methods are more efficient. Similar observation is seen for the two dimensional stiff problems (LP10). LP14 is a three dimension nonstiff problem. For this
problem, we observed that Radau 5 with passive extrapolation (R3 PX) is as efficient as the symmetrized 3-stage Gauss and the symmetrized 4-stage Lobatto IIIA methods with passive extrapolation (G3 PSPX and L4 PSPX). Lastly, LP15 is a six dimensional problem. The stiffness ratio is controlled by the $\alpha$ value (See Appendix B.1). When alpha is small ($\alpha = 10$), we observe that the base methods with extrapolation are as efficient as the symmetrized methods with extrapolation. However, when the $\alpha$ value is large ($\alpha = 10^6$), all the other symmetrized methods behave the same as $\alpha = 10$ but not for the base methods where they perform very poorly in this case. Numerical results for $\alpha = 10^6$ are given in Figure A.6 in Appendix A.

For nonlinear problems, the numerical results are given in Figure 6.3 and Figure 6.4. Figure 6.3 show the efficiency diagrams of the symmetrized 2-stage Gauss and symmetrized 3-stage Lobatto IIIA methods with extrapolation applied to six nonlinear problems. NP1 is a two dimension stiff nonlinear problem. For this problem, we observe that the PSPX by the symmetrized 2-stage Gauss and the symmetrized 3-stage Lobatto IIIA methods are more efficient than the other methods. However, for the order-6 methods, we observe that L4 PSPX is more efficient than the G3 PSPX (see Figure 6.4). The symmetrized 3-stage Gauss method is not as efficient as the symmetrized 4-stage Lo-
6.1. Results for constant stepsize

Figure 6.2: Efficiency diagrams of the symmetrized 3-stage Gauss and symmetrized 4-stage Lobatto IIIA methods with extrapolation applied to the linear test problems B.1.1 using constant stepsize.

The Lobatto IIIA method since for stiff nonlinear problems, the method behaves like order 4 rather than order 6. NP2 is a nonstiff two-dimensional nonlinear problem. For this problem, we observe that the active and passive modes of the 2-stage Gauss method with extrapolation are marginally more efficient that the symmetrization with extrapolation in either mode. The 2-stage Gauss method is more efficient than the 3-stage Lobatto IIIA method. On the other hand, the 4-stage Lobatto IIIA method is more efficient than the 3-stage Gauss method. NP3 is a nonstiff three dimension nonlinear problem. It is shown that L3 PSPX is as efficient as L3 AX. We also observe that the 3-stage Lobatto IIIA method with either mode of extrapolation/symmetrization is more efficient that the 2-stage Gauss method. In the case of order-6 symmetrized methods, we observe that G3 PSPX, G3 PX and G3 AX are more efficient than the corresponding Lobatto IIIA methods with symmetrization/extrapolation. One important observation for nonlinear problems is that when the problem is nonstiff, the base methods with extrapolation is shown to be as efficient as the symmetrized methods using either mode of extrapolation (See BRUS problems).
Figure 6.3: Efficiency diagrams of the symmetrized 2-stage Gauss and symmetrized 3-stage Lobatto IIIA methods with extrapolation applied to the nonlinear problems using constant stepsize.

Figure 6.4: Efficiency diagrams of the symmetrized 3-stage Gauss and symmetrized 4-stage Lobatto IIIA methods with extrapolation applied to the nonlinear problems using constant stepsize.
6.1. Results for constant stepsize

Figure 6.5: Efficiency diagrams of order-4 methods with high level extrapolation applied to the nonlinear problems using constant stepsize.

We have also carried out experiments for higher level extrapolation. Figure 6.5 shows the efficiency diagrams of the high level extrapolation of the IMR and ITR and the 2-stage Gauss and the 3-stage Lobatto IIIA methods with passive symmetrization. These methods are compared to the 3-stage Gauss and the 4-stage Lobatto IIIA methods without symmetrization and extrapolation. Numerical experiments are given for the HIRES, Oregonator and Brusselator problems. It is observed that the 3-stage Gauss method is the most efficient among these methods. This is followed by the 4-stage Lobatto IIIA method. However, as the stepsize decreases and the computational cost increases, it is seen that the second level extrapolations of the IMR and ITR of order 6 are more efficient than the passive symmetrization of the 2-stage Gauss and 3-stage Lobatto IIIA method.

Figure 6.6 shows the efficiency diagram of the order-6 methods such as the first and second level extrapolation of the 2-stage Gauss and the 3-stage Lobatto IIIA methods with symmetrization and the 3-stage Gauss and the 4-stage Lobatto IIIA methods with passive symmetrization. They are compared with the first level extrapolation of the 3-stage Gauss and the 4-stage Lobatto IIIA methods. Numerical experiments are also given for the HIRES, Oregonator and Brussalator problems. The results show that the second level extrapolation of the 2-stage Gauss method with symmetrization is more efficient than the other methods for the Brusselator and Oregonator problems. For the HIRES problem, it is observed that the first level extrapolation of the 4-stage Lobatto IIIA method is as competitive as the 2-stage Gauss method with symmetrization and second level extrapolation.

In the next section, we give numerical results in the variable stepsize setting.
6.2 Results for variable stepsize

In the variable stepsize settings, the symmetrized Gauss and Lobatto IIIA methods with extrapolation are compared with the 2-stage and 3-stage Radau IIA methods. The efficiency is measured in CPU time. In this setting, the local errors for the Gauss and Lobatto IIIA methods are estimated using the difference between the approximations for the base method and the symmetrizer (see Subsection 5.2.2). For the Radau IIA methods, the local errors are estimated using the local extrapolation approach with step-size \( h \) and \( h/2 \) (see Subsection 5.2.1). The problems are solved with the tolerances \( \text{tol} = 10^{-i}, i = 2, 3, \ldots, 10 \) and the nonlinear equations are solved using simplified Newton. Notation used for the numerical experiments is given in Table 6.1. Numerical results are given in Figure 6.7 – Figure 6.11 according to the class of the problems. Detailed information on the problems is given in [38].

Each method is plotted in a different colour. The red dotted lines (\( \cdots \)) represent the symmetrized 2-stage Gauss with active and passive extrapolation (G2 ASAX1 and G2 PSAX), blue dotted lines (\( \cdots \)) represent the symmetrized 3-stage Lobatto methods with active and passive extrapolation (L3 ASAX1 and L3 PSAX), green lines represent (---) the symmetrized 3-stage Gauss with active and passive extrapolation (G3 ASAX1 and G3 PSAX) and the purple lines (---) represent the symmetrized 4-stage Lobatto methods with active and passive extrapolation (L4 ASAX1 and L4 PSAX). Comparisons are made with the 2-stage and the 3-stage Radau IIA methods (light blue lines, ---).
6.2. Results for variable stepsize

Class A and B are linear problems. Numerical results for class A problems are given in Figure 6.7. It is observed that G3 ASAX1 is most efficient followed by L4 ASAX1. For class B problems, the numerical results are given in Figure 6.8. For problem B1, we observe that G3 PSPX is most efficient, while for the problems B2 – B5, R3 with active extrapolation is slightly more efficient than the symmetrized Gauss and Lobatto IIIA methods. Class C – Class E are nonlinear problems and the numerical results are given in Figure 6.9 – Figure 6.11. Again, we observe that although the 3-stage Radau IIA method with active extrapolation (R3 AX) is the most efficient, the symmetrized 3-stage Gauss and the symmetrized 4-stage Lobatto IIIA methods with active extrapolation are almost as efficient as the Radau IIA method.
Chapter 6. Extended Numerical Experiments

Figure 6.8: Efficiency diagrams of the symmetrized Gauss and Lobatto IIIA methods with extrapolation applied to the DETEST class B problems using variable stepsize.

Figure 6.9: Efficiency diagrams of the symmetrized Gauss and Lobatto IIIA methods with extrapolation applied to the DETEST class C problems using variable stepsize.
6.2. Results for variable stepsize

Figure 6.10: Efficiency diagrams of the symmetrized Gauss and Lobatto IIIA methods with extrapolation applied to the DETEST class D problems using variable stepsize.

Figure 6.11: Efficiency diagrams of the symmetrized Gauss and Lobatto IIIA methods with extrapolation applied to the DETEST class E problems using variable stepsize.
6.3 Performance of certain methods

In the previous section, we observed that the G3 ASAX1, G3 PSAX, L4 ASAX1 and L4 PSAX are the most promising methods in the variable stepsize setting. We have therefore applied these methods to selected DETEST problems. Numerical results are shown in Table 6.2 – 6.13. In these tables, we have considered some of the factors such as:

- Time: the total CPU time taken to reach \( x_n \). The CPU time takes into account the number of rejected and accepted steps.
- Rejected steps: the total number of rejected steps.
- Accepted steps: the total number of accepted steps.
- #J: the number of evaluations of the Jacobian.
- #f: the number of evaluations of the derivative function.
- #LU: the number of LU-decompositions.
- #BW: the number of forward and backward substitutions.

Table 6.2 shows the numerical results for the symmetrized 3-stage Gauss, symmetrized 4-stage Lobatto IIIA and 3-stage Radau IIA methods with extrapolation applied to the A1 problem for different tolerances. It is seen that for smaller tolerances \(10^{-2}\), R3 AX has less LU decomposition, function and Jacobian evaluations and backward substitutions, although the method required more computational time than G3 PSAX and L4 PSAX. However, for stringent tolerance \(10^{-8}\), R3 AX requires more steps than the symmetrized methods with extrapolation. R3 AX has two rejected steps compared to the G3 PSAX and L4 PSAX methods. With all methods, it is noted that, when the tolerance is small, all the factors gets bigger. Similar results are seen for the A3 problem except that for stringent tolerance \(10^{-8}\), R3 AX has one rejected step (see Table 6.3). Similar observations are seen for the other classes of problems.

It is hard to draw any definite conclusion based on the results obtained in the Tables. For some problems, we observed that although G3 PSAX and L4 PSAX require less computational time than the R3 AX, they need more Jacobian and function evaluations. Similar observation is made by Enright et al. in their STIFF DETEST paper [38]. The Table 1 on page 23 in that paper shows that although the GEAR method required less
computational cost than the SDBASIC, TRAPEX, GENK and IMPRK methods, the method still required more function and Jacobian evaluations. The 3-stage Radau method is a well known method for solving stiff problems. Many years of experience are required to develop a good solver. However, we can say that the symmetrized 3-stage Gauss and symmetrized 4-stage Lobatto IIIA methods are good enough to solve most of the STIFF DETEST problems with satisfactory efficiency. The variable stepsize algorithm that we have constructed is reasonable enough to give us the results. We have tried solving the DETEST problems using the implicit midpoint rule and implicit trapezoidal rule with smoothing. For most of the problems, these methods fail to converge and we therefore did not include the results here.
Chapter 6. Extended Numerical Experiments

Table 6.2: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST A1 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.2819</td>
<td>0</td>
<td>10</td>
<td>150</td>
<td>210</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0555</td>
<td>0</td>
<td>10</td>
<td>150</td>
<td>210</td>
<td>50</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2535</td>
<td>0</td>
<td>10</td>
<td>150</td>
<td>198</td>
<td>50</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0590</td>
<td>0</td>
<td>10</td>
<td>150</td>
<td>198</td>
<td>50</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1594</td>
<td>0</td>
<td>10</td>
<td>90</td>
<td>138</td>
<td>30</td>
<td>46</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0494</td>
<td>0</td>
<td>13</td>
<td>195</td>
<td>273</td>
<td>65</td>
<td>91</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0424</td>
<td>0</td>
<td>12</td>
<td>180</td>
<td>252</td>
<td>60</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0592</td>
<td>0</td>
<td>13</td>
<td>195</td>
<td>267</td>
<td>65</td>
<td>89</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0522</td>
<td>0</td>
<td>12</td>
<td>180</td>
<td>252</td>
<td>60</td>
<td>84</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0347</td>
<td>0</td>
<td>13</td>
<td>117</td>
<td>195</td>
<td>39</td>
<td>65</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.0654</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>399</td>
<td>95</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0564</td>
<td>0</td>
<td>17</td>
<td>255</td>
<td>357</td>
<td>85</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0795</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>399</td>
<td>95</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0726</td>
<td>0</td>
<td>17</td>
<td>255</td>
<td>357</td>
<td>85</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0508</td>
<td>1</td>
<td>21</td>
<td>198</td>
<td>330</td>
<td>66</td>
<td>110</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.1277</td>
<td>2</td>
<td>34</td>
<td>540</td>
<td>756</td>
<td>180</td>
<td>252</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0987</td>
<td>1</td>
<td>27</td>
<td>420</td>
<td>588</td>
<td>140</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1563</td>
<td>2</td>
<td>34</td>
<td>540</td>
<td>756</td>
<td>180</td>
<td>252</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1219</td>
<td>1</td>
<td>27</td>
<td>420</td>
<td>588</td>
<td>140</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0981</td>
<td>2</td>
<td>41</td>
<td>387</td>
<td>645</td>
<td>129</td>
<td>215</td>
</tr>
</tbody>
</table>
### Table 6.3: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST A3 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3999</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>336</td>
<td>80</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0994</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>336</td>
<td>80</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2992</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>336</td>
<td>80</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1354</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>336</td>
<td>80</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1885</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>240</td>
<td>48</td>
<td>80</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0540</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>399</td>
<td>95</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0474</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>399</td>
<td>95</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0727</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>399</td>
<td>95</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0677</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>399</td>
<td>95</td>
<td>133</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0386</td>
<td>0</td>
<td>20</td>
<td>180</td>
<td>300</td>
<td>60</td>
<td>100</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.0768</td>
<td>0</td>
<td>28</td>
<td>420</td>
<td>588</td>
<td>140</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0691</td>
<td>0</td>
<td>25</td>
<td>375</td>
<td>525</td>
<td>125</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0952</td>
<td>0</td>
<td>28</td>
<td>420</td>
<td>588</td>
<td>140</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0858</td>
<td>0</td>
<td>25</td>
<td>375</td>
<td>525</td>
<td>125</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0583</td>
<td>0</td>
<td>31</td>
<td>279</td>
<td>465</td>
<td>93</td>
<td>155</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.1372</td>
<td>0</td>
<td>49</td>
<td>735</td>
<td>1029</td>
<td>245</td>
<td>343</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1034</td>
<td>0</td>
<td>39</td>
<td>585</td>
<td>819</td>
<td>195</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1696</td>
<td>0</td>
<td>49</td>
<td>735</td>
<td>1029</td>
<td>245</td>
<td>343</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1286</td>
<td>0</td>
<td>39</td>
<td>585</td>
<td>819</td>
<td>195</td>
<td>273</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1111</td>
<td>1</td>
<td>57</td>
<td>522</td>
<td>870</td>
<td>174</td>
<td>290</td>
</tr>
</tbody>
</table>
Table 6.4: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST B2 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3222</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>567</td>
<td>55</td>
<td>189</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0848</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>570</td>
<td>55</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.3205</td>
<td>0</td>
<td>14</td>
<td>231</td>
<td>744</td>
<td>77</td>
<td>248</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0979</td>
<td>0</td>
<td>14</td>
<td>231</td>
<td>744</td>
<td>77</td>
<td>248</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2055</td>
<td>0</td>
<td>14</td>
<td>147</td>
<td>471</td>
<td>49</td>
<td>157</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.1098</td>
<td>0</td>
<td>17</td>
<td>279</td>
<td>1176</td>
<td>93</td>
<td>392</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0973</td>
<td>0</td>
<td>16</td>
<td>261</td>
<td>1113</td>
<td>87</td>
<td>371</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1340</td>
<td>0</td>
<td>17</td>
<td>279</td>
<td>1197</td>
<td>93</td>
<td>399</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1164</td>
<td>0</td>
<td>16</td>
<td>261</td>
<td>1128</td>
<td>87</td>
<td>376</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0763</td>
<td>0</td>
<td>18</td>
<td>186</td>
<td>738</td>
<td>62</td>
<td>246</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.1839</td>
<td>0</td>
<td>29</td>
<td>489</td>
<td>2214</td>
<td>163</td>
<td>738</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1944</td>
<td>0</td>
<td>28</td>
<td>477</td>
<td>2196</td>
<td>159</td>
<td>732</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2430</td>
<td>0</td>
<td>31</td>
<td>534</td>
<td>2451</td>
<td>178</td>
<td>817</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.2157</td>
<td>0</td>
<td>28</td>
<td>483</td>
<td>2277</td>
<td>161</td>
<td>759</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1306</td>
<td>0</td>
<td>30</td>
<td>321</td>
<td>1410</td>
<td>107</td>
<td>470</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.3612</td>
<td>0</td>
<td>53</td>
<td>891</td>
<td>4332</td>
<td>297</td>
<td>1444</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.3215</td>
<td>0</td>
<td>45</td>
<td>768</td>
<td>3855</td>
<td>256</td>
<td>1285</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.4655</td>
<td>0</td>
<td>56</td>
<td>975</td>
<td>4788</td>
<td>325</td>
<td>1596</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.4329</td>
<td>0</td>
<td>51</td>
<td>903</td>
<td>4530</td>
<td>301</td>
<td>1510</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2459</td>
<td>0</td>
<td>58</td>
<td>618</td>
<td>2850</td>
<td>206</td>
<td>950</td>
</tr>
</tbody>
</table>
6.3. Performance of certain methods

Table 6.5: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST B5 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3497</td>
<td>17</td>
<td>258</td>
<td>747</td>
<td>86</td>
<td>249</td>
<td></td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1125</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>801</td>
<td>95</td>
<td>267</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.3289</td>
<td>0</td>
<td>19</td>
<td>303</td>
<td>840</td>
<td>101</td>
<td>280</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1363</td>
<td>0</td>
<td>20</td>
<td>309</td>
<td>822</td>
<td>103</td>
<td>274</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2172</td>
<td>0</td>
<td>21</td>
<td>210</td>
<td>573</td>
<td>70</td>
<td>191</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.1916</td>
<td>37</td>
<td>576</td>
<td>1722</td>
<td>192</td>
<td>574</td>
<td></td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1823</td>
<td>0</td>
<td>37</td>
<td>576</td>
<td>1740</td>
<td>192</td>
<td>580</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2625</td>
<td>0</td>
<td>38</td>
<td>597</td>
<td>1818</td>
<td>199</td>
<td>606</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.2270</td>
<td>0</td>
<td>37</td>
<td>576</td>
<td>1785</td>
<td>192</td>
<td>595</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1322</td>
<td>0</td>
<td>43</td>
<td>411</td>
<td>1179</td>
<td>137</td>
<td>393</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.4129</td>
<td>82</td>
<td>1299</td>
<td>3837</td>
<td>433</td>
<td>1279</td>
<td></td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.3643</td>
<td>0</td>
<td>70</td>
<td>1083</td>
<td>3414</td>
<td>361</td>
<td>1138</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.5354</td>
<td>1</td>
<td>82</td>
<td>1302</td>
<td>3963</td>
<td>434</td>
<td>1321</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.4725</td>
<td>0</td>
<td>75</td>
<td>1179</td>
<td>3810</td>
<td>393</td>
<td>1270</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.3071</td>
<td>1</td>
<td>97</td>
<td>939</td>
<td>2667</td>
<td>313</td>
<td>889</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.9151</td>
<td>2</td>
<td>177</td>
<td>2778</td>
<td>8571</td>
<td>926</td>
<td>2857</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.7276</td>
<td>1</td>
<td>138</td>
<td>2163</td>
<td>7167</td>
<td>721</td>
<td>2389</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>1.1330</td>
<td>2</td>
<td>180</td>
<td>2859</td>
<td>9036</td>
<td>953</td>
<td>3012</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.9037</td>
<td>1</td>
<td>140</td>
<td>2208</td>
<td>7488</td>
<td>736</td>
<td>2496</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.6637</td>
<td>1</td>
<td>212</td>
<td>2010</td>
<td>5940</td>
<td>670</td>
<td>1980</td>
</tr>
</tbody>
</table>
Table 6.6: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST C2 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3166</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>270</td>
<td>55</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0625</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>270</td>
<td>55</td>
<td>90</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2819</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>222</td>
<td>55</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0621</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>222</td>
<td>55</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1784</td>
<td>0</td>
<td>11</td>
<td>99</td>
<td>144</td>
<td>33</td>
<td>48</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0731</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>381</td>
<td>70</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0577</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>381</td>
<td>70</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0657</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>369</td>
<td>70</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0662</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>357</td>
<td>70</td>
<td>119</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0372</td>
<td>0</td>
<td>15</td>
<td>135</td>
<td>237</td>
<td>45</td>
<td>79</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.0809</td>
<td>0</td>
<td>21</td>
<td>315</td>
<td>612</td>
<td>105</td>
<td>204</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0802</td>
<td>0</td>
<td>20</td>
<td>300</td>
<td>588</td>
<td>100</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1098</td>
<td>0</td>
<td>21</td>
<td>315</td>
<td>612</td>
<td>105</td>
<td>204</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0920</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>573</td>
<td>95</td>
<td>191</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0633</td>
<td>1</td>
<td>24</td>
<td>225</td>
<td>444</td>
<td>75</td>
<td>148</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.1722</td>
<td>2</td>
<td>39</td>
<td>615</td>
<td>1443</td>
<td>205</td>
<td>481</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1614</td>
<td>1</td>
<td>33</td>
<td>510</td>
<td>1221</td>
<td>170</td>
<td>407</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2167</td>
<td>2</td>
<td>39</td>
<td>615</td>
<td>1458</td>
<td>205</td>
<td>486</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1694</td>
<td>1</td>
<td>31</td>
<td>480</td>
<td>1182</td>
<td>160</td>
<td>394</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1407</td>
<td>2</td>
<td>47</td>
<td>441</td>
<td>1017</td>
<td>147</td>
<td>339</td>
</tr>
</tbody>
</table>
### 6.3. Performance of certain methods

Table 6.7: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST C5 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#I</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3841</td>
<td>0</td>
<td>18</td>
<td>309</td>
<td>939</td>
<td>103</td>
<td>313</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1340</td>
<td>0</td>
<td>18</td>
<td>315</td>
<td>993</td>
<td>105</td>
<td>331</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.3646</td>
<td>0</td>
<td>19</td>
<td>333</td>
<td>1017</td>
<td>111</td>
<td>339</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1576</td>
<td>0</td>
<td>20</td>
<td>363</td>
<td>1098</td>
<td>121</td>
<td>366</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2399</td>
<td>0</td>
<td>19</td>
<td>222</td>
<td>717</td>
<td>74</td>
<td>239</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.1361</td>
<td>0</td>
<td>23</td>
<td>387</td>
<td>1392</td>
<td>129</td>
<td>464</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1522</td>
<td>0</td>
<td>24</td>
<td>396</td>
<td>1440</td>
<td>132</td>
<td>480</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1767</td>
<td>0</td>
<td>25</td>
<td>417</td>
<td>1422</td>
<td>139</td>
<td>474</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1630</td>
<td>0</td>
<td>24</td>
<td>426</td>
<td>1431</td>
<td>142</td>
<td>477</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1154</td>
<td>0</td>
<td>26</td>
<td>288</td>
<td>1023</td>
<td>96</td>
<td>341</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.2243</td>
<td>1</td>
<td>35</td>
<td>588</td>
<td>2289</td>
<td>196</td>
<td>763</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.2106</td>
<td>0</td>
<td>36</td>
<td>573</td>
<td>2292</td>
<td>191</td>
<td>764</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2678</td>
<td>1</td>
<td>34</td>
<td>606</td>
<td>2337</td>
<td>202</td>
<td>779</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.2343</td>
<td>0</td>
<td>32</td>
<td>537</td>
<td>2118</td>
<td>179</td>
<td>706</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1616</td>
<td>1</td>
<td>41</td>
<td>429</td>
<td>1635</td>
<td>143</td>
<td>545</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.3245</td>
<td>2</td>
<td>55</td>
<td>864</td>
<td>3585</td>
<td>288</td>
<td>1195</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.3678</td>
<td>2</td>
<td>59</td>
<td>921</td>
<td>3945</td>
<td>307</td>
<td>1315</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.3960</td>
<td>1</td>
<td>54</td>
<td>906</td>
<td>3627</td>
<td>302</td>
<td>1209</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.4135</td>
<td>0</td>
<td>57</td>
<td>945</td>
<td>3888</td>
<td>315</td>
<td>1296</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2619</td>
<td>2</td>
<td>72</td>
<td>672</td>
<td>2766</td>
<td>224</td>
<td>922</td>
</tr>
</tbody>
</table>
Table 6.8: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST D2 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3484</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>477</td>
<td>80</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0908</td>
<td>0</td>
<td>17</td>
<td>255</td>
<td>504</td>
<td>85</td>
<td>168</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.3004</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>450</td>
<td>80</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0932</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>450</td>
<td>80</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2029</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>291</td>
<td>48</td>
<td>97</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0935</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>735</td>
<td>95</td>
<td>245</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0820</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>738</td>
<td>95</td>
<td>246</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1015</td>
<td>0</td>
<td>19</td>
<td>285</td>
<td>732</td>
<td>95</td>
<td>244</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1052</td>
<td>0</td>
<td>19</td>
<td>288</td>
<td>753</td>
<td>96</td>
<td>251</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0591</td>
<td>1</td>
<td>20</td>
<td>195</td>
<td>534</td>
<td>65</td>
<td>178</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.1198</td>
<td>0</td>
<td>25</td>
<td>375</td>
<td>1164</td>
<td>125</td>
<td>388</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1083</td>
<td>0</td>
<td>24</td>
<td>360</td>
<td>1146</td>
<td>120</td>
<td>382</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1415</td>
<td>0</td>
<td>25</td>
<td>375</td>
<td>1197</td>
<td>125</td>
<td>399</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1186</td>
<td>0</td>
<td>22</td>
<td>330</td>
<td>1059</td>
<td>110</td>
<td>353</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0763</td>
<td>1</td>
<td>24</td>
<td>225</td>
<td>741</td>
<td>75</td>
<td>247</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.1992</td>
<td>2</td>
<td>38</td>
<td>600</td>
<td>2133</td>
<td>200</td>
<td>711</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1865</td>
<td>0</td>
<td>38</td>
<td>570</td>
<td>2076</td>
<td>190</td>
<td>692</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2338</td>
<td>1</td>
<td>39</td>
<td>600</td>
<td>2193</td>
<td>200</td>
<td>731</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.2012</td>
<td>1</td>
<td>30</td>
<td>465</td>
<td>1734</td>
<td>155</td>
<td>578</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1175</td>
<td>1</td>
<td>37</td>
<td>342</td>
<td>1218</td>
<td>114</td>
<td>406</td>
</tr>
</tbody>
</table>
### 6.3. Performance of certain methods

Table 6.9: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST D5 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3409</td>
<td>0</td>
<td>16</td>
<td>252</td>
<td>414</td>
<td>84</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0754</td>
<td>0</td>
<td>16</td>
<td>252</td>
<td>414</td>
<td>84</td>
<td>138</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2798</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>261</td>
<td>70</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0729</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>258</td>
<td>70</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1871</td>
<td>0</td>
<td>14</td>
<td>126</td>
<td>174</td>
<td>42</td>
<td>58</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0784</td>
<td>0</td>
<td>18</td>
<td>291</td>
<td>570</td>
<td>97</td>
<td>190</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0756</td>
<td>0</td>
<td>18</td>
<td>297</td>
<td>594</td>
<td>99</td>
<td>198</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1037</td>
<td>0</td>
<td>18</td>
<td>294</td>
<td>585</td>
<td>98</td>
<td>195</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0955</td>
<td>0</td>
<td>19</td>
<td>312</td>
<td>600</td>
<td>104</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0439</td>
<td>0</td>
<td>17</td>
<td>159</td>
<td>321</td>
<td>53</td>
<td>107</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.1300</td>
<td>0</td>
<td>27</td>
<td>465</td>
<td>1191</td>
<td>155</td>
<td>397</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1281</td>
<td>0</td>
<td>26</td>
<td>435</td>
<td>1185</td>
<td>145</td>
<td>395</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1544</td>
<td>0</td>
<td>28</td>
<td>489</td>
<td>1236</td>
<td>163</td>
<td>412</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1666</td>
<td>0</td>
<td>28</td>
<td>483</td>
<td>1230</td>
<td>161</td>
<td>410</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0570</td>
<td>0</td>
<td>22</td>
<td>213</td>
<td>591</td>
<td>71</td>
<td>197</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.2242</td>
<td>7</td>
<td>38</td>
<td>705</td>
<td>2232</td>
<td>235</td>
<td>744</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.2351</td>
<td>8</td>
<td>38</td>
<td>711</td>
<td>2106</td>
<td>237</td>
<td>702</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2853</td>
<td>0</td>
<td>36</td>
<td>579</td>
<td>2052</td>
<td>193</td>
<td>684</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.2588</td>
<td>7</td>
<td>36</td>
<td>663</td>
<td>2169</td>
<td>221</td>
<td>723</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1428</td>
<td>9</td>
<td>43</td>
<td>480</td>
<td>1248</td>
<td>160</td>
<td>416</td>
</tr>
</tbody>
</table>
Table 6.10: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST D6 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3388</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>369</td>
<td>80</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0762</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>369</td>
<td>80</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.3017</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>285</td>
<td>80</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0831</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>285</td>
<td>80</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1881</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>183</td>
<td>48</td>
<td>61</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0976</td>
<td>0</td>
<td>22</td>
<td>336</td>
<td>636</td>
<td>112</td>
<td>212</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0884</td>
<td>0</td>
<td>23</td>
<td>351</td>
<td>660</td>
<td>117</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1242</td>
<td>0</td>
<td>23</td>
<td>351</td>
<td>576</td>
<td>117</td>
<td>192</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0948</td>
<td>0</td>
<td>23</td>
<td>351</td>
<td>582</td>
<td>117</td>
<td>194</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0675</td>
<td>0</td>
<td>25</td>
<td>231</td>
<td>414</td>
<td>77</td>
<td>138</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.3633</td>
<td>0</td>
<td>84</td>
<td>1344</td>
<td>2730</td>
<td>448</td>
<td>910</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.3467</td>
<td>0</td>
<td>83</td>
<td>1317</td>
<td>2670</td>
<td>439</td>
<td>890</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.4718</td>
<td>0</td>
<td>90</td>
<td>1458</td>
<td>2904</td>
<td>486</td>
<td>968</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.4855</td>
<td>0</td>
<td>91</td>
<td>1467</td>
<td>2916</td>
<td>489</td>
<td>972</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2488</td>
<td>0</td>
<td>92</td>
<td>858</td>
<td>1725</td>
<td>286</td>
<td>575</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>1.7861</td>
<td>0</td>
<td>436</td>
<td>6603</td>
<td>13239</td>
<td>2201</td>
<td>4413</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>1.8166</td>
<td>0</td>
<td>450</td>
<td>6807</td>
<td>13644</td>
<td>2269</td>
<td>4548</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>2.2628</td>
<td>0</td>
<td>446</td>
<td>6789</td>
<td>13629</td>
<td>2263</td>
<td>4543</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>2.2262</td>
<td>0</td>
<td>457</td>
<td>6936</td>
<td>13914</td>
<td>2312</td>
<td>4638</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>1.4555</td>
<td>0</td>
<td>585</td>
<td>5283</td>
<td>10584</td>
<td>1761</td>
<td>3528</td>
</tr>
</tbody>
</table>
Table 6.11: Numerical results of the 3-stage Gauss the and 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST E1 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.305</td>
<td>0</td>
<td>8</td>
<td>120</td>
<td>138</td>
<td>40</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0487</td>
<td>0</td>
<td>8</td>
<td>120</td>
<td>138</td>
<td>40</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2254</td>
<td>0</td>
<td>8</td>
<td>120</td>
<td>120</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0536</td>
<td>0</td>
<td>8</td>
<td>120</td>
<td>120</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1435</td>
<td>0</td>
<td>8</td>
<td>72</td>
<td>72</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0411</td>
<td>0</td>
<td>9</td>
<td>135</td>
<td>165</td>
<td>45</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0289</td>
<td>0</td>
<td>9</td>
<td>135</td>
<td>165</td>
<td>45</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0752</td>
<td>0</td>
<td>9</td>
<td>135</td>
<td>159</td>
<td>45</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0350</td>
<td>0</td>
<td>9</td>
<td>135</td>
<td>159</td>
<td>45</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0575</td>
<td>0</td>
<td>9</td>
<td>81</td>
<td>105</td>
<td>27</td>
<td>35</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.0420</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>237</td>
<td>55</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0424</td>
<td>0</td>
<td>10</td>
<td>150</td>
<td>219</td>
<td>50</td>
<td>73</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0438</td>
<td>0</td>
<td>11</td>
<td>165</td>
<td>207</td>
<td>55</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0391</td>
<td>0</td>
<td>10</td>
<td>150</td>
<td>195</td>
<td>50</td>
<td>65</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0222</td>
<td>0</td>
<td>11</td>
<td>99</td>
<td>144</td>
<td>33</td>
<td>48</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.0696</td>
<td>1</td>
<td>16</td>
<td>255</td>
<td>432</td>
<td>85</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0477</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>348</td>
<td>70</td>
<td>116</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0782</td>
<td>1</td>
<td>16</td>
<td>255</td>
<td>393</td>
<td>85</td>
<td>131</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0664</td>
<td>0</td>
<td>14</td>
<td>210</td>
<td>315</td>
<td>70</td>
<td>105</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0490</td>
<td>1</td>
<td>18</td>
<td>171</td>
<td>285</td>
<td>57</td>
<td>95</td>
</tr>
</tbody>
</table>
Table 6.12: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST E4 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.2684</td>
<td>0</td>
<td>22</td>
<td>360</td>
<td>804</td>
<td>120</td>
<td>268</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1066</td>
<td>0</td>
<td>20</td>
<td>333</td>
<td>783</td>
<td>111</td>
<td>261</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2757</td>
<td>0</td>
<td>23</td>
<td>387</td>
<td>843</td>
<td>129</td>
<td>281</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1215</td>
<td>1</td>
<td>21</td>
<td>348</td>
<td>726</td>
<td>116</td>
<td>242</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1594</td>
<td>3</td>
<td>20</td>
<td>222</td>
<td>561</td>
<td>74</td>
<td>187</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.1398</td>
<td>0</td>
<td>27</td>
<td>426</td>
<td>1254</td>
<td>142</td>
<td>418</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1367</td>
<td>0</td>
<td>26</td>
<td>435</td>
<td>1296</td>
<td>145</td>
<td>432</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1662</td>
<td>0</td>
<td>27</td>
<td>426</td>
<td>1278</td>
<td>142</td>
<td>426</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.1672</td>
<td>3</td>
<td>26</td>
<td>453</td>
<td>1323</td>
<td>151</td>
<td>441</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0852</td>
<td>0</td>
<td>26</td>
<td>252</td>
<td>783</td>
<td>84</td>
<td>261</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.2287</td>
<td>5</td>
<td>40</td>
<td>675</td>
<td>2292</td>
<td>225</td>
<td>764</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.1782</td>
<td>0</td>
<td>33</td>
<td>519</td>
<td>1758</td>
<td>173</td>
<td>586</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2875</td>
<td>6</td>
<td>42</td>
<td>720</td>
<td>2457</td>
<td>240</td>
<td>819</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.2281</td>
<td>0</td>
<td>36</td>
<td>576</td>
<td>1932</td>
<td>192</td>
<td>644</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1451</td>
<td>5</td>
<td>39</td>
<td>408</td>
<td>1482</td>
<td>136</td>
<td>494</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.3935</td>
<td>6</td>
<td>72</td>
<td>1170</td>
<td>3969</td>
<td>390</td>
<td>1323</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.2906</td>
<td>4</td>
<td>48</td>
<td>792</td>
<td>3021</td>
<td>264</td>
<td>1007</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.4892</td>
<td>6</td>
<td>75</td>
<td>1215</td>
<td>4179</td>
<td>405</td>
<td>1393</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.3685</td>
<td>6</td>
<td>51</td>
<td>867</td>
<td>3312</td>
<td>289</td>
<td>1104</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.2350</td>
<td>7</td>
<td>69</td>
<td>687</td>
<td>2364</td>
<td>229</td>
<td>788</td>
</tr>
</tbody>
</table>
Table 6.13: Numerical results of the 3-stage Gauss and the 4-stage Lobatto IIIA methods with symmetrization and extrapolation applied to the DETEST E5 problem.

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Methods</th>
<th>Time steps</th>
<th>Rejected steps</th>
<th>Accepted steps</th>
<th>#J</th>
<th>#f</th>
<th>#LU</th>
<th>#BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-2}$</td>
<td>G3 ASAX1</td>
<td>0.3240</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>240</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0677</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>240</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.2660</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>240</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0792</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>240</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.1616</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>144</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>$10^{-4}$</td>
<td>G3 ASAX1</td>
<td>0.0928</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>291</td>
<td>80</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0586</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>291</td>
<td>80</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0770</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>240</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0740</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>240</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0377</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>144</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>$10^{-6}$</td>
<td>G3 ASAX1</td>
<td>0.0663</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>321</td>
<td>80</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0551</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>321</td>
<td>80</td>
<td>107</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.1063</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>291</td>
<td>80</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0719</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>291</td>
<td>80</td>
<td>97</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0617</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>189</td>
<td>48</td>
<td>63</td>
</tr>
<tr>
<td>$10^{-8}$</td>
<td>G3 ASAX1</td>
<td>0.0593</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>363</td>
<td>80</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>G3 PSAX</td>
<td>0.0628</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>363</td>
<td>80</td>
<td>121</td>
</tr>
<tr>
<td></td>
<td>L4 ASAX1</td>
<td>0.0754</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>336</td>
<td>80</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>L4 PSAX</td>
<td>0.0648</td>
<td>0</td>
<td>16</td>
<td>240</td>
<td>330</td>
<td>80</td>
<td>110</td>
</tr>
<tr>
<td></td>
<td>R3 AX</td>
<td>0.0503</td>
<td>0</td>
<td>16</td>
<td>144</td>
<td>210</td>
<td>48</td>
<td>70</td>
</tr>
</tbody>
</table>
Conclusions and Future Work

7.1 Conclusions

The main objectives of this thesis are to implement the symmetrized Gauss and Lobatto IIIA methods in a constant and variable stepsize setting and to study the performance of active and passive symmetrization on various problems.

7.1.1 Constant stepsize setting

In the constant stepsize setting, we arrived at the following conclusions for three different classes of problems:

1. Nonstiff problems
   - Active and passive extrapolation of the base method without symmetrization is as efficient as the symmetrized method in either mode of extrapolation.
For the two dimensional nonlinear problems, passive and active extrapolation of the 2-stage Gauss method without symmetrization is more efficient than the 3-stage Lobatto IIIA method without symmetrization in either mode of extrapolation.

For higher dimensional nonlinear problems, passive and active extrapolation of the 3-stage Lobatto IIIA method without symmetrization is more efficient than the 2-stage Gauss method without symmetrization in either mode of extrapolation.

For higher dimensional linear problems, passive symmetrization with passive extrapolation of the Gauss methods is slightly more efficient than passive symmetrization with passive extrapolation of the Lobatto IIIA methods.

For higher dimensional nonlinear problems, passive symmetrization with passive extrapolation of the 3-stage Lobatto IIIA method is more efficient than passive symmetrization of the 2-stage Gauss method with passive extrapolation. On the other hand, passive symmetrization with passive extrapolation of the 3-stage Gauss method is more efficient than passive symmetrization of the 4-stage Lobatto IIIA method with passive extrapolation.

In all cases, passive symmetrization with passive extrapolation is more efficient than active and passive symmetrization with active extrapolation.

2. Mildly stiff problems

Passive symmetrization is more efficient than active symmetrization for linear and nonlinear problems.

Passive symmetrization with passive and active modes of extrapolation is more efficient than active symmetrization in either mode of extrapolation for linear and nonlinear problems.

For scalar linear problems, the symmetrized Lobatto IIIA methods in either mode of extrapolation are more efficient than the symmetrized Gauss methods with extrapolation.

For matrix linear problems, the symmetrized Gauss methods in either mode of extrapolation are more efficient than the symmetrized Lobatto IIIA methods with extrapolation.
3. Stiff problems

- For scalar linear and matrix problems, passive symmetrization with passive extrapolation of the Gauss and Lobatto IIIA methods is more efficient than active symmetrization in either mode of extrapolation.

- For two dimensional nonlinear problems, passive symmetrization with passive extrapolation of the 2-stage Gauss method is more efficient than the 3-stage Lobatto IIIA method with passive symmetrization and extrapolation. On the other hand, passive symmetrization with passive extrapolation of the 4-stage Lobatto IIIA method is more efficient than the 3-stage Gauss method with passive symmetrization and extrapolation.

Overall, for constant stepsizes, we conclude that symmetrization is more efficient when performed with extrapolation for mildly stiff and stiff linear and nonlinear problems. The most efficient way of applying symmetrization with extrapolation is passive symmetrization with passive extrapolation.

7.1.2 Variable stepsize setting

From the numerical results using variable stepsize, we conclude that although the 3-stage Radau IIA method is slightly more efficient than the symmetrized methods, the 3-stage symmetrized Gauss and the 4-stage symmetrized Lobatto IIIA methods with active extrapolation can be competitive with the 3-stage Radau IIA method with and without extrapolation.

In the variable stepsize setting, we draw the following conclusions:

- For linear problems, active extrapolation of the symmetrized 3-stage Gauss method is more efficient than that of the symmetrized 4-stage Lobatto IIIA method.

- For nonlinear problems, active extrapolation of the symmetrized 4-stage Lobatto IIIA method is more efficient than that of the symmetrized 3-stage Gauss method.

- In all cases, active extrapolation combined with passive symmetrization is more efficient than with active symmetrization.

Overall, the numerical results show that the symmetrization of the Gauss and Lobatto IIIA methods can be beneficial in solving the standard stiff problems. In the variable
steps size setting, our code performed well on the STIFF DETEST problems. We have produced a simple algorithm for the implementation of symmetrizer in this setting. We expect that fine tuning in the implementation of this algorithm can make the code more robust.

7.2 Future work

In this thesis, we have shown that the 3-stage symmetrized Gauss and the 4-stage symmetrized Lobatto IIIA methods with extrapolation can be as efficient as the 3-stage Radau IIA method with and without extrapolation. The numerical results indicate that further investigations are needed to classify the effective use of symmetrization/extrapolation on ODEs for higher order Gauss and Lobatto IIIA methods and to extend the study to PDEs. Furthermore, it is interesting to determine the effects of using PI stepsize control (see [53] on page 31) on the performance of symmetrizers.

Additionally, the one step symmetrization can be extended to two-step symmetrization. A two step symmetrizer \( \hat{\mathcal{R}} \) satisfying \( \hat{\mathcal{R}} \circ (-\hat{\mathcal{R}}^{-1}) = \hat{\mathcal{R}}^4 \) has Butcher tableau given by

\[
\begin{align*}
    c & | & A & 0 & 0 & 0 \\
    e + c & | & eb^T & A & 0 & 0 \\
    2e + c & | & eb^T & eb^T & A & 0 \\
    3e + c & | & eb^T & eb^T & eb^T & A \\
    b^T - u^T P & & b^T - u^T P & & u^T & v^T
\end{align*}
\]

where \( u \) and \( v \) are the weight vectors that have \( s \) components which are also determined by the damping and order conditions. The question of interest is whether such two-step symmetrizers can be more efficient than one-step symmetrizers.

One drawback of symmetrization is the need to compute beyond the current step. If computation cannot proceed beyond the endpoint, then the symmetrized value has to be estimated by some other means (see for example, Chan and Gorgey [25]).
In Appendix A, we present additional numerical results. Figure A.1 shows the accuracy diagrams for the linear matrix problems using different stiffness values. The summary of the order behaviour is given in Table 4.1. On the other hand, Figure A.2 – Figure A.5 show the efficiency diagrams of the Gauss and the Lobatto IIIA methods with symmetrization applied to the linear and nonlinear test problems using constant stepsize. Here, the numerical results are given for symmetrization without extrapolation. The numerical results for symmetrization with extrapolation are given in Figure 6.1 – Figure 6.4. As it is shown in Figure A.2, the most efficient method is the 3-stage Radau IIA method. However, we have shown in Chapter 6 (see Figure 6.1) that symmetrization gives greater accuracy when applied with extrapolation. Similar results are seen in Figure A.3.
Appendix A. Additional Numerical Results

Figure A.1: Accuracy diagrams of the Gauss and Lobatto IIIA methods with symmetrization for different $\lambda$ values applied to the linear matrix problem at $x_n = 5$.  

164
Figure A.2: Efficiency diagrams of the 2-stage Gauss and the 3-stage Lobatto IIA methods with symmetrization applied to the linear test problems B.1.1 using constant stepsize.

Figure A.3: Efficiency diagrams of the 3-stage Gauss and the 4-stage Lobatto IIA methods with symmetrization applied to the linear test problems B.1.1 using constant stepsize.
Appendix A. Additional Numerical Results

Figure A.4: Efficiency diagrams of the 2-stage Gauss and 3-stage Lobatto IIIA methods with symmetrization applied to the nonlinear problems using constant stepsize.

Figure A.5: Efficiency diagrams of the 3-stage Gauss and 4-stage Lobatto IIIA methods with symmetrization applied to the nonlinear problems using constant stepsize.
Figure A.6: Efficiency diagrams of the Gauss and Lobatto IIIA methods with symmetrization and extrapolation applied to the test problem LP15 for $\alpha = 10^6$ using constant stepsize.
In Appendix B, we give the list of test problems using constant and variable stepsizes. In the constant stepsize setting, the test problems are taken from the literature and are mainly for solving nonstiff, mildly and stiff problems. On the other hand, the test problems for variable stepsize is taken from the well known stiff DETEST problem set given in [38]. The problems are classified into five classes that represent linear and nonlinear categories with corresponding real and non-real eigenvalues. The differential equations of all problems, their intervals of integration and their initial conditions and step sizes are listed below.
B.1 Test problems using constant stepsize

B.1.1 Linear problems

LP1. C. F. Curtiss and J. O. Hirschfelder, 1952 [29]

\[ y' = -50(y - \cos(x)), \quad (B.1) \]

with \(y(0) = 1\) and \(x_n = 15\).

**Exact solution:** \(y(x) = \frac{2500}{2501} \cos(x) + \frac{50}{2501} \sin(x) + \frac{1}{2501} e^{-50x}, \quad x \geq 0.\)


\[ y' = -200(y - F(x)) + F'(x), \quad F(x) = 10 - (10 + x) e^{-x}, \quad (B.2) \]

with \(y(0) = 10\) and \(x_n = 5\).

**Exact solution:** \(y(x) = 10 - (10 + x) e^{-x} + 10 e^{-200x}.\)

Note: (B.2) is a single ODE. with a solution containing a rapidly decaying component and a slowly decaying component. The eigenvalue is -200, and the solution is desired from \(x = 0\) to \(x = 15\).


\[ y' = \lambda y + e^{-x}, \quad (B.3) \]

with \(y(0) = -\frac{1}{1+\lambda}, \lambda = -10^6\) and \(x_n = 5\).

**Exact solution:** \(y(x) = -\frac{1}{1+\lambda} e^{-x}.\)

LP4. R. Holsapple, R. Iyer and D. Doman, 2007 [55]

\[ y' = -1000y + \sin(x), \quad (B.4) \]

with \(y(0) = -\frac{1}{1000000}.\)

**Exact solution:** \(y(x) = \frac{1000 \sin(x) - \cos(x)}{1000001}.\)
B.1. Test problems using constant stepsize

LP5. A logistic curve (DETest [56])

\[ y' = \frac{1}{4} \left( 1 - \frac{y}{20} \right), \quad (B.5) \]

with \( y(0) = 1 \).

**Exact solution:** \( \frac{20}{1 + 19Ce^{-x/4}}, C = 1 \).


\[ \begin{align*}
y'_1 &= -1000y_1, \\
y'_2 &= 0.909y_1 - y_2, \quad (B.6)\end{align*} \]

with \( y_1(0) = 1 \) and \( y_2(0) = 0.999 \).

**Exact solution:**

\[ \begin{align*}
y_1(x) &= e^{-1000x}, \\
y_2(x) &= \frac{0.909}{999} e^{-1000x} + \frac{998.91}{999} e^{-x}.\end{align*} \]

LP7. Lindberg, 1972 [62]

\[ \begin{align*}
y'_1 &= -11y_1 + 100y_2 + 10, \\
y'_2 &= 10y_1 - 101y_2 + 1, \quad (B.7)\end{align*} \]

with \( y_1(0) = 1 \) and \( y_2(0) = 1 \).

**Exact solution:**

\[ \begin{align*}
y_1(x) &= 10 - \frac{90}{11} e^{-x} - \frac{9}{11} e^{-111x}, \\
y_2(x) &= 1 - \frac{9}{11} e^{-x} - \frac{9}{11} e^{-111x}.\end{align*} \]


\[ \begin{align*}
y'_1 &= -y_1 + 95y_2, \\
y'_2 &= -y_1 - 97y_2, \quad (B.7)\end{align*} \]

with \( y_1(0) = 1 \) and \( y_2(0) = 1 \).
Exact solution:

\[
y_1(x) = \frac{1}{47} \left( 95e^{-2x} - 48e^{-96x} \right),
\]
\[
y_2(x) = \frac{1}{47} \left( 48e^{-96x} - e^{-2x} \right).
\]


\[
y_1' = -100y_1 + 9.901y_2,
\]
\[
y_2' = 0.1y_1 - y_2,
\]

with \(y_1(0) = 1\) and \(y_2(0) = 10\).

Exact solution:

\[
y_1(x) = e^{-0.99x},
\]
\[
y_2(x) = 10e^{-0.99x}.
\]

Note: The eigenvalues of the Jacobian matrix are -100.01 and -0.99.

LP10. Oscillatory problem (S. O. Fatunla, 1980 [40])

\[
y_1' = -10^{-5}y_1 + 100y_2,
\]
\[
y_2' = -100y_1 - 10^{-5}y_2, \quad 0 \leq x \leq 10\pi
\]

with \(y_1(0) = 0\) and \(y_2(0) = 1\).

Exact solution:

\[
y_1(x) = e^{-10^{-5}} \sin(100x),
\]
\[
y_2(x) = e^{-10^{-5}} \cos(100x).
\]


\[
y_1' = \lambda y_1 + y_2,
\]
\[
y_2' = -y_2,
\]

with \(y_1(0) = -\frac{1}{\lambda+1}\) and \(y_2(0) = 1\).
B.1. Test problems using constant stepsize

Exact solution:

\[ y_1(x) = -\frac{e^{-x}}{\lambda + 1}, \]
\[ y_2(x) = e^{-x}. \]

LP12. C. F. Curtiss and J. O. Hirschfelder, 1952 [29]

\[ y_1' = y_2, \]
\[ y_2' = -y_1, \]
\[ y_3' = -Ly_1 + y_2 + Ly_3, \] \hspace{1cm} (B.11)

with \( y_1(0) = 0, y_2(0) = 1 \) and \( y_3(0) = \epsilon \). When \( \epsilon = 0 \),

Exact solution:

\[ y_1(x) = \sin(x), \]
\[ y_2(x) = \cos(x), \]
\[ y_3(x) = \sin(x) \]

and if \( \epsilon \neq 0 \), \( y_3(x) = \sin(x) + \epsilon e^{Lx} \).


\[ y_1' = -500.5y_1 + 499.5y_2, \]
\[ y_2' = 499.5y_1 - 500.5y_2, \] \hspace{1cm} (B.12)

with \( y_1(0) = 2 \) and \( y_2(0) = 1 \).

Exact solution:

\[ y_1(x) = 1.5e^{-x} + 0.5e^{-1000x}, \]
\[ y_2(x) = 1.5e^{-x} - 0.5e^{-1000x}. \]
Appendix B. Test problems


\[ \begin{align*}
  y_1' &= -0.1y_1 - 49.9y_2, \\
  y_2' &= -50y_2, \\
  y_3' &= 70y_2 - 120y_3, \quad 0 \leq x \leq 15 \\
\end{align*} \]  
(B.13)

with \( y_1(0) = 2, y_2(0) = 1 \) and \( y_3(0) = 2 \).

Exact solution:

\[ \begin{align*}
  y_1(x) &= e^{-0.1x} + e^{-50x}, \\
  y_2(x) &= e^{-50x}, \\
  y_3(x) &= e^{-50x} + e^{-120x} \\
\end{align*} \]

Note: (B.13) has eigenvalues -120, -50, and -0.1. This system is interesting because it contains two stiff eigenvalues, so that three different characteristic times appear.


\[ \begin{align*}
  y_1' &= -10y_1 + \alpha y_2, \\
  y_2' &= -\alpha y_1 - 10y_2, \\
  y_3' &= -4y_3, \\
  y_4' &= -y_4, \\
  y_5' &= -0.5y_5, \\
  y_6' &= -0.1y_6 \\
\end{align*} \]  
(B.14)

with \( y_1(0) = 1, y_2(0) = 1, y_3(0) = 1, y_4(0) = 1, y_5(0) = 1 \) and \( y_6(0) = 1 \).

Exact solution:

\[ \begin{align*}
  y_1(x) &= e^{-10x} \sin(\alpha x), \\
  y_2(x) &= e^{-10x} \cos(\alpha x), \\
  y_i(x) &= e^{-\beta_ix}, \quad i = 3, 4, 5, 6 \\
\end{align*} \]

where \( \beta_3 = -4, \beta_4 = -1, \beta_5 = -0.5 \) and \( \beta_6 = -0.1 \).
B.1.2 Nonlinear problems


\[ \begin{align*}
  y_1' &= \lambda y_1 + y_2^2, \\
  y_2' &= -y_2,
\end{align*} \tag{B.15} \]

with \( y_1(0) = -\frac{1}{\lambda+2} \) and \( y_2(0) = 1 \).

Exact solution:

\[ \begin{align*}
  y_1(x) &= -\frac{e^{-2x}}{\lambda+2}, \\
  y_2(x) &= e^{-x}.
\end{align*} \]


\[ \begin{align*}
  y_1' &= y_2 - y_1^2 - (1 + x), \\
  y_2' &= 1 - 20(y_2^2 - (1 + x)^2), \tag{B.16}
\end{align*} \]

with \( y_1(0) = 1 \) and \( y_2(0) = 1 \).

Exact solution:

\[ \begin{align*}
  y_1(x) &= \frac{1}{1 + x}, \\
  y_2(x) &= 1 + x.
\end{align*} \]

NP3. Chemistry reaction, (DETest [56])

\[ \begin{align*}
  y_1' &= y_1, \\
  y_2' &= y_1 - y_2^2, \\
  y_3' &= y_2, \tag{B.17}
\end{align*} \]

with \( y_1(0) = 1, y_2(0) = 0 \) and \( y_3(0) = 0 \).
Appendix B. Test problems

B.2 Test problems for variable stepsize

B.2.1 Problem Class A: Linear with real eigenvalues

A1.

\[ \begin{align*}
  y_1' &= -0.5y_1, \quad y_1(0) = 1, \\
  y_2' &= -y_2, \quad y_2(0) = 1, \\
  y_3' &= -100y_3, \quad y_3(0) = 1, \\
  y_4' &= -90y_4, \quad y_4(0) = 1,
\end{align*} \]

at \( x_n = 20 \) using \( h = 10^{-2} \).

A2.

\[ \begin{align*}
  y_1' &= -1800y_1 + 900y_2, \quad y_1(0) = 0, \\
  y_j' &= y_{j-1} - 2y_j + y_{j+1}, \quad y_j(0) = 0, \quad j = 2, 3, \ldots, 8, \\
  y_9' &= 1000y_8 - 2000y_9 + 1000, \quad y_9(0) = 0,
\end{align*} \]

at \( x_n = 120 \) using \( h = 5 \times 10^{-4} \).

A3.

\[ \begin{align*}
  y_1' &= -10^4y_1 - 100y_2 - 10y_3 + y_4, \quad y_1(0) = 1, \\
  y_2' &= -10^3y_2 + 10y_3 - 10y_4, \quad y_2(0) = 1, \\
  y_3' &= -y_3 + 10y_4, \quad y_3(0) = 1, \\
  y_4' &= -0.1y_4, \quad y_4(0) = 1,
\end{align*} \]

at \( x_n = 20 \) with \( h = 10^{-5} \).

A4.

\[ \begin{align*}
  y_j' &= -j^5y_j, \quad y_j(0) = 1, \quad j = 1, 2, \ldots, 10,
\end{align*} \]

at \( x_n = 1 \) with \( h = 10^{-5} \).
B.2.2 Problem Class B: Linear with complex eigenvalues

B1.

\[ \begin{align*}
  y_1' &= -y_1 + y_2, & y_1(0) &= 1, \\
  y_2' &= -10^2y_1 - y_2, & y_2(0) &= 0, \\
  y_3' &= -10^3y_3 + y_4, & y_3(0) &= 1, \\
  y_4' &= -10^4y_3 - 10^2y_4, & y_4(0) &= 0,
\end{align*} \]

at \( x_n = 20 \) with \( h = 7 \times 10^{-3} \).
(eigenvalues: \(-1 \pm 10i, -100 \pm 100i\)).

B2.

\[ \begin{align*}
  y_1' &= -10y_1 + \mu y_2, & y_1(0) &= 0, \\
  y_2' &= -\mu y_1 - y_2, & y_2(0) &= 0, \\
  y_3' &= -4y_3, & y_3(0) &= 0, \\
  y_4' &= -y_4, & y_4(0) &= 0, \\
  y_5' &= -0.5y_5, & y_5(0) &= 0, \\
  y_6' &= -0.1y_6, & y_6(0) &= 0,
\end{align*} \]

at \( x_n = 20 \) with \( h = 10^{-2} \) and \( \mu = 3 \).
(eigenvalues: \(-0.1, -0.5, -1, -4, -10 \pm i\mu\)).

B3. Same as B2 with \( \mu = 8 \).

B4. Same as B2 with \( \mu = 25 \).

B5. Same as B2 with \( \mu = 100 \).
B.2.3 Problem Class C: Nonlinear coupling

C1.

\[
\begin{align*}
y_1' &= -y_1 + y_2^2 + y_3^2 + y_4^2, & y_1(0) &= 1, \\
y_2' &= -10y_2 + 10y_3^2 + 10y_4^2, & y_2(0) &= 1, \\
y_3' &= -40y_3 + 40y_4^2, & y_3(0) &= 1, \\
y_4' &= -100y_4 + 2, & y_4(0) &= 1,
\end{align*}
\]

at \( x_n = 20 \) with \( h = 10^{-2} \).

(coupling from transient components to smooth components).

C2.

\[
\begin{align*}
y_1' &= -y_1 + 2, & y_1(0) &= 1, \\
y_2' &= -10y_2 + \nu y_1^2, & y_2(0) &= 1, \\
y_3' &= -40y_3 + 4\nu y_1^2 + 4\nu y_2^2, & y_3(0) &= 1, \\
y_4' &= -100y_4 + 10\nu y_1^2 + 10\nu y_2^2 + 10\nu y_3^2, & y_4(0) &= 1,
\end{align*}
\]

at \( x_n = 20 \) with \( h = 10^{-2} \) and \( \nu = 0.1 \).

(coupling from smooth components to transient components).

C3. Same as C2 with \( \nu = 1 \).

C4. Same as C2 with \( \nu = 10 \).

C5. Same as C2 with \( \nu = 20 \).
B.2.4 Problem Class D: Nonlinear with real eigenvalues

D1. Nuclear Reactor Theory: Liniger and Willoughby (1970) [63]

\[
\begin{align*}
y_1' &= 0.2(y_2 - y_1), & y_1(0) &= 0, \\
y_2' &= 10y_1 - (60 - 0.125y_3)y_2 + 0.125y_3, & y_2(0) &= 0, \\
y_3' &= 1, & y_3(0) &= 0,
\end{align*}
\]

at \( x_n = 400 \) with \( h = 1.7 \times 10^{-2} \).
(eigenvalues: \( 0, -0.17 \rightarrow -0.012, -60 \rightarrow -11 \)).

D2. Chemistry: Robertson (1966) [69]

\[
\begin{align*}
y_1' &= -0.04y_1 + 0.01y_2y_3, & y_1(0) &= 1 \\
y_2' &= 400y_1 - 100y_2y_3 - 3000y_2^2, & y_2(0) &= 0, \\
y_3' &= 30y_2^2, & y_3(0) &= 0,
\end{align*}
\]

at \( x_n = 40 \), with \( h = 10^{-5} \).
(eigenvalues: \( 0, 0 \rightarrow -3100, -0.040 \rightarrow -0.40 \rightarrow -0.030 \)).

D3. Chemistry: Bjurel et. al. [8],

\[
\begin{align*}
y_1' &= y_3 - 100y_1y_2, & y_1(0) &= 1, \\
y_2' &= y_3 + 2y_4 - 100y_1y_2 - 2 \times 10^4y_2^2, & y_2(0) &= 1, \\
y_3' &= -y_3 + 100y_1y_2, & y_3(0) &= 0, \\
y_4' &= -y_4 + 10^4y_2^2, & y_4(0) &= 0,
\end{align*}
\]

at \( x_n = 20 \) with \( h = 2.5 \times 10^{-5} \).
(eigenvalues: \( 0, -100 \rightarrow -1.4, -4.0 \times 10^4 \rightarrow -290 \)).


\[
y_j' = -j^5y_j, \quad y_j(0) = 1, \quad j = 1, 2, \ldots, 10,
\]

179
Appendix B. Test problems

at $x_n = 1$ with $h = 10^{-5}$.
(eigenvalues: $0, -9.3 \times 10^{-3} \rightarrow -4.0 \times 10^{-3} \rightarrow -6.3 \times 10^{-3}, -3.5 \times 10^3 \rightarrow -3.8 \times 10^3$).

D5. Reactor kinetics: Liniger and Willoughby (1970) [63]

$$y'_1 = 0.01 - [1 + (y_1 + 1000)(y_1 + 1)](0.01 + y_1 + y_2), \quad y_1(0) = 0,$$
$$y'_2 = 0.01 - (1 + y_2^2)(0.01 + y_1 + y_2), \quad y_2(0) = 0,$$

at $x_n = 100$ with $h = 10^{-4}$.
(eigenvalues: $-0.01 \rightarrow -0.0002 \rightarrow -0.002, -1000 \rightarrow -400$).


$$y'_1 = -y_1 + 10^8 y_3(1 - y_1), \quad y_1(0) = 1,$$
$$y'_2 = -10 y_2 + 3 \times 10^7 y_3(1 - y_2), \quad y_2(0) = 0,$$
$$y'_3 = -y'_1 - y'_2, \quad y_3(0) = 0,$$

at $x_n = 1$ with $h = 3.3 \times 10^{-8}$.
(eigenvalues: $0, -1.0 \rightarrow -8.6, -3.0 \times 10^7 \rightarrow -4.0 \times 10^7$).

B.2.5 Problem Class E: Nonlinear with complex eigenvalues

E1. Control theory: Davison (1971)

$$y'_1 = y_2, \quad y_1(0) = 0,$$
$$y'_2 = y_3, \quad y_2(0) = 0,$$
$$y'_3 = y_4, \quad y_3(0) = 0,$$
$$y'_4 = (y_1^2 - \sin y_1 - 10^8)y_1 + \left(\frac{y_2 y_3}{y_1^2 + 1} - 4 \times 10^6\right)y_2 +$$
$$(1 - 6 \times 10^4)y_3 + (10 \exp(-y_4^2) - 4 \times 10^2)y_4 + 1, y_4(0) = 0,$$
B.2. Test problems for variable stepsize

at \( x_n = 1 \) with \( h = 6.8 \times 10^{-3} \).
(eigenvalues: \(-130 \pm 69i, -64 \pm 22i\)).

E2. Van der Pol’s equation: Davis (1962) [35]

\[
\begin{align*}
    y_1' &= y_2, \quad y_1(0) = 2, \\
    y_2' &= 5(1 - y_1^2)y_2 - y_1, \quad y_2(0) = 0,
\end{align*}
\]

at \( x_n = 1 \) with \( h = 10^{-3} \).
(eigenvalues: \(-0.067 \) and \(-15 \rightarrow 5.7 \) and \(-1.5 \rightarrow 3.6 \) and \(1.4 \rightarrow 2.4 \pm 2.8i \rightarrow -0.052 \pm 8.8i \rightarrow -2.0 \pm 9.5i \rightarrow -5.9 \pm 4.5i \rightarrow -2.0 \) and \(-12 \rightarrow 0.050 \) and \(-15 \rightarrow 1.1 \) and \(-3.4 \)).


\[
\begin{align*}
    y_1' &= -(55 + y_3)y_1 + 65y_2, \quad y_1(0) = 1, \\
    y_2' &= 0.0785(y_1 - y_2), \quad y_2(0) = 1, \\
    y_3' &= 0.1y_1, \quad y_3(0) = 0,
\end{align*}
\]

at \( x_n = 500 \) with \( h = 0.02 \).
(eigenvalues: \(0.0062 \pm 0.01i \rightarrow 0.0014 \pm 0.014i \rightarrow -0.015 \) and \(-4.0 \times 10^{-4}, -55 \rightarrow -81 \)).

E4. Artificial test problem: Krogh (1973) [61]

\[
y' = -Q^T \begin{bmatrix} -10 & -10 & 0 & 0 \\ 10 & -10 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} Qy + G(y), \quad y(0) = \begin{bmatrix} 0 \\ -2 \\ -1 \\ -1 \end{bmatrix},
\]
Appendix B. Test problems

where

\[
Q = \frac{1}{2} \begin{bmatrix}
-1 & 1 & 1 & 1 \\
1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1
\end{bmatrix}
\]

\[Qy + G(y), \quad G(y) = Q^T \begin{bmatrix}
(S_1^2 - S_4^2)/2 \\
S_1S_2 \\
S_3^2 \\
S_4^2
\end{bmatrix}, \quad Qy = \begin{bmatrix}
S_1 \\
S_2 \\
S_3 \\
S_4
\end{bmatrix},
\]

at \(x_n = 1000\) with \(h = 10^{-3}\).

(eigenvalues: \(-2.0 \rightarrow -1.0 \times 10^{-2}, 8.0 \pm 10 i \rightarrow 7.1 \pm 7.9 i \rightarrow 9.0 \pm 0.030 i \rightarrow 12 \pm 13 i \rightarrow 0.19 \pm 29 i \rightarrow -17 \pm 18 i \rightarrow -10 \pm 10 i, -10^3\)).

E5. Chemical pyrolysis: Datta (1967) [34]

\[
y'_1 = -7.89 \times 10^{-10} y_1 - 1.1 \times 10^7 y_1 y_3, \quad y_1(0) = 1.76 \times 10^{-3},
\]

\[
y'_2 = 7.89 \times 10^{-10} y_1 - 1.13 \times 10^9 y_2 y_3, \quad y_2(0) = 0,
\]

\[
y'_3 = -7.89 \times 10^{-10} y_1 - 1.1 \times 10^7 y_1 y_3
\]
\[
+ 1.13 \times 10^3 y_4 - 1.13 \times 10^9 y_2 y_3, \quad y_3(0) = 0
\]

\[
y'_4 = 1.1 \times 10^7 y_1 y_3 - 1.13 \times 10^3 y_4, \quad y_4(0) = 0,
\]

at \(x_n = 1000\) with \(h = 5 \times 10^{-5}\).

(eigenvalues: \(0, -7.5 \times 10^{-10} \pm 9.2 \times 10^{-4} i \rightarrow 2.9 \times 10^{-4} \pm 8.7 \times 10^{-4} i \rightarrow -9.4 \times 10^{-5} \) and \(-0.019, -2.0 \times 10^4\)).
References


References


References


