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

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_oM) (Evans and Yaakub [1995]), the Centroidal mean (C_eM) (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_oM 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_oM), the Centroidal mean (C_eM) 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 $(C_o M)$ 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 4^{th} and 5^{th} order.

TABLE OF CONTENTS

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 busines, 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

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

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

$$y' = 1 + x - y \tag{1.1.2}$$

$$y'' + 101(y')^3 + 100y = 0 \tag{1.1.3}$$

$$(y''')^2 + xy' + y = \sin(x) \tag{1.1.4}$$

$$\frac{\partial^3 u}{\partial x^3} + \frac{\partial^3 u}{\partial t^3} + \frac{\partial^3 u}{\partial y^3} = 4u \tag{1.1.5}$$

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 $y^{(m)} = \dfrac{d^m y}{dx^m}$, 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

$u = u(x,t,y)$. 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

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

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 $h = \dfrac{(b-a)}{N}$ for some integer N and the interval [a,b] is divided into N subinterval as shown in Figure 1.1.

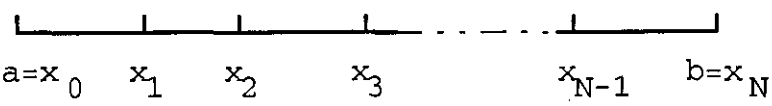


Figure 1.1: The interval divided into N subinterval

Given the initial value $y_0 = y(x_0) = \alpha$ at the point $x = x_0 = a$. Now we want to compute y_1 which approximates $y(x_1)$ by discretizing the differential equation. In general, we assume that we know y_n or $y(x_n)$ and we want to compute y_{n+1}. The differential equation at point $x = x_n$ is

$$y'(x_n) = f(x_n, y(x_n)) \tag{1.2.1}$$

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

$$y'(x_n) = \frac{y(x_{n+1}) - y(x_n)}{h} \tag{1.2.2}$$

and the differential equation then becomes

$$\frac{y(x_{n+1}) - y(x_n)}{h} = f(x_n, y(x_n)) \tag{1.2.3}$$

By replacing $y(x_n)$ and $y(x_{n+1})$ with y_n and y_{n+1} in the difference quotient, we obtain

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{1.2.4}$$

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+k} is calculated using values of y computed at $x_{n+j} = x_n + jh, \quad j = 0,1,2,\ldots,k-1$. 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

$$y^{(m)} = f\left(x, y, y', \ldots, y^{(m-1)}\right) \tag{1.2.5}$$

and the initial value can be written as

$$y^{(i-1)}(\alpha) = \beta_i \quad , \quad i = 1,2,\ldots,m \ . \tag{1.2.6}$$

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

$$\begin{aligned} y_1 &= y, \\ y_2 &= y', \\ y_3 &= y'', \\ \vdots &\quad \vdots \\ y_m &= y^{(m-1)}. \end{aligned} \tag{1.2.7}$$

The equation (1.2.7) now can be written as the first order system of the initial value problem in the form

$$\begin{aligned} y_1' &= y_2 , & y_1(\alpha) &= \beta_1 , \\ y_2' &= y_3 , & y_2(\alpha) &= \beta_2 , \\ \vdots \quad & \quad \vdots & \vdots \quad & \quad \vdots \\ y_m' &= f(x, y_1, y_2, \ldots, y_m) & y_m(\alpha) &= \beta_m \end{aligned} \qquad (1.2.8)$$

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

$$\underline{y}' = \underline{f}(x, \underline{y}) \quad , \quad \underline{y}(\alpha) = \underline{\beta} \qquad (1.2.9)$$

which defines a system of m first order equations or m^{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 (**DY**namic u**NIX**) 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 $i = 1,2,\ldots,n$ and $X_i \geq 0$, the generalised mean is defined by

$$Mean = \left[\frac{1}{n}\sum_{i=1}^{n} X_i^m\right]^{\frac{1}{m}} \tag{2.1.1}$$

a) Arithmetic Mean

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

$$AM = \frac{1}{n}\sum_{i=1}^{n} X_i \tag{2.1.3}$$

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

$$AM = \frac{X_1 + X_2}{2} \tag{2.1.4}$$

b) Harmonic Mean

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

$$H_aM = \left[\frac{1}{n}\sum_{i=1}^{n} X_i^{-1}\right]^{-1} \tag{2.1.5}$$

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

$$H_aM = \frac{2X_1X_2}{X_1 + X_2} \tag{2.1.6}$$

c) Geometric Mean

The geometric mean is defined by

$$GM = \left[\prod_{i=1}^{n} X_i\right]^{\frac{1}{n}} \quad (2.1.7)$$

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

$$GM = \sqrt{X_1 X_2} \quad (2.1.8)$$

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

$$AM \geq GM \geq H_a M \quad .$$

d) Contraharmonic Mean

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

$$C_o M = \frac{\sum_{i=1}^{n} x_i^2}{\sum_{i=1}^{n} x_i}$$

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

$$C_o M = \frac{X_1^2 + X_2^2}{X_1 + X_2} \quad (2.1.9)$$

e) Centroidal Mean

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

$$C_e M = \frac{2\left(X_i^2 + X_i X_{i+1} + X_{i+1}^2\right)}{3\left(X_i + X_{i+1}\right)} \quad (2.1.10)$$

f) Root Mean Square

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

$$RMS = \sqrt{\frac{X_i^2 + X_{i+1}^2}{2}} \quad (2.1.11)$$

g) Heronian Mean

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

$$H_eM = \frac{X_i + X_{i+1} + \sqrt{X_i X_{i+1}}}{3} \qquad (2.1.12)$$

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

$$C_oM \geq RMS \geq C_eM \geq AM \geq H_eM \geq GM \geq H_aM$$

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

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

i) Weighted Arithmetic Mean

Suppose that $X = [X_1, X_2, \dots, X_n]$ and $W = [W_1, W_2, \dots, W_n]$ 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

$$W_{AM} = \frac{W_1 X_i + W_2 X_{i+1}}{W_1 + W_2} \qquad (2.1.14)$$

j) Weighted Geometric Mean

The weighted geometric mean (W_{GM}) with weight W, and for $X_i \neq X_{i+1}$ is defined as

$$W_{GM} = \left[\prod_{i=1}^{n} X_i^{W_i}\right]^{\frac{1}{\sum W_i}}$$

or

$$W_{GM} = \left[X_i^{W_1} + X_{i+1}^{W_2}\right]^{\frac{1}{w_1+w_2}} \qquad (2.1.15)$$

k) Weighted Harmonic Mean

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

$$W_{H_aM} = \frac{\sum_{i=1}^{n} W_i}{\sum_{i=1}^{n}\left(\frac{W_i}{X_i}\right)}$$

or

$$W_{H_aM} = \frac{W_1 + W_2}{\frac{W_1}{X_i} + \frac{W_{i+1}}{X_{i+1}}} \qquad (2.1.16)$$

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

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

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

$$y_0 = \alpha \quad , \quad y_1 = \beta$$

then the difference equation uniquely defines y_0, y_1, y_2 in order to compute $y_3, y_4, y_5\ldots$ 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

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

then the difference equation has the general solution

$$y_n = c_1 r_1^n + c_2 r_2^n + \ldots + c_k r_k^n \quad ,$$

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

$$y_n = P_{m-1}(n) r_i^n \quad ,$$

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

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

and $\qquad y(a) = y_0$

where $\quad y = [y'(x), y''(x), \ldots, y^s(x)]^T$ 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

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

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 $y(x), y'(x), y''(x), \ldots, y^n(x)$ 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

Name	Language	Machine	Authors
ABACUS			Rump et al.
ACRITH-XCS	Fortran 77	IBM S/370	Kulish et al.
AQUARELS	Fortran 77		Erhel et al.
BIAS	Ada	VAX	Corliss
BIAS	Fortran 77	any F77	Neumaier
BIAS	C++		Knueppel
CLEMMENSON	MS Fortran	PC	Wang and Kennedy
C-XSC	C++	PC	Kulisch et al.
FORTRAN-XSC	Fortran 90	any F90	Walter
HAMMER	Pascal-XSC	any P-XSC	Hammer
INTLIB 77	Fortran 77	any F77	Kearfot et al.
INTLIB 90	Fortran 90	any F90	Kearfot
INTPAK	Maple	any Maple	Connell and Corliss
INTPAK	Mathematica	any Math	Kieper
MATHEMATICA	Mathematica	many	Wolfram , S
MODULA-SC			Ullrich et al.
PASCAL-XSC	Pascal	many	Kulisch et al.
PBASIC	Basic	PC	Aberth
RANGE	C++	Some C++	Aberth
TPX	Pascal	PC	Rump et al.

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 inmensely 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.,

```
In[2]:= %
Out[2]:= (-2 + x)(2 + x).
```

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

```
    In[3]:= Out[1]
    Out[3]:= (-2 + x)(2 + x)
or  In[4]:= %1
    Out[4]:= (-2 + x)(2 + x)
```

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

```
In[5]:= a = Factor[x^2-4];
In[6]:= Expand[a]
Out[6]:= -4 + x^2
```

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 subracting,i.e., 2 + 8 * 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,       (* The solution to *)
         3x+2y-6==0},{x,y}]   (* 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^{30}

and In[10]:= Rationalize[0.125]

Out[10]:= $\frac{1}{8}$.

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

$$1 - w_1 - w_2 - w_3 == 0,$$

$$2 - 2a_1w_1 - w_2 - 2a_1w_2 - 3w_3 == 0,$$

$$2 - 3a_1w_2 + 6a_1a_2w_2 - 3w_3 - 3a_1w_3 + 6a_1a_2w_3 + 3a_3w_3 + 3a_4w_3 - 6a_1a_4w_3 = 0$$

$$8 - 12a_1^2w_1 - 3w_2 - 12a_1^2w_2 - 15w_3 = 0$$

$$1 - 6a_1w_3 + 12a_1a_2w_3 + 6a_1a_3w_3 - 12a_1a_2a_3w_3 + 6a_1a_4w_3 - 12a_1a_2a_4w_3 = 0$$

$$8 - 6a_1w_2 - 6a_1^2w_2 + 12a_1a_2w_2 + 12a_1^2a_2w_2 - 15w_3 - 6a_1w_3$$
$$- 6a_1^2w_3 + 12a_1a_2w_3 + 12a_1^2a_2w_3 + 15a_3w_3 - 24a_1a_4w_3 - 12a_1^2a_4w_3 = 0$$

$$4 - 8a_1^3w_1 - w_2 - 8a_1^3w_2 - 9w_3 = 0$$

The above system of equations can be written as a Mathematica program as follows :

In[11]:= s = $Solve[\{1 - w_1 - w_2 - w_3 == 0,$

$2 - 2a_1w_1 - w_2 - 2a_1w_2 - 3w_3 == 0,$

$2 - 3a_1w_2 + 6a_1a_2w_2 - 3w_3 - 3a_1w_3 + 6a_1a_2w_3 + 3a_3w_3 + 3a_4w_3 - 6a_1a_4w_3 == 0,$

$8 - 12a_1^2w_1 - 3w_2 - 12a_1^2w_2 - 15w_3 == 0,$

$1 - 6a_1w_3 + 12a_1a_2w_3 + 6a_1a_3w_3 - 12a_1a_2a_3w_3 + 6a_1a_4w_3 - 12a_1a_2a_4w_3 == 0,$

$8 - 6a_1w_2 - 6a_1^2w_2 + 12a_1a_2w_2 + 12a_1^2a_2w_2 - 15w_3 - 6a_1w_3$

$- 6a_1^2w_3 + 12a_1a_2w_3 + 12a_1^2a_2w_3 + 15a_3w_3 - 24a_1a_4w_3 - 12a_1^2a_4w_3 == 0,$

$4 - 8a_1^3w_1 - w_2 - 8a_1^3w_2 - 9w_3 == 0\},$

$\{a_1, a_2, a_3, a_4, w_1, w_2, w_3\}]$;

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

```
In[12]:= << sam44
```

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

```
In [13]:= s
```

Out[13}:= $\{\{w_1 -> \frac{1}{3}, w_2 -> \frac{1}{3}, a_4 -> 0, a_2 -> 0, a_3 -> 0, a_1 -> \frac{1}{2}, w_3 -> \frac{1}{3}\}\}$

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

```
In[14]:= s >> result1
```

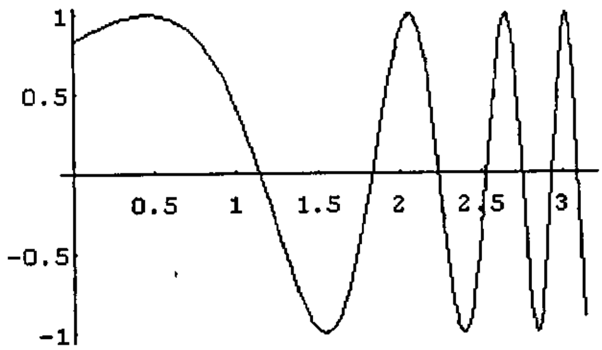
where the result from the above problem was saved in the file named 'result1'. To open the file to read the result from file 'result1' , we use

```
In[15]:= !vi result1 .
```

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 Π is shown as

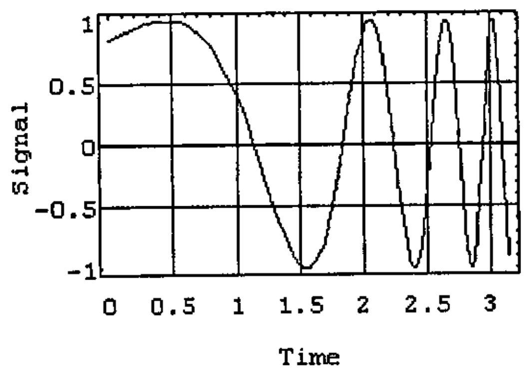
```
In[16]:= Plot[Sin[Exp[x]],{x,0,Pi}]
```




```
Out[16]:= - Graphics -
```

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

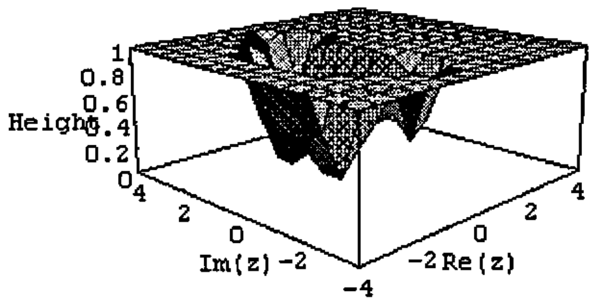
```
In[17]:= Show[%, Frame -> True ,
                FrameLabel -> {"Time","Signal"},
                GridLines -> Automatic]
```




```
Out[17]:= - 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:

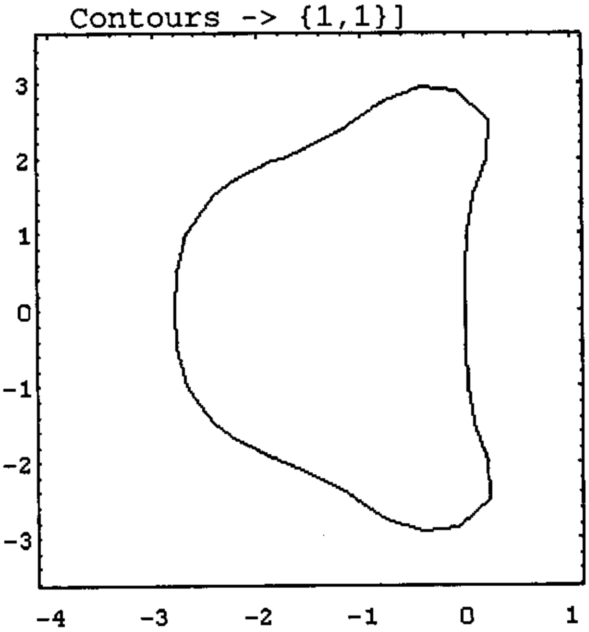
```
In[18]:= f =Abs[((1+z+(z^2/2)+(z^3/6)+(z^4/24))
                    /. z -> (x + I y)] ;
In[19]:= Plot3D[f,{x,-4,4},{y,-4,4},
          AxesLabel -> {"Re(z)","Im(z)","Height"},
          ViewPoint -> {-2,-2,0.8},
          PlotRange -> {0,1}]
```




```
Out[19]:= - SurfaceGraphics -
```

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

```
In[20]:= ContourPlot[f,{x,-4,1},{y,-4,3.5},
         ContourShading -> False,
         Contours -> {1,1}]
```




```
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 multidimentional Taylor series expansions, computed by the ALTRAN system, to analyse the discretization and round-off errors of various numerical methods and also, more inportantly, 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 persecond (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], Hockney and Jesshope [1981], Hwang and Briggs [1984], Khalaf [1990], Freeman and Phillips [1992] and Mohd Saman [1993].

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.

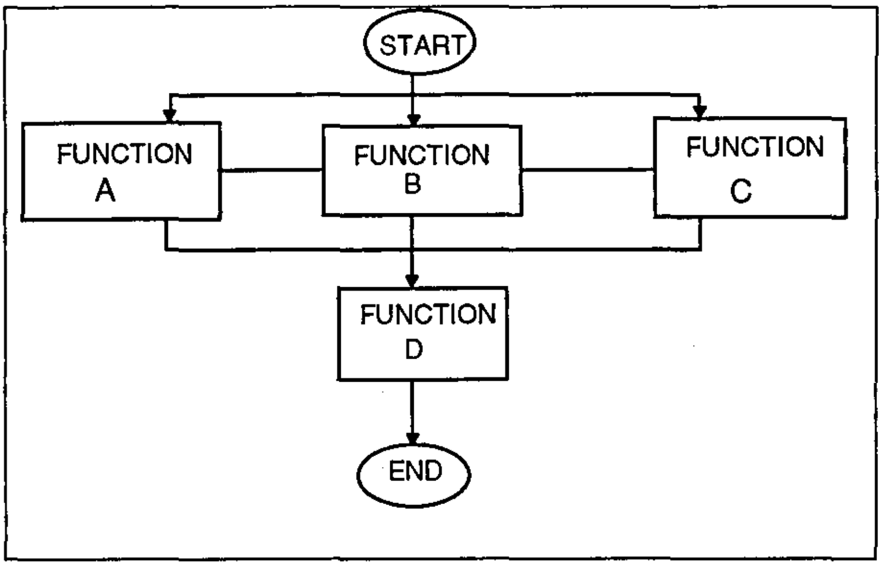


Figure 2.2 :Outline of fork-join technique in function partitioning for parallel program

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

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

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

$$y' = f(x,y) \tag{3.1.1}$$

$$y(x) = y(a) = \alpha \tag{3.1.2}$$

in the neighbourhood of $x = a$, may be expressed as the Taylor series

$$y(x) = y(a) + (x-a)y'(a) + \frac{(x-a)^2}{2!}y''(a) + \frac{(x-a)^3}{3!}y'''(a) + \ldots + \frac{(x-a)^p}{p!}y^p(a) + \ldots \tag{3.1.3}$$

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

$$
\begin{aligned}
y(a) &= \alpha \\
y'(a) &= f(x,y) \\
y''(a) &= \frac{d}{dx} f(x,y) = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx} \\
&= f_x + f_y f \\
y'''(a) &= f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_x f_y + ff_y^2 \\
y^{iv}(a) &= f_{xxx} + 3ff_{xxy} + 3f_x f_{xy} + 5ff_y f_{xy} + 3ff_x f_{yy} + 3f^2 f_{xyy} \\
&\qquad + 4f^2 f_y f_{yy} + f_y f_{xx} + f_x f_y^2 + ff_y^3 + f^3 f_{yyy} \\
y^{v}(a) &= f_{xxxx} + 4ff_{xxxy} + 6f_x f_{xxy} + 9ff_y f_{xxy} + 6f^2 f_{xxyy} + 4f_{xx} f_{xy} \\
&\quad + 8ff_{xy}^2 + 12ff_x f_{xyy} + 7f_x f_y f_{xy} + 9ff_y^2 f_{xy} + 12f^2 f_{xy} f_{yy} + 15f^2 f_y f_{xyy} \\
&\quad + 3f_x^2 f_{yy} + 13ff_x f_y f_{yy} + 4ff_{xx} f_{yy} + 6f^2 f_x f_{yyy} + 4f^3 f_{xyyy} \\
&\quad + 7f^3 f_y f_{yyy} + f_y f_{xxx} + f_{xx} f_y^2 + f_x f_y^3 + ff_y^4 + f^4 f_{yyyy} \\
y^{vi}(a) &= \ldots \quad \text{etc} .
\end{aligned} \tag{3.1.4}
$$

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 $x = x_0$:

$$y(x) = y(x_0) + (x - x_0)y'(x_0) + \frac{(x - x_0)^2}{2!} y''(x_0) + \ldots \quad . \tag{3.1.5}$$

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

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

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

$$y'(x,y) = 1 + x - y \quad , \quad y(0) = 1 \tag{3.1.7}$$

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

$$y'(x_0) = 1 + x - y \quad , \quad y'(0) = 1 - 0 - 1 = 0 \ .$$

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 :

$$\begin{aligned}
y''(x_0) &= 1 - y' \quad , \quad y''(0) = 1 - 0 = 1 \\
y'''(x_0) &= -y'' \quad , \quad y'''(0) = -1 \\
y^{iv}(x_0) &= -y''' \quad , \quad y^{iv}(0) = -(-1) = 1 \\
y^{v}(x_0) &= -y^{iv} \quad , \quad y^{v}(0) = -1 \\
&\vdots \\
&\text{etc .}
\end{aligned}$$

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

$$y(x) = 1 + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{120} + error \ . \qquad (3.1.8)$$

For example , if $x = 0.1$, we obtain

$$\begin{aligned}
y(0.1) &= 1 + \frac{(0.1)^2}{2} - \frac{(0.1)^3}{6} + \frac{(0.1)^4}{24} + \ldots \\
&= 1 + 0.005 - 1.6667 \times 10^{-4} + 4.1667 \times 10^{-6} + \\
&= 1.0048375
\end{aligned}$$

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

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

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 $y(x_1) = y_1$ is used as the tangent through (x_1, y_1) to approximate y_2. 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

$$y_{n+1} = y_n + hf(x_n, y_n) \qquad (3.1.9)$$

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

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n \qquad (3.1.10\text{-i})$$

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n \qquad (3.1.10\text{-ii})$$

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + \frac{h^4}{24} y^{iv}_n \qquad (3.1.10\text{-iii})$$

for solving the problem in equation (3.1.7) where

$$y'_n = 1 + x_n - y_n \qquad (3.1.11\text{-i})$$

$$y''_n = 1 - y'_n \qquad (3.1.11\text{-ii})$$

$$y'''_n = -y''_n \qquad (3.1.11\text{-iii})$$

$$y^{iv}_n = -y'''_n \qquad (3.1.11\text{-iv})$$

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

x_n	First Order	Second Order	Third Order	Fourth Order
0.1	1.0000	1.0050	1.00483	1.00484

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

$$y'' + 101y' + 100y = 0 \quad , \quad y(0) = 1 \, , \, y'(0) = -1 \qquad (3.1.12)$$

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 $y'''(0), y^{iv}(0), y^{v}(0), \ldots$ 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

$$y' = f(x,y) \quad , \quad y(x_0) = \alpha \qquad (3.1.13)$$

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

$$D = \{(x,y) : a \le x \le b, -\infty < y < \infty\} \ ,$$

where a and b are finite and there exists a constant L such that

$$|f(x,y) - f(x,y^*)| \le L|y - y^*| \qquad (3.1.4)$$

for $a \le x \le b$ and all y, y^* , then for any initial value α 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

$$y_{n+1} = y_n + h\phi(x_n, y_n, h) \qquad (3.2.1)$$

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 $\phi(x_n, y_n, h) = y_n' + \frac{h}{2}y_n'' + \frac{h^2}{6}y_n''' + \frac{h^3}{24}y_n^{iv}$ where y_n', y_n'', y_n''' and y_n^{iv} 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_{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).

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

$$LTE = \{y(x_n) + hf(x_n, y_n)\} - \left\{y(x_n) + hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{6}y'''(x_n) + \ldots\right\}$$

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

where $\xi \in [x_n, x_n + h]$ is obtained from the mean value theorem.

Definition 3.2: The Mean Value Theorem

If the function $f(x)$ is continous 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_{n+1} is defined as $y_{n+1} - y(x_{n+1})$ 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 $0(h^{p+1})$ the GTE is $0(h^p)$. 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 satisfiying the differential equation is such that

$$\|y_n - y_n^*\| \le k\|\alpha - \alpha^*\|$$

for all $0 \le h \le h_0$ where $y_n(a) = \alpha$ and $y_n^*(a) = \alpha^*$ and k is the Lipschits 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 $y(x_n)$ at any fixed $x_n \in [a,b]$ as the step length h tends to zero and y_0 tend to α and may be written as

$$\lim_{\substack{h \to 0 \\ x_n \to x}} y_n = y(x)$$

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

$$y' = f(x,y) \quad , \quad y(a) = \eta \quad , \quad f : \Re \times \Re^m \to \Re^m \qquad (3.3.1)$$

is defined by

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

where $$k_i = f\left(x_n + c_i h, y_n + h\sum_{i=1}^{s} a_i k_i\right) \quad , \quad i = 1,2,\ldots,s \qquad (3.3.2)$$

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:

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

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 x s matrix A denoted by

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

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

$$y_{n+1} = y_n + h\sum_{i=1}^{R} w_i k_i \quad , \tag{3.3.3}$$

$$k_1 = f(x, y)$$

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

$$b_i = \sum_{j=1}^{i-1} a_{ij}, \; i = 2, 3, \ldots, R.$$

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,

$$y_{n+1} = y_n + h\sum_{i=1}^{R} w_i k_i \qquad (3.3.5\text{-i})$$

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

The relations in (3.3.5) are used to define the approximation y_{n+1} to $y(x_{n+1})$ 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}, $i, j = 1,2,\ldots,R$ 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

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

$$f_{xy}(\equiv f_{yx}) := \frac{\partial^2 f(x,y)}{\partial x \partial y}$$

all evaluated at the point $(x_n, y(x_n))$. Then, on expanding $y(x_{n+1})$ about x_n as a Taylor series in the form

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2}{2} y''(x_n) + \frac{h^3}{6} y'''(x_n) + 0(h^4) \qquad (3.3.6)$$

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,

$$\begin{aligned} y'(x_n) &= f \\ y''(x_n) &= ff_y \\ y'''(x_n) &= f^2 f_{yy} + ff_y^2 \\ y^{iv}(x_n) &= ff_y^3 + 4f^2 f_y f_{yy} + f^3 f_{yyy} \\ y^{v}(x_n) &= ff_y^4 + 11f^2 f_y^2 f_{yy} + 4f^3 f_{yy}^2 + 7f^3 f_y f_{yyy} + f^4 f_{yyyy} \\ y^{vi}(x_n) &= ff_y^5 + 26f^2 f_y^3 f_{yy} + 34f^3 f_y f_{yy}^2 + 32f^3 f_y^2 f_{yyy} + 15f^4 f_{yy} f_{yyy} \\ &\quad + 11f^4 f_y f_{yyyy} + f^5 f_{yyyyy} \end{aligned} \qquad (3.3.7)$$

By substituting (3.3.7) into (3.3.6) gives the Taylor series as

$$y(x_{n+1}) = y_n + hf + \frac{h^2}{2} ff_y + \frac{h^3}{6}\left(ff_y^2 + f^2 f_{yy}\right) + \frac{h^4}{24}\left(ff_y^3 + 4f^2 f_y f_{yy} + f^3 f_{yyy}\right)$$
$$+ \frac{h^5}{120}\left(ff_y^4 + 11f^2 f_y^2 f_{yy} + 4f^3 f_{yy}^2 + 7f^3 f_y f_{yyy} + f^4 f_{yyyy}\right)$$
$$+ \frac{h^6}{720}\left(ff_y^5 + 26f^2 f_y^3 f_{yy} + 34f^3 f_y f_{yy}^2 + 32f^3 f_y^2 f_{yyy} + 15f^4 f_{yy} f_{yyy}\right.$$
$$\left. + 11f^4 f_y f_{yyyy} + f^5 f_{yyyyy}\right) + 0(h^7) \qquad (3.3.8)$$

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 :

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + \beta_1 h, y_n + ha_1 k_1) \\ k_3 &= f(x_n + \beta_2 h, y_n + ha_2 k_1 + ha_3 k_2) \\ k_4 &= f(x_n + \beta_3 h, y_n + ha_4 k_1 + ha_5 k_2 + ha_6 k_3) \qquad (3.3.9) \\ &\vdots \end{aligned}$$

etc.

where
$$\begin{aligned} \beta_1 &= a_1 \\ \beta_2 &= a_2 + a_3 \\ \beta_3 &= a_4 + a_5 + a_6 \qquad (3.3.10) \\ &\vdots \end{aligned}$$

etc.

where the values of $\beta_1, \beta_2, \beta_3 \ldots$ are obtained after the values for $a_1, a_2, a_3, a_4, \ldots$ 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:

$$k_1 = f(y_n) = f \tag{3.3.11-i}$$

$$k_2 = f(y_n + ha_1k_1)$$

$$= f + ha_1ff_y + \frac{h^2}{2}a_1^2f^2f_{yy} + \frac{h^3}{6}a_1^3f^3f_{yyy} + \frac{h^4}{24}a_1^4f^4f_{yyyy} + \frac{h^5}{120}a_1^5f^5f_{yyyyy} + \ldots \tag{3.3.11-ii}$$

$$k_3 = f(y_n + ha_2k_1 + ha_3k_2)$$

$$= f + h(a_2 + a_3)ff_y + h^2\left[a_1a_3ff_y^2 + \frac{1}{2}(a_2 + a_3)^2f^2f_{yy}\right]$$

$$+ h^3\left[\frac{1}{2}a_1^2a_3f^2f_yf_{yy} + a_1a_3(a_2 + a_3)f^2f_yf_{yy} + \frac{1}{6}(a_2 + a_3)^3f^3f_{yyy}\right]$$

$$+ h^4\left[\frac{1}{6}a_1^3a_3f^3f_yf_{yyy} + \frac{1}{2}a_1^2a_3^2f^2f_y^2f_{yy} + \frac{1}{2}(a_2 + a_3)a_1^2a_3f^3f_{yy}^2\right.$$

$$\left.+ \frac{1}{2}(a_2 + a_3)^2a_1a_3f^3f_yf_{yyy} + \frac{1}{24}(a_2 + a_3)^4f^4f_{yyyy}\right] \tag{3.3.11-iii}$$

$$k_4 = f(y_n + ha_4k_1 + ha_5k_2 + ha_6k_3)$$

$$= f + h(a_4 + a_5 + a_6)ff_y + h^2\left[(a_1a_5 + (a_2 + a_3)a_6)ff_y^2\right.$$

$$\left.+ \frac{1}{2}(a_4 + a_5 + a_6)^2f^2f_{yy}\right] + h^3\left[\left\{\frac{1}{2}a_1^2a_5 + \frac{1}{2}(a_2 + a_3)^2a_6\right.\right.$$

$$\left.+(a_4 + a_5 + a_6)(a_1a_5 + (a_2 + a_3)a_6)\right\}f^2f_yf_{yy}$$

$$\left.+a_1a_3a_6ff_y^3 + \frac{1}{6}(a_4 + a_5 + a_6)^3f^3f_{yyy}\right]$$

$$+h^4\left[\frac{1}{6}\left\{a_1^3a_5 + (a_2 + a_3)^3 + 3(a_4 + a_5 + a_6)^2(a_1a_5 + (a_2 + a_3)a_6)\right\}f^3f_yf_{yyy}\right.$$

$$+ \left\{\left(\frac{1}{2}a_1^2a_3a_6 + a_1a_3a_6(a_2 + a_3)\right) + \frac{1}{2}(a_1a_5 + (a_2 + a_3)a_6)^2\right.$$

$$\left.+ a_1a_3a_6(a_4 + a_5 + a_6)\right\}f^2f_y^2f_{yy}$$

$$+ \frac{1}{2}\left\{(a_4 + a_5 + a_6)(a_1^2a_5 + (a_2 + a_3)^2a_6)\right\}f^3f_{yy}^2$$

$$\left.+ \frac{1}{24}(a_4 + a_5 + a_6)^4f^4f_{yyyy}\right] \tag{3.3.11-iv}$$

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

$$y_{n+1} = y_n + h(w_1 k_1 + w_2 k_2) \qquad (3.3.12)$$

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

$$y_{n+1} = y_n + h(w_1 + w_2)f + h^2(a_1 w_2) f f_y + 0(h^3) \ . \qquad (3.3.13)$$

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 :

$$hf: \quad 1 - w_1 - w_2 = 0 \qquad (3.3.14\text{-i})$$

$$h^2 f f_y: \quad 1 - 2a_1 w_2 = 0 \qquad (3.3.14\text{-ii})$$

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

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

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

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2) \qquad (3.3.15)$$

where $k_1 = f(y_n)$

$k_2 = f(y_n + hk_1)$

which is known as the second order Runge-Kutta method or improved Euler method and can be written in Butcher array form as

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

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

$$y_{n+1} = y_n + h(w_1k_1 + w_2k_2 + w_3k_3) \qquad (3.3.16)$$

where $k_1 = f(y_n)$

$$k_2 = f(y_n + ha_1k_1)$$

$$k_3 = f(y_n + ha_2k_1 + ha_3k_2)$$

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 (3.3.160, we have

$$y_{n+1} = y_n + h(w_1 + w_2 + w_3)f + h^2(a_1w_2 + a_2w_3 + a_3w_3)ff_y$$

$$+ \frac{h^3}{2}\left[(2a_1a_3w_3)ff_y^2 + (a_1^2w_2 + a_2^2w_3 + 2a_2a_3w_3 + a_3^2w_3)f^2f_{yy}\right] + 0(h^4)$$

$$\ldots \qquad (3.3.17)$$

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

$$hf \quad : \quad 1 - w_1 - w_2 - w_3 = 0 \qquad (3.3.18\text{-i})$$

$$h^2ff_y \quad : \quad 1 - 2a_1w_2 - 2a_2w_3 - 2a_3w_3 = 0 \qquad (3.3.18\text{-ii})$$

$$h^3ff_y^2 \quad : \quad 1 - 6a_1a_3w_3 = 0 \qquad (3.3.18\text{-iii})$$

$$h^3f^2f_{yy} \quad : \quad 1 - 3a_1^2w_2 - 3a_2^2w_3 - 6a_2a_3w_3 - 3a_3^2w_3 = 0 \qquad (3.3.18\text{-iv})$$

By taking $a_1 = \frac{1}{2}$ and $a_2 + a_3 = 1$ then using Mathematica to solve equations (3.3.18-i)-(3.3.18-iv), we obtain

$$w_1 = \frac{1}{6}, w_2 = \frac{2}{3}, w_3 = \frac{1}{6}, a_1 = \frac{1}{2}, a_2 = -1, a_3 = 2 \ \ldots \qquad (3.3.19)$$

By substituting (3.3.19) into (3.3.16), the equation (3.3.16) now can be written in the form

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3) \qquad (3.3.20)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f(x_n + h, y_n - hk_1 + 2hk_2)$$

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

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array}$$

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

$$y_{n+1} = y_n + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4] \qquad (3.3.21)$$

where $k_1 = f(y_n)$

$$k_2 = f(y_n + ha_1k_1)$$

$$k_3 = f(y_n + ha_2k_1 + ha_3k_2)$$

$$k_4 = f(y_n + ha_4k_1 + ha_5k_2 + ha_6k_3)$$

By substituting $k_i, i=1,2,3,4$ 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,

$$
\begin{aligned}
y_{n+1} = y_n &+ h(w_1+w_2+w_3+w_4)f \\
&+ h^2(a_1w_2+a_2w_3+a_3w_3+a_4w_4+a_5w_4+a_6w_4)ff_y \\
&+ h^3\Big[(a_1a_3w_3+a_1a_5w_4+a_2a_6w_4+a_3a_6w_4)ff_y^2 \\
&+\frac{1}{2}(a_1^2w_2+a_2^2w_3+2a_2a_3w_3+a_3^2w_3+a_4^2w_4+2a_4a_5w_4 \\
&\qquad +a_5^2w_4+2a_4a_6w_4+2a_5a_6w_4+a_6^2w_4)f^2f_{yy}\Big] \\
&+ h^4\Big[(a_1a_3a_6w_4)ff_y^3+\frac{1}{2}(a_1^2a_3w_3+2a_1a_2a_3w_3+2a_1a_3^2w_3 \\
&+ a_1^2a_5w_4+2a_1a_4a_5w_4+2a_1a_5^2w_4+a_2^2a_6w_4+2a_2a_3a_6w_4 \\
&+ a_3^2a_6w_4+2a_2a_4a_6w_4+2a_3a_4a_6w_4+2a_1a_5a_6w_4+2a_2a_5a_6w_4 \\
&+ 2a_3a_5a_6w_4+2a_2a_6^2w_4+2a_3a_6^2w_4)f^2f_yf_{yy} \\
&+\frac{1}{6}(a_1^3w_2+a_2^3w_3+3a_2^2a_3w_3+3a_2a_3^2w_3+a_3^3w_3+a_4^3w_4 \\
&+ 3a_4^2a_5w_4+3a_4a_5^2w_4+a_5^3w_4+3a_4^2a_6w_4+6a_4a_5a_6w_4 \\
&+ 3a_5^2a_6w_4+3a_4a_6^2w_4+3a_5a_6^2w_4+a_6^3w_4)f^3f_{yyy}\Big]
\end{aligned} \qquad (3.3.22)
$$

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

$hf \quad : \quad 1-w_1-w_2-w_3-w_4=0$ (3.3.23-i)

$h^2ff_y : \quad 1-2a_1w_2-2a_2w_3-2a_3w_3-2a_4w_4-2a_5w_4-2a_6w_4=0$

... (3.3.23-ii)

$h^3ff_y^2 : \quad 1-6a_1a_3w_3-6a_1a_5w_4-6a_2a_6w_4-6a_3a_6w_4=0$ (3.3.23-iii)

$h^3f^2f_{yy} : \quad 1-3a_1^2w_2-3a_2^2w_3-6a_2a_3w_3-3a_3^2w_3-3a_4^2w_4-6a_4a_5w_4$

$-3a_5^2w_4-6a_4a_6w_4-6a_5a_6w_4-3a_6^2w_4=0$ (3.3.23-iv)

$h^4ff_y^3 : \quad 1-24a_1a_3a_6w_4=0$ (3.3.23-v)

$h^4f^2f_yf_{yy} : 1-3a_1^2a_3w_3-6a_1a_2a_3w_3-6a_1a_3^2w_3-3a_1^2a_5w_4-6a_1a_4a_5w_4$

$-6a_1a_5^2w_4-3a_2^2a_6w_4-6a_2a_3a_6w_4-3a_3^2a_6w_4-6a_2a_4a_6w_4$

$-6a_3a_4a_6w_4-6a_1a_5a_6w_4-6a_2a_5a_6w_4-6a_3a_5a_6w_4$

$-6a_2a_6^2w_4-6a_3a_6^2w_4=0$ (3.3.23-vi)

$$h^4 f^3 f_{yyy}: \quad 1 - 4a_1^3 w_2 - 4a_2^3 w_3 - 12a_2^2 a_3 w_3 - 12a_2 a_3^2 w_3 - 4a_3^3 w_3 - 4a_4^3 w_4$$
$$- 12a_4^2 a_5 w_4 - 12a_4 a_5^2 w_4 - 4a_5^3 w_4 - 12a_4^2 a_6 w_4 - 24a_4 a_5 a_6 w_4$$
$$- 12a_5^2 a_6 w_4 - 12a_4 a_6^2 w_4 - 12a_5 a_6^2 w_4 - 4a_6^3 w_4 = 0 \quad (3.3.23\text{-vii})$$

This is an underdetermined system of nonlinear equations and in general more then one solution can be obtained. For solving this system of nonlinear equations, we introduce three more additional equations by fixing the values of β_1, β_2 and β_3 where

$$\beta_1 = a_1 = \frac{1}{2} \quad (3.3.23\text{-viii})$$

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

$$\beta_3 = a_4 + a_5 + a_6 = 1 \quad (3.3.23\text{-x})$$

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

$$w_1 = \frac{1}{6}, \; w_2 = \frac{4 - \frac{1}{a_3}}{6}, \; w_3 = \frac{1}{6a_3}, \; w_4 = \frac{1}{6}$$

$$a_1 = \frac{1}{2}, \; a_2 = \frac{1 - 2a_3}{2}, \; a_4 = 0, \; a_5 = \frac{2 - \frac{1}{a_3}}{2}, \; a_6 = \frac{1}{2a_3}$$

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

$$w_1 = \frac{1}{6}, \; w_2 = \frac{1}{3}, \; w_3 = \frac{1}{3}, \; w_4 = \frac{1}{6}, \; a_1 = \frac{1}{2}$$

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

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

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

where $\quad k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_2\right)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

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, Raltson [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

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

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \left(-\frac{1}{2} + \frac{1}{\sqrt{2}}\right)hk_1 + \left(1 - \frac{1}{\sqrt{2}}\right)hk_2\right)$$

$$k_4 = f\left(x_n + h, y_n - \frac{1}{\sqrt{2}}hk_2 + \left(1 + \frac{1}{\sqrt{2}}\right)hk_3\right)$$

The study of higher order Runge-Kutta methods involves extremly 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

$$y_{n+1} = y_n + \frac{h}{192}\left[23k_1 + 125k_3 - 81k_5 + 125k_6\right] \qquad (3.3.27)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{3}, y_n + \frac{1}{3}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{2}{5}h, y_n + \frac{1}{25}h(4k_1 + 6k_2)\right)$$

$$k_4 = f\left(x_n + h, y_n + \frac{1}{4}h(k_1 - 12k_2 + 15k_3)\right)$$

$$k_5 = f\left(x_n + \frac{2}{3}h, y_n + \frac{1}{81}h(6k_1 + 90k_2 - 50k_3 + 8k_4)\right)$$

$$k_6 = f\left(x_n + \frac{4}{5}h, y_n + \frac{1}{75}h(6k_1 + 36k_2 + 10k_3 + 8k_4)\right) \quad .$$

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

$$y_{n+1} = y_n + \frac{h}{840}\left[41k_1 + 216k_3 + 27k_4 + 272k_5 + 27k_6 + 216k_7 + 41k_8\right] \quad (3.3.28)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{9}, y_n + \frac{1}{9}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{6}, y_n + \frac{1}{24}h(k_1 + 3k_2)\right)$$

$$k_4 = f\left(x_n + \frac{h}{3}, y_n + \frac{1}{6}h(k_1 - 3k_2 + 4k_3)\right)$$

$$k_5 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{8}h(-5k_1 + 27k_2 - 24k_3 + 6k_4)\right)$$

$$k_6 = f\left(x_n + \frac{2}{3}h, y_n + \frac{1}{9}h(221k_1 - 981k_2 + 867k_3 - 102k_4 + k_5)\right)$$

$$k_7 = f\left(x_n + \frac{5}{6}h, y_n + \frac{1}{48}h(-183k_1 + 678k_2 - 472k_3 - 66k_4 + 80k_5 + 3k_6)\right)$$

$$k_8 = f\left(x_n + h, y_n + \frac{1}{82}h(716k_1 - 2079k_2 + 1002k_3 + 834k_4 \right.$$
$$\left. -454k_5 - 9k_6 + 72k_7)\right)$$

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

$$y' = \lambda y \quad , \quad \lambda \in C \quad \text{and} \quad \mathrm{Re}(\lambda) < 0 \ . \qquad (3.3.29)$$

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

$$y_{n+1} = Q(z)y_n \qquad (3.3.30)$$

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

$$|Q(z)| < 1 \qquad (3.3.31)$$

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

$$k_1 = f(y_n)$$

$$= \lambda y_n \qquad (3.3.32\text{-i})$$

$$k_2 = f(y_n + a_1 hk_1)$$

$$= \lambda(1 + a_1 z)y_n \qquad (3.3.32\text{-ii})$$

$$k_3 = f(y_n + a_2 hk_1 + a_3 hk_2)$$

$$= \lambda(1 + (a_2 + a_3)z + a_1 a_3 z^2)y_n \qquad (3.3.32\text{-iii})$$

$$k_4 = f(y_n + a_4 hk_1 + a_5 hk_2 + a_6 hk_3)$$

$$= \lambda\{1 + (a_4 + a_5 + a_6)z + za_6((a_2 + a_3)z + a_1 a_3 z^2)\}y_n$$

$$\ldots \qquad (3.3.32\text{-iv})$$

Substituting (3.3.32-i)-(3.3.32-iv) into equation (3.3.21), we obtain

$$y_{n+1} = \left[1 + z\left\{w_1 + w_2(1 + a_1 z) + w_3\left(1 + (a_2 + a_3)z + a_1 a_3 z^2\right)\right.\right.$$
$$\left.\left. + w_4\left(1 + (a_4 + a_5 + a_6)z + a_1 a_5 z^2 + z a_6\left(1 + (a_2 + a_3)z + a_1 a_3 z^2\right)\right)\right\}\right] y_n \quad \ldots \quad (3.3.33)$$

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

$$y_{n+1} = \left[1 + z + \frac{z^2}{2!} + \ldots + \frac{z^p}{p!}\right] y_n + 0(z^{p+1})$$

or $$Q(z) = 1 + z + \frac{z^2}{2!} + \ldots + \frac{z^p}{p!} + 0(z^{p+1}) \quad (3.3.34)$$

By substituting $k_i, 1 \le i \le 4$ 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

$$Q(z) = 1 + z + 0(z^2) \quad (3.3.35\text{-i})$$

b) Second order Runge-Kutta or improved Euler method

$$Q(z) = 1 + z + \frac{z^2}{2!} + 0(z^3) \quad (3.3.35\text{-ii})$$

c) Third order Runge-Kutta or Kutta method

$$Q(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + 0(z^4) \quad (3.3.35\text{-iii})$$

d) Fourth order Runge-Kutta or classical method

$$Q(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!} + 0(z^5) \quad (3.3.35\text{-iv})$$

and the stability region for all stability functions in (3.3.35-i)-(3.3.35-iv) are shown in Figure 3.1.

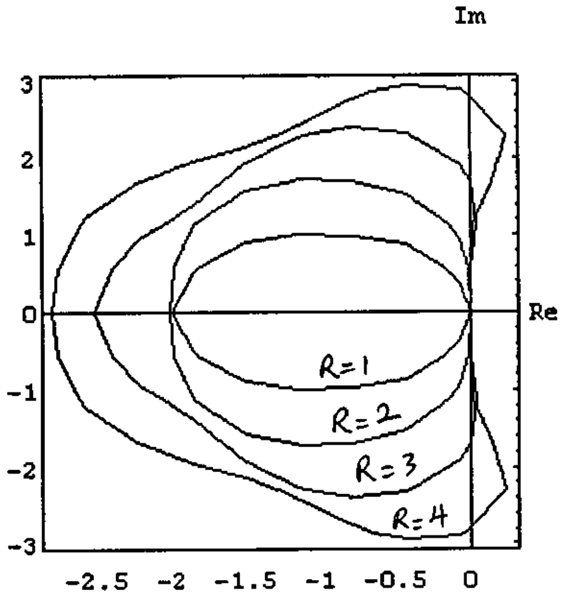


Figure 3.1: Stability region for the Runge-Kutta method for $R = 1, 2, 3, 4$

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

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} \tag{3.4.1}$$

where α_j and β_j are constants; $f_{n+j} = f(x_{n+j}, y_{n+j})$ with the contraints $\alpha_k \neq 0$ and that not both α_0 and β_0 are zero. Equation (3.4.1) can be arranged to give

$$y_{n+k} = h\beta_k f(x_{n+k}, y_{n+k}) + \sum_{j=0}^{k-1} \left(h\beta_j f_{n+j} - \alpha_j y_{n+j}\right) \tag{3.4.2}$$

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

$$\mathcal{L}[y(x); h] = \sum_{j=0}^{k} \left[\alpha_j y(x + jh) - h\beta_j y'(x + jh)\right] \tag{3.4.3}$$

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

$$y(x+jh) = y(x) + \sum_{i=1}^{\infty} \frac{(jh)^i}{i!} y^{(i)}(x) \tag{3.4.4}$$

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

$$y'(x+jh) = y'(x) + \sum_{i=1}^{\infty} \frac{(jh)^i}{i!} y^{(i+1)}(x). \tag{3.4.5}$$

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

$$\mathcal{L}[y(x);h] = C_0 y(x) + C_1 h y'(x) + \ldots + C_q h^q y^{(q)}(x) + \ldots, \tag{3.4.6}$$

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

Definition 3.7:

The difference operator $\mathcal{L}$ 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 α_j and β_j are given as follows:

$$\begin{aligned} C_0 &= \sum_{j=0}^{k} \alpha_j \\ C_1 &= \sum_{j=0}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j \\ &\vdots \\ C_q &= \frac{1}{q!} \sum_{j=0}^{k} j^q \alpha_j - \frac{1}{(q-1)!} \sum_{j=0}^{q-1} \beta_j \quad , \quad q = 2,3,\ldots \end{aligned} \tag{3.4.7}$$

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

$$\sum_{j=0}^{k} \alpha_j = 0$$

and $$\sum_{j=0}^{k} j\alpha_j = \sum_{j=0}^{k} \beta_j \tag{3.4.8}$$

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

$$\rho(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j \tag{3.4.9}$$

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

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

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

$$\ell(x) = h(\rho'(1) - \sigma(1)) = 0 \tag{3.4.11}$$

Thus, for a consistent method, the first characteristic polynomial $\rho(\zeta)$ always has a root at +1 and will be called the principal root with label ζ_1. While the remaining roots, $\zeta_s, s = 2,3,\ldots,k$ 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

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

which has the stability polynomial equation

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

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 ρ and σ, then from (3.4.13) the stability polynomial may be written as

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

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 $|r_j| < 1, j = 1,2,\ldots,k$. 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

$$y_{n+k} - y_{n+k-1} = h\sum_{j=0}^{k}\beta_j f_{n+j} \qquad (3.4.15)$$

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

$$y_{n-s+1} - y_{n-s} = h\sum_{s=0}^{k-t} B_{n-s} f_{n-s} \quad , \quad t = \begin{cases} 1 \text{ if } s = 0 \\ 0 \text{ if } s \geq 1 \end{cases} \qquad (3.4.16)$$

where $B_{n-s} = \dfrac{b_{n-s}}{d}$ with b_{n-s} under columns headed f_{n-s} together with the denominator d, stepnumber k, order p and error constant C_{p+1} 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.

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

f_n	f_{n-1}	f_{n-2}	f_{n-3}	f_{n-4}	f_{n-5}	d	k	p	C_{p+1}
1						1	1	1	$\frac{1}{2}$
3	-1					2	2	2	$\frac{5}{12}$
23	-16	5				12	3	3	$\frac{3}{8}$
55	-59	37	-9			24	4	4	$\frac{251}{720}$
1901	-2774	2616	-1274	251		720	5	5	$\frac{95}{288}$
4277	-7923	9982	-7298	2877	-475	1440	6	6	$\frac{19087}{60480}$

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

f_n	f_{n-1}	f_{n-2}	f_{n-3}	f_{n-4}	f_{n-5}	d	k	p	C_{p+1}
1	1					2	1	2	$-\frac{1}{12}$
5	8	-1				12	2	3	$-\frac{1}{24}$
9	19	-5	1			24	3	4	$-\frac{19}{720}$
251	646	-264	106	-19		720	4	5	$-\frac{3}{160}$
475	1427	-798	482	-173	27	1440	5	6	$-\frac{863}{60480}$

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 $x_{n+1}, x_{n+2}, \ldots, x_{n+N}$. 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

$$y_{n+2} = y_n + \frac{h}{3}\left[k_1 + 4k_5 + k_6\right] \qquad (3.4.17)$$

where

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

$$k_2 = f(x_n + h, y_n + hk_1)$$

$$k_3 = f\left(x_n + h, y_n + \frac{h}{2}k_1 + \frac{h}{2}k_2\right)$$

$$k_4 = f(x_n + 2h, y_n + 2hk_3)$$

$$k_5 = f\left(x_n + h, y_n + \frac{h}{12}(5k_1 + 8k_3 - k_4)\right)$$

$$k_6 = f\left(x_n + 2h, y_n + \frac{h}{3}(k_1 + k_4 + 4k_5)\right).$$

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

$$\sum_{j=0}^{k}\alpha_j y_{n+j} = h\sum_{j=0}^{k}\beta_j f_{n+j} + h\beta_v f_{n+v} \qquad (3.4.18)$$

where $\alpha_k = 1, \alpha_0$ and β_0 are not both zero $v \notin \{0,1,\ldots,k\}$ and for $f_{n+v} = f(x_{n+v}, y_{n+v})$. Equation (3.4.18) is explicit if $\beta_k = 0$ and implicit if $\beta_k \neq 0$ and has a predictor to estimate y_{n+v} of the form

$$y_{n+v} + \sum_{j=0}^{k-1}\bar{\alpha}_j y_{n+j} = h\sum_{j=0}^{k-1}\bar{\beta}_j f_{n+j} \qquad (3.4.19)$$

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 l derivative of y may be written

$$\sum_{j=0}^{k}\alpha_j y_{n+j} = \sum_{i=1}^{l} h^i \sum_{j=0}^{k}\beta_{ij} y_{n+j} \qquad (3.4.20)$$

where $\alpha_k = 1$ and one of $\alpha_0, \beta_{i0}, i = 1,2,\ldots,l$ 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

$$y(x) = C_1 \exp(ax) + C_2 \exp(bx) + C \qquad (3.5.1)$$

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

$$y'' + 101y' + 100y = 0 \text{ , } y(0) = 1.01 \text{ , } y'(0) = -2 \qquad (3.5.2)$$

The equation (3.5.2) can be written as the system

$$y_1' = y_2 \qquad ; \quad y_1(0) = 1.01$$

$$y_2' = -100y_1 - 101y_2 \qquad ; \quad y_2(0) = -2 \qquad (3.5.3)$$

which has the solution

$$y_1(x) = 0.01\exp(-100x) + \exp(-x)$$

$$y_2(x) = -\exp(-100x) - \exp(-x) \text{ .} \qquad (3.5.4)$$

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 $h < \frac{2.78}{100}$. 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

$$y' = Ay + \phi(x) \tag{3.5.5}$$

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

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

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

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

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) $\mathrm{Re}(\lambda_j) < 0, j = 1, 2, \ldots, m$

ii) $Max_{j=1,2,\ldots,m} |\mathrm{Re}(\lambda_j)| >> Min_{j=1,2,\ldots,m} |\mathrm{Re}(\lambda_j)|$

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

$$s = \frac{Max_{j=1,2,\ldots,m} |\mathrm{Re}(\lambda_j)|}{Min_{j=1,2,\ldots,m} |\mathrm{Re}(\lambda_j)|}$$

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

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 $\text{Re}(h\lambda) < 0$.

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 $w_\alpha = \{h\lambda | -\alpha < \Pi - Arg(h\lambda) < \alpha\}$; 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_1 and R_2

ii) it is accurate for all $h \in R_2$ when applied to the scalar test equation $y' = \lambda y$, λ a complex constant with $\mathrm{Re}(\lambda) < 0$, where

$$R_1 = \{h\lambda | \mathrm{Re}(h\lambda) < -a\} \text{ and}$$

$$R_2 = \{h\lambda | -a \le \mathrm{Re}(h\lambda) \le b, -c \le \mathrm{Im}(h\lambda) \le c\}$$

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

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 $y' = \lambda y$, λ a complex constant with $\mathrm{Re}(\lambda) < 0$, it yields $y_{n+1} = Q(h\lambda) y_n$ where $Q(h\lambda) \to 0$ and $\mathrm{Re}(h\lambda) \to -\infty$.

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:

L-stability $\Rightarrow$ A-stability $\Rightarrow$ stiff-stability
$\Rightarrow A(\alpha)$-stability $\Rightarrow A(0)$-stability $\Rightarrow A_0$-stability.

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

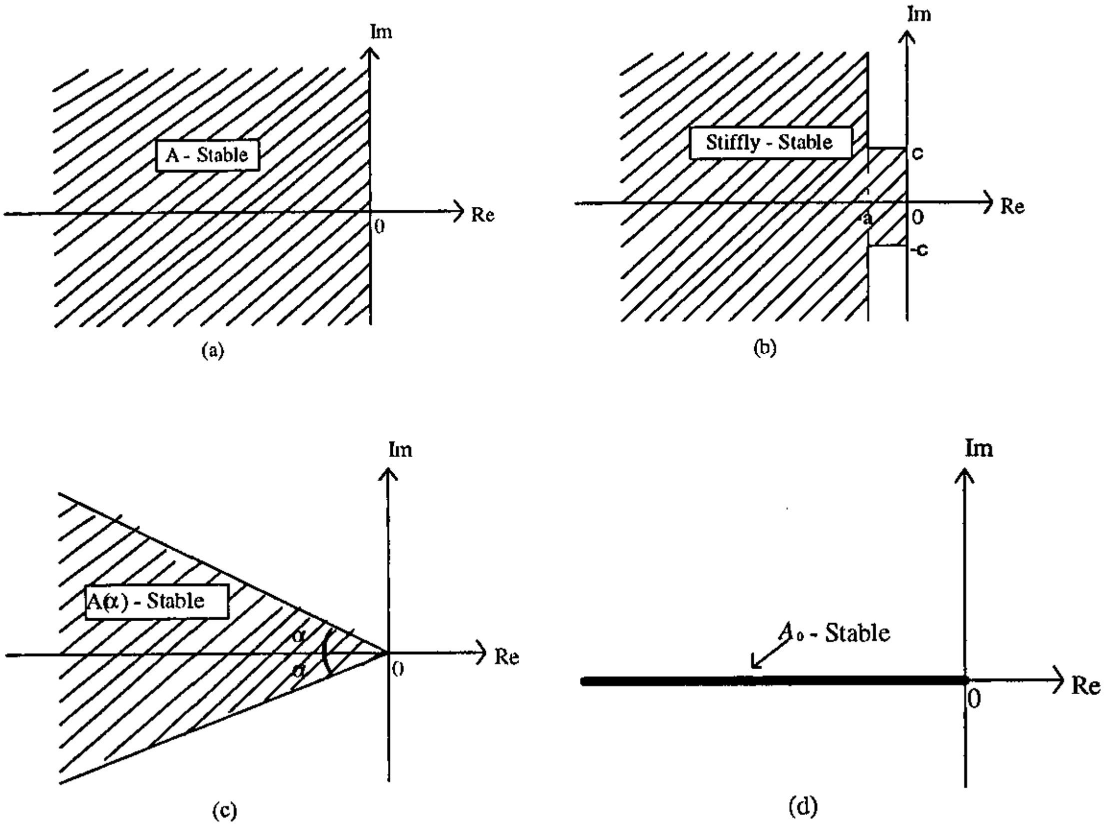


Figure 3.2: The stability region for stiff problems

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

$$\sum_{j=0}^{k}\alpha_j y_{n+j} = h\beta_k f_{n+k} \qquad (3.5.8)$$

where $\alpha_k = 1, \alpha_0 \neq 0$ and $\beta_0 \neq 0$ and k is equal to the step number k. The region of absolute stability and coefficients α_j , β_k 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

k=p	α_6	α_5	α_4	α_3	α_2	α_1	α_0	β_k	$-a$
1						1	-1	1	0
2					1	$-\frac{4}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	0
3				1	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$	$\frac{6}{11}$	0.1
4			1	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$	$\frac{12}{25}$	0.7
5		1	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$	$\frac{60}{137}$	2.4
6	1	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$	$\frac{60}{147}$	6.1

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

$$y_{n+1} - y_n = hf_{n+1}$$

b) second order-method, k=2

$$y_{n+2} - \tfrac{4}{3} y_{n+1} + \tfrac{1}{3} y_n = \tfrac{2}{3} hf_{n+2}$$

c) Third order-method, k=3

$$y_{n+3} - \tfrac{18}{11} y_{n+2} + \tfrac{9}{11} y_{n+1} - \tfrac{2}{11} y_n = \tfrac{6}{11} hf_{n+3}.$$

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

$$y' = f(x, y) \quad , \quad y(x_0) = y_0 \tag{4.0.1}$$

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

$$y_{n+1} = y_n + \frac{h}{3}\left[\frac{k_1 + k_2}{2} + \frac{k_2 + k_3}{2} + \frac{k_3 + k_4}{2}\right] \tag{4.1.1}$$

where ,

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + a_1 h, y_n + h a_1 k_1) \\ k_3 &= f(x_n + (a_2 + a_3)h, y_n + h a_2 k_1 + h a_3 k_2) \\ k_4 &= f(x_n + (a_4 + a_5 + a_6)h, y_n + h a_4 k_1 + h a_5 k_2 + h a_6 k_3) \end{aligned} \tag{4.1.2}$$

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

$$a_1 = \frac{1}{2} \quad , \quad a_2 = 0 \quad , \quad a_3 = \frac{1}{2} \quad , \quad a_4 = 0 \quad , \quad a_5 = 0 \quad , \quad a_6 = 1 .$$

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

$$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3}\frac{(2k_i k_{i+1})}{(k_i + k_{i+1})}\right] \qquad (4.1.3)$$

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

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

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

$$y_{n+1} = y_n + \frac{h}{3}\left(\sum_{i=1}^{3}\sqrt{k_i k_{i+1}}\right) \qquad (4.1.4)$$

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, $a_i, 1 \le i \le 6$ in (4.1.2) , to give

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

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_oM), centroidal mean (C_eM) 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 CONTRAHARMONIC MEAN (C_oM) FORMULA

4.2.1 Second Order Contraharmonic Mean (C_oM) Formula

Following Eves [1983] the largest functional mean is given by the Contraharmonic mean (C_oM) formula

$$y_{n+1} = y_n + h\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) \qquad (4.2.1)$$

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

$$k_2 = f(x_n + a_1 h, y_n + h a_1 k_1) \qquad (4.2.2)$$

Second order accuracy is obtained for formula (4.2.1) from the solution of the equation of condition, i.e.,

$$h^2 f f_y \quad : \qquad 1 - a_1 = 0 \qquad (4.2.3)$$

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

$$y_{n+1} = y_n + \frac{h}{2}\left[\sum_{i=1}^{2}\frac{(k_i^2 + k_{i+1}^2)}{(k_i + k_{i+1})}\right] \qquad (4.2.4)$$

and is obtained by using the related Contraharmonic mean

$$C_o M = \frac{2(AM)^2 - (GM)^2}{(AM)} \quad . \qquad (4.2.5)$$

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,

$$y_{n+1} = y_n + \frac{h}{2}\left[\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) + \left(\frac{k_2^2 + k_3^2}{k_2 + k_3}\right)\right] \qquad (4.2.6)$$

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

$$k_2 = f(x_n + a_1 h, y_n + h a_1 k_1) \qquad (4.2.7)$$

$$k_3 = f(x_n + (a_2 + a_3)h, y_n + a_2 h k_1 + a_3 h k_2)$$

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.,

$$h^2 ff_y: \qquad 2a_1 + a_2 + a_3 = 2, \qquad (4.2.8\text{-i})$$

$$h^3 f^2 f_{yy}: \qquad 3(a_2 + a_3)^2 + 6a_1^2 = 4, \qquad (4.2.8\text{-ii})$$

$$h^3 ff_y^2 \quad : \quad 9a_1^2 + 6a_1a_3 + 3a_1a_2 + 3(a_2 + a_3)^2 = 8 \qquad (4.2.8\text{-iii})$$

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

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

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,

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

as a direct extension of method (4.2.9) where,

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f(x_n + a_1h, y_n + ha_1k_1) \qquad (4.2.10) \\ k_3 &= f(x_n + (a_2 + a_3)h, y_n + a_2hk_1 + a_3hk_2) \\ k_4 &= f(x_n + (a_4 + a_5 + a_6)h, y_n + a_4hk_1 + a_5hk_2 + a_6hk_3) \end{aligned}$$

The Taylor series expansion of $k_i, 1 \le i \le 4$ 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_{n+1} in terms of the function and its derivatives and the parameters $a_i, 1 \le i \le 6$, so that it can be matched with the Taylor series expansion of $y(x_{n+1})$ 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

$$\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \quad , \qquad 1 \le i \le 3 \ . \tag{4.2.11}$$

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

$$(k_1 + k_2)(k_2 + k_3)(k_3 + k_4) \tag{4.2.12}$$

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

$$y_{n+1} = y_n + \frac{UPPER}{LOWER} \tag{4.2.13}$$

where ,

$$UPPER = \frac{h}{3}\left[(k_1^2 + k_2^2)(k_2 + k_3)(k_3 + k_4) + (k_2^2 + k_3^2)(k_1 + k_2)(k_3 + k_4) \right.$$
$$\left. + (k_3^2 + k_4^2)(k_1 + k_2)(k_2 + k_3)\right] \tag{4.2.14}$$

and ,

$$LOWER = (k_1 + k_2)(k_2 + k_3)(k_3 + k_4), \tag{4.2.15}$$

while the Taylor series expansion of $y(x_{n+1})$ was given in equation (3.3.8).

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

$$ERROR = TAYLOR - \frac{UPPER}{LOWER}$$
$$(TAYLOR \times LOWER) - UPPER = (LOWER \times ERROR) \tag{4.2.16}$$

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

$$h^2 ff_y: \quad -2a_1 - 2s_2 - s_3 + 3 = 0 \tag{4.2.17-i}$$

$$h^3 f^2 f_{yy} \quad : \quad -2a_1^2 - 2s_2^2 - s_3^2 + 2 = 0 \tag{4.2.17-ii}$$

$$h^3 ff_y^2 \quad : \quad 2 + 6a_1 - 6a_1^2 - 4a_1a_3 - 2a_1a_5 + 6s_2 - 6a_1s_2 - 2a_6s_2 - 6s_2^2$$
$$+3s_3 - 4a_1s_3 - 2s_2s_3 - 2s_3^2 = 0 \tag{4.2.17-iii}$$

$$h^4 f^3 f_{yyy} \quad : \quad -4a_1^3 - 4s_2^3 - 2s_3^3 + 3 = 0 \qquad (4.2.17\text{-iv})$$

$$h^4 f f_y^3 \quad : \quad 1 + 4a_1 + 3a_1^2 + 12a_1a_3 - 12a_1^2a_3 + 6a_1a_5 - 8a_1^2a_5 - 4a_1a_3a_6$$
$$+ 4s_2 + 9a_1s_2 - 9a_1^2s_2 - 24a_1a_3s_2 - 4a_1^3 - 4a_1a_5s_2 + 6a_6s_2 - 8a_1a_6s_2$$
$$+ 3s_2^2 - 9a_1s_2^2 - 4a_6s_2^2 - 4s_2^3 + 2s_3 + 6a_1s_3 - 7a_1^2s_3 - 4a_1a_3s_3 + 3s_2s_3$$
$$- 8a_1a_5s_3 - 3a_1s_2s_3 - 8a_6s_2s_3 - 2s_2^2s_3 - 4a_1s_3^2 - 2s_2s_3^2 = 0 \qquad (4.2.17\text{-v})$$

$$h^4 f^2 f_y f_{yy} \quad : \quad 4 + 4a_1 + 6a_1^2 - 12a_1^3 - 4a_1^2a_3 - 2a_1^2a_5 + 4s_2 - 6a_1^2s_2$$
$$- 8a_1a_3s_2 + 6s_2^2 - 6a_1s_2^2 - 2a_6s_2^2 - 12s_2^3 + 2s_3 - 4a_1^2s_3$$
$$- 4a_1a_5s_3 - 4a_6s_2s_3 - 2s_2^2s_3 + 3s_3^2 - 4a_1s_3^2 - 2s_2s_3^2 - 4s_3^3 = 0$$
$$\ldots \qquad (4.2.17\text{-vi})$$

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_oM) Runge-Kutta formula in equations (4.2.9)-(4.2.10) can now be written as

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

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{1}{8}hk_1 + \frac{3}{8}hk_2) \qquad (4.2.18)$$

$$k_4 = f(x_n + h, y_n + \frac{1}{4}hk_1 - \frac{3}{4}hk_2 + \frac{3}{2}hk_3)$$

$$y_{n+1} = y_n + \frac{h}{3}\left[\frac{(k_1^2 + k_2^2)}{(k_1 + k_2)} + \frac{(k_2^2 + k_3^2)}{(k_2 + k_3)} + \frac{(k_3^2 + k_4^2)}{(k_3 + k_4)}\right] \qquad (4.2.19)$$

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_0M) method.

a) Truncation Error For The Second Order Contraharmonic Mean

The second order Contraharmonic mean (C_0M) formula is written as

$$y_{n+1} = y_n + h\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) \tag{4.2.20}$$

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

$$y_{n+1} = y_n + h\left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right) \tag{4.2.21}$$

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

$$f_{n+1} = y'_{n+1}$$

$$= y'_n + hy''_n + \frac{1}{2}h^2y'''_n + \frac{1}{6}h^3y_n^{(iv)} + \quad \ldots \tag{4.2.22}$$

or $\quad (f_{n+1})^2 = (y'_{n+1})^2$

$$= (y'_n)^2 + 2hy'_ny''_n + h^2\left[(y''_n)^2 + y'_ny'''_n\right] + \quad \ldots \tag{4.2.23}$$

in the formula (4.2.21) above , we have

$$y_{n+1} = y_n + h\left(\frac{(y'_n)^2 + (y'_{n+1})^2}{y'_n + y'_{n+1}}\right) \tag{4.2.24}$$

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

$$\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} = \frac{2(y'_n)^2 + 2hy'_ny''_n + h^2\left[(y''_n)^2 + y'_ny'''_n\right] + \quad \ldots}{2y'_n + hy''_n + \frac{1}{2}h^2y''' + \quad \ldots} \tag{4.2.25}$$

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

$$\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}} = y_n' + \frac{1}{2}hy_n'' + \frac{h^2}{2y_n'}\left[\frac{1}{2}\left(y_n''\right)^2 + \frac{1}{2}y_n'y_n'''\right] + \quad \ldots \qquad (4.2.26)$$

Thus equation (4.2.21) becomes

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

$$= y_n + hy_n' + \frac{h^2}{2}y_n'' + h^3\left(\frac{y_n'''}{4} + \frac{\left(y_n''\right)^2}{4y_n'}\right) + \quad \ldots \qquad (4.2.27)$$

while the Taylor series expansion of $y(x_{n+1})$ has the form,

$$y(x_{n+1}) = y_n + hy_n' + \frac{h^2}{2}y_n'' + \frac{h^3}{6}y_n''' + \frac{h^4}{24}y^{(iv)} + \quad \ldots \qquad (4.2.28)$$

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

$$LTE = y(x_{n+1}) - y_{n+1}$$

$$= h^3\left(-\frac{y_n'''}{12} - \frac{\left(y_n''\right)^2}{4y_n'}\right). \qquad (4.2.29)$$

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

$$y_{n+1} = y_n + \frac{h}{2}\left[\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) + \left(\frac{k_2^2 + k_3^2}{k_2 + k_3}\right)\right] \qquad (4.2.30)$$

where,

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

$$k_2 = f(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_1) \qquad (4.2.31)$$

$$k_3 = f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_2\right)$$

The formula from equation (4.2.30) can be written as in equation (4.2.13), where

$$UPPER = \frac{h}{2}\left[\left(k_1^2 + k_2^2\right)\left(k_2 + k_3\right) + \left(k_2^2 + k_3^2\right)\left(k_1 + k_2\right)\right] \qquad (4.2.32)$$

and $LOWER = \left(k_1 + k_2\right)\left(k_2 + k_3\right)$ (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 $k_i, 1 \le i \le 3$ in (4.2.31) and then solve by Mathematica , we immediately obtain,

$$\begin{aligned} LOWER \times ERROR &= \frac{f^3\left(13f_y^3 + 4ff_yf_{yy} + f^2 f_{yyy}\right)h^4}{54} + 0\left(h^5\right) \\ &= C_0 + C_1h + C_2h^2 + C_3h^3 + C_4h^4 + C_5h^5 + 0\left(h^6\right) \\ &= P(h) \end{aligned} \qquad (4.2.34)$$

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

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

$$\begin{aligned} LOWER &= \left(k_1 + k_2\right)\left(k_2 + k_3\right) \\ &= 4f^2 + 4f^2 f_y h + \frac{h^2}{9}\left(16f^2 f_y^2 + 12f^3 f_{yy}\right) \\ &\quad + \frac{h^3}{27}\left(8f^2 f_y^3 + 40f^3 f_y f_{yy} + 8f^4 f_{yyy}\right) \\ &\quad + \frac{h^4}{81}\left(48f^3 f_y^2 f_{yy} + 24f^4 f_{yy}^2 + 32f^4 f_y f_{yyy} + 4f^5 f_{yyyy}\right) + 0\left(h^5\right). \\ &= b_0 + b_1h + b_2h^2 + b_3h^3 + b_4h^4 + b_5h^5 + 0\left(h^6\right) \\ &= Q(h) \end{aligned} \qquad (4.2.35)$$

From (4.2.34), we get

$$\begin{aligned} ERROR &= \frac{(TAYLOR \times LOWER) - UPPER}{LOWER} \\ &= \frac{P(h)}{Q(h)} \\ &= a_0 + a_1h + a_2h^2 + a_3h^3 + a_4h^4 + a_5h^5 + 0\left(h^6\right) \end{aligned} \qquad (4.2.36)$$

Butcher [1987] assumes that the component of y_n as an approximation to the corresponding component of $y(x_{n-1}+h)$ 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 $b_0+b_1h+b_2h^2+b_3h^3+\ldots+b_mh^m$ is added or subtracted we simply add or subtract the corresponding coefficients. Multiplying series and truncating at the h^m term gives

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

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

where

$$c_i=\sum_{j=0}^{i}a_{i-j}b_j,$$

and a quotient

$$(C_0+C_1h+C_2h^2+C_3h^3+\ldots+C_mh^m)(b_0+b_1h+b_2h^2+b_3h^3+\ldots+b_mh^m)^{-1}$$
$$=a_0+a_1h+a_2h^2+a_3h^3+\ldots+a_mh^m$$

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

$$a_i=\begin{cases}\dfrac{C_o}{b_o} & i=0,\\[2ex] \dfrac{c_i-\sum\limits_{j=1}^{i}a_{i-j}b_j}{b_o} & i>0.\end{cases}$$

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

$$a_4=\frac{f^3\left(13f_y^3+4ff_yf_{yy}+f^2f_{yyy}\right)h^4}{54\times 4f^2}$$

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

$$LTE=\frac{h^4}{216}\left(13ff_y^3+4f^2f_yf_{yy}+f^3f_{yyy}\right). \qquad (4.2.37)$$

c) Truncation Error For The Fourth Order Contraharmonic Mean

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

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

$$\ldots \qquad (4.2.38)$$

and $LOWER = (k_1 + k_2)(k_2 + k_3)(k_3 + k_4)$ (4.2.39)

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

$$LOWER \times ERROR = \frac{f^4\left(-378f_y^4 - 303ff_y^2f_{yy} - 648f^2f_{yy}^2 + 4f^2f_yf_{yyy} - 8f^3f_{yyyy}\right)h^5}{2880} + 0(h^6)$$

$$= C_0 + C_1h + C_2h^2 + C_3h^3 + C_4h^4 + C_5h^5 + 0(h^6)$$

$$= P(h) \qquad (4.2.40)$$

where $C_0 = C_1 = C_2 = C_3 = C_4 = 0.$

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

$$LOWER = (k_1 + k_2)(k_2 + k_3)(k_3 + k_4)$$

$$= 8f^3 + 12f^3f_yh + \frac{1}{2}f^3\left(17f_y^2 + 8ff_{yy}\right)h^2 + \left(\frac{69}{16}f^3f_y^3 + \frac{13}{2}f^4f_yf_{yy} + f^5f_{yyy}\right)h^3$$

$$+\frac{f^3}{384}\left(567f_y^4 + 2370ff_y^2f_{yy} + 420f^2f_{yy}^2 + 728f^2f_yf_{yyy} + 80f^3f_{yyyy}\right)h^4$$

$$+\frac{f^3}{7680}\left(2295f_y^5 + 27495ff_y^3f_{yy} + 19290f^2f_yf_{yy}^2 + 17160f^2f_y^2f_{yyy} + 4360f^3f_{yy}f_{yyy}\right.$$

$$\left.+3560f^3f_yf_{yyyy} + 2888f^4f_{yyyyy}\right)h^5 + 0(h^6).$$

$$= b_0 + b_1h + b_2h^2 + b_3h^3 + b_4h^4 + b_5h^5 + 0(h^6)$$

$$= Q(h) \qquad (4.2.41)$$

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 ,

$$a_i = \frac{c_i - \sum_{j=1}^{i} a_{i-j} b_j}{b_0} \quad , \quad i = 1,2,3,\ldots\ldots,r$$

where $a_1 = a_2 = a_3 = a_4 = 0$ and

$$a_5 = \frac{f^4\left(-378 f_y^4 - 303 f f_y^2 f_{yy} - 648 f^2 f_{yy}^2 + 4 f^2 f_y f_{yyy} - 8 f^3 f_{yyyy}\right) h^5}{2880 \times 8 f^3}$$

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

$$LTE = \frac{h^5}{23040}\left[-378 f f_y^4 - 303 f^2 f_y^2 f_{yy} - 648 f^3 f_{yy}^2 + 4 f^3 f_y f_{yyy} - 8 f^4 f_{yyyy}\right] + 0\left(h^6\right)$$

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

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

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

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

$$\therefore \quad \frac{y_{n+1}}{y_n} = \frac{LOWER + UPPER}{LOWER} \qquad (4.2.42)$$

where,

$$UPPER = 8z + 16z^2 + \frac{95}{6}z^3 + \frac{523}{48}z^4 + \frac{2207}{384}z^5 + \frac{593}{256}z^6 + 0\left(z^7\right) \qquad (4.2.43)$$

and

$$LOWER = 8 + 12z + \frac{17}{2}z^2 + \frac{69}{16}z^3 + \frac{189}{128}z^4 + \frac{153}{512}z^5 + 0\left(z^6\right)$$

$$= b_0 + b_1 z + b_2 z^2 + b_3 z^3 + b_4 z^4 + b_5 z^4 + 0\left(z^6\right) \qquad (4.2.44)$$

$$LOWER + UPPER = 8 + 20z + \frac{49}{2}z^2 + \frac{967}{48}z^3 + \frac{4751}{384}z^4 + \frac{9287}{1536}z^5 + 0\left(z^6\right).$$

$$= c_0 + c_1 z + c_2 z^2 + c_3 z^3 + c_4 z^4 + c_5 z^5 + 0\left(z^6\right) \qquad (4.2.45)$$

We substitute equations (4.2.43)-(4.2.45) into (4.2.42) and using Butcher's method, we get

$$\frac{y_{n+1}}{y_n} = \frac{8 + 20z + \frac{49}{2}z^2 + \frac{967}{48}z^3 + \frac{4751}{384}z^4 + \frac{9287}{1536}z^5 + 0(z^6)}{8 + 12z + \frac{17}{2}z^2 + \frac{69}{16}z^3 + \frac{189}{128}z^4 + \frac{153}{512}z^5 + 0(z^6)}$$

$$Q(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3 + a_4 z^4 + a_5 z^5 + 0(z^6) \qquad (4.2.46)$$

with , $a_0 = \frac{c_0}{b_0}$

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

$$a_0 = \frac{c_0}{b_0} = 1$$

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

$$a_2 = \frac{c_2 - a_1 b_1 - a_0 b_2}{b_0} = \frac{\frac{49}{2} - 1(12) - 1\left(\frac{17}{2}\right)}{8}$$

$$= \frac{1}{2}$$

$$a_3 = \frac{c_3 - a_2 b_1 - a_1 b_2 - a_0 b_3}{b_0}$$

$$= \frac{\left(\frac{967}{48}\right) - \left(\frac{1}{2}\right)(12) - 1.\left(\frac{17}{2}\right) - 1.\left(\frac{69}{16}\right)}{8}$$

$$= \frac{1}{6}$$

$$a_4 = \frac{c_4 - a_3 b_1 - a_2 b_2 - a_1 b_3 - a_0 b_4}{b_0}$$

$$= \frac{\left(\frac{4751}{384}\right) - \left(\frac{1}{6}\right)(12) - \left(\frac{1}{2}\right)\left(\frac{17}{2}\right) - 1.\left(\frac{69}{16}\right) - 1.\left(\frac{189}{128}\right)}{8}$$

$$= \frac{1}{24}$$

$$a_5 = \frac{c_5 - a_4b_1 - a_3b_2 - a_2b_3 - a_1b_4 - a_0b_5}{b_0}$$

$$= \frac{\left(\frac{9287}{1536}\right) - \left(\frac{1}{24}\right)(12) - \left(\frac{1}{6}\right)\left(\frac{17}{2}\right) - \left(\frac{1}{2}\right)\left(\frac{69}{16}\right) - 1.\left(\frac{189}{128}\right) - 1.\left(\frac{153}{512}\right)}{8}$$

$$= \frac{19}{768}$$

When we substitute $a_0 = 1, a_1 = 1, a_2 = \frac{1}{2}, a_3 = \frac{1}{6}, a_4 = \frac{1}{24}$ *and* $a_5 = \frac{19}{768}$ into equation (4.2.46) , we get

$$Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{19z^5}{768} + 0(z^6) \qquad (4.2.47)$$

The Taylor series expansion of exponential z is

$$e^z = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + 0(z^6) \qquad (4.2.48)$$

When we subtract equation (4.2.48) from equation (4.2.47) we obtain

$$e^z - Q(z) = \left(\frac{1}{120} - \frac{19}{768}\right)z^5 + 0(z^6)$$

$$= -\frac{21}{1280}z^5 + 0(z^6)$$

$$= C_4 z^{p+1} + 0(z^6)$$

This confirms that the contraharmonic mean formula is of order four with error constant $C_4 = -\frac{21}{1280}$. The equation

$$\frac{y_{n+1}}{y_n} = Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} - \frac{21z^5}{1280} + 0(z^6) \qquad (4.2.49)$$

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

$$y' = \lambda y \qquad y(0) = 1$$

to equation (4.2.20) we obtain

$$y_{n+1} = y_n + h\left[\frac{\lambda^2 y_n^2 + \lambda^2 y_{n+1}^2}{\lambda y_n + \lambda y_{n+1}}\right]$$

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

$$\frac{y_{n+1}}{y_n} = \sqrt{\frac{1 + h\lambda}{1 - h\lambda}} \qquad (4.2.50)$$

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,

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

or $$\left|\frac{1+h\lambda}{1-h\lambda}\right| < 1 \tag{4.2.51}$$

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

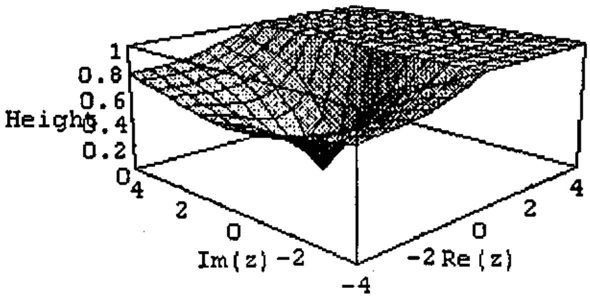


Figure 4.1: Graphic surface defined by (4.2.51)

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

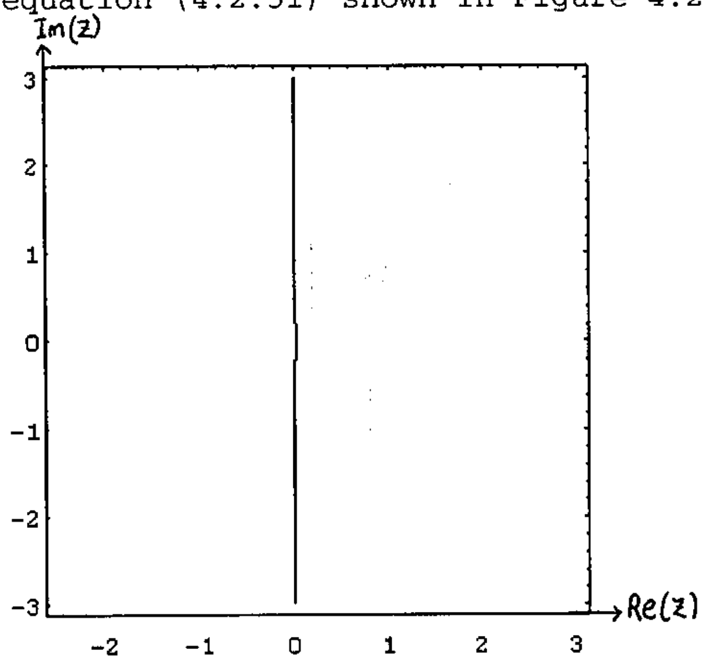


Figure 4.2: Stability region for second order C_oM

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

$$k_1 = \lambda y_n \tag{4.2.52-i}$$

$$k_2 = \lambda\left[y_n + \frac{2}{3}h\lambda y_n\right]$$

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

$$k_3 = \lambda\left[y_n + \frac{2}{3}h(\lambda y_n)\left(1 + \frac{2}{3}z\right)\right]$$

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

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

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

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

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

If we denote

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

$$= 1 + z + \frac{z^2}{2} + \frac{z^3}{6} - \frac{z^4}{54} + 0(z^5) \qquad (4.2.53)$$

we write equation (4.2.53) as the quadratic equation

$$\therefore 2Q^2 - (2+2z+z^2)Q - (\frac{z^3}{3} + \frac{8z^4}{27} + \frac{7z^5}{54} + \frac{z^6}{27} - \frac{z^7}{81} + \frac{z^8}{1458}) = 0 \qquad (4.2.54)$$

The roots of equation (4.2.54) are

$$Q = \frac{(54+54z+27z^2) \pm \left(-54-54z-27z^2-18z^3+2z^4\right)}{108} \qquad (4.2.55)$$

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,

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

The graphic surface for the third order formula is as shown in Figure 4.3.

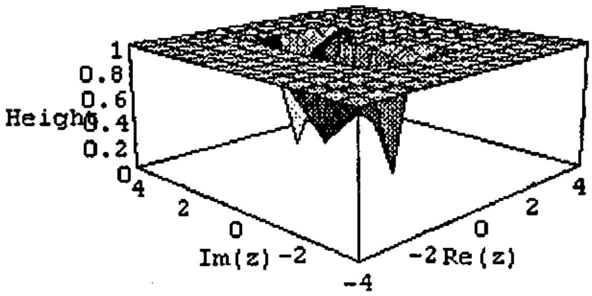


Figure 4.3: Graphic surface define by (4.2.56)

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

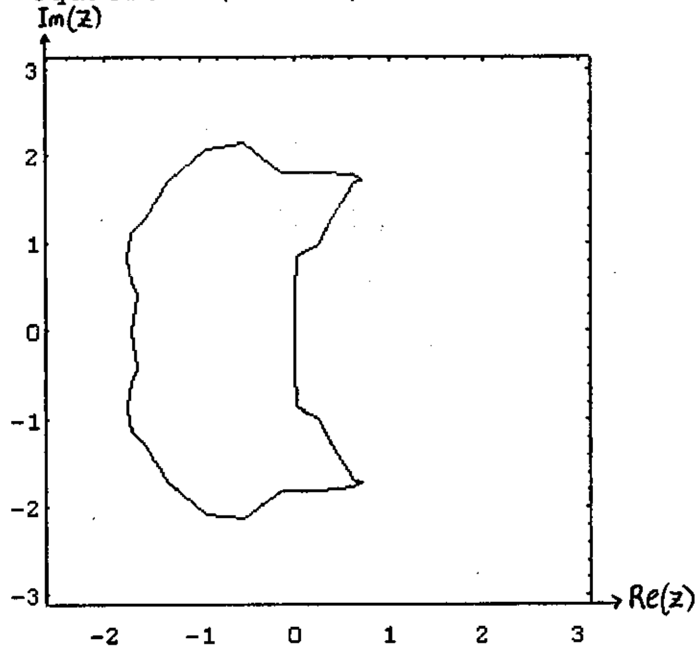


Figure 4.4: Stability region for third order C_oM

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

$$k_1 = \lambda y_n \quad (4.2.57\text{-i})$$

$$k_2 = \lambda\left[y_n + ha_1k_1\right]$$

$$= \lambda y_n\left[1 + za_1\right] \quad (4.2.57\text{-ii})$$

$$k_3 = \lambda\left[y_n + ha_2k_1 + ha_3k_2\right]$$

$$= \lambda y_n\left[1 + z\left(a_2 + a_3\right) + z^2a_1a_3\right] \quad (4.2.57\text{-iii})$$

and

$$k_4 = \lambda\left[y_n + ha_4k_1 + ha_5k_2 + ha_6k_3\right]$$

$$= \lambda\left[y_n + ha_4(\lambda y_n) + ha_5\lambda y_n(1 + za_1) + ha_6\lambda y_n\left(1 + z(a_2 + a_3) + z^2a_1a_3\right)\right]$$

$$= \lambda y_n\left[1 + z(a_4 + a_5 + a_6) + z^2a_1a_5 + z^2a_6(a_2 + a_3) + z^3a_1a_3a_6\right]$$

... (4.2.57-iv)

If we substitute (4.2.57-i) - (4.2.57-iv) and $a_i, i = 1(6)1$, i.e.

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

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

$$y_{n+1} = y_n + \frac{h}{3}\left[\frac{\left(\lambda^2y_n^2 + \lambda^2y_n^2\left(1+\frac{z}{2}\right)^2\right)}{\left(\lambda y_n + \lambda y_n\left(1+\frac{z}{2}\right)\right)} + \frac{\left(\lambda^2y_n^2\left(1+\frac{z}{2}\right)^2 + \lambda^2y_n^2\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2\right)}{\left(\lambda y_n\left(1+\frac{z}{2}\right) + \lambda y_n\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)\right)}\right.$$

$$\left.+\frac{\left(\lambda^2y_n^2\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2 + \lambda^2y_n^2\left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)^2\right)}{\left(\lambda y_n\left(1+\frac{z}{2}+\frac{3}{16}z^2\right) + \lambda y_n\left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)\right)}\right]$$

$$= y_n + \frac{(h\lambda)}{3}\left[\frac{\left(1+\left(1+\frac{z}{2}\right)^2\right)}{\left(1+\left(1+\frac{z}{2}\right)\right)} + \frac{\left(\left(1+\frac{z}{2}\right)^2 + \left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2\right)}{\left(\left(1+\frac{z}{2}\right) + \left(1+\frac{z}{2}+\frac{3}{16}z^2\right)\right)}\right.$$

$$\left.+\frac{\left(\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2 + \left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)^2\right)}{\left(\left(1+\frac{z}{2}+\frac{3}{16}z^2\right) + \left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)\right)}\right] \quad (4.2.58)$$

By substituting $h\lambda = z$ in equation (4.2.58), we can show that

$$y_{n+1} = y_n + y_n\left[\frac{\left(z + z\left(1+\frac{z}{2}\right)^2\right)}{\left(6+\frac{3}{2}z\right)} + \frac{\left(z\left(1+\frac{z}{2}\right)^2 + z\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2\right)}{\left(6+3z+\frac{9}{16}z^2\right)}\right.$$

$$\left.+\frac{z\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2 + z\left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)^2}{\left(6+\frac{9}{2}z+\frac{27}{16}z^2+\frac{27}{32}z^3\right)}\right] \qquad (4.2.59)$$

Dividing both sides of equation (4.2.59) by y_n we obtain

$$\frac{y_{n+1}}{y_n} = 1 + \left[\frac{\left(z + z\left(1+\frac{z}{2}\right)^2\right)}{\left(6+\frac{3}{2}z\right)} + \frac{\left(z\left(1+\frac{z}{2}\right)^2 + z\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2\right)}{\left(6+3z+\frac{9}{16}z^2\right)}\right.$$

$$\left.+\frac{z\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2 + z\left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)^2}{\left(6+\frac{9}{2}z+\frac{27}{16}z^2+\frac{27}{32}z^3\right)}\right] \qquad (4.2.60)$$

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

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

$$\left.+\frac{z\left(1+\frac{z}{2}+\frac{3}{16}z^2\right)^2 + z\left(1+z+\frac{3}{8}z^2+\frac{9}{32}z^3\right)^2}{\left(6+\frac{9}{2}z+\frac{27}{16}z^2+\frac{27}{32}z^3\right)}\right]$$

$$\therefore Q = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{19z^5}{768} + 0(z^6). \qquad (4.2.61)$$

Then we write equation (4.2.61) as the quadratic equation

$$2Q^2 - (2+2z+z^2)Q - (\frac{z^3}{3} + \frac{5z^4}{12} + \frac{115z^5}{384} + \frac{169z^6}{1152} + \frac{121z^7}{2304} + \frac{23z^8}{1152} + \frac{19z^9}{4608} + \frac{361z^{10}}{294912}) = 0. \qquad \ldots (4.2.62)$$

The roots of equation (4.2.62) are

$$Q = \frac{(384+384z+192z^2) \pm (384+384z+192z^2+128z^3+32z^4+19z^5)}{768} \qquad \ldots (4.2.63)$$

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

$$\left|\frac{(384+384z+192z^2) + (384+384z+192z^2+128z^3+32z^4+19z^5)}{768}\right| < 1 \qquad \ldots (4.2.64)$$

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

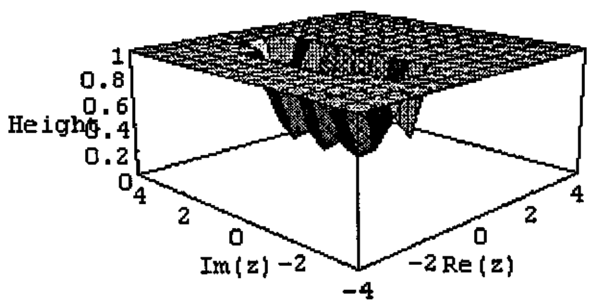


Figure 4.5: Graphic surface defined by equation (4.2.64).

and we can also plot the stability region defined by the formula (4.2.64) as shown in Figure 4.6.

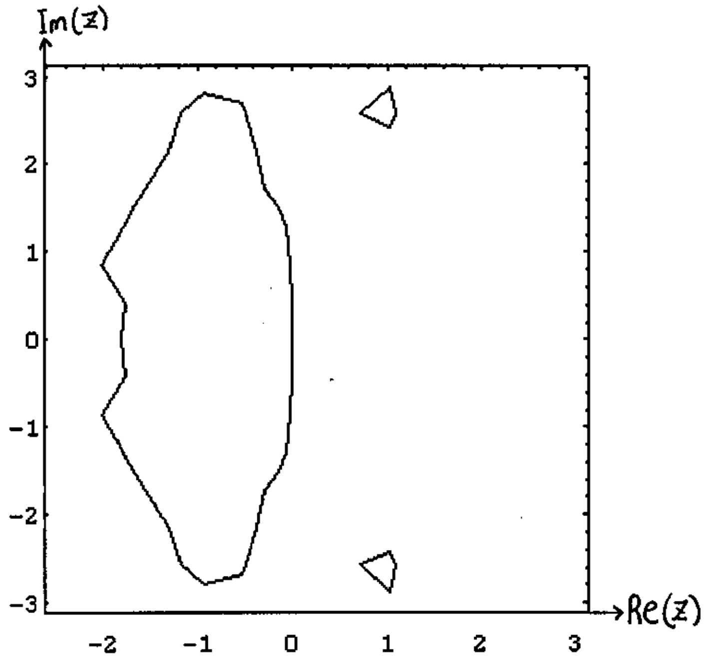


Figure 4.6: Stability region for fourth order C_oM

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_oM) method are A-stable for all orders.

4.2.6 Numerical Example

We consider the initial value problem,

$$y' = y \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 1 \qquad (4.2.65)$$

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_aM) , 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)

x	Exact Solution	Error (C_0M)
.10	.110517E+01	.152133E-06
.20	.122140E+01	.336265E-06
.30	.134986E+01	.557446E-06
.40	.149182E+01	.821431E-06
.50	.164872E+01	.113478E-05
.60	.182212E+01	.150495E-05
.70	.201375E+01	.194043E-05
.80	.222554E+01	.245086E-05
.90	.245960E+01	.304720E-05
1.00	.271828E+01	.374186E-05

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

x	Error (C_0M)	Error (H_0M)	Error (GM)	Error (AM)
.10	.152133E-06	.311267E-06	.190811E-06	.847423E-07
.20	.336265E-06	.688006E-06	.421759E-06	.187309E-06
.30	.557446E-06	.114055E-05	.699173E-06	.310514E-06
.40	.821431E-06	.168066E-05	.103027E-05	.457561E-06
.50	.113478E-05	.232178E-05	.142329E-05	.632103E-06
.60	.150495E-05	.307915E-05	.188757E-05	.838299E-06
.70	.194043E-05	.397015E-05	.243377E-05	.108087E-05
.80	.245086E-05	.501451E-05	.307398E-05	.136520E-05
.90	.304720E-05	.623463E-05	.382193E-05	.169738E-05
1.00	.374186E-05	.765592E-05	.469320E-05	.208432E-05

Table 4.3: No of arithmetic operations for using various formulas

Method	Square root	Division	Multiply	Additions	Total
AM	0	0	1	5	6
GM	3	0	4	2	9
HaM	0	3	4	5	12
CoM	0	3	7	8	18

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

$$AM < C_oM < GM < H_aM$$

whilst the amount of work involved by the H_aM method is twice that of the AM method while the GM and C_oM 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

$$y_{n+1} = y_n + \frac{h}{2}\left[\sum_{i=1}^{2}\frac{2\left(k_i^2 + k_i k_{i+1} + k_{i+1}^2\right)}{3\left(k_i + k_{i+1}\right)}\right] \qquad (4.3.1)$$

by using the related Centroidal Mean

$$C_eM = \frac{4(AM)^2 - (GM)^2}{3(AM)} \qquad (4.3.2)$$

Normally, we proceed to adjust the parameters $a_i, 1 \le i \le 3$ 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.,

$h^2 ff_y \quad : \quad 2a_1 + a_2 + a_3 = 2$ (4.3.3-i)

$h^3 f^2 f_{yy} \quad : \quad 3(a_2 + a_3)^2 + 6a_1^2 = 4$ (4.3.3-ii)

$h^3 ff_y^2 \quad : \quad 7a_1^2 + 8a_1a_3 + 5a_1a_2 + 2(a_2 + a_3)^2 = 8$ (4.3.3-iii)

We immediately obtain the parameters $a_i, 1 \le i \le 3$ in (4.3.3-i)-(4.3.3-iii) by the use of Mathematica. We find that

$$a_1 = \frac{2}{3} \quad , \quad a_2 = -\frac{2}{9} \quad , \quad a_3 = \frac{8}{9}$$

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,

$$y_{n+1} = y_n + \frac{h}{3}\left[\frac{2(k_1^2 + k_1k_2 + k_2^2)}{3(k_1 + k_2)} + \frac{2(k_2^2 + k_2k_3 + k_3^2)}{3(k_2 + k_3)} + \frac{2(k_3^2 + k_3k_4 + k_4^2)}{3(k_3 + k_4)}\right]$$

... (4.3.4)

Normally, we would substitute equations (4.2.10) into (4.3.4) to obtain an expression of y_{n+1}. However, the fraction involved in (4.3.4) is not amenable to direct substitution, i.e.,

$$\frac{k_i^2 + k_ik_{i+1} + k_{i+1}^2}{k_i + k_{i+1}} \quad , \qquad 1 \le i \le 3 \qquad (4.3.5)$$

As shown in equation (4.2.11), the formula from (4.3.4) may be written similar to equation (4.2.13) with

$$UPPER = \frac{2h}{9}\left[(k_1^2 + k_1k_2 + k_2^2)(k_2 + k_3)(k_3 + k_4) + (k_2^2 + k_2k_3 + k_3^2)(k_1 + k_2)(k_3 + k_4)\right.$$
$$\left. + (k_3^2 + k_3k_4 + k_4^2)(k_1 + k_2)(k_2 + k_3)\right] \qquad (4.3.6)$$

and equation (4.2.15). While the Taylor series expansion of $y(x_{n+1})$ 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

$$h^2 ff_y \quad : \quad -2a_1 - 2s_2 - s_3 = 0 \qquad (4.3.7\text{-i})$$

$$h^3 f^2 f_{yy} \quad : \quad -2a_1^2 - 2s_2^2 - s_3^2 + 2 = 0 \qquad (4.3.7\text{-ii})$$

$$h^3 ff_y^2 \quad : \quad -12a_1a_3 - 6a_1a_5 - 22a_1s_2 - 6a_6s_2 - 12a_1s_3 - 10s_2s_3$$
$$-14a_1^2 + 18a_1 - 14s_2^2 + 18s_2 + 9s_3 - 4s_3^2 + 6 = 0 \qquad (4.3.7\text{-iii})$$

$$h^4 f^3 f_{yyy} \quad : \quad -4a_1^3 - 4s_2^3 - 2s_3^3 + 3 = 0 \qquad (4.3.7\text{-iv})$$

$$h^4 ff_y^3 \quad : \quad -12a_1a_3a_6 - 56a_1a_3s_2 - 20a_1a_3s_3 - 24a_1a_6s_2 - 20a_1a_5s_2$$
$$-16a_1a_5s_3 - 21a_1s_2s_3 - 16a_6s_2s_3 - 44a_1^2a_3 - 24a_1^2a_5$$
$$-20a_6s_2^2 - 25a_1s_2^2 - 25a_1^2s_2 - 17a_1^2s_3 - 8s_2^2s_3 - 8a_1s_3^2$$
$$+18a_1s_3 + 9s_2s_3 + 36a_1a_3 + 18a_1a_5 + 18a_6s_2 + 27a_1s_2$$
$$+9s_2^2 + 9a_1^2 + 12a_1 + 12s_2 - 4s_2s_3^2 + 6s_3 - 8a_1^3 - 8s_2^3 + 3 = 0\,,$$
$$\ldots \qquad (4.3.7\text{-v})$$

$$h^4 f^2 f_y f_{yy} \quad : \quad -24a_1a_3s_2 - 12a_1a_5s_3 - 12a_6s_2s_3 - 12a_1^2a_3 - 6a_1^2a_5$$
$$-6a_6s_2^2 - 22a_1s_2^2 - 22a_1^2s_2 - 12a_1s_3^2 - 12a_1^2s_3 - 10s_2s_3^2$$
$$-10s_2^2s_3 - 28s_2^3 + 18s_2^2 - 28a_1^3 - 8s_3^3 + 18a_1^2 + 12s_2 + 12a_1$$
$$+6s_3 + 9s_3^2 + 12 = 0 \quad \ldots \qquad (4.3.7\text{-vi})$$

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

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

Thus , this new method can be written as follows

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

$$k_2 = f(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1)$$

$$k_3 = f(x_n + \frac{h}{2}, y_n + \frac{1}{24}hk_1 + \frac{11}{24}hk_2) \qquad (4.3.8)$$

$$k_4 = f(x_n + h, y_n + \frac{1}{12}hk_1 - \frac{25}{132}hk_2 + \frac{73}{66}hk_3)$$

$$y_{n+1} = y_n + \frac{2h}{9}\left[\frac{\left(k_1^2 + k_1k_2 + k_2^2\right)}{\left(k_1 + k_2\right)} + \frac{\left(k_2^2 + k_2k_3 + k_3^2\right)}{\left(k_2 + k_3\right)} + \frac{\left(k_3^2 + k_3k_4 + k_4^2\right)}{\left(k_3 + k_4\right)}\right] \quad \ldots \quad (4.3.9)$$

4.3.3 Error Analysis

By substituting the values $a_i, 1 \le i \le 6$ into equation (4.3.8)-(4.3.9) using Mathematica and evaluating all the terms up to (h^5) to represent the local truncation error for this method, we have the result

$$LTE = \frac{h^5}{207360}\left[248ff_y^4 - 2727f^2f_y^2f_{yy} + 72f^3f_{yy}^2 - 84f^3f_yf_{yyy} - 72f^4f_{yyyy}\right] \quad \ldots \quad (4.3.10)$$

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 $y' = \lambda y_n$ and we obtain

$$k_1 = \lambda y_n$$

$$k_2 = \lambda\left(y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = \lambda\left(y_n + \frac{1}{24}hk_1 + \frac{11}{24}hk_2\right)$$

$$k_4 = \lambda(y_n + \frac{1}{12}hk_1 - \frac{25}{132}hk_2 + \frac{73}{66}hk_3) \quad (4.3.11)$$

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

$$y_{n+1} = y_n + (h\lambda)y_n + \frac{1}{2}(h\lambda)^2y_n + \frac{1}{6}(h\lambda)^3y_n + \frac{1}{24}(h\lambda)^4y_n + \frac{37}{5184}(h\lambda)^5y_n \quad \ldots \quad (4.3.12)$$

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

$$y_{n+1} = y_n + y_n\left[z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{5184}\right] + 0(z^6) \quad . \quad (4.3.13)$$

We obtain a one step difference equation of the form

$$\frac{y_{n+1}}{y_n} = Q$$

and from equation (4.3.13), obtain

$$Q = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 + \frac{37}{5184}z^5 + 0(z^6) \quad . \qquad (4.3.14)$$

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.,

$$\left|1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{37z^5}{5184}\right| < 1 \quad . \qquad (4.3.15)$$

By the use of Mathematica, we can plot the graphic surface defined by equation (4.3.15) i.e.,

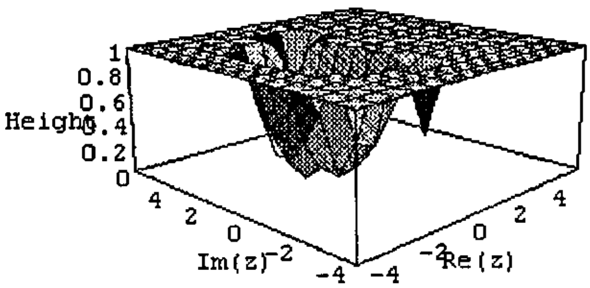


Figure 4.7: Graphic surface defined by fourth order $C_e M$ method

and we can also plot the stability region defined by the formula in equation (4.3.15) as shown in Figure 4.8.

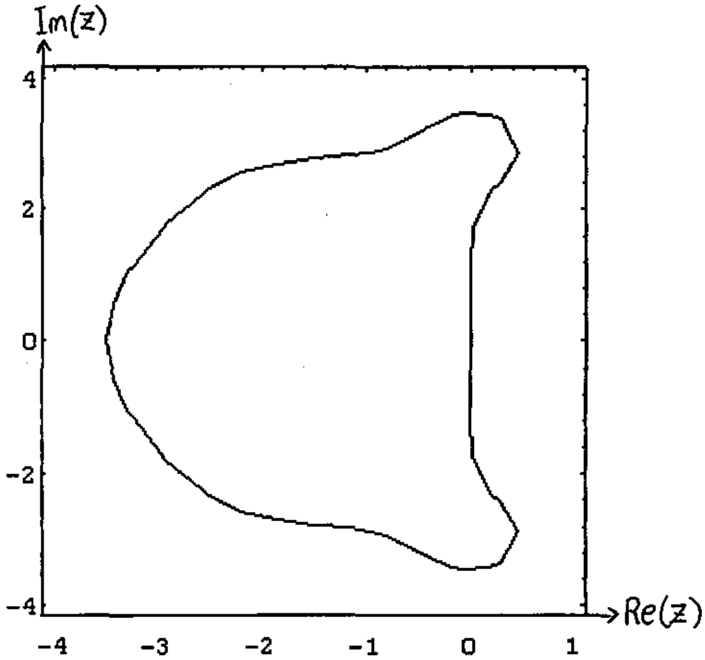


Figure 4.8:Stability region for fourth order C_eM method.

4.3.5 Numerical Example

We solve , the initial value problem

$$y' = -\sqrt{(1-y^2)},\ y(0.1)=\text{Cos}(0.1),\ 0.1 \le x \le 1.0 \qquad (4.3.16)$$

by using the steplength 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_0M), the Harmonic Mean (H_aM) 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)

x	Exact Solution	Error (C_eM)
.10	.995004E+00	.000000E+00
.20	.980067E+00	.101016E-07
.30	.955336E+00	.165567E-07
.40	.921061E+00	.222985E-07
.50	.877583E+00	.276548E-07
.60	.825336E+00	.326692E-07
.70	.764842E+00	.373258E-07
.80	.696707E+00	.415924E-07
.90	.621610E+00	.454337E-07
1.00	.540302E+00	.488152E-07

Table 4.5: Comparison of the errors by using the various order formulae for solving (4.3.16)

x	Error(C_0M)	Error(C_eM)	Error(AM)	Error (H_aM)
.10	.000000E+00	.000000E+00	.000000E+00	.000000E+00
.20	.436823E-09	.101016E-07	.138674E-07	.249999E-07
.30	.714333E-09	.165567E-07	.227342E-07	.409564E-07
.40	.976564E-09	.222985E-07	.306209E-07	.551381E-07
.50	.122684E-08	.276548E-07	.379791E-07	.683668E-07
.60	.146287E-08	.326692E-07	.448692E-07	.807537E-07
.70	.168219E-08	.373258E-07	.512690E-07	.922598E-07
.80	.188261E-08	.415924E-07	.571343E-07	.102806E-06
.90	.206222E-08	.454337E-07	.624160E-07	.112304E-06
1.00	.221936E-08	.488152E-07	.670664E-07	.120668E-06

It can be seen that the two new formula Contraharmonic Mean (C_0M) and the Centroidal Mean (C_eM) 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 $a_i, 1 \le i \le 6$ in (4.2.10) for use in the various formulas

	C_oM	C_eM	AM	GM	H_aM
$132a_1$	66	66	66	66	66
$132a_2$	16.5	5.5	0	-8.25	-16.5
$132a_3$	49.5	60.5	66	74.25	82.5
$132a_4$	33	11	0	-16.5	-33
$132a_5$	-99	-25	0	27.5	46.2
$132a_6$	198	146	132	121	118.8

In Evans and Yaakub [1995] and Wazwaz [1993] it was shown that for $k_i \ne k_{i+1}$, where

$$C_oM > RM > C_eM > AM > H_eM > GM > H_aM \quad (4.3.17)$$

From Table 4.5 it can be seen that the errors satisfy

$$C_oM < C_eM < AM < H_aM \quad (4.3.18)$$

Comparison of the parameters $a_i, 1 \le i \le 6$, 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

$$a_1 = a_2 + a_3 = \frac{1}{2}$$

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, $1 \le i \le 6$.

Table 4.7: The values of parameters a_i, $1 \le i \le 6$ based on the various formulas

	C_oM	C_eM	AM	GM	H_aM
a_1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
a_2	$\frac{1}{8}$	$\frac{1}{24}$	0	$-\frac{1}{16}$	$-\frac{1}{8}$
a_3	$\frac{3}{8}$	$\frac{11}{24}$	$\frac{1}{2}$	$\frac{9}{16}$	$\frac{5}{8}$
a_4	$\frac{1}{4}$	$\frac{11}{132}$	0	$-\frac{1}{8}$	$-\frac{1}{4}$
a_5	$-\frac{3}{4}$	$-\frac{25}{132}$	0	$\frac{5}{24}$	$\frac{7}{20}$
a_6	$\frac{3}{2}$	$\frac{73}{66}$	1	$\frac{11}{12}$	$\frac{9}{10}$

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.,

$$RMS = \sqrt{2(AM)^2 - (GM)^2} \tag{4.4.1}$$

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

$$y_{n+1} = y_n + \frac{h}{2}\left(\sqrt{\frac{k_1^2 + k_2^2}{2}} + \sqrt{\frac{k_2^2 + k_3^2}{2}}\right) \tag{4.4.2}$$

4.4.1 Third Order Root-Mean-Square (RMS) Formula

Now we attempt to find the values of the parameters a_i, $1 \le i \le 3$ 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

$$(1+x)^{\frac{1}{2}}=1+\frac{1}{2}x-\frac{1}{8}x^2+\frac{1}{16}x^3-\frac{5}{128}x^4+\ldots \tag{4.4.3}$$

First we write $\sqrt{\frac{k_1^2+k_2^2}{2}}$ in the form of

$$f(1+x)^{\frac{1}{2}} \tag{4.4.4}$$

where $$x=\frac{k_1^2+k_2^2}{2f^2}-1. \tag{4.4.5}$$

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 $\sqrt{\frac{k_2^2+k_3^2}{2}}$.

By comparing equation (4.4.2) with the Taylor series expansion for $y(x_{n+1})$ in equation (3.3.8), we obtain the following three equations of conditions, i.e.,

$h^2 ff_y \quad : \quad 2a_1+a_2+a_3=2$ (4.4.6-i)

$h^3 f^2 fyy \quad : \quad 6a_1^2+3(a_2+a_3)^2=4$ (4.4.6-ii)

$h^3 ff_y^2 \quad : \quad 6a_1^2+12a_1a_3-6a_1(a_2+a_3)+3(a_2+a_3)^2=8.$ (4.4.6-iii)

By solving the above three equations simultaneously by Mathematica, we obtain the values of $a_i, 1\le i\le 3$, i.e,

$$a_1=\frac{2}{3}\quad,\quad a_2=-\frac{1}{6}\quad,\quad a_3=\frac{5}{6}$$

Thus, the new Root-Mean-Square (RMS) method can be written as follows,

$$\begin{aligned} k_1 &= f(x_n,y_n) \\ k_2 &= f(x_n+\frac{2}{3}h, y_n+\frac{2}{3}hk_1) \\ k_3 &= f(x_n+\frac{2}{3}h, y_n-\frac{1}{6}hk_1+\frac{5}{6}hk_2) \end{aligned} \tag{4.4.7}$$

and

$$y_{n+1} = y_n + \frac{h}{2}\left(\sqrt{\frac{k_1^2 + k_2^2}{2}} + \sqrt{\frac{k_2^2 + k_3^2}{2}}\right) \qquad (4.4.8)$$

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 trunscation error (LTE) for this method, we have the result is given by,

$$LTE = \frac{h^4}{20736}\left[244 f^3 f_{yyy} + 630 f^2 f_y f_{yy} + 1317 f f_y^3\right] ,$$

to achieve third order accuracy.

We solve the initial value problem

$y' = y$, $y(0) = 1$ in $0 \le x \le 1$ 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)

x	Exact Solution	Error (AM)	Error (RMS)
0.1	.110517E+01	.425133E-05	.497404E-05
0.2	.122140E+01	.939687E-05	.109943E-04
0.3	.134986E+01	.155777E-04	.182256E-04
0.4	.149182E+01	.229546E-04	.268568E-04
0.5	.164872E+01	.317109E-04	.371017E-04
0.6	.182212E+01	.420551E-04	.492043E-04
0.7	.201375E+01	.542244E-04	.634422E-04
0.8	.222554E+01	.684881E-04	.801307E-04
0.9	.245960E+01	.851523E-04	.996276E-04
1.0	.271828E+01	.104564E-03	.122339E-03

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

$$y_{n+1} = y_n + \frac{h}{3}\left[\sqrt{\left(\frac{k_1^2 + k_2^2}{2}\right)} + \sqrt{\left(\frac{k_2^2 + k_3^2}{2}\right)} + \sqrt{\left(\frac{k_3^2 + k_4^2}{2}\right)}\right] \quad (4.4.9)$$

By writing k_2^2 in the form ,

$$k_2^2 = \left[f + ha_1 ff_y + \frac{1}{2}h^2 a_1^2 f^2 f_{yy} + \ldots\right]^2$$

$$= f^2\left[1 + ha_1 f_y + \frac{1}{2}h^2 a_1^2 ff_{yy} + \ldots\right]^2$$

we can derive ,

$$k_1^2 + k_2^2 = f^2 + f^2\left[1 + ha_1 f_y + \frac{1}{2}h^2 a_1^2 ff_{yy} + \ldots\right]^2$$

$$= 2f^2 + 2ha_1 f^2 f_y + h^2 a_1^2 f^3 f_{yy} + ..$$

$$\therefore \quad \frac{k_1^2 + k_2^2}{2} = f^2 + ha_1 f^2 f_y + \frac{1}{2}h^2 a_1^2 f^3 f_{yy} + \ldots$$

$$= f^2[1 + ha_1 f_y + \frac{1}{2}h^2 a_1^2 ff_{yy} + \ldots] \quad (4.4.10)$$

Then,

$$\sqrt{\frac{k_1^2 + k_2^2}{2}} = f\left[1 + ha_1 ff_y + \frac{1}{2}h^2 a_1^2 ff_{yy} + \ldots\right]^{\frac{1}{2}} \quad (4.4.11)$$

or

$$\sqrt{\frac{k_1^2 + k_2^2}{2}} = f(1+v)^{\frac{1}{2}} \quad (4.4.12)$$

where $$v = ha_1 f_y + \frac{1}{2}h^2 a_1^2 ff_{yy} + \ldots \quad (4.4.13)$$

From equation (4.4.12), we can show that,

$$v = \frac{k_1^2 + k_2^2}{2f^2} - 1 \quad (4.4.14)$$

By using the Binomial expansion in the form

$$(1+v)^{\frac{1}{2}} = 1+\frac{1}{2}v-\frac{1}{8}v^2+\frac{1}{16}v^3-\frac{5}{128}v^4+ \quad \ldots \qquad (4.4.15)$$

We can finally obtain,

$$f(1+v)^{\frac{1}{2}} = \sqrt{\frac{k_1^2+k_2^2}{2}} \qquad (4.4.16)$$

in terms of $f, f_y, f_{yy}, \ldots, h, h^2, h^3 \ldots etc$

We can obtain similar results for the terms $\sqrt{\frac{k_2^2+k_3^2}{2}}$ and $\sqrt{\frac{k_3^2+k_4^2}{2}}$. By substituting into equation (4.4.9) we obtain an equation of the form

$$y_{n+1} = y_n + hf + \frac{1}{6}h^2(2a_1+2s_2+s_3)ff_y + \frac{1}{12}h^3(2a_1^2+2s_2^2+s_3^2)f^2f_{yy} + \ldots \quad \ldots \qquad (4.4.17)$$

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 $a_2+a_3=s_2$ and $a_4+a_5+a_6=s_3$) , i.e.,

$$h^2ff_y: \quad 2a_1+2s_2+s_3=3 \qquad (4.4.18\text{-i})$$

$$h^3f^2f_{yy}: \quad 2a_1^2+2s_2^2+s_3^2=2 \qquad (4.4.18\text{-ii})$$

$$h^3ff_y^2: \quad 8a_1a_3+4a_1a_5+4a_6s_2+2s_2^2-2s_2s_3+s_3^2+2a_1^2-2a_1s_2=4 \quad \ldots \qquad (4.4.18\text{-iii})$$

$$h^4f^2f_yf_{yy}: \quad 4a_1^2a_3+2a_1^2a_5+8a_1a_3s_2+2a_6s_2^2+2s_2^3+4a_1a_5s_3 + 4a_6s_2s_3-s_2^2s_3-s_2s_3^2+s_3^3+2a_1^3-a_1^2s_2-a_1s_2^2=4 \qquad (4.4.18\text{-iv})$$

$$h^4f^3f_{yyy}: \quad 4a_1^3+4s_2^3+2s_3^3=3 \qquad (4.4.18\text{-v})$$

$$h^4ff_y^3: \quad 8a_1a_3a_6+8a_1a_3s_2-4a_1a_5s_2-4a_6s_2^2-2s_2^3-4a_1a_3s_3+4a_1a_5s_3 + 4a_6s_2s_3+s_2^2s_3+s_2s_3^2-s_3^3-2a_1^3-4a_1^2a_3+a_1^2s_2+a_1s_2^2=2 \quad \ldots \qquad (4.4.18\text{-vi})$$

Solving equations (4.4.18-i)-(4.4.18-vi) by Mathematica, we immediately obtain the values

$$a_1=\frac{1}{2} \quad , a_2=\frac{1}{16} \quad , a_3=\frac{7}{16} \quad , a_4=\frac{1}{8} \quad , a_5=-\frac{17}{56} \quad , a_6=\frac{33}{28}$$

Therefore the Root-Mean-Square (RMS) formula in equation (4.4.9) can now be written as

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

$$k_2 = f(x+\frac{h}{2}\,,\, y_n+\frac{1}{2}hk_1)$$

$$k_3 = f(x_n+\frac{h}{2}\,,\, y_n+\frac{1}{16}hk_1+\frac{7}{16}hk_2) \qquad (4.4.19)$$

$$k_4 = f(x_n+h\,,\, y_n+\frac{1}{8}hk_1-\frac{17}{56}hk_2+\frac{33}{28}hk_3)$$

and

$$y_{n+1} = y_n+\frac{h}{3}\left[\sqrt{\left(\frac{k_1^2+k_2^2}{2}\right)}+\sqrt{\left(\frac{k_2^2+k_3^2}{2}\right)}+\sqrt{\left(\frac{k_3^2+k_4^2}{2}\right)}\right] \qquad (4.4.20)$$

to achieve fourth order accuracy.

By substituting the values of $a_i\,,1\le i\le 6$ again in Mathematica and showing all the terms up to (h^5) to represent the local truncation error (LTE) for this method we have

$$LTE = \frac{h^5}{184320}\left[-429ff_y^4-2454f^2f_y^2f_{yy}-96f^3f_{yy}^2-48f^3f_yf_{yyy}-64f^4f_{yyyy}\right] \qquad \ldots \qquad (4.4.21)$$

4.4.3 Numerical Example

We consider the IVP,

y'=y , y(0)=1, $0\le x\le 1$ (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_oM), the Centroidal Mean (C_eM), 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)

x	Exact Solution	Error (RMS4)	Error (RMS3)
0.1	.110517E+01	.1785807E-07	.497404E-05
0.2	.122140E+01	.3947243E-07	.109943E-04
0.3	.134986E+01	.6543568E-07	.182256E-04
0.4	.149182E+01	.9642348E-07	.268568E-04
0.5	.164872E+01	.1332055E-06	.371017E-04
0.6	.182212E+01	.1766579E-06	.492043E-04
0.7	.201375E+01	.2277766E-06	.634422E-04
0.8	.222554E+01	.2876939E-06	.801307E-04
0.9	.245960E+01	.3576947E-06	.996276E-04
1.0	.271828E+01	.4392376E-06	.122339E-03

Table 4.10: Errors by using the various fourth order formulae for solving (4.4.22)

x	Error(RMS)	Error(C_eM)	Error(C_oM)	Error (AM)	Error (GM)
0.1	.17858E-07	.15965E-07	.15213E-06	.84742E-07	.19081E-06
0.2	.39472E-07	.35288E-07	.33626E-06	.18730E-06	.42175E-06
0.3	.65435E-07	.58500E-07	.55744E-06	.31051E-06	.69917E-06
0.4	.96423E-07	.86203E-07	.82143E-06	.45756E-06	.10302E-05
0.5	.13320E-06	.11908E-06	.11347E-05	.63210E-06	.14232E-05
0.6	.17665E-06	.15793E-06	.15049E-05	.83829E-06	.18875E-05
0.7	.22777E-06	.20363E-06	.19404E-05	.10808E-05	.24337E-05
0.8	.28769E-06	.25720E-06	.24508E-05	.13652E-05	.30739E-05
0.9	.35769E-06	.31978E-06	.30472E-05	.16973E-05	.38219E-05
1.0	.43923E-06	.39268E-06	.37418E-05	.20843E-05	.46932E-05

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

$$C_eM < RMS < AM < C_oM < GM$$

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 method. Butcher [1987] gives six basic reasons for taking a serious interest in implicit Runge-Kutta methods as follows

a) Higher orders of occuracy 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 possesed 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

$$y_{n+1} = y_n + h\Phi(x_n, y_n, h) \qquad (4.5.1\text{-i})$$

where $\Phi(x_n, y_n, h) = \sum_{r=1}^{R} c_r k_r$

and $$k_r = f\left(x + ha_r, y + h\sum_{s=1}^{R} b_{rs}k_s\right) \quad , r = 1,2,\ldots R \qquad (4.5.1\text{-ii})$$

$$a_r = \sum_{s=1}^{R} b_{rs} \quad , \quad r = 1,2,\ldots,R. \qquad (4.5.1\text{-iii})$$

The equations (4.5.1-i)-(4.5.1-iii) are implicit if there is at least one $b_{rs} \neq 0$ for $s \geq r$ 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

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

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

$$y_{n+1} = y_n + \frac{h}{2}[k_1 + k_2] \qquad (4.5.2)$$

where $$k_1 = f\left(x_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h, y_n + \frac{1}{4}hk_1 + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)hk_2\right)$$

$$k_2 = f\left(x_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h, y_n + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)hk_1 + \frac{1}{4}hk_2\right)$$

The fourth order implicit Runge-Kutta method in (4.5.2) can also be written in the Butcher array form as

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

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)

$$\frac{y_{n+1}}{y_n} = r_1 = \frac{1+\frac{1}{2}(h\lambda)+\frac{1}{12}(h\lambda)^2}{1-\frac{1}{2}(h\lambda)+\frac{1}{12}(h\lambda)^2} \qquad (4.5.3)$$

This is the fourth order (2,2) Pade 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 $b_{rs}=0$ for $r<s$, the resulting methods which are termed 'semi-explicit' by Butcher .(See, Lambert [1973], p.154). Semi-explicit Runge-Kutta methods have $b_{rs}=0$ for $r<s$ and when $b_{rs}=0$ for $r\le s$ 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

$$y_{n+1} = y_n + \frac{h}{6}(k_1+4k_2+k_3) \qquad (4.5.4)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n+\frac{h}{2}, y_n+\frac{1}{4}hk_1+\frac{1}{4}hk_2\right)$$

$$k_3 = f(x_n+h, y_n+hk_2)$$

or written in the Butcher array form as

0			
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	
1	0	1	
	$\frac{1}{6}$	$\frac{4}{6}$	$\frac{1}{6}$

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

$$\frac{y_{n+1}}{y_n} = r_1(h\lambda) = \frac{1 + \frac{3}{4}(h\lambda) + \frac{1}{4}(h\lambda)^2 + \frac{1}{24}(h\lambda)^3}{1 - \frac{1}{4}(h\lambda)} \qquad (4.5.5)$$

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

$$k_1 = f\left(x_n + (a_{11} + a_{12})h, y_n + ha_{11}k_1 + ha_{12}k_2\right) \qquad (4.5.6\text{-i})$$

$$k_2 = f\left(x_n + (a_{21} + a_{22})h, y_n + ha_{21}k_1 + ha_{22}k_2\right) \qquad (4.5.6\text{-ii})$$

$$y_{n+1} = y_n + h\left(\frac{k_1 + k_2}{2}\right) \qquad (4.5.6\text{-iii})$$

Expanding equation (4.5.6-i) as a Taylor series about y_n we obtain

$$k_1 = f + (ha_{11}k_1 + ha_{12}k_2)f_y + \tfrac{1}{2}(ha_{11}k_1 + ha_{12}k_2)^2 f_{yy} + \tfrac{1}{6}(ha_{11}k_1 + ha_{12}k_2)^3 f_{yyy} + \quad \ldots$$

$$= f + h(a_{11}k_1 + a_{12}k_2)f_y + \tfrac{1}{2}h^2(a_{11}k_1 + a_{12}k_2)^2 f_{yy}$$
$$+\tfrac{1}{6}h^3(a_{11}k_1 + a_{12}k_2)^3 f_{yyy} + \quad \ldots \qquad (4.5.7)$$

and similarly from equation (4.5.6-ii) we have

$$k_2 = f + h(a_{21}k_1 + a_{22}k_2)f_y + \tfrac{1}{2}h^2(a_{21}k_1 + a_{22}k_2)^2 f_{yy}$$
$$+\tfrac{1}{6}h^3(a_{21}k_1 + a_{22}k_2)^3 f_{yyy} + \quad \ldots \qquad (4.5.8)$$

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

$$k_i = A_i + hB_i + h^2C_i + h^3D_i + 0(h^4)\ , \qquad i = 1,2 \qquad (4.5.9)$$

Substituting for k_1 and k_2 by equation (4.5.9) in (4.5.7) we obtain

$$A_1 + hB_1 + h^2C_1 + h^3D_1 + \quad \ldots$$
$$= f + h\big[a_{11}(A_1 + hB_1 + h^2C_1 + h^3D_1 + \ldots) + a_{12}(A_2 + hB_2 + h^2C_2$$
$$+h^3D_2 + \ldots)\big]f_y + \tfrac{1}{2}h^2\big[a_{11}(A_1 + hB_1 + h^2C_1 + h^3D_1 + \quad \ldots)$$
$$+a_{12}(A_2 + hB_2 + h^2C_2 + h^3D_2 + \ldots)\big]^2 f_{yy} + \tfrac{1}{6}h^3\big[a_{11}(A_1 + hB_1$$
$$+h^2C_1 + h^3D_1 + \ldots) + a_{12}(A_2 + hB_2 + h^2C_2 + h^3D_2 + \ldots)\big]^3 f_{yyy}$$
$$= f + h\big[(a_{11}A_1 + a_{12}A_2) + h(a_{11}B_1 + a_{12}B_2) + h^2(a_{11}C_1 + a_{12}C_2)$$
$$+h^3(a_{11}D_1 + a_{12}D_2)\big]f_y + \tfrac{1}{2}h^2\big[(a_{11}A_1 + a_{12}A_2)^2$$
$$+2h(a_{11}A_1 + a_{12}A_2)(a_{11}B_1 + a_{12}B_2) + h^2\{(a_{11}B_1 + a_{12}B_2)^2$$
$$+2(a_{11}A_1 + a_{12}A_2)(a_{11}C_1 + a_{12}C_2)\}\big]f_{yy} + \tfrac{1}{6}h^3\big[(a_{11}A_1 + a_{12}A_2)^3$$
$$+h\{(a_{11}B_1 + a_{12}B_2)(a_{11}A_1 + a_{12}A_2)^2 + 2(a_{11}A_1 + a_{12}A_2)(a_{11}B_1$$
$$+a_{12}B_2)(a_{11}A_1 + a_{12}A_2)\}\big]f_{yyy}. \qquad (4.5.10)$$

On equating powers of h we obtain ,

Constant : $A_1 = f$

h : $B_1 = (a_{11}A_1 + a_{12}A_2)f_y$

h^2: $C_1 = (a_{11}B_1 + a_{12}B_2)f_y + \tfrac{1}{2}(a_{11}A_1 + a_{12}A_2)^2 f_{yy}$

h^3: $$D_1 = (a_{11}C_1 + a_{12}C_2)f_y + (a_{11}A_1 + a_{12}A_2)(a_{11}B_1 + a_{12}B_2)f_{yy} + \tfrac{1}{6}(a_{11}A_1 + a_{12}A_2)^3 f_{yyy}. \qquad (4.5.11)$$

Similarly, on substituting k_1 and k_2 by (4.5.9) in (4.5.8) we have

Constant : $A_2 = f$

h : $$B_2 = (a_{21}A_1 + a_{22}A_2)f_y$$

h^2: $$C_2 = (a_{21}B_1 + a_{22}B_2)f_y + \tfrac{1}{2}(a_{21}A_1 + a_{22}A_2)^2 f_{yy}$$

h^3: $$D_2 = (a_{21}C_1 + a_{22}C_2)f_y + (a_{21}A_1 + a_{22}A_2)(a_{21}B_1 + a_{22}B_2)f_{yy} + \tfrac{1}{6}(a_{21}A_1 + a_{22}A_2)^3 f_{yyy}. \qquad (4.5.12)$$

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 : $A_1 = f$

$A_2 = f$

h: $B_1 = s_1 f f_y$, where $s_1 = a_{11} + a_{12}$

$B_2 = s_2 f f_y$, where $s_2 = a_{21} + a_{22}$

h^2: $$C_1 = (a_{11}s_1 + a_{12}s_2)f f_y^2 + \tfrac{1}{2}s_1^2 f^2 f_{yy}$$

$$C_2 = (a_{21}s_1 + a_{22}s_2)f f_y^2 + \tfrac{1}{2}s_2^2 f^2 f_{yy}$$

h^3: $$D_1 = \left[a_{11}(a_{11}s_1 + a_{12}s_2) + a_{12}(a_{21}s_1 + a_{22}s_2)\right] f f_y^3 + \left[\tfrac{1}{2}a_{11}s_1^2 + \tfrac{1}{2}a_{12}s_2^2 + s_1(a_{11}s_1 + a_{12}s_2)\right] f^2 f_y f_{yy} + \tfrac{1}{6}s_2^3 f^3 f_{yyy}$$

$$D_2 = \left[a_{21}(a_{11}s_1 + a_{12}s_2) + a_{22}(a_{21}s_1 + a_{22}s_2)\right] f f_y^3 + \left[\tfrac{1}{2}a_{21}s_1^2 + \tfrac{1}{2}a_{22}s_2^2 + s_1(a_{21}s_1 + a_{22}s_2)\right] f^2 f_y f_{yy} + \tfrac{1}{6}s_2^3 f^3 f_{yyy}$$

Therefore, equation (4.5.7) gives

$$k_1 = f + h\left[s_1 f f_y\right] + h^2\left[(a_{11}s_1 + a_{12}s_2)f f_y^2 + \tfrac{1}{2}s_1^2 f^2 f_{yy}\right] + h^3\Big[\left\{a_{11}(a_{11}s_1 + a_{12}s_2) + a_{12}(a_{21}s_1 + a_{22}s_2)\right\} f f_y^3 + \left\{\tfrac{1}{2}a_{11}s_1^2 + \tfrac{1}{2}a_{12}s_2^2 + s_1(a_{11}s_1 + a_{12}s_2)\right\} f^2 f_y f_{yy} + \tfrac{1}{6}s_1^3 f^3 f_{yyy}\Big] . \qquad (4.5.13)$$

Similarly, we also have

$$k_2 = f + h\left[s_2 ff_y\right] + h^2\left[\left(a_{21}s_1 + a_{22}s_2\right)ff_y^2 + \tfrac{1}{2}s_2^2 f^2 f_{yy}\right]$$
$$+h^3\left[\left\{a_{21}\left(a_{11}s_1 + a_{12}s_2\right) + a_{22}\left(a_{21}s_1 + a_{22}s_2\right)\right\}ff_y^3\right.$$
$$+\left\{\tfrac{1}{2}a_{21}s_1^2 + \tfrac{1}{2}a_{22}s_2^2 + s_2\left(a_{21}s_1 + a_{22}s_2\right)\right\}f^2 f_y f_{yy}$$
$$\left. +\tfrac{1}{6}s_2^3 f^3 f_{yyy}\right] \quad . \qquad (4.5.14)$$

Thus,

$$h\left(\frac{k_1 + k_2}{2}\right) = hf + \frac{h^2}{2}\left(s_1 + s_2\right)ff_y + \frac{h^3}{4}\left[2\left(a_{11}s_1 + a_{12}s_2 + a_{21}s_1 + a_{22}s_2\right)ff_y^2\right.$$
$$\left.+\left(s_1^2 + s_2^2\right)f^2 f_{yy}\right] + \frac{h^4}{12}\left\{\left[6\left(a_{11}^2 s_1 + a_{11}a_{21}s_1 + a_{12}a_{21}s_1 + a_{21}a_{22}s_1\right.\right.\right.$$
$$\left.+a_{11}a_{12}s_2 + a_{12}a_{21}s_2 + a_{12}a_{22}s_2 + a_{22}^2 s_2\right]ff_y^3 + \left[9a_{11}s_1^2 + 3a_{21}s_1^2\right.$$
$$\left.+6a_{12}s_1s_2 + 6a_{21}s_1s_2 + 3a_{12}s_2^2 + 9a_{22}s_2^2\right]f^2 f_y f_{yy}$$
$$\left.+\left(s_1^3 + s_2^3\right)f^3 f_{yyy} \quad \right\} \quad + \quad 0\left(h^5\right) \quad . \qquad (4.5.15)$$

By comparing the expansion in (4.5.15) with the Taylor series expansion $y(x_{n+1})$, term by term we obtain,

$$h^2 ff_y: \qquad 1 - s_1 - s_2 = 0 \qquad (4.5.16\text{-i})$$

$$h^3 ff_y^2: \qquad 1 - 3a_{11}s_1 - 3a_{21}s_1 - 3a_{12}s_2 - 3a_{22}s_2 = 0 \qquad (4.5.16\text{-ii})$$

$$h^3 f^2 f_{yy}: \qquad 2 - 3s_1^2 - 3s_2^2 = 0 \qquad (4.5.16\text{-iii})$$

$$h^4 ff_y^3: \qquad 1 - 12a_{11}^2 s_1 - 12a_{11}a_{21}s_1 - 12a_{12}a_{21}s_1 - 12a_{21}a_{22}s_1 - 12a_{11}a_{12}s_2$$
$$-12a_{12}a_{21}s_2 - 12a_{12}a_{22}s_2 - 12a_{22}^2 s_2 = 0 \qquad (4.5.16\text{-iv})$$

$$h^4 f^2 f_y f_{yy}: \quad 2 - 9a_{11}s_1^2 - 3a_{21}s_1^2 - 6a_{12}s_1s_2 - 6a_{21}s_1s_2$$
$$-3a_{12}s_2^2 - 9a_{22}s_2^2 = 0 \qquad (4.5.16\text{-v})$$

$$h^4 f^3 f_{yyy}: \quad 1 - 2s_1^3 - 2s_2^3 = 0 \qquad (4.5.16\text{-vi})$$

When equations (4.5.16-i)-(4.5.16-vi) are solved by Mathematica, we immediately obtain the values

$$s_1 = \frac{3+\sqrt{3}}{6} \ , \quad s_2 = \frac{3-\sqrt{3}}{6} \ , \ a_{11} = \frac{1}{4} \quad , \quad a_{12} = \frac{3+2\sqrt{3}}{12}$$

$$a_{21} = \frac{3-2\sqrt{3}}{12} \quad , a_{22} = \frac{1}{4} \qquad (4.5.17)$$

Therefore the implicit arithmetic mean Runge-Kutta (IAM) formula can now be written as

$$y_{n+1} = y_n + h\left[\frac{k_1 + k_2}{2}\right] \qquad (4.5.18)$$

where $$k_1 = f\left(x_n + \left(\frac{1}{2} + \frac{\sqrt{3}}{6}\right)h, y_n + \frac{1}{4}hk_1 + \left(\frac{1}{4} + \frac{\sqrt{3}}{6}\right)hk_2\right)$$

$$k_2 = f\left(x_n + \left(\frac{1}{2} - \frac{\sqrt{3}}{6}\right)h, y_n + \left(\frac{1}{4} - \frac{\sqrt{3}}{6}\right)hk_1 + \frac{1}{4}hk_2\right)$$

to achieve fourth order accuracy.

Numerical Example

We consider the initial value problem ,

$$y' = -y \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 1 \qquad (4.5.19)$$

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)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90483743E+00	.90483742E+00	.12700828E-07	5
.2000	.81873078E+00	.81873075E+00	.22992994E-07	5
.3000	.74081825E+00	.74081822E+00	.31208545E-07	5
.4000	.67032008E+00	.67032005E+00	.37649560E-07	5
.5000	.60653070E+00	.60653066E+00	.42579110E-07	5
.6000	.54881168E+00	.54881164E+00	.46226273E-07	5
.7000	.49658535E+00	.49658530E+00	.48790219E-07	5
.8000	.44932901E+00	.44932896E+00	.50443821E-07	5
.9000	.40656971E+00	.40656966E+00	.51336861E-07	5
1.000	.36787949E+00	.36787944E+00	.51598857E-07	5

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

$$y_{n+1} = y_n + h\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) \qquad \text{(4.5.20-i)}$$

where $k_1 = f\left(x_n + (a_{11} + a_{12})h, y_n + ha_{11}k_1 + ha_{12}k_2\right)$ (4.5.20-ii

$$k_2 = f\left(x_n + (a_{21} + a_{22})h, y_n + ha_{21}k_1 + ha_{22}k_2\right) \qquad \text{(4.5.20-iii)}$$

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.,

$$\frac{k_1^2 + k_2^2}{k_1 + k_2} \qquad \text{(4.5.21)}$$

Following Evans and Yaakub [1993], the formula from (4.5.20-i) may be written as,

$$y_{n+1} = y_n + \frac{UPPER}{LOWER} \qquad \text{(4.5.22)}$$

where

$$UPPER = h\left(k_1^2 + k_2^2\right) \qquad \text{(4.5.23)}$$

and

$$LOWER = k_1 + k_2 \qquad \text{(4.5.24)}$$

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

$$(TAYLOR \times LOWER) - UPPER = LOWER \times ERROR \qquad \text{(4.5.25)}$$

By comparing coefficients of similar terms in (4.5.25) up to terms in (h^4) yields the following equations of the conditions

$h^2 ff_y$: $\quad 1 - s_1 - s_2 = 0$ (4.5.26-i)

$h^3 ff_y^2$: $\quad 2 + 3s_1 - 6a_{11}s_1 - 6a_{21}s_1 - 6s_1^2 + 3s_2$
$-6a_{12}s_2 - 6a_{22}s_2 - 6s_2^2 = 0$ (4.5.26-ii)

$h^3 f^2 f_{yy}$: $\quad 2 - 3s_1^2 - 3s_2^2 = 0$ (4.5.26-iii)

$h^4 ff_y^3$: $\quad 1 + 2s_1 + 6a_{11}s_1 - 12a_{11}^2 s_1 + 6a_{21}s_1 - 12a_{11}a_{21}s_1 - 12a_{12}a_{21}s_1$
$-12a_{21}a_{22}s_1 - 24a_{11}s_1^2 + 2s_2 + 6a_{12}s_2 - 12a_{11}a_{12}s_2 - 12a_{12}a_{21}s_2$
$+6a_{22}s_2 - 12a_{12}a_{22}s_2 - 12a_{22}^2 s_2 - 24a_{12}s_1s_2 - 24a_{21}s_1s_2 - 24a_{22}s_2^2 = 0$
... (4.5.26-iv)

$h^4 f^2 f_y f_{yy}$: $\quad 4 + 2s_1 + 3s_1^2 - 18a_{11}s_1^2 - 6a_{21}s_1^2 - 12s_1^3 + 2s_2 - 12a_{12}s_1s_2$
$-12a_{21}s_1s_2 + 3s_2^2 - 6a_{12}s_2^2 - 18a_{22}s_2^2 - 12s_2^3 = 0$ (4.5.26-v)

$h^4 f^3 f_{yyy}$: $\quad 1 - 2s_1^3 - 2s_2^3 = 0$ (4.5.26-vi)

There are in effect only four coefficients, namely a_{11}, 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 $a_{ij}, 1 \le i \le 2, 1 \le j \le 2$ 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

$$a_{11}=\frac{3\mp\sqrt{3}}{12} \quad , \quad a_{12}=\frac{1\pm\sqrt{3}}{4}, \quad a_{21}=\frac{1\mp\sqrt{3}}{4} \quad , \quad a_{22}=\frac{3\pm\sqrt{3}}{12}$$

Therefore the first implicit Contraharmonic mean Runge Kutta (IC_0M) formula can now be written as

$$y_{n+1}=y_n+h\left(\frac{k_1^2+k_2^2}{k_1+k_2}\right) \tag{4.5.27}$$

where
$$\left.\begin{aligned} k_1&=f\left(x_n+\left(\tfrac{3+\sqrt{3}}{6}\right)h, y_n+\left(\tfrac{3-\sqrt{3}}{12}\right)hk_1+\left(\tfrac{1+\sqrt{3}}{4}\right)hk_2\right)\\ k_2&=f\left(x_n+\left(\tfrac{3-\sqrt{3}}{6}\right)h, y_n+\left(\tfrac{1-\sqrt{3}}{4}\right)hk_1+\left(\tfrac{3+\sqrt{3}}{12}\right)hk_2\right)\end{aligned}\right\} \tag{4.5.28}$$

to achieve third order accuracy.

Equations (4.5.27) and (4.5.28) can be written in the Butcher array form as

$$\begin{array}{c|cc} \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{12} & \frac{1+\sqrt{3}}{4} \\ \frac{3-\sqrt{3}}{6} & \frac{1-\sqrt{3}}{4} & \frac{3+\sqrt{3}}{12} \\ \hline & 1 & 1 \end{array}$$

with error term = $h^4 ff_y^3$.

By using the second values as

$$a_{11}=\frac{3+\sqrt{3}}{12} \quad , \quad a_{12}=\frac{1-\sqrt{3}}{4}, \quad a_{21}=\frac{1+\sqrt{3}}{4} \quad , \quad a_{22}=\frac{3-\sqrt{3}}{12}$$

the second implicit Contraharmonic mean Runge Kutta (IC_0M) formula can now be written as ,

$$y_{n+1}=y_n+h\left(\frac{k_1^2+k_2^2}{k_1+k_2}\right) \tag{4.5.29}$$

where
$$\left.\begin{aligned} k_1&=f\left(x_n+\left(\tfrac{3-\sqrt{3}}{6}\right)h, y_n+\left(\tfrac{3+\sqrt{3}}{12}\right)hk_1+\left(\tfrac{1-\sqrt{3}}{4}\right)hk_2\right)\\ k_2&=f\left(x_n+\left(\tfrac{3+\sqrt{3}}{6}\right)h, y_n+\left(\tfrac{1+\sqrt{3}}{4}\right)hk_1+\left(\tfrac{3-\sqrt{3}}{12}\right)hk_2\right)\end{aligned}\right\} \tag{4.5.30}$$

to achieve third order accuracy.

In Butcher array form equations (4.5.29) aand (4.5.30) can be written as

$$
\begin{array}{c|cc}
\frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{12} & \frac{1-\sqrt{3}}{4} \\
\frac{3+\sqrt{3}}{6} & \frac{1+\sqrt{3}}{4} & \frac{3-\sqrt{3}}{12} \\
\hline
 & 1 & 1
\end{array}
$$

with error term = $h^4 ff_y^3$.

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)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90483360E+00	.90483742E+00	.38208776E-05	5
.2000	.81872384E+00	.81873075E+00	.69146465E-05	5
.3000	.74080884E+00	.74081822E+00	.93850105E-05	5
.4000	.67030872E+00	.67032005E+00	.11322610E-04	5
.5000	.60651785E+00	.60653066E+00	.12806476E-04	5
.6000	.54879773E+00	.54881164E+00	.13905422E-04	5
.7000	.49657062E+00	.49658530E+00	.14679274E-04	5
.8000	.44931378E+00	.44932896E+00	.15179956E-04	5
.9000	.40655421E+00	.40656966E+00	.15452456E-04	5
1.000	.36786391E+00	.36787944E+00	.15535671E-04	5

Table 4.13:Error in the implicit contraharmonic mean formulae (4.5.29)-(4.5.30) for solving (4.5.19)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90483359E+00	.90483742E+00	.38259721E-05	4
.2000	.81872383E+00	.81873075E+00	.69244328E-05	4
.3000	.74080882E+00	.74081822E+00	.93990142E-05	4
.4000	.67030871E+00	.67032005E+00	.11340445E-04	4
.5000	.60651783E+00	.60653066E+00	.12827833E-04	4
.6000	.54879771E+00	.54881164E+00	.13930057E-04	4
.7000	.49657060E+00	.49658530E+00	.14700813E-04	5
.8000	.44931377E+00	.44932896E+00	.15198669E-04	5
.9000	.40655419E+00	.40656966E+00	.15468580E-04	5
1.000	.36786389E+00	.36787944E+00	.15549421E-04	5

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

$$y_{n+1} = y_n + h\left[\frac{2\left(k_1^2 + k_1k_2 + k_2^2\right)}{3\left(k_1 + k_2\right)}\right] \tag{4.5.31-i}$$

where $k_1 = f\left(x_n + \left(a_{11} + a_{12}\right)h, y_n + ha_{11}k_1 + ha_{12}k_2\right)$ (4.5.31-ii)

$$k_2 = f\left(x_n + \left(a_{21} + a_{22}\right)h, y_n + ha_{21}k_1 + ha_{22}k_2\right) \tag{4.5.31-iii}$$

Normally, we would substitute equations (4.5.31-ii)-(4.5.31-iii) into (4.5.31-i) to obtain an expression for y_{n+1}. However, the fraction involved in (4.5.31-i) is not amenable to direct substitution, i.e.,

$$\frac{k_1^2 + k_1k_2 + k_2^2}{k_1 + k_2} \tag{4.5.32}$$

Following Evans and Yaakub [1995], the formula from (4.5.31-i) may be written as in equation (4.5.22) where,

$$UPPER = \frac{2h}{3}\left(k_1^2 + k_1k_2 + k_2^2\right) \tag{4.5.33}$$

and $LOWER = k_1 + k_2$ (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 $\left(h^4\right)$ yields the following equations of conditions

$h^2 ff_y$: $\quad 1 - s_1 - s_2 = 0$ (4.5.35-i)

$h^3 ff_y^2$: $\quad 2 + 3s_1 - 6a_{11}s_1 - 6a_{21}s_1 - 4s_1^2 + 3s_2$

$\quad -6a_{12}s_2 - 6a_{22}s_2 - 4s_1s_2 - 4s_2^2 = 0$ (4.5.35-ii)

$h^3 f^2 f_{yy}$: $\quad 2 - 3s_1^2 - 3s_2^2 = 0$ (4.5.35-iii)

$$h^4 ff_y^3: \quad 1 + 2s_1 + 6a_{11}s_1 - 12a_{11}^2 s_1 + 6a_{21}s_1 - 12a_{11}a_{21}s_1 - 12a_{12}a_{21}s_1$$
$$-12a_{21}a_{22}s_1 - 16a_{11}s_1^2 - 8a_{21}s_1^2 + 2s_2 + 6a_{12}s_2 - 12a_{11}a_{12}s_2 - 12a_{12}a_{21}s_2$$
$$+6a_{22}s_2 - 12a_{12}a_{22}s_2 - 12a_{22}^2 s_2 - 8a_{11}s_1s_2 - 16a_{12}s_1s_2 - 16a_{21}s_1s_2$$
$$-8a_{22}s_1s_2 - 8a_{12}s_2^2 - 16a_{22}s_2^2 = 0 \qquad (4.5.35\text{-iv})$$

$$h^4 f^2 f_y f_{yy}: \quad 4 + 2s_1 + 3s_1^2 - 18a_{11}s_1^2 - 6a_{21}s_1^2 - 8s_1^3 + 2s_2 - 12a_{12}s_1s_2$$
$$-12a_{21}s_1s_2 - 4s_1^2 s_2 + 3s_2^2 - 6a_{12}s_2^2 - 18a_{22}s_2^2 - 4s_1s_2^2 - 8s_2^3 = 0$$
$$(4.5.35\text{-v})$$

$$h^4 f^3 f_{yyy}: \quad 1 - 2s_1^3 - 2s_2^3 = 0 \qquad (4.5.35\text{-vi})$$

Equations (4.5.35-i)-(4.5.35-iv) are then solved simultaneously by Mathematica to give the required parameters, i.e.,

$$a_{11} = \frac{9+2\sqrt{3}}{72}, a_{12} = \frac{27-14\sqrt{3}}{72}, a_{21} = \frac{27+14\sqrt{3}}{72}, a_{22} = \frac{9-2\sqrt{3}}{72}$$

Thus, this new implicit method can be written as follows

$$y_{n+1} = y_n + h\left[\frac{2(k_1^2 + k_1k_2 + k_2^2)}{3(k_1+k_2)}\right] \qquad (4.5.36)$$

where
$$\left.\begin{aligned} k_1 &= f\left(x_n + \left(\tfrac{3-\sqrt{3}}{6}\right)h, y_n + \left(\tfrac{9+2\sqrt{3}}{72}\right)hk_1 + \left(\tfrac{27-14\sqrt{3}}{72}\right)hk_2\right) \\ k_2 &= f\left(x_n + \left(\tfrac{3+\sqrt{3}}{6}\right)h, y_n + \left(\tfrac{27+14\sqrt{3}}{72}\right)hk_1 + \left(\tfrac{9-2\sqrt{3}}{72}\right)hk_2\right) \end{aligned}\right\} \qquad (4.5.37)$$

to achieve third order accuracy.

The formulae in equations (4.5.36)-(4.5.37) can also be written in Butcher array form as

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{9+2\sqrt{3}}{72} & \frac{27-14\sqrt{3}}{72} \\ \frac{3+\sqrt{3}}{6} & \frac{27+14\sqrt{3}}{72} & \frac{9-2\sqrt{3}}{72} \\ \hline & 1 & 1 \end{array}$$

with error terms = 0.5 $h^4 f^2 f_y f_{yy}$.

Solving equations (4.5.35-i)-(4.5.35-iii) and (4.5.35-v) by Mathematica, we immediately obtain the values

$$a_{11} = \frac{9-\sqrt{3}}{36} \quad , a_{12} = \frac{9+7\sqrt{3}}{36}, a_{21} = \frac{9-7\sqrt{3}}{36} \quad , a_{22} = \frac{9+\sqrt{3}}{36}.$$

Therefore, the second implicit third order centroidal mean Runge Kutta (IC_eM) formulae can be written as

$$y_{n+1} = y_n + h\left[\frac{2(k_1^2 + k_1k_2 + k_2^2)}{3(k_1 + k_2)}\right] \tag{4.5.38}$$

where
$$\left.\begin{aligned} k_1 &= f\left(x_n + \left(\tfrac{3+\sqrt{3}}{6}\right)h, y_n + \left(\tfrac{9-\sqrt{3}}{36}\right)hk_1 + \left(\tfrac{9+7\sqrt{3}}{36}\right)hk_2\right) \\ k_2 &= f\left(x_n + \left(\tfrac{3-\sqrt{3}}{6}\right)h, y_n + \left(\tfrac{9-7\sqrt{3}}{36}\right)hk_1 + \left(\tfrac{9+\sqrt{3}}{36}\right)hk_2\right) \end{aligned}\right\} \tag{4.5.39}$$

or in Butcher array form as

$$\begin{array}{c|cc} \frac{3+\sqrt{3}}{6} & \frac{9-\sqrt{3}}{36} & \frac{9+7\sqrt{3}}{36} \\ \frac{3-\sqrt{3}}{6} & \frac{9-7\sqrt{3}}{36} & \frac{9+\sqrt{3}}{36} \\ \hline & 1 & 1 \end{array}$$

with error terms = 0.3333 $h^4 ff_y^3$.

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)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90483736E+00	.90483742E+00	.59515458E-07	4
.2000	.81873065E+00	.81873075E+00	.10747152E-06	4
.3000	.74081807E+00	.74081822E+00	.14582395E-06	4
.4000	.67031987E+00	.67032005E+00	.17596571E-06	4
.5000	.60653046E+00	.60653066E+00	.19911728E-06	4
.6000	.54881142E+00	.54881164E+00	.21634109E-06	4
.7000	.49658508E+00	.49658530E+00	.22856070E-06	4
.8000	.44932873E+00	.44932896E+00	.23657754E-06	4
.9000	.40656942E+00	.40656966E+00	.24108585E-06	4
1.000	.36787920E+00	.36787944E+00	.24268578E-06	4

Table 4.15: Error in the implicit centroidal mean formulae (4.5.38)-(4.5.39) for solving (4.5.19)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90483616E+00	.90483742E+00	.12609651E-05	4
.2000	.81872847E+00	.81873075E+00	.22864610E-05	5
.3000	.74081512E+00	.74081822E+00	.31053815E-05	5
.4000	.67031630E+00	.67032005E+00	.37477587E-05	5
.5000	.60652642E+00	.60653066E+00	.42397792E-05	5
.6000	.54880703E+00	.54881164E+00	.46042443E-05	5
.7000	.49658044E+00	.49658530E+00	.48609776E-05	5
.8000	.44932394E+00	.44932896E+00	.50271845E-05	5
.9000	.40656454E+00	.40656966E+00	.51177716E-05	5
1.000	.36787430E+00	.36787944E+00	.51456282E-05	5

4.5.4 2-Stage Implicit Harmonic Mean Method

In Sanugi and Evans [1993], the fourth order explicit Harmonic mean (H_aM) method was establised in the form

$$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3} \frac{2k_i k_{i+1}}{k_i + k_{i+1}}\right] \qquad (4.5.40)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \tfrac{h}{2}, y_n + \tfrac{1}{2}hk_1\right)$$

$$k_3 = f\left(x_n + \tfrac{h}{2}, y_n + \tfrac{h}{8}k_1 + \tfrac{5}{8}hk_2\right)$$

$$k_4 = f\left(x_n + h, y_n - \tfrac{1}{4}hk_1 + \tfrac{7}{20}hk_2 + \tfrac{9}{10}hk_3\right) \qquad (4.5.41)$$

Now, we attempt to establish the implicit Harmonic mean (IH_aM) method in the form

$$y_{n+1} = y_n + h\left[\frac{2k_1k_2}{k_1 + k_2}\right] \qquad (4.5.42\text{-i})$$

where $k_1 = f\left(x_n + (a_{11} + a_{12})h, y_n + ha_{11}k_1 + ha_{12}k_2\right)$ (4.5.42-ii)

$$k_2 = f\left(x_n + (a_{21} + a_{22})h, y_n + ha_{21}k_1 + ha_{22}k_2\right) \qquad (4.5.42\text{-iii})$$

By using the same procedure as the implicit Centroidal mean (IC_eM) method we substitute equations (4.5.42-ii)-(4.5.42-iii) into equation (4.5.42-i) to obtain an expression of y_{n+1}. However, the fraction involved in (4.5.42-i) is not amenable to direct substitution, i.e.,

$$\frac{k_1 k_2}{k_1 + k_2} \tag{4.5.43}$$

The formula from (4.5.42-i) can also written as in equation (4.5.22) where

$$UPPER = 2h(k_1 k_2) \tag{4.5.44}$$

and

$$LOWER = k_1 + k_2 \tag{4.5.45}$$

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

$$h^2 ff_y: \quad 1 - s_1 - s_2 = 0 \tag{4.5.46-i}$$

$$h^3 ff_y^2: \quad 2 + 3s_1 - 6a_{11}s_1 - 6a_{21}s_1 + 3s_2 -6a_{12}s_2 - 6a_{22}s_2 - 12s_1s_2 = 0 \tag{4.5.46-ii}$$

$$h^3 f^2 f_{yy}: \quad 2 - 3s_1^2 - 3s_2^2 = 0 \tag{4.5.46-iii}$$

$$\begin{aligned} h^4 ff_y^3: \quad & 1 + 2s_1 + 6a_{11}s_1 - 12a_{11}^2 s_1 + 6a_{21}s_1 - 12a_{11}a_{21}s_1 - 12a_{12}a_{21}s_1 \\ & -12a_{21}a_{22}s_1 - 24a_{21}s_1^2 + 2s_2 + 6a_{12}s_2 - 12a_{11}a_{12}s_2 - 12a_{12}a_{21}s_2 \\ & +6a_{22}s_2 - 12a_{12}a_{22}s_2 - 12a_{22}^2 s_2 - 24a_{11}s_1s_2 - 24a_{22}s_1s_2 - 24a_{12}s_2^2 = 0 \end{aligned} \tag{4.5.46-iv}$$

$$\begin{aligned} h^4 f^2 f_y f_{yy}: \quad & 4 + 2s_1 + 3s_1^2 - 18a_{11}s_1^2 - 6a_{21}s_1^2 + 2s_2 - 12a_{12}s_1s_2 \\ & -12s_1^2 s_2 + 3s_2^2 - 6a_{12}s_2^2 - 18a_{22}s_2^2 - 12s_1s_2^2 - 8s_2^3 = 0 \end{aligned} \tag{4.5.46-v}$$

$$h^4 f^3 f_{yyy}: \quad 1 - 2s_1^3 - 2s_2^3 = 0 \tag{4.5.46-vi}$$

Consequently equations (4.5.46-i)-(4.5.46-iv) are then solved simultaneously by Mathematica to immediately obtain the parameters, i.e.,

$$a_{11}=\frac{9-2\sqrt{3}}{24}\quad,a_{12}=\frac{3-2\sqrt{3}}{24},a_{21}=\frac{3+2\sqrt{3}}{24}\quad,a_{22}=\frac{9+2\sqrt{3}}{24}$$

Therefore, the implicit Harmonic mean (IH_aM) method can be written as

$$y_{n+1}=y_n+h\left(\frac{2k_1k_2}{k_1+k_2}\right) \qquad (4.5.47)$$

where
$$\left.\begin{aligned} k_1&=f\left(x_n+\left(\frac{3-\sqrt{3}}{6}\right)h, y_n+\left(\frac{9-2\sqrt{3}}{24}\right)hk_1+\left(\frac{3-2\sqrt{3}}{24}\right)hk_2\right)\\ k_2&=f\left(x_n+\left(\frac{3+\sqrt{3}}{6}\right)h, y_n+\left(\frac{3+2\sqrt{3}}{24}\right)hk_1+\left(\frac{9+2\sqrt{3}}{24}\right)hk_2\right)\end{aligned}\right\}$$

... (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

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{9-2\sqrt{3}}{24} & \frac{3-2\sqrt{3}}{24}\\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{24} & \frac{9+2\sqrt{3}}{24}\\ \hline & 1 & 1 \end{array}$$

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,

$$a_{11}=\frac{13\sqrt{3}}{54}\quad,a_{12}=\frac{1-\frac{4\sqrt{3}}{27}}{2},a_{21}=\frac{2-\frac{26\sqrt{3}}{27}}{2}\quad,a_{22}=\frac{2\sqrt{3}}{27}$$

Therefore, the second implicit third order Harmonic mean Runge-Kutta (IH_aM) formula can be written as

$$y_{n+1} = y_n + h\left(\frac{2k_1k_2}{k_1+k_2}\right) \qquad (4.5.49)$$

where $k_1 = f\left(x_n + \left(\frac{3+\sqrt{3}}{6}\right)h, y_n + \left(\frac{13\sqrt{3}}{54}\right)hk_1 + \left(\frac{1-\frac{4\sqrt{3}}{27}}{2}\right)hk_2\right)$

$$k_2 = f\left(x_n + \left(\frac{3-\sqrt{3}}{6}\right)h, y_n + \left(\frac{2-\frac{26\sqrt{3}}{27}}{4}\right)hk_1 + \left(\frac{2\sqrt{3}}{27}\right)hk_2\right)$$

... (4.5.50)

or in Butcher array form as

$$\begin{array}{c|cc} \frac{3+\sqrt{3}}{6} & \frac{13}{18\sqrt{3}} & \frac{1-\frac{4}{9\sqrt{3}}}{2} \\ \frac{3-\sqrt{3}}{6} & \frac{2-\frac{26}{9\sqrt{3}}}{4} & \frac{2}{9\sqrt{3}} \\ \hline & 1 & 1 \end{array}$$

with error terms = -0.818899 $h^4 ff_y^3$

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)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90483743E+00	.90483742E+00	.12216797E-07	4
.2000	.81873077E+00	.81873075E+00	.21530529E-07	4
.3000	.74081825E+00	.74081822E+00	.27074405E-07	4
.4000	.67032008E+00	.67032005E+00	.29347467E-07	4
.5000	.60653069E+00	.60653066E+00	.28762721E-07	4
.6000	.54881166E+00	.54881164E+00	.25684771E-07	4
.7000	.49658532E+00	.49658530E+00	.20434691E-07	4
.8000	.44932898E+00	.44932896E+00	.13295030E-07	4
.9000	.40656966E+00	.40656966E+00	.45142284E-08	4
1.000	.36787944E+00	.36787944E+00	.56894454E-08	4

Table 4.17: Error in the implicit harmonic mean formulae (4.5.49)-(4.5.50) method for solving (4.5.19)

xn	Numerical Solution	Exact Solution	Absolute error	No. iteration
.1000	.90484056E+00	.90483742E+00	.31387275E-05	5
.2000	.81873643E+00	.81873075E+00	.56800203E-05	5
.3000	.74082593E+00	.74081822E+00	.77093157E-05	5
.4000	.67032935E+00	.67032005E+00	.93010457E-05	5
.5000	.60654118E+00	.60653066E+00	.10520120E-04	5
.6000	.54882306E+00	.54881164E+00	.11423056E-04	5
.7000	.49659736E+00	.49658530E+00	.12058990E-04	5
.8000	.44934143E+00	.44932896E+00	.12470570E-04	5
.9000	.40658235E+00	.40656966E+00	.12694742E-04	5
1.000	.36789220E+00	.36787944E+00	.12763457E-04	5

From the above discussion, we can see that the results from the third order methods or the error term in implicit harmonic mean (IH_aM) 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_aM) 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_aM) trapezoidal formulas a new class of modified trapezoidal formulas with L-stability is obtained.

The classical trapezoidal formula is given by

$$y_{n+1} = y_n + \frac{h}{2}[f_n + f_{n+1}] \qquad (4.6.1)$$

for the numerical integration of the initial value problem

$$y' = f(x,y), \quad y(a) = y_0 \quad . \tag{4.6.2}$$

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_oM) trapezoidal formula :

$$y_{n+1} = y_n + h\left[\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right] \tag{4.6.3}$$

was discussed by Evans and Yaakub [1995] and a Centroidal mean (C_eM) was proposed by Yaakub and Evans [1995]:

$$y_{n+1} = y_n + h\left[\frac{2\left(f_n^2 + f_n f_{n+1} + f_{n+1}^2\right)}{3\left(f_n + f_{n+1}\right)}\right] \tag{4.6.4}$$

and a root-mean square (RMS) was also proposed by Yaakub and Evans [1993]:

$$y_{n+1} = y_n + h\sqrt{\left(\frac{f_n^2 + f_{n+1}^2}{2}\right)} \quad . \tag{4.6.5}$$

In this section, a new class of trapezoidal formulas is obtained by modifying the contraharmonic mean (C_oM), the centroidal mean (C_eM) and root-mean square (RMS) methods are presented. By using a standard test equation

$$f(x,y) = \lambda y \quad , \quad \lambda \in C, \quad \mathrm{Re}(z) < 0 \tag{4.6.6}$$

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

$$y_{n+1} = y_n + hf_n \tag{4.6.7}$$

applied backwards at x_{n+1} in the negative x-direction gives

$$y_n = y_{n+1} - hf_{n+1} \tag{4.6.8}$$

Formula (4.6.8) was called the explicit backward Euler formula (see Chawla and Al-Zanaidi [1996]) and defined by

$$\hat{y}_n = y_{n+1} - hf_{n+1} \tag{4.6.9}$$

and we set

$$\hat{f}_n = f(x_n, \hat{y}_n) \quad .$$

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

$$y_{n+1} = y_n + \frac{h}{2}\left[\hat{f}_n + f_{n+1}\right] \tag{4.6.10}$$

where we can obtain the local truncation error in equation (4.6.10) since

$$\hat{y}_n = y_n - \frac{h^2}{2} y_n'' + 0(h^3)$$

and

$$\hat{f}_n = f_n - \frac{h^2}{2} y_n'' g_n + 0(h^3) \tag{4.6.11}$$

where $g_n = \dfrac{\partial f_n}{\partial y_n}$. By substituting equation (4.6.11) into (4.6.10) we obtain

$$y_{n+1} = y_n + \left(\frac{h}{2} y_n' + \frac{h}{2} y_n'\right) + \frac{h^2}{2} y_n'' + \frac{h^3}{4}\left(y_n''' - y'' g_n\right) + 0(h^4)$$

$$= y_n + hy_n' + \frac{h^2}{2} y_n'' + \frac{h^3}{12}\left(3y_n''' - 3y_n'' g_n\right) + 0(h^4) \tag{4.6.12}$$

and by Taylor series as

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + 0(h^4) \tag{4.6.13}$$

By subtracting (4.6.12) from (4.6.13), it can be shown that the local truncation error for the MAM trapezoidal formula is

$$LTE_{MAM} = \frac{h^3}{12}\left[-y'''_n + 3y''_n g_n\right] + 0(h^4)\,. \tag{4.6.14}$$

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

$$\hat{y}_n = (1-z)y_{n+1} \tag{4.6.15}$$

and

$$\hat{f}_n = \lambda(1-z)y_{n+1} \tag{4.6.16}$$

By substituting equations (4.6.15) and (4.6.16) in (4.6.10), we obtain

$$y_{n+1} = y_n + \frac{z}{2}\left[(2-z)y_{n+1}\right] \tag{4.6.17}$$

As in equation (3.3.30) then from equation (4.6.17), we obtain

$$Q(z) = \frac{2}{2-2z+z^2} \tag{4.6.18}$$

Now, we set $w = 2-2z+z^2$ and $\mathrm{Re}(z) = -x-iy$ in equation (4.6.18) to give

$$w = \left(2+2x+x^2-y^2\right) + i2xy$$

and $$|w|^2 = \left(2+2x+x^2-y^2\right)^2 + (2xy)^2\,.$$

For $x > 0$, since $|z| \geq x$, it follows that

$$|w|^2 \geq \left(2+2x+x^2\right)^2 \tag{4.6.19}$$

and from (4.6.18) we obtain

$$|Q(z)| < \frac{2}{2+2x+x^2} \qquad (4.6.20)$$

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_oM) Trapezoidal Formula

By using (4.6.9), we extend the modification to the C_oM method in (4.6.3) to obtain

$$y_{n+1} = y_n + h\left[\frac{\hat{f}_n^2 + f_{n+1}^2}{\hat{f}_n + f_{n+1}}\right] \qquad . \qquad (4.6.21)$$

The local truncation error is obtained by using (4.6.11) to give

$$\frac{\hat{f}_n^2 + f_{n+1}^2}{\hat{f}_n + f_{n+1}} = f_n + \frac{h}{2}y_n'' + \frac{h^2}{4}\left(y_n''' - y_n''g_n + \frac{(y'')^2}{f_n}\right)$$

$$+\frac{h^3}{12}\left[y_n^{iv} + 3\frac{y_n''y_n'''}{f_n} - \frac{3(y_n'')^2}{2f_n} + 3\frac{(y_n'')^2 g_n}{f_n}\right] + 0(h^4) \qquad (4.6.22)$$

By substituting (4.6.22) into (4.6.21) we obtain

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}y_n'' + \frac{h^3}{4}\left[y_n''' + \frac{(y_n'')^2}{f_n} - y_n''g_n\right] + 0(h^4) \qquad (4.6.23)$$

From (4.6.13) and (4.6.23) it can be shown that the local truncation error for the MC_oM trapezoidal formula is given by

$$LTE_{MC_oM} = \frac{h^3}{12}\left[3y_n''g_n - 3\frac{(y_n'')^2}{f_n} - y_n'''\right] \qquad . \qquad (4.6.24)$$

Theorem 2. *The MC_oM trapezoidal formula (4.6.21) is L-stable.*

Proof. By applying the MC_oM trapezoidal formula (4.6.21) to the test equation in (4.6.6),we obtain

$$y_{n+1} = y_n + z[(2-z)y_{n+1}] \tag{4.6.25}$$

By writing equation (4.6.25) in a similar form as in equation (3.3.30), we obtain

$$Q(z) = \frac{1}{1-2z+z^2} \tag{4.6.26}$$

Now, we set $w = 1-2z+z^2$ and $\mathrm{Re}(z) = -x-iy$ 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

$$|w|^2 \geq (1+2x+x^2)^2 \tag{4.6.27}$$

and from (4.6.26) we obtain

$$|Q(z)| < \frac{1}{1+2x+x^2} \tag{4.6.28}$$

From equation (4.6.28), $|Q(z)|<1$ and for $\mathrm{Re}(z)<0$ we can see that $|Q(z)| \to 0$ as $x \to \infty$. Thus, we show that our MC_oM trapezoidal formula (4.6.21) is L-stable.

4.6.1.3 Modified Centroidal Mean (MC_eM) Trapezoidal Formula

With (4.6.9), we can modify the C_eM trapezoidal formula (4.6.4) to obtain

$$y_{n+1} = y_n + h\left[\frac{2\left(\hat{f}_n^2 + \hat{f}_n f_{n+1} + f_{n+1}^2\right)}{3\left(\hat{f}_n + f_{n+1}\right)}\right] \tag{4.6.29}$$

By using Mathematica and with (4.6.11), we obtain

$$\frac{2\left(\hat{f}_n^2 + \hat{f}_n f_{n+1} + f_{n+1}^2\right)}{3\left(\hat{f}_n + f_{n+1}\right)} = f_n + \frac{h}{2}y_n'' + \frac{h^2}{12}\left[3y_n''' + \frac{\left(y_n''\right)^2}{f_n} - 3y_n''g_n\right]$$

$$+\frac{h^3}{24}\left[2f_n^2 y_n^{iv} + 2f_n y_n'' y_n''' - \left(y_n''\right)^3 + 2f_n\left(y_n''\right)^2 g_n\right] + 0\left(h^4\right)$$

$$\ldots \qquad (4.6.30)$$

By substituting (4.6.30) into (4.6.29) we obtain

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}y_n'' + \frac{h^3}{12}\left[3y_n''' + \frac{\left(y_n''\right)^2}{f_n} - 3y_n''g_n\right] + 0\left(h^4\right) \qquad (4.6.31)$$

From (4.6.13) and (4.6.31) it can be shown that the local truncation error for the MC_eM trapezoidal formula is given by

$$LTE_{MC_eM} = \frac{h^3}{12}\left[3y_n''g_n - \frac{\left(y_n''\right)^2}{f_n} - y_n'''\right]. \qquad (4.6.32)$$

Theorem 3. *The MC_eM trapezoidal formula (4.6.29) is L-stable.*

Proof. By applying the MC_eM trapezoidal formula (4.6.29) to the test equation in (4.6.6), we obtain

$$y_{n+1} = y_n + \frac{2z}{3}\left[\frac{(3-3z+z^2)}{(2-z)}y_{n+1}\right] \qquad (4.6.33)$$

Equation (4.6.33) can also be written as in equation (3.3.30) to give

$$Q(z) = \frac{6-3z}{6-9z+6z^2-2z^3} \qquad (4.6.34)$$

Now, we set $w = 6-9z+6z^2-2z^3$ and $\mathrm{Re}(z) = -x-iy$ in equation (4.6.34) to give

$$w = \left(6+9x+6x^2+2x^3-6y^2-6xy^2\right) + i(9y+12xy-y^3+6x^2y)$$

and $|w|^2 = \left(6+9x+6x^2+2x^3\right)^2 + \left(9+36x+72x^2+48x^3+12x^4\right)y^2$

$$+\left(24x+12x^2\right)y^4 + 4y^6 \quad .$$

For $x > 0$, since $|z| \geq x$, it follows that

$$|w|^2 \geq \left(6 + 9x + 6x^2 + 2x^3\right)^2 \qquad (4.6.35)$$

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 $\mathrm{Re}(z) < 0$ we can see that $|Q(z)| \to 0$ as $x \to \infty$. Thus, the MC_eM 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

$$y_{n+1} = y_n + h\sqrt{\left(\frac{\hat{f}_n^2 + f_{n+1}^2}{2}\right)} \qquad (4.6.36)$$

By using Mathematica and with (4.6.11), we obtain

$$\sqrt{\left(\frac{\hat{f}_n^2 + f_{n+1}^2}{2}\right)} = f_n + \frac{h}{2}y_n'' + \frac{h^2}{12}\left[3y_n''' + \frac{3\left(y_n''\right)^2}{2f_n} - 3y_n''g_n\right]$$

$$+\frac{h^3}{48}\left[4y_n^{iv} + 6\frac{y_n''y_n'''}{f_n} - 3\frac{\left(y_n''\right)^3}{f_n^2} + 6\frac{\left(y_n''\right)^2}{f_n}\right] + 0\left(h^4\right) \qquad (4.6.37)$$

By substituting (4.6.37) into (4.6.36) we obtain

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}y_n'' + \frac{h^3}{12}\left[3y_n''' + \frac{3\left(y_n''\right)^2}{2f_n} - 3y_n''g_n\right] + 0\left(h^4\right) \qquad (4.6.38)$$

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

$$LTE_{MRMS} = \frac{h^3}{12}\left[3y_n''g_n - \frac{3\left(y_n''\right)^2}{2f_n} - y_n'''\right] \quad . \qquad (4.6.39)$$

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

$$y_{n+1} = y_n + \frac{z}{\sqrt{2}}\left[\left(\sqrt{2 - 2z + z^2}\right) y_{n+1}\right] \tag{4.6.40}$$

As shown previously, we write equation (4.6.40) in the form

$$Q(z) = \frac{\sqrt{2}}{\sqrt{2} - z\left(2 - 2z + z^2\right)^{\frac{1}{2}}} \tag{4.6.41}$$

Now, we set $w = \sqrt{2} - z\left(2 - 2z + z^2\right)^{\frac{1}{2}}$ and $\mathrm{Re}(z) = -x - iy$ in equation (4.6.41) and by setting

$$\sinh(2r) = \frac{(2 + 2x)y}{2 + 2x + x^2} \tag{4.6.42}$$

we obtain

$$\left(2 - 2z + z^2\right)^{\frac{1}{2}} = \left[\left(2 + 2x + x^2 - y^2\right) + i(2 + 2x)y\right]^{\frac{1}{2}} \tag{4.6.43}$$

or equation (4.6.43) can also be written as

$$\begin{aligned}\left(2 - 2z + z^2\right)^{\frac{1}{2}} &= \left[\left(2 + 2x + x^2\right) + i(2 + 2x)y\right]^{\frac{1}{2}} \\ &= \left(2 + 2x + x^2\right)^{\frac{1}{2}}\left[1 + i\sinh(2r)\right]^{\frac{1}{2}} \\ &= \left(2 + 2x + x^2\right)^{\frac{1}{2}}\left[\cosh(r) + i\sinh(r)\right]\end{aligned} \tag{4.6.44}$$

From equation (4.6.41), we set

$$\begin{aligned}w &= \sqrt{2} - z\left(2 - 2z + z^2\right)^{\frac{1}{2}} \\ &= \sqrt{2} + (x + iy)\left(2 + 2x + x^2\right)^{\frac{1}{2}}\left[\cosh(r) + i\sinh(r)\right] \\ &= \left(\sqrt{2} + x\sqrt{2 + 2x + x^2}\cosh(r) - y\sqrt{2 + 2x + x^2}\sinh(r)\right) \\ &\quad + i\left(x\sqrt{2 + 2x + x^2}\sinh(r) + y\sqrt{2 + 2x + x^2}\cosh(r)\right)\end{aligned}$$

and we can show that

$$\begin{aligned}|w|^2 &= 2 + x\sqrt{2 + 2x + x^2}\left[x\sqrt{2 + 2x + x^2}\cosh(2r) + 2^{\frac{3}{2}}\sinh(r)\right] \\ &\quad + \sqrt{2}y\sqrt{2 + 2x + x^2}\left[\frac{1}{\sqrt{2}}y\sqrt{2 + 2x + x^2}\cosh(2r) - 2\sinh(r)\right]\end{aligned} \tag{4.6.45}$$

By substituting y from (4.6.42) into (4.6.45), we obtain

$$|w|^2 = 2 + x\sqrt{2+2x+x^2}\left[x\sqrt{2+2x+x^2}\cosh(2r) + 2^{\frac{3}{2}}\sinh(r)\right]$$

$$+2^{\frac{5}{2}}\frac{(2+2x+x^2)^{\frac{3}{2}}}{(2+2x)}\sinh^2(2r)\cosh(r)\left[\frac{(2+2x+x^2)^{\frac{3}{2}}}{\sqrt{2}(2+2x)}\cosh(r)\cosh(2r)-1\right]$$

... (4.6.46)

From equation (4.6.46), we can see that for $x > 0$

$$\frac{(2+2x+x^2)^{\frac{3}{2}}}{\sqrt{2}(2+2x)}\cosh(r)\cosh(2r) > 1$$

and since $|z| \geq x$, it follows that

$$|w| > 2 + x(2+2x+x^2)^{\frac{1}{2}} \qquad (4.6.47)$$

and from (4.6.41) we obtain

$$|Q(z)| < \frac{2}{2 + x(2+2x+x^2)^{\frac{1}{2}}} \qquad (4.6.48)$$

Thus from equation (4.6.48), $|Q(z)| < 1$ and for $\text{Re}(z) < 0$ 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

<u>Example 1</u>:

We consider the initial value problem

$$y' = -2y \quad , \quad y(0) = 1 \quad , \quad 0 \leq x \leq 1 \ . \qquad (4.6.49)$$

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

x	MAM	MGM	MH_aM	MC_oM	MC_eM
0.1	.44493E-04	.52598E-04	.60703E-04	.28284E-04	.39085E-04
0.2	.72858E-04	.86131E-04	.99403E-04	.46314E-04	.64002E-04
0.3	.89479E-04	.10578E-03	.12208E-03	.56879E-04	.78602E-04
0.4	.97682E-04	.11548E-03	.13327E-03	.62093E-04	.85808E-04
0.5	.99972E-04	.11819E-03	.13639E-03	.63547E-04	.87819E-04
0.6	.98223E-04	.11612E-03	.13401E-03	.62435E-04	.86282E-04
0.7	.93823E-04	.11092E-03	.12801E-03	.59638E-04	.82417E-04
0.8	.87792E-04	.10379E-03	.11978E-03	.55804E-04	.77119E-04
0.9	.80865E-04	.95599E-04	.11033E-03	.51400E-04	.71034E-04
1.0	.73565E-04	.86975E-04	.10038E-03	.46759E-04	.64621E-04

Example 2:

The second-order differential equation IVP given by

$$y'' + 101y' + 100y = 0, \quad y(0) = 1.01, \quad y'(0) = -2 \qquad (4.6.50)$$

over the range $0 \le x \le 1$.

The general solution for (4.6.50) is $y(x) = Ae^{\lambda_1 x} + Be^{\lambda_2 x}$ 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.,

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -100 & -101 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad , \quad \begin{pmatrix} y_1(0) \\ y_2(0) \end{pmatrix} = \begin{pmatrix} 1.01 \\ -2 \end{pmatrix} . \qquad (4.6.51)$$

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 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 MAM, MGM, MH_aM, MC_oM and MC_eM 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

x	MAM	MGM	MH_aM	MC_oM	MC_eM
0.1	.93712E-07	.60644E-05	.12227E-04	.12359E-04	.41858E-05
0.2	.10856E-06	.54558E-05	.11024E-04	.11192E-04	.38037E-05
0.3	.14766E-06	.48779E-05	.99074E-05	.10158E-04	.34827E-05
0.4	.17833E-06	.43606E-05	.89031E-05	.92192E-05	.31885E-05
0.5	.20184E-06	.38976E-05	.80002E-05	.83672E-05	.29186E-05
0.6	.21925E-06	.34832E-05	.71886E-05	.75938E-05	.26713E-05
0.7	.23152E-06	.31124E-05	.64589E-05	.68919E-05	.24446E-05
0.8	.23947E-06	.27806E-05	.58031E-05	.62548E-05	.22369E-05
0.9	.24382E-06	.24838E-05	.52135E-05	.56766E-05	.20466E-05
1.0	.24516E-06	.22183E-05	.46837E-05	.51517E-05	.18722E-05

From Table 4.18, we can see that the modified formulas based on the contraharmonic mean MC_oM 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 MC_eM 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

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

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 $$y_{n+1} = y_n + h\left[\frac{k_1^2 + k_2^2}{k_1 + k_2}\right] \qquad (5.1)$$

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

$$C(h) = y_n = y(x_n; h) \qquad (5.1.1)$$

which are obtained from their respective formulae and the solution is assumed to have an asymptotic expansion of the form

$$y(x;h) = y(x) + \sum_{v=0}^{k} C_v(x)h^{\beta v} + 0(h^{\beta k+v}) \qquad (5.1.2)$$

where β and $C_v(x)$ depend on the numerical formulae and β is an integer ≥ 2. The asymptotic expansion for the Euler method was developed by Gragg [1965] with the expansion prototype equation of the form

$$y' = Ay \quad , \quad y(a) = \alpha$$

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 (H_eM) and Contraharmonic mean (C_oM) 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 (C_oM) METHOD

We will now establish and develop the asymptotic expansion for the contraharmonic mean formulae for the prototype equation

$$y' = Ay \quad , \quad y(a) = \alpha$$

The second order contraharmonic mean (C_oM) method in (5.1) can be written as

$$y_{n+1} = y_n + h\left[\frac{(y_n')^2 + (y_{n-1}')^2}{y_n' + y_{n+1}'}\right] \qquad (5.2.1)$$

By putting $y_n' = \lambda y_n$ in (5.2.1), we get

$$y_{n+1} = y_n + h\lambda\left[\frac{y_n^2 + y_{n+1}^2}{y_n + y_{n+1}}\right] \qquad (5.2.2)$$

By multiplying $[y_n + y_{n+1}]$ on both sides in equation (5.2.2)

$$y_{n+1}y_n + (y_{n+1})^2 = y_n^2 + y_n y_{n+1} + h\lambda[y_n^2 + y_{n+1}^2]$$

which reduces to ,

$$(y_{n+1})^2 = y_n^2 + h\lambda[y_n^2 + y_{n+1}^2]$$

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

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

or
$$\frac{y_{n+1}}{y_n} = \sqrt{\frac{1 + h\lambda}{1 - h\lambda}}$$

by taking the positive sign we have

$$y_{n+1} = y_n\sqrt{\frac{1 + h\lambda}{1 - h\lambda}} \qquad (5.2.3)$$

From the initial condition, $y_o = \alpha$ and by successive substitution, we get

$$y_n = \alpha\left[\sqrt{\frac{1 + h\lambda}{1 - h\lambda}}\right]^n \qquad (5.2.4)$$

Since $x = a + nh$, we substitute $n = \dfrac{x - a}{h}$ in equation (5.2.4) to obtain

$$y(x;h) = \alpha\left[\sqrt{\frac{1 + h\lambda}{1 - h\lambda}}\right]^{\frac{x-a}{h}} \qquad (5.2.5)$$

Taking logarithms on both sides of (5.2.5) gives,

$$\ln[y(x;h)] = \ln\left\{\alpha\left[\sqrt{\frac{1 + h\lambda}{1 - h\lambda}}\right]^{\frac{x-a}{h}}\right\}$$

$$y(x;h) = \alpha \exp\left\{\left(\frac{x-a}{h}\right)\ln\left[\sqrt{\frac{1+h\lambda}{1-h\lambda}}\right]\right\} \qquad (5.2.6)$$

and by putting $h\lambda = z$ and using Mathematica we obtain

$$\sqrt{1+h\lambda} = (1+z)^{\frac{1}{2}}$$

$$= 1 + \frac{z}{2} - \frac{z^2}{8} + \frac{z^3}{16} - \frac{5}{128}z^4 + \frac{7}{256}z^5 + 0(z^6)$$

and $\dfrac{1}{\sqrt{1-h\lambda}} = (1-z)^{-\frac{1}{2}}$

$$= 1 + \frac{z}{2} + \frac{3z^2}{8} + \frac{5z^3}{16} + \frac{35}{128}z^4 + \frac{63}{256}z^5 + 0(z^6)$$

Therefore the expression for

$$\sqrt{\frac{1+h\lambda}{1-h\lambda}} = (1+z)^{\frac{1}{2}}(1-z)^{-\frac{1}{2}}$$

$$= 1 + z + \frac{z^2}{2} + \frac{z^3}{2} + \frac{3}{8}z^4 + \frac{3}{8}z^5 + \frac{5}{16}z^6 + \frac{5}{16}z^7 + 0(z^8)$$

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 $\ln(1+w) = w - \dfrac{w^2}{2} + \dfrac{w^3}{3} - \dfrac{w^4}{4} + \dfrac{w^5}{5}$ (5.2.8)

and putting $w = z + \dfrac{z^2}{2} + \dfrac{z^3}{2} + \dfrac{3z^4}{8} + \dfrac{3z^5}{8} + \dfrac{5}{16}z^6 + \dfrac{5}{16}z^7 + \ldots$ in equation (5.2.8) we get,

$$\ln(1+w) = z + \frac{z^3}{2} + \frac{z^5}{5} + \frac{z^7}{7} + \ldots \qquad (5.2.9)$$

By substituting equation (5.2.9) in equation (5.2.7) we obtain

$$y(x;h) = \alpha \exp\left\{\left(\frac{x-a}{h}\right)\left(z + \frac{z^3}{3} + \frac{z^5}{5} + \frac{z^7}{7} + \ldots\right)\right\}$$

$$= \alpha \exp\left\{\lambda(x-a)\left(1+\frac{z^2}{3}+\frac{z^4}{5}+\frac{z^6}{7}+\ldots\right)\right\}$$

$$= \sum_{j=0}^{k} C_j(x)h^j + 0\left(h^{k+1}\right) \qquad (5.2.10)$$

To find $C_j(x)$, we differentiate equation (5.2.10) with respect to x to give

$$\sum_{j=0}^{k} C_j' h^j = \lambda\left[1+\frac{z^2}{3}+\frac{z^4}{5}+\frac{z^6}{7}+\ldots\right]\alpha \exp\left\{\lambda(x-a)\left(1+\frac{z^2}{3}+\frac{z^4}{5}+\frac{z^6}{7}\right)\right\}$$

$$= \lambda\left[1+\frac{z^2}{3}+\frac{z^4}{5}+\frac{z^6}{7}+\ldots\right]\sum_{j=0}^{k} C_j(x)h^j \qquad (5.2.11)$$

Equating the coefficients $h^j, j = 0,1,2,3,4,5$ we obtain,

h^0: $C_0'(x) = \lambda C_0(x)$ (5.2.12a)

h^1: $C_1'(x) = \lambda C_1(x)$ (5.2.12b)

h^2: $C_2'(x) = \lambda C_2(x) + \frac{\lambda^3}{3} C_0(x)$ (5.2.12c)

h^3: $C_3'(x) = \lambda C_3(x) + \frac{\lambda^3}{3} C_1(x)$ (5.2.12d)

h^4: $C_4'(x) = \lambda C_4(x) + \frac{\lambda^3}{3} C_2(x) + \frac{\lambda^5}{5} C_0(x)$ (5.2.12e)

h^5: $C_5'(x) = \lambda C_5(x) + \frac{\lambda^3}{3} C_3(x) + \frac{\lambda^5}{5} C_1(x)$ (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,

$C_0(x) = A\exp(\lambda x)$ where A is constant.

Now at x=a, $C_0(a) = \alpha$ we obtain

$A = \alpha \exp(-\lambda a)$

Thus,

$$C_0(x) = \alpha \exp(\lambda(x-a)). \qquad (5.2.13)$$

From equation (5.2.12b), we have

$$C_1(x) = 0 \quad \text{when} \quad C_1(a) = 0.$$

From equation (5.2.12c),

$$C_2'(x) = \lambda C_2(x) + \frac{\lambda^3}{3} C_0(x)$$

$$C_2'(x) = \lambda C_2(x) + \frac{\lambda^3}{3} \alpha \exp(\lambda(x-a))$$

With the integrating factor $\exp(-\int f(x)dx) = \exp(-\lambda x)$ and $C_2(a) = 0$, we get

$$\frac{d(C_2(x)\exp(-\lambda x))}{dx} = \int \frac{1}{3}\lambda^3\alpha \exp(\lambda(x-a))dx$$

$$= \frac{1}{3}\lambda^3\alpha \exp(-\lambda a)x + C$$

$$C_2(x) = \frac{1}{3}\lambda^3\alpha \exp(\lambda(x-a))x + C\exp(\lambda x)$$

By substituting $C_2(a) = 0$, we have

$$C = -\frac{1}{3}\lambda^3\alpha a \exp(-\lambda a)$$

Thus, $$C_2(x) = \frac{\lambda^3}{3}(x-a)\alpha \exp(\lambda(x-a)) \qquad (5.2.14)$$

From equation (5.2.12d) we have

$$C_3(x) = 0, \quad \text{since } C_1(x) = 0 \text{ and } C_3(a) = 0 \qquad (5.2.15)$$

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

$$C_4'(x) = \lambda C_4(x) + \frac{\lambda^6}{9}(x-a)\alpha \exp(\lambda(x-a)) + \frac{\lambda^5}{5}\alpha \exp(\lambda(x-a))$$

$$\frac{d}{dx}(C_4 \exp(-\lambda x)) = \int \exp(-\lambda x)\left[\frac{\lambda^6}{9}(x-a)\alpha \exp(\lambda(x-a)) + \frac{\lambda^5}{5}\alpha \exp(\lambda(x-a))\right]dx$$

$$= \int \left[\frac{\lambda^6}{9}\alpha \exp(-\lambda a)x - \frac{\lambda^6}{9}a\alpha \exp(-\lambda a) + \frac{\lambda^5}{5}\alpha \exp(-\lambda a)\right]dx$$

$$C_4(x) = \frac{\lambda^6}{18}\alpha\exp(\lambda(x-a))x^2 - \frac{\lambda^6}{9}a\alpha\exp(\lambda(x-a))x + \frac{\lambda^5}{5}\alpha\exp(\lambda(x-a))x + C\exp(\lambda x)$$

By substituting $C_4(a) = 0$, we get

$$C = \left[\frac{\lambda^6}{18}\alpha a^2 - \frac{\lambda^5}{5}\alpha a\right]\exp(-\lambda a)$$

and

$$C_4(x) = \frac{\lambda^6}{18}\alpha x^2\exp(\lambda(x-a)) - \frac{\lambda^6}{9}a\alpha\exp(\lambda(x-a))x + \frac{\lambda^5}{5}x\alpha\exp(\lambda(x-a))$$

$$+\left[\frac{\lambda^6}{18}\alpha a^2 - \frac{\lambda^5}{5}\alpha a\right]\exp(\lambda(x-a))$$

$$= \frac{\lambda^6}{18}(x^2 + a^2)\alpha\exp(\lambda(x-a)) + \frac{\lambda^5}{5}(x-a)\alpha\exp(\lambda(x-a)) - \frac{\lambda^6}{9}ax\alpha\exp(\lambda(x-a))$$

$$= \frac{\lambda^6}{18}(x-a)^2 C_0(x) + \frac{\lambda^5}{5}(x-a)C_0(x) \qquad (5.2.16)$$

From equation (5.2.12f) we have

$C_5(x) = 0$ since $C_3(x) = 0$, $C_1(x) = 0$ and $C_5(a) = 0$. (5.2.17)

Hence, the asymptotic expansion of equation (5.2.10) can be written as

$$y(x;h) = C_0(x) + C_2(x)h^2 + C_4(x)h^4 + \ldots$$

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 5.1: Coefficients $C_v(x)$ for $v = 0(1)5$

$C_0(x) = \exp[\lambda(x-a)]\alpha$
$C_1(x) = 0$
$C_2(x) = \frac{\lambda^3}{3}(x-a)C_0(x)$
$C_3(x) = 0$
$C_4(x) = \frac{\lambda^6}{18}(x-a)^2 C_0(x) + \frac{\lambda^5}{5}(x-a)C_0(x)$
$C_5(x) = 0$

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 (H_eM) and Contraharmonic mean method (C_oM) are shown in Table 5.2.

Table 5.2: C_i Coefficients of the four methods for the solution of $y' = \lambda y$ and $y(a) = \alpha$

C_i	Contraharmonic Meam	Arithmetic Mean	Geometric Mean	Heronian Mean
C_0	$\alpha \exp[\lambda(x-a)]$	$\alpha \exp[\lambda(x-a)]$	$\alpha \exp[\lambda(x-a)]$	$\alpha \exp[\lambda(x-a)]$
C_1	0	0	0	0
C_2	$\frac{\lambda^3}{3}(x-a)C_0$	$\frac{\lambda^3}{12}(x-a)C_0$	$-\frac{\lambda^3}{24}(x-a)C_0$	$\frac{\lambda^3}{24}(x-a)C_0$
C_3	0	0	0	0
C_4	$\left[\frac{\lambda^6}{18}(x-a)^2 + \frac{\lambda^5}{5}(x-a)\right]C_0$	$\left[\frac{\lambda^6}{288}(x-a)^2 + \frac{\lambda^5}{80}(x-a)\right]C_0$	$\left[\frac{\lambda^6}{1152}(x-a)^2 + \frac{3\lambda^5}{640}(x-a)\right]C_0$	$\left[\frac{\lambda^6}{1152}(x-a)^2 + \frac{17\lambda^5}{5760}(x-a)\right]C_0$
C_5	0	0	0	0

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,

$$C(h) = C_0 + C_2h^2 + C_4h^4 + \dots,$$

or $$T(h) = T_0 + T_2 h^2 + T_4 h^4 + \dots, \qquad (5.3.1)$$

then the process of repeated extrapolation produces the following,

$$P_i^{(0)} = T(h_i) = y(H; h_i) \qquad i = 0, 1, 2, \dots, s,$$

$$P_i^{(j)} = P_{i+1}^{(j-1)} + \frac{\left(P_{i+1}^{(j-1)} - p_i^{(j-1)}\right)}{\left(\frac{h_i}{h_{i+j}}\right)^2 - 1} \qquad j = 1, 2, \dots, s; \quad i = 0, 1, \dots, s-j \qquad (5.3.2)$$

giving $P_i^{(j)} = T_0 + O\left(h_i^{(2j+2)}\right)$, H is the basic steplength and $h_i = \frac{H}{N_i}$ such that $h_i > 0$ and $h_{i+1} < h_i$ and $N \in$ {sequence of numbers } where $N \in$ {2,4,6,8,16,...,1024} i.e. a modified form of the Romberg Sequence gives an extrapolation table of the form,

$$\begin{array}{ccccc}
P_0^{(0)} & & & & \\
 & P_0^{(1)} & & & \\
P_1^{(0)} & & P_0^{(2)} & & \\
 & P_1^{(1)} & & P_0^{(s)} & \\
P_2^{(0)} & & P_{s-2}^{(2)} & & \\
\cdot & & & & \\
\cdot & P_{s-1}^{(1)} & & & \\
\cdot & & & & \\
P_s^{(0)} & & & &
\end{array} \qquad (5.3.3)$$

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

$$\left|P_0^{(j)} - P_0^{(j-1)}\right| < TOL \qquad (5.3.4)$$

where s = min {1,2,...,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

$$y' = f(x,y) \quad , \quad y(a) = \alpha$$

and the tabulation by extrapolation is omitted here.

```
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 - 1][ 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

$$y' = y \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 10 \qquad (5.4.1)$$

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

No. Processor	Sequential Time	Parallel Time	Speedup
1	6.53	6.56	0.995
2		4.38	1.491
3		3.75	1.741
4		3.50	1.866
5		3.38	1.932
6		3.34	1.955
7		3.31	1.973
8		3.30	1.979
9		3.29	1.985
10		3.28	1.991

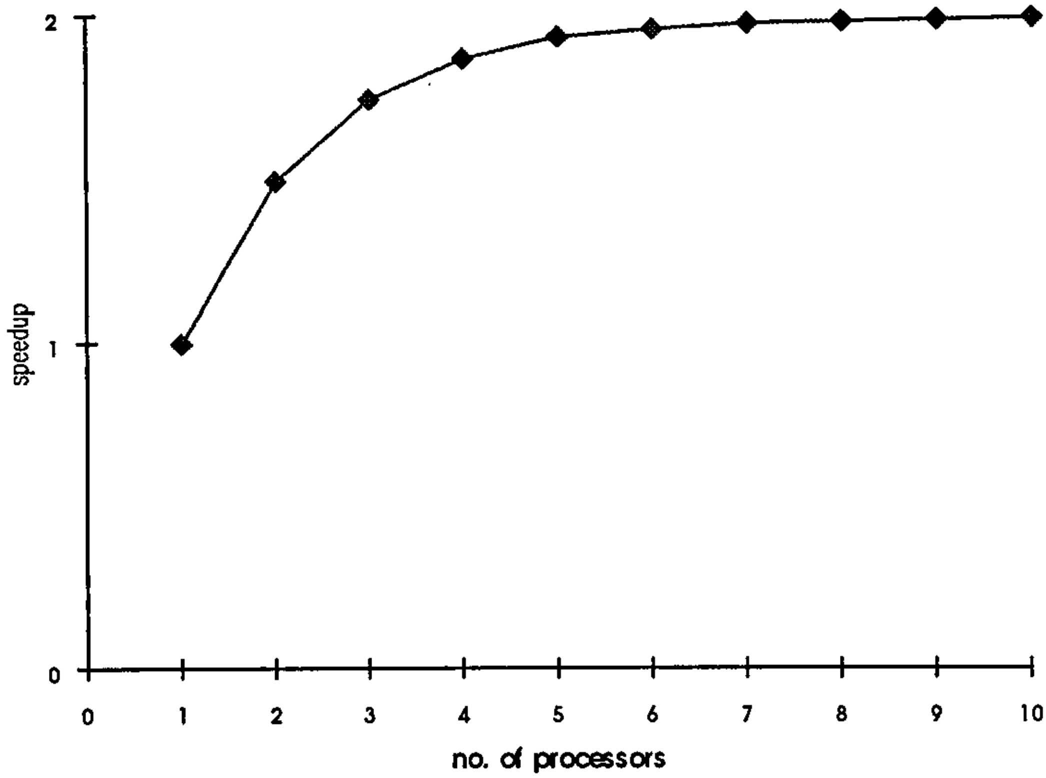


Figure 5.3 : Plot of speedup from Table 5.3

One example of a system of differential equations is

$$\begin{aligned} y'(x) &= y(x) - z(x) + 2x - x^2 - x^3 \\ z'(x) &= y(x) + z(x) - 4x^2 + x^3 \end{aligned} \qquad (5.4.2)$$

with initial condition

$$\begin{cases} y(0) = 1 \\ z(0) = 0 \end{cases}$$

and the exact solution of equation (5.4.2) is

$$\begin{aligned} y(x) &= e^x \cos(x) + x^2 \\ z(x) &= e^x \sin(x) - x^3 \end{aligned}$$

We can write the equation (5.4.2) as the system of equations in the following table

x	y_1	$y_1' = 1$
y	y_2	$y_2' = y_2 - y_3 + 2y_1 - y_1^2 - y_1^3$
z	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

```
void func (yy,F)
double yy [ ] , F [ ]
{
        F [1] = 1.0 ;
        F [2] = y[2] - y[3] + y[1]*(2.0 - y[1]*(1.0 + y[1])) ;
        F [3] = y[2] + y[3] - y[1]*y[1]*(4.0 - y[1] ) ;
}
```

The numerical values obtained by extrapolation method for solving equation (5.4.2) at point $x = 1.0$ is given in the following tableau

2.468672 2.468691 2.468694 2.468694 2.468694 2.468694
2.468691 2.468694 2.468694 2.468694 2.468694
2.468694 2.468694 2.468694 2.468694
2.468694 2.468694 2.468694
2.468694 2.468694
2.468694

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)

No. Processor	Sequential Time	Parallel Time	Speedup
1	12.31	12.42	0.9911
2		8.35	1.4743
3		7.24	1.7003
4		6.87	1.7918
5		6.70	1.8373
6		6.66	1.8483
7		6.62	1.8595
8		6.64	1.8539
9		6.69	1.8401
10		6.79	1.8131

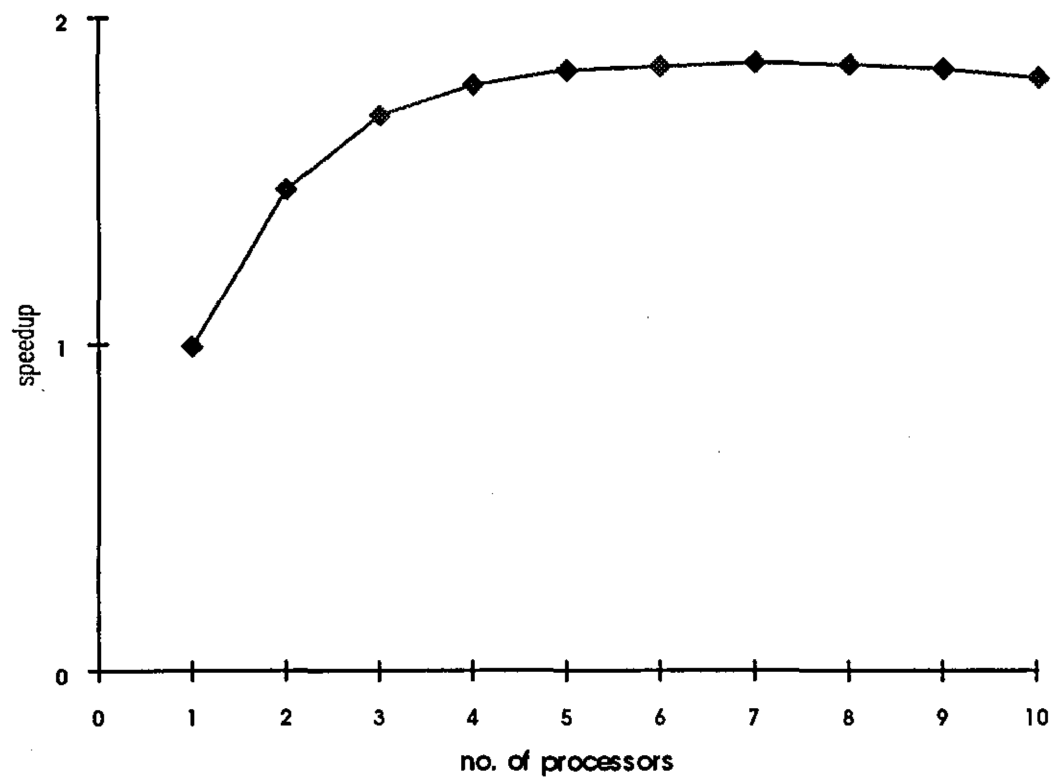


Figure 5.4 : Plot of speedup from Table 5.4.

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.

```
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

```
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

$$y' = f(x,y) \quad , \quad y(a) = \alpha$$

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.

```
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] = x0 ;
        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 ;
            splitt ++ ;
            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

$$y' = 0.92Cosh(x) - Cos(x) \quad , \quad y(0) = 0 \quad , \quad 0 \le x \le 10 \qquad (5.4.3)$$

with the exact solution y(x)=0.92Sinh(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.

No. Processor	Sequential Time	Parallel Time	Speedup
1	5.24	5.24	1.000
2		2.69	1.948
3		2.69	1.948
4		1.34	3.910
5		1.37	3.825
6		1.37	3.825
7		1.37	3.825
8		0.72	7.28
9		0.72	7.28
10		0.70	7.49

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.

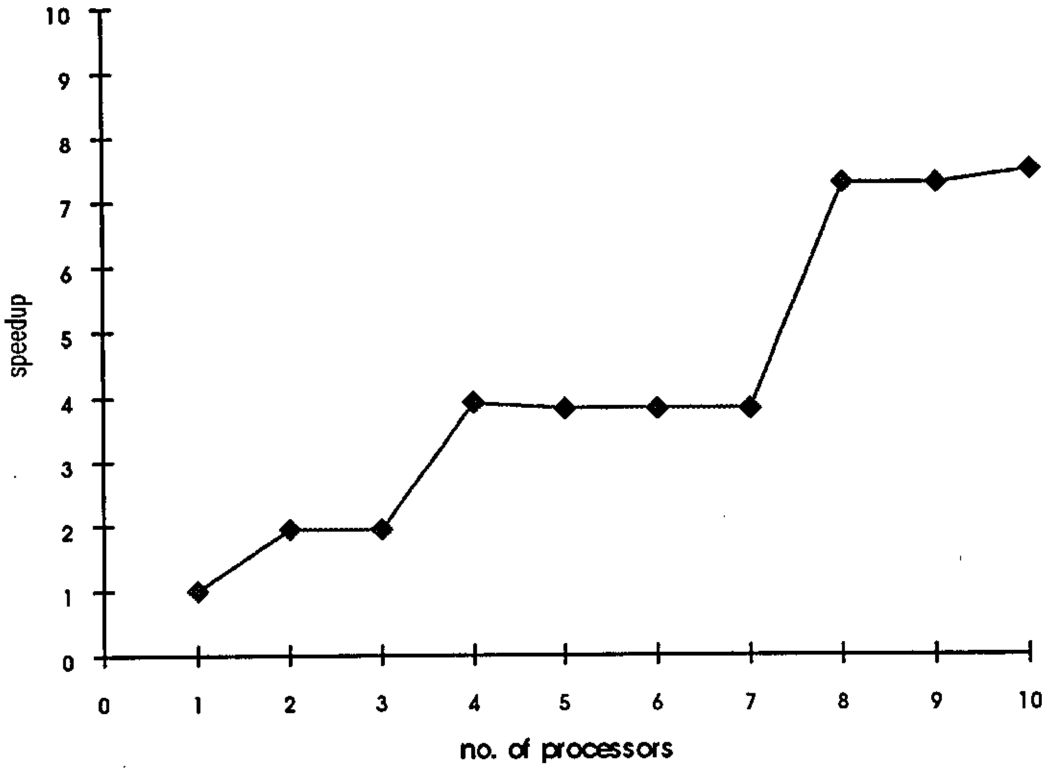


Figure 5.9 : Plot of speedups from Table 5.5

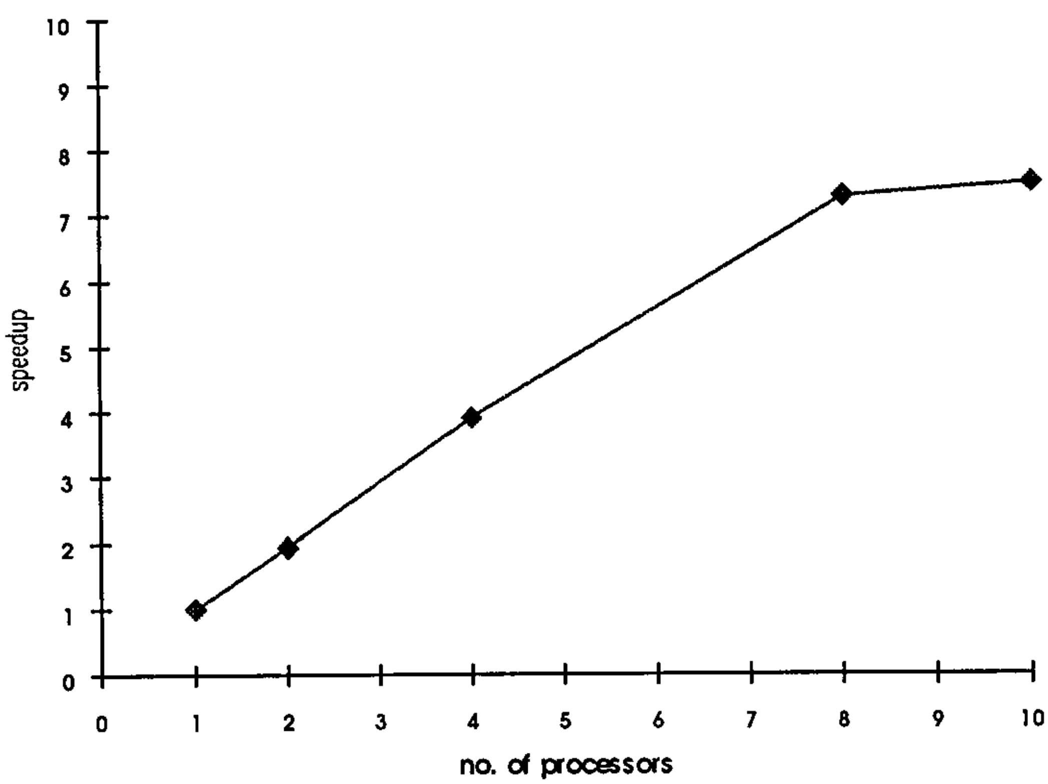


Figure 5.10 : Plot of speedups from Table 5.5 with number of processors of power two

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 * ((integrandl(x)*integrandl (x)) + integrandl (x)) +
integrandl(x+hl)*integrandl(x+hl))/(integrandl(x)+integrandl(x+hl)) ;
        x += hl  + hl ;
    }
part_sums[W_ARG_1][m_get_myid ( )  +  1]  += 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 ;
}
```

Figure 5.12 : Outline of the Sub_Manager for partitioning the nested loop of Figure 5.1

Producing the tableau requires the adding up of the values in the array part_sums[][] and 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

$$P_i^{(0)} = C(h_i) \quad , \qquad P_i^{(j)} = P_{i+1}^{(j-1)} + \frac{\left(P_{i+1}^{(j-1)} - P_i^{(j-1)}\right)}{\left(\dfrac{h_i}{h_{i+j}}\right)^2 - 1} \quad ,$$

where i=0,1,...,s-j;j=1,2,...,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, $h_{10} = (xf - x0)/10$. 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.

No. Processor	Sequential Time	Parallel Time	Speedup
1	28.40	28.49	0.997
2		14.59	1.947
3		9.67	2.937
4		7.45	3.812
5		6.03	4.710
6		5.44	5.221
7		4.54	6.256
8		4.08	6.961
9		3.77	7.533
10		3.60	7.889

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.

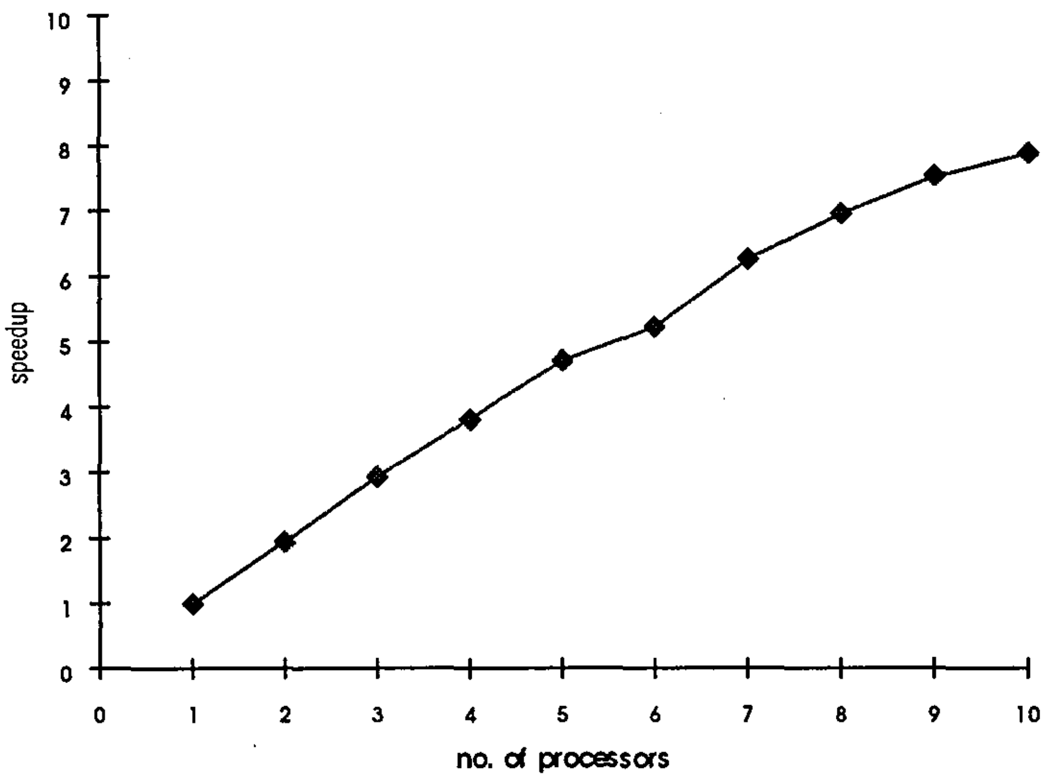


Figure 5.13 : Plot of speedups from Table 5.6

Table 5.7: Times in seconds and speedups for n = 15 and levels = 15 using ExDaTa Schedule

No. Processor	Sequential Time	Parallel Time	Speedup
1	1348.79	1350.31	0.999
2		675.28	1.997
3		450.28	2.995
4		340.75	3.958
5		270.64	4.984
6		225.93	5.970
7		194.40	6.938
8		169.87	7.940
9		151.10	8.926
10		136.21	9.902

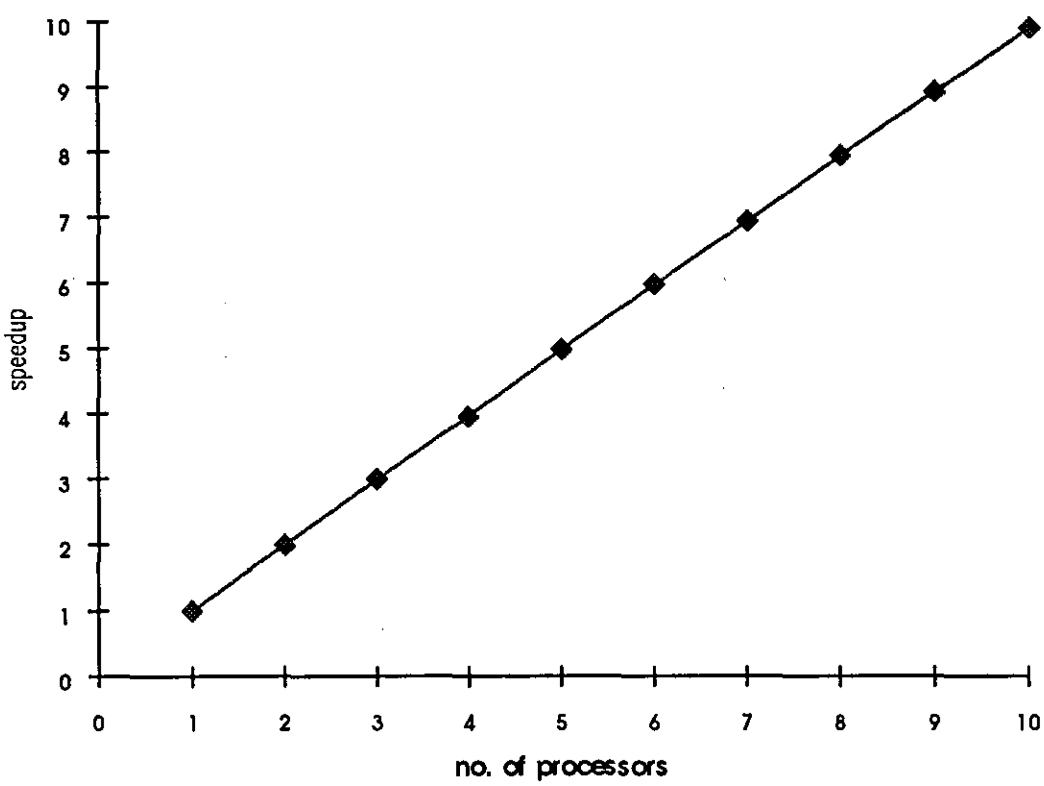


Figure 5.14 : Plot of speedups from Table 5.7

Table 5.8: Times in seconds and speedups for n = 16 and levels = 16 using ExDaTa Schedule.

No. Processor	Sequential Time	Parallel Time	Speedup
1	2844.33	2844.80	0.999
2		1430.03	1.989
3		951.70	2.989
4		714.28	3.982
5		571.65	4.976
6		475.29	5.984
7		408.70	6.959
8		356.53	7.985
9		318.20	8.939
10		285.89	9.949

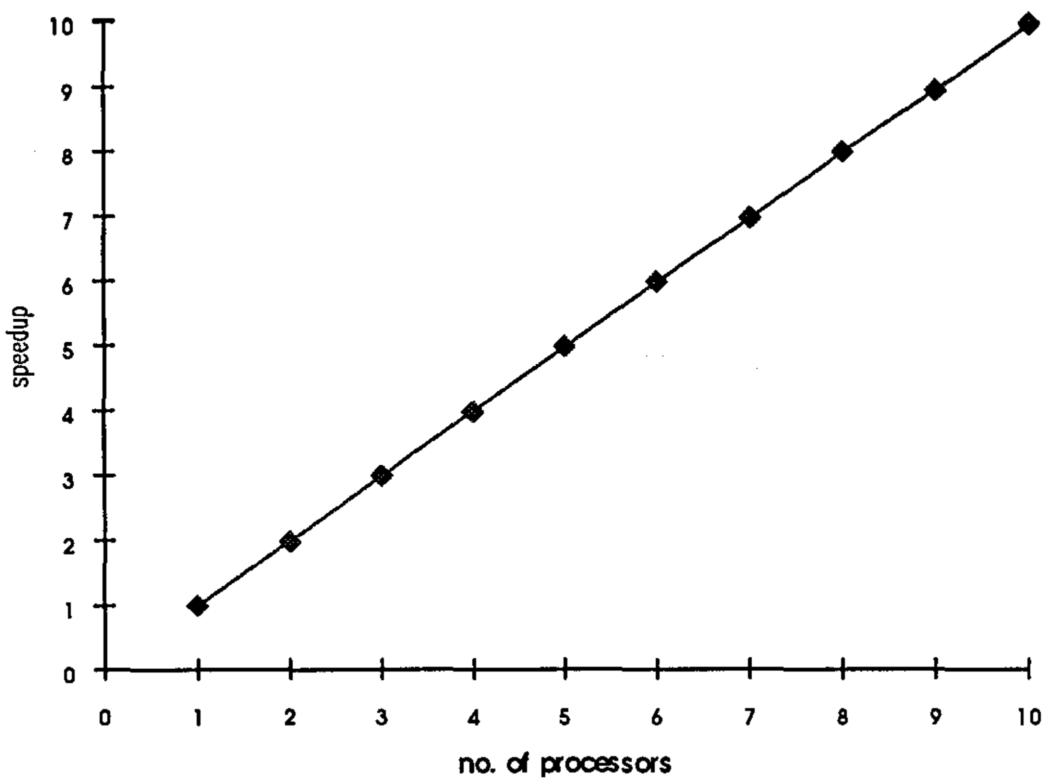


Figure 5.15 : Plot of speedups from Table 5.8

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_oM) 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

$$y(x_n) - \phi(y(x_{n-1})) = C(x_n)h^{p+1} + 0(h^{p+2})$$

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_oM) 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.,

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
c_2 & a_{21} & & & & & \\
c_3 & a_{31} & a_{32} & & & & \\
\vdots & \vdots & \vdots & \ddots & & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & & \\
c_{s+1} & a_{s+1,1} & a_{s+1,2} & \cdots & a_{s+1,s-1} & a_{s+1,s} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s & b_{s+1}
\end{array}
$$

(6.1.1)

for order $p+1$ and the method

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} & \\
\hline
 & a_{s+1,1} & a_{s+1,2} & \cdots & a_{s+1,s-1} & a_{s+1,s}
\end{array}
$$

(6.1.2)

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

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

(6.1.4)

where
$$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3}\left(\frac{k_i + k_{i+1}}{2}\right)\right] \tag{6.1.5}$$

and
$$
\left.
\begin{aligned}
k_1 &= f(x_n, y_n) \\
k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\
k_3 &= f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\
k_4 &= f(x_n + h, y_n + hk_3)
\end{aligned}
\right\} \tag{6.1.6}
$$

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

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

(6.1.7)

and the fourth order RK contraharmonic mean method with the Butcher array in the form

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
\frac{1}{2} & \frac{1}{8} & \frac{3}{8} & \\
1 & \frac{1}{4} & -\frac{3}{4} & \frac{3}{2} \\
\hline
 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{array}
\qquad (6.1.8)
$$

where $$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3}\left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \qquad (6.1.9)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{8}k_1 + \frac{3}{8}hk_2\right) \qquad (6.1.10)$$

$$k_4 = f\left(x_n + h, y_n + \frac{h}{4}k_1 - \frac{3}{4}hk_2 + \frac{3}{2}hk_3\right)$$

In Evans and Yaakub [1995], a new fourth order centroidal mean (C_eM) method can also be written in the form

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
\frac{1}{2} & \frac{1}{24} & \frac{11}{24} & \\
1 & \frac{11}{132} & -\frac{25}{132} & \frac{73}{66} \\
\hline
 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{array}
\qquad (6.1.11)
$$

where $$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3} \frac{2\left(k_i^2 + k_i k_{i+1} + k_{i+1}^2\right)}{3\left(k_i + k_{i+1}\right)}\right] \tag{6.1.12}$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{24}hk_1 + \frac{11}{24}hk_2\right) \tag{6.1.13}$$

$$k_4 = f\left(x_n + h, y_n + \frac{11}{132}hk_1 - \frac{25}{132}hk_2 + \frac{77}{66}hk_3\right)$$

While in Yaakub and Evans [1993] the root mean square (RMS) method in the Butcher array can be written as

$$\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
\frac{1}{2} & \frac{1}{16} & \frac{7}{16} & \\
1 & \frac{7}{56} & -\frac{17}{56} & \frac{33}{28} \\
\hline
 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{array} \tag{6.1.14}$$

where $$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3} \sqrt{\left(\frac{k_i^2 + k_{i+1}^2}{2}\right)}\right] \tag{6.1.15}$$

and $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{16}hk_1 + \frac{7}{16}hk_2\right) \tag{6.1.16}$$

$$k_4 = f\left(x_n + h, y_n + \frac{7}{56}hk_1 - \frac{17}{56}hk_2 + \frac{33}{28}hk_3\right)$$

6.2 ERROR CONTROL AND STEP SIZE SELECTION IN THE RK(4,4) METHOD

The combination between the fourth order classical Runge-Kutta formula

$$y_{AM} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3}\left(\frac{k_i + k_{i+1}}{2}\right)\right] \qquad (6.2.1)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_2\right)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

and the fourth order contraharmonic mean (C_oM) formula

$$y_{C_oM} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3}\left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \qquad (6.2.2)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{1}{8}hk_1 + \frac{3}{8}hk_2\right)$$

$$k_4 = f\left(x_n + h, y_n + \frac{1}{4}hk_1 - \frac{3}{4}hk_2 + \frac{3}{2}hk_3\right)$$

is called the RK(4,4) method. The difference between equation (6.2.1) and (6.2.2), i.e, $|y_{AM} - y_{C_oM}|$ 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

$$
\begin{array}{c|l}
C & A \\
\hline
 & b^T \\
 & \hat{b}^T \\
\hline
 & E^T
\end{array}
$$

(6.2.3)

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_{n+1} obtained by these two methods is then an estimate of the local trunscation error. Therefore, $E^T = \left| b^T - \hat{b}^T \right|$ where $E^T = \left[E_1, E_2, \ldots E_n \right]$.

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

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{1}{2} & \frac{1}{2} & & \\
\frac{1}{2} & 0 & \frac{1}{2} & \\
1 & 0 & 0 & 1 \\
\hline
\frac{1}{2} & \frac{1}{8} & \frac{3}{8} & \\
1 & \frac{1}{4} & -\frac{3}{4} & \frac{3}{2} \\
\hline
 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\hline
 & E^T & &
\end{array}
$$

(6.2.4)

where $$b^T = y_{n+1}^{AM} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3} \frac{k_i + k_{i+1}}{2} \right] \qquad (6.2.5)$$

and $$\hat{b}^T = y_{n+1}^{CoM} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3} \frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}} \right]. \qquad (6.2.6)$$

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

$$\left|\psi(x_n, y_n : h)\right| \leq \frac{73}{720} ML^4 \qquad (6.2.7)$$

where M and L are positive constant.

From the above approach, we can also obtain an error estimate for the four stage explicit $AM - C_oM$ 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

$$y_{n+1}^{AM} = y_n + LTE^{AM} \qquad (6.2.8)$$

and for the (C_oM) method

$$y_{n+1}^{C_oM} = y_n + LTE^{C_oM} \qquad (6.2.9)$$

where y_{n+1}^{AM} and $y_{n+1}^{C_oM}$ are the numerical approximations at x_{n+1} obtained by the AM and (C_oM) methods respectively and LTE_{AM} and LTE_{C_oM} 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+1} by

$$y_{n+1}^{AM} - y_{n+1}^{C_oM} = LTE^{AM} - LTE^{C_oM} \qquad (6.2.10)$$

The local truncation error involves y derivatives of the classical Runge-Kutta method and is given by

$$LTE_{AM} = \frac{h^5}{2880}\left[-24 ff_y^4 + f^4 f_{yyyy} + 2 f^3 f_y f_{yyy} - 6 f^3 f_{yy}^2 + 36 f^2 f_y^2 f_{yy}\right] \qquad (6.2.11)$$

while the local truncation error of the contraharmonic mean (C_oM) method is given by

$$LTE_{C_oM} = \frac{h^5}{23040}\left[-378 f f_y^4 - 8 f^4 f_{yyyy} + 4 f^3 f_y f_{yyy} - 648 f^3 f_{yy}^2 - 303 f^2 f_y^2 f_{yy}\right] \quad \ldots \quad (6.2.12)$$

The absolute difference between LTE_{AM} and LTE_{C_oM} is given by

$$\left|LTE_{AM} - LTE_{C_oM}\right| = h^5\left[\left(-\frac{24}{2880}+\frac{378}{23040}\right)ff_y^4 + \left(\frac{1}{2880}+\frac{8}{23040}\right)f^4 f_{yyyy} + \left(\frac{2}{2880}-\frac{4}{23040}\right)f^3 f_y f_{yyy} + \left(-\frac{6}{2880}+\frac{648}{23040}\right)f^3 f_{yy}^2 + \left(\frac{36}{2880}+\frac{303}{23040}\right)f^2 f_y^2 f_{yy}\right]$$

$$= h^5\left[\frac{31}{3840}ff_y^4 + \frac{1}{1440}f^4 f_{yyyy} + \frac{1}{1920}f^3 f_y f_{yyy} + \frac{5}{192}f^3 f_{yy}^2 + \frac{197}{7680}f^2 f_y^2 f_{yy}\right]$$

$$= \frac{h^5}{23040}\left[186 ff_y^4 + 16 f^4 f_{yyyy} + 12 f^3 f_y f_{yyy} + 600 f^3 f_{yy}^2 + 591 f^2 f_y^2 f_{yy}\right] \quad (6.2.13)$$

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

$$|f(x,y)| < Q \quad , \quad \left|\frac{\partial^{i+j} f(x,y)}{\partial x^i \partial y^j}\right| < \frac{P^{i+j}}{Q^{j-1}} \quad , \quad i+j \le p \quad (6.2.14)$$

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

$$|f_y| < P$$

$$|f_x + ff_y| < 2PQ$$

$$\left.\begin{aligned}
\left|f^4 f_y^4\right| &< \frac{Q^4.P^4}{Q^3}\\
\left|f^4 f_{yyyy}\right| &< \frac{Q^4P^4}{Q^3}\\
\left|f^3 f_y f_{yyy}\right| &< Q^3.P.\frac{P^3}{Q^2}\\
\left|f^3 f_{yy}^2\right| &< Q^3.\left(\frac{P^2}{Q}\right)^2\\
\left|f^2 f_y^2 f_{yy}\right| &< Q^2.P^2.\frac{P^2}{Q}
\end{aligned}\right\} < P^4Q \qquad (6.2.15)$$

From equation (6.2.13) and (6.2.14) we obtain

$$\left|LTE_{AM} - LTE_{C_oM}\right| \le \frac{281}{4608} P^4Qh^5 \qquad (6.2.16)$$

Hence,

$$\left|y_{n+1}^{Am} - y_{n+1}^{C_oM}\right| \le \frac{281}{4608} P^4Qh^5 \qquad (6.2.17)$$

If we suppose that the tolerance TOL, i.e. $\varepsilon < 0.00005$, then by setting

$$\left|y_{n+1}^{Am} - y_{n+1}^{C_oM}\right| \le TOL$$

the error control and step size selection can be determined by (6.2.17) to give the formula

$$\frac{281}{4608} P^4Qh^5 < TOL$$

or

$$h < \left[\frac{16.4 \times TOL}{P^4Q}\right]^{\frac{1}{5}} . \qquad (6.2.18)$$

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

$$y(x_n + h) = y(x_n) + hf + \frac{h^2}{2} ff_y + \frac{h^3}{6}\left(ff_y^2 + f^2 f_{yy}\right)$$

$$+\frac{h^4}{24}\left(f^3 f_{yyy} + 4f^2 f_y f_{yy} + ff_y^3\right) + 0(h^5) \qquad (6.2.19)$$

The third order arithmetic mean method is given in the form

$$y_{n+1} = y_n + \frac{h}{2}\left(\frac{k_1 + k_2}{2} + \frac{k_2 + k_3}{2}\right) \qquad (6.2.20)$$

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 $,1 \le i \le 3$ into equation (6.2.20), we obtain

$$y(x_n + h) = y(x_n) + hf + \frac{h^2}{2} ff_y + \frac{h^3}{6}\left(ff_y^2 + f^2 f_{yy}\right)$$

$$+h^4\left(\frac{1}{27} f^3 f_{yyy} + \frac{1}{6} f^2 f_y f_{yy}\right) + 0(h^5). \qquad (6.2.21)$$

Now we subtract equation (6.2.19) from equation (6.2.21), to obtain

$$y_{n+1} - y(x_n + h) = h^4\left[\left(\frac{1}{27} - \frac{1}{24}\right)f^3 f_{yyy} - \frac{1}{24} ff_y^3\right] + 0(h^5)$$

$$= h^4\left[-\frac{3}{648} f^3 f_{yyy} - \frac{27}{648} ff_y^3\right] + 0(h^5).$$

The local truncation error is

$$y_{n+1} - y(x_n + h) = \frac{h^4}{648} y^{iv}(\xi) = 0(h^4)$$

where $\xi \in |x_n, x_n + h|$.

To evaluate the GTE for the third order arithmetic mean method, we first put

$\epsilon_n = y_n - y(x_n)$ and $\epsilon_{n+1} = y_{n+1} - y(x_n + h)$

We subtract equation (6.2.19) from equation (6.2.21) to obtain

$$\epsilon_{n+1} = \epsilon_n + h[f(x_n, y_n) - f(x_n, y(x_n))] + \frac{h^2}{2}[f_y f(x_n, y_n) - f_y f(x_n, y(x_n))]$$

$$+\frac{h^3}{6}[f_y^2 f(x_n, y_n) - f_y^2 f(x_n, y(x_n))] + \frac{h^4}{648} y^{iv}(\xi)$$

$$\leq \epsilon_n + hL\,\epsilon_n + \frac{h^2}{2} L\,\epsilon_n + \frac{h^3}{6} L\,\epsilon_n + \frac{h^4}{648} y^{iv}(\xi)$$

We assume that f satisfies a Lipschitz condition with constant L and that $|y^{iv}(x)| \leq M$ for all x in the interval of interest, then we can make the estimate

$$|\epsilon_{n+1}| \leq \left[1 + hL + \frac{h^2}{2} L + \frac{h^3}{6} L\right]|\epsilon_n| + \frac{h^4}{648} M$$

$$\leq [1 + C]|\epsilon_n| + B$$

where $C = hL\left(\sum_{p=1}^{3} \frac{h^{p-1}}{p!}\right)$, $A = 1 + C$, $B = \frac{h^4}{648} M$ and $|y^{iv}(x)| \leq M$.

A simple induction proof gives

$$|\epsilon_n| \leq A^n |\epsilon_0| + \left(\sum_{k=0}^{n-1} A^k\right) B$$

i.e., for $A \neq 1$, where $A = (1 + C)$ since $\epsilon_0 = 0$ then from the geometric series we have

$$|\epsilon_n| \le \left(\frac{A^n - 1}{A - 1}\right) B \tag{6.2.22}$$

If we use the inequality

$$1 + x \le e^x$$

we now get

$$A^n = (1+C)^n = \left(1 + hL\left(\sum_{p=1}^{3} \frac{h^{p-1}}{p!}\right)\right)^n \le e^{Cn} = e^{Lhn\left(\sum_{p=1}^{3} \frac{h^{p-1}}{p!}\right)}$$

$$= e^{L(x_n - x_0)\left(\sum_{p=1}^{3} \frac{h^{p-1}}{p!}\right)}$$

$$= e^{DL(x_n - x_0)} \tag{6.2.23}$$

where $D = \sum_{p=1}^{3} \frac{h^{p-1}}{p!}$.

By inserting equation (6.2.23) into the inequality (6.2.22) for ϵ, we finally get

$$|\epsilon_n| \le \frac{h^3}{648LD} M\left(e^{DL(x_n - x_0)} - 1\right) \quad ,$$

hence the GTE is $0\left(h^3\right)$.

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

$$y(x_n + h) = y(x_n) + hf + \frac{h^2}{2} ff_y + \frac{h^3}{6}\left(ff_y^2 + f^2 f_{yy}\right)$$

$$+ \frac{h^4}{24}\left(f^3 f_{yyy} + 4f^2 f_y f_{yy} + ff_y^3\right)$$

$$+ \frac{h^5}{120}\left(ff_y^4 + f^4 f_{yyyy} + 7f^3 f_y f_{yyy} + 11 f^2 f_y^2 f_{yy} + 4f^3 f_{yy}^2\right). \tag{6.2.24}$$

The fourth order arithmetic mean (AM) Runge-Kutta method is

$$y_{n+1} = y_n + \frac{h}{3}\left[\frac{k_1 + k_2}{2} + \frac{k_2 + k_3}{2} + \frac{k_3 + k_4}{2}\right] \tag{6.2.25}$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

and local truncation error (LTE) is obtained by subtracting equation (6.2.24) from equation (6.2.25). The LTE for the fourth the order arithmetic mean method was also discussed in Jayes [1993] and Sanugi [1986], i.e.,

$$y_{n+1} - y(x_n + h) = \frac{h^5}{2880}\left[-24ff_y^4 + f^4 f_{yyyy} + 2f^3 f_y f_{yyy} - 6f^3 f_{yy}^2 + 36f^2 f_y^2 f_{yy}\right]$$
$$= 0(h^5)$$

By using the same procedure to evaluate the global truncation error (GTE) as for the third order classical Runge-Kutta method, we have

$$\epsilon_{n+1} = \epsilon_n + h\left[f(x_n, y_n) - f(x_n, y(x_n))\right] + \frac{h^2}{2}\left[f_y f(x_n, y_n) - f_y f(x_n, y(x_{n)})\right]$$

$$+\frac{h^3}{6}\left[f_y^2 f(x_n, y_n) - f_y^2 f(x_n, y(x_n))\right]$$

$$+\frac{h^4}{24}\left[f_y^4 f^4(x_n, y_n) - f_y^4(x_n, y(x_n)\right] + \frac{h^5}{2880}y^v(\xi)$$

$$\leq \epsilon_n + hL\epsilon_n + \frac{h^2}{2}L\epsilon_n + \frac{h^3}{6}L\epsilon_n + \frac{h^4}{24}L\epsilon_n + \frac{h^5}{2880}y^v(\xi)$$

$$|\epsilon_{n+1}| \leq \left[1 + hL + \frac{h^2}{2}L + \frac{h^3}{6}L + \frac{h^4}{24}L\right]|\epsilon_n| + \frac{h^5}{2880}M$$

$$\leq [1 + C]|\epsilon_n| + B$$

where $C = hL\left(\sum_{p=1}^{4}\frac{h^{p-1}}{p!}\right)$, $A = 1 + C$, $B = \frac{h^5}{2880}M$ and $|y^v(x)| \leq M$

A simple induction proof gives ,

$$|\epsilon_n| \leq A^n|\epsilon_0| + \left(\sum_{k=0}^{n-1}A^k\right)B$$

i.e., for $A \neq 1$, where $A = (1 + C)$ since $\epsilon_0 = 0$ then from the geometric series we have

$$|\epsilon_n| \le \left(\frac{A^n - 1}{A - 1}\right)B \qquad (6.2.26)$$

If we use the inequality

$$1 + x \le e^x$$

we now get

$$A^n = (1 + C)^n = \left(1 + hL\left(\sum_{p=1}^{4}\frac{h^{p-1}}{p!}\right)\right)^n \le e^{Cn} = e^{Lhn\left(\sum_{p=1}^{4}\frac{h^{p-1}}{p!}\right)}$$

$$= e^{DL(x_n - x_0)} \qquad (6.2.27)$$

where $D = \sum_{p=1}^{4}\frac{h^{p-1}}{p!}$.

By inserting equation (6.2.27) into the inequality (6.2.26) for ϵ, we finally get

$$|\epsilon_n| \le \frac{h^4}{2880LD}M\left(e^{DL(x_n - x_0)} - 1\right)$$

and therefore the GTE for the fourth order arithmetic mean (AM) method is $0\left(h^4\right)$.

6.2.1.3 The Fourth Order Contraharmonic Mean Method

The fourth order contraharmonic mean (C_0M) method can be expressed in the form

$$y_{n+1} = y_n + \frac{h}{3}\left[\frac{k_1^2 + k_2^2}{k_1 + k_2} + \frac{k_2^2 + k_3^2}{k_2 + k_3} + \frac{k_3^2 + k_4^2}{k_3 + k_4}\right] \qquad (6.2.28)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{8}k_1 + \frac{3}{8}hk_2\right)$$

$$k_4 = f\left(x_n + h, y_n + \frac{1}{4}hk_1 - \frac{3}{4}hk_2 + \frac{3}{2}hk_3\right)$$

and LTE is the difference between equation (6.2.24) and (6.2.28)

$$y_{n+1} - y(x_n + h) = \frac{h^5}{23040}\left[-378 ff_y^4 - 8 f^4 f_{yyyy} + 4 f^3 f_y f_{yyy} - 648 f^3 f_{yy}^2 - 303 f^2 f_y^2 f_{yy}\right]$$
$$= 0(h^5).$$

By using the same procedure to evaluate the GTE for the fourth order arithmetic mean (AM) method, we have

$$\epsilon_{n+1} \le \epsilon_n + hL\,\epsilon_n + \frac{h^2}{2} L\,\epsilon_n + \frac{h^3}{6} L\,\epsilon_n + \frac{h^4}{24} L\,\epsilon_n + \frac{h^5}{23040} y^v(\xi)$$

$$|\epsilon_{n+1}| \le \left[1 + hL + \frac{h^2}{2} L + \frac{h^3}{6} L + \frac{h^4}{24} L\right]|\epsilon_n| + \frac{h^5}{23040} M$$
$$\le [1 + C]|\epsilon_n| + B$$

where $C = hL\left(\sum_{p=1}^{4} \frac{h^{p-1}}{p!}\right)$, $A = 1 + C$, $B = \frac{h^5}{23040} M$ and $|y^v(x)| \le M$

A simple induction proof gives ,

$$|\epsilon_n| \le A^n |\epsilon_0| + \left(\sum_{k=0}^{n-1} A^k\right) B$$

i.e., for $A \ne 1$, where $A = (1 + C)$ since $\epsilon_0 = 0$ then from the geometric series we have

$$|\epsilon_n| \le \left(\frac{A^n - 1}{A - 1}\right) B \qquad (6.2.29)$$

If we use the inequality

$$1 + x \le e^x$$

we now get

$$A^n = (1 + C)^n = \left(1 + hL\left(\sum_{p=1}^{4} \frac{h^{p-1}}{p!}\right)\right)^n \le e^{Cn} = e^{Lhn\left(\sum_{p=1}^{4} \frac{h^{p-1}}{p!}\right)}$$
$$= e^{DL(x_n - x_0)} \qquad (6.2.30)$$

where $D = \sum_{p=1}^{4} \frac{h^{p-1}}{p!}$.

By inserting equation (6.2.30) into the inequality (6.2.29) for ϵ, we finally get

$$|\epsilon_n| \le \frac{h^4}{23040\, LD} M\left(e^{DL(x_n - x_0)} - 1\right)$$

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_0M) 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)

x	Exact Solution	Numerical Solution	Absolute Error	Estimate Error
h= .50000				
h= .25000				
.25000	.7788086D+00	.7788008D+00	.7810679D-05	.1680223D-05
.50000	.6065428D+00	.6065307D+00	.1216599D-04	.2617097D-05
.75000	.4723808D+00	.4723666D+00	.1421239D-04	.3057273D-05
1.00000	.3678942D+00	.3678794D+00	.1475824D-04	.3174651D-05
1.25000	.2865192D+00	.2865048D+00	.1436723D-04	.3090502D-05
1.50000	.2231436D+00	.2231302D+00	.1342712D-04	.2888240D-05
1.75000	.1737861D+00	.1737739D+00	.1219995D-04	.2624238D-05
2.00000	.1353461D+00	.1353353D+00	.1085872D-04	.2335706D-05
2.25000	.1054087D+00	.1053992D+00	.9513925D-05	.2046415D-05
2.50000	.8209323D-01	.8208500D-01	.8232766D-05	.1770819D-05
2.75000	.6393491D-01	.6392786D-01	.7052889D-05	.1517016D-05
3.00000	.4979306D-01	.4978707D-01	.5992170D-05	.1288848D-05
3.25000	.3877926D-01	.3877421D-01	.5055624D-05	.1087394D-05
3.50000	.3020162D-01	.3019738D-01	.4240217D-05	.9119994D-06
3.75000	.2352128D-01	.2351775D-01	.3538179D-05	.7609933D-06
4.00000	.1831858D-01	.1831564D-01	.2939254D-05	.6321681D-06

h= .50000				
4.50000	.1111518D-01	.1110900D-01	.6182387D-05	.1557395D-05
5.00000	.6744366D-02	.6737947D-02	.6419380D-05	.1655983D-05
5.50000	.4092285D-02	.4086771D-02	.5513370D-05	.1435470D-05
6.00000	.2483079D-02	.2478752D-02	.4326887D-05	.1131865D-05
6.50000	.1506660D-02	.1503439D-02	.3220759D-05	.8447929D-06
h= 1.00000				
7.50000	.5649975D-03	.5530844D-03	.1191311D-04	.5873917D-05
8.50000	.2118741D-03	.2034684D-03	.8405687D-05	.3947403D-05
9.50000	.7945277D-04	.7485183D-04	.4600941D-05	.2028028D-05
10.50000	.2979479D-04	.2753645D-04	.2258340D-05	.9324796D-06

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)

X	H	Y	EXACT	ABS. ERROR	NFC
.26591	.2659148	.766515E+00	.766504E+00	.106068E-04	8
.36591	.1000000	.693572E+00	.693562E+00	.966025E-05	16
.46591	.1000000	.627570E+00	.627561E+00	.879780E-05	24
.56591	.1000000	.567848E+00	.567840E+00	.801202E-05	32
.66591	.1000000	.513811E+00	.513803E+00	.729612E-05	40
.76591	.1000000	.464915E+00	.464908E+00	.664391E-05	48
.86591	.1000000	.420673E+00	.420667E+00	.604977E-05	56
.96591	.1000000	.380640E+00	.380635E+00	.550854E-05	64
1.06591	.1000000	.344418E+00	.344413E+00	.501553E-05	72

PROBLEM 3: Y'+ 2*X*Y = 0

INITIAL CONDITIONS X=0,Y=1

EXACT SOLUTION Y=EXP(-X**2)

X	H	Y	EXACT	ABS. ERROR	NFC
.02239	.0223856	.999499E+00	.999499E+00	.266454E-14	48
.04121	.0188275	.998303E+00	.998303E+00	.101030E-13	56
.06518	.0239672	.995761E+00	.995761E+00	.984768E-13	64
.09604	.0308584	.990819E+00	.990819E+00	.100386E-11	72
.13501	.0389689	.981938E+00	.981938E+00	.822475E-11	80
.18315	.0481424	.967012E+00	.967012E+00	.527802E-10	88
.24158	.0584314	.943309E+00	.943309E+00	.265218E-09	96
.31171	.0701269	.907409E+00	.907409E+00	.999335E-09	104
.39561	.0838992	.855127E+00	.855127E+00	.202847E-08	112
.49561	.1000000	.782214E+00	.782214E+00	.932170E-08	120
.59561	.1000000	.701350E+00	.701350E+00	.685975E-07	128
.69561	.1000000	.616394E+00	.616393E+00	.216906E-06	136
.79561	.1000000	.531002E+00	.531001E+00	.499300E-06	144
.89561	.1000000	.448382E+00	.448381E+00	.953703E-06	152
.99561	.1000000	.371120E+00	.371118E+00	.159881E-05	160
1.09561	.1000000	.301089E+00	.301087E+00	.242451E-05	168

PROBLEM 4: Y' + 3*X**2*Y = 0

INITIAL CONDITIONS X=0,Y=1

EXACT SOLUTION Y=EXP(-X**3)

X	H	Y	EXACT	ABS. ERROR	NFC
.10000	.1000000	.999000E+00	.999000E+00	.312240E-07	16
.19184	.0918385	.992965E+00	.992965E+00	.496714E-07	24
.27295	.0811133	.979870E+00	.979870E+00	.578590E-07	32
.34918	.0762251	.958320E+00	.958320E+00	.629045E-07	40
.42369	.0745160	.926761E+00	.926761E+00	.669893E-07	48
.49850	.0748065	.883488E+00	.883488E+00	.711656E-07	56
.57525	.0767501	.826664E+00	.826664E+00	.745578E-07	64
.65579	.0805445	.754248E+00	.754248E+00	.650008E-07	72
.74281	.0870162	.663744E+00	.663744E+00	.362135E-07	80
.84117	.0983564	.551466E+00	.551465E+00	.770462E-06	88
.94117	.1000000	.434451E+00	.434447E+00	.322746E-05	96
1.04117	.1000000	.323477E+00	.323468E+00	.893874E-05	104

PROBLEM: 5 Y' - Y = 0

INITIAL CONDITIONS X=0,Y=1

EXACT SOLUTION Y=EXP(X)

X	H	Y	EXACT	ABS. ERROR	NFC
.26591	.2659148	.130461E+01	.130462E+01	.115901E-04	8
.36591	.1000000	.144182E+01	.144183E+01	.129196E-04	16
.46591	.1000000	.159346E+01	.159347E+01	.144006E-04	24
.56591	.1000000	.176104E+01	.176106E+01	.160501E-04	32
.66591	.1000000	.194625E+01	.194627E+01	.178874E-04	40
.76591	.1000000	.215094E+01	.215096E+01	.199335E-04	48
.86591	.1000000	.237716E+01	.237718E+01	.222122E-04	56
.96591	.1000000	.262717E+01	.262719E+01	.247498E-04	64
1.06591	.1000000	.290347E+01	.290349E+01	.275753E-04	72

PROBLEM: 6 Y' + Y - X - 1 = 0

INITIAL CONDITIONS X=0,Y=1

EXACT SOLUTION Y=X + EXP(-X)

X	H	Y	EXACT	ABS. ERROR	NFC
.03901	.0390120	.100074E+01	.100075E+01	.824283E-05	32
.07186	.0328495	.100251E+01	.100252E+01	.797615E-05	40
.11518	.0433192	.100638E+01	.100639E+01	.763683E-05	48
.17217	.0569914	.101400E+01	.101401E+01	.720935E-05	56
.24531	.0731424	.102777E+01	.102777E+01	.668635E-05	64
.33685	.0915362	.105086E+01	.105087E+01	.606021E-05	72
.43685	.1000000	.108291E+01	.108292E+01	.542498E-05	80
.53685	.1000000	.112143E+01	.112144E+01	.485577E-05	88
.63685	.1000000	.116580E+01	.116581E+01	.434577E-05	96
.73685	.1000000	.121547E+01	.121547E+01	.388886E-05	104
.83685	.1000000	.126992E+01	.126992E+01	.347956E-05	112
.93685	.1000000	.132871E+01	.132871E+01	.311294E-05	120
1.03685	.1000000	.139142E+01	.139142E+01	.278459E-05	128

PROBLEM 7: Y' - X**2*EXP(X) = 0

INITIAL CONDITIONS X=0,Y=4

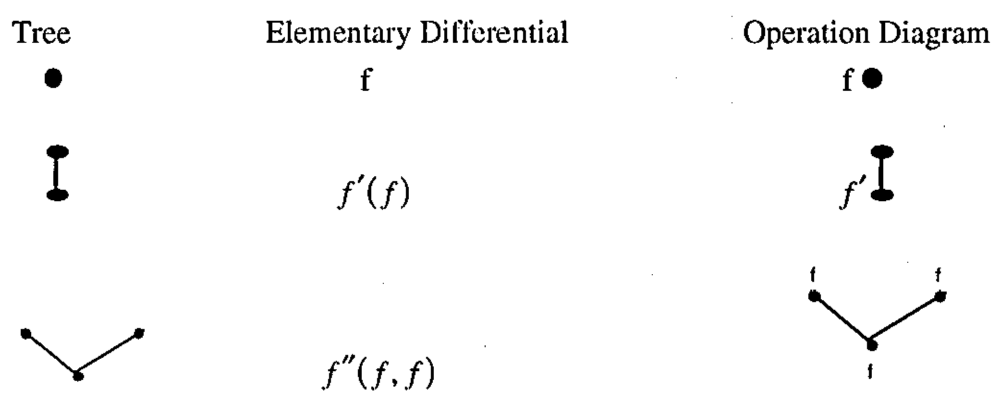
EXACT SOLUTION Y=X**2*EXP(X)-2**X*EXP(X)+2*EXP(X)+2

X	H	Y	EXACT	ABS. ERROR	NFC
.10000	.1000000	.400036E+01	.400036E+01	.452855E-07	16
.20000	.1000000	.400310E+01	.400310E+01	.986423E-07	24
.30000	.1000000	.401129E+01	.401129E+01	.161356E-06	32
.39820	.0981978	.402845E+01	.402845E+01	.228415E-06	40
.49097	.0927681	.405726E+01	.405726E+01	.287053E-06	48
.57721	.0862469	.409943E+01	.409943E+01	.333867E-06	56
.65706	.0798449	.415599E+01	.415599E+01	.370043E-06	64
.73109	.0740295	.422756E+01	.422756E+01	.397919E-06	72
.80000	.0689121	.431456E+01	.431456E+01	.419631E-06	80
.86446	.0644568	.441733E+01	.441732E+01	.436814E-06	88
.92504	.0605795	.453613E+01	.453613E+01	.450650E-06	96
.98222	.0571892	.467123E+01	.467123E+01	.461972E-06	104
1.03643	.0542044	.482287E+01	.482287E+01	.471376E-06	112

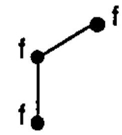
6.3 ANALYSIS OF VARIETY OF MEANS METHOD

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.

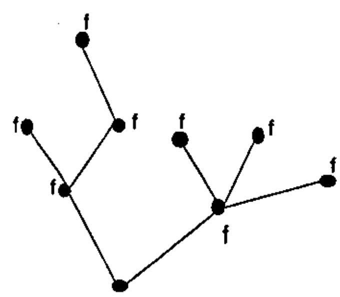


$$f'\left(f'(f)\right)$$



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

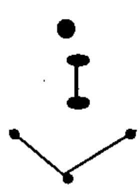


Let $f^i_{j_1 j_2 \cdots j_n} = f^i_{j_1 j_2 \cdots j_n}(y(x))$ denote an nth order partial derivative of component number i of f , where

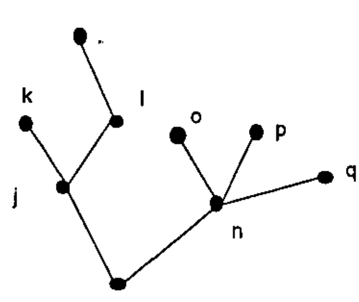
$(f)^i = f^i$, ,

$\left(f'(f)\right)^i = f^i_j f^j$, ,

$\left(f''(f,f)\right)^i = f^i_{jk} f^j f^k$,



For the above example given , we obtain the labelled tree

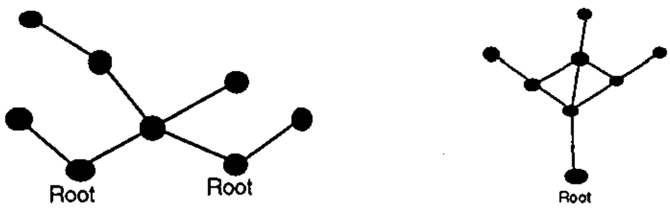


and the vector component , can be written as

$$f^i_{jn} f^j_{kl} f^k f^l_m f^m f^n_{opq} f^o f^p f^q$$

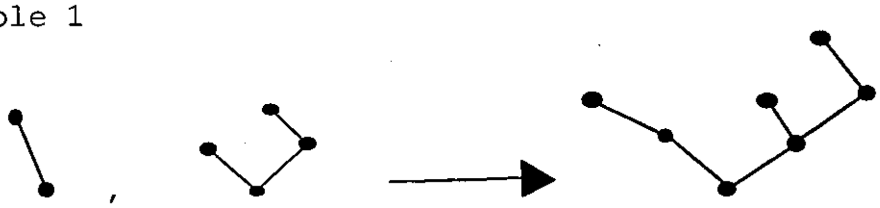
where the number of the system start from **left** to **right.**

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.

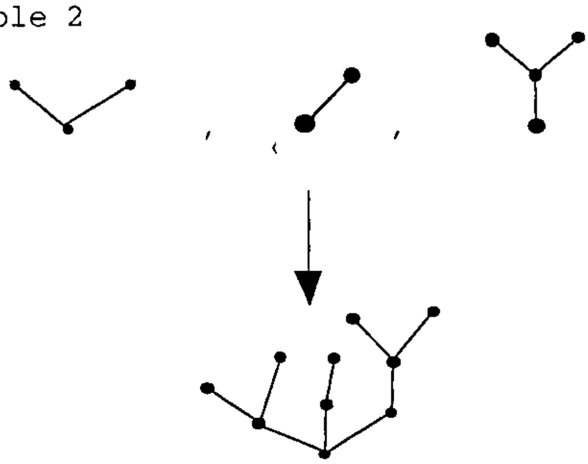


are forbidden.
We also can join two or more trees on a new root to produce a new tree , thus

Example 1



Example 2



In this example all the trees of orders up to 4 are shown in Table 1.

Order	Trees	Number of Trees
1		1
2		1

3 2

4 4

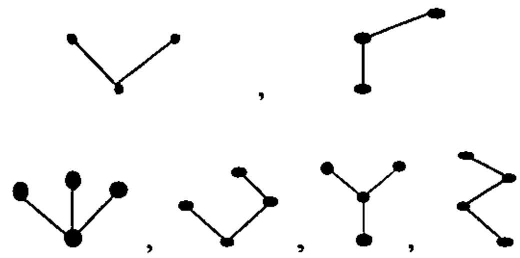


Table 6.1

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 . . . , satisfy termwise the identity

$$a_1 + a_2u + a_3u^2 + \ldots \equiv (1-u)^{-a_1}(1-u^2)^{-a_2}(1-u^3)^{-a_3}\ldots \qquad (6.3.1)$$

From the equation (6.3.1), we obtain the following table:-

n	1	2	3	4	5	6	7	8
a_n	1	1	2	4	9	20	48	115

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

$$y_{n+1} = y_n + h\left[\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5\right] \qquad (6.3.2)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{1}{4}h, y_n + \frac{1}{4}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{3}{8}h, y_n + \frac{3}{32}hk_1 + \frac{9}{32}hk_2\right)$$

$$k_4 = f\left(x_n + \frac{12}{13}h, y_n + \frac{1932}{2197}hk_1 - \frac{7200}{2197}hk_2 + \frac{7296}{2197}hk_3\right)$$

$$k_5 = f\left(x_n + h, y_n + \frac{439}{216}hk_1 - 8hk_2 + \frac{3680}{513}hk_3 - \frac{845}{4104}hk_4\right)$$

This scheme requires one more function evaluation, i.e

$$k_6 = f\left(x_n + \frac{1}{2}h, y_n - \frac{8}{27}hk_1 + 2hk_2 - \frac{3544}{2565}hk_3 + \frac{1859}{4104}hk_4 - \frac{11}{50}hk_5\right)$$

to obtain a Runge-Kutta method of order 5, i.e,

$$y^{*}_{n+1} = y_n + h\left[\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\right] \qquad (6.3.3)$$

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

$$
\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & & & \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & & & & \\
\frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & & & \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & & \\
\frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{50} & \\
\hline
 & \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & \\
 & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55} \\
\hline
 & \frac{1}{360} & 0 & -\frac{128}{4275} & -\frac{2197}{75240} & \frac{1}{50} & \frac{2}{55}
\end{array}
$$

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,

$$LTE = h\left[\frac{k_1}{360} - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{k_5}{50} + \frac{2}{55}k_6\right] \qquad (6.3.4)$$

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

$$
\begin{array}{c|ccccc}
0 & & & & & \\
\frac{1}{3} & \frac{1}{3} & & & & \\
\frac{1}{3} & \frac{1}{6} & \frac{1}{6} & & & \\
\frac{1}{2} & \frac{1}{8} & 0 & \frac{3}{8} & & \\
1 & \frac{1}{2} & 0 & -\frac{3}{2} & 2 & \\
\hline
 & \frac{1}{6} & 0 & 0 & \frac{2}{3} & \frac{1}{6}
\end{array}
\qquad (6.3.5)
$$

which is a 5-stage method of order 4. Merson proposed the principal local truncation error (LTE), i.e.,

$$LTE = \frac{h}{30}\left[-2k_1 + 9k_3 - 8k_4 + k_5\right] \qquad (6.3.6)$$

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 : $y = \exp(-x)$

x	Exact Solution	Numerical Solution	Absolute Error	Estimate Error
h= .50000				
h= .25000				
.25000	.7787992D+00	.7788008D+00	.1579345D-05	.1369379D-05
.50000	.6065282D+00	.6065307D+00	.2459988D-05	.1066471D-05
.75000	.4723637D+00	.4723666D+00	.2873758D-05	.8305667D-06
1.00000	.3678765D+00	.3678794D+00	.2984111D-05	.6468447D-06
1.25000	.2865019D+00	.2865048D+00	.2905032D-05	.5037621D-06
1.50000	.2231274D+00	.2231302D+00	.2714927D-05	.3923295D-06
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	.1425694D-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

h= 1.00000				
6.25000	.1916001D-02	.1930454D-02	.1445294D-04	.9243865D-05
7.25000	.7000774D-03	.7101744D-03	.1009703D-04	.3377566D-05
8.25000	.2557975D-03	.2612586D-03	.5461061D-05	.1234111D-05
9.25000	.9346447D-04	.9611165D-04	.2647182D-05	.4509251D-06
h= 2.00000				
11.25000	.2396525D-05	.1300730D-04	.1061077D-04	.6710270D-05

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 : $y = \exp(-x)$

x	Exact Solution	Numerical Solution	Absolute Error	Estimate Error
h= .50000				
h= .25000				
.25000	.7788018D+00	.7788008D+00	.1028995D-05	.1356337D-05
.50000	.6065323D+00	.6065307D+00	.1602765D-05	.1056318D-05
.75000	.4723684D+00	.4723666D+00	.1872353D-05	.8226620D-06
1.00000	.3678814D+00	.3678794D+00	.1944254D-05	.6406907D-06
1.25000	.2865067D+00	.2865048D+00	.1892735D-05	.4989711D-06
1.50000	.2231319D+00	.2231302D+00	.1768877D-05	.3885996D-06
h= .50000				
2.00000	.1353415D+00	.1353353D+00	.6240554D-05	.9684546D-05
2.50000	.8209192D-01	.8208500D-01	.6919561D-05	.5874198D-05
3.00000	.4979317D-01	.4978707D-01	.6098152D-05	.3563017D-05
3.50000	.3020224D-01	.3019738D-01	.4851913D-05	.2161162D-05
4.00000	.1831928D-01	.1831564D-01	.3642310D-05	.1310861D-05
4.50000	.1111163D-01	.1110900D-01	.2633442D-05	.7951077D-06
5.00000	.6739802D-02	.6737947D-02	.1854606D-05	.4822756D-06
h= 1.00000				
6.00000	.2480621D-02	.2478752D-02	.1869247D-05	.9360836D-05
7.00000	.9130065D-03	.9118820D-03	.1124531D-05	.3445308D-05
8.00000	.3360371D-03	.3354626D-03	.5744853D-06	.1268065D-05
9.00000	.1236803D-03	.1234098D-03	.2705223D-06	.4667182D-06
h= 2.00000				
11.00000	.1374226D-04	.1670170D-04	.2959442D-05	.5496903D-05

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) : $y' + 3x^2 y = 0$
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): $y' - x^2 Sin(x) + \frac{1}{x} + 1 = 0$
Initial conditions : $x_0 = 1$, $y_0 = 4$
Exact solution : $y = -x - Log(x) + x^2 Cos(x) - 2xSin(x) - 2Cos(x) + C$
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_0M) method. These error estimates (ERREST) used together with a constant derived in equation (6.2.1 - 6.2.14) are of the form,

$$ERREST = \left|Y_{AM} - Y_{C_0M}\right| * \frac{281}{4608} \qquad (6.3.7)$$

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.

Problem	TimeTaken		Absolute Error	
	RK(4,4)	RKF(4,5)	RK(4,4)	RKF(4,5)
1	1.80	1.11	0.2258×10^{-5}	0.1061×10^{-4}
2	0.20	0.10	0.3927×10^{-5}	0.1496×10^{-4}
3	0.20	0.15	0.1389×10^{-5}	0.3170×10^{-4}
4	0.20	0.10	0.8308×10^{-6}	0.2984×10^{-5}
5	0.25	0.05	0.3253×10^{-6}	0.9172×10^{-5}
6	0.22	0.04	0.2949×10^{-6}	0.2427×10^{-5}
7	0.12	0.09	0.5516×10^{-4}	0.7625×10^{-5}
8	0.20	0.08	0.1759×10^{-5}	0.2577×10^{-5}

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.

Problem	Time Taken		Absolute Error	
	RK(4,4)	Merson	RK(4,4)	Merson
1	1.80	0.98	0.2258×10^{-5}	0.2959×10^{-5}
2	0.20	0.21	0.3927×10^{-5}	0.2093×10^{-5}
3	0.20	0.90	0.1389×10^{-4}	0.8313×10^{-6}
4	0.20	0.09	0.8308×10^{-6}	0.1944×10^{-5}
5	0.26	0.24	0.3253×10^{-6}	0.3253×10^{-6}
6	0.22	0.20	0.2949×10^{-6}	0.6047×10^{-6}
7	0.12	0.10	0.5516×10^{-4}	0.1446×10^{-4}
8	0.20	0.20	0.1759×10^{-5}	0.2955×10^{-6}

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.

Problem	Time Taken		
	RK(4,4)	RKF(4,5)	Merson
1	1.80	1.11	0.98
2	0.20	0.10	0.21
3	0.20	0.15	0.90
4	0.20	0.10	0.09
5	0.26	0.02	0.24
6	0.22	0.04	0.20
7	0.12	0.09	0.10
8	0.20	0.08	0.20

Table 6.5

Problem	Absolute Error		
	RK(4,4)	Merson	RKF(4,5)
1	0.2258×10^{-5}	0.2959×10^{-5}	0.1061×10^{-4}
2	0.3927×10^{-5}	0.2093×10^{-5}	0.1496×10^{-4}
3	0.1389×10^{-4}	0.8313×10^{-6}	0.3170×10^{-4}
4	0.8308×10^{-6}	0.1944×10^{-5}	0.2984×10^{-5}
5	0.3253×10^{-6}	0.3253×10^{-6}	0.1175×10^{-4}
6	0.2949×10^{-6}	0.6047×10^{-6}	0.2427×10^{-5}
7	0.5516×10^{-4}	0.1446×10^{-4}	0.7625×10^{-5}
8	0.1759×10^{-5}	0.2955×10^{-6}	0.2577×10^{-5}

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.

Problem	TimeTaken		Absolute Error	
	RK(4,4)	RKF(4,5)	RK(4,4)	RKF(4,5)
1	1.80	1.11	0.2258×10^{-5}	0.1061×10^{-4}
2	0.09	0.10	0.5505×10^{-4}	0.1496×10^{-4}
3	0.09	0.15	0.9762×10^{-4}	0.3170×10^{-4}
4	0.09	0.10	0.1476×10^{-4}	0.2984×10^{-5}
5	0.12	0.05	0.5462×10^{-5}	0.9172×10^{-5}
6	0.11	0.04	0.4635×10^{-5}	0.2427×10^{-5}
7	0.10	0.09	0.7189×10^{-4}	0.7625×10^{-5}
8	0.09	0.08	0.2862×10^{-4}	0.2577×10^{-5}

Table 6.7

Problem	Time Taken		Absolute Error	
	RK(4,4)	Merson	RK(4,4)	Merson
1	1.80	0.98	0.2258×10^{-5}	0.2959×10^{-5}
2	0.09	0.10	0.5505×10^{-4}	0.3719×10^{-4}
3	0.09	0.10	0.9762×10^{-4}	0.1861×10^{-4}
4	0.09	0.09	0.1476×10^{-4}	0.1944×10^{-5}
5	0.12	0.11	0.5462×10^{-5}	0.5462×10^{-5}
6	0.11	0.10	0.4635×10^{-5}	0.4635×10^{-5}
7	0.10	0.09	0.7189×10^{-4}	0.1446×10^{-4}
8	0.09	0.09	0.2862×10^{-4}	0.5504×10^{-5}

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 $\left(\frac{1}{2}\right)^4\varepsilon$ or $2^{-4}\varepsilon$, then we double the stepsize. We use 2^{-4} instead of 2^{-5}, because it is an $0\left(h^4\right)$ method.

Thus in the RK(4,4) method with error control, we can delete 'h' because it is a $0(h^4)$ method using the fourth order classical arithmetic mean (AM) method and contraharmonic mean (C_0M) 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

$$ERREST = \left|Y_{AM} - Y_{C_0M}\right| * h * \frac{281}{4608} \quad (6.3.8)$$

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.

Problem	Time Taken		Absolute Error	
	RK(4,4)	RKF(4,5)	RK(4,4)	RKF(4,5)
1	1.15	1.11	0.1803×10^{-5}	0.1061×10^{-4}
2	0.21	0.10	0.3927×10^{-5}	0.1496×10^{-4}
3	0.21	0.15	0.1389×10^{-4}	0.3170×10^{-4}
4	0.20	0.10	0.8308×10^{-6}	0.2984×10^{-5}
5	0.16	0.05	0.5462×10^{-5}	0.9172×10^{-5}
6	0.22	0.04	0.2949×10^{-6}	0.2427×10^{-5}
7	0.12	0.09	0.7189×10^{-4}	0.7625×10^{-5}
8	0.21	0.08	0.1759×10^{-5}	0.2577×10^{-5}

Table 6.9

Problem	Time Taken		Absolute Error	
	RK(4,4)	Merson	RK(4,4)	Merson
1	1.15	0.98	0.1803×10^{-5}	0.2959×10^{-5}
2	0.21	0.21	0.3927×10^{-5}	0.2093×10^{-5}
3	0.21	0.90	0.1389×10^{-4}	0.8313×10^{-6}
4	0.20	0.09	0.8308×10^{-6}	0.1944×10^{-5}
5	0.16	0.24	0.5462×10^{-5}	0.3253×10^{-6}
6	0.22	0.20	0.2949×10^{-6}	0.6047×10^{-6}
7	0.12	0.10	0.7189×10^{-4}	0.1446×10^{-4}
8	0.20	0.20	0.1759×10^{-5}	0.2955×10^{-6}

Table 6.10

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.

Problem	Time Taken		
	RK(4,4)	RKF(4,5)	Merson
1	1.15	1.11	0.98
2	0.20	0.10	0.21
3	0.20	0.15	0.90
4	0.20	0.10	0.09
5	0.16	0.05	0.24
6	0.22	0.04	0.20
7	0.12	0.09	0.10
8	0.20	0.08	0.20

Table 6.11

Problem	Absolute Error		
	RK(4,4)	Merson	RKF(4,5)
1	0.1803×10^{-5}	0.2959×10^{-5}	0.1061×10^{-4}
2	0.3927×10^{-5}	0.2093×10^{-5}	0.1496×10^{-4}
3	0.1389×10^{-4}	0.8313×10^{-6}	0.3170×10^{-4}
4	0.8308×10^{-6}	0.1944×10^{-5}	0.2984×10^{-5}
5	0.5462×10^{-5}	0.3253×10^{-6}	0.1175×10^{-4}
6	0.2949×10^{-6}	0.6047×10^{-6}	0.2427×10^{-5}
7	0.7189×10^{-4}	0.1446×10^{-4}	0.7625×10^{-5}
8	0.1759×10^{-5}	0.2955×10^{-6}	0.2577×10^{-5}

Table 6.12

6.3.5 Automatic Selection of the Initial Stepsize 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 stepsize (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.

Problem	Initial Stepsize
1	$h = 0.25$
2	$h = 0.0625$
3	$h = 0.125$
4	$h = 0.0625$
5	$h = 0.125$
6	$h = 0.0625$
7	$h = 0.25$
8	$h = 0.125$

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 : $y = \exp(-x)$

x	Exact Solution	Numerical Solution	Absolute Error	Estimate Error
h= .50000				
h= .25000				
.25000	.7788086D+00	.7788008D+00	.7810679D-05	.4200556D-06
.50000	.6065428D+00	.6065307D+00	.1216599D-04	.6542743D-06
.75000	.4723808D+00	.4723666D+00	.1421239D-04	.7643182D-06
1.00000	.3678942D+00	.3678794D+00	.1475824D-04	.7936627D-06
1.25000	.2865192D+00	.2865048D+00	.1436723D-04	.7726255D-06
1.50000	.2231436D+00	.2231302D+00	.1342712D-04	.7220601D-06
1.75000	.1737861D+00	.1737739D+00	.1219995D-04	.6560594D-0
h= .50000				
2.25000	.1054484D+00	.1053992D+00	.4913849D-04	.6365842D-05
2.75000	.6398299D-01	.6392786D-01	.5512992D-04	.7236275D-05
3.25000	.3882301D-01	.3877421D-01	.4880501D-04	.6434257D-05
3.75000	.2355667D-01	.2351775D-01	.3892600D-04	.5141905D-05
4.25000	.1429350D-01	.1426423D-01	.2926750D-04	.3869708D-05
4.75000	.8672880D-02	.8651695D-02	.2118456D-04	.2802165D-05
5.25000	.5262450D-02	.5247518D-02	.1493208D-04	.1975352D-05
5.75000	.3193101D-02	.3182781D-02	.1032067D-04	.1365208D-05
6.25000	.1937481D-02	.1930454D-02	.7026700D-05	.9292943D-06
6.75000	.1175607D-02	.1170880D-02	.4727241D-05	.6250015D-06
h= 1.00000				
7.75000	.4408526D-03	.4307425D-03	.1011003D-04	.4768757D-05
8.75000	.1653197D-03	.1584613D-03	.6858390D-05	.3138291D-05
9.75000	.6199489D-04	.5829466D-04	.3700229D-05	.1600700D-05
10.75000	.2324808D-04	.2144541D-04	.1802677D-05	.7333294D-06

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 : $y = \exp(-x)$

x	Exact Solution	Numerical Solution	Absolute Error	Estimate Error
h= .50000				
h= .25000				
h= .50000				
h= 1.00000				
10.75000	.2324808D-04	.2144541D-04	.1802677D-05	.7333294D-06

Total Time = .12

Initial Stepsize For Problem No 1 is h = .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

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

with $c_2 = a_{21} = 1$ is of order 2 iff

$$b_1 + b_2 = 1 \quad , \qquad \bullet$$

$$b_2 c_2 = \frac{1}{2} \quad , \qquad I$$

6.4.2 3 - Stage of Third Order Method

Three stages of order 3 shows a method in the form

$$
\begin{array}{c|ccc}
0 & & & \\
c_2 & a_{21} & & \\
c_3 & a_{31} & a_{32} & \\
\hline
 & b_1 & b_2 & b_3
\end{array}
$$

or

$$
\begin{array}{c|ccc}
0 & & & \\
\frac{2}{3} & \frac{2}{3} & & \\
\frac{2}{3} & \frac{1}{3} & \frac{1}{3} & \\
\hline
 & \frac{1}{4} & 0 & \frac{3}{4}
\end{array}
$$

with $c_2 = a_{21} = \dfrac{2}{3}$, $c_3 = a_{31} + a_{32} = \dfrac{2}{3}$ is of order 3 iff

$$b_1 + b_2 + b_3 = 1 \quad ,$$

$$b_2c_2 + b_3c_3 = \frac{1}{2} \quad ,$$

$$b_3c_2^2 + b_3c_3^2 = \frac{1}{3} \quad ,$$

$$b_3a_{32}c_2 = \frac{1}{6} \quad ,$$

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

$$
\begin{array}{c|cccc}
0 & & & & \\
c_2 & a_{21} & & & \\
c_3 & a_{31} & a_{32} & & \\
c_4 & a_{41} & a_{42} & a_{43} & \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
$$

and from equation (6.1.4) with $c_2 = a_{21} = \dfrac{1}{2}$ we obtain the following eight equations :-

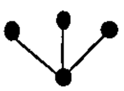
$$b_1 + b_2 + b_3 + b_4 = 1 \quad ,$$

$$b_2c_2 + b_3c_3 + b_4c_4 = \frac{1}{3} \quad ,$$

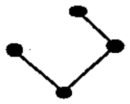
$$b_2c_2^2 + b_3c_3^2 + b_4c_4^2 = \frac{1}{2} \quad ,$$

$$b_3a_{32}c_2 + b_4a_{42}c_2 + b_4a_{43}c_3 = \frac{1}{6} \quad ,$$

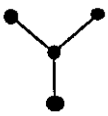
$$b_2c_2^3 + b_3c_3^3 + b_4c_4^3 = \frac{1}{4} \quad ,$$



$$b_3c_3a_{32}c_2 + b_4c_4a_{42}c_2 + b_4c_4a_{43}c_3 = \frac{1}{8} \quad ,$$



$$b_3a_{32}c_2^2 + b_4a_{42}c_2^2 + b_4a_{43}c_3^2 = \frac{1}{12} \quad ,$$



$$b_4a_{43}a_{32}c_2 = \frac{1}{24} \quad ,$$

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

0					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
c_4	a_{41}	a_{42}	a_{43}		
c_5	a_{51}	a_{52}	a_{53}	a_{54}	
	b_1	b_2	b_3	b_4	b_5

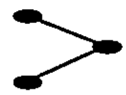
with the following seventeen equations where:-

$$b_1 + b_2 + b_3 + b_4 + b_5 = 1 \quad ,$$

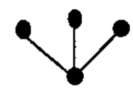
$$b_2c_2 + b_3c_3 + b_4c_4 + b_5c_5 = \frac{1}{3} \quad ,$$

$$b_2c_2^2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^5 = \frac{1}{2} \quad ,$$

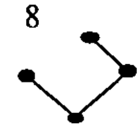
$$b_3a_{32}c_2 + b_4a_{42}c_2 + b_4a_{43}c_3 + b_5a_{52}c_2 + b_5a_{53}c_3 + b_5a_{54}c_4 = \frac{1}{6} \quad ,$$



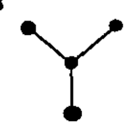
$$b_2c_2^3 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3 = \frac{1}{4} \quad ,$$



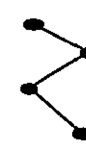
$$b_3c_3a_{32}c_2 + b_4c_4a_{42}c_2 + b_4c_4a_{43}c_3 + b_5c_5a_{52}c_2 + b_5c_5a_{53}c_3 + b_5c_5a_{54}c_4 = \frac{1}{8}$$



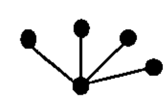
. . .



$$b_3a_{32}c_2^2 + b_4a_{42}c_2^2 + b_5a_{52}c_2^2 + b_4a_{43}c_3^2 + b_5a_{53}c_3^2 + b_5a_{54}c_4^2 = \frac{1}{12} \quad ,$$

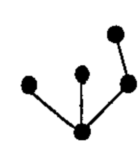


$$b_4a_{43}a_{32}c_2 + b_5a_{53}a_{32}c_2 + b_5a_{54}a_{43}c_3 = \frac{1}{24} \quad ,$$



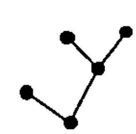
$$b_2c_2^4 + b_3c_3^4 + b_4c_4^4 + b_5c_5^4 = \frac{1}{5} \quad ,$$

$$b_3c_3^2a_{32}c_2 + b_4c_4^2a_{42}c_2 + b_5c_5^2c_{52}c_2 + b_4c_4^2c_{43}c_3 + b_5c_5^2c_{53}c_3 + b_5c_5^2c_{54}c_4 = \frac{1}{10}$$



. . .

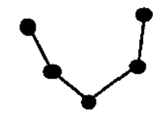
$$b_3c_3a_{32}c_2^2 + b_4c_4a_{42}c_2^2 + b_5c_5a_{52}c_2^2 + b_4c_4a_{43}c_3^2 + b_5c_5a_{53}c_3^2 + b_5c_5a_{54}c_4^2 = \frac{1}{15}$$



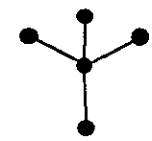
. . .

$$b_4c_4a_{43}a_{32}c_2 + b_5c_5a_{53}a_{32}c_2 + b_5c_5a_{54}a_{43}c_3 = \frac{1}{30} \quad ,$$

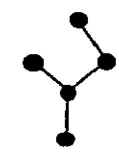
$$b_3\left(a_{32}c_2\right)^2 + b_4\left(a_{42}c_2 + a_{43}c_3\right)^2 + b_5\left(a_{52}c_2 + a_{53}c_3 + a_{54}c_4\right)^2 = \frac{1}{20}$$



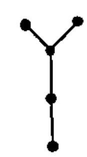
$$b_3a_{32}c_2^3 + b_4a_{42}c_2^3 + b_5a_{52}c_2^3 + b_4a_{43}c_3^3 + b_5a_{53}c_3^3 + b_5a_{54}c_4^3 = \frac{1}{20} \quad ,$$



$$b_4a_{43}c_3a_{32}c_2 + b_5a_{53}c_3a_{32}c_2 + b_5a_{54}c_4a_{43}c_3 = \frac{1}{40} \quad ,$$



$$b_4a_{43}a_{32}c_2^2 + b_5a_{53}a_{32}c_2^2 + b_5a_{54}a_{43}c_3^2 = \frac{1}{60} \quad ,$$



$$b_5a_{54}a_{43}a_{32}c_2 = \frac{1}{120} \quad ,$$

6.5 THE EXPLICIT CONTRAHARMONIC 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

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

where $k_1 = f(x_n, y_n)$

$$k_2 = f(x_n + h, y_n + hk_1)$$

also can be written as a Butcher array as

0		
c_2	a_{21}	
	b_1	b_2

or

$$
\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & 0 & 1
\end{array}
$$

and the four coefficient a_{21}, b_1, b_2 and c_2 can be formulated in two equations as

$$b_1 + b_2 = 1 \quad ,$$

$$b_2 c_2 = 1 \quad ,$$

6.5.2 A 3 - Stage Third Order (C_oM) Method

A 3-stage order 3 (C_oM) method in the form

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

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_2\right)$$

and can be shown in the Butcher array form as

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

with $c_2 = a_{21} = \frac{2}{3}$, $c_3 = a_{31} + a_{32} = \frac{2}{3}$ is of order 3 iff

$b_1 + b_2 = 1$, 

$b_2 c_2 = \frac{1}{3}$, 

$b_2 c_2^2 = \frac{2}{9}$, 

$b_2 a_{32} c_2 = \frac{2}{9}$, 

6.5.4 A 4 - Stage Fourth Order (C_oM) Method

The number of coefficients involved in the 4-stage fourth order (C_oM) 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

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ c_4 & a_{41} & a_{42} & a_{43} \\ \hline & b_1 & b_2 & b_3 \end{array}$$

or

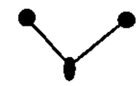
$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & \frac{1}{8} & \frac{3}{8} & \\ 1 & \frac{1}{4} & -\frac{3}{4} & \frac{3}{2} \\ \hline & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array}$$

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

$b_1 + b_2 + b_3 = 1$, 

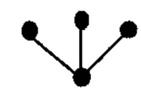
$$b_2c_2 + b_3c_3 = \frac{1}{3} \quad ,$$

$$b_2c_2^2 + b_3c_3^2 = \frac{1}{6} \quad ,$$

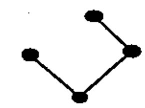


$$b_2a_{32}c_2 + b_3a_{42}c_2 + b_3a_{43}c_3 = \frac{7}{16} \quad ,$$

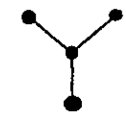
$$b_2c_2^3 + b_3c_3^3 = \frac{1}{6} \quad ,$$



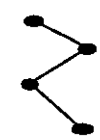
$$b_2c_3a_{32}c_2 + b_3c_4a_{42}c_2 + b_3c_4a_{43}c_3 = \frac{5}{32} \quad ,$$



$$b_3a_{32}c_2^2 + b_3a_{42}c_2^2 + b_3a_{43}c_3^2 = \frac{3}{32} \quad ,$$



$$b_3a_{43}a_{32}c_2 = \frac{3}{32} \quad ,$$



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

$$y_{n+1} = y_n + h\left[\frac{2(k_1^2 + k_1k_2 + k_2^2)}{3(k_1 + k_2)}\right]$$

where $k_1 = f(x_n, y_n)$

$k_2 = f(x_n + h, y_n + hk_1)$

and the four coefficient a_{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

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

with $c_2 = a_{21} = 1$ is of order 2 iff

$$b_1 + b_2 = 1 \quad ,$$

$$b_2 c_2 = \frac{1}{2} \quad ,$$

6.6.2 A 3 - Stage Third Order $(C_e M)$ Method

Three stages for an order 3 method can be shown in the form

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

where $\quad k_1 = f\left(x_n, y_n\right)$

$$k_2 = f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{2}{3}h, y_n - \frac{2}{9}hk_1 + \frac{8}{9}hk_2\right)$$

and in Butcher array form as

$$\begin{array}{c|cc} 0 & & \\ \frac{2}{3} & \frac{2}{3} & \\ \frac{2}{3} & -\frac{2}{9} & \frac{8}{9} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

with $c_2 = a_{21} = \frac{2}{3}$, $c_3 = a_{31} + a_{32} = \frac{2}{3}$ of order 3 with the conditions

$b_1 + b_2 = 1$,

$b_2 c_2 = \frac{1}{3}$,

$b_2 c_2^2 = \frac{2}{9}$,

$b_2 a_{32} c_2 = \frac{8}{27}$,

6.6.3 A 4 - Stage Fourth Order $(C_e M)$ 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

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

or

0			
$\frac{1}{2}$	$\frac{1}{2}$		
$\frac{1}{2}$	$\frac{1}{24}$	$\frac{11}{24}$	
1	$\frac{11}{132}$	$-\frac{25}{132}$	$\frac{73}{66}$
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

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 with the conditions

$$b_1 + b_2 + b_3 = 1 \quad ,$$

$$b_2c_2 + b_3c_3 = \frac{1}{3} \quad ,$$

$$b_2c_2^2 + b_3c_3^2 = \frac{1}{6} \quad ,$$

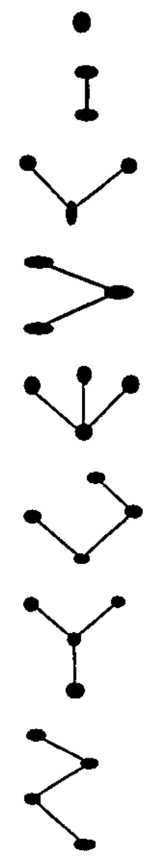
$$b_2a_{32}c_2 + b_3a_{42}c_2 + b_3a_{43}c_3 = \frac{11}{48} \quad ,$$

$$b_2c_2^3 + b_3c_3^3 = \frac{1}{12} \quad ,$$

$$b_2c_3a_{32}c_2 + b_3c_4a_{42}c_2 + b_3c_4a_{43}c_3 = \frac{55}{288} \quad ,$$

$$b_3a_{32}c_2^2 + b_4a_{42}c_2^2 + b_4a_{43}c_3^2 = \frac{11}{288} \quad ,$$

$$b_3a_{43}a_{32}c_2 = \frac{73}{864} \quad ,$$



which is the same number of coefficients as (C_0M).

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

$$\begin{array}{c|cc} 0 & & \\ c_2 & a_{21} & \\ \hline & b_1 & b_2 \end{array}$$

or

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

with $c_2 = a_{21} = 1$ and the four coefficient a_{21}, b_1, b_2 and c_2 can be formulated in two equations of conditions as

$$b_1 + b_2 = 1 \quad ,$$

$$b_2 c_2 = \frac{1}{2} \quad ,$$

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

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

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{2}{3}h, y_n + \frac{2}{3}hk_1\right)$$

$$k_3 = f\left(x_n + \frac{2}{3}h, y_n - \frac{1}{6}hk_1 + \frac{5}{6}hk_2\right)$$

and can be shown in Butcher array form as

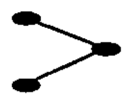
$$\begin{array}{c|cc} 0 & & \\ \frac{2}{3} & \frac{2}{3} & \\ \frac{2}{3} & -\frac{1}{6} & \frac{5}{6} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

with $c_2 = a_{21} = \dfrac{2}{3}$, $c_3 = a_{31} + a_{32} = \dfrac{2}{3}$ is of order 3 with four equations of conditions, i.e,

$$b_1 + b_2 = 1 \quad ,$$


$$b_2 c_2 = \frac{1}{3} \quad ,$$


$$b_2 c_2^2 = \frac{2}{9} \quad ,$$


$$b_2 a_{32} c_2 = \frac{5}{18} \quad ,$$


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 $(C_o M)$ 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

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ c_4 & a_{41} & a_{42} & a_{43} \\ \hline & b_1 & b_2 & b_3 \end{array}$$

or

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & \frac{1}{16} & \frac{7}{16} & \\ 1 & \frac{7}{56} & -\frac{17}{56} & \frac{33}{28} \\ \hline & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{array}$$

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 with eight equations of conditions, i.e,

$$b_1 + b_2 + b_3 = 1 \quad ,$$

$$b_2c_2 + b_3c_3 = \frac{1}{3} \quad ,$$

$$b_2c_2^2 + b_3c_3^2 = \frac{1}{6} \quad ,$$

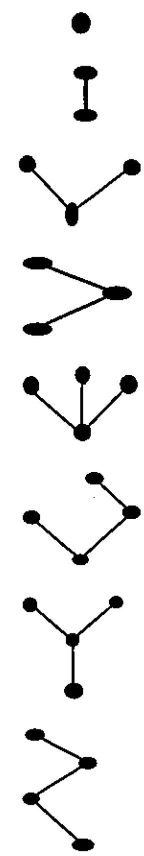
$$b_2a_{32}c_2 + b_3a_{42}c_2 + b_3a_{43}c_3 = \frac{5}{32} \quad ,$$

$$b_2c_2^3 + b_3c_3^3 = \frac{1}{12} \quad ,$$

$$b_2c_3a_{32}c_2 + b_3c_4a_{42}c_2 + b_3c_4a_{43}c_3 = \frac{7}{32} \quad ,$$

$$b_3a_{32}c_2^2 + b_3a_{42}c_2^2 + b_3a_{43}c_3^2 = \frac{49}{192} \quad ,$$

$$b_3a_{43}a_{32}c_2 = \frac{11}{128} \quad ,$$



From the above equations of conditions, we have shown that they have same number of terms as (C_oM).

From the above discussion, we conclude with the following hyphothesis:

Hyphothesis 6.7.4

If an s-stage explicit Runge-Kutta using non-linear method, i.e,. C_oM, C_eM and RMS methods has order N where $N \leq 3$ then $s \geq N$.

CHAPTER 7

NUMERICAL SOLUTION OF ODEs BY NONLINEAR MULTISTEP METHODS

We are given a p^{th} order initial value problem in the form

$$y^{(p)} = f\left(x, y, y^{(i)}, \ldots, y^{(p-1)}\right) \qquad (7.0.1)$$

with initial conditions

$$x_0 = \alpha_0, y(x_0) = y_0, y^{(1)}(x_0) = y_0^{(1)}, \ldots, y^{(p-1)}(x_0) = y_0^{(p-1)}.$$

Evans and Jayes [1993], introduced an approximation of $y(x_{n+1})$ as a geometric mean (GM) combination of the values $y_n, \ldots, y_{n-k-1}$ and the derivatives computed at $y_{n+1}, y_n, \ldots, y_{n-k-1}$. Thus,

$$y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \ldots + \alpha_k y_{n-k-1} + h\left\{\sum_{i,j=0}^{k-1} \beta_{i,j}\left[\sqrt{f(y_{n-i})f(y_{n-j})}\right]\right\} \qquad (7.0.2)$$

for some fixed numbers

$$\alpha_1, \alpha_2, \ldots, \alpha_k, \beta_{0,0}, \beta_{0,1}, \beta_{0,2}, \ldots, \beta_{0,k-1} \text{ and } \beta_{1,1}, \ldots, \beta_{k-1,k-1}.$$

We now study equation (7.0.2) with the C_oM method in the form

$$y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \ldots + \alpha_k y_{n-k-1} + h\left\{\sum_{i,j=0}^{k-1} \beta_{i,j}\left[\frac{f^2(y_{n-i}) + f^2(y_{n-j})}{f(y_{n-i}) + f(y_{n-j})}\right]\right\} \qquad (7.0.3)$$

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

$$y_{n+1} - y_n = h\left\{\alpha_1 f_n + \alpha_2 f_{n+1} + \alpha_3\left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right)\right\} \qquad (7.1.1)$$

The values of α_1, α_2 and α_3 are to be determined to give the highest order of accuracy. The Taylor series of y_{n+1} at x_n give

$$y_{n+1} - y_n = hf_n + \frac{h^2}{2!}f_n^{(1)} + \frac{h^3}{3!}f_n^{(2)} + \frac{h^4}{4!}f_n^{(3)} + 0(h^5) \qquad (7.1.2)$$

and the Taylor series expansion of f_{n+1} about x_n , we have

$$f_{n+1} = f_n + hf_n^{(1)} + \frac{h^2}{2!}f_n^{(2)} + \frac{h^3}{3!}f_n^{(3)} + \frac{h^4}{4!}f_n^{(4)} + 0(h^5) \tag{7.1.3}$$

By substituting (7.1.3) into (7.1.1), we obtain the right-hand side of (7.1.1) as

$$(\alpha_1 + \alpha_2 + \alpha_3)hf_n + \tfrac{1}{2}(2\alpha_2 + \alpha_3)h^2 f_n^{(1)} + h^3\left(\frac{2\alpha_2 + \alpha_3}{4}\right)f_n^{(2)} + h^3\left(\frac{\alpha_3}{4}\right)\frac{\left[f_n^{(1)}\right]^2}{f_n}$$

$$+h^4\left(\frac{2\alpha_2 + \alpha_3}{12}\right)f_n^{(3)} + h^4\left(\frac{\alpha_3}{4}\right)\frac{f_n^{(1)}f_n^{(2)}}{f_n} + h^4\left(\frac{\alpha_3}{8}\right)\frac{\left(f_n^{(1)}\right)^3}{\left(f_n\right)^2} + 0(h^5). \tag{7.1.4}$$

By equating the coefficients terms in (7.1.2) and (7.1.4), the following equations of conditions are obtained :

$$hf_n: \quad -1 + \alpha_1 + \alpha_2 + \alpha_3 = 0 \tag{7.1.5-i}$$

$$h^2 f_n^{(1)}: \quad -1 + 2\alpha_2 + \alpha_3 = 0 \tag{7.1.5-ii}$$

$$h^3 f_n^{(2)}: \quad -2 + 6\alpha_2 + 3\alpha_3 = 0 \tag{7.1.5-iii}$$

By solving (7.1.5-i) and (7.1.5-ii) using Mathematica , we have

$$\alpha_1 = \frac{1-\alpha_3}{2} \qquad and \qquad \alpha_2 = \frac{1-\alpha_3}{2} \tag{7.1.6}$$

or by solving equation (7.1.5-i) and (7.1.5-iii) by Mathematica , we obtain

$$\alpha_1 = \frac{4-2\alpha_3}{6} \qquad and \qquad \alpha_2 = \frac{2-3\alpha_3}{6} \tag{7.1.7}$$

By substituting $\alpha_3 = \beta$ approximately in (7.1.6) and (7.1.7) , we have

$$\alpha_1 = \frac{1-\beta}{2} \qquad and \qquad \alpha_2 = \frac{1-\beta}{2} \tag{7.1.8}$$

and $\alpha_1 = \dfrac{4-2\beta}{6} \qquad and \qquad \alpha_2 = \dfrac{2-3\beta}{6}$ (7.1.9)

When we subtitute equation (7.1.8) into (7.1.1), we obtain

$$y_{n+1} - y_n = h\left\{\left(\frac{1-\beta}{2}\right)f_n + \left(\frac{1-\beta}{2}\right)f_{n+1} + \beta\left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right)\right\} \qquad (7.1.10)$$

where β is the parameter defined as α_3 .

By subtituting $\beta = 0$, the well known Trapezoidal formula (AM) can be obtained as

$$y_{n+1} - y_n = \frac{h}{2}[f_n + f_{n+1}] \qquad (7.1.11)$$

Similarly, the original C_oM formula can be derived from (7.1.10) by putting $\beta = 1$ to obtain

$$y_{n+1} - y_n = h\left[\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right] \qquad (7.1.12)$$

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

$$\begin{aligned} LTE^{AM} &= \left(\frac{\alpha_2}{2} - \frac{1}{6}\right)h^3 f_n^{(2)} \\ &= \frac{h^3}{12}(6\alpha_2 - 2)f_n^{(2)} \\ &= \frac{h^3}{12} f_n^{(2)} \qquad (7.1.13) \end{aligned}$$

By subtituting $\alpha_1 = \alpha_2 = \beta$, then $\alpha_3 = 1 - 2\beta$ into equation (7.1.1), we have

$$y_{n+1} - y_n = h\left\{\beta(f_n + f_{n+1}) + (1-2\beta)\left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right)\right\} \qquad (7.1.14)$$

By substituting $\beta = \frac{1}{2}$, 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 C_oM in the form equation (7.1.12).

The LTE with $\beta = \frac{1}{2}$, $\alpha_3 = 0$ for equation (7.1.14) is

$$LTE^{AM} = -\frac{h^3}{6} f_n^{(2)} + \frac{h^3}{4} f_n^{(2)}$$

$$= \frac{h^3}{12} f_n^{(2)} \qquad (7.1.15)$$

and LTE with $\beta = 0$ *and* $\alpha_3 = 1$ in equation (7.1.14) is given by

$$LTE^{C_oM} = (-\tfrac{1}{6} + \tfrac{1}{4})h^3 f_n^{(2)} + \frac{\alpha_3}{4} \frac{\left(f_n^{(1)}\right)^2}{f_n}$$

$$= \frac{h^3}{12}\left(f_n^{(2)} + 3\frac{\left(f_n^{(1)}\right)^2}{f_n}\right) \qquad (7.1.16)$$

7.2 DERIVATION OF THE C_oM METHOD FOR PROBLEMS OF THE TYPE $y^{(2)} = f(x, y)$

Evans and Jayes [1993] have shown that for the case of p=2 the GM method developed in the form

$$y_{n+1} - 2y_n + y_{n-1} = h^2\{\alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4 \sqrt{f_n f_{n-1}}$$
$$+ \alpha_5 \sqrt{f_n f_{n+1}} + \alpha_6 \sqrt{f_{n-1} f_{n+1}}\} \qquad (7.2.1)$$

A fourth order accuracy is obtained through the standard procedure of adjustment of the parameters $\alpha_i, 1 \le i \le 6$ for formula (7.2.1) where

$$\alpha_1 = \frac{1}{6}(5 + 12\alpha)$$

$$\alpha_2 = \alpha_3 = \frac{1}{12}(1 + 6\alpha) \qquad (7.2.2)$$

$$\alpha_4 = \alpha_5 = -2\alpha$$

$$\alpha_6 = \alpha$$

The GM formula is obtained by substituting $\alpha = -\frac{1}{6}$ into (7.2.2) and from (7.2.1) we obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{2}\left\{3f_n + 2\left[\sqrt{f_n f_{n+1}} + \sqrt{f_n f_{n+1}}\right] - \sqrt{f_{n-1} f_{n+1}}\right\} \qquad (7.2.3)$$

and another form of the GM formula was obtained by substituting $\alpha = -\frac{5}{12}$ in (7.2.2) and from (7.2.10) to give

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{2}\left\{10\left[2\left(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}\right) - \sqrt{f_{n-1} f_{n+1}}\right] - 3\left(f_{n-1} + f_{n+1}\right)\right\} \qquad (7.2.4)$$

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

$$y_{n+1} - 2y_n + y_{n-1} = h^2\left\{\alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4\left[\frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}}\right] + \alpha_5\left[\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right] + \alpha_6\left[\frac{f_{n-1}^2 + f_{n+1}^2}{f_{n-1} + f_{n+1}}\right]\right\} \qquad (7.2.5)$$

Consider the left hand side of equation (7.2.5). The Taylor series of y_{n+1} and y_{n-1} at x_n give

$$y_{n+1} - y_n = hy_n^{(1)} + \frac{h^2}{2!}y_n^{(2)} + \frac{h^3}{3!}y_n^{(3)} + \frac{h^4}{4!}y_n^{(4)} + \frac{h^5}{5!}y_n^{(5)} + \frac{h^6}{6!}y_n^{(6)} + 0\left(h^7\right) \qquad (7.2.6)$$

$$y_{n-1} - y_n = -hy_n^{(1)} + \frac{h^2}{2!}y_n^{(2)} - \frac{h^3}{3!}y_n^{(3)} + \frac{h^4}{4!}y_n^{(4)} - \frac{h^5}{5!}y_n^{(5)} + \frac{h^6}{6!}y_n^{(6)} + 0\left(h^7\right) \qquad (7.2.7)$$

From equations (7.2.6) and (7.2.7), we have

$$y_{n+1} + y_{n-1} - 2y_n = h^2 f_n + \frac{h^4}{12}f_n^{(2)} + \frac{h^6}{360}f_n^{(4)} + 0\left(h^8\right) \qquad (7.2.8)$$

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

$$f_{n+1} = f_n + hf_n^{(1)} + \frac{h^2}{2!}f_n^{(2)} + \frac{h^3}{3!}f_n^{(3)} + \frac{h^4}{4!}f_n^{(4)} + \frac{h^5}{5!}f_n^{(5)} + 0\left(h^6\right) \qquad (7.2.9)$$

and

$$f_{n-1} = f_n - hf_n^{(1)} + \frac{h^2}{2!}f_n^{(2)} - \frac{h^3}{3!}f_n^{(3)} + \frac{h^4}{4!}f_n^{(4)} - \frac{h^5}{5!}f_n^{(5)} + 0\left(h^6\right) \qquad (7.2.10)$$

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 :

$$\alpha_2 f_{n-1} = \alpha_2 f_n\left[1-\frac{hf_n^{(1)}}{f_n}+\frac{h^2 f_n^{(2)}}{2!f_n}-\frac{h^3 f_n^{(3)}}{3!f_n}+\frac{h^4 f_n^{(4)}}{4!f_n}-\frac{h^5 f_n^{(5)}}{5!f_n}\right]+0(h^6) \qquad (7.2.11)$$

$$\alpha_3 f_{n+1} = \alpha_3 f_n\left[1+\frac{hf_n^{(1)}}{f_n}+\frac{h^2 f_n^{(2)}}{2!f_n}+\frac{h^3 f_n^{(3)}}{3!f_n}+\frac{h^4 f_n^{(4)}}{4!f_n}+\frac{h^5 f_n^{(5)}}{5!f_n}\right]+0(h^6) \qquad (7.2.12)$$

$$\alpha_4\left[\frac{f_n^2+f_{n-1}^2}{f_n+f_{n-1}}\right] = \alpha_4 f_n\left[1-\frac{hf_n^{(1)}}{2f_n}+\frac{h^2}{4}\left(\left(\frac{f_n^{(1)}}{f_n}\right)^2+\frac{f_n^{(2)}}{f_n}\right)\right.$$

$$+\frac{h^3}{24}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^3-6\frac{f_n^{(1)}f_n^{(2)}}{(f_n)^2}-2\frac{f_n f_n^{(3)}}{(f_n)^2}\right)$$

$$+\frac{h^4}{48}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^4-9\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{(f_n)^3}+3\frac{f_n\left(f_n^{(2)}\right)^2}{(f_n)^3}+4\frac{f_n f_n^{(1)} f_n^{(3)}}{(f_n)^3}+\frac{(f_n)^2 f_n^{(4)}}{(f_n)^3}\right)$$

$$+\frac{h^5}{480}\left(15\left(\frac{f_n^{(1)}}{f_n}\right)^5-60\frac{\left(f_n^{(1)}\right)^3 f_n^{(2)}}{(f_n)^4}+45\frac{f_n f_n^{(1)}\left(f_n^{(2)}\right)^2}{(f_n)^4}+30\frac{f_n\left(f_n^{(1)}\right)^2 f_n^{(3)}}{(f_n)^4}\right.$$

$$\left.-20\frac{(f_n)^2 f_n^{(2)} f_n^{(3)}}{(f_n)^4}-10\frac{(f_n)^2 f_n^{(1)} f_n^{(4)}}{(f_n)^4}-2\frac{(f_n)^3 f_n^{(5)}}{(f_n)^4}\right)+0(h^6) \qquad (7.2.13)$$

$$\alpha_5\left[\frac{f_n^2+f_{n+1}^2}{f_n+f_{n+1}}\right] = \alpha_5 f_n\left[1+\frac{hf_n^{(1)}}{2f_n}+\frac{h^2}{4}\left(\left(\frac{f_n^{(1)}}{f_n}\right)^2+\frac{f_n^{(2)}}{f_n}\right)\right.$$

$$+\frac{h^3}{24}\left(-3\left(\frac{f_n^{(1)}}{f_n}\right)^3+6\frac{f_n^{(1)}f_n^{(2)}}{(f_n)^2}+2\frac{f_n f_n^{(3)}}{(f_n)^2}\right)$$

$$+\frac{h^4}{48}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^4-9\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{(f_n)^3}+3\frac{f_n\left(f_n^{(2)}\right)^2}{(f_n)^3}+4\frac{f_n f_n^{(1)} f_n^{(3)}}{(f_n)^3}+\frac{(f_n)^2 f_n^{(4)}}{(f_n)^3}\right)$$

$$+\frac{h^5}{480}\left(-15\left(\frac{f_n^{(1)}}{f_n}\right)^5+60\frac{\left(f_n^{(1)}\right)^3 f_n^{(2)}}{\left(f_n\right)^4}-45\frac{f_n f_n\left(f_n^{(2)}\right)^2}{\left(f_n\right)^4}-30\frac{f_n\left(f_n^{(1)}\right)^2 f_n^{(3)}}{\left(f_n\right)^4}\right.$$

$$\left.+20\frac{\left(f_n\right)^2 f_n^{(2)} f_n^{(3)}}{\left(f_n\right)^4}+10\frac{\left(f_n\right)^2 f_n^{(1)} f_n^{(4)}}{\left(f_n\right)^4}+2\frac{\left(f_n\right)^3 f_n^{(5)}}{\left(f_n\right)^4}\right)\Bigg]+0\left(h^6\right) \quad (7.2.14)$$

$$\alpha_6\left[\frac{f_{n-1}^2+f_{n+1}^2}{f_{n-1}+f_{n+1}}\right]=\alpha_6 f_n\left[1+\frac{h^2}{2}\left(2\left(\frac{f_n^{(1)}}{f_n}\right)^2+\frac{f_n^{(2)}}{f_n}\right)\right.$$

$$\left.+\frac{h^4}{24}\left(-12\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{\left(f_n\right)^3}+8\frac{f_n^{(1)} f_n^{(3)}}{\left(f_n\right)^2}+\frac{f_n^{(4)}}{f_n}\right)\right]+0\left(h^6\right) \quad (7.2.15)$$

By substituting equations (7.2.11)-(7.2.15) into equation (7.3.5), we obtain

$$h^2 f_n\left\{\sum_{i=1}^{6}\alpha_i+\left(-2\alpha_2+2\alpha_3-\alpha_4+\alpha_5\right)\frac{hf_n^{(1)}}{2f_n}+\left(2\alpha_2+2\alpha_3+\alpha_4+\alpha_5+2\alpha_6\right)\frac{h^2 f_n^{(2)}}{4f_n}\right.$$

$$+\left(\alpha_4+\alpha_5+4\alpha_6\right)\frac{h^2}{4}\left[\frac{f_n^{(1)}}{f_n}\right]^2-\frac{1}{12}h^2 f_n^{(3)}+\left(-2\alpha_2+2\alpha_3-\alpha_4+\alpha_5\right)\frac{h^3 f_n^{(3)}}{12 f_n}$$

$$\left.+\left(\alpha_4-\alpha_5\right)\frac{h^3}{8}\left[\frac{f_n^{(1)}}{f_n}\right]^3+\left(-\alpha_4+\alpha_5\right)\frac{h^3 f_n^{(1)} f_n^{(2)}}{4\left[f_n\right]^2}\right\}+0\left(h^6\right) \quad (7.2.16)$$

Therefore, by equating equations (7.2.8) and (7.2.16), we obtain the following equations of condition, i.e.,

$$h^2 f_n \quad : \quad -1+\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6=0 \quad (7.2.17\text{-i})$$

$$h^3 f_n^{(1)} \quad : \quad -2\alpha_2+2\alpha_3-\alpha_4+\alpha_5=0 \quad (7.2.17\text{-ii})$$

$$h^4 f_n^{(2)} \quad : \quad -1+6\alpha_2+6\alpha_3+3\alpha_4+3\alpha_5+6\alpha_6=0 \quad (7.2.17\text{-iii})$$

$$\frac{h^4\left[f_n^{(1)}\right]^2}{f_n} \quad : \quad \alpha_4+\alpha_5+4\alpha_6=0 \quad (7.2.17\text{-iv})$$

$$h^5 f_n^{(3)} \quad : \quad -2\alpha_2+2\alpha_3-\alpha_4+\alpha_5=0 \quad (7.2.17\text{-v})$$

$$\frac{h^5\left[f_n^{(1)}\right]^3}{\left(f_n\right)^2} \quad : \quad \alpha_4-\alpha_5=0 \quad (7.2.17\text{-vi})$$

$$\frac{h^5 f_n^{(1)} f_n^{(2)}}{f_n} \quad : \quad -\alpha_4 + \alpha_5 = 0 \qquad (7.2.17\text{-vii})$$

For equation (7.2.17-ii), (7.2.17-v) and (7.2.17-vi) to be symmetric, we impose the condition that

$$-\alpha_2 + \alpha_3 = 0 \qquad (7.2.17\text{-viii})$$

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

$$\alpha_1 = \frac{5}{6} + 2\alpha_6 \quad , \quad \alpha_2 = \alpha_3 = \frac{1}{12} + \frac{\alpha_6}{2} \quad , \quad \alpha_4 = \alpha_5 = -2\alpha_6 \qquad (7.2.18)$$

By substituting $\alpha_6 = \beta$ into (7.2.18) and from (7.2.5), we obtain

$$y_{n+1} = 2y_n - y_{n-1} + h^2\left\{\left(\frac{5}{6} + 2\beta\right)f_n + \left(\frac{1}{12} + \frac{\beta}{2}\right)(f_{n-1} + f_{n+1})\right.$$

$$\left. -2\beta\left[\left(\frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}}\right) + \left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right)\right] + \beta\left(\frac{f_{n-1}^2 + f_{n+1}^2}{f_{n-1} + f_{n+1}}\right)\right\} \qquad (7.2.19)$$

where β is an arbitrary constant .
From equation (7.2.19), by substituting $\beta = 0$, gives the Numerov (AM) method as

$$y_{n+1} = 2y_n - y_{n-1} + h^2\left[\frac{5}{6}f_n + \frac{1}{12}(f_{n-1} + f_{n+1})\right]$$

$$= 2y_n - y_{n-1} + \frac{h^2}{12}\left[10f_n + f_{n-1} + f_{n+1}\right] \qquad (7.2.20)$$

and the C_oM formula is obtained by substituting $\beta = \frac{1}{144}$ into equation (7.2.19) to obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288}\left\{244f_n + 25(f_{n-1} + f_{n+1})\right.$$

$$\left. -4\left[\left(\frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}}\right) + \left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right)\right] + 2\left(\frac{f_{n-1}^2 + f_{n+1}^2}{f_{n-1} + f_{n+1}}\right)\right\} \qquad (7.2.21)$$

By substituting $\beta = -\frac{1}{144}$ into equation (7.2.19) we obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288}\{236 f_n + 23(f_{n-1} + f_{n+1})$$

$$+4\left[\left(\frac{f_n^2 + f_{n-1}^2}{f_n + f_{n-1}}\right) + \left(\frac{f_n^2 + f_{n+1}^2}{f_n + f_{n+1}}\right)\right] - 2\left(\frac{f_{n-1}^2 + f_{n+1}^2}{f_{n-1} + f_{n+1}}\right)\Bigg\} \quad (7.2.22)$$

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

$$LTE^{C_oM} = h^6\Bigg\{\frac{1}{720}(-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)}$$

$$+\frac{1}{12}(\alpha_4 + \alpha_5 + 4\alpha_6)\frac{f_n^{(1)} f_n^{(3)}}{f_n} + \frac{1}{16}(\alpha_4 + \alpha_5)\frac{\left(f_n^{(2)}\right)^2}{f_n}$$

$$+\frac{1}{16}(-3\alpha_4 - 3\alpha_5 - 8\alpha_6)\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{(f_n)^2} + \frac{1}{16}(\alpha_4 + \alpha_5)\frac{\left(f_n^{(1)}\right)^4}{(f_n)^3}\Bigg) \quad (7.2.23)$$

$$= \frac{h^6}{720}\{(-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)}$$

$$+60(\alpha_4 + \alpha_5 + 4\alpha_6)\frac{f_n^{(1)} f_n^{(3)}}{f_n} + 45(\alpha_4 + \alpha_5)\frac{\left(f_n^{(2)}\right)^2}{f_n}$$

$$+45(-3\alpha_4 - 3\alpha_5 - 8\alpha_6)\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{(f_n)^2} + 45(\alpha_4 + \alpha_5)\frac{\left(f_n^{(1)}\right)^4}{(f_n)^3}\Bigg) \quad (7.2.24)$$

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

$$LTE^{C_oM} = \frac{h^6}{720}\left\{3f_n^{(4)} - 180\beta\frac{\left(f_n^{(2)}\right)^2}{f_n} + 180\beta\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{\left(f_n\right)^2} - 180\beta\frac{\left(f_n^{(4)}\right)^4}{\left(f_n\right)^3}\right\}$$

$$= \frac{h^6}{720}\left\{3f_n^{(4)} + 180\beta\left[\frac{f_n f_n^{(2)}\left(f_n^{(1)}\right)^2 - \left(f_n\right)^2\left(f_n^{(2)}\right)^2 - \left(f_n^{(1)}\right)^4}{\left(f_n\right)^3}\right]\right\} \qquad (7.2.25)$$

and we define

$$Mn = \frac{f_n f_n^{(2)}\left(f_n^{(1)}\right)^2 - \left(f_n\right)^2\left(f_n^{(2)}\right)^2 - \left(f_n^{(1)}\right)^4}{\left(f_n\right)^3} \quad .$$

For $\beta = 0$, the local truncation error (LTE) for Numerov (AM) method in equation (7.2.20) is given as

$$LTE^{AM} = \frac{h^6}{240} f_n^{(4)} . \qquad (7.2.26)$$

For $\beta = \frac{1}{144}$, the local truncation error (LTE) for the C_oM formula in equation (7.2.21) is given by

$$LTE^{C_oM} = \frac{h^6}{2880}\left\{12 f_n^{(4)} + 5Mn\right\} \qquad (7.2.27)$$

and by substituting $\beta = -\frac{1}{144}$ in equation (7.2.25) the local truncation error (LTE) for the C_oM formula in (7.2.22) is given as

$$LTE^{C_oM} = \frac{h^6}{2880}\left\{12 f_n^{(4)} - 5Mn\right\}. \qquad (7.2.28)$$

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 an accuracy between these methods are shown in Tables (7.1)-(7.5).

Problem 1: $y'' + xy = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y'_o = 2$

Exact Solution : $y = \left(1 - \frac{x^3}{3} + \frac{x^6}{180} - \ldots\right) + 2\left(x - \frac{x^4}{12} + \frac{x^7}{504} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.119965000595E+01	.119965000595E+01	.000000000000E+00
.20	.139806708690E+01(X) .139807014100E+01(Y) .139806403280E+01(Z)	.139706707302E+01	.715795183538E-03 .717981261443E-03 .713609105633E-03
.30	.159365496416E+01(X) .159366166169E+01(Y) .159364826663E+01(Z)	.158965491786E+01	.251629850849E-02 .252051170910E-02 .251208531616E-02
.40	.178442927854E+01(X) .178443985924E+01(Y) .178441869788E+01(Z)	.177442925714E+01	.563562698202E-02 .564158985500E-02 .562966412911E-02
.50	.196803408010E+01(X) .196804864796E+01(Y) .196801951229E+01(Z)	.194803447421E+01	.102665564462E-01 .102740346854E-01 .102590782401E-01
.60	.214176811045E+01(X) .214178668501E+01(Y) .214174953598E+01(Z)	.210677028571E+01	.166120744014E-01 .166208910065E-01 .166032578425E-01
.70	.230262257646E+01(X) .230264510783E+01(Y) .230260004523E+01(Z)	.224663040833E+01	.249227322500E-01 .249327612106E-01 .249127033487E-01
.80	.244733187537E+01(X) .244735824775E+01(Y) .244730550316E+01(Z)	.236335522540E+01	.355328090634E-01 .355439679374E-01 .355216502615E-01
.90	.257243840106E+01(X) .257246843201E+01(Y) .257240837031E+01(Z)	.245250045357E+01	.489043528245E-01 .489165978591E-01 .488921078748E-01
1.00	.267437213004E+01(X) .267440556832E+01(Y) .267433869201E+01(Z)	.250952380952E+01	.656890840780E-01 .657024086297E-01 .656757596241E-01

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 C_oM formula (7.2.21)

Z denote formula C_oM formula (7.2.22)

Problem 2: $y'' + 2x^2 y = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y_o^{(1)} = 1$

Exact Solution : $y = \left(1 - \frac{x^4}{6} + \frac{x^8}{168} - \ldots\right) + \left(x - \frac{x^5}{10} + \frac{x^9}{360} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.109998233340E+01	.109998233340E+01	.000000000000E+00
.20	.119970133876E+01(X) .119970275584E+01(Y) .119969992168E+01(Z)	.119970134999E+01	.936544166655E-08 .117183061999E-05 .119056150351E-05
.30	.129840729714E+01(X) .129841126327E+01(Y) .129840333101E+01(Z)	.129840744521E+01	.114042419673E-06 .294057112078E-05 .316864816689E-05
.40	.139471310095E+01(X) .139472036629E+01(Y) .139470583565E+01(Z)	.139471396246E+01	.617701420252E-06 .459149460478E-05 .582687034057E-05
.50	.148648365085E+01(X) .148649473480E+01(Y) .148647256698E+01(Z)	.148648701017E+01	.225990528824E-05 .519656853923E-05 .971632141034E-05
.60	.157074164286E+01(X) .157075689858E+01(Y) .157072638729E+01(Z)	.157075197074E+01	.657512117010E-05 .313724479653E-05 .162873889669E-04
.70	.164360423379E+01(X) .164362386518E+01(Y) .164358460263E+01(Z)	.164363156960E+01	.166313519292E-04 .468743749893E-05 .285751193898E-04
.80	.170027071388E+01(X) .170029476769E+01(Y) .170024666040E+01(Z)	.170033680417E+01	.388689400721E-04 .247224373692E-04 .530152399252E-04
.90	.173508683101E+01(X) .173511517526E+01(Y) .173505848723E+01(Z)	.173523947285E+01	.879658605241E-04 .716313756378E-04 .104300080438E-03
1.00	.174171581559E+01(X) .174174811168E+01(Y) .174168352008E+01(Z)	.174206349206E+01	.199577383509E-03 .181038400430E-03 .218116033568E-03

Table 7.2

Problem 3: $y^{(2)} + x^2 y = 1 + x + x^2$

Initial values : $x_o = 0$, $y_o = 2$, $y_o^{(1)} = 2$

Exact Solution : $y = 2\left(1 - \frac{x^4}{12} + \frac{x^8}{672} - \ldots\right) + 2\left(x - \frac{x^5}{20} + \frac{x^9}{1440} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.220515731629E+01	.220515731629E+01	.000000000000E+00
.20	.242116719231E+01(X) .242116717267E+01(Y) .242116721194E+01(Z)	.242116688706E+01	.126074661035E-06 .117965271145E-06 .134184050926E-06
.30	.264857051467E+01(X) .264857044271E+01(Y) .264857058664E+01(Z)	.264856910711E+01	.531441912139E-06 .504270769870E-06 .558613056588E-06
.40	.288743990707E+01(X) .288743972700E+01(Y) .288744008714E+01(Z)	.288743590441E+01	.138623499026E-05 .132387036964E-05 .144859962533E-05
.50	.313722608903E+01(X) .313722569779E+01(Y) .313722648028E+01(Z)	.313721710689E+01	.286309062579E-05 .273837975232E-05 .298780156579E-05
.60	.339659195430E+01(X) .339659111444E+01(Y) .339659279418E+01(Z)	.339657430537E+01	.519609809508E-05 .494882859352E-05 .544336791383E-05
.70	.366323827217E+01(X) .366323603200E+01(Y) .366324051235E+01(Z)	.366320554629E+01	.893367162527E-05 .832213777637E-05 .954520814619E-05
.80	.393372811772E+01(X) .393370393028E+01(Y) .393375230618E+01(Z)	.393366680462E+01	.155867549444E-04 .943792519809E-05 .217358423771E-04
.90	.420332107723E+01(X) .420312309456E+01(Y) .420351905588E+01(Z)	.420319930665E+01	.289709273196E-04 .181319240924E-04 .760728224634E-04
1.00	.446583286944E+01(X) .446547279445E+01(Y) .446619293003E+01(Z)	.446557539683E+01	.576572081016E-04 .229762941082E-04 .138287488676E-03

Table 7.3

Problem 4: $y^{(2)} - y = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y_o^{(1)} = -1$

Exact Solution : $y = \exp(-x)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.904837418036E+00	.904837418036E+00	.000000000000E+00
.20	.818723210252E+00(X)	.818730753078E+00	.921282811814E-05
	.818723208325E+00(Y)		.921518204101E-05
	.818723212180E+00(Z)		.921047419541E-05
.30	.740796234571E+00(X)	.740818220682E+00	.296781450339E-04
	.740796228953E+00(Y)		.296857288372E-04
	.740796240189E+00(Z)		.296705612313E-04
.40	.670277221236E+00(X)	.670320046036E+00	.638870942600E-04
	.670277210291E+00(Y)		.639034216655E-04
	.670277232180E+00(Z)		.638707668567E-04
.50	.606460980113E+00(X)	.606530659713E+00	.114882238736E-03
	.606460962302E+00(Y)		.114911602615E-03
	.606460997923E+00(Z)		.114852874862E-03
.60	.548709348791E+00(X)	.548811636094E+00	.186379618579E-03
	.548709322642E+00(Y)		.186427263684E-03
	.548709374939E+00(Z)		.186331973487E-03
.70	.496444810957E+00(X)	.496585303791E+00	.282917826556E-03
	.496444775037E+00(Y)		.282990160538E-03
	.496444846877E+00(Z)		.282845492598E-03
.80	.449144721232E+00(X)	.449328964117E+00	.410040081691E-03
	.449144674119E+00(Y)		.410144932908E-03
	.449144768345E+00(Z)		.409935230517E-03
.90	.406336078720E+00(X)	.406569659741E+00	.574516605226E-03
	.406336018982E+00(Y)		.574663537699E-03
	.406336138458E+00(Z)		.574369672825E-03
1.00	.367590796995E+00(X)	.367879441171E+00	.784616219919E-03
	.367590723162E+00(Y)		.784816918115E-03
	.367590870828E+00(Z)		.784415521840E-03

Table 7.4

Problem 5: $y^{(2)} - y\left\{\left[(Q + Bx) / x\right]^2 - Q / x^2\right\} = 0$

Initial values : $x_o = 1$, $y_o = 10e$, $y_o^{(1)} = 10e(Q + B)$

Exact Solution : $y = Cx^Q e^{Bx}$

$$\left(Set \quad B = 1 , C = 10 , Q = \frac{3}{2}\right)$$

xn	Numerical Solution	Exact Solution	Absolute Error
1.10	.346587549802E+02	.346587549802E+02	.000000000000E+00
1.20	.436390152385E+02(X) .436390125972E+02(Y) .436390178797E+02(Z)	.436440703713E+02	.115826337471E-03 .115886854901E-03 .115765820041E-03
1.30	.543715401009E+02(X) .543715320262E+02(Y) .543715481755E+02(Z)	.543873445416E+02	.290590410221E-03 .290738875845E-03 .290441944473E-03
1.40	.671413377291E+02(X) .671413211479E+02(Y) .671413543102E+02(Z)	.671744823128E+02	.493410333302E-03 .493657170275E-03 .493163495968E-03
1.50	.822756510828E+02(X) .822756225211E+02(Y) .822756796445E+02(Z)	.823338855610E+02	.707296611036E-03 .707643511888E-03 .706949709502E-03
1.60	.100149815184E+03(X) .100149770648E+03(Y) .100149859721E+03(Z)	.100242328229E+03	.922894012738E-03 .923338299662E-03 .922449724767E-03
1.70	.121193894010E+03(X) .121193828860E+03(Y) .121193959159E+03(Z)	.121331621407E+03	.113513192929E-02 .113566888199E-02 .113459497516E-02
1.80	.145900197789E+03(X) .145900106610E+03(Y) .145900288968E+03(Z)	.146096168080E+03	.134137871608E-02 .134200281759E-02 .134075461276E-02
1.90	.174831793908E+03(X) .174831670357E+03(Y) .174831917459E+03(Z)	.175101520393E+03	.154040058834E-02 .154110618393E-02 .153969499054E-02
2.00	.208632138690E+03(X) .208631975333E+03(Y) .208632302047E+03(Z)	.208994066965E+03	.173176339644E-02 .173254503037E-02 .173098175995E-02

Table 7.5

7.3 DERIVATION OF THE C_eM METHOD FOR PROBLEMS OF THE TYPE $y^{(2)} = f(x,y)$

By carrying out a similar procedure as the contraharmonic mean (C_oM) method for solving problems of the type $y^{(2)} = f(x,y)$, a new centroidal mean (C_eM) can be established as

$$y_{n+1} - 2y_n + y_{n-1} = h^2\left\{\alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4\left[\frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})}\right]\right.$$

$$\left. + \alpha_5\left[\frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})}\right] + \alpha_6\left[\frac{2(f_{n-1}^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_{n-1} + f_{n+1})}\right]\right\} \qquad (7.3.1)$$

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 :

$$\alpha_4\left[\frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})}\right] = \alpha_4 f_n\left[1 - \frac{hf_n^{(1)}}{2f_n} + \frac{h^2}{12}\left(\left(\frac{f_n^{(1)}}{f_n}\right)^2 + \frac{3f_n^{(2)}}{f_n}\right)\right.$$

$$+ \frac{h^3}{24}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^3 - 2\frac{f_n^{(1)}f_n^{(2)}}{(f_n)^2} - 2\frac{f_n f_n^{(3)}}{(f_n)^2}\right)$$

$$+ \frac{h^4}{144}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^4 - 9\frac{(f_n^{(1)})^2 f_n^{(2)}}{(f_n)^3} + 3\frac{f_n(f_n^{(2)})^2}{(f_n)^3} + 4\frac{f_n f_n^{(1)} f_n^{(3)}}{(f_n)^3} + \frac{4(f_n)^2 f_n^{(4)}}{(f_n)^3}\right)$$

$$+ \frac{h^5}{1440}\left(15\left(\frac{f_n^{(1)}}{f_n}\right)^5 - 60\frac{(f_n^{(1)})^3 f_n^{(2)}}{(f_n)^4} + 45\frac{f_n f_n^{(1)}(f_n^{(2)})^2}{(f_n)^4} + 30\frac{f_n(f_n^{(1)})^2 f_n^{(3)}}{(f_n)^4}\right.$$

$$\left.\left. -20\frac{(f_n)^2 f_n^{(2)} f_n^{(3)}}{(f_n)^4} - 10\frac{(f_n)^2 f_n^{(1)} f_n^{(4)}}{(f_n)^4} - 6\frac{(f_n)^3 f_n^{(5)}}{(f_n)^4}\right) + 0(h^6)\right. \qquad (7.3.2)$$

$$\alpha_5\left[\frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})}\right] = \alpha_5 f_n\left[1 + \frac{hf_n^{(1)}}{2f_n} + \frac{h^2}{12}\left(\left(\frac{f_n^{(1)}}{f_n}\right)^2 + \frac{3f_n^{(2)}}{f_n}\right)\right.$$

$$+\frac{h^3}{24}\left(-\left(\frac{f_n^{(1)}}{f_n}\right)^3+2\frac{f_n^{(1)}f_n^{(2)}}{(f_n)^2}+2\frac{f_nf_n^{(3)}}{(f_n)^2}\right)$$

$$+\frac{h^4}{144}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^4-9\frac{\left(f_n^{(1)}\right)^2f_n^{(2)}}{(f_n)^3}+3\frac{f_n\left(f_n^{(2)}\right)^2}{(f_n)^3}+4\frac{f_nf_n^{(1)}f_n^{(3)}}{(f_n)^3}+\frac{3(f_n)^2f_n^{(4)}}{(f_n)^3}\right)$$

$$+\frac{h^5}{1440}\left(-15\left(\frac{f_n^{(1)}}{f_n}\right)^5+60\frac{\left(f_n^{(1)}\right)^3f_n^{(2)}}{(f_n)^4}-45\frac{f_nf_n\left(f_n^{(2)}\right)^2}{(f_n)^4}-30\frac{f_n\left(f_n^{(1)}\right)^2f_n^{(3)}}{(f_n)^4}\right.$$

$$\left.\left.+20\frac{(f_n)^2f_n^{(2)}f_n^{(3)}}{(f_n)^4}+10\frac{(f_n)^2f_n^{(1)}f_n^{(4)}}{(f_n)^4}+6\frac{(f_n)^3f_n^{(5)}}{(f_n)^4}\right)\right]+0\left(h^6\right) \qquad (7.3.3)$$

$$\alpha_6\left[\frac{2\left(f_{n-1}^2+f_{n-1}f_{n+1}+f_{n+1}^2\right)}{3\left(f_{n-1}+f_{n+1}\right)}\right]=\alpha_6f_n\left[1+\frac{h^2}{6}\left(2\left(\frac{f_n^{(1)}}{f_n}\right)^2+\frac{3f_n^{(2)}}{f_n}\right)\right.$$

$$\left.+\frac{h^4}{72}\left(-12\frac{\left(f_n^{(1)}\right)^2f_n^{(2)}}{(f_n)^3}+8\frac{f_n^{(1)}f_n^{(3)}}{(f_n)^2}+3\frac{f_n^{(4)}}{f_n}\right)\right]+0\left(h^6\right) \qquad (7.3.4)$$

By substituting equations (7.2.11)-(7.2.12) and (7.3.2)-(7.3.4) into equation (7.3.1) , we obtain

$$h^2f_n\left\{\sum_{i=1}^{6}\alpha_i+\left(-2\alpha_2+2\alpha_3-\alpha_4+\alpha_5\right)\frac{hf_n^{(1)}}{2f_n}+\left(2\alpha_2+2\alpha_3+\alpha_4+\alpha_5+2\alpha_6\right)\frac{h^2f_n^{(2)}}{4f_n}\right.$$

$$+\left(\alpha_4+\alpha_5+4\alpha_6\right)\frac{h^2}{12}\left[\frac{f_n^{(1)}}{f_n}\right]^2+\left(-2\alpha_2+2\alpha_3-\alpha_4+\alpha_5\right)\frac{h^3f_n^{(3)}}{12f_n}$$

$$\left.+\left(\alpha_4-\alpha_5\right)\frac{h^3}{24}\left[\frac{f_n^{(1)}}{f_n}\right]^3+\left(-\alpha_4+\alpha_5\right)\frac{h^3f_n^{(1)}f_n^{(2)}}{12[f_n]^2}\right\}+0\left(h^6\right) \qquad (7.3.5)$$

On equating equations (7.2.8) and (7.3.5), we obtain the following equations of condition , i.e.,

$$h^2f_n \quad : \quad -1+\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6=0 \qquad (7.3.6\text{-i})$$

$$h^3f_n^{(1)} \quad : \quad -2\alpha_2+2\alpha_3-\alpha_4+\alpha_5=0 \qquad (7.3.6\text{-ii})$$

$$h^4f_n^{(2)} \quad : \quad -1+6\alpha_2+6\alpha_3+3\alpha_4+3\alpha_5+6\alpha_6=0 \qquad (7.3.6\text{-iii})$$

$$\frac{h^4\left[f_n^{(1)}\right]^2}{f_n} \quad : \quad \alpha_4+\alpha_5+4\alpha_6=0 \tag{7.3.6-iv}$$

$$h^5 f_n^{(3)} \quad : \quad -2\alpha_2+2\alpha_3-\alpha_4+\alpha_5=0 \tag{7.3.6-v}$$

$$\frac{h^5\left[f_n^{(1)}\right]^3}{(f_n)^2} \quad : \quad \alpha_4-\alpha_5=0 \tag{7.3.6-vi}$$

$$\frac{h^5 f_n^{(1)} f_n^{(2)}}{f_n} \quad : \quad -\alpha_4+\alpha_5=0 \tag{7.3.6-vii}$$

For equation (7.3.6-ii), (7.3.6-v) and (7.3.6-vi) to be symmetric, we impose the condition that

$$-\alpha_2+\alpha_3=0 \tag{7.3.6-viii}$$

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

$$\alpha_1=\frac{5}{6}+2\alpha_6 \quad , \quad \alpha_2=\alpha_3=\frac{1}{12}+\frac{\alpha_6}{2} \quad , \quad \alpha_4=\alpha_5=-2\alpha_6 \tag{7.3.7}$$

By substituting $\alpha_6=\beta$ into (7.3.7) and from (7.3.1), we obtain the equation

$$y_{n+1}=2y_n-y_{n-1}+h^2\left\{\left(\frac{5}{6}+2\beta\right)f_n+\left(\frac{1}{12}+\frac{\beta}{2}\right)(f_{n-1}+f_{n+1})\right.$$

$$\left.-2\beta\left[\left(\frac{2\left(f_n^2+f_nf_{n-1}+f_{n-1}^2\right)}{3(f_n+f_{n-1})}\right)+\left(\frac{2\left(f_n^2+f_nf_{n+1}+f_{n+1}^2\right)}{3(f_n+f_{n+1})}\right)\right]+\beta\left(\frac{2\left(f_{n-1}^2+f_{n-1}f_{n+1}+f_{n+1}^2\right)}{3(f_{n-1}+f_{n+1})}\right)\right\} \tag{7.3.8}$$

where β is an arbitrary constant .

From equation (7.3.8), by substituting $\beta=0$, gives the Numerov (AM) method as

$$y_{n+1}=2y_n-y_{n-1}+h^2\left[\frac{5}{6}f_n+\frac{1}{12}(f_{n-1}+f_{n+1})\right]$$

$$=2y_n-y_{n-1}+\frac{h^2}{12}\left[10f_n+f_{n-1}+f_{n+1}\right] \tag{7.3.9}$$

and the C_eM formula is obtained by substituting $\beta = \frac{1}{2880}$ into equation (7.3.8) to obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{5760}\{4804 f_n + 481(f_{n-1} + f_{n+1})$$

$$-4\left[\left(\frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})}\right) + \left(\frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})}\right)\right] + 2\left(\frac{2(f_{n-1}^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_{n-1} + f_{n+1})}\right)\Bigg\}$$

... (7.3.10)

By substituting $\beta = -\frac{1}{2880}$ into equation (7.3.8) to obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{5760}\{4796 f_n + 479(f_{n-1} + f_{n+1})$$

$$+4\left[\left(\frac{2(f_n^2 + f_n f_{n-1} + f_{n-1}^2)}{3(f_n + f_{n-1})}\right) + \left(\frac{2(f_n^2 + f_n f_{n+1} + f_{n+1}^2)}{3(f_n + f_{n+1})}\right)\right] - 2\left(\frac{2(f_{n-1}^2 + f_{n-1}f_{n+1} + f_{n+1}^2)}{3(f_{n-1} + f_{n+1})}\right)\Bigg\}$$

... (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

$$LTE^{C_oM} = h^6\Bigg\{\frac{1}{720}(-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)}$$

$$+\frac{1}{36}(\alpha_4 + \alpha_5 + 4\alpha_6)\frac{f_n^{(1)} f_n^{(3)}}{f_n} + \frac{1}{48}(\alpha_4 + \alpha_5)\frac{(f_n^{(2)})^2}{f_n}$$

$$+\frac{1}{48}(-3\alpha_4 - 3\alpha_5 - 8\alpha_6)\frac{f_n^{(2)}(f_n^{(1)})^2}{(f_n)^2} + \frac{1}{48}(\alpha_4 + \alpha_5)\frac{(f_n^{(1)})^4}{(f_n)^3}\Bigg) \quad (7.3.12)$$

$$= \frac{h^6}{720}\{(-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6) f_n^{(4)}$$

$$+20(\alpha_4+\alpha_5+4\alpha_6)\frac{f_n^{(1)}f_n^{(3)}}{f_n}+15(\alpha_4+\alpha_5)\frac{\left(f_n^{(2)}\right)^2}{f_n}$$

$$+15(-3\alpha_4-3\alpha_5-8\alpha_6)\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{\left(f_n\right)^2}+15(\alpha_4+\alpha_5)\frac{\left(f_n^{(1)}\right)^4}{\left(f_n\right)^3}\Bigg) \tag{7.3.13}$$

By substituting $\alpha_6=\beta$ and (7.3.7) into equation (7.3.13), we obtain the local truncation error of equation (7.3.11) as

$$LTE^{C_oM}=\frac{h^6}{240}\left\{f_n^{(4)}-20\beta\frac{\left(f_n^{(2)}\right)^2}{f_n}+20\beta\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{\left(f_n\right)^2}-20\beta\frac{\left(f_n^{(4)}\right)^4}{\left(f_n\right)^3}\right\}$$

$$=\frac{h^6}{240}\left\{f_n^{(4)}+20\beta\left[\frac{f_nf_n^{(2)}\left(f_n^{(1)}\right)^2-\left(f_n\right)^2\left(f_n^{(2)}\right)^2-\left(f_n^{(1)}\right)^4}{\left(f_n\right)^3}\right]\right\} \tag{7.3.14}$$

and if we define

$$Mn=\frac{f_nf_n^{(2)}\left(f_n^{(1)}\right)^2-\left(f_n\right)^2\left(f_n^{(2)}\right)^2-\left(f_n^{(1)}\right)^4}{\left(f_n\right)^3}\ .$$

For $\beta=\frac{1}{2880}$, the local truncation error (LTE) for the C_eM formula in equation (7.3.10) is given by

$$LTE^{C_eM}=\frac{h^6}{34560}\left\{144f_n^{(4)}+Mn\right\} \tag{7.3.15}$$

and by substituting $\beta=-\frac{1}{2880}$ in equation (7.3.14) the local truncation error (LTE) for the C_cM formula in (7.3.11) is given as

$$LTE^{C_eM}=\frac{h^6}{34560}\left\{144f_n^{(4)}+Mn\right\} \tag{7.3.16}$$

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 : $x_o = 0$, $y_o = 1$, $y_o' = 2$

Exact Solution : $y = \left(1 - \frac{x^3}{3} + \frac{x^6}{180} - \ldots\right) + 2\left(x - \frac{x^4}{12} + \frac{x^7}{504} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.119965000595E+01	.119965000595E+01	.000000000000E+00
.20	.139806708690E+01(X)	.139706707302E+01	.715795183538E-03
	.139806576794E+01(Y)		.714851092616E-03
	.139806840586E+01(Z)		.716739274460E-03
.30	.159365496416E+01(X)	.158965491786E+01	.251629850849E-02
	.159364934757E+01(Y)		.251276529493E-02
	.159366058075E+01(Z)		.251983172609E-02
.40	.178442927854E+01(X)	.177442925714E+01	.563562698202E-02
	.178441433382E+01(Y)		.562720471252E-02
	.178444422331E+01(Z)		.564404928021E-02
.50	.196803408010E+01(X)	.194803447421E+01	.102665564462E-01
	.196800236728E+01(Y)		.102502770544E-01
	.196806579314E+01(Z)		.102828359542E-01
.60	.214176811045E+01(X)	.210677028571E+01	.166120744014E-01
	.214170947714E+01(Y)		.165842435024E-01
	.214182674450E+01(Z)		.166399056527E-01
.70	.230262257646E+01(X)	.224663040833E+01	.249227322500E-01
	.230252393295E+01(Y)		.248788249310E-01
	.230272122199E+01(Z)		.249666404617E-01
.80	.244733187537E+01(X)	.236335522540E+01	.355328090634E-01
	.244717707374E+01(Y)		.354673082762E-01
	.244748668172E+01(Z)		.355983118480E-01
.90	.257243840106E+01(X)	.245250045357E+01	.489043528245E-01
	.257220824572E+01(Y)		.488105076499E-01
	.257266856641E+01(Z)		.489982020836E-01
1.00	.267437213004E+01(X)	.250952380952E+01	.656890840780E-01
	.267404454710E+01(Y)		.655585481784E-01
	.267469973258E+01(Z)		.658196277839E-01

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 C_eM formula (7.3.10)

Z denote formula C_eM formula (7.3.11)

Problem 2: $y'' + 2x^2y = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y_o^{(1)} = 1$

Exact Solution : $y = \left(1 - \frac{x^4}{6} + \frac{x^8}{168} - \ldots\right) + \left(x - \frac{x^5}{10} + \frac{x^9}{360} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.109998233340E+01	.109998233340E+01	.000000000000E+00
.20	.119970133876E+01(X) .119970101932E+01(Y) .119970165820E+01(Z)	.119970134999E+01	.936544166655E-08 .275629277623E-06 .256898394105E-06
.30	.129840729714E+01(X) .129840554781E+01(Y) .129840904647E+01(Z)	.129840744521E+01	.114042419673E-06 .146133016009E-05 .123324590116E-05
.40	.139471310095E+01(X) .139470735942E+01(Y) .139471884249E+01(Z)	.139471396246E+01	.617701420252E-06 .473433837777E-05 .349894234915E-05
.50	.148648365085E+01(X) .148646913736E+01(Y) .148649816440E+01(Z)	.148648701017E+01	.225990528824E-05 .120235190288E-04 .750374997611E-05
.60	.157074164286E+01(X) .157071052442E+01(Y) .157077276158E+01(Z)	.157075197074E+01	.657512117010E-05 .263862954209E-04 .32362307040E-04
.70	.164360423379E+01(X) .164354478252E+01(Y) .164366368605E+01(Z)	.164363156960E+01	.166313519292E-04 .528020323092E-04 .195399325706E-04
.80	.170027071388E+01(X) .170016656480E+01(Y) .170037486593E+01(Z)	.170033680417E+01	.388689400721E-04 .100120967343E-03 .223848362394E-04
.90	.173508683101E+01(X) .173491649468E+01(Y) .173525717515E+01(Z)	.173523947285E+01	.879658605241E-04 .186128870414E-03 .102016483915E-04
1.00	.174171581559E+01(X) .174145265621E+01(Y) .174197899342E+01(Z)	.174206349206E+01	.199577383509E-03 .350639258562E-03 .485049185424E-04

Table 7.7

Problem 3: $y^{(2)} + x^2 y = 1 + x + x^2$

Initial values : $x_o = 0$, $y_o = 2$, $y_o^{(1)} = 2$

$$\text{Exact Solution : } y = 2\left(1 - \frac{x^4}{12} + \frac{x^8}{672} - \ldots\right) + 2\left(x - \frac{x^5}{20} + \frac{x^9}{1440} - \ldots\right)$$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.220515731629E+01	.220515731629E+01	.000000000000E+00
.20	.242116719231E+01(X) .242117846901E+01(Y) .242115591560E+01(Z)	.242116688706E+01	.126074661035E-06 .478362362933E-05 .453147430726E-05
.30	.264857051467E+01(X) .264860489112E+01(Y) .264853613821E+01(Z)	.264856910711E+01	.531441912139E-06 .135106951500E-04 .124478163520E-04
.40	.288743990707E+01(X) .288750924092E+01(Y) .288737057312E+01(Z)	.288743590441E+01	.138623499026E-05 .253984906416E-04 .226260563602E-04
.50	.313722608903E+01(X) .313734156911E+01(Y) .313711060850E+01(Z)	.313721710689E+01	.286309062579E-05 .396728102554E-04 .339467718105E-04
.60	.339659195430E+01(X) .339676318035E+01(Y) .339642072683E+01(Z)	.339657430537E+01	.519609809508E-05 .556074919201E-04 .452157177116E-04
.70	.366323827217E+01(X) .366347210125E+01(Y) .366300443933E+01(Z)	.366320554629E+01	.893367162527E-05 .727654931235E-04 .548991751322E-04
.80	.393372811772E+01(X) .393402727697E+01(Y) .393342894995E+01(Z)	.393366680462E+01	.155867549444E-04 .916377434813E-04 .604664003490E-04
.90	.420332107723E+01(X) .420368256725E+01(Y) .420295956989E+01(Z)	.420319930665E+01	.289709273196E-04 .114974468059E-03 .570367334037E-04
1.00	.446583286944E+01(X) .446624621826E+01(Y) .446541948844E+01(Z)	.446557539683E+01	.576572081016E-04 .150220604312E-03 .349133923420E-04

Table 7.8

Problem 4: $y^{(2)} - y = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y_o^{(1)} = -1$

Exact Solution : $y = \exp(-x)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.904837418036E+00	.904837418036E+00	.000000000000E+00
.20	.818723210252E+00(X) .818732635642E+00(Y) .818713784863E+00(Z)	.818730753078E+00	.921282811814E-05 .229936894503E-05 .207250251814E-04
.30	.740796234571E+00(X) .740823708070E+00(Y) .740768761269E+00(Z)	.740818220682E+00	.296781450339E-04 .740719874887E-05 .667632237556E-04
.40	.670277221236E+00(X) .670330734491E+00(Y) .670223708947E+00(Z)	.670320046036E+00	.638870942600E-04 .159453021364E-04 .143718047987E-03
.50	.606460980113E+00(X) .606548050870E+00(Y) .606373912218E+00(Z)	.606530659713E+00	.114882238736E-03 .286731707725E-04 .258432929178E-03
.60	.548709348791E+00(X) .548837165966E+00(Y) .548581538215E+00(Z)	.548811636094E+00	.186379618579E-03 .465184595057E-04 .419265670619E-03
.70	.496444810957E+00(X) .496620369775E+00(Y) .496269265204E+00(Z)	.496585303791E+00	.282917826556E-03 .706142202927E-04 .636423560507E-03
.80	.449144721232E+00(X) .449374950412E+00(Y) .448914515374E+00(Z)	.449328964117E+00	.410040081691E-03 .102344380291E-03 .922372641460E-03
.90	.406336078720E+00(X) .406627961541E+00(Y) .406044234504E+00(Z)	.406569659741E+00	.574516605226E-03 .143399289503E-03 .129233754661E-02
1.00	.367590796995E+00(X) .367951487994E+00(Y) .367230166352E+00(Z)	.367879441171E+00	.784616219919E-03 .195843567524E-03 .176491194365E-02

Table 7.9

Problem 5: $y^{(2)} - y\left\{\left[(Q+Bx)/x\right]^2 - Q/x^2\right\} = 0$

Initial values : $x_o = 1$, $y_o = 10e$, $y_o^{(1)} = 10e(Q+B)$

Exact Solution : $y = Cx^Q e^{Bx}$

$$\left(Set \quad B = 1 \text{ , } C = 10 \text{ , } Q = \frac{3}{2}\right)$$

xn	Numerical Solution	Exact Solution	Absolute Error
1.10	.346587549802E+02	.346587549802E+02	.000000000000E+00
1.20	.436390152385E+02(X) .436405798154E+02(Y) .436374506616E+02(Z)	.436440703713E+02	.115826337471E-03 .799777818272E-04 .151674893115E-03
1.30	.543715401009E+02(X) .543765543140E+02(Y) .543665260158E+02(Z)	.543873445416E+02	.290590410221E-03 .198395926046E-03 .382782538445E-03
1.40	.671413377291E+02(X) .671521082286E+02(Y) .671305678773E+02(Z)	.671744823128E+02	.493410333302E-03 .333074158730E-03 .653736865316E-03
1.50	.822756510828E+02(X) .822950166448E+02(Y) .822562874947E+02(Z)	.823338855610E+02	.707296611036E-03 .472088933389E-03 .942480313587E-03
1.60	.100149815184E+03(X) .100181271096E+03(Y) .100118363974E+03(Z)	.100242328229E+03	.922894012738E-03 .609095315974E-03 .123664580781E-02
1.70	.121193894010E+03(X) .121241734348E+03(Y) .121146063314E+03(Z)	.121331621407E+03	.113513192929E-02 .740837865543E-03 .152934651886E-02
1.80	.145900197789E+03(X) .145969679621E+03(Y) .145830733834E+03(Z)	.146096168080E+03	.134137871608E-02 .865789024863E-03 .181684604938E-02
1.90	.174831793908E+03(X) .174929327678E+03(Y) .174734290944E+03(Z)	.175101520393E+03	.154040058834E-02 .983387890907E-03 .209723735182E-02
2.00	.208632138690E+03(X) .208765509450E+03(Y) .208498818168E+03(Z)	.208994066965E+03	.173176339644E-02 .109360767111E-02 .236967874033E-02

Table 7.10

7.4 DERIVATION OF THE H_aM METHOD FOR PROBLEMS OF THE TYPE $y^{(2)} = f(x,y)$

For solving problems of the type $y^{(2)} = f(x,y)$, we now consider the harmonic mean formula H_aM as

$$y_{n+1} - 2y_n + y_{n-1} = h^2\left\{\alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} + \alpha_4\left[\frac{2(f_n f_{n-1})}{(f_n + f_{n-1})}\right]\right.$$

$$\left. + \alpha_5\left[\frac{2(f_n f_{n+1})}{(f_n + f_{n+1})}\right] + \alpha_6\left[\frac{2(f_{n-1} f_{n+1})}{(f_{n-1} + f_{n+1})}\right]\right\} \qquad (7.4.1)$$

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:

$$\alpha_4\left[\frac{2f_n f_{n-1}}{f_n + f_{n-1}}\right] = \alpha_4 f_n\left[1 - \frac{hf_n^{(1)}}{2f_n} + \frac{h^2}{4}\left(-\left(\frac{f_n^{(1)}}{f_n}\right)^2 + \frac{f_n^{(2)}}{f_n}\right)\right.$$

$$+\frac{h^3}{24}\left(-3\left(\frac{f_n^{(1)}}{f_n}\right)^3 + 6\frac{f_n^{(1)} f_n^{(2)}}{(f_n)^2} - 2\frac{f_n f_n^{(3)}}{(f_n)^2}\right)$$

$$+\frac{h^4}{48}\left(-3\left(\frac{f_n^{(1)}}{f_n}\right)^4 + 9\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{(f_n)^3} - 3\frac{f_n\left(f_n^{(2)}\right)^2}{(f_n)^3} - 4\frac{f_n f_n^{(1)} f_n^{(3)}}{(f_n)^3} + \frac{(f_n)^2 f_n^{(4)}}{(f_n)^3}\right)$$

$$+\frac{h^5}{480}\left(-15\left(\frac{f_n^{(1)}}{f_n}\right)^5 + 60\frac{\left(f_n^{(1)}\right)^3 f_n^{(2)}}{(f_n)^4} - 45\frac{f_n f_n^{(1)}\left(f_n^{(2)}\right)^2}{(f_n)^4} - 30\frac{f_n\left(f_n^{(1)}\right)^2 f_n^{(3)}}{(f_n)^4}\right.$$

$$\left.+20\frac{(f_n)^2 f_n^{(2)} f_n^{(3)}}{(f_n)^4} + 10\frac{(f_n)^2 f_n^{(1)} f_n^{(4)}}{(f_n)^4} - 2\frac{(f_n)^3 f_n^{(5)}}{(f_n)^4}\right) + 0(h^6) \qquad (7.4.2)$$

$$\alpha_5\left[\frac{2f_n f_{n+1}}{f_n + f_{n+1}}\right] = \alpha_5 f_n\left[1 + \frac{hf_n^{(1)}}{2f_n} + \frac{h^2}{4}\left(-\left(\frac{f_n^{(1)}}{f_n}\right)^2 + \frac{f_n^{(2)}}{f_n}\right)\right.$$

$$+\frac{h^3}{24}\left(3\left(\frac{f_n^{(1)}}{f_n}\right)^3-6\frac{f_n^{(1)}f_n^{(2)}}{(f_n)^2}+2\frac{f_nf_n^{(3)}}{(f_n)^2}\right)$$

$$+\frac{h^4}{48}\left(-3\left(\frac{f_n^{(1)}}{f_n}\right)^4+9\frac{\left(f_n^{(1)}\right)^2f_n^{(2)}}{(f_n)^3}-3\frac{f_n\left(f_n^{(2)}\right)^2}{(f_n)^3}-4\frac{f_nf_n^{(1)}f_n^{(3)}}{(f_n)^3}+\frac{(f_n)^2f_n^{(4)}}{(f_n)^3}\right)$$

$$+\frac{h^5}{480}\left(15\left(\frac{f_n^{(1)}}{f_n}\right)^5-60\frac{\left(f_n^{(1)}\right)^3f_n^{(2)}}{(f_n)^4}+45\frac{f_nf_n\left(f_n^{(2)}\right)^2}{(f_n)^4}+30\frac{f_n\left(f_n^{(1)}\right)^2f_n^{(3)}}{(f_n)^4}\right.$$

$$\left.\left.-20\frac{(f_n)^2f_n^{(2)}f_n^{(3)}}{(f_n)^4}-10\frac{(f_n)^2f_n^{(1)}f_n^{(4)}}{(f_n)^4}+2\frac{(f_n)^3f_n^{(5)}}{(f_n)^4}\right)\right]+0\left(h^6\right) \qquad (7.4.3)$$

$$\alpha_6\left[\frac{2f_{n+1}f_{n-1}}{f_{n+1}+f_{n-1}}\right]=\alpha_6f_n\left[1+\frac{h^2}{2}\left(-2\left(\frac{f_n^{(1)}}{f_n}\right)^2+\frac{f_n^{(2)}}{f_n}\right)\right.$$

$$\left.+\frac{h^4}{24}\left(12\frac{\left(f_n^{(1)}\right)^2f_n^{(2)}}{(f_n)^3}-8\frac{f_n^{(1)}f_n^{(3)}}{(f_n)^2}+\frac{f_n^{(4)}}{f_n}\right)\right]+0\left(h^6\right) \qquad (7.4.4)$$

By substituting equations (7.2.11)-(7.2.12) and (7.4.2)-(7.4.4) into equation (7.4.1), we obtain

$$h^2f_n\left\{\sum_{i=1}^{6}\alpha_i+(-2\alpha_2+2\alpha_3-\alpha_4+\alpha_5)\frac{hf_n^{(1)}}{2f_n}+(2\alpha_2+2\alpha_3+\alpha_4+\alpha_5+2\alpha_6)\frac{h^2f_n^{(2)}}{4f_n}\right.$$

$$+(\alpha_4+\alpha_5+4\alpha_6)\frac{h^2}{4}\left[\frac{f_n^{(1)}}{f_n}\right]^2+(-2\alpha_2+2\alpha_3-\alpha_4+\alpha_5)\frac{h^3f_n^{(3)}}{12f_n}$$

$$\left.+(\alpha_4-\alpha_5)\frac{h^3}{8}\left[\frac{f_n^{(1)}}{f_n}\right]^3+(-\alpha_4+\alpha_5)\frac{h^3f_n^{(1)}f_n^{(2)}}{4[f_n]^2}\right\}+0\left(h^6\right) \qquad (7.4.5)$$

By equating equations (7.2.8) and (7.4.5), we obtain the following equations of condition , i.e.,

$$h^2f_n \quad : \quad -1+\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_6=0 \qquad (7.4.6\text{-i})$$

$$h^3f_n^{(1)} \quad : \quad -2\alpha_2+2\alpha_3-\alpha_4+\alpha_5=0 \qquad (7.4.6\text{-ii})$$

$$h^4f_n^{(2)} \quad : \quad -1+6\alpha_2+6\alpha_3+3\alpha_4+3\alpha_5+6\alpha_6=0 \qquad (7.4.6\text{-iii})$$

$$\frac{h^4\left[f_n^{(1)}\right]^2}{f_n} \quad : \quad -\alpha_4 - \alpha_5 - 4\alpha_6 = 0 \qquad (7.4.6\text{-iv})$$

$$h^5 f_n^{(3)} \quad : \quad -2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5 = 0 \qquad (7.4.6\text{-v})$$

$$\frac{h^5\left[f_n^{(1)}\right]^3}{(f_n)^2} \quad : \quad \alpha_4 - \alpha_5 = 0 \qquad (7.4.6\text{-vi})$$

$$\frac{h^5 f_n^{(1)} f_n^{(2)}}{f_n} \quad : \quad -\alpha_4 + \alpha_5 = 0 \qquad (7.4.6\text{-vii})$$

For equations (7.4.6-ii), (7.4.6-v) and (7.4.6-vi) to be symmetric, we impose the condition that

$$-\alpha_2 + \alpha_3 = 0 \qquad (7.4.6\text{-viii})$$

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

$$\alpha_1 = \frac{5}{6} + 2\alpha_6 \quad , \quad \alpha_2 = \alpha_3 = \frac{1}{12} + \frac{\alpha_6}{2} \quad , \quad \alpha_4 = \alpha_5 = -2\alpha_6 \qquad (7.4.7)$$

By substituting $\alpha_6 = \beta$ into (7.4.7) and from (7.4.1), we obtain

$$y_{n+1} = 2y_n - y_{n-1} + h^2\left\{\left(\frac{5}{6} + 2\beta\right)f_n + \left(\frac{1}{12} + \frac{\beta}{2}\right)(f_{n-1} + f_{n+1})\right.$$

$$\left. -2\beta\left[\left(\frac{2f_n f_{n-1}}{f_n + f_{n-1}}\right) + \left(\frac{2f_n f_{n+1}}{f_n + f_{n+1}}\right)\right] + \beta\left(\frac{2f_{n+1}f_{n-1}}{f_{n+1} + f_{n-1}}\right)\right\} \qquad (7.4.8)$$

where β is an arbitrary constant .

From equation (7.4.8), by substituting $\beta = 0$, gives the Numerov (AM) method as

$$y_{n+1} = 2y_n - y_{n-1} + h^2\left[\frac{5}{6}f_n + \frac{1}{12}(f_{n-1} + f_{n+1})\right]$$

$$= 2y_n - y_{n-1} + \frac{h^2}{12}\left[10f_n + f_{n-1} + f_{n+1}\right] \qquad (7.4.9)$$

and the H_aM formula is obtained by substituting $\beta = \dfrac{1}{144}$ into equation (7.4.8) to obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288}\{144f_n + 25(f_{n-1} + f_{n+1})$$

$$-4\left[\left(\frac{2f_nf_{n-1}}{f_n + f_{n-1}}\right) + \left(\frac{2f_nf_{n+1}}{f_n + f_{n+1}}\right)\right] + 2\left(\frac{2f_{n+1}f_{n-1}}{f_{n+1} + f_{n-1}}\right)\}$$

... (7.4.10)

By substituting $\beta = -\frac{1}{144}$ into equation (7.4.8) to obtain

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{288}\{236f_n + 23(f_{n-1} + f_{n+1})$$

$$+4\left[\left(\frac{2f_nf_{n-1}}{f_n + f_{n-1}}\right) + \left(\frac{2f_nf_{n+1}}{f_n + f_{n+1}}\right)\right] - 2\left(\frac{2f_{n+1}f_{n-1}}{f_{n+1} + f_{n-1}}\right)\}$$

... (7.4.11)

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

$$LTE^{H_aM} = h^6\left\{\frac{1}{720}(-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6)f_n^{(4)}\right.$$

$$+\frac{1}{12}(-\alpha_4 - \alpha_5 - 4\alpha_6)\frac{f_n^{(1)}f_n^{(3)}}{f_n} + \frac{1}{16}(-\alpha_4 - \alpha_5)\frac{\left(f_n^{(2)}\right)^2}{f_n}$$

$$\left.+\frac{1}{16}(3\alpha_4 + 3\alpha_5 + 8\alpha_6)\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{(f_n)^2} + \frac{1}{16}(-\alpha_4 - \alpha_5)\frac{\left(f_n^{(1)}\right)^4}{(f_n)^3}\right\} \qquad (7.4.12)$$

$$= \frac{h^6}{720}\{(-2 + 30\alpha_2 + 30\alpha_3 + 15\alpha_4 + 15\alpha_5 + 30\alpha_6)f_n^{(4)}$$

$$+60(-\alpha_4 - \alpha_5 - 4\alpha_6)\frac{f_n^{(1)}f_n^{(3)}}{f_n} + 15(-\alpha_4 - \alpha_5)\frac{\left(f_n^{(2)}\right)^2}{f_n}$$

$$+45(3\alpha_4 + 3\alpha_5 + 8\alpha_6)\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{(f_n)^2} + 45(-\alpha_4 - \alpha_5)\frac{\left(f_n^{(1)}\right)^4}{(f_n)^3}\Bigg) \qquad (7.4.13)$$

By substituting $\alpha_6 = \beta$ and (7.4.7) into equation (7.4.13), we obtain the local truncation error of equation (7.4.11) as

$$LTE^{H_aM} = \frac{h^6}{240}\left\{f_n^{(4)} + 60\beta\frac{\left(f_n^{(2)}\right)^2}{f_n} - 60\beta\frac{f_n^{(2)}\left(f_n^{(1)}\right)^2}{(f_n)^2} + 60\beta\frac{\left(f_n^{(4)}\right)^4}{(f_n)^3}\right\}$$

$$= \frac{h^6}{240}\left\{f_n^{(4)} - 60\beta\left[\frac{f_n f_n^{(2)}\left(f_n^{(1)}\right)^2 - (f_n)^2\left(f_n^{(2)}\right)^2 - \left(f_n^{(1)}\right)^4}{(f_n)^3}\right]\right\} \qquad (7.4.14)$$

and we define

$$Mn = \frac{f_n f_n^{(2)}\left(f_n^{(1)}\right)^2 - (f_n)^2\left(f_n^{(2)}\right)^2 - \left(f_n^{(1)}\right)^4}{(f_n)^3}\ .$$

For $\beta = \frac{1}{144}$, the local truncation error (LTE) for H_aM formula in equation (7.4.10) is given by

$$LTE^{H_aM} = \frac{h^6}{2880}\left\{12 f_n^{(4)} + 5Mn\right\} \qquad (7.4.15)$$

and by substituting $\beta = -\frac{1}{144}$ in equation (7.4.14) the local truncation error (LTE) for the H_aM formula in (7.4.11) is given as

$$LTE^{H_aM} = \frac{h^6}{2880}\left\{12 f_n^{(4)} - 5Mn\right\} \qquad (7.4.16)$$

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 an accuracy between these methods are shown in Tables (7.11)-(7.15).

Problem 1: $y'' + xy = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y'_o = 2$

Exact Solution : $y = \left(1 - \frac{x^3}{3} + \frac{x^6}{180} - \ldots\right) + 2\left(x - \frac{x^4}{12} + \frac{x^7}{504} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.119965000595E+01	.119965000595E+01	.000000000000E+00
.20	.139806708690E+01(X) .139806403280E+01(Y) .139807014100E+01(Z)	.139706707302E+01	.715795183538E-03 .713609105633E-03 .717981261443E-03
.30	.159365496416E+01(X) .159364826663E+01(Y) .159366166169E+01(Z)	.158965491786E+01	.251629850849E-02 .251208531616E-02 .252051170910E-02
.40	.178442927854E+01(X) .178441869788E+01(Y) .178443985924E+01(Z)	.177442925714E+01	.563562698202E-02 .562966412911E-02 .564158985500E-02
.50	.196803408010E+01(X) .196801951229E+01(Y) .196804864796E+01(Z)	.194803447421E+01	.102665564462E-01 .102590782401E-01 .102740346854E-01
.60	.214176811045E+01(X) .214174953598E+01(Y) .214178668501E+01(Z)	.210677028571E+01	.166120744014E-01 .166032578425E-01 .166208910065E-01
.70	.230262257646E+01(X) .230260004523E+01(Y) .230264510783E+01(Z)	.224663040833E+01	.249227322500E-01 .249127033487E-01 .249327612106E-01
.80	.244733187537E+01(X) .244730550316E+01(Y) .244735824775E+01(Z)	.236335522540E+01	.355328090634E-01 .355216502615E-01 .355439679374E-01
.90	.257243840106E+01(X) .257240837031E+01(Y) .257246843201E+01(Z)	.245250045357E+01	.489043528245E-01 .488921078748E-01 .489165978591E-01
1.00	.267437213004E+01(X) .267433869201E+01(Y) .267440556832E+01(Z)	.250952380952E+01	.656890840780E-01 .656757596241E-01 .657024086297E-01

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 $H_a M$ formula (7.4.10)

Z denote formula $H_a M$ formula (7.4.11)

Problem 2: $y'' + 2x^2 y = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y_o^{(1)} = 1$

Exact Solution : $y = \left(1 - \frac{x^4}{6} + \frac{x^8}{168} - \ldots\right) + \left(x - \frac{x^5}{10} + \frac{x^9}{360} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.109998233340E+01	.109998233340E+01	.000000000000E+00
.20	.119970133876E+01(X) .119969992168E+01(Y) .119970275584E+01(Z)	.119970134999E+01	.936544166655E-08 .119056150351E-05 .117183061999E-05
.30	.129840729714E+01(X) .129840333101E+01(Y) .129841126327E+01(Z)	.129840744521E+01	.114042419673E-06 .316864816689E-05 .294057112078E-05
.40	.139471310095E+01(X) .139470583565E+01(Y) .139472036629E+01(Z)	.139471396246E+01	.617701420252E-06 .582687034057E-05 .459149460478E-05
.50	.148648365085E+01(X) .148647256698E+01(Y) .148649473480E+01(Z)	.148648701017E+01	.225990528824E-05 .971632141034E-05 .519656853923E-05
.60	.157074164286E+01(X) .157072638729E+01(Y) .157075689858E+01(Z)	.157075197074E+01	.657512117010E-05 .162873889669E-04 .313724479653E-05
.70	.164360423379E+01(X) .164358460263E+01(Y) .164362386518E+01(Z)	.164363156960E+01	.166313519292E-04 .285751193898E-04 .468743749893E-05
.80	.170027071388E+01(X) .170024666040E+01(Y) .170029476769E+01(Z)	.170033680417E+01	.388689400721E-04 .530152399252E-04 .247224373693E-04
.90	.173508683101E+01(X) .173505848723E+01(Y) .173511517526E+01(Z)	.173523947285E+01	.879658605241E-04 .104300080438E-03 .716313756380E-04
1.00	.174171581559E+01(X) .174168352008E+01(Y) .174174811168E+01(Z)	.174206349206E+01	.199577383509E-03 .218116033568E-03 .181038400430E-03

Table 7.12

Problem 3: $y^{(2)} + x^2 y = 1 + x + x^2$

Initial values : $x_o = 0$, $y_o = 2$, $y_o^{(1)} = 2$

Exact Solution : $y = 2\left(1 - \frac{x^4}{12} + \frac{x^8}{672} - \ldots\right) + 2\left(x - \frac{x^5}{20} + \frac{x^9}{1440} - \ldots\right)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.220515731629E+01	.220515731629E+01	.000000000000E+00
.20	.242116719231E+01(X) .242116721194E+01(Y) .242116717267E+01(Z)	.242116688706E+01	.126074661035E-06 .134184050926E-06 .117965271145E-06
.30	.264857051467E+01(X) .264857058664E+01(Y) .264857044271E+01(Z)	.264856910711E+01	.531441912139E-06 .558613056588E-06 .504270769870E-06
.40	.288743990707E+01(X) .288744008714E+01(Y) .288743972700E+01(Z)	.288743590441E+01	.138623499026E-05 .144859962533E-05 .132387036964E-05
.50	.313722608903E+01(X) .313722648028E+01(Y) .313722569779E+01(Z)	.313721710689E+01	.286309062579E-05 .298780156579E-05 .273837975232E-05
.60	.339659195430E+01(X) .339659279418E+01(Y) .339659111444E+01(Z)	.339657430537E+01	.519609809508E-05 .544336791383E-05 .494882859352E-05
.70	.366323827217E+01(X) .366324051235E+01(Y) .366323603200E+01(Z)	.366320554629E+01	.893367162527E-05 .954520814619E-05 .832213777637E-05
.80	.393372811772E+01(X) .393375230618E+01(Y) .393370393028E+01(Z)	.393366680462E+01	.155867549444E-04 .217358423771E-04 .943792519809E-05
.90	.420332107723E+01(X) .420351905588E+01(Y) .420312309456E+01(Z)	.420319930665E+01	.289709273196E-04 .760728224634E-04 .181319240924E-04
1.00	.446583286944E+01(X) .446619293003E+01(Y) .446547279445E+01(Z)	.446557539683E+01	.576572081016E-04 .138287488676E-03 .229762941082E-04

Table 7.13

Problem 4: $y^{(2)} - y = 0$

Initial values : $x_o = 0$, $y_o = 1$, $y_o^{(1)} = -1$

Exact Solution : $y = \exp(-x)$

xn	Numerical Solution	Exact Solution	Absolute Error
.10	.904837418036E+00	.904837418036E+00	.000000000000E+00
.20	.818723210252E+00(X) .818723212180E+00(Y) .818723208325E+00(Z)	.818730753078E+00	.921282811814E-05 .921047419541E-05 .921518204101E-05
.30	.740796234571E+00(X) .740796240189E+00(Y) .740796228953E+00(Z)	.740818220682E+00	.296781450339E-04 .296705612313E-04 .296857288372E-04
.40	.670277221236E+00(X) .670277232180E+00(Y) .670277210291E+00(Z)	.670320046036E+00	.638870942600E-04 .638707668567E-04 .639034216655E-04
.50	.606460980113E+00(X) .606460997923E+00(Y) .606460962302E+00(Z)	.606530659713E+00	.114882238736E-03 .114852874862E-03 .114911602615E-03
.60	.548709348791E+00(X) .548709374939E+00(Y) .548709322642E+00(Z)	.548811636094E+00	.186379618579E-03 .186331973487E-03 .186427263684E-03
.70	.496444810957E+00(X) .496444846877E+00(Y) .496444775037E+00(Z)	.496585303791E+00	.282917826556E-03 .282845492598E-03 .282990160538E-03
.80	.449144721232E+00(X) .449144768345E+00(Y) .449144674119E+00(Z)	.449328964117E+00	.410040081691E-03 .409935230517E-03 .410144932908E-03
.90	.406336078720E+00(X) .406336138458E+00(Y) .406336018982E+00(Z)	.406569659741E+00	.574516605226E-03 .574369672825E-03 .574663537700E-03
1.00	.367590796995E+00(X) .367590870828E+00(Y) .367590723162E+00(Z)	.367879441171E+00	.784616219919E-03 .784415521840E-03 .784816918115E-03

Table 7.14

Problem 5: $y^{(2)} - y\left\{\left[(Q+Bx)/x\right]^2 - Q/x^2\right\} = 0$

Initial values : $x_o = 1$, $y_o = 10e$, $y_o^{(1)} = 10e(Q+B)$

Exact Solution : $y = Cx^Q e^{Bx}$

$$\left(Set \quad B = 1\,,\, C = 10\,,\, Q = \frac{3}{2}\right)$$

xn	Numerical Solution	Exact Solution	Absolute Error
1.10	.346587549802E+02	.346587549802E+02	.000000000000E+00
1.20	.436390152385E+02(X)	.436440703713E+02	.115826337471E-03
	.436390178797E+02(Y)		.115765820041E-03
	.436390125972E+02(Z)		.115886854901E-03
1.30	.543715401009E+02(X)	.543873445416E+02	.290590410221E-03
	.543715481755E+02(Y)		.290441944473E-03
	.543715320262E+02(Z)		.290738875845E-03
1.40	.671413377291E+02(X)	.671744823128E+02	.493410333302E-03
	.671413543102E+02(Y)		.493163495968E-03
	.671413211479E+02(Z)		.493657170275E-03
1.50	.822756510828E+02(X)	.823338855610E+02	.707296611036E-03
	.822756796445E+02(Y)		.706949709502E-03
	.822756225211E+02(Z)		.707643511888E-03
1.60	.100149815184E+03(X)	.100242328229E+03	.922894012738E-03
	.100149859721E+03(Y)		.922449724767E-03
	.100149770648E+03(Z)		.923338299662E-03
1.70	.121193894010E+03(X)	.121331621407E+03	.113513192929E-02
	.121193959159E+03(Y)		.113459497516E-02
	.121193828860E+03(Z)		.113566888199E-02
1.80	.145900197789E+03(X)	.146096168080E+03	.134137871608E-02
	.145900288968E+03(Y)		.134075461276E-02
	.145900106610E+03(Z)		.134200281759E-02
1.90	.174831793908E+03(X)	.175101520393E+03	.154040058834E-02
	.174831917459E+03(Y)		.153969499054E-02
	.174831670357E+03(Z)		.154110618393E-02
2.00	.208632138690E+03(X)	.208994066965E+03	.173176339644E-02
	.208632302047E+03(Y)		.173098175995E-02
	.208631975333E+03(Z)		.173254503037E-02

Table 7.15

Table 7.16 : Computational work of various integration formulas

Method	Square Root	Division	Multiplication	Additions	Total
Numerov (7.2.20)	0	0	3	4	7
GM (7.2.3)	3	0	7	5	15
GM (7.2.4)	3	0	8	6	17
CoM (7.2.21)	0	3	6	13	21
CeM (7.3.10)	0	3	15	16	34
HaM (7.4.10)	0	3	8	10	21

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_oM and H_aM methods are three times more work than the Numerov formulae while the C_eM 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

$$y_{n+1} = y_{n-1} + 2hf_n(x,y) \qquad (7.5.1)$$

We may now to replace $2f_n(x,y)$ by alternative means, i.e., geometric mean (GM), Contraharmonic mean (C_oM), Centroidal Mean (C_eM) and Harmonic Mean (H_aM). 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 $\left(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}\right)$ in equation (7.5.1) to produce a new formula in the form

$$y_{n+1} = y_{n-1} + h\left(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}\right) \qquad (7.5.2)$$

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

$$y_{n+1} = y_n + hy_n^{(1)} + \frac{h^2}{2!}y_n^{(2)} + \frac{h^3}{3!}y_n^{(3)} + \frac{h^4}{4!}y_n^{(4)} + \frac{h^5}{5!}y_n^{(5)} + 0(h^6) \qquad (7.5.3)$$

$$y_{n-1} = y_n - hy_n^{(1)} + \frac{h^2}{2!}y_n^{(2)} - \frac{h^3}{3!}y_n^{(3)} + \frac{h^4}{4!}y_n^{(4)} - \frac{h^5}{5!}y_n^{(5)} + 0(h^6) \qquad (7.5.4)$$

$$f_{n+1} = f_n + hf_n^{(1)} + \frac{h^2}{2!}f_n^{(2)} + \frac{h^3}{3!}f_n^{(3)} + \frac{h^4}{4!}f_n^{(4)} + \frac{h^5}{5!}f_n^{(5)} + 0(h^6) \qquad (7.5.5)$$

$$f_{n-1} = f_n - hf_n^{(1)} + \frac{h^2}{2!}f_n^{(2)} - \frac{h^3}{3!}f_n^{(3)} + \frac{h^4}{4!}f_n^{(4)} - \frac{h^5}{5!}f_n^{(5)} + 0(h^6) \qquad (7.5.6)$$

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 :

$$\sqrt{f_n f_{n-1}} = f_n - \frac{h}{2} f_n^{(1)} + \frac{h^2}{8}\left(2 f_n^{(2)} - \frac{\left(f_n^{(1)}\right)^2}{f_n}\right) + \frac{h^3}{48}\left(-4 f_n^{(3)} - 3\frac{\left(f_n^{(1)}\right)^3}{f_n^2} + 6\frac{f_n^{(1)} f_n^{(2)}}{f_n}\right)$$

$$+\frac{h^4}{384}\left(8 f_n^{(4)} - 16\frac{f_n^{(1)} f_n^{(3)}}{f_n} - 12\frac{\left(f_n^{(2)}\right)^2}{f_n} + 36\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{f_n^2} - 15\frac{\left(f_n^{(1)}\right)^4}{f_n^3}\right) + 0\left(h^5\right)$$

$$\ldots \qquad (7.5.7)$$

$$\sqrt{f_n f_{n+1}} = f_n + \frac{h}{2} f_n^{(1)} + \frac{h^2}{8}\left(2 f_n^{(2)} - \frac{\left(f_n^{(1)}\right)^2}{f_n}\right) + \frac{h^3}{48}\left(4 f_n^{(3)} + 3\frac{\left(f_n^{(1)}\right)^3}{f_n^2} - 6\frac{f_n^{(1)} f_n^{(2)}}{f_n}\right)$$

$$+\frac{h^4}{384}\left(8 f_n^{(4)} - 16\frac{f_n^{(1)} f_n^{(3)}}{f_n} - 12\frac{\left(f_n^{(2)}\right)^2}{f_n} + 36\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{f_n^2} - 15\frac{\left(f_n^{(1)}\right)^4}{f_n^3}\right) + 0\left(h^5\right)$$

$$\ldots \qquad (7.5.8)$$

By adding equations (7.5.7) and (7.5.8), we obtain

$$\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}} = 2 f_n + \frac{h^2}{4}\left(2 f_n^{(2)} - \frac{\left(f_n^{(1)}\right)^2}{f_n}\right)$$

$$+\frac{h^4}{192}\left(8 f_n^{(4)} - 16\frac{f_n^{(1)} f_n^{(3)}}{f_n} - 12\frac{\left(f_n^{(2)}\right)^2}{f_n} + 36\frac{\left(f_n^{(1)}\right)^2 f_n^{(2)}}{f_n^2} - 15\frac{\left(f_n^{(1)}\right)^4}{f_n^3}\right) + 0\left(h^5\right)$$

$$\ldots \qquad (7.5.9)$$

By substituting equations (7.5.4) and (7.5.9) into equation (7.5.2), we obtain

$$y_{n+1} = f_n h + \frac{h^2}{2} f_n^{(1)} + \frac{h^3}{12}\left(4 f_n^{(2)} - 3\frac{\left(f_n^{(1)}\right)^2}{f_n}\right) + \frac{h^4}{24} f_n^{(3)} + 0\left(h^5\right) \qquad (7.5.10)$$

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

$$LTE = \left[\frac{\left(f_n^{(1)}\right)^2}{4f_n} - \frac{f_n^{(2)}}{6} \right] h^3$$

$$= \frac{h^3}{12} \left[3\frac{\left(f_n^{(1)}\right)^2}{f_n} - 2f_n^{(2)} \right].$$

Stability of Formula (7.5.2)

We apply the stability method to test equation $y' = \lambda y$ in equation (7.5.2) to produce

$$y_{n+1} = y_{n-1} + h\lambda\left[\sqrt{y_n y_{n-1}} + \sqrt{y_n y_{n+1}}\right] \tag{7.5.11}$$

By substituting $y_n = \mu^n$, $y_{n+1} = \mu^{n+1}$, $y_{n-1} = \mu^{n-1}$ into equation (7.5.11) we obtain

$$\mu^{n+1} = \mu^{n-1} + (h\lambda)\left(\sqrt{\mu^n \mu^{n-1}} + \sqrt{\mu^n \mu^{n+1}}\right)$$

$$= \mu^{n-1} + (h\lambda)\left(\mu^{n-\frac{1}{2}} + \mu^{n+\frac{1}{2}}\right)$$

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

$$\mu - \frac{1}{\mu} - h\lambda\left(\frac{1}{\sqrt{\mu}} + \sqrt{\mu}\right) = 0 \tag{7.5.12}$$

We see that $\mu = 0$ is not root of this equation. Let $\sqrt{\mu} = v$ and substituting into equation (7.5.12) we have

$$v^2 - \frac{1}{v^2} - h\lambda\left(\frac{1}{v} + v\right) = 0$$

$$v^4 - 1 - h\lambda\left(v + v^3\right) = 0$$

$$v^4 - (h\lambda)v^3 - (h\lambda)v - 1 = 0 \tag{7.5.13}$$

By solving equation (7.5.13) using Mathematica i.e.,

$In[1] := Solve[v\wedge 4 - h\lambda\ v\wedge 3 - h\lambda\ v - 1 == 0, v]$

$$Out[1]:= v -> \frac{h\lambda \pm \sqrt{4+(h\lambda)^2}}{2} \quad , \quad v = \pm I \tag{7.5.14}$$

From equation (7.5.14) and $\mu = v^2$, we have

$$v_1^2 = \left[\frac{h\lambda + \sqrt{4+(h\lambda)^2}}{2}\right]^2$$

$$= 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{8} + 0(h^4)$$

$$= e^{h\lambda} + \frac{(h\lambda)^3}{8} + 0(h^4) \tag{7.5.15}$$

and

$$v_2^2 = \left[\frac{h\lambda - \sqrt{4+(h\lambda)^2}}{2}\right]^2$$

$$= -1 - h\lambda + \frac{(h\lambda)^2}{2} - \frac{(h\lambda)^3}{8} + 0(h^4)$$

$$= -e^{-h\lambda} - \frac{(h\lambda)^3}{8} + 0(h^4) \tag{7.5.16}$$

The general solution of the difference equation (7.5.12) is

$$y_n = C_1\mu_1^n + C_2\mu_2^n$$

$$= C_1(v_1^2)^n + C_2(v_2^2)^n$$

$$= C_1e^{h\lambda n} + C_2(-1)^n e^{-h\lambda n} + 0(h^3).$$

Since, $x_n = nh$ we now obtain

$$y_n = C_1e^{x_n\lambda} + C_2(-1)^n e^{x_n\lambda} \tag{7.5.17}$$

To determine the stability region of the second order geometric mean (GM) formula (7.5.11) in the complex plane that satisfy the condition

$$|\mu^n| = |(v^2)^n| < 1$$

i.e.,

$$|\mu_1^n| = |(v_1^2)^n| = \left|\left(\frac{h\lambda}{2} + \sqrt{1+\left(\frac{h\lambda}{2}\right)^2}\right)^2\right| < 1 \tag{7.5.18}$$

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.

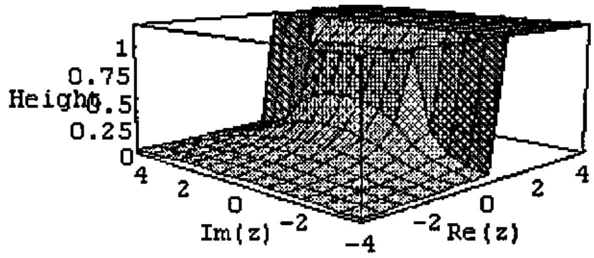


Figure 7.1: Graphic surface defined by equation (7.5.18)

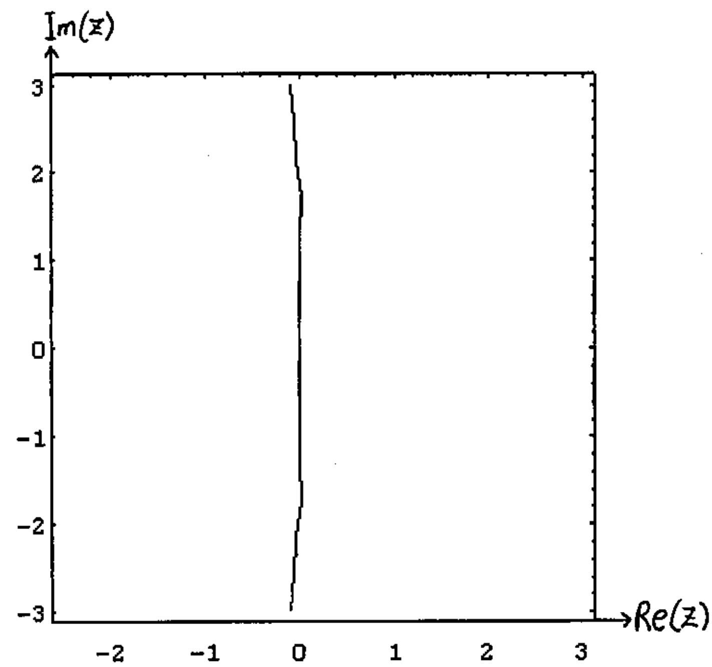


Figure 7.2: Stability region for second order GM method

Numerical Example

We consider the initial value problem

$$y' = -2y \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 1 \tag{7.5.19}$$

which has the solution $y(x) = \exp(-2x)$.

To use the equation (7.5.2), we must also compute an approximation to y_1, which we obtain from Euler's method. It gives

$$y_1 = y_0 + h(-2y_0).$$

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).

xn	exact solution	error (GM)	error(Midpoint)
.1000	.818730753078E+00	-.187307530780E-01	-.187307530780E-01
.2000	.670320046036E+00	-.920548423562E-02	.967995396436E-02
.3000	.548811636094E+00	-.264844229043E-01	-.208116360940E-01
.4000	.449328964117E+00	-.102073952000E-01	.194710358828E-01
.5000	.367879441171E+00	-.291607044110E-01	-.273994411714E-01
.6000	.301194211912E+00	-.694970431920E-02	.314137880878E-01
.7000	.246596963942E+00	-.298670140055E-01	-.391601639416E-01
.8000	.201896517995E+00	-.150407820120E-02	.477367620053E-01
.9000	.165298888222E+00	-.303276582224E-01	-.577154002216E-01
1.0000	.135335283237E+00	.517083608217E-02	.712646015634E-01

7.5.2 Contraharmonic Mean (C_oM) 2 - Step Method

Now, we attempt to replace $2f_n(x,y)$ by $\left(\frac{f_n^2+f_{n-1}^2}{f_n+f_{n-1}}+\frac{f_n^2+f_{n+1}^2}{f_n+f_{n+1}}\right)$ in equation (7.5.1) to obtain a new formula defined by

$$y_{n+1}=y_{n-1}+h\left[\left(\frac{f_n^2+f_{n-1}^2}{f_n+f_{n-1}}\right)+\left(\frac{f_n^2+f_{n+1}^2}{f_n+f_{n+1}}\right)\right] \tag{7.5.20}$$

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 :

$$\left(\frac{f_n^2+f_{n-1}^2}{f_n+f_{n-1}}\right)=f_n-\frac{h}{2}f_n^{(1)}+\frac{h^2}{4}\left(f_n^{(2)}+\frac{\left(f_n^{(1)}\right)^2}{f_n}\right)+\frac{h^3}{24}\left(3\frac{\left(f_n^{(1)}\right)^3}{f_n^2}-6\frac{f_n^{(1)}f_n^{(2)}}{f_n}-2f_n^{(3)}\right)$$

$$+\frac{h^4}{48f_n^3}\left(3\left(f_n^{(1)}\right)^4-9f_n\left(f_n^{(1)}\right)^2f_n^{(2)}+3f_n^2\left(f_n^{(2)}\right)^2+4f_n^2f_n^{(1)}f_n^{(3)}+f_n^3f_n^{(4)}\right)+0(h^5)$$

$$\ldots \tag{7.5.21}$$

$$\left(\frac{f_n^2+f_{n+1}^2}{f_n+f_{n+1}}\right)=f_n+\frac{h}{2}f_n^{(1)}+\frac{h^2}{4}\left(f_n^{(2)}+\frac{\left(f_n^{(1)}\right)^2}{f_n}\right)+\frac{h^3}{24}\left(-3\frac{\left(f_n^{(1)}\right)^3}{f_n^2}+6\frac{f_n^{(1)}f_n^{(2)}}{f_n}+2f_n^{(3)}\right)$$

$$+\frac{h^4}{48f_n^3}\left(3\left(f_n^{(1)}\right)^4-9f_n\left(f_n^{(1)}\right)^2f_n^{(2)}+3f_n^2\left(f_n^{(2)}\right)^2+4f_n^2f_n^{(1)}f_n^{(3)}+f_n^3f_n^{(4)}\right)+0\left(h^5\right)$$

$$\ldots \qquad (7.5.22)$$

By adding equations (7.5.21) and (7.5.22), we have

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

$$+\frac{h^4}{24f_n^3}\left(3\left(f_n^{(1)}\right)^4-9f_n\left(f_n^{(1)}\right)^2f_n^{(2)}+3f_n^2\left(f_n^{(2)}\right)^2+4f_n^2f_n^{(1)}f_n^{(3)}+f_n^3f_n^{(4)}\right)+0\left(h^5\right)$$

$$\ldots \qquad (7.5.23)$$

and substituting equations (7.5.4) and (7.5.23) into equation (7.5.20), we obtain

$$y_{n+1}=f_nh+\frac{h^2}{2}f_n^{(1)}+\frac{h^3}{6}\left(2f_n^{(2)}+3\frac{\left(f_n^{(1)}\right)^2}{f_n}\right)+\frac{h^4}{24}f_n^{(3)}+0\left(h^5\right) \qquad (7.5.24)$$

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

$$LTE=\left[-\frac{\left(f_n^{(1)}\right)^2}{2f_n}-\frac{f_n^{(2)}}{6}\right]h^3+0\left(h^4\right)$$

$$=\frac{h^3}{6}\left[-3\frac{\left(f_n^{(1)}\right)^2}{f_n}-f_n^{(2)}\right]+0\left(h^4\right).$$

Stability of Formula (7.5.20)

By applying the stability method to test equation $y'=\lambda y$ in equation (7.5.20), we obtain

$$y_{n+1} = y_{n-1} + h\lambda\left[\left(\frac{y_n^2 + y_{n-1}^2}{y_n + y_{n-1}}\right) + \left(\frac{y_n^2 + y_{n+1}^2}{y_n + y_{n+1}}\right)\right] \tag{7.5.25}$$

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

$$\mu^{n+1} = \mu^{n-1} + h\lambda\left[\left(\frac{\mu^{2n} + \mu^{2n-2}}{\mu^n + \mu^{n-1}}\right) + \left(\frac{\mu^{2n} + \mu^{2n+2}}{\mu^n + \mu^{n+1}}\right)\right]$$

$$= \mu^{n-1} + h\lambda\mu^n\left[\left(\frac{1+\mu^{-2}}{1+\mu^{-1}}\right) + \left(\frac{1+\mu^2}{1+\mu}\right)\right]$$

or

$$\mu^n\left(\mu - \frac{1}{\mu} - h\lambda\left[\frac{\mu^3 + \mu^2 + \mu + 1}{\mu(1+\mu)}\right]\right) = 0$$

$$\mu^3 + \mu^2 = 1 + \mu + h\lambda(\mu^3 + \mu^2 + \mu + 1) \tag{7.5.26}$$

By solving equation (7.5.26) using Mathematica i.e.,

$$In[2] := Solve[(1-h\lambda)\mu \wedge 3 + (1-h\lambda)\mu^2 - (1+h\lambda)\mu - (1+h\lambda) == 0, \mu]$$

$$Out[2] := \mu -> -1 \quad , \quad \mu -> \pm\sqrt{\left(\frac{2}{1-h\lambda}\right) - 1} \tag{7.5.27}$$

To determine the stability region of the second order implicit contraharmonic mean (C_oM) formula (7.5.25) in the complex plane that satisfy the condition

$$|\mu^n| < 1$$

i.e.,

$$|\mu^n| = \left|\sqrt{\left(\frac{2}{1-h\lambda}\right) - 1}\right| < 1 \tag{7.5.28}$$

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.

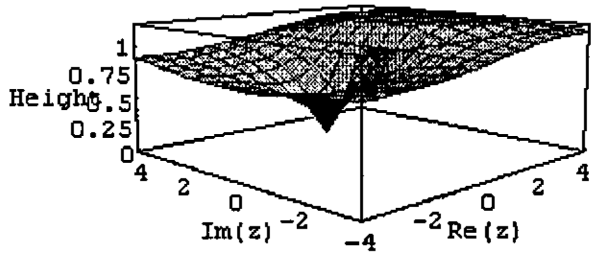


Figure 7.3: Graphic surface defined by equation (7.5.28)

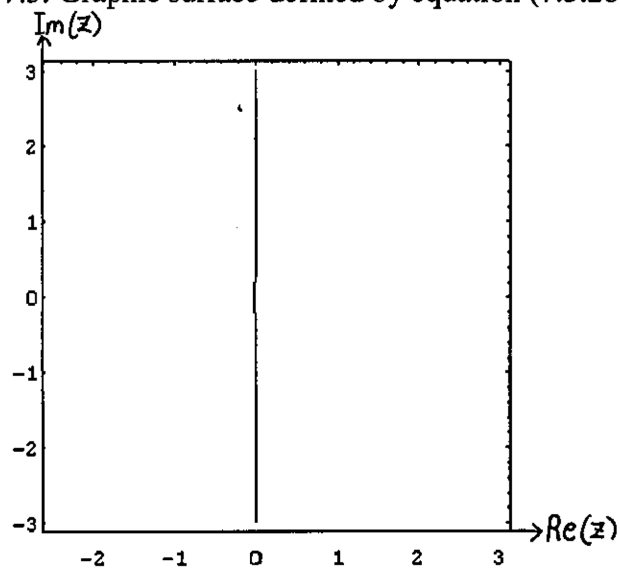


Figure 7.4: Stability region for second order implicit C_oM method

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 7.18: Error by using the C_oM second order implicit formulae for solving (7.5.19)

xn	Numerical Solution	Exact Solution	Absolute Error
.1000	.800000000000E+00	.818730753078E+00	-.187307530780E-01
.2000	.657777777778E+00	.670320046036E+00	-.125422682579E-01
.3000	.521279132791E+00	.548811636094E+00	-.275325033027E-01
.4000	.434036024305E+00	.449328964117E+00	-.152929398123E-01
.5000	.338143674105E+00	.367879441171E+00	-.297357670661E-01
.6000	.287998490248E+00	.301194211912E+00	-.131957216638E-01
.7000	.217528167201E+00	.246596963942E+00	-.290687967403E-01
.8000	.192957836056E+00	.201896517995E+00	-.893868193837E-02
.9000	.137740929809E+00	.165298888222E+00	-.275579584129E-01
1.0000	.131417814599E+00	.135335283237E+00	-.391746863806E-02

7.5.3 Centroidal Mean $(C_e M)$ 2 - Step Method

By replacing $2f_n(x,y)$ with $\left[\dfrac{2\left(f_n^2+f_nf_{n-1}+f_{n-1}^2\right)}{3\left(f_n+f_{n-1}\right)}+\dfrac{2\left(f_n^2+f_nf_{n+1}+f_{n+1}^2\right)}{3\left(f_n+f_{n+1}\right)}\right]$

in equation (7.5.1) we obtain a new implicit method as

$$y_{n+1}=y_{n-1}+h\left[\left(\frac{2\left(f_n^2+f_nf_{n-1}+f_{n-1}^2\right)}{3\left(f_n+f_{n+1}\right)}\right)+\left(\frac{2\left(f_n^2+f_nf_{n+1}+f_{n+1}^2\right)}{3\left(f_n+f_{n+1}\right)}\right)\right] \qquad (7.5.29)$$

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 :

$$\frac{2\left(f_n^2+f_nf_{n-1}+f_{n-1}^2\right)}{3\left(f_n+f_{n-1}\right)}=f_n-\frac{h}{2}f_n^{(1)}+\frac{h^2}{12}\left(3f_n^{(2)}+\frac{\left(f_n^{(1)}\right)^2}{f_n}\right)+\frac{h^3}{24}\left(\frac{\left(f_n^{(1)}\right)^3}{f_n^2}-2\frac{f_n^{(1)}f_n^{(2)}}{f_n}-2f_n^{(3)}\right)$$
$$+\frac{h^4}{144f_n^3}\left(3\left(f_n^{(1)}\right)^4-9f_n\left(f_n^{(1)}\right)^2f_n^{(2)}+3f_n^2\left(f_n^{(2)}\right)^2+4f_n^2f_n^{(1)}f_n^{(3)}+3f_n^3f_n^{(4)}\right)+0\left(h^5\right)$$
$$\ldots \qquad (7.5.30)$$

$$\frac{2\left(f_n^2+f_nf_{n+1}+f_{n+1}^2\right)}{3\left(f_n+f_{n+1}\right)}=f_n+\frac{h}{2}f_n^{(1)}+\frac{h^2}{12}\left(3f_n^{(2)}+\frac{\left(f_n^{(1)}\right)^2}{f_n}\right)+\frac{h^3}{24}\left(-\frac{\left(f_n^{(1)}\right)^3}{f_n^2}+2\frac{f_n^{(1)}f_n^{(2)}}{f_n}+2f_n^{(3)}\right)$$
$$+\frac{h^4}{144f_n^3}\left(3\left(f_n^{(1)}\right)^4-9f_n\left(f_n^{(1)}\right)^2f_n^{(2)}+3f_n^2\left(f_n^{(2)}\right)^2+4f_n^2f_n^{(1)}f_n^{(3)}+3f_n^3f_n^{(4)}\right)+0\left(h^5\right)$$
$$\ldots \qquad (7.5.31)$$

By substituting equations (7.5.4) and (7.5.30)-(7.5.31) into equation (7.5.29), we have

$$y_{n+1}=f_nh+\frac{h^2}{2}f_n^{(1)}+\frac{h^3}{6}\left(2f_n^{(2)}+\frac{\left(f_n^{(1)}\right)^2}{f_n}\right)+\frac{h^4}{24}f_n^{(3)}+0\left(h^5\right) \qquad (7.5.32)$$

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

$$LTE = \left[-\frac{\left(f_n^{(1)}\right)^2}{6f_n} - \frac{f_n^{(2)}}{6} \right] h^3 + 0\left(h^4\right)$$

$$= -\frac{h^3}{6}\left[\frac{\left(f_n^{(1)}\right)^2}{f_n} + f_n^{(2)} \right] + 0\left(h^4\right)$$

Stability of Formula (7.5.29)

By substituting the test equation $y' = \lambda y$ into equation (7.5.29), we have

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

$$= y_{n-1} + \frac{2}{3}h\lambda\left[\left(\frac{\left(y_n^2 + y_n y_{n-1} + y_{n-1}^2\right)}{\left(y_n + y_{n+1}\right)}\right) + \left(\frac{\left(y_n^2 + y_n y_{n+1} + y_{n+1}^2\right)}{\left(y_n + y_{n+1}\right)}\right)\right] \qquad (7.5.33)$$

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

$$\mu^{n+1} = \mu^{n-1} + \frac{2}{3}h\lambda\left[\left(\frac{\mu^{2n} + \mu^n\mu^{n-1} + \mu^{2n-2}}{\mu^n + \mu^{n-1}}\right) + \left(\frac{\mu^{2n} + \mu^n\mu^{n+1} + \mu^{2n+2}}{\mu^n + \mu^{n+1}}\right)\right]$$

and divided by μ^n we obtain

$$\mu = \frac{1}{\mu} + \frac{2}{3}h\lambda\left[\left(\frac{1 + \mu^{-1} + \mu^{-2}}{1 + \mu^{-1}}\right) + \left(\frac{1 + \mu + \mu^2}{1 + \mu}\right)\right]$$

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

$$\left(1 - \frac{2}{3}h\lambda\right)\mu^3 + \left(1 - \frac{4}{3}h\lambda\right)\mu^2 - \left(1 + \frac{4}{3}h\lambda\right)\mu - \left(1 + \frac{2}{3}h\lambda\right) = 0 \qquad (7.5.34)$$

By solving equation (7.5.34) using Mathematica, i.e.,

$$In[3]:= Solve\left[\left(1-\frac{2}{3}h\lambda\right)\mu^3+\left(1-\frac{4}{3}h\lambda\right)\mu^2-\left(1+\frac{4}{3}h\lambda\right)\mu-\left(1+\frac{2}{3}h\lambda\right)==0,\mu\right]$$

$$Out[3]:=\mu->-1 \quad , \quad \mu->\frac{-1+\frac{3}{3-2h\lambda}\pm\frac{2\sqrt{3}\sqrt{(3-(h\lambda)^2)}}{-3+2h\lambda}}{2} \qquad (7.5.35)$$

To determine the stability region of the second order implicit centroidal mean (C_eM) formula (7.5.35) in the complex plane that satisfy the condition

$$|\mu^n|<1$$

i.e.,

$$|\mu^n|=\left|\frac{-1+\frac{3}{3-2h\lambda}+\frac{2\sqrt{3}\sqrt{3-(h\lambda)^2}}{-3+2h\lambda}}{2}\right|<1 \qquad (7.5.36)$$

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.

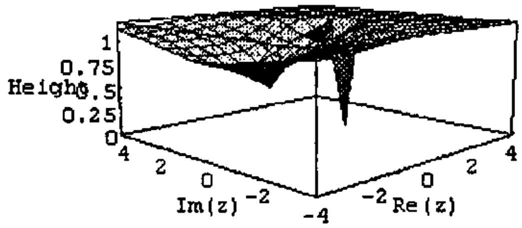


Figure 7.5: Graphic surface defined by equation (7.5.36)

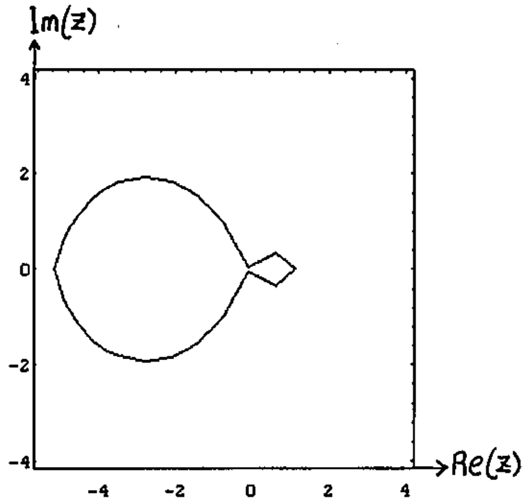


Figure 7.6: Stability region for second order implicit C_eM method

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_eM second order implicit formulae for solving (7.5.19).

xn	Numerical Solution	Exact Solution	Absolute Error
.1000	.800000000000E+00	.818730753078E+00	-.187307530780E-01
.2000	.659259259259E+00	.670320046036E+00	-.110607867764E-01
.3000	.521769756220E+00	.548811636094E+00	-.270418798742E-01
.4000	.436268878411E+00	.449328964117E+00	-.130600857060E-01
.5000	.338457764072E+00	.367879441171E+00	-.294216770998E-01
.6000	.290693031682E+00	.301194211912E+00	-.105011802303E-01
.7000	.217283202546E+00	.246596963942E+00	-.293137613956E-01
.8000	.196085142085E+00	.201896517995E+00	-.581137591002E-02
.9000	.136693104206E+00	.165298888222E+00	-.286057840159E-01
1.0000	.135115366812E+00	.135335283237E+00	-.219916424883E-03

7.5.4 Harmonic Mean (H_aM) 2 - Step Method

Finally, we attempt to replace $2f_n(x,y)$ by $\left[\frac{2f_nf_{n-1}}{f_n+f_{n-1}}+\frac{2f_nf_{n+1}}{f_n+f_{n+1}}\right]$ in equation (7.5.1) to obtain a new implicit method as

$$y_{n+1} = y_{n-1} + h\left[\left(\frac{2f_nf_{n-1}}{f_n + f_{n-1}}\right) + \left(\frac{2f_nf_{n+1}}{f_n + f_{n+1}}\right)\right] \tag{7.5.37}$$

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 :

$$\frac{2f_nf_{n-1}}{f_n + f_{n-1}} = f_n - \frac{h}{2}f_n^{(1)} + \frac{h^2}{4}\left(f_n^{(2)} - \frac{\left(f_n^{(1)}\right)^2}{f_n}\right) + \frac{h^3}{24}\left(-3\frac{\left(f_n^{(1)}\right)^3}{f_n^2} + 6\frac{f_n^{(1)}f_n^{(2)}}{f_n} - 2f_n^{(3)}\right)$$

$$+\frac{h^4}{48f_n^3}\left(-3\left(f_n^{(1)}\right)^4 + 9f_n\left(f_n^{(1)}\right)^2 f_n^{(2)} - 3f_n^2\left(f_n^{(2)}\right)^2 - 4f_n^2f_n^{(1)}f_n^{(3)} + f_n^3f_n^{(4)}\right) + 0\left(h^5\right)$$

$$\ldots \tag{7.5.38}$$

$$\frac{2f_nf_{n+1}}{f_n + f_{n+1}} = f_n + \frac{h}{2}f_n^{(1)} + \frac{h^2}{4}\left(f_n^{(2)} - \frac{\left(f_n^{(1)}\right)^2}{f_n}\right) + \frac{h^3}{24}\left(3\frac{\left(f_n^{(1)}\right)^3}{f_n^2} - 6\frac{f_n^{(1)}f_n^{(2)}}{f_n} + 2f_n^{(3)}\right)$$

$$+\frac{h^4}{48f_n^3}\left(-3\left(f_n^{(1)}\right)^4 + 9f_n\left(f_n^{(1)}\right)^2 f_n^{(2)} - 3f_n^2\left(f_n^{(2)}\right)^2 - 4f_n^2f_n^{(1)}f_n^{(3)} + f_n^3f_n^{(4)}\right) + 0\left(h^5\right)$$

$$\ldots \tag{7.5.39}$$

By substituting equations (7.5.4) and (7.5.38)-(7.5.39) into equation (7.5.37), we have

$$y_{n+1} = f_nh + \frac{h^2}{2}f_n^{(1)} + \frac{h^3}{6}\left(2f_n^{(2)} - 3\frac{\left(f_n^{(1)}\right)^2}{f_n}\right) + \frac{h^4}{24}f_n^{(3)} + 0\left(h^5\right) \tag{7.5.40}$$

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

$$LTE = \left[\frac{\left(f_n^{(1)}\right)^2}{2f_n} - \frac{f_n^{(2)}}{6}\right]h^3 + 0\left(h^4\right)$$

$$= \frac{h^3}{6}\left[3\frac{\left(f_n^{(1)}\right)^2}{f_n} - f_n^{(2)}\right] + 0\left(h^4\right) \tag{7.5.41}$$

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

$$y_{n+1} = y_{n-1} + h\lambda\left[\left(\frac{2y_n y_{n-1}}{y_n + y_{n-1}}\right) + \left(\frac{2y_n y_{n+1}}{y_n + y_{n+1}}\right)\right] \qquad (7.5.42)$$

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

$$\mu^{n+1} = \mu^{n-1} + 2h\lambda\left[\frac{\mu^n \mu^{n-1}}{\mu^n + \mu^{n-1}} + \frac{\mu^n \mu^{n+1}}{\mu^n + \mu^{n+1}}\right]$$

$$\mu = \frac{1}{\mu} + 2h\lambda\left[\frac{\mu^{-1}}{1+\mu^{-1}} + \frac{\mu}{1+\mu}\right] \qquad (7.5.43)$$

where

$$\frac{\mu^{-1}}{1+\mu^{-1}} = \frac{1}{1+\mu}.$$

Therefore, equation (7.5.43) can be written as

$$\mu^2 - 2h\lambda\mu - 1 = 0$$

and solving using Mathematica , i.e.,

$$In[4] := Solve[\mu\wedge 2 - 2\ h\lambda\ \mu - 1 == 0, \mu]$$

$$Out[4] := \mu -> \frac{2h\lambda \pm 2\sqrt{1+(h\lambda)^2}}{2} = h\lambda \pm \sqrt{1+(h\lambda)^2} \qquad (7.5.44)$$

To determine the stability region of the second order implicit Harmonic mean $(H_a M)$ formula (7.5.44) in the complex plane that satisfy the condition

$$|\mu^n| < 1$$

i.e.,

$$|\mu^n| = \left|h\lambda + \sqrt{1+(h\lambda)^2}\right| < 1 \qquad (7.5.45)$$

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.

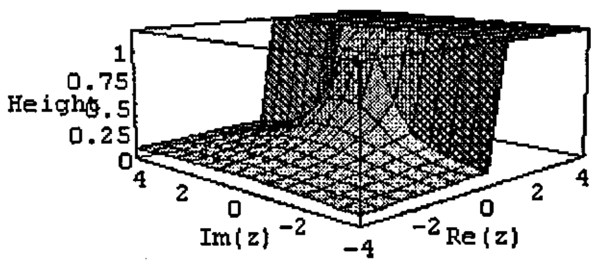


Figure 7.7: Graphic surface defined by equation (7.5.45)

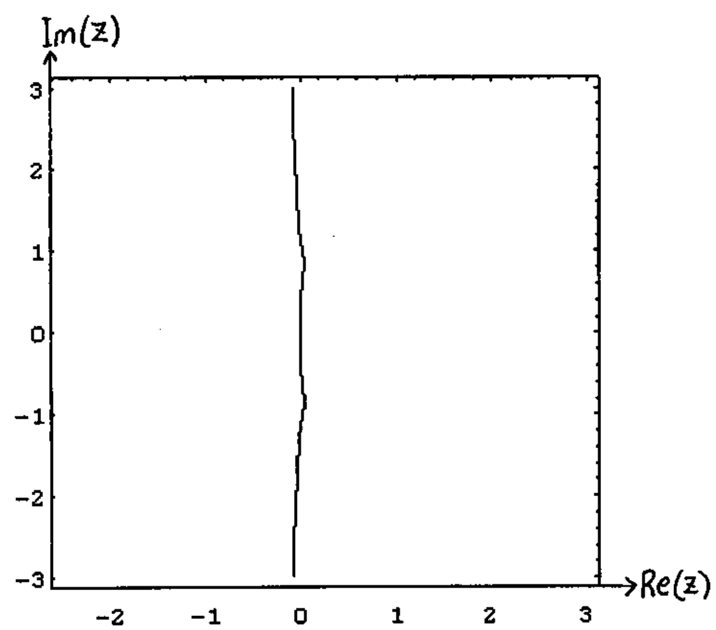


Figure 7.8: Stability region for second order implicit H_aM method

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_aM second order implicit formulae for solving (7.5.19).

xn	Numerical Solution	Exact Solution	Absolute Error
.1000	.800000000000E+00	.818730753078E+00	-.187307530780E-01
.2000	.662222222222E+00	.670320046036E+00	-.809782381342E-02
.3000	.522631543398E+00	.548811636094E+00	-.261800926965E-01
.4000	.440855090780E+00	.449328964117E+00	-.847387333750E-02
.5000	.338805943918E+00	.367879441171E+00	-.290734972539E-01
.6000	.296463510831E+00	.301194211912E+00	-.473070108102E-02
.7000	.216268520083E+00	.246596963942E+00	-.303284438582E-01
.8000	.203190911266E+00	.201896517995E+00	.129439327177E-02
.9000	.133725167137E+00	.165298888222E+00	-.315737210847E-01
1.0000	.144186522692E+00	.135335283237E+00	.885123945553E-02

From the above results and discussions, it was shown that the geometric mean (GM), contraharmonic mean (C_oM), centroidal mean (C_eM) and harmonic mean (H_aM) 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

Number of Stages	Highest Order
1	1
2	2
3	3
4	4
5	4
6	5
7	6
8	6
9	7
R	R - 2 R = 10 , 11

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_oM) 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_oM), centroidal mean (C_eM) 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

$$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3} \text{Means}\right] \qquad (8.0.1)$$

where *Means* = principal means include arithmetic mean (AM), geometric mean (GM) [1987], contraharmonic mean (C_oM) [1993], centroidal mean (C_eM) [1993], root mean square (RMS) [1993], harmonic mean (H_aM) [1993] and heronian mean (H_eM) [1993] which involve $k_i, 1 < i < 4$ where

$$\begin{aligned}
k_1 &= f(x_n, y_n) \\
k_2 &= f(x_n + a_1 h, y_n + a_1 h k_1) \\
k_3 &= f(x_n + (a_2 + a_3)h, y_n + a_2 h k_1 + a_3 h k_2) \\
k_4 &= f(x_n + (a_4 + a_5 + a_6)h, y_n + a_4 h k_1 + a_5 h k_2 + a_6 h k_3)
\end{aligned} \qquad (8.0.2)$$

From the above discussion, a comparison of the parameters $a_i, 1 \le i \le 6$ in equation (8.0.2), show that for the parameters, a_1 is fixed, a_2, a_4, a_6 are decreasing and a_3, a_5 are increasing are shown in Table 8.2.

Table 8.2 : The values of parameters $a_i, 1 \le i \le 6$ based on the various formulas

	C_oM	C_eM	RMS	AM	GM	H_eM	H_aM
a_1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
a_2	$\frac{1}{8}$	$\frac{1}{24}$	$\frac{1}{16}$	0	$-\frac{1}{16}$	$-\frac{1}{48}$	$-\frac{1}{8}$
a_3	$\frac{3}{8}$	$\frac{11}{24}$	$\frac{7}{16}$	$\frac{1}{2}$	$\frac{9}{16}$	$\frac{25}{48}$	$\frac{5}{8}$
a_4	$\frac{1}{4}$	$\frac{1}{12}$	$\frac{1}{8}$	0	$-\frac{1}{8}$	$-\frac{1}{24}$	$-\frac{1}{4}$
a_5	$-\frac{3}{4}$	$-\frac{25}{132}$	$-\frac{17}{56}$	0	$\frac{5}{24}$	$\frac{47}{600}$	$\frac{7}{20}$
a_6	$\frac{3}{2}$	$\frac{73}{66}$	$\frac{33}{28}$	1	$\frac{11}{12}$	$\frac{289}{300}$	$\frac{9}{10}$

From Table 8.2, we can see that $a_2 + a_3 = \frac{1}{2}$ and $a_4 + a_5 + a_6 = 1$. In the following discussion, our concern is to establish a new fifth order formula based on the last two k values i.e., k_3 and k_4 which involve the parameters $a_2 + a_3 = \frac{1}{2}$ and $a_4 + a_5 + a_6 = 1$ and are weighted so we can establish a new weighted Runge-Kutta formula (WRK) using both the arithmetic mean (AM) and contraharmonic mean (C_oM).

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

$$y_{n+1} = y_n + \frac{h}{3}\left[\sum_{i=1}^{3}\left(\frac{k_i + k_{i+1}}{2}\right)\right] \tag{8.1.1}$$

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

where $k_i, 1 \le i \le 4$ as we mentioned above.

Now , based on the parameters involved in the last two k values i.e., k_3 and k_4 , we attempt to develop a new fourth order method called the weighted Runge-Kutta (WRK) formula in the form

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{3} w_i\left(\frac{k_i + k_{i+1}}{2}\right)\right] \tag{8.1.2}$$

where $\sum_{i=1}^{3} w_i = 1$, $k_1 = f(y_n)$

$$k_2 = f(y_n + a_1 hk_1)$$
$$k_3 = f(y_n + a_2 hk_1 + (\tfrac{1}{2} - a_2)hk_2)$$
$$k_4 = f(y_n + a_3 hk_1 + a_4 hk_2 + (1 - a_3 - a_4)hk_3). \tag{8.1.3}$$

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: 1. - x(1) - x(2) - x(3) = 0 , (8.1.4-i)

$h^2 ff_y$: 2. - 2.*x(4)*x(1) - x(2) - 2.*x(4)*x(2) - 3.*x(3) = 0 , (8.1.4-ii)

$h^3 ff_y^2$: 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, (8.1.4-iii)

$h^3 f^2 f_{yy}$:   8. - 12.*x(4)**2*x(1) - 3.*x(2) - 12.*x(4)**2*x(2)
- 15.*x(3) = 0 ,   (8.1.4-iv)

$h^4 f f_y^3$:   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) = 0 ,   (8.1.4-v)

$h^4 f^2 f_y f_{yy}$:   8. - 6.*x(4)*x(2) - 6.*x(4)**2*x(2) +
12.*x(4)*x(5)*x(2) + 12.*x(4)**2*x(5)*x(2) - 15.*x(3) -
6.*x(4)*x(3) - 6.*x(4)**2*x(3) + 12.*x(4)*x(5)*x(3) +
12.*x(4)**2*x(5)*x(3) + 15.*x(6)*x(3) + 15.*x(7)*x(3) -
24.*x(4)*x(7)*x(3) - 12.*x(4)**2*x(7)*x(3) = 0 ,   (8.1.4-vi)

$h^4 f^3 f_{yyy}$:   4. - 8.*x(4)**3*x(1) - x(2) - 9.*x(3) - 8.*x(4)**3*x(2) = 0   (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
$x(6) = a_3$, $x(7) = a_4$.

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.,

$$x(1) = w_1 = 0.3333333333, \quad x(2) = w_2 = 0.3333333333, \quad x(3) = w_3 = 0.3333333333$$
$$x(4) = a_1 = 0.5000000000 \quad , \quad x(5) = a_2 = 0 \quad , \quad x(6) = a_3 = 0 \quad , \quad x(7) = a_4 = 0$$

Thus, this new WRK method gives the same result as the standard fourth order arithmetic mean (AM) Runge-Kutta method $y_{n+1} = y_n + \frac{h}{3}\left[\frac{k_1 + k_2}{2} + \frac{k_2 + k_3}{2} + \frac{k_3 + k_4}{2}\right]$

where $k_1 = f(y_n)$

$$k_2 = f\left(y_n + \frac{1}{2}hk_1\right)$$

$$k_3 = f\left(y_n + \frac{1}{2}hk_2\right)$$

$$k_4 = f(y_n + hk_3) \; .$$

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

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i + k_{i+1}}{2}\right)\right] \qquad (8.2.1)$$

where $\sum_{i=1}^{4} w_i = 1$ and

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + a_1hk_1)$$

$$k_3 = f(y_n + a_2hk_1 + a_3hk_2)$$

$$k_4 = f\left(y_n + a_4hk_1 + a_5hk_2 + \left(\tfrac{1}{2} - a_4 - a_5\right)hk_3\right)$$

$$k_5 = f\left(y_n + a_6hk_1 + a_7hk_2 + a_8hk_3 + (1 - a_6 - a_7 - a_8)hk_4\right) \qquad (8.2.2)$$

The Taylor series expansion of $y(x_{n+h})$ up to sixth order is given by

$$y(x_n + h) = y_n + hf + \frac{1}{2}h^2 ff_y + \frac{1}{6}h^3\left(ff_y^2 + f^2 f_{yy}\right)$$

$$+\frac{1}{24}h^4\left(f^3 f_{yyy} + 4f^2 f_y f_{yy} + ff_y^3\right)$$

$$+\frac{1}{120}h^5\left(ff_y^4 + 11f^2 f_y^2 f_{yy} + 4f^3 f_{yy}^2 + 7f^3 f_y f_{yyy} + f^4 f_{yyyy}\right)$$

$$+\frac{1}{720}h^6\left(f^5 f_{yyyyy} + 11f^4 f_y f_{yyyy} + 15f^4 f_{yy} f_{yyy} + 32f^3 f_y^2 f_{yyy}\right.$$

$$\left. + 34f^3 f_y f_{yy}^2 + 26f^2 f_y^3 f_{yy} + ff_y^5\right) + 0(h^7) \quad \ldots \qquad (8.2.3)$$

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: 1 - x(1) - x(2) - x(3) - x(4) = 0 , (8.2.4-i)

$h^2 ff_y$: 2 - 2*x(5)*x(1) - 2*x(5)*x(2) - 2*x(6)*x(2) -
2*x(7)*x(2)- x(3) - 2*x(6)*x(3) - 2*x(7)*x(3) - 3*x(4) = 0 , (8.2.4-ii)

$h^3 ff_y^2$: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) +

6*x(6)*x(8)*x(4) + 6*x(7)*x(8)*x(4) - 6*x(5)*x(9)*
x(4) + 6*x(6)*x(9)*x(4) + 6*x(7)*x(9)*x(4) +
3*x(10)*x(4) + 3*x(11)*x(4) - 6*x(5)*x(11)*x(4) +
3*x(12)*x(4) - 6*x(6)*x(12)*x(4) - 6*x(7)*x(12)*x(4) = 0 , (8.2.4-iii)

$h^3 f^2 f_{yy}$: 8 - 12*x(5)**2*x(1) - 12*x(5)**2*x(2) - 12*x(6)**2*x(2)
- 24*x(6)*x(7)*x(2) - 12*x(7)**2*x(2) - 3*x(3) -
12*x(6)**2*x(3) - 24*x(6)*x(7)*x(3)-12*x(7)**2*x(3)-15*x(4) = 0 ,
... (8.2.4-iv)

$h^4 f f_y^3$: 1 - 6*x(5)*x(7)*x(3) + 12*x(5)*x(7)*x(8)*x(3)
+ 12*x(5)*x(7)*x(9)*x(3) - 6*x(6)*x(4) - 6*x(7)*x(4) -
6*x(5)*x(7)*x(4) + 12*x(6)*x(8)*x(4) + 12*x(7)*x(8)*x(4)
+ 12*x(5)*x(7)*x(8)*x(4) - 12*x(5)*x(9)*x(4) +
12*x(6)*x(9)*x(4) + 12*x(7)*x(9)*x(4) + 12*x(5)*x(7)*x(9)*x(4)
+ 6*x(6)*x(10)*x(4) + 6*x(7)*x(10)*x(4) -
12*x(6)*x(8)*x(10)*x(4) - 12*x(7)*x(8)*x(10)*x(4) +
12*x(5)*x(9)*x(10)*x(4) - 12*x(6)*x(9)*x(10)*x(4) -
12*x(7)*x(9)*x(10)*x(4) + 6*x(6)*x(11)*x(4) +
6*x(7)*x(11)*x(4) - 12*x(6)*x(8)*x(11)*x(4) -
12*x(7)*x(8)*x(11)*x(4) + 12*x(5)*x(9)*x(11)*x(4) -
12*x(6)*x(9)*x(11)*x(4) - 12*x(7)*x(9)*x(11)*x(4) +
6*x(6)*x(12)*x(4) + 6*x(7)*x(12)*x(4) -
12*x(5)*x(7)*x(12)*x(4) - 12*x(6)*x(8)*x(12)*x(4) -
12*x(7)*x(8)*x(12)*x(4) + 12*x(5)*x(9)*x(12)*x(4) -
12*x(6)*x(9)*x(12)*x(4) - 12*x(7)*x(9)*x(12)*x(4) = 0 , (8.2.4-v)

$h^4 f^2 f_y f_{yy}$: 8 - 12*x(5)**2*x(7)*x(2) - 24*x(5)*x(6)*x(7)*x(2) -
24*x(5)*x(7)**2*x(2) - 6*x(6)*x(3) - 6*x(6)**2*x(3) -
6*x(7)*x(3) - 12*x(5)**2*x(7)*x(3) - 12*x(6)*x(7)*x(3) -
24*x(5)*x(6)*x(7)*x(3) - 6*x(7)**2*x(3) -
24*x(5)*x(7)**2*x(3) + 12*x(6)*x(8)*x(3) +
12*x(6)**2*x(8)*x(3) + 12*x(7)*x(8)*x(3) +
24*x(6)*x(7)*x(8)*x(3) + 12*x(7)**2*x(8)*x(3) -
12*x(5)*x(9)*x(3) - 12*x(5)**2*x(9)*x(3) +
12*x(6)*x(9)*x(3) + 12*x(6)**2*x(9)*x(3) + 12*x(7)*x(9)*x(3)
+ 24*x(6)*x(7)*x(9)*x(3) +
12*x(7)**2*x(9)*x(3) - 15*x(4) - 6*x(6)*x(4) -
6*x(6)**2*x(4) - 6*x(7)*x(4) - 12*x(6)*x(7)*x(4) -
6*x(7)**2*x(4) + 12*x(6)*x(8)*x(4) + 12*x(6)**2*x(8)*x(4) +
12*x(7)*x(8)*x(4) + 24*x(6)*x(7)*x(8)*x(4) +
12*x(7)**2*x(8)*x(4) - 12*x(5)*x(9)*x(4) -
12*x(5)**2*x(9)*x(4) + 12*x(6)*x(9)*x(4) +
12*x(6)**2*x(9)*x(4) + 12*x(7)*x(9)*x(4) +
24*x(6)*x(7)*x(9)*x(4)+ 12*x(7)**2*x(9)*x(4)+15*x(10)*x(4) +
15*x(11)*x(4) - 24*x(5)*x(11)*x(4) - 12*x(5)**2*x(11)*x(4) +
15*x(12)*x(4) - 24*x(6)*x(12)*x(4) -
12*x(6)**2*x(12)*x(4) - 24*x(7)*x(12)*x(4) -
24*x(6)*x(7)*x(12)*x(4) - 12*x(7)**2*x(12)*x(4) = 0 , (8.2.4-vi)

$h^4 f^3 f_{yyy}$: 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 , (8.2.4-vii)

$h^5 f f_y^4$: 1 - 30*x(5)*x(7)*x(4) + 60*x(5)*x(7)*x(8)*x(4) +
60*x(5)*x(7)*x(9)*x(4) + 30*x(5)*x(7)*x(10)*x(4) -
60*x(5)*x(7)*x(8)*x(10)*x(4) - 60*x(5)*x(7)*x(9)*x(10)*x(4)
+ 30*x(5)*x(7)*x(11)*x(4) + 30*x(5)*x(7)*x(12)*x(4) -
60*x(5)*x(7)*x(8)*x(11)*x(4) - 60*x(5)*x(7)*x(9)*x(11)*x(4)
- 60*x(5)*x(7)*x(8)*x(12)*x(4) - 60*x(5)*x(7)*x(9)*x(12)*x(4) = 0 ,
... (8.2.4-viii)

$h^5 f^2 f_y^2 f_{yy}$: 22 - 60*x(5)**2*x(7)**2*x(2) - 15*x(6)**2*x(3)
- 30*x(5)*x(7)*x(3) - 30*x(5)**2*x(7)*x(3) -
30*x(6)*x(7)*x(3) - 60*x(5)*x(6)*x(7)*x(3) -
15*x(7)**2* x(3) - 60*x(5)*x(7)**2*x(3) -
60*x(5)**2*x(7)**2*x(3) + 60*x(6)**2*x(8)*x(3) +
60*x(5)*x(7)*x(8)*x(3) + 60*x(5)**2*x(7)*x(8)*x(3) +
120*x(6)*x(7)*x(8)*x(3) + 120*x(5)*x(6)*x(7)*x(8)*x(3) +
60*x(7)**2*x(8)*x(3) +
120*x(5)*x(7)**2*x(8)*x(3) - 60*x(6)**2*x(8)**2*x(3) -
120*x(6)*x(7)*x(8)**2*x(3) -
60*x(7)**2*x(8)**2*x(3) - 60*x(5)*x(6)*x(9)*x(3) +
60*x(6)**2*x(9)*x(3) + 60*x(5)**2*x(7)*x(9)*x(3) +
120*x(6)*x(7)*x(9)*x(3) + 120*x(5)*x(6)*x(7)*x(9)*x(3)
+ 60*x(7)**2*x(9)*x(3) +
120*x(5)*x(7)**2*x(9)*x(3) + 120*x(5)*x(6)*x(8)*x(9)*x(3)
- 120*x(6)**2*x(8)*x(9)*x(3) +
120*x(5)*x(7)*x(8)*x(9)*x(3) - 240*x(6)*x(7)*x(8)*x(9)*x(3)
- 120*x(7)**2*x(8)*x(9)*x(3) -
60*x(5)**2*x(9)**2*x(3) + 120*x(5)*x(6)*x(9)**2*x(3) -
60*x(6)**2*x(9)**2*x(3) +
120*x(5)*x(7)*x(9)**2*x(3) - 120*x(6)*x(7)*x(9)**2*x(3) -
60*x(7)**2*x(9)**2*x(3) - 15*x(4) -
90*x(6)*x(4) - 45*x(6)**2*x(4) - 90*x(7)*x(4) -
30*x(5)*x(7)*x(4) - 30*x(5)**2*x(7)*x(4) -
90*x(6)*x(7)*x(4) - 60*x(5)*x(6)*x(7)*x(4) - 45*x(7)**2
*x(4) - 60*x(5)*x(7)**2*x(4) +
180*x(6)*x(8)*x(4) + 120*x(6)**2*x(8)*x(4) + 180*x(7)*
x(8)*x(4)+ 60*x(5)*x(7)*x(8)*x(4) +
60*x(5)**2*x(7)*x(8)*x(4) + 240*x(6)*x(7)*x(8)*x(4) +
120*x(5)*x(6)*x(7)*x(8)*x(4) +
120*x(7)**2*x(8)*x(4) + 120*x(5)*x(7)**2*x(8)*x(4) -
60*x(6)**2*x(8)**2*x(4) -
120*x(6)*x(7)*x(8)**2*x(4) - 60*x(7)**2*x(8)**2*x(4) -
180*x(5)*x(9)*x(4) - 60*x(5)**2*x(9)*x(4) +
180*x(6)*x(9)*x(4) - 60*x(5)*x(6)*x(9)*x(4) +
120*x(6)**2*x(9)*x(4) + 180*x(7)*x(9)*x(4) +
60*x(5)**2*x(7)*x(9)*x(4) + 240*x(6)*x(7)*x(9)*x(4) +

120*x(5)*x(6)*x(7)*x(9)*x(4) +
120*x(7)**2*x(9)*x(4) + 120*x(5)*x(7)**2*x(9)*x(4) +
120*x(5)*x(6)*x(8)*x(9)*x(4) -
120*x(6)**2*x(8)*x(9)*x(4) + 120*x(5)*x(7)*x(8)*x(9)*x(4) -
240*x(6)*x(7)*x(8)*x(9)*x(4) -
120*x(7)**2*x(8)*x(9)*x(4) - 60*x(5)**2*x(9)**2*x(4) +
120*x(5)*x(6)*x(9)**2*x(4) -
60*x(6)**2*x(9)**2*x(4) + 120*x(5)*x(7)*x(9)**2*x(4) -
120*x(6)*x(7)*x(9)**2*x(4) -
60*x(7)**2*x(9)**2*x(4) + 30*x(10)*x(4) + 90*x(6)*x(10)*
x(4) + 30*x(6)**2*x(10)*x(4) + 90*x(7)*x(10)*x(4) +
60*x(6)*x(7)*x(10)*x(4) + 30*x(7)**2*x(10)*x(4) -
180*x(6)*x(8)*x(10)*x(4) - 60*x(6)**2*x(8)*x(10)*x(4) -
180*x(7)*x(8)*x(10)*x(4) - 120*x(6)*x(7)*x(8)*x(10)*x(4)
- 60*x(7)**2*x(8)*x(10)*x(4) +
180*x(5)*x(9)*x(10)*x(4) + 60*x(5)**2*x(9)*x(10)*x(4) -
180*x(6)*x(9)*x(10)*x(4) - 60*x(6)**2*x(9)*x(10)*x(4) -
180*x(7)*x(9)*x(10)*x(4) - 120*x(6)*x(7)*x(9)*x(10)*x(4) -
60*x(7)**2*x(9)*x(10)*x(4) - 15*x(10)**2*x(4) +
30*x(11)*x(4) - 60*x(5)*x(11)*x(4) + 90*x(6)*x(11)*x(4) +
30*x(6)**2*x(11)*x(4) + 90*x(7)*x(11)*x(4) +
60*x(6)*x(7)*x(11)*x(4) + 30*x(7)**2*x(11)*x(4) -
180*x(6)*x(8)*x(11)*x(4) - 60*x(6)**2*x(8)*x(11)*x(4) -
180*x(7)*x(8)*x(11)*x(4) - 120*x(6)*x(7)*x(8)*x(11)*x(4) -
60*x(7)**2*x(8)*x(11)*x(4) +
180*x(5)*x(9)*x(11)*x(4) + 60*x(5)**2*x(9)*x(11)*x(4) -
180*x(6)*x(9)*x(11)*x(4) - 60*x(6)**2*x(9)*x(11)*x(4) -
180*x(7)*x(9)*x(11)*x(4) - 120*x(6)*x(7)*x(9)*x(11)*x(4) -
60*x(7)**2*x(9)*x(11)*x(4) - 30*x(10)*x(11)*x(4) +
60*x(5)*x(10)*x(11)*x(4) - 15*x(11)**2*x(4) +
60*x(5)*x(11)**2*x(4) - 60*x(5)**2*x(11)**2*x(4) +
30*x(12)*x(4) + 30*x(6)*x(12)*x(4) + 30*x(6)**2*x(12)*x(4)
+ 30*x(7)*x(12)*x(4) - 120*x(5)*x(7)*x(12)*x(4) -
60*x(5)**2*x(7)*x(12)*x(4) + 60*x(6)*x(7)*x(12)*x(4) -
120*x(5)*x(6)*x(7)*x(12)*x(4) + 30*x(7)**2*x(12)*x(4) -
120*x(5)*x(7)**2*x(12)*x(4) - 180*x(6)*x(8)*x(12)*x(4) -
60*x(6)**2*x(8)*x(12)*x(4) -
180*x(7)*x(8)*x(12)*x(4) - 120*x(6)*x(7)*x(8)*x(12)*x(4) -
60*x(7)**2*x(8)*x(12)*x(4) +
180*x(5)*x(9)*x(12)*x(4) + 60*x(5)**2*x(9)*x(12)*x(4) -
180*x(6)*x(9)*x(12)*x(4) - 60*x(6)**2*x(9)*x(12)*x(4) -
180*x(7)*x(9)*x(12)*x(4) - 120*x(6)*x(7)*x(9)*x(12)*x(4) -
60*x(7)**2*x(9)*x(12)*x(4) - 30*x(10)*x(12)*x(4) +
60*x(6)*x(10)*x(12)*x(4) + 60*x(7)*x(10)*x(12)*x(4) -
30*x(11)*x(12)*x(4) + 60*x(5)*x(11)*x(12)*x(4) +
60*x(6)*x(11)*x(12)*x(4) - 120*x(5)*x(6)*x(11)*x(12)*x(4) +
60*x(7)*x(11)*x(12)*x(4)-120*x(5)*x(7)*x(11)*x(12)*x(4)
- 15*x(12)**2*x(4) + 60*x(6)*x(12)**2*x(4) - 60*x(6)**2*x(12)**2*x(4) +
60*x(7)*x(12)**2*x(4) - 120*x(6)*x(7)*x(12)**2*x(4) -
60*x(7)**2*x(12)**2*x(4) = 0 , (8.2.4 - ix)

$h^5 f^3 f_{yy}^2$: 8 - 60*x(5)**2*x(6)*x(7)*x(2) - 60*x(5)**2*x(7)**2*x(2)
- 15*x(6)**2*x(3) - 30*x(6)*x(7)*x(3) -
60*x(5)**2*x(6)*x(7)*x(3) - 15*x(7)**2*x(3) -
60*x(5)**2*x(7)**2*x(3) + 30*x(6)**2*x(8)*x(3) +
60*x(6)*x(7)*x(8)*x(3) + 30*x(7)**2*x(8)*x(3) -
30*x(5)**2*x(9)*x(3) + 30*x(6)**2*x(9)*x(3) +
60*x(6)*x(7)*x(9)*x(3) + 30*x(7)**2*x(9)*x(3) - 15*x(4)
- 15*x(6)**2*x(4) - 30*x(6)*x(7)*x(4) -
15*x(7)**2*x(4) + 30*x(6)**2*x(8)*x(4) +
60*x(6)*x(7)*x(8)*x(4) + 30*x(7)**2*x(8)*x(4) -
30*x(5)**2*x(9)*x(4) + 30*x(6)**2*x(9)*x(4) +
60*x(6)*x(7)*x(9)*x(4) + 30*x(7)**2*x(9)*x(4) +
15*x(10)*x(4) + 15*x(11)*x(4) - 60*x(5)**2*x(11)*x(4) +
15*x(12)*x(4) - 60*x(6)**2*x(12)*x(4) -
120*x(6)*x(7)*x(12)*x(4) - 60*x(7)**2*x(12)*x(4) = 0 , (8.2.4 - x)

$h^5 f^3 f_y f_{yyy}$: 28 - 40*x(5)**3*x(7)*x(2) - 120*x(5)*x(6)**2*x(7)*x(2)
- 240*x(5)*x(6)*x(7)**2*x(2) -
120*x(5)*x(7)**3*x(2) - 15*x(6)*x(3) - 20*x(6)**3*x(3)
- 15*x(7)*x(3) - 40*x(5)**3*x(7)*x(3) -
60*x(6)**2*x(7)*x(3) - 120*x(5)*x(6)**2*x(7)*x(3) -
60*x(6)*x(7)**2*x(3) - 240*x(5)*x(6)*x(7)**2*x(3) -
20*x(7)**3*x(3) - 120*x(5)*x(7)**3*x(3) + 30*x(6)*x(8)*x(3)
+ 40*x(6)**3*x(8)*x(3) + 30*x(7)*x(8)*x(3) +
120*x(6)**2*x(7)*x(8)*x(3) + 120*x(6)*x(7)**2*x(8)*x(3) +
40*x(7)**3*x(8)*x(3) - 30*x(5)*x(9)*x(3) -
40*x(5)**3*x(9)*x(3) + 30*x(6)*x(9)*x(3) +
40*x(6)**3*x(9)*x(3) + 30*x(7)*x(9)*x(3) +
120*x(6)**2*x(7)*x(9)*x(3) + 120*x(6)*x(7)**2*x(9)*x(3)
+ 40*x(7)**3*x(9)*x(3) - 65*x(4) -
15*x(6)*x(4) - 20*x(6)**3*x(4) - 15*x(7)*x(4) -
60*x(6)**2*x(7)*x(4) - 60*x(6)*x(7)**2*x(4) -
20*x(7)**3*x(4) + 30*x(6)*x(8)*x(4) + 40*x(6)**3*x(8)*x(4)
+ 30*x(7)*x(8)*x(4) +
120*x(6)**2*x(7)*x(8)*x(4) + 120*x(6)*x(7)**2*x(8)*x(4)
+ 40*x(7)**3*x(8)*x(4) - 30*x(5)*x(9)*x(4) -
40*x(5)**3*x(9)*x(4) + 30*x(6)*x(9)*x(4) +
40*x(6)**3*x(9)*x(4) + 30*x(7)*x(9)*x(4) +
120*x(6)**2*x(7)*x(9)*x(4) + 120*x(6)*x(7)**2*x(9)*x(4)
+ 40*x(7)**3*x(9)*x(4) + 65*x(10)*x(4) +
65*x(11)*x(4) - 120*x(5)*x(11)*x(4) - 40*x(5)**3*x(11)*x(4)
+ 65*x(12)*x(4) - 120*x(6)*x(12)*x(4) -
40*x(6)**3*x(12)*x(4) - 120*x(7)*x(12)*x(4) -
120*x(6)**2*x(7)*x(12)*x(4) - 120*x(6)*x(7)**2*x(12)*x(4) -
40*x(7)**3*x(12)*x(4) = 0 , (8.2.4 - xi)

$h^5 f^4 f_{yyyy}$: 32 - 80*x(5)**4*x(1) - 80*x(5)**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) - 85*x(4) = 0 (8.2.4 - xii)

where

$$x(1) = w_1 \ , \ x(2) = w_2 \ , \ x(3) = w_3 \ , \ x(4) = w_4 \ , \ x(5) = a_1 \ , \ x(6) = a_2$$

$$x(7) = a_3 \ , \ x(8) = a_4 \ , \ x(9) = a_5 \ , \ x(10) = a_6 \ , \ x(11) = a_7 \ \textit{and} \ x(12) = a_8 \ .$$

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.,

$$w_1 = x(1) = 0.2615038147, \ w_2 = x(2) = -\ 0.2765809214, \ w_3 = x(3) = 0.5947141647$$

$$w_4 = x(4) = 0.4203629420, \ a_1 = x(5) = 1.5471214403, \ a_2 = x(6) = 0.1756458393$$

$$a_3 = x(7) = 0.1243059001, \ a_4 = x(8) = 0.1009316694, \ a_5 = x(9) = 0.1100539630$$

$$a_6 = x(10) = 0.99974318 \ , \ x_7 = x(11) = -\ 0.0928890403, \ x_8 = x(12) = -\ 0.6201812828 \qquad \ldots \qquad (8.2.5)$$

Thus, this new fifth order WRK method can be written as follows

$$y_{n+1} = y_n + h\left[0.2615038147\left(\frac{k_1 + k_2}{2}\right) - 0.2765809214\left(\frac{k_2 + k_3}{2}\right)\right.$$

$$\left. + \ 0.5947141647\left(\frac{k_3 + k_4}{2}\right) + 0.4203629420\left(\frac{k_4 + k_5}{2}\right)\right] \qquad (8.2.6)$$

where $k_1 = f(y_n)$

$$k_2 = f(y_n + 1.5471214403hk_1)$$

$$k_3 = f(y_n + 0.1756458393hk_1 + 0.1243059001hk_2)$$

$$k_4 = f(y_n + 0.1009316694hk_1 + 0.1100539630hk_2 + 0.2890143692hk_3)$$

$$k_5 = f(y_n + 0.9997431862hk_1 - 0.0928890403hk_2 - 0.6201812828hk_3$$

$$+ \ 0.7133271396hk_4) \ . \qquad \ldots \qquad (8.2.7)$$

By use of Mathematica to rationalize the coefficients in equation (8.2.5), we obtain

$$w_1 = 0.2615038147 = \frac{28449}{108790} \quad , \quad w_2 = -\ 0.2765809214 = \frac{40890}{147841}$$

$$w_3 = 0.5947141647 = \frac{211318}{355327} \quad , \quad w_4 = 0.4203629420 = \frac{65601}{156058}$$

$$a_1 = 1.5471214403 = \frac{479767}{310103} \quad , \quad a_2 = 0.1756458393 = \frac{14081}{80167}$$

$$a_3 = 0.1243059001 = \frac{7768}{62491} \quad , \quad a_4 = 0.1009316694 = \frac{20551}{203613}$$

$$a_5 = 0.1100539630 = \frac{15153}{137687} \quad , \quad a_6 = 0.9997431862 = \frac{151822}{151861}$$

$$a_7 = -0.0928890403 = \frac{4909}{52848} \quad , \quad a_8 = -0.6201812828 = \frac{91411}{147394}$$

$$a_{11} = \frac{1}{2} - a_4 - a_5 = 0.2890143692 = \frac{32604}{112811}$$

$$a_{22} = 1 - a_6 - a_7 - a_8 = 0.7133271396 = \frac{548973}{769595} .$$

Hence this new fifth order WRK method can be written in rational form as

$$y_{n+1} = y_n + h\left[\frac{28449}{108790}\left(\frac{k_1 + k_2}{2}\right) - \frac{40890}{147841}\left(\frac{k_2 + k_3}{2}\right) + \frac{211318}{355327}\left(\frac{k_3 + k_4}{2}\right) + \frac{65601}{156058}\left(\frac{k_4 + k_5}{2}\right)\right] \tag{8.2.8}$$

where $k_1 = f(y_n)$

$$k_2 = f\left(y_n + \frac{479767}{310103}hk_1\right)$$

$$k_3 = f\left(y_n + \frac{14081}{80167}hk_1 + \frac{7768}{62491}hk_2\right)$$

$$k_4 = f\left(y_n + \frac{20551}{203613}hk_1 + \frac{15153}{137687}hk_2 + \frac{32604}{112811}hk_3\right)$$

$$k_5 = f\left(y_n + \frac{151822}{151861}hk_1 - \frac{4909}{52848}hk_2 - \frac{91411}{147394}hk_3 + \frac{548973}{769595}hk_4\right) .$$

... (8.2.9)

8.2.1 Error Analysis

By substituting the values of $a_i, 1 \le i \le 8$ and $w_i, 1 \le i \le 4$ 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

$$LTE = h^6\left[\frac{1}{720}ff_y^5 + 0.0018022816f^2f_y^3f_{yy} - 0.0166861138f^3f_yf_{yy}^2 + 0.0082646021f^3f_y^2f_{yyy} + 0.0041171137f^4f_{yy}f_{yyy} - 0.0023096163f^4f_yf_{yyyy} + 0.0000588245f^5f_{yyyyy}\right] + O(h^7) \text{ as } h \to 0 . \quad \ldots \quad (8.2.10)$$

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

$$\begin{aligned} k_1 &= \lambda y_n \\ k_2 &= \lambda(y_n + 1.5471214403hk_1) \\ k_3 &= \lambda(y_n + 0.1756458393hk_1 + 0.1243059001hk_2) \\ k_4 &= \lambda(y_n + 0.1009316694hk_1 + 0.1100539630hk_2 + 0.2890143692hk_3) \\ k_5 &= \lambda(y_n + 0.9997431862hk_1 - 0.0928890403hk_2 - 0.6201812828hk_3 \\ &\quad + 0.7133271396hk_4) . \quad \ldots \quad (8.2.11) \end{aligned}$$

By substituting $k_i, 1 \le i \le 5$ in equation (8.2.11) and $w_i, 1 \le i \le 4$ in equation (8.2.5) into the fifth order WRK formula, i.e.,

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{k_i + k_{i+1}}{2}\right)\right] \quad (8.2.12)$$

to obtain

$$y_{n+1} = y_n + (h\lambda)y_n + 0.5(h\lambda)^2 y_n + 0.166667(h\lambda)^3 y_n + 0.0416667(h\lambda)^4 y_n + 0.00833333(h\lambda)^5 y_n + 0(h^6) . \quad \ldots \quad (8.2.13)$$

By rationalizing the coefficients in equation (8.2.13) we have

$$y_{n+1} = y_n + (h\lambda)y_n + \frac{1}{2}(h\lambda)^2 y_n + \frac{1}{6}(h\lambda)^3 y_n + \frac{1}{24}(h\lambda)^4 y_n + \frac{1}{120}(h\lambda)^5 + 0(h^6)$$

$$\ldots \qquad (8.2.14)$$

By substituting $h\lambda = z$ in (8.2.14), we can show that

$$y_{n+1} = y_n + y_n\left[z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120}\right] + 0(z^6) \; . \qquad (8.2.15)$$

Following equation (3.3.30), we write $\frac{y_{n+1}}{y_n} = Q$ in the equation (8.2.15), to obtain

$$Q = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 + \frac{1}{120}z^5 + 0(z^6)\,. \qquad (8.2.16)$$

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.,

$$\left|1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120}\right| < 1 \qquad (8.2.17)$$

by the use of Mathematica, we can plot the graphic surface defined by equation (8.2.17) i.e.,

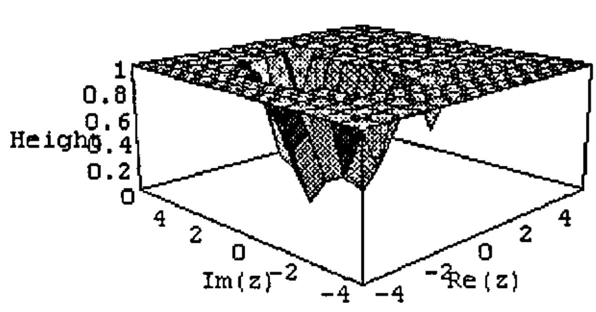


Figure 8.1: Graphic surface defined by the fifth order WRK formula

and the stability region defined by the formula in equation (8.2.17) as shown in Figure 8.2.

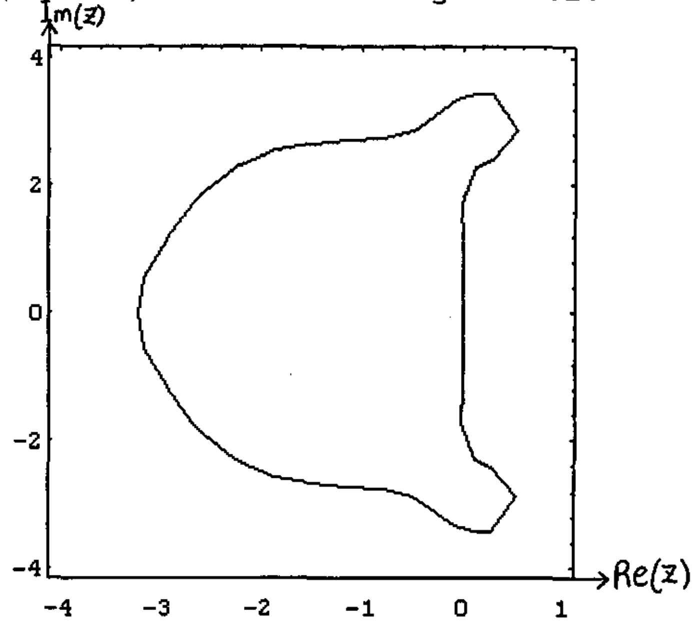


Figure 8.2: Stability region for fifth order WRK method

8.2.3 Numerical Example

We consider the IVP

$$y' = x - y + 1 \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 1 \tag{8.2.18}$$

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 8.3 : Absolute errors by various formula for solving equation (8.2.18)

x	Classical AM-RK4	RK4(5) Merson	RK5(6) Nystrom	AM-RK5(5)
0.1	0.8196E-07	0.1252E-07	0.1707E-08	0.1380E-08
0.2	0.1483E-06	0.2266E-07	0.3089E-08	0.2498E-08
0.3	0.2013E-06	0.3075E-07	0.4192E-08	0.3391E-08
0.4	0.2429E-06	0.3710E-07	0.5058E-08	0.4091E-08
0.5	0.2747E-06	0.4196E-07	0.5721E-08	0.4627E-08
0.6	0.2983E-06	0.4556E-07	0.6211E-08	0.5024E-08
0.7	0.3149E-06	0.4810E-07	0.6557E-08	0.5303E-08
0.8	0.3256E-06	0.4974E-07	0.6781E-08	0.5484E-08
0.9	0.3315E-06	0.5063E-07	0.6902E-08	0.5583E-08
1.0	0.3332E-06	0.5090E-07	0.6939E-08	0.5613E-08

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 :

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \qquad (8.3.1)$$

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

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{3} w_i \left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \qquad (8.3.2)$$

where $\sum_{i=1}^{3} w_i = 1$

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + a_1 h k_1)$$

$$k_3 = f\left(y_n + a_2 h k_1 + \left(\tfrac{1}{2} - a_2\right) h k_2\right)$$

$$k_4 = f\left(y_n + a_3 h k_1 + a_4 h k_2 + \left(1 - a_3 - a_4\right) h k_3\right) \qquad (8.3.3)$$

By comparing the right hand side of equation (8.3.1) with the Taylor series expansion for $y(x_{n+1})$, we obtain the following 7 equations involving 7 variables which must be satisfied to obtain fifth order accuracy, i.e.,

hf: 1 - x(1) - x(2) - x(3) = 0 (8.3.4-i)

$h^2 ff_y$: 2 - 2*x(4)*x(1) - x(2) - 2*x(4)*x(2) - 3*x(3) = 0 (8.3.4-ii)

$h^3 ff_y^2$: 8 - 12*x(4)**2*x(1) - 3*x(2) - 12*x(4)**2*x(2) +
24*x(4)*x(5)*x(2) - 15*x(3) - 12*x(4)*x(3) +
24*x(4)*x(5)*x(3) + 12*x(6)*x(3) + 12*x(7)*x(3) -
24*x(4)*x(7)*x(3) = 0 (8.3.4-iii)

$h^3 f^2 f_{yy}$: 8 - 12*x(4)**2*x(1) - 3*x(2) - 12*x(4)**2*x(2) - 15*x(3) = 0 (8.3.4-iv)

$h^4 ff_y^3$: 8 + 24*x(4)**3*x(1) + 3*x(2) - 30*x(4)*x(2) + 36*x(4)**2*x(2) +
24*x(4)**3*x(2) + 48*x(4)*x(5)*x(2) -
96*x(4)**2*x(5)*x(2) - 15*x(3) - 24*x(4)*x(3) +
48*x(4)*x(5)*x(3) + 24*x(6)*x(3) + 48*x(4)*x(6)*x(3) -
96*x(4)*x(5)*x(6)*x(3) + 24*x(7)*x(3) -
96*x(4)*x(5)*x(7)*x(3) = 0 (8.3.4-v)

$h^4 f^2 f_y f_{yy}$: 16 - 24*x(4)**3*x(1) - 3*x(2) - 6*x(4)*x(2) -
24*x(4)**3*x(2) + 24*x(4)*x(5)*x(2) +
24*x(4)**2*x(5)*x(2) - 39*x(3) - 12*x(4)*x(3) -
12*x(4)**2*x(3) + 24*x(4)*x(5)*x(3) +
24*x(4)**2*x(5)*x(3) + 30*x(6)*x(3) + 30*x(7)*x(3) -
48*x(4)*x(7)*x(3) - 24*x(4)**2*x(7)*x(3) = 0 (8.3.4-vi)

$h^4 f^3 f_{yyy}$: 4 - 8*x(4)**3*x(1) - x(2) - 8*x(4)**3*x(2) - 9*x(3) = 0 (8.3.4-vii)

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.,

$w_1 = 0.3333333333$, $w_2 = 0.3333333333$, $w_3 = 0.3333333333$

$a_1 = 0.5000000000$, $a_2 = 0.1250000000$, $a_3 = 0.2500000000$

$a_4 = -0.7500000000$ ... (8.3.5)

By rationalizing the coefficients in equation (8.3.5) we have

$$w_1 = w_2 = w_3 = \frac{1}{3} \quad , \quad a_1 = \frac{1}{2} \quad , \quad a_2 = \frac{1}{8}$$

$$a_3 = \frac{1}{4} \quad , \quad a_4 = -\frac{3}{4}$$

Thus, this new WRK method gives the same result as the fourth order contraharmonic mean (C_oM) 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 $k_1 = f(y_n)$

$$k_2 = f\left(y_n + \tfrac{1}{2}hk_1\right)$$

$$k_3 = f\left(y_n + \tfrac{1}{8}hk_1 + \tfrac{3}{8}hk_2\right)$$

$$k_4 = f\left(y_n + \tfrac{1}{4}hk_1 - \tfrac{3}{4}hk_2 + \tfrac{3}{2}hk_3\right) .$$

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_oM method including the extra parameters involved in the last two k values i.e.,

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \tag{8.3.6}$$

where $\sum_{i=1}^{4} w_i = 1$

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + a_1 hk_1)$$

$$k_3 = f(y_n + a_2 hk_1 + a_3 hk_2)$$

$$k_4 = f\left(y_n + a_4 hk_1 + a_5 hk_2 + \left(\tfrac{1}{2} - a_4 - a_5\right)hk_3\right)$$

$$k_5 = f\left(y_n + a_6 hk_1 + a_7 hk_2 + a_8 hk_3 + \left(1 - a_6 - a_7 - a_8\right)hk_4\right)$$

... (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.,

hf : 1 - x(1) - x(2) - x(3) - x(4) = 0 (8.3.8-i)

$h^2 ff_y$: 2 - 2*x(5)*x(1) - 2*x(5)*x(2) - 2*x(6)*x(2) -
2*x(7)*x(2) - x(3) - 2*x(6)*x(3) - 2*x(7)*x(3) - 3*x(4) = 0 (8.3.8-ii)

$h^3 ff_y^2$: 8 - 12*x(5)**2*x(1) - 12*x(5)**2*x(2) +
24*x(5)*x(6)*x(2)-12*x(6)**2*x(2) - 24*x(6)*x(7)*x(2) -
12*x(7)**2*x(2) - 3*x(3) - 12*x(6)**2*x(3) -
24*x(5)*x(7)*x(3) - 24*x(6)*x(7)*x(3) - 12*x(7)**2*x(3) +
24*x(6)*x(8)*x(3)+24*x(7)*x(8)*x(3)-24*x(5)*x(9)*x(3)+
24*x(6)*x(9)*x(3) + 24*x(7)*x(9)*x(3) -
15*x(4) - 12*x(6)*x(4) - 12*x(7)*x(4) + 24*x(6)*x(8)*x(4) +
24*x(7)*x(8)*x(4) - 24*x(5)*x(9)*x(4) +
24*x(6)*x(9)*x(4) + 24*x(7)*x(9)*x(4) + 12*x(10)*x(4) +
12*x(11)*x(4) - 24*x(5)*x(11)*x(4) +
12*x(12)*x(4) - 24*x(6)*x(12)*x(4) -
24*x(7)*x(12)*x(4) = 0 (8.3.8-iii)

$h^3 f^2 f_{yy}$: 8-12*x(5)**2*x(1)-12*x(5)**2*x(2) - 12*x(6)**2*x(2) -
24*x(6)*x(7)*x(2) - 12*x(7)**2*x(2) -
3*x(3) - 12*x(6)**2*x(3) - 24*x(6)*x(7)*x(3) -
12*x(7)**2*x(3) - 15*x(4) = 0 (8.3.8-iv)

$h^4 ff_y^3$: 8 + 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(5)**2*x(7)*x(2) - 144*x(5)*x(6)*x(7)*x(2) +
72*x(6)**2*x(7)*x(2) - 120*x(5)*x(7)**2*x(2) +
72*x(6)*x(7)**2*x(2) + 24*x(7)**3*x(2) + 3*x(3) -
30*x(6)*x(3) + 36*x(6)**2*x(3) + 24*x(6)**3*x(3) -
30*x(7)*x(3) + 72*x(6)*x(7)*x(3) - 96*x(5)*x(6)
*x(7)*x(3) + 72*x(6)**2*x(7)*x(3) + 36*x(7)**2
*x(3) - 96*x(5)*x(7)**2*x(3) + 72*x(6)*x(7)**2
*x(3) + 24*x(7)**3*x(3) + 48*x(6)*x(8)*x(3) -
96*x(6)**2*x(8)*x(3) + 48*x(7)*x(8)*x(3) +
96*x(5)*x(7)*x(8)*x(3) - 192*x(6)*x(7)*x(8)*x(3) -
96*x(7)**2*x(8)*x(3) - 48*x(5)*x(9)*x(3) +
48*x(6)*x(9)*x(3) + 96*x(5)*x(6)*x(9)*x(3) -
96*x(6)**2*x(9)*x(3) + 48*x(7)*x(9)*x(3) +
192*x(5)*x(7)*x(9)*x(3) - 192*x(6)*x(7)*x(9)*x(3) -
96*x(7)**2*x(9)*x(3) - 15*x(4) - 24*x(6)*x(4) -
24*x(7)*x(4) - 48*x(5)*x(7)*x(4) + 48*x(6)*x(8)*x(4) +
48*x(7)*x(8)*x(4) + 96*x(5)*x(7)*x(8)*x(4) -
48*x(5)*x(9)*x(4) + 48*x(6)*x(9)*x(4) + 48*x(7)*x(9)
*x(4) + 96*x(5)*x(7)*x(9)*x(4) + 24*x(10)*x(4) + 48*x(6)

*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(8)*x(12)*x(4) - 96*x(7)*x(8)*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) = 0 (8.3.8-v)

$h^4 f^2 f_y f_{yy}$: 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) -
24*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) = 0 (8.3.8-vi)

$h^4 f^3 f_{yyy}$: 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)

$h^5 f f_y^4$: 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) -

1440*x(5)**2*x(7)**2*x(2) + 2880*x(5)*x(6)*x(7)**2*x(2)
- 1440*x(6)**2*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)*x(3) - 600*x(5)*x(7)*x(3) -
720*x(6)*x(7)*x(3) + 1440*x(5)*x(6)*x(7)*x(3) -
720*x(6)**2*x(7)*x(3) + 1440*x(5)*x(6)**2*x(7)*x(3) -
960*x(6)**3*x(7)*x(3) - 360*x(7)**2*x(3) +
1440*x(5)*x(7)**2*x(3) - 960*x(5)**2*x(7)**2*x(3) -
720*x(6)*x(7)**2*x(3) + 2880*x(5)*x(6)*x(7)**2*x(3) -
1440*x(6)**2*x(7)**2*x(3) - 240*x(7)**3*x(3) +
1440*x(5)*x(7)**3*x(3) - 960*x(6)*x(7)**3*x(3) -
240*x(7)**4*x(3) - 360*x(6)*x(8)*x(3) +
1440*x(6)**2*x(8)*x(3) + 480*x(6)**3*x(8)*x(3) -
360*x(7)*x(8)*x(3) + 960*x(5)*x(7)*x(8)*x(3) +
2880*x(6)*x(7)*x(8)*x(3) - 3840*x(5)*x(6)*x(7)
*x(8)*x(3) + 1440*x(6)**2*x(7)*x(8)*x(3) +
1440*x(7)**2*x(8)*x(3) - 3840*x(5)*x(7)**2*x(8)*x(3)
+ 1440*x(6)*x(7)**2*x(8)*x(3) +
480*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) + 360*x(5)*x(9)*x(3) -
360*x(6)*x(9)*x(3) - 1440*x(5)*x(6)*x(9)*x(3) +
1440*x(6)**2*x(9)*x(3) - 480*x(5)*x(6)**2*x(9)*x(3) +
480*x(6)**3*x(9)*x(3) - 360*x(7)*x(9)*x(3) -
480*x(5)*x(7)*x(9)*x(3) + 1920*x(5)**2*x(7)*x(9)*x(3) +
2880*x(6)*x(7)*x(9)*x(3) -
4800*x(5)*x(6)*x(7)*x(9)*x(3) + 1440*x(6)**2*x(7)*x(9)
*x(3) + 1440*x(7)**2*x(9)*x(3) -
4320*x(5)*x(7)**2*x(9)*x(3) + 1440*x(6)*x(7)**2*x(9)
*x(3) + 480*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)
+ 45*x(4) - 300*x(6)*x(4) - 240*x(6)**2*x(4) -
300*x(7)*x(4) - 480*x(5)*x(7)*x(4) - 480*x(6)*x(7)*x(4)
- 240*x(7)**2*x(4) + 600*x(6)*x(8)*x(4) +
960*x(6)**2*x(8)*x(4) + 600*x(7)*x(8)*x(4) +
960*x(5)*x(7)*x(8)*x(4) + 1920*x(6)*x(7)*x(8)*x(4) +
960*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) - 600*x(5)*x(9)*x(4) +
600*x(6)*x(9)*x(4) - 960*x(5)*x(6)*x(9)*x(4) +
960*x(6)**2*x(9)*x(4) + 600*x(7)*x(9)*x(4) +
1920*x(6)*x(7)*x(9)*x(4) + 960*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*x(4) +
1920*x(5)*x(6)*x(9)**2*x(4) - 960*x(6)**2*x(9)**2*x(4) +
1920*x(5)*x(7)*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(10)*x(4) + 960*x(5)*x(7)*x(10)*x(4) -
1920*x(5)*x(7)*x(8)*x(10)*x(4) - 1920*x(5)*x(7)*x(9)
*x(10)*x(4) - 240*x(10)**2*x(4) + 60*x(11)*x(4) -
120*x(5)*x(11)*x(4) + 960*x(5)*x(6)*x(11)*x(4) + 1920*
x(5)*x(7)*x(11)*x(4) - 1920*x(5)*x(6)*x(8)*x(11)*x(4) -
3840*x(5)*x(7)*x(8)*x(11)*x(4) + 1920*x(5)**2*x(9)
*x(11)*x(4) - 1920*x(5)*x(6)*x(9)*x(11)*x(4) -
3840*x(5)*x(7)*x(9)*x(11)*x(4) - 480*x(10)*x(11)*x(4) +
960*x(5)*x(10)*x(11)*x(4) - 240*x(11)**2*x(4) +
960*x(5)*x(11)**2*x(4) - 960*x(5)**2*x(11)**2*x(4) +
60*x(12)*x(4) - 120*x(6)*x(12)*x(4) +
960*x(6)**2*x(12)*x(4) - 120*x(7)*x(12)*x(4) +
1920*x(6)*x(7)*x(12)*x(4) + 960*x(7)**2*x(12)*x(4) -
1920*x(6)**2*x(8)*x(12)*x(4) - 1920*x(5)*x(7)*x(8)
*x(12)*x(4) - 3840*x(6)*x(7)*x(8)*x(12)*x(4) -
1920*x(7)**2*x(8)*x(12)*x(4) + 1920*x(5)*x(6)*x(9)
*x(12)*x(4) - 1920*x(6)**2*x(9)*x(12)*x(4) -
3840*x(6)*x(7)*x(9)*x(12)*x(4) - 1920*x(7)**2*x(9)
*x(12)*x(4) - 480*x(10)*x(12)*x(4) +
960*x(6)*x(10)*x(12)*x(4) + 960*x(7)*x(10)*x(12)*x(4)
- 480*x(11)*x(12)*x(4) + 960*x(5)*x(11)*x(12)*x(4) +
960*x(6)*x(11)*x(12)*x(4) - 1920*x(5)*x(6)*x(11)
*x(12)*x(4) + 960*x(7)*x(11)*x(12)*x(4) -
1920*x(5)*x(7)*x(11)*x(12)*x(4) - 240*x(12)**2*x(4) +
960*x(6)*x(12)**2*x(4) - 960*x(6)**2*x(12)**2*x(4) +
960*x(7)*x(12)**2*x(4) - 1920*x(6)*x(7)*x(12)**2*x(4)
- 960*x(7)**2*x(12)**2*x(4) = 0 (8.3.8-viii)

$h^5 f^2 f_y^2 f_{yy}$: 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) + 1440*x(5)
3*x(7)*x(2) - 4320*x(5)*x(6)2*x(7)*x(2) +
2880*x(6)**3*x(7)*x(2) - 480*x(5)**2*x(7)**2*x(2) -
7200*x(5)*x(6)*x(7)**2*x(2) +
4320*x(6)**2*x(7)**2*x(2) - 3360*x(5)*x(7)**3*x(2) +
2880*x(6)*x(7)**3*x(2) + 720*x(7)**4*x(2) +
45*x(3) - 420*x(6)*x(3) - 120*x(6)**2*x(3) +
720*x(6)**3*x(3) + 720*x(6)**4*x(3) - 420*x(7)*x(3) -
240*x(5)*x(7)*x(3) - 240*x(6)*x(7)*x(3) - 960*x(5)**2
*x(6)*x(7)*x(3) + 2160*x(6)**2*x(7)*x(3) -
2880*x(5)*x(6)**2*x(7)*x(3) + 2880*x(6)**3*x(7)*x(3) -
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) +

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

*x(4) + 1920*x(5)*x(6)*x(9)**2*x(4) -
960*x(6)**2*x(9)**2*x(4) + 1920*x(5)*x(7)*x(9)**2
*x(4) - 1920*x(6)*x(7)*x(9)**2*x(4) -
960*x(7)**2*x(9)**2*x(4) + 1440*x(10)*x(4) +
1440*x(6)*x(10)*x(4) + 480*x(6)**2*x(10)*x(4) +
1440*x(7)*x(10)*x(4) + 960*x(6)*x(7)*x(10)*x(4) +
480*x(7)**2*x(10)*x(4) - 2880*x(6)*x(8)*x(10)*x(4) -
960*x(6)**2*x(8)*x(10)*x(4) - 2880*x(7)*x(8)*x(10)
*x(4) - 1920*x(6)*x(7)*x(8)*x(10)*x(4) -
960*x(7)**2*x(8)*x(10)*x(4) + 2880*x(5)*x(9)*x(10)
*x(4) + 960*x(5)**2*x(9)*x(10)*x(4) -
2880*x(6)*x(9)*x(10)*x(4) - 960*x(6)**2*x(9)*x(10)
*x(4) - 2880*x(7)*x(9)*x(10)*x(4) -
1920*x(6)*x(7)*x(9)*x(10)*x(4) - 960*x(7)**2*x(9)*
x(10)*x(4) - 240*x(10)**2*x(4) + 1440*x(11)*x(4) -
2640*x(5)*x(11)*x(4) - 480*x(5)**2*x(11)*x(4) +
1440*x(6)*x(11)*x(4) + 480*x(6)**2*x(11)*x(4) +
1440*x(7)*x(11)*x(4) + 960*x(6)*x(7)*x(11)*x(4) +
480*x(7)**2*x(11)*x(4) - 2880*x(6)*x(8)*x(11)*x(4) -
960*x(6)**2*x(8)*x(11)*x(4) - 2880*x(7)*x(8)*x(11)
*x(4) - 1920*x(6)*x(7)*x(8)*x(11)*x(4) -
960*x(7)**2*x(8)*x(11)*x(4) + 2880*x(5)*x(9)*x(11)
*x(4) + 960*x(5)**2*x(9)*x(11)*x(4) -
2880*x(6)*x(9)*x(11)*x(4) - 960*x(6)**2*x(9)*x(11)
*x(4) - 2880*x(7)*x(9)*x(11)*x(4) -
1920*x(6)*x(7)*x(9)*x(11)*x(4) - 960*x(7)**2
*x(9)*x(11)*x(4) - 480*x(10)*x(11)*x(4) +
960*x(5)*x(10)*x(11)*x(4) - 240*x(11)**2*x(4) +
960*x(5)*x(11)**2*x(4) - 960*x(5)**2*x(11)**2*
x(4)+1440*x(12)*x(4) - 1200*x(6)*x(12)*x(4) -
1200*x(7)*x(12)*x(4) - 1920*x(5)*x(7)*x(12)*x(4) -
960*x(5)**2*x(7)*x(12)*x(4) - 1920*x(5)*x(6)
*x(7)*x(12)*x(4) - 1920*x(5)*x(7)**2*x(12)*x(4) -
2880*x(6)*x(8)*x(12)*x(4) - 960*x(6)**2*x(8)
*x(12)*x(4) - 2880*x(7)*x(8)*x(12)*x(4) -
1920*x(6)*x(7)*x(8)*x(12)*x(4)-960*x(7)**2
*x(8)*x(12)*x(4) + 2880*x(5)*x(9)*x(12)*x(4)+
960*x(5)**2*x(9)*x(12)*x(4) - 2880*x(6)
*x(9)*x(12)*x(4) - 960*x(6)**2*x(9)*x(12)*x(4)-
2880*x(7)*x(9)*x(12)*x(4) - 1920*x(6)*x(7)
*x(9)*x(12)*x(4) - 960*x(7)**2*x(9)*x(12)*x(4)-
480*x(10)*x(12)*x(4) + 960*x(6)*x(10)*x(12)*x(4)
+960*x(7)*x(10)*x(12)*x(4) - 480*x(11)*x(12)
*x(4)+960*x(5)*x(11)*x(12)*x(4) + 960*x(6)*x(11)
*x(12)*x(4) - 1920*x(5)*x(6)*x(11)*x(12)*x(4)+
960*x(7)*x(11)*x(12)*x(4) - 1920*x(5)*x(7)*x(11)
*x(12)*x(4) - 240*x(12)**2*x(4) + 960*x(6)*x(12)
2*x(4)-960*x(6)2*x(12)**2*x(4) + 960*x(7)
*x(12)**2*x(4) - 1920*x(6)*x(7)*x(12)**2
*x(4)-960*x(7)**2*x(12)**2*x(4) = 0 (8.3.8 -ix)

$h^5 f^3 f_{yy}^2$: 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)*x(2) - 480*x(5)**2*x(7)**2*x(2) -
1440*x(6)**2*x(7)**2*x(2) - 960*x(6)*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)*x(7)*x(3) -
960*x(5)**2*x(6)*x(7)*x(3) - 960*x(6)**3*x(7)*x(3) -
120*x(7)**2*x(3) - 960*x(5)**2*x(7)**2*x(3) -
1440*x(6)**2*x(7)**2*x(3) - 960*x(6)*x(7)**3*x(3) -
240*x(7)**4*x(3) + 480*x(6)**2*x(8)*x(3) +
960*x(6)*x(7)*x(8)*x(3) + 480*x(7)**2*x(8)*x(3) -
480*x(5)**2*x(9)*x(3) + 480*x(6)**2*x(9)*x(3) +
960*x(6)*x(7)*x(9)*x(3) + 480*x(7)**2*x(9)*x(3) -
375*x(4) - 240*x(6)**2*x(4) - 480*x(6)*x(7)*x(4) -
240*x(7)**2*x(4) + 480*x(6)**2*x(8)*x(4) +
960*x(6)*x(7)*x(8)*x(4) + 480*x(7)**2*x(8)*x(4) -
480*x(5)**2*x(9)*x(4) + 480*x(6)**2*x(9)*x(4) +
960*x(6)*x(7)*x(9)*x(4) + 480*x(7)**2*x(9)*x(4) +
240*x(10)*x(4) + 240*x(11)*x(4) - 960*x(5)**2*x(11)
*x(4) + 240*x(12)*x(4) - 960*x(6)**2*x(12)*x(4) -
1920*x(6)*x(7)*x(12)*x(4) - 960*x(7)**2*x(12)*x(4) = 0 (8.3.8 -x)

$h^5 f^3 f_y f_{yyy}$: 56 - 80*x(5)**4*x(1) - 80*x(5)**4*x(2) + 80*x(5)
3*x(6)*x(2) + 80*x(5)*x(6)3*x(2) - 80*x(6)**4*x(2)
-320*x(6)**3*x(7)*x(2) - 240*x(5)*x(6)*x(7)**2*x(2) -
480*x(6)**2*x(7)**2*x(2) - 160*x(5)*x(7)**3*x(2) -
320*x(6)*x(7)**3*x(2) - 80*x(7)**4*x(2) - 5*x(3) -
20*x(6)*x(3) - 80*x(6)**4*x(3) - 20*x(7)*x(3) -
80*x(5)**3*x(7)*x(3) - 240*x(5)*x(6)**2*x(7)*x(3) -
320*x(6)**3*x(7)*x(3) - 480*x(5)*x(6)*x(7)**2*x(3) -
480*x(6)**2*x(7)**2*x(3) - 240*x(5)*x(7)**3*x(3) -
320*x(6)*x(7)**3*x(3) - 80*x(7)**4*x(3) +
60*x(6)*x(8)*x(3) + 80*x(6)**3*x(8)*x(3) +
60*x(7)*x(8)*x(3) + 240*x(6)**2*x(7)*x(8)*x(3) +
240*x(6)*x(7)**2*x(8)*x(3) + 80*x(7)**3*x(8)*x(3) -
60*x(5)*x(9)*x(3) - 80*x(5)**3*x(9)*x(3) +
60*x(6)*x(9)*x(3) + 80*x(6)**3*x(9)*x(3) +
60*x(7)*x(9)*x(3) + 240*x(6)**2*x(7)*x(9)*x(3) +
240*x(6)*x(7)**2*x(9)*x(3) + 80*x(7)**3*x(9)*x(3) -
165*x(4) - 30*x(6)*x(4) - 40*x(6)**3*x(4) -
30*x(7)*x(4) - 120*x(6)**2*x(7)*x(4) - 120*x(6)
*x(7)**2*x(4) - 40*x(7)**3*x(4) + 60*x(6)*x(8)*x(4) +
80*x(6)**3*x(8)*x(4) + 60*x(7)*x(8)*x(4) + 240*x(6)
2*x(7)*x(8)*x(4) + 240*x(6)*x(7)2*x(8)*x(4) +
80*x(7)**3*x(8)*x(4) - 60*x(5)*x(9)*x(4) -
80*x(5)**3*x(9)*x(4) + 60*x(6)*x(9)*x(4) +
80*x(6)**3*x(9)*x(4) + 60*x(7)*x(9)*x(4) + 240*x(6)
2*x(7)*x(9)*x(4) + 240*x(6)*x(7)2*x(9)*x(4) +
80*x(7)**3*x(9)*x(4) + 130*x(10)*x(4) + 130*x(11)

*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(7)*x(12)*x(4) -
240*x(6)**2*x(7)*x(12)*x(4) - 240*x(6)*x(7)**2
*x(12)*x(4) - 80*x(7)**3*x(12)*x(4) = 0 (8.3.8-xi)

$h^5 f^4 f_{yyyy}$: 32 - 80*x(5)**4*x(1) - 80*x(5)**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) - 85*x(4) = 0 (8.3.8-xii)

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.,

$$\begin{array}{lll} w_1 = -0.1773157366, & w_2 = 1.0254553152, & w_3 = -0.0779114700 \\ w_4 = 0.2297718914, & a_1 = 0.1017275411, & a_2 = -0.5236574475 \\ a_3 = 1.1653361910, & a_4 = 4.7450804540, & a_5 = -4.2354437705 \\ a_6 = -0.5736403905, & a_7 = 0.9301175162, & a_8 = 0.4667978567 \end{array} \quad \ldots \quad (8.3.9)$$

Thus, the new fifth order C_oM WRK method can be written as follows :-

$$y_{n+1} = y_n + h\left[-0.1773157366\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) + 1.0254553152\left(\frac{k_2^2 + k_3^2}{k_2 + k_3}\right)\right.$$
$$\left. - 0.0779114700\left(\frac{k_3^2 + k_4^2}{k_3 + k_4}\right) + 0.2297718914\left(\frac{k_4^2 + k_5^2}{k_4 + k_5}\right)\right] \quad (8.3.10)$$

where $k_1 = f(y_n)$

$$k_2 = f(y_n + 0.1017275411hk_1)$$

$$k_3 = f(y_n - 0.5236574475hk_1 + 1.1653361910hk_2)$$

$$k_4 = f(y_n + 4.7450804540hk_1 - 4.2354437705hk_2 - 0.0096366835hk_3)$$

$$k_5 = f(y_n - 0.5736403905hk_1 + 0.9301175162hk_2 + 0.4667978567hk_3 + 0.1767250176hk_4) \,. \quad \ldots \quad (8.3.11)$$

By the use of Mathematica to rationalize the coefficients in equation (8.3.9), we obtain

$$w_1 = -\frac{8639}{48721} \quad , \quad w_2 = \frac{135638}{132271} \quad , \quad w_3 = -\frac{2480}{31831} \quad , \quad w_4 = \frac{22241}{96796}$$

$$a_1 = \frac{6236}{61301} \quad , \quad a_2 = -\frac{162659}{310621} \quad , \quad a_3 = \frac{225101}{193164} \quad , \quad a_4 = \frac{814792}{171713}$$

$$a_5 = -\frac{424527}{100232} \quad , \quad a_6 = -\frac{27973}{48764} \quad , \quad a_7 = \frac{78594}{84499} \quad , \quad a_8 = \frac{54796}{117387}$$

$$\frac{1}{2} - a_4 - a_5 = -\frac{1353}{140401} \quad , \quad 1 - a_6 - a_7 - a_8 = \frac{26844}{151897}.$$

and the new fifth order contraharmonic WRK method can be written in rational form as

$$y_{n+1} = y_n + h\left[-\frac{8639}{48721}\left(\frac{k_1^2 + k_2^2}{k_1 + k_2}\right) + \frac{135638}{132271}\left(\frac{k_2^2 + k_3^2}{k_2 + k_3}\right) - \frac{2480}{31831}\left(\frac{k_3^2 + k_4^2}{k_3 + k_4}\right) + \frac{22241}{96796}\left(\frac{k_4^2 + k_5^2}{k_4 + k_5}\right)\right] \quad (8.3.12)$$

where $k_1 = f(y_n)$

$$k_2 = f\left(y_n + \frac{6236}{61301}hk_1\right)$$

$$k_3 = f\left(y_n - \frac{162659}{310621}hk_1 + \frac{225101}{193164}hk_2\right)$$

$$k_4 = f\left(y_n + \frac{814792}{171713}hk_1 - \frac{424527}{100232}hk_2 - \frac{1353}{140401}hk_3\right)$$

$$k_5 = f\left(y_n - \frac{27973}{48764}hk_1 + \frac{78594}{84499}hk_2 + \frac{54796}{117387}hk_3 + \frac{26844}{151897}hk_4\right)$$

8.3.3 Error Analysis Fifth Order C_oM Method

By substituting the values of $a_i, 1 \le i \le 8$ and $w_i, 1 \le i \le 4$ in (8.3.9) into (8.3.6) and (8.3.7) using Mathematica and evaluating all the terms up to (h^6) to represent the local truncation error (LTE) for this method we have

$$LTE = h^6\Big[0.0132485733 ff_y^5 + 0.0202501069 f^2 f_y^3 f_{yy}$$
$$+ 0.0095106268 f^3 f_y f_{yy}^2 - 0.0022879188 f^3 f_y^2 f_{yyy}$$
$$- 0.0001379536 f^4 f_{yy} f_{yyy} - 0.0003448339 f^4 f_y f_{yyyy}$$
$$- 0.0000178190 f^5 f_{yyyyy}\Big] \quad \ldots \quad (8.3.13)$$

8.3.4. Stability Analysis Fifth Order C_oM Method

We examine the stability region for the fifth order contraharmonic mean WRK method with the test equation $y' = \lambda y_n$ and we obtain

$$k_1 = \lambda y_n$$
$$k_2 = \lambda(y_n + 0.1017275411 hk_1)$$
$$k_3 = \lambda(y_n - 0.5236574475 hk_1 + 1.1653361910 hk_2)$$
$$k_4 = \lambda(y_n + 4.7450804540 hk_1 - 4.2354437705 hk_2 - 0.0096366835 hk_3)$$
$$k_5 = \lambda(y_n - 0.5736403905 hk_1 + 0.9301175162 hk_2 + 0.4667978567 hk_3$$
$$+ 0.1767250176 hk_4)\,. \quad \ldots \quad (8.3.14)$$

By substituting $k_i, 1 \le i \le 5$ in equation (8.3.14) and $w_i, 1 \le i \le 4$ in equation (8.3.9) into the fifth order C_oM WRK formula, i.e.,

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \quad (8.3.15)$$

we obtain

$$y_{n+1} = y_n + (h\lambda)y_n + 0.5(h\lambda)^2 y_n + 0.166667(h\lambda)^3 y_n + 0.0416667(h\lambda)^4 y_n$$
$$+ 0.008333333(h\lambda)^5 y_n - 0.0118597(h\lambda)^6 y_n + 0(h^7) \ldots \quad (8.3.16)$$

By rationalizing the coefficients in equation (8.3.16) we have

$$y_{n+1} = y_n + (h\lambda)y_n + \frac{1}{2}(h\lambda)^2 y_n + \frac{1}{6}(h\lambda)^3 y_n + \frac{1}{24}(h\lambda)^4 y_n$$
$$+\frac{1}{120}(h\lambda)^5 y_n - \frac{47}{3963}(h\lambda)^6 y_n + 0(h^7) \quad \ldots \quad (8.3.17)$$

By substituting $h\lambda = z$ in (8.3.17), we can show that

$$y_{n+1} = y_n + y_n\left[z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} - \frac{47}{3963}z^6\right] + 0\left(z^7\right). \qquad (8.3.18)$$

Following equation (3.3.30), we write $\frac{y_{n+1}}{y_n} = Q$ in the equation (8.3.18), to obtain

$$Q = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 + \frac{1}{120}z^5 - \frac{47}{3963}z^6 + 0\left(z^7\right). \qquad (8.3.19)$$

To determine the stability region of the fifth order C_oM WRK formula in the complex plane that satisfy the condition

$$\left|\frac{y_{n+1}}{y_n}\right| = |Q| < 1$$

i.e.,

$$\left|1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} - \frac{47}{3963}z^6\right| < 1 \qquad (8.3.20)$$

by the use of Mathematica, we can plot the graphic surface defined by equation (8.3.20) i.e.,

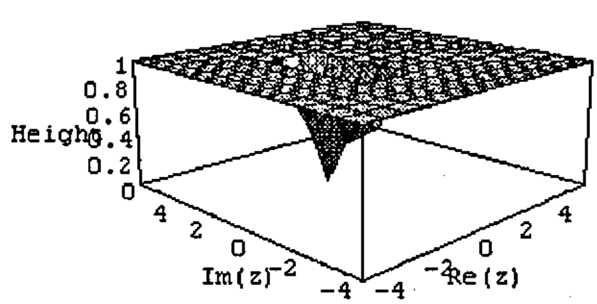


Figure 8.3: Graphic surface defined by the fifth order C_oM WRK formula

and the stability region defined by the formula in equation (8.3.20) as shown in Figure 8.4.

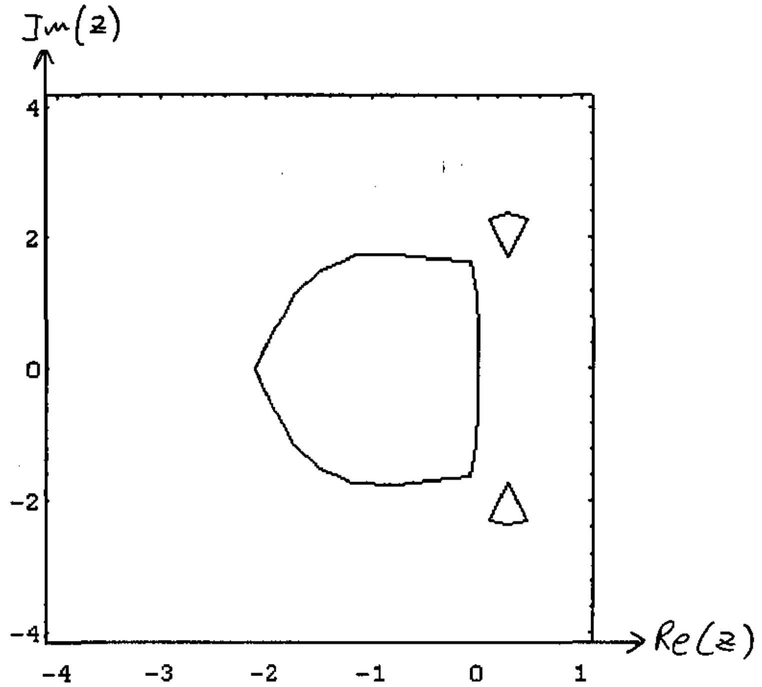


Figure 8.4: Stability region for fifth order C_oM WRK method

8.3.5. Numerical Example

We consider the standard test IVP

$$y' = y \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 1 \qquad (8.3.21)$$

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)

X	Classical AM-RK4	RK4(5) Merson	GM-RK5(5)	WRK-RK5(5)	CoM-RK5(5)
0.1	.847423E-07	.152979E-07	.110389E-07	.142041E-08	.801866E-08
0.2	.187310E-06	.338135E-07	.243998E-07	.313960E-08	.177240E-07
0.3	.310514E-06	.560546E-07	.404490E-07	.520468E-08	.293821E-07
0.4	.457561E-06	.825999E-07	.596040E-07	.766942E-08	.432963E-07
0.5	.632103E-06	.114109E-06	.823408E-07	.105950E-07	.598122E-07
0.6	.838299E-06	.151332E-06	.109201E-06	.140512E-07	.793233E-07
0.7	.108087E-05	.195122E-06	.140799E-06	.181171E-07	.102277E-06
0.8	.136520E-05	.246449E-06	.177837E-06	.228829E-07	.129181E-06
0.9	.169738E-05	.306414E-06	.221108E-06	.284507E-07	.160613E-06
1.0	.208432E-05	.376267E-06	.271514E-06	.349365E-07	.197227E-06

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_oM) 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_oM) 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

$$y_{AM} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i + k_{i+1}}{2}\right)\right] \tag{8.4.1}$$

where $\sum_{i=1}^{4} w_i = 1$, $w_1 = 0.2615038147$, $w_2 = -0.2765809214$

$w_3 = 0.5947141647$, $w_4 = 0.4203629420$

$$k_1 = f(y_n)$$
$$k_2 = f(y_n + 1.5471214403hk_1)$$
$$k_3 = f(y_n + 0.1756458393hk_1 + 0.1243059001hk_2)$$
$$k_4 = f(y_n + 0.1009316694hk_1 + 0.1100539630hk_2 + 0.2890143692hk_3)$$
$$k_5 = f(y_n + 0.9997431862hk_1 - 0.0928890403hk_2 - 0.6201812828hk_3 + 0.7133271396hk_4) \quad (8.4.2)$$

and the fifth order contraharmonic mean (C_oM) formula in the form

$$y_{CoM} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i^2 + k_{i+1}^2}{k_i + k_{i+1}}\right)\right] \quad (8.4.3)$$

where $\sum_{i=1}^{4} w_i = 1$, $w_1 = -0.1773157366$, $w_2 = 1.0254553152$

$$w_3 = -0.0779114700 \quad , \quad w_4 = 0.2297718914$$
$$k_1 = f(y_n)$$
$$k_2 = f(y_n + 0.1017275411hk_1)$$
$$k_3 = f(y_n - 0.5236574475hk_1 + 1.1653361910hk_2)$$
$$k_4 = f(y_n + 4.7450804540hk_1 - 4.2354437705hk_2 - 0.0096366835hk_3)$$
$$k_5 = f(y_n - 0.5736403905hk_1 + 0.9301175162hk_2 + 0.4667978567hk_3 + 0.1767250176hk_4) \quad (8.4.4)$$

is called RK(5,5) method. The difference between equation (8.4.1) and (8.4.3), i.e., $|y_{AM} - y_{CoM}|$ 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 $AM - C_oM$ 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

$$y_{n+1}^{AM} = y_n + LTE^{AM} \tag{8.4.5}$$

and for the contraharmonic mean method

$$y_{n+1}^{C_oM} = y_n + LTE^{C_oM} \tag{8.4.6}$$

where y_{n+1}^{AM} and $y_{n+1}^{C_oM}$ are the numerical approximations at x_{n+1} obtained by the arithmetic mean and contraharmonic mean methods respectively and LTE^{AM} and LTE^{C_oM} 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

$$y_{n+1}^{AM} - y_{n+1}^{C_oM} = LTE^{AM} - LTE^{C_oM} \tag{8.4.7}$$

The local trunscation error for the fifth order arithmetic mean Runge-Kutta method involves y derivatives given by

$$\begin{aligned} LTE^{AM} = h^6\Big[& \frac{1}{720} ff_y^5 + 0.0018022816 f^2 f_y^3 f_{yy} - 0.0166861138 f^3 f_y f_{yy}^2 \\ & + 0.0082646021 f^3 f_y^2 f_{yyy} + 0.0041171137 f^4 f_{yy} f_{yyy} \\ & - 0.0023096163 f^4 f_y f_{yyyy} + 0.0000588245 f^5 f_{yyyyy} \Big] \quad \ldots \end{aligned} \tag{8.4.8}$$

while the local truncation error for the contraharmonic mean method is given by

$$\begin{aligned} LTE^{C_oM} = h^6\Big[& 0.0132485733 ff_y^5 + 0.0202501069 f^2 f_y^3 f_{yy} \\ & + 0.0095106268 f^3 f_y f_{yy}^2 - 0.0022879188 f^3 f_y^2 f_{yyy} \\ & - 0.0001379536 f^4 f_{yy} f_{yyy} - 0.0003448339 f^4 f_y f_{yyyy} \\ & - 0.0000178190 f^5 f_{yyyyy} \Big] \quad \ldots \end{aligned} \tag{8.4.9}$$

The absolute difference between LTE^{AM} and LTE^{C_oM} is given by

$$\left|LTE^{AM} - LTE^{C_oM}\right| = h^6\left[\left(\frac{1}{720} - 0.0132485733\right)ff_y^5 + (0.0018022816 - 0.0202501069)f^2f_y^3f_{yy}\right.$$
$$+(-0.0166861138 - 0.0095106268)f^3f_yf_{yy}^2$$
$$+(0.0082646021 + 0.0022879188)f^3f_y^2f_{yyy}$$
$$+(0.0041171137 + 0.0001379536)f^4f_{yy}f_{yyy}$$
$$+(-0.0023096163 + 0.0003448339)f^4f_yf_{yyyy}$$
$$\left.+(0.0000588245 + 0.0000178190)f^5f_{yyyyy}\right]$$

$$= h^6\left[-0.0118597ff_y^5 - 0.0184478f^2f_y^3f_{yy}\right.$$
$$-0.0261967f^3f_yf_{yy}^2 + 0.0105525f^3f_y^2f_{yyy}$$
$$0.00425507f^4f_{yy}f_{yyy} - 0.00196478f^4f_yf_{yyyy}$$
$$\left.+0.0000766435f^5f_{yyyyy}\right] \quad \ldots \quad (8.4.10)$$

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,

$$|f(x,y)| < Q \quad , \quad \left|\frac{\partial^{i+j}f(x,y)}{\partial x^i \partial y^j}\right| < \frac{P^{i+j}}{Q^{j-1}} \quad , \quad i + j \le p \qquad (8.4.11)$$

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

$$\left.\begin{array}{r} \left|ff_y^5\right| < Q\left(\dfrac{P^{0+1}}{Q^{1-1}}\right)^5 \\ \left|f^2f_y^3f_{yy}\right| < Q^2\left(\dfrac{P^1}{Q^0}\right)^3\dfrac{P^2}{Q} \\ \left|f^3f_yf_{yy}^2\right| < Q^3P\left(\dfrac{P^2}{Q}\right)^2 \\ \left|f^3f_y^2f_{yyy}\right| < Q^3P^2\left(\dfrac{P^3}{Q^2}\right) \\ \left|f^4f_{yy}f_{yyy}\right| < Q^4\dfrac{P^2}{Q}.\dfrac{P^3}{Q^2} \\ \left|f^4f_yf_{yyyy}\right| < Q^4P.\dfrac{P^4}{Q^3} \\ \left|f^5f_{yyyyy}\right| < Q^5\dfrac{P^5}{Q^4} \end{array}\right\} \quad P^5Q \qquad \cdots \qquad (8.4.12)$$

From the equations (8.4.9)-(8.4.12), we obtain

$$\left|LTE^{AM} - LTE^{C_oM}\right| \leq 0.0435848P^5Qh^6 \qquad (8.4.13)$$

Hence,

$$\left|y_{n+1}^{AM} - y_{n+1}^{C_oM}\right| \leq 0.0435848P^5Qh^6 \qquad (8.4.14)$$

or

$$\left|y_{n+1}^{AM} - y_{n+1}^{C_oM}\right| \leq \frac{89}{2042}P^5Qh^6 \ .$$

By taking the tolerence as TOL, i.e., $\varepsilon < 0.00005$, then by setting

$$\left|y_{n+1}^{AM} - y_{n+1}^{C_oM}\right| \leq TOL$$

the error control and step size selection can be determined by (8.4.14) to give the formula as

$$0.0435848P^5Qh^6 < TOL$$

or

$$h < \left[\frac{TOL}{0.0435848P^5Q}\right]^{\frac{1}{6}} \qquad . \qquad (8.4.15)$$

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 : $y = \exp(-x)$

x	Exact Solution	Numerical Solution	Absolute Error	Estimate Error
h= .50000				
h= .25000				
h= .50000				
0.50000	.6065104D+00	.6065307D+00	.2024324D-04	.1283099D-04
1.00000	.3678549D+00	.3678794D+00	.2455588D-04	.1556049D-04
1.50000	.2231078D+00	.2231302D+00	.2234047D-04	.1415296D-04
2.00000	.1353172D+00	.1353353D+00	.1806660D-04	.1144245D-04
2.50000	.8207130D-01	.8208500D-01	.1369721D-04	.8672849D-05
3.00000	.4977710D-01	.4978707D-01	.9969165D-05	.6310677D-05
3.50000	.3019033D-01	.3019738D-01	.7054254D-05	.4464323D-05
4.00000	.1831075D-01	.1831564D-01	.4889771D-05	.3093716D-05
4.50000	.1110566D-01	.1110900D-01	.3336465D-05	.2110405D-05
5.00000	.6735699D-02	.6737947D-02	.2248483D-05	.1421858D-05
5.50000	.4085271D-02	.4086771D-02	.1500126D-05	.9483790D-06
6.00000	.2477760D-02	.2478752D-02	.9925716D-06	.6273406D-06
h= 1.00000				
7.00000	.9085119D-03	.9118820D-03	.3370110D-05	.4554334D-05
8.00000	.3331210D-03	.3354626D-03	.2341614D-05	.3081334D-05
9.00000	.1221444D-03	.1234098D-03	.1265432D-05	.1590493D-05
10.00000	.4478627D-04	.4539993D-04	.6136600D-06	.7335390D-06

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 : $x_0 = 1 \quad , \quad y_0 = 4$

Exact solution : $y = -x - Log(x) + x^2 Cos(x) - 2xSin(x) - 2Cos(x) + C$

Problem 4 (NPB 12): $y' + \ln(x^2) = 0$

Initial conditions : $x_0 = 1 \quad , \quad y_0 = 2$

Exact solution : $y = -2(x\ln(x) - x)$

Problem 5 (NPB 4) : Y' - Y = 0

Initial conditions : $x_0 = 0 \quad , \quad y_0 = 1$

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 ERREST used together with a constant derived in equation (8.4.14 - 8.4.15) are in the form

$$ERREST = \left|Y_{AM} - Y_{CoM}\right| * \frac{89}{2042} \tag{8.4.16}$$

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

Problem	Time Taken			
	RK(4,4)	RK4(5)-Merson	RKF(4,5)	RK(5,5)
1	1.80	0.98	1.11	0.93
2	0.20	0.09	0.10	0.10
3	0.26	0.24	0.02	0.13
4	0.22	0.20	0.04	0.12
5	0.12	0.10	0.09	0.08

Table 8.6

Problem	Absolute Error			
	RK(4,4)	RK4(5)-Merson	RKF(4,5)	RK(5,5)
1	0.2258×10^{-5}	0.2959×10^{-5}	0.1061×10^{-4}	0.6137×10^{-6}
2	0.8308×10^{-6}	0.1944×10^{-5}	0.2984×10^{-5}	0.6186×10^{-6}
3	0.3253×10^{-6}	0.3253×10^{-6}	0.1175×10^{-4}	0.2449×10^{-6}
4	0.2949×10^{-6}	0.6047×10^{-6}	0.2427×10^{-5}	0.5352×10^{-6}
5	0.5516×10^{-4}	0.1446×10^{-4}	0.7625×10^{-5}	0.3999×10^{-4}

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 focussed their attention on the second order initial value problem (IVP)

$$y'' = f(x,y), \quad y(x_0) = y_0 \quad , \quad y'(x_0) = \eta. \qquad (8.5.1)$$

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

$$y'' = f(x,y,y'), \quad y(x_0) = y_0 \quad , \quad y'(x_0) = \eta. \qquad (8.5.2)$$

In Evans and Yaakub [1996], a new fifth order five stage arithmetic mean (AM) weighted Runge-Kutta (WRK) method was presented in the form

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{k_i + k_{i+1}}{2}\right)\right] \tag{8.5.3}$$

where $\sum_{i=1}^{4} w_i = 1$

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + a_1 hk_1)$$

$$k_3 = f(y_n + a_2 hk_1 + a_3 hk_2)$$

$$k_4 = f\left(y_n + a_4 hk_1 + a_5 hk_2 + \left(\tfrac{1}{2} - a_4 - a_5\right)hk_3\right)$$

$$k_5 = f\left(y_n + a_6 hk_1 + a_7 hk_2 + a_8 hk_3 + \left(1 - a_6 - a_7 - a_8\right)hk_4\right) \tag{8.5.4}$$

where fifth order accuracy is obtained by choosing, e.g.,

$w_1 = 0.2615038147$, $w_2 = -0.2765809214$, $w_3 = 0.5947141647$
$w_4 = 0.4203629420$, $a_1 = 1.5471214403$, $a_2 = 0.1756458393$
$a_3 = 0.1243059001$, $a_4 = 0.1009316694$, $a_5 = 0.1100539630$
$a_6 = 0.9997431862$, $a_7 = -0.0928890403$, $a_8 = -0.6201812828$
$a_{11} = 0.5 - a_4 - a_5 = 0.2890143692$, $a_{22} = 1 - a_6 - a_7 - a_8 = 0.7133271396$.

Thus, the new fifth order arithmetic mean WRK method can be written as follows:-

$$y_{n+1} = y_n + h\left[0.2615038147\left(\frac{k_1 + k_2}{2}\right) - 0.2765809214\left(\frac{k_2 + k_3}{2}\right)\right.$$

$$\left. + 0.5947141647\left(\frac{k_3 + k_4}{2}\right) + 0.4203629420\left(\frac{k_4 + k_5}{2}\right)\right] \tag{8.5.5}$$

where $k_1 = f(y_n)$

$$k_2 = f(y_n + 1.5471214403hk_1)$$

$$k_3 = f(y_n + 0.1756458393hk_1 + 0.1243059001hk_2)$$

$$k_4 = f\left(y_n + 0.1009316694hk_1 + 0.1100539630hk_2 + 0.2890143692hk_3\right)$$

$$k_5 = f\left(y_n + 0.9997431862hk_1 - 0.0928890403hk_2 - 0.6201812828hk_3 + 0.7133271396hk_4\right) . \quad \ldots \quad (8.5.6)$$

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

$$\begin{aligned} y' &= F(x, y, v) \quad , \quad y(x_0) = \alpha \\ v' &= G(x, y, v) \quad , \quad v(x_0) = \beta \end{aligned} \qquad (8.5.7)$$

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:

$$\begin{aligned}
k_1 &= F(x_n, y_n, v_n) \\
k_2 &= F(x_n + a_1h, y_n + a_1hk_1, v_n + a_1hm_1) \\
k_3 &= F(x_n + (a_2 + a_3)h, y_n + a_2hk_1 + a_3hk_2, v_n + a_2hm_1 + a_3hm_2) \\
k_4 &= F(x_n + (a_4 + a_5 + a_{11})h, y_n + a_4hk_1 + a_5hk_2 + a_{11}hk_3, \\
&\qquad v_n + a_4hm_1 + a_5hm_2 + a_{11}hm_3) \\
k_5 &= F(x_n + (a_6 + a_7 + a_8 + a_{22})h, y_n + a_6hk_1 + a_7hk_2 + a_8hk_3 + a_{22}hk_4, \\
&\qquad v_n + a_6hm_1 + a_7hm_2 + a_8hm_3 + a_{22}hm_4) \qquad (8.5.8)
\end{aligned}$$

and $m_1 = G(x_n, y_n, v_n)$

$$m_2 = G(x_n + a_1h, y_n + a_1hk_1, v_n + a_1hm_1)$$

$$m_3 = G(x_n + (a_2 + a_3)h, y_n + a_2hk_1 + a_3hk_2, v_n + a_2hm_1 + a_3hm_2)$$

$$m_4 = G(x_n + (a_4 + a_5 + a_{11})h, y_n + a_4hk_1 + a_5hk_2 + a_{11}hk_3, v_n + a_4hm_1 + a_5hm_2 + a_{11}hm_3)$$

$$m_5 = G(x_n + (a_6 + a_7 + a_8 + a_{22})h, y_n + a_6hk_1 + a_7hk_2 + a_8hk_3 + a_{22}hk_4, v_n + a_6hm_1 + a_7hm_2 + a_8hm_3 + a_{22}hm_4) \quad (8.5.9)$$

where $y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{k_i + k_{i+1}}{2}\right)\right]$ (8.5.10)

$$v_{n+1} = v_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{m_i + m_{i+1}}{2}\right)\right] \quad (8.5.11)$$

8.5.2 Weighted Fifth-Order Formulas for Second-Order Differential Equations

We consider initial value problems in the form

$$y'' = f(x, y, y')$$
$$y(x_0) = \alpha \quad , \quad y'(x_0) = \beta \quad (8.5.12)$$

If let $y' = v$, the equation (8.5.12) is equivalent to the system

$$y' = v \qquad y(x_0) = \alpha$$
$$v' = f(x, y, v) \qquad v(x_0) = \beta \quad (8.5.13)$$

Now the system of equation in (8.5.13) can be written similarly to equation (8.5.7) as

$$\left.\begin{aligned} y' &= F(x, y, v) = v \\ v' &= G(x, y, v) = f(x, y, v) \end{aligned}\right\} \quad (8.5.14)$$

Therefore, a set of k's and m's in (8.5.8) and (8.5.9) can be written in the form

$$k_1 = v_n$$
$$k_2 = v_n + a_1hm_1$$
$$k_3 = v_n + a_2hm_1 + a_3hm_2$$
$$k_4 = v_n + a_4hm_1 + a_5hm_2 + a_{11}hm_3$$
$$k_5 = v_n + a_6hm_1 + a_7hm_2 + a_8hm_3 + a_{22}hm_4 \quad (8.5.15)$$

By substituting equations (8.5.14)-(8.5.15) into equation (8.5.9), we obtain

$$m_1 = f(x_n, y_n, v_n)$$

$$m_2 = f(x_n + a_1 h, y_n + a_1 h v_n, v_n + a_1 h m_1)$$

$$m_3 = f(x_n + (a_2 + a_3)h, y_n + a_2 h v_n + a_3 h v_n + a_1 a_3 h^2 m_1 , v_n + a_2 h m_1 + a_3 h m_2)$$

$$m_4 = f(x_n + (a_4 + a_5 + a_{11})h, y_n + (a_4 + a_5 + a_{11})h v_n + (a_1 a_5 + a_2 a_{11})h^2 m_1 + a_3 a_{11} h^2 m_2, v_n + a_4 h m_1 + a_5 h m_2 + a_{11} h m_3)$$

$$m_5 = f(x_n + (a_6 + a_7 + a_8 + a_{22})h, y_n + (a_6 + a_7 + a_8 + a_{22})h v_n + (a_1 a_7 + a_2 a_8 + a_4 a_{22})h^2 m_1 + (a_3 a_8 + a_5 a_{22})h^2 m_2 + a_{11} a_{22} h^2 m_3, v_n + a_6 h m_1 + a_7 h m_2 + a_8 h m_3 + a_{22} h m_4) \tag{8.5.16}$$

and then equation (8.5.10) and (8.5.11) can be written as

$$y_{n+1} = y_n + h\left[w_1\left(\frac{k_1 + k_2}{2}\right) + w_2\left(\frac{k_2 + k_3}{2}\right) + w_3\left(\frac{k_3 + k_4}{2}\right) + w_4\left(\frac{k_4 + k_5}{2}\right)\right].$$

$$= y_n + \frac{h}{2}\left[w_1(2v_n + a_1 h m_1) + w_2(2v_n + (a_1 + a_2)h m_1 + a_3 h m_2) + w_3(2v_n + (a_2 + a_4)h m_1 + (a_3 + a_5)h m_2 + a_{11} h m_3) + w_4(2v_n + (a_4 + a_6)h m_1 + (a_5 + a_7)h m_2 + (a_8 + a_{11})h m_3 + a_{22} h m_4\right]$$

$$= y_n + h\sum_{i=1}^{4} w_i v_n + \frac{h^2}{2}\{[(a_1 + a_2)w_2 + (a_2 + a_4)w_3 + (a_4 + a_6)w_4]m_1 + [(a_3 + a_5)w_3 + (a_5 + a_7)w_4]m_2 + [(a_8 + a_{11})w_4]m_3 + w_1 a_1 m_1 + w_2 a_3 m_2 + w_3 a_{11} m_3 + w_4 a_{22} m_4\}$$

$$= y_n + h v_n + \frac{h^2}{2}\{[(a_1 + a_2)w_2 + (a_2 + a_4)w_3 + (a_4 + a_6)w_4]m_1 + [(a_3 + a_5)w_3 + (a_5 + a_7)w_4]m_2 + [(a_8 + a_{11})w_4]m_3 + w_1 a_1 m_1 + w_2 a_3 m_2 + w_3 a_{11} m_3 + w_4 a_{22} m_4\}$$

$$y_{n+1} = y_n + hv_n + \frac{h^2}{2}[0.5552610794m_1 + 0.1122119872m_2 + 0.0326706395m_3 + 0.2998562939m_4]$$

or $$y_{n+1} = y_n + hv_n + \frac{h^2}{2}\left[\frac{31636}{56975}m_1 + \frac{9185}{81854}m_2 + \frac{3262}{99845}m_3 + \frac{56964}{189971}m_4\right] \quad \ldots \quad (8.5.17)$$

and $$v_{n+1} = v_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{m_i + m_{i+1}}{2}\right)\right] \quad (8.5.18)$$

8.5.3 Numerical Example

We solve the second-order equation

$$\begin{aligned} &y'' + 3y' + 2y = 2\exp(-3x) \\ &y(0) = 1 \quad , \quad y'(0) = -2 \end{aligned} \quad (8.5.19)$$

If we denote $y' = v$, the differential equation can be written as

$$v' + 3v + 2y = 2\exp(-3x).$$

We can write equation (8.5.19) as the system of equations

$$\begin{aligned} y' &= v \\ v' &= -2y - 3v + 2\exp(-3x) \end{aligned} \quad (8.5.20)$$

with exact solution $y(x) = \exp(-x) - \exp(-2x) + \exp(-3x)$ and initial values $y(0) = 1$, $v(0) = -2$. By using h=0.01, we obtain a solution in the region $0 \le x \le 1$ with this new equation in (8.5.17) and (8.5.18) as shown in Table 8.7.

Table 8.7

x	Numerical Solution	Exact Solution	Absolute Error
0.1	0.8269249454E+00	0.8269248856E+00	0.5972778871E-07
0.2	0.6972224306E+00	0.6972223431E+00	0.8745288960E-07
0.3	0.5985763393E+00	0.5985762443E+00	0.9500676212E-07
0.4	0.5221853843E+00	0.5221852938E+00	0.9046166094E-07
0.5	0.4617814579E+00	0.4617813787E+00	0.7923542494E-07
0.6	0.4129163773E+00	0.4129163124E+00	0.6488580956E-07
0.7	0.3724448178E+00	0.3724447681E+00	0.4967923989E-07
0.8	0.3381504344E+00	0.3381503994E+00	0.3499606382E-07
0.9	0.3084763059E+00	0.3084762843E+00	0.2161763496E-07
1.0	0.2823312362E+00	0.2823312263E+00	0.9928264599E-08

8.6 NEW RUNGE KUTTA STARTERS FOR MULTISTEP METHODS

In many publications e.g. Gear [1980], Hindmarsh [1974], Krogh [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 $(x_i, y_i),(x_{i-1}, y_{i-1}),(x_{i-2}, y_{i-2}),\ldots$. 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

$$y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i \left(\frac{k_i + k_{i+1}}{2}\right)\right] \tag{8.6.1}$$

where $\sum_{i=1}^{4} w_i = 1$

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + a_1 hk_1)$$

$$k_3 = f(y_n + a_2 hk_1 + a_3 hk_2)$$

$$k_4 = f\left(y_n + a_4 hk_1 + a_5 hk_2 + \left(\tfrac{1}{2} - a_4 - a_5\right) hk_3\right)$$

$$k_5 = f\left(y_n + a_6 hk_1 + a_7 hk_2 + a_8 hk_3 + (1 - a_6 - a_7 - a_8) hk_4\right)$$

$$\ldots \qquad (8.6.2)$$

where fifth order accuracy is obtained by choosing, e.g.,

$w_1 = 0.2615038147$, $w_2 = -0.2765809214$, $w_3 = 0.5947141647$
$w_4 = 0.4203629420$, $a_1 = 1.5471214403$, $a_2 = 0.1756458393$
$a_3 = 0.1243059001$, $a_4 = 0.1009316694$, $a_5 = 0.1100539630$
$a_6 = 0.9997431862$, $a_7 = -0.0928890403$, $a_8 = -0.6201812828$

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

$$y_{n+1} = y_n + \frac{h}{24}\left[55 f_n - 59 f_{n-1} + 37 f_{n-2} - 9 f_{n-3}\right] . \qquad (8.6.3)$$

For solving equation (8.6.3), we need $f_i, i = 0,1,2,3$ by using the fourth order method in equation (8.0.1).

The five-step fifth order multistep Adam's method is of the form

$$y_{n+1} = y_n + \frac{h}{720}\left[1901 f_n - 2774 f_{n-1} + 2616 f_{n-2} - 1274 f_{n-3} + 251 f_{n-4}\right] \qquad (8.6.4)$$

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 $x_{i+1} = x_i + h$

$$y_{n+1} = y_n + \frac{h}{24}\left[55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}\right]$$

$f_{i-3} = f_{i-2}$

$f_{i-2} = f_{i-1}$

$f_{i-1} = f_i$

$f_i = f(x_{i+1}, y_{i+1})$

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 $x_{i+1} = x_i + h$

$$y_{n+1} = y_n + \frac{h}{720}\left[1901f_n - 2774f_{n-1} + 2616f_{n-2} - 1274f_{n-3} + 251f_{n-4}\right]$$

$f_{i-4} = f_{i-3}$

$f_{i-3} = f_{i-2}$

$f_{i-2} = f_{i-1}$

$f_{i-1} = f_i$

$f_i = f(x_{i+1}, y_{i+1})$

Figure 8.6 : Algorithm for the fifth order Adam's multistep method

8.6.3 Numerical Example

We consider the IVPs

$$y' = 1 + x - y \quad , \quad y(0) = 1 \quad , \quad 0 \le x \le 1 \qquad (8.6.5)$$

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

	X	Adcom44	Adcem44	Adrms44	Adgm44	Adam44	Adam54
10	0.1	.144E-04	.478E-05	.648E-05	.110E-04	.354E-09	.228E-09
20	0.2	.130E-04	.433E-05	.586E-05	.996E-05	.101E-08	.496E-09
30	0.3	.118E-04	.392E-05	.531E-05	.901E-05	.180E-08	.711E-09
40	0.4	.106E-04	.355E-05	.480E-05	.816E-05	.271E-08	.881E-09
50	0.5	.963E-05	.321E-05	.435E-05	.738E-05	.369E-08	.101E-08
60	0.6	.871E-05	.291E-05	.393E-05	.667E-05	.473E-08	.111E-08
70	0.7	.788E-05	.263E-05	.356E-05	.604E-05	.582E-08	.118E-08
80	0.8	.714E-05	.238E-05	.322E-05	.546E-05	.692E-08	.123E-08
90	0.9	.646E-05	.216E-05	.292E-05	.494E-05	.803E-08	.125E-08
100	1.0	.585E-05	.195E-05	.264E-05	.447E-05	.913E-08	.126E-08

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

	X	Adam45	Adam55
10	0.1	.1212808E-11	.2942535E-11
20	0.2	.1659117E-11	.9436896E-13
30	0.3	.3995471E-11	.2579714E-11
40	0.4	.5871748E-11	.4590328E-11
50	0.5	.7355228E-11	.6195489E-11
60	0.6	.8502976E-11	.7453371E-11
70	0.7	.9365841E-11	.8415935E-11
80	0.8	.9987344E-11	.9127810E-11
90	0.9	.1040568E-10	.9628298E-11
100	1.0	.1065370E-10	.9950707E-11

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

$$L[x^r;h]=0, \quad r=0,1,\ldots,p; \quad L[x^{p+1};h]\neq 0, \tag{9.0.1}$$

where L, the associated linear difference operator, is defined by the equation

$$L[y(x);h]=\sum_{j=0}^{k}\left[\alpha_j y(x+jh)-h\beta_j y'(x+jh)\right]. \tag{9.0.2}$$

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

$$\sum_{j=0}^{k}\alpha_j y_{n+j}=h\sum_{j=0}^{k}\beta_j(v)f_{n+j}, \quad v=\omega h, \quad \alpha_k=+1 \tag{9.0.3}$$

is said to be of trigonometric order q relative to the frequency ω if the associated linear difference operator

$$L_\omega[y(x);h]=\sum_{j=0}^{k}\left[\alpha_j y(x+jh)-h\beta_j y'(x+jh)\right] \tag{9.0.4}$$

satisfies

i) $L_\omega[1;h]=0,$

ii) $L_\omega[\cos(r\omega x);h]\equiv L_\omega[\sin(r\omega x);h]\equiv 0, \quad r=1,2,\ldots,q,$ (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 $\alpha_0, \beta_1(v)$, *and* $\beta_0(v)$ we obtain a base formula in the form

$$y_{n+1} = y_n + h[\beta_0 f_n + \beta_1 f_{n+1}] \tag{9.0.6}$$

which must satisfy the conditions

$$L_\omega[1;h] = 1 + \alpha_0 = 0, \tag{9.0.6-i}$$

$$L_\omega[\cos(\omega x);h] = \cos[\omega(x+h)] - \alpha_0 \cos(\omega x) + h\omega\{\beta_1 \sin[\omega(x+h)] + \beta_0 \sin(\omega x)\} \equiv 0, \tag{9.0.6-ii}$$

$$L_\omega[\sin(\omega x);h] = \sin[\omega(x+h)] - \alpha_0 \sin(\omega x) - h\omega\{\beta_1 \cos[\omega(x+h)] + \beta_0 \cos(\omega x)\} \equiv 0, \tag{9.0.6-iii}$$

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

$$\alpha_0 = -1, \quad \beta_0 = \beta_1 = \frac{[\tan(\frac{v}{2})]}{v} \qquad \text{where } v = \omega h$$

which is
$$y_{n+1} = y_n + \frac{h\tan(\frac{v}{2})}{v}(f_n + f_{n+1}) \tag{9.0.7}$$

and is equvalent to the well known Trapezoidal rule. We now, expand $\tan(\frac{v}{2})$ in a power series where

$$\tan(v) = v + \frac{1}{3}v^3 + \frac{2}{15}v^5 + \frac{17}{315}v^7 + \ldots$$

and by replacing v with $(\frac{v}{2})$ we have

$$\tan(\tfrac{v}{2}) = \frac{v}{2} + \frac{1}{3}(\tfrac{v}{2})^3 + \frac{2}{15}(\tfrac{v}{2})^5 + \frac{17}{315}(\tfrac{v}{2})^7 + \ldots$$

$$= \frac{v}{2} + \frac{v^3}{24} + \frac{v^5}{240} + \frac{17v^7}{40320} + \ldots$$

and $$\frac{\tan(\tfrac{v}{2})}{v} = \frac{1}{2} + \frac{v^2}{24} + \frac{v^4}{240} + \frac{17v^6}{40320} + \ldots . \tag{9.0.8}$$

Substituting (9.0.8) into (9.0.7) we obtain

$$y_{n+1} = y_n + h\left[\frac{1}{2} + \frac{v^2}{24} + \frac{v^4}{240} + \frac{17v^6}{40320} + \ldots\right](f_n + f_{n+1}) . \tag{9.0.9}$$

and expanding the right hand-side (9.0.9) at x_n we have

$$y_{n+1} = y_n + h\left[\frac{1}{2} + \frac{v^2}{24} + \frac{v^4}{240} + \frac{17v^6}{40320} + \ldots\right](2y_n' +$$

$$+hy_n'' + \frac{h^2}{2}y_n''' + \frac{h^3}{6}y_n^{iv} + \ldots)$$

$$= y_n + hy_n' + \frac{h^2 y_n''}{2} + h^3\left(\frac{y_n'''}{4} + \frac{y_n'}{12}\right) + \ldots \tag{9.0.10}$$

where the local truncation error (LTE) is given by

$$LTE = h^3\left[\frac{y_n'''}{6} - \frac{y_n'''}{4} - \frac{\omega^2 y_n'}{12}\right] + 0(h^4)$$

$$= -\frac{h^3}{12}\left(y_n''' + \omega^2 y_n'\right) + 0(h^4). \tag{9.0.11}$$

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

(9.0.3) whose trigonometric order is $q = \frac{k}{2}$ $\left(or\ q = \frac{1}{2}(k+1) \right)$.
In both cases, when $\omega = 0$ the method will reduce to a linear multistep method of algebraic order $2q$.

9.1 RUNGE-KUTTA METHODS FOR THE OSCILLATORY PROBLEM

The general explicit one step method as a particular case for a classical Runge-Kutta method can be written as

$$y_{n+1} = y_n + h\phi(x_n, y_n, h) \tag{9.1.1}$$

For example, two classical second order methods are the modified Euler method and the improved Euler method given as

$$y_{n+1} = y_n + hk_2$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1 \right) \tag{9.1.2}$$

and

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

where $k_1 = f(x_n, y_n)$

$$k_2 = f(x_n + h, y_n + hk_1) \tag{9.1.3}$$

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

$$y_{n+1} = y_n + h\frac{\tan(\frac{v}{2})}{v}(k_1 + k_2)$$

where $k_1 = f(x_n, y_n)$

$$k_2 = f(x_n + h, y_n + hk_1) \tag{9.1.4}$$

We now establish the formula in (9.1.4) for solving the oscillatory problem by writing

$$k_1 = f$$

$$k_2 = f + hff_y + \frac{1}{2}h^2 f_{yy} + \ldots$$

and substituted in equation (9.1.4) to give,

$$y_{n+1} = y_n + h\left[\tfrac{1}{2} + \frac{\omega^2 h^2}{24} + \frac{\omega^4 h^4}{240} + \ldots\right]\left[2f + hff_y + \frac{1}{2}h^2 f^2 f_{yy} + \ldots\right]$$

$$= y_n + h\left\{f + \frac{1}{2}hff_y + h^2\left(\frac{\omega^2}{12} + \frac{1}{4}f^2 f_{yy}\right) + \ldots\right\}$$

$$= y_n + hf + \frac{h^2}{2}ff_y + h^3\left\{\frac{\omega^2}{12} + \frac{1}{4}f^2 f_{yy}\right\} + \ldots \qquad (9.1.5)$$

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

$$LTE = h^3\left(\frac{1}{6}ff_y^2 + \frac{1}{6}f^2 f_{yy} - \frac{1}{12}\omega^2 - \frac{1}{4}f^2 f_{yy}\right)$$

$$= h^3\left(\frac{1}{6}ff_y^2 - \frac{1}{12}ff_{yy} - \frac{1}{12}\omega^2\right).$$

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

$$y_{n+3} = y_n + h\left[\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2} + \beta_3 f_{n+3}\right] \tag{9.2.1}$$

and by using the difference operator in (9.1.3) and (9.1.4) we obtain

a) $L_\omega[1:h] = 0$ (9.2.2-i)

b) For r = 1 , we have

i) $L_\omega\left[\cos(\omega x):h\right] = 0$

$$-\cos(\omega x) + \cos(\omega x + 3v) = -v\beta_0 \sin(\omega x) - v\beta_1 \sin(\omega x + v) - v\beta_2 \sin(\omega x + 2v) - v\beta_3 \sin(\omega x + 3v)$$

and when $\omega x = 0$, we have

$$-v\beta_1 \sin(v) - v\beta_2 \sin(2v) - v\beta_3 \sin(3v) = \cos(3v) - 1 \tag{9.2.2-ii}$$

ii) $L_\omega\left[\sin(\omega x):h\right] = 0$

$$-\sin(\omega x) + \sin(\omega x + 3v) = v\beta_0 \cos(\omega x) + v\beta_1 \cos(\omega x + v) + v\beta_2 \cos(\omega x + 2v) + v\beta_3 \cos(\omega x + 3v)$$

and for $\omega x = 0$, we obtain

$$v\beta_0 + v\beta_1 \cos(v) + v\beta_2 \cos(2v) + v\beta_3 \cos(3v) = \sin(3v) \tag{9.2.2-iii}$$

c) For r = 2 ;

i) $L_\omega\left[\cos(2\omega x):h\right] = 0$

$$-\cos(2\omega x) + \cos(2\omega x + 6v) = -2v\beta_0 \sin(2\omega x) - 2v\beta_1 \sin(2\omega x + 2v) - 2v\beta_2 \sin(2\omega x + 4v) - 2v\beta_3 \sin(2\omega x + 6v)$$

and for $\omega x = 0$, we have

$$-2v\beta_1 \sin(2v) - 2v\beta_2 \sin(4v) - 2v\beta_3 \sin(6v) = \cos(6v) - 1 \tag{9.2.2-iv}$$

ii) $L_\omega\left[\sin(2\omega x):h\right] = 0$

$$-\sin(2\omega x) + \sin(2\omega x + 6v) = 2v\beta_0 \cos(2\omega x) + 2v\beta_1 \cos(2\omega x + 2v) + 2v\beta_2 \cos(2\omega x + 4v) + 2v\beta_3 \cos(2\omega x + 6v)$$

and for $\omega x = 0$, we obtain

$$2v\beta_0 + 2v\beta_1\cos(2v) + 2v\beta_2\cos(4v) + 2v\beta_3\cos(6v) = \sin(6v) \qquad (9.2.2\text{-v})$$

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.,

$$\beta_0 = \big(3(2940000 - 4165000v^2 + 29473500v^4 - 270016250v^6 + 451324475v^8$$
$$-12319825v^{10} - 781610v^{12} - 4280400v^{14} + 412992v^{16})\big) /$$
$$\big(140(168000 - 280000v^2 + 207900v^4 - 1341000v^6 + 946545v^8 - 104600v^{10} + 2016v^{12})\big)$$
$$\ldots \qquad (9.2.3\text{-i})$$

$$\beta_1 = \big(27(14000 - 24500v^2 - 107800v^4 + 830425v^6 - 1780875v^8 +$$
$$750600v^{10} - 62208v^{12})\big) /$$
$$\big(2(168000 - 280000v^2 + 207900v^4 - 1341000v^6 + 946545v^8 - 104600v^{10} + 2016v^{12})\big)$$
$$\ldots \qquad (9.2.3\text{-ii})$$

$$\beta_2 = \big(15(28000 - 49000v^2 + 294700v^4 - 1570450v^6 + 2144295v^8 -$$
$$287685v^{10} + 8262v^{12})\big) /$$
$$\big(4(168000 - 280000v^2 + 207900v^4 - 1341000v^6 + 946545v^8 - 104600v^{10} + 2016v^{12})\big)$$
$$\ldots \qquad (9.2.3\text{-iii})$$

$$\beta_3 = \big(15(8400 - 11900v^2 - 68880v^4 + 149315v^6 - 118043v^8 + 14536v^{10} - 384v^{12})\big) /$$
$$\big(2(168000 - 280000v^2 + 207900v^4 - 1341000v^6 + 946545v^8 - 104600v^{10} + 2016v^{12})\big)$$
$$\ldots \qquad (9.2.3\text{-iv})$$

Equations (9.2.3-i)-(9.2.3-iv) can also be simplified by Mathematica, i.e.,

$$In[1] := Series[\beta_0, \{v, 0, 12\}]$$

and we obtain $\beta_0, \beta_1, \beta_2$ and β_3 as

$$\beta_0 = \frac{3}{8}\left(1 + \frac{v^2}{4} + \frac{2209v^4}{240} - \frac{1387597v^6}{20160} + \frac{28749227v^8}{1209600} + \frac{2805517639v^{10}}{14515200} - \frac{1881719979239v^{12}}{6096384000} + 0(v^{13})\right)$$

$$\beta_1 = \frac{9}{8}\left(1 - \frac{v^2}{12} - \frac{1307v^4}{144} + \frac{3161533v^6}{60480} - \frac{18221411v^8}{518400} - \frac{6140536111v^{10}}{43545600} + \frac{4981300261121v^{12}}{18289152000} + 0(v^{13})\right)$$

$$\beta_2 = \frac{9}{8}\left(1 - \frac{v^2}{12} + \frac{6587v^4}{720} - \frac{396199v^6}{12096} + \frac{15858841v^8}{3628800} + \frac{972466693v^{10}}{8709120} - \frac{2415491349181v^{12}}{18289152000} + 0(v^{13})\right)$$

$$\beta_3 = \frac{3}{8}\left(1 + \frac{v^2}{4} - \frac{433v^4}{48} + \frac{209939v^6}{20160} + \frac{13097489v^8}{1209600} - \frac{136652999v^{10}}{2073600} + \frac{65908719619v^{12}}{6096384000} + 0(v^{13})\right)$$

Thus the proposed method is given by ,

$$y_{n+3} = y_n + 3h(\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2} + \beta_3 f_{n+3}) \tag{9.2.4}$$

where

$$\beta_0 = \frac{1}{8}\left(1 + \frac{v^2}{4} + \frac{2209v^4}{240} - \frac{1387597v^6}{20160} + \frac{28749227v^8}{1209600} + \frac{2805517639v^{10}}{14515200} - \frac{1881719979239v^{12}}{6096384000} + 0(v^{13})\right)$$

$$\beta_1 = \frac{3}{8}\left(1 - \frac{v^2}{12} - \frac{1307v^4}{144} + \frac{3161533v^6}{60480} - \frac{18221411v^8}{518400} - \frac{6140536111v^{10}}{43545600} + \frac{4981300261121v^{12}}{18289152000} + 0(v^{13})\right)$$

$$\beta_2 = \frac{3}{8}\left(1 - \frac{v^2}{12} + \frac{6587v^4}{720} - \frac{396199v^6}{12096} + \frac{15858841v^8}{3628800} + \frac{972466693v^{10}}{8709120} - \frac{2415491349181v^{12}}{18289152000} + 0(v^{13})\right)$$

$$\beta_3 = \frac{1}{8}\left(1 + \frac{v^2}{4} - \frac{433v^4}{48} + \frac{209939v^6}{20160} + \frac{13097489v^8}{1209600} - \frac{136652999v^{10}}{2073600} + \frac{65908719619v^{12}}{6096384000} + 0(v^{13})\right)$$

Now, the formula in equation (9.2.4) developed for solving oscillatory problems is transformed into

$$y_{n+1} = y_n + h\left[w_1 f_n + w_2 f_{n+\frac{1}{3}} + w_3 f_{n+\frac{2}{3}} + w_4 f_{n+1}\right] \tag{9.2.5-i}$$

where $w_1 = \frac{1}{8}$, $w_2 = \frac{3}{8}$, $w_3 = \frac{3}{8}$, $w_4 = \frac{1}{8}$ and we replace $k_1 = f_n$, $k_2 = f_{n+\frac{1}{3}}$, $k_3 = f_{n+\frac{2}{3}}$ and $k_4 = f_{n+1}$ in the form

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + ha_1k_1)$$

$$k_3 = f\left(y_n + ha_2k_1 + h\left(\tfrac{2}{3} - a_2\right)k_2\right)$$

$$k_4 = f\left(y_n + ha_3k_1 + ha_4k_2 + h(1 - a_3 - a_4)k_3\right)$$

and $$y_{n+1} = y_n + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4] \tag{9.2.5-ii}$$

The reason for taking v = 0 is obvious in that we do not intend to relate a_i with v (see Sanugi [1986], p 311).

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.,

$$h^2ff_y: \quad 1-3a_1=0 \tag{9.2.6-i}$$

$$h^3ff_y^2: \quad 2-6a_1+9a_1a_2+2a_3+2a_4-3a_1a_4=0 \tag{9.2.6-ii}$$

$$h^3f^2f_{yy}: \quad (1-3a_1)(1+3a_1)=0 \tag{9.2.6-iii}$$

$$h^4ff_y^3: \quad 1-2a_1+3a_1a_2+2a_1a_3-3a_1a_2a_3+2a_1a_4-3a_1a_2a_4=0 \tag{9.2.6-iv}$$

$$h^4f^2f_yf_{yy}: 8-24a_1-18a_1^2+36a_1a_2+27a_1^2a_2+16a_3+16a_4-18a_1a_4-9a_1^2a_4=0 \quad \ldots \tag{9.2.6-v}$$

$$h^4f^3f_{yyy}: \quad (1-3a_1)(1+3a_1+9a_1^2)=0 \tag{9.2.6-vi}$$

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.,

Parameter	Set 1	Set 2
a_1	$\frac{1}{3}$	$\frac{1}{3}$
a_2	$-\frac{1}{3}$	1
$a_{21}=\frac{2}{3}-a_2$	1	$-\frac{1}{3}$
a_3	1	-7
a_4	-1	11
$a_{31}=1-a_3-a_4$	1	-3

Thus, two sets of fourth order Runge-Kutta formula for solving oscillatory problem can be written as

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

$$k_2=f\left(x_n+\frac{h}{3},y_n+\frac{1}{3}hk_1\right)$$

$$k_3=f\left(x_n+\frac{2}{3}h,y_n-\frac{1}{3}hk_1+hk_2\right)$$

$$k_4=f(x_n+h,y_n+hk_1-hk_2+hk_3)$$

and $$y_{n+1}=y_n+h[w_1k_1+w_2k_2+w_3k_3+w_4k_4] \tag{9.2.7}$$

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

$$y_{n+1} = y_n + h[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4] \qquad (9.2.8)$$

where in equations (9.2.7) and (9.2.8) we have the values w_i , $i = 1,2,3,4$ given by

$$w_1 = \frac{1}{8}\left(1 + \frac{v^2}{4} + \frac{2209v^4}{240} - \frac{1387597v^6}{20160} + \frac{28749227v^8}{1209600} + \frac{2805517639v^{10}}{14515200} - \frac{1881719979239v^{12}}{6096384000} + 0(v^{13})\right)$$

$$w_2 = \frac{3}{8}\left(1 - \frac{v^2}{12} - \frac{1307v^4}{144} + \frac{3161533v^6}{60480} - \frac{18221411v^8}{518400} - \frac{6140536111v^{10}}{43545600} + \frac{4981300261121v^{12}}{18289152000} + 0(v^{13})\right)$$

$$w_3 = \frac{3}{8}\left(1 - \frac{v^2}{12} + \frac{6587v^4}{720} - \frac{396199v^6}{12096} + \frac{15858841v^8}{3628800} + \frac{972466693v^{10}}{8709120} - \frac{2415491349181v^{12}}{18289152000} + 0(v^{13})\right)$$

$$w_4 = \frac{1}{8}\left(1 + \frac{v^2}{4} - \frac{433v^4}{48} + \frac{209939v^6}{20160} + \frac{13097489v^8}{1209600} - \frac{136652999v^{10}}{2073600} + \frac{65908719619v^{12}}{6096384000} + 0(v^{13})\right)$$

9.2.2 Error Analysis

By substituting the values a_i , $1 \le i \le 4$ in set one into equation (9.2.7) using Mathematica and evaluating all the terms up to (h^5) to represent the local truncation error for this method, we have

$$LTE = \frac{h^5}{6480}\left[54ff_y^4 - 81ff_y^2f_{yy} - 9f^2f_{yy}^2 + 13f^2f_yf_{yyy} - f^3f_{yyyy}\right] + O(h^6) \text{ as } h \to 0. \qquad (9.2.9)$$

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

$$k_1 = \lambda y_n$$

$$k_2 = \lambda\left(y_n + \frac{1}{3}hk_1\right)$$

$$k_3 = \lambda\left(y_n - \frac{1}{3}hk_1 + hk_2\right)$$

$$k_4 = \lambda\left(y_n + hk_1 - hk_2 + hk_3\right) \qquad (9.2.10)$$

By substituting $k_i, 1 \le i \le 4$ in equation (9.2.10) and $w_i, 1 \le i \le 4$ into the new fourth order formula, i.e.,

$$y_{n+1} = y_n + h\left[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4\right] \qquad (9.2.11)$$

we obtain

$$y_{n+1} = y_n + (h\lambda)y_n + \frac{1}{2}(h\lambda)^2 y_n + \frac{1}{6}(h\lambda)^3 y_n + \frac{1}{24}(h\lambda)^4 y_n \, . \; \ldots \qquad (9.2.12)$$

By substituting $h\lambda = z$ in (9.2.12), we can show that

$$y_{n+1} = y_n + y_n\left[z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}\right] + 0\left(z^5\right) \; . \qquad (9.2.13)$$

Following equation (3.3.30), we write $\frac{y_{n+1}}{y_n} = Q$ in the equation (9.2.13), to obtain

$$Q = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4 + 0\left(z^5\right). \qquad (9.2.14)$$

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.,

$$\left|1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}\right| < 1 \quad . \qquad (9.2.15)$$

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.,

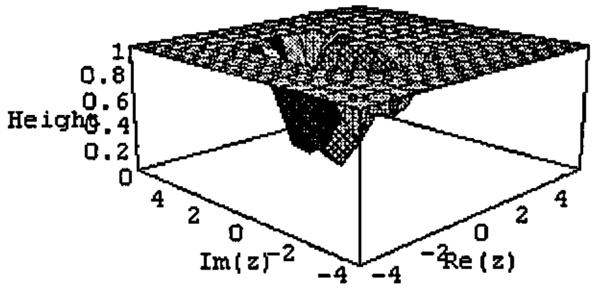


Figure 9.1: Graphic surface defined by fourth order method

and the stability region defined by the formula in equation (9.2.15) as shown in Figure 9.2.

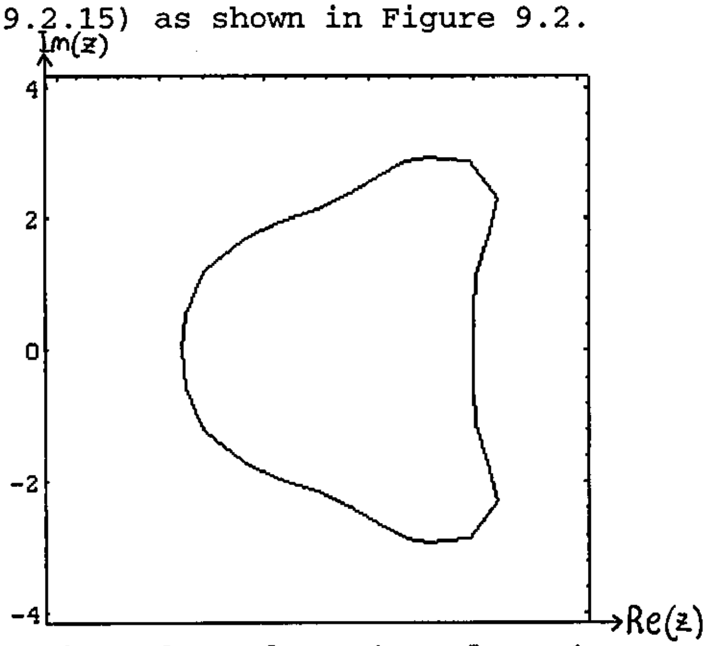


Figure 9.2: Stability region for fourth order method.

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) $L_\infty[1:h] = 0$ (9.3.2-i)

b) For r = 1

i) $L_\infty\left[\cos(\omega x):h\right] = 0$

$$-\cos(\omega x) + \cos(\omega x + 5v) = -v\beta_0 \sin(\omega x) - v\beta_1 \sin(\omega x + v) -v\beta_2 \sin(\omega x + 2v) - v\beta_3 \sin(\omega x + 3v) -v\beta_4 \sin(\omega x + 4v) - v\beta_5 \sin(\omega x + 5v)$$

and for $\omega x = 0$, we have

$$-v\beta_1 \sin(v) - v\beta_2 \sin(2v) - v\beta_3 \sin(3v) - v\beta_4 \sin(4v) - v\beta_5 \sin(5v) = \cos(5v) - 1 \quad \ldots \quad (9.3.2\text{-ii})$$

ii) $L_\infty\left[\sin(\omega x):h\right] = 0$

$$-\sin(\omega x) + \sin(\omega x + 5v) = v\beta_0 \cos(\omega x) + v\beta_1 \cos(\omega x + v) +v\beta_2 \cos(\omega x + 2v) + v\beta_3 \cos(\omega x + 3v) +v\beta_4 \cos(\omega x + 4v) + v\beta_5 \cos(\omega x + 5v)$$

and for $\omega x = 0$, we obtain

$$v\beta_0 + v\beta_1 \cos(v) + v\beta_2 \cos(2v) + v\beta_3 \cos(3v) + v\beta_4 \cos(4v) + v\beta_5 \cos(5v) = \sin(5v) \quad \ldots \quad (9.3.2\text{-iii})$$

c) For r = 2;

i) $L_\infty\left[\cos(2\omega x):h\right] = 0$

$$-\cos(2\omega x) + \cos(2\omega x + 10v) = -2v\beta_0 \sin(2\omega x) - 2v\beta_1 \sin(2\omega x + 2v) -2v\beta_2 \sin(2\omega x + 4v) - 2v\beta_3 \sin(2\omega x + 6v) -2v\beta_4 \sin(2\omega x + 8v) - 2v\beta_5 \sin(2\omega x + 10v)$$

and for $\omega x = 0$, we have

$$-2v\beta_1 \sin(2v) - 2v\beta_2 \sin(4v) - 2v\beta_3 \sin(6v) - 2v\beta_4 \sin(8v) - 2v\beta_5 \sin(10v) = \cos(10v) - 1 \quad \ldots \quad (9.3.2\text{-iv})$$

ii) $L_\infty\left[\sin(2\omega x):h\right] = 0$

$$-\sin(2\omega x) + \sin(2\omega x + 10v) = 2v\beta_0 \cos(2\omega x) + 2v\beta_1 \cos(2\omega x + 2v) +2v\beta_2 \cos(2\omega x + 4v) + 2v\beta_3 \cos(2\omega x + 6v) +2v\beta_4 \cos(2\omega x + 8v) + 2v\beta_5 \cos(2\omega x + 10v)$$

and for $\omega x = 0$, we obtain

$$2v\beta_0 + 2v\beta_1 \cos(2v) + 2v\beta_2 \cos(4v) + 2v\beta_3 \cos(6v) + 2v\beta_4 \cos(8v) + 2v\beta_5 \cos(10v) = \sin(10v) \quad \ldots \quad (9.3.2\text{-v})$$

d) For r = 3 ;

i) $L_\infty[\cos(3\omega x):h]=0$

$$-\cos(3\omega x)+\cos(3\omega x+15v)=-3v\beta_0\sin(3\omega x)-3v\beta_1\sin(3\omega x+3v)$$
$$-3v\beta_2\sin(3\omega x+6v)-3v\beta_3\sin(3\omega x+9v)$$
$$-3v\beta_4\sin(3\omega x+12v)-3v\beta_5\sin(3\omega x+15v)$$

and for $\omega x=0$, we have

$$-3v\beta_1\sin(3v)-3v\beta_2\sin(6v)-3v\beta_3\sin(9v)-3v\beta_4\sin(12v)-3v\beta_5\sin(15v)=\cos(15v)-1 \quad \ldots \quad (9.3.2\text{-vi})$$

ii) $L_\infty[\sin(3\omega x):h]=0$

$$-\sin(3\omega x)+\sin(3\omega x+15v)=3v\beta_0\cos(3\omega x)+3v\beta_1\cos(3\omega x+3v)$$
$$+3v\beta_2\cos(3\omega x+6v)+3v\beta_3\cos(3\omega x+9v)$$
$$+3v\beta_4\cos(3\omega x+12v)+3v\beta_5\cos(3\omega x+15v)$$

and for $\omega x=0$, we obtain

$$3v\beta_0+3v\beta_1\cos(3v)+3v\beta_2\cos(6v)+3v\beta_3\cos(9v)+3v\beta_4\cos(12v)+3v\beta_5\cos(15v)=\sin(15v) \quad \ldots \quad (9.3.2\text{-vii})$$

Equations (9.3.2-ii)-(9.3.2-vii) when solved simultaneously by Mathematica gives

$$\begin{aligned}\beta_0 =&(5\,(136217604096000 - 887804209152000\,v^2 + 166125005541734400\,v^4 -\\ &803845765365511680\,v^6 + 765544756910277254 40\,v^8 +\\ &27320330170447910400\,v^{10} + 83562156096513784536\,v^{12} -\\ &85698938485489371700 0\,v^{14} + 109737970465268155107 3\,v^{16} +\\ &431383337817696138 00\,v^{18} - 1076049130167396000\,v^{20} -\\ &922736375164800000\,v^{22} + 264776345856000000\,v^{24}))\,/\\ &(10752\,(192036096000 - 1344252672000\,v^2 + 4469640134400\,v^4 -\\ &508347691084800\,v^6 + 2320138789660440\,v^8 -\\ &844650807096600\,v^{10} - 69298164524589\,v^{12} -\\ &490888716667650\,v^{14} + 373375548002198\,v^{16} -\\ &38027971022400\,v^{18} + 531038592000\,v^{20}))\end{aligned} \quad (9.3.3\text{-i})$$

$$\begin{aligned}\beta_1 =&(625\,(691329945600 - 5092797265920\,v^2 - 1032231463038720\,v^4 +\\ &43387018368365952\,v^6 - 451634639711485104\,v^8 +\\ &3443112667193895 36\,v^{10} - 2425649423164625 07\,v^{12} +\\ &3150455038032486450\,v^{14} - 4326807844514380750\,v^{16} +\\ &1354683029790600000\,v^{18} - 59808221424000000\,v^{20}))\,/\\ &(1728\,(192036096000 - 1344252672000\,v^2 + 4469640134400\,v^4 -\\ &508347691084800\,v^6 + 2320138789660440\,v^8 -\\ &844650807096600\,v^{10} - 69298164524589\,v^{12} -\\ &490888716667650\,v^{14} + 373375548002198\,v^{16} -\\ &38027971022400\,v^{18} + 531038592000\,v^{20}))\end{aligned} \quad (9.3.3\text{ - ii})$$

$$\beta_2 = (625\,(1843546521600 - 12228858593280\,v^2 + 8439092811233280\,v^4 - 32979780319681843\,2\,v^6 + 3249145065440486016\,v^8 - 3905230975285925376\,v^{10} + 2346099604550380728\,v^{12} - 1229668040227699380\,0\,v^{14} + 126842116710914211\,25\,v^{16} - 1487586380157900000\,v^{18} + 29923076376000000\,v^{20}))\,/\,(6912\,(192036096000 - 1344252672000\,v^2 + 4469640134400\,v^4 - 508347691084800\,v^6 + 2320138789660440\,v^8 - 844650807096600\,v^{10} - 69298164524589\,v^{12} - 490888716667650\,v^{14} + 373375548002198\,v^{16} - 38027971022400\,v^{18} + 531038592000\,v^{20})) \quad (9.3.3 - iii)$$

$$\beta_3 = (625\,(25604812800 - 169845258240\,v^2 - 116167577621760\,v^4 + 3882134419599744\,v^6 - 35580793663355472\,v^8 + 40089266837582592\,v^{10} - 12854717663329101\,v^{12} + 7198712260759635\,0\,v^{14} - 4528409941262725\,0\,v^{16} + 4725345607800000\,v^{18} - 82067472000000\,v^{20}))\,/\,(96\,(192036096000 - 1344252672000\,v^2 + 4469640134400\,v^4 - 508347691084800\,v^6 + 2320138789660440\,v^8 - 844650807096600\,v^{10} - 69298164524589\,v^{12} - 490888716667650\,v^{14} + 373375548002198\,v^{16} - 38027971022400\,v^{18} + 531038592000\,v^{20})) \quad (9.3.3 - iv)$$

$$\beta_4 = (625\,(5530639564800 - 40742378127360\,v^2 + 8545306695690240\,v^4 - 254785366461072384\,v^6 + 1914797080967799168\,v^8 - 1370934790423091712\,v^{10} + 426843818430175944\,v^{12} - 197561824708099740\,0\,v^{14} + 1146898629298170875\,v^{16} - 115966038986325000\,v^{18} + 1888565260500000\,v^{20}))\,/\,(13824\,(192036096000 - 1344252672000\,v^2 + 4469640134400\,v^4 - 508347691084800\,v^6 + 2320138789660440\,v^8 - 844650807096600\,v^{10} - 69298164524589\,v^{12} - 490888716667650\,v^{14} + 373375548002198\,v^{16} - 38027971022400\,v^{18} + 531038592000\,v^{20})) \quad (9.3.3 - v)$$

$$\beta_5 = (5\,(21892114944000 - 142682819328000\,v^2 - 25811208395078400\,v^4 + 542547449348284800\,v^6 - 3237915694006329840\,v^8 + 1988009710663485600\,v^{10} - 589709412485954271\,v^{12} + 2608473533984918250\,v^{14} - 1482044278630418678\,v^{16} + 148151676311323200\,v^{18} - 2341045701504000\,v^{20}))\,/\,(1728\,(192036096000 - 1344252672000\,v^2 + 4469640134400\,v^4 - 508347691084800\,v^6 + 2320138789660440\,v^8 - 844650807096600\,v^{10} - 69298164524589\,v^{12} - 490888716667650\,v^{14} + 373375548002198\,v^{16} - 38027971022400\,v^{18} + 531038592000\,v^{20})) \quad (9..3.3- vi)$$

Equations (9.3.3-i)-(9.3.3-vi) can also be simplified by Mathematica to give

$$\beta_0 = \frac{95}{288}\left(1+\frac{55v^2}{114}+\frac{3282265v^4}{2736}-\frac{5513289335v^6}{114912}+\frac{258450119345v^8}{1378944}+\right.$$
$$\left.\frac{68594391998759v^{10}}{11819520}-\frac{727108666308742877v^{12}}{6949877760}+0\left(v^{13}\right)\right) \qquad (9.3.4\text{-i})$$

$$\beta_1 = \frac{125}{96}\left(1-\frac{11v^2}{30}-\frac{218729v^4}{144}+\frac{4969804253v^6}{90720}-\frac{673623326819v^8}{2721600}-\right.$$
$$\left.\frac{304273318350997v^{10}}{46656000}+\frac{337775919756561241 3v^{12}}{27433728000}+0\left(v^{13}\right)\right) \qquad (9.3.4\text{-ii})$$

$$\beta_2 = \frac{125}{144}\left(1+\frac{11v^2}{30}+\frac{3280991v^4}{720}-\frac{4365325187v^6}{30240}+\frac{1151739351529v^8}{1814400}+\right.$$
$$\left.\frac{276014322723379v^{10}}{15552000}-\frac{2985268375870083433v^{12}}{9144576000}+0\left(v^{13}\right)\right) \qquad (9.3.4\text{-iii})$$

$$\beta_3 = \frac{125}{144}\left(1+\frac{11v^2}{30}-\frac{3281509v^4}{720}+\frac{3699937313v^6}{30240}-\frac{198757705399v^8}{453600}-\right.$$
$$\left.\frac{255272260074121v^{10}}{15552000}+\frac{2503319588144210317v^{12}}{9144576000}+0\left(v^{13}\right)\right) \qquad (9.3.4\text{-iv})$$

$$\beta_4 = \frac{125}{96}\left(1-\frac{11v^2}{30}+\frac{218771v^4}{144}-\frac{2973583247v^6}{90720}+\frac{372098638237v^8}{5443200}+\right.$$
$$\left.\frac{234401042464003v^{10}}{46656000}-\frac{1960667278140426337v^{12}}{27433728000}+0\left(v^{13}\right)\right) \qquad (9.3.4\text{-v})$$

$$\beta_5 = \frac{95}{288}\left(1+\frac{55v^2}{114}-\frac{3280235v^4}{2736}+\frac{2186348165v^6}{114912}+\frac{1641506735v^8}{689472}-\right.$$
$$\left.\frac{41492416771241v^{10}}{11819520}+\frac{279364542028634873v^{12}}{6949877760}+0\left(v^{13}\right)\right) \qquad (9.3.4\text{-vi})$$

Thus, the method in equation (9.3.1) is now written as

$$y_{n+5} = y_n + 5h\left[\beta_0 f_n + \beta_1 f_{n+1} + \beta_2 f_{n+2} + \beta_3 f_{n+3} + \beta_4 f_{n+4} + \beta_5 f_{n+5}\right] \qquad (9.3.5)$$

where

$$\beta_0 = \frac{19}{288}\left(1 + \frac{55v^2}{114} + \frac{3282265v^4}{2736} - \frac{5513289335v^6}{114912} + \frac{258450119345v^8}{1378944} + \right.$$

$$\left. \frac{68594391998759v^{10}}{11819520} - \frac{727108666308742877v^{12}}{6949877760} + 0(v^{13})\right) \qquad (9.3.6\text{-i})$$

$$\beta_1 = \frac{75}{288}\left(1 - \frac{11v^2}{30} - \frac{218729v^4}{144} + \frac{4969804253v^6}{90720} - \frac{673623326819v^8}{2721600} - \right.$$

$$\left. \frac{304273318350997v^{10}}{46656000} + \frac{337775919756561 2413v^{12}}{27433728000} + 0(v^{13})\right) \qquad (9.3.6\text{-ii})$$

$$\beta_2 = \frac{50}{288}\left(1 + \frac{11v^2}{30} + \frac{3280991v^4}{720} - \frac{4365325187v^6}{30240} + \frac{1151739351529v^8}{1814400} + \right.$$

$$\left. \frac{276014322723379v^{10}}{15552000} - \frac{2985268375870083433v^{12}}{9144576000} + 0(v^{13})\right) \qquad (9.3.6\text{-iii})$$

$$\beta_3 = \frac{50}{288}\left(1 + \frac{11v^2}{30} - \frac{3281509v^4}{720} + \frac{3699937313v^6}{30240} - \frac{198757705399v^8}{453600} - \right.$$

$$\left. \frac{255272260074121v^{10}}{15552000} + \frac{2503319588144210317v^{12}}{9144576000} + 0(v^{13})\right) \qquad (9.3.6\text{-iv})$$

$$\beta_4 = \frac{75}{288}\left(1 - \frac{11v^2}{30} + \frac{218771v^4}{144} - \frac{2973583247v^6}{90720} + \frac{372098638237v^8}{5443200} + \right.$$

$$\left. \frac{234401042464003v^{10}}{46656000} - \frac{1960667278140426337v^{12}}{27433728000} + 0(v^{13})\right) \qquad (9.3.6\text{-v})$$

$$\beta_5 = \frac{19}{288}\left(1 + \frac{55v^2}{114} - \frac{3280235v^4}{2736} + \frac{2186348165v^6}{114912} + \frac{1641506735v^8}{689472} - \right.$$

$$\left. \frac{41492416771241v^{10}}{11819520} + \frac{279364542028634873v^{12}}{6949877760} + 0(v^{13})\right) \qquad (9.3.6\text{-vi})$$

By taking (9.3.5) for solving oscillatory problems we have

$$y_{n+1} = y_n + h\left[w_1 f_n + w_2 f_{n+\frac{1}{5}} + w_3 f_{n+\frac{2}{5}} + w_4 f_{n+\frac{3}{5}} + w_5 f_{n+\frac{4}{5}} + w_6 f_{n+1}\right] \qquad (9.3.7)$$

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 $k_1 = f_n$, $k_2 = f_{n+\frac{1}{5}}$, $k_3 = f_{n+\frac{2}{5}}$, $k_4 = f_{n+\frac{3}{5}}$, $k_5 = f_{n+\frac{4}{5}}$ and $k_6 = f_{n+1}$ in the form

$$k_1 = f(y_n)$$

$$k_2 = f(y_n + a_1 h k_1)$$

$$k_3 = f\left(y_n + a_2 h k_1 + \left(\tfrac{2}{5} - a_2\right) h k_2\right)$$

$$k_4 = f\left(y_n + a_3 h k_1 + a_4 h k_2 + \left(\tfrac{3}{5} - a_3 - a_4\right) h k_3\right)$$

$$k_5 = f\left(y_n + a_5 h k_1 + a_6 h k_2 + a_7 h k_3 + \left(\tfrac{4}{5} - a_5 - a_6 - a_7\right) h k_4\right)$$

$$k_6 = f\left(y_n + a_8 h k_1 + a_9 h k_2 + a_{10} h k_3 + a_{11} h k_4 + \left(1 - a_8 - a_9 - a_{10} - a_{11}\right) h k_5\right)$$

and $$y_{n+1} = y_n + h\left[w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 + w_5 k_5 + w_6 k_6\right] \tag{9.3.8}$$

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.,

$h^2 f f_y$: 5 - 25*x(1) = 0 (9.3.9-i)

$h^3 f f_y^2$: -76-100*x(1)+38*x(10)+19*x(11)+250*x(1)*x(2)+
100*x(3)+100*x(4)-250*x(1)*x(4)+
225*x(5) + 225*x(6)-375*x(1)*x(6)+75*x(7)+76*x(8) +
76*x(9) - 95*x(1)*x(9) = 0 (9.3.9-ii)

$h^3 f^2 f_{yy}$: 1 - 25*x(1)**2 = 0 (9.3.9-iii)

$h^4 f f_y^3$: -288-300*x(1)+228*x(10)-190*x(1)*x(10)+114*x(11)+
750*x(1)*x(2)+475*x(1)*x(10)*x(2)+
600*x(3)+500*x(1)*x(3)+190*x(11)*x(3)-
1250*x(1)*x(2)*x(3)+600*x(4)-1000*x(1)*x(4)+
190*x(11)*x(4)-475*x(1)*x(11)*x(4)-
1250*x(1)*x(2)*x(4)+735*x(5)-285*x(10)*x(5)-
285*x(11)*x(5)-750*x(3)*x(5)-750*x(4)*x(5)+
1875*x(1)*x(4)*x(5)+735*x(6)-475*x(1)*x(6)-
285*x(10)*x(6)+475*x(1)*x(10)*x(6)-285*x(11)*x(6)+
475*x(1)*x(11)*x(6)-750*x(3)*x(6)-
750*x(4)*x(6)+1875*x(1)*x(4)*x(6)+545*x(7)-
750*x(1)*x(7)-95*x(10)*x(7)-95*x(11)*x(7)+
1875*x(1)*x(2)*x(7)-750*x(3)*x(7)-750*x(4)*x(7)+
1875*x(1)*x(4)*x(7)+228*x(8)-

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)

$h^4 f^2 f_y f_{yy}$: -1124-400*x(1)-500*x(1)**2+608*x(10)+323*x(11)+
1000*x(1)*x(2)+1250*x(1)**2*x(2)+
800*x(3)+800*x(4)-1500*x(1)*x(4)-
1250*x(1)**2*x(4)+2475*x(5)+2475*x(6)-
3000*x(1)*x(6) - 1875*x(1)**2*x(6) + 975*x(7)+
1064*x(8)+1064*x(9)-950*x(1)*x(9)-
475*x(1)**2*x(9) = 0 (9.3.9-v)

$h^4 f^3 f_{yyy}$: 1 - 125*x(1)**3 = 0 (9.3.9-vi)

$h^5 f f_y^4$: -156-1800*x(1)+456*x(10) + 456*x(11) -
570*x(1)*x(11) + 4500*x(1)*x(2) +
1425*x(1)*x(11)*x(2)+760*x(3)+3000*x(1)*x(3)-
760*x(10)*x(3) - 760*x(11)*x(3) +
950*x(1)*x(11)*x(3)-7500*x(1)*x(2)*x(3)-
2375*x(1)*x(11)*x(2)*x(3)+760*x(4)+1100*x(1)*x(4)-
760*x(10)*x(4) + 1900*x(1)*x(10)*x(4) -
760*x(11)*x(4) + 2850*x(1)*x(11)*x(4) -
7500*x(1)*x(2)*x(4)-2375*x(1)*x(11)*x(2)*x(4)+
570*x(5)+2250*x(1)*x(5)-570*x(10)*x(5)-
570*x(11)*x(5)-5625*x(1)*x(2)*x(5)-950*x(3)*x(5)-
3750*x(1)*x(3)*x(5)+950*x(10)*x(3)*x(5)+
950*x(11)*x(3)*x(5)+ 9375*x(1)*x(2)*x(3)*x(5) -
950*x(4)*x(5) - 1375*x(1)*x(4)*x(5) +
950*x(10)*x(4)*x(5)-2375*x(1)*x(10)*x(4)*x(5)+
950*x(11)*x(4)*x(5)-2375*x(1)*x(11)*x(4)*x(5)+
9375*x(1)*x(2)*x(4)*x(5)+570*x(6)+2250*x(1)*x(6)-
570*x(10)*x(6)-570*x(11)*x(6) -
5625*x(1)*x(2)*x(6) - 950*x(3)*x(6) -
3750*x(1)*x(3)*x(6) + 950*x(10)*x(3)*x(6) +
950*x(11)*x(3)*x(6) + 9375*x(1)*x(2)*x(3)*x(6) -
950*x(4)*x(6) - 1375*x(1)*x(4)*x(6) +
950*x(10)*x(4)*x(6)-2375*x(1)*x(10)*x(4)*x(6)+
950*x(11)*x(4)*x(6)-2375*x(1)*x(11)*x(4)*x(6)+
9375*x(1)*x(2)*x(4)*x(6)+570*x(7)+1300*x(1)*x(7)-
570*x(10)*x(7)+950*x(1)*x(10)*x(7) -
570*x(11)*x(7)+950*x(1)*x(11)*x(7)-
3250*x(1)*x(2)*x(7)-2375*x(1)*x(10)*x(2)*x(7) -
2375*x(1)*x(11)*x(2)*x(7)-950*x(3)*x(7)-
3750*x(1)*x(3)*x(7) + 950*x(10)*x(3)*x(7) +
950*x(11)*x(3)*x(7)+9375*x(1)*x(2)*x(3)*x(7)-
950*x(4)*x(7) - 1375*x(1)*x(4)*x(7) +
950*x(10)*x(4)*x(7)-2375*x(1)*x(10)*x(4)*x(7)+
950*x(11)*x(4)*x(7)-2375*x(1)*x(11)*x(4)*x(7)+
9375*x(1)*x(2)*x(4)*x(7)+456*x(8)-760*x(3)*x(8)-
760*x(4)*x(8)+1900*x(1)*x(4)*x(8) -

570*x(5)*x(8)+950*x(3)*x(5)*x(8)+
950*x(4)*x(5)*x(8)-2375*x(1)*x(4)*x(5)*x(8)-
570*x(6)*x(8)+
950*x(3)*x(6)*x(8) + 950*x(4)*x(6)*x(8) -
2375*x(1)*x(4)*x(6)*x(8) - 570*x(7)*x(8) +
950*x(1)*x(7)*x(8)-2375*x(1)*x(2)*x(7)*x(8)+
950*x(3)*x(7)*x(8)+950*x(4)*x(7)*x(8)-
2375*x(1)*x(4)*x(7)*x(8)+456*x(9)-760*x(3)*x(9)-
760*x(4)*x(9)+1900*x(1)*x(4)*x(9) -
570*x(5)*x(9)+950*x(3)*x(5)*x(9)+
950*x(4)*x(5)*x(9)-2375*x(1)*x(4)*x(5)*x(9)-
570*x(6)*x(9)+
950*x(3)*x(6)*x(9)+950*x(4)*x(6)*x(9)-
2375*x(1)*x(4)*x(6)*x(9)-570*x(7)*x(9)+
950*x(1)*x(7)*x(9)-2375*x(1)*x(2)*x(7)*x(9)+
950*x(3)*x(7)*x(9) + 950*x(4)*x(7)*x(9) -
2375*x(1)*x(4)*x(7)*x(9) = 0 (9.3.9-vii)

$h^5 f^2 f_y^2 f_{yy}$: -7988-3000*x(1)-2500*x(1)**2+6308*x(10)-
2660*x(1)*x(10)-950*x(1)**2*x(10)-
380*x(10)**2 + 3496*x(11) - 380*x(10)*x(11) -
95*x(11)**2 + 7500*x(1)*x(2) +
8750*x(1)**2*x(2)+6650*x(1)*x(10)*x(2)+
2375*x(1)**2*x(10)*x(2)-6250*x(1)**2*x(2)**2+
10800*x(3)+5000*x(1)*x(3)+2500*x(1)**2*x(3)+
3420*x(11)*x(3)-12500*x(1)*x(2)*x(3) -
6250*x(1)**2*x(2)*x(3)-1000*x(3)**2+
10800*x(4)-19000*x(1)*x(4)-5000*x(1)**2*x(4)+
3420*x(11)*x(4)-7600*x(1)*x(11)*x(4)-
2375*x(1)**2*x(11)*x(4)-12500*x(1)*x(2)*x(4)-
6250*x(1)**2*x(2)*x(4)-2000*x(3)*x(4)+
5000*x(1)*x(3)*x(4)-1000*x(4)**2+5000*x(1)*x(4)**2-
6250*x(1)**2*x(4)**2+18585*x(5)-
5985*x(10)*x(5)-5985*x(11)*x(5)-12000*x(3)*x(5)-
12000*x(4)*x(5)+26250*x(1)*x(4)*x(5)+
9375*x(1)**2*x(4)*x(5)-3375*x(5)**2+18585*x(6)-
17550*x(1)*x(6)-2375*x(1)**2*x(6) -
5985*x(10)*x(6) + 8550*x(1)*x(10)*x(6) +
2375*x(1)**2*x(10)*x(6)-5985*x(11)*x(6)+
8550*x(1)*x(11)*x(6)+2375*x(1)**2*x(11)*x(6)-
12000*x(3)*x(6)-12000*x(4)*x(6)+
26250*x(1)*x(4)*x(6)+9375*x(1)**2*x(4)*x(6) -
6750*x(5)*x(6) + 11250*x(1)*x(5)*x(6) -
3375*x(6)**2 + 11250*x(1)*x(6)**2 -
9375*x(1)**2*x(6)**2+11185*x(7)-
9000*x(1)*x(7)-3750*x(1)**2*x(7)-2185*x(10)*x(7)-
2185*x(11)*x(7)+22500*x(1)*x(2)*x(7)+
9375*x(1)**2*x(2)*x(7)-12000*x(3)*x(7) -
12000*x(4)*x(7)+26250*x(1)*x(4)*x(7)+
9375*x(1)**2*x(4)*x(7)-2250*x(5)*x(7) -

2250*x(6)*x(7)+3750*x(1)*x(6)*x(7)-
375*x(7)**2+7828*x(8)-1520*x(10)*x(8) -
760*x(11)*x(8)-5985*x(5)*x(8)-5985*x(6)*x(8)+
8550*x(1)*x(6)*x(8)+2375*x(1)**2*x(6)*x(8)-
2185*x(7)*x(8)-1520*x(8)**2+7828*x(9)-
3800*x(1)*x(9) - 1520*x(10)*x(9) +
1900*x(1)*x(10)*x(9)-760*x(11)*x(9)+
950*x(1)*x(11)*x(9)-5985*x(5)*x(9)-5985*x(6)*x(9)+
8550*x(1)*x(6)*x(9)+2375*x(1)**2*x(6)*x(9) -
2185*x(7)*x(9) - 3040*x(8)*x(9) +
3800*x(1)*x(8)*x(9)-1520*x(9)**2 +
3800*x(1)*x(9)**2 - 2375*x(1)**2*x(9)**2 = 0 (9.3.9-viii)

$h^5 f^3 f_{yy}^2$: -328-200*x(1)**2+228*x(10)+133*x(11)+
500*x(1)**2*x(2)+120*x(3)+120*x(4)-
750*x(1)**2*x(4)+540*x(5)+540*x(6)-
1500*x(1)**2*x(6)+300*x(7)+304*x(8)+304*x(9)-
475*x(1)**2*x(9) = 0 (9.3.9-ix)

$h^5 f^3 f_y f_{yyy}$: -6436-1200*x(1)-2500*x(1)**3+3914*x(10)+2128*x(11)+
3000*x(1)*x(2)+
6250*x(1)**3*x(2)+3100*x(3)+3100*x(4)-
6750*x(1)*x(4)-6250*x(1)**3*x(4)+12825*x(5)+
12825*x(6)-18000*x(1)*x(6)-9375*x(1)**3*x(6)+
5025*x(7)+6916*x(8)+6916*x(9)-
7125*x(1)*x(9) - 2375*x(1)**3*x(9) = 0 (9.3.9-x)

$h^5 f^4 f_{yyyy}$: 1 - 625*x(1)**4 = 0 (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 CO5NBF) for solving a system of non-linear equations to give the required parameters, i.e.,

$$a_1 = 0.2000000000 \quad , \quad a_2 = -0.2027706499 \quad , \quad a_3 = 0.5320433054$$
$$a_4 = -0.8706804955 \quad , \quad a_5 = -0.4215000613 \quad , \quad a_6 = 1.3566987431$$
$$a_7 = -0.4405835645 \quad , \quad a_8 = 1.0285379694 \quad , \quad a_9 = -1.7032738709$$
$$a_{10} = 1.1167966402 \quad , \quad a_{11} = -0.1408342610$$

$$a_{31} = \frac{2}{5} - a_2 = 0.6027706499 \quad , \quad a_{41} = \frac{3}{5} - a_3 - a_4 = 0.9386371900$$

$$a_{51} = \frac{4}{5} - a_5 - a_6 - a_7 = 0.3053848827$$

$$a_{61} = 1 - a_8 - a_9 - a_{10} - a_{11} = 0.6987735224$$

$$\dots \qquad (9.3.10)$$

Thus, the new fifth order method for solving oscillatory problems can be written as follows:

$$y_{n+1} = y_n + h\left[w_1k_1 + w_2k_2 + w_3k_3 + w_4k_4 + w_5k_5 + w_6k_6\right] \qquad (9.3.11)$$

where $k_1 = f(y_n)$

$$k_2 = f(y_n + 0.2000000000hk_1)$$

$$k_3 = f(y_n - 0.2027706499hk_1 + 0.6027706499hk_2)$$

$$k_4 = f(y_n + 0.5320433054hk_1 - 0.8706804955hk_2 + 0.9386371900hk_3)$$

$$k_5 = f(y_n - 0.4215000613hk_1 + 1.3566987431hk_2 - 0.4405835645hk_3 + 0.3053848827hk_4)$$

$$k_6 = f(y_n + 1.0285379694hk_1 - 1.7032738709hk_2 + 1.1167966402hk_3 - 0.1408342610hk_4 + 0.6987735224hk_5) \qquad (9.3.12)$$

and

$$w_1 = \frac{19}{288}\left(1 + \frac{55v^2}{114} + \frac{3282265v^4}{2736} - \frac{5513289335v^6}{114912} + \frac{258450119345v^8}{1378944} + \frac{68594391998759v^{10}}{11819520} - \frac{727108666308742877v^{12}}{6949877760} + 0(v^{13})\right)$$

$$w_2 = \frac{75}{288}\left(1 - \frac{11v^2}{30} - \frac{218729v^4}{144} + \frac{4969804253v^6}{90720} - \frac{673623326819v^8}{2721600} - \frac{304273318350997v^{10}}{46656000} + \frac{337775919756561241 3v^{12}}{27433728000} + 0(v^{13})\right)$$

$$w_3 = \frac{50}{288}\left(1 + \frac{11v^2}{30} + \frac{3280991v^4}{720} - \frac{4365325187v^6}{30240} + \frac{1151739351529v^8}{1814400} + \frac{276014322723379v^{10}}{15552000} - \frac{2985268375870083433v^{12}}{9144576000} + 0(v^{13})\right)$$

$$w_4 = \frac{50}{288}\left(1 + \frac{11v^2}{30} - \frac{3281509v^4}{720} + \frac{3699937313v^6}{30240} - \frac{198757705399v^8}{453600} - \frac{255272260074121v^{10}}{15552000} + \frac{2503319588144210317v^{12}}{9144576000} + 0(v^{13})\right)$$

$$w_5 = \frac{75}{288}\left(1 - \frac{11v^2}{30} + \frac{218771v^4}{144} - \frac{2973583247v^6}{90720} + \frac{372098638237v^8}{5443200} + \frac{234401042464003v^{10}}{46656000} - \frac{19606672781404263337v^{12}}{27433728000} + 0(v^{13})\right)$$

$$w_6 = \frac{19}{288}\left(1 + \frac{55v^2}{114} - \frac{3280235v^4}{2736} + \frac{2186348165v^6}{114912} + \frac{1641506735v^8}{689472} - \frac{41492416771241v^{10}}{11819520} + \frac{279364542028634873v^{12}}{6949877760} + 0(v^{13})\right) \quad (9.3.13)$$

By the use of Mathematica to rationalize the parameters in equations (9.3.10), we obtain

$$a_1 = \frac{1}{5}, \quad a_2 = -\frac{122711}{230641}, \quad a_3 = \frac{11797}{23249}, \quad a_4 = -\frac{127014}{145879}$$

$$a_5 = -\frac{24047}{57051}, \quad a_6 = \frac{70916}{52271}, \quad a_7 = -\frac{80844}{183493}, \quad a_8 = \frac{73776}{71729}$$

$$a_9 = -\frac{90994}{53423}, \quad a_{10} = \frac{143065}{128103}, \quad a_{11} = -\frac{20369}{144631}$$

$$a_{31} = \frac{2}{5} - a_2 = \frac{37115}{61574}, \quad a_{41} = \frac{3}{5} - a_3 - a_4 = \frac{64536}{68755}$$

$$a_{51} = \frac{4}{5} - a_5 - a_6 - a_7 = \frac{41303}{135249}$$

$$a_{61} = 1 - a_8 - a_9 - a_{10} - a_{11} = \frac{54809}{78436}$$

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 $k_1 = f(y_n)$

$$k_2 = f\left(y_n + \frac{1}{5}hk_1\right)$$

$$k_3 = f\left(y_n - \frac{11695}{57676}hk_1 + \frac{37115}{61574}hk_2\right)$$

$$k_4 = f\left(y_n + \frac{122711}{230641}hk_1 - \frac{127014}{145879}hk_2 + \frac{64536}{68755}hk_3\right)$$

$$k_5 = f\left(y_n - \frac{24047}{57051}hk_1 + \frac{70916}{52271}hk_2 - \frac{80844}{183493}hk_3 + \frac{41303}{135249}hk_4\right)$$

$$k_6 = f\left(y_n + \frac{73776}{71729}hk_1 - \frac{90994}{53423}hk_2 + \frac{143065}{128103}hk_3 - \frac{20369}{144631}hk_4 + \frac{54809}{78436}hk_5\right) .$$

9.3.1 Error Analysis

By substituting the values $a_i, 1 \le i \le 11$ 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

$$\begin{aligned} LTE = h^6\big[&-0.000204145 f f_y^5 - 0.000938195 f^2 f_y^3 f_{yy} + 0.000176875 f^3 f_y f_{yy}^2 \\ &- 0.0004346 f^3 f_y^2 f_{yyy} - 0.0003058 f^4 f_{yy} f_{yyy} \\ &- 0.000081732 f^4 f_y f_{yyyy} - 4.33681 \times 10^{-19} f^5 f_{yyyyy}\big] \end{aligned} \qquad (9.3.14)$$

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

$$\begin{aligned} k_1 &= \lambda y_n \\ k_2 &= \lambda(y_n + 0.2000000000hk_1) \\ k_3 &= \lambda(y_n - 0.2027706499hk_1 + 0.6027706499hk_2) \\ k_4 &= \lambda(y_n + 0.5320433054hk_1 - 0.8706804955hk_2 + 0.9386371900hk_3) \\ k_5 &= \lambda(y_n - 0.4215000613hk_1 + 1.3566987431hk_2 - 0.4405835645hk_3 \\ &\qquad + 0.3053848827hk_4) \\ k_6 &= \lambda(y_n + 1.0285379694hk_1 - 1.7032738709hk_2 + 1.1167966402hk_3 \\ &\qquad - 0.1408342610hk_4 + 0.6987735224hk_5) \end{aligned} \qquad (9.3.15)$$

By substituting $k_i, 1 \le i \le 6$ in equation (9.3.15) and $w_i, 1 \le i \le 6$ in equation (9.3.13) into the new fifth order formula, i.e.,

$$y_{n+1} = y_n + h\left[w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 + w_5 k_5 + w_6 k_6\right] \qquad (9.3.16)$$

we obtain

$$y_{n+1} = y_n + (h\lambda) y_n + 0.5(h\lambda)^2 y_n + 0.166667(h\lambda)^3 y_n + 0.0416667(h\lambda)^4 y_n$$
$$+ 0.008333333(h\lambda)^5 y_n + 0.00159303(h\lambda)^6 y_n + 0(h^7) \ldots \qquad (9.3.17)$$

By rationalizing the coefficients in equation (9.3.17) we have

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

$$+ \frac{1}{120}(h\lambda)^5 y_n + \frac{15}{9416}(h\lambda)^6 y_n + 0(h^7) \quad \ldots \qquad (9.3.18)$$

By substituting $h\lambda = z$ in (9.3.18), we can show that

$$y_{n+1} = y_n + y_n\left[z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{15}{9416} z^6\right] + 0(z^7) \; . \qquad (9.3.19)$$

Following equation (3.3.30), we write $\frac{y_{n+1}}{y_n} = Q$ in the equation (9.3.19), to obtain

$$Q = 1 + z + \frac{1}{2} z^2 + \frac{1}{6} z^3 + \frac{1}{24} z^4 + \frac{1}{120} z^5 + \frac{15}{9416} z^6 + 0(z^7) \; . \qquad (9.3.20)$$

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

$$\left|1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} + \frac{z^5}{120} + \frac{15}{9416} z^6\right| < 1 \qquad . \qquad (9.3.21)$$

By the use of Mathematica , we can plot the graphic surface defined by equation (9.3.21) i.e.,

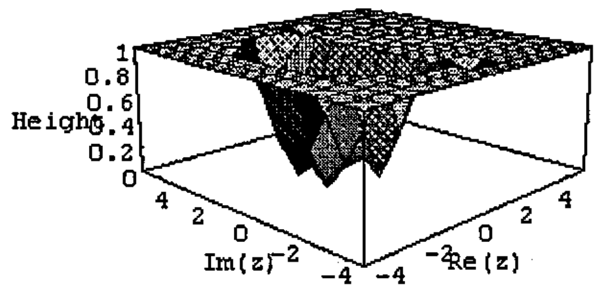


Figure 9.3: Graphic surface defined by new fifth order method

and the stability region defined by the formula in equation (9.3.21) as shown in Figure 9.4

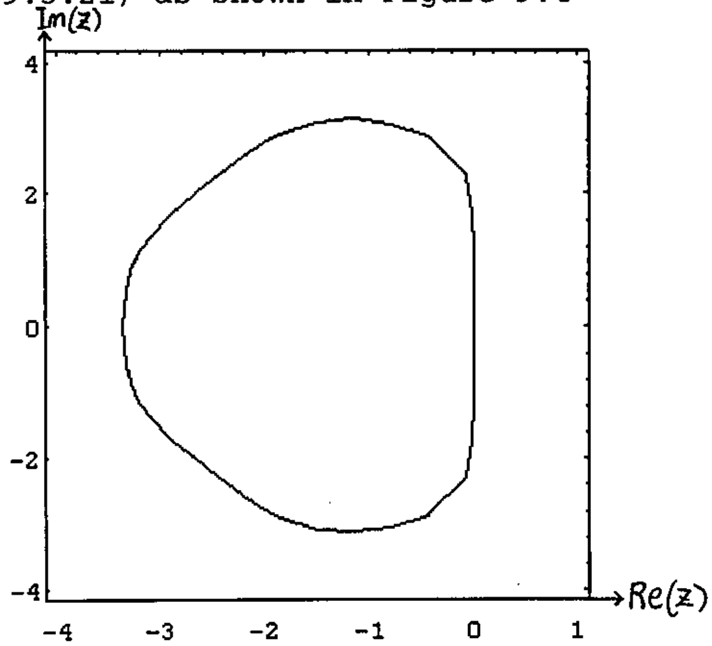


Figure 9.4: Stability region for new fifth order method

9.3.3 Numerical Example

We consider the problem of solving the system of equations

$$y'' + 100(1 - \alpha \cos(2x))y = 0 \quad . \tag{9.3.22}$$

which are oscillatory functions. The oscillatory problem in (9.3.22) can also be written as

$$\frac{d\underline{y}}{dx} = \begin{bmatrix} 0 & 1 \\ -100(1-\alpha\cos(2x)) & 0 \end{bmatrix} \underline{y} \quad , \quad \underline{y}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} . \qquad (9.3.23)$$

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 $\omega = 0$.

x	Fourth Order	Fifth Order	Exact 7 Digit Values
0	1.0000000	1.0000000	1.0000000
0.5	0.0691559	0.0692086	0.0692085
1.0	-0.9084381	-0.9084172	-0.9084179
1.5	-0.6938100	-0.6939603	-0.6939608
2.0	0.2312066	0.2309582	0.2309590
2.5	0.9763441	0.9763679	0.9763699
3.0	0.2053593	0.2057661	0.2057667
3.5	-0.9617184	-0.9616773	-0.9616794
4.0	-0.4260467	-0.4265307	-0.4265317
4.5	0.6026120	0.6022346	0.6022367
5.0	0.9415266	0.9417339	0.9417373

Table 9.2: Fourth order and fifth order method for solving problem (9.3.23) with $\omega = 1$.

x	Fourth Order	Fifth Order	Exact 7 Digit Values
0	1.0000000	1.0000000	1.0000000
0.5	0.0691559	0.0692078	0.0692085
1.0	-0.9084383	-0.9084167	-0.9084179
1.5	-0.6938103	-0.6939573	-0.6939608
2.0	0.2312068	0.2309610	0.2309590
2.5	0.9763449	0.9763651	0.9763699
3.0	0.2053594	0.2057601	0.2057667
3.5	-0.9617195	-0.9616746	-0.9616794
4.0	-0.4260471	-0.4265225	-0.4265317
4.5	0.6026129	0.6022370	0.6022367
5.0	0.9415281	0.9417265	0.9417373

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_oM, C_eM and RMS. The second part is the extension of the concepts and techniques of the C_oM, C_eM 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_oM 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_oM, C_eM 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_oM and C_eM 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_aM method attained nearly fourth order accuracy compared to the other non-linear 2-stage third order methods. For these three methods, C_oM, C_eM 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_oM) gives greater accuracy but for a system of stiff equations in the same class of nonlinear modified Trapezoidal formula, the modified centroidal mean (MC_eM) method performs better.

In Chapter 5, we compare the speedups obtained on a parallel computer by using a second order C_oM 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 then 2 no matter how many processors were used. However, by implementating 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_oM 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_oM, C_eM and H_aM approach of deriving numerical methods for the solution of ODEs. For the second order ODE problems of special type, we have obtained the C_oM, C_eM and H_aM modified form of the Numerov method. The C_oM, C_eM and H_aM 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_oM, C_eM and H_aM 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., $y_{n+1} = y_n + h\left[\sum_{i=1}^{4} w_i\left(\frac{k_i + k_{i+1}}{2}\right)\right]$ where $\sum_{i=1}^{4} w_i = 1$ and $k_i, 1 \le i \le 5$ 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)-Nystrom 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 C_oM. 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

Abromowitz, M. and I.A.Stegun. [1964],Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables, National Bureau of Standard, New York.

Atkinson, L.V., P.J.Harley and J.D.Hudson.[1989], Numerical Methods With Fortran 77 : A Practical Introduction, Addison-Wesley Publishing Company, New York.

Axelsson, O. [1969], A Class of A-Stable Methods, B.I.T., 9, pp. 185-199.

Bahoshy, N.M. [1992],Parallelization Of Algorithms By Explicit Partitioning, Ph.D Thesis, Loughborough University of Technology, UK.

Brown, W.S., and A.C. Hearn. [1978], Application of Symbolic Algebraic Computation, Proceedings Computational Atomic and Molecular Physics Conference, Nottingham, England.

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

Caudill, M. [1990], Neural Networks Application Mathematica V.1.2, AI Expert, Vol.5, pp. 61-65.

Chawla, M.M., and M.A. Al-Zanaidi. [1996], New L-Stable Modified Trapezoidal Formulas for The Numerical Integration of $y' = f(x,y)$, Department of Mathematics and Computer Science, Kuwait University, Kuwait. Submitted to Intern. J. Comp. Maths.

Cloutman, L.D., and L.W. Fullerton. [1977], Some Applications of Automated Heuristic Stability Analysis, Report LA-6885-MS, Los Alamos Scientific Laboratory, Los Alamos, N.M.

Corliss,G.F. [1994], Guaranteed Error Bounds For Ordinary Differential Equations, Lecture Notes at VI-th SERC Numerical Analysis Summer School, Leicester University, 25 - 29 July 1994.

Cryer, C.W. [1973], A New Class of Highly Stable Methods: A_o-Stable Methods, B.I.T., 13, pp. 153-159.

Dahlquist, G. [1963], A Special Stability Problem for Linear Multistep Methods, B.I.T., 3, pp. 27-43.

Ehle, B.L. [1969], On Pade Approximations to the Exponential Function and A-Stable Methods for the Numerical Solution of Initial Value Problems, University of Waterloo, Department of Applied Analysis and Computer Science, Research Report No. CSRR 2010.

Evans,D.J. and M.I.Jayes. [1993],"New Numerical Methods For A Special Class of Second Order ODEs",Intern. J. Computer Math., Vol. 48, pp.191-201.

Evans,D.J. and A.R.Yaakub. [1995], A New Runge-Kutta RK(4,4) Method, Loughborough University of Technology, Department of Computer Studies, Report No.968 and Intern. J. Computer Math., Vol. 58, pp. 169 - 187.

Evans,D.J. and A.R.Yaakub. [1996], A New Fifth Order Weighted Runge-Kutta Formula, Loughborough University of Technology, Department of Computer Studies, Report No. 989 and Intern. J. Computer Math.,Vol. 59, pp.227-243. [Paper Presented at Southeastern Atlantic Regional Conference In Differential Equations,October 13-14,1995, North Carolina State University, Raleigh, NC]

Evans,D.J. and A.R.Yaakub. [1994], Truncation Error and Stability Analysis For the Runge-Kutta C_oM method , Loughborough University of Technology, Department of Computer Studies , Report No.889.

Evans,D.J. and A.R.Yaakub. [1993], A New Fourth Order Runge-Kutta Formula Based On The Contraharmonic (C_0M) Mean, Loughborough University of Technology , Department of Computer Studies, Report No. 849 [Contributed Talks at the VI-th SERC Numerical Analysis Summer School, Leicester University , 25 - 29 July 1994] and in Intern. J. Computer Math., Vol. 57, pp. 249 - 256.

Evans, D.J. and A.R.Yaakub.[1993], A New Third Order Runge-Kutta Method Based On The Root-Mean Square Formula, Loughborough University of Technology, Department of Computer Studies, Report No.857.

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.

Eves , J.H. [1983] , An Introduction To The History of Mathematics , Saunders College Publishing , Philadelphia.

Evans,D.J., and B.B. Sanugi. [1991], A Comparison of Nonlinear Trapezoidal Formulae for Solving Initial Value Problems, Intern.J.Computer Math.,41, pp. 65 - 79.

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 B.B. Sanugi.[1993], Numerical Solution of Initial Value Problems By Nonlinear Trapezoidal Formulae and Rational Extrapolation Method II, Loughborough University of Technology , Department of Computer Studies, Report No. 865.

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.

Evans , D.J., and G.Walz. [1993] , Numerical Solution of Initial Value Problems By Nonlinear Trapezoidal Formulae and Extrapolation Method 1, Loughborough University of Technology, Department of Computer Studies, Report No. 864.

Elden , L . and L. Wittmeyer-Koch. [1990] , Numerical Analysis : An Introduction , Academic Press Limited , London.

England , R. [1969] , Error Estimates For Runge-Kutta Type Solutions To Systems of Ordinary Differential Equations , Comput. J., 12 , pp. 166 - 170.

Fatunla , S.O. [1995] , A Class of Block Methods For Second Order IVP , Intern. J. Computer Math., Vol.55, pp. 119 - 133.

Fehlberg , E. [1969] , Low Order Classical Runge-Kutta Formulas With Step-Size Control and Their Application To Some Heat Transfer Problems, NASA Tech. Report 315 (1969) , Extract Published in Computing , 6 , pp. 66 - 71, (1970).

Flynn, M.J. [1972], Some Computer Organizations and Their Effectiveness, IEEE Trans On Computers , Vol. C-21, No.9, pp. 948-960.

Fox, L. [1962], Numerical Solution of Ordinary and Partial Differential Equations, Pergamon Press, New York.

Freeman, T.L., and C.Phillips. [1992], Parallel Numerical Algorithms, Prentice Hall.

Gautschi , W. [1961] , Numerical Integration of Ordinary Differential Equations Based on Trigonometric Polinomials, Numerische Mathematik , 3, pp. 381- 397.

Gaylord, R.J, Kanin, S.N. and P.R.Wellin. [1996], An Introduction to Programming with Mathematica, Second Edition, Springer-Verlag Publishers, New York.

Gear, C.W. [1971], Numerical Initial Value Problems In Ordinary Differential Equations, Prentice Hal, London.

Gear, C.W. [1987], Parallel Methods for Ordinary Differential Equations, University of Illinois at Urbana-Champaign , Report No. UIUCDCS-R-87-1369.

Gerald, C.F., and P.O. Wheatley. [1994], Applied Numerical Analysis, Fifth Edition, Addison-Wesley Publishing Companay, New York.

Gladwell, I., and D.K. Sayers. [1980], Computational Techniques for Ordinary Differential Equations, Academic Press, London.

Gladwell , I, Shampine, L.F. and R.W. Brankin. [1985] ,"Automatic Selection of the Initial Step Size for ODE Solver, University of Manchester, Department of Mathematics , Report No. 104.

Gragg, W.B. [1965], On Extrapolation Algorithms for Ordinary Initial Value Problems, SINUM. 2, pp. 384-403.

Greenspan,D., and V.Casulli. [1988], Numerical Analysis ForApplied Mathematics, Science, and Engineering , Addison-Wesley Publishing Company, Inc. United Kingdom.

Hall , G. [1994] ,"A New Stepsize Strategy for Runge-Kutta Codes", University of Manchester/UMIST, Manchester Centre for Computational Mathematics, Report No. 245.

Hammer, P.C., and J.W. Hollingsworth. [1955], Trapezoidal Methods of Approximating Solutions of Differential Equations, M.T.A.C, 9, pp. 92-96.

Henrici, P. [1962], Discrete Variable Methods In Ordinary Differential Equations, John Wiley and Sons, New York.

Hindmarsh, A.C. [1974], GEAR: Ordinary Differential Equation Solver, Report UCIO - 30001, Revision 3, Lawrence Livermore Laboratory.

Hwang, K., and F.A. Briggs. [1984], Computer Architecture and Parallel Processing, McGraw-Hill.

Jamieson, L.H., Gannon, D. and R.J.Douglass. [1987], The Characteristic of Parallel Algorithms , The MIT Press , Cambridge.

Jayes, M.I., and D.J. Evans. [1991], A New Adaptive Runge-Kutta Method, Loughborough University of Technology,Department of Computer Studies, Report No. 592.

Jayes, M.I. [1992] , Numerical Solution of Ordinary and Partial Diffrential Equations Occuring in Scientific Applications , Ph.D Thesis, Loughborough University of Technology , UK.

Jeltsch, R.[1978] , Complete Characterization Of Multistep Methods With Interval Of Periodicity For Solving $y'' = f(x, y)$, Math. Comp., 32, pp. 1108-1114.

Jenson, J., and F.Niordson. [1977], Symbolic and Algebraic Manipulation Languages and Their Application in Mechanics, in Structural Mechanics Software Series, 1, pp. 541-576, University Press of Virginia.

Kernighan, B.W., and D.M. Ritchie. [1988], The C Programming Language, Prentice-Hall.

Khalaf, B.M.S. [1990], Parallel Numerical Algorithms For Solving Ordinary Differential Equations, PhD Thesis, University of Leed, UK.

Kohn, M.C. [1986] ,Practical Numerical Methods : Algorithms and Programs, John Wiley & Sons , London.

Kuck, D.J. [1968] , ILLIAC IV : Software and Application Programming, IEEE Transactions On Computer 17, No.8, pp. 758-770.

Lambert, J.D. [1973], Computational Methods in Ordinary Differential Equations, John Wiley & Sons , London.

Lambert, J.D. [1991], Numerical Methods For Ordinary Differential Systems, John Wiley & Sons , London.

Lewis, T.G., and H. El-Rewini. [1992], Introduction to Parallel Computing, Prentice Hall.

Lotkin , M. [1951] , On The Accuracy of RK methods , MTAC , 5 , pp. 128-132.

Maeder , R. [1990] , Programming In Mathematica , First Edition, Addison-Wesley Publishing, United Kingdom.

Mathews , J.H. [1992], Numerical Methods For Mathematics, Science and Engineering , Prentice - Hall International Editions , London .

Merson , R.H. [1957] , An Operational Method For The Study of Integrating Processes , Proc. Symp. Data Processing , Weapon Research Establishment, Salisbury , S . Australia.

Noye , J. [1984] , Computational Techniques for Differential Equations, Elsever Science Publishes , New York.

Osterhaug, A.[1989], Guide To Parallel Programming On Sequent Computer System, Second Edition, Prentice Hall , London.

Ralston , A. [1962] , Runge - Kutta Methods With Minimum Error Bounds , Math. Comput. , 16 , pp. 431 - 437.

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 Differential Equation , Ph.D. Thesis, Loughborough University of Technology. UK.

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, Part 1 , Mathematical Software 3, Academic Press, pp. 257-275.

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, Math. Comp. Vol. 20 , pp. 21 - 38.

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 A Variety of Means, Intern. J . Computer Math, 50, pp. 105-112.

Widlund, O.B. [1967], A Note on Unconditionally Stable Linear Multistep Methods, B.I.T., 7, pp. 65-70.

Wolfram, S. [1991], Mathematica : A System For Doing Mathematics By Computer, Second Edition. Addison-Wesley Publishing Company.

Yaakub, A.R. [1988], A Survey of Computational and Implementation of Self Generated Code For Solving the Initial Value Problem. M.Sc. Thesis, Universiti Teknologi Malaysia.

Yaakub, A.R. and D.J.Evans. [1993], A New Fourth Order Runge-Kutta Method Based On The Root-Mean-Square Formula, Loughborough University of Technology, Department of Computer Studies, Report No.862.

Yaakub, A.R. and D.J.Evans. [1995],The Parallel Implementation Of The Runge-Kutta Contraharmonic Mean Method With Extrapolation, Parallel Algorithms and Application, Vol. 6, pp. 111 - 128.

Yaakub, A.R. and D.J.Evans. [1995], A Fourth Order Runge-Kutta RK(4,4) Method With Error Control , Loughborough University of Technology, Department of Computer Studies , Report No.977.

Yaakub, A.R. and Evans, D.J. [1995], A New Fourth Order Runge-Kutta Method Based on The Centroidal Mean Formula, Paper Presented at Northern England Universities Numerical Analysis Colloquium ,November 29,1995. University of Liverpool, UK.

Yaakub, A.R. and D.J.Evans. [1996], New Runge-Kutta Starters For Multistep Methods, Loughborough University of Technology, Department of Computer Studies, Report No.1008.

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<parallel/parallel.h>
#include<parallel/microtask.h>
#define m 3
        shared double tableau[100][100], H;
        double exact, yend;
        int     si, time1, time2;
        shared int   x0,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(intergers only) x0,xf,nprocs \n");
        scanf("%d %d %d", &x0,&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);
        print_table();
        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;
        htam[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]+htam[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.-3.*h[j]*k2[i]/4.+3.*h[j]*k3[i]/2.;
        }
         func3(zz,k4);
        for (i=1; i<=m ; i++) {
        yy[i] +=(h[j]/3.)*(((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[%f] = %f\n",j,h[j] ,xh[j], tableau[1][j]);
        }
                printf("\n");
        return;
}

/* --------------- end of find_func_values() -------------------------------------------- */
```

```
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][k] = tableau[i-1][k+1] + (tableau[i-1][k+1]-tableau[i-1][k])/fp;
        if (fabs(tableau[i][1]-tableau[i-1][1]) <= 0.0000000001 ) break;
                yend=tableau[i-1][1];
                        si=i-1;
        }
        return;
}

/* ---------------- end of tabulate() ---------------------------------------------------------*/

void    print_table()
{
int     i, j, k;

        for (i=1, j=si; j>0; i++, j--) {          /* for each row of tableau */
                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("Error = %E\n",fabs(exact-yend));
                printf("\n");
        return;
}

/*------------- defining the function------------------------------------------------------------*/

void func3(yy,F)
double yy[ ], F[ ];
{
        F[1] = 1.0;
        F[2] = yy[2] - yy[3] + yy[1] * (2.0 - yy[1] * (1.0 + yy[1]));
        F[3] = yy[2] + yy[3] - yy[1] * yy[1] * (4.0 - yy[1]);
        return;
}
```

APPENDIX 2

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)
      ak2=f(npb,x+a1*h,y+h*a1*ak1)
      ak3=f(npb,x+(a2+a3)*h,y+h*a2*ak1+h*a3*ak2)
      ak4=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
      cm1=f(npb,x,y1)
      cm2=f(npb,x+c1*h,y1+h*c1*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))/3.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
```

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 1TO 6'
1     read*,num
      call problem(num,x0,y0,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(1x,'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*f(num,xn,yn)
   ~   + a2*f(num,xn0,yn0)+a3*f(num,xn1,yn1)
   ~   + a4*((f(num,xn,yn)*f(num,xn,yn)+f(num,xn1,yn1)*f(num,xn1,yn1))
   ~   /(f(num,xn,yn)+f(num,xn1,yn1)))
   ~   + a5*((f(num,xn0,yn0)*f(num,xn0,yn0)+f(num,xn,yn)*f(num,xn,yn))
   ~   /(f(num,xn0,yn0)+f(num,xn,yn)))
```

```
     ~     + a6*((f(num,xn1,yn1)*f(num,xn1,yn1)+f(num,xn0,yn0)*f(num,xn0,yn0))
     ~      /(f(num,xn1,yn1)+f(num,xn0,yn0))))
c-------compute the exact solution of the problem
        exct = exact(num,xn)
c--------compute the absolute difference between exact and computed solutions
     if(exct.ne.0)then
        err = abs(exct-yn)/abs(exct)
     else
        err = abs(exct-yn)
     endif
     write(*,100)xn,yn,exct,err
c--------reset appropriate values of xn0,xn,xn1,yn0,yn
        xn0 = xn
        xn = xn1
        yn0 = yn
        yn = yn1
100  format(f7.4,3e23.12)
10   continue
9999 continue
        stop
        end
c
     subroutine predic(h,xn,yn1,yn,yn0)
c
        implicit double precision (a-h,o-z)
        yn1 = 2.*yn-yn0+h**2.*f(num,xn,yn)
        return
        end
c
     subroutine problem(num,x0,y0,xend,nsteps)
c
        implicit double precision (a-h,o-z)
        common/blk1/b,c,q
c
     if(num.eq.1)then
c
c-----PROBLEM:1   Y'' + X*Y = 0
c-----INITIAL CONDITIONS X0=0,Y0=1,Y'=2
c-----EXACT SOLUTION Y=(1 - X**3/3 + X**6/180 - ...)
c-----+ 2*(X - X**4/12 + X**7/504 - ...)
c-----CHOOSE SOLUTION DOMAIN [0,1]
       write(*,*)' PROBLEM:1   Y'''' + X*Y = 0'
       write(*,*)'  INITIAL CONDITIONS X0=0,Y0=1,Y''=2'
       write(*,*)'  EXACT SOLUTION Y=(1 - X**3/3 + X**6/180 - ...)'
       write(*,*)'          + 2*(X - X**4/12 + X**7/504 - ...)'
       write(*,*)'     CHOOSE SOLUTION DOMAIN [0,1]'
       write(*,*)'INPUT VALUES OF x0 y0  xend nsteps'
       read(*,*)x0,y0,xend,nsteps
       return
```

```
c

c
   elseif(num.eq.2)then
c
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]'
      write(*,*)'INPUT VALUES OF x0 y0 xend nsteps'
      read*,x0,y0,xend,nsteps
      return
c
   elseif(num.eq.3)then
c
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
c
   elseif(num.eq.4)then
c
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
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
c-----PROBLEM:6  Y'' - Y*(((Q + B*X)/X)**2 - Q/(X**2)) = 0
c-----INITIAL CONDITIONS X0=1,Y0=10*e,Y'0=10*e*(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=10*e,Y''0=10*e*(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
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
c
c-----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.)
    1   + x**2/2. + x**3/6. + x**4/12. - x**6/60. - x**7/252. - x**8/672.
        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 := Abs[((((z+Sqrt[z^2+4])/2)^2)
            /.z->(x+I y))]
gm1:= Plot3D[gm,{x,-4,4},{y,-4,4},
        AxesLabel ->{"Re(z)","Im(z)","Height"},
        ViewPoint ->{-2,-2,0.5},
        PlotRange ->{0,1.2}]
gm2:=ContourPlot[gm,{x,-2.5,3},{y,-3,3},ContourShading->False,
        Contours->{1,1}]
co := Abs[(Sqrt[((2/(1-z))-1)]
            /.z->(x+I y))]
co1:= Plot3D[co,{x,-4,4},{y,-4,4},
        AxesLabel ->{"Re(z)","Im(z)","Height"},
        ViewPoint ->{-2,-2,0.5},
        PlotRange ->{0,1.2}]
co2:=ContourPlot[co,{x,-2.5,3},{y,-3,3},ContourShading->False,
        Contours->{1,1}]
ce:= Abs[((-1+(3/(3-2 z))+((2 Sqrt[3]) (Sqrt[3-z^2])/(-3+2 z))/2)
        /.z->(x+I y))]
ce1:= Plot3D[ce,{x,-4,4},{y,-4,4},
        AxesLabel ->{"Re(z)","Im(z)","Height"},
        ViewPoint ->{-2,-2,0.5},
        PlotRange ->{0,1.2}]
ce2:=ContourPlot[ce,{x,-5.5,4},{y,-4,4},ContourShading->False,
        Contours->{1,1}]
ha := Abs[((z+Sqrt[z^2+1])
            /.z->(x+I y))]
ha1:= Plot3D[ha,{x,-4,4},{y,-4,4},
        AxesLabel ->{"Re(z)","Im(z)","Height"},
        ViewPoint ->{-2,-2,0.5},
        PlotRange ->{0,1.2}]
ha2:=ContourPlot[ha,{x,-2.5,3},{y,-3,3},ContourShading->False,
        Contours->{1,1}]
f:= Abs[((1+z+(1/2) z^2+(1/6) z^3+(1/24) z^4)
            /.z->(x+I y))]
r1:= Plot3D[f,{x,-4,4},{y,-4,4},
        AxesLabel ->{"Re(z)","Im(z)","Height"},
        ViewPoint ->{-2,-2,0.5},
        PlotRange ->{0,1.2}]
r2:=ContourPlot[f,{x,-4,4},{y,-4,4},ContourShading->False,
        Contours->{1,1}]
s1:=Show[GraphicsArray[{{gm1,co1,ce1,ha1}, {gm2,co2,ce2,ha2}}]]
s2:=Show[%, Frame->True, 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
20      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) 'ifail =', 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
        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     1.0000000000
*********************************************************
```

APPENDIX 6

Program System For Second Order ODEs

```
$INCLUDE probl.f
        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 probleml
        write(*,*)'Enter the value of x0,y(X0),V(X0) and xend'
        read(*,*)x0,y0,v0,xend
        xn = x0
        yn = y0
        vn = v0
        w1=0.2615038147351447
        w2=-0.2765809214083533
        w3=0.5947141647174489
        w4=0.4203629419557595
        a1=1.5471214402823019
        a2=0.1756458393315915
        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(5x,'xn ',11x,' 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))
```

```
c
        xn1=xn+h
        xn = xn1
        yn = yn1
        vn = vn1
        exct = exact(npb,xn)
          err = dabs(exct - yn)
        write(*,100)xn,yn,exct,err
100     format(f7.4,3e23.10)
c write and format for problem no.1 where x0=0,y0=1,vn=0
c       write(*,100)xn,yn,vn
c100    format(f7.4,3e23.10)
        if(xn.le.xend)goto 200
        stop
        end
c
c               SUBROUTINE PROBLEM1
c
      subroutine problem1
      implicit double precision (a-h,o-z)
      common/blk1/b,c,q
      common/blk3/x0,xend,y0,v0,npb
      pi=22.d0/7.
      if(npb.eq.1)then
c
c     For npb=1 , see Scraton,R.E [1986], pp 77
c     PROBLEM:1  Y'' + 3*Y'+2*Y = 2*EXP(-3X)
c     INITIAL CONDITIONS X0=0,Y(X0)=1,Y'(X0)=V(X0)=-2
c     EXACT SOLUTION Y=EXP(-X)-EXP(-2*X)+EXP(-3*X)
c     CHOOSE SOLUTION DOMAIN [0,1]
c
      write(*,11)
11    format(7x,'FOR PROBLEM:1, SEE SCRATON,R.E [1986], PP 77'
     1/10x,'PROBLEM:1  Y'''' + 3*Y''+2*y = 2*exp(-3*x)'/10x,
     2' INITIAL CONDITIONS X0=0,Y(X0)=1,Y''(X0)=V(X0)=-2'
     3/12x,'  EXACT SOLUTION Y=EXP(-X)-EXP(-2*X)+EXP(-3*X)'
     4/10x,'      CHOOSE SOLUTION DOMAIN [0,1]'/)
      return
c
      elseif(npb.eq.2)then
c
c-----For npb=2 , see Greenspan,D and V.Casulli [1988], pp 139
c-----PROBLEM:2   Y'' + (Y')**3 - 8*X*Y = 0
c-----INITIAL CONDITIONS X0=0,Y(X0)=0,Y'(X0)=0
c-----EXACT SOLUTION Y=X**2
c-----CHOOSE SOLUTION DOMAIN [0,1]
c
```

```
   write(*,12)
12  format(7x,'FOR PROBLEM:2,SEE GREENSPAN,D ANDV.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
c
       else
       stop
       endif
       end
c
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
c
c-----SUBROUTINE OF THE EXACT SOLUTION
c
    double precision function exact(npb,x)
    implicit double precision(a-h,o-z)
    common/blk1/b,c,q
    if(npb.eq.1)then
        exact = exp(-x)-exp(-2*x)+exp(-3*x)
    return
    elseif(npb.eq.2)then
      exact = x**2
    return
     endif
    end
```

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(0:100)
     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(0)= 0
      y(0)= 1.do
        x0 = 0
        xend = 1
     n= 100
     h= (xend - x0)/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) = x0 + 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,e10.5,1x,3e18.7)
      print 12,jj,h
12    format(/,1x,'RESULTS ARE PRINTED FOR EVERY',I4,1x,'STEPS:h=',f6.4)
      stop
      end
c
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)
       exact = 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 - (v^3/6) + (v^5/120)-(v^7/5040)+(v^9/362880)
cos[v_]:= 1 - (v^2/2) + (v^4/24)-(v^6/720)+(v^8/40320)

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
        -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
        -3 v sin[15v] b5 - cos[15v] + 1 ==0,
        3 v b0 + 3 v cos[3v] b1 + 3 v cos[6v] b2 + 3v cos[9v] b3 + 3 v cos[12v] b4
        + 3 v cos[15v] b5 - sin[15v] == 0},
            {b0,b1,b2,b3,b4,b5}])
```

APPENDIX 9

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=0.8-a5-a6-a7
      a61=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.)-
      (5513289335.*v**6 /114912.)+(258450119345.*v**8/1378944.)+
      (68594391998759.*v**10/11819520.)-
      (727108666308742877.*v**12/6949877760.)+
      (910220251463182217.*v**14/4389396480.))
      w2=(75./288.)*(1-(11.*v**2/30.)-
      (218729.*v**4/144.)+(4969804253.*v**6/90720.)-
      (673623326019.*v**8/2721600.)-
      (304273318350997.*v**10/46656000.)+
      (3377759197565612413.*v**12/27433728000.)-
      (402311025659343378 7.*v**14/13168189440.))
      w3=(50./288.)*(1+(11.*v**2/30.)+(3280991.*v**4/720.)-
      (4365325187.*v**6/30240.)+(1151739351529.*v**8/1814400.)+
      (276014322723379.*v**10/15552000.)-
      (2985268375870083433.*v**12/9144576000.)+
      (8115490920303284377 9.*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.)+
```

```
     (25033195881442103l7.*v**12/9144576000.)-
     (391872106451530l0971.*v**14/109734912000.))
     w5=(75./288.)*(1-(11.*v**2/30.)+(218771.*v**4/144.)-
     (2973583247.*v**6/90720.)+(372098638237.*v**8/5443200.)+
     (234401042464003*v**10/46656000.)-
     (1960667278140426337.*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.)+
     (279364542028634873.*v**12/6949877760.)+
     (11205900105824818873.*v**14/83398533120.))
     Write(*,7)
7    format(5x,'xn ',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,vn+a1*h*aM1)
     aM2=g(xn+a1*h,yn+a1*h*ak1,vn+a1*h*aM1)
    ak3=f(xn+(a2+a31)*h,yn+a2*h*ak1+a31*h*ak2,vn+a2*h*aM1+a31*h*aM2)
     aM3=g(xn+(a2+a31)*h,yn+a2*h*ak1+a31*h*ak2,
            vn+a2*h*aM1+a31*h*aM2)
     ak4=f(xn+(a3+a4+a41)*h,yn+a3*h*ak1+a4*h*ak2+a41*h*ak3,
            vn+a3*h*aM1+a4*h*aM2+a41*h*aM3)
     aM4=g(xn+(a3+a4+a41)*h,yn+a3*h*ak1+a4*h*ak2+a41*h*ak3,
            vn+a3*h*aM1+a4*h*aM2+a41*h*aM3)
     ak5=f(xn+(a5+a6+a7+a51)*h,yn+a5*h*ak1+a6*h*ak2+
            a7*h*ak3+a51*h*ak,vn+a5*h*aM1+a6*h*aM2+
            a7*h*aM3+a51*h*aM4)
     aM5=g(xn+(a5+a6+a7+a51)*h,yn+a5*h*ak1+a6*h*ak2+
            a7*h*ak3+a51*h*a4,vn+a5*h*aM1+a6*h*aM2+
            a7*h*aM3+a51*h*aM4)
     ak6=f(xn+(a8+a9+a10+a11+a61)*h,yn+a8*h*ak1+
            a9*h*ak2+a10*h*ak3+a11*h*ak4
            +a61*h*ak5,vn+a8*h*aM1+a9*h*aM2+a10*h*aM3+
            a11*h*aM4+a61*h*aM5)
     aM6=g(xn+(a8+a9+a10+a11+a61)*h,yn+a8*h*ak1+a9*h*ak2+
            a10*h*ak3+a11*h*ak4
            +a61*h*ak5,vn+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
     exct = exact(xn)
       err = (exct - yn)
     excta = exacta(xn)
```

```
      erra = (excta - vn)
c     write(*,100)xn,yn,exct,err
      write(*,100)xn,excta,err,erra
100   format(f7.4,3e23.5)
      if(xn.le.xend)goto 200
      stop
      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)
       exact = dcos(x)
      return
      end
c
      double precision function exacta(x,y)
      implicit double precision (a-h,o-z)
       exacta = dsin(x)
      return
      end
```