Computer solution of non-linear integration formula for solving initial value problems

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/25381

Publisher: © Abdul Razak bin Yaakub

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

Please cite the published version.
COMPUTER SOLUTION OF NON-LINEAR INTEGRATION FORMULA FOR SOLVING INITIAL VALUE PROBLEMS

by

ABDUL RAZAK BIN YAAKUB

B.Sc, M.Sc

A Doctoral Thesis
Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of the Loughborough University
Jun, 1996

Supervisor
PROFESSOR D. J. EVANS, Ph.D., D.Sc.
Parallel Algorithms And Architectures Research Centre Department of Computer Studies

© by Abdul Razak Bin Yaakub, 1996
ACKNOWLEDGEMENTS

I would like to express my thanks and gratitude to my Supervisor, Professor D. J. Evans for his invaluable help, guidance, supervision and support throughout my research.

I also would like to thank the British Aerospace, United Kingdom for providing me with financial support during my research for the last three years.

I also would like to thank Universiti Utara Malaysia for giving me study leave and providing my family with financial support.

I also would like to thank the staff in the Department of Computer Studies, Loughborough University for the assistance they gave and all of my friends who have helped me in one way or another and wish to extend my sincere appreciation and best wishes.

Finally, I would like to thank my father and mother for their love and encouragement, my wife Zaini Azmi and our children Zafirah Hanis, Farah Yasmeen and Nabilah Hanim for their constant love, devotion and patience. May Allah reward them all.
ABSTRACT

This thesis is concerned with the numerical solutions of initial value problems with ordinary differential equations and covers the various aspects of single step integration methods. Specifically, its main focus is to study the numerical methods of non-linear integration formula with a variety of means based on the Contraharmonic mean \((C,M)\) (Evans and Yaakub [1995]), the Centroidal mean \((C,M)\) (Yaakub and Evans [1995]) and the Root-Mean-Square \((RMS)\) (Yaakub and Evans [1993]) for solving initial value problems. It includes a study of the applications of the second order \(C,M\) method for parallel implementation of extrapolation methods for ordinary differential equations with the ExDaTa schedule by Bahoshy [1992]. Another important topic presented in this thesis is that a fifth order five-stage explicit Runge-Kutta method or weighted Runge Kutta formula [Evans and Yaakub [1996]] exists which is contrary to Butcher [1987] and the theorem in Lambert ([1991], pp 181).

The thesis is organized as follows. An introduction to initial value problems in ordinary differential equations and parallel computers and software in Chapter 1, the basic preliminaries and fundamental concepts in mathematics, an algebraic manipulation package, e.g., Mathematica and basic parallel processing techniques are discussed in Chapter 2. Following in Chapter 3 is a survey of single step methods to solve ordinary differential equations. In this chapter, several single step methods including the Taylor series method, Runge Kutta method and a linear multistep method for non-stiff and stiff problems are also considered.

Chapter 4 gives a new Runge Kutta formula for solving initial value problems using the Contraharmonic mean \((C,M)\), the Centroidal mean \((C,M)\) and the Root-Mean-Square \((RMS)\). An error and stability analysis for these variety of means and numerical examples are also
presented. Chapter 5 discusses the parallel implementation on the Sequent 8000 parallel computer of the Runge-Kutta contraharmonic mean \((C,M)\) method with extrapolation procedures using explicit data task assignment scheduling (EXDATA) strategies. A new Runge-Kutta RK\((4,4)\) method is introduced and the theory and analysis of its properties are investigated and compared with the more popular RKF\((4,5)\) method, are given in Chapter 6. Chapter 7 presents a new integration method with error control for the solution of a special class of second order ODEs. In Chapter 8, a new weighted Runge-Kutta fifth order method with 5 stages is introduced. By comparison with the currently recommended RK4\((5)\) Merson and RK5\((6)\) Nystrom methods, the new method gives improved results. Chapter 9 proposes a new fifth order Runge-Kutta type method for solving oscillatory problems by the use of trigonometric polynomial interpolation which extends the earlier work of Gautschi [1961]. An analysis of the convergence and stability of the new method is given with comparison with the standard Runge-Kutta methods.

Finally, Chapter 10 summarises and presents conclusions on the topics discussed throughout the thesis.

**Keywords:** ordinary differential equations (ODEs), arithmetic mean, contraharmonic mean, centroidal mean, extrapolation, parallel solution, trigonometric polynomial and Runge-Kutta 4th and 5th order.
# TABLE OF CONTENTS

## CHAPTER 1: INTRODUCTION

1.1 DIFFERENTIAL EQUATIONS ................................................................. 2
1.2 NUMERICAL SOLUTION OF INITIAL VALUE PROBLEMS ...................... 4
1.3 THE SEQUENT BALANCE SYSTEM .................................................... 7
1.4 PARALLEL PROCESSING AND PARALLEL PROGRAMS ......................... 8
1.5 PARALLEL PROGRAMMING LANGUAGE IN C ON THE SEQUENT
   BALANCE SYSTEM ............................................................................. 9

## CHAPTER 2: BASIC PRELIMINARIES, SYMBOLIC COMPUTATION
   AND BASIC IN PARALLEL PROCESSING

2.1 BASIC PRELIMINARIES .................................................................... 11
   2.1.1 Mathematical Means ................................................................. 11
   2.1.2 Difference Equations ................................................................. 14
   2.1.3 Homogeneous Difference Equation ............................................. 15
   2.1.4 Taylor Series ............................................................................ 15
2.2 SYMBOLIC COMPUTATION - AN INTRODUCTION TO MATHEMATICA ...... 15
2.3 BASIC CONCEPTS IN PARALLEL PROCESSING .................................. 25
   2.3.1 Data Partitioning and Function Partitioning .................................. 26
   2.3.2 Execution Time, Speedup and Efficiency .................................... 27

## CHAPTER 3: SURVEY OF SINGLE STEP METHODS TO SOLVE
   ORDINARY DIFFERENTIAL EQUATIONS

3.1 SINGLE STEP AND TAYLOR SERIES METHODS ................................. 30
   3.1.1 Introduction .............................................................................. 30
   3.1.2 The Solution By Taylor Series .................................................. 30
   3.1.3 Euler's Method ......................................................................... 33
   3.1.4 Higher Order Taylor Series Method ......................................... 34
   3.1.5 The Existence and Uniqueness of Solutions ................................ 35
3.2 ERRORS, CONVERGENCE, CONSISTENCY AND STABILITY .............. 36
   3.2.1 Introduction .............................................................................. 36
   3.2.2 Local Truncation Error and Global Truncation Error ...................... 36
   3.2.3 Consistency, Stability and Convergence ..................................... 37
3.3 RUNGE-KUTTA METHOD ............................................................................... 39
  3.3.1 Introduction .......................................................................................... 39
  3.3.2 Second Order Runge-Kutta Method .................................................... 44
  3.3.3 Third Order Runge-Kutta Method ....................................................... 45
  3.3.4 Higher Order Runge-Kutta Methods .................................................... 46
  3.3.5 Stability Properties Of Runge-Kutta Methods ....................................... 51
3.4 LINEAR MULTISTEP METHOD AND EXTRAPOLATION METHOD .......... 53
  3.4.1 Introduction .......................................................................................... 53
  3.4.2 The Adams Methods ............................................................................ 56
  3.4.3 The Extrapolation Method .................................................................... 57
  3.4.4 The Other Methods ............................................................................. 58
3.5 STIFF EQUATIONS AND METHOD OF ABSOLUTE STABILITY .......... 60
  3.5.1 Introduction .......................................................................................... 60
  3.5.2 Stiffness of Initial Value Problem ........................................................ 60
  3.5.3 Stability Theory and Method For Stiff Problems ................................... 62

CHAPTER 4: NUMERICAL SOLUTION OF PROBLEMS INVOLVING ODEs BY USING THE VARIETY OF MEANS
4.1 DERIVATION OF A VARIETY OF MEANS FOR SOLVING ODE's ............... 67
4.2 NEW RUNGE-KUTTA FORMULA BASED ON THE CONTRAHARMONIC MEAN FORMULA ........................................................................................................ 68
  4.2.1 Second Order Contraharmonic Mean Formula ...................................... 68
  4.2.2 Third Order Contraharmonic Mean Formula ......................................... 69
  4.2.3 New Fourth Order Contraharmonic Mean Formula .............................. 70
  4.2.4 Truncation Error For The Contraharmonic Mean ................................. 73
  4.2.5 Stability Analysis Of The Contraharmonic Mean .................................. 81
  4.2.6 Numerical Example ............................................................................. 89
4.3 NEW RUNGE-KUTTA METHOD BASED ON THE CENTROIDAL MEAN FORMULA ........................................................................................................ 91
  4.3.1 Third Order Centroidal Mean Formula ............................................... 91
  4.3.2 New Fourth Order Centroidal Mean Formula ...................................... 92
  4.3.3 Error Analysis .................................................................................... 94
  4.3.4 Stability Analysis .............................................................................. 94
  4.3.5 Numerical Example ............................................................................ 96
4.4 NEW RUNGE-KUTTA METHOD BASED ON THE ROOT-MEAN-SQUARE (RMS) FORMULA ................................................................................................. 99
  4.4.1 Third Order Root-Mean-Square (RMS) Formula .................................. 99
  4.4.2 New Fourth Order Root-Mean-Square (RMS) Formula ....................... 102
  4.4.3 Numerical Example .......................................................................... 104

vii
4.5 THE IMPLICIT RUNGE-KUTTA METHODS .................................................. 106
  4.5.1 2-Stage Implicit Arithmetic Mean Runge - Kutta Method .................. 109
  4.5.2 2-Stage Implicit Contraharmonic Mean Runge - Kutta Method ............ 114
  4.5.3 2-Stage Implicit Centroidal Mean Method ........................................ 118
  4.5.4 2-Stage Implicit Harmonic Mean Method .......................................... 121

4.6 L-STABLE MODIFIED TRAPEZOIDAL METHODS FOR IVPs .................. 125
  4.6.1 L-Stable Modified Trapezoidal Formulas ........................................... 127
    4.6.1.1 Modified Arithmetic Mean (MAM) Trapezoidal Formula .................. 127
    4.6.1.2 Modified Contraharmonic Mean (MC_c) Trapezoidal Formula .......... 129
    4.6.1.3 Modified Centroidal Mean (MC_c) Trapezoidal Formula ................. 130
    4.6.1.4 Modified Root-Mean-Square (MRMS) Trapezoidal Formula .............. 132
  4.6.2 Numerical Example ........................................................................... 134

CHAPTER 5: THE PARALLEL IMPLEMENTATION OF THE RUNGE-KUTTA
CONTRAHARMONIC MEAN METHOD WITH EXTRAPOLATION

5.1 THE EXTRAPOLATION METHOD FOR ORDINARY DIFFERENTIAL
  EQUATIONS .................................................................................................. 138

5.2 ASYMPTOTIC EXPANSION OF THE CONTRAHARMONIC MEAN
  METHOD ........................................................................................................ 139

5.3 ROMBERG EXTRAPOLATION ................................................................. 145

5.4 PARALLEL IMPLEMENTATION OF ODE EXTRAPOLATION METHOD ... 147
  5.4.1 Parallel Implementation of the ODE Extrapolation Method ................ 147
  5.4.2 Parallelization Techniques ................................................................ 152
  5.4.3 Explicit Parallel Programming By ExDaTa Schedule ...................... 152
  5.4.4 ExDaTa : Scheduling Tool For Parallel Programming ..................... 153
  5.4.5 Parallel Implementation of the ODE Extrapolation Method Using the
        ExDaTa Schedule .................................................................................. 154

CHAPTER 6: THE THEORY OF RK(4,4) METHOD

6.1 RK(4,4) METHOD FOR ERROR-ESTIMATE ............................................ 166

6.2 ERROR CONTROL AND STEP SIZE SELECTION IN THE RK(4,4)
  METHOD ...................................................................................................... 171
  6.2.1 Local And Global Truncation Error ................................................... 175
    6.2.1.1 The Third Order Arithmetic Mean Method .................................. 176
    6.2.1.2 The Fourth Order Arithmetic Mean Method ............................. 178
    6.2.1.3 The Fourth Order Contraharmonic Mean Method ..................... 180
  6.2.2 Experimental Results For RK(4,4) ................................................... 182
6.3 ANALYSIS OF VARIETY OF MEANS METHOD ............................................. 186
  6.3.1 Elementary Differentials, Trees and Operation Diagram .................. 186
  6.3.2 Comparison of RKF(4,5), Merson and RK(4,4) Methods ................... 189
  6.3.3 Experimental Results For RKF(4,5) .................................................. 192
  6.3.4 Experimental Results For The Merson Method ............................... 193
  6.3.5 Automatic Selection of the Initial Stepsize for an ODE Solver .......... 200

6.4 THE EXPLICIT RUNGE-KUTTA METHOD .................................................. 202
  6.4.1 2-Stage Second Order Method ......................................................... 202
  6.4.2 3-Stage of Third Order Method ....................................................... 202
  6.4.3 4-Stage Fourth-Order Method ............................................................ 203
  6.4.4 5-Stage Fifth-Order Method ............................................................... 204

6.5 THE EXPLICIT CONTRAHARMONIC MEAN (C0M) METHOD .................... 206
  6.5.1 2-Stage Second Order C0M Method ................................................... 206
  6.5.2 A 3-Stage Third Order C0M Method .................................................. 207
  6.5.4 A 4-Stage Fourth Order C0M Method ................................................ 208

6.6 THE EXPLICIT CENTROIDAL MEAN (C,M) METHOD ................................ 209
  6.6.1 A 2-Stage Second Order C,M Method .................................................. 209
  6.6.2 A 3-Stage Third Order C,M Method ................................................... 210
  6.6.3 A 4-Stage Fourth Order C,M Method .................................................. 211

6.7 THE EXPLICIT ROOT-MEAN-SQUARE (RMS) METHOD ............................ 212
  6.7.1 A 2-Stage Second Order RMS Method .............................................. 212
  6.7.2 A 3-Stage Third Order RMS Method .................................................. 213
  6.7.3 A 4-Stage Fourth Order RMS Method ................................................ 214

CHAPTER 7: NUMERICAL SOLUTION OF ODEs BY NONLINEAR
MULTISTEP METHODS

7.1 NUMERICAL METHODS FOR FIRST ORDER ODEs .................................... 217

7.2 DERIVATION OF THE C0M METHOD FOR PROBLEMS OF THE TYPE y'' = f(x,y)220
  7.2.1 Error Analysis of (7.2.20), (7.2.21) and (7.2.22) ....................... 225
  7.2.2 Numerical Results For (7.2.16), (7.2.17) and (7.2.18) .................... 226

7.3 DERIVATION OF THE C,M METHOD FOR PROBLEMS OF THE TYPE y'' = f(x,y)232
  7.3.1 Error Analysis of (7.3.10) and (7.3.11) .......................................... 235
  7.3.2 Numerical Results For (7.3.10) and (7.3.11) .................................... 237

7.4 DERIVATION OF THE H,M METHOD FOR PROBLEMS OF THE TYPE y'' = f(x,y)242
  7.4.1 Error Analysis of (7.4.10) and (7.4.11) .......................................... 245
  7.4.2 Numerical Results For (7.4.10) and (7.4.11) .................................... 246

7.5 NUMERICAL SOLUTION FOR SOLVING ODEs WITH NONLINEAR
2-STEP METHOD ......................................................................................... 253
7.5.1 Geometric Mean (GM) 2-Step Method ................................................................. 253
7.5.2 Contraharmonic Mean (C_oM) 2-Step Method .................................................. 258
7.5.3 Centroidal Mean (C_M) 2-Step Method ............................................................... 262
7.5.4 Harmonic Mean (H_M) 2-Step Method ................................................................. 265

CHAPTER 8: A NEW FIFTH ORDER WEIGHTED RUNGE-KUTTA FORMULA

8.1 THE FOURTH ORDER ARITHMETIC MEAN WEIGHTED RUNGE-KUTTA FORMULA ................................................................. 273
8.2 A NEW FIFTH ORDER ARITHMETIC MEAN WEIGHTED RUNGE-KUTTA FORMULA ................................................................................. 275
  8.2.1 Error Analysis ................................................................................................. 282
  8.2.2 Stability Analysis of the Fifth Order WRK Method ........................................ 282
  8.2.3 Numerical Example ....................................................................................... 284
8.3 NEW FIFTH ORDER CONTRAHRARMONIC MEAN WEIGHTED RUNGE-KUTTA FORMULA ............................................................. 285
  8.3.1 The Fourth Order Contrahmonic Mean Weighted Runge-Kutta Formula ........ 285
  8.3.2 A New Fifth Order Contrahmonic Mean Weighted Runge-Kutta Formula .... 287
  8.3.3 Error Analysis Fifth Order C_oM Method ...................................................... 296
  8.3.4 Stability Analysis Fifth Order C_oM Method ................................................ 297
  8.3.5 Numerical Example ....................................................................................... 299
8.4 THE THEORY OF WEIGHTED RK(5,5) METHOD ............................................. 300
  8.4.1 RK(5,5) Method For Error Estimate and Error Control ................................. 300
  8.4.2 Experimental Results For RK(5,5) ................................................................. 305
8.5 WEIGHTED FIFTH-ORDER RUNGE-KUTTA FORMULAS FOR SECOND- ORDER DIFFERENTIAL EQUATIONS ......................................................... 307
  8.5.1 Weighted Fifth-Order Formula For A System of Two First-Order Equations 309
  8.5.2 Weighted Fifth-Order Formulas for Second-Order Differential Equations ... 310
  8.5.3 Numerical Example ....................................................................................... 312
8.6 NEW RUNGE-KUTTA STARTERS FOR MULTISTEP METHODS ...................... 313
  8.6.1 Starting By Linear and Nonlinear Methods .................................................... 313
  8.6.2 Explicit Fourth and Fifth Order Multistep Adam Method ............................. 314
  8.6.3 Numerical Example ....................................................................................... 316

CHAPTER 9: HIGH ORDER INTEGRATION FORMULA USING TRIGONOMETRIC POLYNOMIALS FOR PERIODIC IVPs

9.1 RUNGE-KUTTA METHODS FOR THE OSCILLATORY PROBLEM ...................... 322
9.2 THE NUMERICAL SOLUTION OF OSCILLATORY PROBLEM .............................. 323
CHAPTER 1

INTRODUCTION
The introduction of computers have brought a fundamental change in the nature of research in education, business, management, science and technology. Users or experimentalist use computers to collect and analyze data while scientists or theoreticians use computers in mathematical problems and to manipulate equations numerically and symbolically. For both, the computer is a tool which if used correctly, can lead to significantly reduced times and costs. The use of computers in business, management, science and technology has been rapid and computer hardware and software development has progressed where nearly most people can afford to buy a computer. In some places in the world today, universities and colleges are struggling to obtain funds and financial support to establish a computer laboratory and to introduce new courses related to the application of computers. Many problems in business, management, science and technology can be formulated in terms of linear or nonlinear equations and differential equations. The major motivations for building the early computers came from the need to compute the complex linear or nonlinear equations and differential equations accurately and quickly. Thus, a computational or numerical method is used to solve the above problems. Therefore, the computational solution and the numerical solution of differential equations becomes important because of the numerous physical problems in business, management, science and technology which lead to differential equations that cannot be solved analytically.

1.1 DIFFERENTIAL EQUATIONS

Mathematical models of problems in business, management, science and technology involve differential equations or systems of differential equations. For instance, problems in mechanics such as the motion of projectiles or orbiting bodies, in population growth, in
chemical kinetics and in economic growth may be modelled by differential equations.

Differential equations are equations involving an unknown function of time or space and one or several of its derivatives. The study of differential equations is normally divided into two categories namely ordinary differential equations (ODEs) and partial differential equations (PDEs). Our concern in this study is ODEs where differential equations involving only ordinary derivatives \( y', y'', y''', \ldots, y^{(m)} \) of unknown functions or the dependent variable \( y \) with respect to the independent variable \( x \) in the form of

\[
f(x, y, y', y'', \ldots, y^{(m)}) = 0. \quad (1.1.1)
\]

Those differential equations which involve partial derivatives are called partial differential equations. For example,

\[
y' = 1 + x - y \quad (1.1.2)
\]

\[
y'' + 101(y')^3 + 100y = 0 \quad (1.1.3)
\]

\[
(y''')^2 + xy' + y = \sin(x) \quad (1.1.4)
\]

\[
\frac{\partial^3 u}{\partial x^3} + \frac{\partial^3 u}{\partial t^3} + \frac{\partial^3 u}{\partial y^3} = 4u \quad (1.1.5)
\]

are all differential equations. Equations (1.1.2)-(1.1.4) with unknown function \( y \) of an independent variable \( x \) and \( y = y(x) \). The differential equation is said to be of order \( m \) if the derivative of the highest order in the equation is \( y^{(m)} = \frac{d^m y}{dx^m} \), whilst the power to the highest derivative in the equation is called the degree. Therefore, equations (1.1.2) and (1.1.3) are of the first degree while equation (1.1.4) is of second degree because the highest derivative is power 2. The terms of \( y', y'' \) and \( y''' \) in (1.1.2)-(1.1.4) respectively are first, second and third derivative of the function \( y \) with respect to \( x \). While in (1.1.5) the unknown function \( u \) is the function respect to the independent variables \( x, t \) and \( y \) and that
Thus, equation (1.1.5) is the third order linear partial differential equation of the function $u(x, t, y)$ with respect to $x$, $t$ and $y$. From the above example, equations (1.1.2)-(1.1.4) are called ODEs while equation (1.1.5) only involves partial derivatives is called PDE.

Differential equations generally have many solutions and extra conditions known as boundary conditions, must be imposed to single out a particular solution. These boundary conditions usually takes the form of the solution or derivatives for a particular value of $x$ and can be shown that a differential equation of order $m$ requires $m$ boundary conditions. If all the boundary conditions apply at one value of $x$ they are called initial conditions and the differential equation together with the initial conditions is called an initial value problem. If more than one value of $x$ is involved in the boundary conditions it is called a boundary value problem.

A differential equation is linear if $y$ and its derivatives occur linearly and be written in the form

$$a_n(x)y^{(n)} + a_{n-1}(x)y^{(n-1)} + \ldots + a_0(x)y = g(x). \quad (1.1.6)$$

If $g(x) = 0$ the equation in (1.1.6) is said to be homogeneous and if $g(x) \neq 0$, it is said to be non-homogeneous. Equation (1.1.1) is linear and homogeneous but equation (1.1.3) is non-linear homogeneous, equation (1.1.4) is non-linear and non-homogeneous and equation (1.1.2) is linear non-homogeneous. A linear homogeneous boundary value problem is called the eigenvalue problem.

1.2 NUMERICAL SOLUTION OF INITIAL VALUE PROBLEMS

Differential equations are commonly used in many practical problems which lead to ordinary differential equations and most differential equations which arise in applications cannot be solved using analytical methods.
Therefore, it is necessary to make numerical approximations. The analytical solutions can only be determined in a few simple cases. For this reason, the advancement of sophisticated modern computers from the personal computer to the parallel computer are being widely used as a tool for solving numerical methods.

The idea of numerical methods which are described by Elden and Wittneyer [1990] are based on the following technique. Since we cannot determine the function \(y(x)\) for all \(x\) in the interval \([a, b]\) we will be satisfied with computing approximations \(y_i\) of \(y(x_i)\) for some points \((x_i)_{i=0}^N\) in the interval. We assume that the points are equidistant, i.e., \(x_i = a + ih, \quad i = 0, 1, \ldots, N\), where the step length \(h\) is defined as \(h = \frac{b - a}{N}\) for some integer \(N\) and the interval \([a, b]\) is divided into \(N\) subinterval as shown in Figure 1.1.

\[
\begin{array}{cccccccc}
\text{a=}&x_0&x_1&x_2&x_3&\ldots&x_{N-1}&b=x_N
\end{array}
\]

Figure 1.1: The interval divided into \(N\) subinterval

Given the initial value \(y_0 = y(x_0) = \alpha\) at the point \(x = x_0 = a\). Now we want to compute \(y_i\) which approximates \(y(x_i)\) by discretizing the differential equation. In general, we assume that we know \(y_{s-1}\) or \(y(x_s)\) and we want to compute \(y_{s+1}\). The differential equation at point \(x = x_s\) is

\[
y'(x_s) = f(x_s, y(x_s)) \quad (1.2.1)
\]

and by replacing the derivative by a difference quotient, we have

\[
y'(x_s) = \frac{y(x_{s+1}) - y(x_s)}{h} \quad (1.2.2)
\]

and the differential equation then becomes

\[
\frac{y(x_{s+1}) - y(x_s)}{h} = f(x_s, y(x_s)) \quad (1.2.3)
\]
By replacing \( y(x) \) and \( y(x_{n+1}) \) with \( y_n \) and \( y_{n+1} \) in the difference quotient, we obtain

\[
y_{n+1} = y_n + hf(x_n, y_n)
\]

what we know today as the first order classical method for the numerical solution of initial value problems and called Euler method.

In general there are two classes of methods, a single step method and a multistep method. A single step or one-step method is where the numerical solution for some value \( x_{n+1} \) of the independent variable is calculated using information from only the previous \( x_n \). In a multistep method or a k-step method, information is used from the previous k values which means, \( y_{n+1} \) is calculated using values of \( y \) computed at \( x_{n+j} = x_n + jh, \ j = 0, 1, 2, \ldots, k-1 \).

In general, a special procedure is needed for \( k \geq 2 \) as to begin with only one value is known from the initial condition. For this type of solution, any suitable single step method can be applied to calculate the required values.

Most cases of ordinary differential equations of higher order, can be reduced to a first order system of ordinary differential equations. For the \( m^{th} \) order initial value problem in the form

\[
y^{(m)} = f(x, y, y', \ldots, y^{(m-1)})
\]

and the initial value can be written as

\[
y^{(0)}(\alpha) = \beta_i, \quad i = 1, 2, \ldots, m.
\]

By replacing with variable \( y_i, i = 1, 2, \ldots, m \) where

\[
y_1 = y,

y_2 = y',

y_3 = y'',

\vdots

y_m = y^{(m-1)}.
\]
The equation (1.2.7) now can be written as the first order system of the initial value problem in the form

\[
\begin{align*}
  y'_1 &= y_2, & y_1(\alpha) &= \beta_1, \\
  y'_2 &= y_3, & y_2(\alpha) &= \beta_2, \\
  \vdots & \quad \vdots & \quad \vdots \\
  y'_n &= f(x, y_1, y_2, \ldots, y_n), & y_n(\alpha) &= \beta_n \\
\end{align*}
\]  

and by using the vector form, the equation (1.2.8) can be written as

\[
\begin{align*}
  y' &= f(x, y), & y(\alpha) &= \beta \\
\end{align*}
\]

which defines a system of \( m \) first order equations or \( m^\text{th} \) order initial value problem. Throughout the rest of this thesis only first order equation or a system of \( m \) first order equation will be discussed. Thus, to solve this first order equation or a system of \( m \) first order equation, a method and program was developed and run on the Sequent Balance Machine.

1.3 THE SEQUENT BALANCE SYSTEM

The Sequent Balance system is a multiprocessor computer that incorporates multiple identical central processing units (CPUs) and a single common memory. The Sequent Balance offers two kinds of series: the Balance Series and Symmetry Series. The Balance Series support parallel programming and is used in a variety of fields of research which lead to commercial applications.

The Balance Series is divided in two: namely the Sequent Balance 8000 and Balance 21000 systems. The Balance 8000 computer, is the smaller system designed to employ from 2 to 12 processors and can deliver processing speeds of 10 million instructions per second (MIPS). The Sequent Balance 21000 includes from 4 to 30 processors is the larger system. In the Balance Systems, all the processors share from 4 to 28 Megabytes of physical
memory and can provide 16 Megabytes of virtual address space per process. Balance computers run the DYNIX (DYnamic uNIX) operating system, a version of UNIX 4.2bsd that supports most utilities, libraries and system calls provided by UNIX System V. This machine can be used for both sequential programming and parallel programming.

1.4 PARALLEL PROCESSING AND PARALLEL PROGRAMS

Parallel processing is the technique of using many processors together in concert arranging for each to perform a separate portion of a task. In other words, processing by a number of different processors simultaneously on one particular process, each processor dealing with a different slice of the process. In this way, many tasks can be completed very quickly by sharing out the work. All these tasks can be extended as required and can be simplified by separation into self sustaining modules.

By definition, parallel programs are executed simultaneously or concurrently meaning that at any instant, the system is in the process of executing multiple programs. In adapting an application for efficient parallel programming, the following objectives should be borne in mind:

a) Choose the right programming method
b) Run as much of the program in parallel as possible
c) Balance the computational load as possible in parallel processes
d) To achieve ultimately the execution speed of the program.
1.5 PARALLEL PROGRAMMING LANGUAGE IN C ON THE SEQUENT BALANCE SYSTEM

By definition, C is a general purpose and a high level programming language designed for system programming, usually for software development in the UNIX environment. UNIX has been written in C and therefore a simple way to install UNIX on a particular system is to use a C compiler to compile UNIX for that particular system. It is flexible, convenient, powerful and efficient. C is rapidly becoming one of the most popular and widely used programming languages for the development of applications.

Basically, there are two kinds of parallel programming: multiprogramming and multitasking. Multiprogramming is an operating system feature that allows a computer to execute multiple unrelated programs concurrently. Multitasking is a programming technique that allows a single application to consist of multiple processes executing concurrently. The Sequent Balance system language software includes multitasking extensions to C, Pascal and Fortran. The DYNIX Parallel Programming Libraries includes routines to create, synchronise and terminate parallel processes from C programs and it includes:

a) allocation of memory for shared data
b) creation of processes to execute subprograms in parallel
c) identification of individual processes
d) suspension of processes during serial program sections
e) mutual exclusion on shared data
f) synchronisation of processes during critical sections.
CHAPTER 2

BASIC PRELIMINARIES, SYMBOLIC COMPUTATION AND BASIC CONCEPTS IN PARALLEL PROCESSING
This chapter covers the discussion of basic topics that are required in the following chapters, together with certain definitions related to the discussion for the solution of ordinary differential equations and parallel processing.

2.1 BASIC PRELIMINARIES

The definitions of terms and basic software tools for numerical methods will be discussed in the following sections.

2.1.1 Mathematical Means

With a set of \( n \) numbers \( X_i \) for \( i = 1, 2, ..., n \) and \( X_i \geq 0 \), the generalised mean is defined by

\[
\text{Mean} = \left[ \frac{1}{n} \sum_{i=1}^{n} X_i^m \right]^{\frac{1}{m}}
\]  

(2.1.1)

a) Arithmetic Mean

For \( m = 1 \) in equation (2.1.1), we have the arithmetic mean (AM) which can be written in the form

\[
AM = \frac{1}{n} \sum_{i=1}^{n} X_i
\]  

(2.1.3)

For \( X_i \neq X_{i+1} \) and \( n=2 \), the arithmetic mean (AM) is defined as

\[
AM = \frac{X_i + X_{i+1}}{2}
\]  

(2.1.4)

b) Harmonic Mean

If \( m = -1 \) is substituted in equation (2.1.1), the harmonic mean \( (H_aM) \) is shown as

\[
H_aM = \left[ \frac{1}{n} \sum_{i=1}^{n} X_i^{-1} \right]^{-1}
\]  

(2.1.5)

and for \( X_i \neq X_{i+1} \) and \( n=2 \), the harmonic mean \( (H_aM) \) can be written as

\[
H_aM = \frac{2X_iX_{i+1}}{X_i + X_{i+1}}
\]  

(2.1.6)
c) Geometric Mean

The geometric mean is defined by

$$GM = \left( \prod_{i=1}^{n} X_i \right)^{\frac{1}{n}}$$ \hspace{1cm} (2.1.7)

and for $X_i \neq X_{i+1}$ and $n=2$, the geometric mean (GM) has the form

$$GM = \sqrt{X_i X_2}$$ \hspace{1cm} (2.1.8)

The equations (2.1.4), (2.1.6) and (2.1.8) satisfy the following inequality

$$AM \geq GM \geq H_{\#}M \, .$$

d) Contraharmonic Mean

For two positive numbers $X_i$ and $X_{i+1}$, the contraharmonic mean ($C_{oM}$) is defined as

$$C_{oM} = \frac{\sum_{i=1}^{n} X_i^2}{\sum_{i=1}^{n} X_i}$$

and for $X_i \neq X_{i+1}$ and $n=2$,

$$C_{oM} = \frac{X_1^2 + X_2^2}{X_1 + X_2}$$ \hspace{1cm} (2.1.9)

e) Centroidal Mean

For $X_i \neq X_{i+1}$, the centroidal mean ($C_{eM}$) is defined by

$$C_{eM} = \frac{2(X_i^2 + X_{i+1}^2 + X_{i+1})}{3(X_i + X_{i+1})}$$ \hspace{1cm} (2.1.10)

f) Root Mean Square

Another new mean for $X_i \neq X_{i+1}$, the root mean square ($RMS$) is defined by

$$RMS = \sqrt{\frac{X_i^2 + X_{i+1}^2}{2}}$$ \hspace{1cm} (2.1.11)
g) Heronian Mean

For two positive numbers $X_i$ and $X_{i+1}$, the heronian mean ($H_iM$) is defined by

$$H_iM = \frac{X_i + X_{i+1} + \sqrt{X_i X_{i+1}}}{3}$$  \hspace{1cm} (2.1.12)

Eves (1983, pp 153) showed that, for $X_i \neq X_{i+1}$, all types of means are related to each other by the decreasing inequality

$$C_iM \geq RMS \geq C_iM \geq AM \geq H_iM \geq GM \geq H_iM$$

h) Logarithmic Mean

In addition to the above means, we also have another mean called logarithmic mean ($LM$). For $X_i \neq X_{i+1}$, the logarithmic mean ($LM$) is defined by

$$LM = \frac{X_{i+1} - X_i}{\ln \left( \frac{X_{i+1}}{X_i} \right)}$$ \hspace{1cm} (2.1.13)

i) Weighted Arithmetic Mean

Suppose that $X = [X_1, X_2, ..., X_n]$ and $W = [W_1, W_2, ..., W_n]$ be two ordered sets of values in $R$. We now define the three types of weighted means where the weighted arithmetic mean for $X_i \neq X_{i+1}$, is defined by

$$W_{AM} = \frac{\sum_{i=1}^{n} W_i X_i}{\sum_{i=1}^{n} W_i}$$

or

$$W_{AM} = \frac{W_i X_i + W_{i+1} X_{i+1}}{W_i + W_{i+1}}$$ \hspace{1cm} (2.1.14)

j) Weighted Geometric Mean

The weighted geometric mean ($W_{GM}$) with weight $W$, and for $X_i \neq X_{i+1}$ is defined as
or
\[ W_{GM} = \left[ \prod_{i=1}^{n} X_i^{w_i} \right]^{\frac{1}{\sum w_i}} \]

or
\[ W_{GM} = \left[ X_i^{w_i} + X_{i+1}^{w_{i+1}} \right]^{\frac{1}{w_i + w_{i+1}}} \] (2.1.15)

**k) Weighted Harmonic Mean**

The weighted harmonic mean \( W_{H,M} \) with weight \( W \) and for two positive numbers \( X_i \) and \( X_{i+1} \), is defined by

\[ W_{H,M} = \frac{\sum_{i=1}^{n} W_i}{\sum_{i=1}^{n} \left( \frac{W_i}{X_i} \right)} \]

or

\[ W_{H,M} = \frac{W_i + W_{i+1}}{X_i + X_{i+1}} \] (2.1.16)

In most cases of applications, the weights \( W \) is defined by \( \sum W_i = 1 \).

**2.1.2 Difference Equations**

A linear difference equation of second order is an equation of the type

\[ y_{n+2} + by_{n+1} + cy_n = 0, \quad n = 0, 1, 2, \ldots \] (2.1.17)

The equation (2.1.17) is homogeneous, i.e., the right hand side is equal to zero. If we have the initial values,

\[ y_0 = \alpha \quad \text{and} \quad y_1 = \beta \]

then the difference equation uniquely defines \( y_0, y_1, y_2, \ldots \) in order to compute \( y_0, y_1, y_2, \ldots \) from the equation. In order to get an expression for the solution, we use \( y_n = r^n \) to obtain

\[ r^{n+2} + br^{n+1} + cr^n = 0 \]

and divided by \( r^n \), we have

\[ r^2 + br + c = 0 \]

which we called the characteristic equation.
2.1.3 Homogeneous Difference Equation

A linear homogeneous difference equation of order \( k \) is an equation of the type

\[ y_{n+k} + a_k r^{k-1} + \ldots + a_1 r + a_0 = 0 \]

then the difference equation has the general solution

\[ y_n = c_1 r_1^n + c_2 r_2^n + \ldots + c_k r_k^n, \]

where \( c_1, c_2, \ldots, c_k \) can be determined from \( k \) initial values. If \( r_i \) is a root of multiplicity \( m \) of the characteristic equation, then

\[ y_n = P_{m-1}(n) r_i^n \]

where \( P_{m-1} \) is a polynomial of degree \( m - 1 \) which satisfies the difference equation.

2.1.4 Taylor Series

For solving the ordinary differential equation

\[ y' = f(x, y), \quad x \in [a, b] \]

and \( y(a) = y_0 \)

where \( y = [y(x), y'(x), \ldots, y^{(m)}(x)]^T \) and \( y_0 \) is a constant, then from point \( x_n \) and \( x_{n+1} \), we can use the Taylor Series method in the form

\[ y(x + h) = y(x) + hy'(x) + \frac{h^2}{2!} y''(x) + \frac{h^3}{3!} y'''(x) + \ldots + \frac{h^p}{p!} y^{(p)}(x) + \ldots \]

For numerical purposes, the Taylor series truncated after \( (n+1) \) terms enables us to compute \( y(x + h) \) rather accurately if \( h \) is small and \( y(x), y'(x), y''(x), \ldots, y^{(p)}(x) \) are known. When terms through \( \frac{h^p}{p!} y^{(p)}(x) \) are included in the Taylor series, it is called the Taylor Series method of order \( p \) and will be discussed in the next chapter.

2.2 SYMBOLIC COMPUTATION - AN INTRODUCTION TO MATHEMATICA

Symbolic computation is referred to as the technique of the manipulation of symbolic expressions on the computer. This section introduces the new symbolic
manipulation systems as a powerful tool for symbolic algebraic computation and its potential in several areas: numerical mathematics, mathematical modelling, simulation, graphics applications, neural networks and financial modelling.

There are several symbolic manipulation systems available and suitable which have appeared on the market. Following Corliss [1994] , the symbolic computation systems which have emerged to date are listed in Table 2.1.

Table 2.1 : Survey of Available Software for Symbolic Computation

<table>
<thead>
<tr>
<th>Name</th>
<th>Language</th>
<th>Machine</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABACUS</td>
<td></td>
<td></td>
<td>Rump et al.</td>
</tr>
<tr>
<td>ACRITH-XCS</td>
<td>Fortran 77</td>
<td>IBM S/370</td>
<td>Kulish et al.</td>
</tr>
<tr>
<td>AQUARELS</td>
<td>Fortran 77</td>
<td></td>
<td>Erhel et al.</td>
</tr>
<tr>
<td>BIAS</td>
<td>Ada</td>
<td>VAX</td>
<td>Corliss</td>
</tr>
<tr>
<td>BIAS</td>
<td>Fortran 77</td>
<td>any F77</td>
<td>Neumaier</td>
</tr>
<tr>
<td>BIAS</td>
<td>C++</td>
<td></td>
<td>Knuempel</td>
</tr>
<tr>
<td>CLEMMENSON</td>
<td>MS Fortran</td>
<td>PC</td>
<td>Wang and Kennedy</td>
</tr>
<tr>
<td>C-XSC</td>
<td>C++</td>
<td>PC</td>
<td>Kulisch et al.</td>
</tr>
<tr>
<td>FORTRAN-XSC</td>
<td>Fortran 90</td>
<td>any F90</td>
<td>Walter</td>
</tr>
<tr>
<td>HAMMER</td>
<td>Pascal-XSC</td>
<td>any P-XSC</td>
<td>Hammer</td>
</tr>
<tr>
<td>INTLIB 77</td>
<td>Fortran 77</td>
<td>any F77</td>
<td>Kearfot et al.</td>
</tr>
<tr>
<td>INTLIB 90</td>
<td>Fortran 90</td>
<td>any F90</td>
<td>Kearfot</td>
</tr>
<tr>
<td>INTPAK</td>
<td>Maple</td>
<td>any Maple</td>
<td>Connell and Corliss</td>
</tr>
<tr>
<td>INTPAK</td>
<td>Mathematica</td>
<td>any Math</td>
<td>Kieper</td>
</tr>
<tr>
<td>MATHEMATICA</td>
<td>Mathematica</td>
<td>many</td>
<td>Wolfram , S</td>
</tr>
<tr>
<td>MODULA-SC</td>
<td>Mathematica</td>
<td>many</td>
<td>Ullrich et al.</td>
</tr>
<tr>
<td>PASCAL-XSC</td>
<td>Pascal</td>
<td>many</td>
<td>Kulisch et al.</td>
</tr>
<tr>
<td>PBASIC</td>
<td>Basic</td>
<td>PC</td>
<td>Aberth</td>
</tr>
<tr>
<td>RANGE</td>
<td>C++</td>
<td>Some C++</td>
<td>Aberth</td>
</tr>
<tr>
<td>TPX</td>
<td>Pascal</td>
<td>PC</td>
<td>Rump et al.</td>
</tr>
</tbody>
</table>
Jenson and Niordson [1977] also has given a comparative study of some of the other packages which include: ALPAK, ALTRAN, CAYLEY, FORMAC, MACSYMA, MATLAB, muMath, REDUCE, SAC-1, SCHOONSCHIP, SCTRATCHPAD, SMP, SYMBAL and CONFORM for the purpose of symbolic manipulation. One of the examples, REDUCE is also used widely as a tool in symbolic manipulation by Sanugi [1986] and Jayes [1992] in their research work on the VAXII at Loughborough University.

Here we describe Mathematica. Mathematica is a general system for doing numerical, symbolic and graphical computation, used regularly both as an interactive calculation tool and as a programming language. Following Gaylord et. al [1996], Mathematica is a very large and seemingly complex system. It contains hundreds of functions for performing various tasks in Science, Mathematics, Management, Financial Analysis and Engineering. Mathematica has its own language with well defined rules, syntax and combines characteristics of a word processor, outliner, spreadsheet, interactive computer language, statistical graphics program and more in one immensely powerful package (see Caudill [1990]). In other word, Mathematica is a powerful tool for computing programming, data analysis, knowledge representation and visualization of information and can be run on over 20 platforms including Macintosh, Power Macintosh, Microsoft Windows, Windows NT, MS-DOS, IBM OS/2, Sun SPARC, DEC OSF/1 AXP, Open VMS AXP, RISC Ultrix, VAX/VMS, HP, Hitachi, IBM RISC, MIPS, NEC PC, NEW EWS, NEXTSTEP for Intel, HP PA-RISC, Motorola and others.

Mathematica can be run by using two basic types of interface: the "textual" or command line interface and the "notebook" or graphical interface. The textual interface is used on a PC not running Windows or a VAX workstation while the notebook interface is used on a Macintosh, a PC running Windows and most Unix workstations. However, using Mathematica on a computer that does not have a notebook interface is not a disadvantage if the user can
just remember a few additional commands. But by using a notebook we can write text, perform computation and create graphics all in one document. Notebook also has a facility which makes it easy to copy and paste graphics or to import to any files in Microsoft Word.

We shall now illustrate the use of the textual interface of Mathematica. After a successful logging-in, the user can run Mathematica on the PARC-PC-2 at Loughborough University by typing the word 'math' at the prompt 'C:\MATH'. The following output results depend upon the computer system being used, i.e.,

C:\MATH>math
Mathematica 2.0 for MS-DOS 386
Copyright 1988-91 Wolfram Research, Inc.

In[1]:=

We can also run Mathematica using a notebook interface on the PARC-PC-1 by a double click on the Mathematica icon which will look something like in three dimensional graphics as a stellated icosahedron and when a blank notebook first appears on the screen, select New in the Notebook menu which then prompts the user for input by

In[1]:=

After typing the input, press Shift-Return or Shift-Enter. For example, if one wants to work on the factor polynomials, it can be written as

In[1]:= Factor[x^2-4]

and by pressing the Enter or Return key, the system evaluate it and gives the output in the form

Out[1]:= (-2 + x)(2 + x)

In[2]:=
where (In [2]:=) is automatically assigned to the second command. Mathematica has a convenient user interface where the previous result or calculation can be used by the symbol %, i.e.,

\begin{align*}
\text{In}[2] & := \% \\
\text{Out}[2] & := (-2 + x)(2 + x).
\end{align*}

Mathematica also can refer to the result of any earlier calculation by using Out[i] or %i. For example

\begin{align*}
\text{In}[3] & := \text{Out}[1] \\
\text{Out}[3] & := (-2 + x)(2 + x)
\end{align*}
or

\begin{align*}
\text{In}[4] & := \%1 \\
\text{Out}[4] & := (-2 + x)(2 + x)
\end{align*}

when the expression ends with a semicolon (;), Mathematica computes its values but does not print it. This is very helpful when the results of the expression would be very long and can be shown as

\begin{align*}
\text{In}[5] & := a = \text{Factor}[x^2-4]; \\
\text{In}[6] & := \text{Expand}[a] \\
\text{Out}[6] & := -4 + x^2
\end{align*}

The syntax of inputs in Mathematica consist of expressions and numerical. For example, a slash (/) for division, a caret (^) for exponentiation. Mathematica also allows to indicate multiplication by juxtaposing two expression or using the asterisk (*) for the multiplication as is traditional in computer programming.

The operations in Mathematica is given the same precedence as in mathematics. For example, multiplication and division have a higher precedence than addition and subtracting, i.e., \(2 + 8 \times 5\) equals 42 and not 50. While the square brackets [ and ] are used to enclose the arguments to function while the curly braces { and } to indicate a list or range of values. For example
In[7]:= Solve [{x+y-2==0,3x+2y-6==0},{x,y}]
Out[7]:={x->2,y->0}.

To enter a comment or some words that are not evaluated, the words are entered between (* and *) and is shown by

In[8]:= Solve[({x+y-2==0, 3x+2y-6==0},{x,y}) (* The solution to *) (*a simple set of *) (*simultaneous equations*)
Out[8]:={x->2,y->0}.

Another example of the usefulness of Mathematica is that we may use it as a tool to convert from a certain number to scientific notation by using the N function. While to convert from decimal form to rational form the Rationalize function is used.

In[9]:= N[2^100]
Out[9]:= 1.26765×10^{} and In[10]:= Rationalize[0.125]

A Mathematica program consists of a set of functional commands which are evaluated sequentially by the computer. These commands are constructed from statements and expressions which were explained in the previous paragraphs. Now, we shall illustrate a simple Mathematica program for solving a system of nonlinear equations in the form

1 - \(w_i - w_j - w_k = 0\),
2 - \(2a_iw_i - w_j - 2a_jw_j - 3w_k = 0\),
2 - \(3a_iw_i + 6a_ja_kw_j - 3w_j - 3a_kw_j + 6a_iw_i + 3a_kw_j + 3a_kw_j - 6a_iw_j = 0\),
8 - \(12a_i^2w_i - 3w_j - 12a_i^2w_j - 15w_k = 0\),
1 - \(6a_iw_i + 12a_iw_j + 6a_iw_j - 12a_ja_kw_j + 6a_iw_j - 12a_iw_j = 0\),
8 - \(6a_iw_j - 6a_iw_j + 12a_iw_j + 12a_i^2w_j - 15a_j - 6a_iw_j - 6a_iw_j + 12a_iw_j + 15a_j - 24a_iw_j - 12a_i^2w_j = 0\),
4 - \(8a_i^3w_i - w_j - 8a_i^3w_j - 9w_k = 0\).
The above system of equations can be written as a Mathematica program as follows:

\[
\begin{align*}
\text{In}[11]:= \text{s} &= \text{Solve}\{1-w_1-w_2-w_3 == 0, \\
&\quad 2-2a_1w_1-w_2-2a_1w_2-3w_3 == 0, \\
&\quad 2-3a_1w_2+6a_1a_2w_2-3w_2-3a_1w_3+6a_1a_2w_3+3a_1w_3-6a_1a_2w_3 == 0, \\
&\quad 8-12a_1^3w_1-3w_2-12a_1^3w_2-15w_3 == 0, \\
&\quad 1-6a_1w_1+12a_1a_2w_1+6a_1a_2w_2-12a_1a_2a_3w_1+6a_1a_2w_3-12a_1a_2a_3w_3 == 0, \\
&\quad 8-6a_1w_2-6a_1^3w_2+12a_1a_2w_2+12a_1^3a_2w_2-15w_3-6a_1w_3 \\
&\quad -6a_1^3w_3+12a_1a_2w_3+12a_1^2a_2w_3+15a_1w_3-24a_1a_2w_3-12a_1^2a_2w_3 == 0, \\
&\quad 4-8a_1^3w_1-w_2-8a_1^3w_2-9w_3 == 0\}, \\
\{a_1,a_2,a_3,a_4,w_1,w_2,w_3\} ;
\end{align*}
\]

If the system of equations is kept in a file named 'sam44' then

\[
\begin{align*}
\text{In}[12]:= << \text{sam44}
\end{align*}
\]

will load the file named 'sam44'. By using the interactive mode, the output from this program can be shown by

\[
\begin{align*}
\text{In}[13]:= \text{s} \\
\text{Out}[13]:&=\{\{w_1->\frac{1}{2},w_2->\frac{1}{3},a_1->0,a_2->0,a_3->0,a_4->\frac{1}{2},w_3->\frac{1}{3}\}\}
\end{align*}
\]

When using the batch mode if we need to transfer the result from the file 'sam44' to the other file, the following command is used

\[
\begin{align*}
\text{In}[14]:= \text{s} >> \text{result1}
\end{align*}
\]

where the result from the above problem was saved in the file named 'result1'. To open the file to read the result from file 'result1', we use

\[
\begin{align*}
\text{In}[15]:= \text{!vi result1}
\end{align*}
\]
Mathematica also provides a wide range of graphics capabilities including two and three-dimensional plots of functions or data sets and contours. In addition, the Mathematica programming language allows the construction of the graphical images from the ground up using the primitive elements (see Gaylord et.al [1996]).

Plotting functions of one variable of the function \( \sin(\exp(x)) \) with \( x \) range from 0 to \( \pi \) is shown as

\[
\text{In}[16]:= \text{Plot}[\sin[\exp[x]],\{x,0,\pi\}]
\]

\[
\text{Out}[16]:= -\text{Graphics}-
\]

Mathematica provides many options that we can set to determine exactly how the graphics will appear, i.e.,

\[
\text{In}[17]:= \text{Show}[%, \text{Frame} \to \text{True}, \text{FrameLabel} \to \{"Time","Signal"}, \text{GridLines} \to \text{Automatic}]
\]

\[
\text{Out}[17]:= -\text{Graphics}-
\]

Functions of one variable can be represented as curves in the plane, while functions of two variables are visualized as surfaces in space and Mathematica has
numerous constructions for visualizing these surfaces. For example, the three-dimensional fourth order standard Runge-Kutta formula can be shown as a three-dimensional or graphic surface as follows:

\textbf{In[18]:= } f = \text{Abs}[((1+z+(z^2/2)+(z^3/6)+(z^4/24)) / z \rightarrow (x + I y)];
\textbf{In[19]:= } \text{Plot3D}[f, \{x,-4,4\}, \{y,-4,4\},
\text{AxesLabel} \rightarrow \{"Re(z)","Im(z)","Height"},
\text{ViewPoint} \rightarrow \{-2,-2,0.8\},
\text{PlotRange} \rightarrow \{0,1\}]

\textbf{Out[19]:= - SurfaceGraphics -}

The stability region for the fourth order classical Runge-Kutta formula is constructed by a contour plot which is shown in the form

\textbf{In[20]:= } \text{ContourPlot}[f, \{x,-4,1\}, \{y,-4,3.5\},
\text{ContourShading} \rightarrow \text{False},
\text{Contours} \rightarrow \{1,1\}]

\textbf{Out[20]:= - ContourGraphics -}
Following Gaylord et.al [1996], Mathematica is one of the software packages with a fully integrated technical computing environment, including a programming language suitable for individuals who are not full time programmers and it includes:

1) Built-in mathematical and graphical capabilities, that are both powerful and flexible.
2) A programming language that can be used to extend its capabilities virtually without limit. The language is interactive, has the capability to perform both numeric and symbolic manipulation, makes broad use of pattern matching, and supports the functional style of programming favored by many computer scientist.
3) Extensive on-line help facilities, including the Function Browser, is a new feature in Version 2.2 that makes it easy to learn about built-in functions and get their syntax correct.
4) The ability to connect Mathematica to other computing environments and other languages.
5) An interface that allows text and graphics to appear together in documents.

Finally, from the above discussion we conclude that symbolic computation is practically useful especially in the context of modelling and the solution of problems. For example, one of the reasons cited by Brown and Hearn [1978] is in the realm of partial differential equations, Cloutman and Fullerton [1977] have used symbolic multidimensional Taylor series expansions, computed by the ALTRAN system, to analyse the discretization and round-off errors of various numerical methods and also, more importantly, to eliminate inaccurate or unstable methods prior to coding and testing, and to develop methods in which the lowest order errors cancel each other out. Therefore, in the following chapters we will use this type of application to develop new linear and nonlinear methods for solving ordinary differential equations.
2.3 BASIC CONCEPTS IN PARALLEL PROCESSING

The process of parallelism actually exists and can be seen in action in our daily lives. For example, a bank or post office which has more than one customer counter provides a service concurrently in a faster and more efficient way. These objectives also gives the same application for the solution of a large problem on a computer and led to the construction of the parallel computer or supercomputer. Following Chandor et.al [1985], a parallel computer is defined as a computer which contains a unit giving it simultaneity of operation. While a supercomputer is the fastest computer at any point of time. The difference between a supercomputer and mainframe computer, is that the supercomputer is many times faster and gives greater performance.

Throughout the historical development of computers, faster and faster machines have been built, but today we have reached the limits of speed achievement. The fastest machines now can operate with clock times of about one to three nanoseconds, giving the order of 10^9 floating point operations per second (flops). Machines with high performance such as supercomputers usually are used for large scale problems, such as short-term weather forecasting, simulation to predict aerodynamics performance, image processing and artificial intelligence. Many of these applications involve the solution of very large sets of simultaneous equations by numerical methods.

The research into parallel computation has brought improvements for many years in the area of architecture, the development of memory-store electronic computer and algorithmic and programming systems. For example, ILLIAC IV was the first parallel computer built (Karplus [1989], Kuck [1968]) followed by STAR-100, Cray-1, Cray-2 and Cray-3. Further discussion about parallel computers can be found in many publications, e.g., Flynn [1972],

The current trend is to use parallel processing, that is, to put several machines to work on a single problem, dividing the solution process into many steps that can be performed simultaneously. Following Gerald and Wheatly [1994], not all problems permit such parallel operations but many important problems of Applied Mathematics can be so structured. For example, parallel processing can be applied to Romberg integration where each of the function evaluations can be done at the same time and the summations can be speeded up by using the fan algorithm. In this thesis, the extrapolation methods for solving ordinary differential equations using parallel processing will be discussed in chapter 5.

In this section, we present the fundamental concepts in parallel processing. These include data partitioning, function partitioning, execution time, speedup and efficiency. All these concepts will be discussed and apply in the ensuing chapters.

2.3.1 Data Partitioning and Function Partitioning

Data partitioning means the creation of identical processes which run concurrently with each operating on different data from those operated on by the others. The data is distributed or partitioned in several processes. Data partitioning is appropriate for applications that perform the same operations repeatedly on large collections of data. In case of programming, data partitioning is appropriate for applications that require loops to perform calculations on arrays or matrices: data partitioning is done by executing the loop iterations in parallel.

Function partitioning means the creation of different processes which the functions that the program has to perform are partitioned. In other words, function
partitioning involves creating multiple processes which then perform different operations on the same data set. Function partitioning is suitable for applications which are performed in different operations on the same data. In the case of programming, function partitioning is appropriate for applications that include subroutines or functions. Two basic techniques in function partitioning is the fork-join technique and the pipeline technique. An example of function partitioning use the m_fork( ) in order to create the fork-join technique flow in one of the outline parallel program is shown in Figure 2.2.

![Diagram of fork-join technique in function partitioning for parallel program](image)

**Figure 2.2**: Outline of fork-join technique in function partitioning for parallel program

### 2.3.2 Execution Time, Speedup and Efficiency

In parallel processing we will assume that each process has a machine to itself, so there is no contention amongst the processes. Applying more processes to any problem will allow the algorithm to be solved more quickly. Here, we can achieve this aim by defining several terms.
Definition 2.1

Execution time \( T(p,n,A) \) is the time needed by algorithm \( A \) to compute a problem of size \( n \) on \( p \) processes. We will omit the indices \( p \), \( n \) and \( A \) when it is clear what is meant. Execution time includes initialization and communication time, and is measured from the time the first process starts to the time the last one terminates.

Definition 2.2

Speedup \( S(p,n,A) \) is defined as

\[
S(p,n,A) = \frac{\text{time required by the best serial algorithm}}{T(p,n,A)}
\]

Values of \( S \) range from 0 to \( p \). In other words, the term speedup is used to describe the increased performance of a parallel system compared to a single processor. It is the ratio of the execution time for the original sequential process, to the time for the same job using parallel processing. In computing this speedup, we use the definition for the time for the optimum sequential procedure and for the best parallel algorithm.

Definition 2.3

Efficiency \( E(p,n) \) is defined as

\[
E(p,n,A) = \frac{S(p,n,A)}{p}
\]

Values of \( E \) range from 0 to 1. From this definition, the efficiency is based on how the speedup compares to the number of processors used. Theoretically, if we have \( n \) processors, we should be able to do the job \( n \) times as fast. The efficiency is less than 1.00 because some of the processors are idle.
CHAPTER 3

SURVEY OF SINGLE STEP METHODS TO SOLVE ORDINARY DIFFERENTIAL EQUATIONS
In this chapter, we describe the standard basic single step methods for solving ordinary differential equations. However, the linear multistep method will be briefly described at the end of this chapter. The reason is that two thirds of the material in this thesis is focussed on single step methods which form the basis of the research work accounted in Chapter 4, 5 and 6. While the remaining one-third of the material is on linear multistep methods and will be discussed in Chapter 7.

3.1 SINGLE STEP AND TAYLOR SERIES METHODS

3.1.1 Introduction

Single step methods for solving initial value problems in (1.2.9) require only a knowledge of the numerical solution \( y_n \) and the initial value \( y_0 \) in order to compute the next value \( y_{n+1} \). The well known single step method is the Runge Kutta method which will be described in Section 3.3. While the simplest single step method is based on using the Taylor series.

3.1.2 The Solution By Taylor Series

The solution of the initial value problem
\[
\begin{align*}
y' &= f(x, y) \\
y(x) &= y(a) = \alpha
\end{align*}
\]
(3.1.1)
(3.1.2)
in the neighbourhood of \( x = a \), may be expressed as the Taylor series
\[
y(x) = y(a) + (x - a)y'(a) + \frac{(x - a)^2}{2!} y''(a) \\
+ \frac{(x - a)^3}{3!} y'''(a) + \ldots + \frac{(x - a)^p}{p!} y^{(p)}(a) + \ldots
\]
(3.1.3)
which is differentiable at \( x = a \). The second and higher values of the successive derivatives in (3.1.3) may be obtained by repeated differentiating the differential equation using the chain rule given by
\[ y(a) = \alpha \]
\[ y'(a) = f(x, y) \]
\[ y''(a) = \frac{d}{dx} f(x, y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} \]
\[ = f_x + f_y \frac{dy}{dx} \]
\[ y'''(a) = f_{xx} + 2ff_{xy} + f^2f_{yy} + f_xf_y + ff_y^2 \]
\[ y^{(4)}(a) = f_{xxx} + 3ff_{xxy} + 3f_xf_{xy} + 5ff_{xyy} + 3ff_{yy} + 3f^2f_{yyy} \]
\[ + 4f^3f_{yy} + f_xf_{xx} + f_xf_{yy} + ff_y^3 + f^3f_{yyy} \]
\[ y^{(5)}(a) = f_{xxxx} + 4ff_{xxyy} + 6f_{xxyy} + 9ff_{xyyy} + 6f^2f_{xxyy} + 4f_{xx}f_{yy} \]
\[ + 8f^2f_{yy} + 12ff_{xxyy} + 7f_{xxyy} + 9ff_{xyyy} + 12f^3f_{yy} + 15f^2f_{yyyy} \]
\[ + 3f^4f_{yy} + 13ff_{xxyy} + 4ff_{xxyy} + 6f^2f_{xyy} + 4f^3f_{xyy} \]
\[ + 7f^3f_{xyy} + f_{xx}f_{xx} + f_{xx}f_{yy} + f_{xx}f_{yy} + ff_y^3 + f^4f_{xyy} \]
\[ y''(a) = \ldots \quad \text{etc.} \quad (3.1.4) \]

This method is not strictly numerical, but is important because several numerical methods are related to it. Basically, the Taylor series method produces an infinite series of powers of \( x \) which is near to the initial point and a good approximation to the analytic solution \( y(x) \). We develop the relation between \( y \) and \( x \) by finding the coefficients of the Taylor series in which we expand \( y \) about the point \( x = x_0 \):

\[ y(x) = y(x_0) + (x - x_0)y'(x_0) + \frac{(x - x_0)^2}{2!}y''(x_0) + \ldots \quad (3.1.5) \]

If \( h = x - x_0 \), equation (3.1.5) can be written as

\[ y(x) = y(x_0) + hy'(x_0) + \frac{h^2}{2!}y''(x_0) + \frac{h^3}{3!}y'''(x_0) + \ldots \quad (3.1.6) \]

where the first term is known as the initial condition. Equation (3.1.6) is actually the Maclaurin series because the expansion is about the point \( x = 0 \).

Consider the example problem

\[ y'(x, y) = 1 + x - y \quad , \quad y(0) = 1 \quad (3.1.7) \]
and the analytical solution is given by \( y(x) = x + \exp(-x) \).

The coefficient of the second term is obtained by substituting \( x=0, y=1 \) in the equation for the first derivative in (3.1.7) given by

\[
y'(x) = 1 + x - y \quad , \quad y'(0) = 1 - 0 - 1 = 0
\]

The second and higher order derivatives can be obtained by successively differentiating the equation of the first derivative and evaluated corresponding to \( x=0 \) to obtain the following coefficients:

\[
\begin{align*}
y''(x) &= 1 - y' \quad , \quad y''(0) = 1 - 0 = 1 \\
y'''(x) &= -y'' \quad , \quad y'''(0) = -1 \\
y''''(x) &= -y''' \quad , \quad y''''(0) = -(-1) = 1 \\
y''''''(x) &= -y'''' \quad , \quad y''''''(0) = -1 \\
&\vdots \\
&\text{etc}.
\end{align*}
\]

The Taylor series solution for \( y \) is obtained by letting \( h = x \) and substituting (3.1.7) into (3.1.6), to determine \( y \) as a function of \( x \) and can be written in the form

\[
y(x) = 1 + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{120} + \text{error}.
\]  

(3.1.8)

For example, if \( x = 0.1 \), we obtain

\[
y(0.1) = 1 + \frac{(0.1)^2}{2} - \frac{(0.1)^3}{6} + \frac{(0.1)^4}{24} + \ldots \\
= 1 + 0.005 - 1.6667 \times 10^{-2} + 4.1667 \times 10^{-3} + \\
= 1.0048375
\]

which agrees well with the analytical solution 1.004837418. However, to obtain the same accuracy as in analytical solutions, many more terms in (3.1.8) would be needed.
In Butcher [1987], many people have made important contributions to the use of Taylor series methods. For example, the program by Gibbons [1960] using a computer with the limited memory available at that time, used a recursive technique to generate the Taylor coefficients automatically. A similar approach using greater sophistication and more powerful computational tools was used by Barton, Willers and Zahar [1971]. The most interesting work uses interval arithmetic and supplies rigorous error bounds for the computed solution was introduced by Moore [1964].

### 3.1.3 Euler's Method

The Euler method for solving the initial value problem

\[ y' = f(x, y), \quad y(x_0) = y_0 \]

was described by Euler in 1768. These are the simplest of the single step methods which uses is the first two terms of the Taylor series. The solution is approximated locally by the tangent at the point \((x_0, y_0)\) and can be written in the form \( y_1 = y_0 + hf(x_0, y_0) \). While the initial condition \( y(x_0) = y_1 \) is used as the tangent through \((x_0, y_0)\) to approximate \( y_1 \). By repeating this procedure, the approximation to \( y(x_{n+1}) \) through \((x_n, y_n)\) and tangent passing through \((x_n, y_n)\) until it intersects the ordinate at \( x_{n+1} \) is given by

\[ y_{n+1} = y_n + hf(x_n, y_n) \]  \hspace{1cm} (3.1.9)

is called the Euler method, the first-order Taylor series method or the first-order Runge-Kutta method. This method is not particularly accurate and so are rarely used in practice but it provides a useful introduction to the other Runge-Kutta methods discussed in Section (3.3).
3.1.4 Higher Order Taylor Series Method

We now consider the second, third and fourth order Taylor series method written in the form

\[ y_{n+1} = y_n + h y' + \frac{h^2}{2} y'' \]  
\[ (3.1.10-i) \]

\[ y_{n+1} = y_n + h y' + \frac{h^2}{2} y'' + \frac{h^3}{6} y''' \]  
\[ (3.1.10-ii) \]

\[ y_{n+1} = y_n + h y' + \frac{h^2}{2} y'' + \frac{h^3}{6} y''' + \frac{h^4}{24} y^{(iv)} \]  
\[ (3.1.10-iii) \]

for solving the problem in equation (3.1.7) where

\[ y' = 1 + x - y \]  
\[ (3.1.11-i) \]

\[ y'' = 1 - y' \]  
\[ (3.1.11-ii) \]

\[ y''' = -y'' \]  
\[ (3.1.11-iii) \]

\[ y^{(iv)} = -y''' \]  
\[ (3.1.11-iv) \]

The comparison of the numerical results given by the Taylor series methods of order 1, 2, 3 and 4 at point \( x = 0.1 \) with \( h = 0.1 \) for solving differential equation in (3.1.7) are shown in Table 3.1.

Table 3.1: Numerical solution of \( y' = 1 + x - y, y(0) = 1 \) using Taylor series method at point \( x = 0.1 \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>First Order</th>
<th>Second Order</th>
<th>Third Order</th>
<th>Fourth Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.0000</td>
<td>1.0050</td>
<td>1.00483</td>
<td>1.00484</td>
</tr>
</tbody>
</table>

From figures in Table 3.1, clearly show that the accuracy of the methods are increased by the order of the method. The same characteristic also will be applied to develop the linear and nonlinear method for solving ordinary differential equation and will be shown in the following chapter.
The Taylor series method is easily applied to a higher order equation. For example, if the initial value problem is given by
\[ y'' + 101y' + 100y = 0 \quad , \quad y(0) = 1 , y'(0) = -1 \] (3.1.12)
then the derivative terms using Taylor series method as follows:

a) \( y(0) \) and \( y'(0) \) can be taken from the initial condition
b) \( y''(0) \) is obtained by substituting into the differential equation from \( y(0) \) and \( y'(0) \)
c) and \( y'''(0) , y''(0) , y'(0) , \ldots \) are obtained by differentiating the previous order equations.

3.1.5 The Existence and Uniqueness of Solutions

The solution of \( y' = f(x,y) \) are generally a family of curves and the initial condition \( y(x_0) = \alpha \) usually give a unique solution. For example, \( y' = 2y \) has the solution \( y(x) = c \exp(2x) \) and \( y(0) = 1 \) implies that \( c = 1 \) which gives the unique or exact solution as \( y(x) = \exp(2x) \). The initial value problem
\[ y' = f(x,y) \quad , \quad y(x_0) = \alpha \] (3.1.13)
has a unique solution on some interval \([a,b]\) if \( f(x,y) \) satisfies the conditions proved in the well known theorem by Henrici [1962].

Theorem 3.1:

If function \( f(x,y) \) be defined and continous for all points in the region
\[ D = \{(x,y) : a \leq x \leq b, -\infty < y < \infty \} \ , \]
where \( a \) and \( b \) are finite and there exists a constant \( L \) such that
\[ |f(x,y) - f(x,y')| \leq L|y - y'| \] (3.1.4)
for \( a \leq x \leq b \) and all \( y,y' \), then for any initial value \( \alpha \) there exist a unique solution of the initial value
problem \( y' = f(x, y) \), \( y(x_0) = \alpha \) and the constant \( L \) is called the Lipschitz constant and the condition in (3.1.14) is called the Lipschitz condition. This theorem applies to a system of equations by putting the vector sign as shown in equation (1.2.9).

3.2 ERRORS, CONVERGENCE, CONSISTENCY AND STABILITY

3.2.1 Introduction

One of the main reasons why the Taylor series method is seldom used for the solution of differential equations is because of the complexities of evaluating the higher order derivatives. For example, in the Euler method the approximation error only dominates the third term of the Taylor series. In this section, we now discuss the two sources of error in the numerical solution of differential equations which are the local truncation error (LTE) and the global truncation error (GTE).

3.2.2 Local Truncation Error and Global Truncation Error

The Taylor series method, as discussed in the previous section belongs to the class of single step methods. Since only the values \( x_n, y_n \) are required to calculate \( y_{n+1} \). A general explicit single step method is written as

\[
y_{n+1} = y_n + h\phi(x_n, y_n, h)
\]

where \( \phi(x_n, y_n, h) \) is called the increment function. For example, the first order Taylor series method has \( \phi(x_n, y_n, h) = f(x_n, y_n) \) while the fourth-order Taylor series method has \( \phi(x_n, y_n, h) = y' + \frac{h}{2} y'' + \frac{h^2}{6} y''' + \frac{h^3}{24} y^{(3)} \) where \( y', y'', y''' \) and \( y^{(3)} \) is obtained by using similar steps to equation (3.1.11). We now give the definition between the LTE and GTE as follows:
Definition 3.1:

The local truncation error (LTE) at the point \( x_{\ast i} \) is the difference between the computed value \( y_{\ast i} \) and the value at the point \( x_{\ast i} \) on the solution curve that goes through the point \((x_{\ast i}, y_{\ast i})\).

According to the above definition, the LTE for the Euler method may be written as

\[
LTE = \{y(x_{\ast i}) + hf(x_{\ast i}, y_{\ast i})\} - \left\{y(x_{\ast i}) + hy'(x_{\ast i}) + \frac{h^2}{2} y''(x_{\ast i}) + \frac{h^3}{6} y'''(x_{\ast i}) + \ldots \right\}
\]

\[
= -\frac{h^2}{2} y''(\xi) = O(h^2) \text{ as } h \to 0.
\]

where \( \xi \in [x_{\ast i}, x_{\ast i} + h] \) is obtained from the mean value theorem.

Definition 3.2: The Mean Value Theorem

If the function \( f(x) \) is continuous and differentiable in \([a, b]\), then there exist at least one point \( \xi \in [a, b] \) such that \( \Delta f = f(b) - f(a) = f'(\xi)(b - a) \).

Definition 3.3:

The global truncation error (GTE) at the point \( x_{\ast i} \) is defined as \( y_{\ast i} - y(x_{\ast i}) \) where \( y(x) \) denotes the solution of the given initial value problem.

Many texts, including Henrici [1962] and Gear [1971] show that the definition of GTE with the opposite sign does not effect any of the results. In Elden and Wittmeyer [1990], it is stated that generally, if the LTE of a numerical method is \( O(h^{n+1}) \) the GTE is \( O(h^n) \). The comparison of the LTE and GTE for linear and nonlinear methods will be discussed in the chapter 6.

3.2.3 Consistency, Stability and Convergence

Consistency, stability and convergence of a single step method of solving the initial value problem (3.1.13) which satisfies the condition of the uniqueness theorem of Section (3.1.4) is specified by the following definition:
Definition 3.4:

The single step method defined by equation (3.2.1) is said to be consistent with the initial value problem (3.1.13) if the increment function satisfies the following relation

\[ \phi(x_n, y_n, 0) = f(x_n, y_n) \]

Definition 3.5:

The single step method defined by equation (3.2.1) is said to be stable if for each differential equation satisfying a Lipschitz condition there exists a positive constant \( h_0 \) such that the difference between two different numerical solutions \( y_n \) and \( y_n' \), each satisfying the differential equation is such that

\[ \| y_n - y_n' \| \leq k \| \alpha - \alpha' \| \]

for all \( 0 \leq h \leq h_0 \) where \( y_n(a) = \alpha \) and \( y_n'(a) = \alpha' \) and \( k \) is the Lipschitz constant.

Definition 3.6:

A single step method for solving the initial value problem is said to be convergent if the numerical solution \( y_n \) approaches the analytic solution \( y(x) \) at any fixed \( x_n \in [a, b] \) as the step length \( h \) tends to zero and \( y_n \) tend to \( \alpha \) and may be written as

\[ \lim_{h \to 0} y_n = y(x) \]

\[ x_n \to x \]

Noye [1990] proved that a single step method is convergent if GTE or \( e_n \to 0 \) for all fixed \( nh = x_n - a \) as \( h \to 0 \). The theory of convergence for a single step method is also discussed in many texts e.g., Gear [1971] and Henrici [1962]. For example, Henrici [1962] proved that if the increment function \( \phi = \phi(x_n, y_n, h) \) is continuous in the interval \([a, b]\) with respect to \( x_n, y_n \) and \( h \) and satisfies the Lipschitz condition with respect to \( y_n \) in the region \( R \), then the single step method is convergent if and only if it is consistent.
3.3 RUNGE KUTTA METHOD

3.3.1 Introduction

The idea of extending the Euler method by allowing a multiplicity of evaluations of the function $f$ within each step was originally introduced by Runge (1856 - 1927) in 1895. Further contributions was made by Heun in 1900 and was modified by Kutta (1867 - 1944) in 1901. The main characteristic of the Runge-Kutta method

a) It is one step method, to obtain $y_{n+1}$, we only use the existence of the previous point, that is the value of the function at point $(x_n, y_n)$

b) The accuracy of the method is approximately similar to the Taylor Series method

c) The important criteria in the Runge-Kutta method is the function evaluation of $f(x,y)$ only and does not involve the higher derivative.

In the Runge-Kutta method, $f(x,y)$ is evaluated at one point in the interval which we want to find the derivative. The number of function evaluations $f(x,y)$ depends on the order of the Runge-Kutta method to be used. The general $s$-stage Runge-Kutta method for the problem

$$ y' = f(x,y) \quad , \quad y(a) = \eta \quad , \quad f: \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n \quad (3.3.1) $$

is defined by

$$ y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i $$

where

$$ k_i = f \left( x_n + c_i h, y_n + h \sum_{j=1}^{s} a_{ij} k_j \right) \quad , \quad i = 1, 2, \ldots, s \quad (3.3.2) $$

and

$$ c_i = \sum_{j=1}^{s} a_{ij} \quad , \quad i = 1, 2, \ldots, s. $$

The coefficients in equation (3.3.2) can be written in the form, known as a Butcher array:
or can be specified by its Butcher array as

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

or can be specified by its Butcher array as

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

with \( s \)-dimensional vectors \( c \) and \( b \) and the \( s \times s \) matrix \( A \) denoted by

\[
c = [c_1, c_2, \ldots, c_s]^T, \quad b = [b_1, b_2, b_3, \ldots, b_s]^T \quad \text{and} \quad A = [a_{ij}].
\]

The general \( R \)-stage Runge-Kutta method for solving \( y' = f(x, y) \) is defined by

\[
y_{r+1} = y_r + h \sum_{i=1}^{s} w_i k_i, \quad (3.3.3)
\]

\[
k_i = f(x, y)
\]

\[
k_i = f\left(x + h b_i, y + h \sum_{j=1}^{R} a_{ij} k_j\right), \quad i = 2, 3, \ldots, R, \quad (3.3.4)
\]

\[
b_i = \sum_{j=1}^{n} a_{ij}, \quad i = 2, 3, \ldots, R.
\]

Equation (3.3.3) is a basic formula which is widely used by most earlier authors to develop or to prove the classical Runge-Kutta method. In the following chapter, the above definition will be used in the development of the nonlinear Runge-Kutta formulae based on the variety of mathematical means. This type of nonlinear Runge-Kutta formulae have also been discussed by many authors, i.e., Evans and Yaakub [1993], Evans and Yaakub [1995], Yaakub and Evans [1995], Evans and Yaacob [1993] and Wazwaz [1994].

In equation (3.3.4), the sum of the weighted values for \( x \) and \( y \) are equal. From these reason, a great deal of tedious algebra manipulation is saved in deriving Runge-Kutta methods of high order, the general R-stage Runge-
Kutta method also can be defined only in terms of $y$ by the following relations,

$$y_{n+1} = y_n + h \sum_{i=1}^{R} w_i k_i \tag{3.3.5-i}$$

$$k_i = f \left( y_n + h \sum_{j=1}^{i} a_{ij} k_j \right) \tag{3.3.5-ii}$$

The relations in (3.3.5) are used to define the approximation $y_{n+1}$ to $y(x_n)$ in terms of an approximation $y(x_n)$ and was denoted by $y_n$ in (3.3.5) to be used in all the function evaluations in the following chapter. While the coefficient $w_i, a_{ij}, i,j = 1,2,...,R$ in the R-stage Runge-Kutta method will be determined by solving the equations of conditions.

By assuming that $f(x,y)$ is sufficiently smooth, Lambert [1973] introduced the shortened notation

$$f := f(x,y), \quad f_x := \frac{\partial f(x,y)}{\partial x}, \quad f_{xx} := \frac{\partial^2 f(x,y)}{\partial x^2}$$

all evaluated at the point $(x_n,y(x_n))$. Then, on expanding $y(x_n)$ about $x_n$ as a Taylor series in the form

$$y(x_n) = y(x_n) + hy'(x_n) + \frac{h^2}{2} y''(x_n) + \frac{h^3}{6} y'''(x_n) + O(h^4) \tag{3.3.6}$$

and by taking $f$ as a function of $y$ and independent of $x$, the repeated differentiating of the differential equation can be obtained in the form,

$$y'(x_n) = f$$

$$y''(x_n) = f f_x$$

$$y'''(x_n) = f^2 f_{xx} + f f_{xx}$$

$$y''''(x_n) = f f_{xx} + 4 f^2 f_{xx} f_x + f^3 f_{xxx} + f^2 f_{xxx}$$

$$y''''''(x_n) = f f_{xx} + 11 f^2 f_{xx} f_{xx} + 4 f^3 f_{xxx} f_x + f^4 f_{xxxx} + f^3 f_{xxxx}$$

$$y'''''''(x_n) = f f_{xx} + 26 f^2 f_{xx} f_{xx} f_x + 34 f^3 f_{xxx} f_{xx} + 32 f^4 f_{xxxx} f_x + 15 f^3 f_{xxxx} f_{xx} + 11 f^4 f_{xxxx} f_{xxx} + f^5 f_{xxxxx} \tag{3.3.7}$$
By substituting (3.3.7) into (3.3.6) gives the Taylor series as

\[ y(x,\ldots, h) = y + hf + \frac{h^2}{2} f_x + \frac{h^3}{6} (f_{xx} + f^2 f_y) + \frac{h^4}{24} (f_{xxx} + 4 f^2 f_x f_y + f^3 f_{yy}) \]

\[ + \frac{h^5}{120} (f_{xxxx} + 11 f^2 f_x f_{yy} + 4 f^3 f_{yy} + 7 f^3 f_x f_{yy} + f^4 f_{yy}) \]

\[ + \frac{h^6}{720} (f_{xxxxx} + 26 f^2 f_x f_{yy} + 34 f^3 f_y f_{yy} + 32 f^3 f_x f_{yy} + 15 f^4 f_{yy}) \]

\[ + 11 f^4 f_x f_{yy} + f^5 f_{yyy}) + O(h^7) \] (3.3.8)

and can be applied to confirm the order and the local truncation error (LTE) of the methods.

The expansion of \( k_i, i=1,2,3,4 \) in terms of the functional derivatives and the parameters \( a_i, i=1,2,3,4,5,6 \) from equation (3.3.5-ii) is obtained as follows:

\[ k_1 = f(x, y) \]
\[ k_2 = f(x + \beta_1 h, y + ha_k) \]
\[ k_3 = f(x, y, h + ha_k + ha_h) \]
\[ k_4 = f(x + \beta_3 h, y + ha_k + ha_h + ha_h) \]

\[ \vdots \]

\[ \text{etc.} \]

where \( \beta_i = a_i \)
\[ \beta_2 = a_1 + a \]
\[ \beta_3 = a_4 + a_5 + a_6 \] (3.3.10)

\[ \vdots \]

\[ \text{etc.} \]

where the values of \( \beta_i, \beta_i, \beta \ldots \) are obtained after the values for \( a_i, a_1, a_5, a_6 \ldots \) have been determined. For simplification of the accompanying algebra, it is sufficient only to consider \( f \) as a function of \( y \) only rather than \( x \).

By using the expression of the Taylor series of \( k_i, i=1,2,3,4 \) in terms of the functional derivatives as follows:
\[ k_i = f(y_i) = f \]  
\[ k_s = f(y_s + h \alpha k_i) \]  
\[ = f + h \alpha_f + \frac{h^2}{2} a_i^2 f^2 f_m + \frac{h^3}{6} a_i^3 f^3 f_m + \frac{h^4}{24} a_i^4 f^4 f_{mm} + \frac{h^5}{120} a_i^5 f^5 f_{mmm} + \ldots \]  
\[ k_s = f(y_s + h \alpha k_i + h \alpha k_s) \]  
\[ = f + h(a_i + a_s)ff + h^2 \left[ a_i a_s ff + \frac{1}{2} (a_i + a_s)^2 f^2 f_m \right] \]  
\[ + h^3 \left[ \frac{1}{2} a_i^2 a_s f^2 f_m + a_i a_s (a_i + a_s) f^2 f_{m} + \frac{1}{6} (a_i + a_s)^3 f^3 f_{mm} \right] \]  
\[ + h^4 \left[ \frac{1}{6} a_i^3 a_s f^3 f_{mm} + \frac{1}{2} a_i^2 f^2 f_m + \frac{1}{2} (a_i + a_s) a_i a_s f^3 f_m \right] \]  
\[ + \frac{1}{2} (a_i + a_s)^2 a_i a_s f^2 f_{m} + \frac{1}{24} (a_i + a_s)^4 f^4 f_{mm} \]  
\[ k_4 = f(y_s + h \alpha k_i + h \alpha k_s + h \alpha k_4) \]  
\[ = f + h(a_i + a_s + a_s)ff + h^2 \left[ (a_i a_s + (a_i + a_s)a_s) ff \right] \]  
\[ + \frac{1}{2} (a_i + a_s + a_s)^2 f^2 f_m \right] + h^3 \left[ \left( \frac{1}{2} a_i^2 a_s + \frac{1}{2} (a_i + a_s)^2 a_s \right) \right. \]  
\[ + (a_i + a_s + a_s)(a_i a_s + (a_i + a_s)a_s) f^2 f_m \]  
\[ + a_i a_s a_s ff + \frac{1}{6} (a_i + a_s + a_s)^3 f^3 f_{mm} \]  
\[ + h^4 \left[ \frac{1}{6} \left( a_i^2 a_s + (a_i + a_s)^2 + 3(a_i + a_s + a_s)^2 (a_i a_s + (a_i + a_s)a_s) \right) f^3 f_{mm} \right. \]  
\[ + \left\{ \left( \frac{1}{2} a_i^2 a_s a_s + a_i a_s (a_i + a_s) \right) + \frac{1}{2} (a_i a_s + (a_i + a_s)a_s)^2 \right. \]  
\[ + a_i a_i a_s (a_i + a_s + a_s) \} f^2 f^2 f_m \]  
\[ + \frac{1}{2} \left\{ (a_i + a_s + a_s) (a_i a_s + (a_i + a_s)^2 a_s) \right\} f^3 f_m^2 \]  
\[ + \frac{1}{24} (a_i + a_s + a_s)^4 f^4 f_{mm} \]  
\[ (3.3.11-iv) \]
3.3.2 Second Order Runge-Kutta Method

The 2-stage formulae of order 2 where \( R = p = 2 \) can be derived from equation (3.3.5-ii), where we use the variable \( a_1 \) to represent \( a_{11} \) and then determine the parameters \( w_1 \) and \( w_2 \). By putting \( R = 2 \) in (3.3.5-i) we have

\[
y_{n,1} = y_n + h(y_n) k_1 + w_2 k_2
\] (3.3.12)

where
\[
k_1 = f(y_n)
\]
\[
k_2 = f(y_n + h a_1 k_1).
\]

By using the expression for the Taylor series for \( k_1 \) and \( k_2 \) in (3.3.11-i) and (3.3.11-ii) and substituting in equation (3.3.12), we obtain

\[
y_{n,1} = y_n + h(y_n) f + h^2(y_n) w_2 + O(h^3)
\] (3.3.13)

By comparing the equation (3.3.13) with the equation in (3.3.8), we obtain two equations of conditions with 3 parameters to be determined:

\[
h f: \quad 1 - w_1 - w_2 = 0 \quad \text{(3.3.14-i)}
\]
\[
h f^2: \quad 1 - 2 a_1 w_1 = 0 \quad \text{(3.3.14-ii)}
\]

Solving the equations (3.3.14-i)-(3.3.14-ii) using Mathematica, we immediately obtain

\[
w_1 = 1 - w_2 \quad \text{and} \quad a_1 = \frac{1}{2 w_2}
\]

and by taking \( w_2 = \frac{1}{2} \), equation (3.3.12) can be written in the form

\[
y_{n,1} = y_n + \frac{h}{2} (k_1 + k_2)
\] (3.3.15)

where
\[
k_1 = f(y_n)
\]
\[
k_2 = f(y_n + h k_1)
\]
which is known as the second order Runge-Kutta method or improved Euler method and can be written in Butcher array form as

\[
\begin{array}{c|ccc}
0 & 0 & 0 \\
1 & 1 & 0 \\
\hline
1 & 1 & 0
\end{array}
\]

### 3.3.3 Third Order Runge-Kutta Method

Now, we extend the idea further by deriving the 3-stage formulae of order three by putting \( R=3 \) in equation (3.3.5-i) for the form

\[
y_{n+1} = y_n + h(w_1 k_1 + w_2 k_2 + w_3 k_3)
\]  

(3.3.16)

where

\[
k_1 = f(y_n),
\]

\[
k_2 = f(y_n + ha_1 k_1),
\]

\[
k_3 = f(y_n + ha_2 k_1 + ha_3 k_2)
\]

By the same procedure using the expressions for the Taylor series for \( k_1, k_2, \) and \( k_3 \) in (3.3.11-i)-(3.3.11-iii) and substituting into equation (3.3.16), we have

\[
y_{n+1} = y_n + h(w_1 + w_2 + w_3) f + h^3 (a_1 w_2 + a_2 w_3 + a_3 w_4) f f_f f
\]

\[
+ \frac{h^3}{2} [(2a_1 a_2 w_5) f f_f f_f + (a_1^2 w_2 + a_2^2 w_3 + 2a_1 a_2 w_5 + a_3^2 w_4) f^2 f_f] + O(h^4)
\]

(3.3.17)

By comparing the equation (3.3.17) with equation (3.3.8), we obtain 6 parameters with 4 equations of conditions:

\[
hf : 1 - w_1 - w_2 - w_3 = 0 \quad (3.3.18-i)
\]

\[
h^2 f f_f : 1 - 2a_1 w_2 - 2a_2 w_3 - 2a_3 w_4 = 0 \quad (3.3.18-ii)
\]

\[
h^2 f f_f^2 : 1 - 6a_1 a_2 w_5 = 0 \quad (3.3.18-iii)
\]

\[
h^3 f f_{f_f} : 1 - 3a_1^2 w_2 - 3a_2^2 w_3 - 6a_1 a_2 w_5 - 3a_3^2 w_4 = 0 \quad (3.3.18-iv)
\]
By taking \( a_i = \frac{1}{2} \) and \( a_i + a_j = 1 \) then using Mathematica to solve equations (3.3.18-i)-(3.3.18-iv), we obtain
\[
\begin{align*}
  w_1 = \frac{1}{6}, w_2 = \frac{2}{3}, w_3 = \frac{1}{6}, a_i = \frac{1}{2}, a_j = -1, a_j = 2 \ldots
\end{align*}
\] (3.3.19)

By substituting (3.3.19) into (3.3.16), the equation (3.3.16) now can be written in the form
\[
y_{n+1} = y_n + \frac{h}{6} (k_1 + 4k_2 + k_3)
\] (3.3.20)

where \( k_1 = f(x_n, y_n) \)
\[
k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_1\right)
\]
\[
k_3 = f\left(x_n + h, y_n - hk_1 + 2hk_2\right)
\]

which achieves third order accuracy and is known as the Kutta third order method. The Butcher array form for this method can be shown as
\[
\begin{array}{c|cccc}
0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & 0 \\
1 & -1 & 2 & 0 \\
\hline
\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6}
\end{array}
\]

3.3.4 Higher Order Runge-Kutta Methods

The derivation of fourth order Runge-Kutta methods involves tedious algebra manipulation. However, by extending the work using \( R = 4 \) in (3.1.5-i), we obtain
\[
y_{n+1} = y_n + h \left[ w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 \right]
\] (3.3.21)

where
\[
k_1 = f(y_n)
\]
\[
k_2 = f(y_n + ha, k_1)
\]
\[
k_3 = f(y_n + ha, k_1 + ha, k_2)
\]
\[
k_4 = f(y_n + ha, k_1 + ha, k_2 + ha, k_3)
\]
By substituting $k_i=1,2,3,4$ from equations (3.3.11-i)-(3.3.11-iv) into equation (3.3.21), the equation (3.3.21) now can be written in the form, 

$$y_{ni} = y_n + h(w_1 + w_2 + w_3 + w_4)f$$

$$+ h^3(a_iw_2 + a_2w_3 + a_3w_4 + a_4w_5 + a_5w_6)ff$$

$$+ h^4[(a_i a_iw_1 + a_2a_iw_2 + a_3a_iw_3 + a_4a_iw_4)ff]$$

$$+ \frac{1}{2}(a_i w_2 + a_i w_3 + a_i w_4 + a_i w_5 + a_i w_6 + 2a_i a_iw_4)$$

$$+ a_i w_4 + 2a_i a_iw_4 + 2a_i a_iw_4 + a_i a_iw_4 + a_i a_iw_4 + a_i a_iw_4 + a_i a_iw_4]$$

By comparing the equation (3.3.22) with the Taylor series expansion in (3.3.8), we obtain the following seven equations of conditions with ten parameters as

$$hf = 1 - w_1 - w_2 - w_3 - w_4 = 0$$

$$h^3ff = 1 - 2a_iw_2 - 2a_2w_3 - 2a_3w_4 - 2a_4w_5 - 2a_5w_6 = 0$$

$$h^3f^2f = 1 - 6a_2a_2w_3 - 6a_2a_3w_4 - 6a_2a_4w_5 - 6a_2a_5w_6 = 0$$

$$h^3f^2f = 1 - 3a_iw_2 - 3a_iw_3 - 6a_2a_2w_4 - 3a_iw_4 - 3a_iw_5 = 0$$

$$h^4f^2f = 1 - 24a_i a_iw_4 = 0$$

$$h^4f^2f = 1 - 3a_i a_iw_3 - 6a_2a_3w_4 - 6a_i a_iw_5 - 3a_i a_iw_6 = 0$$

By comparing the equation (3.3.22) with the Taylor series expansion in (3.3.8), we obtain the following seven equations of conditions with ten parameters as

$$hf : 1 - w_1 - w_2 - w_3 - w_4 = 0$$

$$h^3ff : 1 - 2a_iw_2 - 2a_2w_3 - 2a_3w_4 - 2a_4w_5 - 2a_5w_6 = 0$$

$$h^3f^2f : 1 - 6a_2a_2w_3 - 6a_2a_3w_4 - 6a_2a_4w_5 - 6a_2a_5w_6 = 0$$

$$h^3f^2f : 1 - 3a_iw_2 - 3a_iw_3 - 6a_2a_2w_4 - 3a_iw_4 - 3a_iw_5 = 0$$

$$h^4f^2f : 1 - 24a_i a_iw_4 = 0$$

$$h^4f^2f : 1 - 3a_i a_iw_3 - 6a_2a_3w_4 - 6a_i a_iw_5 - 3a_i a_iw_6 = 0$$
This is an underdetermined system of nonlinear equations and in general more than one solution can be obtained. For solving this system of nonlinear equations, we introduce three more additional equations by fixing the values of \( \beta_1, \beta_2, \) and \( \beta_3, \) where

\[
\beta_1 = a_1 = \frac{1}{2} \quad (3.3.23-viii)
\]

\[
\beta_2 = a_2 + a_3 = \frac{1}{2} \quad (3.3.23-ix)
\]

\[
\beta_3 = a_4 + a_5 + a_6 = 1 \quad (3.3.23-x)
\]

to obtain a set of solutions. By solving this system of nonlinear equations using Mathematica, we immediately obtain

\[
w_1 = \frac{1}{6}, \quad w_2 = \frac{4 - \frac{a}{6}}{6}, \quad w_3 = \frac{1}{6a}, \quad w_4 = \frac{1}{6}
\]

\[
a_1 = \frac{1}{2}, \quad a_2 = \frac{1 - 2a}{2}, \quad a_4 = 0, \quad a_5 = \frac{2 - \frac{a}{6}}{2}, \quad a_6 = \frac{1}{2a}
\]

By taking \( a_3 = \frac{1}{2} \), the ten parameters from the system of equations can be written as

\[
w_1 = \frac{1}{6}, \quad w_2 = \frac{1}{3}, \quad w_3 = \frac{1}{3}, \quad w_4 = \frac{1}{6}, \quad a_1 = \frac{1}{2}
\]

\[
a_2 = 0, \quad a_3 = \frac{1}{2}, \quad a_4 = 0, \quad a_5 = 0, \quad a_6 = 1 \quad (3.3.24)
\]

By substituting the parameters \( w_i, 1 \leq i \leq 4 \) and \( a_i, 1 \leq i \leq 6 \) into equation (3.3.21), we obtain

\[
y_{m+1} = y_n + \frac{h}{6}[k_1 + 2k_2 + 2k_3 + k_4] \quad (3.3.25)
\]

where \( k_i = f(x_n, y_n) \)
which is the most popular of all the Runge-Kutta methods and is referred to as the classical Runge-Kutta method. Another technique to deriving the fourth order standard method which does not involve tedious algebra manipulation will be discussed in the chapter 8. In the other alternative schemes, Ralston [1965] shows that to obtain a fourth-order 4-stage method, 13 equations in 11 unknown must be satisfied.

Another fourth order Runge-Kutta method is due to Gill [1951]. The scheme, which referred to as the Runge-Kutta-Gill method, is

\[ y_{n+1} = y_n + \frac{h}{6} \left[ k_1 + 2(1 - \frac{1}{3})k_2 + 2(1 + \frac{1}{3})k_3 + k_4 \right] \] (3.3.26)

where

\[ k_1 = f(x_n, y_n) \]
\[ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1) \]
\[ k_3 = f(x_n + \frac{h}{2}, y_n + \left(1 - \frac{1}{\sqrt{2}}\right)hk_1 + \left(1 + \frac{1}{\sqrt{2}}\right)hk_2) \]
\[ k_4 = f(x_n + h, y_n - \frac{1}{\sqrt{2}}hk_2 + \left(1 + \frac{1}{\sqrt{2}}\right)hk_3) \]

The study of higher order Runge-Kutta methods involves extremely complicated algebra. In Butcher [1987] and Lambert [1973] & [1991], Butcher proves the non-existence of a five-stage method of order five. However, the first example is the fifth-order Kutta-Nystrom six-stage method given by

\[ y_{n+1} = y_n + \frac{h}{192} \left[ 23k_1 + 125k_2 - 81k_3 + 125k_4 \right] \] (3.3.27)

where \[ k_1 = f(x_n, y_n) \]
\[ k_2 = f\left(x_1 + \frac{h}{3}, y_1 + \frac{1}{3} h k_1\right) \]
\[ k_3 = f\left(x_1 + \frac{2}{5} h, y_1 + \frac{1}{25} h(4k_1 + 6k_2)\right) \]
\[ k_4 = f\left(x_1 + \frac{1}{4} h, y_1 + \frac{1}{4} h(k_1 - 12k_2 + 15k_3)\right) \]
\[ k_5 = f\left(x_1 + \frac{2}{3} h, y_1 + \frac{1}{8} h(6k_1 + 90k_2 - 50k_3 + 8k_4)\right) \]
\[ k_6 = f\left(x_1 + \frac{4}{5} h, y_1 + \frac{1}{75} h(6k_1 + 36k_2 + 10k_3 + 8k_4)\right) \]

While Huta (see in Lambert [1973]) derived the following sixth-order eight-stage method in the form

\[ y_{n+1} = y_n + \frac{h}{840} \left[ 41k_1 + 216k_2 + 27k_3 + 272k_4 + 27k_5 + 216k_6 + 41k_7 \right] \quad (3.3.28) \]

where \( k_1 = f(x_n, y_n) \)
\[ k_2 = f\left(x_n + \frac{h}{9}, y_n + \frac{1}{9} h k_1\right) \]
\[ k_3 = f\left(x_n + \frac{h}{6}, y_n + \frac{1}{24} h(k_1 + 3k_2)\right) \]
\[ k_4 = f\left(x_n + \frac{h}{3}, y_n + \frac{1}{6} h(k_1 - 3k_2 + 4k_3)\right) \]
\[ k_5 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{8} h(-5k_1 + 27k_2 - 24k_3 + 8k_4)\right) \]
\[ k_6 = f\left(x_n + \frac{2}{3} h, y_n + \frac{1}{9} h(221k_1 - 981k_2 + 867k_3 - 102k_4 + k_5)\right) \]
\[ k_7 = f\left(x_n + \frac{5}{6} h, y_n + \frac{1}{48} h(-183k_1 + 678k_2 - 472k_3 - 66k_4 + 80k_5 + 3k_6)\right) \]
\[ k_8 = f\left(x_n + h, y_n + \frac{1}{82} h(716k_1 - 2079k_2 + 1002k_3 + 834k_4 - 454k_5 - 9k_6 + 72k_7)\right) \]

The fifth-order with five-stage Runge-Kutta method often referred to as the weighted Runge-Kutta method is
an interesting topics to study and will be discussed in chapter 8.

3.3.5 Stability Properties Of Runge-Kutta Methods

The purpose of stability analysis in numerical analysis is to study such qualitatives properties as boundedness and convergence to zero of the numerical solutions when these properties are possessed by the exact solution. In other words, stability analysis is studying the growth of numerical errors in computed solution to a differential equation.

We consider the standard test problem differential equation

\[ y' = Ay, \quad A \in \mathbb{C} \text{ and } \Re(\lambda) < 0. \]  

(3.3.29)

By applying the general Runge-Kutta method (3.3.5-i)-(3.3.5-ii) to (3.3.29) with \( h \lambda = z \), we obtain a one step difference equation of the form

\[ y_{n+1} = Q(z)y_n \]  

(3.3.30)

with \( Q(z) \) as the stability function of the method if it is shown clearly that \( y_n \to 0 \) as \( n \to \infty \) if and only if

\[ |Q(z)| < 1 \]  

(3.3.31)

then the method is absolutely stable. By substituting (3.3.29) into (3.3.11-i)-(3.3.11-iv), we obtain

\[ k_1 = f(y_n) = \lambda y_n \]  

(3.3.32-i)

\[ k_2 = f(y_n + a_1 h k_1) = \lambda (1 + a_1 z) y_n \]  

(3.3.32-ii)

\[ k_3 = f(y_n + a_2 h k_1 + a_3 h k_2) = \lambda (1 + (a_2 + a_3) z + a_2 a_3 z^2) y_n \]  

(3.3.32-iii)

\[ k_4 = f(y_n + a_4 h k_1 + a_5 h k_2 + a_6 h k_3) = \lambda \{1 + (a_4 + a_5 + a_6) z + za_5 ((a_2 + a_3) z + a_2 a_3 z^2)\} y_n \]  

(3.3.32-iv)

Substituting (3.3.32-i)-(3.3.32-iv) into equation (3.3.21), we obtain
\[ y_{*1} = \left[ 1 + z \left\{ w_1 + w_2 (1 + a_5 z) + w_3 (1 + (a_4 + a_5) z + a_6 a_5 z^2) \\
+ w_4 (1 + (a_4 + a_5 + a_6) z + a_6 a_5 z^2 + za_6 (1 + (a_4 + a_5) z + a_6 a_5 z^2)) \right\} \right] y_* \]

\[ \ldots \quad (3.3.33) \]

which is similar to the form in equation (3.3.30).

The general stability function, from repeatedly differentiating equation (3.3.29) for the R-stage explicit Runge-Kutta method order \( p \) can be written as

\[ y_{*1} = \left[ 1 + z \frac{z^2}{2!} + \ldots + \frac{z^p}{p!} \right] y_* + O(z''') \]

or

\[ Q(z) = 1 + z \frac{z^2}{2!} + \ldots + \frac{z^p}{p!} + O(z''') \quad (3.3.34) \]

By substituting \( k_i, 1 \leq i \leq 4 \) into the explicit Runge-Kutta method of order one to four then the stability function can be written as follows:

a) First order Runge-Kutta method or Euler method

\[ Q(z) = 1 + z + O(z^2) \quad (3.3.35-i) \]

b) Second order Runge-Kutta or improved Euler method

\[ Q(z) = 1 + z + \frac{z^2}{2!} + O(z^3) \quad (3.3.35-ii) \]

c) Third order Runge-Kutta or Kutta method

\[ Q(z) = 1 + z + \frac{z^2}{2!} \frac{z^3}{3!} + O(z^4) \quad (3.3.35-iii) \]

d) Fourth order Runge-Kutta or classical method

\[ Q(z) = 1 + z + \frac{z^2}{2!} \frac{z^3}{3!} \frac{z^4}{4!} + O(z^5) \quad (3.3.35-iv) \]

and the stability region for all stability functions in (3.3.35-i)-(3.3.35-iv) are shown in Figure 3.1.
Figure 3.1: Stability region for the Runge-Kutta method for \( R = 1, 2, 3, 4 \)

3.4 LINEAR MULTISTEP METHOD AND EXTRAPOLATION METHOD

3.4.1 Introduction

A general linear multistep method (LMM) of \( k \)-step is defined by the equation

\[
\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}
\]

where \( \alpha_j \) and \( \beta_j \) are constants; \( f_{n+j} = f(x_{n+j}, y_{n+j}) \) with the constraints \( \alpha_0 \neq 0 \) and that not both \( \alpha_0 \) and \( \beta_0 \) are zero. Equation (3.4.1) can be arranged to give

\[
y_{n+k} = h \beta_k f(x_{n+k}, y_{n+k}) + \sum_{j=0}^{k-1} (h \beta_j f_{n+j} - \alpha_j y_{n+j})
\]

If \( \beta_k = 0 \) then the RHS of (3.4.2) is known and the method is explicit but if \( \beta_k \neq 0 \), the method is implicit.

With the LMM in (3.4.1), (as in Lambert [1973]) we associate the linear difference operator \( \mathcal{L} \) defined by

\[
\mathcal{L}[y(x); h] = \sum_{j=0}^{k} \left[ \alpha_j y(x + jh) - h \beta_j y'(x + jh) \right]
\]

where \( y(x) \) is an arbitrary function and continuously differentiable on \( [a, b] \).
By expanding the Taylor series for the test function \(y(x + jh)\) at \(x\) gives

\[
y(x + jh) = y(x) + \sum_{i=1}^{\infty} \frac{(jh)^i}{i!} y^{(i)}(x)
\]

and the expansion of \(y'(x + jh)\) at \(x\) by Taylor series gives

\[
y'(x + jh) = y'(x) + \sum_{i=1}^{\infty} \frac{(jh)^i}{i!} y^{(i+1)}(x).
\]

By substituting equations (3.4.4) and (3.4.5) into equation (3.4.3) and collecting terms in (3.4.3) gives

\[
\ell[y(x);h] = C_0 y(x) + C_1 h y'(x) + \ldots + C_q h^q y^{(q)}(x) + \ldots,
\]

(3.4.6)

where \(C_i, i = 0, 1, 2, \ldots\) are constants.

Definition 3.7:
The difference operator \(\ell\) in (3.4.3) and the associated LMM given by equation (3.4.1) are said to be of order \(p\) if in (3.4.6), \(C_0 = C_1 = \ldots = C_p = 0, C_{p+1} \neq 0\).

The constant \(C_q\) in (3.4.6) in terms of the coefficient \(\alpha_j\) and \(\beta_j\) are given as follows:

\[
C_0 = \sum_{j=0}^{\infty} \alpha_j,
\]

\[
C_1 = \sum_{j=0}^{\infty} j\alpha_j - \sum_{j=0}^{\infty} \beta_j,
\]

\[
\vdots
\]

\[
C_q = \frac{1}{q!} \sum_{j=0}^{\infty} j^q \alpha_j - \frac{1}{(q-1)!} \sum_{j=0}^{\infty} \beta_j, \quad q = 2, 3, \ldots
\]

(3.4.7)

The parameters \(\alpha_j, \beta_j, j = 0, 1, \ldots, k\) can be obtained from equation (3.4.7).

Definition 3.8:
The LMM in (3.4.1) is said to be consistent if it has order \(p \geq 1\) and from (3.4.7), the LMM in (3.4.2) is consistent if and only if

\[
\sum_{j=0}^{\infty} \alpha_j = 0
\]
and $\sum_{j=0}^{k} \alpha_j = \sum_{j=0}^{k} \beta_j$ \hspace{1cm} (3.4.8)

The first characteristic polynomials of the LMM (3.4.1) is defined as

$$p(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j$$ \hspace{1cm} (3.4.9)

and the second characteristic polynomials of the LMM (3.4.1) is defined by

$$\sigma(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j$$ \hspace{1cm} (3.4.10)

It follows from (3.4.8) that the LMM is consistent if and only if

$$\tau(x) = h(p'(1) - \sigma(1)) = 0$$ \hspace{1cm} (3.4.11)

Thus, for a consistent method, the first characteristic polynomial $p(\zeta)$ always has a root at +1 and will be called the principal root with label $\zeta_1$. While the remaining roots, $\zeta_s, s = 2, 3, ..., k$ are spurious roots when $k > 1$.

Using the general linear k-step method defined by (3.4.1) to solve the differential equation $y' = \lambda y, y(0) = 0, \lambda$ complex gives the k-order difference equation

$$\sum_{j=0}^{k} (\alpha_j - h\lambda \beta_j) y_{n+j} = 0$$ \hspace{1cm} (3.4.12)

which has the stability polynomial equation

$$\Pi(r, h\lambda) = \sum_{j=0}^{k} (\alpha_j - h\lambda \beta_j) r^j = 0$$ \hspace{1cm} (3.4.13)

The polynomial $\Pi(r, h\lambda)$ is often referred as the characteristic polynomial or stability polynomial. From (3.4.9) and (3.4.10), if the first and second stability polynomial of the LMM are $p$ and $\sigma$, then from (3.4.13) the stability polynomial may be written as

$$\Pi(r, h\lambda) = \rho(r) - h\lambda \sigma(r) = 0$$ \hspace{1cm} (3.4.14)

Thus, from equations (3.4.12)-(3.4.14), we can make the following definition.
Definition 3.9
The LMM (3.4.1) is said to be absolutely stable if the roots $r_j$ of its stability polynomial satisfy $|r_j|<1, j=1,2,...,k$. The region of absolute stability is the region in the complex $(h\lambda)$ plane in which the method is stable and the interval of absolute stability its intersection with the real axis.

The other type of stability and the general methods for finding the region and interval of absolute stability are discussed, for example, in Lambert [1973], Scraton [1986], Butcher [1987] and Yaakub [1988].

3.4.2 The Adams Methods
Adams methods constitute a sub-family of LMM and is defined by

$$y_{n+k} - y_{n+k-1} = h \sum_{j=0}^{k} b_j f_{n+j}$$

(3.4.15)

The explicit Adams-Bashforth methods having been first introduced in a numerical investigation of capillary attraction by Bashforth and Adams in 1883. While the implicit Adams-Moulton method first appeared in connection with problems of ballistics by Moulton in 1926. These two most popular family of the LMM still remain in the basic form as predictor-corrector code for solving non-stiff initial value problems.

For convenience of presentation, Lambert [1991] introduced the explicit and implicit Adams methods as the k-step s-Adams method an expressed in the form

$$y_{n+s} - y_n = h \sum_{m=0}^{k-s} B_{m,s} f_{n+m}$$

(3.4.16)

where $B_{m,s} = \frac{b_{m-s}}{d}$ with $b_{m-s}$ under columns headed $f_{m-s}$ together with the denominator d, stepnumber k, order p and error constant $C_{p,s}$ are listed in the following tables. The coefficient of 0-Adams or the explicit Adams-Bashforth method with order $p=1,2,3,4,5,6$ are listed in Table 3.2.

56
While the coefficient of 1-Adams methods or the implicit Adams-Moulton method are shown in Table 3.3. The fourth and fifth order explicit Adams-Bashforth methods in Table 3.2 using a new fourth and fifth order single method as a starting value are discussed in chapter 8.

Table 3.2: Coefficients of 0-Adams methods or the explicit Adams-Bashforth methods

<table>
<thead>
<tr>
<th>$f_n$</th>
<th>$f_{n-1}$</th>
<th>$f_{n-2}$</th>
<th>$f_{n-3}$</th>
<th>$f_{n-4}$</th>
<th>$f_{n-5}$</th>
<th>$d$</th>
<th>$k$</th>
<th>$p$</th>
<th>$C_{p+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-1/2</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>-1/2</td>
</tr>
<tr>
<td>23</td>
<td>-16</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>3</td>
<td>3</td>
<td>-1/15</td>
</tr>
<tr>
<td>55</td>
<td>-59</td>
<td>37</td>
<td>-9</td>
<td></td>
<td></td>
<td>24</td>
<td>4</td>
<td>4</td>
<td>-1/15</td>
</tr>
<tr>
<td>1901</td>
<td>-2774</td>
<td>2616</td>
<td>-1274</td>
<td>251</td>
<td></td>
<td>720</td>
<td>5</td>
<td>5</td>
<td>-3/20</td>
</tr>
<tr>
<td>4277</td>
<td>-7923</td>
<td>9982</td>
<td>-7298</td>
<td>2877</td>
<td>-475</td>
<td>1440</td>
<td>6</td>
<td>6</td>
<td>-3/20</td>
</tr>
</tbody>
</table>

Table 3.3: Coefficients of 1-Adams methods or the implicit Adams-Moulton methods

<table>
<thead>
<tr>
<th>$f_n$</th>
<th>$f_{n-1}$</th>
<th>$f_{n-2}$</th>
<th>$f_{n-3}$</th>
<th>$f_{n-4}$</th>
<th>$f_{n-5}$</th>
<th>$d$</th>
<th>$k$</th>
<th>$p$</th>
<th>$C_{p+1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-1/12</td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td>2</td>
<td>3</td>
<td>-1/12</td>
</tr>
<tr>
<td>9</td>
<td>19</td>
<td>-5</td>
<td>1</td>
<td></td>
<td></td>
<td>24</td>
<td>3</td>
<td>4</td>
<td>-1/12</td>
</tr>
<tr>
<td>251</td>
<td>646</td>
<td>-264</td>
<td>106</td>
<td>-19</td>
<td></td>
<td>720</td>
<td>4</td>
<td>5</td>
<td>-1/20</td>
</tr>
<tr>
<td>475</td>
<td>1427</td>
<td>-798</td>
<td>482</td>
<td>-173</td>
<td>27</td>
<td>1440</td>
<td>5</td>
<td>6</td>
<td>-1/20</td>
</tr>
</tbody>
</table>

3.4.3 The Extrapolation Method

The extrapolation method is another way to improve the accuracy of the estimates in the numerical solution of initial value problems involving ordinary differential equations. These techniques have been studied by several authors, e.g., Fox [1962], Gragg [1965], Burlirsh and Stoer [1966], Lambert [1973] and Noye [1984]. The study of the extrapolation method is based on the idea known as Richardson extrapolation method, where the function
values are evaluated for the steplength or h-values to form a tableau.

From this tableau, higher order extrapolations with the h-value halved at each stage will be obtained. In practice, the number of stages is typically in the range 4 to 7. The discussion comparing the polynomial and the rational extrapolation methods by a parallel procedure will be presented in the following chapter.

3.4.4 The Other Methods

Lambert [1973] has presented some of the less popular methods namely block method, hybrid method, Obrechkoff method and the other methods for special problems.

a) Block Methods

The property of this method is to simultaneously produce approximations to the solution of the initial value problem at a block of points \( x_{n}, x_{n+1}, \ldots, x_{n+N} \). It can be formulated either in terms of linear multistep methods or their equivalent to certain Runge-Kutta methods. The block method equivalent to the fourth order six-stage explicit Runge-Kutta method given by Rosser [1967] are of the form

\[
y_{n+2} = y_{n} + \frac{h}{3} [k_{1} + 4k_{2} + k_{3}] \tag{3.4.17}
\]

where

\[
k_{1} = f(x_{n}, y_{n})
\]

\[
k_{2} = f(x_{n} + h, y_{n} + h k_{1})
\]

\[
k_{3} = f\left( x_{n} + h, y_{n} + \frac{h}{2} k_{1} + \frac{h}{2} k_{2} \right)
\]

\[
k_{4} = f(x_{n} + 2h, y_{n} + 2hk_{1})
\]

\[
k_{5} = f\left( x_{n} + h, y_{n} + \frac{h}{12} (5k_{1} + 8k_{2} - k_{3}) \right)
\]

\[
k_{6} = f\left( x_{n} + 2h, y_{n} + \frac{h}{3} (k_{1} + k_{4} + 4k_{5}) \right).
\]
One of the advantages over the classical Runge-Kutta method lies in the fact that they are less expensive in terms of function evaluations for a given order.

b) Hybrid Methods

The hybrid method has certain characteristics of the LMM and also uses information from the previous point or off-step point like by using a single step method. A k-step hybrid scheme is defined by

\[ \sum_{j=0}^{k} \alpha_j y_{sv,j} = h \sum_{j=0}^{k} \beta_j f_{sv,j} + h \beta_v f_{sv} \]  \hspace{1cm} (3.4.18)

where \( \alpha_i = 1, \alpha_v \) and \( \beta_v \) are not both zero \( v \not\in \{0,1,...,k\} \) and for \( f_{sv} = f(x_{sv},y_{sv}) \). Equation (3.4.18) is explicit if \( \beta_v = 0 \) and implicit if \( \beta_v \neq 0 \) and has a predictor to estimate \( y_{sv} \) of the form

\[ y_{sv} + \sum_{j=0}^{k} \alpha_j y_{sv,j} = h \sum_{j=0}^{k} \beta_j f_{sv,j} \]  \hspace{1cm} (3.4.19)

The order \( p \) in equation (3.4.19) is obtained in a similar way as in LMM by using constant as defined in definition 3.6.

c) Obrechkoff Methods

A Obrechkoff method is a modification of LMM but uses higher derivatives similar to the Taylor series methods. The k-step Obrechkoff methods or a multiderivative multistep method using the first derivative of \( y \) may be written

\[ \sum_{j=0}^{l} \alpha_j y_{sv,j} = \sum_{j=0}^{l} h^j \sum_{j=0}^{l} \beta_j y_{sv,j} \]  \hspace{1cm} (3.4.20)

where \( \alpha_i = 1 \) and one of \( \alpha_v, \beta_v, i=1,2,...,l \) is non-zero. As for the LMM of the same order of the implicit method, the Obrechkoff method is more accurate and has a better stability property compared to the explicit method.

The other methods for use in specific problems, e.g., problems with oscillatory solutions will be discussed in Chapter 9.
3.5 STIFF EQUATIONS AND ABSOLUTE STABILITY

3.5.1 Introduction

The initial value problems involving a system of ODEs which exhibit a phenomena which has come to be known as stiffness was applied in many fields of application especially in chemical engineering and control theory. One of the problems which can occur is when the solution to the system of equations contains components with widely differing time scales. For example, the general solution to a second order differential equation might be of the form

$$y(x) = C_1 \exp(ax) + C_2 \exp(bx) + C$$

(3.5.1)

where both $a$ and $b$ are negative but $b$ is much smaller than $a$ and the second term decays very much more rapidly than the first term. Typical of such equations is the existence of some components in the solution that decrease very fast and some components that decrease quite slowly. The study to solve these type of problems known as stiffness by using the methods or special techniques has attracted and received a lot of attention to many numerical analysts.

In this section, we discuss the stability and the method which involve these stiff problems of differential equations.

3.5.2 Stiffness of Initial Value Problem

Now, we consider the initial value problem

$$y'' + 101y' + 100y = 0, \quad y(0) = 1.01, \quad y'(0) = -2$$

(3.5.2)

The equation (3.5.2) can be written as the system

$$\begin{align*}
y_1' & = y_2 \\
y_2' & = -100y_1 - 101y_2 \end{align*}$$

(3.5.3)

which has the solution

$$\begin{align*}
y_1(x) & = 0.01 \exp(-100x) + \exp(-x) \\
y_2(x) & = -\exp(-100x) - \exp(-x) \end{align*}$$

(3.5.4)

The terms $\exp(-100x)$ and $\exp(-x)$ are classified as fast and slow components respectively. If we use the single step
method namely the fourth order Runge Kutta method to solve equation (3.5.2), we would have to use a small steplength to obtain a better accuracy. For absolute stability, we require $h \lambda \in (2.78,0)$ and since $h < \frac{1}{10}$. However, when one solves such a system over a large interval, one would like to take large value of $h$ as soon as the rapidly decaying components have disappeared. This phenomena cannot be done with the explicit method because of their stability property but it is possible to be solved by the implicit method.

In Lambert [1973], if the general linear constant coefficient system

$$y' = Ay + \phi(x) \quad (3.5.5)$$

where the matrix $A$ has distinct eigenvalues $\lambda_j$ and corresponding eigenvectors $c_j, j = 1, 2, ..., m$ has a general solution of the form

$$y(x) = \sum_{j=1}^{m} c_j k_j e^{\lambda_j x} + \psi(x) \quad (3.5.6)$$

with $\text{Re}(\lambda_j) < 0, j = 1, 2, ..., m$ then the term

$$\sum_{j=1}^{m} c_j k_j e^{\lambda_j x} \to 0 \quad \text{as} \quad x \to \infty \quad (3.5.7)$$

which we call the transient solution and the remaining term $\psi(x)$ is the steady-state solution.

A more formal definition of stiffness is given by Lambert [1973] as follows:

Definition 3.10:

The linear system (3.5.5) is said to be stiff if

i) $\text{Re}(\lambda_j) < 0, j = 1, 2, ..., m$

ii) $\text{Max}_{j=1,2,...,m} |\text{Re}(\lambda_j)| >> \text{Min}_{j=1,2,...,m} |\text{Re}(\lambda_j)|$

where $\lambda_j, j = 1, 2, ..., m$ are the eigenvalues of $A$ and the ratio

$$s = \frac{\text{Max}_{j=1,2,...,m} |\text{Re}(\lambda_j)|}{\text{Min}_{j=1,2,...,m} |\text{Re}(\lambda_j)|}$$

is called the stiffness ratio.
Another statement claimed by Lambert [1991] will be adopted as our definition is given as follows:

**Definition 3.11:**

If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use in a certain interval of integration a steplength which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be stiff in that interval.

### 3.5.3 Stability Theory and Method For Stiff Problems

The problem of stiffness as in equation (3.5.5) can be overcome if the method employed has a region of absolute stability which includes the whole of the left-half complex plane, i.e., \( \{h\lambda | \Re(h\lambda) < 0 \} \).

The basic stability theory of stiffness has been proposed in the following definition:

**Definition 3.12 (Dahlquist [1963]):**

A numerical method is said to be \( A \)-stable if its region of absolute stability contains the whole of the left-hand half-plane \( \Re(h\lambda) < 0 \).

Dahlquist also proved that a LMM of \( p > 2 \) cannot be \( A \)-stable and the most accurate \( A \)-stable LMM is the Trapezoidal method. In view of this, several less demanding stability or relaxed stability definition have been given by Widlund [1967] and Gear [1969].

**Definition 3.13: (Widlund [1967]):**

A numerical method is said to be \( A(\alpha) \)-stable, \( \alpha \in (0, \frac{\pi}{2}) \), if its region of absolute stability contains the infinite wedge \( \omega_a = \{h\lambda | -\alpha < \Pi - \Arg(h\lambda) < \alpha \} \); it is said to be \( A(0) \)-stable if it is \( A(\alpha) \)-stable for some sufficiently small \( \alpha \in (0, \frac{\pi}{2}) \).
The above definitions are concerned only with stability but Gear [1971] gives a definition of more complex property involving stability and accuracy of approximation.

**Definition 3.14:** (Gear [1971])

A numerical method is said to be stiffly stable if
i) its region of absolute stability contain $R_i$ and $R_i$,
ii) it is accurate for all $h \in R_i$ when applied to the scalar test equation $y' = \lambda y$, $\lambda$ a complex constant with $\text{Re}(\lambda) < 0$, where

$$R_i = \{ h \lambda | \text{Re}(h \lambda) < -a \} \text{ and }$$

$$R_i = \{ h \lambda | -a \leq \text{Re}(h \lambda) \leq b, -c \leq \text{Im}(h \lambda) \leq c \}$$

where $a, b$ and $c$ are positive constants. The second condition of accuracy requires absolute stability to the left of the imaginary axis and relative stability to the right of the imaginary axis.

A weaker condition where the region of absolute stability contains the whole negative real axis is given by Cryer [1973] in the following definition:

**Definition 3.15:** (Cryer [1973])

A numerical method is said to be $A_0$-stable if

$$\{ h \lambda | \text{Re}(h \lambda) < 0, \text{Im}(h \lambda) = 0 \}.$$

The above definition of stability are applicable to for the LMM while the definition for a single step method is defined by Ehle [1969] and Axelsson [1969] as follows:

**Definition 3.16:** (Ehle [1969] and Axelsson [1969])

A single step method is said to be L-stable if it is $A$-stable and in addition when applied to the scalar test equation $y' = \lambda y$, $\lambda$ a complex constant with $\text{Re}(\lambda) < 0$, it yields $y_{n+1} = Q(h \lambda) y_n$, where $Q(h \lambda) \to 0$ and $\text{Re}(h \lambda) \to -\infty$.

This property also called stiff $A$-stability or strong $A$-stability indicates their the region to the left
of the origin is required. From the above definition, we make the hierarchy of stability definitions of stiffness as follows:

\[
\text{L-stability} \Rightarrow \text{A-stability} \Rightarrow \text{stiff-stability} \\
\Rightarrow \text{A}(\alpha)\text{-stability} \Rightarrow \text{A}(0)\text{-stability} \Rightarrow A_0\text{-stability}.
\]

Lambert [1991], showed that the region of absolute stability from the above definition are shown in Figure 3.2.

![Figure 3.2: The stability region for stiff problems](image)

The most common method of solution of stiff systems is the use of backward differentiation formulas which are LMM of the form

\[
\sum_{j=0}^{k} \alpha_j y_{n+j} = h \beta_0 f_{n+k} 
\]  

(3.5.8)

where \( \alpha_j = 1, \alpha_0 \neq 0 \) and \( \beta_0 \neq 0 \) and \( k \) is equal to the step number \( k \). The region of absolute stability and
coefficients $\alpha_i$, $\beta_i$ and the parameter $a$ used as in Figure 3.2(b) for the definition of stiff stability are given in Gear [1971, pp 215 and 216] and Lambert [1991] are shown in Table 3.4.

Table 3.4: Coefficients of the Backward Differentiation Formula

<table>
<thead>
<tr>
<th>$k=p$</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
<th>$\alpha_4$</th>
<th>$\alpha_5$</th>
<th>$\alpha_6$</th>
<th>$\beta_1$</th>
<th>$-a$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-(\frac{1}{2})</td>
<td>(\frac{1}{2})</td>
<td>(\frac{1}{2})</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>-(\frac{2}{3})</td>
<td>(\frac{2}{3})</td>
<td>-(\frac{2}{3})</td>
<td>(\frac{1}{3})</td>
<td>0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>-(\frac{3}{8})</td>
<td>(\frac{3}{8})</td>
<td>-(\frac{3}{8})</td>
<td>(\frac{1}{8})</td>
<td></td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-(\frac{5}{10})</td>
<td>(\frac{5}{10})</td>
<td>-(\frac{5}{10})</td>
<td>(\frac{1}{10})</td>
<td>(\frac{2}{9})</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>-(\frac{7}{16})</td>
<td>(\frac{7}{16})</td>
<td>-(\frac{7}{16})</td>
<td>(\frac{1}{16})</td>
<td>(\frac{2}{9})</td>
<td>6.1</td>
<td></td>
</tr>
</tbody>
</table>

Following Hindmarsh [1974], the low order formulae of this type, provide the basis of the well known Gear's package for solving stiff differential equations.

From Table 3.4, the first three backward differentiation formulae are:

a) The backward Euler method, $k=1$

\[ y_{n+1} = y_n + hf_n, \]

b) Second order-method, $k=2$

\[ y_{n+2} - \frac{1}{2} y_{n+1} + \frac{1}{2} y_n = \frac{1}{2} hf_{n+2} \]

c) Third order-method, $k=3$

\[ y_{n+3} - \frac{15}{10} y_{n+2} + \frac{15}{10} y_{n+1} - \frac{1}{10} y_n = \frac{1}{10} hf_{n+3}. \]

The implicit Runge Kutta methods are another class of methods which have suitable stability characteristics for solving stiff systems of ODEs. Ehle [1969], proved that the R-stage implicit Runge Kutta method of order $2R$ is A-stable.

Finally new L-stable modified Trapezoidal methods for solving stiff systems of differential equations in initial value problems have been proposed by Yaakub and Evans [1996] and will be discussed in Chapter 4.
CHAPTER 4

NUMERICAL SOLUTION OF PROBLEMS INVOLVING ODEs BY USING THE VARIETY OF MEANS
The new Runge-Kutta formulae using different means rather than the conventional arithmetic mean (AM) has resulted in the introduction of a number of new formulae for the numerical solution of initial value problems

\[ y' = f(x, y), \quad y(x_0) = y_0 \]  

(4.0.1)

to be solved by the explicit single step method similar to equation (3.2.1). This chapter will discuss the derivation of second, third and fourth order Runge-Kutta method based on a variety of means to be developed. The accuracy of several modified third order and fourth order Runge-Kutta based on these variety of means will also be discussed.

4.1 DERIVATION OF A VARIETY OF MEANS FOR SOLVING ODE's

In many publications, e.g., Evans and Sanugi [1986], Evans and Yaakub [1993] and Yaakub and Evans [1995] it was shown that the standard fourth order arithmetic mean (AM) Runge-Kutta formula for solving IVPs of the form

\[ y' = f(x, y) \]  

may be written as

\[ y_{n+1} = y_n + \frac{h}{3} \left[ k_1 + \frac{k_2 + k_3 + k_4}{2} \right] \]  

(4.1.1)

where,

\[ k_1 = f(x_n, y_n) \]
\[ k_2 = f(x_n + a_2 h, y_n + h a_2 k_1) \]
\[ k_3 = f(x_n + (a_3 + a_5) h, y_n + h a_3 k_1 + h a_5 k_2) \]  

(4.1.2)
\[ k_4 = f(x_n + (a_4 + a_5 + a_6) h, y_n + h a_4 k_1 + h a_5 k_2 + h a_6 k_3) \]

A fourth order accurate formulas is obtained through the standard procedure of the adjustment of the parameters \( a_i, 1 \leq i \leq 6 \) for formula (4.1.1) where

\[ a_1 = \frac{1}{2}, \quad a_2 = 0, \quad a_3 = \frac{1}{2}, \quad a_4 = 0, \quad a_5 = 0, \quad a_6 = 1. \]
In Evans and Sanugi [1993], a new fourth order Runge-Kutta formula based on the concept of averaging the harmonic mean functional is given in the form

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \frac{1}{2} \left(\frac{2(k_{i+1})}{(k_i + k_{i+1})}\right) \right] \]  

(4.1.3)

The improved accuracy of (4.1.3) was achieved by adjusting the parameters \( a_i, 1 \leq i \leq 6 \) in (4.1.2), where

\[ a_1 = \frac{1}{2}, \quad a_2 = -\frac{1}{8}, \quad a_3 = \frac{5}{8}, \quad a_4 = -\frac{1}{4}, \quad a_5 = \frac{7}{20}, \quad a_6 = \frac{9}{10} \]

The geometric mean (GM) Runge-Kutta formula was developed by Evans and Sanugi [1986] in the form

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \sum_{i=1}^{3} \sqrt{k_{i+1}} \right] \]  

(4.1.4)

by replacing the arithmetic mean (AM) of the functional values in (4.1.1) by the average of the geometric mean (GM) values and adjusting the parameters, \( a_i, 1 \leq i \leq 6 \) in (4.1.2), to give

\[ a_1 = \frac{1}{2}, \quad a_2 = -\frac{1}{16}, \quad a_3 = \frac{9}{16}, \quad a_4 = -\frac{1}{8}, \quad a_5 = \frac{5}{24}, \quad a_6 = \frac{11}{12} \]

In this chapter, our concern is to establish a new fourth order Runge-Kutta formula based on the concept of averaging the contraharmonic mean \((C,M)\), centroidal mean \((C,M)\) and root-mean-square \((RMS)\) functional values. The establishment of the new methods are identical to the above discussion with emphasis on the numerical comparison.

4.2 A NEW RUNGE-KUTTA FORMULA BASED ON THE CONTRA哈ARMONIC MEAN \((C,M)\) FORMULA

4.2.1 Second Order Contraharmonic Mean \((C,M)\) Formula

Following Eves [1983] the largest functional mean is given by the Contraharmonic mean \((C,M)\) formula
\[ y_{n+1} = y_n + h \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) \]  
\[(4.2.1)\]

where, \( k_1 = f(x_n, y_n) \)
\[ k_2 = f(x_n + a_1 h, y_n + h a_1 k_1) \]  
\[(4.2.2)\]

Second order accuracy is obtained for formula (4.2.1) from the solution of the equation of condition, i.e.,
\[ h^2 f'_{y_y} : 1 - a_1 = 0 \]  
\[(4.2.3)\]
to give \( a_1 = 1 \).

A detailed explanation confirming the second order Contraharmonic mean \((C_o M)\) method is discussed in section 4.2.4.

### 4.2.2 Third Order Contraharmonic Mean \((C_o M)\) Formula

The third order Contraharmonic Mean \((C_o M)\) Runge-Kutta formula can be written in the form
\[ y_{n+1} = y_n + \frac{h}{2} \left[ \sum_{i=1}^{2} \left( \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right) \right] \]  
\[(4.2.4)\]

and is obtained by using the related Contraharmonic mean
\[ C_o M = \frac{2(AM)^2 - (GM)^2}{(AM)} \]  
\[(4.2.5)\]

Before the extension of (4.2.4), three function evaluations were proposed in Wazwaz [1993] which uses \( k_i, i = 1, 2, 3 \) in the form,
\[ y_{n+1} = y_n + \frac{h}{2} \left[ \frac{k_1^2 + k_2^2}{k_1 + k_2} + \frac{k_2^2 + k_3^2}{k_2 + k_3} \right] \]  
\[(4.2.6)\]

where, \( k_1 = f(x_n, y_n) \)
\[ k_2 = f(x_n + a_1 h, y_n + h a_1 k_1) \]  
\[(4.2.7)\]

\[ k_3 = f(x_n + (a_1 + a_2) h, y_n + a_1 h k_1 + a_2 h k_2) \]

Through the standard procedure of the adjustment of the parameters, a third order accurate formula is
obtained for formula (4.2.6) from the solution of the equations of condition, i.e.,

\[ h'^{2}f_{,i} = 2a_1 + a_i + a_i = 2, \]  
(4.2.8-i)

\[ h'^{2}f_{,i} = 3(a_1 + a_i)^2 + 6a_i^2 = 4, \]  
(4.2.8-ii)

\[ h'^{2}f_{,i} = 9a_i^2 + 6a_1a_i + 3a_2a_i + 3(a_1 + a_i)^2 = 8 \]  
(4.2.8-iii)

We can easily obtain the parameters \( a_i, 1 \leq i \leq 3 \) in (4.2.8-i)-(4.2.8-iii) by the use of Mathematica. It follows that

\[ a_1 = \frac{2}{3}, \quad a_2 = 0, \quad a_3 = \frac{2}{3} \]

4.2.3 New Fourth Order Contraharmonic Mean \((C_{oM})\) Formula

Now we attempt to establish a 4-stage Runge-Kutta formula based on the Contraharmonic Mean in the form,

\[ y_{m+1} = y_n + \frac{h}{3} \left[ \frac{k_1^2 + k_2^2 + k_3^2 + k_4^2}{k_1 + k_2 + k_3 + k_4} \right] \]  
(4.2.9)

as a direct extension of method (4.2.9) where,

\[ k_1 = f(x_n, y_n) \]

\[ k_2 = f(x_n + a_1 h, y_n + h a_1 k_1) \]  
(4.2.10)

\[ k_3 = f(x_n + (a_i + a_s) h, y_n + a_i h k_1 + a_s h k_2) \]

\[ k_4 = f(x_n + (a_i + a_s + a_e) h, y_n + a_i h k_1 + a_s h k_2 + a_e h k_3) \]

The Taylor series expansion of \( k_i, 1 \leq i \leq 4 \) where for simplicity we have considered \( f \) as a function of \( y \) only in equations (3.3.11-i)-(3.3.11-iv).

Normally, we would substitute equations (3.3.11-i)-(3.3.11-iv) into (4.2.9) to obtain an expression for \( y_{m+1} \) in terms of the function and its derivatives and the parameters \( a_i, 1 \leq i \leq 6 \), so that it can be matched with the Taylor series expansion of \( y(x_{m+1}) \) as in equation (3.3.8)
through terms of order \((h^4)\). The fraction involved in (4.2.9) however is not amenable to direct substitution of the series, i.e.

\[
\frac{k_i^2 + k_{i_{w1}}^2}{k_i + k_{i_{w1}}} \quad , \quad 1 \leq i \leq 3 . \quad (4.2.11)
\]

However, these difficulties can be alleviated by multiplying the terms across with the common denominator

\[
(k_i + k_{j})(k_i + k_{j})(k_i + k_{j}) \quad (4.2.12)
\]

Following Sanugi and Evans [1993], the formula from (4.2.9) may be written as,

\[
y_{w1} = y_s + \frac{\text{UPPER}}{\text{LOWER}} \quad (4.2.13)
\]

where,

\[
\text{UPPER} = \frac{h}{3} [(k_i^2 + k_j^2)(k_i + k_j)(k_i + k_j) + (k_i^2 + k_j^2)(k_i + k_j)(k_i + k_j) + (k_i^2 + k_j^2)(k_i + k_j)(k_i + k_j)] \quad (4.2.14)
\]

and,

\[
\text{LOWER} = (k_i + k_j)(k_i + k_j)(k_i + k_j) \quad (4.2.15)
\]

while the Taylor series expansion of \(y(x_{w1})\) was given in equation (3.3.8).

From (4.2.13)-(4.2.15) and the Taylor series, we can now write

\[
\text{ERROR} = \text{TAYLOR} - \frac{\text{UPPER}}{\text{LOWER}}
\]

\[
(\text{TAYLOR} \times \text{LOWER}) - \text{UPPER} = (\text{LOWER} \times \text{ERROR}) \quad (4.2.16)
\]

By comparing coefficient of similar terms in (4.2.16) up to terms in \((h^4)\) yields the following equations of conditions:

\[
h^1 f f_i : \quad -2a_i - 2s_i - s_i + 3 = 0 \quad (4.2.17-i)
\]

\[
h^1 f^2 f_w : \quad -2a_i^2 - 2s_i^2 - s_i^2 + 2 = 0 \quad (4.2.17-ii)
\]

\[
h^2 f f_i^2 : \quad 2 + 6a_i - 6a_i^2 - 4a_i a_s - 2a_i a_s + 6s_i - 6a_s s_i - 2a_s s_i - 6s_i^2
\]
\[
+ 3s_i - 4a_i s_i - 2s_i s_i - 2s_i^2 = 0 \quad (4.2.17-iii)
\]
\[ h^4 f^3 f'' : -4a^2 - 4s^2 - 2s^2 + 3 = 0 \]  \hspace{1cm} (4.2.17-iv)

\[ h^4 f^3 \ : \ 1 + 4a + 3a^2 + 12a_1a_2 - 12a_1^2a_2 + 6a_1a_5 - 8a_1^2a_5 - 4a_1a_5a_6 + 4s_1 + 9a_1s_1 - 9a_1^2s_1 - 24a_1s_1 - 4a_1^2s_1 + 6a_1s_2 - 8a_1a_5s_2 + 3s_1s_2
\]
\[ + 3s_1^2 - 9a_1s_2^2 - 4a_1s_2^2 - 4s_2^2 + 2s_1 + 6a_5s_1 - 7a_1^2s_1 - 4a_1a_5s_1 + 3s_1s_2
\]
\[ - 8a_1a_5s_2 - 3a_1s_2s_3 - 8a_1s_2s_3 - 2s_1^2s_3 - 4a_1s_2^2 - 2s_2s_3 = 0 \]  \hspace{1cm} (4.2.17-v)

\[ h^4 f^2 f'' : 4 + 4a + 6a^2 - 12a_1 - 4a_1^2a_5 - 2a_1^2a_5 + 4s_1 - 6a_1s_2
\]
\[ - 8a_1a_5s_2 + 6s_1^2 - 6a_1s_2^2 - 2a_1s_2^2 - 12s_2^2 + 2s_1 - 4a_1s_3
\]
\[ - 4a_1a_5s_3 + 4a_1s_2s_3 - 2s_1^2s_3 + 3s_3^2 - 4a_1s_2^2 - 2s_2s_3^2 - 4s_3^2 = 0 \]  \hspace{1cm} (4.2.17-vi)

Consequently equations (4.2.17-i)-(4.2.17-vi) are then solved simultaneously by Mathematica to immediately obtain the parameters, i.e.,

\[ a_1 = \frac{1}{2}, a_2 = \frac{1}{8}, a_3 = \frac{3}{8}, a_4 = \frac{1}{4}, a_5 = -\frac{3}{4} \text{ and } a_6 = \frac{3}{2} \]

Therefore, the Contraharmonic mean (C,M) Runge-Kutta formula in equations (4.2.9)-(4.2.10) can now be written as

\[ k_1 = f(x, y) \]
\[ k_2 = f(x + h, y + \frac{1}{2}hk) \]
\[ k_3 = f(x + \frac{h}{2}, y + \frac{1}{8}hk + \frac{3}{8}hk) \]  \hspace{1cm} (4.2.18)
\[ k_4 = f(x + h, y + \frac{1}{4}hk - \frac{3}{4}hk + \frac{3}{2}hk) \]
\[ y_{n+1} = y_n + \frac{h}{3} \left[ \frac{k_1^2 + k_2^2}{(k_1 + k_2) + \frac{k_3^2 + k_4^2}{(k_3 + k_4)} + \frac{k_3^2 + k_4^2}{(k_3 + k_4)} \right] \]  \hspace{1cm} (4.2.19)

to achieve fourth order accuracy. Confirmation of this fourth order method will be discussed in the next section.
4.2.4 Truncation Error For The Contraharmonic Mean

In this section we determine the truncation errors and develop the stability analysis of the contraharmonic mean \((C_0 M)\) method.

a) Truncation Error For The Second Order Contraharmonic Mean

The second order Contraharmonic mean \((C_0 M)\) formula is written as

\[
y_{n+1} = y_n + h \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right)
\]  

(4.2.20)

By substituting \(k_1\) and \(k_2\) into the incremental function, we obtain a new formula given by the relation

\[
y_{n+1} = y_n + h \left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right)
\]  

(4.2.21)

We will establish the truncation error by expanding the right hand side of (4.2.21) as a Taylor Series expansion about \(x_n\). By putting

\[
f_{n+1} = y_{n+1}'
\]

\[
= y_n' + h y_{n}'' + \frac{1}{2} h^2 y_{n}''' + \frac{1}{6} h^3 y_{n}^{(4)} + \ldots
\]

(4.2.22)

or \((f_{n+1})^2 = (y_{n+1}')^2\)

\[
= (y_n')^2 + 2 h y_n'' y_n' + h^2 [ (y_n')^2 + y_n y_n'' ] + \ldots
\]

(4.2.23)

in the formula (4.2.21) above, we have

\[
y_{n+1} = y_n + h \left( \frac{(y_n')^2 + (y_{n+1}')^2}{y_n' + y_{n+1}'} \right)
\]

(4.2.24)

Therefore, the increment function in (4.2.21) can be written as

\[
\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} = \frac{2(y_n')^2 + 2 h y_n'' y_n' + h^2 [(y_n')^2 + y_n y_n'']}{2 y_n' + h y_n''' + \frac{1}{2} h^2 y_n'''} + \ldots
\]

(4.2.25)
By direct division of the quotient on the right hand side in equation (4.2.25) we have

\[
\frac{f_n^2 + f_m^2}{f_n + f_m} = y_n' + \frac{1}{2} h y_n'' + \frac{h^2}{2 y_n} \left[ \frac{1}{2} (y_n')^2 + \frac{1}{2} y_n' y_n'' \right] + \ldots
\]  

(4.2.26)

Thus equation (4.2.21) becomes

\[
y_{n+1} = y_n + h \left[ y_n' + \frac{h}{2} y_n'' + \frac{h^2}{2 y_n} \left[ \frac{1}{2} (y_n')^2 + \frac{1}{2} y_n' y_n'' \right] + \ldots \right] \ldots
\]  

(4.2.27)

while the Taylor series expansion of \( y(x_n) \) has the form,

\[
y(x_n) = y_n + hy_n' + \frac{h^2}{2} y_n'' + h^3 \left( \frac{y_n'''}{4} + \frac{(y_n'^2)}{4y_n'} \right) + \ldots
\]  

(4.2.28)

Thus, the local truncation error (LTE) is given by

\[
LTE = y(x_{n+1}) - y_{n+1}
\]

\[
= h^3 \left( -\frac{y_{n}'''}{12} - \frac{(y_n'^2)}{4y_n'} \right).
\]  

(4.2.29)

This confirms that the equation (4.2.21) is of order two.

b) Truncation Error For The Third Order Contraharmonic Mean

The third order contraharmonic mean formula is written as

\[
y_{n+1} = y_n + \frac{h}{2} \left[ \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) + \left( \frac{k_3^2 + k_4^2}{k_3 + k_4} \right) \right]
\]  

(4.2.30)

where,

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + \frac{2}{3} h, y_n + \frac{2}{3} h k_1)
\]  

(4.2.31)

\[
k_3 = f \left( x_n + \frac{2}{3} h, y_n + \frac{2}{3} h k_1 \right)
\]
The formula from equation (4.2.30) can be written as in equation (4.2.13), where

$$\text{UPPER} = \frac{h}{2} \left[\left( k_i^2 + k_j^2 \right)(k_i + k_j) + \left( k_i^2 + k_j^2 \right)(k_i + k_j) \right]$$  \hspace{1cm} (4.2.32)

and

$$\text{LOWER} = (k_i + k_j)(k_i + k_j)$$ \hspace{1cm} (4.2.33)

while the Taylor series expansion of $y(x_{n+1})$ may be given as in equation (3.3.8). From equations (4.2.13) and (4.2.32)-(4.2.33), we can write the equation as in (4.2.16). When we substitute $k_i, 1 \leq i \leq 3$ in (4.2.31) and then solve by Mathematica, we immediately obtain,

$$\text{LOWER} \times \text{ERROR} = \frac{f^3(13f^3 + 4ff'f'' + f^2f'''h^4)}{54} + O(h^5)$$

$$= C_0 + C_1h^3 + C_2h^5 + C_4h^7 + C_5h^9 + O(h^5)$$

$$= P(h)$$ \hspace{1cm} (4.2.34)

where \( C_0 = C_1 = C_2 = C_3 = 0. \)

From equation (4.2.33), we also immediately obtain, i.e,

$$\text{LOWER} = (k_i + k_j)(k_i + k_j)$$

$$= 4f^4 + 4f^2f'h + \frac{h^2}{9} \left(16f^2f'_h + 12f^3f'' \right)$$

$$+ \frac{h^2}{27} \left(8f^2f'_h + 40f^3f'' + 8f^4f''' \right)$$

$$+ \frac{h^4}{81} \left(48f^2f'_h + 24f^3f'' + 32f^4f''' + 4f^5f'''' \right) + O(h^5).$$

$$= b_0 + b_1h + b_2h^2 + b_3h^3 + b_4h^4 + b_5h^5 + O(h^6)$$

$$= Q(h)$$ \hspace{1cm} (4.2.35)

From (4.2.34), we get

$$\text{ERROR} = \frac{\text{TAYLOR} \times \text{LOWER} - \text{UPPER}}{\text{LOWER}}$$

$$= \frac{P(h)}{Q(h)}$$

$$= a_0 + a_1h + a_2h^2 + a_3h^3 + a_4h^4 + a_5h^5 + O(h^6)$$ \hspace{1cm} (4.2.36)
Butcher [1987] assumes that the component of $y_n$ as an approximation to the corresponding component of $y(x_{i-1}+h)$ takes the form $a_0 + a_1h + a_2h^2 + a_3h^3 + \ldots + a_mh^m$ for the manipulation of power series. If a second expansion takes the form $b_0 + b_1h + b_2h^2 + b_3h^3 + \ldots + b_nh^n$ is added or subtracted we simply add or subtract the corresponding coefficients. Multiplying series and truncating at the $h^m$ term gives

$$ (a_0 + a_1h + a_2h^2 + a_3h^3 + \ldots + a_mh^m)(b_0 + b_1h + b_2h^2 + b_3h^3 + \ldots + b_nh^n) $$

$$ = C_0 + C_1h + C_2h^2 + C_3h^3 + \ldots + C_mh^m, $$

where

$$ c_i = \sum_{j=0}^{i} a_i \cdot b_j $$

and a quotient

$$ (C_0 + C_1h + C_2h^2 + C_3h^3 + \ldots + C_mh^m)(b_0 + b_1h + b_2h^2 + b_3h^3 + \ldots + b_nh^n)^{-1} $$

$$ = a_0 + a_1h + a_2h^2 + a_3h^3 + \ldots + a_mh^m $$

is found by re-interpreting the relationship between $a_i, b_i, c_i$ to give

$$ a_i = \begin{cases} 
C_0 & \text{if } i = 0, \\
\frac{C_i - \sum_{j=0}^{i-1} a_i \cdot b_j}{b_0} & \text{if } i > 0.
\end{cases} $$

where $C_0 = C_1 = C_2 = C_3 = 0$ and

$$ a_i = \frac{f^3(13f_i^2 + 4ff_i f_n + f^2f_m)}{54 \times 4f^3}h^i $$

which achieves third order accuracy and the local truncation error can be written as

$$ LTE = \frac{h^4}{216} \left(13ff_i^2 + 4f^2f_i f_n + f^3f_m\right). $$

(4.2.37)
c) Truncation Error For The Fourth Order Contraharmonic Mean

The formula from (4.2.19) can be written as in (4.2.13) where,

\[
UPPER = \frac{h}{3} \left[ (k_1^2 + k_2^2)(k_2 + k_3) + (k_3^2 + k_4^2)(k_3 + k_4) + (k_4^2 + k_5^2)(k_4 + k_5)(k_5 + k_6) \right]
\]

... (4.2.38)

and

\[
LOWER = (k_1 + k_2)(k_2 + k_3)(k_3 + k_4)
\]

(4.2.39)

while the Taylor series expansion of \( y(x_{ni}) \) may be given as in equation (3.3.8). When we substitute \( k_i, 1 \leq i \leq 4 \) from (4.2.18) into (4.2.16) and then solve by Mathematica, we immediately obtain,

\[
LOWER \times ERROR = \frac{f'(-378f_s - 303ff_s f_f - 648f^3f_f f_f - 8f^3f_f f_f f_f)}{2880} + O(h^6)
\]

\[
= C_0 + C_1 h + C_2 h^2 + C_3 h^3 + C_4 h^4 + C_5 h^5 + 0(h^6)
\]

\[
= P(h)
\]

(4.2.40)

where \( C_0 = C_1 = C_2 = C_3 = C_4 = 0. \)

From equation (4.2.39), we also immediately obtain, i.e,

\[
LOWER = (k_1 + k_2)(k_2 + k_3)(k_3 + k_4)
\]

\[
= 8f^3 + 12f^3f_f h + \frac{1}{2} f^3(17f_s^2 + 8ff_f^2)h^2 + \left( \frac{69}{16} f^3f_f^2 + \frac{13}{2} f^3f_f f_f + f^3f_f f_f f_f \right)h^3
\]

\[
+ \frac{f^3}{384} \left( 567f_s^2 + 2370ff_f^2 f_f + 420f^2f_f^2 + 728f^2f_f f_f + 80f^2f_f f_f f_f \right)h^4
\]

\[
+ \frac{f^3}{7680} \left( 2295f_s^2 + 27495ff_f^2 f_f + 19290f^2f_f^2 + 17160f^2f_f f_f + 4360f^2f_f f_f f_f + 3560f^2f_f f_f f_f f_f \right)h^5 + O(h^6)
\]

\[
= b_0 + b_1 h + b_2 h^2 + b_3 h^3 + b_4 h^4 + b_5 h^5 + 0(h^6)
\]

\[
= Q(h)
\]

(4.2.41)

Using the technique used by Butcher [1987] for the division of two series, we get

\[
a_0 = \frac{c_0}{b_0} = \frac{0}{8f^3} = 0
\]

and,

77
\[ a_i = \frac{\sum_{j=1}^{i} a_j b_j}{b_i} \quad , \quad i = 1, 2, 3, \ldots, r \]

where \( a_1 = a_2 = a_3 = a_4 = 0 \) and

\[ a_5 = \frac{f^4(-378f_1^4 - 303ff_1^2f_2 + 648f_2f_1^3 + 4f^3f_2f_3 - 8f^3f_3f_4)h^5}{2880 \times 8f^3} \]

This is the Local Truncation Error (LTE) and can be written as

\[ LTE = \frac{h^5}{23040} \left[-378ff_1^4 - 303ff_1^2f_2 + 648f_2f_1^3 + 4f^3f_2f_3 - 8f^3f_3f_4\right] + O(h^6) \]
as \( h \to 0 \).

When we substitute equation (4.2.38) and (4.2.39) into (4.2.13), the equation (4.2.13) can be simplified to

\[ y_{n+1} = y_n + \frac{UPPER}{LOWER} y_n \]

\[ = y_n \left(1 + \frac{UPPER}{LOWER}\right) \]

\[ = y_n \left(\frac{UPPER + LOWER}{LOWER}\right) \]

\[ : \quad \frac{y_{n+1}}{y_n} = \frac{LOWER + UPPER}{LOWER} \quad (4.2.42) \]

where,

\[ UPPER = 8z + 16z^2 + \frac{95}{6} z^3 + \frac{523}{48} z^4 + \frac{2207}{384} z^5 + \frac{593}{256} z^6 + O(z^7) \quad (4.2.43) \]

and

\[ LOWER = 8 + 12z + \frac{17}{2} z^2 + \frac{69}{16} z^3 + \frac{189}{128} z^4 + \frac{153}{512} z^5 + O(z^6) \]

\[ = b_0 + b_1 z + b_2 z^2 + b_3 z^3 + b_4 z^4 + b_5 z^5 + O(z^6) \quad (4.2.44) \]

\[ LOWER + UPPER = 8 + 20z + \frac{49}{2} z^2 + \frac{967}{48} z^3 + \frac{4751}{384} z^4 + \frac{9287}{1536} z^5 + O(z^6). \]

\[ = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + c_4 z^4 + c_5 z^5 + O(z^6) \quad (4.2.45) \]

We substitute equations (4.2.43)-(4.2.45) into (4.2.42) and using Butcher's method, we get
\[
\frac{y_{\text{asym}}} = 8 + 20z + \frac{49}{2} z^2 + \frac{967}{48} z^3 + \frac{4751}{384} z^4 + \frac{9287}{1536} z^5 + \mathcal{O}(z^6)
\]

\[
y = 8 + 12z + \frac{17}{2} z^2 + \frac{69}{16} z^3 + \frac{189}{128} z^4 + \frac{153}{512} z^5 + \mathcal{O}(z^6)
\]

\[
Q(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4 + a_5 z^5 + \mathcal{O}(z^6)
\]

\[ (4.2.46) \]

with, \( a_0 = \frac{c_0}{b_0} \)

or

\[ a_i = \frac{c_i - \sum_{j=1}^{i} a_i b_j}{b_0} \quad i = 1, 2, \ldots, r \]

\[ a_0 = \frac{c_0}{b_0} = 1 \]

\[ a_1 = \frac{c_1 - a_0 b_1}{b_0} = \frac{20 - 1(12)}{8} = 1 \]

\[ a_2 = \frac{c_2 - a_0 b_2}{b_0} = \frac{49 - 1(12) - 1(17/2)}{8} = \frac{1}{2} \]

\[ a_3 = \frac{c_3 - a_0 b_3}{b_0} = \frac{967 - \frac{1}{2}(12) - 1(17/2) - 1(69/16)}{8} = \frac{1}{6} \]

\[ a_4 = \frac{c_4 - a_0 b_4}{b_0} = \frac{4751 - \frac{1}{6}(12) - \frac{1}{2}(17/2) - 1(69/16) - 1(189/128)}{8} = \frac{1}{24} \]
\[ a_n = c_n - a_n b_1 - a_n b_2 - a_n b_3 - a_n b_4 - a_n b_5 \]
\[ = \left( \frac{9287}{1536} \right) - \left( \frac{1}{24} \right)(12) - \left( \frac{1}{6} \right)(\frac{17}{2}) - \left( \frac{1}{2} \right)(\frac{69}{16}) - 1(\frac{189}{128}) - 1(\frac{153}{512}) \]
\[ = \frac{19}{768} \]

When we substitute \( a_n = 1, a_1 = 1, a_2 = \frac{1}{2}, a_3 = \frac{1}{6}, a_4 = \frac{1}{24} \) and \( a_5 = \frac{19}{768} \)
into equation (4.2.46), we get
\[ Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{19z^5}{768} + O(z^6) \]

The Taylor series expansion of exponential \( z \) is
\[ e^z = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + O(z^6) \]

When we subtract equation (4.2.48) from equation (4.2.47) we obtain
\[ e^z - Q(z) = \left( \frac{1}{120} - \frac{19}{768} \right)z^5 + O(z^6) \]
\[ = -\frac{21}{1280}z^5 + O(z^6) \]
\[ = C_4z^5 + O(z^6) \]

This confirms that the contraharmonic mean formula is of order four with error constant \( C_4 = -\frac{21}{1280} \). The equation
\[ \frac{y_{n+1}}{y_n} = Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} - \frac{21z^5}{1280} + O(z^6) \]

is called the "Stability Polynomial". For absolute stability we must have \( |Q(z)| < 1 \) and for the stability region, we can plot the graph for \( |Q(z)| = 1 \) by using Mathematica.
4.2.5 Stability Analysis Of The Contraharmonic Mean \((C_0M)\)

Let us now to discuss the stability analysis and the properties of the absolute stability region for the Contraharmonic mean formula from the second order to fourth order. For various stages and orders, we can show that the stability regions are different.

The first order method in the family of the \(C_0M\) formulae is the Euler's method. Its stability property has been discussed in various references, namely, in Lambert [1973] and Yaakub [1988].

a) Stability Analysis Of The Second Order Contraharmonic Mean

By applying the test equation
\[
y' = \lambda y \quad y(0) = 1
\]
to equation \((4.2.20)\) we obtain
\[
y_{n+1} = y_n + h \left[ \frac{\lambda^2 y_n^2 + \lambda y_{n+1}^2}{\lambda y_n + \lambda y_{n+1}} \right]
\]
or
\[
y_{n+1} = y_n + h\lambda \left( \frac{y_n^2 + y_{n+1}^2}{y_n + y_{n+1}} \right)
\]
On multiplying both sides by \(y_n + y_{n+1}\), we obtain
\[
y_{n+1}(y_n + y_{n+1}) = y_n(y_n + y_{n+1}) + h\lambda(y_n^2 + y_{n+1}^2)
\]
or
\[
y_{n+1}^2 = y_n^2 + h\lambda(y_n^2 + y_{n+1}^2)
\]
Dividing by \(y_n^2\) on both sides, we have
\[
\left( \frac{y_{n+1}}{y_n} \right)^2 = 1 + h\lambda \left[ 1 + \left( \frac{y_{n+1}}{y_n} \right)^2 \right]
\]
\[
(1 - h\lambda) \left( \frac{y_{n+1}}{y_n} \right)^2 = 1 + h\lambda
\]
By taking the positive sign we obtain
\[
\frac{y_{n+1}}{y_n} = \sqrt{1 + \frac{h\lambda}{1 - h\lambda}} \quad (4.2.50)
\]
The stability region of the formula is determined by the values of $h\lambda$ in the complex plane where $\frac{y_{n+1}}{y_n}$ as given in equation (4.2.50) satisfies the inequality,

$$\left|\frac{y_{n+1}}{y_n}\right| < 1 \quad \text{or} \quad \left|\frac{1 + h\lambda}{1 - h\lambda}\right| < 1$$

or

$$\left|\frac{1 + h\lambda}{1 - h\lambda}\right| < 1 \quad (4.2.51)$$

The graphic surface for the second order Contraharmonic mean $(C_0M)$ formula is as shown in Figure 4.1.

![Figure 4.1: Graphic surface defined by (4.2.51)](image)

The stability region for the second order formula is define by equation (4.2.51) shown in Figure 4.2 i.e.,

![Figure 4.2: Stability region for second order $C_0M$](image)
b) Stability Analysis Of The Third Order Contraharmonic Mean

Let us now proceed with examining the stability of the third order 3-stage formula. For this purpose we will use the formula in equations (4.2.6) to (4.2.7). By applying formula (4.2.7) to the test equation \( y' = \lambda y \) and \( h\lambda = z \), we obtain

\[
k_1 = \lambda y_n \tag{4.2.52-i}
\]

\[
k_2 = \lambda \left[ y_n + \frac{2}{3} h\lambda y_n \right] = \lambda y_n \left[ 1 + \frac{2}{3} z \right] \tag{4.2.52-ii}
\]

\[
k_3 = \lambda \left[ y_n + \frac{2}{3} h(\lambda y_n) \left( 1 + \frac{2}{3} z \right) \right] = \lambda y_n \left[ 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right] \tag{4.2.52-iii}
\]

\[
y_{n+1} = y_n + \frac{h}{2} \left[ \frac{\lambda^2 y_n^2 + \lambda^2 y_n^2 \left( 1 + \frac{2}{3} z \right)^2}{\lambda y_n + \lambda y_n \left( 1 + \frac{2}{3} z \right)} + \frac{\lambda^2 y_n^2 \left( 1 + \frac{2}{3} z \right)^2 + \lambda^2 y_n^2 \left( 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right)^2}{\lambda y_n \left( 1 + \frac{2}{3} z \right) + \lambda y_n \left( 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right)} \right]
\]

\[
y_{n+1} = y_n + \frac{h}{2} \left[ \frac{\lambda^2 y_n^2 \left( 1 + \frac{2}{3} z \right)^2}{\lambda y_n \left( 1 + \frac{2}{3} z \right)} + \frac{\lambda^2 y_n^2 \left( 1 + \frac{2}{3} z \right)^2 + \lambda^2 y_n^2 \left( 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right)^2}{\lambda y_n \left( 1 + \frac{2}{3} z \right) + \lambda y_n \left( 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right)} \right]
\]

\[
y_{n+1} = y_n + \frac{h}{2} \left( \frac{\left( 1 + \frac{2}{3} z \right)^2}{2 + \frac{2}{3} z} \right) + \frac{\left( 1 + \frac{2}{3} z \right)^2 + \left( 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right)^2}{\left( 1 + \frac{2}{3} z \right) + \left( 1 + \frac{2}{3} z + \frac{4}{9} z^2 \right)} \right)
\]
If we denote

\[
\frac{\gamma_{\text{mil}}}{\gamma_n} = 1 + \frac{z}{2} \left[ \frac{\left(1 + \left(1 + \frac{2}{3} z\right) \right) + \left( \left(1 + \frac{2}{3} z\right)^2 + \left(1 + \frac{2}{3} z + \frac{4}{9} z^2\right)^2 \right)}{2 + \frac{2}{3} z} + \left( \frac{2 + 4}{9} z + \frac{4}{9} z^2 \right) \right]
\]

we write equation (4.2.53) as the quadratic equation

\[
Q(z) = 1 + \frac{z}{2} \left[ \frac{z + \left(1 + \frac{2}{3} z\right)}{4 + \frac{4}{3} z} + \frac{z \left(1 + \frac{2}{3} z\right)^2 + z \left(1 + \frac{2}{3} z + \frac{4}{9} z^2\right)^2}{4 + \frac{8}{3} z + \frac{8}{9} z^2} \right]
\]

\[
= 1 + z + \frac{z^2}{6} + \frac{z^3}{54} + O(z^4)
\]

we write equation (4.2.53) as the quadratic equation

\[
2Q^2 - (2 + 2z + z^2)Q - \left(\frac{z}{3} + \frac{8z}{27} + \frac{7z^2}{54} + \frac{z^3}{27} - \frac{z^4}{81} + \frac{z^4}{1458}\right) = 0 \quad (4.2.54)
\]

The roots of equation (4.2.54) are

\[
Q = \frac{(54 + 54z + 27z^2) \pm (-54 - 54z - 27z^2 - 18z^3 + 2z^4)}{108}
\]

To determine the stability region of the third order \(C_0M\) formula, we need to find all values of \(z\) in the complex plane that satisfy the condition in equation (3.3.31), i.e.,

\[
\left| \frac{(54 + 54z + 27z^2) + (-54 - 54z - 27z^2 - 18z^3 + 2z^4)}{108} \right| < 1 \quad (4.2.56)
\]

The graphic surface for the third order formula is as shown in Figure 4.3.
The stability region for the third order formula is defined by equation (4.2.56) as shown in Figure 4.4, i.e.,

\[
\text{Re}(z) \\
\text{Im}(z)
\]

Figure 4.4: Stability region for third order \( C_0M \)

c) Stability Analysis Of The Fourth Order Contraharmonic Mean

Finally we examine the stability region for the fourth order contraharmonic mean with the test equation \( y' = \lambda y \) and we obtain

\[
k_1 = \lambda y, \quad (4.2.57-i)
\]

\[
k_2 = \lambda [y + ha, k_1] \\
= \lambda y, [1 + za_1], \quad (4.2.57-ii)
\]

\[
k_3 = \lambda [y + ha, k_1 + ha, k_2] \\
= \lambda y, [1 + z(a_1 + a_2) + z^2a_1a_2], \quad (4.2.57-iii)
\]

and
\[ k_i = \lambda [y_i + ha_i + ha_k + ha_i k_i] \]
\[ = \lambda [y_i + ha_i (\lambda y_i) + ha_i \lambda y_i (1 + za_i) + ha_i \lambda y_i (1 + za_i + za_i + za_i)] \]
\[ = \lambda y_i [1 + za_i + za_i + za_i + za_i + za_i + za_i + za_i] \]
\[ \ldots \text{(4.2.57-iv)} \]

If we substitute (4.2.57-i) - (4.2.57-iv) and \( a_{i=1(6)1} \), i.e.

\[ a_1 = \frac{1}{2}, \quad a_2 = \frac{1}{8}, \quad a_3 = \frac{3}{8}, \quad a_4 = \frac{1}{4}, \quad a_5 = -\frac{3}{4}, \quad a_6 = \frac{3}{2} \]

in the fourth order contraharmonic mean \((C_0M)\) formula in equation (4.2.19), we get

\[ y_{*1} = y_* + \frac{h}{3} \left[ \frac{\lambda^2 y_*^2 + \lambda^2 y_*^2 \left(1 + \frac{z}{2}\right)^2}{\lambda y_* + \lambda y_* \left(1 + \frac{z}{2}\right)} + \frac{\lambda^2 y_*^2 \left(1 + \frac{z}{2}\right)^2 + \lambda^2 y_*^2 \left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2}{\lambda y_* \left(1 + \frac{z}{2}\right) + \lambda y_* \left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)} \right] \]
\[ + \frac{\lambda^2 y_*^2 \left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2 + \lambda^2 y_*^2 \left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)^2}{\lambda y_* \left(1 + \frac{z}{2} + \frac{3}{16} z^2\right) + \lambda y_* \left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)} \] 
\[ = y_* + \frac{(h\lambda)}{3} \left[ \frac{1 + \left(1 + \frac{z}{2}\right)^2}{1 + \left(1 + \frac{z}{2}\right)} + \frac{\left(1 + \frac{z}{2}\right)^2 + \left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2}{\left(1 + \frac{z}{2}\right) + \left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)} \right] \]
\[ + \frac{\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2 + \left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)^2}{\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right) + \left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)} \] 
\[ \text{(4.2.58)} \]

By substituting \( h\lambda = z \) in equation (4.2.58), we can show that
\[ y_{m} = y_{n} + y_{s} \left[ \frac{z + z(1 + \frac{z}{2})^2}{6 + \frac{3}{2} z} + \frac{z\left(1 + \frac{z}{2}\right)^2 + z\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2}{6 + 3z + \frac{9}{16} z^3} \right] \]

\[ y_{m} = y_{n} + y_{s} \left[ \frac{z\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2 + z\left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)^2}{6 + \frac{9}{2} z + \frac{27}{16} z^2 + \frac{27}{32} z^3} \right] \]

Dividing both sides of equation (4.2.59) by \( y_{n} \) we obtain

\[ \frac{y_{m} - y_{n}}{y_{s}} = 1 + \left[ \frac{z + z(1 + \frac{z}{2})^2}{6 + \frac{3}{2} z} + \frac{z\left(1 + \frac{z}{2}\right)^2 + z\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2}{6 + 3z + \frac{9}{16} z^3} \right] \]

\[ y_{m} = 1 + \left[ \frac{z\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2 + z\left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)^2}{6 + \frac{9}{2} z + \frac{27}{16} z^2 + \frac{27}{32} z^3} \right] \]

Following equation (3.3.31), we write equation (4.2.60) as

\[ Q = 1 + \left[ \frac{z + z(1 + \frac{z}{2})^2}{6 + \frac{3}{2} z} + \frac{z\left(1 + \frac{z}{2}\right)^2 + z\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2}{6 + 3z + \frac{9}{16} z^3} \right] \]

\[ z\left(1 + \frac{z}{2} + \frac{3}{16} z^2\right)^2 + z\left(1 + z + \frac{3}{8} z^2 + \frac{9}{32} z^3\right)^2 \]
\[ Q = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{19z^5}{768} + O(z^6). \quad (4.2.61) \]

Then we write equation (4.2.61) as the quadratic equation

\[ 2Q^2 - (2 + 2z + z^2)Q - \left( \frac{z^2}{3} \left( \frac{5z^4}{54} + \frac{115z^6}{1152} + \frac{121z^7}{2304} + \frac{23z^8}{1152} + \frac{19z^9}{4608} + \frac{361z^{10}}{294912} \right) \right) = 0. \]

\[ \ldots \quad (4.2.62) \]

The roots of equation (4.2.62) are

\[ Q = \frac{(384 + 384z + 192z^2) \pm (384 + 384z + 192z^2 + 128z^3 + 32z^4 + 19z^5)}{768} \]

\[ \ldots \quad (4.2.63) \]

To determine the stability region of the fourth order contraharmonic mean \((C_0 M)\) formula (4.2.63) in the complex plane that satisfy the condition in equation (3.3.31), i.e.,

\[ \left| \frac{(384 + 384z + 192z^2) + (384 + 384z + 192z^2 + 128z^3 + 32z^4 + 19z^5)}{768} \right| < 1 \]

\[ \ldots \quad (4.2.64) \]

and using Mathematica, we can plot the graphic surface defined by the formula (4.2.64) i.e.,

Figure 4.5: Graphic surface defined by equation (4.2.64)

and we can also plot the stability region defined by the formula (4.2.64) as shown in Figure 4.6.
The regions for which inequality (4.2.51), (4.2.56) and (4.2.64) are satisfied are shown in Figure 4.2, Figure 4.4 and Figure 4.6 which corresponds with the left half plane. Thus, from the stability definition the Contraharmonic mean \((C_o M)\) method are A-stable for all orders.

4.2.6 Numerical Example

We consider the initial value problem,

\[
y' = y, \quad y(0) = 1, \quad 0 \leq x \leq 1 \quad (4.2.65)
\]

where the exact solution is \(y(x) = \exp(x)\).

The numerical solution using formulae (4.2.18)-(4.2.19) compared with the exact solution are shown in Table 4.1. Table 4.2 illustrates the errors obtained by using the fourth order Harmonic Mean \((H_a M)\), the Geometric Mean (GM) and the classical Runge-Kutta formula or Arithmetic Mean (AM) methods.
Table 4.1: Errors of the Contraharmonic Mean formulae (4.2.18)-(4.2.19) method for solving (4.2.65)

<table>
<thead>
<tr>
<th>x</th>
<th>Exact Solution</th>
<th>Error (C,M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.110517E+01</td>
<td>.152133E-06</td>
</tr>
<tr>
<td>.20</td>
<td>.122140E+01</td>
<td>.336265E-06</td>
</tr>
<tr>
<td>.30</td>
<td>.134986E+01</td>
<td>.557446E-06</td>
</tr>
<tr>
<td>.40</td>
<td>.149182E+01</td>
<td>.821431E-06</td>
</tr>
<tr>
<td>.50</td>
<td>.164872E+01</td>
<td>.113478E-05</td>
</tr>
<tr>
<td>.60</td>
<td>.182212E+01</td>
<td>.150495E-05</td>
</tr>
<tr>
<td>.70</td>
<td>.201375E+01</td>
<td>.194043E-05</td>
</tr>
<tr>
<td>.80</td>
<td>.222554E+01</td>
<td>.245086E-05</td>
</tr>
<tr>
<td>.90</td>
<td>.245960E+01</td>
<td>.304720E-05</td>
</tr>
<tr>
<td>1.00</td>
<td>.271828E+01</td>
<td>.374186E-05</td>
</tr>
</tbody>
</table>

Table 4.2: Errors by using the various fourth order formulae for solving (4.2.65)

<table>
<thead>
<tr>
<th>x</th>
<th>Error (C,M)</th>
<th>Error (H,M)</th>
<th>Error (GM)</th>
<th>Error (AM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.152133E-06</td>
<td>.311267E-06</td>
<td>.190811E-06</td>
<td>.847423E-07</td>
</tr>
<tr>
<td>.20</td>
<td>.336265E-06</td>
<td>.688006E-06</td>
<td>.421759E-06</td>
<td>.187309E-06</td>
</tr>
<tr>
<td>.30</td>
<td>.557446E-06</td>
<td>.114055E-05</td>
<td>.699173E-06</td>
<td>.310514E-06</td>
</tr>
<tr>
<td>.40</td>
<td>.821431E-06</td>
<td>.168066E-05</td>
<td>.103027E-05</td>
<td>.457561E-06</td>
</tr>
<tr>
<td>.50</td>
<td>.113478E-05</td>
<td>.232178E-05</td>
<td>.142329E-05</td>
<td>.632103E-06</td>
</tr>
<tr>
<td>.60</td>
<td>.150495E-05</td>
<td>.307915E-05</td>
<td>.188757E-05</td>
<td>.838299E-06</td>
</tr>
<tr>
<td>.70</td>
<td>.194043E-05</td>
<td>.397015E-05</td>
<td>.243377E-05</td>
<td>.108087E-05</td>
</tr>
<tr>
<td>.80</td>
<td>.245086E-05</td>
<td>.501451E-05</td>
<td>.307398E-05</td>
<td>.136520E-05</td>
</tr>
<tr>
<td>.90</td>
<td>.304720E-05</td>
<td>.623463E-05</td>
<td>.382193E-05</td>
<td>.169738E-05</td>
</tr>
<tr>
<td>1.00</td>
<td>.374186E-05</td>
<td>.765592E-05</td>
<td>.469320E-05</td>
<td>.208432E-05</td>
</tr>
</tbody>
</table>
Table 4.3: No of arithmetic operations for using various formulas

<table>
<thead>
<tr>
<th>Method</th>
<th>Square root</th>
<th>Division</th>
<th>Multiply</th>
<th>Additions</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>GM</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>HaM</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>CoM</td>
<td>0</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 4.3 illustrate the number of arithmetic operations involved by using the various formulas for fourth order accuracy. From Table 4.2 it can be seen that the errors satisfy

\[ AM < C_eM < GM < H_aM \]

whilst the amount of work involved by the \( H_aM \) method is twice that of the AM method while the GM and \( CoM \) methods are approximately three times.

4.3 NEW RUNGE-KUTTA METHOD BASED ON THE CENTROIDAL MEAN \((C_eM)\) FORMULA

4.3.1 Third Order Centroidal Mean \((C_eM)\) Formula

Carrying out the procedure outlined in Section (4.2.2), a third order formula based on the centroidal mean \((C_eM)\) is established in the form

\[
y_{n+1} = y_n + \frac{h}{2} \left[ \frac{2(k_1^2 + k_{1n} + k_{11})}{3(k_1 + k_{11})} \right]
\]

(4.3.1)

by using the related Centroidal Mean

\[
C_eM = \frac{4(AM)^2 - (GM)^2}{3(AM)}
\]

(4.3.2)

Normally, we proceed to adjust the parameters \( a_i, 1 \leq i \leq 3 \) in (4.2.7) to achieve a high order accuracy in (4.3.1). Through the standard procedure of the adjustment of the parameters, a third order accurate formula is obtained from satisfying the equations of condition, i.e.,
\[ h^2 f_y' = 2a_1 + a_2 + a_3 = 2 \quad (4.3.3-i) \]
\[ h^3 f^2 f_{yy} = 3(a_2 + a_3)^2 + 6a_1^2 = 4 \quad (4.3.3-ii) \]
\[ h^3 f_y'^2 = 7a_1^2 + 8a_2a_3 + 5a_2a_3 + 2(a_2 + a_3)^2 = 8 \quad (4.3.3-iii) \]

We immediately obtain the parameters \( a_i, 1 \leq i \leq 3 \) in (4.3.3-i)-(4.3.3-iii) by the use of Mathematica. We find that
\[ a_1 = \frac{2}{3}, \quad a_2 = -\frac{2}{9}, \quad a_3 = \frac{8}{9} \]

4.3.2 New Fourth Order Centroidal Mean (C_eM) Formula

We now extend the procedure for the Centroidal Mean (C_eM) to obtain the fourth order formula in the form,
\[
y_{m+1} = y_m + \frac{h}{3} \left[ \frac{2(k_1^2 + k_2k_3 + k_4^2)}{3(k_1 + k_2)} + \frac{2(k_3^2 + k_2k_3 + k_4^2)}{3(k_3 + k_4)} + \frac{2(k_4^2 + k_4k_4 + k_4^2)}{3(k_4 + k_4)} \right] \]
\[ \ldots \quad (4.3.4) \]

Normally, we would substitute equations (4.2.10) into (4.3.4) to obtain an expression of \( y_{m+1} \). However, the fraction involved in (4.3.4) is not amenable to direct substitution, i.e.,
\[
\frac{k_1^2 + k_2k_3 + k_4^2}{k_1 + k_4}, \quad 1 \leq i \leq 3 \quad (4.3.5) \]

As shown in equation (4.2.11), the formula from (4.3.4) may be written similar to equation (4.2.13) with
\[
UPPER = \frac{2h}{9} \left[ (k_1^2 + k_2k_3 + k_4^2)(k_2 + k_3)(k_3 + k_4) + (k_2^2 + k_2k_3 + k_4^2)(k_1 + k_2)(k_3 + k_4) \right. \\
\left. + (k_3^2 + k_2k_3 + k_4^2)(k_1 + k_3)(k_2 + k_4) \right] \quad (4.3.6) \]

and equation (4.2.15). While the Taylor series expansion of \( y(x_{m+1}) \) is given in equation (3.3.8).

By using equations (3.3.8), (4.2.15) and (4.3.6), we can also write the equation similar to equation (4.2.16) and by comparing coefficient of similar terms in (3.3.8)
up to terms in \((h^4)\) yields the following equations of conditions

\[ h^{f'}f' = -2a_1 - 2s_2 - s_3 = 0 \]  (4.3.7-i)

\[ h^{f^2}f'' = -2a^2 - 2s^2_2 - s^2_3 + 2 = 0 \]  (4.3.7-ii)

\[ h^{f'}f'' : -12a_1a_4 - 6a_1s_2 - 22a_4s_2 - 12a_3s_2 - 10s_3s_2 \]
\[ -14a_1^2 - 14s_2^2 + 18s_2 + 9s_3 - 4s_3^2 + 6 = 0 \]  (4.3.7-iii)

\[ h^{f^3}f''' = -4a^3 - 4s_2^2 - 2s_3^2 + 3 = 0 \]  (4.3.7-iv)

\[ h^{f'}f''' : -12a_3a_5 - 56a_3s_2 - 20a_5s_2 - 20a_5s_3 \]
\[ -16a_5s_3 - 21a_7s_3 - 16a_8s_3 - 44a_7a_5 - 24a_7^2a_5 \]
\[ -20a_3s^2_2 - 25a_3s^2_2 - 25a_3^2s_2 - 17a_7^2s_3 - 8s^3_2s_3 - 8a_3^2 \]
\[ + 18a_3s_3 + 9s_3s_3 + 36a_3a_5 + 18a_3a_5 + 18a_5s_2 + 27a_7s_2 \]
\[ + 9s_2^2 + 9a_3^2 + 12a_4 + 12a_2 - 4s_2s^2_2 + 6s_3 - 8a_3^2 - 8s_3^2 + 3 = 0 \]  (4.3.7-v)

\[ h^{f^2}f''': -24a_3a_5 - 24a_5s_2 - 12a_5s_3 - 12a_5s_3 - 12a_3^2a_5 - 6a_3^2a_5 \]
\[ -6a_3s^2_2 - 22a_3s^2_2 - 12a_3s^2_2 - 12a_3s^2_2 - 10s_3^2s_2 \]
\[ -10s_3^2s_3 - 28s_3^2 - 28a_3^2 - 8s_3^2 + 18a_3^2 + 12a_2 + 12a_4 \]
\[ + 6s_3 + 9s_3^2 + 12 = 0 \]  (4.3.7-vi)

Equations (4.3.7-i)-(4.3.7-vi) are then solved simultaneously by Mathematica to give the required parameters, i.e.,

\[ a_1 = \frac{1}{2}, a_2 = \frac{1}{24}, a_3 = \frac{11}{24}, a_4 = \frac{11}{132}, a_5 = -\frac{25}{132} \text{ and } a_6 = \frac{73}{66} \]

Thus, this new method can be written as follows

\[ k_1 = f(x_1, y_1) \]

\[ k_2 = f(x_1 + \frac{h}{2}, y_1 + \frac{1}{2}hk_1) \]

\[ k_3 = f(x_1 + \frac{h}{2}, y_1 + \frac{1}{2}hk_1 + \frac{11}{24}hk_2) \]  (4.3.8)

\[ k_4 = f(x_1 + h, y_1 + \frac{1}{12}hk_1 - \frac{25}{132}hk_2 + \frac{73}{66}hk_3) \]
By substituting the values $a_i, 1 \leq i \leq 6$ into equation (4.3.8)-(4.3.9) using Mathematica and evaluating all the terms up to $(h')^2$ to represent the local truncation error for this method, we have the result

$$
LTE = \frac{h'}{207360} \left[248f_{y-} - 2727f_{y-}^2f_{y+} + 72f_{y-}^3 - 84f_{y-}f_{y+} - 72f_{y-}^4\right]
$$

### 4.3.4 Stability Analysis

We examine the stability region of the fourth order method for solving the initial value problem with the test equation $y' = \lambda y$, and we obtain

$$
k_i = \lambda\left(y_n + \frac{1}{2}hk_i\right)
$$

$$
k_3 = \lambda\left(y_n + \frac{1}{2}hk_i + \frac{11}{24}hk_3\right)
$$

$$
k_4 = \lambda\left(\frac{1}{12}hk - \frac{25}{132}hk_2 + \frac{73}{66}hk_3\right)
$$

By substituting $k_i, 1 \leq i \leq 6$ in (4.3.11) into (4.3.9) the new fourth order formula, we obtain

$$
y_{n+1} = y_n + (h\lambda)y + \frac{1}{2}(h\lambda)^2y + \frac{1}{6}(h\lambda)^3y + \frac{1}{24}(h\lambda)^4y + \frac{37}{5184}(h\lambda)^5y
$$

By substituting $h\lambda = z$ in (4.3.12), we can show that

$$
y_{n+1} = y_n + y_n\left[z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{5184}\right] + 0(z^6)
$$

We obtain a one step difference equation of the form...
\[ \frac{y_{n+1}}{y_n} = Q \]

and from equation (4.3.13), obtain

\[ Q = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 + \frac{37}{5184} z^5 + O(z^6). \]  

(4.3.14)

We now determine the stability region of this fourth order formula in the complex plane that satisfy the condition as in equation (3.3.31), i.e.,

\[ \left| 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{37z^5}{5184} \right| < 1. \]  

(4.3.15)

By the use of Mathematica, we can plot the graphic surface defined by equation (4.3.15) i.e.,

![Graphic surface defined by fourth order formula](image)

**Figure 4.7: Graphic surface defined by fourth order $C,J \text{M}$ method**

and we can also plot the stability region defined by the formula in equation (4.3.15) as shown in Figure 4.8.
4.3.5 Numerical Example

We solve the initial value problem

\[ y' = -\sqrt{1-y^2}, \quad y(0.1) = \cos(0.1), \quad 0.1 \leq x \leq 1.0 \quad (4.3.16) \]

by using the step length \( h = 0.01 \) where the exact solution is \( y(x) = \cos(x) \).

The numerical solutions are printed every ten steps using formulae (4.3.8)-(4.3.9) and compared with the exact solution as shown in Table 4.4. Table 4.5 illustrate the errors obtained by using the fourth order Contraharmonic Mean (\( C_0 M \)), the Harmonic Mean (\( H_a M \)) and the classical Runge-Kutta formula or Arithmetic Mean (AM) methods.
Table 4.4: The errors in the Centroidal Mean formulae (4.3.8)-(4.3.9) method for solving (4.3.16)

<table>
<thead>
<tr>
<th>x</th>
<th>Exact Solution</th>
<th>Error (C\textsubscript{e}M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.995004E+00</td>
<td>.000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.980067E+00</td>
<td>.101016E-07</td>
</tr>
<tr>
<td>.30</td>
<td>.955336E+00</td>
<td>.165567E-07</td>
</tr>
<tr>
<td>.40</td>
<td>.921061E+00</td>
<td>.222985E-07</td>
</tr>
<tr>
<td>.50</td>
<td>.877583E+00</td>
<td>.276548E-07</td>
</tr>
<tr>
<td>.60</td>
<td>.825336E+00</td>
<td>.326692E-07</td>
</tr>
<tr>
<td>.70</td>
<td>.764842E+00</td>
<td>.373258E-07</td>
</tr>
<tr>
<td>.80</td>
<td>.696707E+00</td>
<td>.415924E-07</td>
</tr>
<tr>
<td>.90</td>
<td>.621610E+00</td>
<td>.454337E-07</td>
</tr>
<tr>
<td>1.00</td>
<td>.540302E+00</td>
<td>.488152E-07</td>
</tr>
</tbody>
</table>

Table 4.5: Comparison of the errors by using the various order formulae for solving (4.3.16)

<table>
<thead>
<tr>
<th>x</th>
<th>Error (C\textsubscript{e}M)</th>
<th>Error (C\textsubscript{e}M)</th>
<th>Error (AM)</th>
<th>Error (H,M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.000000E+00</td>
<td>.000000E+00</td>
<td>.000000E+00</td>
<td>.000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.436823E-09</td>
<td>.101016E-07</td>
<td>.138674E-07</td>
<td>.249999E-07</td>
</tr>
<tr>
<td>.30</td>
<td>.714333E-09</td>
<td>.165567E-07</td>
<td>.227342E-07</td>
<td>.409564E-07</td>
</tr>
<tr>
<td>.40</td>
<td>.976564E-09</td>
<td>.222985E-07</td>
<td>.306209E-07</td>
<td>.551381E-07</td>
</tr>
<tr>
<td>.50</td>
<td>.122684E-08</td>
<td>.276548E-07</td>
<td>.379791E-07</td>
<td>.683668E-07</td>
</tr>
<tr>
<td>.60</td>
<td>.146287E-08</td>
<td>.326692E-07</td>
<td>.448692E-07</td>
<td>.807537E-07</td>
</tr>
<tr>
<td>.70</td>
<td>.168219E-08</td>
<td>.373258E-07</td>
<td>.512690E-07</td>
<td>.922598E-07</td>
</tr>
<tr>
<td>.80</td>
<td>.188261E-08</td>
<td>.415924E-07</td>
<td>.571343E-07</td>
<td>.102806E-06</td>
</tr>
<tr>
<td>.90</td>
<td>.206222E-08</td>
<td>.454337E-07</td>
<td>.624160E-07</td>
<td>.112304E-06</td>
</tr>
<tr>
<td>1.00</td>
<td>.221936E-08</td>
<td>.488152E-07</td>
<td>.670664E-07</td>
<td>.120668E-06</td>
</tr>
</tbody>
</table>

It can be seen that the two new formula Contraharmonic Mean (C\textsubscript{p}M) and the Centroidal Mean (C\textsubscript{e}M) perform better than the classical Arithmetic Mean Runge-Kutta formula in terms of accuracy for this particular problem.
Table 4.6: Comparison of the parameters $a_i, 1 \leq i \leq 6$ in (4.2.10) for use in the various formulas

<table>
<thead>
<tr>
<th></th>
<th>$C_0 M$</th>
<th>$C_e M$</th>
<th>$AM$</th>
<th>$GM$</th>
<th>$H_0 M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$132a_1$</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
</tr>
<tr>
<td>$132a_2$</td>
<td>16.5</td>
<td>5.5</td>
<td>0</td>
<td>-8.25</td>
<td>-16.5</td>
</tr>
<tr>
<td>$132a_3$</td>
<td>49.5</td>
<td>60.5</td>
<td>66</td>
<td>74.25</td>
<td>82.5</td>
</tr>
<tr>
<td>$132a_4$</td>
<td>33</td>
<td>11</td>
<td>0</td>
<td>-16.5</td>
<td>-33</td>
</tr>
<tr>
<td>$132a_5$</td>
<td>-99</td>
<td>-25</td>
<td>0</td>
<td>27.5</td>
<td>46.2</td>
</tr>
<tr>
<td>$132a_6$</td>
<td>198</td>
<td>146</td>
<td>132</td>
<td>121</td>
<td>118.8</td>
</tr>
</tbody>
</table>

In Evans and Yaakub [1995] and Wazwaz [1993] it was shown that for $k_i \neq k_{n-i}$, where

$$C_e M > RM > C_0 M > AM > H_0 M > GM > H_0 M$$ (4.3.17)

From Table 4.5 it can be seen that the errors satisfy

$$C_e M < C_0 M < AM < H_0 M$$ (4.3.18)

Comparison of the parameters $a_i, 1 \leq i \leq 6$, for all types of means, show that of the parameters, $a_1$ is fixed, $a_2, a_4, a_6$ are decreasing and $a_3, a_5$ are increasing following equation (4.3.17).

Furthermore, we see that

$$a_i = a_2 + a_3 = \frac{1}{2}$$

and

$$a_4 + a_5 + a_6 = 1.$$ 

These can be found in Table 4.6 and Table 4.7 where we show our results for the parameters $a_i, 1 \leq i \leq 6$.  

98
4.4 NEW RUNGE-KUTTA METHOD BASED ON THE ROOT-MEAN-SQUARE (RMS) FORMULA

In Eves [1983] and Wazwaz [1993] another formula may be developed by using the Root-Mean-Square (RM), i.e.,

\[
RMS = \sqrt{2(AM)^2 - (GM)^2}
\]  

Thus by replacing the Arithmetic Mean (AM) in (4.4.1) and Geometric Mean (GM) of the form \(GM = \sqrt{k_i k_{i+1}}\) by Root-Mean-Square (RMS), we obtain a new third order Root-Mean-Square formula of the form

\[
y_{i+1} = y_i + \frac{h}{2} \left( \sqrt{\frac{k_1^2 + k_2^2}{2}} + \sqrt{\frac{k_3^2 + k_4^2}{2}} \right)
\]  

4.4.1 Third Order Root-Mean-Square (RMS) Formula

Now we attempt to find the values of the parameters \(a_i, 1 \leq i \leq 3\) from (4.2.7) to make formula (4.4.2) of third order accuracy. To calculate the right hand side (RHS) in equation (4.4.2) we use a binomial expansion with the help of Mathematica. The binomial expansion of \((1+x)^{\frac{1}{2}}\) is given as

\[
C_0 M, C_e M, AM, GM, H_a M
\]

<table>
<thead>
<tr>
<th>(a_i)</th>
<th>(\frac{1}{2})</th>
<th>(\frac{1}{2})</th>
<th>(\frac{1}{2})</th>
<th>(\frac{1}{2})</th>
<th>(\frac{1}{2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_2)</td>
<td>(\frac{1}{8})</td>
<td>(\frac{1}{24})</td>
<td>0</td>
<td>(-\frac{1}{8})</td>
<td>(-\frac{1}{8})</td>
</tr>
<tr>
<td>(a_3)</td>
<td>(\frac{3}{8})</td>
<td>(\frac{11}{24})</td>
<td>(\frac{9}{16})</td>
<td>(\frac{5}{8})</td>
<td>(\frac{5}{8})</td>
</tr>
<tr>
<td>(a_4)</td>
<td>(\frac{1}{4})</td>
<td>(\frac{11}{32})</td>
<td>0</td>
<td>(-\frac{1}{4})</td>
<td>(-\frac{1}{4})</td>
</tr>
<tr>
<td>(a_5)</td>
<td>(-\frac{1}{4})</td>
<td>(-\frac{1}{12})</td>
<td>0</td>
<td>(\frac{5}{24})</td>
<td>(\frac{7}{20})</td>
</tr>
<tr>
<td>(a_6)</td>
<td>(\frac{3}{2})</td>
<td>(\frac{73}{66})</td>
<td>(\frac{11}{12})</td>
<td>(\frac{9}{10})</td>
<td>(\frac{9}{10})</td>
</tr>
</tbody>
</table>

Table 4.7: The values of parameters \(a_i, 1 \leq i \leq 6\) based on the various formulas.
(1 + x)^\frac{1}{2} = 1 + \frac{1}{2} - \frac{1}{8} x + \frac{1}{16} x^2 - \frac{5}{128} x^3 + \ldots \quad (4.4.3)

First we write \( \sqrt{\frac{k_1^2 + k_2^2}{2}} \) in the form of

\[ f(1 + x)^\frac{1}{2} \quad (4.4.4) \]

where

\[ x = \frac{k_1^2 + k_2^2}{2f^2} - 1. \quad (4.4.5) \]

With \( x \) given by equation (4.4.5) and substituting in equation (4.4.3) we can evaluate (4.4.4) to obtain the values of \( \sqrt{\frac{k_1^2 + k_2^2}{2}} \) given in powers of \( h \). From this step, we take \( h \) only to third order. With a similar technique, we can also obtain the values of \( \sqrt{\frac{k_1^2 + k_2^2}{2}} \).

By comparing equation (4.4.2) with the Taylor series expansion for \( y(x,\alpha) \) in equation (3.3.8), we obtain the following three equations of conditions, i.e.,

\[
\begin{align*}
    h^2 f f_x & : \quad 2a_1 + a_2 + a_3 = 2 \quad (4.4.6-i) \\
    h^3 f f_{xy} & : \quad 6a_1^2 + 3(a_2 + a_3)^2 = 4 \quad (4.4.6-ii) \\
    h^3 f_{xx} & : \quad 6a_1^2 + 12a_1a_2 - 6a_1(a_1 + a_2) + 3(a_2 + a_3)^2 = 8. \quad (4.4.6-iii)
\end{align*}
\]

By solving the above three equations simultaneously by Mathematica, we obtain the values of \( a_i, 1 \leq i \leq 3 \), i.e.,

\[ a_1 = \frac{2}{3}, \quad a_2 = -\frac{1}{6}, \quad a_3 = \frac{5}{6} \]

Thus, the new Root-Mean-Square (RMS) method can be written as follows,

\[ k_1 = f(x_s, y_s) \]

\[ k_2 = f(x_s + \frac{2}{3} h, y_s + \frac{2}{3} hk_1) \quad (4.4.7) \]

\[ k_3 = f(x_s + \frac{2}{3} h, y_s - \frac{1}{6} hk_1 + \frac{5}{6} hk_2) \]

and
\[ y_{n+1} = y_n + \frac{h}{2} \left( \sqrt{k_1^2 + k_2^2} + \sqrt{k_1^2 + k_2^2} \right) \]  
\hspace{1cm} (4.4.8)

By substituting the values of \( a_1 = \frac{2}{3}, a_2 = -\frac{1}{6}, a_3 = \frac{5}{6} \) in (4.4.8) and again in Mathematica exhibiting all the terms up to \( h^4 \) to represent the local truncation error (LTE) for this method, we have the result is given by,

\[ LTE = \frac{h^4}{20736} \left[ 244 f^3 f_{x_{,x}} + 630 f^2 f_{x_{,x}} + 1317 f f_{x_{,x}} \right], \]

to achieve third order accuracy.

We solve the initial value problem

\[ y' = y, \quad y(0) = 1 \quad \text{in} \quad 0 \leq x \leq 1 \]

by using \( h = 0.1 \) and we compare the third order Runge-Kutta formula with the new formula (4.4.8). The results are shown in Table 4.8.

Table 4.8: The comparison of errors with AM and RMS third order for solving \( y' = y \) with exact solution \( y(x) = \exp(x) \)

<table>
<thead>
<tr>
<th>x</th>
<th>Exact Solution</th>
<th>Error (AM)</th>
<th>Error (RMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>.110517E+01</td>
<td>.425133E-05</td>
<td>.497404E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>.122140E+01</td>
<td>.939687E-05</td>
<td>.109943E-04</td>
</tr>
<tr>
<td>0.3</td>
<td>.134986E+01</td>
<td>.155777E-04</td>
<td>.182256E-04</td>
</tr>
<tr>
<td>0.4</td>
<td>.149182E+01</td>
<td>.229546E-04</td>
<td>.268568E-04</td>
</tr>
<tr>
<td>0.5</td>
<td>.164872E+01</td>
<td>.317109E-04</td>
<td>.371017E-04</td>
</tr>
<tr>
<td>0.6</td>
<td>.182212E+01</td>
<td>.420551E-04</td>
<td>.492043E-04</td>
</tr>
<tr>
<td>0.7</td>
<td>.201375E+01</td>
<td>.542244E-04</td>
<td>.634422E-04</td>
</tr>
<tr>
<td>0.8</td>
<td>.222554E+01</td>
<td>.684881E-04</td>
<td>.801307E-04</td>
</tr>
<tr>
<td>0.9</td>
<td>.245960E+01</td>
<td>.851523E-04</td>
<td>.996276E-04</td>
</tr>
<tr>
<td>1.0</td>
<td>.271828E+01</td>
<td>.104564E-03</td>
<td>.122339E-03</td>
</tr>
</tbody>
</table>
4.4.2 New Fourth Order Root-Mean Square (RMS) Formula

By substituting the Arithmetic Mean (AM) in (4.1.1) with the Root-Mean-Square (RMS), we obtain a new formula of the form

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \sqrt{\left( \frac{k_1^2 + k_2^2}{2} \right)} + \sqrt{\left( \frac{k_3^2 + k_4^2}{2} \right)} + \sqrt{\left( \frac{k_5^2 + k_6^2}{2} \right)} \right] \quad (4.4.9) \]

By writing \( k_i^2 \) in the form,

\[ k_i^2 = \left[ f + ha_if_i + \frac{1}{2} h^2 a_i^2 f_i^2 f_i + \ldots \right]^2 \]

we can derive,

\[ k_1^2 + k_2^2 = f^2 + f^2 \left[ 1 + ha_if_i + \frac{1}{2} h^2 a_i^2 f_i^2 f_i + \ldots \right]^2 \]

\[ = \frac{f^2}{2} + 2ha_if_i f_i + h^2 a_i^2 f_i^2 f_i + \ldots \]

\[ \therefore \frac{k_1^2 + k_2^2}{2} = f^2 + ha_if_i^2 f_i + \frac{1}{2} h^2 a_i^2 f_i^3 f_i + \ldots \]

\[ = f^2 \left[ 1 + ha_if_i + \frac{1}{2} h^2 a_i^2 f_i^2 f_i + \ldots \right] \quad (4.4.10) \]

Then,

\[ \frac{k_1^2 + k_2^2}{2} = f \left[ 1 + ha_if_i + \frac{1}{2} h^2 a_i^2 f_i^2 f_i + \ldots \right]^2 \quad (4.4.11) \]

or

\[ \sqrt{\frac{k_1^2 + k_2^2}{2}} = f (1 + \nu)^{\frac{1}{2}} \quad (4.4.12) \]

where \( \nu = ha_if_i + \frac{1}{2} h^2 a_i^2 f_i^2 f_i + \ldots \) \quad (4.4.13)

From equation (4.4.12), we can show that,

\[ \nu = \frac{k_1^2 + k_2^2}{2f^2} - 1 \quad (4.4.14) \]

By using the Binomial expansion in the form
\[(1 + \nu)^{\frac{1}{2}} = 1 + \frac{1}{2} \nu - \frac{1}{8} \nu^2 + \frac{1}{16} \nu^3 - \frac{5}{128} \nu^4 + \ldots\]  

(4.4.15)

We can finally obtain,

\[f(1 + \nu)^{\frac{1}{2}} = \sqrt{\frac{k^2 + k_2^2}{2}}\]  

(4.4.16)

in terms of \(f, f', f'', \ldots, h, h^2, h^3 \ldots\ etc\)

We can obtain similar results for the terms \(\sqrt{\frac{k^2 + k_2^2}{2}}\) and \(\sqrt{\frac{k^2 + k_2^2}{2}}\). By substituting into equation (4.4.9) we obtain an equation of the form

\[y_{n+1} = y_n + h f + \frac{1}{6} h^2 (2a_1 + 2s_1 + s_1) f' f' + \frac{1}{12} h^3 (2a_1^2 + 2s_1^2 + s_1^2) f'' f'' + \ldots\]  

(4.4.17)

By comparing equation (3.3.8) and (4.4.17) the first two terms are satisfied but the remaining terms yield the following six equations of conditions (putting \(a_1 + a_2 = s_1\) and \(a_4 + a_5 + a_6 = s_1\)), i.e.,

\[h^2 f' f' = 2a_1 + 2s_1 + s_1 = 3\]  

(4.4.18-i)

\[h^3 f' f'' = 2a_1^2 + 2s_1^2 + s_1^2 = 2\]  

(4.4.18-ii)

\[h^2 f' f'' = 8a_4 a_5 + 4a_6 a_7 + 4a_8 s_8 + 2s_7^2 - 2s_7 s_8 + s_8^2 + 2a_4^2 - 2a_7 s_7 = 4\]  

(4.4.18-iii)

\[h^4 f' f'' f' = 4a_1 a_5 + 2a_2 a_7 + 8a_3 a_8 + 2a_4 s_8 + 2s_7^2 + 2a_5 a_8 s_8 + 4a_4 a_8\]  

\[+ 4a_5 s_7^2 - s_7 s_8^2 + s_7^2 + 2a_1^3 - a_5^2 s_7 - a_7 s_7^2 = 4\]  

(4.4.18-iv)

\[h^4 f' f'' f' = 4a_1^2 + 4s_1^2 + 2s_1^2 = 3\]  

(4.4.18-v)

\[h^4 f' f'' f' = 8a_4 a_6 a_8 + 8a_5 a_7 s_7 - 4a_6 a_7 s_7 - 4a_7 s_7^2 - 2s_7^3 - 4a_8 s_8 + 4a_4 a_8 s_8\]  

\[+ 4a_5 s_7^2 + s_7^2 s_8 + s_7 s_8^2 - s_7^3 - 2a_1^4 - 4a_5^2 a_7 + a_7^2 s_7 + a_7^2 s_7 = 2\]  

(4.4.18-vi)

Solving equations (4.4.18-i)-(4.4.18-vi) by Mathematica, we immediately obtain the values

\[a_1 = \frac{1}{2}, a_2 = \frac{1}{16}, a_3 = \frac{7}{16}, a_4 = \frac{1}{8}, a_5 = -\frac{17}{56}, a_6 = \frac{33}{28}\]
Therefore the Root-Mean-Square (RMS) formula in equation (4.4.9) can now be written as

\[ k_i = f(x_i, y_i) \]

\[ k_2 = f(x + \frac{h}{2}, y + \frac{1}{2}hk_y) \]

\[ k_3 = f(x + \frac{h}{2}, y + \frac{1}{16}hk_y + \frac{7}{16}hk_y) \]

\[ k_4 = f(x + h, y + \frac{1}{8}hk_y - \frac{17}{56}hk_y + \frac{33}{28}hk_y) \]

and

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \sqrt{\left( \frac{k_1^2 + k_2^2}{2} \right)} + \sqrt{\left( \frac{k_3^2 + k_4^2}{2} \right)} + \sqrt{\left( \frac{k_1^2 + k_4^2}{2} \right)} \right] \]

(4.4.20)

to achieve fourth order accuracy.

By substituting the values of \( a_i, 1 \leq i \leq 6 \) again in Mathematica and showing all the terms up to \( (h^5) \) to represent the local truncation error (LTE) for this method we have

\[ LTE = \frac{h^5}{184320} \left[ -429f'f' - 2454f^2f''f''' - 96f^3f''' - 48f^4f'''f''' - 64f^4f'''' \right] \]

\( \ldots \) (4.4.21)

### 4.4.3 Numerical Example

We consider the IVP,

\[ y' = y, \quad y(0) = 1, \quad 0 \leq x \leq 1 \]

(4.4.22)

where the exact solution is \( y(x) = \exp(x) \).

The error in the numerical solution using formulae (4.4.19)-(4.4.20) compared with the third order Root-Mean-Square (RMS) in Section (4.4.1) and the exact solution are shown in Table 4.9. While Table 4.10 shows the errors obtained by the fourth order Contraharmonic Mean (\( C_p M \)), the Centroidal Mean (\( C_e M \)), the Geometric Mean (GM) and the classical Runge-Kutta Arithmetic Mean (AM) formulas.
Table 4.9: Error in the Root-Mean-Square formula
(4.4.19)-(4.4.20) method for solving (4.4.22)

<table>
<thead>
<tr>
<th>x</th>
<th>Exact Solution</th>
<th>Error (RMS)</th>
<th>Error (RMS3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.110517E+01</td>
<td>0.1785807E-07</td>
<td>0.497404E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>0.122140E+01</td>
<td>0.3947243E-07</td>
<td>0.109943E-04</td>
</tr>
<tr>
<td>0.3</td>
<td>0.134986E+01</td>
<td>0.6543568E-07</td>
<td>0.182256E-04</td>
</tr>
<tr>
<td>0.4</td>
<td>0.149182E+01</td>
<td>0.9642348E-07</td>
<td>0.268568E-04</td>
</tr>
<tr>
<td>0.5</td>
<td>0.164872E+01</td>
<td>0.1332055E-06</td>
<td>0.371017E-04</td>
</tr>
<tr>
<td>0.6</td>
<td>0.182212E+01</td>
<td>0.1766579E-06</td>
<td>0.492043E-04</td>
</tr>
<tr>
<td>0.7</td>
<td>0.201375E+01</td>
<td>0.2277766E-06</td>
<td>0.634422E-04</td>
</tr>
<tr>
<td>0.8</td>
<td>0.222554E+01</td>
<td>0.2876939E-06</td>
<td>0.801307E-04</td>
</tr>
<tr>
<td>0.9</td>
<td>0.245960E+01</td>
<td>0.3576947E-06</td>
<td>0.996276E-04</td>
</tr>
<tr>
<td>1.0</td>
<td>0.271828E+01</td>
<td>0.4392376E-06</td>
<td>0.122339E-03</td>
</tr>
</tbody>
</table>

Table 4.10: Errors by using the various fourth order formulae for solving (4.4.22)

<table>
<thead>
<tr>
<th>x</th>
<th>Error (RMS)</th>
<th>Error (C,M)</th>
<th>Error (C,M)</th>
<th>Error (AM)</th>
<th>Error (GM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.17858E-07</td>
<td>0.15965E-07</td>
<td>0.15213E-06</td>
<td>0.84742E-07</td>
<td>0.19081E-06</td>
</tr>
<tr>
<td>0.2</td>
<td>0.39472E-07</td>
<td>0.35288E-07</td>
<td>0.33626E-06</td>
<td>0.18730E-06</td>
<td>0.42175E-06</td>
</tr>
<tr>
<td>0.3</td>
<td>0.65435E-07</td>
<td>0.58500E-07</td>
<td>0.55744E-06</td>
<td>0.31051E-06</td>
<td>0.69917E-06</td>
</tr>
<tr>
<td>0.4</td>
<td>0.96423E-07</td>
<td>0.86203E-07</td>
<td>0.82143E-06</td>
<td>0.45756E-06</td>
<td>0.10302E-05</td>
</tr>
<tr>
<td>0.5</td>
<td>0.13320E-06</td>
<td>0.11908E-06</td>
<td>0.11347E-06</td>
<td>0.63210E-06</td>
<td>0.14232E-05</td>
</tr>
<tr>
<td>0.6</td>
<td>0.17665E-06</td>
<td>0.15793E-06</td>
<td>0.15049E-05</td>
<td>0.83829E-06</td>
<td>0.18875E-05</td>
</tr>
<tr>
<td>0.7</td>
<td>0.22777E-06</td>
<td>0.20363E-06</td>
<td>0.19404E-05</td>
<td>0.10808E-05</td>
<td>0.24337E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>0.28769E-06</td>
<td>0.25720E-06</td>
<td>0.24508E-05</td>
<td>0.13652E-05</td>
<td>0.30739E-05</td>
</tr>
<tr>
<td>0.9</td>
<td>0.35769E-06</td>
<td>0.31978E-06</td>
<td>0.30472E-05</td>
<td>0.16973E-05</td>
<td>0.38219E-05</td>
</tr>
<tr>
<td>1.0</td>
<td>0.43923E-06</td>
<td>0.39268E-06</td>
<td>0.37418E-05</td>
<td>0.20843E-05</td>
<td>0.46932E-05</td>
</tr>
</tbody>
</table>

From Table 4.9 it can be noted that the Root-Mean-Square (RMS) gives the smallest error for the fourth order method compared to the third order formula. From Table 4.10, it can be seen that the errors satisfy

\[ C,M < RMS < AM < C,M < GM \]
4.5 THE IMPLICIT RUNGE-KUTTA METHODS

All the methods discussed so far in the previous chapters have been explicit. Following Sanugi [1986] in the search for reliable efficient one step methods for solving general ODE problems, explicit Runge-Kutta methods are sometimes avoided for two reasons. The first is that the computational cost, particularly as measured in terms of derivative evaluations increases rapidly as high order requirements are imposed. The second reason is specific to stiff problems and is concerned with the stability properties of these methods. Butcher [1987] gives six basic reasons for taking a serious interest in implicit Runge-Kutta methods as follows

a) Higher orders of accuracy can be obtained than for explicit methods.
b) For linear systems of differential equations implicit methods can be implemented explicitly.
c) For stiff problems explicit methods are never satisfactory whereas some implicit methods are.
d) Implicit methods are closely related to Rosenbrock methods.
e) The structure of certain high-order explicit methods can be derived directly from some related implicit methods.
f) Implicit methods have an algebraic nicety not possessed by explicit methods in that the set of implicit methods make a very natural operation homomorphic to a certain group whereas the subset corresponding to explicit methods is only a semigroup.

As pointed out by Butcher [1963], it is also possible to consider implicit Runge-Kutta methods. The general R-stage implicit Runge-Kutta method is defined by

$$y_{n+i} = y_n + h\Phi(x_n, y_n, h)$$

(4.5.1-i)

where $$\Phi(x_n, y_n, h) = \sum_{r=1}^{R} c_r k_r.$$
and  
\[ k_r = f\left(x + h\alpha_r, y + h\sum_{i=1}^{s} b_i k_i\right), \quad r = 1, 2, \ldots, R \]  
(4.5.1-ii)

\[ a_r = \sum_{i=1}^{s} b_i, \quad r = 1, 2, \ldots, R. \]  
(4.5.1-iii)

The equations (4.5.1-i)-(4.5.1-iii) are implicit if there is at least one \( b_i \neq 0 \) for \( s > r \) so that at least one \( k_r \) is defined implicitly. We can also write the equation (4.5.1-i)-(4.5.1-iii) as \( S \)-stage implicit methods with the Butcher array in the form

\[
\begin{array}{c|cccc}
\cdot & a_{11} & a_{12} & \cdots & a_{1s} \\
\cdot & a_{21} & a_{22} & \cdots & a_{2s} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\cdot & \cdot & \cdot & \cdots & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
c_{s} & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
b_{1} & b_{2} & \cdots & b_{s}
\end{array}
\]

Butcher [1963] has shown that for any \( R \geq 2 \) there exists an \( R \)-stage implicit Runge Kutta method of order \( 2R \). For example, by setting \( R=2 \) the following 2-stage scheme of Hammer and Hollingsworth [1955] has order four and is defined by

\[ y_{n+1} = y_n + \frac{h}{2} \left[ k_1 + k_2 \right] \]  
(4.5.2)

where

\[ k_1 = f\left(x_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right) h, y_n + \frac{1}{4} h k_1 + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right) h k_2\right) \]

\[ k_2 = f\left(x_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right) h, y_n + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right) h k_1 + \frac{1}{4} h k_2\right) \]

The fourth order implicit Runge-Kutta method in (4.5.2) can also be written in the Butcher array form as
The main advantage of implicit schemes is their improved stability characteristics. (See, for example, Lambert [1973], p.155). Lambert [1973] has also shown that for the fourth-order scheme in equation (4.5.2)

\[
\frac{y_{n+1}}{y_n} = 1 + \frac{1}{4} (h \lambda) + \frac{1}{12} (h \lambda)^2
\]

(4.5.3)

This is the fourth order (2,2) Padé approximation to \(\exp(h \lambda)\). From equation (4.5.3) the interval of absolute stability is \((-\infty, 0)\) compared with the stability interval of \((-2.78, 0)\) for the 4-stage fourth order explicit Runge-Kutta methods.

A somewhat less formidable problem arises in the case when \(b_n = 0\) for \(r < s\), the resulting methods which are termed 'semi-explicit' by Butcher. (See, Lambert [1973], p.154). Semi-explicit Runge-Kutta methods have \(b_n = 0\) for \(r < s\) and when \(b_n = 0\) for \(r \leq s\) the method is explicit. An example of a 3-stage fourth order R-stage semi-explicit method quoted by Butcher (See, Lambert [1973], p.154) is

\[
y_{n+1} = y_n + \frac{h}{6} (k_1 + 4k_2 + k_3)
\]

(4.5.4)

where

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + \frac{h}{2}, y_n + \frac{1}{4} hk_1 + \frac{1}{4}hk_2)
\]

\[
k_3 = f(x_n + h, y_n + hk_2)
\]

or written in the Butcher array form as
The method in equation (4.5.4) has a larger stability interval than the four stage fourth order explicit Runge-Kutta methods and easily shown by Noye [1984] as

\[
\frac{y_{n+1}}{y_n} = r(h\lambda) = \frac{1 + \frac{1}{4}(h\lambda) + \frac{1}{4}(h\lambda)^2 + \frac{1}{6}(h\lambda)^3}{1 - \frac{1}{4}(h\lambda)}
\] (4.5.5)

This is the fourth order (3,1) Pade approximation to \( \exp(h\lambda) \) and the interval of absolute stability is \((-5.41,0)\) compared with \((-2.78,0)\) for the explicit Runge-Kutta methods.

4.5.1 2-Stage Implicit Arithmetic Mean Runge-Kutta Method

The development of the implicit Runge-Kutta formulae or the implicit classical Runge-Kutta formulae is based on the concept of the arithmetic mean (AM) as shown by the implicit AM Runge-Kutta method proposed by Hammer and Hollingsworth [1955]. We consider here only the two-stage method given by the following relations

\[
k_1 = f(x_n + (a_{11} + a_{12})h, y_n + h a_{11} k_1 + h a_{12} k_2)
\] (4.5.6-i)
\[
k_2 = f(x_n + (a_{21} + a_{22})h, y_n + h a_{21} k_1 + h a_{22} k_2)
\] (4.5.6-ii)
\[
y_{n+1} = y_n + h \left( \frac{k_1 + k_2}{2} \right)
\] (4.5.6-iii)

Expanding equation (4.5.6-i) as a Taylor series about \( y_n \) we obtain

\[
k_i = f + (h a_{11} k_1 + h a_{12} k_2) f_r + \frac{1}{6} (h a_{11} k_1 + h a_{12} k_2)^2 f_r^2
+ \frac{1}{6} (h a_{11} k_1 + h a_{12} k_2)^3 f_r^3 + \ldots
\]
\[ f + h(a_1 k_1 + a_2 k_2) f_x + \frac{1}{2} h^2 (a_1 k_1 + a_2 k_2)^2 f_{xx} + \frac{1}{6} h^3 (a_1 k_1 + a_2 k_2)^3 f_{xxx} + \ldots \] (4.5.7)

and similarly from equation (4.5.6-ii) we have
\[ k = f + h(a_1 k_1 + a_2 k_2) f_x + \frac{1}{2} h^2 (a_1 k_1 + a_2 k_2)^2 f_{xx} + \frac{1}{6} h^3 (a_1 k_1 + a_2 k_2)^3 f_{xxx} + \ldots \] (4.5.8)

Since these two equations are implicit, we can no longer proceed by successive substitution as in the explicit case. Let us assume, instead that the solutions for \( k_1 \) and \( k_2 \) may be expressed in the form
\[ k_i = A_i + hB_i + h^2 C_i + h^3 D_i + O(h^4), \quad i = 1, 2 \] (4.5.9)

Substituting for \( k_1 \) and \( k_2 \) by equation (4.5.9) in (4.5.7) we obtain
\[ A_i + hB_i + h^2 C_i + h^3 D_i + \ldots \]
\[ f + h \left[ a_1 (A_i + hB_i + h^2 C_i + h^3 D_i + \ldots) + a_2 (A_i + hB_i + h^2 C_i + h^3 D_i + \ldots) \right] f_x + \frac{1}{2} h^2 \left[ a_1 (A_i + hB_i + h^2 C_i + h^3 D_i + \ldots) \right]^2 f_{xx} + \frac{1}{6} h^3 \left[ a_1 (A_i + hB_i + h^2 C_i + h^3 D_i + \ldots) \right]^3 f_{xxx} + \ldots \]
\[ = f + h \left[ (a_1 A_i + a_2 A_i) + h(a_1 B_i + a_2 B_i) + h^2 (a_1 C_i + a_2 C_i) \right. \]
\[ + h^3 \left[ (a_1 A_i + a_2 A_i) \right] f_x + \frac{1}{2} h^2 \left[ (a_1 A_i + a_2 A_i) \right]^2 \]
\[ + 2h(a_1 A_i + a_2 A_i)(a_1 B_i + a_2 B_i) + h^2 \left[ (a_1 B_i + a_2 B_i) \right]^2 \]
\[ + 2(a_1 A_i + a_2 A_i)(a_1 C_i + a_2 C_i) \right] f_{xx} + \frac{1}{6} h^3 \left[ (a_1 A_i + a_2 A_i) \right]^3 \]
\[ + h \left[ (a_1 B_i + a_2 B_i)(a_1 A_i + a_2 A_i)^2 + 2(a_1 A_i + a_2 A_i)(a_1 B_i + a_2 B_i) \right] f_{xxx} \] (4.5.10)

On equating powers of \( h \) we obtain,
Constant: \( A_i = f \)
\[ h : \quad B_i = (a_1 A_i + a_2 A_i) f_x \]
\[ h^2 : \quad C_i = (a_1 B_i + a_2 B_i) f_x + \frac{1}{2} (a_1 A_i + a_2 A_i)^2 f_{xx} \]
Similarly, on substituting $k_1$ and $k_2$ by (4.5.9) in (4.5.8) we have

**Constant**: $A_2 = f$

**$h^2$**: $B_2 = (a_1 a_4 + a_2 a_3)f_y$

**$h^3$**: $C_2 = (a_1 B_1 + a_2 B_2)f_y + \frac{1}{2} (a_1 A_1 + a_2 A_2)^2 f_{yy}$

The set of equations (4.5.11) and (4.5.12) is seen to be explicit and can be solved by successive substitution.

Hence, after simplifying, we have

**Constant**: $A_4 = f$

**$h^2$**: $B_1 = s_1 f_y$, where $s_1 = a_1 + a_2$

**$h^3$**: $C_1 = (a_1 s_1 + a_2 s_2) f_y^2 + \frac{1}{2} s_1^2 f^3 f_{yy}$

Therefore, equation (4.5.7) gives

$$k_i = f + h[s_1 f_y] + h^2[(a_1 s_1 + a_2 s_2) f_y^2 + \frac{1}{2} s_1^2 f^3 f_{yy}]$$

$$+ h^3[(a_1 s_1 + a_2 s_2) + a_2 (a_2 s_1 + a_2 s_2)] f_y^3$$

$$+ \left[\frac{1}{2} a_1 s_1^2 + \frac{1}{2} a_2 s_2^2 + s_1 (a_1 s_1 + a_2 s_2)\right] f^2 f_y f_{yy}$$

$$+ \frac{1}{2} s_1^2 f^3 f_{yy}$$

(4.5.13)
Similarly, we also have

\[ k_2 = f + h[s_1 f_{s_1}] + h^2[(a_{11}s_1 + a_{22}s_2) f_{s_1}^2 + \frac{1}{2}s_1^2 f^2 f_{s_1}] \]
\[ + h^3\left[\left\{a_{11}(a_{11}s_1 + a_{12}s_2) + a_{22}(a_{11}s_1 + a_{22}s_2)\right\} f_{s_1}^3 \right. \]
\[ + \left. \left\{\frac{1}{2}a_{11}s_1^2 + \frac{1}{2}a_{22}s_2^2 + s_2(a_{11}s_1 + a_{22}s_2)\right\} f^2 f_{s_1} \right. \]
\[ + \left. \frac{1}{2}s_1^2 f^3 f_{s_1} \right] \]. \tag{4.5.14} \]

Thus,

\[ h^2 f_{s_1} f_{s_1} = h^2 f + \frac{h^2}{2} (s_1 + s_2) f_{s_1} + \frac{h^3}{4} \left\{ 2(a_{11}s_1 + a_{12}s_2 + a_{21}s_1 + a_{22}s_2) f_{s_1}^2 \right. \]
\[ + \left. (s_1^2 + s_2^2) f^2 f_{s_1} \right\} + \frac{h^4}{12} \left\{ [6(a_{11}s_1 + a_{12}s_2 + a_{21}s_1 + a_{22}s_2) f_{s_1}^3 \right. \]
\[ + a_{11}a_{21}s_1 + a_{11}a_{22}s_1 + a_{12}a_{21}s_2 + a_{12}a_{22}s_2 \right\} f_{s_1} \]
\[ + \left. [9a_{11}s_1^2 + 3a_{12}s_2^2 \right. \]
\[ + 6a_{11}s_1 s_2 + 6a_{12}s_2 s_2 + 3a_{12}s_1^2 + 9a_{22}s_2^2] f^2 f_{s_1} + \left. (s_1^2 + s_2^2) f^3 f_{s_1} \right\} \right] + 0(h^5). \tag{4.5.15} \]

By comparing the expansion in (4.5.15) with the Taylor series expansion \( y(x,\dot{y},\ddot{y},\dddot{y}) \), term by term we obtain,

\[ h^2 f_{s_1} f_{s_1} : 1 - s_1 - s_2 = 0 \tag{4.5.16-i} \]
\[ h^3 f_{s_1}^2 : 1 - 3a_{11}s_1 - 3a_{21}s_1 - 3a_{12}s_2 - 3a_{22}s_2 = 0 \tag{4.5.16-ii} \]
\[ h^3 f^2 f_{s_1} : 2 - 3s_1^2 - 3s_2^2 = 0 \tag{4.5.16-iii} \]
\[ h^4 f_{s_1}^3 : 1 - 12a_{11}^2 s_1 - 12a_{11}a_{21}s_1 - 12a_{12}a_{21}s_1 - 12a_{12}a_{22}s_2 - 12a_{22}a_{22}s_2 = 0 \tag{4.5.16-iv} \]
\[ h^4 f^2 f_{s_1} f_{s_1} : 2 - 9a_{11}s_1^2 - 3a_{21}s_1^2 - 6a_{12}s_2 s_2 - 6a_{22}s_2 s_2 = 0 \tag{4.5.16-v} \]
\[ h^4 f^3 f_{s_1} : 1 - 2s_1^3 - 2s_2^3 = 0 \tag{4.5.16-vi} \]

When equations (4.5.16-i)-(4.5.16-vi) are solved by Mathematica, we immediately obtain the values

\[ s_1 = \frac{3 + \sqrt{3}}{6}, \quad s_2 = \frac{3 - \sqrt{3}}{6}, \quad a_{11} = \frac{1}{4}, \quad a_{12} = \frac{3 + 2\sqrt{3}}{12} \]
\[ a_{21} = \frac{3 - 2\sqrt{3}}{12}, \quad a_{22} = \frac{1}{4} \tag{4.5.17} \]

112
Therefore the implicit arithmetic mean Runge-Kutta (IAM) formula can now be written as

\[
y_{n+1} = y_n + h \left[ \frac{k_1 + k_2}{2} \right]
\]  
(4.5.18)

where

\[
k_1 = f\left(x_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h, y_n + \frac{1}{4}hk_1 + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)hk_2\right)
\]

\[
k_2 = f\left(x_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h, y_n + \frac{1}{4} - \frac{\sqrt{3}}{6}\right)hk_1 + \frac{1}{4}hk_2
\]

to achieve fourth order accuracy.

Numerical Example

We consider the initial value problem

\[
y' = -y, \quad y(0) = 1 \quad 0 \leq x \leq 1
\]  
(4.5.19)

with the exact solution is \( y(x) = \exp(-x) \).

The numerical solution using formulae (4.5.18) with the exact solution is shown in Table 4.11.

Table 4.11: Error in the implicit arithmetic mean formulae (4.5.18) for solving (4.5.19)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>.90483743E+00</td>
<td>.90483742E+00</td>
<td>.12700828E-07</td>
<td>5</td>
</tr>
<tr>
<td>.2000</td>
<td>.81873078E+00</td>
<td>.81873075E+00</td>
<td>.22992994E-07</td>
<td>5</td>
</tr>
<tr>
<td>.3000</td>
<td>.74081825E+00</td>
<td>.74081822E+00</td>
<td>.31208545E-07</td>
<td>5</td>
</tr>
<tr>
<td>.4000</td>
<td>.67032008E+00</td>
<td>.67032005E+00</td>
<td>.37649560E-07</td>
<td>5</td>
</tr>
<tr>
<td>.5000</td>
<td>.60653070E+00</td>
<td>.60653066E+00</td>
<td>.42579110E-07</td>
<td>5</td>
</tr>
<tr>
<td>.6000</td>
<td>.54881168E+00</td>
<td>.54881164E+00</td>
<td>.46226273E-07</td>
<td>5</td>
</tr>
<tr>
<td>.7000</td>
<td>.49658535E+00</td>
<td>.49658530E+00</td>
<td>.48790219E-07</td>
<td>5</td>
</tr>
<tr>
<td>.8000</td>
<td>.44932901E+00</td>
<td>.44932896E+00</td>
<td>.50443821E-07</td>
<td>5</td>
</tr>
<tr>
<td>.9000</td>
<td>.40656971E+00</td>
<td>.40656966E+00</td>
<td>.51330861E-07</td>
<td>5</td>
</tr>
<tr>
<td>1.000</td>
<td>.36787949E+00</td>
<td>.36787944E+00</td>
<td>.51598575E-07</td>
<td>5</td>
</tr>
</tbody>
</table>
4.5.2 2-Stage Implicit Contraharmonic Mean Runge-Kutta Method

In Evans and Yaakub [1995], the fourth order Contraharmonic mean \((C_0M)\) explicit Runge-Kutta method was presented for solving IVPs in the form \(y' = f(x,y)\). Now we consider the two stage implicit Runge-Kutta using Contraharmonic mean \((C_0M)\) given by

\[
y_{n+1} = y_n + h \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) (4.5.20-i)
\]

where

\[
k_1 = f(x_n + (a_1 + a_2)h, y_n + ha_1 k_1 + ha_2 k_2) (4.5.20-ii)
\]

\[
k_2 = f(x_n + (a_1 + a_2)h, y_n + ha_1 k_1 + ha_2 k_2) (4.5.20-iii)
\]

We proceed as before by substituting equations (4.5.13)-(4.5.14) into (4.5.20-i) to obtain an expression of \(y_{n+1}\) in terms of the function and its derivatives and the parameter \(a_{ij}\), so that it can be matched with the Taylor series expansion through terms of order \((h^4)\). The fraction in (4.5.20-i) however is not amenable to direct substitution of the series, i.e.,

\[
\frac{k_1^2 + k_2^2}{k_1 + k_2} (4.5.21)
\]

Following Evans and Yaakub [1993], the formula from (4.5.20-i) may be written as,

\[
y_{n+1} = y_n + \frac{UPPER}{LOWER} (4.5.22)
\]

where

\[
UPPER = h(k_1^2 + k_2^2) (4.5.23)
\]

and

\[
LOWER = k_1 + k_2 (4.5.24)
\]

while the Taylor series expansion of \(y(x_{n+1})\) may be given as in equation (3.3.8). From equation (3.3.8), (4.5.22)-(4.5.24), we can now write

\[
(TAYLOR \times LOWER) - UPPER = LOWER \times ERROR (4.5.25)
\]
By comparing coefficients of similar terms in (4.5.25) up to terms in \((h^4)\) yields the following equations of the conditions:

- **\(h^2 f_{yy} \):** \(1 - s_1 - s_2 = 0 \) \hspace{1cm} (4.5.26-i)

- **\(h^3 f_{yy}^2 \):** \(2 + 3s_i - 6a_{iu}s_i - 6a_{ii}s_i - 6s_i^2 + 3s_2 + 6a_{iu}s_i - 6a_{ii}s_i - 6s_i^2 = 0 \) \hspace{1cm} (4.5.26-ii)

- **\(h^3 f^2_{f_{yy}} \):** \(2 - 3s_i^2 - 3s_2^2 = 0 \) \hspace{1cm} (4.5.26-iii)

- **\(h^4 f_{yy}^3 \):** \(1 + 2s_1 + 6a_{ii}s_1 - 12a_{ii}^2 s_1 + 6a_{ii}s_i - 12a_{ii}a_{ii} s_1 - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_2 - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_2 - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_2 - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_i - 12a_{ii}a_{ii} s_2 = 0 \) \(\ldots\) \hspace{1cm} (4.5.26-iv)

- **\(h^4 f^2_{f_{yy}} \):** \(4 + 2s_1 + 3s_i^2 - 18a_{ii} s_i^2 - 6a_{ii} s_i^2 - 12s_i^3 + 2s_2 - 12a_{ii} s_2 s_2 - 12a_{ii} s_2 s_2 - 18a_{ii} s_2 s_2 - 12s_2^3 = 0 \) \hspace{1cm} (4.5.26-v)

- **\(h^4 f^3_{f_{yy}} \):** \(1 - 2s_1^3 - 2s_2^3 = 0 \) \hspace{1cm} (4.5.26-vi)

There are in effect only four coefficients, namely \(a_{ii}, a_{12}, a_{21}\) and \(a_{22}\) to be determined. In solving this nonlinear system of equations we will certainly use the first four equations to obtain an order of at least 3. With these values of \(a_{ii}, 1 \leq i \leq 2, 1 \leq j \leq 2\) satisfied we now substitute them into the last two equations.

If the solution obtained is satisfied by the last two equations then we have a method of order 4. Otherwise, the unchosen equations will provide an error term for the method, thus producing two different methods with two different error terms. The following are two sets of parameters which give rise to third order formulae.

By solving equations (4.5.26-i)-(4.5.26-iv) by Mathematica, we can see that the solution is not generic and is rejected by Mathematica. But, by solving equation (4.5.26-i)-(4.5.26-iii) and (4.5.26-v) we obtain the values...
\[ a_i = \frac{3 + \sqrt{3}}{12}, \quad a_n = \frac{1 + \sqrt{3}}{4}, \quad a_n = \frac{1 + \sqrt{3}}{4}, \quad a_n = \frac{3 + \sqrt{3}}{12} \]

Therefore the first implicit Contraharmonic mean Runge Kutta (IC\textsubscript{0}\textsubscript{M}) formula can now be written as

\[ y_{n+1} = y_n + h \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) \]  
(4.5.27)

where

\[ k_1 = f(x_n + \frac{3+\sqrt{3}}{6}h, y_n + \left( \frac{3+\sqrt{3}}{12} \right)hk_1 + \left( \frac{1+\sqrt{3}}{4} \right)hk_2) \]
\[ k_2 = f(x_n + \frac{3-\sqrt{3}}{6}h, y_n + \left( \frac{3-\sqrt{3}}{12} \right)hk_1 + \left( \frac{1-\sqrt{3}}{4} \right)hk_2) \]  
(4.5.28)

to achieve third order accuracy.

Equations (4.5.27) and (4.5.28) can be written in the Butcher array form as

\[
\begin{array}{ccc|ccc}
3 + \sqrt{3} & 3 - \sqrt{3} & 1 + \sqrt{3} \\
6 & 12 & 4 \\
3 - \sqrt{3} & 1 - \sqrt{3} & 3 + \sqrt{3} \\
6 & 4 & 12 \\
\hline
1 & 1 &
\end{array}
\]

with error term = \( h^4 f f_y^3 \).

By using the second values as

\[ a_i = \frac{3 + \sqrt{3}}{12}, \quad a_n = \frac{1 - \sqrt{3}}{4}, \quad a_n = \frac{1 + \sqrt{3}}{4}, \quad a_n = \frac{3 - \sqrt{3}}{12} \]

the second implicit Contraharmonic mean Runge Kutta (IC\textsubscript{0}\textsubscript{M}) formula can now be written as,

\[ y_{n+1} = y_n + h \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) \]  
(4.5.29)

where

\[ k_1 = f(x_n + \frac{3-\sqrt{3}}{6}h, y_n + \left( \frac{3-\sqrt{3}}{12} \right)hk_1 + \left( \frac{1-\sqrt{3}}{4} \right)hk_2) \]
\[ k_2 = f(x_n + \frac{3+\sqrt{3}}{6}h, y_n + \left( \frac{3+\sqrt{3}}{12} \right)hk_1 + \left( \frac{1+\sqrt{3}}{4} \right)hk_2) \]  
(4.5.30)

to achieve third order accuracy.

In Butcher array form equations (4.5.29) and (4.5.30) can be written as

116
The numerical solution using formulae (4.5.27)-(4.5.28) and (4.5.29)-(4.5.30) for solving problem (4.5.19) compared with the exact solution are shown in Table 4.12 and Table 4.13.

Table 4.12: Error in the implicit Contraharmonic mean formulae (4.5.27)-(4.5.28) for solving (4.5.19)

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100</td>
<td>.90483360E+00</td>
<td>.90483742E+00</td>
<td>.38208776E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.200</td>
<td>.81872384E+00</td>
<td>.81873075E+00</td>
<td>.69146465E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.300</td>
<td>.74080844E+00</td>
<td>.74081822E+00</td>
<td>.93850105E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.400</td>
<td>.67030872E+00</td>
<td>.67032005E+00</td>
<td>.11322610E-04</td>
<td>5</td>
</tr>
<tr>
<td>0.500</td>
<td>.60517858E+00</td>
<td>.60653066E+00</td>
<td>.12806476E-04</td>
<td>5</td>
</tr>
<tr>
<td>0.600</td>
<td>.54879773E+00</td>
<td>.54881164E+00</td>
<td>.13905422E-04</td>
<td>5</td>
</tr>
<tr>
<td>0.700</td>
<td>.49657062E+00</td>
<td>.49658530E+00</td>
<td>.14679274E-04</td>
<td>5</td>
</tr>
<tr>
<td>0.800</td>
<td>.44931378E+00</td>
<td>.44932896E+00</td>
<td>.15179956E-04</td>
<td>5</td>
</tr>
<tr>
<td>0.900</td>
<td>.40655421E+00</td>
<td>.40656966E+00</td>
<td>.15452456E-04</td>
<td>5</td>
</tr>
<tr>
<td>1.000</td>
<td>.36786391E+00</td>
<td>.36787944E+00</td>
<td>.15535671E-04</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 4.13: Error in the implicit contraharmonic mean formulae (4.5.29)-(4.5.30) for solving (4.5.19)

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100</td>
<td>.90483359E+00</td>
<td>.90483742E+00</td>
<td>.38259721E-05</td>
<td>4</td>
</tr>
<tr>
<td>0.200</td>
<td>.81872384E+00</td>
<td>.81873075E+00</td>
<td>.69146465E-05</td>
<td>4</td>
</tr>
<tr>
<td>0.300</td>
<td>.74080844E+00</td>
<td>.74081822E+00</td>
<td>.93850105E-05</td>
<td>4</td>
</tr>
<tr>
<td>0.400</td>
<td>.67030872E+00</td>
<td>.67032005E+00</td>
<td>.11340445E-04</td>
<td>4</td>
</tr>
<tr>
<td>0.500</td>
<td>.60517858E+00</td>
<td>.60653066E+00</td>
<td>.12827833E-04</td>
<td>4</td>
</tr>
<tr>
<td>0.600</td>
<td>.54879773E+00</td>
<td>.54881164E+00</td>
<td>.13930057E-04</td>
<td>4</td>
</tr>
<tr>
<td>0.700</td>
<td>.49657062E+00</td>
<td>.49658530E+00</td>
<td>.14700813E-04</td>
<td>4</td>
</tr>
<tr>
<td>0.800</td>
<td>.44931378E+00</td>
<td>.44932896E+00</td>
<td>.15198669E-04</td>
<td>5</td>
</tr>
<tr>
<td>0.900</td>
<td>.40655419E+00</td>
<td>.40656966E+00</td>
<td>.15468580E-04</td>
<td>5</td>
</tr>
<tr>
<td>1.000</td>
<td>.36786389E+00</td>
<td>.36787944E+00</td>
<td>.15549421E-04</td>
<td>5</td>
</tr>
</tbody>
</table>
4.5.3 2-Stage Implicit Centroidal Mean Method

By carrying out the procedure outlined in the implicit contraharmonic mean \((IC_oM)\) method, a new implicit centroidal mean \((IC_eM)\) method can be established in the form

\[ y_{on} = y_o + h \left[ \frac{2(k_1^2 + k_2^2 + k_3^2)}{3(k_1 + k_2)} \right] \quad (4.5.31-i) \]

where

\[ k_1 = f(x_o + (a_{11} + a_{12})h, y_o + ha_{21}k_1 + ha_{22}k_2) \quad (4.5.31-ii) \]

\[ k_2 = f(x_o + (a_{21} + a_{22})h, y_o + ha_{31}k_1 + ha_{32}k_2) \quad (4.5.31-iii) \]

Normally, we would substitute equations (4.5.31-ii)-(4.5.31-iii) into (4.5.31-i) to obtain an expression for \(y_{on}\). However, the fraction involved in (4.5.31-i) is not amenable to direct substitution, i.e.,

\[ \frac{k_1^2 + k_2^2 + k_3^2}{k_1 + k_2} \quad (4.5.32) \]

Following Evans and Yaakub [1995], the formula from (4.5.31-i) may be written as in equation (4.5.22) where,

\[ UPPER = \frac{2h}{3} (k_1^2 + k_1k_2 + k_2^2) \quad (4.5.33) \]

and \( LOWER = k_1 + k_2 \quad (4.5.34) \)

By substituting equations (4.5.33)-(4.5.34) into equation (4.5.25) and comparing the coefficients of similar terms up to terms in \((h^4)\) yields the following equations of conditions

\[ h^2 f_{yy} : \quad 1 - s_1 - s_2 = 0 \quad (4.5.35-i) \]

\[ h^3 f_{yy}^2 : \quad 2 + 3s_1 - 6a_{11}s_1 - 6a_{12}s_1 - 4s_1^2 + 3s_2 \\
-6a_{22}s_2 - 6a_{22}s_2 - 4s_1s_2 - 4s_2^2 = 0 \quad (4.5.35-ii) \]

\[ h^3 f_{yy}^2 : \quad 2 - 3s_1^2 - 3s_2^2 = 0 \quad (4.5.35-iii) \]
Equations (4.5.35-i)-(4.5.35-iv) are then solved simultaneously by Mathematica to give the required parameters, i.e.,

\[
an_n = \frac{9 + 2\sqrt{3}}{72}, \quad a_n = \frac{27 - 14\sqrt{3}}{72}, \quad a_n = \frac{27 + 14\sqrt{3}}{72}, \quad a_2 = \frac{9 - 2\sqrt{3}}{72}
\]

Thus, this new implicit method can be written as follows

\[
y_{n+1} = y_n + h \left[ 2 \left( k_1^2 + k_1 k_2 + k_2^2 \right) \right] / 3 \left( k_1 + k_2 \right).
\]

(4.5.36)

where

\[
k_1 = f \left( x_n + \left( \frac{3\sqrt{3}}{6} \right) h, y_n + \left( \frac{3\sqrt{3}}{12} \right) h k_1 + \left( \frac{3\sqrt{3}}{18} \right) h k_2 \right)
\]

and

\[
k_2 = f \left( x_n + \left( \frac{3\sqrt{3}}{6} \right) h, y_n + \left( \frac{3\sqrt{3}}{12} \right) h k_1 + \left( \frac{3\sqrt{3}}{18} \right) h k_2 \right)
\]

(4.5.37)

to achieve third order accuracy.

The formulae in equations (4.5.36)-(4.5.37) can also be written in Butcher array form as

\[
\begin{array}{c|ccc}
3 - \sqrt{3} & 9 + 2\sqrt{3} & 27 - 14\sqrt{3} \\
\hline
6 & 72 & 72 \\
3 + \sqrt{3} & 27 + 14\sqrt{3} & 9 - 2\sqrt{3} \\
6 & 72 & 72
\end{array}
\]

with error terms $= 0.5 \ h^4 f^2 f_y f_{yy}$. 

119
Solving equations (4.5.35-i)-(4.5.35-iii) and (4.5.35-v) by Mathematica, we immediately obtain the values

\[ a_n = \frac{9 - \sqrt{3}}{36}, \quad a_{n+1} = \frac{9 + 7\sqrt{3}}{36}, \quad a_{n+2} = \frac{9 - 7\sqrt{3}}{36}, \quad a_{n+3} = \frac{9 + \sqrt{3}}{36}. \]

Therefore, the second implicit third order centroidal mean Runge Kutta (ICeM) formulae can be written as

\[ y_{n+1} = y_n + h \left[ \frac{2(k_1^2 + k_1k_2 + k_2^2)}{3(k_1 + k_2)} \right] \quad (4.5.38) \]

where

\[ k_1 = f(x_n + \frac{3\sqrt{3}h}{6}y_n + \frac{2+\sqrt{3}}{36}hk_1 + \frac{2+\sqrt{3}}{36}hk_2) \]

\[ k_i = f(x_n + \frac{3\sqrt{3}h}{6}y_n + \frac{2-\sqrt{3}}{36}hk_1 + \frac{2-\sqrt{3}}{36}hk_2) \quad (4.5.39) \]

or in Butcher array form as

\[
\begin{array}{ccc}
3 + \sqrt{3} & 9 - \sqrt{3} & 9 + 7\sqrt{3} \\
6 & 36 & 36 \\
3 - \sqrt{3} & 9 - 7\sqrt{3} & 9 + \sqrt{3} \\
6 & 36 & 36 \\
\end{array}
\]

with error terms = 0.3333 \( h^4 f^{\frac{3}{2}} \).

The numerical solution using formulae (4.5.36)-(4.5.37) and (4.5.38)-(4.5.39) compared with the exact solution are shown in Table 4.14 and Table 4.15.

Table 4.14: Error in the implicit centroidal mean formulae (4.5.36)-(4.5.37) for solving (4.5.19)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>.90483736E+00</td>
<td>.90483742E+00</td>
<td>.59515458E-07</td>
<td>4</td>
</tr>
<tr>
<td>.2000</td>
<td>.81873065E+00</td>
<td>.81873075E+00</td>
<td>.10741252E-06</td>
<td>4</td>
</tr>
<tr>
<td>.3000</td>
<td>.74081807E+00</td>
<td>.74081822E+00</td>
<td>.14582395E-06</td>
<td>4</td>
</tr>
<tr>
<td>.4000</td>
<td>.67031987E+00</td>
<td>.67032005E+00</td>
<td>.17596571E-06</td>
<td>4</td>
</tr>
<tr>
<td>.5000</td>
<td>.60653046E+00</td>
<td>.60653066E+00</td>
<td>.19911728E-06</td>
<td>4</td>
</tr>
<tr>
<td>.6000</td>
<td>.54881142E+00</td>
<td>.54881164E+00</td>
<td>.21634109E-06</td>
<td>4</td>
</tr>
<tr>
<td>.7000</td>
<td>.49658508E+00</td>
<td>.49658530E+00</td>
<td>.22856076E-06</td>
<td>4</td>
</tr>
<tr>
<td>.8000</td>
<td>.44932873E+00</td>
<td>.44932896E+00</td>
<td>.23657754E-06</td>
<td>4</td>
</tr>
<tr>
<td>.9000</td>
<td>.40656942E+00</td>
<td>.40656966E+00</td>
<td>.24108585E-06</td>
<td>4</td>
</tr>
<tr>
<td>1.000</td>
<td>.36787920E+00</td>
<td>.36787944E+00</td>
<td>.24268578E-06</td>
<td>4</td>
</tr>
</tbody>
</table>

120
Table 4.15: Error in the implicit centroidal mean formulae (4.5.38)-(4.5.39) for solving (4.5.19)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000</td>
<td>0.90483616E+00</td>
<td>0.90483742E+00</td>
<td>0.12609651E-05</td>
<td>4</td>
</tr>
<tr>
<td>0.2000</td>
<td>0.81872847E+00</td>
<td>0.81873075E+00</td>
<td>0.22864610E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.3000</td>
<td>0.74081512E+00</td>
<td>0.74081822E+00</td>
<td>0.31053815E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.4000</td>
<td>0.67031630E+00</td>
<td>0.67032005E+00</td>
<td>0.37477587E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.60652642E+00</td>
<td>0.60653066E+00</td>
<td>0.42397792E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.6000</td>
<td>0.54880703E+00</td>
<td>0.54881164E+00</td>
<td>0.46042443E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.7000</td>
<td>0.49658444E+00</td>
<td>0.49658530E+00</td>
<td>0.48609776E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.8000</td>
<td>0.44932394E+00</td>
<td>0.44932896E+00</td>
<td>0.50271845E-05</td>
<td>5</td>
</tr>
<tr>
<td>0.9000</td>
<td>0.40656454E+00</td>
<td>0.40656966E+00</td>
<td>0.51177716E-05</td>
<td>5</td>
</tr>
<tr>
<td>1.0000</td>
<td>0.36787430E+00</td>
<td>0.36787944E+00</td>
<td>0.51456282E-05</td>
<td>5</td>
</tr>
</tbody>
</table>

4.5.4.2 2-Stage Implicit Harmonic Mean Method

In Sanugi and Evans [1993], the fourth order explicit Harmonic mean \((H_aM)\) method was established in the form

\[
y_{n+1} = y_n + \frac{h}{3} \left[ \frac{2k_1k_{11}}{k_1 + k_{11}} \right]
\]

(4.5.40)

where

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + \frac{h}{3}, y_n + \frac{h}{3} k_1)
\]

\[
k_3 = f(x_n + \frac{h}{3}, y_n + \frac{h}{3} k_1 + \frac{h}{3} k_2)
\]

\[
k_4 = f(x_n + h, y_n - \frac{h}{6} k_1 + \frac{h}{6} k_2 + \frac{h}{6} k_3)
\]

(4.5.41)

Now, we attempt to establish the implicit Harmonic mean \((IH_aM)\) method in the form

\[
y_{n+1} = y_n + h \left[ \frac{2k_1k_2}{k_1 + k_2} \right]
\]

(4.5.42-i)

where

\[
k_1 = f(x_n + (a_{11} + a_{12})h, y_n + ha_{11}k_1 + ha_{12}k_2)
\]

(4.5.42-ii)

\[
k_2 = f(x_n + (a_{11} + a_{12})h, y_n + ha_{11}k_1 + ha_{12}k_2)
\]

(4.5.42-iii)
By using the same procedure as the implicit Centroidal mean \( (I_{c,m}) \) method we substitute equations (4.5.42-ii)-(4.5.42-iii) into equation (4.5.42-i) to obtain an expression of \( y_{*i} \). However, the fraction involved in (4.5.42-i) is not amenable to direct substitution, i.e.,

\[
\frac{k_i k_j}{k_i + k_j}
\]

(4.5.43)

The formula from (4.5.42-i) can also written as in equation (4.5.22) where

\[
UPPER = 2h(k_i k_j)
\]

and

\[
LOWER = k_i + k_j
\]

(4.5.44)

(4.5.45)

By substituting equations (3.3.8) and (4.5.44)-(4.5.45) into equation (4.5.25) and comparing coefficient of similar terms up to terms in \( (h^4) \) yields the following equations of conditions

\[
h^2 f_{yy} : 1 - s_i - s_2 = 0
\]

(4.5.46-i)

\[
h^3 f_{yy}^2 : 2 + 3s_i - 6a_{ii}s_i - 6a_{i2}s_i + 3s_2 - 6a_{i2}s_2 - 6a_{22}s_2 - 12s_i s_2 = 0
\]

(4.5.46-ii)

\[
h^3 f_{yy}^3 : 2 - 3s_i^2 - 3s_2^2 = 0
\]

(4.5.46-iii)

\[
h^4 f_{yy}^4 : 1 + 2s_i + 6a_{ii}s_i - 12a_{ii}^2s_i + 6a_{ii}s_i - 12a_{i2}a_{ii}s_i - 12a_{ii}a_{ii}s_i - 12a_{i2}a_{ii}s_i
-12a_{i2}a_{22}s_i - 24a_{i2}a_{22}s_i + 2s_i + 6a_{ii}s_i - 12a_{ii}a_{ii}s_i - 12a_{ii}a_{ii}s_i
+6a_{i2}s_2 - 12a_{i2}a_{22}s_i - 12a_{22}s_2 - 24a_{i2}s_i s_2 - 24a_{i2}s_i s_2 - 24a_{i2}a_{i2}s_i^2 = 0
\]

(4.5.46-iv)

\[
h^4 f_{yy}^4 : 4 + 2s_i + 3s_i^2 - 18a_{ii}s_i^2 - 6a_{ii}s_i^2 + 2s_2 - 12a_{i2}s_2 - 12a_{i2}s_2 - 18a_{22}s_i^2 - 12s_i s_2 - 8s_2^2 = 0
\]

(4.5.46-v)

\[
h^4 f_{yy}^2 : 1 - 2s_i^3 - 2s_2^3 = 0
\]

(4.5.46-vi)
Consequently equations (4.5.46-i)-(4.5.46-iv) are then solved simultaneously by Mathematica to immediately obtain the parameters, i.e.,

\[ a_u = \frac{9 - 2\sqrt{3}}{24}, a_v = \frac{3 - 2\sqrt{3}}{24}, a_u = \frac{3 + 2\sqrt{3}}{24}, a_v = \frac{9 + 2\sqrt{3}}{24} \]

Therefore, the implicit Harmonic mean \((IH_\alpha M)\) method can be written as

\[ y_{u+1} = y_n + h \left( \frac{2k_1k_2}{k_1 + k_2} \right) \]  

(4.5.47)

\[ k_1 = f \left( x_n + \left( \frac{3 - \sqrt{3}}{6} \right) h, y_n + \left( \frac{9 - 2\sqrt{3}}{24} \right) h_k + \left( \frac{3 - 2\sqrt{3}}{24} \right) h_k \right) \]

where

\[ k_2 = f \left( x_n + \left( \frac{3 + \sqrt{3}}{6} \right) h, y_n + \left( \frac{9 + 2\sqrt{3}}{24} \right) h_k + \left( \frac{3 + 2\sqrt{3}}{24} \right) h_k \right) \]

(4.5.48)

... to achieve third order accuracy. The formulae in equations (4.5.47)-(4.5.48) can also be written in Butcher array form as

\[
\begin{array}{ccc}
\frac{3 - \sqrt{3}}{6} & 9 - 2\sqrt{3} & 3 - 2\sqrt{3} \\
\frac{3 + \sqrt{3}}{6} & 3 + 2\sqrt{3} & 9 + 2\sqrt{3} \\
\frac{3 - \sqrt{3}}{6} & 9 - 2\sqrt{3} & 3 - 2\sqrt{3} \\
\frac{3 + \sqrt{3}}{6} & 3 + 2\sqrt{3} & 9 + 2\sqrt{3} \\
\end{array}
\]

with error terms \(= -3.88583 h^4 f^2 f_y f_{yy} \).

Solving equations (4.5.46-i)-(4.5.46-iii) and (4.5.46-v) by Mathematica, we immediately obtain the value,

\[ a_u = \frac{13\sqrt{3}}{54}, a_v = \frac{1 - \frac{4\sqrt{3}}{3}}{2}, a_u = \frac{2 - \frac{2\sqrt{3}}{3}}{2}, a_v = \frac{2\sqrt{3}}{27} \]
Therefore, the second implicit third order Harmonic mean Runge-Kutta ($IH_aM$) formula can be written as

\[ y_{n+1} = y_n + h \left( \frac{2k_1k_2}{k_1 + k_2} \right) \]  

(4.5.49)

where

\[ k_1 = f \left( x_n + \left( \frac{3 + \sqrt{3}}{6} \right) h, y_n + \left( \frac{13\sqrt{3}}{54} \right) h k_1 + \left( \frac{1 - \frac{2\sqrt{3}}{9}}{2} \right) h k_2 \right) \]

\[ k_2 = f \left( x_n + \left( \frac{3 - \sqrt{3}}{6} \right) h, y_n + \left( \frac{2 - \frac{2\sqrt{3}}{9}}{4} \right) h k_1 + \left( \frac{2\sqrt{3}}{27} \right) h k_2 \right) \]

or in Butcher array form as

\[
\begin{array}{ccc}
3 + \sqrt{3} & 13 & 1 - \frac{1}{3} \\
6 & 18\sqrt{3} & 2 \\
3 - \sqrt{3} & 2 - \frac{2}{9} & 2 \\
6 & 4 & 9\sqrt{3} \\
& & 1 \\
\end{array}
\]  

with error terms = $-0.818899 \ h^4 \frac{f_y^3}{2}$

The numerical solution using formulae (4.5.47)-(4.5.48) and (4.5.49)-(4.5.50) compared with exact solution are shown in Table 4.16 and Table 4.17.

Table 4.16: Error in the implicit harmonic mean formulae (4.5.47)-(4.5.48) method for solving (4.5.19)

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>.90483743E+00</td>
<td>.90483742E+00</td>
<td>.12216797E-07</td>
<td>4</td>
</tr>
<tr>
<td>.2000</td>
<td>.81873077E+00</td>
<td>.81873075E+00</td>
<td>.21530529E-07</td>
<td>4</td>
</tr>
<tr>
<td>.3000</td>
<td>.74081825E+00</td>
<td>.74081822E+00</td>
<td>.29347467E-07</td>
<td>4</td>
</tr>
<tr>
<td>.4000</td>
<td>.67032005E+00</td>
<td>.67032005E+00</td>
<td>.32937467E-07</td>
<td>4</td>
</tr>
<tr>
<td>.5000</td>
<td>.60653069E+00</td>
<td>.60653066E+00</td>
<td>.28762721E-07</td>
<td>4</td>
</tr>
<tr>
<td>.6000</td>
<td>.54881166E+00</td>
<td>.54881164E+00</td>
<td>.25684771E-07</td>
<td>4</td>
</tr>
<tr>
<td>.7000</td>
<td>.49658532E+00</td>
<td>.49658530E+00</td>
<td>.20434691E-07</td>
<td>4</td>
</tr>
<tr>
<td>.8000</td>
<td>.44932898E+00</td>
<td>.44932896E+00</td>
<td>.13295030E-07</td>
<td>4</td>
</tr>
<tr>
<td>.9000</td>
<td>.40656966E+00</td>
<td>.40656966E+00</td>
<td>.45142284E-08</td>
<td>4</td>
</tr>
<tr>
<td>1.000</td>
<td>.36787944E+00</td>
<td>.36787944E+00</td>
<td>.56894545E-08</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 4.17: Error in the implicit harmonic mean formulae (4.5.49)-(4.5.50)

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute error</th>
<th>No. iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>9.0484056E+00</td>
<td>9.0483742E+00</td>
<td>3.1387275E-05</td>
<td>5</td>
</tr>
<tr>
<td>.2000</td>
<td>8.1873643E+00</td>
<td>8.1873075E+00</td>
<td>5.6800203E-05</td>
<td>5</td>
</tr>
<tr>
<td>.3000</td>
<td>7.4082593E+00</td>
<td>7.4081822E+00</td>
<td>7.7093157E-05</td>
<td>5</td>
</tr>
<tr>
<td>.4000</td>
<td>6.7032935E+00</td>
<td>6.7032005E+00</td>
<td>9.3010457E-05</td>
<td>5</td>
</tr>
<tr>
<td>.5000</td>
<td>6.0654118E+00</td>
<td>6.0653066E+00</td>
<td>1.0520120E-04</td>
<td>5</td>
</tr>
<tr>
<td>.6000</td>
<td>5.4882306E+00</td>
<td>5.4881164E+00</td>
<td>1.1423056E-04</td>
<td>5</td>
</tr>
<tr>
<td>.7000</td>
<td>4.9659736E+00</td>
<td>4.9658530E+00</td>
<td>1.2058990E-04</td>
<td>5</td>
</tr>
<tr>
<td>.8000</td>
<td>4.4934143E+00</td>
<td>4.4932896E+00</td>
<td>1.2470570E-04</td>
<td>5</td>
</tr>
<tr>
<td>.9000</td>
<td>4.0658235E+00</td>
<td>4.0656966E+00</td>
<td>1.2694742E-04</td>
<td>5</td>
</tr>
<tr>
<td>1.000</td>
<td>3.6789220E+00</td>
<td>3.6787944E+00</td>
<td>1.2763457E-04</td>
<td>5</td>
</tr>
</tbody>
</table>

From the above discussion, we can see that the results from the third order methods or the error term in implicit harmonic mean ($IH_M$) methods performs better than the fourth order arithmetic mean (AM) method or the 2-stage scheme developed by Hammer and Hollingsworth [1955]. With smaller error terms in the third order implicit harmonic mean ($IH_M$) method, we can conclude that the 2-stage scheme in the implicit harmonic mean method is 'nearly' fourth order accuracy.

### 4.6 L-STABLE MODIFIED TRAPEZOIDAL METHODS FOR IVPs

In Sanugi [1986], it was shown that the geometric mean (GM) trapezoidal formula is L-stable. In recent work, Chawla and Al-Zanaidi [1996] proposed that by modifying the classical arithmetic mean (AM), Geometric mean (GM) and harmonic mean ($H_M$) trapezoidal formulas a new class of modified trapezoidal formulas with L-stability is obtained.

The classical trapezoidal formula is given by

$$y_{n+1} = y_n + \frac{h}{2} [f_n + f_{n+1}] \quad (4.6.1)$$

for the numerical integration of the initial value problem.
Alternative formula to (4.6.1) have been proposed by using various other types of means for \( f_n \) and \( f_{n+1} \). A contraharmonic mean (\( C,M \)) trapezoidal formula:

\[
y_{n+1} = y_n + h \left[ \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right]
\]

was discussed by Evans and Yaakub [1995] and a Centroidal mean (\( C,M \)) was proposed by Yaakub and Evans [1995]:

\[
y_{n+1} = y_n + h \left[ \frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})} \right]
\]

and a root-mean square (RMS) was also proposed by Yaakub and Evans [1993]:

\[
y_{n+1} = y_n + h \sqrt{\left( \frac{f_n^2 + f_{n+1}^2}{2} \right)}
\]

In this section, a new class of trapezoidal formulas is obtained by modifying the contraharmonic mean (\( C,M \)), the centroidal mean (\( C,M \)) and root-mean square (RMS) methods are presented. By using a standard test equation

\[
f(x,y) = \lambda y , \quad \lambda \in \mathbb{C}, \quad \text{Re}(z) < 0
\]

we show that each of these modified trapezoidal formulas is second order and L-stable for autonomous problems.

The stability theory for ordinary differential equations include zero stability, weak stability and absolute stability or A-stability are discussed in Lambert [1973]. In our discussion, following Axelsson [1969] , Ehle [1969] and Lambert [1973] the term L-stability was defined as in Section 3.5.
4.6.1 L-Stable Modified Trapezoidal Formulas

The classical Euler method
\[ y_{n+1} = y_n + hf_n \]  
(4.6.7)

applied backwards at \( x_{n+1} \), in the negative x-direction gives
\[ y_n = y_{n+1} - hf_{n+1} \]  
(4.6.8)

Formula (4.6.8) was called the explicit backward Euler formula (see Chawla and Al-Zanaidi [1996]) and defined by
\[ \hat{y}_n = y_{n+1} - hf_{n+1} \]  
(4.6.9)

and we set
\[ \hat{f}_n = f(x_n, \hat{y}_n) \).

4.6.1.1 Modified Arithmetic Mean (MAM) Trapezoidal Formula

The classical AM trapezoidal formula (4.6.1) was modified with equation (4.6.9) and written as
\[ y_{n+1} = y_n + \frac{h}{2} [\hat{f}_n + f_{n+1}] \]  
(4.6.10)

where we can obtain the local truncation error in equation (4.6.10) since
\[ \hat{y}_n = y_n - \frac{h^2}{2} y'' + 0(h^3) \]

and
\[ \hat{f}_n = f_n - \frac{h^2}{2} y'' g_n + 0(h^3) \]  
(4.6.11)

where \( g_n = \frac{\partial f_n}{\partial y_n} \). By substituting equation (4.6.11) into (4.6.10) we obtain
\[ y_{n+1} = y_n + \left(\frac{h}{2} y'' + \frac{h}{2} y''\right) + \frac{h^2}{2} y'' + \frac{h^3}{4} (y''' - y'' g_n) + 0(h^4) \]

\[ = y_n + hy'' + \frac{h^2}{2} y'' + \frac{h^3}{12} (3y''' - 3y'' g_n) + 0(h^4) \]  
(4.6.12)
and by Taylor series as

\[ y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + O(h^4) \]  \hspace{1cm} (4.6.13)

By subtracting (4.6.12) from (4.6.13), it can be shown that the local truncation error for the MAM trapezoidal formula is

\[ LTE_{MAM} = \frac{h^3}{12} [-y''' + 3y'' y_n] + O(h^4). \]  \hspace{1cm} (4.6.14)

**Theorem 1.** The MAM trapezoidal formula (4.6.10) is L-stable.

Proof. By applying the test equation (4.6.6) in (4.6.9) we obtain

\[ \hat{y}_n = (1 - z)y_{n+1} \]  \hspace{1cm} (4.6.15)

and

\[ \hat{f}_n = \lambda(1 - z)y_{n+1} \]  \hspace{1cm} (4.6.16)

By substituting equations (4.6.15) and (4.6.16) in (4.6.10), we obtain

\[ y_{n+1} = y_n + \frac{z}{2} [(2 - z)y_n] \]  \hspace{1cm} (4.6.17)

As in equation (3.3.30) then from equation (4.6.17), we obtain

\[ Q(z) = \frac{2}{2 - 2z + z^2} \]  \hspace{1cm} (4.6.18)

Now, we set \( w = 2 - 2z + z^2 \) and \( \text{Re}(z) = -x - iy \) in equation (4.6.18) to give

\[ w = (2 + 2x + x^2 - y^2) + i2xy \]

and

\[ |w|^2 = (2 + 2x + x^2 - y^2)^2 + (2xy)^2. \]

For \( x > 0 \), since \( |z| \geq x \), it follows that

\[ |w|^2 \geq (2 + 2x + x^2)^2 \]  \hspace{1cm} (4.6.19)

and from (4.6.18) we obtain
\[ |Q(z)| < \frac{2}{2 + 2x + x^2} \quad \text{(4.6.20)} \]

From equation (4.6.20), \(|Q(z)| < 1\) and for \(\text{Re}(z) < 0\) we can see that \(|Q(z)| \to 0\) as \(x \to \infty\). This proves that the MAM trapezoidal formula (4.6.20) is L-stable.

### 4.6.1.2 Modified Contraharmonic Mean (MC,M) Trapezoidal Formula

By using (4.6.9), we extend the modification to the \(C,M\) method in (4.6.3) to obtain

\[
y_{n+1} = y_n + h \left[ \frac{\hat{f}_n^2 + \hat{f}_{n+1}^2}{\hat{f}_n + \hat{f}_{n+1}} \right]. \quad \text{(4.6.21)}
\]

The local truncation error is obtained by using (4.6.11) to give

\[
\frac{\hat{f}_n^2 + \hat{f}_{n+1}^2}{\hat{f}_n + \hat{f}_{n+1}} = f_n + \frac{h}{2} y_n'' + \frac{h^2}{4} \left( y_n''' - y_n g_n + \frac{(y_n')^2}{f_n} \right)
\]

\[+ \frac{h^3}{12} \left[ \frac{y_n'''}{f_n} + 3 \frac{y_n' y_n'''}{2 f_n} - 3 \left( \frac{y_n''}{f_n} \right)^2 + 3 \frac{(y_n''')^2}{f_n} \right] + O(h^4) \quad \text{(4.6.22)}
\]

By substituting (4.6.22) into (4.6.21) we obtain

\[
y_{n+1} = y_n + hf_n + \frac{h^2}{2} y_n'' + \frac{h^2}{4} \left[ \frac{y_n'''}{f_n} + \frac{(y_n')^2}{f_n^2} - y_n' g_n \right] + O(h^4) \quad \text{(4.6.23)}
\]

From (4.6.13) and (4.6.23) it can be shown that the local truncation error for the \(MC,M\) trapezoidal formula is given by

\[
LTE_{MC,M} = \frac{h^3}{12} \left[ 3y_n' g_n - 3 \left( \frac{y_n''}{f_n} \right)^2 \right] \quad \text{(4.6.24)}
\]

**Theorem 2.** The \(MC,M\) trapezoidal formula (4.6.21) is L-stable.

**Proof.** By applying the \(MC,M\) trapezoidal formula (4.6.21) to the test equation in (4.6.6), we obtain
\[ y_{n+1} = y_n + z[(2-z)y_{n+1}] \]  \hspace{1cm} (4.6.25)

By writing equation (4.6.25) in a similar form as in equation (3.3.30), we obtain

\[ Q(z) = \frac{1}{1-2z+z^2} \]  \hspace{1cm} (4.6.26)

Now, we set \( w = 1 - 2z + z^2 \) and \( \text{Re}(z) = -x - iy \) in equation (4.6.26) to give

\[ w = (1 + 2x + x^2 - y^2) + i(2xy + 2y) \]

and

\[ |w|^2 = (1 + 2x + x^2 - y^2)^2 + (2xy + 2y)^2 \]

For \( x > 0 \), since \(|z| \geq x\), it follows that

\[ |w|^2 \geq (1 + 2x + x^2)^2 \]  \hspace{1cm} (4.6.27)

and from (4.6.26) we obtain

\[ |Q(z)| < \frac{1}{1 + 2x + x^2} \]  \hspace{1cm} (4.6.28)

From equation (4.6.28), \(|Q(z)| < 1\) and for \( \text{Re}(z) < 0 \) we can see that \(|Q(z)| \to 0\) as \( x \to \infty \). Thus, we show that our \( MC,M \) trapezoidal formula (4.6.21) is L-stable.

### 4.6.1.3 Modified Centroidal Mean \( (MC,M) \) Trapezoidal Formula

With (4.6.9), we can modify the \( C,M \) trapezoidal formula (4.6.4) to obtain

\[ y_{n+1} = y_n + h \left[ \frac{2(f_n^2 + \hat{f}_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})} \right] \]  \hspace{1cm} (4.6.29)

By using Mathematica and with (4.6.11), we obtain
By substituting (4.6.30) into (4.6.29) we obtain

\[ y_{n+1} = y_n + hf_n + \frac{h^2}{2} y_n'' + \frac{h^3}{12} \left[ 3 y_n''' + \frac{(y_n')^2}{f_n} - 3 y_n' g_n \right] + O(h^4) \]  

(4.6.31)

From (4.6.13) and (4.6.31) it can be shown that the local truncation error for the \( MC,M \) trapezoidal formula is given by

\[ LTE_{MC,M} = \frac{h^3}{12} \left[ 3 y_n''' + \frac{(y_n')^2}{f_n} - y_n'' \right]. \]  

(4.6.32)

**Theorem 3.** The \( MC,M \) trapezoidal formula (4.6.29) is \( L \)-stable.

**Proof.** By applying the \( MC,M \) trapezoidal formula (4.6.29) to the test equation in (4.6.6), we obtain

\[ y_{n+1} = y_n + \frac{2}{3} \left[ \frac{(3 - 3z + z^2)}{(2 - z)} y_{n+1} \right] \]  

(4.6.33)

Equation (4.6.33) can also be written as in equation (3.3.30) to give

\[ Q(z) = \frac{6 - 3z}{6 - 9z + 6z^2 - 2z^3} \]  

(4.6.34)

Now, we set \( w = 6 - 9z + 6z^2 - 2z^3 \) and \( \text{Re}(z) = -x - iy \) in equation (4.6.34) to give

\[ w = (6 + 9x + 6x^2 + 2x^3 - 6y^2 - 6xy^2) + i(9y + 12xy - y^3 + 6x^2y) \]

and \( |w|^2 = (6 + 9x + 6x^2 + 2x^3)^2 + (9 + 36x + 72x^2 + 48x^3 + 12x^4)y^2 \)

\[ + (24x + 12x^3)y^4 + 4y^6 \].
For $x > 0$, since $|z| \geq x$, it follows that

$$|w|^2 \geq (6 + 9x + 6x^2 + 2x^3)^2$$

and from (4.6.34) we obtain

$$|Q(z)| < \frac{6 + 3x}{6 + 9x + 6x^2 + 2x^3}$$

From equation (4.6.35), $|Q(z)| < 1$ and for $\text{Re}(z) < 0$ we can see that $|Q(z)| \rightarrow 0$ as $x \rightarrow \infty$. Thus, the $MC,M$ trapezoidal formula (4.6.29) is L-stable.

### 4.6.1.4 Modified Root-Mean-Square (MRMS) Trapezoidal Formula

With (4.6.9), we also modify the RMS trapezoidal formula (4.6.5) to obtain

$$y_{n+1} = y_n + h \sqrt{\frac{f_n^2 + f_{n+1}^2}{2}}$$

By using Mathematica and with (4.6.11), we obtain

$$\sqrt{\frac{f_n^2 + f_{n+1}^2}{2}} = f_n + \frac{h}{2} y_n'' + \frac{h^2}{12} \left[ 3y_n''' + \frac{3(y_n'')^2}{2f_n} - 3y_n''g_n \right]$$

$$+ \frac{h^3}{48} \left[ 4y_n + 6 \frac{y_n y_{n-1}'}{f_n} - 3 \frac{(y_n')^3}{f_n^2} + 6 \frac{(y_n')^2}{f_n} \right] + O(h^4)$$

By substituting (4.6.37) into (4.6.36) we obtain

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2} y_n'' + \frac{h^3}{12} \left[ 3y_n''' + \frac{3(y_n'')^2}{2f_n} - 3y_n''g_n \right] + O(h^4)$$

From (4.6.13) and (4.6.38) it can be shown that the local truncation error for the MRMS trapezoidal formula is given by

$$LTE_{MRMS} = \frac{h^3}{12} \left[ 3y_n''g_n - \frac{3(y_n'')^2}{2f_n} - y_n''' \right]$$

132
Theorem 4. The MRMS trapezoidal formula (4.6.36) is L-stable.

Proof. By applying the MRMS trapezoidal formula (4.6.36) to the test equation in (4.6.6), we obtain

$$y_{n+1} = y_n + \frac{\tau}{\sqrt{2}} \left[ \left( \sqrt{2 - 2z + z^2} \right) y_{n+1} \right]$$ \hspace{1cm} (4.6.40)

As shown previously, we write equation (4.6.40) in the form

$$Q(z) = \frac{\sqrt{2}}{\sqrt{2} - z(2 - 2z + z^2)^{\frac{1}{2}}}$$ \hspace{1cm} (4.6.41)

Now, we set $$w = \sqrt{2} - z(2 - 2z + z^2)^{\frac{1}{2}}$$ and $$\text{Re}(z) = -x - iy$$ in equation (4.6.41) and by setting

$$\sinh(2r) = \frac{(2 + 2x)y}{2 + 2x + x^2}$$ \hspace{1cm} (4.6.42)

we obtain

$$(2 - 2z + z^2)^{\frac{1}{2}} = [(2 + 2x + x^2) - y^2] + i(2 + 2x)y$$ \hspace{1cm} (4.6.43)

or equation (4.6.43) can also be written as

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

$$= (2 + 2x + x^2) \left[ 1 + i \sinh(2r) \right]^{\frac{1}{2}}$$

$$= (2 + 2x + x^2) \left[ \cosh(r) + i \sinh(r) \right]$$ \hspace{1cm} (4.6.44)

From equation (4.6.41), we set

$$w = \sqrt{2} - z(2 - 2z + z^2)^{\frac{1}{2}}$$

$$= \sqrt{2} + (x + iy)(2 + 2x + x^2)^{\frac{1}{2}} \left[ \cosh(r) + i \sinh(r) \right]$$

$$= \left( \sqrt{2} + x\sqrt{2 + 2x + x^2} \cosh(r) - y\sqrt{2 + 2x + x^2} \sinh(r) \right)$$

$$+ i \left( x\sqrt{2 + 2x + x^2} \sinh(r) + y\sqrt{2 + 2x + x^2} \cosh(r) \right)$$

and we can show that

$$\left| w \right|^2 = 2 + x\sqrt{2 + 2x + x^2} \left[ x\sqrt{2 + 2x + x^2} \cosh(2r) + 2^{\frac{1}{2}} \sinh(r) \right]$$

$$+ \sqrt{2}y\sqrt{2 + 2x + x^2} \left[ \frac{1}{\sqrt{2}} y\sqrt{2 + 2x + x^2} \cosh(2r) - 2 \sinh(r) \right]$$ \hspace{1cm} (4.6.45)
By substituting $y$ from (4.6.42) into (4.6.45), we obtain

$$|w|^2 = 2 + x \sqrt{2 + 2x + x^2} \left[ x \sqrt{2 + 2x + x^2} \cosh(2r) + 2 \tanh(r) \right]$$

$$+ \frac{2}{(2 + 2x)} \frac{(2 + 2x + x^2)\sinh(r)}{\sqrt{2 + 2x}} \cosh(r) \cosh(2r) - 1$$

From equation (4.6.46), we can see that for $x > 0$

$$\frac{(2 + 2x + x^2)\tanh(r)}{\sqrt{2 + 2x}} \cosh(r) \cosh(2r) > 1$$

and since $|z| \geq x$, it follows that

$$|w| > 2 + x (2 + 2x + x^2)$$

and from (4.6.41) we obtain

$$|Q(z)| < \frac{2}{2 + x (2 + 2x + x^2)}$$

Thus from equation (4.6.48), $|Q(z)| < 1$ and for $\Re(z) < 0$ we can see that $|Q(z)| \to 0$ as $x \to \infty$. Thus, the MRMS trapezoidal formula (4.6.36) is L-stable.

4.6.2 Numerical Example

**Example 1:**

We consider the initial value problem

$$y' = -2y \quad y(0) = 1 \quad 0 \leq x \leq 1$$

where the exact solution is $y(x) = \exp(-2x)$. The absolute error in the numerical solution using this new class of modified trapezoidal formulae obtained with step-size $h = 0.01$ after every ten steps are shown in Table 4.18.
Table 4.18: The absolute error for solving equation (4.6.49) using the various modified trapezoidal formulas

<table>
<thead>
<tr>
<th>x</th>
<th>MAM</th>
<th>MGM</th>
<th>MH₂M</th>
<th>MCₐM</th>
<th>MCₐME-04</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.44493E-04</td>
<td>0.52598E-04</td>
<td>0.60703E-04</td>
<td>0.28284E-04</td>
<td>0.39085E-04</td>
</tr>
<tr>
<td>0.2</td>
<td>0.72858E-04</td>
<td>0.86131E-04</td>
<td>0.99403E-04</td>
<td>0.46314E-04</td>
<td>0.64002E-04</td>
</tr>
<tr>
<td>0.3</td>
<td>0.89479E-04</td>
<td>1.0578E-03</td>
<td>1.2208E-03</td>
<td>0.56879E-04</td>
<td>0.78602E-04</td>
</tr>
<tr>
<td>0.4</td>
<td>0.97682E-04</td>
<td>1.1548E-03</td>
<td>1.3327E-03</td>
<td>0.62093E-04</td>
<td>0.85808E-04</td>
</tr>
<tr>
<td>0.5</td>
<td>0.99722E-04</td>
<td>1.1819E-03</td>
<td>1.3639E-03</td>
<td>0.63547E-04</td>
<td>0.87819E-04</td>
</tr>
<tr>
<td>0.6</td>
<td>0.9823E-04</td>
<td>1.1612E-03</td>
<td>1.3401E-03</td>
<td>0.62435E-04</td>
<td>0.86282E-04</td>
</tr>
<tr>
<td>0.7</td>
<td>0.93823E-04</td>
<td>1.1092E-03</td>
<td>1.2801E-03</td>
<td>0.59638E-04</td>
<td>0.82417E-04</td>
</tr>
<tr>
<td>0.8</td>
<td>0.87792E-04</td>
<td>1.0379E-03</td>
<td>1.1978E-03</td>
<td>0.55804E-04</td>
<td>0.77119E-04</td>
</tr>
<tr>
<td>0.9</td>
<td>0.80865E-04</td>
<td>0.95599E-04</td>
<td>1.1033E-03</td>
<td>0.51400E-04</td>
<td>0.71034E-04</td>
</tr>
<tr>
<td>1</td>
<td>0.73565E-04</td>
<td>0.86975E-04</td>
<td>1.0038E-03</td>
<td>0.46759E-04</td>
<td>0.64621E-04</td>
</tr>
</tbody>
</table>

Example 2:
The second-order differential equation IVP given by
\[ y'' + 101y' + 100y = 0, \quad y(0) = 1.01, \quad y'(0) = -2 \] (4.6.50)
over the range \(0 \leq x \leq 1\).

The general solution for (4.6.50) is \( y(x) = Ae^{100x} + Be^{-x} \)
and the theoretical solution of problem (4.6.50) in the specified range is \( y(x) = 0.01 \exp(-100x) + \exp(-x) \).

The problem (4.6.50) can also be written as a system, i.e.,
\[
\begin{pmatrix}
y_1' \\
y_2'
\end{pmatrix} =
\begin{pmatrix}
0 & 1 \\
-100 & -101
\end{pmatrix}
\begin{pmatrix}
y_1 \\
y_2
\end{pmatrix},
\begin{pmatrix}
y_1(0) \\
y_2(0)
\end{pmatrix} = \begin{pmatrix}
1.01 \\
-2
\end{pmatrix}.
\] (4.6.51)

The matrix of the system given by equation (4.6.51) has the eigenvalues \( \lambda_1 = -100 \) and \( \lambda_2 = -1 \) and the solutions of system (4.6.51) are \( y_1(x) = y(x) = 0.01 \exp(-100x) + \exp(-x) \) and \( y_2(x) = z(x) = -\exp(-100x) - \exp(-x) \).

According to the analysis of the trapezoidal method for absolute stability we need \( -2.0 < h\lambda < 0 \) or \( -2.0 < -h < 0 \) and \( -2.0 < -100h < 0 \). These inequalities are both satisfied

135
only if \( h < \frac{2}{100} = 0.02 \) in order to get a decreasing solution. The absolute errors in the numerical solutions for solving equation (4.6.51) using the modified trapezoidal formulas \( \text{MAM}, \text{MGM}, \text{MH}_M, \text{MC}_M \) and \( \text{MC}_M \) obtained with \( h = 0.001 \) are shown in Table 4.19 after every hundred steps.

Table 4.19: The absolute error for solving equation (4.6.51) by the various modified trapezoidal formulas

<table>
<thead>
<tr>
<th>x</th>
<th>MAM</th>
<th>MGM</th>
<th>( \text{MH}_M )</th>
<th>( \text{MC}_M )</th>
<th>( \text{MC}_M )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>.93712E-07</td>
<td>.60644E-05</td>
<td>.12227E-04</td>
<td>.12359E-04</td>
<td>.41858E-05</td>
</tr>
<tr>
<td>0.2</td>
<td>.10856E-06</td>
<td>.54558E-05</td>
<td>.11024E-04</td>
<td>.11192E-04</td>
<td>.38037E-05</td>
</tr>
<tr>
<td>0.3</td>
<td>.14766E-06</td>
<td>.48779E-05</td>
<td>.99074E-05</td>
<td>.10158E-04</td>
<td>.34827E-05</td>
</tr>
<tr>
<td>0.4</td>
<td>.17833E-06</td>
<td>.43606E-05</td>
<td>.89031E-05</td>
<td>.92192E-05</td>
<td>.31885E-05</td>
</tr>
<tr>
<td>0.5</td>
<td>.20184E-06</td>
<td>.38976E-05</td>
<td>.80002E-05</td>
<td>.83672E-05</td>
<td>.29186E-05</td>
</tr>
<tr>
<td>0.6</td>
<td>.21925E-06</td>
<td>.34832E-05</td>
<td>.71886E-05</td>
<td>.75938E-05</td>
<td>.26713E-05</td>
</tr>
<tr>
<td>0.7</td>
<td>.23152E-06</td>
<td>.31124E-05</td>
<td>.64589E-05</td>
<td>.68919E-05</td>
<td>.24446E-05</td>
</tr>
<tr>
<td>0.8</td>
<td>.23947E-06</td>
<td>.27806E-05</td>
<td>.58031E-05</td>
<td>.62548E-05</td>
<td>.22369E-05</td>
</tr>
<tr>
<td>0.9</td>
<td>.24382E-06</td>
<td>.24838E-05</td>
<td>.52135E-05</td>
<td>.56766E-05</td>
<td>.20466E-05</td>
</tr>
<tr>
<td>1.0</td>
<td>.24516E-06</td>
<td>.22183E-05</td>
<td>.46837E-05</td>
<td>.51517E-05</td>
<td>.18722E-05</td>
</tr>
</tbody>
</table>

From Table 4.18, we can see that the modified formulas based on the contraharmonic mean \( \text{MC}_M \) gives greater accuracy. While the results in Table 4.19 as expected show that the linear modified trapezoidal formula based on the arithmetic mean (AM) for solving system of stiff equation gives more accuracy. However in the same class of nonlinear modified trapezoidal formulas, we see that the centroidal mean \( \text{MC}_M \) method performs better.
CHAPTER 5

THE PARALLEL IMPLEMENTATION OF THE RUNGE-KUTTA CONTRAHARMONIC MEAN METHOD WITH EXTRAPOLATION
We consider the initial value problem (IVP) for the first order ordinary differential equation (ODE) in the form

\[ y' = f(x, y), \quad y(x_0) = y_0 \]

in order to compute a numerical approximation value of \( y(x) \) by using parallel computers. To complete this numerical computation we use parallelizing strategies and extrapolation techniques. We consider the second order contraharmonic mean \((C_0M)\) method in the form

\[
y_{n+1} = y_n + h \left[ \frac{f_1^2 + f_2^2}{f_1 + f_2} \right]
\]

or

\[
y_{n+1} = y_n + h \left[ \frac{k_1^2 + k_2^2}{k_1 + k_2} \right]
\]

where

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + h, y_n + hk_1).
\]

In this chapter, we will discuss and compare of speedups obtained by using the contraharmonic mean \((C_0M)\) method with the extrapolation technique in both sequential and parallel programs. The parallel implementation of ODEs extrapolation methods will also be discussed using parallel programs developed by using the explicit data and task assignment (ExDaTa Schedule) software developed by Bahoshy [1992].

5.1 THE EXTRAPOLATION METHOD FOR ORDINARY DIFFERENTIAL EQUATIONS

To increase the accuracy of a second order ODE formula by extrapolation depends on the approximations

\[
C(h) = y_n = y(x_n; h)
\]

(5.1.1)
which are obtained from their respective formulae and the solution is assumed to have an asymptotic expansion of the form

\[ y(x; h) = y(x) + \sum_{\nu=0}^{\beta} C_{\nu}(x) h^\nu + O(h^{\beta+1}) \]  

(5.1.2)

where \( \beta \) and \( C_{\nu}(x) \) depend on the numerical formulae and \( \beta \) is an integer \( \geq 2 \). The asymptotic expansion for the Euler method was developed by Gragg [1965] with the expansion prototype equation of the form

\[ y' = Ay, \quad y(a) = \alpha \]

where \( A \) is a constant. The asymptotic expansion for the geometric mean method can be found in Sanugi [1986]. The coefficients of the Euler method, Geometric mean method (GM), Heronian mean method (HeM) and Contraharmonic mean (CoM) methods are listed in Table 5.2. The asymptotic expansion of the Contraharmonic mean method are now considered.

5.2 ASYMPTOTIC EXPANSION OF THE CONTRAHARMONIC MEAN (CoM) METHOD

We will now establish and develop the asymptotic expansion for the contraharmonic mean formulae for the prototype equation

\[ y' = Ay, \quad y(a) = \alpha \]

The second order contraharmonic mean (CoM) method in (5.1) can be written as

\[ y_{n+1} = y_n + h \left[ \frac{(y'_n)^3 + (y'_{n+1})^3}{y'_n + y'_{n+1}} \right] \]  

(5.2.1)

By putting \( y'_n = \lambda y_n \) in (5.2.1), we get
\[ y_{n+1} = y_n + h\lambda \left[ \frac{y_n^2 + y_{n+1}^2}{y_n + y_{n+1}} \right] \]  

(5.2.2)

By multiplying \([y_n + y_{n+1}]\) on both sides in equation (5.2.2)

\[ y_{n+1}y_n + (y_{n+1})^2 = y_n^2 + y_{n+1}y_n + h\lambda[y_n^2 + y_{n+1}^2] \]

which reduces to,

\[ (y_{n+1})^2 = y_n^2 + h\lambda[y_n^2 + y_{n+1}^2] \]

or

\[ \left( \frac{y_{n+1}}{y_n} \right)^2 = 1 + h\lambda \left[ 1 + \left( \frac{y_{n+1}}{y_n} \right)^2 \right] \]

\[ \left( \frac{y_{n+1}}{y_n} \right)^2 = 1 + h\lambda \]

or

\[ \frac{y_{n+1}}{y_n} = \sqrt{\frac{1 + h\lambda}{1 - h\lambda}} \]

by taking the positive sign we have

\[ y_{n+1} = y_n \sqrt{\frac{1 + h\lambda}{1 - h\lambda}} \]  

(5.2.3)

From the initial condition, \( y_0 = \alpha \) and by successive substitution, we get

\[ y_n = \alpha \left[ \frac{1 + h\lambda}{\sqrt{1 - h\lambda}} \right]^n \]  

(5.2.4)

Since \( x = a + nh \), we substitute \( n = \frac{x - a}{h} \) in equation (5.2.4) to obtain

\[ y(x; h) = \alpha \left[ \frac{1 + h\lambda}{\sqrt{1 - h\lambda}} \right]^\frac{x-a}{h} \]  

(5.2.5)

Taking logarithms on both sides of (5.2.5) gives,

\[ \ln[y(x; h)] = \ln \left\{ \alpha \left[ \frac{1 + h\lambda}{\sqrt{1 - h\lambda}} \right]^\frac{x-a}{h} \right\} \]

140
\[ y(x; h) = \alpha \exp \left\{ \left( \frac{x - a}{h} \right) \ln \left[ \frac{1 + h\lambda}{\sqrt{1 - h\lambda}} \right] \right\} \] (5.2.6)

and by putting \( h\lambda = z \) and using Mathematica we obtain
\[
\sqrt{1 + h\lambda} = (1 + z)^{\frac{1}{2}}
\]
\[
= 1 + \frac{z}{2} - \frac{z^2}{8} + \frac{5z^4}{128} - \frac{7z^5}{256} + O(z^6)
\]
and
\[
\frac{1}{\sqrt{1 - h\lambda}} = (1 - z)^{-\frac{1}{2}}
\]
\[
= 1 + \frac{z}{2} + \frac{3z^2}{8} + \frac{5z^3}{16} + \frac{35z^4}{128} + \frac{63z^5}{256} + O(z^6)
\]

Therefore the expression for
\[
\sqrt{1 + h\lambda} = (1 + z)^{\frac{1}{2}} (1 - z)^{-\frac{1}{2}}
\]
\[
= 1 + z + \frac{z^2}{2} + \frac{z^3}{2} + \frac{3z^4}{8} + \frac{3z^5}{8} + \frac{3z^6}{16} + \frac{5z^7}{16} + O(z^8)
\]

The equation (5.2.6) can thus be written as
\[
y(x; h) = \alpha \exp \left\{ \left( \frac{x - a}{h} \right) \ln \left[ 1 + z + \frac{z^2}{2} + \frac{z^3}{2} + \frac{3z^4}{8} + \frac{3z^5}{8} + \frac{5z^6}{16} + \frac{5z^7}{16} + \ldots \right] \right\}
\]...
(5.2.7)

where
\[
\ln(1 + w) = w - \frac{w^2}{2} + \frac{w^3}{3} - \frac{w^4}{4} + \frac{w^5}{5}
\] (5.2.8)

and putting \( w = z + \frac{z^2}{2} + \frac{z^3}{2} + \frac{3z^4}{8} + \frac{3z^5}{8} + \frac{5z^6}{16} + \frac{5z^7}{16} + \ldots \) in equation (5.2.8) we get,
\[
\ln(1 + w) = z + \frac{z^2}{2} + \frac{z^3}{5} + \frac{z^7}{7} + \ldots
\] (5.2.9)

By substituting equation (5.2.9) in equation (5.2.7) we obtain
\[
y(x; h) = \alpha \exp \left\{ \left( \frac{x - a}{h} \right) \left( z + \frac{z^3}{3} + \frac{z^5}{5} + \frac{z^7}{7} + \ldots \right) \right\}
\]
\[= \alpha \exp \left\{ \lambda (x - a) \left( 1 + \frac{z^2}{3} + \frac{z^4}{5} + \frac{z^6}{7} + \ldots \right) \right\} \]

\[= \sum_{j=0}^{\ell} C_j(x) h^j + 0(h^{\infty}) \quad \text{(5.2.10)}\]

To find \( C_j(x) \), we differentiate equation \((5.2.10)\) with respect to \( x \) to give

\[
\sum_{j=0}^{\ell} C_j' h^j = \lambda \left[ 1 + \frac{z^2}{3} + \frac{z^4}{5} + \frac{z^6}{7} + \ldots \right] \alpha \exp \left\{ \lambda (x - a) \left( 1 + \frac{z^2}{3} + \frac{z^4}{5} + \frac{z^6}{7} \right) \right\}
\]

\[= \lambda \left[ 1 + \frac{z^2}{3} + \frac{z^4}{5} + \frac{z^6}{7} + \ldots \right] \sum_{j=0}^{\ell} C_j(x) h^j \quad \text{(5.2.11)}\]

Equating the coefficients \( h^j, j = 0, 1, 2, 3, 4, 5 \) we obtain,

\[
h^0: \quad C_0'(x) = \lambda C_0(x) \quad \text{(5.2.12a)}
\]

\[
h^1: \quad C_1'(x) = \lambda C_1(x) \quad \text{(5.2.12b)}
\]

\[
h^2: \quad C_2'(x) = \lambda C_2(x) + \frac{\lambda^3}{3} C_0(x) \quad \text{(5.2.12c)}
\]

\[
h^3: \quad C_3'(x) = \lambda C_3(x) + \frac{\lambda^3}{3} C_1(x) \quad \text{(5.2.12d)}
\]

\[
h^4: \quad C_4'(x) = \lambda C_4(x) + \frac{\lambda^3}{3} C_2(x) + \frac{\lambda^5}{5} C_0(x) \quad \text{(5.2.12e)}
\]

\[
h^5: \quad C_5'(x) = \lambda C_5(x) + \frac{\lambda^3}{3} C_3(x) + \frac{\lambda^5}{5} C_1(x) \quad \text{(5.2.12f)}
\]

The initial values requires that \( C_0(a) = \alpha \) and \( C_j(a) = 0 \) for \( j \geq 1 \). The next task is to obtain \( C_j(x), j = 0, 1, 2, \ldots \) from the above differential equation.

Hence from \((5.2.12a)\) we have,

\[ C_0(x) = A \exp(\lambda x) \quad \text{where} \ A \ \text{is constant.} \]

Now at \( x=a, \ C_0(a) = \alpha \) we obtain

\[ A = \alpha \exp(-\lambda a) \]

Thus,

\[ C_0(x) = \alpha \exp(\lambda (x - a)). \quad \text{(5.2.13)} \]
From equation (5.2.12b), we have
\[ C_1(x) = 0 \quad \text{when} \quad C_1(a) = 0. \]

From equation (5.2.12c),
\[ C'_1(x) = \lambda C_1(x) + \frac{\lambda^3}{3} C_0(x) \]
\[ C'_1(x) = \lambda C_1(x) + \frac{\lambda^3}{3} \alpha \exp(\lambda(x - a)) \]

With the integrating factor \( \exp(-\int f(x)dx) = \exp(-\lambda x) \) and \( C_1(a) = 0 \), we get
\[ \frac{d}{dx}(C_1(x)\exp(-\lambda x)) = \frac{1}{3} \lambda^3 \alpha \exp(\lambda(x - a))dx \]
\[ = \frac{1}{3} \lambda^3 \alpha \exp(-\lambda a)x + C \]
\[ C_1(x) = \frac{1}{3} \lambda^3 \alpha \exp(\lambda(x - a))x + C \exp(\lambda x) \]

By substituting \( C_1(a) = 0 \), we have
\[ C = -\frac{1}{3} \lambda^3 \alpha a \exp(-\lambda a) \]

Thus,
\[ C_1(x) = \frac{\lambda^3}{3} (x - a) \alpha \exp(\lambda(x - a)) \quad (5.2.14) \]

From equation (5.2.12d) we have
\[ C_3(x) = 0, \quad \text{since} \quad C_1(x) = 0 \quad \text{and} \quad C_1(a) = 0 \quad (5.2.15) \]

From equation (5.2.12e),
\[ C'_4(x) = \lambda C_4(x) + \frac{\lambda^3}{3} C_2(x) + \frac{\lambda^5}{5} C_0(x) \]

By substituting \( C_0(x) \) and \( C_2(x) \) equation (5.2.12e) will give
\[ C'_4(x) = \lambda C_4(x) + \frac{\lambda^5}{9} (x - a) \alpha \exp(\lambda(x - a)) + \frac{\lambda^5}{5} \alpha \exp(\lambda(x - a)) \]
\[ \frac{d}{dx}(C_4 \exp(-\lambda x)) = \int \exp(-\lambda x) \left[ \frac{\lambda^5}{9} (x - a) \alpha \exp(\lambda(x - a)) + \frac{\lambda^5}{5} \alpha \exp(\lambda(x - a)) \right]dx \]
\[ = \int \left[ \frac{\lambda^5}{9} \alpha \exp(-\lambda a) - \frac{\lambda^6}{9} a \alpha \exp(-\lambda a) + \frac{\lambda^5}{5} \alpha \exp(-\lambda a) \right]dx \]
\[ C_i(x) = \frac{\lambda^6}{18} \alpha \exp(\lambda(x-a))x^2 - \frac{\lambda^6}{9} a\alpha \exp(\lambda(x-a))x + \frac{\lambda^5}{5} \alpha \exp(\lambda(x-a))x + C \exp(\lambda x) \]

By substituting \( C_i(a) = 0 \), we get

\[ C = \left[ \frac{\lambda^6}{18} \alpha a^2 - \frac{\lambda^5}{5} \alpha a \right] \exp(-\lambda a) \]

and

\[ C_i(x) = \frac{\lambda^6}{18} \alpha x^2 \exp(\lambda(x-a)) - \frac{\lambda^5}{9} a\alpha \exp(\lambda(x-a))x + \frac{\lambda^5}{5} \alpha \exp(\lambda(x-a))x \]

\[ + \left[ \frac{\lambda^6}{18} \alpha a^2 - \frac{\lambda^5}{5} \alpha a \right] \exp(\lambda(x-a)) \]

\[ = \frac{\lambda^6}{18} (x^2 + a^2) \alpha \exp(\lambda(x-a)) + \frac{\lambda^5}{5} (x-a)\alpha \exp(\lambda(x-a)) - \frac{\lambda^4}{9} ax\alpha \exp(\lambda(x-a)) \]

\[ = \frac{\lambda^6}{18} (x-a)^2 C_0(x) + \frac{\lambda^5}{5} (x-a)C_0(x) \] \hspace{1cm} (5.2.16)

From equation (5.2.12f) we have
\[ C_i(x) = 0 \] since \( C_i(x) = 0 \), \( C_i(x) = 0 \) and \( C_i(a) = 0 \). \hspace{1cm} (5.2.17)

Hence, the asymptotic expansion of equation (5.2.10) can be written as

\[ y(x;h) = C_0(x) + C_i(x)h + C_4(x)h^4 + \ldots \]

Higher degree coefficients can be obtained by continuing the substitutions and equating powers of \( h \). The coefficient \( C_v(x) \) for \( v = 0(1)5 \) are shown in Table 5.1.

<table>
<thead>
<tr>
<th>( C_v(x) ) for ( v = 0(1)5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_0(x) = \exp[\lambda(x-a)] \alpha )</td>
</tr>
<tr>
<td>( C_1(x) = 0 )</td>
</tr>
<tr>
<td>( C_2(x) = \frac{\lambda^5}{3} (x-a)C_0(x) )</td>
</tr>
<tr>
<td>( C_3(x) = 0 )</td>
</tr>
<tr>
<td>( C_4(x) = \frac{\lambda^6}{18} (x-a)^2 C_0(x) + \frac{\lambda^5}{5} (x-a)C_0(x) )</td>
</tr>
<tr>
<td>( C_5(x) = 0 )</td>
</tr>
</tbody>
</table>
For comparison the first six terms of the coefficients $C_v(x)$ of the asymptotic expansion of the Trapezoidal (AM method), Geometric Mean method (GM), Heronian mean method (HeAM) and Contraharmonic mean method (CoAM) are shown in Table 5.2.

Table 5.2: $C_i$ Coefficients of the four methods for the solution of $y' = \lambda y$ and $y(a) = \alpha$

<table>
<thead>
<tr>
<th>$C_i$</th>
<th>Contraharmonic Mean</th>
<th>Arithmetic Mean</th>
<th>Geometric Mean</th>
<th>Heronian Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_0$</td>
<td>$\alpha \exp[\lambda(x-a)]$</td>
<td>$\alpha \exp[\lambda(x-a)]$</td>
<td>$\alpha \exp[\lambda(x-a)]$</td>
<td>$\alpha \exp[\lambda(x-a)]$</td>
</tr>
<tr>
<td>$C_1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_2$</td>
<td>$\frac{\lambda^2}{3}(x-a)C_0$</td>
<td>$\frac{\lambda^3}{12}(x-a)C_0$</td>
<td>$-\frac{\lambda^3}{24}(x-a)C_0$</td>
<td>$\frac{\lambda^3}{24}(x-a)C_0$</td>
</tr>
<tr>
<td>$C_3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$C_4$</td>
<td>$\left(\frac{\lambda^6}{18}(x-a)^2 + \frac{\lambda^6}{288}(x-a)^2 + \frac{\lambda^6}{1152}(x-a)^2 + \frac{\lambda^6}{1152}(x-a)^2 + \frac{\lambda^6}{5760}(x-a)^2\right)C_0$</td>
<td>$\frac{\lambda^5}{80}(x-a)C_0$</td>
<td>$\frac{3\lambda^5}{640}(x-a)C_0$</td>
<td>$\frac{17\lambda^5}{5760}(x-a)C_0$</td>
</tr>
<tr>
<td>$C_5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5.3 ROMBERG EXTRAPOLATION

In Evans and Yaakub [1995] and Evans and Walz [1993] it was shown that extrapolation techniques can improve the accuracy of integration methods for solving initial value problems. The benefits of repeated extrapolation are clearly greatly enhanced if it happens that the asymptotic expansion for $C(h)$ and $T(h)$ contain only even powers of $h$.

If the asymptotic expansion of $C(h)$ or $T(h)$ has the form,

$$C(h) = C_0 + C_1h^2 + C_2h^4 + \ldots,$$
or \[ T(h) = T_0 + T_1 h + T_2 h^2 + T_3 h^3 + \ldots, \quad (5.3.1) \]

then the process of repeated extrapolation produces the following,
\[ p_i^{(0)} = T(h_i) = y(H; h_i) \quad i = 0, 1, 2, \ldots, s, \]
\[ p_i^{(j)} = p_i^{(j-1)} + \frac{(p_i^{(j-1)} - p_i^{(j-1)})}{\left(\frac{h}{h_{i+1}}\right)^2 - 1} \quad j = 1, 2, \ldots, s; \quad i = 0, 1, \ldots, s - j \quad (5.3.2) \]
giving \[ p_i^{(j)} = T_0 + O(h_i^{j+1}), \] \(H\) is the basic steplength and \(h_i = \frac{H}{N_i}\) such that \(h_i > 0\) and \(h_{i+1} < h_i\) and \(N \in \{\text{sequence of numbers}\} \) where \(N \in \{2, 4, 6, 8, 16, \ldots, 1024\}\) i.e. a modified form of the Romberg Sequence gives an extrapolation table of the form,
\[ p_0^{(0)} \quad p_0^{(3)} \]
\[ p_i^{(0)} \quad p_i^{(3)} \quad p_0^{(2)} \]
\[ p_2^{(0)} \quad p_2^{(3)} \quad p_0^{(1)} \]
\[ p_3^{(0)} \quad p_3^{(3)} \quad p_0^{(1)} \]
\[ \vdots \quad \vdots \quad \vdots \]
\[ p_s^{(0)} \]

This scheme is built up by generating at the \(s\) th stage the upward sloping diagonal commencing with \(p_s^{(0)}\) and to make this triangle as small as possible (when \(s\) is the smallest integer). This can be achieved for some pre-set tolerance \(TOL\) such that
\[ |p_s^{(0)} - p_s^{(0)}| < TOL \quad (5.3.4) \]
where \(s = \min \{1, 2, \ldots, j\}\).
5.4 PARALLEL IMPLEMENTATION OF ODE EXTRAPOLATION METHOD

The algorithm is made up of three parts: finding the initial estimate, finding the estimates as the interval is successively halved, and producing the tableau. The first two parts require finding the function values at the appropriate points.

5.4.1 Parallel Implementation of the ODE Extrapolation Method

Let $H$ be the initial interval and $n$ the initial number of divisions. Two subroutines of the algorithm are given in Figure 5.1 and 5.2 for solving the initial value problem

$$y' = f(x, y), \quad y(a) = \alpha$$

and the tabulation by extrapolation is omitted here.

```c
void find_func_value()
{
    int p, i, j, nprocs;
    nprocs = m_get_numprocs ( );
    for (i = m_get_myid() + 1 ; i <= levels - logp ; i += nprocs ) {
        p = (int)pow(2.0,(double) i);
        h[i] = H/p ;
        yy[i] = Y ;
        x[i] = x0 ;
        do {
            k1[i] = function ( x[i], yy[i] ) ;
            k2[i] = function ( x[i] + h[i] , yy[i] + h[i] * k1[i] ) ;
            yy[i] += h[i] *((k1[i] * k1[i] ) + (k2[i] * k2[i] ) (k1[i] + k2[i] ));
            x[i] += h[i] ;
        } while ( x[i] < xf ) ;
    }
    return ;
}
```

Figure 5.1 : Producing the initial estimate.
void tabulate (){  
  int i, j, k;  
  long fp;  
  for (i = 1; i < levels + 1; i++) {/*for each cell in column one */  
    tab[1][i] = yy(i);  
  }  
  for (i = 2, j = levels; j > 0; i++, j--) {  
    fp = (long) pow (2.0, (double) (i-1));  
    fp = fp * fp - 1;  
    for (k = 1; k <= j; k++)  
      tab[i][j] = tab[i][k+1] + (tab[i-1][k+1] - tab[i-1][k]) / fp;  
  }  
  return;  
}

Figure 5.2 : Producing the successive estimates and a tableau.

The numerical solution of the initial value problem
\[ y' = y, \quad y(0) = 1, \quad 0 \leq x \leq 10 \]  
with the exact solution \( y(x) = \exp(x) \) was evaluated using a parallel program partitioned in the way described above. Table 5.3 gives the times and the speedups obtained. The speedups and plotted in Figure 5.3, and show that the best speedup achieved was less then 2 no matter how many processors were used.

Table 5.3 : Times in seconds and speedups for \( x = 10 \)

<table>
<thead>
<tr>
<th>No. Processor</th>
<th>Sequential Time</th>
<th>Parallel Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.53</td>
<td>6.56</td>
<td>0.995</td>
</tr>
<tr>
<td>2</td>
<td>4.38</td>
<td>1.491</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3.75</td>
<td>1.741</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.50</td>
<td>1.866</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3.38</td>
<td>1.932</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.34</td>
<td>1.955</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3.31</td>
<td>1.973</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.30</td>
<td>1.979</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.29</td>
<td>1.985</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3.28</td>
<td>1.991</td>
<td></td>
</tr>
</tbody>
</table>
One example of a system of differential equations is

\[ y'(x) = y(x) - z(x) + 2x - x^2 - x^3 \]
\[ z'(x) = y(x) + z(x) - 4x^2 + x^3 \]  
(5.4.2)

with initial condition

\[
\begin{align*}
  y(0) &= 1 \\
  z(0) &= 0
\end{align*}
\]

and the exact solution of equation (5.4.2) is

\[ y(x) = e^x \cos(x) + x^3 \]
\[ z(x) = e^x \sin(x) - x^3 \]

We can write the equation (5.4.2) as the system of equations in the following table

<table>
<thead>
<tr>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
</table>
| \( y' \) | \( y_1 \) | \( y_1' = 1 \)
| \( y_2 \) | \( y_2 \) | \( y_2' = y_2 - y_3 + 2y_1 - y_1^2 - y_1'^2 \)
| \( y_3 \) | \( y_3 \) | \( y_3' = y_2 + y_3 - 4y_1^2 + y_1'^3 \)
and a subroutine in C for computing at RHS for above equation are

```c
void func (yy,F) 
double yy [], F []
{
    F [1] = 1.0;
}
```

The numerical values obtained by extrapolation method for solving equation (5.4.2) at point \( x = 1.0 \) is given in the following tableau

<table>
<thead>
<tr>
<th></th>
<th>2.468672</th>
<th>2.468691</th>
<th>2.468694</th>
<th>2.468694</th>
<th>2.468694</th>
<th>2.468694</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.468691</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
</tr>
<tr>
<td></td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
</tr>
<tr>
<td></td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
</tr>
<tr>
<td></td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
<td>2.468694</td>
</tr>
</tbody>
</table>

with exact solution is 2.468694E+00 and error 6.8027E-10.

Table 5.4: Times in seconds and speedups for solving problem (5.4.2)

<table>
<thead>
<tr>
<th>No. Processor</th>
<th>Sequential Time</th>
<th>Parallel Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.31</td>
<td>12.42</td>
<td>0.9911</td>
</tr>
<tr>
<td>2</td>
<td>8.35</td>
<td>1.4743</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7.24</td>
<td>1.7003</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6.87</td>
<td>1.7918</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6.70</td>
<td>1.8373</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6.66</td>
<td>1.8483</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6.62</td>
<td>1.8595</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>6.64</td>
<td>1.8539</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6.69</td>
<td>1.8401</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6.79</td>
<td>1.8131</td>
<td></td>
</tr>
</tbody>
</table>
The reason for this is that there are two loops in the algorithms procedure which are independent of each other, so that both can be executed concurrently. Secondly, the amount of work in the second loop is far greater than that in the first. The partitioning of the outer loop leads to uneven load balance and a limiting value of speed up equal to 2 (see Bahoshy [1992]). Partitioning the inner loop is straightforward but the granularity of the outer loop might be too small to make the parallelization of the inner loop worthwhile. Alternatively both parts of the algorithm can be executed concurrently by the use of function partitioning. These difficulties can be avoided by the use of the EXDATA submanager. The second loop can be partitioned at both the outer and inner level simultaneously with an accompanying increase in efficiency.
5.4.2 Parallelization Techniques

Explicit data and task assignment (ExDaTa Schedule) is a scheduling software package for parallel computers developed by Bahoshy [1992]. The software is a strategy for parallel programming which is based on separating and collecting the parallel control statements of a program into separate functions. In other words, it is based on a model of parallel execution which treats a parallel program as a collection of tasks with a schedule in which functional parallelism is performed.

5.4.3 Explicit Parallel Programming By ExDaTa Schedule

There are two ways of applying parallelism in a program, i.e., data partitioning and function partitioning. In the execution of a parallel program, the processors are assigned to perform different functions on different data on which to operate. In explicit parallel programming we require that the statements which carry out these assignments be logically and functionally separate from the routine themselves. The manager/worker technique can be used to achieve explicit partitioning. It is based on the use of two generic processes, a manager and several workers. A worker requests work from the manager and acts in accordance with the received instructions.

In data partitioning, the request will be for data and in function partitioning the request will be for functions. When a program is arranged for parallel processing, it has to be partitioned into tasks. Of course the program may be parallelized using both data and function partitioning, so that there might be different tasks, with different functions. The former will be called main tasks or just tasks and the latter sub-tasks. The process which performs the function partitioning will be called simply the manager and those which partition and assign the data will be called sub-managers. Clearly a parallel program with function and data partitioning can have more than one manager.
In general, there are three routines, `manager()` which is the partitioning routines, `worker()` which sends requests for work and receive the data, as well as the routine to be parallelized with data partitioning, `par()`. The full routines of these actions are shown in Figure 5.5. While the same procedure for parallelizing the function partitioning are shown in Figure 5.6.

```c
Manager()
{
    return next data to be operated on;
}
Worker()
{
    loop : data = manager();
    par(data);
}
par(){ ... }
```

Figure 5.5 : The manager/worker technique for data partitioning

```c
Manager()
{
    return next function to be executed;
}
Worker()
{
    loop : (manager())();
}
Par(){ ... }
```

Figure 5.6 : The manager/worker technique for function partitioning

### 5.4.4 ExDaTa : Scheduling Tool For Parallel Programming

The ExDaTa Schedule was built as a scheduling tool for parallel programming using both explicit function and data partitioning. ExDaTa is also built using existing parallel library routines. It is therefore not too difficult to incorporate other routines and facilities.
5.4.5 Parallel Implementation of the ODE Extrapolation Method Using the ExDaTa Schedule

Let us now to discuss the parallel implementation of the ODE extrapolation method using the ExDaTa Schedule with the second order Contraharmonic mean formula to solve the initial value problem

\[ y' = f(x, y) \quad y(a) = \alpha \]

From the previous result, the best speedup of 2 was obtained no matter how many processors were used. For a further increase in speedup, the inner loop should be scheduled where by using a suitable sub_manager technique a better result can be achieved. The main_task and sub_manager used in the ExDaTa Schedule are shown in Figure 5.7 and Figure 5.8.

```c
void ffv_ml (MAIN_TASK)
mtarg
{
    int p, fp, i, j;
    for (i = 1; i <= levels * logp; i++) {
        p = (int) pow (2.0, (double) i);
        h[i] = H/p;
        yy[i] = Y;
        x[i] = xo;
        do {
            k1[i] = function( x[i], yy[i] ) ;
            k2[i] = function( x[i] + h[i] * k1[i] ) ;
            yy[i] += h[i] * (( k1[i] *k1[i] ) +( k2[i]*k2[i] ) )/(k1[i] + k2[i] );
            x[i] += h[i];
        } while ( x[i] < xf ) ;
    }
    return;
}
```

Figure 5.7 : Main_Task program for solving \( y' = y \), \( y(0) = 1 \) at \( x = 1.0 \)
Sub_Man_ffv_sl_m (TASK)

smarg
{
  ismarg
int start_node, nnodes, splitter, l;
  if ( logp == 0 ) MORE_W = 0;
  if ( MORE_W ) {
    STOP_GO = 1;
    l = levels - logp + loop_var;
    splitter = (int) pow (2.0, (double)(loop_var - 1));
    start_node = (int) pow (2.0, (double) (l-1));
    nnodes = start_node / splitter;
    W_ARG_1 = l;
    W_ARG_2 = start_node + nnodes * (split - 1);
    W_ARG_3 = job.x2 + nnodes;
    W_ARG_4 = f;
    split ++ ;
    f += 2 * nnodes;
    if ( split > splitter ) {
      split = 1;
      loop_var ++ ;
      f = 1;
    }
    if ( loop_var > logp )
      MORE_W = 0;
  }
  else
    STOP_GO = 0;
  return JOB;
}

Figure 5.8 : Sub_Manager program for solving \( y' = y \), \( y(0) = 1 \) at \( x = 1.0 \)
The initial value problem

\[ y' = 0.92 \cosh(x) - \cos(x) \quad y(0) = 0 \quad 0 \leq x \leq 10 \quad (5.4.3) \]

with the exact solution \( y(x) = 0.92 \sinh(x) - \sin(x) \) was evaluated using a parallel program with the ExDaTa Schedule and Sub_Manager technique described above. Table 5.5 give the times and the speedup obtained for \( x = 1.0 \) and the speedup is plotted in Figure 5.9.

Table 5.5: Times in seconds and speedups for \( x = 1.0 \) in program with ExDaTa Schedule.

<table>
<thead>
<tr>
<th>No. Processor</th>
<th>Sequential Time</th>
<th>Parallel Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.24</td>
<td>5.24</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>2.69</td>
<td>1.948</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.69</td>
<td>1.948</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.34</td>
<td>3.910</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.37</td>
<td>3.825</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.37</td>
<td>3.825</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.37</td>
<td>3.825</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.72</td>
<td>7.28</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.72</td>
<td>7.28</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.70</td>
<td>7.49</td>
<td></td>
</tr>
</tbody>
</table>

From the experiments that we have completed on the above problem using the ExDaTa program we found that the speedups are best when the number of processors are a power of two i.e 1, 2, 4 and 8. The results for \( p = 3, 5, 6, 7, 9 \) and 10 processors are affected by unequal load balancing in the scheduling strategy chosen. The plot of speedup with number of processor of power 2 are shown in Figure 5.10.
Figure 5.9: Plot of speedups from Table 5.5

Figure 5.10: Plot of speedups from Table 5.5 with number of processors of power two
Now we attempt by using 3, 5, 6, 7, 9 and 10 processors with the ExDaTa Schedule to obtain an improved speedup. In the previous program the manager and sub_manager is replaced by the program shown in Figure 5.11. An outline of the Sub_Manager for partitioning the second task is shown in Figure 5.12. Note the use of the array variable Part_Points[] for holding the intermediate results.

```
Main_Task Ests (JOB)
mtarg
{
    double hl, x, y = 0.0;
    int l, nl, f, i;
    f = W_ARG_4;
    hl = h/(double)f;
    nl = f * n;
    x = a + (double)(W_ARG_2 - 1) * hl;
    for (i = W_ARG_2; i <= W_ARG_3; i+=2) {
        y += hl * ((integrand1(x)*integrand1(x)) + integrand1(x) + integrand1(x+hl)) / (integrand1(x)+integrand1(x+hl));
        x += hl + hl;
    }
    part_sums[W_ARG_1][m_get_myid () + l] += y;
    return;
}
```

Figure 5.11: Outline of the main task of implementing Figure 5.1
Sub_Man Ests_m(TASK)
smarg
{
int nl;

if(MORE_W) {
    if( level > levels ) {
        STOP_GO = 0;
        MORE_W = 0;
    }
    else {
        W_ARG_4 = (int) pow(2.0 , (double)( level - 1 ));
        nl = n * W_ARG_4 ;
        W_ARG_1 = level ;
        W_ARG_2 = ili ;
        if ( ili >= nl ) {
            W_ARG_3 = nl ;
            ili = 2 ;
            level ++ ;
        }
        else {
            W_ARG_3 = ili ;
            ili += 2 ;
        }
        STOP_GO = 1 ;
    }
}
else
    STOP_GO = 0 ;
return JOB ;
}
Producing the tableau requires the adding up of the values in the array part_sums[][]. Multiplying by half the size of the interval to give the first column of the tableau, the rest of the tableau is obtained by repeatedly applying the equation

\[ P_i^{(0)} = C(h_{i}) \]

\[ P_i^{(j)} = P_{i+1}^{(j-1)} + \left( \frac{P_{i+1}^{(j-1)} - P_{i}^{(j-1)}}{h_{i+1}} \right)^2 \]

where \( i=0,1,\ldots,s-j; j=1,2,\ldots,s \).

The number of elements in each successive column of the tableau is one less than that of the preceding column. The tableau itself is implemented in the same way as an upper triangular matrix. The amount of work required for producing the tableau is very small, since the number of elements in the first column level is very small.

If \( n \) is initially 10, say, then with \( l=10 \), \( h_{10} = (xf - xo)/10 \). The columns of the tableau have to be produced sequentially because each column depends on the previous column. The solution of the initial value problem in equation (5.4.3) using a parallel program with the ExDaTa Schedule including the Main_Task and Sub_Manager technique described above was carried out. Table 5.6 gives the times and speedups obtained for \( n=10 \) and \( l=10 \) and the speedups are plotted in Figure 5.13. For different values of \( n \) and \( l \) are shown respectively in Table 5.7 and Table 5.8. While the speedups obtained from both tables are plotted in Figure 5.14 and Figure 5.15.
Table 5.6: Times in seconds and speedups for \( n = 10 \) and levels = 10 using ExDaTa Schedule.

<table>
<thead>
<tr>
<th>No. Processor</th>
<th>Sequential Time</th>
<th>Parallel Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28.40</td>
<td>28.49</td>
<td>0.997</td>
</tr>
<tr>
<td>2</td>
<td>14.59</td>
<td>1.947</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>9.67</td>
<td>2.937</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7.45</td>
<td>3.812</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6.03</td>
<td>4.710</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5.44</td>
<td>5.221</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4.54</td>
<td>6.256</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4.08</td>
<td>6.961</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.77</td>
<td>7.533</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>7.889</td>
<td></td>
</tr>
</tbody>
</table>

Thus, we see that by applying the scheduling strategy to the inner nested loop we achieve an almost equal load balancing objective in distributing the tasks to all the processors which results in the improved results.

Figure 5.13: Plot of speedups from Table 5.6
Table 5.7: Times in seconds and speedups for $n = 15$ and levels = 15 using ExDaTa Schedule

<table>
<thead>
<tr>
<th>No. Processor</th>
<th>Sequential Time</th>
<th>Parallel Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1348.79</td>
<td>1350.31</td>
<td>0.999</td>
</tr>
<tr>
<td>2</td>
<td>675.28</td>
<td>1350.31</td>
<td>1.997</td>
</tr>
<tr>
<td>3</td>
<td>450.28</td>
<td>1350.31</td>
<td>2.995</td>
</tr>
<tr>
<td>4</td>
<td>340.75</td>
<td>1350.31</td>
<td>3.958</td>
</tr>
<tr>
<td>5</td>
<td>270.64</td>
<td>1350.31</td>
<td>4.984</td>
</tr>
<tr>
<td>6</td>
<td>225.93</td>
<td>1350.31</td>
<td>5.970</td>
</tr>
<tr>
<td>7</td>
<td>194.40</td>
<td>1350.31</td>
<td>6.938</td>
</tr>
<tr>
<td>8</td>
<td>169.87</td>
<td>1350.31</td>
<td>7.940</td>
</tr>
<tr>
<td>9</td>
<td>151.10</td>
<td>1350.31</td>
<td>8.926</td>
</tr>
<tr>
<td>10</td>
<td>136.21</td>
<td>1350.31</td>
<td>9.902</td>
</tr>
</tbody>
</table>

Figure 5.14: Plot of speedups from Table 5.7
Table 5.8: Times in seconds and speedups for $n = 16$ and levels = 16 using ExDaTa Schedule.

<table>
<thead>
<tr>
<th>No. Processor</th>
<th>Sequential Time</th>
<th>Parallel Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2844.33</td>
<td>2844.80</td>
<td>0.999</td>
</tr>
<tr>
<td>2</td>
<td>1430.03</td>
<td>1.989</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>951.70</td>
<td>2.989</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>714.28</td>
<td>3.982</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>571.65</td>
<td>4.976</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>475.29</td>
<td>5.984</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>408.70</td>
<td>6.959</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>356.53</td>
<td>7.985</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>318.20</td>
<td>8.939</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>285.89</td>
<td>9.949</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.15: Plot of speedups from Table 5.8
As might be expected from such a potentially highly parallel algorithm, the speedups achieved are very close to the ideal. From these experiments we have shown that by using a second order Contraharmonic mean \((C_M)\) method together with the extrapolation technique when solved on a parallel computer converges faster by the speedup factor than its counterpart on a sequential computer.
CHAPTER 6

THE THEORY OF RK(4,4) METHOD
In this chapter, we will described the techniques used in the adaptive implementation of numerical integration methods for assessing the step size in order to obtain better accuracy. The local truncation error using the formula $y_n = \phi(y_{n-1})$ is given by

$$y(x_n) - \phi(y(x_{n-1})) = C(x_n)h^{p+1} + o(h^{p+2})$$

where $h = x_n - x_{n-1}$.

This technique was proposed by Richardson [1927] and called "the deferred approach to the limit". The repeated integration from $x_{n-1}$ to $x_n$ with different step sizes results in the error estimate.

Butcher [1987] declared a modern approach with special methods RK(4,5) which are actually two methods built into one. The first method is of order $p$ and the second has order $p+1$. The difference between these methods provides an error estimate for the first method with order $p$. Error estimates by these methods have been derived by R.Merson [1957], Sarafyan [1966], E.Fehlberg [1968, 1969] and England [1969].

Now, we attempt to develop a new RK(4,4) strategy which are actually two different RK methods but of the same order $p$. The difference between these two approximations is taken to obtain an estimate of their accuracy. This approach is based on the use of the fourth order classical Runge-Kutta method and the Contraharmonic mean ($C_0M$) method (see Evans and Yaakub [1995]). The combination of these formula will be written as the RK(4,4) method.

### 6.1 RK(4,4) Method for Error-Estimate

The study of RK($p$, $p+1$) methods with a built in error estimate has been proposed by Merson [1957], Sarafyan [1966], Fehlberg [1969] and by a number of other authors where the method, is given in the form of a tableau, i.e.,
for order \( p + 1 \) and the method

\[
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
& \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,p-1} & \\
c_{n+1} & a_{n+1,1} & a_{n+1,2} & \cdots & a_{n+1,p-1} & a_{n+1,p} \\
\hline
b_1 & b_2 & \cdots & b_{n-1} & b_n & b_{n+1}
\end{array}
\] (6.1.1)

for order \( p \). The method in (6.1.2) is used to compute \( y_n \) from a given value of \( y_{n-1} \), whereas the difference of the results computed by these two methods is used for the error estimate.

We can also write the equations (6.1.1) and (6.1.2) as four stage methods with the Butcher array in the form

\[
\begin{array}{c|cccc}
0 & & & & \\
c_2 & a_{21} & & & \\
c_3 & a_{31} & a_{32} & & \\
c_4 & a_{41} & a_{42} & a_{43} & \\
\hline
b_1 & b_2 & b_3 & b_4
\end{array}
\] (6.1.3)

The fourth order classical Runge-Kutta method can be written in the Butcher array form as
where \( y_{n+1} = y_n + \frac{h}{3} \left( \frac{1}{2} k_1 + k_2 + \frac{1}{2} k_3 \right) \) (6.1.5)

\[
\begin{align*}
\begin{array}{c|cccc}
0 & 1 & 1 \\
\frac{1}{2} & \frac{1}{2} & \\
\frac{1}{2} & 0 & \frac{1}{2} \\
1 & 0 & 0 & 1 \\
\frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6}
\end{array}
\end{align*}
\]

(6.1.4)

and (6.1.6)

\[
\begin{align*}
k_1 &= f(x_n, y_n) \\
k_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_1) \\
k_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_2) \\
k_4 &= f(x_n + h, y_n + h k_3)
\end{align*}
\]

From equation (6.1.3), (6.1.5) and (6.1.6) the classical Runge-Kutta method with the Butcher array can also be written in the new form as

\[
\begin{align*}
\begin{array}{c|cccc}
0 & 1 & 1 \\
\frac{1}{2} & \frac{1}{2} & \\
\frac{1}{2} & 0 & \frac{1}{2} \\
1 & 0 & 0 & 1 \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{array}
\end{align*}
\]

(6.1.7)

and the fourth order RK contraharmonic mean method with the Butcher array in the form
where

\[ y_{n+i} = y_n + \frac{h}{3} \left[ \sum_{i=1}^{3} \left( \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right) \right] \]  

(6.1.9)

where \( k_i = f(x_n, y_n) \)

\[ k_2 = f \left( x_n + \frac{h}{2}, y_n + \frac{h}{2}k_i \right) \]

\[ k_3 = f \left( x_n + \frac{h}{2}, y_n + \frac{h}{8}k_i + \frac{3}{8}hk_2 \right) \]  

(6.1.10)

\[ k_4 = f \left( x_n + h, y_n + \frac{h}{4}k_i - \frac{3}{4}hk_3 + \frac{3}{2}hk_2 \right) \]

In Evans and Yaakub [1995], a new fourth order centroidal mean \((C_eM)\) method can also be written in the form

\[
\begin{array}{c|ccc|}
0 & \frac{1}{2} & 1 & \frac{1}{2} \\
\frac{1}{2} & 1 & \frac{3}{2} & 2 \\
\frac{1}{2} & \frac{3}{4} & \frac{1}{4} & 3 \\
1 & \frac{1}{4} & \frac{3}{4} & 2 \\
\hline
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 3 \\
\end{array}
\]

(6.1.8)
where \( y_{n+1} = y_n + \frac{h}{3} \left[ \sum_{i=1}^{3} \frac{2(k_i^2 + k_i k_{i+1} + k_{i+1}^2)}{3(k_i + k_{i+1})} \right] \) \hspace{1cm} (6.1.12)

where \( k_i = f(x_n, y_n) \)

\[
\begin{align*}
k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_i\right) \\
k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{1}{12} h k_i + \frac{11}{24} h k_2\right) \\
k_4 &= f\left(x_n + h, y_n + \frac{11}{132} h k_i - \frac{25}{132} h k_2 + \frac{77}{66} h k_3\right)
\end{align*}
\]

\( k_i = f(x_n, y_n) \)

\[
\begin{align*}
k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_i\right) \\
k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{1}{16} h k_i + \frac{7}{16} h k_2\right) \\
k_4 &= f\left(x_n + h, y_n + \frac{7}{56} h k_i - \frac{17}{56} h k_2 + \frac{33}{28} h k_3\right)
\end{align*}
\]

While in Yaakub and Evans [1993] the root mean square (RMS) method in the Butcher array can be written as

\[
\begin{array}{c|cccc}
0 & 0 \\
\frac{1}{2} & 0 & 1 \\
1 & 0 & 1 & 7 \\
1 & 0 & \frac{1}{16} & \frac{17}{16} & 33 \\
1 & 0 & \frac{7}{56} & \frac{17}{56} & \frac{33}{28} \\
1 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{array}
\]

\hspace{1cm} (6.1.14)

where \( y_{n+1} = y_n + \frac{h}{3} \left[ \sum_{i=1}^{3} \left( k_i^2 + k_{i+1}^2 \right) \right] \) \hspace{1cm} (6.1.15)

and \( k_i = f(x_n, y_n) \)

\[
\begin{align*}
k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2} k_i\right) \\
k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{1}{16} h k_i + \frac{7}{16} h k_2\right) \\
k_4 &= f\left(x_n + h, y_n + \frac{7}{56} h k_i - \frac{17}{56} h k_2 + \frac{33}{28} h k_3\right)
\end{align*}
\]

170
6.2 ERROR CONTROL AND STEP SIZE SELECTION IN THE RK(4,4) METHOD

The combination between the fourth order classical Runge-Kutta formula

\[ y_{aM} = y_n + \frac{h}{3} \sum_{i=1}^{3} \left( \frac{k_i + k_{i+1}}{2} \right) \]  \hspace{1cm} (6.2.1)

where \( k_i = f(x_n, y_n) \)

\[ k_1 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2} hk_1\right) \]
\[ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2} hk_2\right) \]
\[ k_3 = f(x_n + h, y_n + hk_3) \]

and the fourth order contraharmonic mean (C\(\alpha\)M) formula

\[ y_{C\alpha M} = y_n + \frac{h}{3} \sum_{i=1}^{2} \left( \frac{k_i^2 + k_{i+1}}{k_i + k_{i+1}} \right) \]  \hspace{1cm} (6.2.2)

where \( k_i = f(x_n, y_n) \)

\[ k_1 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2} hk_1\right) \]
\[ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2} hk_2\right) \]
\[ k_3 = f\left(x_n + \frac{3}{8} h k_1 + \frac{3}{8} h k_2\right) \]
\[ k_4 = f\left(x_n + h, y_n + \frac{1}{4} hk_1 - \frac{3}{4} h k_2 + \frac{3}{2} h k_3\right) \]

is called the RK(4,4) method. The difference between equation (6.2.1) and (6.2.2), i.e., \( |y_{aM} - y_{C\alpha M}| \) provides an error estimate for the fourth order classical Runge-Kutta.

Following Lambert [1991], the Merson idea is to derive Runge-Kutta methods of order \( p \) and \( p+1 \), which share the same set of vectors \( \{k_i\} \) and this process is known as embedded methods. The embedded methods is written in the Butcher array in the form as
The symbol defined by $C, A$ and $b^T$ has order $p$ and that defined by $C, A$ and $\hat{b}^T$ has order $(p + 1)$. The difference between the values for $y_{\ast i}$ obtained by these two methods is then an estimate of the local truncation error. Therefore, $E' = |b^T - \hat{b}^T|$ where $E' = [E_1, E_2, \ldots, E_n]$.

The RK(4,4) methods from equation (6.2.1) and (6.2.2) are written in the Butcher form of equation (6.2.3) as

\[
\begin{array}{c|cccc}
0 & & & & \\
\frac{1}{2} & & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & \\
1 & 0 & 0 & 1 & \\
\frac{1}{2} & \frac{1}{8} & \frac{3}{8} & & \\
\frac{1}{4} & & -\frac{3}{4} & \frac{3}{2} & \\
1 & & & & \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & \\
E^T & & & & \\
\end{array}
\]

where

\[
b^T = y_{\ast i}^{\text{AT}} = y_* + \frac{h}{3} \left[ \sum_{i=1}^{3} k_i + k_{i,1} \right]
\]

and

\[
\hat{b}^T = y_{\ast i}^{\text{CO}} = y_* + \frac{h}{3} \left[ \sum_{i=1}^{3} k_i^2 + k_{i,1}^2 \right].
\]

In the RK(4,4) method four stages are required to obtain the solution which only share the same set of
vectors \( k_1 \) and \( k_2 \) using in \( b^T \) and \( \hat{b}^T \) approximately, but \( k_3 \) and \( k_4 \) use \( b^T \). While \( k_3^* \) and \( k_4^* \) use \( \hat{b}^T \).

Lotkin [1951] and Ralston [1962] has provided an error estimate for the classical fourth order Runge-Kutta scheme as

\[
\left| \psi(x^*_n, y^*_n, h) \right| \leq \frac{73}{720} ML^4
\]

where \( M \) and \( L \) are positive constant.

From the above approach, we can also obtain an error estimate for the four stage explicit \( AM-C,M \) method of order four by implementing the local truncation error for the classical Runge-Kutta method and contraharmonic mean \((C_oM)\) methods.

For the classical fourth order Runge-Kutta method

\[
y_{*n}^{AM} = y_n + LTE_{AM}
\]

and for the \((C_oM)\) method

\[
y_{*n}^{C,M} = y_n + LTE_{C,M}
\]

where \( y_{*n}^{AM} \) and \( y_{*n}^{C,M} \) are the numerical approximations at \( x_{*n} \), obtained by the \( AM \) and \((C_oM)\) methods respectively and \( LTE_{AM} \) and \( LTE_{C,M} \) are the local truncation errors of the classical Runge-Kutta and the contraharmonic mean \((C_oM)\) methods.

The difference between the classical Runge-Kutta method and the contraharmonic mean \((C_oM)\) method give an error estimate for numerical approximations at \( x_{*n} \) by

\[
y_{*n}^{AM} - y_{*n}^{C,M} = LTE_{AM} - LTE_{C,M}
\]

The local truncation error involves \( y \) derivatives of the classical Runge-Kutta method and is given by

\[
LTE_{AM} = \frac{h^5}{2880} \left[ -24 f_{v}^4 + f_{v}^4 f_{y,v} + 2 f_{v}^3 f_{y,v} f_{v} - 6 f_{v}^2 f_{y,v} + 36 f_{v}^3 f_{y,v} f_{y,v} \right]
\]
while the local truncation error of the contraharmonic mean \((C_{0}M)\) method is given by

\[
LTE_{c,M} = \frac{h^3}{23040} \left[ -378 f_{yy}^4 - 8 f_{yy}^4 f_{y} + 4 f_{y}^4 f_{y} f_{yy} - 648 f_{y}^4 f_{yy}^2 - 303 f_{y}^2 f_{yy}^2 \right]
\]

(6.2.12)

The absolute difference between \(LTE_{AM}\) and \(LTE_{C_{0}M}\) is given by

\[
|LTE_{AM} - LTE_{C_{0}M}| = h^3 \left[ \left( -\frac{24}{2880} + \frac{378}{23040} \right) f_{yy}^4 + \left( \frac{1}{2880} + \frac{8}{23040} \right) f_{y}^4 f_{y} f_{yy} + \left( \frac{2880}{2880} + \frac{23040}{23040} \right) f_{y}^2 f_{yy}^2 \right]
\]

\[
+ \left( \frac{36}{2880} + \frac{303}{23040} \right) f_{y}^4 f_{y} f_{yy}^2 \]  

(6.2.13)

By following an argument suggested by Lotkin [1951], if we assume that the following bounds for \(f\) and its partial derivatives hold for \(x \in [a,b]\) and \(y \in [-\infty, \infty]\), we have

\[
|f(x,y)| < Q \quad , \quad \left| \frac{\partial^{i+j} f(x,y)}{\partial x^i \partial y^j} \right| < \frac{P^{i+j}}{Q^{i+j}} \quad , \quad i + j \leq p \]  

(6.2.14)

where \(P\) and \(Q\) are positive constants and \(p\) is the order of the method. In this case, we have \(p = 4\). Hence using (6.2.14), we obtain

\[
|f_s| < P
\]

\[
|f_s + f_{y}^4| < 2PQ
\]
\[ |f^4 f^4| < \frac{Q^3}{Q^3} \]
\[ |f^4 f^4| < \frac{Q^4 P^4}{Q^4} \]
\[ |f^4 f^4| < Q^4 P^4 \left(\frac{P^4}{Q^4}\right)^2 < P^4 Q \]
\[ (6.2.15) \]
\[ |f^4 f^4| < Q^4 P^4 \frac{P^4}{Q^4} \]

From equation (6.2.13) and (6.2.14) we obtain
\[ |LTE_{AM} - LTE_{GM}| \leq \frac{281}{4608} P^4 Q h^3 \]
\[ (6.2.16) \]

Hence,
\[ |\dot{y}_{nm} - \ddot{y}_{nm}| \leq \frac{281}{4608} P^4 Q h^3 \]
\[ (6.2.17) \]

If we suppose that the tolerance TOL, i.e. \( \varepsilon < 0.00005 \),
then by setting
\[ |\dot{y}_{nm} - \ddot{y}_{nm}| \leq \text{TOL} \]
the error control and step size selection can be
determined by (6.2.17) to give the formula
\[ \frac{281}{4608} P^4 Q h^3 < \text{TOL} \]
or
\[ h < \left[ \frac{16.4 \times \text{TOL}}{P^4 Q} \right]^{1/3} \].
\[ (6.2.18) \]

6.2.1 Local And Global Truncation Error

The primary types of error in the numerical solution
of ordinary differential equations are truncation error
and rounding error. Now, we compare the local truncation error (LTE) and global truncation error (GTE).
Definition 6.1 Local Truncation Error

The local truncation error at the point \( x_{n+1} \) is the difference between the computed value \( y_{n+1} \) and the value at the point \( x_{n+1} \) on the solution curve that goes through the point \((x_n, y_n)\).

The local truncation error at the point \( x_{n+1} \) is defined as \( y_{n+1} - y(x_{n+1}) \), where \( y(x) \) denotes the solution of the given initial value problem.

6.2.1.1 The Third Order Arithmetic Mean Method

The Taylor expansion around \( x = x_n \) is

\[
y(x_n + h) = y(x_n) + hf + \frac{h^2}{2} ff + \frac{h^3}{6} (ff' + f'f)
\]

\[
+ \frac{h^4}{24} (f''f + 4f'f', + f''') + O(h^5)
\]  

(6.2.19)

The third order arithmetic mean method is given in the form

\[
y_{n+1} = y_n + \frac{h}{2} \left( \frac{k_1 + k_2}{2} \right)
\]  

(6.2.20)

where \( k_1 = f(x_n, y_n) \)

\[
k_2 = f \left( x_n + \frac{2}{3} h, y_n + \frac{2}{3} hk_1 \right)
\]

\[
k_3 = f \left( x_n + \frac{2}{3} h, y_n + \frac{h}{3} k_1 + \frac{h}{3} k_2 \right)
\]

When we substitute \( k_i, 1 \leq i \leq 3 \) into equation (6.2.20), we obtain

\[
y(x_n + h) = y(x_n) + hf + \frac{h^2}{2} ff + \frac{h^3}{6} (ff' + f'f)
\]

\[
+ \frac{h^4}{24} \left( \frac{1}{27} f''f + \frac{1}{6} f''f' + f'' f' \right) + O(h^5).
\]  

(6.2.21)

Now we subtract equation (6.2.19) from equation (6.2.21), to obtain
The local truncation error is

\[ y_{n+1} - y(x_n + h) = \frac{h^4}{648} y^{(5)}(\xi) = O(h^4) \]

where \( \xi \in [x_n, x_n + h] \).

To evaluate the GTE for the third order arithmetic mean method, we first put

\[ \varepsilon_n = y_n - y(x_n) \quad \text{and} \quad \varepsilon_{n+1} = y_{n+1} - y(x_n + h) \]

We subtract equation (6.2.19) from equation (6.2.21) to obtain

\[
\varepsilon_{n+1} = \varepsilon_n + h \left[ f(x_n, y_n) - f(x_n, y(x_n)) \right] + \frac{h^3}{2} \left[ f(x_n, y_n) - f(x_n, y(x_n)) \right] + \frac{h^4}{648} y^{(5)}(\xi) 
\]

\[
\leq \varepsilon_n + hL\varepsilon_n + \frac{h^2}{2}L\varepsilon_n + \frac{h^3}{6}L\varepsilon_n + \frac{h^4}{648} y^{(5)}(\xi) 
\]

We assume that \( f \) satisfies a Lipschitz condition with constant \( L \) and that \( |y^{(5)}(x)| \leq M \) for all \( x \) in the interval of interest, then we can make the estimate

\[
|\varepsilon_{n+1}| \leq \left[ 1 + hL + \frac{h^2}{2}L + \frac{h^3}{6}L \right] |\varepsilon_n| + \frac{h^4}{648} M 
\]

\[
\leq [1 + C]|\varepsilon_n| + B 
\]

where \( C = hL \left( \sum_{p=1}^{5} \frac{h^{p-1}}{p!} \right) \), \( A = 1 + C \), \( B = \frac{h^4}{648} M \) and \( |y^{(5)}(x)| \leq M \).

A simple induction proof gives

\[
|\varepsilon_n| \leq A^n |\varepsilon_0| + \left( \sum_{i=0}^{n} A^i \right) B 
\]

i.e., for \( A \neq 1 \), where \( A = (1 + C) \) since \( \varepsilon_0 = 0 \) then from the geometric series we have
If we use the inequality 
\[1 + x \leq e^x\]
we now get
\[A^n = (1 + C)^n = \left(1 + hL\left(\sum_{p=1}^{3} \frac{h^{p-1}}{p!}\right)\right)^n \leq e^{cn} = e^{\frac{\sum h^{p-1}}{p!}}\]
\[= e^{\frac{u(x_n-x_{n-1})}{6}(\sum h^{p-1})} = e^{uL(x_n-x_{n-1})} (6.2.23)\]
where \(D = \sum_{p=1}^{3} \frac{h^{p-1}}{p!}\).

By inserting equation (6.2.23) into the inequality (6.2.22) for \(\varepsilon\), we finally get
\[|\varepsilon_n| \leq \frac{h^3}{648LD} M(e^{uL(x_n-x_{n-1})} - 1),\]

hence the GTE is \(O(h^3)\).

6.2.1.2 The Fourth Order Arithmetic Mean Method

The Taylor expansion series up to \(h^5\) at \(x = x_n\), can be written as
\[y(x_n + h) = y(x_n) + hf + \frac{h^2}{2} f_f + \frac{h^3}{6} (ff_f + f^2 f_f)\]
\[+ \frac{h^4}{24} (f^3 f_{yy} + 4f^2 f_f f_f + ff_f^2)\]
\[+ \frac{h^5}{120} (ff_f^4 + f^4 f_{yyy} + 7f^3 f_y f_{yy} + 11f^2 f^2 f_y f_y + 4f^3 f_{yy}^2). (6.2.24)\]
The fourth order arithmetic mean (AM) Runge-Kutta method is
\[y_{n+1} = y_n + \frac{h}{3} \left[ k_1 + \frac{k_2}{2} + \frac{k_2 + k_3}{2} + \frac{k_1 + k_4}{2} \right] (6.2.25)\]
where \(k_i = f(x_n, y_n)\)
\[ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \]

\[ k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \]

\[ k_4 = f(x_n + h, y_n + hk_3) \]

and local truncation error (LTE) is obtained by subtracting equation (6.2.24) from equation (6.2.25). The LTE for the fourth order arithmetic mean method was also discussed in Jayes [1993] and Sanugi [1986], i.e.,

\[ y_{n+1} - y(x_n + h) = \frac{h^4}{2880} \left[ -24 f'_x + f'_x f_{xx} + 2 f'_y f_{xy} - 6 f'_y f_{yy} + 36 f'_y f_{xy} \right] = 0(h^4) \]

By using the same procedure to evaluate the global truncation error (GTE) as for the third order classical Runge-Kutta method, we have

\[ \epsilon_{n+1} = \epsilon_n + h\left[ f(x_n, y_n) - f(x_n, y(x_n)) \right] + \frac{h^2}{2} \left[ f, f(x_n, y_n) - f, f(x_n, y(x_n)) \right] \]

\[ + \frac{h^3}{6} \left[ f, f, f(x_n, y_n) - f, f, f(x_n, y(x_n)) \right] \]

\[ + \frac{h^4}{24} \left[ f, f, f, f(x_n, y_n) - f, f, f, f(x_n, y(x_n)) \right] + \frac{h^5}{2880} y''(\xi) \]

\[ \leq \epsilon_n + hL \epsilon_n + \frac{h^2}{2} L \epsilon_n + \frac{h^3}{6} L \epsilon_n + \frac{h^4}{24} L \epsilon_n + \frac{h^5}{2880} y''(\xi) \]

\[ |\epsilon_{n+1}| \leq \left[ 1 + hL + \frac{h^2}{2} L + \frac{h^3}{6} L + \frac{h^4}{24} L \right] |\epsilon_n| + \frac{h^5}{2880} M \]

\[ \leq [1 + C]|\epsilon_n| + B \]

where \( C = hL \left( \sum_{p=1}^{r-1} \frac{h^{p-1}}{p!} \right) \), \( A = 1 + C \), \( B = \frac{h^5}{2880} M \) and \( |y''(x)| \leq M \)

A simple induction proof gives,

\[ |\epsilon_n| \leq A^n |\epsilon_0| + \left( \sum_{i=0}^{r-1} A^i \right) B \]

i.e., for \( A \neq 1 \), where \( A = (1 + C) \) since \( \epsilon_0 = 0 \) then from the geometric series we have

179
\[ |\varepsilon_n| \leq \left( \frac{A^n - 1}{A - 1} \right) B \]  
(6.2.26)

If we use the inequality
\[ 1 + x \leq e^x \]
we now get
\[
A^n = (1 + C)^n = \left( 1 + hL\left( \sum_{p=1}^{t} \frac{h^{p-1}}{p!} \right) \right)^n \leq e^{\nu_n} = e^{\frac{\sum k^{p+1}}{2h}} = e^{n(h(x))} \]
(6.2.27)

where \( D = \sum_{p=1}^{t} \frac{h^{p-1}}{p!} \).

By inserting equation (6.2.27) into the inequality (6.2.26) for \( \varepsilon_n \), we finally get
\[
|\varepsilon_n| \leq \frac{h^4}{2880LD} \cdot M(e^{n(h(x))} - 1)
\]
and therefore the GTE for the fourth order arithmetic mean (AM) method is \( O(h^4) \).

6.2.1.3 The Fourth Order Contraharmonic Mean Method

The fourth order contraharmonic mean \((C_0M)\) method can be expressed in the form
\[
y_{n+1} = y_n + \frac{h}{3} \left[ \frac{k_1^2 + k_2^2}{k_1 + k_2} + \frac{k_3^2 + k_4^2}{k_3 + k_4} \right]
\]
(6.2.28)

where \( k_1 = f(x_n, y_n) \)
\[
k_2 = f\left( x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1 \right)
\]
\[
k_3 = f\left( x_n + \frac{h}{2}, y_n + \frac{h}{8}k_1 + \frac{3}{8}hk_2 \right)
\]
\[
k_4 = f\left( x_n + h, y_n + \frac{h}{4}k_1 - \frac{3}{4}hk_2 + \frac{3}{2}hk_3 \right)
\]
and LTE is the difference between equation (6.2.24) and (6.2.28)
\[ y_{n+1} - y(x_n + h) = \frac{h^4}{23040} \left[ -378 f_n^4 - 8 f_n^4 f_{nn} + 4 f_n^3 f_{n} f_{nn} - 648 f_n^3 f_n^2 f_{nn} - 303 f_n^4 f_n^2 f_{n} f_{n} \right] \]
\[ = 0(h^5). \]

By using the same procedure to evaluate the GTE for the fourth order arithmetic mean (AM) method, we have

\[ \varepsilon_{n+1} \leq \varepsilon_n + hL \varepsilon_n + \frac{h^2}{2} L \varepsilon_n - \varepsilon_n + \frac{h^3}{6} L \varepsilon_n + \frac{h^4}{24} L \varepsilon_n + \frac{h^3}{23040} y''(\xi) \]

\[ |\varepsilon_{n+1}| \leq \left[ 1 + hL + \frac{h^2}{2} L + \frac{h^3}{6} L + \frac{h^4}{24} L \right] |\varepsilon_n| + \frac{h^3}{23040} M \]
\[ \leq [1 + C]|\varepsilon_n| + B \]

where \( C = hL \left( \sum_{p=1}^{\infty} \frac{h^{p-1}}{p!} \right) \), \( A = 1 + C \), \( B = \frac{h^3}{23040} M \) and \( |y''(x)| \leq M \)

A simple induction proof gives

\[ |\varepsilon_n| \leq A^n |\varepsilon_0| + \left( \sum_{i=1}^{n-1} A^i \right) B \]

i.e., for \( A \neq 1 \), where \( A = (1 + C) \) since \( \varepsilon_0 = 0 \) then from the geometric series we have

\[ |\varepsilon_n| \leq \left( \frac{A^n - 1}{A - 1} \right) B \]
\[ \text{(6.2.29)} \]

If we use the inequality \( 1 + x \leq e^x \)

we now get

\[ A^n = (1 + C)^n = \left( 1 + hL \left( \sum_{p=1}^{\infty} \frac{h^{p-1}}{p!} \right) \right)^n \leq e^{hL \left( \sum_{p=1}^{\infty} \frac{h^{p-1}}{p!} \right)} = e^{hL(n_x - x_0)} \]
\[ \text{(6.2.30)} \]

where \( D = \sum_{p=1}^{\infty} \frac{h^{p-1}}{p!} \).

By inserting equation (6.2.30) into the inequality (6.2.29) for \( \varepsilon \), we finally get

\[ |\varepsilon_n| \leq \frac{h^4}{23040LD} M(e^{hL(n_x - x_0)} - 1) \]
and therefore the GTE for the fourth order Contraharmonic
mean method is \( o(h^4) \).

From the above discussion, we can conclude that for
first order ODEs if the local truncation error (LTE) of a
numerical method is \( O(h^{p+1}) \) then the global truncation
error (GTE) is \( o(h^p) \). The estimate of the GTE that was
derived in the fourth order arithmetic mean (AM) and
Contraharmonic mean \((C_0 M)\) method, cannot be used for
practical error estimation or error control because the
value from the GTE is less accurate than the LTE.

6.2.2 Experimental Results For \( RK(4,4) \)

The following are the numerical results of testing
\( RK(4,4) \) on the error control on some of the sample
problems:

PROBLEM: 1 \( Y' + Y = 0 \)
INITIAL CONDITIONS \( X0=0, Y0=1 \)
EXACT SOLUTION \( Y=\exp(-X) \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>Exact Solution</th>
<th>Numerical Solution</th>
<th>Absolute Error</th>
<th>Estimate Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h = .50000 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.25000</td>
<td>.77880868D+00</td>
<td>.77880088D+00</td>
<td>.7810679D-05</td>
<td>.1680223D-05</td>
</tr>
<tr>
<td>.50000</td>
<td>.6065428D+00</td>
<td>.6065307D+00</td>
<td>.1216599D-04</td>
<td>.2617097D-05</td>
</tr>
<tr>
<td>.75000</td>
<td>.4723808D+00</td>
<td>.4723666D+00</td>
<td>.1421239D-04</td>
<td>.3057273D-05</td>
</tr>
<tr>
<td>1.0000</td>
<td>.3678942D+00</td>
<td>.3678794D+00</td>
<td>.1475824D-04</td>
<td>.3174651D-05</td>
</tr>
<tr>
<td>1.2500</td>
<td>.2865192D+00</td>
<td>.2865048D+00</td>
<td>.1436723D-04</td>
<td>.3090520D-05</td>
</tr>
<tr>
<td>1.5000</td>
<td>.2231436D+00</td>
<td>.2231302D+00</td>
<td>.1342712D-04</td>
<td>.2888240D-05</td>
</tr>
<tr>
<td>1.7500</td>
<td>.1737861D+00</td>
<td>.1737739D+00</td>
<td>.1219995D-04</td>
<td>.2624238D-05</td>
</tr>
<tr>
<td>2.0000</td>
<td>.1353461D+00</td>
<td>.1353353D+00</td>
<td>.1085872D-04</td>
<td>.2335706D-05</td>
</tr>
<tr>
<td>2.2500</td>
<td>.1054087D+00</td>
<td>.1053992D+00</td>
<td>.9513925D-05</td>
<td>.2046415D-05</td>
</tr>
<tr>
<td>2.5000</td>
<td>.8209323D-01</td>
<td>.8208500D-01</td>
<td>.8232766D-05</td>
<td>.1770819D-05</td>
</tr>
<tr>
<td>2.7500</td>
<td>.6393491D-01</td>
<td>.6392786D-01</td>
<td>.7052889D-05</td>
<td>.1517016D-05</td>
</tr>
<tr>
<td>3.0000</td>
<td>.4979360D-01</td>
<td>.4978707D-01</td>
<td>.5992170D-05</td>
<td>.1288848D-05</td>
</tr>
<tr>
<td>3.2500</td>
<td>.3877926D-01</td>
<td>.3877421D-01</td>
<td>.5055624D-05</td>
<td>.1087394D-05</td>
</tr>
<tr>
<td>3.5000</td>
<td>.3020162D-01</td>
<td>.3019738D-01</td>
<td>.4240217D-05</td>
<td>.9119994D-06</td>
</tr>
<tr>
<td>3.7500</td>
<td>.2352128D-01</td>
<td>.2351775D-01</td>
<td>.3538179D-05</td>
<td>.7609933D-06</td>
</tr>
<tr>
<td>4.0000</td>
<td>.1831858D-01</td>
<td>.1831564D-01</td>
<td>.2939254D-05</td>
<td>.6321681D-06</td>
</tr>
</tbody>
</table>
The following is a list of the numerical experiments performed. The notation NFC defines as the number of function evaluations.

**PROBLEM: 2 \ Y' + Y = 0**

**INITIAL CONDITIONS X=0,Y=1**

**EXACT SOLUTION Y=EXP(-X)**

<table>
<thead>
<tr>
<th>X</th>
<th>H</th>
<th>Y</th>
<th>EXACT</th>
<th>ABS. ERROR</th>
<th>NFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>.26591</td>
<td>.2659148</td>
<td>.766515E+00</td>
<td>.766504E+00</td>
<td>.106068E-04</td>
<td>8</td>
</tr>
<tr>
<td>.36591</td>
<td>.1000000</td>
<td>.693572E+00</td>
<td>.693562E+00</td>
<td>.966025E-05</td>
<td>16</td>
</tr>
<tr>
<td>.46591</td>
<td>.1000000</td>
<td>.627570E+00</td>
<td>.627561E+00</td>
<td>.879780E-05</td>
<td>24</td>
</tr>
<tr>
<td>.56591</td>
<td>.1000000</td>
<td>.567848E+00</td>
<td>.567840E+00</td>
<td>.801202E-05</td>
<td>32</td>
</tr>
<tr>
<td>.66591</td>
<td>.1000000</td>
<td>.513811E+00</td>
<td>.513803E+00</td>
<td>.729612E-05</td>
<td>40</td>
</tr>
<tr>
<td>.76591</td>
<td>.1000000</td>
<td>.464915E+00</td>
<td>.464908E+00</td>
<td>.664391E-05</td>
<td>48</td>
</tr>
<tr>
<td>.86591</td>
<td>.1000000</td>
<td>.420673E+00</td>
<td>.420667E+00</td>
<td>.604977E-05</td>
<td>56</td>
</tr>
<tr>
<td>.96591</td>
<td>.1000000</td>
<td>.380640E+00</td>
<td>.380635E+00</td>
<td>.550854E-05</td>
<td>64</td>
</tr>
<tr>
<td>1.06591</td>
<td>.1000000</td>
<td>.344418E+00</td>
<td>.344413E+00</td>
<td>.501553E-05</td>
<td>72</td>
</tr>
</tbody>
</table>
**PROBLEM 3:** \( Y' + 2X^2Y = 0 \)

**INITIAL CONDITIONS** \( X=0, Y=1 \)

**EXACT SOLUTION** \( Y = \exp(-X^2) \)

<table>
<thead>
<tr>
<th>( X )</th>
<th>( H )</th>
<th>( Y )</th>
<th>( \text{EXACT} )</th>
<th>( \text{ABS. ERROR} )</th>
<th>( \text{NFC} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02239</td>
<td>0.0223856</td>
<td>9.99499E+00</td>
<td>9.99499E+00</td>
<td>2.66454E-14</td>
<td>48</td>
</tr>
<tr>
<td>0.04121</td>
<td>0.0188275</td>
<td>9.98303E+00</td>
<td>9.98303E+00</td>
<td>1.01030E-13</td>
<td>56</td>
</tr>
<tr>
<td>0.06518</td>
<td>0.0239672</td>
<td>9.95761E+00</td>
<td>9.95761E+00</td>
<td>9.84768E-13</td>
<td>64</td>
</tr>
<tr>
<td>0.09604</td>
<td>0.0308584</td>
<td>9.90819E+00</td>
<td>9.90819E+00</td>
<td>1.00386E-11</td>
<td>72</td>
</tr>
<tr>
<td>0.13501</td>
<td>0.0389689</td>
<td>9.81938E+00</td>
<td>9.81938E+00</td>
<td>8.22475E-11</td>
<td>80</td>
</tr>
<tr>
<td>0.18315</td>
<td>0.0481424</td>
<td>9.67012E+00</td>
<td>9.67012E+00</td>
<td>5.27802E-10</td>
<td>88</td>
</tr>
<tr>
<td>0.24158</td>
<td>0.0584314</td>
<td>9.43309E+00</td>
<td>9.43309E+00</td>
<td>2.65218E-09</td>
<td>96</td>
</tr>
<tr>
<td>0.31171</td>
<td>0.0701269</td>
<td>9.07409E+00</td>
<td>9.07409E+00</td>
<td>9.99335E-09</td>
<td>104</td>
</tr>
<tr>
<td>0.39561</td>
<td>0.0838992</td>
<td>8.55127E+00</td>
<td>8.55127E+00</td>
<td>2.02847E-08</td>
<td>112</td>
</tr>
<tr>
<td>0.49561</td>
<td>0.1000000</td>
<td>7.82214E+00</td>
<td>7.82214E+00</td>
<td>9.32170E-08</td>
<td>120</td>
</tr>
<tr>
<td>0.59561</td>
<td>0.1000000</td>
<td>7.01350E+00</td>
<td>7.01350E+00</td>
<td>6.85975E-07</td>
<td>128</td>
</tr>
<tr>
<td>0.69561</td>
<td>0.1000000</td>
<td>6.16394E+00</td>
<td>6.16394E+00</td>
<td>2.16906E-06</td>
<td>136</td>
</tr>
<tr>
<td>0.79561</td>
<td>0.1000000</td>
<td>5.31002E+00</td>
<td>5.31002E+00</td>
<td>4.99300E-06</td>
<td>144</td>
</tr>
<tr>
<td>0.89561</td>
<td>0.1000000</td>
<td>4.48382E+00</td>
<td>4.48382E+00</td>
<td>9.53703E-06</td>
<td>152</td>
</tr>
<tr>
<td>0.99561</td>
<td>0.1000000</td>
<td>3.71120E+00</td>
<td>3.71118E+00</td>
<td>1.59881E-05</td>
<td>160</td>
</tr>
<tr>
<td>1.09561</td>
<td>0.1000000</td>
<td>3.01089E+00</td>
<td>3.01087E+00</td>
<td>2.42451E-05</td>
<td>168</td>
</tr>
</tbody>
</table>

**PROBLEM 4:** \( Y' + 3X^2Y = 0 \)

**INITIAL CONDITIONS** \( X=0, Y=1 \)

**EXACT SOLUTION** \( Y = \exp(-X^3) \)

<table>
<thead>
<tr>
<th>( X )</th>
<th>( H )</th>
<th>( Y )</th>
<th>( \text{EXACT} )</th>
<th>( \text{ABS. ERROR} )</th>
<th>( \text{NFC} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10000</td>
<td>0.1000000</td>
<td>9.99000E+00</td>
<td>9.99000E+00</td>
<td>3.12240E-07</td>
<td>16</td>
</tr>
<tr>
<td>0.19184</td>
<td>0.0918385</td>
<td>9.92965E+00</td>
<td>9.92965E+00</td>
<td>4.96714E-07</td>
<td>24</td>
</tr>
<tr>
<td>0.27295</td>
<td>0.0811133</td>
<td>9.79870E+00</td>
<td>9.79870E+00</td>
<td>5.78590E-07</td>
<td>32</td>
</tr>
<tr>
<td>0.34918</td>
<td>0.0762251</td>
<td>9.58320E+00</td>
<td>9.58320E+00</td>
<td>6.29045E-07</td>
<td>40</td>
</tr>
<tr>
<td>0.42369</td>
<td>0.0745160</td>
<td>9.26761E+00</td>
<td>9.26761E+00</td>
<td>6.69893E-07</td>
<td>48</td>
</tr>
<tr>
<td>0.49850</td>
<td>0.0748065</td>
<td>8.83488E+00</td>
<td>8.83488E+00</td>
<td>7.11656E-07</td>
<td>56</td>
</tr>
<tr>
<td>0.57525</td>
<td>0.0767501</td>
<td>8.26664E+00</td>
<td>8.26664E+00</td>
<td>7.45578E-07</td>
<td>64</td>
</tr>
<tr>
<td>0.65579</td>
<td>0.0805445</td>
<td>7.54248E+00</td>
<td>7.54248E+00</td>
<td>6.50008E-07</td>
<td>72</td>
</tr>
<tr>
<td>0.74281</td>
<td>0.0870162</td>
<td>6.63744E+00</td>
<td>6.63744E+00</td>
<td>3.62135E-07</td>
<td>80</td>
</tr>
<tr>
<td>0.84117</td>
<td>0.0983564</td>
<td>5.51466E+00</td>
<td>5.51466E+00</td>
<td>7.70462E-06</td>
<td>88</td>
</tr>
<tr>
<td>0.94117</td>
<td>0.1000000</td>
<td>4.34451E+00</td>
<td>4.34447E+00</td>
<td>3.22746E-05</td>
<td>96</td>
</tr>
<tr>
<td>1.04117</td>
<td>0.1000000</td>
<td>3.23477E+00</td>
<td>3.23468E+00</td>
<td>8.93874E-05</td>
<td>104</td>
</tr>
</tbody>
</table>
**PROBLEM: 5** $Y' - Y = 0$
INITIAL CONDITIONS $X=0, Y=1$
EXACT SOLUTION $Y=\exp(X)$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$H$</th>
<th>$Y$</th>
<th>EXACT</th>
<th>ABS. ERROR</th>
<th>NFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>.26591</td>
<td>.2659148</td>
<td>.130461E+01</td>
<td>.130462E+01</td>
<td>.115901E-04</td>
<td>8</td>
</tr>
<tr>
<td>.36591</td>
<td>.1000000</td>
<td>.144182E+01</td>
<td>.144183E+01</td>
<td>.129196E-04</td>
<td>16</td>
</tr>
<tr>
<td>.46591</td>
<td>.1000000</td>
<td>.159346E+01</td>
<td>.159347E+01</td>
<td>.144006E-04</td>
<td>24</td>
</tr>
<tr>
<td>.56591</td>
<td>.1000000</td>
<td>.176104E+01</td>
<td>.176106E+01</td>
<td>.160501E-04</td>
<td>32</td>
</tr>
<tr>
<td>.66591</td>
<td>.1000000</td>
<td>.194625E+01</td>
<td>.194627E+01</td>
<td>.17884E-04</td>
<td>40</td>
</tr>
<tr>
<td>.76591</td>
<td>.1000000</td>
<td>.215094E+01</td>
<td>.215096E+01</td>
<td>.199335E-04</td>
<td>48</td>
</tr>
<tr>
<td>.86591</td>
<td>.1000000</td>
<td>.237716E+01</td>
<td>.237718E+01</td>
<td>.222122E-04</td>
<td>56</td>
</tr>
<tr>
<td>.96591</td>
<td>.1000000</td>
<td>.262717E+01</td>
<td>.262719E+01</td>
<td>.247498E-04</td>
<td>64</td>
</tr>
<tr>
<td>1.06591</td>
<td>.1000000</td>
<td>.290347E+01</td>
<td>.290349E+01</td>
<td>.275753E-04</td>
<td>72</td>
</tr>
</tbody>
</table>

**PROBLEM: 6** $Y' + Y - X - 1 = 0$
INITIAL CONDITIONS $X=0, Y=1$
EXACT SOLUTION $Y=X + \exp(-X)$

<table>
<thead>
<tr>
<th>$X$</th>
<th>$H$</th>
<th>$Y$</th>
<th>EXACT</th>
<th>ABS. ERROR</th>
<th>NFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>.03901</td>
<td>.0390120</td>
<td>.100074E+01</td>
<td>.100075E+01</td>
<td>.824283E-05</td>
<td>32</td>
</tr>
<tr>
<td>.07186</td>
<td>.0328495</td>
<td>.100251E+01</td>
<td>.100252E+01</td>
<td>.797615E-05</td>
<td>40</td>
</tr>
<tr>
<td>.11518</td>
<td>.0433192</td>
<td>.100638E+01</td>
<td>.100639E+01</td>
<td>.763683E-05</td>
<td>48</td>
</tr>
<tr>
<td>.17217</td>
<td>.0569914</td>
<td>.101400E+01</td>
<td>.101401E+01</td>
<td>.720935E-05</td>
<td>56</td>
</tr>
<tr>
<td>.24531</td>
<td>.0731424</td>
<td>.102777E+01</td>
<td>.102777E+01</td>
<td>.668635E-05</td>
<td>64</td>
</tr>
<tr>
<td>.33685</td>
<td>.0915362</td>
<td>.105086E+01</td>
<td>.105087E+01</td>
<td>.606021E-05</td>
<td>72</td>
</tr>
<tr>
<td>.43685</td>
<td>.1000000</td>
<td>.108291E+01</td>
<td>.108292E+01</td>
<td>.542498E-05</td>
<td>80</td>
</tr>
<tr>
<td>.53685</td>
<td>.1000000</td>
<td>.112143E+01</td>
<td>.112144E+01</td>
<td>.485577E-05</td>
<td>88</td>
</tr>
<tr>
<td>.63685</td>
<td>.1000000</td>
<td>.116580E+01</td>
<td>.116581E+01</td>
<td>.434577E-05</td>
<td>96</td>
</tr>
<tr>
<td>.73685</td>
<td>.1000000</td>
<td>.121547E+01</td>
<td>.121547E+01</td>
<td>.388886E-05</td>
<td>104</td>
</tr>
<tr>
<td>.83685</td>
<td>.1000000</td>
<td>.126992E+01</td>
<td>.126992E+01</td>
<td>.347956E-05</td>
<td>112</td>
</tr>
<tr>
<td>.93685</td>
<td>.1000000</td>
<td>.132871E+01</td>
<td>.132871E+01</td>
<td>.311294E-05</td>
<td>120</td>
</tr>
<tr>
<td>1.03685</td>
<td>.1000000</td>
<td>.139142E+01</td>
<td>.139142E+01</td>
<td>.278459E-05</td>
<td>128</td>
</tr>
</tbody>
</table>
PROBLEM 7: \( Y' - X^2 \exp(X) = 0 \)

INITIAL CONDITIONS \( X=0, Y=4 \)

EXACT SOLUTION \( Y = X^2 \exp(X) - 2X \exp(X) + 2 \exp(X) + 2 \)

<table>
<thead>
<tr>
<th>( X )</th>
<th>( H )</th>
<th>( Y )</th>
<th>EXACT</th>
<th>ABS. ERROR</th>
<th>NFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10000</td>
<td>.10000</td>
<td>.400036E+01</td>
<td>.400036E+01</td>
<td>.452855E-07</td>
<td>16</td>
</tr>
<tr>
<td>.20000</td>
<td>.10000</td>
<td>.400310E+01</td>
<td>.400310E+01</td>
<td>.986423E-07</td>
<td>24</td>
</tr>
<tr>
<td>.30000</td>
<td>.10000</td>
<td>.401129E+01</td>
<td>.401129E+01</td>
<td>.161356E-06</td>
<td>32</td>
</tr>
<tr>
<td>.39820</td>
<td>.0981978</td>
<td>.402845E+01</td>
<td>.402845E+01</td>
<td>.228415E-06</td>
<td>40</td>
</tr>
<tr>
<td>.49097</td>
<td>.0927681</td>
<td>.405726E+01</td>
<td>.405726E+01</td>
<td>.287053E-06</td>
<td>48</td>
</tr>
<tr>
<td>.57721</td>
<td>.0862469</td>
<td>.409943E+01</td>
<td>.409943E+01</td>
<td>.333867E-06</td>
<td>56</td>
</tr>
<tr>
<td>.65706</td>
<td>.0798449</td>
<td>.415599E+01</td>
<td>.415599E+01</td>
<td>.370043E-06</td>
<td>64</td>
</tr>
<tr>
<td>.73109</td>
<td>.0740295</td>
<td>.422756E+01</td>
<td>.422756E+01</td>
<td>.397919E-06</td>
<td>72</td>
</tr>
<tr>
<td>.80000</td>
<td>.0689121</td>
<td>.431456E+01</td>
<td>.431456E+01</td>
<td>.419631E-06</td>
<td>80</td>
</tr>
<tr>
<td>.86446</td>
<td>.0644568</td>
<td>.441733E+01</td>
<td>.441732E+01</td>
<td>.436814E-06</td>
<td>88</td>
</tr>
<tr>
<td>.92504</td>
<td>.0605795</td>
<td>.453613E+01</td>
<td>.453613E+01</td>
<td>.450650E-06</td>
<td>96</td>
</tr>
<tr>
<td>.98222</td>
<td>.0571892</td>
<td>.467123E+01</td>
<td>.467123E+01</td>
<td>.461972E-06</td>
<td>104</td>
</tr>
<tr>
<td>1.03643</td>
<td>.0542044</td>
<td>.482287E+01</td>
<td>.482287E+01</td>
<td>.471376E-06</td>
<td>112</td>
</tr>
</tbody>
</table>

6.3 ANALYSIS OF VARIETY OF MEANS METHOD

6.3.1 Elementary Differentials, Trees and Operation Diagram

In the numerical computations involving ODEs, we compute \( y'(x) \), \( y''(x) \), \( y'''(x) \) or in the vector form: \( f \), \( f'(f), f''(f,f), f'''(f'(f)) \) where \( f = f(y(x)), f' = f'(y(x)), f'' = f''(y(x)) \) are termed elementary differentials and can be shown in the form of rooted trees and operation diagrams.

Tree

[Diagram]

Elementary Differential

[Diagram]

Operation Diagram

[Diagram]
For any rooted tree the operation diagram corresponds to an elementary differential. For example,
\[ f^{(2)} \left( f^{(2)} \left( f, f^{(1)}(f) \right), f^{(3)}(f, f, f) \right) \]
can be written as the operation diagram

Let \( f_{i_{h_{i^{-i}}}} \) denote an nth order partial derivative of component number \( i \) of \( f \), where
\[
\begin{align*}
(f) &= f' \\
(f')(f) &= f'f' \\
(f'')(f, f) &= f'f'f
\end{align*}
\]
For the above example given, we obtain the labelled tree

and the vector component, can be written as
\[
\begin{align*}
&f_{j_{i_{k_{l_{m_{n_{o_{p_{q_{}}}}}}}}}}} \\
&f_{k_{l_{m_{n_{o_{p_{q_{}}}}}}}}
\end{align*}
\]
where the number of the system start from left to right.

Following Lambert [1991], the Butcher rooted tree (henceforth, just 'tree') of order n is a set of n points or nodes joined by lines to give a picture as above. There must be just one root and branches are not allowed to grow together, i.e.

are forbidden.
We also can join two or more trees on a new root to produce a new tree, thus

Example 1

Example 2

In this example all the trees of orders up to 4 are shown in Table 1.

<table>
<thead>
<tr>
<th>Order</th>
<th>Trees</th>
<th>Number of Trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

188
From Table 6.1, we can obtain the further orders and the number of trees by the following rule. Let \( a_n \) be the number of trees of order \( n \). Then \( a_1, a_2, \ldots \), satisfy termwise the identity

\[
a_1 + a_2 u + a_3 u^2 + \ldots = (1-u)^{-1}(1-u^2)^{-1}(1-u^3)^{-1} \ldots \tag{6.3.1}
\]

From the equation (6.3.1), we obtain the following table:

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_n )</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>20</td>
<td>48</td>
<td>115</td>
</tr>
</tbody>
</table>

Table 6.2

From Table 6.2, we can show that all nine trees are of order 5.

6.3.2 Comparison of RKF(4,5), Merson and RK(4,4) Methods

RKF(4,5) is a subroutine for solving initial value problems in ODEs. It is based on the RK formulae developed by E. Fehlberg in 1969 and implemented by L.F. Shampine and H.A. Watts in 1974. It requires six function evaluations per step, four of these function values are combined with a set of coefficients to produce a fourth order method, and all six values are combined with another set of coefficients to produce a fifth order method. As in many other popular RK routines, the RKF(4,5) adopts error per step criterion and local extrapolation.
The RKF(4,5) method of order 4 is of Runge-Kutta type and uses the formula

\[ y_{n+1} = y_n + h \left[ \frac{25}{216} k_1 + \frac{1408}{2565} k_2 + \frac{2197}{4104} k_3 - \frac{1}{5} k_4 \right] \]  \hspace{1cm} (6.3.2)

where \( k_i = f(x_n, y_n) \)

\[ k_2 = f(x_n + \frac{1}{4} h, y_n + \frac{1}{4} hk_1) \]

\[ k_3 = f(x_n + \frac{3}{8} h, y_n + \frac{3}{32} hk_1 + \frac{9}{32} hk_2) \]

\[ k_4 = f(x_n + \frac{12}{13} h, y_n + \frac{1932}{2197} hk_1 - \frac{7200}{2197} hk_2 + \frac{7296}{2197} hk_3) \]

\[ k_5 = f(x_n + h, y_n + \frac{439}{216} hk_1 - 8hk_2 + \frac{3680}{513} hk_3 - \frac{845}{4104} hk_4) \]

This scheme requires one more function evaluation, i.e.

\[ k_s = f\left(x_n + \frac{1}{2} h, y_n - \frac{8}{27} hk_1 + 2hk_2 - \frac{3544}{2565} hk_3 + \frac{1859}{4104} hk_4 - \frac{11}{50} hk_5\right) \]

to obtain a Runge-Kutta method of order 5, i.e.

\[ y_{n+1} = y_n + h \left[ \frac{16}{135} k_1 + \frac{6656}{12825} k_2 + \frac{28561}{56430} k_3 - \frac{9}{50} k_4 + \frac{2}{55} k_5 \right] \]  \hspace{1cm} (6.3.3)

Comparison of the two values \((y_{n+1}, y_{n+1}')\) yields an error estimate of the local truncation error in the fourth order procedure is used for step size control.

The most popular embedded method in the form \((p, p+1)\) or \((4,5)\) method is RKF(4,5), one of a class of methods developed by Fehlberg (1968, 1969). The RKF(4,5) method in equation (6.3.2-6.3.3) or as in Butcher array is written in the form
which share the same set of vectors \( \{ k_i \} \) and five stages are required to obtain the solution. While six stages are required when the error estimate is used. The last row proposed by Butcher [1987] represents an estimate of the local truncation error (LTE), i.e,

\[
LTE = h \left[ k_i \frac{k}{360} - \frac{128}{4275} k_i - \frac{2197}{75240} k_i + \frac{k_i}{50} + \frac{2}{55} k_i \right]
\]  

(6.3.4)

Following Lambert [1991], an early example of a Runge-Kutta method with an error estimate in terms of the computed values \( k_i \) was proposed by Merson [1957]. Merson's method is defined by the Butcher array

\[
\begin{array}{c|cccc}
0 & 0 & 0 & \frac{2}{3} & \frac{1}{6} \\
\frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{6} \\
\frac{1}{3} & 0 & \frac{1}{6} & \frac{1}{6} & 0 & \frac{1}{6} \\
\frac{1}{2} & 0 & 0 & \frac{3}{8} & \frac{3}{8} & 0 \\
1 & 0 & -\frac{1}{2} & 2 & 0 & 0 \\
\end{array}
\]  

(6.3.5)
which is a 5-stage method of order 4. Merson proposed the principal local truncation error (LTE), i.e.,

\[ LTE = \frac{h}{30}[-2k_i + 9k_i - 8k_i + k_i] \]  

(6.3.6)

It is clearly seen that, both the RK(4,4) and RKF(4,5) methods in equation (6.2.1-6.2.2) and (6.3.2-6.3.3) for fourth order embedded method type use a minimum of six stages, whilst the Merson method in equation (6.3.5-6.3.6) use only 5 stages.

From Table 2, we can see that for RKF(4,5) and the Merson method only 9 trees are used but for RK(4,4) only 8 trees are used.

6.3.3 Experimental Results For RKF(4,5)

The following are the numerical results of testing RKF(4,5) for error control on the sample problems:

Problem 1: \( Y' + Y = 0 \)

Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution: \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>( x )</th>
<th>Exact Solution</th>
<th>Numerical Solution</th>
<th>Absolute Error</th>
<th>Estimate Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h = .50000 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( h = .25000 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( .50000 )</td>
<td>( .7787992D+00 )</td>
<td>( .7788008D+00 )</td>
<td>( .1579345D-05 )</td>
<td>( .1369379D-05 )</td>
</tr>
<tr>
<td>( .50000 )</td>
<td>( .6065282D+00 )</td>
<td>( .6065307D+00 )</td>
<td>( .2459988D-05 )</td>
<td>( .1066471D-05 )</td>
</tr>
<tr>
<td>( .75000 )</td>
<td>( .4723637D+00 )</td>
<td>( .4723666D+00 )</td>
<td>( .2873758D-05 )</td>
<td>( .8305667D-06 )</td>
</tr>
<tr>
<td>( 1.00000 )</td>
<td>( .3678765D+00 )</td>
<td>( .3678794D+00 )</td>
<td>( .2984111D-05 )</td>
<td>( .6468447D-06 )</td>
</tr>
<tr>
<td>( 1.25000 )</td>
<td>( .2865019D+00 )</td>
<td>( .2865048D+00 )</td>
<td>( .2905032D-05 )</td>
<td>( .5037621D-06 )</td>
</tr>
<tr>
<td>( 1.50000 )</td>
<td>( .2231274D+00 )</td>
<td>( .2231302D+00 )</td>
<td>( .2714927D-05 )</td>
<td>( .3923295D-06 )</td>
</tr>
</tbody>
</table>

| \( h = .50000 \) | | | | |
| \( h = .25000 \) | | | | |
| \( 1.75000 \) | \( .1737715D+00 \) | \( .1737739D+00 \) | \( .2466782D-05 \) | \( .3055459D-06 \) |

| \( h = .50000 \) | | | | |
| \( 2.25000 \) | \( .1053872D+00 \) | \( .1053992D+00 \) | \( .1197584D-04 \) | \( .8267373D-05 \) |
| \( 2.75000 \) | \( .6391424D-01 \) | \( .6392786D-01 \) | \( .1361932D-04 \) | \( .5013917D-05 \) |
| \( 3.25000 \) | \( .3876209D-01 \) | \( .3877421D-01 \) | \( .1211502D-04 \) | \( .3040792D-05 \) |
| \( 3.75000 \) | \( .2350806D-01 \) | \( .2351775D-01 \) | \( .9685763D-05 \) | \( .1844150D-05 \) |
| \( 4.25000 \) | \( .1426594D-01 \) | \( .1426423D-01 \) | \( .7292416D-05 \) | \( .1118422D-05 \) |
| \( 4.75000 \) | \( .8646412D-02 \) | \( .8651695D-02 \) | \( .5282869D-05 \) | \( .6782900D-06 \) |
| \( 5.25000 \) | \( .5243793D-02 \) | \( .5247518D-02 \) | \( .3725663D-05 \) | \( .4113628D-06 \) |

192
6.3.4 Experimental Results For The Merson Method

The following are the numerical results of testing the Merson Method for error control on the sample problems:

**Problem 1:** \( Y' + Y = 0 \)

Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution: \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>x</th>
<th>Exact Solution</th>
<th>Numerical Solution</th>
<th>Absolute Error</th>
<th>Estimate Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>\hbox{h=1.00000}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.25000</td>
<td>.1916001D-02</td>
<td>.1930454D-02</td>
<td>.1445294D-04</td>
<td>.9243865D-05</td>
</tr>
<tr>
<td>7.25000</td>
<td>.7000774D-03</td>
<td>.7101744D-03</td>
<td>.1009703D-04</td>
<td>.3377566D-05</td>
</tr>
<tr>
<td>8.25000</td>
<td>.2557975D-03</td>
<td>.2612586D-03</td>
<td>.5461061D-05</td>
<td>.1234111D-05</td>
</tr>
<tr>
<td>9.25000</td>
<td>.9346447D-04</td>
<td>.9611165D-04</td>
<td>.2647182D-05</td>
<td>.4509251D-06</td>
</tr>
<tr>
<td>\hbox{h=2.00000}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.25000</td>
<td>.2396525D-05</td>
<td>.1300730D-04</td>
<td>.1061077D-04</td>
<td>.6710270D-05</td>
</tr>
</tbody>
</table>

6.3.4 Experimental Results For The Merson Method

The following are the numerical results of testing the Merson Method for error control on the sample problems:

**Problem 1:** \( Y' + Y = 0 \)

Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution: \( y = \exp(-x) \)
The following is a list of sample problems used in the numerical experiments. The notation NPB defines the number of problem solution. The comparison of the time taken and accuracy between the RK(4,4) and RKF(4,5) methods are shown in Table 6.3.

Problem 2 (NPB 1): \( y' + 2xy = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)
Exact solution: \( y = \exp(-x^2) \)

Problem 3 (NPB 2): \( y' + 3x^2y = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)
Exact solution: \( y = \exp(-x^3) \)

Problem 4 (NPB 7): \( y' + y - x - 1 = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)
Exact solution: \( y = x + \exp(-x) \)

Problem 5 (NPB 10): \( y' - x^3\sin(x) + \frac{1}{x} + 1 = 0 \)
Initial conditions: \( x_0 = 1 \), \( y_0 = 4 \)
Exact solution: \( y = -x - \log(x) + x^4\cos(x) - 2x\sin(x) - 2\cos(x) + C \)

Problem 6 (NPB 12): \( y' + \ln(x^2) = 0 \)
Initial conditions: \( x_0 = 1 \), \( y_0 = 2 \)
Exact solution: \( y = -2(x\ln(x) - x) \)

Problem 7 (NPB 4): \( y' - y = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)
Exact solution: \( y = \exp(x) \)

Problem 8 (NPB 18): \( y' + y - x^2 - 1 = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)
Exact solution: \( y = -2\exp(-x) + x^2 - 2x + 3 \)

In the RK(4,4) with error control program, we choose the error estimate as the difference between the fourth order classical arithmetic mean (AM) method and the
contraharmonic mean \((C_{OM})\) method. These error estimates (ERREST) used together with a constant derived in equation (6.2.1 - 6.2.14) are of the form,

\[
ERREST = |y_{\text{est}} - y_{\text{CM}}| \times \frac{281}{4608} \quad (6.3.7)
\]

By using the error estimate in equation (6.3.7), the comparison of the time taken and the accuracy between solutions from the RK(4,4) and RKF(4,5) methods are shown in Table 6.3.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>RKF(4,5)</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
<td>1.11</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>0.20</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.04</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>8</td>
<td>0.20</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6.3

By using the error estimate in equation (6.3.7), the comparison of the time taken and the accuracy between solutions from the RK(4,4) and Merson methods are shown in Table 6.4.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>Merson</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
<td>0.2258x10^-5</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
<td>0.3927x10^-5</td>
</tr>
<tr>
<td>3</td>
<td>0.20</td>
<td>0.1389x10^-5</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.8308x10^-6</td>
</tr>
<tr>
<td>5</td>
<td>0.26</td>
<td>0.3253x10^-6</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.2949x10^-6</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.5516x10^-4</td>
</tr>
<tr>
<td>8</td>
<td>0.20</td>
<td>0.1759x10^-3</td>
</tr>
</tbody>
</table>

Table 6.4
By using the error estimate in equation (6.3.7), the comparison of the time taken and the accuracy between solutions from the RK(4,4), Merson and RKF(4,5) methods are shown in Table 6.5 and Table 6.6.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>RKF(4,5)</td>
<td>Merson</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
<td>1.11</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
<td>0.10</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>0.20</td>
<td>0.15</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.10</td>
<td>0.09</td>
</tr>
<tr>
<td>5</td>
<td>0.26</td>
<td>0.02</td>
<td>0.24</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.04</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>8</td>
<td>0.20</td>
<td>0.08</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 6.5

<table>
<thead>
<tr>
<th>Problem</th>
<th>Absolute Error</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>Merson</td>
<td>RKF(4,5)</td>
</tr>
<tr>
<td>1</td>
<td>0.2258×10⁻³</td>
<td>0.2959×10⁻³</td>
<td>0.1061×10⁻⁴</td>
</tr>
<tr>
<td>2</td>
<td>0.3927×10⁻³</td>
<td>0.2093×10⁻³</td>
<td>0.1496×10⁻⁴</td>
</tr>
<tr>
<td>3</td>
<td>0.1389×10⁻⁴</td>
<td>0.8313×10⁻⁶</td>
<td>0.3170×10⁻⁴</td>
</tr>
<tr>
<td>4</td>
<td>0.8308×10⁻⁶</td>
<td>0.1944×10⁻⁵</td>
<td>0.2984×10⁻⁵</td>
</tr>
<tr>
<td>5</td>
<td>0.3253×10⁻⁶</td>
<td>0.3253×10⁻⁶</td>
<td>0.1175×10⁻⁴</td>
</tr>
<tr>
<td>6</td>
<td>0.2949×10⁻⁶</td>
<td>0.6047×10⁻⁶</td>
<td>0.2427×10⁻⁵</td>
</tr>
<tr>
<td>7</td>
<td>0.5516×10⁻⁴</td>
<td>0.1446×10⁻⁴</td>
<td>0.7625×10⁻⁵</td>
</tr>
<tr>
<td>8</td>
<td>0.1759×10⁻⁵</td>
<td>0.2955×10⁻⁶</td>
<td>0.2577×10⁻⁵</td>
</tr>
</tbody>
</table>

Table 6.6

From Table 6.5, we can see that the solution for problems 1,2,3,4,5,6,7,8 by RKF(4,5) performed faster than the solution by Merson and RK(4,4). But in Table 6.6, the accuracy of the RK(4,4) and Merson method is more accurate compared to RKF(4,5). However, by reducing the step size below a certain value, i.e \( \frac{h}{2} \) and \( \frac{h}{4} \), the solution by the RK(4,4), Merson and RKF(4,5) methods are...
comparable in terms of the time taken and the accuracy as shown in Table 6.7 and Table 6.8.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>RKF(4,5)</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
<td>1.11</td>
</tr>
<tr>
<td>2</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>0.09</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.12</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>0.11</td>
<td>0.04</td>
</tr>
<tr>
<td>7</td>
<td>0.10</td>
<td>0.09</td>
</tr>
<tr>
<td>8</td>
<td>0.09</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6.7

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>Merson</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>0.09</td>
<td>0.10</td>
</tr>
<tr>
<td>4</td>
<td>0.09</td>
<td>0.09</td>
</tr>
<tr>
<td>5</td>
<td>0.12</td>
<td>0.11</td>
</tr>
<tr>
<td>6</td>
<td>0.11</td>
<td>0.10</td>
</tr>
<tr>
<td>7</td>
<td>0.10</td>
<td>0.09</td>
</tr>
<tr>
<td>8</td>
<td>0.09</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Table 6.8

The computer program gives a facility to test whether the estimated truncation error ERREST given by equation (6.3.7) exceeds a certain pre-set tolerance ε. If this happens the routines halves the stepsize, recomputes ERREST and tests again. If ERREST is smaller than \( \left( \frac{1}{2} \right) ε \) or \( 2^{-1} ε \), then we double the stepsize. We use \( 2^{-4} \) instead of \( 2^{-3} \), because it is an \( O(h^4) \) method.

197
Thus in the RK(4,4) method with error control, we can delete 'h' because it is a $O(h^4)$ method using the fourth order classical arithmetic mean (AM) method and contraharmonic mean ($C_pM$) method with ERREST. However if necessary we can make the test more stringent by increasing the ERREST in equation (6.3.7) from fourth order to fifth order by multiplying the equation (6.3.7) with a constant 'h' ($h<1$). The equation (6.3.7) is now fifth order in the form

$$ERREST = |Y_{AM} - Y_{C_pM}| \times h \times \frac{281}{4608}$$  \hspace{1cm} (6.3.8)

However by using ERREST in equation (6.3.8) and maintaining the error tolerance ERRTOL, i.e., ERRTOL=1.0×10^{-4} we can reduce the time taken by 36% and still maintain the higher accuracy. The comparison of times taken and accuracy achieved using ERREST in equation (6.3.8) between solutions from RK(4,4), Merson and RKF(4,5) methods are shown in Table 6.9 and Table 6.10.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>RKF(4,5)</td>
</tr>
<tr>
<td>1</td>
<td>1.15</td>
<td>1.11</td>
</tr>
<tr>
<td>2</td>
<td>0.21</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>0.21</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.16</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.04</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>8</td>
<td>0.21</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6.9
By using the error estimate in equation (6.3.8), the comparison of the time taken and the accuracy between solutions from RK(4,4), Merson and RKF(4,5) methods are shown in Table 6.11 and Table 6.12.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>Merson</td>
</tr>
<tr>
<td>1</td>
<td>1.15</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.21</td>
<td>0.21</td>
</tr>
<tr>
<td>3</td>
<td>0.21</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.09</td>
</tr>
<tr>
<td>5</td>
<td>0.16</td>
<td>0.24</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.10</td>
</tr>
<tr>
<td>8</td>
<td>0.20</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 6.10

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
<td>Merson</td>
</tr>
<tr>
<td>1</td>
<td>1.15</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>3</td>
<td>0.20</td>
<td>0.15</td>
</tr>
<tr>
<td>4</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.16</td>
<td>0.05</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.04</td>
</tr>
<tr>
<td>7</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>8</td>
<td>0.20</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 6.11
6.3.5 Automatic Selection of the Initial Step size for an ODE Solver

In most cases in our experience, users of ODE solvers do not wish to specify an initial stepsize and are often unable to provide a suitable value. In our approach, a new strategy to produce an effective scheme for the automatic choice of the initial stepsize is given. There have been a number of efforts to devise algorithms for the automatic selection of the initial step size (Gladwell et.al. [1985]). In our approach, we divide the process into two phases to provide the initial stepsize. In the first phase, the first RK(4, 4) method with error control is used to obtain a value of the stepsize. This value of stepsize is used as an initial stepsize to solve the problem by using the second formula RK(4, 4) with error control.

With this strategy, all the problems (1-8) can determine the initial stepsize. The value of the initial stepsize for certain problem are shown in Table 6.13.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Initial Stepsize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>h = 0.25</td>
</tr>
<tr>
<td>2</td>
<td>h = 0.0625</td>
</tr>
<tr>
<td>3</td>
<td>h = 0.125</td>
</tr>
<tr>
<td>4</td>
<td>h = 0.0625</td>
</tr>
<tr>
<td>5</td>
<td>h = 0.125</td>
</tr>
<tr>
<td>6</td>
<td>h = 0.0625</td>
</tr>
<tr>
<td>7</td>
<td>h = 0.25</td>
</tr>
<tr>
<td>8</td>
<td>h = 0.125</td>
</tr>
</tbody>
</table>

Table 6.13

The following are the numerical results of testing RK(4, 4) for the automatic selection of the initial stepsize with error control on the sample problem.
**Problem 1 :** \( Y' + Y = 0 \)

Initial conditions : \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution : \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>( h )</th>
<th>Exact Solution</th>
<th>Numerical Solution</th>
<th>Absolute Error</th>
<th>Estimate Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.25000</td>
<td>.7788086D+00</td>
<td>.7788008D+00</td>
<td>.7810679D-05</td>
<td>.4200556D-06</td>
</tr>
<tr>
<td>.50000</td>
<td>.6065428D+00</td>
<td>.6065307D+00</td>
<td>.1216599D-04</td>
<td>.6542743D-06</td>
</tr>
<tr>
<td>.75000</td>
<td>.4723808D+00</td>
<td>.4723666D+00</td>
<td>.1421239D-04</td>
<td>.7643182D-06</td>
</tr>
<tr>
<td>1.00000</td>
<td>.3678942D+00</td>
<td>.3678794D+00</td>
<td>.1475824D-04</td>
<td>.7936627D-06</td>
</tr>
<tr>
<td>1.25000</td>
<td>.2865192D+00</td>
<td>.2865048D+00</td>
<td>.1436723D-04</td>
<td>.7726255D-06</td>
</tr>
<tr>
<td>1.50000</td>
<td>.2231436D+00</td>
<td>.2231302D+00</td>
<td>.1342712D-04</td>
<td>.7220601D-06</td>
</tr>
<tr>
<td>1.75000</td>
<td>.1737861D+00</td>
<td>.1737739D+00</td>
<td>.1219995D-04</td>
<td>.6560594D-06</td>
</tr>
<tr>
<td>.50000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.25000</td>
<td>.1054484D+00</td>
<td>.1053992D+00</td>
<td>.4913849D-04</td>
<td>.6365842D-05</td>
</tr>
<tr>
<td>2.75000</td>
<td>.6398299D-01</td>
<td>.6392786D-01</td>
<td>.5512992D-04</td>
<td>.7236275D-05</td>
</tr>
<tr>
<td>3.25000</td>
<td>.3882301D-01</td>
<td>.3877421D-01</td>
<td>.4880501D-04</td>
<td>.6434257D-05</td>
</tr>
<tr>
<td>3.75000</td>
<td>.2355667D-01</td>
<td>.2351775D-01</td>
<td>.3892600D-04</td>
<td>.5141905D-05</td>
</tr>
<tr>
<td>4.25000</td>
<td>.1429350D-01</td>
<td>.1426423D-01</td>
<td>.2926750D-04</td>
<td>.3869708D-05</td>
</tr>
<tr>
<td>4.75000</td>
<td>.8672880D-02</td>
<td>.8651695D-02</td>
<td>.2118456D-04</td>
<td>.2802165D-05</td>
</tr>
<tr>
<td>5.25000</td>
<td>.5262450D-02</td>
<td>.5247518D-02</td>
<td>.1493208D-04</td>
<td>.1975352D-05</td>
</tr>
<tr>
<td>5.75000</td>
<td>.3193101D-02</td>
<td>.3182781D-02</td>
<td>.1032067D-04</td>
<td>.1365208D-05</td>
</tr>
<tr>
<td>6.25000</td>
<td>.1937481D-02</td>
<td>.1930454D-02</td>
<td>.6858390D-05</td>
<td>.9292943D-06</td>
</tr>
<tr>
<td>6.75000</td>
<td>.1175607D-02</td>
<td>.1170880D-02</td>
<td>.4727241D-05</td>
<td>.6250015D-06</td>
</tr>
<tr>
<td>.50000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.25000</td>
<td>.4408526D-03</td>
<td>.4307425D-03</td>
<td>.1011003D-04</td>
<td>.4768757D-05</td>
</tr>
<tr>
<td>7.75000</td>
<td>.1653197D-03</td>
<td>.1584613D-03</td>
<td>.6858390D-05</td>
<td>.3138291D-05</td>
</tr>
<tr>
<td>8.25000</td>
<td>.6199489D-04</td>
<td>.5829466D-04</td>
<td>.3700229D-05</td>
<td>.1600700D-05</td>
</tr>
<tr>
<td>8.75000</td>
<td>.2324808D-04</td>
<td>.2144541D-04</td>
<td>.1802677D-05</td>
<td>.7333294D-06</td>
</tr>
</tbody>
</table>

Total Time = .54

Initial Stepsize For Problem No 1 is \( h = .25000 \)

The time taken can be reduced significantly and depends on the user which value of final point is needed. If the user needs the value at the final point at \( x = 10 \) without printing out the results, the time taken in Problem No 1 can be reduced from 0.54s to 0.12s. The numerical results of testing RK(4,4) using the automatic selection of initial stepsize with the final point needed, i.e., at \( x = 10 \) are shown below
Problem 1: \( Y' + Y = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)
Exact solution: \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>( h )</th>
<th>Exact Solution</th>
<th>Numerical Solution</th>
<th>Absolute Error</th>
<th>Estimate Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50000</td>
<td>0</td>
<td>0.2324808D-04</td>
<td>2.144541D-04</td>
<td>.1802677D-05</td>
</tr>
<tr>
<td>0.25000</td>
<td>0.1802677D-05</td>
<td>.7333294D-06</td>
<td>.1802677D-05</td>
<td>.7333294D-06</td>
</tr>
<tr>
<td>0.50000</td>
<td>0.2324808D-04</td>
<td>0.2144541D-04</td>
<td>.7333294D-06</td>
<td>.7333294D-06</td>
</tr>
<tr>
<td>1.00000</td>
<td>0.1802677D-05</td>
<td>.7333294D-06</td>
<td>.7333294D-06</td>
<td>.7333294D-06</td>
</tr>
</tbody>
</table>

Total Time = .12

Initial Stepsize For Problem No 1 is \( h = 0.25000 \)

6.4 THE EXPLICIT RUNGE-KUTTA METHOD

6.4.1 2-Stage Second Order Method

For second order accuracy, the four coefficient \( a_{21}, b_1, b_2 \) and \( c_2 \) can be shown in Butcher array form as

\[
\begin{array}{c|cc}
0 & c_2 & a_{21} \\
\hline
b_1 & b_2 \\
\end{array}
\]

or

\[
\begin{array}{c|cc}
0 & 1 & 1 \\
\hline
1 & 1 & 1 \\
\hline
1/2 & 1/2 \\
\end{array}
\]

with \( c_i = a_i = 1 \) is of order 2 iff

\[
\begin{aligned}
b_1 + b_2 &= 1 , \\
b_2 c_2 &= \frac{1}{2}
\end{aligned}
\]

6.4.2 3-Stage of Third Order Method

Three stages of order 3 shows a method in the form
\[
\begin{array}{c|ccc}
0 & c_2 & a_{21} \\
c_2 & & & a_{31} \\
c_3 & a_{31} & a_{32} \\
\hline
b_1 & b_2 & b_3 & b_4
\end{array}
\]

or

\[
\begin{array}{c|ccc}
0 & 2/3 & 2/3 \\
2/3 & 1/3 & 1/3 \\
1/4 & 0 & 3/4 \\
\hline

\end{array}
\]

with \( c_2 = a_{21} = \frac{2}{3} \), \( c_3 = a_{31} + a_{32} = \frac{2}{3} \) is of order 3 iff

\[
\begin{align*}
& b_1 + b_2 + b_3 = 1, \\
& b_2 c_2 + b_3 c_3 = \frac{1}{2}, \\
& b_2 c_2^2 + b_3 c_3^2 = \frac{1}{3}, \\
& b_2 a_{21} c_2 = \frac{1}{6}
\end{align*}
\]

6.4.3 A 4 - Stage Fourth - Order Method

For a 4-stage fourth order method, we can write the coefficient \( c_2, c_3, c_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}, b_1, b_2, b_3, b_4 \) and put in Butcher array form as

\[
\begin{array}{c|ccc}
0 & c_2 & a_{21} \\
c_2 & & & a_{31} \\
c_3 & a_{31} & a_{32} \\
c_4 & a_{41} & a_{42} & a_{43} \\
\hline
b_1 & b_2 & b_3 & b_4
\end{array}
\]

203
and from equation (6.1.4) with \( c_2 = a_{21} = \frac{1}{2} \) we obtain the following eight equations:

\[
\begin{align*}
&b_1 + b_2 + b_3 + b_4 = 1, \\
&b_2 c_1 + b_3 c_2 + b_4 c_3 = \frac{1}{3}, \\
&b_2 c_1^2 + b_3 c_2^2 + b_4 c_3^2 = \frac{1}{2}, \\
&b_2 a_{22} c_2 + b_3 a_{32} c_2 + b_4 a_{42} c_3 = \frac{1}{6}, \\
&b_2 c_2^3 + b_3 c_3^3 + b_4 c_4^3 = \frac{1}{4}, \\
&b_2 c_1 a_{22} c_2 + b_3 c_1 a_{32} c_2 + b_4 c_1 a_{42} c_3 = \frac{1}{8}, \\
&b_2 a_{22} c_2^2 + b_3 a_{32} c_2^2 + b_4 a_{42} c_3^2 = \frac{1}{12}, \\
&b_2 a_{22} a_{33} c_3 = \frac{1}{24}.
\end{align*}
\]

### 6.4.4 5-stage Fifth-order Method

For 5-stages or fifth order method, we can write the coefficients

\[ c_2, c_3, c_4, c_5, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}, a_{51}, a_{52}, a_{53}, a_{54}, b_1, b_2, b_3, b_4, b_5 \]

and put them in Butcher array form as

\[
\begin{array}{c|cccc}
0 & c_2 & a_{21} \\
c_2 & c_3 & a_{31} & a_{32} \\
c_3 & c_4 & a_{41} & a_{42} & a_{43} \\
c_4 & c_5 & a_{51} & a_{52} & a_{53} & a_{54} \\
& b_1 & b_2 & b_3 & b_4 & b_5 \\
\end{array}
\]

with the following seventeen equations where:
\[ b_1 + b_2 + b_3 + b_4 + b_5 = 1 \]
\[ b_2c_2 + b_3c_3 + b_4c_4 + b_5c_5 = \frac{1}{3} \]
\[ b_2c_2^2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2 = \frac{1}{2} \]
\[ b_2a_0c_2 + b_4a_0c_4 + b_3a_0c_3 + b_3a_0c_2 + b_5a_0c_5 + b_5a_0c_4 = \frac{1}{6} \]
\[ b_2c_2^3 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3 = \frac{1}{4} \]
\[ b_2c_2a_2c_2 + b_4c_4a_4c_4 + b_3c_3a_2c_3 + b_3c_3a_2c_2 + b_5c_5a_2c_5 + b_5c_5a_2c_4 = \frac{1}{8} \]
\[ b_2a_0c_2 + b_4a_0c_4 + b_3a_0c_3 + b_3a_0c_2 + b_5a_0c_5 + b_5a_0c_4 = \frac{1}{12} \]
\[ b_4a_0a_0c_2 + b_3a_0a_0c_2 + b_3a_0a_0c_3 = \frac{1}{24} \]
\[ b_2c_2^4 + b_3c_3^4 + b_4c_4^4 + b_5c_5^4 = \frac{1}{5} \]
\[ b_2a_2c_2 + b_4a_2c_4 + b_3a_2c_3 + b_3a_2c_2 + b_5a_2c_5 + b_5a_2c_4 = \frac{1}{10} \]
\[ b_2a_0c_2 + b_4a_0c_4 + b_3a_0c_3 + b_3a_0c_2 + b_5a_0c_5 + b_5a_0c_4 = \frac{1}{15} \]
\[ b_4a_0a_0a_0c_2 + b_5a_0a_0a_0c_2 + b_5a_0a_0a_0c_3 + b_5a_0a_0a_0c_3 = \frac{1}{30} \]
\[ b_3(a_a c_1)^2 + b_4(a_a c_2 + a_a c_3)^2 + b_5(a_a c_3 + a_a c_4)^2 = \frac{1}{20} \]

\[ b_3 a_a c_1^2 + b_4 a_a c_2^2 + b_5 a_a c_3^2 + b_6 a_a c_4^2 = \frac{1}{20} \]

\[ b_3 a_a c_1 a_a c_2 + b_4 a_a c_2 a_a c_2 + b_5 a_a c_3 a_a c_2 = \frac{1}{40} \]

\[ b_3 a_a c_2^2 + b_4 a_a c_3^2 + b_5 a_a c_4^2 = \frac{1}{60} \]

\[ b_3 a_a c_2 = \frac{1}{120} \]

### 6.5 THE EXPLICIT CONTRAHRARMONIC MEAN (\(C_oM\)) METHOD

#### 6.5.1 2-Stage Second Order \((C_oM)\) Method

A 2-stage formula of second order method for contraharmonic mean \((C_oM)\) method in the form is

\[ y_{n+1} = y_n + h \left[ \frac{k_1 + k_2}{k_1 + k_2} \right] \]

where

\[ k_1 = f(x_n, y_n) \]

\[ k_2 = f(x_n + h, y_n + hk_1) \]

also can be written as a Butcher array as

\[
\begin{array}{c|cc}
0 & \ \\
\hline
c_2 & a_{21} & \\
\end{array}
\]

\[
\begin{array}{cc}
b_1 & b_2 \\
\end{array}
\]
and the four coefficients \( a_{21}, b_1, b_2 \) and \( c_2 \) can be formulated in two equations as

\[
\begin{align*}
  b_1 + b_2 &= \frac{1}{2}, \\
  b_c c_2 &= \frac{1}{3},
\end{align*}
\]

### 6.5.2 A 3 - Stage Third Order \((C_0M)\) Method

A 3-stage order 3 \((C_0M)\) method in the form

\[
y_{n+1} = y_n + h \left[ k_1^2 + k_2^2 + \frac{k_2^3 + k_3^3}{k_2 + k_3} \right]
\]

where

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f(x_n + \frac{2}{3} h, y_n + \frac{2}{3} h k_1)
\]

\[
k_3 = f(x_n + \frac{2}{3} h, y_n + \frac{2}{3} h k_2)
\]

and can be shown in the Butcher array form as

\[
\begin{array}{ccc}
  0 & & \\
  1 & 1 & \\
  0 & 1 & \\
  2 & 2 & 2 \\
  3 & 3 & 3 \\
  1 & 1 & 1
\end{array}
\]

with \( c_2 = a_{21} = \frac{2}{3}, \ c_j = a_{2j} + a_{3j} = \frac{2}{3} \) is of order 3 iff
The number of coefficients involved in the 4-stage fourth order \((C_0 M)\) method is 19 and less compared to the 4-stage fourth order classical Runge-Kutta method which requires 23. The coefficients \(c_2, c_3, c_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}, b_1, b_2, b_3\) in the Butcher array form are

\[
\begin{array}{c|ccc}
0 & c_2 & a_{21} \\
c_3 & a_{31} & a_{32} \\
c_4 & a_{41} & a_{42} & a_{43} \\
& b_1 & b_2 & b_3 \\
\end{array}
\]

or

\[
\begin{array}{c|ccc}
0 & 1 & 1 \\
\frac{1}{2} & 1 & 2 \\
\frac{1}{2} & 1 & 3 \\
\frac{1}{2} & 8 & 8 \\
1 & \frac{1}{4} & \frac{3}{4} & \frac{3}{2} \\
1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\end{array}
\]

with \(c_2 = a_{21} = \frac{1}{2}, c_3 = a_{31} + a_{32} = \frac{1}{2}, c_4 = a_{41} + a_{42} + a_{43} = 1\) is of order 4 iff

\[
b_1 + b_2 + b_3 = 1
\]
\[ b_2c_2 + b_3c_3 = \frac{1}{3} \]
\[ b_2c_2^2 + b_3c_3^2 = \frac{1}{6} \]
\[ b_2a_2c_2 + b_3a_2c_2 + b_3a_3c_3 = \frac{7}{16} \]
\[ b_2c_2^3 + b_3c_3^3 = \frac{1}{6} \]
\[ b_2c_2a_2c_2 + b_3c_2a_2c_2 + b_3c_2a_3c_3 = \frac{5}{32} \]
\[ b_2a_2a_3c_2 + b_3a_2a_3c_2 + b_3a_2a_3c_3 = \frac{3}{32} \]
\[ b_3a_2a_3c_2 = \frac{3}{32} \]

6.6 THE EXPLICIT CENTROIDAL MEAN \((C_eM)\) METHOD

6.6.1 A 2-Stage Second Order \((C_eM)\) Method

For a second order centroidal mean \((C_eM)\) method in the form

\[ y_{n+1} = y_n + h \left[ \frac{2(k_1 + k_2 + k_3)}{3(k_1 + k_2)} \right] \]

where

\[ k_1 = f(x_n, y_n) \]
\[ k_2 = f(x_n + h, y_n + hk_1) \]

and the four coefficient \(a_2, b_1, b_2\) and \(c_2\) can be shown in Butcher array form as

\[
\begin{array}{c|ccc}
0 & & & \\
c_2 & a_{21} & & \\
& b_1 & b_2 & \\
\end{array}
\]

or

209
with \( c_2 = a_2 = 1 \) is of order 2 iff

\[
\begin{align*}
b_1 + b_2 &= 1, \\
b_2 c_2 &= \frac{1}{2},
\end{align*}
\]

6.6.2 A 3 - Stage Third Order \((C_eM)\) Method

Three stages for an order 3 method can be shown in the form

\[
y_{n+1} = y_n + \frac{h}{2} \left[ 2(k_1^2 + k_2 k_3 + k_4^2) + 2(k_2^2 + k_4 k_5 + k_7^2) \right]
\]

where

\[
k_1 = f(x_n, y_n)
\]

\[
k_2 = f\left(x_n + \frac{2}{3} h, y_n + \frac{2}{3} h k_1\right)
\]

\[
k_3 = f\left(x_n + \frac{2}{3} h, y_n - \frac{2}{9} h k_1 + \frac{8}{9} h k_2\right)
\]

and in Butcher array form as

\[
\begin{array}{l|lll}
0 & 0 & 0 & 0 \\
\frac{2}{3} & \frac{2}{3} & 0 & 0 \\
\frac{2}{3} & -\frac{2}{9} & \frac{8}{9} & 0 \\
\frac{1}{2} & \frac{1}{2} & 0 & 0
\end{array}
\]

with \( c_2 = a_{21} = \frac{2}{3} \), \( c_3 = a_{32} + a_{33} = \frac{2}{3} \) of order 3 with the conditions
\[ b_1 + b_2 = 1 \]
\[ b_2 c_2 = \frac{1}{3} \]
\[ b_2 c_2^2 = \frac{2}{9} \]
\[ b_2 a_2 c_2 = \frac{8}{27} \]

### 6.6.3 A 4 - Stage Fourth Order \((C_4M)\) Method

For a 4-stage fourth order method, we can write the coefficients \(c_2, c_3, c_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}, b_1, b_2, b_3\) and when put in the Butcher array form we have

\[
\begin{array}{cccc}
0 & a_{21} & a_{31} & a_{41} \\
\hline
c_2 & b_1 & b_2 & b_3 \\
\end{array}
\]

or

\[
\begin{array}{ccccccc}
0 & & & & & & \\
\hline
\frac{1}{2} & 1 & 1 & 11 & 25 & 73 & \\
\frac{1}{2} & & 24 & 24 & & & \\
1 & & 132 & & 132 & 66 & \\
\hline
\end{array}
\]
with \( c_2 = a_{m} = \frac{1}{2} \), \( c_3 = a_{m} + a_{m} = \frac{1}{2} \), \( c_4 = a_{m} + a_{m} + a_{m} = 1 \)

is of order 4 with the conditions

\[
\begin{align*}
    b_1 + b_2 + b_3 &= 1, \\
    b_2c_2 + b_3c_3 &= \frac{1}{3}, \\
    b_2c_2^2 + b_3c_3^2 &= \frac{1}{6}, \\
    b_2a_2c_2 + b_3c_2c_2 + b_4c_2c_3 &= \frac{11}{48}, \\
    b_2c_2^3 + b_3c_3^3 &= \frac{1}{12}, \\
    b_2c_2a_2c_2 + b_3c_2a_2c_2 + b_4c_2a_3c_3 &= \frac{55}{288}, \\
    b_3a_3c_2^2 + b_4a_3c_2^2 + b_5a_3c_3^2 &= \frac{11}{288}, \\
    b_3a_3c_2 &= \frac{73}{864}
\end{align*}
\]

which is the same number of coefficients as \((C_6M)\).

6.7 THE EXPLICIT ROOT-MEAN-SQUARE (RMS) METHOD

6.7.1 A 2-Stage Second Order (RMS) Method

A formula with 2-stages of second order for the root-mean-square (RMS) method has the form

\[
y_{n+1} = y_n + h \left[ \sqrt{\left( \frac{k_1^2 + k_2^2}{2} \right)} \right]
\]

where \( k_1 = f(x_n, y_n) \)

\[
k_2 = f(x_n + h, y_n + hk_1)
\]

It can also be written in the Butcher array form as
with \( c_2 = a_2 = 1 \) and the four coefficient \( a_{21}, b_1, b_2 \) and \( c_2 \) can be formulated in two equations of conditions as

\[
\begin{align*}
b_1 + b_2 &= 1 , \\
b_2 c_2 &= \frac{1}{2} ,
\end{align*}
\]

6.7.2 A 3 - Stage Third Order (RMS) Method

An order 3 method with 3-stages for the root-mean-square (RMS) method has the form

\[
y_{n+1} = y_n + \frac{h}{2} \left[ \frac{k_2^2 + k_1^2}{2} + \sqrt{\left( \frac{k_2^2 + k_3^2}{2} \right)} \right]
\]

where \( k_i = f(x_i, y_i) \)

\[
\begin{align*}
k_2 &= f\left( x_n + \frac{2}{3} h, y_n + \frac{2}{3} h k_1 \right) \\
k_3 &= f\left( x_n + \frac{2}{3} h, y_n - \frac{1}{6} h k_1 + \frac{5}{6} h k_2 \right)
\end{align*}
\]

and can be shown in Butcher array form as

\[
\begin{array}{c|ccc}
0 & a_{21} \\
\hline
0 & b_1 & b_2 \\
\hline
\frac{1}{2} & 1 & \frac{1}{2}
\end{array}
\]
with \( c_2 = a_{21} = \frac{2}{3}, \ c_3 = a_{31} + a_{32} = \frac{2}{3} \) is of order 3 with four equations of conditions, i.e.,

\[
\begin{align*}
    b_1 + b_2 &= 1, \\
    b_2c_2 &= \frac{1}{3}, \\
    b_2c_2^2 &= \frac{2}{9}, \\
    b_3a_3c_3 &= \frac{5}{18},
\end{align*}
\]

6.7.3 A 4-Stage Fourth Order (RMS) Method

The numbers of coefficient involved in the 4-stage fourth order (RMS) method are similar with the 4-stage fourth order contraharmonic mean (\(C_oM\)) method. The coefficient \( c_2, c_3, c_4, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{43}, b_1, b_2, b_3 \) in the Butcher array form are

\[
\begin{array}{c|ccc}
0 & c_2 & c_3 & c_4 \\
    & a_{21} & a_{31} & a_{41} \\
    & & a_{32} & a_{42} \\
    & & & a_{43} \\
    b_1 & b_2 & b_3
\end{array}
\]

or

\[
\begin{array}{c|ccc}
0 & \frac{1}{2} & 1 & 1 \\
    & \frac{1}{2} & \frac{7}{16} & \frac{33}{28} \\
\frac{1}{2} & \frac{1}{16} & \frac{17}{56} & \frac{1}{3} \\
\frac{1}{2} & \frac{17}{56} & \frac{1}{3} & \frac{1}{3} \\
1 & \frac{33}{28} & \frac{1}{3} & \frac{1}{3}
\end{array}
\]
with \( c_2 = a_{\pi} = \frac{1}{2} \), \( c_3 = a_{\pi} + a_{2\pi} = \frac{1}{2} \), \( c_4 = a_{4\pi} + a_{5\pi} = 1 \) is of order 4 with eight equations of conditions, i.e.,

\[
\begin{align*}
    b_1 + b_2 + b_3 &= 1 \\
    b_2 c_1 + b_3 c_3 &= \frac{1}{3} \\
    b_2 c_1^2 + b_3 c_3^2 &= \frac{1}{6} \\
    b_2 a_{\pi} c_2 + b_3 a_{\pi} c_3 + b_3 a_{\pi} c_3 &= \frac{5}{32} \\
    b_2 c_1^3 + b_3 c_3^3 &= \frac{1}{12} \\
    b_2 c_1 a_{\pi} c_2 + b_3 c_4 a_{\pi} c_3 + b_3 c_4 a_{\pi} c_3 &= \frac{7}{32} \\
    b_3 c_1^2 a_{\pi} c_2 + b_3 a_{\pi} c_3^2 + b_3 a_{\pi} c_3^2 &= \frac{49}{192} \\
    b_3 a_{\pi} a_{\pi} c_2 &= \frac{11}{128}
\end{align*}
\]

From the above equations of conditions, we have shown that they have same number of terms as \((C_0M)\).

From the above discussion, we conclude with the following hypothesis:

**Hypothesis 6.7.4**

If an \( s \)-stage explicit Runge-Kutta using non-linear method, i.e., \( C_s M \), \( C_s M \) and RMS methods has order \( N \) where \( N \leq 3 \) then \( s \geq N \).
NUMERICAL SOLUTION OF ODEs BY NONLINEAR MULTISTEP METHODS
We are given a $p^{th}$ order initial value problem in the form
\[ y^{(p)} = f(x, y, y^{(1)}, \ldots, y^{(p-1)}) \]  
with initial conditions
\[ x_0 = \alpha_0, y(x_0) = \gamma_0, y^{(0)}(x_0) = \gamma_0^{(0)}, \ldots, y^{(p-1)}(x_0) = \gamma_0^{(p-1)}. \]

Evans and Jayes [1993], introduced an approximation of $y(x_{n+1})$ as a geometric mean (GM) combination of the values $y_1, y_2, \ldots, y_{n+1}$ and the derivatives computed at $y_{n+1}, y_n, \ldots, y_{n-k+1}$. Thus,
\[ y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \cdots + \alpha_k y_{n-k+1} + h \left\{ \sum_{i,j=0}^{k-1} \beta_{i,j} \left[ f(y_{n-1}) f(y_{n-j}) \right] \right\} \]  
for some fixed numbers
\[ \alpha_1, \alpha_2, \ldots, \alpha_k, \beta_{00}, \beta_{01}, \beta_{02}, \ldots, \beta_{k-1} \] and \[ \beta_{11}, \ldots, \beta_{k-1,k-1}. \]

We now study equation (7.0.2) with the $C_0M$ method in the form
\[ y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \cdots + \alpha_k y_{n-k+1} + h \left\{ \sum_{i,j=0}^{k-1} \beta_{i,j} \left[ f^2(y_{n-1}) + f^2(y_{n-j}) \right] \right\} \]  
(7.0.3)

### 7.1 NUMERICAL METHODS FOR FIRST ORDER ODES

Consider equation (7.0.1) for the case of $p=1$, then the general form of the formula which approximates $y_{n+1}$ at $x_n$ is defined by
\[ y_{n+1} - y_n = h \left\{ \alpha_1 f_n + \alpha_2 f_{n+1} + \alpha_3 \left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right) \right\} \]  
(7.1.1)

The values of $\alpha_1, \alpha_2$ and $\alpha_3$ are to be determined to give the highest order of accuracy. The Taylor series of $y_{n+1}$ at $x_n$ give
\[ y_{n+1} - y_n = hf_n + \frac{h^2}{2!} f_n^{(2)} + \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} + O(h^5) \]  
(7.1.2)
and the Taylor series expansion of $f_{n+1}$ about $x_n$, we have

$$f_{n+1} = f_n + h f_n^{(1)} + \frac{h^2}{2!} f_n^{(2)} + \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} + O(h^5)$$  \hspace{1cm} (7.1.3)

By substituting (7.1.3) into (7.1.1), we obtain the right-hand side of (7.1.1) as

$$(\alpha_1 + \alpha_2 + \alpha_3) f_n + \frac{1}{4}(2\alpha_2 + \alpha_3) h^2 f_n^{(2)} + h^3 \left( \frac{2\alpha_2 + \alpha_3}{4} \right) f_n^{(3)} + h^4 \left( \frac{\alpha_3}{4} \right) \frac{(f_n^{(3)})^2}{f_n} + O(h^5).$$  \hspace{1cm} (7.1.4)

By equating the coefficients terms in (7.1.2) and (7.1.4), the following equations of conditions are obtained:

- $h f_n : -1 + \alpha_1 + \alpha_2 + \alpha_3 = 0$  \hspace{1cm} (7.1.5-i)
- $h^2 f_n^{(1)} : -1 + 2\alpha_2 + \alpha_3 = 0$  \hspace{1cm} (7.1.5-ii)
- $h^3 f_n^{(2)} : -2 + 6\alpha_2 + 3\alpha_3 = 0$  \hspace{1cm} (7.1.5-iii)

By solving (7.1.5-i) and (7.1.5-ii) using Mathematica, we have

$$\alpha_1 = \frac{1 - \alpha_3}{2} \quad \text{and} \quad \alpha_2 = \frac{1 - \alpha_3}{2}$$  \hspace{1cm} (7.1.6)

or by solving equation (7.1.5-i) and (7.1.5-iii) by Mathematica, we obtain

$$\alpha_1 = \frac{4 - 2\alpha_3}{6} \quad \text{and} \quad \alpha_2 = \frac{2 - 3\alpha_3}{6}$$  \hspace{1cm} (7.1.7)

By substituting $\alpha_3 = \beta$ approximately in (7.1.6) and (7.1.7), we have

$$\alpha_1 = \frac{1 - \beta}{2} \quad \text{and} \quad \alpha_2 = \frac{1 - \beta}{2}$$  \hspace{1cm} (7.1.8)

and

$$\alpha_1 = \frac{4 - 2\beta}{6} \quad \text{and} \quad \alpha_2 = \frac{2 - 3\beta}{6}$$  \hspace{1cm} (7.1.9)
When we substitute equation (7.1.8) into (7.1.1), we obtain

$$y_{n+1} - y_n = h \left\{ \left( \frac{1-\beta}{2} \right) f_n + \left( \frac{1-\beta}{2} \right) f_{n+1} + \beta \left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right) \right\}$$

(7.1.10)

where $\beta$ is the parameter defined as $\alpha_2$.

By substituting $\beta = 0$, the well known Trapezoidal formula (AM) can be obtained as

$$y_{n+1} - y_n = \frac{h}{2} \left[ f_n + f_{n+1} \right]$$

(7.1.11)

Similarly, the original $CJM$ formula can be derived from (7.1.10) by putting $\beta = 1$ to obtain

$$y_{n+1} - y_n = h \left[ \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right]$$

(7.1.12)

The formula given by (7.1.11) and (7.1.12) is accurate up to order two and the LTE of (7.1.11) with $\beta = 1, \alpha_1 = \alpha_2 = \frac{1}{2}$ is obtained as

$$LTE_{AM} = \left( \frac{\alpha_2}{2} - \frac{1}{6} \right) h^3 f_n^{(3)}$$

$$= \frac{h^3}{12} (6\alpha_2 - 2) f_n^{(3)}$$

$$= \frac{h^3}{12} f_n^{(3)}$$

(7.1.13)

By substituting $\alpha_1 = \alpha_2 = \beta$, then $\alpha_3 = 1 - 2\beta$ into equation (7.1.1), we have

$$y_{n+1} - y_n = h \left\{ \beta (f_n + f_{n+1}) + (1-2\beta) \left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right) \right\}$$

(7.1.14)

By substituting $\beta = \frac{1}{4}$, the well known Trapezoidal formula (AM) can be obtained as in equation (7.1.11) and by replacing $\beta = 0$ in (7.1.14) we obtain $CJM$ in the form equation (7.1.12).

The LTE with $\beta = \frac{1}{4}$, $\alpha_2 = 0$ for equation (7.1.14) is
\[ LTE^{CM} = \frac{-h^3}{6} f_n^{(2)} + \frac{h^3}{4} f_n^{(2)} \]

\[ = \frac{h^3}{12} f_n^{(2)} \quad (7.1.15) \]

and LTE with \( \beta = 0 \) and \( \alpha, = 1 \) in equation \((7.1.14)\) is given by

\[ LTE^{CM} = (-\frac{1}{6} + \frac{1}{4})h^3 f_n^{(2)} + \frac{\alpha_5 (f_n^{(0)})^2}{f_n} \]

\[ = \frac{h^3}{12} \left( f_n^{(2)} + 3 \frac{(f_n^{(0)})^2}{f_n} \right) \quad (7.1.16) \]

### 7.2 DERIVATION OF THE CM METHOD FOR PROBLEMS OF THE TYPE

\( \gamma^{(0)} = f(x, y) \)

Evans and Jayes [1993] have shown that for the case of \( p=2 \) the CM method developed in the form

\[ y_{n+1} - 2y_n + y_{n-1} = h^2 \left\{ \alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4 \sqrt{f_{n} f_{n-1}} + \alpha_5 \sqrt{f_{n} f_{n+1}} + \alpha_6 \sqrt{f_{n-1} f_{n+1}} \right\} \quad (7.2.1) \]

A fourth order accuracy is obtained through the standard procedure of adjustment of the parameters \( \alpha_i, 1 \leq i \leq 6 \) for formula \((7.2.1)\) where

\[ \alpha_1 = \frac{1}{6} (5 + 12 \alpha) \]

\[ \alpha_2 = \alpha_3 = \frac{1}{12} \left(1 + 6 \alpha \right) \quad (7.2.2) \]

\[ \alpha_4 = \alpha_5 = \frac{1}{2} \alpha \]

\[ \alpha_6 = \alpha \]

The CM formula is obtained by substituting \( \alpha = -\frac{1}{6} \) into \((7.2.2)\) and from \((7.2.1)\) we obtain

\[ y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{2} \left\{ 3f_n + 2 \left[ \sqrt{f_n f_{n+1}} + \sqrt{f_n f_{n+1}} \right] - \sqrt{f_{n-1} f_{n+1}} \right\} \quad (7.2.3) \]

and another form of the CM formula was obtained by substituting \( \alpha = -\frac{5}{12} \) in \((7.2.2)\) and from \((7.2.10)\) to give
\[ y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{2} \left\{ 10 \left[ 2 \left( \sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}} \right) - \sqrt{f_{n-1} f_{n+1}} \right] - 3(f_{n-1} + f_{n+1}) \right\} \]  
(7.2.4)

Now, we consider equation (7.0.1) for the case of \( p=2 \) and the general form of the formula which approximates \( y_{n+1} \) at \( x_n \) is defined by

\[ y_{n+1} - 2y_n + y_{n-1} = h^2 \left[ \alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4 \left[ \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right] + \alpha_5 \left[ \frac{f_{n-1}^2 + f_{n+1}^2}{f_{n-1} + f_{n+1}} \right] \right] \]  
(7.2.5)

Consider the left hand side of equation (7.2.5). The Taylor series of \( y_{n+1} \) and \( y_n \) at \( x_n \) give

\[ y_{n+1} - y_n = h y_n^{(1)} + \frac{h^2}{2!} y_n^{(2)} + \frac{h^3}{3!} y_n^{(3)} + \frac{h^4}{4!} y_n^{(4)} + \frac{h^5}{5!} y_n^{(5)} + \frac{h^6}{6!} y_n^{(6)} + O(h^7) \]  
(7.2.6)

\[ y_{n-1} - y_n = -h y_n^{(1)} + \frac{h^2}{2!} y_n^{(2)} - \frac{h^3}{3!} y_n^{(3)} + \frac{h^4}{4!} y_n^{(4)} - \frac{h^5}{5!} y_n^{(5)} + \frac{h^6}{6!} y_n^{(6)} + O(h^7) \]  
(7.2.7)

From equations (7.2.6) and (7.2.7), we have

\[ y_{n+1} + y_{n-1} - 2y_n = h^2 f_n + \frac{h^4}{12} f_n^{(2)} + \frac{h^6}{360} f_n^{(4)} + O(h^7) \]  
(7.2.8)

since \( y_n^{(2)} = f_n \).

Now consider the right hand side of equation (7.2.5). By the Taylor series expansion of \( f_{n+1} \) and \( f_{n-1} \) at \( x_n \), we have

\[ f_{n+1} = f_n + h f_n^{(1)} + \frac{h^2}{2!} f_n^{(2)} + \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} + \frac{h^5}{5!} f_n^{(5)} + O(h^6) \]  
(7.2.9)

and

\[ f_{n-1} = f_n - h f_n^{(1)} + \frac{h^2}{2!} f_n^{(2)} - \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} - \frac{h^5}{5!} f_n^{(5)} + O(h^6) \]  
(7.2.10)
By substituting equations (7.2.9) and (7.2.10) into the right hand side of equation (7.2.5) and using the Mathematica program for algebraic manipulation, we obtain the following results:

\[
\alpha_2 f_{n-1} = \alpha_2 f_n \left[ 1 - \frac{hf_n^{(0)}}{f_n} + \frac{h^2 f_n^{(2)}}{2! f_n} - \frac{h^3 f_n^{(3)}}{3! f_n} + \frac{h^4 f_n^{(4)}}{4! f_n} - \frac{h^5 f_n^{(5)}}{5! f_n} \right] + O(h^6) \quad (7.2.11)
\]

\[
\alpha_3 f_{n+1} = \alpha_3 f_n \left[ 1 + \frac{hf_n^{(0)}}{f_n} + \frac{h^2 f_n^{(2)}}{2! f_n} + \frac{h^3 f_n^{(3)}}{3! f_n} + \frac{h^4 f_n^{(4)}}{4! f_n} + \frac{h^5 f_n^{(5)}}{5! f_n} \right] + O(h^6) \quad (7.2.12)
\]

\[
\alpha_4 f_n \left[ \frac{f_n^2 + f_n^{(2)}}{f_n + f_n^{-1}} \right] = \alpha_4 f_n \left[ 1 - \frac{hf_n^{(0)}}{2f_n} + \frac{h^2}{4} \left( \frac{f_n^{(0)}}{f_n} \right)^2 + \frac{f_n^{(2)}}{f_n} \right] \\
+ \frac{h^3}{24} \left( 3 \left( \frac{f_n^{(0)}}{f_n} \right)^3 - 6 \frac{f_n^{(0)} f_n^{(2)}}{(f_n)^3} - 2 \frac{f_n f_n^{(0)} f_n^{(2)}}{(f_n)^3} \right) \\
+ \frac{h^4}{48} \left( 3 \left( \frac{f_n^{(0)}}{f_n} \right)^4 - 9 \frac{(f_n^{(0)})^2 f_n^{(2)}}{(f_n)^3} + 3 \frac{f_n(f_n^{(2)})^2}{(f_n)^3} + 4 \frac{f_n f_n^{(0)} f_n^{(2)}}{(f_n)^3} + \frac{(f_n)^2 f_n^{(4)}}{(f_n)^3} \right) \\
+ \frac{h^5}{480} \left( 15 \left( \frac{f_n^{(0)}}{f_n} \right)^5 - 60 \frac{(f_n^{(0)})^3 f_n^{(3)}}{(f_n)^4} + 45 \frac{f_n f_n^{(0)} (f_n^{(2)})^2}{(f_n)^4} + 30 \frac{f_n(f_n^{(0)})^2 f_n^{(3)}}{(f_n)^4} \right) \\
- 20 \frac{(f_n)^3 f_n^{(2)} f_n^{(3)}}{(f_n)^4} - 10 \frac{(f_n)^2 f_n^{(0)} f_n^{(4)}}{(f_n)^4} - 2 \frac{(f_n)^3 f_n^{(5)}}{(f_n)^4} \right] + O(h^6) \quad (7.2.13)
\]

\[
\alpha_5 f_n \left[ \frac{f_n^2 + f_n^{(2)}}{f_n + f_n^{-1}} \right] = \alpha_5 f_n \left[ 1 + \frac{hf_n^{(0)}}{2f_n} + \frac{h^2}{4} \left( \frac{f_n^{(0)}}{f_n} \right)^2 + \frac{f_n^{(2)}}{f_n} \right] \\
+ \frac{h^3}{24} \left( -3 \left( \frac{f_n^{(0)}}{f_n} \right)^3 + 6 \frac{f_n^{(0)} f_n^{(2)}}{(f_n)^2} + 2 \frac{f_n f_n^{(0)} f_n^{(2)}}{(f_n)^2} \right) \\
+ \frac{h^4}{48} \left( 3 \left( \frac{f_n^{(0)}}{f_n} \right)^4 - 9 \frac{(f_n^{(0)})^2 f_n^{(2)}}{(f_n)^3} + 3 \frac{f_n(f_n^{(2)})^2}{(f_n)^3} + 4 \frac{f_n f_n^{(0)} f_n^{(2)}}{(f_n)^3} + \frac{(f_n)^2 f_n^{(4)}}{(f_n)^3} \right)
\]
By substituting equations (7.2.11)-(7.2.15) into equation (7.3.5), we obtain

\[
\frac{h^2}{480} \left[ -15 \left( \frac{f_n^{(0)}}{f_n} \right)^3 + 60 \left( \frac{f_n^{(0)} f_n^{(2)}}{f_n} \right)^2 - 45 \frac{f_n f_n^{(0)} f_n^{(2)}}{f_n} - 30 \frac{f_n (f_n^{(0)})^2 f_n^{(3)}}{f_n} \\
+ 20 \frac{(f_n^{(2)})^2 f_n^{(3)}}{(f_n)^4} + 10 \frac{(f_n^{(2)})^2 f_n^{(3)}}{(f_n)^4} + 2 \left( \frac{f_n^{(3)}}{f_n} \right)^2 + 0(h^6) \right] (7.2.14)
\]

\[
\alpha_0 \left[ \frac{f_{n+1}^2}{f_{n+1}^2 + f_{n+1}^2} \right] = \alpha_0 f_n \left[ 1 + \frac{h^2}{2} \left( 2 \left( \frac{f_n^{(0)}}{f_n} \right)^2 + \frac{f_n^{(2)}}{f_n} \right) + \frac{h^4}{24} \left( -12 \frac{(f_n^{(0)})^2 f_n^{(2)}}{(f_n)^4} + 8 \frac{f_n^{(0)} f_n^{(3)}}{(f_n)^4} + \frac{f_n^{(4)}}{f_n} \right) \right] + 0(h^6) (7.2.15)
\]

Therefore, by equating equations (7.2.8) and (7.2.16), we obtain the following equations of condition, i.e.,

\[
h^2 f_n \left( \sum \alpha_i + (-2 \alpha_2 + 2 \alpha_3 - \alpha_4 + \alpha_5) \frac{h h_n^{(0)}}{2 f_n} + (2 \alpha_2 + 2 \alpha_3 + \alpha_4 + \alpha_5 + 2 \alpha_6) \frac{h^2 f_n^{(2)}}{4 f_n} \right. \\
+ (\alpha_4 + \alpha_5 + 4 \alpha_6) \frac{h^2}{4} \left[ \frac{f_n^{(0)}}{f_n} \right]^3 - \frac{1}{12} \frac{h^2 f_n^{(3)}}{f_n} + (-2 \alpha_2 + 2 \alpha_3 - \alpha_4 + \alpha_5) \frac{h^3 f_n^{(3)}}{12 f_n} \left. \\
+ (\alpha_4 - \alpha_5) \frac{h^3}{8} \left[ \frac{f_n^{(0)}}{f_n} \right]^3 + (-\alpha_4 + \alpha_5) \frac{h^3 f_n^{(0)} f_n^{(3)}}{4 [f_n]^2} \right] + 0(h^6) (7.2.16)
\]

Therefore, by equating equations (7.2.8) and (7.2.16), we obtain the following equations of condition, i.e.,

\[
h^2 f_n : -1 + \alpha_i + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = 0 \quad (7.2.17-i)
\]

\[
h^2 f_n^{(0)} : -2 \alpha_2 + 2 \alpha_3 - \alpha_4 + \alpha_5 = 0 \quad (7.2.17-ii)
\]

\[
h^4 f_n^{(2)} : -1 + 6 \alpha_2 + 3 \alpha_3 + 3 \alpha_4 + 3 \alpha_5 + 6 \alpha_6 = 0 \quad (7.2.17-iii)
\]

\[
\frac{h^4 \left[ f_n^{(0)} \right]^3}{f_n} : \alpha_4 + \alpha_5 + 4 \alpha_6 = 0 \quad (7.2.17-iv)
\]

\[
h^4 f_n^{(3)} : -2 \alpha_2 + 2 \alpha_3 - \alpha_4 + \alpha_5 = 0 \quad (7.2.17-v)
\]

\[
\frac{h^4 \left[ f_n^{(0)} \right]^3}{(f_n)^5} : \alpha_4 - \alpha_5 = 0 \quad (7.2.17-vi)
\]
\[
\frac{h^2 f_x^{(1)} f_x^{(2)}}{f_x} : -\alpha_x + \alpha_z = 0 \tag{7.2.17-vii}
\]

For equation (7.2.17-ii), (7.2.17-v) and (7.2.17-vi) to be symmetric, we impose the condition that

\(-\alpha_z + \alpha_y = 0 \tag{7.2.17-viii}\)

Solving equations (7.2.17-i)-(7.2.17-iv), (7.2.17-vi) and (7.2.17-viii) simultaneously by Mathematica, we immediately obtain the values

\[\alpha_x = \frac{5}{6} + 2\alpha_z, \quad \alpha_y = \alpha_z = \frac{1}{12} + \frac{\alpha_z}{2}, \quad \alpha_\mu = \alpha_z = -2\alpha_x \tag{7.2.18}\]

By substituting \(\alpha_z = \beta\) into (7.2.18) and from (7.2.5), we obtain

\[
y_{n+1} = 2y_n - y_{n-1} + h^2 \left\{ \left( \frac{5}{6} + 2\beta \right) f_n + \left( \frac{1}{12} + \frac{\beta}{2} \right) (f_{n-1} + f_{n+1}) \right\} \\
- 2\beta \left\{ \left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right) + \left( \frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}} \right) \right\} + \beta \left( \frac{f_{n+1}^2 + f_{n-1}^2}{f_{n-1} + f_{n+1}} \right) \quad \tag{7.2.19}\]

where \(\beta\) is an arbitrary constant.

From equation (7.2.19), by substituting \(\beta = 0\), gives the Numerov (AM) method as

\[
y_{n+1} = 2y_n - y_{n-1} + h^2 \left\{ \frac{5}{6} f_n + \frac{1}{12} (f_{n-1} + f_{n+1}) \right\} \\
= 2y_n - y_{n-1} + \frac{h^2}{12} [10f_n + f_{n-1} + f_{n+1}] \quad \tag{7.2.20}\]

and the \(C_{\mu}\) formula is obtained by substituting \(\beta = \frac{1}{144}\) into equation (7.2.19) to obtain

\[
y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288} \left\{ 244f_n + 25(f_{n-1} + f_{n+1}) \right\} \\
- 4 \left\{ \left( \frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}} \right) + \left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right) \right\} + 2 \left( \frac{f_{n+1}^2 + f_{n-1}^2}{f_{n-1} + f_{n+1}} \right) \quad \tag{7.2.21}\]

224
By substituting $\beta = -\frac{1}{144}$ into equation (7.2.19) we obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288} \{236f_n + 23(f_{n-1} + f_{n+1})$$

$$+ 4 \left[ \frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}} + \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right] - 2 \left( \frac{f_{n-1}^2 + f_{n+1}^2}{f_{n-1} + f_{n+1}} \right) \right\}$$  \hspace{1cm} (7.2.22)

### 7.2.1 Error Analysis of (7.2.20), (7.2.21) and (7.2.22)

Now, we consider the derivation of the local truncation error (LTE) from the formula (7.2.19). From the above discussion, we observe that the equation (7.2.19) is accurate to the fourth order. To obtain the local truncation error (LTE) of equation (7.2.19), we substitute equations (7.2.11)-(7.2.15) into equation (7.2.5) and equating with equation (7.2.8), we have

\[
LTE = h^6 \left\{ \frac{1}{720} (-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)} \\
+ \frac{12}{12} (\alpha_4 + \alpha_5 + 4\alpha_6) f_n^{(0)} f_n^{(3)} + \frac{1}{16} (\alpha_4 + \alpha_5) \left( \frac{f_n^{(2)}}{f_n} \right)^2 \\
+ \frac{1}{16} (-3\alpha_4 - 3\alpha_5 - 8\alpha_6) \frac{f_n^{(3)}(f_n^{(0)})^2}{(f_n)^2} + \frac{1}{16} (\alpha_4 + \alpha_5) \left( \frac{f_n^{(0)}}{f_n} \right)^4 \right\} \hspace{1cm} (7.2.23)
\]

\[
= \frac{h^6}{720} \left\{ (-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)} \\
+ 60(\alpha_4 + \alpha_5 + 4\alpha_6) \frac{f_n^{(0)} f_n^{(3)}}{f_n} + 45(\alpha_4 + \alpha_5) \left( \frac{f_n^{(2)}}{f_n} \right)^2 \\
+ 45(-3\alpha_4 - 3\alpha_5 - 8\alpha_6) \frac{f_n^{(3)}(f_n^{(0)})^2}{(f_n)^2} + 45(\alpha_4 + \alpha_5) \left( \frac{f_n^{(0)}}{f_n} \right)^4 \right\} \hspace{1cm} (7.2.24)
\]

By substituting $\alpha_6 = \beta$ and (7.2.18) into equation (7.2.24), we obtain the local truncation error of equation (7.2.19) as
and we define

\[ Mn = \frac{f_n^{(2)}(f_n^{(1)})^2 - (f_n^{(0)})^2}{(f_n)^3} \]  

For \( \beta = 0 \), the local truncation error (LTE) for Numerov (AM) method in equation (7.2.20) is given as

\[ LTE_{AM} = \frac{h^6}{720} \left\{ 3f_n^{(4)} - 180\beta \frac{(f_n^{(0)})^2}{f_n} + 180\beta \frac{f_n^{(2)}(f_n^{(0)})^2}{(f_n)^2} - 180\beta \frac{(f_n^{(4)})^4}{(f_n)^3} \right\} \]

\[ = \frac{h^6}{720} \left\{ 3f_n^{(4)} + 180\beta \left[ f_n f_n^{(2)}(f_n^{(1)})^2 - (f_n)^2(f_n^{(2)})^2 - (f_n^{(0)})^4 \right] \right\} \]  

(7.2.25)

and by substituting \( \beta = \frac{1}{144} \) in equation (7.2.25) the local truncation error (LTE) for the \( C_M \) formula in equation (7.2.22) is given as

\[ LTE_{C_M} = \frac{h^6}{720} \left\{ 12f_n^{(4)} + 5Mn \right\} \]  

(7.2.27)

\[ = \frac{h^6}{2880} \left\{ 12f_n^{(4)} - 5Mn \right\} \]  

(7.2.28)

7.2.2 Numerical Results For (7.2.16), (7.2.17) and (7.2.18)

The following are the numerical results of testing equations (7.2.16), (7.2.17) and (7.2.18). The comparison of accuracy between these methods are shown in Tables (7.1)-(7.5).
Problem 1: \( y'' + xy = 0 \)

Initial values: \( x_0 = 0, y_0 = 1, y'_0 = 2 \)

Exact Solution: 
\[
y = \left(1 - \frac{x^3}{3} + \frac{x^6}{180} - \ldots\right) + 2 \left(x - \frac{x^4}{12} + \frac{x^7}{504} - \ldots\right)
\]

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>1.19965000595E+01</td>
<td>1.19965000595E+01</td>
<td>0.00000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>1.1990708690E+01(X)</td>
<td>1.19706707302E+01</td>
<td>7.15795183538E-03</td>
</tr>
<tr>
<td></td>
<td>1.19907014100E+01(Y)</td>
<td>1.197981261443E-03</td>
<td>7.13609105633E-03</td>
</tr>
<tr>
<td>.30</td>
<td>1.1986403280E+01(Z)</td>
<td>1.15965491786E+01</td>
<td>2.51629850849E-02</td>
</tr>
<tr>
<td></td>
<td>1.19864826669E+01(Y)</td>
<td>2.52051170910E-02</td>
<td>2.51208531616E-02</td>
</tr>
<tr>
<td>.40</td>
<td>1.18442927854E+01(X)</td>
<td>1.177442925714E+01</td>
<td>5.63562968202E-02</td>
</tr>
<tr>
<td></td>
<td>1.18443985924E+01(Y)</td>
<td>5.64158985500E-02</td>
<td>5.62966412911E-02</td>
</tr>
<tr>
<td>.50</td>
<td>1.19680340801E+01(Z)</td>
<td>1.19480344721E+01</td>
<td>1.0266556442E-01</td>
</tr>
<tr>
<td></td>
<td>1.19680486496E+01(Y)</td>
<td>1.02740346854E-01</td>
<td>1.02590782401E-01</td>
</tr>
<tr>
<td>.60</td>
<td>1.21417681104E+01(Z)</td>
<td>1.210677028571E+01</td>
<td>1.66120740014E-01</td>
</tr>
<tr>
<td></td>
<td>1.21417866850E+01(Y)</td>
<td>1.66208910065E-01</td>
<td>1.66032578425E-01</td>
</tr>
<tr>
<td>.70</td>
<td>1.230262257646E+01(Z)</td>
<td>1.224663040833E+01</td>
<td>2.49227322500E-01</td>
</tr>
<tr>
<td></td>
<td>1.230264510783E+01(Y)</td>
<td>2.49327612106E-01</td>
<td>2.49127033487E-01</td>
</tr>
<tr>
<td>.80</td>
<td>1.24473187537E+01(Z)</td>
<td>1.236335522540E+01</td>
<td>3.55328090634E-01</td>
</tr>
<tr>
<td></td>
<td>1.24473582475E+01(Y)</td>
<td>3.55439679374E-01</td>
<td>3.55216502615E-01</td>
</tr>
<tr>
<td>.90</td>
<td>1.257243840106E+01(Z)</td>
<td>1.245250045357E+01</td>
<td>4.89043528245E-01</td>
</tr>
<tr>
<td></td>
<td>1.257246843201E+01(Y)</td>
<td>4.89165978591E-01</td>
<td>4.88921078748E-01</td>
</tr>
<tr>
<td>1.00</td>
<td>1.267437213004E+01(Z)</td>
<td>1.250952380952E+01</td>
<td>6.56890840780E-01</td>
</tr>
<tr>
<td></td>
<td>1.267440556832E+01(Y)</td>
<td>6.57024086297E-01</td>
<td>6.56757596241E-01</td>
</tr>
</tbody>
</table>

Table 7.1

Notations

The following notations are used in Tables (7.1)-(7.5)

- \( X \) denote formula \( AM \) formula (7.2.20)
- \( Y \) denote formula \( CM \) formula (7.2.21)
- \( Z \) denote formula \( CM \) formula (7.2.22)
Problem 2: \( y'' + 2x^2y = 0 \)

Initial values: \( x_0 = 0, y_0 = 1, y^{(1)}_0 = 1 \)

Exact Solution: \( y = \left( 1 - \frac{x^4}{6} + \frac{x^8}{168} - \cdots \right) + \left( x - \frac{x^4}{10} + \frac{x^9}{360} - \cdots \right) \)

<table>
<thead>
<tr>
<th>x_n</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.109998233340E+01</td>
<td>.109998233340E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.119970133876E+01(X)</td>
<td>.119970134999E+01</td>
<td>.936544166655E-08</td>
</tr>
<tr>
<td></td>
<td>.119970275584E+01(Y)</td>
<td>.117183061999E-05</td>
<td>.119056150351E-05</td>
</tr>
<tr>
<td></td>
<td>.119969992168E+01(Z)</td>
<td>.119056150351E-05</td>
<td>.119056150351E-05</td>
</tr>
<tr>
<td>.30</td>
<td>.129840729714E+01(X)</td>
<td>.12984074521E+01</td>
<td>.114042419673E-06</td>
</tr>
<tr>
<td></td>
<td>.129841126327E+01(Y)</td>
<td>.129841126327E+01</td>
<td>.114042419673E-06</td>
</tr>
<tr>
<td></td>
<td>.12984033101E+01(Z)</td>
<td>.119970275584E+01</td>
<td>.114042419673E-06</td>
</tr>
<tr>
<td>.40</td>
<td>.139471310095E+01(X)</td>
<td>.139471396246E+01</td>
<td>.617701420252E-06</td>
</tr>
<tr>
<td></td>
<td>.139472036629E+01(Y)</td>
<td>.139472036629E+01</td>
<td>.459149460478E-05</td>
</tr>
<tr>
<td></td>
<td>.139470583565E+01(Z)</td>
<td>.139470583565E+01</td>
<td>.582687034057E-05</td>
</tr>
<tr>
<td>.50</td>
<td>.148648365085E+01(X)</td>
<td>.148648701017E+01</td>
<td>.225990528824E-05</td>
</tr>
<tr>
<td></td>
<td>.148649473480E+01(Y)</td>
<td>.148649473480E+01</td>
<td>.519656853923E-05</td>
</tr>
<tr>
<td></td>
<td>.148647256698E+01(Z)</td>
<td>.148647256698E+01</td>
<td>.971632141034E-05</td>
</tr>
<tr>
<td>.60</td>
<td>.157074164286E+01(X)</td>
<td>.157075197074E+01</td>
<td>.657512117010E-05</td>
</tr>
<tr>
<td></td>
<td>.157075689858E+01(Y)</td>
<td>.157075689858E+01</td>
<td>.313724479653E-05</td>
</tr>
<tr>
<td></td>
<td>.157072638729E+01(Z)</td>
<td>.157072638729E+01</td>
<td>.16287389669E-04</td>
</tr>
<tr>
<td>.70</td>
<td>.164360423379E+01(X)</td>
<td>.164363156960E+01</td>
<td>.166313519292E-04</td>
</tr>
<tr>
<td></td>
<td>.164362386518E+01(Y)</td>
<td>.164362386518E+01</td>
<td>.468743749893E-05</td>
</tr>
<tr>
<td></td>
<td>.164358460263E+01(Z)</td>
<td>.164358460263E+01</td>
<td>.28575119389E-04</td>
</tr>
<tr>
<td>.80</td>
<td>.170027071388E+01(X)</td>
<td>.170033680417E+01</td>
<td>.388689400721E-04</td>
</tr>
<tr>
<td></td>
<td>.170029476769E+01(Y)</td>
<td>.170029476769E+01</td>
<td>.247224373692E-04</td>
</tr>
<tr>
<td></td>
<td>.17002466040E+01(Z)</td>
<td>.17002466040E+01</td>
<td>.53015299252E-04</td>
</tr>
<tr>
<td>.90</td>
<td>.173508683101E+01(X)</td>
<td>.173523947285E+01</td>
<td>.879658605241E-04</td>
</tr>
<tr>
<td></td>
<td>.173511517526E+01(Y)</td>
<td>.173511517526E+01</td>
<td>.71631756378E-04</td>
</tr>
<tr>
<td></td>
<td>.173505848723E+01(Z)</td>
<td>.173505848723E+01</td>
<td>.10430080438E-03</td>
</tr>
<tr>
<td>1.00</td>
<td>.174171581559E+01(X)</td>
<td>.174206349206E+01</td>
<td>.199577383509E-03</td>
</tr>
<tr>
<td></td>
<td>.174174811168E+01(Y)</td>
<td>.174174811168E+01</td>
<td>.18103840043E-03</td>
</tr>
<tr>
<td></td>
<td>.174168352008E+01(Z)</td>
<td>.174168352008E+01</td>
<td>.218116033568E-03</td>
</tr>
</tbody>
</table>

Table 7.2
Problem 3: \( y'' + x^2 y = 1 + x + x^2 \)

Initial values: \( x_0 = 0, \ y_0 = 2, \ y_0^{(0)} = 2 \)

Exact Solution: \( y = 2 \left( 1 - \frac{x^4}{12} + \frac{x^6}{672} - \ldots \right) + 2 \left( x - \frac{x^5}{20} + \frac{x^7}{1440} - \ldots \right) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>.220515731629E+01</td>
<td>.220515731629E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>0.20</td>
<td>.2421116719231E+01(X)</td>
<td>.242116688706E+01</td>
<td>.126074661035E-06</td>
</tr>
<tr>
<td></td>
<td>.2421116717267E+01(Y)</td>
<td></td>
<td>.117965771145E-06</td>
</tr>
<tr>
<td></td>
<td>.242116721194E+01(Z)</td>
<td></td>
<td>.134184050926E-06</td>
</tr>
<tr>
<td>0.30</td>
<td>.264857051467E+01(X)</td>
<td>.264856910711E+01</td>
<td>.531441912139E-06</td>
</tr>
<tr>
<td></td>
<td>.264857044271E+01(Y)</td>
<td></td>
<td>.504270769870E-06</td>
</tr>
<tr>
<td></td>
<td>.264857058664E+01(Z)</td>
<td></td>
<td>.558613056588E-06</td>
</tr>
<tr>
<td>0.40</td>
<td>.288743990707E+01(X)</td>
<td>.288743590441E+01</td>
<td>.138623499026E-05</td>
</tr>
<tr>
<td></td>
<td>.288743972700E+01(Y)</td>
<td></td>
<td>.132387036964E-05</td>
</tr>
<tr>
<td></td>
<td>.288744008714E+01(Z)</td>
<td></td>
<td>.144859962533E-05</td>
</tr>
<tr>
<td>0.50</td>
<td>.313722608903E+01(X)</td>
<td>.313721710689E+01</td>
<td>.286309062579E-05</td>
</tr>
<tr>
<td></td>
<td>.313722569779E+01(Y)</td>
<td></td>
<td>.273837975232E-05</td>
</tr>
<tr>
<td></td>
<td>.313722648028E+01(Z)</td>
<td></td>
<td>.298780156579E-05</td>
</tr>
<tr>
<td>0.60</td>
<td>.339659195430E+01(X)</td>
<td>.339657430537E+01</td>
<td>.519609809508E-05</td>
</tr>
<tr>
<td></td>
<td>.339659111444E+01(Y)</td>
<td></td>
<td>.494882859352E-05</td>
</tr>
<tr>
<td></td>
<td>.339659279418E+01(Z)</td>
<td></td>
<td>.544336791383E-05</td>
</tr>
<tr>
<td>0.70</td>
<td>.366323827217E+01(X)</td>
<td>.366320554629E+01</td>
<td>.893367162527E-05</td>
</tr>
<tr>
<td></td>
<td>.366323603200E+01(Y)</td>
<td></td>
<td>.832213777637E-05</td>
</tr>
<tr>
<td></td>
<td>.366324051235E+01(Z)</td>
<td></td>
<td>.954520814619E-05</td>
</tr>
<tr>
<td>0.80</td>
<td>.393372811772E+01(X)</td>
<td>.393366680462E+01</td>
<td>.155867549444E-04</td>
</tr>
<tr>
<td></td>
<td>.393370393028E+01(Y)</td>
<td></td>
<td>.943792519809E-05</td>
</tr>
<tr>
<td></td>
<td>.393375230618E+01(Z)</td>
<td></td>
<td>.217358423771E-04</td>
</tr>
<tr>
<td>0.90</td>
<td>.420332107723E+01(X)</td>
<td>.420319930665E+01</td>
<td>.289709273196E-04</td>
</tr>
<tr>
<td></td>
<td>.420312309456E+01(Y)</td>
<td></td>
<td>.181319240924E-04</td>
</tr>
<tr>
<td></td>
<td>.420351905588E+01(Z)</td>
<td></td>
<td>.760728224634E-04</td>
</tr>
<tr>
<td>1.00</td>
<td>.446583286944E+01(X)</td>
<td>.446557539683E+01</td>
<td>.576572081016E-04</td>
</tr>
<tr>
<td></td>
<td>.446547279445E+01(Y)</td>
<td></td>
<td>.229762941082E-04</td>
</tr>
<tr>
<td></td>
<td>.446619293003E+01(Z)</td>
<td></td>
<td>.138287488676E-03</td>
</tr>
</tbody>
</table>

Table 7.3
**Problem 4:** \( y^{(2)} - y = 0 \)

Initial values: \( x_0 = 0, y_0 = 1, y_0^{(1)} = -1 \)

Exact Solution : \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>0.904837418036E+00</td>
<td>0.904837418036E+00</td>
<td>0.000000000000E+00</td>
</tr>
<tr>
<td>0.20</td>
<td>0.818723210252E+00</td>
<td>0.818730753078E+00</td>
<td>9.21282811814E-05</td>
</tr>
<tr>
<td>0.30</td>
<td>0.740796234571E+00</td>
<td>0.740818220682E+00</td>
<td>2.96781450339E-04</td>
</tr>
<tr>
<td>0.40</td>
<td>0.670277212366E+00</td>
<td>0.670320046036E+00</td>
<td>6.38870942600E-04</td>
</tr>
<tr>
<td>0.50</td>
<td>0.606460980113E+00</td>
<td>0.60650659713E+00</td>
<td>1.14882238736E-03</td>
</tr>
<tr>
<td>0.60</td>
<td>0.548709348791E+00</td>
<td>0.548811636094E+00</td>
<td>1.86379618579E-03</td>
</tr>
<tr>
<td>0.70</td>
<td>0.496444810957E+00</td>
<td>0.496585303791E+00</td>
<td>2.82917826556E-03</td>
</tr>
<tr>
<td>0.80</td>
<td>0.449144721232E+00</td>
<td>0.449328964117E+00</td>
<td>4.10040081691E-03</td>
</tr>
<tr>
<td>0.90</td>
<td>0.406363078720E+00</td>
<td>0.406569659741E+00</td>
<td>5.74516605226E-03</td>
</tr>
<tr>
<td>1.00</td>
<td>0.367590796995E+00</td>
<td>0.367879441171E+00</td>
<td>7.84616219919E-03</td>
</tr>
</tbody>
</table>

Table 7.4
Problem 5: \( y^{(5)} - y \left( \frac{(Q + Bx)}{x} \right)^2 - Q / x^2 = 0 \)

Initial values: \( x_o = 1, y_o = 10e, y_o^{(5)} = 10e(Q + B) \)

Exact Solution: \( y = Cx^2e^{Bx} \)

\[ \left( \text{Set } B = 1, C = 10, Q = \frac{1}{2} \right) \]

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.10</td>
<td>.346587549802E+02</td>
<td>.346587549802E+02</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>1.20</td>
<td>.436390152385E+02</td>
<td>.436440703713E+02</td>
<td>.115826337471E-03</td>
</tr>
<tr>
<td>1.30</td>
<td>.543715401009E+02</td>
<td>.54387344516E+02</td>
<td>.290459041021E-03</td>
</tr>
<tr>
<td>1.40</td>
<td>.67143377291E+02</td>
<td>.671744823128E+02</td>
<td>.49341033302E-03</td>
</tr>
<tr>
<td>1.50</td>
<td>.822756510828E+02</td>
<td>.82338855610E+02</td>
<td>.707296611036E-03</td>
</tr>
<tr>
<td>1.60</td>
<td>.100149815184E+03</td>
<td>.100242328229E+03</td>
<td>.92284012738E-03</td>
</tr>
<tr>
<td>1.70</td>
<td>.12119389401E+03</td>
<td>.121331621407E+03</td>
<td>.11351319292E-02</td>
</tr>
<tr>
<td>1.80</td>
<td>.145900197789E+03</td>
<td>.146096168080E+03</td>
<td>.134137871608E-02</td>
</tr>
<tr>
<td>1.90</td>
<td>.174831793908E+03</td>
<td>.175101520393E+03</td>
<td>.154040058834E-02</td>
</tr>
<tr>
<td>2.00</td>
<td>.208632138690E+03</td>
<td>.208994069665E+03</td>
<td>.173176339644E-02</td>
</tr>
</tbody>
</table>

Table 7.5
7.3 DERIVATION OF THE C\(_M\) METHOD FOR PROBLEMS OF THE TYPE 
\( y^{(a)} = f(x, y) \)

By carrying out a similar procedure as the contraharmonic mean \((C,M)\) method for solving problems of the type \( y^{(a)} = f(x, y) \), a new centroidal mean \((C,M)\) can be established as

\[
y_{n+1} - 2y_n + y_{n-1} = h^2 \left\{ \alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_t \left[ \frac{2(f_n^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n-1})} \right] + \alpha_i \left[ \frac{2(f_n^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n-1})} \right] + \alpha_e \left[ \frac{2(f_{n-1}^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_{n-1} + f_{n+1})} \right] \right\}
\]  

(7.3.1)

By substituting equations (7.2.9)-(7.2.10) into the right hand side of equation (7.3.1) and using the Mathematica program for algebraic manipulation, we obtain the following results:

\[
\alpha_i \left[ \frac{2(f_n^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n-1})} \right] = \alpha_i f_n \left[ 1 - \frac{h^4 f_n^{(4)}}{2f_n} + \frac{h^2}{12} \left( \left( \frac{f_n^{(2)}}{f_n} \right)^2 + \frac{3f_n^{(2)}}{f_n} \right) \right]
\]

\[
+ \frac{h^4}{24} \left( 3 \left( \frac{f_n^{(3)}}{f_n} \right)^2 - 2 \frac{f_n^{(2)}}{f_n^2} - 2 \frac{f_n^{(2)}}{f_n^2} \right)
\]

\[
+ \frac{h^4}{144} \left( 15 \left( \frac{f_n^{(5)}}{f_n} \right)^5 - 60 \left( \frac{f_n^{(3)}}{f_n} \right)^2 f_n^{(2)} \right) + \frac{f_n^{(5)}}{f_n^5} \right) + 4 \frac{f_n^{(4)}}{f_n^3} \frac{f_n^{(3)}}{f_n^3} + 4 \frac{f_n^{(4)}}{f_n^3} \frac{f_n^{(3)}}{f_n^3} \right)
\]

\[
+ \frac{h^4}{1440} \left( 15 \left( \frac{f_n^{(5)}}{f_n} \right)^5 - 60 \left( \frac{f_n^{(3)}}{f_n} \right)^2 f_n^{(2)} \right) + 45 \frac{f_n^{(5)}}{f_n^5} \frac{f_n^{(3)}}{f_n^3} + 30 \frac{f_n^{(4)}}{f_n^3} \frac{f_n^{(3)}}{f_n^3} \right)
\]

\[
- \frac{20}{f_n^5} \left( \frac{f_n^{(2)}}{f_n} \right)^2 f_n^{(3)} - \frac{10}{f_n^4} \left( \frac{f_n^{(2)}}{f_n} \right)^2 f_n^{(3)} - \frac{6}{f_n^3} \left( \frac{f_n^{(3)}}{f_n^3} \right) + 0(h^4)
\]  

(7.3.2)

\[
\alpha_i \left[ \frac{2(f_n^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n-1})} \right] = \alpha_i f_n \left[ 1 + \frac{h^4 f_n^{(4)}}{2f_n} + \frac{h^2}{12} \left( \left( \frac{f_n^{(2)}}{f_n} \right)^2 + \frac{3f_n^{(2)}}{f_n} \right) \right]
\]
By substituting equations (7.2.11)-(7.2.12) and (7.3.2)-(7.3.4) into equation (7.3.1), we obtain

\( h^3 f_n \left\{ \sum_{i=0}^{n} \alpha_i + (-2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5) \frac{h^2 f_n^{(1)}}{2f_n} + (2\alpha_3 + 2\alpha_4 + \alpha_5 + 2\alpha_6) \frac{h^2 f_n^{(3)}}{4f_n} \right\} \)

\( + (\alpha_4 + \alpha_5 + 4\alpha_6) \frac{h^2}{12} \left[ \frac{f_n^{(4)}}{f_n} \right] + (-2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5) \frac{h^3 f_n^{(0)}}{12f_n} \)

\( + (\alpha_4 - \alpha_5) \frac{h^3}{24} \left[ \frac{f_n^{(0)}}{f_n} \right] + (-\alpha_4 + \alpha_5) \frac{h^3 f_n^{(2)}}{12[f_n]^2} \right\} + 0(h^6) \)

On equating equations (7.2.8) and (7.3.5), we obtain the following equations of condition, i.e.,

\( h^3 f_n : -1 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = 0 \) \hspace{2cm} (7.3.6-i)

\( h^3 f_n^{(1)} : -2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5 = 0 \) \hspace{2cm} (7.3.6-ii)

\( h^4 f_n^{(2)} : -1 + 6\alpha_2 + 6\alpha_3 + 3\alpha_4 + 3\alpha_5 + 6\alpha_6 = 0 \) \hspace{2cm} (7.3.6-iii)
For equation (7.3.6-ii), (7.3.6-v) and (7.3.6-vi) to be symmetric, we impose the condition that
\[-\alpha_2 + \alpha_3 = 0\]  
(7.3.6-viii)

Solving equations (7.3.6-i)-(7.3.6-iv), (7.3.6-vi) and (7.3.6-viii) simultaneously by Mathematica, we immediately obtain the values
\[\alpha_1 = \frac{5}{6} + 2\alpha_6, \quad \alpha_2 = \alpha_3 = \frac{1}{12} + \frac{\alpha_6}{2}, \quad \alpha_4 = \alpha_5 = -2\alpha_6\]  
(7.3.7)

By substituting \(\alpha_6 = \beta\) into (7.3.7) and from (7.3.1), we obtain the equation
\[y_{n+1} = 2y_n - y_{n-1} + h^2\left[\left(\frac{5}{6} + 2\beta\right)f_n + \left(\frac{1}{12} + \frac{\beta}{2}\right)(f_{n-1} + f_{n+1})\right]
-2\beta\left[\left(\frac{2(f_{n-1}^2 + f_{f_{n+1}^2} + f_{n+1}^2)}{3(f_n + f_{n-1})}\right) + \left(\frac{2(f_{n-1}^2 + f_{f_{n+1}^2} + f_{n+1}^2)}{3(f_n + f_{n-1})}\right)\right] + \beta\left(\frac{2(f_{n-1}^2 + f_{f_{n+1}^2} + f_{n+1}^2)}{3(f_n + f_{n-1})}\right)\]
(7.3.8)

where \(\beta\) is an arbitrary constant.

From equation (7.3.8), by substituting \(\beta = 0\), gives the Numerov (AM) method as
\[y_{n+1} = 2y_n - y_{n-1} + h^2\left[\frac{5}{6}f_n + \frac{1}{12}(f_{n-1} + f_{n+1})\right]\]
\[= 2y_n - y_{n-1} + \frac{h^2}{12}[10f_n + f_{n+1} + f_{n-1}]\]  
(7.3.9)
and the \(C_M\) formula is obtained by substituting \(\beta = \frac{1}{2880}\) into equation (7.3.8) to obtain

\[
y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{5760} \left\{ 4804f_n + 481(f_{n-1} + f_{n+1}) \right\}
\]

\[
-4 \left\{ \left( \frac{2(f_n^2 + f_{n-1}^2 + f_{n+1}^2)}{3(f_n + f_{n-1})} \right) + \left( \frac{2(f_n^2 + f_{n-1}^2 + f_{n+1}^2)}{3(f_n + f_{n+1})} \right) \right\} + 2 \left( \frac{2(f_n^2 + f_{n-1}^2 + f_{n+1}^2)}{3(f_n + f_{n+1})} \right)
\]

\[= 2880 (7.3.10)\]

By substituting \(\beta = \frac{1}{2880}\) into equation (7.3.8) to obtain

\[
y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{5760} \left\{ 4796f_n + 479(f_{n-1} + f_{n+1}) \right\}
\]

\[
+ 4 \left\{ \left( \frac{2(f_n^2 + f_{n-1}^2 + f_{n+1}^2)}{3(f_n + f_{n-1})} \right) + \left( \frac{2(f_n^2 + f_{n-1}^2 + f_{n+1}^2)}{3(f_n + f_{n+1})} \right) \right\} - 2 \left( \frac{2(f_n^2 + f_{n-1}^2 + f_{n+1}^2)}{3(f_n + f_{n+1})} \right)
\]

\[= 2880 (7.3.11)\]

7.3.1 Error Analysis of (7.3.10) and (7.3.11)

To obtain the local truncation error (LTE) of equation (7.3.8), we substitute equations (7.2.11)-(7.2.12) and (7.3.2)-(7.3.4) into equation (7.3.1) and equating with equation (7.2.8), we have the result

\[
LTE_{C_M} = h^6 \left( \frac{1}{720} (-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(e)}
\]

\[+ \frac{1}{36} (\alpha_4 + \alpha_3 + 4\alpha_6) \frac{f_n^{(e)}}{f_n} + \frac{1}{48} (\alpha_4 + \alpha_3) \left( \frac{f_n^{(e)}}{f_n} \right)^2
\]

\[+ \frac{1}{48} (-3\alpha_4 - 3\alpha_3 - 8\alpha_6) \frac{f_n^{(e)}}{(f_n)^2} + \frac{1}{48} (\alpha_4 + \alpha_5) \left( \frac{f_n^{(e)}}{(f_n)^2} \right)^3 \right) \]

\[= \frac{h^6}{720} (-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(e)} \]

235
By substituting \( \alpha_s = \beta \) and (7.3.7) into equation (7.3.13), we obtain the local truncation error of equation (7.3.11) as

\[
LTE_{c,M} = \frac{h^6}{240} \left[ f_s^{(4)} - 20 \beta \left( \frac{f_s^{(2)}}{f_s} \right)^2 + 20 \beta \frac{f_s^{(3)}(f_s^{(1)})^2}{(f_s)^3} - 20 \beta \frac{(f_s^{(1)})^4}{(f_s)^3} \right]
\]

(7.3.14)

and if we define

\[ Mn = \frac{f_s f_s^{(2)}(f_s^{(1)})^2 - (f_s)^2(f_s^{(2)})^2 - (f_s^{(1)})^4}{(f_s)^3} \]

For \( \beta = \frac{1}{2880} \), the local truncation error (LTE) for the \( C_M \) formula in equation (7.3.10) is given by

\[
LTE_{c,M} = \frac{h^6}{34560} \left[ 144 f_s^{(4)} + Mn \right]
\]

(7.3.15)

and by substituting \( \beta = -\frac{1}{2880} \) in equation (7.3.14) the local truncation error (LTE) for the \( C_M \) formula in (7.3.11) is given as

\[
LTE_{c,M} = \frac{h^6}{34560} \left[ 144 f_s^{(4)} + Mn \right]
\]

(7.3.16)
7.3.2 Numerical Results For (7.3.10) and (7.3.11)

The following tables are the numerical results of testing equations (7.3.10) and (7.3.11). The comparison on accuracy between these methods are shown in Tables (7.6)-(7.10).

Problem 1: \( y'' + xy = 0 \)

Initial values: \( x_0 = 0, \ y_0 = 1, \ y_0' = 2 \)

Exact Solution: \( y = \left( 1 - \frac{x^3}{3} + \frac{x^6}{180} - \ldots \right) + 2 \left( x - \frac{x^4}{12} + \frac{x^7}{504} - \ldots \right) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.119965000595E+01</td>
<td>.119965000595E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.139806708690E+01(X)</td>
<td>.139706707302E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.139806576794E+01(Y)</td>
<td>.139806405866E+01(Z)</td>
<td>.716739274460E-03</td>
</tr>
<tr>
<td>.30</td>
<td>.159365496416E+01(X)</td>
<td>.158965491786E+01</td>
<td>.251629850849E-02</td>
</tr>
<tr>
<td></td>
<td>.159364934757E+01(Y)</td>
<td>.159366058075E+01(Z)</td>
<td>.251276529493E-02</td>
</tr>
<tr>
<td>.40</td>
<td>.178442927854E+01(X)</td>
<td>.177442925714E+01</td>
<td>.563562698202E-02</td>
</tr>
<tr>
<td></td>
<td>.178441433382E+01(Y)</td>
<td>.178444223311E+01(Z)</td>
<td>.562720471252E-02</td>
</tr>
<tr>
<td>.50</td>
<td>.196803408010E+01(X)</td>
<td>.194803447421E+01</td>
<td>.102665564462E-01</td>
</tr>
<tr>
<td></td>
<td>.19680236728E+01(Y)</td>
<td>.196805793140E+01(Z)</td>
<td>.102502705441E-01</td>
</tr>
<tr>
<td>.60</td>
<td>.214176811045E+01(X)</td>
<td>.210677028571E+01</td>
<td>.166120744014E-01</td>
</tr>
<tr>
<td></td>
<td>.214170947714E+01(Y)</td>
<td>.214182674450E+01(Z)</td>
<td>.165842350244E-01</td>
</tr>
<tr>
<td>.70</td>
<td>.230262257646E+01(X)</td>
<td>.224663040833E+01</td>
<td>.249227322500E-01</td>
</tr>
<tr>
<td></td>
<td>.23025393295E+01(Y)</td>
<td>.230272122199E+01(Z)</td>
<td>.248788249310E-01</td>
</tr>
<tr>
<td>.80</td>
<td>.24473187537E+01(X)</td>
<td>.236335522540E+01</td>
<td>.355328090634E-01</td>
</tr>
<tr>
<td></td>
<td>.24471770734B+01(Y)</td>
<td>.2447486868172E+01(Z)</td>
<td>.354673082762E-01</td>
</tr>
<tr>
<td>.90</td>
<td>.257243840106E+01(X)</td>
<td>.245250045357E+01</td>
<td>.48904328245E-01</td>
</tr>
<tr>
<td></td>
<td>.25722082457E+01(Y)</td>
<td>.257268566414E+01(Z)</td>
<td>.488105076499E-01</td>
</tr>
<tr>
<td>1.00</td>
<td>.267437213004E+01(X)</td>
<td>.250952380952E+01</td>
<td>.656890840780E-01</td>
</tr>
<tr>
<td></td>
<td>.267404454710E+01(Y)</td>
<td>.26749973258E+01(Z)</td>
<td>.655585481784E-01</td>
</tr>
</tbody>
</table>

Table 7.6
**Notations**

The following notations are used in Tables (7.6)-(7.10)

- **X** denote formula $AM$ formula (7.3.9)
- **Y** denote formula $CM$ formula (7.3.10)
- **Z** denote formula $CM$ formula (7.3.11)

**Problem 2:** $y'' + 2x^2y = 0$

Initial values: $x_o = 0, y_o = 1, y_o^{(0)} = 1$

Exact Solution: $y = \left(1 - \frac{x^4}{6} + \frac{x^8}{168} - \cdots\right) + \left(x - \frac{x^5}{10} + \frac{x^9}{360} - \cdots\right)$

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>1.09998233340E+01</td>
<td>1.09998233340E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>1.19970133876E+01(X)</td>
<td>1.199701344999E+01</td>
<td>.936544166655E-08</td>
</tr>
<tr>
<td></td>
<td>1.19970101932E+01(Y)</td>
<td></td>
<td>2.75629277623E-06</td>
</tr>
<tr>
<td></td>
<td>1.19970163820E+01(Z)</td>
<td></td>
<td>2.56898394155E-06</td>
</tr>
<tr>
<td>.30</td>
<td>1.29840729714E+01(X)</td>
<td>1.2984074521E+01</td>
<td>1.14042419673E-06</td>
</tr>
<tr>
<td></td>
<td>1.2984055781E+01(Y)</td>
<td></td>
<td>1.4633016009E-05</td>
</tr>
<tr>
<td></td>
<td>1.29840904647E+01(Z)</td>
<td></td>
<td>1.2338495166E-05</td>
</tr>
<tr>
<td>.40</td>
<td>1.3947310095E+01(X)</td>
<td>1.39473196246E+01</td>
<td>.617701420252E-06</td>
</tr>
<tr>
<td></td>
<td>1.39470735942E+01(Y)</td>
<td></td>
<td>.473433837777E-05</td>
</tr>
<tr>
<td></td>
<td>1.39471884249E+01(Z)</td>
<td></td>
<td>.34989423915E-05</td>
</tr>
<tr>
<td>.50</td>
<td>1.48648365085E+01(X)</td>
<td>1.48648701017E+01</td>
<td>.225990528824E-05</td>
</tr>
<tr>
<td></td>
<td>1.48646913736E+01(Y)</td>
<td></td>
<td>1.20235190288E-04</td>
</tr>
<tr>
<td></td>
<td>1.48648164408E+01(Z)</td>
<td></td>
<td>.75037499761E-05</td>
</tr>
<tr>
<td>.60</td>
<td>1.57074164286E+01(X)</td>
<td>1.57075197074E+01</td>
<td>.65751211701E-05</td>
</tr>
<tr>
<td></td>
<td>1.57071052442E+01(Y)</td>
<td></td>
<td>2.63862954209E-04</td>
</tr>
<tr>
<td></td>
<td>1.5707726758E+01(Z)</td>
<td></td>
<td>3.236230704E-04</td>
</tr>
<tr>
<td>.70</td>
<td>1.64360423379E+01(X)</td>
<td>1.64363156960E+01</td>
<td>1.66313519292E-04</td>
</tr>
<tr>
<td></td>
<td>1.64354478252E+01(Y)</td>
<td></td>
<td>.52802032902E-04</td>
</tr>
<tr>
<td></td>
<td>1.64366368058E+01(Z)</td>
<td></td>
<td>.19539925706E-04</td>
</tr>
<tr>
<td>.80</td>
<td>1.70027071388E+01(X)</td>
<td>1.70033680417E+01</td>
<td>.38868400721E-04</td>
</tr>
<tr>
<td></td>
<td>1.70016656480E+01(Y)</td>
<td></td>
<td>1.00120967343E-03</td>
</tr>
<tr>
<td></td>
<td>1.70037486593E+01(Z)</td>
<td></td>
<td>.223848632394E-04</td>
</tr>
<tr>
<td>.90</td>
<td>1.73508683101E+01(X)</td>
<td>1.73523947285E+01</td>
<td>.87965805241E-04</td>
</tr>
<tr>
<td></td>
<td>1.73491649468E+01(Y)</td>
<td></td>
<td>1.86128870414E-03</td>
</tr>
<tr>
<td></td>
<td>1.73525717515E+01(Z)</td>
<td></td>
<td>1.02016483915E-04</td>
</tr>
<tr>
<td>1.00</td>
<td>1.74171581559E+01(X)</td>
<td>1.74206349206E+01</td>
<td>.199577383509E-03</td>
</tr>
<tr>
<td></td>
<td>1.74145265621E+01(Y)</td>
<td></td>
<td>.350639258562E-03</td>
</tr>
<tr>
<td></td>
<td>1.74197899342E+01(Z)</td>
<td></td>
<td>.485049185424E-04</td>
</tr>
</tbody>
</table>

Table 7.7
Problem 3: \( y'' + x^2y = 1 + x + x^2 \)

Initial values: \( x_0 = 0, \ y_0 = 2, \ y_0' = 2 \)

Exact Solution: \( y = 2\left(1 - \frac{x^4}{12} + \frac{x^8}{672} - \ldots\right) + 2\left(x - \frac{x^5}{20} + \frac{x^9}{1440} - \ldots\right) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( .10 )</td>
<td>( .220515731629E+01 )</td>
<td>( .220515731629E+01 )</td>
<td>( .00000000000000E+00 )</td>
</tr>
<tr>
<td>( .20 )</td>
<td>( .2421116719231E+01(X) ) ( .242116688706E+01(Y) ) ( .242115591560E+01(Z) )</td>
<td>( .2421116719231E+01(X) ) ( .242116688706E+01(Y) ) ( .242115591560E+01(Z) )</td>
<td>( .126074661035E-06 ) ( .478362362933E-05 ) ( .453147430726E-05 )</td>
</tr>
<tr>
<td>( .30 )</td>
<td>( .264857051467E+01(X) ) ( .264860489112E+01(Y) ) ( .26485613821E+01(Z) )</td>
<td>( .264857051467E+01(X) ) ( .264860489112E+01(Y) ) ( .26485613821E+01(Z) )</td>
<td>( .531441912139E-06 ) ( .135106951500E-04 ) ( .124478163520E-04 )</td>
</tr>
<tr>
<td>( .40 )</td>
<td>( .288743990707E+01(X) ) ( .288750924092E+01(Y) ) ( .288737057312E+01(Z) )</td>
<td>( .288743990707E+01(X) ) ( .288750924092E+01(Y) ) ( .288737057312E+01(Z) )</td>
<td>( .138623499026E-05 ) ( .253984906416E-04 ) ( .22620563602E-04 )</td>
</tr>
<tr>
<td>( .50 )</td>
<td>( .313722608903E+01(X) ) ( .313734156911E+01(Y) ) ( .313711060850E+01(Z) )</td>
<td>( .313722608903E+01(X) ) ( .313734156911E+01(Y) ) ( .313711060850E+01(Z) )</td>
<td>( .286309062579E-05 ) ( .396728102554E-04 ) ( .339467718105E-04 )</td>
</tr>
<tr>
<td>( .60 )</td>
<td>( .339659195430E+01(X) ) ( .339676318035E+01(Y) ) ( .339642072683E+01(Z) )</td>
<td>( .339659195430E+01(X) ) ( .339676318035E+01(Y) ) ( .339642072683E+01(Z) )</td>
<td>( .519609809508E-05 ) ( .556074919201E-04 ) ( .452157171116E-04 )</td>
</tr>
<tr>
<td>( .70 )</td>
<td>( .366323827217E+01(X) ) ( .366347210125E+01(Y) ) ( .366300443933E+01(Z) )</td>
<td>( .366323827217E+01(X) ) ( .366347210125E+01(Y) ) ( .366300443933E+01(Z) )</td>
<td>( .893367162527E-05 ) ( .727654931235E-04 ) ( .548991753122E-04 )</td>
</tr>
<tr>
<td>( .80 )</td>
<td>( .393372811772E+01(X) ) ( .393402727697E+01(Y) ) ( .393342894995E+01(Z) )</td>
<td>( .393372811772E+01(X) ) ( .393402727697E+01(Y) ) ( .393342894995E+01(Z) )</td>
<td>( .155867549444E-04 ) ( .916377434813E-04 ) ( .604664003490E-04 )</td>
</tr>
<tr>
<td>( .90 )</td>
<td>( .420332107723E+01(X) ) ( .420368256725E+01(Y) ) ( .420295959898E+01(Z) )</td>
<td>( .420332107723E+01(X) ) ( .420368256725E+01(Y) ) ( .420295959898E+01(Z) )</td>
<td>( .289709273196E-04 ) ( .114974468059E-03 ) ( .570367334037E-04 )</td>
</tr>
<tr>
<td>1.00</td>
<td>( .446583286944E+01(X) ) ( .44662461826E+01(Y) ) ( .44654194844E+01(Z) )</td>
<td>( .446583286944E+01(X) ) ( .44662461826E+01(Y) ) ( .44654194844E+01(Z) )</td>
<td>( .576572081016E-04 ) ( .150220604312E-03 ) ( .349133923420E-04 )</td>
</tr>
</tbody>
</table>

Table 7.8
Problem 4: \( y^{(2)} - y = 0 \)

Initial values: \( x_0 = 0, y_0 = 1, y_0^{(1)} = -1 \)

Exact Solution: \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.904837418036E+00</td>
<td>.904837418036E+00</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.818723210252E+00(X)</td>
<td>.818730753078E+00</td>
<td>.921282811814E-05</td>
</tr>
<tr>
<td></td>
<td>.818732636424E+00(Y)</td>
<td></td>
<td>.229936894503E-05</td>
</tr>
<tr>
<td></td>
<td>.818713784863E+00(Z)</td>
<td></td>
<td>.207252051814E-04</td>
</tr>
<tr>
<td>.30</td>
<td>.740796234571E+00(X)</td>
<td>.740818220682E+00</td>
<td>.296781450339E-04</td>
</tr>
<tr>
<td></td>
<td>.740823708070E+00(Y)</td>
<td></td>
<td>.740719874887E-05</td>
</tr>
<tr>
<td></td>
<td>.740768761269E+00(Z)</td>
<td></td>
<td>.667632237556E-04</td>
</tr>
<tr>
<td>.40</td>
<td>.670277221236E+00(X)</td>
<td>.670320046036E+00</td>
<td>.638870942600E-04</td>
</tr>
<tr>
<td></td>
<td>.670330734491E+00(Y)</td>
<td></td>
<td>.159453021364E-04</td>
</tr>
<tr>
<td></td>
<td>.670223708947E+00(Z)</td>
<td></td>
<td>.143718047987E-03</td>
</tr>
<tr>
<td>.50</td>
<td>.606460980113E+00(X)</td>
<td>.606530659713E+00</td>
<td>.114882238736E-03</td>
</tr>
<tr>
<td></td>
<td>.606548050870E+00(Y)</td>
<td></td>
<td>.286731707725E-04</td>
</tr>
<tr>
<td></td>
<td>.606373912218E+00(Z)</td>
<td></td>
<td>.258432929178E-03</td>
</tr>
<tr>
<td>.60</td>
<td>.548709348791E+00(X)</td>
<td>.548811636094E+00</td>
<td>.186379618579E-03</td>
</tr>
<tr>
<td></td>
<td>.548837165965E+00(Y)</td>
<td></td>
<td>.465184595057E-04</td>
</tr>
<tr>
<td></td>
<td>.548581538215E+00(Z)</td>
<td></td>
<td>.419265670619E-03</td>
</tr>
<tr>
<td>.70</td>
<td>.496444810957E+00(X)</td>
<td>.496585303791E+00</td>
<td>.282917826556E-03</td>
</tr>
<tr>
<td></td>
<td>.496620369775E+00(Y)</td>
<td></td>
<td>.706142202927E-04</td>
</tr>
<tr>
<td></td>
<td>.496269265204E+00(Z)</td>
<td></td>
<td>.636423560507E-03</td>
</tr>
<tr>
<td>.80</td>
<td>.449144721232E+00(X)</td>
<td>.449328964117E+00</td>
<td>.410040081691E-03</td>
</tr>
<tr>
<td></td>
<td>.449374950412E+00(Y)</td>
<td></td>
<td>.102344380291E-03</td>
</tr>
<tr>
<td></td>
<td>.448914515374E+00(Z)</td>
<td></td>
<td>.922372641460E-03</td>
</tr>
<tr>
<td>.90</td>
<td>.406336078720E+00(X)</td>
<td>.406569659741E+00</td>
<td>.574516605226E-03</td>
</tr>
<tr>
<td></td>
<td>.406627961541E+00(Y)</td>
<td></td>
<td>.143399289503E-03</td>
</tr>
<tr>
<td></td>
<td>.406044234504E+00(Z)</td>
<td></td>
<td>.129233754661E-02</td>
</tr>
<tr>
<td>1.00</td>
<td>.367590796995E+00(X)</td>
<td>.367879441171E+00</td>
<td>.784616219919E-03</td>
</tr>
<tr>
<td></td>
<td>.367951487994E+00(Y)</td>
<td></td>
<td>.195843567524E-03</td>
</tr>
<tr>
<td></td>
<td>.367230166352E+00(Z)</td>
<td></td>
<td>.176491194365E-02</td>
</tr>
</tbody>
</table>

Table 7.9
Problem 5: \( y^{(0)} - y\left[\left(\frac{(Q + Bx)}{x}\right)^2 - \frac{Q}{x^2}\right] = 0 \)

Initial values: \( x_0 = 1, y_0 = 10e, y_0^{(0)} = 10e(Q + B) \)

Exact Solution: \( y = Cx^0e^{\frac{t}{2}} \)

\[
\left( \text{Set } B = 0, C = 10, Q = \frac{3}{2} \right)
\]

<table>
<thead>
<tr>
<th>( xn )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.10</td>
<td>.346587549802E+02</td>
<td>.346587549802E+02</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>1.20</td>
<td>.436390152395E+02</td>
<td>.436440703713E+02</td>
<td>.115826337471E-03</td>
</tr>
<tr>
<td>1.30</td>
<td>.543715401099E+02</td>
<td>.543873445416E+02</td>
<td>.290590410221E-03</td>
</tr>
<tr>
<td>1.40</td>
<td>.67141377291E+02</td>
<td>.671744823128E+02</td>
<td>.493410335370E-03</td>
</tr>
<tr>
<td>1.50</td>
<td>.822756510828E+02</td>
<td>.82338855610E+02</td>
<td>.707296611036E-03</td>
</tr>
<tr>
<td>1.60</td>
<td>.100149815184E+03</td>
<td>.100242328229E+03</td>
<td>.922894012738E-03</td>
</tr>
<tr>
<td>1.70</td>
<td>.121193894010E+03</td>
<td>.121331621407E+03</td>
<td>.113513192929E-02</td>
</tr>
<tr>
<td>1.80</td>
<td>.145900197789E+03</td>
<td>.146096168080E+03</td>
<td>.134137871608E-02</td>
</tr>
<tr>
<td>1.90</td>
<td>.174831793908E+03</td>
<td>.175101520393E+03</td>
<td>.154004058834E-02</td>
</tr>
<tr>
<td>2.00</td>
<td>.208632138699E+03</td>
<td>.208994066965E+03</td>
<td>.17317639644E-02</td>
</tr>
</tbody>
</table>

Table 7.10
7.4 DERIVATION OF THE \( H_M \) METHOD FOR PROBLEMS OF THE TYPE \( y^{(3)} = f(x, y) \)

For solving problems of the type \( y^{(3)} = f(x, y) \), we now consider the harmonic mean formula \( H_M \) as

\[
y_{n+1} - 2y_n + y_{n-1} = h^2 \left\{ \alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4 \left[ \frac{2(f_n f_{n-1})}{f_n + f_{n-1}} \right] \right\}
\]

\[
+ \alpha_5 \left[ \frac{2(f_n f_{n+1})}{f_n + f_{n+1}} \right] + \alpha_6 \left[ \frac{2(f_{n-1} f_{n+1})}{f_{n-1} + f_{n+1}} \right] \right\}
\]

(7.4.1)

By substituting equations (7.2.9)-(7.2.10) into the right hand side of equation (7.4.1) and using the Mathematica program for algebraic manipulation, we obtain the following results:

\[
\alpha_i \left[ \frac{2f_n f_{n-1}}{f_n + f_{n-1}} \right] = \alpha_i f_n \left[ 1 - \frac{h^2 f_n^{(1)}}{2f_n} + \frac{h^2}{4} \left( -\frac{f_n^{(1)}}{f_n} + \frac{f_n^{(2)}}{f_n} \right) \right] 
\]

\[
+ \frac{h^3}{24} \left( -3 \left( \frac{f_n^{(1)}}{f_n} \right)^3 \right) + 6 \frac{f_n^{(1)} f_n^{(2)}}{(f_n)^2} - 2 \frac{f_n^{(3)}}{(f_n)^2} \right)
\]

\[
+ \frac{h^4}{48} \left( -3 \left( \frac{f_n^{(1)}}{f_n} \right)^4 \right) + 9 \left( \frac{f_n^{(1)}}{f_n} \right)^2 \left( \frac{f_n^{(2)}}{(f_n)^3} \right) - 3 \frac{f_n \left( \frac{f_n^{(2)}}{(f_n)^2} \right)^2}{(f_n)^3} - 4 \frac{f_n f_n^{(1)} f_n^{(2)}}{(f_n)^3} + \left( \frac{f_n}{(f_n)^3} \right) \right)
\]

\[
+ \frac{h^5}{480} \left( -15 \left( \frac{f_n^{(1)}}{f_n} \right)^5 \right) + 60 \left( \frac{f_n^{(1)}}{f_n} \right)^3 \left( \frac{f_n^{(2)}}{(f_n)^4} \right) - 45 \frac{f_n f_n^{(1)} \left( \frac{f_n^{(2)}}{(f_n)^2} \right)^2}{(f_n)^4} - 30 \frac{f_n \left( \frac{f_n}{(f_n)^3} \right)^2}{(f_n)^3} \right)
\]

\[
+ 20 \left( \frac{f_n^{(2)}}{f_n^{(1)}} \right)^2 \frac{f_n^{(3)}}{(f_n)^4} - 10 \left( \frac{f_n^{(2)}}{f_n^{(1)}} \right)^2 \left( \frac{f_n^{(3)}}{(f_n)^4} \right) - 2 \left( \frac{f_n^{(2)}}{(f_n)^4} \right)^2 \right) \right\}
\]

(7.4.2)

\[
\alpha_i \left[ \frac{2f_n f_{n+1}}{f_n + f_{n+1}} \right] = \alpha_i f_n \left[ 1 + \frac{h^2 f_n^{(1)}}{2f_n} + \frac{h^2}{4} \left( -\frac{f_n^{(1)}}{f_n} + \frac{f_n^{(2)}}{f_n} \right) \right] 
\]
By substituting equations (7.2.11)-(7.2.12) and (7.4.2)-(7.4.4) into equation (7.4.1), we obtain

\[ h^2 f_s \left[ \sum_{i=1}^{k} \alpha_i + (-2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_s) \frac{h^2 f_s^{(1)}}{2 f_n} + (2\alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_5 + 2\alpha_s) \frac{h^2 f_s^{(2)}}{4 f_n} \right. \]
\[ \left. + (\alpha_4 + \alpha_5 + 4\alpha_s) \frac{h^2}{4} \left[ \frac{f_s^{(1)}}{f_n} \right]^2 + (-2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_s) \frac{h^3 f_s^{(3)}}{12 f_n} \right. \]
\[ \left. + (2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5 + 2\alpha_s) \frac{h^3 f_s^{(3)} f_s^{(1)}}{4 f_n^2} \right] + 0(h^6) \]

By equating equations (7.2.8) and (7.4.5), we obtain the following equations of condition, i.e.,

\[ h^2 f_s : -1 + \alpha_1 + \alpha_s + \alpha_s + \alpha_s + \alpha_5 + \alpha_5 = 0 \]  \hspace{1cm} (7.4.6-i)
\[ h^2 f_s^{(1)} : -2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_s = 0 \]  \hspace{1cm} (7.4.6-ii)
\[ h^3 f_s^{(2)} : -1 + 6\alpha_2 + 6\alpha_3 + 3\alpha_4 + 3\alpha_s + 6\alpha_s = 0 \]  \hspace{1cm} (7.4.6-iii)

243
\[
\frac{h^4 f_n^{(3)}}{f_n} : -\alpha_4 - \alpha_3 - 4\alpha_2 = 0 \quad \text{(7.4.6-iv)}
\]
\[
\frac{h^4 f_n^{(3)}}{f_n} : -2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5 = 0 \quad \text{(7.4.6-v)}
\]
\[
\frac{h^4 f_n^{(3)}}{(f_n)^2} : \alpha_4 - \alpha_5 = 0 \quad \text{(7.4.6-vi)}
\]
\[
\frac{h^4 f_n^{(3)} f_n^{(3)}}{f_n} : -\alpha_4 + \alpha_5 = 0 \quad \text{(7.4.6-vii)}
\]

For equations (7.4.6-ii), (7.4.6-v) and (7.4.6-vi) to be symmetric, we impose the condition that
\[
-\alpha_2 + \alpha_3 = 0 \quad \text{(7.4.6-viii)}
\]

Solving equations (7.4.6-i)-(7.4.6-iv), (7.4.6-vi) and (7.4.6-viii) simultaneously by Mathematica, we immediately obtain the values
\[
\alpha_1 = \frac{5}{6} + 2\alpha_2, \quad \alpha_2 = \alpha_3 = \frac{1}{12} + \frac{\alpha_4}{2}, \quad \alpha_4 = \alpha_5 = -2\alpha_2 \quad \text{(7.4.7)}
\]

By substituting \(\alpha_4 = \beta\) into (7.4.7) and from (7.4.1), we obtain
\[
y_{n+1} = 2y_n - y_{n-1} + h^2 \left( \left( \frac{5}{6} + 2\beta \right) f_n + \left( \frac{1}{12} + \frac{\beta}{2} \right) (f_{n-1} + f_{n+1}) \right)
\]
\[
-2\beta \left[ \frac{2f_n f_{n-1}}{f_n + f_{n-1}} + \frac{2f_n f_{n+1}}{f_n + f_{n+1}} \right] + \beta \left( \frac{2f_n f_{n-1}}{f_{n-1} + f_n} \right) \quad \text{(7.4.8)}
\]

where \(\beta\) is an arbitrary constant.

From equation (7.4.8), by substituting \(\beta = 0\), gives the Numerov (AM) method as
\[
y_{n+1} = 2y_n - y_{n-1} + h^2 \left[ \frac{5}{6} f_n + \frac{1}{12} (f_{n-1} + f_{n+1}) \right]
\]
\[
= 2y_n - y_{n-1} + \frac{h^2}{12} [10f_n + f_{n-1} + f_{n+1}] \quad \text{(7.4.9)}
\]

and the \(H_nM\) formula is obtained by substituting \(\beta = \frac{1}{144}\) into equation (7.4.8) to obtain

244
\[ y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288} \left[ 144f_n + 25(f_{n-1} + f_{n+1}) \right] \]

\[ -4 \left[ \left( \frac{2f_nf_{n-1}}{f_n + f_{n-1}} \right) + \left( \frac{2f_nf_{n+1}}{f_n + f_{n+1}} \right) \right] + 2\left( \frac{2f_nf_{n-1}}{f_{n-1} + f_{n-2}} \right) \right] \]

By substituting \( \beta = -\frac{1}{144} \) into equation (7.4.8) to obtain

\[ y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288} \left[ 236f_n + 23(f_{n-1} + f_{n+1}) \right] \]

\[ + 4 \left[ \left( \frac{2f_nf_{n-1}}{f_n + f_{n-1}} \right) + \left( \frac{2f_nf_{n+1}}{f_n + f_{n+1}} \right) \right] - 2\left( \frac{2f_nf_{n-1}}{f_{n-1} + f_{n-2}} \right) \]

7.4.1 Error Analysis of (7.4.10) and (7.4.11)

To obtain the local truncation error (LTE) of equation (7.4.8), we substitute equations (7.2.11)-(7.2.12) and (7.4.2)-(7.4.4) into equation (7.4.1) and on equating with equation (7.2.8), we have

\[ \text{LTE}^{h,M} = h^6 \left\{ \frac{1}{720} (-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)} \right. \]

\[ + \frac{1}{12} (-\alpha_4 - \alpha_5 - 4\alpha_6) \frac{f_n^{(0)} f_n^{(3)}}{f_n} + \frac{1}{16} (-\alpha_4 - \alpha_5) \left( \frac{f_n^{(2)}}{f_n} \right)^2 \]

\[ + \frac{1}{16} (3\alpha_4 + 3\alpha_5 + 8\alpha_6) \frac{f_n^{(2)} f_n^{(0)}}{(f_n)^2} + \frac{1}{16} (-\alpha_4 - \alpha_5) \left( \frac{f_n^{(0)}}{f_n^3} \right) \]

\[ = \frac{h^6}{720} \left\{ (-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)} \right. \]

\[ + 60 (-\alpha_4 - \alpha_5 - 4\alpha_6) \frac{f_n^{(0)} f_n^{(3)}}{f_n} + 15 (-\alpha_4 - \alpha_5) \left( \frac{f_n^{(2)}}{f_n} \right)^2 \]

245
By substituting \(\alpha_s = \beta\) and (7.4.7) into equation (7.4.13), we obtain the local truncation error of equation (7.4.11) as

\[
LTE''_{\mu''} = \frac{h^5}{240} \left\{ \frac{f_s^{(4)}}{f_s} + 60\beta \frac{f_s^{(3)}}{(f_s)^2} - 60\beta \frac{f_s^{(2)}}{(f_s)^2} + 60\beta \frac{f_s^{(4)}}{(f_s)^3} \right\}
\]

and we define

\[Mn = \frac{f_s f_s^{(2)} (f_s^{(3)})^2 - (f_s)^2 (f_s^{(3)})^2 - (f_s)^4}{(f_s)^3}\]

For \(\beta = \frac{1}{144}\), the local truncation error (LTE) for \(H,M\) formula in equation (7.4.10) is given by

\[
LTE''_{\mu''} = \frac{h^5}{2880} \left\{ 12 f_s^{(4)} + 5 Mn \right\} \tag{7.4.15}
\]

and by substituting \(\beta = -\frac{1}{144}\) in equation (7.4.14) the local truncation error (LTE) for the \(H,M\) formula in (7.4.11) is given as

\[
LTE''_{\mu''} = \frac{h^5}{2880} \left\{ 12 f_s^{(4)} - 5 Mn \right\} \tag{7.4.16}
\]

7.4.2 Numerical Results For (7.4.10) and (7.4.11)

The following are the numerical results of testing equations (7.4.10) and (7.4.11). The comparison of accuracy between these methods are shown in Tables (7.11)-(7.15).
Problem 1: \( y'' + xy = 0 \)

Initial values: \( x_0 = 0, y_0 = 1, y'_0 = 2 \)

Exact Solution: \( y = \left( 1 - \frac{x^3}{3} + \frac{x^5}{180} - \ldots \right) + 2 \left( x - \frac{x^4}{12} + \frac{x^7}{504} - \ldots \right) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.119965000595E+01</td>
<td>.119965000595E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.139806707690E+01(X)</td>
<td>.139706707302E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.13980403280E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.139807014100E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.30</td>
<td>.15936549616E+01(X)</td>
<td>.158965491786E+01</td>
<td>.251629850849E-02</td>
</tr>
<tr>
<td></td>
<td>.159364826663E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.159366166169E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.40</td>
<td>.178442927854E+01(X)</td>
<td>.177442925714E+01</td>
<td>.713609105633E-02</td>
</tr>
<tr>
<td></td>
<td>.178441869788E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.178443985924E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.50</td>
<td>.19680340801E+01(X)</td>
<td>.194803447421E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.196801951229E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.196804864796E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.60</td>
<td>.214176811045E+01(X)</td>
<td>.210677028571E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.214174953598E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.214178668501E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.70</td>
<td>.230262257646E+01(X)</td>
<td>.224663040833E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.230260004523E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.230264510783E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.80</td>
<td>.244733187537E+01(X)</td>
<td>.236335522540E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.244730550316E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.244735824775E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.90</td>
<td>.257243840106E+01(X)</td>
<td>.245250045357E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.257240837031E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.257246843201E+01(Z)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>.267437213004E+01(X)</td>
<td>.250952380952E+01</td>
<td>.715795183538E-03</td>
</tr>
<tr>
<td></td>
<td>.26743369201E+01(Y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.267440556832E+01(Z)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.11

Notations

The following notations are used in Tables (7.11)-(7.15):

- \( X \) denote formula \( AM \) formula (7.4.9)
- \( Y \) denote formula \( H,M \) formula (7.4.10)
- \( Z \) denote formula \( H,M \) formula (7.4.11)
**Problem 2:** \( y'' + 2x^2y = 0 \)

Initial values: \( x_0 = 0, y_0 = 1, y'_0 = 1 \)

Exact Solution: \( y = \left( 1 - \frac{x^4}{6} + \frac{x^8}{168} - \ldots \right) + \left( x - \frac{x^4}{10} + \frac{x^9}{360} - \ldots \right) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>.109998233340E+01</td>
<td>.109998233340E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>0.20</td>
<td>.1199701333876E+01(X)</td>
<td>.119970134999E+01</td>
<td>.936544166655E-08</td>
</tr>
<tr>
<td>0.20</td>
<td>.119969992168E+01(Y)</td>
<td>.119056150351E-05</td>
<td></td>
</tr>
<tr>
<td>0.20</td>
<td>.119970275584E+01(Z)</td>
<td>.117183061999E-05</td>
<td></td>
</tr>
<tr>
<td>0.30</td>
<td>.129840729714E+01(X)</td>
<td>.129840744521E+01</td>
<td>.11404219673E-06</td>
</tr>
<tr>
<td>0.30</td>
<td>.12984033101E+01(Y)</td>
<td>.31684816689E-05</td>
<td></td>
</tr>
<tr>
<td>0.30</td>
<td>.129841126327E+01(Z)</td>
<td>.294057112078E-05</td>
<td></td>
</tr>
<tr>
<td>0.40</td>
<td>.139471310095E+01(X)</td>
<td>.139471396246E+01</td>
<td>.617701420252E-06</td>
</tr>
<tr>
<td>0.40</td>
<td>.139470583565E+01(Y)</td>
<td>.582687034057E-05</td>
<td></td>
</tr>
<tr>
<td>0.40</td>
<td>.139472036629E+01(Z)</td>
<td>.459149460478E-05</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>.148648365085E+01(X)</td>
<td>.148648701017E+01</td>
<td>.225990528824E-05</td>
</tr>
<tr>
<td>0.50</td>
<td>.148647256698E+01(Y)</td>
<td>.971632141034E-05</td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td>.14864973480E+01(Z)</td>
<td>.519656850478E-05</td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>.157074164286E+01(X)</td>
<td>.157075197074E+01</td>
<td>.657512117010E-05</td>
</tr>
<tr>
<td>0.60</td>
<td>.157072638729E+01(Y)</td>
<td>.162873889669E-04</td>
<td></td>
</tr>
<tr>
<td>0.60</td>
<td>.157075689588E+01(Z)</td>
<td>.313724479653E-05</td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td>.164360423379E+01(X)</td>
<td>.164363156960E+01</td>
<td>.166313519292E-04</td>
</tr>
<tr>
<td>0.70</td>
<td>.164358460263E+01(Y)</td>
<td>.285751193898E-04</td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td>.16436286518E+01(Z)</td>
<td>.468743749893E-05</td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>.170027071388E+01(X)</td>
<td>.170033680417E+01</td>
<td>.388689400721E-04</td>
</tr>
<tr>
<td>0.80</td>
<td>.170024666040E+01(Y)</td>
<td>.530152399252E-04</td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>.170029476769E+01(Z)</td>
<td>.247224373693E-04</td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td>.173508681301E+01(X)</td>
<td>.173523947285E+01</td>
<td>.879658605241E-04</td>
</tr>
<tr>
<td>0.90</td>
<td>.173505848723E+01(Y)</td>
<td>.104300080438E-03</td>
<td></td>
</tr>
<tr>
<td>0.90</td>
<td>.17351157526E+01(Z)</td>
<td>.716313756380E-04</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>.174171581559E+01(X)</td>
<td>.174206349206E+01</td>
<td>.199577383509E-03</td>
</tr>
<tr>
<td>1.00</td>
<td>.174168352008E+01(Y)</td>
<td>.218116033568E-03</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>.174174811168E+01(Z)</td>
<td>.181038400430E-03</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.12
Problem 3: \( y^{(3)} + x^2 y = 1 + x + x^2 \)

Initial values: \( x_0 = 0, y_0 = 2, y_1^{(0)} = 2 \)

Exact Solution: \( y = 2 \left( 1 - \frac{x^4}{12} + \frac{x^6}{672} - \cdots \right) + 2 \left( x - \frac{x^5}{20} + \frac{x^7}{1440} - \cdots \right) \)

<table>
<thead>
<tr>
<th>x_n</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.220515731629E+01</td>
<td>.220515731629E+01</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.242116719231E+01</td>
<td>.242116688706E+01</td>
<td>.126074661035E-06</td>
</tr>
<tr>
<td></td>
<td>.242116719231E+01</td>
<td>.242116719231E+01</td>
<td>.134184059026E-06</td>
</tr>
<tr>
<td>.30</td>
<td>.264857051467E+01</td>
<td>.264856910711E+01</td>
<td>.117965271145E-06</td>
</tr>
<tr>
<td></td>
<td>.264857051467E+01</td>
<td>.264857051467E+01</td>
<td>.131441912139E-06</td>
</tr>
<tr>
<td>.40</td>
<td>.288743990707E+01</td>
<td>.288743590441E+01</td>
<td>.504270769870E-06</td>
</tr>
<tr>
<td></td>
<td>.288743990707E+01</td>
<td>.288743990707E+01</td>
<td>.558613056588E-06</td>
</tr>
<tr>
<td>.50</td>
<td>.313722608903E+01</td>
<td>.313721710689E+01</td>
<td>.138623499026E-05</td>
</tr>
<tr>
<td></td>
<td>.313722608903E+01</td>
<td>.313722608903E+01</td>
<td>.144859962533E-05</td>
</tr>
<tr>
<td>.60</td>
<td>.339659195430E+01</td>
<td>.339657430537E+01</td>
<td>.12387036964E-05</td>
</tr>
<tr>
<td></td>
<td>.339659195430E+01</td>
<td>.339659195430E+01</td>
<td>.134888259352E-05</td>
</tr>
<tr>
<td>.70</td>
<td>.366323827217E+01</td>
<td>.366320554629E+01</td>
<td>.893367162527E-05</td>
</tr>
<tr>
<td></td>
<td>.366323827217E+01</td>
<td>.366323827217E+01</td>
<td>.954520814619E-05</td>
</tr>
<tr>
<td>.80</td>
<td>.393372811772E+01</td>
<td>.393366680462E+01</td>
<td>.832213777637E-05</td>
</tr>
<tr>
<td></td>
<td>.393372811772E+01</td>
<td>.393372811772E+01</td>
<td>.155867549444E-04</td>
</tr>
<tr>
<td>.90</td>
<td>.420332107723E+01</td>
<td>.420319930665E+01</td>
<td>.217358423771E-04</td>
</tr>
<tr>
<td></td>
<td>.420332107723E+01</td>
<td>.420332107723E+01</td>
<td>.943792519809E-05</td>
</tr>
<tr>
<td>1.00</td>
<td>.446583286944E+01</td>
<td>.446557539683E+01</td>
<td>.576572081016E-04</td>
</tr>
<tr>
<td></td>
<td>.446583286944E+01</td>
<td>.446583286944E+01</td>
<td>.138287488676E-03</td>
</tr>
<tr>
<td></td>
<td>.446583286944E+01</td>
<td>.446583286944E+01</td>
<td>.229762941082E-04</td>
</tr>
</tbody>
</table>

Table 7.13
Problem 4: \( y^{(2)} - y = 0 \)

Initial values: \( x_0 = 0, \quad y_0 = 1, \quad y_0^{(2)} = -1 \)

Exact Solution: \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>( x_n )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.10</td>
<td>.904837418036E+00</td>
<td>.904837418036E+00</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>.20</td>
<td>.818723210252E+00(X)</td>
<td>.818730753078E+00(Y)</td>
<td>.921282811814E-05(Z)</td>
</tr>
<tr>
<td>.30</td>
<td>.740796234571E+00(X)</td>
<td>.740818220682E+00(Y)</td>
<td>.296781450339E-04(Z)</td>
</tr>
<tr>
<td>.40</td>
<td>.670277221236E+00(X)</td>
<td>.670320046036E+00(Y)</td>
<td>.921047419541E-04(Z)</td>
</tr>
<tr>
<td>.50</td>
<td>.606460980113E+00(X)</td>
<td>.60650659713E+00(Y)</td>
<td>.296857288372E-04(Z)</td>
</tr>
<tr>
<td>.60</td>
<td>.548709348791E+00(X)</td>
<td>.548811636094E+00(Y)</td>
<td>.921518204101E-04(Z)</td>
</tr>
<tr>
<td>.70</td>
<td>.496444810957E+00(X)</td>
<td>.49658303791E+00(Y)</td>
<td>.28291786256E-03(Z)</td>
</tr>
<tr>
<td>.80</td>
<td>.449144721232E+00(X)</td>
<td>.449328964117E+00(Y)</td>
<td>.41004008196E-03(Z)</td>
</tr>
<tr>
<td>.90</td>
<td>.406336078720E+00(X)</td>
<td>.40659659741E+00(Y)</td>
<td>.57451605226E-03(Z)</td>
</tr>
<tr>
<td>1.00</td>
<td>.367590796995E+00(X)</td>
<td>.367879441171E+00(Y)</td>
<td>.78461621991E-03(Z)</td>
</tr>
</tbody>
</table>

Table 7.14
Problem 5: \( y^{(n)} - y \left( \frac{(Q + Bx)}{x} \right) - Q / x^2 = 0 \)

Initial values: \( x_0 = 1, y_0 = 10e, y^{(0)}_0 = 10e(Q + B) \)

Exact Solution: \( y = Cx^Qe^{Bx} \)

\[
\begin{pmatrix}
\text{Set} & B = 1, C = 10, Q = \frac{3}{2}
\end{pmatrix}
\]

<table>
<thead>
<tr>
<th>( x_0 )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.10</td>
<td>.346587549802E+02</td>
<td>.346587549802E+02</td>
<td>.000000000000E+00</td>
</tr>
<tr>
<td>1.20</td>
<td>.436390152385E+02</td>
<td>.436440703713E+02</td>
<td>.115826337471E-03</td>
</tr>
<tr>
<td>1.30</td>
<td>.543715401009E+02</td>
<td>.543873445416E+02</td>
<td>.290590410221E-03</td>
</tr>
<tr>
<td>1.40</td>
<td>.671413377291E+02</td>
<td>.671744823128E+02</td>
<td>.493410333002E-03</td>
</tr>
<tr>
<td>1.50</td>
<td>.822756510828E+02</td>
<td>.82338855610E+02</td>
<td>.707296611036E-03</td>
</tr>
<tr>
<td>1.60</td>
<td>.100149815184E+03</td>
<td>.100242328229E+03</td>
<td>.922894012738E-03</td>
</tr>
<tr>
<td>1.70</td>
<td>.121193849010E+03</td>
<td>.121331621407E+03</td>
<td>.113513929292E-02</td>
</tr>
<tr>
<td>1.80</td>
<td>.145900197789E+03</td>
<td>.146096168080E+03</td>
<td>.134137871608E-02</td>
</tr>
<tr>
<td>1.90</td>
<td>.174831793908E+03</td>
<td>.175101520393E+03</td>
<td>.154040058834E-02</td>
</tr>
<tr>
<td>2.00</td>
<td>.208632138690E+03</td>
<td>.208994066965E+03</td>
<td>.173176396444E-02</td>
</tr>
</tbody>
</table>

Table 7.15
Table 7.16: Computational work of various integration formulas

<table>
<thead>
<tr>
<th>Method</th>
<th>Square Root</th>
<th>Division</th>
<th>Multiplication</th>
<th>Additions</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerov (7.2.20)</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>GM (7.2.3)</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>GM (7.2.4)</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>CoM (7.2.21)</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>CeM (7.3.10)</td>
<td>0</td>
<td>3</td>
<td>15</td>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>HaM (7.4.10)</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>10</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 7.16 illustrates the number of arithmetic operations involved in using the various formulas. The GM formulae are found to involve two times more work than the Numerov formulae. In addition, the $C,M$ and $H_s M$ methods are three times more work than the Numerov formulae while the $C_s M$ method is approximately four times.
7.5 NUMERICAL SOLUTION FOR SOLVING ODEs WITH NONLINEAR 2-STEP METHOD

A well known second order explicit formula for the integration of ODEs, i.e., \( y' = f(x, y) \) is the unstable mid-point (leap frog) method or the arithmetic mean (AM) formula with interval of length 2h of the form

\[
y_{n+1} = y_{n-1} + 2hf_n(x, y) \tag{7.5.1}
\]

We may now to replace \( 2f_n(x, y) \) by alternative means, i.e., geometric mean (GM), Contraharmonic mean (C,M), Centroidal Mean (C,M) and Harmonic Mean (H,M). In the following we will establish a stable second order implicit formula to use in a predictor corrector scheme.

7.5.1 Geometric Mean (GM) 2 - Step Method

We replace \( 2f_n(x, y) \) by \( \sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}} \) in equation (7.5.1) to produce a new formula in the form

\[
y_{n+1} = y_{n-1} + h(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}) \tag{7.5.2}
\]

Now consider the right hand side of equation (7.5.2). By the Taylor series expansion of \( y_{n+1}, y_{n-1}, f_{n+1} \) and \( f_{n-1} \) at \( x_n \) we have

\[
y_{n+1} = y_n + hy_n^{(0)} + \frac{h^2}{2!} y_n^{(2)} + \frac{h^3}{3!} y_n^{(3)} + \frac{h^4}{4!} y_n^{(4)} + \frac{h^5}{5!} y_n^{(5)} + O(h^6) \tag{7.5.3}
\]

\[
y_{n-1} = y_n - hy_n^{(0)} + \frac{h^2}{2!} y_n^{(2)} - \frac{h^3}{3!} y_n^{(3)} + \frac{h^4}{4!} y_n^{(4)} - \frac{h^5}{5!} y_n^{(5)} + O(h^6) \tag{7.5.4}
\]

\[
f_{n+1} = f_n + hf_n^{(0)} + \frac{h^2}{2!} f_n^{(2)} + \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} + \frac{h^5}{5!} f_n^{(5)} + O(h^6) \tag{7.5.5}
\]

\[
f_{n-1} = f_n - hf_n^{(0)} + \frac{h^2}{2!} f_n^{(2)} - \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} - \frac{h^5}{5!} f_n^{(5)} + O(h^6) \tag{7.5.6}
\]
By substituting equations (7.5.3)-(7.5.6) into the right hand side of equation (7.5.2) and using Mathematica for algebraic manipulation, we obtain the following results:

\[
\sqrt{f_n f_{n+1}} = f_n - \frac{h}{2} f_n^{(0)} + \frac{h^2}{8} \left( 2 f_n^{(2)} - \frac{(f_n^{(0)})^2}{f_n} \right) + \frac{h^3}{48} \left( -4 f_n^{(3)} - 3 \left( \frac{f_n^{(0)}}{f_n^2} \right)^2 + 6 \frac{f_n^{(0)} f_n^{(2)}}{f_n} \right) + O(h^4)
\]

\[
+ \frac{h^4}{384} \left( 8 f_n^{(4)} - 16 \frac{f_n^{(0)} f_n^{(3)}}{f_n} - 12 \frac{(f_n^{(0)})^2}{f_n} + 36 \left( \frac{f_n^{(0)}}{f_n^2} \right)^2 \frac{f_n^{(2)}}{f_n} - 15 \left( \frac{f_n^{(0)}}{f_n^3} \right)^4 \right) + O(h^5)
\]

\(\ldots\) (7.5.7)

\[
\sqrt{f_n f_{n+1}} = f_n + \frac{h}{2} f_n^{(0)} + \frac{h^2}{8} \left( 2 f_n^{(2)} - \frac{(f_n^{(0)})^2}{f_n} \right) + \frac{h^3}{48} \left( 4 f_n^{(3)} + 3 \left( \frac{f_n^{(0)}}{f_n^2} \right)^2 - 6 \frac{f_n^{(0)} f_n^{(2)}}{f_n} \right) + O(h^4)
\]

\[
+ \frac{h^4}{384} \left( 8 f_n^{(4)} - 16 \frac{f_n^{(0)} f_n^{(3)}}{f_n} - 12 \frac{(f_n^{(0)})^2}{f_n} + 36 \left( \frac{f_n^{(0)}}{f_n^2} \right)^2 \frac{f_n^{(2)}}{f_n} - 15 \left( \frac{f_n^{(0)}}{f_n^3} \right)^4 \right) + O(h^5)
\]

\(\ldots\) (7.5.8)

By adding equations (7.5.7) and (7.5.8), we obtain

\[
\sqrt{f_n f_{n+1}} + \sqrt{f_n f_{n+1}} = 2 f_n + \frac{h^2}{4} \left( 2 \frac{f_n^{(0)}}{f_n} - \frac{(f_n^{(0)})^2}{f_n} \right) + O(h^4)
\]

\[
+ \frac{h^4}{192} \left( 8 f_n^{(4)} - 16 \frac{f_n^{(0)} f_n^{(3)}}{f_n} - 12 \frac{(f_n^{(0)})^2}{f_n} + 36 \left( \frac{f_n^{(0)}}{f_n^2} \right)^2 \frac{f_n^{(2)}}{f_n} - 15 \left( \frac{f_n^{(0)}}{f_n^3} \right)^4 \right) + O(h^5)
\]

\(\ldots\) (7.5.9)

By substituting equations (7.5.4) and (7.5.9) into equation (7.5.2), we obtain

\[
y_{n+1} = f_n h + \frac{h^2}{2} f_n^{(0)} + \frac{h^3}{12} \left( 4 f_n^{(2)} - 3 \left( \frac{f_n^{(0)}}{f_n^2} \right)^2 \right) + \frac{h^4}{24} f_n^{(0)} + O(h^5)
\]

(7.5.10)

Therefore, by comparing (7.5.3) and (7.5.10), we can see that the formula of equation (7.5.2) is second order with its local truncation error (LTE) given by
\[
LTE = \left[ \frac{(f_n(0))^2}{4f_n} - \frac{f_n^{(2)}}{6} \right] h^3
\]

\[
= \frac{h^3}{12} \left[ 3 \left( \frac{f_n(0)^2}{f_n} - 2f_n^{(2)} \right) \right].
\]

**Stability of Formula (7.5.2)**

We apply the stability method to test equation \( y' = \lambda y \) in equation (7.5.2) to produce

\[
y_{n+1} = y_{n-1} + h\lambda \left[ \sqrt{y_n y_{n-1}} + \sqrt{y_n y_{n+1}} \right] \quad \text{(7.5.11)}
\]

By substituting \( y_n = \mu^n, \quad y_{n+1} = \mu^{n+1}, \quad y_{n-1} = \mu^{n-1} \) into equation (7.5.11) we obtain

\[
\mu^{n+1} = \mu^{n-1} + (h\lambda) \left( \sqrt{\mu^n \mu^{n-1}} + \sqrt{\mu^n \mu^{n+1}} \right)
\]

or

\[
\mu^n \left( \mu - \frac{1}{\mu} - (h\lambda) \left( \frac{1}{\sqrt{\mu}} + \sqrt{\mu} \right) \right) = 0
\]

i.e. \( \mu = 0 \) is a root with multiplicity \( n \). Now we want to determine the roots of

\[
\mu - \frac{1}{\mu} - h\lambda \left( \frac{1}{\sqrt{\mu}} + \sqrt{\mu} \right) = 0 \quad \text{(7.5.12)}
\]

We see that \( \mu = 0 \) is not root of this equation. Let \( \sqrt{\mu} = v \) and substituting into equation (7.5.12) we have

\[
v^2 - \frac{1}{v^2} - h\lambda \left( \frac{1}{v} + v \right) = 0
\]

\[
v^4 - 1 - h\lambda (v + v^3) = 0
\]

\[
v^4 - (h\lambda) v^3 - (h\lambda) v - 1 = 0 \quad \text{(7.5.13)}
\]

By solving equation (7.5.13) using Mathematica i.e.,

\[
In[1]:= \text{Solve}[v^4 - h\lambda v^3 - h\lambda v - 1 == 0, v]
\]

255
\[ \text{Out[1]} = v \leftarrow \frac{h\lambda \pm \sqrt{4 + (h\lambda)^2}}{2}, \quad v = \pm 1 \]  

(7.5.14)

From equation (7.5.14) and \( \mu = v^2 \), we have

\[ v^2_i = \left[ \frac{h\lambda + \sqrt{4 + (h\lambda)^2}}{2} \right]^2 \]

\[ = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{8} + O(h^4) \]

\[ = e^{h\lambda} + \frac{(h\lambda)^3}{8} + O(h^4) \]  

(7.5.15)

and

\[ v^2_i = \left[ \frac{h\lambda - \sqrt{4 + (h\lambda)^2}}{2} \right]^2 \]

\[ = -1 - h\lambda + \frac{(h\lambda)^2}{2} - \frac{(h\lambda)^3}{8} + O(h^4) \]

\[ = -e^{-h\lambda} - \frac{(h\lambda)^3}{8} + O(h^4) \]  

(7.5.16)

The general solution of the difference equation (7.5.12) is

\[ y_s = C_1 \mu^s_i + C_2 \mu^s_{-i} \]

\[ = C_1 (v^2_i)^s + C_2 (v^2_{-i})^s \]

\[ = C_1 e^{s h \lambda} + C_2 (-1)^s e^{-s h \lambda} + O(h^3). \]

Since, \( x_s = nh \) we now obtain

\[ y_s = C_1 e^{s x_h} + C_2 (-1)^s e^{-s x_h} \]  

(7.5.17)

To determine the stability region of the second order geometric mean (GM) formula (7.5.11) in the complex plane that satisfy the condition

\[ |\mu^s| = |(v^2)^s| < 1 \]

i.e.,

\[ |\mu^s_i| = |(v^2_i)^s| = \left| \frac{h\lambda}{2} + \sqrt{1 + \left( \frac{h\lambda}{2} \right)^2} \right|^2 < 1 \]  

(7.5.18)
and using Mathematica, we can plot the graphic surface defined by equation (7.5.18) as shown in Figure 7.1 and plot the stability region defined by equation (7.5.18) as shown in Figure 7.2.

Figure 7.1: Graphic surface defined by equation (7.5.18)

Figure 7.2: Stability region for second order GM method

**Numerical Example**

We consider the initial value problem

\[ y' = -2y, \quad y(0) = 1, \quad 0 \leq x \leq 1 \]  

(7.5.19)

which has the solution \( y(x) = \exp(-2x) \).

To use the equation (7.5.2), we must also compute an approximation to \( y_i \), which we obtain from Euler's method. It gives

\[ y_i = y_0 + h(-2y_0). \]
The error in the numerical solution using formula (7.5.2) compared with the second order Midpoint method in equation (7.5.1) and the exact solution are shown in Table 7.17.

Table 7.17: Errors in equation (7.5.1) and (7.5.2) method for solving (7.5.19).

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>exact solution</th>
<th>error (GM)</th>
<th>error (Midpoint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>.818730753078E+00</td>
<td>-1.87307530780E-01</td>
<td>-1.87307530780E-01</td>
</tr>
<tr>
<td>.2000</td>
<td>.670320046036E+00</td>
<td>-9.20548423562E-02</td>
<td>.96795396436E-02</td>
</tr>
<tr>
<td>.3000</td>
<td>.548811636094E+00</td>
<td>-2.64844229043E-01</td>
<td>-2.08116360940E-01</td>
</tr>
<tr>
<td>.4000</td>
<td>.449328964117E+00</td>
<td>-1.02073952000E-01</td>
<td>.19471035828E-01</td>
</tr>
<tr>
<td>.5000</td>
<td>.367879441171E+00</td>
<td>-2.91607044110E-01</td>
<td>.27399441114E-01</td>
</tr>
<tr>
<td>.6000</td>
<td>.301194211912E+00</td>
<td>-6.94970431920E-02</td>
<td>.31413788078E-01</td>
</tr>
<tr>
<td>.7000</td>
<td>.246596963942E+00</td>
<td>-2.98670140055E-01</td>
<td>.39160163941E-01</td>
</tr>
<tr>
<td>.8000</td>
<td>.201896517995E+00</td>
<td>-1.50407820120E-02</td>
<td>.47736762005E-01</td>
</tr>
<tr>
<td>.9000</td>
<td>.165298888222E+00</td>
<td>-3.03276582224E-01</td>
<td>.57715400221E-01</td>
</tr>
<tr>
<td>1.0000</td>
<td>.135335283237E+00</td>
<td>-.517083608217E-01</td>
<td>.71264601563E-01</td>
</tr>
</tbody>
</table>

7.5.2 Contraharmonic Mean ($C_M$) 2-Step Method

Now, we attempt to replace $2f_n(x,y)$ by

$$\left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right)$$

in equation (7.5.1) to obtain a new formula defined by

$$y_{n+1} = y_{n-1} + h \left[ \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right]$$

By substituting equations (7.5.3)-(7.5.6) into the right hand side of equation (7.5.20) and using Mathematica for algebraic manipulation, we obtain the following results:

$$\left( \frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} \right) = f_n - \frac{h}{2} f_n^{(1)} + \frac{h^2}{4} \left( f_n^{(2)} + \frac{f_n^{(0)} f_n^{(2)}}{f_n} \right) + \frac{h^3}{24} \left( 3 \left( f_n^{(0)} \right)^3 - 6 f_n^{(0)} f_n^{(2)} f_n^{(3)} - 2 f_n^{(3)} \right)$$

$$+ \frac{h^4}{48 f_n^3} \left( 3 \left( f_n^{(0)} \right)^4 - 9 f_n^{(0)} f_n^{(2)} f_n^{(3)} + 3 f_n^{(2)} f_n^{(3)} + 4 f_n^{(3)} f_n^{(3)} + f_n^{(3)} f_n^{(4)} \right) + O(h^5) \right)$$

(7.5.21)
By adding equations (7.5.21) and (7.5.22), we have

\[
\left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right) = f_n + \frac{h}{2} f_n^{(0)} + \frac{h^2}{4} \left(f_n^{(2)} + \left(\frac{f_n^{(0)}}{f_n}\right)^2\right) + \frac{h^3}{24} \left(-3 \left(\frac{f_n^{(0)}}{f_n^2}\right) + 6 \frac{f_n^{(0)} f_n^{(2)}}{f_n} + 2 f_n^{(3)}\right)
\]

\[+ \frac{h^4}{48 f_n} \left(3 \left(\frac{f_n^{(0)}}{f_n}\right)^4 - 9 f_n \left(\frac{f_n^{(0)}}{f_n}\right)^2 f_n^{(2)} + 3 f_n^2 \left(\frac{f_n^{(0)}}{f_n}\right)^2 + 4 f_n^2 f_n^{(0)} f_n^{(3)} + f_n^3 f_n^{(4)}\right) + O(h^5)
\]

... (7.5.22)

and substituting equations (7.5.4) and (7.5.23) into equation (7.5.20), we obtain

\[
y_{n+1} = f_n h + \frac{h^2}{2} f_n^{(3)} + \frac{h^3}{6} \left(2 f_n^{(2)} + 3 \left(\frac{f_n^{(0)}}{f_n}\right)^2\right) + \frac{h^4}{24} f_n^{(3)} + O(h^5)
\]

... (7.5.23)

By comparing (7.5.3) and (7.5.24), we can see that the formula of equation (7.5.20) is second order with its local truncation error (LTE) given by

\[
LTE = -\frac{f_n^{(0)^2}}{2 f_n} - \frac{f_n^{(2)}}{6} h^2 + O(h^4)
\]

\[
= \frac{h^2}{6} \left[ -3 \left(\frac{f_n^{(0)}}{f_n}\right)^2 - f_n^{(2)} \right] + O(h^4).
\]

Stability of Formula (7.5.20)

By applying the stability method to test equation \(y' = \lambda y\) in equation (7.5.20), we obtain
By substituting $y_n = \mu^n$, $y_{n+1} = \mu^{n+1}$, $y_{n-1} = \mu^{n-1}$ into equation (7.5.25) we obtain

$$y_{n+1} = y_{n-1} + h\lambda \left[ \frac{y_n^2 + y_{n+1}^2}{y_n + y_{n+1}} + \left( \frac{y_n^2 + y_{n+1}^2}{y_n + y_{n+1}} \right) \right]$$

(7.5.25)

By solving equation (7.5.26) using Mathematica i.e.,

$$\text{In}[2] := \text{Solve}[\mu^n + (1 - h\lambda)\mu^3 + (1 - h\lambda)\mu^2 - (1 + h\lambda)\mu + (1 + h\lambda) = 0, \mu]$$

$$\text{Out}[2] := \mu > -1, \quad \mu > \pm \sqrt{\left(\frac{2}{1 - h\lambda}\right) - 1}$$

(7.5.27)

To determine the stability region of the second order implicit contraharmonic mean (C,M) formula (7.5.25) in the complex plane that satisfy the condition $|\mu^n| < 1$

i.e.,

$$|\mu^n| = \sqrt{\left(\frac{2}{1 - h\lambda}\right) - 1} < 1$$

(7.5.28)

and using Mathematica, we can plot the graphic surface defined by equation (7.5.28) as shown in Figure 7.3 and plot the stability region defined by equation (7.5.28) as shown in Figure 7.4.
Figure 7.3: Graphic surface defined by equation (7.5.28)

Figure 7.4: Stability region for second order implicit $CM$ method

**Numerical Example of formula (7.5.20)**

The error in the numerical solution using formulae (7.5.20) and (7.5.1) for solving problem (7.5.19) are shown in Table 7.18.

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1000</td>
<td>0.8000000000000E+00</td>
<td>0.818730753078E+00</td>
<td>-0.187307530780E-01</td>
</tr>
<tr>
<td>0.2000</td>
<td>0.657777777778E+00</td>
<td>0.670320046036E+00</td>
<td>-0.125422682579E-01</td>
</tr>
<tr>
<td>0.3000</td>
<td>0.521279132791E+00</td>
<td>0.548811636094E+00</td>
<td>-0.275325033027E-01</td>
</tr>
<tr>
<td>0.4000</td>
<td>0.434036024305E+00</td>
<td>0.449328964117E+00</td>
<td>-0.152929398123E-01</td>
</tr>
<tr>
<td>0.5000</td>
<td>0.338143674105E+00</td>
<td>0.367879441171E+00</td>
<td>-0.297357670661E-01</td>
</tr>
<tr>
<td>0.6000</td>
<td>0.287998490248E+00</td>
<td>0.301194211912E+00</td>
<td>-0.131957216638E-01</td>
</tr>
<tr>
<td>0.7000</td>
<td>0.217528167201E+00</td>
<td>0.246596963942E+00</td>
<td>-0.290687967403E-01</td>
</tr>
<tr>
<td>0.8000</td>
<td>0.192957836056E+00</td>
<td>0.201896517995E+00</td>
<td>-0.893868193837E-02</td>
</tr>
<tr>
<td>0.9000</td>
<td>0.137740929809E+00</td>
<td>0.165298888222E+00</td>
<td>-0.275579584129E-01</td>
</tr>
<tr>
<td>1.0000</td>
<td>0.131417814599E+00</td>
<td>0.135335283237E+00</td>
<td>-0.391746863806E-02</td>
</tr>
</tbody>
</table>
7.5.3 Centroidal Mean \((C,M)\) 2-Step Method

By replacing \(2f_n(x,y)\) with \[
\left[ \frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})} + \frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})} \right]
\]
in equation (7.5.1) we obtain a new implicit method as

\[
y_{n+1} = y_{n-1} + h \left[ \left( \frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})} \right) + \left( \frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})} \right) \right] \tag{7.5.29}
\]

By substituting equations (7.5.4)-(7.5.6) into the right hand side of equation (7.5.29) and using Mathematica for algebraic manipulation, we obtain the following results:

\[
\frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})} = f_n - \frac{h}{2} f_n^{(0)} + \frac{h^2}{12} \left[ 3f_n^{(2)} + \left( \frac{f_n^{(0)}}{f_n} \right)^2 \right] + \frac{h^3}{24} \left( \frac{(f_n^{(0)})^3}{f_n^2} - 2 \frac{f_n^{(0)} f_n^{(2)}}{f_n} - 2f_n^{(0)} \right)
\]

\[
+ \frac{h^4}{144 f_n^3} \left( 3(f_n^{(0)})^4 - 9f_n(f_n^{(0)})^2 f_n^{(2)} + 3 f_n f_n^{(2)} (f_n^{(0)})^2 + 4 f_n^2 f_n^{(0)} f_n^{(3)} + 3 f_n^2 f_n^{(4)} \right) + 0(h^5)
\]

\[
\ldots \tag{7.5.30}
\]

\[
\frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})} = f_n + \frac{h}{2} f_n^{(0)} + \frac{h^2}{12} \left[ 3f_n^{(2)} + \left( \frac{f_n^{(0)}}{f_n} \right)^2 \right] + \frac{h^3}{24} \left( \frac{(f_n^{(0)})^3}{f_n^2} + 2 \frac{f_n^{(0)} f_n^{(2)}}{f_n} + 2f_n^{(0)} \right)
\]

\[
+ \frac{h^4}{144 f_n^3} \left( 3(f_n^{(0)})^4 - 9f_n(f_n^{(0)})^2 f_n^{(2)} + 3 f_n f_n^{(2)} (f_n^{(0)})^2 + 4 f_n^2 f_n^{(0)} f_n^{(3)} + 3 f_n^2 f_n^{(4)} \right) + 0(h^5)
\]

\[
\ldots \tag{7.5.31}
\]

By substituting equations (7.5.4) and (7.5.30)-(7.5.31) into equation (7.5.29), we have

\[
y_{n+1} = f_n h + \frac{h^2}{2} f_n^{(0)} + \frac{h^3}{6} \left[ 2f_n^{(2)} + \left( \frac{f_n^{(0)}}{f_n} \right)^2 \right] + \frac{h^4}{24} f_n^{(3)} + 0(h^5) \tag{7.5.32}
\]

By comparing (7.5.3) and (7.5.32), we conclude that the formula of equation (7.5.29) is second order with its local truncation error (LTE) given by
\[ \text{LTE} = \left[ -\frac{(f_n^{(0)})^2}{6f_n} - \frac{f_n^{(2)}}{6} \right] h^3 + 0(h^4) \]

\[ = -\frac{h^3}{6} \left[ \frac{(f_n^{(0)})^2}{f_n} + f_n^{(2)} \right] + 0(h^4) \]

**Stability of Formula (7.5.29)**

By substituting the test equation \( y' = \lambda y \) into equation (7.5.29), we have

\[ y_{n+1} = y_{n-1} + h\lambda \left[ \left( \frac{2(y_n^2 + y_n y_{n-1} + y_{n-1}^2)}{3(y_n + y_{n-1})} \right) + \left( \frac{2(y_n^2 + y_n y_{n+1} + y_{n+1}^2)}{3(y_n + y_{n+1})} \right) \right] \]

\[ = y_{n-1} + \frac{2}{3} h\lambda \left[ \left( \frac{(y_n^2 + y_n y_{n-1} + y_{n-1}^2)}{y_n + y_{n-1}} \right) + \left( \frac{(y_n^2 + y_n y_{n+1} + y_{n+1}^2)}{y_n + y_{n+1}} \right) \right] \quad (7.5.33) \]

By substituting \( y_n = \mu^n, y_{n+1} = \mu^{n+1}, y_{n-1} = \mu^{n-1} \) into equation (7.5.33) we obtain

\[ \mu^{n+1} = \mu^{n-1} + \frac{2}{3} h\lambda \left[ \left( \frac{\mu^{2n} + \mu^n \mu^{n-1} + \mu^{n-2}}{\mu^n + \mu^{n-1}} \right) + \left( \frac{\mu^{2n} + \mu^n \mu^{n+1} + \mu^{n+2}}{\mu^n + \mu^{n+1}} \right) \right] \]

and divided by \( \mu^n \) we obtain

\[ \mu = \frac{1}{\mu} + \frac{2}{3} h\lambda \left[ \left( \frac{1 + \mu^{-1} + \mu^2}{1 + \mu^{-1}} \right) + \left( \frac{1 + \mu^2}{1 + \mu} \right) \right] \]

where

\[ \frac{1 + \frac{1}{\mu} + \frac{1}{\mu^2}}{1 + \frac{1}{\mu}} = \frac{1 + \mu + \mu^2}{\mu(1 + \mu)} \]

Therefore,

\[ \mu = \frac{1}{\mu} + \frac{2}{3} h\lambda \left[ \left( \frac{1 + \mu + \mu^2}{\mu(1 + \mu)} \right) + \left( \frac{\mu + \mu^2 + \mu^3}{\mu(1 + \mu)} \right) \right] \]

or

\[ \left( 1 - \frac{2}{3} h\lambda \right) \mu^3 + \left( 1 - \frac{4}{3} h\lambda \right) \mu^2 - \left( 1 + \frac{4}{3} h\lambda \right) \mu - \left( 1 + \frac{2}{3} h\lambda \right) = 0 \quad (7.5.34) \]

263
By solving equation (7.5.34) using Mathematica, i.e.,

\[ In[3]:= \text{Solve}\left[\left(1-\frac{2}{3}h\lambda\right)\mu^3 + \left(1-\frac{4}{3}h\lambda\right)\mu^2 - \left(1+\frac{4}{3}h\lambda\right)\mu - \left(1+\frac{2}{3}h\lambda\right) = 0, \mu\right] \]

\[ Out[3]:= -1 + \sqrt{-\frac{1}{2} + \frac{3 + 2\lambda}{2}} \]

(7.5.35)

To determine the stability region of the second order implicit centroidal mean (C,M) formula (7.5.35) in the complex plane that satisfy the condition

\[ |\mu^n| < 1 \]

i.e.,

\[ |\mu^n| = \left|\frac{-1 + \frac{3}{3 - 2h\lambda} \pm \frac{2\sqrt{3\lambda - (h\lambda)^2}}{3 - 2h\lambda}}{2}\right| < 1 \]

(7.5.36)

and using Mathematica, we can plot the graphic surface defined by equation (7.5.36) as shown in Figure 7.5 and plot the stability region defined by equation (7.5.36) as shown in Figure 7.6.

![Figure 7.5: Graphic surface defined by equation (7.5.36)](image-url)
Figure 7.6: Stability region for second order implicit $C,M$ method

Numerical Example of formula (7.5.29)

The error in the numerical solution using formulae (7.5.29) for solving problem (7.5.19) are shown in Table 7.19.

Table 7.19: Error by using the $C,M$ second order implicit formulae for solving (7.5.19).

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>.800000000000E+00</td>
<td>.818730753078E+00</td>
<td>-.187307530780E-01</td>
</tr>
<tr>
<td>.2000</td>
<td>.659259259259E+00</td>
<td>.670320046036E+00</td>
<td>-.11067867764E-01</td>
</tr>
<tr>
<td>.3000</td>
<td>.521769756220E+00</td>
<td>.548811636094E+00</td>
<td>-.270418798742E-01</td>
</tr>
<tr>
<td>.4000</td>
<td>.436268878411E+00</td>
<td>.449328964117E+00</td>
<td>-.13060857060E-01</td>
</tr>
<tr>
<td>.5000</td>
<td>.338457764072E+00</td>
<td>.367879441171E+00</td>
<td>-.294216770998E-01</td>
</tr>
<tr>
<td>.6000</td>
<td>.290693031682E+00</td>
<td>.301194211912E+00</td>
<td>-.105011820303E-01</td>
</tr>
<tr>
<td>.7000</td>
<td>.217283202546E+00</td>
<td>.246596963942E+00</td>
<td>-.293137613956E-01</td>
</tr>
<tr>
<td>.8000</td>
<td>.196085142085E+00</td>
<td>.201896517995E+00</td>
<td>-.581137591002E-02</td>
</tr>
<tr>
<td>.9000</td>
<td>.136693104206E+00</td>
<td>.165298888222E+00</td>
<td>-.286057840159E-01</td>
</tr>
<tr>
<td>1.0000</td>
<td>.135115366812E+00</td>
<td>.13535283237E+00</td>
<td>-.219916424883E-03</td>
</tr>
</tbody>
</table>

7.5.4 Harmonic Mean ($H,M$) 2 - Step Method

Finally, we attempt to replace $2f_n(x,y)$ by

$$\left[ \frac{2f_n f_{n-1}}{f_n + f_{n-1}} + \frac{2f_{n+1} f_{n+2}}{f_{n+1} + f_{n+2}} \right]$$

in equation (7.5.1) to obtain a new implicit method as
\[ y_{n+1} = y_{n-1} + h \left[ \frac{2f_n f_{n+1}}{f_n + f_{n+1}} + \frac{2f_n f_{n+1}}{f_n + f_{n+1}} \right] \tag{7.5.37} \]

By using the same procedure previously, we substitute equations (7.5.4)-(7.5.6) into the right hand side of equation (7.5.37) and using Mathematica for algebraic manipulation, we obtain the following results:

\[ \frac{2f_n f_{n+1}}{f_n + f_{n+1}} = f_n - \frac{h}{2} f_n^{(i)} + \frac{h^2}{4} \left( f_n^{(i)} - \frac{(f_n^{(i)})^2}{f_n} \right) + \frac{h^3}{24} \left( \frac{(f_n^{(i)})^3}{f_n^2} - \frac{3(f_n^{(i)})^2}{f_n} + 6 \frac{f_n^{(i)} f_n^{(2)}}{f_n} - 2 f_n^{(3)} \right) + \frac{h^4}{48 f_n^3} \left( -3(f_n^{(i)})^4 + 9 f_n (f_n^{(i)})^2 f_n^{(2)} - 3 f_n^2 (f_n^{(2)})^2 - 4 f_n^2 f_n^{(i)} f_n^{(3)} + f_n^3 f_n^{(4)} \right) + O(h^5) \tag{7.5.38} \]

\[ \frac{2f_n f_{n+1}}{f_n + f_{n+1}} = f_n + \frac{h}{2} f_n^{(i)} + \frac{h^2}{4} \left( f_n^{(i)} - \frac{(f_n^{(i)})^2}{f_n} \right) + \frac{h^3}{24} \left( \frac{(f_n^{(i)})^3}{f_n^2} - \frac{3(f_n^{(i)})^2}{f_n} + 6 \frac{f_n^{(i)} f_n^{(2)}}{f_n} + 2 f_n^{(3)} \right) + \frac{h^4}{48 f_n^3} \left( -3(f_n^{(i)})^4 + 9 f_n (f_n^{(i)})^2 f_n^{(2)} - 3 f_n^2 (f_n^{(2)})^2 - 4 f_n^2 f_n^{(i)} f_n^{(3)} + f_n^3 f_n^{(4)} \right) + O(h^5) \tag{7.5.39} \]

By substituting equations (7.5.4) and (7.5.38)-(7.5.39) into equation (7.5.37), we have

\[ y_{n+1} = f_n h + \frac{h^2}{2} f_n^{(i)} + \frac{h^3}{6} \left( 2 f_n^{(2)} - 3 \frac{(f_n^{(i)})^2}{f_n} \right) + \frac{h^4}{24} f_n^{(4)} + O(h^5) \tag{7.5.40} \]

By comparing (7.5.3) and (7.5.40), we conclude that the formula of equation (7.5.37) is second order with its local truncation error (LTE) given by

\[ LTE = \left[ \frac{(f_n^{(i)})^2}{2 f_n} - \frac{f_n^{(2)}}{6} \right] h^3 + O(h^4) \]

\[ = \frac{h^3}{6} \left[ 3 \frac{(f_n^{(i)})^2}{f_n} - f_n^{(2)} \right] + O(h^4) \tag{7.5.41} \]
Stability of Formula (7.5.37)

By applying the stability method to the test equation $y' = \lambda y$ in equation (7.5.37), we obtain

$$y_{n+1} = y_n + h \lambda \left[ \left( \frac{2y_n y_{n-1}}{y_n + y_{n-1}} \right) + \left( \frac{2y_n y_{n+1}}{y_n + y_{n+1}} \right) \right]$$

(7.5.42)

By substituting $y_n = \mu^n, y_{n+1} = \mu^{n+1}, y_{n-1} = \mu^{n-1}$ into equation (7.5.42) we obtain

$$\mu^{n+1} = \mu^n + 2h \lambda \left[ \frac{\mu^n \mu^{n-1}}{\mu^n + \mu^{n-1}} + \frac{\mu^n \mu^{n+1}}{\mu^n + \mu^{n+1}} \right]$$

(7.5.43)

where

$$\frac{\mu^n}{1 + \mu^n} = \frac{1}{1 + \mu}.$$

Therefore, equation (7.5.43) can be written as

$$\mu^2 - 2h \lambda \mu - 1 = 0$$

and solving using Mathematica, i.e.,

$$In[4] := \text{Solve}[\mu^2 - 2h \lambda \mu - 1 == 0, \mu]$$

$$Out[4] := \mu \rightarrow \frac{2h \lambda \pm 2\sqrt{1 + (h \lambda)^2}}{2} = h \lambda \pm \sqrt{1 + (h \lambda)^2}$$

(7.5.44)

To determine the stability region of the second order implicit Harmonic mean ($H.M$) formula (7.5.44) in the complex plane that satisfy the condition

$$|\mu^n| < 1$$

i.e.,

$$|\mu^n| = |h \lambda + \sqrt{1 + (h \lambda)^2}| < 1$$

(7.5.45)

and using Mathematica, we can plot the graphic surface defined by equation (7.5.45) as shown in Figure 7.7 and plot the stability region defined by equation (7.5.45) as shown in Figure 7.8.
Numerical Example of formula (7.5.37)

The error in the numerical solution using formulae (7.5.37) for solving problem (7.5.19) are shown in Table 7.20.
Table 7.20: Error by using the $H_M$ second order implicit formulae for solving (7.5.19).

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1000</td>
<td>.8000000000000E+00</td>
<td>.818730753078E+00</td>
<td>-.187307530780E-01</td>
</tr>
<tr>
<td>.2000</td>
<td>.6622222222222E+00</td>
<td>.670320046036E+00</td>
<td>-.809782381342E-02</td>
</tr>
<tr>
<td>.3000</td>
<td>.522631543398E+00</td>
<td>.548811636094E+00</td>
<td>-.26180926965E-01</td>
</tr>
<tr>
<td>.4000</td>
<td>.440855090780E+00</td>
<td>.449328964117E+00</td>
<td>-.847387333750E-02</td>
</tr>
<tr>
<td>.5000</td>
<td>.338805943918E+00</td>
<td>.367879441171E+00</td>
<td>-.290734972539E-01</td>
</tr>
<tr>
<td>.6000</td>
<td>.296463510831E+00</td>
<td>.301194211912E+00</td>
<td>-.473070108102E-02</td>
</tr>
<tr>
<td>.7000</td>
<td>.216268520083E+00</td>
<td>.246596963942E+00</td>
<td>-.303284438582E-01</td>
</tr>
<tr>
<td>.8000</td>
<td>.203190911266E+00</td>
<td>.201896517995E+00</td>
<td>.129439327177E-02</td>
</tr>
<tr>
<td>.9000</td>
<td>.133725167137E+00</td>
<td>.165298888222E+00</td>
<td>-.315737210847E-01</td>
</tr>
<tr>
<td>1.0000</td>
<td>.144186522692E+00</td>
<td>.135335283237E+00</td>
<td>.885123945553E-02</td>
</tr>
</tbody>
</table>

From the above results and discussions, it was shown that the geometric mean ($GM$), contraharmonic mean ($C_M$), centroidal mean ($C_M$) and harmonic mean ($H_M$) are formulae second order implicit methods. We also show that, all these new methods are unstable and in the same class as the second order explicit Midpoint method in equation (7.5.1).
CHAPTER 8

A NEW FIFTH ORDER WEIGHTED RUNGE-KUTTA FORMULA
In Butcher [1987], a general outline about the attainable orders for explicit Runge-Kutta methods was given. It was proved that there does not exist five stage linear methods of order five (see Lambert [1973], pp 122). In Butcher's findings the orders of explicit linear methods are shown in the Table 8.1.

Table 8.1

<table>
<thead>
<tr>
<th>Number of Stages</th>
<th>Highest Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>R</td>
<td>R = 2</td>
</tr>
<tr>
<td>R</td>
<td>R = 10, 11</td>
</tr>
</tbody>
</table>

In this chapter, a new five stage explicit fifth order linear and non-linear Runge-Kutta method is developed based on the arithmetic mean (AM) and contraharmonic mean (C,M) formulation in the functional values. Evans and Yaakub [1993], [1994] and [1995] show that the existence of non-linear Runge-Kutta methods based on various means formulation such as contraharmonic mean (C,M), centroidal mean (C,M) and root mean square (RMS) was revealed compared to the arithmetic mean which is more usually employed.

From recent publications see Evans & Yaakub [1993] and Evans & Sanugi [1987], [1993], fourth order linear and non-linear methods using a variety of means for solving initial value problems of the form \( y' = f(x, y) \) are shown to have the form

\[
y_{n+1} = y_n + \frac{h}{3} \left[ \sum_{i=1}^{3} \text{Means} \right]
\]

(8.0.1)
where Means = principal means include arithmetic mean (AM), geometric mean (GM) [1987], contraharmonic mean (C,M) [1993], centroidal mean (C,M) [1993], root mean square (RMS) [1993], harmonic mean (H,M) [1993] and heronian mean (H,M) [1993] which involve $k_i, 1 < i < 4$ where


k_i = f(x_i, y_i)
k_2 = f(x_i + a_i h, y_i + a_i h k_i)
k_3 = f(x_i + (a_i + a_2) h, y_i + a_i h k_i + a_2 h k_2)
k_4 = f(x_i + (a_i + a_2 + a_3) h, y_i + a_i h k_i + a_2 h k_2 + a_3 h k_3) \quad (8.0.2)

From the above discussion, a comparison of the parameters $a_i, 1 \leq i \leq 6$ in equation (8.0.2), show that for the parameters, $a_i$ is fixed, $a_i, a_2, a_3$ are decreasing and $a_i, a_2$ are increasing are shown in Table 8.2.

Table 8.2: The values of parameters $a_i, 1 \leq i \leq 6$ based on the various formulas

<table>
<thead>
<tr>
<th>$a_i$</th>
<th>$C_i M$</th>
<th>$C_i M$</th>
<th>RMS</th>
<th>AM</th>
<th>GM</th>
<th>H,M</th>
<th>H,M</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
</tr>
<tr>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
<td>$-\frac{1}{4}$</td>
</tr>
<tr>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
<td>$\frac{1}{4}$</td>
</tr>
</tbody>
</table>

From Table 8.2, we can see that $a_i + a_2 = \frac{1}{2}$ and $a_i + a_2 + a_3 = 1$. In the following discussion, our concern is to establish a new fifth order formula based on the last two $k_i$ values i.e., $k_i$ and $k_4$ which involve the parameters $a_i + a_2 = \frac{1}{2}$ and $a_i + a_2 + a_3 = 1$ and are weighted so we can establish a new weighted Runge-Kutta formula (WRK) using both the arithmetic mean (AM) and contraharmonic mean (C,M).
8.1 THE FOURTH ORDER ARITHMETIC MEAN WEIGHTED RUNGE-KUTTA FORMULA

The standard fourth order arithmetic mean (AM) Runge-Kutta formula for solving IVPs may be written in the form

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \frac{k_1 + k_{11}}{2} \right] \tag{8.1.1} \]

or

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \frac{k_1 + k_2 + k_2 + k_3 + k_3 + k_{11}}{2} \right] \]

where \( k_i, 1 \leq i \leq 4 \) as we mentioned above.

Now, based on the parameters involved in the last two \( k \) values i.e., \( k_2 \) and \( k_3 \), we attempt to develop a new fourth order method called the weighted Runge-Kutta (WRK) formula in the form

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \sum_{i=1}^{3} w_i \left( \frac{k_i + k_{11}}{2} \right) \right] \tag{8.1.2} \]

where \( \sum_{i=1}^{3} w_i = 1 \), \( k_i = f(y_n) \)

\[ k_2 = f(y_n + a_1hk_i) \]
\[ k_3 = f(y_n + a_1hk_i + (\frac{1}{2} - a_1)hk_j) \]
\[ k_4 = f(y_n + a_1hk_i + a_1hk_j + (1 - a_1 - a_2)hk_j). \tag{8.1.3} \]

By use of the standard procedure of adjustment of the parameters, a fourth order accuracy is obtained for the 7 equations involving 7 variables, i.e.,

\[ hf: \quad 1. - x(1) - x(2) - x(3) = 0, \tag{8.1.4-i} \]
\[ h^2 ff_y: \quad 2. - 2. x(4) x(1) - x(2) - 2. x(4) x(2) - 3. x(3) = 0, \tag{8.1.4-ii} \]
\[ h^3 ff_y: \quad 2. - 3. x(4) x(2) + 6. x(4) x(5) x(2) - 3. x(3)
- 3. x(4) x(3) + 6. x(4) x(5) x(3) + 3. x(6) x(3)
+ 3. x(7) x(3) - 6. x(4) x(7) x(3) = 0. \tag{8.1.4-iii} \]
\[ h^3 f^2 f_n : \quad 8. - 12.*x(4)**2*x(1) - 3.*x(2) - 12.*x(4)**2*x(2) \]
\[ \quad - 15.*x(3) = 0 , \quad (8.1.4-iv) \]

\[ h^4 f^2 f_n^3 : \quad 1. - 6.*x(4)*x(3) + 12.*x(4)*x(5)*x(3) - \]
\[ \quad 12.*x(4)*x(5)*x(6)*x(3) + 6.*x(4)*x(7)*x(3) - 12.*x(4)*x(5)*x(7)*x(3) \]
\[ \quad + 6.*x(4)*x(6)*x(3) = 0 , \quad (8.1.4-v) \]

\[ h^4 f^2 f_n f_n' : \quad 8. - 6.*x(4)*x(2) - 6.*x(4)**2*x(2) + \]
\[ \quad 12.*x(4)*x(5)*x(2) + 12.*x(4)**2*x(5)*x(2) - 15.*x(3) - \]
\[ \quad 6.*x(4)*x(3) - 6.*x(4)**2*x(3) + 12.*x(4)*x(5)*x(3) + \]
\[ \quad 12.*x(4)**2*x(5)*x(3) + 15.*x(6)*x(3) + 15.*x(7)*x(3) - \]
\[ \quad 24.*x(4)*x(7)*x(3) - 12.*x(4)**2*x(7)*x(3) = 0 , \quad (8.1.4-vi) \]

\[ h^4 f^3 f_m : \quad 4. - 8.*x(4)**3*x(1) - x(2) - 9.*x(3) - 8.*x(4)**3*x(2) = 0 \quad (8.1.4-vii) \]

where \( x(1) = w_1 , x(2) = w_2 , x(3) = w_3 , x(4) = a_1 , x(5) = a_2 \), and
\[ x(6) = a_3 , \quad x(7) = a_4 \, . \]

Equations \((8.1.4-i)-(8.1.4-vii)\) are then solved simultaneously using the NAG routine (Subroutine C05NBF) for solving a system of non-linear equations to give the required parameters, i.e.,

\[ x(1) = w_1 = 0.3333333333 , \quad x(2) = w_2 = 0.3333333333 , \quad x(3) = w_3 = 0.3333333333 \]
\[ x(4) = a_1 = 0.5000000000 , \quad x(5) = a_2 = 0 , \quad x(6) = a_3 = 0 , \quad x(7) = a_4 = 0 \, . \]

Thus, this new WRK method gives the same result as the standard fourth order arithmetic mean (AM) Runge-Kutta method

\[ y_{n+1} = y_n + \frac{h}{3} \left[ k_1 + k_2 + \frac{k_3 + k_4}{2} \right] \]

where \( k_1 = f(y_n) \)
\[ k_2 = f\left(y_n + \frac{1}{2} hk_1\right) \]
\[ k_3 = f\left(y_n + \frac{1}{2} hk_2\right) \]
\[ k_4 = f\left(y_n + hk_3\right) \, . \]
8.2 A NEW FIFTH ORDER ARITHMETIC MEAN WEIGHTED RUNGE-KUTTA FORMULA

We now extend the same procedure as used in the fourth order (AM) method to obtain the fifth order formula in the form

\[ y_{n+1} = y_n + h \left[ \frac{\sum_{i=1}^{s} w_i (k_i + k_{i+1})}{2} \right] \]  

(8.2.1)

where \( \sum_{i=1}^{s} w_i = 1 \) and

\[ k_i = f(y_n) \]

\[ k_2 = f(y_n + a_1 h_k) \]

\[ k_3 = f(y_n + a_1 h_k + a_2 h_k) \]

\[ k_4 = f(y_n + a_1 h_k + a_2 h_k + (1 - a_1 - a_2) h k_3) \]

\[ k_5 = f(y_n + a_1 h_k + a_2 h_k + a_3 h k_2 + a_4 h k_4 + (1 - a_3 - a_4) h k_5) \]  

(8.2.2)

The Taylor series expansion of \( y(x_{n+1}) \) up to sixth order is given by

\[ y(x_n + h) = y_n + hf + \frac{1}{2} h^2 f_n + \frac{1}{6} h^3 (f_{n}^2 + f_{n} f_{n+1}) \]

\[ + \frac{1}{24} h^4 (f_{n}^3 f_{n+1} + 4 f_{n}^2 f_{n+1} + f_{n} f_{n+1}) \]

\[ + \frac{1}{120} h^5 (f_{n}^4 + 11 f_{n}^2 f_{n+1} + 4 f_{n} f_{n+1} + 7 f_{n} f_{n+1} + f_{n+1} f_{n+2}) \]

\[ + \frac{1}{720} h^6 (f_{n}^5 f_{n+1} + 11 f_{n}^4 f_{n+1} + 15 f_{n}^3 f_{n+1} + 32 f_{n}^3 f_{n+1} + 34 f_{n}^2 f_{n+1} + 26 f_{n} f_{n+1} + f_{n+1}) + O(h^7) \]  

(8.2.3)

By substituting equation (8.2.2) into (8.2.1) and subtract from equation (8.2.3), we obtain 12 equations with 12 parameters, i.e.,

\[ hf: \quad 1 - x(1) - x(2) - x(3) - x(4) = 0, \]  

(8.2.4-i)

\[ h^2 f_n: \quad 2 - 2 x(5) * x(1) - 2 x(5) * x(2) - 2 x(6) * x(2) - 2 x(7) * x(2) - x(3) - 3 x(6) * x(3) - 3 x(7) * x(3) - 3 x(4) = 0, \]  

(8.2.4-ii)

\[ h^3 f_n: \quad 2 - 6 x(5) * x(7) * x(2) - 3 x(6) * x(3) - 3 x(7) * x(3) - 6 x(5) * x(7) * x(3) + 6 x(6) * x(8) * x(3) + 6 x(7) * x(8) * x(3) - 6 x(5) * x(9) * x(3) + 6 x(6) * x(9) * x(3) + 6 x(6) * x(7) * x(3) - 3 x(4) - 3 x(6) * x(4) - 3 x(7) * x(4) + \]

275
\[ 6 \times x(6) \times x(8) \times x(4) + 6 \times x(7) \times x(8) \times x(4) - 6 \times x(5) \times x(9) \times x(4) + x(4) + 6 \times x(6) \times x(9) \times x(4) + 6 \times x(7) \times x(9) \times x(4) + 3 \times x(10) \times x(4) + 3 \times x(11) \times x(4) - 6 \times x(5) \times x(11) \times x(4) + 3 \times x(12) \times x(4) - 6 \times x(6) \times x(12) \times x(4) - 6 \times x(7) \times x(12) \times x(4) = 0 , \] 

(8.2.4-iii)

\[ h^2 f^2 f_{w} : 8 - 12 \times x(5) \times x(7) \times x(2) - 12 \times x(6) \times x(7) \times x(2) - 3 \times x(3) - 12 \times x(6) \times x(7) \times x(3) - 12 \times x(7) \times x(3) - 15 \times x(4) = 0 , \] 

(8.2.4-iv)

\[ h^4 f^2 f : 1 - 6 \times x(5) \times x(7) \times x(3) + 12 \times x(5) \times x(7) \times x(8) \times x(3) + 12 \times x(5) \times x(7) \times x(9) \times x(3) - 12 \times x(6) \times x(8) \times x(3) - 12 \times x(7) \times x(8) \times x(3) - 12 \times x(5) \times x(9) \times x(3) + 12 \times x(6) \times x(9) \times x(3) - 12 \times x(7) \times x(9) \times x(3) + 6 \times x(5) \times x(10) \times x(3) - 6 \times x(6) \times x(10) \times x(3) - 6 \times x(7) \times x(10) \times x(3) - 6 \times x(8) \times x(10) \times x(3) - 6 \times x(9) \times x(10) \times x(3) + 6 \times x(10) \times x(10) \times x(3) = 0 , \] 

(8.2.4-v)

\[ h^4 f^2 f_{w} : 8 - 12 \times x(5) \times x(7) \times x(2) - 12 \times x(6) \times x(7) \times x(2) - 3 \times x(3) - 12 \times x(6) \times x(7) \times x(3) - 12 \times x(7) \times x(3) - 15 \times x(4) = 0 , \] 

(8.2.4-vi)
\[ h^4 f^3 f_{y} = 4 - 8x(5)^2x(1) - 8x(5)^2x(2) - 8x(6)^2x(2) - 24x(6)^2x(2) - 24x(6)^2x(2) - 8x(7)^2x(3) - 8x(7)^2x(3) - 24x(6)^2x(7) - 8x(3) - 24x(6)^2x(7) - 8x(7)^2x(3) - 9x(4) = 0, \quad (8.2.4-vii) \]

\[ h^5 f^4 f_{y} = 1 - 30x(5)x(7)x(4) + 60x(5)x(7)x(8)x(4) + 60x(5)x(7)x(9)x(4) + 60x(5)x(7)x(10)x(4) - 60x(5)x(7)x(8)x(10)x(4) - 60x(5)x(7)x(9)x(10)x(4) + 30x(5)x(7)x(11)x(4) + 30x(5)x(7)x(12)x(4) - 60x(5)x(7)x(8)x(11)x(4) - 60x(5)x(7)x(9)x(11)x(4) - 60x(5)x(7)x(12)x(4) - 60x(5)x(7)x(9)x(12)x(4) = 0, \quad (8.2.4-viii) \]

\[ h^5 f^2 f_{y} = 22 - 60x(5)^2x(7)^2x(2) - 15x(6)^2x(3) - 30x(5)x(7)x(3) - 30x(5)^2x(7)x(3) - 30x(6)x(7)x(3) - 60x(5)x(6)x(7)x(3) - 15x(7)^2x(3) - 60x(5)x(7)^2x(3) - 60x(5)^2x(7)^2x(3) + 60x(6)^2x(8)x(3) + 60x(5)x(7)x(8)x(3) + 60x(5)^2x(7)x(8)x(3) + 120x(6)x(7)x(8)x(3) + 120x(5)x(6)x(7)x(8)x(3) + 60x(7)^2x(8)x(3) + 120x(5)x(7)^2x(8)x(3) - 60x(6)^2x(8)^2x(3) - 120x(6)x(7)x(8)^2x(3) - 60x(7)^2x(8)^2x(3) - 15x(4) - 90x(6)x(4) - 45x(6)^2x(4) - 90x(7)x(4) - 30x(5)x(7)x(4) - 30x(5)^2x(7)x(4) - 90x(6)x(7)x(4) - 60x(5)x(6)x(7)x(4) - 45x(7)^2x(4) - 60x(5)x(7)^2x(4) + 180x(6)x(8)x(4) + 120x(6)^2x(8)x(4) + 180x(7)x(8)x(4) + 60x(5)x(7)^2x(8)x(4) - 180x(6)x(8)^2x(4) - 120x(6)x(7)^2x(8)x(4) - 120x(5)x(6)x(8)x(4) - 180x(5)x(7)x(8)x(4) + 60x(5)^2x(7)x(8)x(4) - 120x(6)x(7)^2x(8)x(4) - 120x(5)x(6)^2x(8)x(4) - 180x(6)x(7)^2x(8)x(4) + 60x(5)^2x(7)^2x(8)x(4) - 120x(6)x(7)^2x(8)x(4) - 120x(5)^2x(7)^2x(8)x(4) - 180x(6)x(7)^2x(8)x(4) - 180x(5)^2x(7)^2x(8)x(4) - 277
120 \times x(5) \times x(6) \times x(7) \times x(9) \times x(4) + \\
120 \times x(7) \times x(9) \times x(4) + 120 \times x(5) \times x(7) \times x(9) \times x(4) + \\
120 \times x(5) \times x(6) \times x(8) \times x(9) \times x(4) - \\
120 \times x(6) \times x(8) \times x(9) \times x(4) + 120 \times x(5) \times x(7) \times x(8) \times x(9) \times x(4) - \\
240 \times x(6) \times x(7) \times x(9) \times x(4) - \\
120 \times x(7) \times x(8) \times x(9) \times x(4) - 60 \times x(5) \times x(9) \times x(2) \times x(4) + \\
120 \times x(5) \times x(6) \times x(9) \times x(2) \times x(4) - \\
60 \times x(6) \times x(9) \times x(2) \times x(4) + 120 \times x(5) \times x(7) \times x(9) \times x(2) \times x(4) - \\
120 \times x(6) \times x(7) \times x(9) \times x(2) \times x(4) - \\
60 \times x(7) \times x(9) \times x(2) \times x(4) + 30 \times x(10) \times x(4) + 90 \times x(6) \times x(10) \times x(4) + \\
30 \times x(6) \times x(10) \times x(4) + 90 \times x(7) \times x(10) \times x(4) + \\
60 \times x(6) \times x(7) \times x(10) \times x(4) + 30 \times x(7) \times x(10) \times x(4) - \\
180 \times x(6) \times x(8) \times x(10) \times x(4) - 60 \times x(6) \times x(8) \times x(10) \times x(4) - \\
180 \times x(7) \times x(8) \times x(10) \times x(4) - 120 \times x(6) \times x(7) \times x(8) \times x(10) \times x(4) - \\
60 \times x(7) \times x(9) \times x(10) \times x(4) - 15 \times x(10) \times x(11) \times x(4) + \\
30 \times x(11) \times x(4) - 60 \times x(5) \times x(11) \times x(4) + 90 \times x(6) \times x(11) \times x(4) + \\
30 \times x(6) \times x(11) \times x(4) + 90 \times x(7) \times x(11) \times x(4) + \\
60 \times x(6) \times x(7) \times x(11) \times x(4) + 30 \times x(7) \times x(11) \times x(4) - \\
180 \times x(6) \times x(8) \times x(11) \times x(4) - 60 \times x(6) \times x(8) \times x(11) \times x(4) - \\
180 \times x(7) \times x(8) \times x(11) \times x(4) - 120 \times x(6) \times x(7) \times x(8) \times x(11) \times x(4) - \\
60 \times x(7) \times x(9) \times x(11) \times x(4) - 60 \times x(7) \times x(9) \times x(11) \times x(4) + \\
180 \times x(5) \times x(9) \times x(11) \times x(4) + 60 \times x(5) \times x(9) \times x(11) \times x(4) - \\
180 \times x(6) \times x(9) \times x(11) \times x(4) - 60 \times x(6) \times x(9) \times x(11) \times x(4) - \\
180 \times x(7) \times x(9) \times x(11) \times x(4) - 120 \times x(6) \times x(7) \times x(9) \times x(11) \times x(4) - \\
60 \times x(7) \times x(9) \times x(11) \times x(4) - 30 \times x(11) \times x(11) \times x(4) + \\
60 \times x(5) \times x(10) \times x(11) \times x(4) - 15 \times x(11) \times x(11) \times x(4) + \\
60 \times x(5) \times x(11) \times x(11) \times x(4) - 60 \times x(5) \times x(11) \times x(11) \times x(4) + \\
30 \times x(12) \times x(4) + 30 \times x(6) \times x(12) \times x(4) + 30 \times x(6) \times x(12) \times x(4) + \\
30 \times x(7) \times x(12) \times x(4) - 120 \times x(5) \times x(7) \times x(12) \times x(4) - \\
60 \times x(5) \times x(7) \times x(12) \times x(4) + 60 \times x(6) \times x(7) \times x(12) \times x(4) - \\
120 \times x(5) \times x(6) \times x(7) \times x(12) \times x(4) + 30 \times x(7) \times x(12) \times x(4) - \\
120 \times x(5) \times x(7) \times x(12) \times x(4) - 180 \times x(6) \times x(8) \times x(12) \times x(4) - \\
60 \times x(6) \times x(9) \times x(12) \times x(4) - \\
180 \times x(7) \times x(8) \times x(12) \times x(4) - 120 \times x(6) \times x(7) \times x(8) \times x(12) \times x(4) - \\
60 \times x(7) \times x(9) \times x(12) \times x(4) + \\
180 \times x(5) \times x(9) \times x(12) \times x(4) + 60 \times x(5) \times x(9) \times x(12) \times x(4) - \\
180 \times x(6) \times x(9) \times x(12) \times x(4) - 60 \times x(6) \times x(9) \times x(12) \times x(4) - \\
180 \times x(7) \times x(9) \times x(12) \times x(4) - 120 \times x(6) \times x(7) \times x(9) \times x(12) \times x(4) - \\
60 \times x(7) \times x(9) \times x(12) \times x(4) - 30 \times x(10) \times x(12) \times x(4) + \\
60 \times x(6) \times x(10) \times x(12) \times x(4) + 60 \times x(7) \times x(10) \times x(12) \times x(4) - \\
30 \times x(11) \times x(12) \times x(4) + 60 \times x(5) \times x(11) \times x(12) \times x(4) + \\
60 \times x(6) \times x(11) \times x(12) \times x(4) - 120 \times x(5) \times x(6) \times x(11) \times x(12) \times x(4) + \\
60 \times x(7) \times x(11) \times x(12) \times x(4) - 120 \times x(5) \times x(7) \times x(11) \times x(12) \times x(4) - \\
15 \times x(12) \times x(12) \times x(4) + 60 \times x(6) \times x(12) \times x(12) \times x(4) - \\
60 \times x(7) \times x(12) \times x(12) \times x(4) - 120 \times x(6) \times x(7) \times x(12) \times x(12) \times x(4) - \\
60 \times x(7) \times x(12) \times x(12) \times x(4) = 0 \ . 
(8.2.4 - ix)
\[ h^5 f^j f_{\gamma} = 8 - 60 \times x(5) \times 2 \times x(6) \times x(7) \times x(2) - 60 \times x(5) \times 2 \times x(7) \times 2 \times x(2) - 15 \times x(6) \times 2 \times x(3) - 30 \times x(6) \times x(7) \times x(3) - 60 \times x(5) \times 2 \times x(6) \times x(7) \times x(3) - 15 \times x(7) \times 2 \times x(3) - 60 \times x(5) \times 2 \times x(7) \times 2 \times x(3) + 60 \times x(6) \times x(7) \times 2 \times x(3) + 30 \times x(5) \times 2 \times x(9) \times x(3) + 30 \times x(6) \times 2 \times x(9) \times x(3) + 60 \times x(6) \times x(7) \times 2 \times x(9) \times x(3) - 30 \times x(7) \times 2 \times x(9) \times x(3) - 15 \times x(4) - 15 \times x(6) \times 2 \times x(4) - 30 \times x(6) \times x(7) \times x(4) - 15 \times x(7) \times 2 \times x(4) + 30 \times x(6) \times x(8) \times x(4) + 60 \times x(6) \times x(7) \times 2 \times x(8) \times x(4) - 30 \times x(5) \times 2 \times x(9) \times x(4) + 30 \times x(6) \times 2 \times x(9) \times x(4) + 60 \times x(6) \times x(7) \times 2 \times x(9) \times x(4) - 30 \times x(7) \times 2 \times x(9) \times x(4) - 15 \times x(4) \times 10 \times x(4) + 15 \times x(11) \times x(4) - 15 \times x(4) \times 12 \times x(4) - 60 \times x(5) \times 2 \times x(11) \times x(4) + 15 \times x(12) \times x(4) - 60 \times x(6) \times 2 \times x(12) \times x(4) - 120 \times x(6) \times x(7) \times x(12) \times x(4) - 60 \times x(7) \times 2 \times x(12) \times x(4) = 0, \] (8.2.4 - x)

\[ h^5 f^j f_{\gamma'} = 8 - 40 \times x(5) \times 3 \times x(7) \times x(2) - 120 \times x(5) \times x(6) \times 2 \times x(7) \times x(2) - 240 \times x(5) \times x(6) \times 2 \times x(7) \times x(2) - 120 \times x(5) \times x(7) \times 3 \times x(3) - 15 \times x(6) \times x(3) - 20 \times x(6) \times x(3) - 60 \times x(5) \times 2 \times x(6) \times x(3) - 120 \times x(5) \times x(6) \times x(3) - 40 \times x(5) \times x(6) \times 3 \times x(7) \times x(3) - 60 \times x(6) \times 2 \times x(7) \times x(3) - 120 \times x(5) \times x(6) \times 2 \times x(7) \times x(3) - 60 \times x(6) \times x(7) \times 2 \times x(3) - 240 \times x(5) \times x(6) \times x(7) \times 2 \times x(3) - 20 \times x(7) \times 3 \times x(3) - 120 \times x(5) \times x(7) \times 3 \times x(3) + 30 \times x(6) \times x(8) \times x(3) + 40 \times x(6) \times 3 \times x(8) \times x(3) + 30 \times x(7) \times x(8) \times x(3) + 120 \times x(6) \times 2 \times x(7) \times x(8) \times x(3) + 120 \times x(6) \times x(7) \times 2 \times x(8) \times x(3) + 40 \times x(7) \times 3 \times x(8) \times x(3) - 30 \times x(5) \times x(9) \times x(3) - 40 \times x(5) \times 3 \times x(9) \times x(3) + 30 \times x(6) \times x(9) \times x(3) + 40 \times x(6) \times 3 \times x(9) \times x(3) + 30 \times x(7) \times x(9) \times x(3) + 120 \times x(6) \times 2 \times x(7) \times x(9) \times x(3) + 120 \times x(6) \times x(7) \times 2 \times x(9) \times x(3) + 40 \times x(7) \times 3 \times x(9) \times x(3) - 65 \times x(4) - 15 \times x(6) \times x(4) - 20 \times x(6) \times 3 \times x(4) - 15 \times x(7) \times x(4) - 60 \times x(6) \times 2 \times x(7) \times x(4) - 60 \times x(6) \times x(7) \times 2 \times x(4) - 20 \times x(7) \times 3 \times x(4) + 30 \times x(6) \times x(8) \times x(4) + 40 \times x(6) \times 3 \times x(8) \times x(4) + 30 \times x(7) \times x(8) \times x(4) + 120 \times x(6) \times 2 \times x(7) \times x(8) \times x(4) + 120 \times x(6) \times x(7) \times 2 \times x(8) \times x(4) + 40 \times x(7) \times 3 \times x(8) \times x(4) - 30 \times x(5) \times x(9) \times x(4) - 40 \times x(6) \times 3 \times x(9) \times x(4) + 30 \times x(6) \times x(9) \times x(4) + 40 \times x(6) \times 3 \times x(9) \times x(4) + 30 \times x(7) \times x(9) \times x(4) + 120 \times x(6) \times 2 \times x(7) \times x(9) \times x(4) + 120 \times x(6) \times x(7) \times 2 \times x(9) \times x(4) + 40 \times x(7) \times 3 \times x(9) \times x(4) + 65 \times x(10) \times x(4) + 65 \times x(11) \times x(4) - 120 \times x(5) \times x(11) \times x(4) - 40 \times x(5) \times 3 \times x(11) \times x(4) + 65 \times x(12) \times x(4) - 120 \times x(6) \times x(12) \times x(4) - 40 \times x(6) \times 3 \times x(12) \times x(4) - 120 \times x(7) \times x(12) \times x(4) - 120 \times x(6) \times 2 \times x(7) \times x(12) \times x(4) - 120 \times x(6) \times x(7) \times 2 \times x(12) \times x(4) - 40 \times x(7) \times 3 \times x(12) \times x(4) = 0, \] (8.2.4 - xi)

\[ h^5 f^j f_{\gamma''} = 32 - 80 \times x(5) \times 4 \times x(1) - 80 \times x(5) \times 4 \times x(2) - 80 \times x(6) \times 4 \times x(2) - 320 \times x(6) \times 3 \times x(7) \times x(2) - 279 \]
\[480x(6)^2x(7)^2x(2) - 320x(6)x(7)^3x(2) - 80x(7)^4x(2) - 5x(3) - 80x(6)^4x(3) - 320x(6)x(7)^3x(3) - 480x(6)^2x(7)^2x(3) - 320x(6)x(7)^3x(3) - 80x(7)^4x(3) - 85x(4) = 0\] 

(8.2.4 - xii)

where

\[x(1) = w, x(2) = w, x(3) = w, x(4) = w, x(5) = a, x(6) = a,\]
\[x(7) = a, x(8) = a, x(9) = a, x(10) = a, x(11) = a, and x(12) = a.\]

Similarly, equations (8.2.4-i)-(8.2.4-xii) are solved simultaneously by using the NAG routine (Subroutine C05NBF) for solving a system of non-linear equations to give the required parameters, i.e.,
\[w_1 = x(1) = 0.2615038147, w_2 = x(2) = -0.2765809214, w_3 = x(3) = 0.5947141647\]
\[w_4 = x(4) = 0.4203629420, a_1 = x(5) = 1.5471214403, a_2 = x(6) = 0.1756458393\]
\[a_3 = x(7) = 0.1243059001, a_4 = x(8) = 0.1009316694, a_5 = x(9) = 0.1100539630\]
\[a_6 = x(10) = 0.99974318, x_7 = x(11) = -0.0928890403, x_8 = x(12) = -0.6201812828\]

(8.2.5)

Thus, this new fifth order WRK method can be written as follows

\[y_{\text{new}} = y_n + h \left[ 0.2615038147\left(\frac{k_1 + k_2}{2}\right) - 0.2765809214\left(\frac{k_3}{2}\right) + 0.5947141647\left(\frac{k_3}{2}\right) + 0.4203629420\left(\frac{k_3}{2}\right) \right] \] 

(8.2.6)

where \(k_1 = f(y_n)\)
\[k_2 = f(y_n + 1.5471214403hk_1)\]
\[k_3 = f(y_n + 0.1756458393hk_1 + 0.1243059001hk_2)\]
\[k_4 = f(y_n + 0.1009316694hk_1 + 0.1100539630hk_2 + 0.2890143692hk_3)\]
\[k_5 = f(y_n + 0.9997431862hk_1 - 0.0928890403hk_2 - 0.6201812828hk_3 + 0.17133271396hk_4) . \]

(8.2.7)

By use of Mathematica to rationalize the coefficients in equation (8.2.5), we obtain
\[w_1 = 0.2615038147 = \frac{28449}{108790}, \quad w_2 = -0.2765809214 = \frac{40890}{147841}\]

280
\[ w_3 = 0.5947141647 = \frac{211318}{355327}, \quad w_4 = 0.4203629420 = \frac{65601}{156058} \]

\[ a_1 = 1.5471214403 = \frac{479767}{310103}, \quad a_2 = 0.1756458393 = \frac{14081}{80167} \]

\[ a_3 = 0.1243059001 = \frac{7768}{62491}, \quad a_4 = 0.1009316694 = \frac{20551}{203613} \]

\[ a_5 = 0.0928890403 = \frac{15153}{137687}, \quad a_6 = 0.9997431862 = \frac{151822}{151861} \]

\[ a_7 = -0.0928890403 = \frac{4909}{52848}, \quad a_8 = -0.6201812828 = \frac{91411}{147394} \]

\[ a_{11} = \frac{1}{2} - a_4 - a_5 = 0.2890143692 = \frac{32604}{112811} \]

\[ a_{21} = 1 - a_6 - a_7 - a_8 = 0.7133271396 = \frac{548973}{769595} \]

Hence this new fifth order WRK method can be written in rational form as

\[ y_{n+1} = y_n + h \left[ \frac{28449}{108790} \left( \frac{k_1 + k_2}{2} \right) - \frac{40890}{147841} \left( \frac{k_2 + k_3}{2} \right) \right. \\
\left. + \frac{211318}{355327} \left( \frac{k_3 + k_4}{2} \right) + \frac{65601}{156058} \left( \frac{k_4 + k_5}{2} \right) \right] \]

\[ (8.2.8) \]

where \( k_1 = f(y_n) \)

\[ k_2 = f \left( y_n + \frac{479767}{310103} h k_1 \right) \]

\[ k_3 = f \left( y_n + \frac{14081}{80167} h k_1 + \frac{7768}{62491} h k_2 \right) \]

\[ k_4 = f \left( y_n + \frac{20551}{203613} h k_1 + \frac{15153}{137687} h k_2 + \frac{32604}{112811} h k_3 \right) \]

\[ k_5 = f \left( y_n + \frac{151822}{151861} h k_1 - \frac{4909}{52848} h k_2 - \frac{91411}{147394} h k_3 + \frac{548973}{769595} h k_4 \right) \]

\[ (8.2.9) \]
8.2.1 Error Analysis

By substituting the values of \( a_i, 1 \leq i \leq 8 \) and \( w_i, 1 \leq i \leq 4 \) in (8.2.5) into (8.2.1) and (8.2.2) using Mathematica and evaluating all the terms up to \((h^6)\) to represent the local truncation error (LTE) for this method we have

\[
LTE = h^6 \left[ \frac{1}{720} f_y^5 + 0.0018022816 f_y^3 f_y x - 0.0166861138 f_y^3 f_x y \right. \\
\left. + 0.0008266021 f_y^4 f_y y + 0.0041171137 f_y^4 f_y y y \\
- 0.0023096134 f_y^4 f_y y y y + 0.0000588245 f_y^5 f_y y y y \right] + o(h^7) \text{ as } h \to 0. \\
\]

(8.2.10)

8.2.2 Stability Analysis of the Fifth Order WRK Method

We examine the stability region for the fifth order arithmetic mean WRK method with the test equation \( y' = \lambda y \) and we obtain

\[
k_1 = \lambda y, \\
k_2 = \lambda (y + 1.5471214403hk_1), \\
k_3 = \lambda (y + 0.1756458393hk_2 + 0.1243059001hk_1), \\
k_4 = \lambda (y + 0.1009316694hk_3 + 0.1100539630hk_2 + 0.2890143692hk_1), \\
k_5 = \lambda (y + 0.9997431862hk_4 - 0.0928890403hk_2 - 0.6201812828hk_3) \\
+ 0.7133271396hk_5). 
\]

(8.2.11)

By substituting \( k_i, 1 \leq i \leq 5 \) in equation (8.2.11) and \( w_i, 1 \leq i \leq 4 \) in equation (8.2.5) into the fifth order WRK formula, i.e.,

\[
y_{m_i} = y + h \left[ \sum_{i=1}^{5} w_i \left( \frac{k_i + k_{m_i}}{2} \right) \right] 
\]

(8.2.12)

to obtain

\[
y_{m_i} = y + (\lambda h)y + 0.5(\lambda h)^2 y + 0.166667(\lambda h)^3 y + 0.0416667(\lambda h)^4 y \\
+ 0.00833333(\lambda h)^5 y + o(h^6). 
\]

(8.2.13)
By rationalizing the coefficients in equation (8.2.13) we have

\[ y_{n+1} = y_n + (h\lambda)y_n + \frac{1}{2}(h\lambda)^2 y_n + \frac{1}{6}(h\lambda)^3 y_n + \frac{1}{24}(h\lambda)^4 y_n + \frac{1}{120}(h\lambda)^5 y_n + O(h^6) \]  

By substituting \( h\lambda = z \) in (8.2.14), we can show that

\[ y_{n+1} = y_n + y_n \left[ z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} \right] + O(z^6) \]  

Following equation (3.3.30), we write \( \frac{y_{n+1}}{y_n} = Q \) in the equation (8.2.15), to obtain

\[ Q = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 + \frac{1}{120}z^5 + O(z^6) \]  

To determine the stability region of the fifth order WRK formula in the complex plane that satisfy the condition

\[ \left| \frac{y_{n+1}}{y_n} \right| = |Q| < 1 \]

i.e.,

\[ \left| 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} \right| < 1 \]  

by the use of Mathematica, we can plot the graphic surface defined by equation (8.2.17) i.e.,

![Figure 8.1: Graphic surface defined by the fifth order WRK formula](image)

Figure 8.1: Graphic surface defined by the fifth order WRK formula
and the stability region defined by the formula in equation (8.2.17) as shown in Figure 8.2.

![Stability region for fifth order WRK method](image)

**Figure 8.2: Stability region for fifth order WRK method**

### 8.2.3 Numerical Example

We consider the IVP

\[ y' = x - y + 1 \quad , \quad y(0) = 1 \quad , \quad 0 \leq x \leq 1 \tag{8.2.18} \]

where the exact solution is \( y(x) = x + \exp(-x) \). The absolute error in the numerical solution using formula (8.2.1)-(8.2.2) or (8.2.8)-(8.2.9) compared with the fourth order WRK formula (classical formula), RK5(6)-Nystrom (Lambert [1973]) and RK4(5)-Merson (Merson [1957]) are shown in Table 8.3.

<table>
<thead>
<tr>
<th>x</th>
<th>Classical AM-RK4</th>
<th>RK4(5) Merson</th>
<th>RK5(6) Nystrom</th>
<th>AM-RK5(5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.8196E-07</td>
<td>0.1252E-07</td>
<td>0.1707E-08</td>
<td>0.1380E-08</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1483E-06</td>
<td>0.2266E-07</td>
<td>0.3089E-08</td>
<td>0.2498E-08</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2013E-06</td>
<td>0.3075E-07</td>
<td>0.4192E-08</td>
<td>0.3391E-08</td>
</tr>
<tr>
<td>0.4</td>
<td>0.2429E-06</td>
<td>0.3710E-07</td>
<td>0.5058E-08</td>
<td>0.4091E-08</td>
</tr>
<tr>
<td>0.5</td>
<td>0.2747E-06</td>
<td>0.4196E-07</td>
<td>0.5721E-08</td>
<td>0.4627E-08</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2983E-06</td>
<td>0.4556E-07</td>
<td>0.6211E-08</td>
<td>0.5024E-08</td>
</tr>
<tr>
<td>0.7</td>
<td>0.3149E-06</td>
<td>0.4810E-07</td>
<td>0.6557E-08</td>
<td>0.5303E-08</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3256E-06</td>
<td>0.4974E-07</td>
<td>0.6781E-08</td>
<td>0.5484E-08</td>
</tr>
<tr>
<td>0.9</td>
<td>0.3315E-06</td>
<td>0.5063E-07</td>
<td>0.6902E-08</td>
<td>0.5583E-08</td>
</tr>
<tr>
<td>1.0</td>
<td>0.3332E-06</td>
<td>0.5090E-07</td>
<td>0.6939E-08</td>
<td>0.5613E-08</td>
</tr>
</tbody>
</table>

284
From Table 8.3, we can see that the accuracy obtained from using AM-RK5(5) is better than the AM-RK4, RK4(5)-Merson and RK5(6)-Nystrom methods. When we make a work comparison with the fifth order RK5(6)-Nystrom method, then this new fifth order method saves one function evaluation. From the above discussion we can conclude that contrary to Butcher [1987] a fifth order arithmetic mean weighted Runge-Kutta method with five stages does exist. The study of fifth order weighted Runge-Kutta methods for the variety of means are under investigation.

8.3 NEW FIFTH ORDER CONTRAHARMONIC MEAN WEIGHTED RUNGE-KUTTA FORMULA

Now by replacing the arithmetic means of \( \left( \frac{k_i + k_{i+1}}{2} \right) \) with their corresponding contraharmonic means in equation (8.1.1) we obtain a new integration formula which can be written as follows:

\[
y_{n+1} = y_n + h \left[ \sum_{i=1}^{4} w_i \left( \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right) \right]
\]

(8.3.1)

8.3.1 The Fourth Order Contraharmonic Mean Weighted Runge-Kutta Formula

By using a similar procedure as in (8.2) we can establish a new fourth order contraharmonic mean weighted WRK formula in the form

\[
y_{n+1} = y_n + h \left[ \sum_{i=1}^{3} w_i \left( \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right) \right]
\]

(8.3.2)

where \( \sum_{i=1}^{3} w_i = 1 \)

\[
k_1 = f(y_n)
k_2 = f(y_n + a_1hk_1)
k_3 = f(y_n + a_2hk_1 + (\frac{1}{3} - a_1)hk_2)
k_4 = f(y_n + a_3hk_1 + a_4hk_2 + (1 - a_2 - a_4)hk_3)
\]

(8.3.3)
By comparing the right hand side of equation (8.3.1) with the Taylor series expansion for \( y(x_n) \), we obtain the following 7 equations involving 7 variables which must be satisfied to obtain fifth order accuracy, i.e.,

\[
\begin{align*}
  h f &: \quad 1 - x(1) - x(2) - x(3) = 0 \\
  h^2 ff_x & : \quad 2 - 2x(4)x(1) - x(2) - 2x(4)x(2) - 3x(3) = 0 \\
  h^2 ff_y & : \quad 8 - 12x(4)x(1) - 3x(2) - 12x(4)x(2) + 12x(4)x(5)x(2) - 15x(3) - 12x(4)x(3) + 12x(4)x(7)x(3) - 24x(4)x(7)x(3) = 0 \\
  h^3 f^2 f_y & : \quad 8 - 12x(4)x(1) - 2x(2) - 12x(4)x(2) - 15x(3) = 0 \\
  h^4 f^3 f_y & : \quad 8 + 24x(4)x(1) + 3x(2) - 30x(4)x(2) + 36x(4)x(2) + 24x(4)x(5)x(2) - 96x(4)x(5)x(3) + 24x(4)x(7)x(3) + 24x(4)x(7)x(3) - 24x(4)x(7)x(3) = 0 \\
  h^4 f^2 f_x & : \quad 16 - 24x(4)x(1) - 3x(2) - 6x(4)x(2) - 24x(4)x(5)x(2) + 24x(4)x(5)x(3) + 24x(4)x(5)x(3) + 24x(4)x(7)x(3) - 24x(4)x(7)x(3) = 0 \\
  h^5 f^3 f_x & : \quad 4 - 8x(4)x(1) - x(2) - 8x(4)x(2) - 9x(3) = 0
\end{align*}
\]

where \( x(1) = w_1 \), \( x(2) = w_2 \), \( x(3) = w_3 \), \( x(4) = a_1 \), \( x(5) = a_2 \), \( x(6) = a_3 \), \( x(7) = a_4 \).

Equations (8.3.4-i)-(8.3.4-vii) are solved simultaneously using the NAG routine (Subroutine C05NBF) for solving a system of non-linear equations to give the required parameters, i.e.,

\[
\begin{align*}
  w_1 &= 0.3333333333 \quad , \quad w_2 = 0.3333333333 \quad , \quad w_3 = 0.3333333333 \\
  a_1 &= 0.5000000000 \quad , \quad a_2 = 0.1250000000 \quad , \quad a_3 = 0.2500000000 \\
  a_4 &= -0.7500000000 \quad \ldots \quad (8.3.5)
\end{align*}
\]
By rationalizing the coefficients in equation (8.3.5) we have

\[ w_1 = w_2 = w_3 = \frac{1}{3}, \quad a_1 = \frac{1}{2}, \quad a_2 = \frac{1}{8} \]

\[ \begin{align*}
        a_3 &= \frac{1}{4}, \\
        a_4 &= \frac{-3}{4}
\end{align*} \]

Thus, this new WRK method gives the same result as the fourth order contraharmonic mean (C_M) method as in Chapter 4 in the form

\[ y_{n+1} = y_n + \frac{h}{3} \left[ \frac{k_1^2 + k_2^2}{k_1 + k_2} + \frac{k_2^2 + k_3^2}{k_2 + k_3} + \frac{k_3^2 + k_4^2}{k_3 + k_4} \right] \]

where \( k_i = f(y_n) \)

\[ \begin{align*}
        k_1 &= f(y_n + \frac{1}{3}hk_1) \\
        k_2 &= f(y_n + \frac{1}{6}hk_1 + \frac{1}{3}hk_2) \\
        k_3 &= f(y_n + \frac{1}{6}hk_1 - \frac{1}{3}hk_2 + \frac{1}{3}hk_3) \\
        k_4 &= f(y_n + \frac{1}{6}hk_1 + \frac{1}{6}hk_2 + \frac{1}{6}hk_3 + (\frac{1}{3} - a_4 - a_5)hk_4)
\end{align*} \]

### 8.3.2 A New Fifth Order Contraharmonic Mean Weighted Runge-Kutta Formula

We now attempt to develop a fifth order method by using the same procedure as in the fourth order C_M method including the extra parameters involved in the last two \( k \) values i.e.,

\[ y_{n+1} = y_n + h \left[ \sum_{i=1}^{5} w_i \left( \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right) \right] \quad \text{(8.3.6)} \]

where \( \sum_{i=1}^{5} w_i = 1 \)

\[ \begin{align*}
        k_1 &= f(y_n) \\
        k_2 &= f(y_n + a_1hk_1) \\
        k_3 &= f(y_n + a_2hk_1 + a_3hk_2) \\
        k_4 &= f(y_n + a_3hk_1 + a_4hk_2 + (\frac{1}{3} - a_5 - a_6)hk_3) \\
        k_5 &= f(y_n + a_6hk_1 + a_7hk_2 + a_6hk_3 + (1 - a_6 - a_7)hk_4)
\end{align*} \]

\[ \ldots \quad \text{(8.3.7)} \]
By substituting equation (8.3.7) into (8.3.6) and subtracting from equation (8.2.3), we obtain 12 equations with 12 parameters, i.e.,

\[
h_f: 1 - x(1) - x(2) - x(3) - x(4) = 0 \quad (8.3.8-i)
\]

\[
h^{ff}_f: 2 - 2x(5)x(1) - 2x(5)x(2) - 2x(6)x(2) - 2x(7)x(2) - x(3) - 2x(6)x(3) - 2x(7)x(3) - 3x(4) = 0 \quad (8.3.8-ii)
\]

\[
h^{ff}_f: 8 - 12x(5)*2x(1) - 12x(5)*2x(2) + 24x(5)x(6)x(2) - 12x(6)*2x(2) - 24x(6)x(7)x(2) - 12x(7)*2x(3) - 24x(6)x(7)*3x(2) - 12x(8)*2x(3) - 24x(6)x(7)*3x(3) - 24x(6)*x(9)*x(3) - 15x(4) - 12x(6)x(4) - 12x(7)x(4) - 30x(4) = 0 \quad (8.3.8-iii)
\]

\[
h^{ff}_f: 8 + 24x(5)*3x(1) + 24x(5)*3x(2) - 24x(5)x(6)*2x(2) - 24x(6)*x(7)x(2) - 12x(7)*2x(3) - 15x(4) = 0 \quad (8.3.8-iv)
\]

\[
h^4f_f: 8 + 24x(5)*3x(1) + 24x(5)*3x(2) - 24x(5)x(6)*2x(2) - 24x(6)*x(7)x(2) - 12x(7)*2x(3) - 15x(4) = 0 \quad (8.3.8-iv)
\]
\*x(10)\*x(4) + 48\*x(7)\*x(10)\*x(4) - 96\*x(6)\*x(8)\*x(10) \
\*x(4) - 96\*x(7)\*x(8)\*x(10)\*x(4) + 96\*x(5)\*x(9)\*x(10) \
\*x(4) - 96\*x(6)\*x(9)\*x(10)\*x(4) - 96\*x(7)\*x(9)\*x(10) \
\*x(4)+24\*x(11)\*x(4) - 48\*x(5)\*x(11)\*x(4) + 48\*x(6)\*x(11) \
\*x(4) + 48\*x(7)\*x(11)\*x(4) - 96\*x(6)\*x(8)\*x(11)\*x(4) - 
\*96\*x(7)\*x(8)\*x(11)\*x(4) + 96\*x(5)\*x(9)\*x(11)\*x(4) - 
\*96\*x(6)\*x(9)\*x(11)\*x(4) - 96\*x(7)\*x(9)\*x(11)\*x(4) + 
\*24\*x(12)\*x(4) - 96\*x(5)\*x(7)\*x(12)\*x(4) - 96\*x(6)\*x(7)\*x(12)\*x(4) + 
\*96\*x(5)\*x(9)\*x(12)\*x(4) - 96\*x(6)\*x(9)\*x(12)\*x(4) - 
\*96\*x(7)\*x(9)\*x(12)\*x(4)\*x(4) = 0 

(8.3.8-v)

\[ h^4 f_2 f_{1r} : \] 16 - 24\*x(5)\*3\*x(1) - 24\*x(5)\*3\*x(2) + 
24\*x(5)\*2\*x(6)\*x(2) + 24\*x(5)\*x(6)\*2\*x(2) - 
24\*x(6)\*3\*x(2) - 
72\*x(6)\*2\*x(7)\*x(2) - 24\*x(5)\*x(7)\*2\*x(2) -
72\*x(6)\*x(7)\*2\*x(2) - 24\*x(7)\*3\*x(2) - 3\*x(3) - 
6\*x(6)\*x(3) - 24\*x(6)\*3\*x(3) - 6\*x(7)\*x(3) - 
24\*x(5)\*2\*x(7)\*x(3) - 48\*x(5)\*x(6)\*x(7)\*x(3) - 
72\*x(6)\*2\*x(7)\*x(3) - 48\*x(5)\*x(7)\*2\*x(3) - 
72\*x(6)\*x(7)\*2\*x(3) - 24\*x(7)\*3\*x(3) + 
24\*x(6)\*x(8)\*x(3) + 24\*x(6)\*2\*x(8)\*x(3) + 
24\*x(7)\*x(8)\*x(3) + 48\*x(6)\*x(7)\*x(8)\*x(3) + 
24\*x(7)\*2\*x(8)\*x(3) - 24\*x(5)\*x(9)\*x(3) - 
24\*x(5)\*2\*x(9)\*x(3) + 24\*x(6)\*x(9)\*x(3) + 
24\*x(6)\*2\*x(9)\*x(3) + 24\*x(7)\*x(9)\*x(3) + 
48\*x(6)\*x(7)\*x(9)\*x(3) + 24\*x(7)\*2\*x(9)\*x(3) - 39\*x(4) - 
12\*x(6)\*x(4) - 12\*x(6)\*2\*x(4) - 12\*x(7)\*x(4) - 
12\*x(6)\*x(7)\*x(4) - 12\*x(7)\*2\*x(4) + 
24\*x(6)\*x(8)\*x(4) + 24\*x(6)\*2\*x(8)\*x(4) + 
24\*x(7)\*x(8)\*x(4) + 48\*x(6)\*x(7)\*x(8)\*x(4) + 
24\*x(7)\*2\*x(8)\*x(4) - 24\*x(5)\*x(9)\*x(4) - 
24\*x(5)\*2\*x(9)\*x(4) + 24\*x(6)\*x(9)\*x(4) + 
24\*x(6)\*2\*x(9)\*x(4) + 24\*x(7)\*x(9)\*x(4) + 
48\*x(6)\*x(7)\*x(9)\*x(4) + 24\*x(7)\*2\*x(9)\*x(4) + 
30\*x(10)\*x(4) + 30\*x(11)\*x(4) - 
48\*x(5)\*x(11)\*x(4) - 24\*x(5)\*2\*x(11)\*x(4) + 
30\*x(12)\*x(4) - 48\*x(6)\*x(12)\*x(4) - 
24\*x(6)\*2\*x(12)\*x(4) - 48\*x(7)\*x(12)\*x(4) - 
48\*x(6)\*x(7)\*x(12)\*x(4) - 24\*x(7)\*2\*x(12)\*x(4)\*x(4) = 0 

(8.3.8-vi)

\[ h^4 f_3 f_{1r} : \] 4 - 8\*x(5)\*3\*x(1) - 8\*x(5)\*3\*x(2) - 8\*x(6)\*3\*x(2) 
- 24\*x(6)\*2\*x(7)\*x(2) - 24\*x(6)\*x(7)\*2\*x(2) - 
8\*x(7)\*3\*x(2) - x(3) - 8\*x(6)\*3\*x(3) - 24\*x(6)\*2\*x(7) 
\*x(3) - 24\*x(6)\*x(7)\*2\*x(3) - 8\*x(7)\*3\*x(3) - 9\*x(4) = 0 

(8.3.8-vii)

\[ h^5 f f_{1r} : \] 32 - 240\*x(5)\*4\*x(1) - 240\*x(5)\*4\*x(2) + 
480\*x(5)\*2\*x(6)\*2\*x(2) - 240\*x(6)\*4\*x(2) - 
480\*x(5)\*3\*x(7)\*x(2) + 1440\*x(5)\*x(6)\*2\*x(7)\*x(2) 
- 960\*x(6)\*3\*x(7)\*x(2) - 

289
1440*x(5)**2*x(7)**2*x(2) + 2880*x(5)*x(6)*x(7)**2*x(2) - 1440*x(6)*x(7)**2*x(2) + 1440*x(5)*x(7)**3*x(2) - 960*x(6)*x(7)**3*x(2) - 240*x(7)**4*x(2) - 15*x(3) + 180*x(6)*x(3) - 360*x(6)**2*x(3) - 240*x(6)**3*x(3) - 240*x(6)**4*x(3) + 180*x(7)**2*x(3) - 600*x(5)*x(7)**3*x(3) - 720*x(6)**2*x(7)**2*x(3) + 720*x(6)**3*x(7)**2*x(3) - 1440*x(5)**2*x(7)**2*x(3) + 1440*x(5)*x(6)**2*x(7)**2*x(3) - 720*x(6)**2*x(7)**3*x(3) - 1440*x(5)*x(7)**2*x(8)**2*x(3) - 960*x(6)*x(7)**2*x(8)**2*x(3) - 240*x(7)**3*x(8)**2*x(3) - 1440*x(5)*x(7)**2*x(9)**2*x(3) - 960*x(6)*x(7)**2*x(9)**2*x(3) - 240*x(7)**3*x(9)**2*x(3) - 1440*x(5)*x(7)**3*x(9)**2*x(3) - 240*x(6)**2*x(7)**3*x(9)**2*x(3) + 1440*x(5)**2*x(7)**3*x(9)**2*x(3) - 720*x(6)**2*x(7)**3*x(9)**2*x(3) - 240*x(7)**3*x(8)**3*x(3) - 1440*x(6)**2*x(7)**3*x(8)**3*x(3) - 720*x(6)**2*x(9)**3*x(3) - 1440*x(7)**2*x(9)**3*x(3) - 360*x(8)*x(9)
1920*\(x(5)*x(6)*x(8)*x(9)*x(4)\) - 1920*\(x(6)**2*x(8)*x(9)\) 
\(x(4) + 1920*x(5)*x(7)*x(8)*x(9)*x(4) - 3840*x(6)*x(7)*x(8)*x(9)*x(4) - 1920*x(7)**2*x(8)*x(9)\) 
\(x(4) - 960*x(5)**2*x(9)**2*x(4) + 1920*x(5)*x(6)*x(9)**2*x(4) - 1920*x(6)*x(7)*x(9)**2*x(4) - 960*x(7)**2*x(9)**2*x(4) + 60*x(l0)*x(4) + 960*x(5)*x(7)*x(l0)*x(4) - 1920*x(5)*x(7)*x(8)*x(l0)*x(4) - 1920*x(5)*x(7)*x(9)*x(l0)*x(4) - 240*x(l0)**2*x(4) + 60*x(ll)*x(4) + 960*x(5)*x(ll)*x(4) - 120*x(6)*x(ll)*x(4) + 960*x(5)*x(ll)**2*x(4) + 960*x(5)**2*x(ll)**2*x(4) + 60*x(l2)*x(4) - 120*x(6)*x(l2)*x(4) + 960*x(6)**2*x(l2)*x(4) + 1920*x(6)*x(7)*x(l2)**2*x(4) - 960*x(7)**2*x(l2)**2*x(4) = 0

(8.3.8-viii)

\(h^i f^j f^k f_m : 352 + 720*x(5)**4*x(1) + 720*x(5)**4*x(2) - 480*x(5)**3*x(6)*x(2) - 480*x(5)**2*x(6)**2*x(2) - 480*x(5)*x(6)**3*x(2) + 720*x(6)**4*x(2) + 720*x(6)**4*x(3) - 480*x(6)**3*x(7)*x(2) - 7200*x(6)**2*x(7)*x(3) - 2880*x(6)*x(7)**2*x(3) - 7200*x(6)*x(7)**2*x(4) - 7200*x(6)*x(7)**2*x(5) - 120*x(7)**2*x(3) - 1920*x(5)**2*x(7)**2*x(3) + 2160*x(6)*x(7)**2*x(3) - 5760*x(5)*x(6)*x(7)**2*x(3) + 291
4320*x(6)**2*x(7)**2*x(3) + 720*x(7)**3*x(3)2880*x(5)*x(7)**3*x(3) + 2880*x(6)*x(7)**3*x(3) +
720*x(7)**4*x(3) + 720*x(6)*x(8)*x(3) +
480*x(6)**2*x(8)*x(3)- 1920*x(6)**3*x(8)*x(3) +
720*x(7)*x(8)*x(3) + 960*x(5)*x(7)*x(8)*x(3) +
960*x(5)**2*x(7)*x(8)*x(3) + 960*x(6)*x(7)*x(8)*x(3)
+ 1920*x(5)*x(6)*x(7)*x(8)*x(3)5760*x(6)**2*x(7)*x(8)*x(3) + 480*x(7)**2*x(8)*x(3)
+ 1920*x(5)*x(7)**2*x(8)*x(3)5760*x(6)*x(7)**2*x(8)*x(3) - 1920*x(7)**3*x(8)*x(3)
- 960*x(6)**2*x(8)**2*x(3)1920*x(6)*x(7)*x(8)**2*x(3) - 960*x(7)**2*x(8)**2
*x(3) - 720*x(5)*x(9)*x(3) - 480*x(5)**2*x(9)*x(3) +
720*x(6)*x(9)*x(3) + 960*x(5)**2*x(6)*x(9)*x(3) +
480*x(6)**2*x(9)*x(3) + 960*x(5)*x(6)**2*x(9)*x(3)1920*x(6)**3*x(9)*x(3) + 720*x(7)*x(9)*x(3) +
960*x(5)*x(7)*x(9)*x(3) + 1920*x(5)**2*x(7)*x(9)*x(3)
+ 960*x(6)*x(7)*x(9)*x(3) + 3840*x(5)*x(6)*x(7)*x(9)
*x(3)- 5760*x(6)**2*x(7)*x(9)*x(3) +
480*x(7)**2*x(9)*x(3) + 2880*x(5)*x(7)**2*x(9)*x(3)5760*x(6)*x(7)**2*x(9)*x(3) 1920*x(7)**3*x(9)*x(3) + 1920*x(5)*x(6)*x(8)*x(9)
*x(3)- 1920*x(6)**2*x(8)*x(9)*x(3) +
1920*x(5)*x(7)*x(8)*x(9)*x(3)- 3840*x(6)*x(7)*x(8)
*x(9)*x(3)- 1920*x(7)**2*x(8)*x(9)*x(3)960*x(5)**2*x(9)**2*x(3) + 1920*x(5)*x(6)*x(9)**2*x(3)
- 960*x(6)**2*x(9)**2*x(3) +
1920*x(5)*x(7)*x(9)**2*x(3) - 1920*x(6)*x(7)*x(9)**2
*x(3)- 960*x(7)**2*x(9)**2*x(3)- 855*x(4)840*x(6)*x(4)- 480*x(6)**2*x(4)- 840*x(7)*x(4) 480*x(5)*x(7)*x(4)- 480*x(5)**2*x(7)*x(4)960*x(6)*x(7)*x(4)- 960*x(5)*x(6)*x(7)*x(4)480*x(7)**2*x(4)- 960*x(5)*x(7)**2*x(4) +
1680*x(6)*x(8)*x(4) + 1440*x(6)**2*x(8)*x(4)
1680*x(7)*x(8)*x(4) + 960*x(5)*x(7)*x(8)*x(4) +
960*x(5)**2*x(7)*x(8)*x(4) + 2880*x(6)*x(7)*x(8)*x(4)
+ 1920*x(5)*x(6)*x(7)*x(8)*x(4) +
1440*x(7)**2*x(8)*x(4) + 1920*x(5)*x(7)**2*x(8)*x(4)
- 960*x(6)**2*x(8)**2*x(4) 1920*x(6)*x(7)*x(8)**2*x(4)- 960*x(7)**2*x(8)**2*x(4)
- 1680*x(5)*x(9)*x(4)- 480*x(5)**2*x(9)*x(4) +
1680*x(6)*x(9)*x(4)- 960*x(5)*x(6)*x(9)*x(4) +
1440*x(6)**2*x(9)*x(4) + 1680*x(7)*x(9)*x(4) +
960*x(5)**2*x(7)*x(9)*x(4) + 2880*x(6)*x(7)*x(9)*x(4)
+ 1920*x(5)*x(6)*x(7)*x(9)*x(4) +
1440*x(7)**2*x(9)*x(4) + 1920*x(5)*x(7)**2*x(9)*x(4)
+ 1920*x(5)*x(6)*x(8)*x(9)*x(4)1920*x(6)**2*x(8)*x(9)*x(4) + 1920*x(5)*x(7)*x(8)
*x(9)*x(4)- 3840*x(6)*x(7)*x(8)*x(9)*x(4)1920*x(7)**2*x(8)*x(9)*x(4)- 960*x(5)**2*x(9)**2
292


\[ 8.3.8 - ix = 0 \]
\[ h^3 f^3 f_y f_{yy} = 128 - 240 x(5)**4 x(1) - 240 x(5)**4 x(2) + \\
480 x(5)**2 x(6)**2 x(2) - 240 x(6)**4 x(2) - \\
960 x(6)**3 x(7)**4 x(2) - 240 x(6)**4 x(2) - \\
1440 x(6)**2 x(7)**2 x(2) - 960 x(6)**3 x(7)**3 x(2) - \\
240 x(7)**4 x(2) - 15 x(3) - 120 x(6)**2 x(3) - \\
240 x(6)**4 x(3) - 240 x(6)**2 x(7)**3 x(3) - \\
960 x(5)**2 x(6)**3 x(7)**3 x(3) - \\
120 x(7)**2 x(3) - 960 x(6)**3 x(7)**3 x(3) - \\
1440 x(6)**2 x(7)**2 x(3) - 960 x(6)**3 x(7)**3 x(3) - \\
240 x(7)**4 x(3) + 480 x(6)**2 x(8)**2 x(3) + \\
960 x(6)**2 x(7)**2 x(3) + 480 x(7)**2 x(8)**2 x(3) - \\
480 x(5)**2 x(9)**2 x(3) + 480 x(6)**2 x(9)**2 x(3) + \\
960 x(6)**2 x(7)**2 x(3) + 480 x(7)**2 x(9)**2 x(3) - \\
375 x(4) - 240 x(6)**2 x(7)**2 x(4) - 480 x(6)**2 x(7)**2 x(4) - \\
240 x(7)**2 x(4) + 480 x(6)**2 x(8)**2 x(4) + \\
960 x(6)**2 x(8)**2 x(4) + 480 x(7)**2 x(8)**2 x(4) - \\
480 x(5)**2 x(9)**2 x(4) + 480 x(6)**2 x(9)**2 x(4) + \\
960 x(6)**2 x(7)**2 x(4) + 480 x(7)**2 x(9)**2 x(4) + \\
240 x(10)*x(4) + 240 x(11)*x(4) - 960 x(5)**2 x(11) x(4) - \\
g x(4) + 240 x(12)**2 x(4) - 960 x(6)**2 x(12)**2 x(4) - \\
1920 x(6)**2 x(7)**2 x(12)**2 x(4) = 0 \\
(8.3.8 - x) \]

\[ h^3 f^3 f_y f_{yy} = 56 - 80 x(5)**4 x(1) - 80 x(5)**4 x(2) + 80 x(5) \\
**x(6)**3 x(6)**3 x(2) + 80 x(5)**3 x(6)**3 x(2) - 80 x(6)**4 x(2) - \\
-320 x(6)**3 x(7)**4 x(2) - 240 x(5)**x(6)**4 x(2) - \\
-480 x(6)**2 x(7)**2 x(2) - 160 x(5)**x(7)**3 x(2) - \\
-320 x(6)**2 x(7)**2 x(2) - 80 x(6)**4 x(2) - 5 x(3) - \\
20 x(6)**x(3) - 80 x(6)**4 x(3) - 20 x(7)**3 x(3) - \\
80 x(5)**3 x(7)**4 x(3) - 240 x(5)**x(6)**4 x(3) - \\
-320 x(6)**3 x(7)**4 x(3) - 480 x(5)**x(6)**4 x(3) - \\
-480 x(6)**2 x(7)**2 x(3) - 240 x(5)**x(6)**4 x(3) - \\
-320 x(6)**2 x(7)**2 x(3) - 80 x(6)**4 x(3) + \\
60 x(6)**x(8)**4 x(3) + 80 x(6)**3 x(8)**4 x(3) + \\
60 x(7)**3 x(7)**4 x(3) + 240 x(6)**2 x(7)**4 x(3) + \\
240 x(6)**2 x(7)**4 x(3) + 80 x(7)**3 x(8)**4 x(3) + \\
60 x(5)**x(9)**4 x(3) - 80 x(5)**3 x(9)**4 x(3) + \\
60 x(6)**x(9)**4 x(3) + 80 x(6)**3 x(9)**4 x(3) + \\
60 x(7)**3 x(9)**4 x(3) + 240 x(6)**2 x(7)**4 x(3) + \\
240 x(6)**2 x(7)**4 x(3) + 80 x(7)**3 x(8)**4 x(3) + \\
60 x(5)**x(9)**4 x(3) - 80 x(5)**3 x(9)**4 x(3) + \\
60 x(6)**x(9)**4 x(3) + 80 x(6)**3 x(9)**4 x(3) + \\
60 x(7)**3 x(9)**4 x(3) + 240 x(6)**2 x(7)**4 x(3) + \\
240 x(6)**2 x(7)**4 x(3) + 80 x(7)**3 x(8)**4 x(3) + \\
60 x(5)**x(9)**4 x(3) - 80 x(5)**3 x(9)**4 x(3) + \\
60 x(6)**x(9)**4 x(3) + 80 x(6)**3 x(9)**4 x(3) + \\
60 x(7)**3 x(9)**4 x(3) + 240 x(6)**2 x(7)**4 x(3) + \\
240 x(6)**2 x(7)**4 x(3) + 80 x(7)**3 x(8)**4 x(3) + 
(8.3.8 - x) \]
\[ x(4) - 240*x(5)*x(11)*x(4) - 80*x(5)**3*x(11)*x(4) + 130*x(12)*x(4) - 240*x(6)*x(12)*x(4) - 80*x(6)**3*x(12)*x(4) - 240*x(6)**2*x(7)*x(12)*x(4) - 240*x(6)**2*x(7)*x(12)*x(4) - 240*x(6)**2*x(7)*x(12)*x(4) - 240*x(6)**2*x(7)*x(12)*x(4) = 0 \]  

\[ h^T f_{y_2} : 32 - 80*x(5)**4*x(1) - 80*x(5)**4*x(2) - 80*x(6)**4*x(2) - 80*x(6)**4*x(2) - 320*x(6)**3*x(7)*x(2) - 480*x(6)**2*x(7)**2*x(2) - 320*x(6)*x(7)**3*x(2) - 80*x(7)**4*x(2) - 5*x(3) - 80*x(6)**4*x(3) - 320*x(6)**3*x(7)*x(3) - 480*x(6)**2*x(7)**2*x(3) - 320*x(6)*x(7)**3*x(3) - 80*x(7)**4*x(3) - 320*x(6)**4*x(3) - 85*x(4) = 0 \]  

where \( w_1 = x(1), w_2 = x(2), w_3 = x(3), w_4 = x(4), a_1 = x(5), a_2 = x(6) \)
\( a_3 = x(7), a_4 = x(8), a_5 = x(9), a_6 = x(10), a_7 = x(11), a_8 = x(12). \)

Similarly, equations (8.3.8-i)-(8.3.8-xii) are solved simultaneously by using the NAG routine (Subroutine C05NBF) for solving a system of non-linear equations to give the required parameters, i.e.,

\[
\begin{align*}
w_1 &= -0.1773157366, & w_2 &= 1.0254553152, & w_3 &= -0.0779114700 \\
 w_4 &= 0.2297718914, & a_1 &= 0.1017275411, & a_2 &= -0.5236574475 \\
 a_3 &= 1.1653361910, & a_4 &= 4.7450804540, & a_5 &= -4.2354437705 \\
a_6 &= -0.5736403905, & a_7 &= 0.9301175162, & a_8 &= 0.4667978567
\end{align*}
\]

Thus, the new fifth order \(C_M^5\) WRK method can be written as follows:

\[
y_{y_1} = y_1 + h \left[ -0.1773157366 \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) + 1.0254553152 \left( \frac{k_2^2 + k_3^2}{k_2 + k_3} \right) \\
- 0.0779114700 \left( \frac{k_2^2 + k_3^2}{k_2 + k_3} \right) + 0.2297718914 \left( \frac{k_2^2 + k_3^2}{k_2 + k_3} \right) \right] \]  

where \( k_1 = f(y_1) \)
\( k_2 = f(y_1 + 0.1017275411hk) \)
\( k_3 = f(y_1 - 0.5236574475hk + 1.1653361910hk) \)
\( k_4 = f(y_1 + 4.7450804540hk - 4.2354437705hk + 0.0096366835hk) \)
\( k_5 = f(y_1 - 0.5736403905hk + 0.9301175162hk + 0.4667978567hk \) 
\[ + 0.1767250176hk \) . \]  

... (8.3.9)
By the use of Mathematica to rationalize the coefficients in equation (8.3.9), we obtain

\[
\begin{align*}
    w_1 &= -\frac{8639}{48721}, \quad w_2 = \frac{135638}{132271}, \quad w_3 = -\frac{2480}{31831}, \quad w_4 = \frac{22241}{96796}, \\
    a_1 &= \frac{6236}{61301}, \quad a_2 = -\frac{162659}{310621}, \quad a_3 = \frac{225101}{193164}, \quad a_4 = \frac{814792}{171713}, \\
    a_5 &= -\frac{424527}{100232}, \quad a_6 = -\frac{27973}{48764}, \quad a_7 = \frac{78594}{84499}, \quad a_8 = \frac{54796}{117387}, \\
    \frac{1}{2} - a_4 - a_5 &= -\frac{1353}{140401}, \quad 1 - a_6 - a_7 - a_8 = \frac{26844}{151897}.
\end{align*}
\]

and the new fifth order contraharmonic WRK method can be written in rational form as

\[
y_{n+1} = y_n + h \left[ -\frac{8639}{48721} \left( \frac{k_1^2 + k_2^2}{k_1 + k_2} \right) + \frac{135638}{132271} \left( \frac{k_2^2 + k_3^2}{k_2 + k_3} \right) \\
    -\frac{2480}{31831} \left( \frac{k_3^2 + k_4^2}{k_3 + k_4} \right) + \frac{22241}{96796} \left( \frac{k_4^2 + k_5^2}{k_4 + k_5} \right) \right],
\]

where \( k_1 = f(y_n) \)

\[
\begin{align*}
    k_2 &= f \left( y_n + \frac{6236}{61301} hk_1 \right), \\
    k_3 &= f \left( y_n - \frac{162659}{310621} hk_1 + \frac{225101}{193164} hk_2 \right), \\
    k_4 &= f \left( y_n + \frac{814792}{171713} hk_1 - \frac{424527}{100232} hk_2 - \frac{1353}{140401} hk_3 \right), \\
    k_5 &= f \left( y_n - \frac{27973}{48764} hk_1 + \frac{78594}{84499} hk_2 + \frac{54796}{117387} hk_3 + \frac{26844}{151897} hk_4 \right).
\end{align*}
\]

**8.3.3 Error Analysis Fifth Order \(C_sM\) Method**

By substituting the values of \( a_i, 1 \leq i \leq 8 \) and \( w_i, 1 \leq i \leq 4 \) in (8.3.9) into (8.3.6) and (8.3.7) using Mathematica and evaluating all the terms up to \((h^4)\) to
represent the local truncation error (LTE) for this method we have

\[ \text{LTE} = h^4 \left[ 0.0132485733 f_{n}^2 + 0.0202501069 f_{n}^2 f_{w} + 0.0095106268 f_{n}^2 f_{w}^2 - 0.0022879188 f_{n}^2 f_{w}^3 - 0.0001379536 f_{n}^4 f_{w}^3 - 0.0003448339 f_{n}^4 f_{w}^4 - 0.000178190 f_{n}^4 f_{w}^5 \right] \]

(8.3.13)

8.3.4. Stability Analysis Fifth Order \(C_{5}M\) Method

We examine the stability region for the fifth order contraharmonic mean WRK method with the test equation \(y' = \lambda y\), and we obtain

\[
\begin{align*}
k_1 &= \lambda y_n \\
k_2 &= \lambda(y_n + 0.1017275411hk_1) \\
k_3 &= \lambda(y_n - 0.5236574475hk_1 + 1.1653361910hk_2) \\
k_4 &= \lambda(y_n + 4.7450804540hk_1 - 4.2354437705hk_2 - 0.0096366835hk_3) \\
k_5 &= \lambda(y_n - 0.5736403905hk_1 + 0.9301175162hk_2 + 0.4667978567hk_3 + 0.1767250176hk_4).
\end{align*}
\]

(8.3.14)

By substituting \(k_{1,1}i\leq i\leq 5\) in equation (8.3.14) and \(w_{1,1}i\leq i\leq 4\) in equation (8.3.9) into the fifth order \(C_{5}M\) WRK formula, i.e.,

\[
y_{n+1} = y_n + h \left[ \sum_{i=1}^{4} w_i \left( \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right) \right]
\]

(8.3.15)

we obtain

\[
y_{n+1} = y_n + (h\lambda) y_n + 0.5(h\lambda)^2 y_n + 0.1666667(h\lambda)^3 y_n + 0.0416667(h\lambda)^4 y_n + 0.008333333(h\lambda)^5 y_n - 0.0118597(h\lambda)^6 y_n + 0(h^7) \ldots
\]

(8.3.16)

By rationalizing the coefficients in equation (8.3.16) we have

\[
y_{n+1} = y_n + (h\lambda) y_n + \frac{1}{2} (h\lambda)^2 y_n + \frac{1}{6} (h\lambda)^3 y_n + \frac{1}{24} (h\lambda)^4 y_n + \frac{1}{120} (h\lambda)^5 y_n - \frac{47}{3963} (h\lambda)^6 y_n + 0(h^7) \ldots
\]

(8.3.17)
By substituting $hA = z$ in (8.3.17), we can show that

$$y_{n+1} = y_n + y_n\left[ z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} - \frac{47}{3963} z^6 \right] + O(z^7). \quad (8.3.18)$$

Following equation (3.3.30), we write $\frac{y_{n+1}}{y_n} = Q$ in the equation (8.3.18), to obtain

$$Q = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 + \frac{1}{120} z^5 - \frac{47}{3963} z^6 + O(z^7). \quad (8.3.19)$$

To determine the stability region of the fifth order $C_\infty M \ WRK$ formula in the complex plane that satisfy the condition

$$\left| \frac{y_{n+1}}{y_n} \right| = |Q| < 1$$

i.e.,

$$\left| 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} - \frac{47}{3963} z^6 \right| < 1 \quad (8.3.20)$$

by the use of Mathematica, we can plot the graphic surface defined by equation (8.3.20) i.e.,

![Figure 8.3: Graphic surface defined by the fifth order $C_\infty M \ WRK$ formula](image)

and the stability region defined by the formula in equation (8.3.20) as shown in Figure 8.4.
8.3.5. Numerical Example

We consider the standard test IVP

\[ y' = y, \quad y(0) = 1, \quad 0 \leq x \leq 1 \] (8.3.21)

where the exact solution is \( y(x) = \exp(x) \). The absolute errors in the numerical solution using formula (8.3.6)-(8.3.7) compared with the fourth order WRK formula (classical formula), RK4(5)-Merson [1957], GM-RK5(5) [1995] and WRK-RK5(5) [1996] methods are shown in Table 8.4.

Table 8.4: Absolute errors by various formula for solving equation (8.3.21)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>.847423E-07</td>
<td>.152979E-07</td>
<td>.110389E-07</td>
<td>.142041E-08</td>
<td>.801866E-08</td>
</tr>
<tr>
<td>0.2</td>
<td>.187310E-06</td>
<td>.338135E-07</td>
<td>.243998E-07</td>
<td>.313960E-08</td>
<td>.177240E-07</td>
</tr>
<tr>
<td>0.3</td>
<td>.310514E-06</td>
<td>.560546E-07</td>
<td>.404490E-07</td>
<td>.520468E-08</td>
<td>.293821E-07</td>
</tr>
<tr>
<td>0.4</td>
<td>.457561E-06</td>
<td>.825999E-07</td>
<td>.596040E-07</td>
<td>.766942E-08</td>
<td>.432963E-07</td>
</tr>
<tr>
<td>0.5</td>
<td>.632103E-06</td>
<td>.114109E-06</td>
<td>.105950E-07</td>
<td>.140512E-07</td>
<td>.598122E-07</td>
</tr>
<tr>
<td>0.6</td>
<td>.838299E-06</td>
<td>.151332E-06</td>
<td>.109201E-06</td>
<td>.140512E-07</td>
<td>.793233E-07</td>
</tr>
<tr>
<td>0.7</td>
<td>.108087E-05</td>
<td>.195122E-06</td>
<td>.140799E-06</td>
<td>.181171E-07</td>
<td>.102277E-06</td>
</tr>
<tr>
<td>0.8</td>
<td>.136520E-05</td>
<td>.246449E-06</td>
<td>.177837E-06</td>
<td>.228829E-07</td>
<td>.129181E-06</td>
</tr>
<tr>
<td>0.9</td>
<td>.169738E-05</td>
<td>.306414E-06</td>
<td>.221108E-06</td>
<td>.284507E-07</td>
<td>.160613E-06</td>
</tr>
<tr>
<td>1.0</td>
<td>.208432E-05</td>
<td>.376267E-06</td>
<td>.271514E-06</td>
<td>.349365E-07</td>
<td>.197227E-06</td>
</tr>
</tbody>
</table>
From Table 8.4, we can see that the accuracy obtained from using CoM-RK5(5) is better than the AM-RK4, RK4(5)-Merson and GM-RK5(5) methods. When we make a work comparison with the fifth order RK5(6)-Nystrom method, then the new fifth order method saves one function evaluation. From the above discussion we can conclude that a fifth order contraharmonic mean weighted Runge-Kutta method with five stages does exist.

8.4 THE THEORY OF WEIGHTED RK(5,5) METHOD

In Evans and Yaakub [1995], a new method called the RK(4,4) was introduced using two different RK methods but of the same order p. The difference between these two approximations is taken to obtain an estimate of their accuracy. The RK(4,4) method is based on the use of the fourth order classical Runge-Kutta method and the fourth order contraharmonic mean \((C,M)\) method (see Evans and Yaakub [1995]). Now, we establish a new weighted RK(5,5) strategy where we extend the RK(4,4) method by using the fifth order RK methods. This approach is based on the use of the new fifth order arithmetic mean (AM) weighted Runge-Kutta method (Evans and Yaakub [1996]) and the fifth order contraharmonic mean \((C,M)\) weighted Runge-Kutta method (Evans and Yaakub [1995]). The combination of these two formula will be denoted as the RK(5,5) method.

8.4.1 RK(5,5) Method For Error Estimate and Error Control

The combination of the fifth order arithmetic mean (AM) weighted Runge-Kutta formula

\[
y_{AM} = y_n + h \left[ \sum_{i=1}^{4} w_i \left( \frac{k_i + k_{i+1}}{2} \right) \right]
\]  

(8.4.1)

where \(\sum w_i = 1\), \(w_1 = 0.2615038147\), \(w_2 = -0.2765809214\), \(w_3 = 0.5947141647\), \(w_4 = 0.4203629420\)
\[
\begin{align*}
k_1 &= f(y_n) \\
k_2 &= f(y_n + 1.5471214403hk_i) \\
k_3 &= f(y_n + 0.1756458393hk_i + 0.1243059001hk_j) \\
k_4 &= f(y_n + 0.1009316694hk_i + 0.1100539630hk_k + 0.2890143692hk_l) \\
k_5 &= f(y_n + 0.9997431862hk_i - 0.0928890403hk_j - 0.6201812828hk_k + 0.7133271396hk_l)
\end{align*}
\]

(8.4.2)

and the fifth order contraharmonic mean \((C_M)\) formula in the form

\[
y_{C_M} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \tag{8.4.3}
\]

where \(\sum w_i = 1\), \(\ w_1 = -0.1773157366\), \(\ w_2 = 1.0254553152\)

\(\ w_3 = -0.0779114700\), \(\ w_4 = 0.2297718914\)

\(k_1 = f(y_n)\)

\(k_2 = f(y_n + 0.1017275411hk_i)\)

\(k_3 = f(y_n - 0.5236574475hk_i + 1.1653361910hk_j)\)

\(k_4 = f(y_n + 4.7450804540hk_i - 4.2354437705hk_k - 0.0096366835hk_l)\)

\(k_5 = f(y_n - 0.5736403905hk_i + 0.9301175162hk_k + 0.4667978567hk_l + 0.1767250176hk_j)\)  

(8.4.4)

is called RK(5,5) method. The difference between equation (8.4.1) and (8.4.3), i.e., \(|y_{AM} - y_{C_M}|\) provides an error estimate for the approximation to the numerical solution.

By using the same procedure as in the RK(4,4) method, we can also obtain an error estimate for the five stage explicit \(AM-C_M\) method of order five by implementing the local truncation error for the fifth order arithmetic mean Runge-Kutta method and fifth order contraharmonic mean method.

For the fifth order arithmetic mean Runge-Kutta method, we have
\[ y_{n+1}^{AM} = y_n + LTE^{AM} \quad (8.4.5) \]

and for the contraharmonic mean method

\[ y_{n+1}^{CM} = y_n + LTE^{CM} \quad (8.4.6) \]

where \( y_{n+1}^{AM} \) and \( y_{n+1}^{CM} \) are the numerical approximations at \( x_{n+1} \), obtained by the arithmetic mean and contraharmonic mean methods respectively and \( LTE^{AM} \) and \( LTE^{CM} \) are the local truncation errors of the fifth order arithmetic mean Runge-Kutta method and the fifth order contraharmonic mean methods.

An error estimate is obtained by taking the difference between these two methods for the numerical approximations at \( x_{n+1} \) by

\[ y_{n+1}^{AM} - y_{n+1}^{CM} = LTE^{AM} - LTE^{CM} \quad (8.4.7) \]

The local truncation error for the fifth order arithmetic mean Runge-Kutta method involves \( y \) derivatives given by

\[
LTE^{AM} = h^5 \left[ \frac{1}{720} f f_y^5 + 0.0018022816 f^2 f_y^3 f_y - 0.0166861138 f^3 f_y f_y^2 \\
+ 0.0082646021 f^3 f_y^2 f_y f_y + 0.0041171137 f^4 f_y f_y^2 f_y \\
- 0.0023096163 f^4 f_y f_y f_y f_y + 0.0000588245 f^5 f_y f_y f_y f_y \right]
\]

... (8.4.8)

while the local truncation error for the contraharmonic mean method is given by

\[
LTE^{CM} = h^5 \left[ 0.0132485733 f f_y^5 + 0.0202501069 f^2 f_y^3 f_y \\
+ 0.0095106268 f^3 f_y^2 f_y - 0.0022879188 f^4 f_y f_y^2 f_y \\
- 0.000179536 f^4 f_y f_y^2 f_y f_y - 0.0003448339 f^4 f_y f_y f_y \\
- 0.0000178190 f^5 f_y f_y f_y f_y \right]
\]

... (8.4.9)
The absolute difference between $LTE^{al}$ and $LTE^{cm}$ is given by

$$|LTE^{al} - LTE^{cm}| = h\left[\frac{1}{720} - 0.0132485733\right]f_r^2 + (0.0018022816 - 0.0202501069)f_r^3 f_{rr}$$
$$+ (-0.0166861138 - 0.0095106268)f_r^3 f_{rr}^2$$
$$+ (0.0082646021 + 0.0022879188)f_r^4 f_{rr}$$
$$+ (0.004171137 + 0.0001379536)f_r^4 f_{rrr}$$
$$+ (-0.0023096163 + 0.0003448339)f_r^4 f_{rrrr}$$
$$+ (0.0000588245 + 0.0000178190)f_r^5 f_{yyyy}$$

$$= h\left[-0.0118597f_r^3 - 0.0184478f_r^3 f_{rr}^2ight.$$
$$+ (-0.0261967 + 0.0105525)f_r^3 f_{rr}^2 + 0.00425507 f_r^4 f_{rrr}$$
$$- 0.00196478 f_r^4 f_{rrrr}$$
$$+ 0.0000766435 f_r^5 f_{yyyy}\right] \ldots \quad \text{(8.4.10)}$$

Following Lotkin [1951], if the following bounds for $f$ and its partial derivatives hold for $x \in [a,b]$ and $y \in [-\infty, \infty]$ we have,

$$|f(x,y)| < Q \quad , \quad \left|\frac{\partial^{n+j}f(x,y)}{\partial x^i \partial y^j}\right| < \frac{P^{n+j}}{Q^{n+i}} \quad , \quad i + j \leq p \quad \text{(8.4.11)}$$

where $P$ and $Q$ are positive constants and $p$ is the order of the method. In this case, we have $p = 5$. Hence using (8.4.11), we have
\begin{align*}
|f f'_{n}| & < Q \left( \frac{P^{n+1}}{Q^{n+1}} \right)^{2} \\
|f' f_{n} f_{n}'| & < Q^{3} \left( \frac{P^{n}}{Q^{n}} \right)^{3} \\
|f' f_{n} f_{n}'| & < Q^{3} P^{3} \left( \frac{P^{3}}{Q^{3}} \right) \\
|f' f_{n} f_{n}'| & < Q^{3} P \left( \frac{P^{4}}{Q^{4}} \right) \\
|f' f_{n} f_{n}'| & < Q^{3} P \left( \frac{P^{5}}{Q^{5}} \right) \\
|f' f_{n} f_{n}'| & < Q^{3} P \left( \frac{P^{6}}{Q^{6}} \right)
\end{align*}

From the equations (8.4.9)-(8.4.12), we obtain

\begin{align*}
|LTE_{\text{adv}} - LTE_{\text{CM}}| & \leq 0.0435848 P^{3} Q h^{6} \quad (8.4.13) \\
\text{Hence,} & \\
|y^{\text{adv}}_{m} - y^{\text{CM}}_{m}| & \leq 0.0435848 P^{3} Q h^{6} \quad (8.4.14) \\
\text{or} & \\
|y^{\text{adv}}_{m} - y^{\text{CM}}_{m}| & \leq \frac{89}{2042} P^{3} Q h^{6}.
\end{align*}

By taking the tolerance as TOL, i.e., \( \varepsilon < 0.00005 \), then by setting

\begin{align*}
|y^{\text{adv}}_{m} - y^{\text{CM}}_{m}| & \leq TOL
\end{align*}

the error control and step size selection can be determined by (8.4.14) to give the formula as

\begin{align*}
0.0435848 P^{3} Q h^{6} < TOL
\end{align*}

or

\begin{align*}
\frac{TOL}{0.0435848 P^{3} Q} \frac{1}{h} \quad (8.4.15)
\end{align*}
8.4.2 Experimental Results For RK(5,5)

The following are the numerical results of testing the RK(5,5) method for error control on the sample problems:

**Problem 1**: \( y' + y = 0 \)
Initial condition: \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution: \( y = \exp(-x) \)

<table>
<thead>
<tr>
<th>x</th>
<th>Exact Solution</th>
<th>Numerical Solution</th>
<th>Absolute Error</th>
<th>Estimate Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h = .50000 )</td>
<td>( .6065104D+00 )</td>
<td>( .6065307D+00 )</td>
<td>( .2024324D-04 )</td>
<td>( .1283099D-04 )</td>
</tr>
<tr>
<td>( h = .25000 )</td>
<td>( .3678549D+00 )</td>
<td>( .3678794D+00 )</td>
<td>( .2455588D-04 )</td>
<td>( .1556049D-04 )</td>
</tr>
<tr>
<td>( h = .50000 )</td>
<td>( .2231078D+00 )</td>
<td>( .2231302D+00 )</td>
<td>( .2234047D-04 )</td>
<td>( .1415296D-04 )</td>
</tr>
<tr>
<td>( 2.00000 )</td>
<td>( .1353172D+00 )</td>
<td>( .1353353D+00 )</td>
<td>( .1806660D-04 )</td>
<td>( .1144245D-04 )</td>
</tr>
<tr>
<td>( 2.50000 )</td>
<td>( .8207130D-01 )</td>
<td>( .8208500D-01 )</td>
<td>( .1369721D-04 )</td>
<td>( .8672849D-05 )</td>
</tr>
<tr>
<td>( 3.00000 )</td>
<td>( .4977710D-01 )</td>
<td>( .4978070D-01 )</td>
<td>( .9969165D-05 )</td>
<td>( .5310677D-05 )</td>
</tr>
<tr>
<td>( 3.50000 )</td>
<td>( .3019033D-01 )</td>
<td>( .3019383D-01 )</td>
<td>( .7054254D-05 )</td>
<td>( .4464323D-05 )</td>
</tr>
<tr>
<td>( 4.00000 )</td>
<td>( .1831075D-01 )</td>
<td>( .1831564D-01 )</td>
<td>( .4889771D-05 )</td>
<td>( .3093716D-05 )</td>
</tr>
<tr>
<td>( 4.50000 )</td>
<td>( .1110566D-01 )</td>
<td>( .1110900D-01 )</td>
<td>( .3336465D-05 )</td>
<td>( .2110405D-05 )</td>
</tr>
<tr>
<td>( 5.00000 )</td>
<td>( .6735699D-02 )</td>
<td>( .6737947D-02 )</td>
<td>( .2248483D-05 )</td>
<td>( .1421858D-05 )</td>
</tr>
<tr>
<td>( 5.50000 )</td>
<td>( .4085271D-02 )</td>
<td>( .4086771D-02 )</td>
<td>( .1500126D-05 )</td>
<td>( .9483790D-06 )</td>
</tr>
<tr>
<td>( 6.00000 )</td>
<td>( .2477760D-02 )</td>
<td>( .2478752D-02 )</td>
<td>( .9925716D-06 )</td>
<td>( .6273406D-06 )</td>
</tr>
<tr>
<td>( h = 1.00000 )</td>
<td>( .9085119D-03 )</td>
<td>( .9118820D-03 )</td>
<td>( .3370110D-05 )</td>
<td>( .4554334D-05 )</td>
</tr>
<tr>
<td>( 7.00000 )</td>
<td>( .3331210D-03 )</td>
<td>( .3354626D-03 )</td>
<td>( .2341614D-05 )</td>
<td>( .3081334D-05 )</td>
</tr>
<tr>
<td>( 8.00000 )</td>
<td>( .1221444D-03 )</td>
<td>( .1234098D-03 )</td>
<td>( .1265432D-05 )</td>
<td>( .1590943D-05 )</td>
</tr>
<tr>
<td>( 9.00000 )</td>
<td>( .4478627D-04 )</td>
<td>( .4539993D-04 )</td>
<td>( .6136600D-06 )</td>
<td>( .7335390D-06 )</td>
</tr>
</tbody>
</table>

The following is a list of sample problems used in the numerical experiments. The notation NPB defines the number of problem solution. The comparison of the time taken and accuracy between the RK(4,4) (see Evans and Yaakub [1995]) and RKF(4,5) methods are shown in Tables 8.5 and 8.6.

**Problem 2 (NPB 7)**: \( y' + y - x - 1 = 0 \)
Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution: \( y = x + \exp(-x) \)
Problem 3 (NPB 10): \( y' - x^2 \sin(x) + \frac{1}{x} + 1 = 0 \)

Initial conditions: \( x_0 = 1 \), \( y_0 = 4 \)

Exact solution: \( y = -x - \log(x) + x^2 \cos(x) - 2x \sin(x) - 2 \cos(x) + C \)

Problem 4 (NPB 12): \( y' + \ln(x^2) = 0 \)

Initial conditions: \( x_0 = 1 \), \( y_0 = 2 \)

Exact solution: \( y = -2(x \ln(x) - x) \)

Problem 5 (NPB 4): \( Y' - Y = 0 \)

Initial conditions: \( x_0 = 0 \), \( y_0 = 1 \)

Exact solution: \( y = \exp(x) \)

In the new RK(5,5) method with error control strategy, we use the error estimate as the difference between the fifth order AM Runge-Kutta method and the fifth order contraharmonic mean method. These error estimates \( \text{ERREST} \) used together with a constant derived in equation (8.4.14 - 8.4.15) are in the form

\[
\text{ERREST} = \left| Y_{av} - Y_{c,M} \right| \times \frac{89}{2042} \tag{8.4.16}
\]

By using the error estimate in equation (8.4.16), the comparison of the time taken and the accuracy between solutions from the RK(5,5), RK(4,4), RKF(4,5) and RK4(5)-Merson methods are shown in Table 8.5 and Table 8.6.

Table 8.5

<table>
<thead>
<tr>
<th>Problem</th>
<th>Time Taken</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
</tr>
<tr>
<td>1</td>
<td>1.80</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
</tr>
<tr>
<td>3</td>
<td>0.26</td>
</tr>
<tr>
<td>4</td>
<td>0.22</td>
</tr>
<tr>
<td>5</td>
<td>0.12</td>
</tr>
</tbody>
</table>
Table 8.6

<table>
<thead>
<tr>
<th>Problem</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RK(4,4)</td>
</tr>
<tr>
<td>1</td>
<td>0.2258x10^-3</td>
</tr>
<tr>
<td>2</td>
<td>0.8308x10^-4</td>
</tr>
<tr>
<td>3</td>
<td>0.3253x10^-4</td>
</tr>
<tr>
<td>4</td>
<td>0.2949x10^-4</td>
</tr>
<tr>
<td>5</td>
<td>0.5516x10^-4</td>
</tr>
</tbody>
</table>

From Table 8.5, we can see that the solution for problems 1-5 by RK(5,5) and RKF(4,5) performed faster than the solution by Merson and RK(4,4) methods. But in Table 8.6, the accuracy for problems 1, 2, 3 and 4 of the RK(5,5) is more accurate compared to RKF(4,5), RK(4,4) and Merson. However by reducing the step size to a certain value, i.e., $\frac{h}{2}$ and $\frac{h}{4}$ the solution by the RK(5,5), RK(4,4), Merson and RKF(4,5) methods are comparable in terms of the time taken and the accuracy.

8.5 WEIGHTED FIFTH-ORDER RUNGE-KUTTA FORMULAS FOR SECOND-ORDER DIFFERENTIAL EQUATIONS

Many publications, i.e., Fatunla [1995], Evans & Jayes [1993], Jeltsch [1978] have focused their attention on the second order initial value problem (IVP)

$$y'' = f(x, y), \quad y(x_0) = y_0, \quad y'(x_0) = \eta. \quad (8.5.1)$$

This type of special second order IVP often occurs in the mathematical modelling of mechanical systems without dissipation (Henrici, [1962]). However, interestingly very little attention has been given to the second order initial value problem (see Fatunla [1995]) in the form

$$y'' = f(x, y, y'), \quad y(x_0) = y_0, \quad y'(x_0) = \eta. \quad (8.5.2)$$
In Evans and Yaakub [1996], a new fifth order five stage arithmetic mean (AM) weighted Runge-Kutta (WRK) method was presented in the form

\[ y_{n+1} = y_n + h \left[ \sum_{i=1}^{4} w_i \left( \frac{k_i + k_{i+1}}{2} \right) \right] \]  
(8.5.3)

where \( \sum_{i=1}^{4} w_i = 1 \)

\[ k_1 = f(y_n) \]

\[ k_2 = f(y_n + a_1 h k_1) \]

\[ k_3 = f(y_n + a_2 h k_1 + a_3 h k_2) \]

\[ k_4 = f(y_n + a_4 h k_1 + a_5 h k_2 + (\frac{1}{2} - a_4 - a_5) h k_3) \]

\[ k_5 = f(y_n + a_6 h k_1 + a_7 h k_2 + a_8 h k_3 + (1 - a_6 - a_7 - a_8) h k_4) \]  
(8.5.4)

where fifth order accuracy is obtained by choosing, e.g.,

\[ w_1 = 0.2615038147 \quad w_2 = -0.2765809214 \quad w_3 = 0.5947141647 \]
\[ w_4 = 0.4203629420 \quad a_1 = 1.5471214403 \quad a_2 = 0.1756458393 \]
\[ a_3 = 0.1243059001 \quad a_4 = 0.1100539634 \quad a_5 = 0.1009316694 \]
\[ a_6 = 0.9997431862 \quad a_7 = -0.0928890403 \quad a_8 = -0.6201812828 \]
\[ a_{11} = 0.5 - a_4 - a_5 = 0.2890143692 \quad a_{22} = 1 - a_6 - a_7 - a_8 = 0.7133271396 \]

Thus, the new fifth order arithmetic mean WRK method can be written as follows:

\[ y_{n+1} = y_n + h \left[ 0.2615038147 \left( \frac{k_1 + k_2}{2} \right) - 0.2765809214 \left( \frac{k_2 + k_3}{2} \right) + 0.5947141647 \left( \frac{k_3 + k_4}{2} \right) + 0.4203629420 \left( \frac{k_4 + k_5}{2} \right) \right] \]

(8.5.5)

where \( k_1 = f(y_n) \)

\[ k_2 = f(y_n + 1.5471214403 h k_1) \]

\[ k_3 = f(y_n + 0.1756458393 h k_1 + 0.1243059001 h k_2) \]
\[ k_4 = f\left(y_n + 0.1009316694hk_1 + 0.1100539630hk_2 + 0.2890143692hk_3 \right) \]

\[ k_5 = f\left(y_n + 0.9997431862hk_1 - 0.0928890403hk_2 - 0.6201812828hk_3 + 0.7133271396hk_4 \right) \] ...

(8.5.6)

In this section, we propose using the new fifth order arithmetic mean (AM) weighted in equations (8.5.5) and (8.5.6) for solving the second order differential equation in initial value problem (8.5.2).

8.5.1 Weighted Fifth-Order Formula For a System of Two First-Order Equations

Now by using the formula in equations (8.5.5) and (8.5.6), we have developed a new strategy for solving a system of two first-order equations with initial values in the form

\[ y' = F(x, y, v) , \quad y(x_0) = \alpha \]

\[ v' = G(x, y, v) , \quad v(x_0) = \beta \]

(8.5.7)

Following Greenspan et al [1988], since there are two equations we must have two sets of k's. For notational simplicity, we take instead a set of k's and a set of m's as follows:

\[ k_1 = F(x_n, y_n, v_n) \]

\[ k_2 = F(x_n + a_1h, y_n + a_1hk_1, v_n + a_1hm_1) \]

\[ k_3 = F(x_n + (a_2 + a_3)h, y_n + a_2hk_1 + a_2hk_2, v_n + a_2hm_1 + a_2hm_2) \]

\[ k_4 = F(x_n + (a_4 + a_5 + a_6 + a_{11})h, y_n + a_4hk_1 + a_4hk_2 + a_{11}hk_3, v_n + a_4hm_1 + a_4hm_2 + a_{11}hm_3) \]

\[ k_5 = F(x_n + (a_6 + a_7 + a_8 + a_{12})h, y_n + a_6hk_1 + a_6hk_2 + a_8hk_3 + a_{12}hk_4, v_n + a_6hm_1 + a_6hm_2 + a_8hm_3 + a_{12}hm_4) \]

(8.5.8)

and

\[ m_1 = G(x_n, y_n, v_n) \]

\[ m_2 = G(x_n + a_1h, y_n + a_1hk_1, v_n + a_1hm_1) \]
8.5.2 Weighted Fifth-Order Formulas for Second-Order Differential Equations

We consider initial value problems in the form

\[ y'' = f(x, y, y') \]

\[ y(x_0) = \alpha, \quad y'(x_0) = \beta \]  

(8.5.12)

If let \( y' = v \), the equation (8.5.12) is equivalent to the system

\[ y' = v \]

\[ y(x_0) = \alpha \]

\[ v' = f(x, y, v) \]

\[ v(x_0) = \beta \]  

(8.5.13)

Now the system of equation in (8.5.13) can be written similarly to equation (8.5.7) as

\[
\begin{align*}
    y' &= F(x, y, v) = v \\
    v' &= G(x, y, v) = f(x, y, v)
\end{align*}
\]  

(8.5.14)

Therefore, a set of \( k \)'s and \( m \)'s in (8.5.8) and (8.5.9) can be written in the form

\[
k_1 = v_n \\
k_2 = v_n + a_i h_n \\
k_3 = v_n + a_i h_n + a_i h_m \\
k_4 = v_n + a_i h_n + a_i h_m + a_i h_m \\
k_5 = v_n + a_i h_n + a_i h_m + a_i h_m + a_i h_m
\]  

(8.5.15)
By substituting equations (8.5.14)-(8.5.15) into equation (8.5.9), we obtain
\[ m_1 = f(x_*, y_*, v_*) \]
\[ m_2 = f(x_*, a_1h, y_*, a_1hv_*, v_* + a_1hm_1) \]
\[ m_3 = f(x_*, (a_2 + a_3)h, y_*, a_2h, y_*, a_2hv_*, a_2a_3h^2m_1, v_* + a_2hm_1 + a_2hm_2) \]
\[ m_4 = f(x_*, (a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} + a_{11})h, y_*, (a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} + a_{11} + a_{12})hv_*, \]
\[ (a_5a_6 + a_7a_8 + a_9a_{10} + a_{11}a_{12})h^2m_1 + (a_5a_6 + a_7a_8 + a_9a_{10} + a_{11}a_{12})h^2m_2 + a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}h^2m_3, \]
\[ v_* + a_2hm_1 + a_2hm_2 + a_2h, m_1 + a_2hm_3 + a_3hm_5 \]
and then equation (8.5.10) and (8.5.11) can be written as
\[ y_{n+1} = y_n + \frac{\hbar}{2} \left[ w_1 \left( \frac{k_1 + k_2}{2} \right) + w_2 \left( \frac{k_3 + k_4}{2} \right) + \right. \]
\[ w_3 \left( \frac{k_5 + k_6}{2} \right) + w_4 \left( \frac{k_7 + k_8}{2} \right) \]
\[ = y_n + \frac{\hbar}{2} \left[ w_1 (2v_n + a_1hm_1) + w_2 (2v_n + (a_1 + a_2)hm_1 + a_2hm_2) + \right. \]
\[ w_3 (2v_n + (a_2 + a_3)hm_1 + (a_3 + a_4)hm_2 + a_3hm_3) + \]
\[ w_4 (2v_n + (a_4 + a_5)hm_1 + (a_5 + a_6)hm_2 + \]
\[ (a_5 + a_6)hm_3 + a_7hm_5 \]
\[ = y_n + \hbar \sum_{i=1}^{\infty} w_i v_* + \frac{\hbar^2}{2} \left[ [(a_1 + a_2)w_2 + (a_2 + a_3)w_3 + (a_3 + a_4)w_4]m_1, \right. \]
\[ +[(a_3 + a_4)w_3 + (a_4 + a_5)w_4]m_2 + [(a_4 + a_5)w_4]m_3 \]
\[ +w_1a_1m_1 + w_2a_2m_2 + w_3a_3m_3 + w_4a_4m_4 \}
\[ = y_n + \hbar v_* + \frac{\hbar^2}{2} \left[ [(a_1 + a_2)w_2 + (a_2 + a_3)w_3 + (a_3 + a_4)w_4]m_1, \right. \]
\[ +[(a_3 + a_4)w_3 + (a_4 + a_5)w_4]m_2 + [(a_4 + a_5)w_4]m_3 \]
\[ +w_1a_1m_1 + w_2a_2m_2 + w_3a_3m_3 + w_4a_4m_4 \}
\[ = 311 \]
\[ y_{n+1} = y_n + hv_n + \frac{h^2}{2} \left[ 0.5552610794m_1 + 0.1122119872m_2 + 0.0326706395m_3 + 0.2998562939m_4 \right] \]

or \[ y_{n+1} = y_n + hv_n + \frac{h^2}{2} \left[ \begin{array}{c} 31636 \ 56975 \ 81854 \ 99845 \ 189971 \end{array} \right] \left[ \begin{array}{c} m_1 \ 56975 \ 81854 \ 99845 \ 189971 \end{array} \right] \]

and \[ v_{n+1} = v_n + h \left[ \sum_{i=1}^{n} w_i \left( \frac{m_i + m_{i+1}}{2} \right) \right] \]

8.5.3 Numerical Example

We solve the second-order equation

\[ y'' + 3y' + 2y = 2 \exp(-3x) \]
\[ y(0) = 1, \quad y'(0) = -2 \]

(8.5.19)

If we denote \( \dot{y} = v \), the differential equation can be written as

\[ \dot{v} + 3v + 2y = 2 \exp(-3x). \]

We can write equation (8.5.19) as the system of equations

\[ \dot{y} = v \]
\[ \dot{v} = -2y - 3v + 2 \exp(-3x) \]

(8.5.20)

with exact solution \( y(x) = \exp(-x) - \exp(-2x) + \exp(-3x) \) and initial values \( y(0) = 1, \quad v(0) = -2 \). By using \( h = 0.01 \), we obtain a solution in the region \( 0 \leq x \leq 1 \) with this new equation in (8.5.17) and (8.5.18) as shown in Table 8.7.

Table 8.7

<table>
<thead>
<tr>
<th>( x )</th>
<th>Numerical Solution</th>
<th>Exact Solution</th>
<th>Absolute Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.8269249454E+00</td>
<td>0.8269248856E+00</td>
<td>0.5972778871E-07</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6972224306E+00</td>
<td>0.6972223431E+00</td>
<td>0.8745288960E-07</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5985763938E+00</td>
<td>0.5985762443E+00</td>
<td>0.9500676212E-07</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5221853843E+00</td>
<td>0.5221852938E+00</td>
<td>0.9046160949E-07</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4617814579E+00</td>
<td>0.4617813787E+00</td>
<td>0.7923542494E-07</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4129163773E+00</td>
<td>0.4129163124E+00</td>
<td>0.6488580956E-07</td>
</tr>
<tr>
<td>0.7</td>
<td>0.3724448178E+00</td>
<td>0.3724447681E+00</td>
<td>0.4967923989E-07</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3381504344E+00</td>
<td>0.3381503994E+00</td>
<td>0.3499606382E-07</td>
</tr>
<tr>
<td>0.9</td>
<td>0.3084763059E+00</td>
<td>0.3084762843E+00</td>
<td>0.2161763496E-07</td>
</tr>
<tr>
<td>1.0</td>
<td>0.2823312362E+00</td>
<td>0.2823312263E+00</td>
<td>0.9928264599E-08</td>
</tr>
</tbody>
</table>

312
8.6 NEW RUNGE KUTTA STARTERS FOR MULTISTEP METHODS

In many publications e.g. Gear [1980], Hindmarsh [1974], Krog [1969] and Shampine and Gordon [1975], there has been much interest in the discussion of starting a multistep method by using \((k-1)\) values needed to start a \(k\)-step method when just the initial value is given. For example, a four step explicit Adam's formula can be started with three fourth order Runge-Kutta steps and a final function evaluation to obtain \(f_i, i=0, 1, 2, 3\).

In this section, our discussion presents fourth order linear and non linear methods and fifth order method (Evans and Yaakub [1995]) as starters for solving the explicit fourth and fifth order multistep methods or the s-step Adam formula (Lambert [1991], pp 96).

8.6.1 Starting By Linear and Nonlinear Methods

In the numerical solution of initial value problems, single step methods use information from only the last computed point \((x_i, y_i)\) to evaluate \((x_{i+1}, y_{i+1})\). In contrast, multistep methods use information from several previous points \((x_i, y_i), (x_{i-1}, y_{i-1}), (x_{i-2}, y_{i-2}), ....\). It is obvious that the multistep methods cannot be used until sufficient few points in the numerical solution are obtained. To obtain these several previous values, we can use a linear or non-linear method as a starter for solving a multistep method. Therefore, for the four-step Adam's formula, we can apply fourth order linear or non-linear methods using a variety of means which can be written as in equations (8.0.1)-(8.0.2).

To apply the five-step Adam's formula, we use a new fifth order arithmetic mean (AM) method (see Evans and Yaakub [1995]) which can be written as

\[
y_{m1} = y_n + h \left[ \sum_{i=1}^{4} w_i \left( \frac{k_i + k_{m1}}{2} \right) \right]
\]  

(8.6.1)
where \[ \sum_{i=1}^{4} w_i = 1 \]

\[
\begin{align*}
    k_1 &= f(y_i) \\
    k_2 &= f(y_i + \alpha_i h_k) \\
    k_3 &= f(y_i + \alpha_i h_k + \alpha_i h_k) \\
    k_4 &= f(y_i + \alpha_i h_k + \alpha_i h_k + (\frac{1}{2} - \alpha_i - \alpha_i)h_k) \\
    k_5 &= f(y_i + \alpha_i h_k + \alpha_i h_k + \alpha_i h_k + (1 - \alpha_i - \alpha_i - \alpha_i)h_k)
\end{align*}
\]

(8.6.2)

where fifth order accuracy is obtained by choosing, e.g.,

\[
\begin{align*}
    w_1 &= 0.2615038147, \quad w_2 = -0.2765809214, \quad w_3 = 0.5947141647 \\
    w_4 &= 0.4203629420, \quad \alpha_i = 1.5471214403, \quad \alpha_2 = 0.1756458393 \\
    \alpha_3 &= 0.1243059001, \quad \alpha_4 = 0.1009316694, \quad \alpha_5 = 0.1100539630 \\
    \alpha_6 &= 0.9997431862, \quad \alpha_7 = -0.0928890403, \quad \alpha_8 = -0.6201812828
\end{align*}
\]

8.6.2 Explicit Fourth and Fifth Order Multistep Adam Method

A well known four-step explicit Adam-Bashforth method (see Lambert [1991], pp. 96) is of the form

\[
y_{n+1} = y_n + \frac{h}{24} \left[ 55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3} \right]
\]

(8.6.3)

For solving equation (8.6.3), we need \( f_i, i = 0,1,2,3 \) by using the fourth order method in equation (8.6.1).

The five-step fifth order multistep Adam's method is of the form

\[
y_{n+1} = y_n + \frac{h}{720} \left[ 1901f_n - 2774f_{n-1} + 2616f_{n-2} - 2174f_{n-3} + 251f_{n-4} \right]
\]

(8.6.4)

and we need \( f_i, i = 0,1,2,3,4 \) by using the fifth order method in equations (8.6.1) and (8.6.2). Algorithms for solving the multistep Adam's formula in equations (8.6.3) and (8.6.4) are shown in Figure 8.5 and 8.6.
Given \( f(x, y), (x_i, y_i) \) \( \{i = 0, 1, 2, 3\}, h, n \)

For \( i = 0, 1, 2, 3 \)

Set \( f_i = f(x_i, y_i) \) \( \Rightarrow \) By using a fourth order method as a starter

For \( i = 3, 4, 5, \ldots, n - 1 \)

Set \( x_{i+1} = x_i + h \)

\[
\begin{align*}
y_{i+1} &= y_i + \frac{h}{24} [55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}] \\
f_{i-3} &= f_{i-2} \\
f_{i-2} &= f_{i-1} \\
f_{i-1} &= f_i \\
f_i &= f(x_{i+1}, y_{i+1})
\end{align*}
\]

Figure 8.5: Algorithm for the fourth order Adam's multistep method

Given \( f(x, y), (x_i, y_i) \) \( \{i = 0, 1, 2, 3, 4\}, h, n \)

For \( i = 0, 1, 2, 3, 4 \)

Set \( f_i = f(x_i, y_i) \) \( \Rightarrow \) By using a fifth order method as a starter

For \( i = 4, 5, 6, \ldots, n - 1 \)

Set \( x_{i+1} = x_i + h \)

\[
\begin{align*}
y_{i+1} &= y_i + \frac{h}{720} [1901f_i - 2774f_{i-1} + 2616f_{i-2} - 1274f_{i-3} + 251f_{i-4}] \\
f_{i-4} &= f_{i-3} \\
f_{i-3} &= f_{i-2} \\
f_{i-2} &= f_{i-1} \\
f_{i-1} &= f_i \\
f_i &= f(x_{i+1}, y_{i+1})
\end{align*}
\]

Figure 8.6: Algorithm for the fifth order Adam's multistep method
8.6.3 Numerical Example

We consider the IVPs

\[ y' = 1 + x - y, \quad y(0) = 1, \quad 0 \leq x \leq 1 \]  \hspace{1cm} (8.6.5)

where the exact solution is \( y(x) = x + \exp(-x) \). The absolute error in the numerical solution obtained by using the fourth order Adam's formula in equation (8.6.3) with the fourth order AM formula (adam44), Contraharmonic mean (adcom44), Centroidal mean (adcem44), Geometric mean (adgm44), Root mean square (adrms44) and a new fifth order method in equation (8.6.1) or (adam54) as starting formulas are shown in Table 8.8.

Table 8.8: The absolute errors for solving equation (8.6.5) by the fourth order Adam's formula (8.6.3) using various fourth order methods as starters

<table>
<thead>
<tr>
<th>X</th>
<th>Adcom44</th>
<th>Adcem44</th>
<th>Adrms44</th>
<th>Adgm44</th>
<th>Adam44</th>
<th>Adam54</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.144E-04</td>
<td>0.478E-05</td>
<td>0.648E-05</td>
<td>0.110E-04</td>
<td>0.354E-09</td>
<td>0.228E-09</td>
</tr>
<tr>
<td>20</td>
<td>0.130E-04</td>
<td>0.433E-05</td>
<td>0.586E-05</td>
<td>0.996E-05</td>
<td>0.101E-08</td>
<td>0.496E-09</td>
</tr>
<tr>
<td>30</td>
<td>0.118E-04</td>
<td>0.392E-05</td>
<td>0.531E-05</td>
<td>0.901E-05</td>
<td>0.180E-08</td>
<td>0.711E-09</td>
</tr>
<tr>
<td>40</td>
<td>0.106E-04</td>
<td>0.355E-05</td>
<td>0.480E-05</td>
<td>0.816E-05</td>
<td>0.271E-08</td>
<td>0.881E-09</td>
</tr>
<tr>
<td>50</td>
<td>0.963E-05</td>
<td>0.321E-05</td>
<td>0.435E-05</td>
<td>0.738E-05</td>
<td>0.369E-08</td>
<td>0.101E-08</td>
</tr>
<tr>
<td>60</td>
<td>0.871E-05</td>
<td>0.291E-05</td>
<td>0.393E-05</td>
<td>0.667E-05</td>
<td>0.473E-08</td>
<td>0.111E-08</td>
</tr>
<tr>
<td>70</td>
<td>0.788E-05</td>
<td>0.263E-05</td>
<td>0.356E-05</td>
<td>0.604E-05</td>
<td>0.582E-08</td>
<td>0.118E-08</td>
</tr>
<tr>
<td>80</td>
<td>0.714E-05</td>
<td>0.238E-05</td>
<td>0.322E-05</td>
<td>0.546E-05</td>
<td>0.692E-08</td>
<td>0.123E-08</td>
</tr>
<tr>
<td>90</td>
<td>0.646E-05</td>
<td>0.216E-05</td>
<td>0.292E-05</td>
<td>0.494E-05</td>
<td>0.803E-08</td>
<td>0.125E-08</td>
</tr>
<tr>
<td>100</td>
<td>0.585E-05</td>
<td>0.195E-05</td>
<td>0.264E-05</td>
<td>0.447E-05</td>
<td>0.913E-08</td>
<td>0.126E-08</td>
</tr>
</tbody>
</table>

While the absolute errors in the numerical solution by using the fifth order multistep Adam's method in equation (8.6.4) with fourth order AM (adam45) and the new fifth order AM (adam55) as starters are shown in Table 8.9.
Table 8.9: The absolute errors for solving equation (8.6.5) by the fifth order Adam's formula (8.6.4) using fourth and fifth order AM method as starters

<table>
<thead>
<tr>
<th>X</th>
<th>Adam45</th>
<th>Adam55</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>.1212808E-11</td>
<td>.2942535E-11</td>
</tr>
<tr>
<td>20</td>
<td>.1659117E-11</td>
<td>.9436896E-13</td>
</tr>
<tr>
<td>30</td>
<td>.3995471E-11</td>
<td>.2579714E-11</td>
</tr>
<tr>
<td>40</td>
<td>.5871748E-11</td>
<td>.4590328E-11</td>
</tr>
<tr>
<td>50</td>
<td>.7355228E-11</td>
<td>.6195489E-11</td>
</tr>
<tr>
<td>60</td>
<td>.8502976E-11</td>
<td>.7453371E-11</td>
</tr>
<tr>
<td>70</td>
<td>.9365841E-11</td>
<td>.8415935E-11</td>
</tr>
<tr>
<td>80</td>
<td>.9987344E-11</td>
<td>.9127810E-11</td>
</tr>
<tr>
<td>90</td>
<td>.1040568E-10</td>
<td>.9628298E-11</td>
</tr>
<tr>
<td>100</td>
<td>.1065370E-10</td>
<td>.9950707E-11</td>
</tr>
</tbody>
</table>

From Tables 8.8 and 8.9, we can see that the new fifth order method in equation (8.6.1) used as a starting single step method gives greater accuracy compared with using the standard fourth order Runge Kutta method.
CHAPTER 9

HIGH ORDER INTEGRATION FORMULA
USING TRIGONOMETRIC POLYNOMIALS
FOR PERIODIC IVPs
An interesting and important class of initial value problems which can arise in practice consists of problems whose solutions are known to be periodic, or to oscillate with a known frequency (Lambert [1973]). If this frequency, or a reasonable estimate for it, is known in advance, then a class of methods based on trigonometrical polynomials, developed by Gautschi [1961] is particularly appropriate. The linear multistep method of order p may be defined by the requirement

\[ L[x';h] = 0, \quad r = 0,1,\ldots,p; \quad L[x^{r+1};h] \neq 0, \]  

(9.0.1)

where \( L \), the associated linear difference operator, is defined by the equation

\[ L[y(x);h] = \sum_{j=0}^{s} [\alpha_j y(x + jh) - h\beta_j y'(x + jh)]. \]  

(9.0.2)

The linear difference operator \( L \), of order \( p \) annihilates all algebraic polynomials of order \( \leq p \). The methods developed by Gautschi similarly annihilate trigonometric polynomials up to a certain degree where the estimated period of the solution is \( T \), and the frequency is defined as \( \omega = \frac{2\pi}{T} \). The method proposed by Gautschi

\[ \sum_{j=0}^{s} \alpha_j y_{n+j} = h \sum_{j=0}^{s} \beta_j (v) f_{n+j}, \quad v = \omega h, \quad \alpha_s = +1 \]  

(9.0.3)

is said to be of trigonometric order \( q \) relative to the frequency \( \omega \) if the associated linear difference operator

\[ L_\omega[y(x);h] = \sum_{j=0}^{s} [\alpha_j y(x + jh) - h\beta_j y'(x + jh)] \]  

(9.0.4)

satisfies

i) \( L_\omega[1;h] = 0 \).

ii) \( L_\omega[\cos(r\omega x);h] = L_\omega[\sin(r\omega x);h] = 0, \quad r = 1,2,\ldots,q \), \hspace{1cm} (9.0.5)

iii) \( L_\omega[\cos((q + 1)\omega x);h] \) and \( L_\omega[\sin((q + 1)\omega x);h] \)

are not both identically zero.
For the case $k=1$ or if we require it to be of trigonometric order 1, then for the three unspecified coefficients $\alpha_0, \beta_1(v), \text{and} \beta_0(v)$ we obtain a base formula in the form

$$y_{n+1} = y_n + h[\beta_0 f_n + \beta_1 f_{n+1}]$$  \hspace{1cm} (9.0.6)

which must satisfy the conditions

$$L_0[1; h] = 1 + \alpha_0 = 0, \quad (9.0.6-i)$$
$$L_0[\cos(\omega x); h] = \cos[\omega(x + h)] - \alpha_0 \cos(\omega x) + h\omega [\beta_1 \sin[\omega(x + h)] + \beta_0 \sin(\omega x)] = 0, \quad (9.0.6-ii)$$
$$L_0[\sin(\omega x); h] = \sin[\omega(x + h)] - \alpha_0 \sin(\omega x) - h\omega [\beta_1 \cos[\omega(x + h)] + \beta_0 \cos(\omega x)] = 0, \quad (9.0.6-iii)$$

The equations (9.0.6-ii)-(9.0.6-iii) can also be written in matrix form as

$$\begin{bmatrix} v \sin(\omega x) & v \sin(\omega x + v) \\ -v \cos(\omega x) & -v \cos(\omega x + v) \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \alpha_0 \cos(\omega x) - \cos(\omega x + v) \\ \alpha_0 \sin(\omega x) - \sin(\omega x + v) \end{bmatrix}$$

and by taking $\omega x = 0$, the solution of this set of equations is found to be

$$\alpha_0 = -1, \quad \beta_0 = \beta_1 = \frac{[\tan(\frac{x}{2})]}{v} \quad \text{where} \quad v = \omega h$$

which is

$$y_{n+1} = y_n + \frac{h \tan(\frac{x}{2})}{v} (f_n + f_{n+1}) \quad (9.0.7)$$

and is equivalent to the well known Trapezoidal rule. We now, expand $\tan(\frac{x}{2})$ in a power series where

$$\tan(v) = v + \frac{1}{3} v^3 + \frac{2}{15} v^5 + \frac{17}{315} v^7 + ...$$

and by replacing $v$ with $(\frac{x}{2})$ we have
Substituting (9.0.8) into (9.0.7) we obtain

\[
y_{n+1} = y_n + h \left[ \frac{1}{2} \frac{v^2}{24} + \frac{v^4}{240} + \frac{17v^6}{40320} + \ldots \right] (f_n + f_{n+1}).
\]  

(9.0.9)

and expanding the right hand-side (9.0.9) at \( x_n \) we have

\[
y_{n+1} = y_n + h \left[ \frac{1}{2} \frac{v^2}{24} + \frac{v^4}{240} + \frac{17v^6}{40320} + \ldots \right] \left( 2y_n' + h y_n'' + \frac{h^2}{2} y_n''' + \frac{h^3}{6} y_n^{(4)} + \ldots \right)
\]

\[
= y_n + h y_n' + h^2 y_n'' + h^3 \left( \frac{y_n'''}{4} + \frac{y_n^{(4)}}{12} \right) + \ldots
\]

(9.0.10)

where the local truncation error (LTE) is given by

\[
LTE = h^3 \left[ \frac{y_n'''}{6} - \frac{y_n^{(4)}}{4} - \omega^3 y_n' \right] + o(h^4)
\]

\[
= -\frac{h^3}{12} (y_n''' + \omega^3 y_n') + o(h^4).
\]

(9.0.11)

On the trigonometric order attainable by a method of equation (9.0.3) of given stepnumber, Gautschi [1961] proves the following result (see for example, Lambert [1973], p.207). Let \( v = \omega h \) be given and let \( k \) be a given even (or odd) integer. Then, for any given set of coefficients \( \alpha_j, j = 0,1,\ldots,k \), subject to \( \sum_{j=0}^{k} \alpha_j = 0 \), there exists a unique explicit (or implicit) method of equation
The general explicit one step method as a particular case for a classical Runge-Kutta method can be written as

\[ y_{n+1} = y_n + h\phi(x_n, y_n, h) \]  

(9.1.1)

For example, two classical second order methods are the modified Euler method and the improved Euler method given as

\[ y_{n+1} = y_n + hk_1 \]

where \( k_1 = f(x_n, y_n) \)

(9.1.2)

and

\[ y_{n+1} = y_n + h\left(\frac{k_1 + k_2}{2}\right) \]

where \( k_1 = f(x_n, y_n) \)

\[ k_2 = f(x_n + h, y_n + hk_1) \]

(9.1.3)

For the improved Euler method in (9.1.3) we may consider the trapezoidal rule applied in a predictor-corrector mode with the Euler method as a predictor. By using formula (9.0.7) another predictor-corrector method involving a 2-stage method for solving the oscillatory problem can be written as

\[ y_{n+1} = y_n + h\tan(\xi)\left(\frac{k_1 + k_2}{2}\right) \]

where \( k_1 = f(x_n, y_n) \)

\[ k_2 = f(x_n + h, y_n + hk_1) \]

(9.1.4)
We now establish the formula in (9.1.4) for solving the oscillatory problem by writing

\[ k_1 = f \]
\[ k_2 = f + hff + \frac{1}{2} h^2 f'' + \ldots \]

and substituted in equation (9.1.4) to give,

\[ y_{n+1} = y_n + h \left[ \frac{1}{2} + \frac{\omega^2 h^2}{24} + \frac{\omega h^4}{240} + \ldots \right] \left[ 2f + hff + \frac{1}{2} h^2 f^2 f'' + \ldots \right] \]
\[ = y_n + h \left\{ f + \frac{1}{2} hff + h^2 \left( \frac{\omega^2}{12} + \frac{1}{4} f^2 f'' \right) + \ldots \right\} \]
\[ = y_n + hf + \frac{h^2}{2} ff + h^3 \left( \frac{\omega^2}{12} + \frac{1}{4} f^2 f'' \right) + \ldots \quad (9.1.5) \]

By comparing (9.1.5) with the Taylor series expansion of \( y(x_{n+1}) \) in terms of \( f \) and its partial derivatives we obtain the local truncation error (LTE) as

\[ LTE = h^3 \left( \frac{1}{6} ff'^2 + \frac{1}{6} f^2 f'' - \frac{1}{12} \omega^2 - \frac{1}{4} f^2 f'' \right) \]
\[ = h^3 \left( \frac{1}{6} ff'^2 - \frac{1}{12} ff'' - \frac{1}{12} \omega^2 \right). \]

This error expression confirms that equation (9.1.5) is of second order accuracy.

9.2 THE NUMERICAL SOLUTION OF OSCILLATORY PROBLEM

In Sanugi [1986] the trigonometric order of a multistep integration formula is shown to be maximal, where the number of parameters is sufficient to solve all the equations of conditions, if the stepnumber is odd and the formula is implicit. Therefore, we seek an implicit \( k \)-step method for oscillatory problems with \( k \) an odd number.
9.2.1 A Fourth Order Method For Oscillatory Problems

By applying \( k = 3 \) in equation (9.1.2) and \( \alpha_0 = -1, \alpha_1 = \alpha_2 = 0 \) and \( \alpha_3 = 1 \) we obtain a base formula in the form

\[
y_{n+3} = y_n + h[\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2} + \beta_3 f_{n+3}]
\]

(9.2.1)

and by using the difference operator in (9.1.3) and (9.1.4) we obtain

a) \( L_n[1: h] = 0 \) (9.2.2-i)

b) For \( r = 1 \), we have
   i) \( L_n[\cos(\alpha x): h] = 0 \)
   \[-\cos(\alpha x) + \cos(\alpha x + 3v) = -\nu \beta_0 \sin(\alpha x) - \nu \beta_1 \sin(\alpha x + v) - \nu \beta_2 \sin(\alpha x + 2v) - \nu \beta_3 \sin(\alpha x + 3v)\]
   and when \( \alpha x = 0 \), we have
   \[-\nu \beta_0 \sin(v) - \nu \beta_1 \sin(2v) - \nu \beta_2 \sin(3v) = \cos(3v) - 1 \] (9.2.2-ii)

   ii) \( L_n[\sin(\alpha x): h] = 0 \)
   \[-\sin(\alpha x) + \sin(\alpha x + 3v) = \nu \beta_0 \cos(\alpha x) + \nu \beta_1 \cos(\alpha x + v) + \nu \beta_2 \cos(\alpha x + 2v) + \nu \beta_3 \cos(\alpha x + 3v)\]
   and for \( \alpha x = 0 \), we obtain
   \[\nu \beta_0 + \nu \beta_1 \cos(v) + \nu \beta_2 \cos(2v) + \nu \beta_3 \cos(3v) = \sin(3v) \] (9.2.2-iii)

c) For \( r = 2 \);
   i) \( L_n[\cos(2\alpha x): h] = 0 \)
   \[-\cos(2\alpha x) + \cos(2\alpha x + 6v) = -2\nu \beta_0 \sin(2\alpha x) - 2\nu \beta_1 \sin(2\alpha x + 2v) - 2\nu \beta_2 \sin(2\alpha x + 4v) - 2\nu \beta_3 \sin(2\alpha x + 6v)\]
   and for \( \alpha x = 0 \), we have
   \[-2\nu \beta_1 \sin(2v) - 2\nu \beta_2 \sin(4v) - 2\nu \beta_3 \sin(6v) = \cos(6v) - 1 \] (9.2.2-iv)

   ii) \( L_n[\sin(2\alpha x): h] = 0 \)
   \[-\sin(2\alpha x) + \sin(2\alpha x + 6v) = 2\nu \beta_0 \cos(2\alpha x) + 2\nu \beta_1 \cos(2\alpha x + 2v) + 2\nu \beta_2 \cos(2\alpha x + 4v) + 2\nu \beta_3 \cos(2\alpha x + 6v)\]
and for $\omega x = 0$, we obtain

$$2\nu \beta_0 + 2\nu \beta_1 \cos(2\nu) + 2\nu \beta_2 \cos(4\nu) + 2\nu \beta_3 \cos(6\nu) = \sin(6\nu) \quad (9.2.2-v)$$

Consequently equations (9.2.2-ii)-(9.2.2-v) are then solved simultaneously by Mathematica, we obtain the values of the parameters, i.e.,

$$\beta_0 = \left(3(2940000 - 4165000\nu^2 + 29473500\nu^4 - 270016250\nu^6 + 451324475\nu^8 \right.$$

$$\left. -12319825\nu^{10} - 781610\nu^{12} - 4280400\nu^{14} + 412992\nu^{16}\right) / (140(168000 - 280000\nu^2 + 207900\nu^4 - 1341000\nu^6 + 946545\nu^8 - 104600\nu^{10} + 2016\nu^{12}))$$

\[\beta_1 = \left(27(140000 - 24500\nu^2 + 830425\nu^6 - 1780875\nu^8 + 750600\nu^{10} - 62208\nu^{12})\right) / (2(168000 - 280000\nu^2 + 207900\nu^4 - 1341000\nu^6 + 946545\nu^8 - 104600\nu^{10} + 2016\nu^{12}))$$

\[\beta_2 = \left(15(28000 - 49000\nu^2 + 294700\nu^4 - 1570450\nu^6 + 2144295\nu^8 - 287685\nu^{10} + 8262\nu^{12})\right) / (4(168000 - 280000\nu^2 + 207900\nu^4 - 1341000\nu^6 + 946545\nu^8 - 104600\nu^{10} + 2016\nu^{12}))$$

\[\beta_3 = \left(15(8400 - 11900\nu^2 - 68880\nu^4 + 149315\nu^6 - 118043\nu^8 + 14536\nu^{10} - 384\nu^{12})\right) / (2(168000 - 280000\nu^2 + 207900\nu^4 - 1341000\nu^6 + 946545\nu^8 - 104600\nu^{10} + 2016\nu^{12}))$$

Equations (9.2.3-i)-(9.2.3-iv) can also be simplified by Mathematica, i.e.,

\[\text{In}[1]:= \text{Series}[\beta_n,\{\nu,0,12\}]\]

and we obtain $\beta_0, \beta_1, \beta_2$, and $\beta_3$ as

\[
\beta_0 = \frac{3}{8} \left(1 - \frac{\nu^4}{4} + \frac{2209\nu^4}{240} - \frac{1387597\nu^6}{20160} + \frac{28749227\nu^6}{1209600} + \frac{2805517639\nu^8}{14515200} - \frac{1881719979239\nu^{10}}{6096384000} + O(\nu^{12}) \right)
\]

\[
\beta_1 = \frac{9}{8} \left(1 - \frac{\nu^4}{12} - \frac{1307\nu^4}{144} + \frac{3161533\nu^6}{60480} - \frac{18221411\nu^8}{518400} - \frac{6140536111\nu^{10}}{43545600} + \frac{4981300261121\nu^{12}}{18289152000} + O(\nu^{14}) \right)
\]
\[ \beta_2 = \frac{9}{8} \left( 1 - \frac{\nu^2}{12} + \frac{6587\nu^4}{720} - \frac{396199\nu^6}{12096} + \frac{15858841\nu^8}{3628800} + \frac{972466693\nu^{10}}{8709120} - \frac{2415491349181\nu^{12}}{18289152000} + O(\nu^{14}) \right) \]

\[ \beta_3 = \frac{3}{8} \left( 1 + \frac{\nu^2}{4} - \frac{433\nu^4}{48} + \frac{209939\nu^6}{20160} + \frac{13097489\nu^8}{1209600} - \frac{136652999\nu^{10}}{2073600} + \frac{65908719619\nu^{12}}{6096384000} + O(\nu^{14}) \right) \]

Thus the proposed method is given by,

\[ y_{n+3} = y_n + 3h(\beta_{n} f_n + \beta_{n+1} f_{n+1} + \beta_{n+2} f_{n+2} + \beta_{n+3} f_{n+3}) \quad (9.2.4) \]

where

\[ \beta_n = \frac{1}{8} \left( 1 + \frac{\nu^2}{4} + \frac{2209\nu^4}{240} - \frac{1387597\nu^6}{20160} + \frac{28749227\nu^8}{1209600} + \frac{2805517639\nu^{10}}{14515200} - \frac{1881719979239\nu^{12}}{6096384000} + O(\nu^{14}) \right) \]

\[ \beta_n = \frac{3}{8} \left( 1 + \frac{\nu^2}{12} + \frac{1307\nu^4}{144} - \frac{3161533\nu^6}{60480} + \frac{18221411\nu^8}{518400} + \frac{6140536111\nu^{10}}{43545600} - \frac{498130026111\nu^{12}}{18289152000} + O(\nu^{14}) \right) \]

\[ \beta_n = \frac{3}{8} \left( 1 - \frac{\nu^2}{12} + \frac{6587\nu^4}{720} - \frac{396199\nu^6}{12096} + \frac{15858841\nu^8}{3628800} + \frac{972466693\nu^{10}}{8709120} - \frac{2415491349181\nu^{12}}{18289152000} + O(\nu^{14}) \right) \]

\[ \beta_n = \frac{1}{8} \left( 1 + \frac{\nu^2}{4} + \frac{433\nu^4}{48} + \frac{209939\nu^6}{20160} + \frac{13097489\nu^8}{1209600} - \frac{136652999\nu^{10}}{2073600} + \frac{65908719619\nu^{12}}{6096384000} + O(\nu^{14}) \right) \]

Now, the formula in equation (9.2.4) developed for solving oscillatory problems is transformed into

\[ y_{n+1} = y_n + h \left[ w_1 f_n + w_2 f_{n+1} + w_3 f_{n+2} + w_4 f_{n+3} \right] \quad (9.2.5-i) \]

where \( w_1 = \frac{1}{8}, w_2 = \frac{3}{8}, w_3 = \frac{3}{8}, w_4 = \frac{1}{8} \) and we replace

\[ k_1 = f_n, k_2 = f_{n+1}, k_3 = f_{n+2}, k_4 = f_{n+3} \]

and we relate \( a_i \) with \( v \) (see Sanugi [1986], p 311).

\[ y_{n+1} = y_n + h \left[ w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 \right] \quad (9.2.5-ii) \]
Through the standard procedure of adjustment of the parameters, fourth order accuracy is obtained for formula (9.2.5) from the solution of six equations of conditions, i.e.,

\[ h^2 f_{ff}: \quad 1 - 3a_i = 0 \quad (9.2.6-i) \]

\[ h^2 f_{ff}^2: \quad 2 - 6a_i + 9a_1a_2 + 2a_i^2 + 2a_i - 3a_i^2 = 0 \quad (9.2.6-ii) \]

\[ h^2 f f_{f}: \quad (1 - 3a_i)(1 + 3a_i) = 0 \quad (9.2.6-iii) \]

\[ h^2 f_{ff}^3: \quad 1 - 2a_i + 3a_1a_2 + 2a_i^2 - 3a_i^3a_2 = 0 \quad (9.2.6-iv) \]

\[ h^2 f f_{f}: \quad 8 - 24a_i - 18a_i^2 + 36a_1a_2 + 27a_1^2a_2 + 16a_i + 16a_i^2 - 18a_i^2 - 9a_i^2a_2 = 0 \quad (9.2.6-v) \]

\[ h^2 f f_{f}: \quad (1 - 3a_i)(1 + 3a_i + 9a_i^i) = 0 \quad (9.2.6-vi) \]

Equations (9.2.6-i)-(9.2.6-vi) are then solved simultaneously by Mathematica to give two sets of the required parameters, i.e.,

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Set 1</th>
<th>Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_i)</td>
<td>(\frac{1}{3})</td>
<td>(\frac{1}{3})</td>
</tr>
<tr>
<td>(a_2)</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>(a_1 = \frac{1}{3} - a_2)</td>
<td>1</td>
<td>-(\frac{1}{3})</td>
</tr>
<tr>
<td>(a_3)</td>
<td>1</td>
<td>-7</td>
</tr>
<tr>
<td>(a_4)</td>
<td>-1</td>
<td>11</td>
</tr>
<tr>
<td>(a_5 = 1 - a_3 - a_4)</td>
<td>1</td>
<td>-3</td>
</tr>
</tbody>
</table>

Thus, two sets of fourth order Runge-Kutta formula for solving oscillatory problem can be written as

a) \[ k_1 = f(x_i, y_i) \]

\[ k_2 = f\left(x_i + \frac{h}{3}, y_i + \frac{1}{3}hk_i\right) \]

\[ k_3 = f\left(x_i + \frac{2}{3}h, y_i - \frac{1}{3}hk_i + hk_2\right) \]

\[ k_4 = f(x_i + h, y_i + hk_i - hk_2 + hk_3) \]

and \[ y_{i+1} = y_i + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4] \] (9.2.7)
b) \[ k_1 = f(x_n, y_n) \]
\[ k_2 = f\left(x_n + \frac{h}{3}, y_n + \frac{1}{3}hk_1\right) \]
\[ k_3 = f\left(x_n + \frac{2}{3}h, y_n + hk_1 - \frac{1}{3}hk_2\right) \]
\[ k_4 = f(x_n + h, y_n - 7hk_1 + 11hk_2 - 3hk_3) \]
and \[ y_{n+1} = y_n + h\left[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4\right] \quad (9.2.8) \]

where in equations (9.2.7) and (9.2.8) we have the values \( w_i, i = 1, 2, 3, 4 \) given by

\[ w_1 = \frac{1}{8}\left(1 + \frac{v^4}{4} + \frac{2209v^6}{240} - \frac{1387597v^8}{20160} + \frac{28749227v^{10}}{1209600} + \frac{2805517639v^{12}}{14515200} - \frac{188171997239v^{14}}{6096384000} + O(v^{16})\right) \]
\[ w_2 = \frac{3}{8}\left(1 - \frac{v^4}{12} - \frac{1307v^6}{144} + \frac{3161533v^8}{60480} - \frac{18221411v^{10}}{518400} + \frac{6140536111v^{12}}{43545600} + \frac{4981300261121v^{14}}{18289152000} + O(v^{16})\right) \]
\[ w_3 = \frac{3}{8}\left(1 - \frac{v^4}{12} + \frac{6587v^6}{720} - \frac{396199v^8}{12096} - \frac{15858841v^{10}}{3628800} + \frac{972466693v^{12}}{8709120} + \frac{2415491349181v^{14}}{18289152000} + O(v^{16})\right) \]
\[ w_4 = \frac{1}{8}\left(1 + \frac{v^4}{4} - \frac{433v^6}{48} + \frac{209939v^8}{20160} + \frac{13097489v^{10}}{1209600} - \frac{1366529999v^{12}}{2073600} + \frac{65908719619v^{14}}{6096384000} + O(v^{16})\right) \]

9.2.2 Error Analysis

By substituting the values \( a_i, 1 \leq i \leq 4 \) in set one into equation (9.2.7) using Mathematica and evaluating all the terms up to \( (h^4) \) to represent the local truncation error for this method, we have

\[ LTE = \frac{h^4}{6480}\left[54ff'_r - 81ff'_m - 9f^2f'_r + 13f^2f'_m - f^3f''_m\right] + O(h^6) \quad (9.2.9) \]

9.2.3 Stability Analysis

We examine the stability region of the fourth order method for solving oscillatory problem with the test equation \( y' = \lambda y \) and we obtain
By substituting \( k_i, 1 \leq i \leq 4 \) in equation (9.2.10) and \( w_i, 1 \leq i \leq 4 \) into the new fourth order formula, i.e.,

\[
y_{n+1} = y_n + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4]
\]

we obtain

\[
y_{n+1} = y_n + (h\lambda)y_n + \frac{1}{2}(h\lambda)^2 y_n + \frac{1}{6}(h\lambda)^3 y_n + \frac{1}{24}(h\lambda)^4 y_n + \ldots
\]

(9.2.11)

By substituting \( h\lambda = z \) in (9.2.12), we can show that

\[
y_{n+1} = y_n + y_n \left[ z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \right] + O(z^5).
\]

(9.2.12)

Following equation (3.3.30), we write \( \frac{y_{n+1}}{y_n} = Q \) in the equation (9.2.13), to obtain

\[
Q = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 + 0(z^5).
\]

(9.2.13)

We now determine the stability region of this fourth order formula in the complex plane that satisfy the condition as in equation (3.3.31), i.e.,

\[
\left| 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \right| < 1
\]

(9.2.14)

By the use of Mathematica, we can plot the graphic surface defined by equation (9.2.15), i.e.,
By the use of Mathematica, we can plot the graphic surface defined by equation (9.2.15), i.e.,

\[
\text{Figure 9.1: Graphic surface defined by fourth order method}
\]

and the stability region defined by the formula in equation (9.2.15) as shown in Figure 9.2.

\[
\text{Figure 9.2: Stability region for fourth order method.}
\]

9.3 A NEW FIFTH ORDER METHOD FOR OSCILLATORY PROBLEMS

We now extend the same procedure to obtain the fifth order equation by taking \( k = 5 \) in equation (9.0.3) and \( \alpha_0 = -1, \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0 \) and \( \alpha_5 = 1 \) we obtain a base formula in the form
and by using the difference operator in (9.1.3) and (9.1.4) we obtain

a) a) \( L_\omega[1:h] = 0 \)  
\begin{align*}
  & b) \quad \text{For } r = 1 \\
  & \quad \text{i) } \quad L_\omega[\cos(\omega x):h] = 0 \\
  & \quad \quad -\cos(\omega x) + \cos(\omega x + 5v) = -\nu \beta_1 \sin(\omega x) - \nu \beta_2 \sin(\omega x + v) \\
  & \quad \quad -\nu \beta_1 \sin(\omega x + 2v) - \nu \beta_2 \sin(\omega x + 3v) \\
  & \quad \quad -\nu \beta_1 \sin(\omega x + 4v) - \nu \beta_2 \sin(\omega x + 5v) \\
  & \end{align*}

and for \( \omega x = 0 \), we have

\[-\nu \beta_1 \sin(v) - \nu \beta_2 \sin(2v) - \nu \beta_2 \sin(3v) - \nu \beta_2 \sin(4v) - \nu \beta_2 \sin(5v) = \cos(5v) - 1 \]

\[ \cdots \tag{9.3.2-i} \]

\[ \text{ii) } L_\omega[\sin(\omega x):h] = 0 \\
-\sin(\omega x) + \sin(\omega x + 5v) = \nu \beta_1 \cos(\omega x) + \nu \beta_1 \cos(\omega x + v) \\
+ \nu \beta_2 \cos(\omega x + 2v) + \nu \beta_3 \cos(\omega x + 3v) \\
+ \nu \beta_4 \cos(\omega x + 4v) + \nu \beta_5 \cos(\omega x + 5v) \]

and for \( \omega x = 0 \), we obtain

\[ \nu \beta_1 \cos(v) + \nu \beta_2 \cos(2v) + \nu \beta_1 \cos(3v) + \nu \beta_4 \cos(4v) + \nu \beta_5 \cos(5v) = \sin(5v) \]

\[ \cdots \tag{9.3.2-ii} \]

b) For \( r = 1 \)

\[ \text{c) For } r = 2; \]

\[ \text{i) } L_\omega[\cos(2\omega x):h] = 0 \\
-\cos(2\omega x) + \cos(2\omega x + 10v) = -2\nu \beta_1 \sin(2\omega x) - 2\nu \beta_2 \sin(2\omega x + 2v) \\
-2\nu \beta_1 \sin(2\omega x + 4v) - 2\nu \beta_2 \sin(2\omega x + 6v) \\
-2\nu \beta_1 \sin(2\omega x + 8v) - 2\nu \beta_2 \sin(2\omega x + 10v) \]

and for \( \omega x = 0 \), we have

\[-2\nu \beta_1 \sin(2v) - 2\nu \beta_2 \sin(4v) - 2\nu \beta_1 \sin(6v) - 2\nu \beta_2 \sin(8v) - 2\nu \beta_1 \sin(10v) = \cos(10v) - 1 \]

\[ \cdots \tag{9.3.2-iii} \]

\[ \text{ii) } L_\omega[\sin(2\omega x):h] = 0 \\
-\sin(2\omega x) + \sin(2\omega x + 10v) = 2\nu \beta_1 \cos(2\omega x) + 2\nu \beta_1 \cos(2\omega x + 2v) \\
+ 2\nu \beta_2 \cos(2\omega x + 4v) + 2\nu \beta_3 \cos(2\omega x + 6v) \\
+ 2\nu \beta_4 \cos(2\omega x + 8v) + 2\nu \beta_5 \cos(2\omega x + 10v) \]

and for \( \omega x = 0 \), we obtain

\[ 2\nu \beta_1 \cos(2v) + 2\nu \beta_1 \cos(4v) + 2\nu \beta_1 \cos(6v) + 2\nu \beta_4 \cos(8v) + 2\nu \beta_5 \cos(10v) = \sin(10v) \]

\[ \cdots \tag{9.3.2-iv} \]
d) For $r = 3$;

i) $L_\omega[\cos(3\omega x): h] = 0$

$-\cos(3\omega x) + \cos(3\omega x + 15\nu) = -3\beta_3\sin(3\omega x) - 3\beta_3\sin(3\omega x + 3\nu) - 3\beta_3\sin(3\omega x + 6\nu) - 3\beta_3\sin(3\omega x + 9\nu) - 3\beta_3\sin(3\omega x + 12\nu) - 3\beta_3\sin(3\omega x + 15\nu)$

and for $\omega x = 0$, we have

$-3\beta_3\sin(3\nu) - 3\beta_3\sin(6\nu) - 3\beta_3\sin(9\nu) - 3\beta_3\sin(12\nu) - 3\beta_3\sin(15\nu) = \cos(15\nu) - 1$  \hspace{1cm} (9.3.2-vi)

ii) $L_\omega[\sin(3\omega x): h] = 0$

$-\sin(3\omega x) + \sin(3\omega x + 15\nu) = 3\beta_3\cos(3\omega x) + 3\beta_3\cos(3\omega x + 3\nu) + 3\beta_3\cos(3\omega x + 6\nu) + 3\beta_3\cos(3\omega x + 9\nu) + 3\beta_3\cos(3\omega x + 12\nu) + 3\beta_3\cos(3\omega x + 15\nu)$

and for $\omega x = 0$, we obtain

$3\beta_3 + 3\beta_3\cos(3\nu) + 3\beta_3\cos(6\nu) + 3\beta_3\cos(9\nu) + 3\beta_3\cos(12\nu) + 3\beta_3\cos(15\nu) = \sin(15\nu)$  \hspace{1cm} (9.3.2-vii)

Equations (9.3.2-ii)-(9.3.2-vii) when solved simultaneously by Mathematica gives

$\beta_\omega = (5(13621760409600 - 88780420915200 v^3 + 166125005541734400 v^4 - 8038457653655116800 v^6 + 76554475691027725440 v^8 + 27320330170447910400 v^{10} + 83562156096513784536 v^{12} - 856989384854893717000 v^{14} + 1097379704652681551073 v^{16} + 4313833781769613800 v^{18} - 1076049130167396000 v^{20} - 922736375164800000 v^{22} + 264776345856000000 v^{24} - 10752 (192036096000 - 1344252672000 v^2 + 4469640134400 v^4 - 508347691084800 v^6 + 2320138789660440 v^8 - 844650807096600 v^{10} - 69298164524589 v^{12} - 490888716667650 v^{14} + 373375548002198 v^{16} - 38027971022400 v^{18} + 531038592000 v^{20} + (9.3.3-i)) / (1728 (192036096000 - 1344252672000 v^2 + 4469640134400 v^4 - 508347691084800 v^6 + 2320138789660440 v^8 - 844650807096600 v^{10} - 69298164524589 v^{12} - 490888716667650 v^{14} + 373375548002198 v^{16} - 38027971022400 v^{18} + 531038592000 v^{20} + (9.3.3 - ii))}
\[ \beta_2 = (625 \left( 1843546521600 - 1222885893280 \, v^2 + 8439092811233280 \, v^4 - \\
329797803196818432 \, v^6 + 3249145065440486016 \, v^8 - \\
3905230975285925376 \, v^{10} + 2346099604550380728 \, v^{12} - \\
12296680402276993800 \, v^{14} + 12684211671091421125 \, v^{16} - \\
14875863801579000000 \, v^{18} + 29923076376000000000 \, v^{20} \right)) / \\
(6912 \left( 192036096000 - 1344252672000 \, v^2 + 4469640134400 \, v^4 - \\
508347691084800 \, v^6 + 2320138789660440 \, v^8 - \\
844650807096600 \, v^{10} - 69298164524589 \, v^{12} - \\
490888716667650 \, v^{14} + 373375548002198 \, v^{16} - \\
38027971022400 \, v^{18} + 531038592000 \, v^{20} \right)) / \\
(9.3.3 - iii) \]

\[ \beta_3 = (625 \left( 25604812800 - 169845258240 \, v^2 - 116167577621760 \, v^4 + \\
3882134419599744 \, v^6 - 35580793663355472 \, v^8 + \\
40089266837582592 \, v^{10} - 12854717663329101 \, v^{12} + \\
719871212607563500 \, v^{14} - 45284099412627250 \, v^{16} + \\
47253456078000000 \, v^{18} - 820674720000000 \, v^{20} \right)) / \\
(96 \left( 192036096000 - 1344252672000 \, v^2 + 4469640134400 \, v^4 - \\
508347691084800 \, v^6 + 2320138789660440 \, v^8 - \\
844650807096600 \, v^{10} - 69298164524589 \, v^{12} - \\
490888716667650 \, v^{14} + 373375548002198 \, v^{16} - \\
38027971022400 \, v^{18} + 531038592000 \, v^{20} \right)) / \\
(9.3.3 - iv) \]

\[ \beta_4 = (625 \left( 5530639564800 - 40742378127360 \, v^2 + 8545306695690240 \, v^4 - \\
254785366461072384 \, v^6 + 1914797080967799168 \, v^8 - \\
1370934790423091712 \, v^{10} + 426843818430175944 \, v^{12} - \\
1975612470809974000 \, v^{14} + 1146898629298170875 \, v^{16} - \\
115966038868325000 \, v^{18} + 1888565260500000 \, v^{20} \right)) / \\
(13824 \left( 192036096000 - 1344252672000 \, v^2 + 4469640134400 \, v^4 - \\
508347691084800 \, v^6 + 2320138789660440 \, v^8 - \\
844650807096600 \, v^{10} - 69298164524589 \, v^{12} - \\
490888716667650 \, v^{14} + 373375548002198 \, v^{16} - \\
38027971022400 \, v^{18} + 531038592000 \, v^{20} \right)) / \\
(9.3.3 - v) \]

\[ \beta_5 = (5 \left( 21892114944000 - 142682819328000 \, v^2 - 25811208395078400 \, v^4 + \\
542547449348284800 \, v^6 - 3237915694006329840 \, v^8 + \\
1988009710663485600 \, v^{10} - 589709412485954271 \, v^{12} + \\
2608473533984918250 \, v^{14} - 1482044278630418678 \, v^{16} + \\
148151676311323200 \, v^{18} - 2341045701504000 \, v^{20} \right)) / \\
(1728 \left( 192036096000 - 1344252672000 \, v^2 + 4469640134400 \, v^4 - \\
508347691084800 \, v^6 + 2320138789660440 \, v^8 - \\
844650807096600 \, v^{10} - 69298164524589 \, v^{12} - \\
490888716667650 \, v^{14} + 373375548002198 \, v^{16} - \\
38027971022400 \, v^{18} + 531038592000 \, v^{20} \right)) / \\
(9.3.3 - vi) \]
Equations (9.3.3-i)-(9.3.3-vi) can also be simplified by Mathematica to give

\[
\beta_0 = \frac{95}{288} \left(1 + \frac{55v^2}{114} \right. + \frac{3282265v^4}{2736} - \frac{5513289335v^6}{114912} + \frac{258450119345v^8}{1378944} + \\
\frac{68594391998759v^{10}}{11819520} - \frac{727108666308742877v^{12}}{6949877760} + O(\nu^3) \left. \right) 
\quad (9.3.4-i)
\]

\[
\beta_1 = \frac{125}{96} \left(1 - \frac{11v^2}{30} + \frac{218729v^4}{144} + \frac{4969804253v^6}{90720} - \frac{673623326819v^8}{2721600} - \\
\frac{304273318350997v^{10}}{46656000} + \frac{337775919756582144313v^{12}}{27433728000} + O(\nu^3) \right) 
\quad (9.3.4-ii)
\]

\[
\beta_2 = \frac{125}{144} \left(1 + \frac{11v^2}{30} - \frac{3280991v^4}{720} + \frac{4365325187v^6}{30240} + \frac{115179351529v^8}{1814400} + \\
\frac{276014322723379v^{10}}{15552000} + \frac{2985268375870834333v^{12}}{9144576000} + O(\nu^3) \right) 
\quad (9.3.4-iii)
\]

\[
\beta_3 = \frac{125}{144} \left(1 + \frac{11v^2}{30} - \frac{3281509v^4}{720} + \frac{3699937313v^6}{30240} - \frac{198757705399v^8}{453600} - \\
\frac{255272260074121v^{10}}{15552000} - \frac{2503319588144210317v^{12}}{9144576000} + O(\nu^3) \right) 
\quad (9.3.4-iv)
\]

\[
\beta_4 = \frac{125}{96} \left(1 - \frac{11v^2}{30} + \frac{218771v^4}{144} - \frac{2973583247v^6}{90720} + \frac{372098638237v^8}{5443200} + \\
\frac{234401042464003v^{10}}{46656000} - \frac{1960667278140426337v^{12}}{27433728000} + O(\nu^3) \right) 
\quad (9.3.4-v)
\]

\[
\beta_5 = \frac{95}{288} \left(1 + \frac{55v^2}{114} - \frac{3280235v^4}{2736} + \frac{2186348165v^6}{114912} + \frac{1641506735v^8}{689472} - \\
\frac{41492416771241v^{10}}{11819520} + \frac{279364542028634873v^{12}}{6949877760} + O(\nu^3) \right) 
\quad (9.3.4-vi)
\]

Thus, the method in equation (9.3.1) is now written as

\[
y_{n+5} = y_n + 5h \left[ \beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2} + \beta_3 f_{n+3} + \beta_4 f_{n+4} + \beta_5 f_{n+5} \right] 
\quad (9.3.5)
\]

where
\[ \beta_0 = \frac{19}{288} \left( 1 + \frac{55v^2}{114} + \frac{3282265v^4}{2736} - \frac{5513289335v^6}{114912} + \frac{258450119345v^8}{1378944} + \frac{68594391998759v^{10}}{11819520} - \frac{727108666308742877v^{12}}{6949877760} + O(v^{13}) \right) \] (9.3.6-i)

\[ \beta_1 = \frac{75}{288} \left( 1 - \frac{11v^2}{30} - \frac{218729v^4}{144} + \frac{4969804253v^6}{90720} - \frac{673623326819v^8}{2721600} - \frac{304273318350997v^{10}}{46656000} + \frac{3377759197565612413v^{12}}{27433728000} + O(v^{13}) \right) \] (9.3.6-ii)

\[ \beta_2 = \frac{50}{288} \left( 1 + \frac{11v^2}{30} + \frac{3280991v^4}{720} - \frac{4365325187v^6}{30240} + \frac{1151739351529v^8}{1814400} + \frac{276014322723379v^{10}}{15552000} - \frac{2985268375870083433v^{12}}{9144576000} + O(v^{13}) \right) \] (9.3.6-iii)

\[ \beta_3 = \frac{50}{288} \left( 1 + \frac{11v^2}{30} - \frac{3281509v^4}{720} + \frac{3699937313v^6}{30240} - \frac{198757705399v^8}{453600} - \frac{255272260074121v^{10}}{15552000} + \frac{2503319588144210317v^{12}}{9144576000} + O(v^{13}) \right) \] (9.3.6-iv)

\[ \beta_4 = \frac{75}{288} \left( 1 - \frac{11v^2}{30} + \frac{218771v^4}{144} - \frac{2973583247v^6}{90720} + \frac{372098638237v^8}{5443200} + \frac{234401042464003v^{10}}{46656000} - \frac{1960667278140426337v^{12}}{27433728000} + O(v^{13}) \right) \] (9.3.6-v)

\[ \beta_5 = \frac{19}{288} \left( 1 + \frac{55v^2}{114} - \frac{3280235v^4}{2736} + \frac{2186348165v^6}{114912} + \frac{1641506735v^8}{689472} - \frac{41492416771241v^{10}}{11819520} + \frac{279364542028634873v^{12}}{6949877760} + O(v^{13}) \right) \] (9.3.6-vi)

By taking (9.3.5) for solving oscillatory problems we have

\[ y_{n+1} = y_n + h \left[ w_1 f_1 + w_2 f_{n+1} + w_3 f_{n+2} + w_4 f_{n+3} + w_5 f_{n+4} + w_6 f_{n+5} \right] \] (9.3.7)
where \( w_1 = \frac{19}{288}, w_2 = \frac{75}{288}, w_3 = w_4 = \frac{50}{288}, w_5 = \frac{75}{288}, w_6 = \frac{19}{288} \) and we replace \( k_1 = f_1, k_2 = f_{1+\frac{1}{2}}, k_3 = f_{1+\frac{3}{2}}, k_4 = f_{1+2}, k_5 = f_{1+3} \) and \( k_6 = f_{1+4} \) in the form

\[
\begin{align*}
  k_1 &= f(y_n) \\
  k_2 &= f(y_n + ahk_1) \\
  k_3 &= f(y_n + ahk_1 + \left(\frac{1}{2} - a_1\right)hk_2) \\
  k_4 &= f(y_n + ahk_1 + a_1hk_2 + \left(\frac{3}{2} - a_3 - a_4\right)hk_3) \\
  k_5 &= f(y_n + a_1hk_1 + a_2hk_2 + a_3hk_3 + \left(\frac{1}{2} - a_5 - a_6 - a_7\right)hk_4) \\
  k_6 &= f(y_n + a_1hk_1 + a_2hk_2 + a_3hk_3 + a_4hk_4 + a_5hk_5 + a_6hk_6 + (1 - a_9 - a_6 - a_7 - a_8)hk_7)
\end{align*}
\]

and \( y_{n+1} = y_n + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4 + w_5k_5 + w_6k_6] \) (9.3.8)

Through the standard procedure of adjustment of the parameters, a new method of fifth order accuracy is obtained for equation (9.3.8) from the solution of the 11 equations of conditions, i.e.,

\[
\begin{align*}
  h^2f_{y_1} &= 5 - 25\times x(1) = 0 \quad (9.3.9-i) \\
  h^2f_{y_2} &= -76 - 100\times x(1) + 38\times x(10) + 19\times x(11) + 250\times x(1)\times x(2) + 100\times x(3) + 100\times x(4) - 250\times x(1)\times x(4) + 225\times x(5) + 225\times x(6) - 375\times x(1)\times x(6) + 75\times x(7) + 76\times x(8) + 76\times x(9) - 95\times x(1)\times x(9) = 0 \quad (9.3.9-ii) \\
  h^2f_{y_3} &= 1 - 25\times x(1)\times x(2) = 0 \quad (9.3.9-iii) \\
  h^2f_{y_4} &= -288 - 300\times x(1) + 228\times x(10) - 190\times x(1)\times x(10) + 114\times x(11) + 750\times x(1)\times x(2) + 475\times x(1)\times x(10)\times x(2) + 600\times x(3) + 500\times x(1)\times x(3) + 190\times x(11)\times x(3) - 1250\times x(1)\times x(2)\times x(3) + 600\times x(4) - 1000\times x(1)\times x(4) + 190\times x(11)\times x(4) - 475\times x(1)\times x(11)\times x(4) - 1250\times x(1)\times x(2)\times x(4) + 735\times x(5) - 285\times x(10)\times x(5) - 285\times x(11)\times x(5) - 750\times x(3)\times x(5) - 750\times x(4)\times x(5) + 1875\times x(1)\times x(4)\times x(5) + 735\times x(6) - 475\times x(1)\times x(6) - 285\times x(10)\times x(6) + 475\times x(1)\times x(10)\times x(6) - 285\times x(11)\times x(6) + 1875\times x(1)\times x(11)\times x(6) + 475\times x(1)\times x(11)\times x(6) - 750\times x(4)\times x(6) + 1875\times x(1)\times x(4)\times x(6) + 545\times x(7) - 750\times x(1)\times x(7) - 95\times x(10)\times x(7) - 95\times x(11)\times x(7) + 1875\times x(1)\times x(2)\times x(7) - 750\times x(3)\times x(7) - 750\times x(4)\times x(7) + 1875\times x(1)\times x(4)\times x(7) + 228\times x(8) - 336
\end{align*}
\]
\begin{align*}
285 x(5)x(8) - 285 x(6)x(8) + 475 x(1)x(6)x(8) - \\
95 x(7)x(8) + 228 x(9) - 285 x(5)x(9) - \\
285 x(6)x(9) + 475 x(1)x(6)x(9) - 95 x(7)x(9) &= 0 \\
(9.3.9-iv) \\

h'^2 f_{fs} : & -1124-400x(1)-500x(1)^2+608x(10)+323x(11) + \\
1000x(1)x(2)+1250x(1)^2 x(2)+
800x(3)+800x(4)-1500x(1)x(4)-
1250x(1)^2 x(4)+2475x(5)+2475x(6)-
3000x(1)x(6)-1875x(1)^2 x(6)+975x(7)+
1064x(8)+1064x(9)-950x(1)x(9) - \\
475x(1)^2 x(9) &= 0 \\
(9.3.9-v) \\

h'^4 f_{f_{pp}} : & 1 - 125x(1)^3 = 0 \\
(9.3.9-vi) \\

h'^2 f'^2 : & -156-1800x(1)+456x(10)+456x(11) - \\
570x(1)x(11) + 4500x(1)x(2) + \\
1425x(1)x(11)x(2)+760x(3)+3000x(1)x(3)-
760x(10)x(3) - 760x(11)x(3) + \\
950x(1)x(11)x(3)-7500x(1)x(2)x(3)-
2375x(1)x(11)x(2)x(3)+760x(4)+1100x(1)x(4)-
760x(10)x(4) + 1900x(1)x(10)x(4) - \\
760x(11)x(4) + 2850x(1)x(11)x(4) - \\
7500x(1)x(2)x(4)-2375x(1)x(11)x(2)x(4)+
570x(5)+2250x(1)x(5)-570x(10)x(5)-
570x(11)x(5)-5625x(1)x(2)x(5)-950x(3)x(5)-
3750x(1)x(3)x(5)+950x(10)x(3)x(5)+
950x(11)x(3)x(5)+9375x(1)x(2)x(3)x(5)-
950x(4)x(5)-1375x(1)x(4)x(5)+
950x(10)x(4)x(5)-2375x(1)x(10)x(4)x(5)+
950x(11)x(4)x(5)-2375x(1)x(11)x(4)x(5)+
9375x(1)x(2)x(4)x(5)+570x(6)+2250x(1)x(6)-
570x(10)x(6)-570x(11)x(6) - \\
5625x(1)x(2)x(6)-950x(3)x(6)-
3750x(1)x(3)x(6)+950x(10)x(3)x(6)+
950x(11)x(3)x(6)+9375x(1)x(2)x(3)x(6)-
950x(4)x(6)-1375x(1)x(4)x(6)+
950x(10)x(4)x(6)-2375x(1)x(10)x(4)x(6)+
950x(11)x(4)x(6)-2375x(1)x(11)x(4)x(6)+
9375x(1)x(2)x(4)x(6)+570x(7)+1300x(1)x(7)-
570x(10)x(7)+950x(1)x(10)x(7) - \\
570x(11)x(7)+950x(1)x(11)x(7)-
3250x(1)x(2)x(7)-2375x(1)x(10)x(2)x(7)-
2375x(1)x(11)x(2)x(7)-950x(3)x(7)-
3750x(1)x(3)x(7)+950x(10)x(3)x(7)+
950x(11)x(3)x(7)+9375x(1)x(2)x(3)x(7)-
950x(4)x(7)-1375x(1)x(4)x(7)+
950x(10)x(4)x(7)-2375x(1)x(10)x(4)x(7)+
950x(11)x(4)x(7)-2375x(1)x(11)x(4)x(7)+
9375x(1)x(2)x(4)x(7)+456x(8)-760x(3)x(8)-
760x(4)x(8)+1900x(1)x(4)x(8) - \\
337
\end{align*}
\[ 570 \times x(5) \times x(8) + 950 \times x(3) \times x(5) \times x(8) +
950 \times x(4) \times x(5) \times x(8) - 2375 \times x(1) \times x(4) \times x(5) \times x(8) -
570 \times x(6) \times x(8) +
950 \times x(3) \times x(6) \times x(8) + 950 \times x(4) \times x(6) \times x(8) -
2375 \times x(1) \times x(4) \times x(6) \times x(8) - 570 \times x(7) \times x(8) +
950 \times x(3) \times x(7) \times x(8) + 950 \times x(4) \times x(7) \times x(8) -
2375 \times x(1) \times x(4) \times x(7) \times x(8) + 456 \times x(9) - 760 \times x(3) \times x(9) -
760 \times x(4) \times x(9) + 1900 \times x(1) \times x(4) \times x(9) -
570 \times x(5) \times x(9) + 950 \times x(3) \times x(5) \times x(9) +
950 \times x(4) \times x(5) \times x(9) - 2375 \times x(1) \times x(4) \times x(5) \times x(9) -
570 \times x(6) \times x(9) +
950 \times x(3) \times x(6) \times x(9) + 950 \times x(4) \times x(6) \times x(9) -
2375 \times x(1) \times x(4) \times x(6) \times x(9) - 570 \times x(7) \times x(9) +
950 \times x(1) \times x(7) \times x(9) - 2375 \times x(1) \times x(4) \times x(7) \times x(9) +
950 \times x(3) \times x(7) \times x(9) + 950 \times x(4) \times x(7) \times x(9) -
2375 \times x(1) \times x(4) \times x(7) \times x(9) = 0 \]
\[2250 \times x(6) \times x(7) + 3750 \times x(1) \times x(6) \times x(7) - 375 \times x(7)^2 + 1520 \times x(10)^2 - 760 \times x(11)^2 - 2185 \times x(7)^2 - 1520 \times x(8) \times x(10) - 760 \times x(11) - 5985 \times x(5)^2 - 5985 \times x(6)^2 + 8550 \times x(1) \times x(6)^2 - 2375 \times x(11)^2 - 3800 \times x(1)^2 - 1520 \times x(10)^2 + 1900 \times x(1) \times x(10)^2 - 760 \times x(11) - 950 \times x(1)^2 \times x(6)^2 + 8550 \times x(1) \times x(6)^2 - 2185 \times x(7)^2 - 3040 \times x(8)^2 + 3800 \times x(1) \times x(9)^2 - 2375 \times x(11)^2 = 0 \]

\[\text{(9.3.9-viii)}\]

\[h^tf^f_{\mu} : -328 - 200 \times x(1)^2 + 228 \times x(10) + 133 \times x(11) + 500 \times x(1)^2 \times x(2) + 120 \times x(3) + 120 \times x(4) - 750 \times x(1)^2 \times x(4) + 540 \times x(5) + 540 \times x(6) - 1500 \times x(1)^2 \times x(6) + 300 \times x(7) + 304 \times x(8) + 304 \times x(9) - 475 \times x(1)^2 \times x(9) = 0 \]

\[\text{(9.3.9-ix)}\]

\[h^tf^f_{\mu} : -6436 - 1200 \times x(1)^3 - 2500 \times x(1)^3 + 3914 \times x(10)^2 + 2128 \times x(11) + 300 \times x(1)^3 \times x(2) + 3100 \times x(1)^3 \times x(4) + 3100 \times x(1)^3 \times x(4) + 12825 \times x(5) + 12825 \times x(6) - 18000 \times x(1)^3 \times x(6) + 12825 \times x(7) + 6916 \times x(8) + 6916 \times x(9) - 7125 \times x(1)^3 \times x(9) - 2375 \times x(1)^3 \times x(9) = 0 \]

\[\text{(9.3.9-x)}\]

\[h^tf^f_{\mu} : 1 - 625 \times x(1)^4 = 0 \]

\[\text{(9.3.9-xi)}\]

where \(x(1) = a_1, x(2) = a_2, x(3) = a_3, x(4) = a_4, x(5) = a_5, x(6) = a_6, x(7) = a_7, x(8) = a_8, x(9) = a_9, x(10) = a_{10}, x(11) = a_{11}\).

Equations \(9.3.9-i)-(9.3.9-xi)\) are solved simultaneously using the NAG routine (Subroutine C05NBF) for solving a system of non-linear equations to give the required parameters, i.e.,

\[a_1 = 0.2000000000, \quad a_2 = -0.2027706499, \quad a_3 = 0.5320433054, \quad a_4 = -0.8706804955, \quad a_5 = -0.421500613, \quad a_6 = 1.3566987431, \quad a_7 = -0.4405835645, \quad a_8 = 1.0285379694, \quad a_9 = -1.7032738709, \quad a_{10} = 1.1167966402, \quad a_{11} = -0.1408342610\]

\[a_{12} = \frac{2}{5} - a_2 = 0.6027706499, \quad a_{13} = \frac{3}{5} - a_3 - a_6 = 0.9386371900\]
Thus, the new fifth order method for solving oscillatory problems can be written as follows:
\[ y_{n+1} = y_n + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4 + w_5k_5 + w_6k_6] \]  
(9.3.11)

where \( k_i = f(y_n) \)
\[ k_2 = f(y_n + 0.2000000000hk_1) \]
\[ k_3 = f(y_n - 0.2027706499hk_1 + 0.6027706499hk_2) \]
\[ k_4 = f(y_n + 0.5320433054hk_1 - 0.8706804955hk_2 + 0.9386371900hk_3) \]
\[ k_5 = f(y_n - 0.4215000613hk_1 + 1.3566987431hk_2 - 0.4405835645hk_3 \]
\[ + 0.3053848827hk_4) \]
\[ k_6 = f(y_n + 1.0285379694hk_1 - 1.7032738709hk_2 + 1.1167966402hk_3 \]
\[ - 0.1408342610hk_4 + 0.6987735224hk_5) \]  
(9.3.12)
\[ w_i = \frac{75}{288} \left( 1 - \frac{11v^2}{30} + \frac{21871v^4}{144} - \frac{2973583247v^6}{90720} + \frac{372098638237v^8}{5443200} + \frac{234401042464003v^{10}}{46656000} - \frac{1960667278140426337v^{12}}{27433728000} + 0(v^{13}) \right) \]

\[ w_e = \frac{19}{288} \left( 1 + \frac{55v^2}{114} - \frac{3280235v^4}{2736} + \frac{2186348165v^6}{114912} + \frac{1641506735v^8}{689472} - \frac{41492416771241v^{10}}{11819520} + \frac{279364542028634873v^{12}}{6949877760} + 0(v^{13}) \right) \]

By the use of Mathematica to rationalize the parameters in equations (9.3.10), we obtain

\[
\begin{align*}
    a_1 &= \frac{1}{5}, \quad a_2 = \frac{-122711}{230641}, \quad a_3 = \frac{11797}{23249}, \quad a_4 = \frac{-127014}{145879} \\
    a_5 &= \frac{-24047}{57051}, \quad a_6 = \frac{70916}{52271}, \quad a_7 = \frac{-80844}{183493}, \quad a_8 = \frac{73776}{71729} \\
    a_9 &= \frac{-90994}{53423}, \quad a_{10} = \frac{143065}{128103}, \quad a_{11} = \frac{-20369}{144631} \\
    a_{12} &= \frac{2}{5} - a_2 = \frac{37115}{61574}, \quad a_{13} = \frac{3}{5} - a_3 - a_4 = \frac{64536}{68755} \\
    a_{14} &= \frac{4}{5} - a_5 - a_6 - a_7 = \frac{41303}{135249} \\
    a_{15} &= 1 - a_8 - a_9 - a_{10} - a_{11} = \frac{54809}{78436}
\end{align*}
\]

and the new fifth order method for solving oscillatory problems can be written in rational form as

\[ y_{n+1} = y_n + h \left[ w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4 + w_5k_5 + w_6k_6 \right] \]

where \( k_1 = f(y_n) \)

\[ k_2 = f \left( y_n + \frac{1}{5}hk_1 \right) \]

\[ k_3 = f \left( y_n - \frac{11695}{57676}hk_1 + \frac{37115}{61574}hk_2 \right) \]
By substituting the values \( a_i, 1 \leq i \leq 11 \) in equation (9.3.10) into equations (9.3.11) and (9.3.12) using Mathematica and evaluating all the terms up to \( (h^6) \) to represent the local truncation error for this method, we have

\[
LTE = h^6 \left[ -0.000204145 f y' - 0.000938195 f^2 y'' f y' + 0.000176875 f^3 y'' f y' \\
- 0.0004346 f^3 y'' f y' + 0.0003058 f^4 y'' y y'' + 0.000081732 f^4 y'' y y'' - 4.33681 \times 10^{-9} f^5 y'' y y'' \right]
\] (9.3.14)

9.3.2 Stability Analysis

We examine the stability region for the new fifth order method for solving the oscillatory problem with the test equation \( y' = \lambda y \) and we obtain

\[
k_1 = \lambda y_n
\]
\[
k_2 = \lambda (y_n + 0.20000000000 h k_1)
\]
\[
k_3 = \lambda (y_n - 0.2027706499 h k_2 + 0.6027706499 h k_1)
\]
\[
k_4 = \lambda (y_n + 0.5320433054 h k_2 - 0.8706804955 h k_2 + 0.9386371900 h k_1) + 0.3053848827 h k_4
\]
\[
k_5 = \lambda (y_n + 1.0285379694 h k_2 - 1.7032738709 h k_2 + 1.1167966402 h k_1 - 0.1408342610 h k_4 + 0.6987735224 h k_3)
\] (9.3.15)
By substituting $k_i, 1 \leq i \leq 6$ in equation (9.3.15) and $w_i, 1 \leq i \leq 6$ in equation (9.3.13) into the new fifth order formula, i.e.,

$$y_{in1} = y_n + h \left[ w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 + w_5 k_5 + w_6 k_6 \right]$$  \hspace{1cm} (9.3.16)

we obtain

$$y_{m1} = y_n + (h \lambda)y_n + 0.5(h \lambda)^2 y_n + 0.166667(h \lambda)^3 y_n + 0.416667(h \lambda)^4 y_n + 0.00833333(h \lambda)^5 y_n + 0.00159303(h \lambda)^6 y_n + O(h^7) \ldots$$  \hspace{1cm} (9.3.17)

By rationalizing the coefficients in equation (9.3.17) we have

$$y_{m1} = y_n + (h \lambda)y_n + \frac{1}{2}(h \lambda)^2 y_n + \frac{1}{6}(h \lambda)^3 y_n + \frac{1}{24}(h \lambda)^4 y_n + \frac{1}{120}(h \lambda)^5 y_n + \frac{15}{9416}(h \lambda)^6 y_n + O(h^7) \ldots$$  \hspace{1cm} (9.3.18)

By substituting $h \lambda = z$ in (9.3.18), we can show that

$$y_{m1} = y_n + y_n \left[ z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{15}{9416} z^6 \right] + O(z^7) \ldots$$  \hspace{1cm} (9.3.19)

Following equation (3.3.30), we write $\frac{y_{m1}}{y_n} = Q$ in the equation (9.3.19), to obtain

$$Q = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 + \frac{1}{120}z^5 + \frac{15}{9416} z^6 + O(z^7) \ldots$$  \hspace{1cm} (9.3.20)

We now determine the stability region of this new fifth order formula in the complex plane that satisfy the condition in equation (3.3.31) as

$$\left| 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{15}{9416} z^6 \right| < 1 \ldots$$  \hspace{1cm} (9.3.21)
By the use of Mathematica, we can plot the graphic surface defined by equation (9.3.21) i.e.,

![Figure 9.3: Graphic surface defined by new fifth order method](image)

and the stability region defined by the formula in equation (9.3.21) as shown in Figure 9.4

![Figure 9.4: Stability region for new fifth order method](image)

9.3.3 Numerical Example

We consider the problem of solving the system of equations

\[ y'' + 100(1 - \alpha \cos(2x))y = 0 \quad (9.3.22) \]

which are oscillatory functions. The oscillatory problem in (9.3.22) can also be written as
\[
\frac{dy}{dx} = \begin{bmatrix}
0 & 1 \\
-100(1 - \alpha \cos(2x)) & 0
\end{bmatrix} y , \quad y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} .
\] (9.3.23)

In this example we solve this problem with \( \alpha = 0.1 \) and \( h = 0.02 \) by using formula (9.2.25-i)-(9.2.25-ii) and (9.3.11)-(9.3.12) with \( \omega = 1 \) and the Runge-Kutta type formula given by (9.2.25-i) and (9.3.11) with \( \omega = 0 \). By putting \( \omega = 0 \) formula (9.2.25-i) and (9.3.11) is equivalent to using a fourth order classical Runge-Kutta formula which has an algebraic order 4 and 5. Putting \( \omega = 1 \) in (9.2.25-i)-(9.2.25-ii) and (9.3.11)-(9.3.12) is equivalent to using the new Runge-Kutta formula for oscillatory problems which has an algebraic order 4 and 5 with a trigonometric order 2.

The errors in the numerical solution with the frequency \( \omega = 1 \) by using the new fifth order method in equation (9.3.11)-(9.3.12) are compared with the fourth order formula in equations (9.2.25-i)-(9.2.25-ii) and are shown in Tables 9.1 and 9.2.

Table 9.1: Fourth order and fifth order method for solving problem (9.3.23) with \( \omega = 0 \).

<table>
<thead>
<tr>
<th>( x )</th>
<th>Fourth Order</th>
<th>Fifth Order</th>
<th>Exact 7 Digit Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000000</td>
<td>1.0000000</td>
<td>1.0000000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0691559</td>
<td>0.0692086</td>
<td>0.0692085</td>
</tr>
<tr>
<td>1.0</td>
<td>-0.9084381</td>
<td>-0.9084172</td>
<td>-0.9084179</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.6938100</td>
<td>-0.6939603</td>
<td>-0.6939608</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2312066</td>
<td>0.2309582</td>
<td>0.2309590</td>
</tr>
<tr>
<td>2.5</td>
<td>0.9763441</td>
<td>0.9763679</td>
<td>0.9763699</td>
</tr>
<tr>
<td>3.0</td>
<td>0.2053593</td>
<td>0.2057661</td>
<td>0.2057667</td>
</tr>
<tr>
<td>3.5</td>
<td>-0.9617184</td>
<td>-0.9616773</td>
<td>-0.9616794</td>
</tr>
<tr>
<td>4.0</td>
<td>-0.4260467</td>
<td>-0.4265307</td>
<td>-0.4265317</td>
</tr>
<tr>
<td>4.5</td>
<td>0.6026120</td>
<td>0.6022346</td>
<td>0.6022367</td>
</tr>
<tr>
<td>5.0</td>
<td>0.9415266</td>
<td>0.9417339</td>
<td>0.9417373</td>
</tr>
</tbody>
</table>

345
Table 9.2: Fourth order and fifth order method for solving problem (9.3.23) with $\omega = 1$.

<table>
<thead>
<tr>
<th>$x$</th>
<th>Fourth Order</th>
<th>Fifth Order</th>
<th>Exact 7 Digit Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0000000</td>
<td>1.0000000</td>
<td>1.0000000</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0691559</td>
<td>0.0692078</td>
<td>0.0692085</td>
</tr>
<tr>
<td>1.0</td>
<td>-0.9084383</td>
<td>-0.9084167</td>
<td>-0.9084179</td>
</tr>
<tr>
<td>1.5</td>
<td>-0.6938103</td>
<td>-0.6939573</td>
<td>-0.6939608</td>
</tr>
<tr>
<td>2.0</td>
<td>0.2312068</td>
<td>0.2309610</td>
<td>0.2309590</td>
</tr>
<tr>
<td>2.5</td>
<td>0.9763449</td>
<td>0.9763651</td>
<td>0.9763699</td>
</tr>
<tr>
<td>3.0</td>
<td>0.2053594</td>
<td>0.2057601</td>
<td>0.2057667</td>
</tr>
<tr>
<td>3.5</td>
<td>-0.9617195</td>
<td>-0.9616746</td>
<td>-0.9616794</td>
</tr>
<tr>
<td>4.0</td>
<td>-0.4260471</td>
<td>-0.4265225</td>
<td>-0.4265317</td>
</tr>
<tr>
<td>4.5</td>
<td>0.6026129</td>
<td>0.6022370</td>
<td>0.6022367</td>
</tr>
<tr>
<td>5.0</td>
<td>0.9415281</td>
<td>0.9417265</td>
<td>0.9417373</td>
</tr>
</tbody>
</table>

From Table 9.1, by using $\omega = 0$ in formula (9.2.25-i)-(9.2.25-ii) and (9.3.11)-(9.3.12) the new fifth order Runge-Kutta equivalent formula is more accurate than the fourth order formula. Thus, from results in Tables 9.1 and 9.2 we can see that, the higher order formula gives results comparable to those given by Gautschi [1961] and improves the results given by the fourth order formula of Sanugi [1986].
CHAPTER 10

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK
The work presented in this thesis can be summarized into four parts. The first part is concerned with the numerical solution of problems involving ODEs using fourth order non-linear integration formulae with a variety of means based on $C_M$, $C_M$ and $RMS$. The second part is the extension of the concepts and techniques of the $C_M$, $C_M$ and $RMS$ strategies to the Runge-Kutta method, the implementation of the extrapolation processes and the parallelization strategies used, the theory of RK(4,4) method and the solution of a special class of second order ODEs. The third part is concerned with the development of a new fifth order five stage explicit Runge-Kutta method based on $AM$ and $C_M$ techniques. The final part in this thesis is concerned with the development of new fourth and fifth order methods for solving oscillatory problems by the use of trigonometric polynomial interpolation.

In Chapter 4, a detailed study was carried out on three types of non-linear formula based on $C_M$, $C_M$ and $RMS$ concepts together with the analysis of its local truncation error (LTE), accuracy and its stability properties. In terms of accuracy, it was found that the two fourth order formula i.e., $C_M$ and $C_M$ gives better accuracy when compared with the fourth order classical Runge-Kutta method but the fourth order $RMS$ method was less encouraging when compared with the more established methods. For the implicit Runge-Kutta method, we concluded that the 2-stage scheme for the implicit $H_{4,4}$ method attained nearly fourth order accuracy compared to the other non-linear 2-stage third order methods. For these three methods, $C_M$, $C_M$ and $RMS$ it was shown that when these formulas are considered, a new class of modified Trapezoidal formula with L-stability is obtained. Our numerical examples show that the modified contraharmonic mean ($MC_M$) gives greater accuracy but for a system of stiff equations in the same class of non-
linear modified Trapezoidal formula, the modified centroidal mean (\(MC,M\)) method performs better.

In Chapter 5, we compare the speedups obtained on a parallel computer by using a second order \(C,M\) method together with the extrapolation technique for both sequential and parallel programs. By solving the numerical solution of the ODEs and system of ODEs, the best speedups achieved was less than 2 no matter how many processors were used. However, by implementing the parallel program on the explicit data and task assignment (EXDATA) schedule, linear speedups was achieved very close to the ideal.

In Chapter 6, we propose a new strategy for adaptive error control by using two different Runge-Kutta methods with the same order \(p\). The combination of the fourth order classical Runge-Kutta method and the \(C,M\) method of RK(4,4) form. The numerical solution of ODEs by RK(4,4), Merson and RKF(4,5) by Fehlberg are comparable in terms of the time taken and the accuracy obtained. For this reason, the new adaptive RK(4,4) method gives an alternative method for solving ODEs problems. Further study of the combinations with the other two fourth order method are planned.

In Chapter 7 we investigated the feasibility of extending the \(C,M\), \(C,M\) and \(H,M\) approach of deriving numerical methods for the solution of ODEs. For the second order ODE problems of special type, we have obtained the \(C,M\), \(C,M\) and \(H,M\) modified form of the Numerov method. The \(C,M\), \(C,M\) and \(H,M\) versions of the Numerov method is found to be comparable in accuracy with the classical Numerov method. We have also investigated solving ODEs with a variety of means 2-step methods. Numerical results for selected problems show that the \(GM\), \(C,M\), \(C,M\) and \(H,M\) second order implicit method are unstable as with the second order explicit midpoint method.
In Chapter 8, a new explicit fifth order linear Runge-Kutta method with 5-stages for solving initial value problems based on the arithmetic mean ($AM$) formulation, i.e., $y_{n+1} = y_n + h\left[\sum_{i=1}^{5} w_i \left(\frac{k_i + k_{ni}}{2}\right)\right]$ where $\sum_{i} w_i = 1$ and $k_i, 1 \leq i \leq 5$ was derived. The derivation of this linear method is given in detail along with the LTE and stability analysis. Experimental results show that, the new fifth order linear Runge-Kutta method is superior when compared with the fourth order classical Runge-Kutta method, fourth order 5-stages RK4(5)-Merson method and the fifth order 6-stages RK5(6)-Nyström method. In the present state of knowledge, these new fifth order methods with 5-stages have made a significant contribution to the numerical solution of linear ODEs. We also study a fifth order with 5-stages Runge-Kutta method based on the $CM$. The combination of these two fifth order method form an adaptive RK(5,5) method. In addition, studies of a fifth order Runge-Kutta method was also carried out for solving second order ODEs in the form $y'' = f(x, y, y')$ and the numerical results obtained are convincing. Further studies of fifth order Runge-Kutta methods for the variety of means is warranted especially for solving non-linear initial value problems in ODEs.

Finally, in Chapter 9 we are concerned with the numerical solution of periodic problems involving ODEs. With the help of Mathematica, we have developed fourth and fifth order methods for solving oscillatory problems by the use of trigonometric interpolation. The numerical results obtained show that the fifth order method gives results comparable to those given by Gautschi [1961] and improves the results given by a fourth order formula of Sanugi [1986].
REFERENCES


Evans, D.J. and N.Yaacob. [1994], A Fourth Order Runge-Kutta Method Based On The Heronian Mean Formula, Loughborough University of Technology, Department of Computer Studies, Report No. 879.


Evans, D.J., and B.B. Sanugi. [1993], A New Fourth Order Runge-Kutta Formula Based On Harmonic Mean, Loughborough University of Technology Department of Computer Studies, Report No. 799.


Evans, D.J., and N.M. Bahoshy. [1994], Parallel Romberg Integration of Multidimensional Integrals, Loughborough University of Technology, Department of Computer Studies, Report No. 874.


Gear, C.W. [1987], Parallel Methods for Ordinary Differential Equations, University of Illinois at Urbana-Champaign, Report No. UIUCDCS-R-87-1369.


Ralston , A. [1962], Runge - Kutta Methods With Minimum Error Bounds ,

Rauber, T., and G. Runger. [1993], Solving ODE's on Distributed Memory
Multiprocessors, Universitat des Saarlandes , Computer Science Department ,
Germany.

Saman, M.Y. [1993], Automatic Parallelization of Programs, PhD Thesis,
Loughborough University of Technology, UK.

Sanugi, B.B. [1986], New Numerical Strategies For Initial Value Type Ordinary

Sanugi, B.B. [1993], A New Fourth Order Runge-Kutta Method Based On The
Harmonic Mean , Loughborough University of Technology , Department of
Computer Studies , Report No. 799.

Scraton, R.E. [1986], Basic Numerical Method : An Introduction to Numerical
Mathematics on a Microcomputer, Edward Arnold, London.

Shampine, L.F and H.A. Watts. [1977],"The Art Of Writting A Runge-Kutta Code,

Shampine, L.F and H.A. Watts. [1979],"The Art Of Writting A Runge-Kutta Code,
Part 2 , Applied Math Comp, Vol 5, pp. 93-121

Shanks, E.B. [1966], Solutions of Differential Equations by Evaluations of Functions,

Wazwaz, A.M. [1993], On The Numerical Solution of $y' = f(x, y)$ By A Class of
Nonlinear Trapezoidal Formulas , Department of Maths. and Comp. Science,
Saint Xavier University, Chicago, Illinois.

Wazwaz, A.M. [1994], A Comparison of Modified Runge-Kutta Formula Based On
Widlund, O.B. [1967], A Note on Unconditionally Stable Linear Multistep Methods, B.I.T., 7, pp. 65-70.


Yaakub, A.R. and D.J.Evans. [1996], New L-Stable Modified Trapezoidal Methods For The Initial Value Problems, Loughborough University of Technology, Department of Computer Studies, Report No.1009

Yaakub, A.R. and D.J.Evans. [1996], High Order Integration Formula Using Trigonometric Polynomials for Periodic IVPs, Loughborough University, Department of Computer Studies, Report No.1017.
APPENDICES
APPENDIX 1

/* This program solves a system of ordinary equation by using the fourth order Contraharmonic mean method with extrapolation. */

#include<stdio.h>
#include<math.h>
#include<paralle/lparalle h>
#include<parallellmicrotask.h>
#define m3
shared double tableau[100][100], H;
double exact, yend;
int si, time1, time2;
shared int xO,xf,levels,logp;
shared double Y0[1000],htam[1000],xh[1000],h[1000];
void func3(),find_func_values(), tabulate(), print_table();

/* **************************************************************************
*/

main()
{
    int i,j,nprocs;
    printf("Enter(integers only) x0,xf,nprocs \n");
    scanf("%d %d %d", &xO,&xf,&nprocs);
    levels = (int)(log((double)1024.0)/log(2.0));
    logp = (int)(log((double)1.0)/log(2.0));
    H = (double)0.5;
    Y0[1]=0.0;
    Y0[2]=1.0;
    Y0[3]=0.0;
    time1 = clock_time();
    m_set__procs(nprocs);
    m_fork(find_func_values);
    tabulate();
    time2 = clock_time();
    printf("x = %d\n",xf);
    printf("time for %d processor is = %f\n",m_get_numprocs(),(float)(time2 - time1)/100.0);
    printf("\n");
    return;
}

/* ---------------- end of main() ---------------------------------------------*/
void find_func_values()
{
    int p, i,j,nprocs;
    double yy[1000],zz[1000],k1[1000],k2[1000],k3[1000],k4[1000];
    nprocs = m_get_numprocs();
    for (j=m_get_myid()+1; j<=levels-logp; j+=nprocs){
        p=(int)pow(2.0,(double)j);
        h[j]= H/p;
        xh[j]= x0;
        htm[j]=0.5*h[j];
        for (i=1; i<=m; i++) {
            yy[i]=Y0[i];
        }  
        do {
            func3(yy,k1);
            for (i=1; i<=m; i++) {
                zz[i]=yy[i]+htm[j]*k1[i];
            }
            func3(zz,k2);
            for (i=1; i<=m; i++) {
                zz[i]=yy[i]+h[j]*k1[i]/8.0+3.0*h[j]*k2[i]/8.0;
            }
            func3(zz,k3);
            for (i=1; i<=m; i++) {
                zz[i]=yy[i]+h[j]*k1[i]/4.0+3.0*h[j]*k2[i]/4.0+3.0*h[j]*k3[i]/2.0;
            }
            func3(zz,k4);
            for (i=1; i<=m; i++) {
                yy[i] +=(h[j]/3.0)*((k1[i]*k1[i]+k2[i]*k2[i]/(k1[i]+k2[i]))+
                ((k2[i]*k2[i]+k3[i]*k3[i])/(k2[i]+k3[i]))+
                ((k3[i]*k3[i]+k4[i]*k4[i])/(k3[i]+k4[i])));
            }
            xh[j] +=h[j];
        } while ( xh[j] < xf );
    tableau[1][j] = yy[2];
    printf("h[%d] = %f, yy[%d] = %f \n",j,h[j] ,xh[j], tableau[1][j]);
    printf("\n");
    return;
}

/* ---------------- end of find_func_values() --------------------------------.InputStreamReader*/
void tabulate()
{
int i, j, k;
long fp;
for (i=2, j=levels; j>0; i++, j--)
{
    fp = (long)pow(2.0, (double)(i-1));
    fp = fp*fp - 1;
    for (k=1; k<=j; k++)
        tableau[i][j] = tableau[i-1][k] + (tableau[i-1][k+1]-tableau[i-1][k])/fp;
    if(fabs(tableau[i][j]-tableau[i-1][1]) <= 0.0000000001) break;
    si=i-1;
}
return;
/* ----------------
end of tabulate() -----------------------------------------------------------*/

void print_table()
{
int i, j, k;
for (i=1, j=si; j>0; i++, j--)
{
    for (k=3; k<=j; k++)
        printf("%f ", tableau[k][i]);
    printf("\n");
}
exact = exp((double)xf)*cos((double)xf)+xf*xf;
printf("\nExact Solution = %E\n", exact);
printf("\nError = %E\n", fabs(exact-yend));
printf("\n");
return;
/*----------------- defining the function-------------------------------------------*/

void func3(yy, F)
do double yy[ 1, F[ ] :
{
    F[1] = 1.0;
    return;
}
PROGRAM ERRRK(4,4)
$INCLUDE prob.f
Implicit double precision(a-h,o-z)
common/blk3/xo,xend,yo,npb,nsteps
integer time1,time2
write(*,*)'Enter The Value of npb, errtol and h'
read(*,* )npb,errtol,h
call problem
a1=1.d0/2.d0
a2=0
a3=1.d0/2.d0
a4=0
a5=0
a6=1
c1=1.d0/2.d0
c2=1.d0/8.d0
c3=3.d0/8.d0
c4=1.d0/4.d0
c5=-3.d0/4.d0
c6=3.d0/2.d0
k=1
x=x0
y=y0
icount=0
write(1,19)errtol
19 format(5x,'Error tolerance='d6.1)
write(1,21)x,y
21 Format(f19.5,d15.7)
c--------Start Timing
call _clock_time(time1)
10 continue
do 30 i=1,k
11 ak1=f(npb,x,y)
a k2=f(npb,x+a1*h,y+h*a1*ak1)
ak3=f(npb,x+(a2+a3)*h,y+h*a2*ak1+h*a3*ak2)
a k4=f(npb,x+(a4+a5+a6)*h,y+h*a4*ak1+h*a5*ak2+h*a6*ak3)
yam = y + h*(ak1 + 2*ak2 + 2*ak3 + ak4)/6.d0

c

cml = f(npb, x, y1)
cm2 = f(npb, x + cl*h, y1 + h*cl*cm1)
cm3 = f(npb, x + (c2 + c3)*h, y1 + h*c2*cm1 + h*c3*cm2)
cm4 = f(npb, x + (c4 + c5 + c6)*h, y1 + h*c4*cm1 + h*c5*cm2 + h*c6*cm3)
ycm = y1 + h*(cm1**2 + cm2**2)/(cm1 + cm2) + (cm2**2 + cm3**2)/(cm2 + cm3) +
    (cm3**2 + cm4**2)/(cm3 + cm4))**.16.d0

errest = abs((yam - ycm) * 281.d0/4608.d0)
if(abs(errest).gt.errtol) then
  h = h/2
  write(1, 99) h
99  format(1x, 'h = ', f8.5)
go to 11
end if
if(abs(errest).lt.errtol/32) then
  icount = icount + 1
  if(icount.ge.2) go to 22
  h = 2*h
  write(1, 99) h
  go to 11
end if
22  icount = 0
  x = x + h
  y = yam
  y1 = ycm
30 continue

c------ Finish Timing
  call _clock_time(time2)
exct = exact(npb, x)
err = abs(exct - y)
write(1, 20)x, exct, y, err, errest
20  format(f9.5, 4d15.7)
if(x.lt.xend) go to 10
write(1, 200)(time2 - time1)/100.0
200 format(/, 'Total Time = ', f10.2)
stop
end

365
APPENDIX 3

program ode2_cases

c-----This program investigates all possible cases of the parameter for the
c-----new CoM formula for solving the special second order ODE problems

implicit double precision (a-h,o-z)
character answer,Y,N
c-----choose problem number
write(*,*)
11 write(*,'PLEASE TYPE THE CORRECT PROBLEM NUMBER FROM 1 TO 6')
1   read*,num
call problem(num,xO,yO,xend,nsteps)
2   do 9999 ll=1,3
   print*,'INPUT THE NUMBER OF an and ad'
   read*,an,ad
   a=an/ad
   a1 = (12.*a+5.)/6.
   a2 = (6.*a+1.)/12.
   a3 = (6.*a+1.)/12.
c------ a2 = a3 and a4 = a5
   a4 = -2.*a
   a5 = -2.*a
   a6 = a
   print*,
   print*,'PARAMETER OF THE EQUATION'
   write(*,3) an, ad, a1, a2, a3, a4, a5, a6
3   format( lx, 'a = ',f5.2, '/', f7.2/*1x,6(f8.3,2x))
   print*,
xn0 = x0
yn0 = y0
h = abs(xend - x0)/nsteps
c-------use the exact solution to obtain y1,x1=x0+h
   xn = xn0+h
   yn = exact(num,xn)
write(*,7)
7   format(5x,'xn ',7x,' computed',12x,' exact',13x,'relative error')
do 10 j = 1, nsteps
c-------call predictor to obtain yn1
   call predic(h,xn,yn1,yn,yn0)
xn1 = xn + h
   yn1 = 2.*yn - yn0 + (h**2.)*(a1*fn(num,xn,yn)
   ~~ + a2*fn(num,xn0,yn0)+a3*fn(num,xn1,yn1))
   ~~ + a4*fn(num,xn,yn)*fn(num,xn1,yn1)+fn(num,xn1,yn1)*fn(num,xn1,yn1)))
   ~~ /fn(num,xn,yn)+fn(num,xn1,yn1)))
   ~~ + a5*fn(num,xn0,yn0)*fn(num,xn0,yn0)+fn(num,xn,yn)*fn(num,xn,yn))
   ~~ /fn(num,xn0,yn0)+fn(num,xn,yn)))
\[ a6^*((f(num,xn1,yn1) + f(num,xn0,yn0))*f(num,xn0,yn0)) \]

\[ \frac{(f(num,xn1,yn1) + f(num,xn0,yn0)))}{(f(num,xn1,yn1))} \]

\[ \text{compute the exact solution of the problem} \]
\[ \text{exact} = \text{exact}(num,xn) \]

\[ \text{compute the absolute difference between exact and computed solutions} \]
\[ \text{if} (\text{exct} \neq 0) \text{then} \]
\[ \text{err} = \text{abs}(\text{exct} - \text{yn})/\text{abs}(\text{exct}) \]
\[ \text{else} \]
\[ \text{err} = \text{abs}(\text{exct} - \text{yn}) \]
\[ \text{endif} \]
\[ \text{write}(*,100)xn,yn,\text{exct},\text{err} \]

\[ \text{reset appropriate values of } xn0,xn,xn1,yn0,yn \]
\[ xn0 = xn \]
\[ xn = xn1 \]
\[ yn0 = yn \]
\[ yn = yn1 \]

\(100 \text{ format(j7.4,e23.12)}\)

\(10 \text{ continue}\)

\(9999 \text{ continue}\)

\[ \text{stop} \]
\[ \text{end} \]

\[ \text{subroutine predic(h,xn,yn1,yn,yn0)} \]
\[ \text{implicit double precision (a-h,o-z)} \]
\[ yn1 = 2. * \text{yn} - \text{yn0} + h**2. * f(num,xn,yn) \]
\[ \text{return} \]
\[ \text{end} \]

\[ \text{subroutine problem(num,x0,y0,xend,nsteps)} \]
\[ \text{implicit double precision (a-h,o-z)} \]
\[ \text{common/bk1/b,c,q} \]
\[ \text{if(num.eq.1)then} \]
\[ \text{---PROBLEM: 1 \ Y'' + X*Y = 0} \]
\[ \text{---INITIAL CONDITIONS X0=0,Y0=1,Y'=2} \]
\[ \text{---EXACT SOLUTION Y=(1 - X**3/3 + X**6/180 - ...)} \]
\[ \text{---+ 2*(X - X**4/12 + X**7/504 - ...)} \]
\[ \text{---CHOOSE SOLUTION DOMAIN [0,1]} \]
\[ \text{write(*,'PROBLEM: 1 \ Y'' + X*Y = 0')} \]
\[ \text{write(*,'INITIAL CONDITIONS X0=0,Y0=1,Y'=2')} \]
\[ \text{write(*,'EXACT SOLUTION Y=(1 - X**3/3 + X**6/180 - ...')} \]
\[ \text{write(*,'+ 2*(X - X**4/12 + X**7/504 - ...')} \]
\[ \text{write(*,'CHOOSE SOLUTION DOMAIN [0,1]} \]
\[ \text{write(*,'INPUT VALUES OF x0 y0 xend nsteps')} \]
\[ \text{read(*,'x0,y0,xend,nsteps')} \]
\[ \text{return} \]
elseif(num.eq.2) then
  c-----PROBLEM:2 \ Y'' + 2*X**2*Y = 0
  c-----INITIAL CONDITIONS X0=0,Y0=1,Y'=1
  c-----EXACT SOLUTION Y=(1 - X**4/6 + X**8/168 - ...)
  c-----+ (X - X**5/10 + X**9/360 - ...)
  c-----CHOOSE SOLUTION DOMAIN [0,1]
  write(*,*)' PROBLEM:2 \ Y'' + 2*X**2*Y = 0'
  write(*,*)' INITIAL CONDITIONS X0=0,Y0=1,Y'=1'
  write(*,*)' EXACT SOLUTION Y=(1 - X**4/6 + X**8/168 - ...)' +
  write(*,*)' + (X - X**5/10 + X**9/360 - ...)'
  write(*,*)' CHOOSE SOLUTION DOMAIN [0,1]'
  read*,x0,y0,xend,nsteps
  return
elseif(num.eq.3) then
  c-----PROBLEM:3 \ Y'' + X**2*Y = 1 + X + X**2
  c-----INITIAL CONDITIONS X0=0,Y0=2,Y'=2
  c-----EXACT SOLUTION Y=2*(1 - X**4/12 + X**8/672 - ...)
  c-----+ 2*(X - X**5/20 + X**9/1440 - ...)
  c-----CHOOSE SOLUTION DOMAIN [0,1]
  write(*,*)' PROBLEM:3 \ Y'' + X**2*Y = 1 + X + X**2'
  write(*,*)' INITIAL CONDITIONS X0=0,Y0=2,Y'=2'
  write(*,*)' EXACT SOLUTION Y=2*(1 - X**4/12 + X**8/672 - ...)' +
  write(*,*)' + 2*(X - X**5/20 + X**9/1440 - ...)'
  write(*,*)' CHOOSE SOLUTION DOMAIN [0,1]'
  write(*,*)' INPUT VALUES OF x0 y0 xend nsteps'
  read*,x0,y0,xend,nsteps
  return
elseif(num.eq.4) then
  c-----PROBLEM:4 \ Y'' - Y = 0
  c-----INITIAL CONDITIONS X0=0,Y0=1,Y'0=-1
  c-----EXACT SOLUTION Y=exp(-X)
  c-----CHOOSE SOLUTION DOMAIN [0,1]
  write(*,*)' PROBLEM:4 \ Y'' - Y = 0'
  write(*,*)' INITIAL CONDITIONS X0=0,Y0=1,Y'0=-1'
  write(*,*)' EXACT SOLUTION Y=exp(-X)'
  write(*,*)' CHOOSE SOLUTION DOMAIN [0,1]'
  write(*,*)' INPUT VALUES OF x0 xend nsteps'
  read*,x0,y0,xend,nsteps
  return
c
elseif(num.eq.5)then

c-----PROBLEM:5  Y'' - 220*(2-X)**(-12) = 0

c-----INITIAL CONDITIONS X0=1,Y0=-2,Y'0=-1

c-----EXACT SOLUTION Y=2*(2-X)**(-10)-X-1

c-----CHOOSE SOLUTION DOMAIN [0,1]

write(*,*)' PROBLEM:5  Y'' - 220*(2-X)**(-12) = 0'
write(*,*)' INITIAL CONDITIONS X0=1,Y0=0,Y'0=19'
write(*,*)' EXACT SOLUTION Y=2*(2-X)**(-10)-X-1'
write(*,*)' CHOOSE SOLUTION DOMAIN [0,1]'
write(*,*)' INPUT VALUES OF x0 xend nsteps'
read*,x0,y0,xend,nsteps
return

c
elseif(num.eq.6)then

c-----PROBLEM:6  Y'' - Y*((Q + B*X)/X)**2 - Q/(X**2) = 0

c-----INITIAL CONDITIONS X0=1,Y0=10e,Y'0=10e*(Q + B)

c-----EXACT SOLUTION Y=C*X**Q*EXP(B*X)

c-----USE B=1,C=10,Q=3/2

write(*,*)' PROBLEM:6  Y'' - Y*((Q + B*X)/X)**2 - Q/(X**2) = 0'
write(*,*)' INITIAL CONDITIONS X0=1,Y0=10e,Y'0=10e*(Q + B)'
write(*,*)' EXACT SOLUTION Y=C*X**Q*EXP(B*X)'
write(*,*)' CHOOSE SOLUTION DOMAIN [1,2]'
write(*,*)' USE B=1,C=10,Q=3/2'
write(*,*)' INPUT VALUES OF b c q x0 xend nsteps'
read*,b,c,q,x0,xend,nsteps
y0=exact(num,x0)
return
else
print*, 'YOU HAVE NO SUCH PROBLEM NUMBER'
stop
endif
end
c

---
defining the function :

function f(num,x,y)
  implicit double precision(a-h,o-z)
  if(num.eq.1) then
    f=-x*y
    return
  elseif(num.eq.2) then
    f=-2.d0*(x**2)*y
    return
  elseif(num.eq.3) then
    f=-x**2*y + 1. + x + x**2
    return
  elseif(num.eq.4) then
    f=y
    return
  elseif(num.eq.5) then
    f=220.d0*(2.d0-x)**(-12)
    return
  elseif(num.eq.6) then
    f=y*(((q+b*x)/x)**2 - q/x**2)
    return
  endif
end

---
defining the solution :

function exact(num,x)
  implicit double precision(a-h,o-z)
  if(num.eq.1) then
    exact = (1.-x**3/3.+x**6/180.) + 2*(x-x**4/12.+x**7/504)
    return
  elseif(num.eq.2) then
    exact = (1.-x**4/6.+x**8/168.) + (x-x**5/10.+x**9/360.)
    return
  elseif(num.eq.3) then
    exact = 2*(1.-x**4/12.+x**8/672.) + 2*(x-x**5/20.+x**9/1440.)
    return
  elseif(num.eq.4) then
    exact = exp(-x)
    return
  elseif(num.eq.5) then
    exact = 2.d0*(2.d0-x)**(-10)-x-1.d0
    return
  elseif(num.eq.6) then
    exact = c*x**q*exp(b*x)
    return
  endif
end
APPENDIX 4

PROGRAM STA2STEP ( Using Mathematica )

\[ gm := \text{Abs}[((z+\sqrt{z^2+4})/2)^2] \]
\[ /z->(x+I y)) \]

\[ gm1 := \text{Plot3D}[gm,\{x,-4,4\},\{y,-4,4\}, \]
\[ \text{AxesLabel ->"Re}(z),"Im}(z),"Height"], \]
\[ \text{ViewPoint ->\{-2,-2,0.5\},} \]
\[ \text{PlotRange ->\{0,1.2\}} \]

\[ gm2 := \text{ContourPlot}[gm,\{x,-2.5,3\},\{y,-3,3\},\text{ContourShading}->\text{False}, \]
\[ \text{Contours->\{1,1\}} \]

\[ co := \text{Abs}\left[(\frac{\sqrt{2}}{l-z}\right) \]
\[ /z->(x+I y)) \]

\[ col := \text{Plot3D}[co,\{x,-4,4\},\{y,-4,4\}, \]
\[ \text{AxesLabel ->"Re}(z),"Im}(z),"Height"], \]
\[ \text{ViewPoint ->\{-2,-2,0.5\},} \]
\[ \text{PlotRange ->\{0,1.2\}} \]

\[ co2 := \text{ContourPlot}[co,\{x,-2.5,3\},\{y,-3,3\},\text{ContourShading}->\text{False}, \]
\[ \text{Contours->\{1,1\}} \]

\[ ce := \text{Abs}\left[(1+(3(3-2 z))+((2 \sqrt{3})) (Sqrt(3-z^2))/(-3+2 z))/2 \]
\[ /z->(x+I y)) \]

\[ ce1 := \text{Plot3D}[ce,\{x,-4,4\},\{y,-4,4\}, \]
\[ \text{AxesLabel ->"Re}(z),"Im}(z),"Height"], \]
\[ \text{ViewPoint ->\{-2,-2,0.5\},} \]
\[ \text{PlotRange ->\{0,1.2\}} \]

\[ ce2 := \text{ContourPlot}[ce,\{x,-5.5,4\},\{y,-4,4\},\text{ContourShading}->\text{False}, \]
\[ \text{Contours->\{1,1\}} \]

\[ ha := \text{Abs}\left[(-1+(3(3-2 z))+((2 \sqrt{3})) (Sqrt(3-z^2))/(-3+2 z))/2 \]
\[ /z->(x+I y)) \]

\[ ha1 := \text{Plot3D}[ha,\{x,-4,4\},\{y,-4,4\}, \]
\[ \text{AxesLabel ->"Re}(z),"Im}(z),"Height"], \]
\[ \text{ViewPoint ->\{-2,-2,0.5\},} \]
\[ \text{PlotRange ->\{0,1.2\}} \]

\[ ha2 := \text{ContourPlot}[ha,\{x,-2.5,3\},\{y,-3,3\},\text{ContourShading}->\text{False}, \]
\[ \text{Contours->\{1,1\}} \]

\[ f := \text{Abs}\left[((1+z^2)/(1/2) z^2+(1/6) z^3+(1/24) z^4) \]
\[ /z->(x+I y)) \]

\[ r1 := \text{Plot3D}[f,\{x,-4,4\},\{y,-4,4\}, \]
\[ \text{AxesLabel ->"Re}(z),"Im}(z),"Height"], \]
\[ \text{ViewPoint ->\{-2,-2,0.5\},} \]
\[ \text{PlotRange ->\{0,1.2\}} \]

\[ r2 := \text{ContourPlot}[f,\{x,-4,4\},\{y,-4,4\},\text{ContourShading}->\text{False}, \]
\[ \text{Contours->\{1,1\}} \]

\[ s1 := \text{Show}[\text{GraphicsArray}\left[\left\{\{gm1,co1,ce1,ha1\}, \{gm2,co2,ce2,ha2\}\right\}\right]] \]
\[ s2 := \text{Show}[\%\ , \text{Frame->True}, \text{FrameTicks->None}] \]
APPENDIX 5

Program Weighted AM Formula
* c05nbf example program text
* mark 14 revised. nag copyright 1989
* ..parameter..
  implicit double precision (a-h,o-z)
  integer n,lwa
  parameter (n=7,lwa=(n*(3*n+13))/2)
  integer nout
  parameter (nout=7)
* ..local scalars..
  c real fnorm,tol
  integer i, ifail, j
* ..local arrays..
  dimension fvec(n), wa(lwa), x(n)
* ..external functions..
  c real F06EJF, X02AJF
  external f06ejf, x02ajf
* ..external subroutines..
  external c05nbf, fcn
* ..intrinsic functions..
  intrinsic sqrt
* ..executable statements..
  write (nout,*)'c05nbf AM-RK4 program results'
  write (nout,*)
  the following starting values provide a rough solution.
  do 20 j = 1, n
    x(j) = 0.50e0
  continue
  tol = sqrt(x02ajf())
  ifail = 1
  call c05nbf(fcn,n,x,fvec,tol,wa,lwa,ifail)
  if (ifail.eq.0) then
    fnorm = f06ejf(n,fvec,1)
    write (nout,99999)'final 2-norm of the residuals =',fnorm
    write (nout,*)
    write (nout,*) 'final approximate solution'
    write (nout,*)
    write (nout,99998) (x(j),j=1,n)
    write (nout,*)
    write (nout,99998) (0.5-x(5), 1-x(6)-x(7))
  else
    write (nout,99997)'fail =', ifail
    if (ifail.gt.1) then
      write (nout,*)
      write (nout,*) 'approximate solution'
      write (nout,*)
      write (nout,99998) (x(i),i=1,n)
    end if
  end if
99999 format (a17,e13.6)
99998 format (a5,e13.6)
372
end if
end if
stop

99999 format (1x,a,e12.4) 99998 format (1x,3f18.10) 99997 format (1x,a,i2)
end

subroutine fcn(n,x,fvec,iflag)
implicit double precision (a-h,o-z)
.. parameters ..
.. scalar arguments ..
integer iflag, n
.. array arguments ..
dimension fvec(n), x(n)
.. local scalars ..
integer k
.. executable statements ..
do 20 k = 1, n
fvec(1) = 1. - x(1) - x(2) - x(3)
fvec(2) = 2. - 2. * x(4) * x(1) - x(2) - 2. * x(4) * x(2) - 3. * x(3)
fvec(3) = 2. - 3. * x(4) * x(2) + 6. * x(4) * x(5) * x(2) - 3. * x(3)
- 3. * x(4) * x(3) + 6. * x(4) * x(5) * x(3) + 3. * x(6) * x(3)
+ 3. * x(7) * x(3) - 6. * x(4) * x(7) * x(3)
fvec(4) = 8. - 12. * x(4) * x(4) * x(1) - 3. * x(2) - 12. * x(4) * x(4) * x(2)
- 15. * x(3)
fvec(5) = 1. - 6. * x(4) * x(3) + 12. * x(4) * x(5) * x(3) -
12. * x(4) * x(5) * x(6) * x(3) + 6. * x(4) * x(7) * x(3) - 12. * x(4) * x(5) * x(7)
* x(3) + 6. * x(4) * x(6) * x(3)
fvec(6) = 8. - 6. * x(4) * x(2) - 6. * x(4) * x(4) * x(2) +
12. * x(4) * x(5) * x(2) + 12. * x(4) * x(4) * x(5) * x(2) - 15. * x(3) -
6. * x(4) * x(3) - 6. * x(4) * x(4) * x(3) + 12. * x(4) * x(5) * x(3) +
12. * x(4) * x(4) * x(5) * x(3) + 15. * x(6) * x(3) + 15. * x(7) * x(3) -
24. * x(4) * x(7) * x(3) - 12. * x(4) * x(4) * x(7) * x(3)
fvec(7) = 4. - 8. * x(4) * x(4) * x(4) * x(1) - x(2) - 9. * x(3) -
8. * x(4) * x(4) * x(4) * x(2)
20 continue
return
end

**********************************************************
c05nbf AM-RK4 program results
final 2-norm of the residuals =  .3313E-10
final approximate solution
.3333333333 .3333333333 .3333333333
.5000000000 .0000000000 .0000000000
.0000000000 .5000000000 .0000000000

************************************************************
APPENDIX 6

Program System For Second Order ODEs

\texttt{SINCLUDE probl.f}

\begin{verbatim}
implicit double precision (a-h,o-z)
common/blk3/x0,xend,y0,v0,npb
write(*,*)'Enter the value of npb and h'
read(*,*)npb,h
call problem
write(*,*)'Enter the value of x0,y(X0),V(X0) and xend'
read(*,*)x0,y0,v0,xend

xn = x0
yn = y0
vn = v0
wl = 0.2615038147351447
w2 = -0.2765809214083533
w3 = 0.5947141647174489
w4 = 0.4203629419557595
w5 = 1.5471214402823019
w6 = 0.1756458393315915
w7 = 0.124305900880404
w8 = 0.100931693726120
w9 = 0.1100539629764319
w10 = 0.9997431862131075
w11 = -0.0928890403263464
w12 = -0.6201812828225984
w13 = 0.5-a4-a5
w14 = 1-a6-a7-a8

print*,a11,a22,a4+a5+a11,a6+a7+a8+a22
write(*,7)

7 format(5x, 'xn ',llx, 'computed',12x, 'exact',13x, 'absolute error')

200 aM1 = f(npb,xn,yn,vn)
aM2 = f(npb,xn+a1*h,yn+h*a1*vn,vn+a1*h*aM1)
aM3 = f(npb,xn+(a2+a3)*h,yn+(a2+a3)*h*vn+a1*a3*h**2*aM1,
 vn+a2*h*aM1+a3*h*aM2)
aM4 = f(npb,xn+(a4+a5+a11)*h,yn+(a4+a5+a11)*h*vn+
   (a1*a5+a2*a11)*h**2*aM1+
   a3*a11*h**2*aM2, vn+a4*h*aM1+a5*h*aM2+a11*h*aM3)
aM5 = f(npb,xn+(a6+a7+a8+a22)*h,yn+(a6+a7+a8+a22)*h*vn+
   (a1*a7+a2*a8+a4*a22)*h**2*aM1+(a3*a8+a5*a22)*h**2*aM2+
   a11*a22*h**2*aM3,
   vn+a6*h*aM1+a7*h*aM2+a8*h*aM3+a22*h*aM4)
yn1 = yn+(h/2.)*(w1*(2*vn+a1*h*aM1)+w2*(2*vn+(a1+a2)*h*aM1+
   a3*h*aM2)+
   w3*(2*vn+(a2+a4)*h*aM1+(a3+a5)*h*aM2+a11*h*aM3)+
   w4*(2*vn+(a4+a6)*h*aM1+(a5+a7)*h*aM2+(a8+a11)*h*aM3+
   a22*h*aM4))
vn1 = vn+(h/2.)*(w1*(aM1+aM2)+w2*(aM2+aM3)+w3*(aM3+aM4)+
   w4*(aM4+aM5))
\end{verbatim}
\( x_{n+1} = x_n + h \\
\text{xn} = x_{n+1} \\
\text{yn} = y_{n+1} \\
\text{vn} = v_{n+1} \\
\text{exact} = \text{exact}(\text{nbp}, \text{xn}) \\
\text{err} = \text{dabs}(\text{exact} - \text{yn}) \\
\text{write}(*,100)\text{xn, yn, exact, err} \\
100 \text{ format(f7.4,3e23.10)} \\
\text{write and format for problem no. 1 where } x_0=0, y_0=1, v_0=0 \\
\text{write(*,100)xn,yn,vn} \\
c100 \text{ format(f7.4,3e23.10)} \\
\text{if}(\text{xn}.le.\text{xend})\text{goto 200} \\
\text{stop} \\
\text{end} \\

\text{SUBROUTINE PROBLEM1}\n
\text{subroutine problem1} \\
\text{implicit double precision (a-h,o-z)} \\
\text{common/blk1/b,c,q} \\
\text{common/blk3/x0,xend,y0,v0,npb} \\
\pi=22.d0!1. \\
\text{if}(\text{nbp}.eq.1)\text{then} \\
\text{For npb=1, see Scraton,R.E [1986], pp 77} \\
\text{PROBLEM: 1 } Y'' + 3*Y' + 2*Y = 2*\text{EXP}(-3*X) \\
\text{INITIAL CONDITIONS } x_0=0, y(x_0)=1, y'(x_0)=v(x_0)=-2 \\
\text{EXACT SOLUTION } y=\text{EXP}(-X)-\text{EXP}(-2*X)+\text{EXP}(-3*X) \\
\text{CHOOSE SOLUTION DOMAIN [0,1]} \\
\text{write(*,11)} \\
11 \text{ format(7x,'FOR PROBLEM:1, SEE SCRATON,R.E [1986], PP 77'} \\
1/10x,' PROBLEM: 1 } Y'' + 3*Y' + 2*Y = 2*\text{EXP}(-3*X)/10x, \\
2' \text{ INITIAL CONDITIONS } x_0=0, y(x_0)=1, y'(x_0)=v(x_0)=-2' \\
3/12x, ' EXACT SOLUTION } y=\text{EXP}(-X)-\text{EXP}(-2*X)+\text{EXP}(-3*X)' \\
4/10x,' \text{ CHOOSE SOLUTION DOMAIN [0,1]}' \\
\text{return} \\
\text{elseif(nbp.eq.2)then} \\
\text{-----For npb=2, see Greenspan,D and V.Casulli [1988], pp 139} \\
\text{-----PROBLEM: 2 } Y'' + (Y')^**3 - 8*X*Y = 0 \\
\text{-----INITIAL CONDITIONS } x_0=0, y(x_0)=0, y'(x_0)=0 \\
\text{-----EXACT SOLUTION } y=X^**2 \\
\text{-----CHOOSE SOLUTION DOMAIN [0,1]}
write(*,12)
12 format(7x,'FOR PROBLEM:2,SEE GREENSPAN,D AND V.CASULLI [1988],PP 139'
1/10x,'PROBLEM:2  Y'''' + (Y'')**3 - 8*x*y = 0')/10x,
2' INITIAL CONDITIONS X0=0,Y(X0)=0,Y''(X0)=0
3/12x,' EXACT SOLUTION Y=X**2'
4/10x,' CHOOSE SOLUTION DOMAIN [0,1]')
return
else
stop
endif
end

c-----SUBROUTINE FOR THE RIGHT HAND SIDE OF EQUATION
c
double precision function f(npb,x,y,v)
implicit double precision(a-h,o-z)
common/blk1/b,c,q
c-----For npb=1 , see Scraton,R.E [1986], pp 77
if(npb.eq.1)then
  f=-2.d0*y-3.d0*v+2.d0*exp(-3*x)
return
c-----For npb=2, see Greenspan,D and V.Casulli [1988], pp 139
elseif(npb.eq.2)then
  f=2.d0+8.d0*x*y-v**3
return
•endif
end

376
APPENDIX 7

c----program fifth order Adam multistep method using

c----a fifth order arithmetic mean as a starter

implicit double precision (a-h,o-z)
dimension x(0:100),y(0:100),exact(O:100)
integer i,k

external f, exact

w1=0.2615038147351447
w2=-0.2765809214083533
w3=0.5947141647174489
w4=0.4203629419557595
a1=1.5471214402823019
a2=0.1756459393315915
a3=0.1243059000880404
a4=0.1009316693726120
a5=0.1100539629764319
a6=0.9997431862131075
a7=-0.0928890403263464
a8=0.6201812828225984
a11=0.5-a4-a5
a22=1-a6-a7-a8

print*,a11 ,a22,a4+a5+a11,a6+a7+a8+a22

write(*,7)

7 format(/,10x,'xn ',12x,' computed',7x,' exact',8x,'absolute error',/)

x(O)= 0
y(O)= 1.do
xO = 0
xend = 1
n= 100
h= (xend - xO)/n
jj = 10

do 20 i=0,4
  xn = x(i)
  yn = y(i)
  ak1 = f(xn,yn)
  ak2 = f(xn+a1*h,yn+a1*h*ak1)
  ak3 = f(xn+(a2+a3)*h,yn+a2*h*ak1+a3*h*ak2)
  ak4 = f(xn+(a4+a5+a11)*h,yn+a4*h*ak1+a5*h*ak2+a11*h*ak3)
  ak5 = f(xn+(a6+a7+a8+a22)*h,yn+a6*h*ak11+a7*h*ak2+a8*h*ak3+a22*h*ak4)

  y(i+1)=yn+(h/2)*(w1 *( ak1 +ak2)+w2*( ak2+ak3)+w3*( ak3+ak4)+w4*( ak4+ak5))
  x(i+1) = xO + h*(i+1)

20 continue

f0 = f(x(0),y(0))
f1 = f(x(1),y(1))
f2 = f(x(2),y(2))
f3 = f(x(3),y(3))
f4 = f(x(4),y(4))
do 30 k = 4, n-1
y(k+1) = y(k) + (h/720.)*(1901*f4-2774*f3+2616*f2-1274*f1+251*f0)
x(k+1) = x0 + h*(k+1)
f0 = f1
f1 = f2
f2 = f3
f3 = f4
f4 = f(x(k+1),y(k+1))

30 continue
do 40 i = 0, n, jj
PRINT 10,i,x(i),y(i),exact(x(i)),abs(y(i)-exact(x(i)))

40 continue
10 format(1x,i3,2x,el0.5,1x,3e18.7)
print 12,jj,h
12 format(/,lx, 'RESULTS ARE PRINTED FOR EVERY',/4,lx,'STEPS:h=',f6.4)
stop
end

c-------------------end of main program-------------------------------------------------------
c
function f(x,y)
implicit double precision (a-h,o-z)
f=1.d0+x-y
return
end

c
function exact(x)
implicit double precision (a-h,o-z)
extact = x+depp(-x)
return
end
APPENDIX 8

TRIGONOMETRIC PROGRAM

This MATHEMATICA program is used for solving the system of equations using trigonometric polynomial interpolation to obtain the third, fourth and fifth order methods:

\[
\sin[v_] := v - \frac{v^3}{6} + \frac{(v^5/120)-(v^7/5040)+(v^9/362880)}{1} \\
\cos[v_] := 1 - \frac{v^2}{2} + \frac{(v^4/24)-(v^6/720)+(v^8/40320)}{1} \\
k1 := (Solve[{v \sin[v] b1 + \cos[v] - 1 == 0,} \\
{-v b0 - v \cos[v] b1 + \sin[v] == 0},} \\
{b0, b1}]) \\
k3 := (Solve[{-v \sin[v] b1 - v \sin[2v] b2 - v \sin[3v] b3 + 1 - \cos[3v] == 0,} \\
{v b0 + v \cos[v] b1 + v \cos[2v] b2 + v \cos[3v] b3 - \sin[3v] == 0,} \\
{-2 v \sin[2v] b1 - 2 v \sin[4v] b2 - 2 v \sin[6v] b3 - \cos[6v] + 1 == 0,} \\
{2 v b0 + 2 v \cos[2v] b1 + 2 v \cos[4v] b2 + 2 v \cos[6v] b3 - \sin[6v] == 0},} \\
{b0, b1, b2, b3}]) \\
k5 := (Solve[{-v \sin[v] b1 - v \sin[2v] b2 - v \sin[3v] b3 - v \sin[4v] b4 - v \sin[5v] b5 + 1 - \cos[5v] == 0,} \\
v b0 + v \cos[v] b1 + v \cos[2v] b2 + v \cos[3v] b3 + v \cos[4v] b4 + v \cos[5v] b5 - \sin[5v] == 0,} \\
{-2 v \sin[2v] b1 - 2 v \sin[4v] b2 - 2 v \sin[6v] b3 - 2 v \sin[8v] b4 = 0,} \\
{-2 v \sin[10v] b5 - \cos[10v] + 1 == 0,} \\
{2 v b0 + 2 v \cos[2v] b1 + 2 v \cos[4v] b2 + 2 v \cos[6v] b3 + 2 v \cos[8v] b4 + 2 v \cos[10v] b5 - \sin[10v] == 0,} \\
{-3 v \sin[3v] b1 - 3 v \sin[6v] b2 - 3 v \sin[9v] b3 - 3 v \sin[12v] b4 = 0,} \\
{-3 v \sin[15v] b5 - \cos[15v] + 1 == 0,} \\
{3 v b0 + 3 v \cos[3v] b1 + 3 v \cos[6v] b2 + 3 v \cos[9v] b3 + 3 v \cos[12v] b4 + 3 v \cos[15v] b5 - \sin[15v] == 0},} \\
{b0, b1, b2, b3, b4, b5}])
PROGRAM PERIODIC_CASES
program periodic_cases
implicit double precision (a-h,o-z)
common/blk3/x0,xend,y0,v0,npb
write(*,*)'Enter the value of h'
read(*,*)h
1111 write(*,*)'Enter the value of w,x0 and xend'
1 read*,w,x0,xend
xn = x0
yn = dcos(xn)
vn = dsin(xn)
v = w*h
a1 = 0.2000000000
a2 = -0.2027706499
a3 = 0.5320433054
a4 = -0.8706804955
a5 = -0.4215000613
a6 = 1.3566987431
a7 = -0.4405835645
a8 = 1.0285379694
a9 = -1.7032738709
a10 = 1.1167966402
a11 = -0.1408342610
a31 = 0.4-a2
a41 = 0.6-a3-a4
a51 = 1-a8-a9-a10-a11
print*,a2+a31,a3+a4+a41,a5+a6+a7+a51,a8+a9+a10+a11+a61
w1 = (19./288.)*((1+(55.*v**2/114.)+(3282265.*v**4/2736.)·
(551328935.*v**6/114912.)+(258450119345.*v**8/1378944.)+
(68594391998759.*v**10/11819520.)·
(727108666308742877.*v**12/694987760.)·
(91022051463182217.*v**14/4389396480.))
w2 = (75./288.)*(1-(11.*v**2/144.)·
(218729.*v**4/144.)+(4969804253.*v**6/90720.)·
(67362326019.*v**8/2721600.)·
(304273318350997.*v**10/46656000.)·
(3377759197565612413.*v**12/27433728000.)·
(4023110255693433787.*v**14/13168189440.)·
w3 = (50./288.)*((1+(11.*v**2/30.)+(3280991.*v**4/720.)·
(4365325187.*v**6/30240.)+(115173951529.*v**8/1814400.)·
(276014322723379.*v**10/15552000.)·
(2985268375870083433.*v**12/29144576000.)·
(81154909203032843779.*v**14/109734912000.)·
w4 = (50./288.)*((1+(11.*v**2/30.)·
(3281509.*v**4/720.)+(3699937313.*v**6/30240.)·
(198757705399.*v**8/453600.)·(255272260074121.*v**10/15552000.)·+
(2503319588144210317.*v**12/9144576000.-
(39187210645153010971.*v**14/109734912000.))

\[ w_5 = (75./288.)*(1-(11.*v**2/30.)+(218771.*v**4/144.-(
(297353247.*v**6/690720.)+(37209863827.*v**8/5443200.)+
(234401042464003.*v**10/46656000.))-
(196066728140426337.*v**12/27433728000.-(
(446663546619991577.*v**14/13168189440.))
\]

w6=(19./288.)*(1+(55.*v**2/114.)-
(3280235.*v**4/2736.+(2186348165.*v**6/114912.)+
(1641506735.*v**8/689472.-(41492416771241.*v**10/11819520.)+
(279664542028634873.*v**12/6949877760.)+
(11205900105824818873.*v**14/83398533120.))

Write(*,7)

7  format(5x,'x',7x,' computed',12x, exact,13x,' error')

200 ak1=f(xn,yn,vn)

\[ aM1 = g(xn,yn,vn) \]

ak2=f(xn+a1*h,yn+a1*h*ak1,yn+a1*h*aM1)

\[ aM2 = g(xn+a1*h,yn+a1*h*ak1,yn+a1*h*aM1) \]

ak3=f(xn+(a2+a3)*h,yn+a2*h*ak1+a3*h*ak2,yn+a2*h*aM1+a3*h*aM2)

\[ aM3 = g(xn+(a2+a3)*h,yn+a2*h*ak1+a3*h*ak2,\
   vn+a2*h*aM1+a3*h*aM2) \]

ak4=f(xn+(a3+a4+a4)*h,yn+a3*h*ak1+a4*h*ak2+a4*h*ak3,
   vn+a3*h*aM1+a4*h*aM2+a4*h*aM3)

\[ aM4 = g(xn+(a3+a4+a4)*h,yn+a3*h*ak1+a4*h*ak2+a4*h*ak3,\
   vn+a3*h*aM1+a4*h*aM2+a4*h*aM3) \]

ak5=f(xn+(a5+a6+a7+a5)*h,yn+a5*h*ak1+a6*h*ak2+
   a7*h*ak3+a5*h*ak4,yn+a5*h*aM1+a6*h*aM2+
   a7*h*aM3+a5*h*aM4)

\[ aM5 = g(xn+(a5+a6+a7+a5)*h,yn+a5*h*ak1+a6*h*ak2+\
   a7*h*ak3+a5*h*ak4,yn+a5*h*aM1+a6*h*aM2+\
   a7*h*aM3+a5*h*aM4) \]

ak6=f(xn+(a8+a9+a10+a11+a6)*h,yn+a8*h*ak1+
   a9*h*ak2+a10*h*ak3+a11*h*ak4+
   +a61*h*ak5,yn+a8*h*aM1+a9*h*aM2+a10*h*aM3+
   +a11*h*aM4+a61*h*aM5)

\[ aM6 = g(xn+(a8+a9+a10+a11+a6)*h,yn+a8*h*ak1+a9*h*ak2+\
   +a10*h*ak3+a11*h*ak4+
   +a61*h*ak5,yn+a8*h*aM1+a9*h*aM2+\
   +a10*h*aM3+a11*h*aM4+a61*h*aM5) \]

yn1=yn+h*(w1*ak1+w2*ak2+w3*ak3+w4*ak4+w5*ak5+w6*ak6)

vn1=vn+h*(w1*aM1+w2*aM2+w3*aM3+w4*aM4+w5*aM5+w6*aM6)

c

\[ xn1=xn+h \]

\[ xn=xn1 \]

\[ yn = yn1 \]

\[ vn = vn1 \]

\[ exact = exact(xn) \]

\[ err = (exact - yn) \]

\[ exacta = exacta(xn) \]

381
\[ \text{erra} = (\text{exact} - \text{vn}) \]
\[
\text{write}(*,100)\text{xn,yn,exact,erra}
\]
\[
\text{write}(*,100)\text{xn,exacta,err,erra}
\]
\[
100 \quad \text{format(f7.4,3e23.5)}
\]
\[
\text{if(xn.le.xend)goto 200}
\]
\[
\text{stop}
\]
\[
\text{end}
\]

c

double precision function \( f(x,y,v) \)
implicit double precision (a-h,o-z)
   \( f = -v \)
return
end
c
double precision function \( g(x,y,v) \)
implicit double precision (a-h,o-z)
   \( g = y \)
return
end
c
double precision function exact(x,y)
implicit double precision (a-h,o-z)
   \( \text{exact} = \text{dcos}(x) \)
return
end
c
double precision function exacta(x,y)
implicit double precision (a-h,o-z)
   \( \text{exacta} = \text{dsin}(x) \)
return
end