
This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Numerical solution of ordinary and partial differential equations occurring in scientific applications

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

© Mohd. Idris Jayes

PUBLISHER STATEMENT

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

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Jayes, Mohd. Idris. 2019. "Numerical Solution of Ordinary and Partial Differential Equations Occurring in Scientific Applications". figshare. <https://hdl.handle.net/2134/32103>.

BLDSC no:- DX 174570

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR/FILING TITLE

JAYES, M. I.

ACCESSION/COPY NO.

040060474

VOL. NO.

CLASS MARK

- 1 JUL 1994

Loan copy

30 JUN 1995

28 JUN 1996

0400604744



BADMINTON PRESS
18 THE HALFCROFT
SYSTON
LEICESTER LE7 8LD
ENGLAND
TEL: 0533 602917
FAX: 0533 696636

NUMERICAL SOLUTION
OF ORDINARY AND PARTIAL DIFFERENTIAL
EQUATIONS OCCURRING IN SCIENTIFIC
APPLICATIONS

BY

MOHD IDRIS JAYES M.Sc.

A Doctoral Thesis
submitted in partial fulfilment of the requirements
for the award of Doctor of Philosophy
of the Loughborough University of Technology
August, 1992.

Director of Research
PROFESSOR D.J.EVANS, D.Sc.

Supervisor
DR. A.BENSON, Ph.D.

Department of Computer Studies

© by Mohd Idris Jayes, 1992.

LONG BEACH UNIVERSITY
LIBRARY
Dec 92
036000 0400 60474

W9919957

DECLARATION

I declare that the following thesis is a record of research work carried out by me and that the thesis is of my own composition. I also certify that neither this thesis nor the original work contained therein has been submitted to this or any other institution for a degree.

M. I. JAYES

ACKNOWLEDGEMENTS

All praise be unto ALLAH, the God and only one for all mankind. All the blessings unto HIS Prophets from Adam to Muhammad, peace be unto them all without any distinction and their true followers from the day of creation to the Last Hour.

I would like to take this opportunity to thank my Director of research Professor D.J.Evans and supervisor Dr.A.Benson. For their unfailing guidance and warm sense of humour all throughout the work, nothing within my capacity could ever repay them except to wish them the best of life.

To everyone who has helped me in a way or other, I wish to extend my sincere appreciation and best wishes.

ABSTRACT

This thesis is concerned with the numerical solutions of initial value problems with ordinary differential equations and the boundary value problems involving partial differential equations.

Chapter 1 is an introductory chapter on the initial value and boundary value problems in ordinary and partial differential equations. This is then followed by a chapter on the basic mathematical preliminaries and fundamental concepts of Numerical Analysis which are applied in the thesis. A survey of the current numerical algorithms for solving the initial value problems in ordinary differential equations by the step by step marching methods and the boundary value problems derived from elliptic partial differential equations by solving the large sets of linear equations which occur when the partial differential equation is discretized by the finite difference method is described in chapter 3. The discussion on the advantages and disadvantages of several strategies in terms of stability and truncation error is also considered.

Chapter 3 further discusses the partial differential equations solvers, mainly on the boundary value problems involving elliptic partial differential equations. Discretization of the problems leads to solving very large sparsely structured systems of linear equations. Direct and iterative methods of solution are considered such as the LU decomposition, Gaussian Elimination, Cyclic or Odd/Even Reduction, and the Jacobi, Gauss-Seidel and SOR methods.

Chapters 4 and 5 investigate the ordinary differential equations solvers based on the Geometric Mean (GM) strategy. Chapter 4 concentrates on the single-step method application of the GM strategy, mainly on the

modified GM Runge-Kutta (RK-GM) method together with an adaptive strategy for the RK-GM method. Chapter 5 deals with the multistep (specifically, the two-step) method application of the GM strategy. A modified GM Numerov method is also derived.

Chapter 6 concentrates on the numerical solution of elliptic partial differential equations. In this special situation the spectral decomposition method can be efficiently utilised. The direct methods are now extended to block odd/even cyclic reduction and the block tri-reduction algorithm introduced. Then a new direct method, utilising the 'stride of 3' algorithm is devised and the Buneman modified version is also proposed and shown to be stable. Finally, the iterative solvers are considered and an optimum relaxation parameter for the SLOR technique for solving iteratively the system of equations obtained by the discretization of the periodic boundary value problems involving elliptic partial differential equations is also derived.

The final chapter contains the conclusions and recommendations for further research.

CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
CHAPTER 1 INTRODUCTION	1
1.1 DIFFERENTIAL EQUATIONS.....	1
1.2 NUMERICAL SOLUTIONS OF ODES AND PDES.....	2
CHAPTER 2 BASIC PRELIMINARIES AND FUNDAMENTALS OF NUMERICAL METHODS	6
2.1 INTRODUCTION.....	6
2.2 BASIC PRELIMINARIES.....	6
2.2.1 MEANS	6
2.2.2 POWER SERIES	8
2.2.3 SYMBOLIC COMPUTATION	9
2.3 FUNDAMENTALS OF NUMERICAL METHODS.....	20
2.3.1 VECTOR AND MATRIX	20
2.3.2 VECTOR AND MATRIX NORMS	34
2.3.3 CONVERGENCE AND PERTURBATION THEOREMS	37
CHAPTER 3 SURVEYS OF ODE AND PDE SOLVERS ..	40
3.1 INTRODUCTION.....	40
3.2 ODE SOLVERS.....	40
3.2.1 BASIC DEFINITIONS AND THEOREMS	40
3.2.2 SINGLE-STEP AND RK METHODS	45
3.2.3 STABILITY ANALYSIS FOR EXPLICIT RK METHODS	51
3.2.4 LINEAR MULTISTEP METHODS FOR THE SPECIAL CLASS OF ODE PROBLEMS	54
3.2.5 GENERAL OPERATORS FOR SPECIAL SECOND ORDER EQUATIONS	54
3.2.6 BASIC PROPERTIES OF LINEAR MULTISTEP METHODS ...	57
3.2.7 STÖRMER-COWELL METHODS.....	61
3.2.8 BOUNDS FOR THE LOCAL AND GLOBAL TRUNCATION ERRORS OF METHODS (3.2.5-1)	62
3.2.9 ABSOLUTE AND RELATIVE STABILITY OF LINEAR MULTISTEP METHODS	62

3.3 ELLIPTIC PDE: SOLVERS.....	65
3.3.1 CLASSIFICATION OF PDES AND TYPES OF ELLIPTIC PROBLEMS	65
3.3.2 DISCRETIZATION OF THE ELLIPTIC BOUNDARY VALUE PROBLEMS	68
3.3.3 GRID POINTS NEAR THE BONDARY AND GENERAL BOUNDARY CONDITIONS	77
3.3.4 DISCRETIZATION ERRORS	81
3.3.5 METHODS OF SOLUTION OF $Mu = s$	90
3.3.6 DIRECT METHODS OF SOLVING (3.3.2-23)	90
3.3.7 BASIC ITERATIVE METHODS FOR LINEAR EQUATIONS	106
3.3.8 CONVERGENCE OF ITERATIVE METHODS	111
3.3.9 THE OPTIMUM SOR PARAMETER	115

**CHAPTER 4 NUMERICAL SOLUTION OF PROBLEMS
INVOLVING ODES BY USING THE GM SINGLE**

STEP METHODS.....	122
4.1 INTRODUCTION.....	122
4.2 DERIVATION OF COMPOSITE SINGLE STEP GM METHODS.....	123
4.2.1 ACCURACY AND STABILITY ANALYSIS OF EQUATION (4.2-20)	127
4.2.2 NUMERICAL RESULTS FROM USING (4.2.1-5)	129
4.2.3 STABILITY ANALYSIS OF (4.2-20)	132
4.3 DERIVATION OF COMPOSITE RK-GM METHODS.....	136
4.3.1 SECOND ORDER RK-GM METHOD	137
4.3.1.1 ERROR ANALYSIS FOR THE SECOND ORDER METHODS	142
4.3.1.2 NUMERICAL RESULTS	146
4.3.2 THIRD ORDER RK-GM METHOD	147
4.3.2.1 ERROR ANALYSIS OF THE THIRD ORDER RK-GM METHOD	152
4.3.2.2 NUMERICAL RESULTS	152
4.3.3 FOURTH ORDER METHOD	153
4.3.3.1 ERROR ANALYSIS OF (4.3.3-12)	156
4.3.3.2 NUMERICAL RESULTS	158
4.3.4 STABILITY ANALYSIS FOR THE RK-GM METHODS	159
4.3.5 OPTIMAL EXPLICIT TWO-STAGE RK PROCESS	165
4.4 ERROR CONTROL AND ADAPTIVE METHODS.....	168
4.4.1 ERROR ESTIMATION FOR RK PROCESSES	168

4.4.2	ADAPTIVE ERROR CONTROL STRATEGY	172
4.4.3	ERROR CONTROL AND STEP SIZE SELECTION IN THE AM-GM METHOD	176
4.4.4	PRACTICAL ERROR CONTROL	178
4.4.5	MATRIX REPRESENTATION OF THE RK PROCESSES	181
4.4.6	DISCUSSION OF THE IMSL, NAG AND RKF45 ERROR CONTROL STRATEGIES	184
4.4.7	EXPERIMENTAL RESULTS	186
4.5	RK-GM METHOD FOR SYSTEM OF ODES.....	208
4.5.1	NUMERICAL RESULTS FOR SYSTEMS	214
4.6	CONCLUSIONS AND RECOMMENDATIONS.....	215

**CHAPTER 5 NUMERICAL SOLUTION OF PROBLEMS
INVOLVING ODES - GM MULTISTEP METHODS 217**

5.1	INTRODUCTION.....	217
5.2	NUMERICAL METHODS FOR FIRST ORDER ODES.....	218
5.2.1	CONDITIONS UNDER WHICH THE GM FORMULA IS MORE ACCURATE THAN THE AM FORMULA	222
5.2.2	STABILITY ANALYSIS OF (5.2-3)	224
5.3	NUMERICAL METHODS FOR A SPECIAL CLASS OF SECOND ORDER ODES	226
5.3.1	DERIVATION OF THE GM METHOD FOR PROBLEMS OF THE TYPE $y^{(2)} = f(x, y)$	227
5.3.2	ERROR ANALYSIS OF (5.3-9a)	232
5.3.3	NUMERICAL RESULTS	236
5.3.4	DERIVATION OF THE NEW GM STRATEGY WITH ERROR CONTROL	241
5.3.5	NUMERICAL RESULTS	243
5.4	IMPLICIT FORMULA FOR A SPECIAL CLASS OF FOURTH ORDER ODES.	246
5.4.1	VARIANTS OF GM4 FORMULA	256
5.4.2	ERROR ANALYSIS OF THE GM4 FORMULAE	264
5.4.3	COMPUTATIONAL COMPLEXITY OF GM4 FORMULAE	267
5.4.4	NUMERICAL RESULTS	268
5.5	EXPLICIT FORMULA FOR A SPECIAL CLASS OF FOURTH ORDER ODES.....	269
5.5.1	NUMERICAL RESULTS	271

5.5.2 PREDICTOR-CORRECTOR PAIR USING (5.4-13a) AND (5.5-5)	273
5.5.3 NUMERICAL RESULTS	274
5.6 CONCLUSIONS AND RECOMMENDATIONS.....	276
 CHAPTER 6 NUMERICAL SOLUTIONS OF PERIODIC BOUNDARY VALUE PROBLEMS IN ELLIPTIC PDES....	277
6.1 INTRODUCTION.....	277
6.2 FORMULATION OF THE PROBLEM.....	277
6.3 THE DIFFERENCE EQUATIONS.....	278
6.4 SPECTRAL DECOMPOSITION METHOD.....	280
6.5 DIRECT METHOD FOR SOLVING THE SYSTEM (6.4-1).....	284
6.5.1 BLOCK ODD-EVEN/CYCLIC REDUCTION ALGORITHM	285
6.5.2 THE BLOCK TRI-REDUCTION (TR3) ALGORITHM	292
6.5.3 NUMERICAL RESULTS	300
6.5.4 STABILITY ANALYSIS OF THE BLOCK TR3 REDUCTION ALGORITHM	301
6.5.5 STABLE VERSION OF THE TR3 REDUCTION ALGORITHM ..	311
6.6 ITERATIVE METHOD FOR SOLVING (6.4.1).....	322
6.6.1 OUTLINE OF SLOR ITERATIVE METHOD FOR SOLVING (6.4-1)	322
6.6.2 RELATIONSHIP BETWEEN SPECTRAL RADIUS AND OVERRELAXATION FACTOR	326
6.6.3 NUMERICAL RESULTS	339
6.7 CONCLUSIONS AND RECOMMENDATIONS.....	341
 CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK.....	342
 REFERENCES	345
 APPENDICES	359

CHAPTER 1

INTRODUCTION

1.1 DIFFERENTIAL EQUATIONS

Many problems in Applied Mathematics lead to differential equations or systems of differential equations. Differential equations are equations that involve derivatives of some unknown functions of time and/or space. The solutions can only be explicitly determined in a relatively few cases. Therefore, numerical methods are essential to produce a sufficiently accurate approximation to the desired solutions.

Differential equations which involve only ordinary derivatives are called ordinary differential equations (ODEs). Partial differential equations (PDEs) are those differential equations which involve partial derivatives. For example,

$$y^{(1)} - xy = 2 \quad (1.1-1)$$

$$y^{(2)} + 3y^{(1)} + y = \cosh(x) \quad (1.1-2)$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial t^2} = 0 \quad (1.1-3)$$

are all differential equations. In (1.1-1) and (1.1-2) the unknown function is represented by y and is assumed to be a function of the single independent variable x , that is $y = y(x)$. For notational simplicity, we may suppress the argument x in $y(x)$ and its derivatives. The terms $y^{(1)}$ and $y^{(2)}$ in (1.1-1) and (1.1-2) are respectively, the first and second derivatives of the function $y(x)$ with respect to x . In (1.1-3), the unknown u is assumed to be a function of the two independent variables t and x , that is $u = u(t, x)$. Then $\frac{\partial^2 u}{\partial t^2}$ and $\frac{\partial^2 u}{\partial x^2}$ are the second partial derivatives of the function $u(t, x)$ with respect to t and x , respectively. Equations (1.1-1) and (1.1-2) involve ordinary

derivatives only and are therefore called ordinary differential equations. Equation (1.1-3) involves partial derivatives and therefore is a partial differential equation.

The general form of a differential equation can be written as

$$D[y] = f \quad (1.1-4)$$

where D is a differential operator and f is a given function of the independent variables t_i ; $i = 1, 2, \dots, n$ and n is the number of independent variables. The order of a differential equation is the order of its highest derivative and the degree of the derivative of the highest order in the rationalised equation is its degree. A linear equation is one which does not contain the product of the dependent variable with itself or any of its derivatives, otherwise it is said to be nonlinear. The general solution of the m^{th} order differential equation may contain m independent arbitrary constants. In order to determine the arbitrary constants in the general solution, m conditions need to be prescribed. If the m conditions are prescribed at one point, they are called initial conditions. The differential equation together with the initial conditions is known as the initial value problem. If the m conditions are prescribed at more than one point, they are called boundary conditions. The differential equation together with the boundary conditions is termed as the boundary value problem.

1.2 NUMERICAL SOLUTIONS OF ODES AND PDES

Differential equations are at the heart of our perception of the physical universe. The analytical solutions can only be explicitly determined in a relatively few simple cases. For this reason and with the advent of modern computers, numerical methods for their solution are central tools for obtaining

quantitative information on the physical behaviour. In the remaining parts of the thesis, we shall only concentrate on the numerical solutions of ODEs and PDEs. Thus the word solution(s) will only imply numerical solution(s) unless otherwise stated.

In the solution of the ODEs, we can classify two classes of methods, namely a single step method and a multistep method. A single step method determines the approximation y_{k+1} at the abscissa $x_{k+1} = x_k + h$ primarily on the basis of the approximation point (x_k, y_k) . In contrast, a multistep method uses the knowledge at the previous source abscissae $x_{k-1}, x_{k-2}, \dots, x_{k-n}$, to compute y_k . In general, we need to distinguish certain properties when the procedures are applied in practice. For systems of ODEs arising in physical problems, such as physics, chemistry, biology or engineering sciences, we often have specific criteria which influence the choice of the method.

As most ODEs of higher order can be reduced to systems of first order ODEs, the procedure will be presented, for simplicity and clarity; on the basis of the scalar differential equations of first order $y^{(1)} = f(x, y)$ for a single unknown function $y(x)$. An initial condition $y(x_0) = y_0$, stating the value y_0 at a given starting abscissa x_0 , is necessary in order to determine a certain solution among the one parameter family of solutions of ODEs of first order. The existence and uniqueness of a solution is assumed on the basis that the corresponding hypotheses are satisfied. Further treatments of numerical methods for the solution of ODEs are given in Gear[1971], Gekeler[1984], Henrici[1962], Jain[1984], Lambert[1991], Lapidus and Seinfeld[1971], Shampine and Gordon[1975], Aiken[1985], Butcher[1987], Hairer et al[1987]. In Sanugi[1986], a new treatment of numerical methods for the solution of ODEs based on the Geometric Mean (GM) approach is given. This approach is further extended to derive the composite GM method. From the

single step method application of the GM strategy, we will obtain the usual standard methods of Euler, Trapezoidal and Runge-Kutta. The combination of the standard Runge-Kutta and the new GM Runge-Kutta (RK-GM) methods offers an alternative adaptive strategy. From the multistep (specifically, the two step) method application of the GM strategy, a modified GM Numerov method is derived. The derivation of the closed and open formulae offers alternative formulae to be applied in the predictor corrector method.

Many mathematical formulation of problems in physics, chemistry or biology involve functions of several independent variables. This would eventually lead to satisfying certain PDEs. There is an enormous variety of PDEs and systems of PDEs that arise in these applications and their appropriate numerical analysis often requires special strategies. Broadly speaking, their numerical methods can be classified into two groups; namely the direct method and the iterative method. In a direct method, the solution is obtained in a fixed number of steps, subject only to rounding errors. In contrast, an iterative method starts with an initial approximation vector $\mathbf{u}^{(0)}$ to the solution vector \mathbf{u} , and generates a sequence of vectors $\{\mathbf{u}^{(k)}\}_{k=0}^{\infty}$ that converges to the vector \mathbf{u} . Many efficient direct methods exist (Reid[1977], George[1973], and Irons[1970]). However, from a practical point of view, the best solution method is one that accomplishes the job with a minimum total cost. The cost would include computer cost and the man-hour cost to develop and program the solution scheme. If the computer time is irrelevant, then the solution method selected should be one that works well and ^{is} easily implemented, not necessarily the best. However, for large scientific applications which easily saturate the computer capabilities, details of implementation become more important. Thus the most effective iterative methods to be used for large scale

computations are those which converge at a reasonable rate (not necessarily the fastest rate) and which can be easily adapted to the architectural features of the available computer at hand. Given the choice between direct and iterative methods, the usual criteria to decide are storage and work (number of arithmetic operations). For many problems, there is a limit to the number of unknowns above which a 'good' iterative method becomes more cost effective than a 'good' direct method (Young and Hageman[1981]). Further extensive representations of the numerical methods for solving the various types of PDEs can be found in Ames[1977], Collatz[1966], Gladwell and Wait[1979], Jain[1984], Mitchell and Griffiths[1980], Parter[1979], Smith[1985], Twizell[1984], Vemuri and Karplus[1981] and Hackbusch[1986].

In this thesis, we restrict our attention to solving second order PDEs for an unknown function with two independent variables. Moreover, The PDEs are of the self-adjoint elliptic case and the problem is of the periodic boundary value type. An alternative direct method of solving the systems of linear equations which result from the discretization of the problem is obtained. This method, which we called the Tri-Reduction (TR3) direct method in its modified form is proved to be numerically stable. We shall also discuss the iterative methods of solving the systems of linear equations mentioned above. In particular we shall show that the well known optimum relaxation parameter of the SOR theory cannot be applied to the case of the periodic problem. Alternatively, we shall derive the optimum relaxation parameter for the periodic case, which is distinct from that of the SOR theory.

CHAPTER 2

BASIC PRELIMINARIES AND FUNDAMENTALS OF NUMERICAL METHODS

2.1 INTRODUCTION

This chapter contains a short review of basic topics that will be repeatedly required in later chapters, together with an introduction to the terminology related to the discussion of the numerical solutions of differential equations (ODEs and PDEs).

2.2 BASIC PRELIMINARIES

In this section we shall list the definitions of terms and state basic results that are tools for numerical methods discussed in the following chapters.

2.2.1 MEANS

Suppose we are given a set of n numbers x_i for $i = 1, \dots, n$. Assume that $x_i \geq 0$. Define the generalized mean by

$$M_m = \left[\frac{1}{n} \sum_{i=1}^n x_i^m \right]^{1/m}. \quad (2.2.1-1)$$

If at least one of the x_i is zero and m is negative, we set M_m to zero. In particular, we have the arithmetic mean, for $m = 1$,

$$A = M_1 = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.2.1-2)$$

the geometric mean, if

$$G = \lim_{m \rightarrow 0} M_m = \left[\prod_{i=1}^n x_i \right]^{1/n} \quad (2.2.1-3)$$

and the harmonic mean, if

$$H = M_{-1} = \left[\frac{1}{n} \sum_{i=1}^n x_i^{-1} \right]^{-1}. \quad (2.2.1-4)$$

When neither all of the x_i are identical nor some of the x_i are zero and $m \leq 0$, then M_m is strictly monotonically increasing with m ; $\lim_{m \rightarrow \infty} M_m = \max_i x_i$ and $\lim_{m \rightarrow -\infty} M_m = \min_i x_i$.

Thus, we have

$$\min_i x_i \leq M_m \leq \max_i x_i, \quad (2.2.1-5)$$

and in particular

$$H \leq G \leq A. \quad (2.2.1-6)$$

In addition, we may define other means as follows:

Given two positive numbers x_1 and x_2 , the logarithmic mean of x_1 and x_2 is defined as

$$M_{\log} = \frac{x_1 - x_2}{\ln \left[\frac{x_1}{x_2} \right]}. \quad (2.2.1-7)$$

A more general concept is the weighted means, two of which are the weighted arithmetic mean and the weighted geometric mean.

Suppose that we have a set of numbers $x = \{x_i; i = 1, 2, \dots, n\}$ and weight $w = \{w_i; i = 1, 2, \dots, n\}$, then the weighted arithmetic mean x_{am} , is defined as

$$x_{am} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}, \quad (2.2.1-8)$$

and the weighted geometric mean x_{gm} , is defined by

$$x_{gm} = \left[\prod_{i=1}^n x_i^{w_i} \right]^{\frac{1}{\sum w_i}}. \quad (2.2.1-9)$$

In most practical applications, the weight w is normalized so that $\sum_i w_i = 1$.

Thus it is clear from the two computations of the means in (2.2.1-8) and (2.2.1-9), ^{that} the weighted arithmetic mean involves n multiplications and one division, while the weighted geometric mean consists of $n + 1$ powers. This means that any computation in the sense of the geometric mean incurs more computational work.

2.2.2 POWER SERIES

Suppose F is a field, which may be the field of complex numbers. Let a and c_i , for $i = 0, 1, \dots$, be the elements of F . A power series in one variable z , is defined as

$$Z = \sum_{i=0}^{\infty} c_i [z - a]^i, \quad (2.2.2-1)$$

The radius of convergence R , of a power series is a unique real finite number R such that Z converges if $|z - a| < R$ and diverges if $R < |z - a|$. The circle $|z - a| < R$ is called the circle of convergence of Z . The value of R is given by $R = 1 / \{ \limsup_{i \rightarrow \infty} [|c_i|]^{1/i} \}$.

A power series is absolutely and uniformly convergent in its circle of convergence, where it defines a single-valued complex function $\bar{f}(z)$. This function is a holomorphic function of z , since the series is termwise differentiable. Conversely, any holomorphic function in its domain can be represented by a power series in the neighbourhood of each point a of the domain. The representation is called the Taylor's expansion of $\bar{f}(z)$ at point a or in the neighbourhood of a .

Besides (2.2.2-1), we may have a power series of the form

$$\tilde{Z} = \sum_{i=0}^{\infty} c_i z_i^{-1} \quad (2.2.2-2)$$

with centre at infinity and its value at infinity is c_0 . By suitable transformations of (2.2.2-1) and (2.2.2-2), we may write every power series in the form

$$\sum_{i=0}^{\infty} c_i t^i, \quad (2.2.2-3)$$

where t is called the canonical parameter. When t is a local canonical parameter, we obtain the Laurent series given in the form $\sum_{-\infty}^{\infty} c_i t^i$. Taylor series are also called Power series.

2.2.3 SYMBOLIC COMPUTATION

Symbolic computation is a technique of manipulation of symbolic expressions on a computer. The term symbolic computation or computer algebra raises two distinct classes of activity:

- (a) the theoretical and structural development of all computer algebra systems,
- (b) the application of any of the existing systems to problems in mathematics, science and technology.

However, for the purpose of the work in this thesis, we are concerned exclusively with class (b), that is, the manipulation of algebraic formulae. Thus we shall restrict the discussion in this Section to this topic only.

There are several symbolic computation systems available that have been developed over the past thirty years. The first general purpose (non-numerical) symbolic computation systems appeared in the mid 1960's. Between then and ^{the}late 1980's, there have emerged systems such as MACSYMA, Scratchpad, REDUCE, FORMAC, Schoonship, CAMAL, ALTRAN, ALPAK, MATLAB, DERIVE, SMP, SAC-1, MAPLE and Mathematica. There are also some special purpose systems like LAM and Sheep for General Relativity. By restricting ourselves only to general purpose systems, the field may be narrowed down to about six only, namely, MACSYMA, Maple, Mathematica, REDUCE, Scratchpad and Derive.

Derive can only run on PC-DOS/MS-DOS systems and is of limited applicability. At the other extreme, Scratchpad is available only on IBM mainframes and needs about 8-12Mb of the main memory. The other four are the most commonly used. Of these MACSYMA is probably the most extensively developed but it requires about 4-5Mb of main memory and is only available on some machines. Mathematica which is claimed for applications that span all areas of science, technology and business where quantitative methods are used, also needs about 4Mb and upwards of memory. As at the end of 1990, versions of Mathematica are available for a wide variety of computer systems, including Apple Macintosh, CONVEX, DEC VAX (Ultrix and VMS) and RISC, Hewlett-Packard/Apollo, IBM 386-based compatibles (MS-DOS and Microsoft Windows) and IBM RISC, MIPS, NeXT, Silicon Graphics, Sony and Sun (and SPARC compatibles). Reviews of the system have appeared in both the popular and scientific press (see, for example, Barwise[1988], Simon[1989], Taubes[1988] and Wayner[1989]). Some initial impression of Mathematica's capabilities are given by Barwise[1988]. Maple and REDUCE require only a minimum of 1Mb of main memory and with 4Mb one can solve quite complex problems. Also, Maple and REDUCE will run on a large variety of machines ranging from PC's to Crays. Jenson and Niordson[1977] have given a comparative study of some of the systems listed.

We shall now describe REDUCE in some detail. REDUCE is a system for performing algebraic operations accurately; irrespective of the complexity of the expressions. It can do various manipulations of polynomials including expansion and factorization as well as the extraction of parts of polynomials as required. REDUCE has the facilities for defining new functions and extending program syntax and can do analytic differentiation and integration of functions. Other capabilities of REDUCE include facilities for the

solution of a variety of algebraic equations, facilities for the output of expressions in a variety of formats, facilities for manipulation of symbolic arrays and matrix operations, facilities for generating numerical programs from symbolic input, simplifications of expressions and substitutions and pattern matching in various forms. There are also user-contributed packages. The basic REDUCE system is being continually extended by a library of packages contributed by users. At present there are two classes of such packages: those that are distributed with the system and those that have been written since the appearance of the current version of REDUCE. These packages may be available from the REDUCE network library at the e-mail address `reduce-network@rand.org`.

REDUCE can be run in both modes, the batch and interactive modes. Its design of being an interactive system enables the user to input an algebraic expression and inspect its value before moving on to the next calculation. However, if necessary, a sequence of commands can be given to REDUCE and the results obtained without any intervention by the user during the computation.

We shall now illustrate the interactive use of REDUCE. After a successful logging-in, the user can run REDUCE on the SUNA at Loughborough University by typing 'reduce' at the prompt 'suna%', after which REDUCE will respond with a banner message which reports the version number and the current system release date which may change from time to time i.e.,

```
suna% reduce
REDUCE 3.4, 15-Jul-91 ...
```

It then prompts the user for input by

```
1:
```

We can now type a REDUCE expression, terminated by a^a semi-colon to indicate the end of the REDUCE expression, for example:

```
1:(x**4 - y**4)/(x - y);
```

Note that we type the expression exactly like that of the FORTRAN expression except that the REDUCE expression is terminated by a semi-colon. When the end-of-line character is encountered, which is normally the RETURN key on an ASCII terminal, the statement ending with ; or \$ is processed. Thus for the illustration above, we obtain the results as follows:

```
      3      2      2      3
X  + X *Y + X*Y  + Y
2:
```

where (2:) is automatically assigned to the next command. Input may be in the lower or upper case, but the output is in the upper case.

The results of a given calculation are also saved in the variable WS (for Workspace), which enables it to be used in the next calculation for further processing.

For example, if we enter on line (2:) following the results of evaluation of line (1:), the expression

```
int(ws,x);
```

will integrate the function $x^3 + x^2y + xy^2 + y^3$ with respect to x to obtain

```
      3      2      2      3
X*(3*X  + 4*X *Y + 6*X*Y  + 12*Y )
-----
                        12
```

```
3:
```

Note that after each evaluation of an expression a line number command which prompts the user for the next input

follows. If we do not have anything to process further and wish to leave the REDUCE session, we may do so by typing the word 'bye'. This ends the REDUCE session and returns to the system prompt.

However, in many cases, we may continue further and use some previous results in the succeeding calculations. One way of doing this is by assigning a variable name to an expression as follows,

```
u := (x**4 - y**4)/(x - y);
```

This enables the value of the right-hand side of the above to be represented by u and used in later calculations.

REDUCE also has the capability of handling symbolic matrices. For example,

```
matrix m(2,2);
```

declares m to be a 2 x 2 matrix, and

```
m := mat((a,b),(c,d));
```

gives its element values. Expressions which involve matrix operations may now be evaluated. For example, 1/m evaluates the inverse of a matrix m, det(m) calculates the determinant of the matrix m and n**2*m**(-2) gives another combined matrix, assuming that m and n have been declared as matrices.

REDUCE has a wide range of substitution capabilities. The system knows about elementary functions, but does not automatically reckon their well-known characteristics. However, REDUCE has an important class of commands which allows substitutions for variables and expressions to be defined during the evaluation of expressions. Such substitutions use forms of the command LET.

The LET rules will stay in effect until replaced or CLEARED. For example, after assigning the expression $\frac{x^4 - y^4}{x - y}$ to u, we can set the numerical values of x and y and thus obtain the numerical value of u by using the command LET as follows:

```
let x = 1, y = 2;  
u;
```

REDUCE will respond to give the result

15

But if we wish to assign the value to another variable v, then we write as follows:

```
let x = 1, y = 2;  
v := u;
```

REDUCE will then respond with

V := 15

Another very useful command for the purpose of substitution is the OPERATOR command. The user may add new prefix operators to the system by using the declaration OPERATOR. For example,

```
operator p,q,taylor;
```

adds the prefix operators p, q and taylor to the system. This allows symbols such as p(x,y), q(x/y,z), taylor(x,n) to be used in expressions. By associating LET statement with the operator declaration statement, we can have a meaningful operator symbol or a definition of some of its properties. For example, if we wish to arbitrarily define the average of two numbers x and y, we may declare the operator 'av' as follows:

```
operator av;  
for all x,y let av(x,y) = (x + y)/2;
```

Hence, if we have the command:

```
m := av(10,50);
```

REDUCE will give the average of 10 and 50 as

```
m := 30
```

Thus we may use this facility as a tool to simplify complicated algebraic expressions. For example, we may express the product of the cosine of two angles, say $\cos(a) \times \cos(b)$, into the sum of two trigonometric values as $\frac{1}{2}[\cos(a-b) + \cos(a+b)]$. This can be done in REDUCE as follows:

```
operator p;  
for all a,b let p(a)*p(b) = (p(a-b) + p(a+b))/2;
```

This will result in any product of two trigonometric cosines to be simplified into the sum of two related trigonometric values.

Note that we have used the FOR ALL declaration in the above example; this may be used if a substitution for all possible values of a given argument of an operator is required. The LET command may also be used as an asymptotic command. For example, in the expansions of polynomials involving variables which are known to be small, it is often desirable to curtail the expansions after certain finite powers of these variables. Thus the command

```
let x^8 = 0;
```

will cause the system to expand the polynomial up to and including the seventh power of x only. However, this substitution should be used with care because it is applied only during polynomial manipulation rather than to the whole evaluated expression. If several variables are involved, it is necessary to supply an asymptotic

weight to each variable and count up the total weight of each product in an expanded expression before the decision to keep the term or not is made.

There are also a number of reserved operators from the three classes of operators, namely the infix, prefix and mathematical operators supplied together with the system. The user can add further rules for the reduction of expressions involving the reserved mathematical operators by using the LET command. New infix operators may be added by the user by using the declarations INFIX and PRECEDENCE.

We shall now describe another operator which is very handy for the solution of simultaneous algebraic equations. The SOLVE operator allows one to solve one or more simultaneous algebraic equations. For example,

```
solve(log(sin(x+3))^5 = 8,x);  
solve(a*log(sin(x+3))^5 = b, sin(x+3));  
solve({a*x + y = 3,y = -2,{x,y}});
```

SOLVE returns a list of solutions. If there is only one unknown, each solution is an equation for the unknown. If a complete solution was found, the unknown will appear automatically on the left-hand side of the equation. On the other hand, if the solve package could not find a solution, the "solution" will be an equation for the unknown. If there are several unknowns, each solution will be a list of equations for the unknowns. By turning the switch MULTIPLICITIES, a list of the multiplicities of the solutions will be explicitly displayed. There are also several options which can be used with the SOLVE operator. By turning the switch SOLVESINGULAR on (the default setting), degenerate systems may be solved by introducing appropriate arbitrary constants. By switching OFF SOLVESINGULAR suppresses the solutions of consistent singular equations.

A REDUCE program is composed from a set of functional commands which are evaluated sequentially by the computer. These commands are constructed from declarations, statements and expressions which we have just explained in the preceding paragraphs. We shall now illustrate a simple REDUCE program of solving a system of linear equations in four unknowns p, q, r and s . Suppose the equations are given as follows:

$$\begin{aligned}3p + 2q - 4r + s &= 5, \\4p - 4q + r - 5s &= 1, \\p + 2q + 4r + 2s &= 9, \\8p + 6q + r + 9s &= 12\end{aligned}$$

The REDUCE program can be written as follows:

```
%Line begins with '%' is a comment statement;
%REDUCE program to solve system of equations;
solve({3*p + 2*q - 4*r +s = 5,
      4*p - 4*q + r - 5*s = 1,
      p + 2*q + 4*r + 2*s = 9,
      8*p + 6*q + r + 9*s = 12},{p,q,r,s});
end;
```

In many applications, we may need to load previously prepared REDUCE files into the system, or write the output onto other files. The commands IN and OUT in REDUCE offer the facility for this purpose. The command IN takes in a list of file names as argument and directs the system to input each file into the system for processing. For example, if the REDUCE program to solve the system of four equations written above is kept in a file named 'solve', then

```
1: in solve;
```

will load the file named 'solve'. A file to be read using IN must end with ';end;'.

If we are using the interactive mode, on the successful processing of the program the results will automatically be printed on the screen as follows:

```
%Line begins with '%' is a comment statement;
%REDUCE program to solve system of equations;
  solve({3*p + 2*q - 4*r + s = 5,
         4*p - 4*q + r - 5*s = 1,
         p + 2*q + 4*r + 2*s = 9,
         8*p + 6*q + r + 9*s = 12},{p,q,r,s});

      346      2929      361      - 296
{{P=-----,Q=-----,R=-----,S=-----}}
      327      654      327      109

end;

2:
```

To terminate the REDUCE session, we type after the REDUCE prompt '2: ' the word 'bye'.

However, if we are using batch mode, where we need to transfer the results of the REDUCE program to some other file, we may use the command OUT. For the same example above, let the output file be named as 'result'. To run the program we execute the following commands in sequence on entering the REDUCE session.

```
1: out result;
2: in solve;
3: bye;
```

Note that we need to inform the REDUCE system, first the output file where the results are to be directed to, then followed by the relevant input file to the system. The command OUT takes a single file name as argument, and directs the output to the named file until another OUT changes the output file, or SHUT closes it. Again we end the REDUCE session by typing the word 'bye'. Thus the results are contained in the file 'result'.

Symbolic computation is practically useful especially in the context of modelling and field problems. Brown and Hearn[1978] have cited some of these reasons as follows:

(1) Sometimes it is prohibitively expensive, or even impossible, to solve an essentially numerical problem by purely numerical means because it involves too many variables, demands a greater accuracy, or is presented in an ill-conditioned or intractable form. However, a symbolic transformation may reduce the dimensionality, evade a large source of round-off error, finesse the ill-conditioning, and otherwise change the problem into one that can be solved by standard numerical methods. Transformations are a very general and natural way to represent many kinds of information, particularly mathematical relations Wolfram[1991].

(2) The algebraic result obtained from symbolic computation may subsequently be evaluated using a variety of parameter values.

(3) Symbolic computation lends an opportunity for realizing the important computational symbiosis between numerical experiments and symbolic theories.

(4) Symbolic computation can be used to generate a needed numerical subroutine.

(5) Lastly, in the realm of partial differential equations, Cloutman and Fullerton[1977] have used symbolic multidimensional Taylor series expansions, computed by the Altran system, to analyse the discretization and round-off errors of various methods, to eliminate inaccurate and unstable methods prior to coding and testing and to develop methods in which the lowest order errors cancel each other out. We shall utilize this idea in the treatment of the geometric mean (GM) methods for the ordinary differential equations discussed in Chapters 4 and 5.

2.3 FUNDAMENTALS OF NUMERICAL METHODS

The numerical solution of a differential equation on a fixed number of grid points starts with finding how to express the solution on the discrete coordinate points and how to approximate the differential and integral operators in the discrete space. A finite number of dependent variables may be represented by a vector. Numerical approximations of the differential or integral operators may be expressed by matrices. Thus, linear algebra is an important tool in numerical analysis. In the section to follow we shall list some of the relevant definitions and results associated with vectors and matrices.

2.3.1 VECTOR AND MATRIX

One of the fundamental reasons for reformulating problems as equivalent linear algebra problems is to introduce some geometric insight. Vector and matrix algebra offer comprehensive concepts to this process. In the following paragraphs we shall list definitions and results which are useful in this context.

Definition 2.3.1-1 : Let V be a vector space and let $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m \in V$. We say that $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ are linearly dependent if there exists a set of scalars $\alpha_1, \alpha_2, \dots, \alpha_m$ with at least one nonzero scalar such that

$$\sum_{i=1}^m \alpha_i \mathbf{v}_i = \mathbf{0}. \quad (2.3.1-1)$$

Without loss of generality, we assume $\alpha_i \neq 0$, so that

$$\mathbf{v}_i = -\frac{1}{\alpha_i} \sum_{\substack{j=1 \\ j \neq i}}^m \alpha_j \mathbf{v}_j. \quad (2.3.1-1a)$$

We say that \mathbf{v}_i is a linear combination of the vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{i-1}, \mathbf{v}_{i+1}, \dots, \mathbf{v}_m$.

We say the vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$ are linearly independent if they are not dependent. Therefore, the only choice of

scalars $\alpha_1, \dots, \alpha_m$ for (2.3.1-1) to be true is the trivial case of $\alpha_1 = \dots = \alpha_m = 0$.

The set $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$ is a basis for V if for every $\mathbf{v} \in V$, there exists unique scalars $\alpha_1, \dots, \alpha_m$ such that

$$\mathbf{v} = \sum_{j=1}^m \alpha_j \mathbf{v}_j. \quad (2.3.1-1b)$$

Note that this implies $\mathbf{v}_1, \dots, \mathbf{v}_m$ are independent.

Theorem 2.3.1-1 : If V is a vector space with basis $\{\mathbf{v}_1, \dots, \mathbf{v}_m\}$, then every basis for V will contain exactly m vectors. The number m is called the dimension of V .

An array of n numbers may be expressed either as a column or row vector of order n . We shall define a column vector representing a column-wise array of n numbers by

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad (2.3.1-2)$$

and each x_i for $i = 1, 2, \dots, n$, is called the component of the vector \mathbf{x} .

Definition 2.3.1-2 : The transpose of a vector \mathbf{x} is denoted by \mathbf{x}^T and represented by a row-wise vector

$$\mathbf{x}^T = [x_1, x_2, \dots, x_n]. \quad (2.3.1-3)$$

The null vector is represented by $\mathbf{0}$ which means that the vector $\mathbf{0}$ has all its components zero.

Some basic operations and properties of vectors are as follows:

Definition 2.3.1-3 : The addition and subtraction of two vectors \mathbf{x} and \mathbf{y} are defined as

$$\mathbf{x} \pm \mathbf{y} = \begin{bmatrix} x_1 \pm y_1 \\ x_2 \pm y_2 \\ \vdots \\ x_n \pm y_n \end{bmatrix}. \quad (2.3.1-4)$$

Definition 2.3.1-4 : The scalar multiplication of a vector \mathbf{x} by c is defined by

$$c\mathbf{x} = \mathbf{x}c = \begin{bmatrix} cx_1 \\ \vdots \\ cx_n \end{bmatrix} \quad (2.3.1-5)$$

Definition 2.3.1-5 : Two vectors, \mathbf{x} and \mathbf{y} , are said to be equal if $x_i = y_i$ for all $i = 1, 2, \dots, n$.

Definition 2.3.1-6 : The inner (scalar) product of two vectors, \mathbf{x} and \mathbf{y} , is written and defined as

$$(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{x} = \sum_{i=1}^n x_i y_i. \quad (2.3.1-6)$$

Definition 2.3.1-7 : Two vectors, \mathbf{x} and \mathbf{y} , are orthogonal if and only if the scalar product is zero, that is,

$$(\mathbf{x}, \mathbf{y}) = 0. \quad (2.3.1-7)$$

Definition 2.3.1-8 : The Euclidean norm of a vector \mathbf{x} in C^n or R^n is denoted and defined by

$$\|\mathbf{x}\|_2 = \sqrt{(\mathbf{x}, \mathbf{x})} = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2}. \quad (2.3.1-8)$$

If $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ forms a basis for C^n or R^n , and if $(\mathbf{x}_i, \mathbf{x}_j) = 0$, for all $i \neq j$, $1 \leq i, j \leq n$, then we say $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

is an orthogonal basis. If all basis vectors have Euclidean norm 1, i.e. $(\mathbf{x}_i, \mathbf{x}_i) = 1$, then the basis is called orthonormal.

By introducing an orthonormal basis for a vector space V , one can decompose an arbitrary vector into its components in the direction of the basis vectors. Let $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ be an orthonormal basis for V and let $\mathbf{x} \in V$. By using the basis, we may express the vector \mathbf{x} in the form

$$\mathbf{x} = \alpha_1 \mathbf{u}_1 + \dots + \alpha_n \mathbf{u}_n$$

for some unique coefficients $\alpha_1, \dots, \alpha_n$. Now the coefficient α_j , for every $1 \leq j \leq n$, can be found by forming the inner product of \mathbf{x} and \mathbf{u}_j ;

$$(\mathbf{x}, \mathbf{u}_j) = \alpha_1 (\mathbf{u}_1, \mathbf{u}_j) + \dots + \alpha_n (\mathbf{u}_n, \mathbf{u}_j) = \alpha_j$$

since $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$ is an orthonormal basis for V . Thus

$$\mathbf{x} = \sum_{j=1}^n (\mathbf{x}, \mathbf{u}_j) \mathbf{u}_j. \quad (2.3.1-8a)$$

The formula (2.3.1-8a) gives the decomposition of a vector into its components in the direction of the basis vectors.

A matrix A is a rectangular array of real or complex numbers and defined by

$$A = \begin{bmatrix} a_{11} & a_{12} \dots a_{1n} \\ a_{21} & a_{22} \dots a_{2n} \\ \vdots & \vdots \\ a_{m1} & a_{m2} \dots a_{mn} \end{bmatrix} \quad (2.3.1-9)$$

where a_{ij} is called the element in the i^{th} row and j^{th} column. We say the matrix in (2.3.1-9) to be of order $m \times n$. Generally, we denote matrices by capital letters and

their entries by small letters, usually corresponding to the name of the matrix, as above. We may also use the notation (a_{ij}) to denote the matrix A . A column vector or a row vector may be considered as special cases of matrices. When $n = m$, the matrix is said to be a square matrix. A matrix of order n is shorthand for a square matrix of order $n \times n$.

The following definitions give the common operations on matrices.

Definition 2.3.1-9 : Let A be a matrix of order $m \times n$. The transpose of a real matrix A is denoted by A^T and defined by $A^T = (a_{ij})^T = (a_{ji})$. A^T is now of order $n \times m$.

Similarly, the conjugate transpose A^* of a complex matrix A also has order $n \times m$, and $A^* = (\bar{a}_{ji})$, where \bar{a}_{ji} denotes the complex conjugate of the complex number a_{ji} for every i and j . If $\bar{a}_{ji} = a_{ji}$, for every i and j , then we have a real matrix A and vice-versa.

Definition 2.3.1-10 : The identity matrix I is defined as the matrix where the diagonal elements are 1 and off-diagonal elements are 0.

For all matrices A of order $m \times n$ and B of order $n \times p$, we have

$$AI = A \text{ and } IB = B.$$

Definition 2.3.1-11 : Suppose A is a square matrix of order n . The inverse of a matrix A is denoted by A^{-1} and defined by

$$A^{-1}A = AA^{-1} = I. \tag{2.3.1-10}$$

The inverse of a matrix, if it exists is unique.

Definition 2.3.1-12 : A matrix is a null matrix of order $m \times n$ if all its entries are zero. We denote a null matrix by 0 .

Definition 2.3.1-13 : A matrix A is symmetric if it is equal to its transpose. Thus, $A^T = A$ or $a_{ij} = a_{ji}$ for every $i, j = 1, 2, \dots, n$.

Of necessity, all matrices which are symmetric must also be square.

Definition 2.3.1-14 : A matrix A is said to be a diagonal matrix if all the off-diagonal elements are zero. That is $a_{ij} = 0$ for every $i \neq j$.

Definition 2.3.1-15 : A matrix A is lower triangular if $a_{ij} = 0$ for every $i > j$ and A is strictly lower triangular if $a_{ij} = 0$ for every $i \geq j$.

Definition 2.3.1-16 : A matrix A is upper triangular if $a_{ij} = 0$ for every $i < j$ and A is strictly upper triangular if $a_{ij} = 0$ for every $i \leq j$.

Definition 2.3.1-17 : A matrix A is said to be nonsingular if $\det(A) \neq 0$, where $\det(A)$ denotes the determinant of the matrix A . Thus it is clear that if A is nonsingular A^{-1} exists.

Definition 2.3.1-18 : The rank of a square matrix is the maximum number of independent columns.

The rank of A is equal to that of A^T ; which means that the number of independent columns of A is always identical with the number of independent rows. If the rank of A is equal to the order of A , then the determinant of A is nonzero, $\det(A) \neq 0$. If $\det(A) = 0$, the rank of A is less than n and A is said to be singular.

Definition 2.3.1-19 : A is diagonally dominant if $|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$, for $i = 1, 2, \dots, n$ and A is said to be strictly diagonally dominant if the strict inequality holds for every $i = 1, 2, \dots, n$, that is, $|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|$.

Definition 2.3.1-20 : A real matrix A is positive definite if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$. Thus the real matrix A is positive definite if and only if it is symmetric and all its eigenvalues are positive.

Definition 2.3.1-21 : Suppose A and B are two given matrices, each of order $m \times n$. Matrix addition and subtraction are defined by

$$C = A \pm B = (a_{ij} \pm b_{ij}) = (c_{ij}) \quad (2.3.1-11)$$

where $C = (c_{ij})$ denotes the resultant matrix and is of order $m \times n$.

Thus for any matrix A of order $m \times n$, combining definitions (2.3.1-12) and (2.3.1-21), we have

$$A + \mathbf{0} = \mathbf{0} + A = A.$$

Definition 2.3.1-22 : The scalar multiplication of a matrix A of order $m \times n$, by a scalar α gives another matrix C defined by

$$C = \alpha A = A\alpha = (\alpha a_{ij}) = (c_{ij}). \quad (2.3.1-12)$$

The order of C is also $m \times n$.

Definition 2.3.1-23 : Two matrices A and B of order $m \times n$ are said to be equal only if $a_{ij} = b_{ij}$ for every $i = 1, \dots, m$ and $j = 1, \dots, n$.

Theorem 2.3.1-2 : Let A be a square matrix in which the elements may be real or complex numbers. Let the vector space V to be either R^n or C^n , then the following are equivalent statements.

- (1) $Ax = b$ has a unique solution $x \in V$ for every $b \in V$.
- (2) $Ax = 0$ implies $x = 0$.
- (3) A^{-1} exists.
- (4) $\det(A) \neq 0$.
- (5) $\text{Rank}(A) = n$.

Definition 2.3.1-26 : The number λ , complex or real is an eigenvalue of the square matrix A if there is a vector, $x \neq 0$, such that

$$Ax = \lambda x. \quad (2.3.1-16)$$

The vector x is called an eigenvector corresponding to the eigenvalue λ .

From Theorem 2.3.1-2, statements (2) and (4), λ is an eigenvalue of A if and only if

$$\det(A - \lambda I) = 0. \quad (2.3.1-17)$$

The relation (2.3.1-17) is called the characteristic equation for A . If A is of order n , then $f_A(\lambda) = \det(A - \lambda I)$ is a polynomial of degree n exactly, called the characteristic polynomial of A . Therefore we may write $f_A(\lambda)$ in the form

$$f_A(\lambda) = (-1)^n \lambda^n + (-1)^{n-1} (a_{11} + \dots + a_{nn}) \lambda^{n-1} + (\text{terms of degree } \leq n-2). \quad (2.3.1-18)$$

From the coefficient of λ^{n-1} , we define

$$\text{trace}(A) = a_{11} + a_{22} + \dots + a_{nn} \quad (2.3.1-19)$$

which is often a quantity of interest in the study of A .

Since $f_A(\lambda)$ is of degree n , therefore in general, a $n \times n$ matrix has at most, n distinct eigenvalues.

Definition 2.3.1-27 : Let A and B be square matrices of the same order. Then A is similar to B if there exists a nonsingular matrix P such that

$$B = P^{-1}AP. \quad (2.3.1-20)$$

Note that this is a symmetric relation since

$$A = Q^{-1}BQ \quad Q = P^{-1}. \quad (2.3.1-20a)$$

The relation (2.3.1-20) can be interpreted as follows: A and B are matrix representations of the same linear transformation T from V to V , but with respect to different bases for V . The matrix P is known as the change of basis matrix, which relates the two representations of a vector $x \in V$ with respect to the two bases used.

Some of the simple properties of similar matrices and their eigenvalues are as follows:

(1) If A and B are similar, then $f_A(\lambda) = f_B(\lambda)$.

This follows from the definition (2.3.1-20) for

$$\begin{aligned} f_B(\lambda) &= \det(B - \lambda I) \\ &= \det(P^{-1}(A - \lambda I)P) \\ &= \det(P^{-1}) \det(A - \lambda I) \det(P) = f_A(\lambda) \end{aligned}$$

since

$$\det(P) \det(P^{-1}) = \det(PP^{-1}) = \det(I) = 1.$$

(2) Similar matrices have exactly the same eigenvalues and there is a one-to-one correspondence of the eigenvectors. If $Ax = \lambda x$, then using (2.3.1-20),

$$P^{-1}APP^{-1}\mathbf{x} = \lambda P^{-1}\mathbf{x}$$

$$B\mathbf{z} = \lambda\mathbf{z} \quad \mathbf{z} = P^{-1}\mathbf{x}$$

Trivially, $\mathbf{z} \neq \mathbf{0}$, otherwise \mathbf{x} would have been zero. Similarly, given any eigenvector \mathbf{z} of B , this argument can be reversed to obtain a corresponding eigenvector $\mathbf{x} = P\mathbf{z}$ for A .

(3) We have $f_A(\lambda)$ is invariant under similarity transformations of A , therefore the coefficients of $f_A(\lambda)$ are also invariant under such similarity transformations. In particular, if A and B are similar, then

$$\text{trace}(A) = \text{trace}(B) \quad \det(A) = \det(B). \quad (2.3.1-21)$$

We now state some important results about the canonical forms for matrices. These forms relate the structure of a matrix to its eigenvalues and eigenvectors. They find use in many applications in other areas of mathematics and science.

Definition 2.3.1-28 : A square matrix U is called unitary if

$$U^*U = UU^* = I$$

where U^* is the conjugate transpose of U . If U is a real matrix, it is usually called orthogonal and U^* is replaced by the transpose of U . The rows (or columns) of a unitary matrix of order n form an orthonormal basis for C^n . Similarly for orthonormal matrices for R^n .

Theorem 2.3.1-3 (Schur normal form) : Let the matrix be of order n with elements from C . Then there exists a unitary matrix U such that

$$T = U^*AU \quad (2.3.1-22)$$

is upper triangular. Since T is triangular, and $U^* = U^{-1}$,

$$f_A(\lambda) = f_T(\lambda) = (\lambda - t_{11}) \dots (\lambda - t_{nn}) \quad (2.3.1-23)$$

and thus the eigenvalues of A are the diagonal elements of T .

We note that by combining (2.3.1-21) and (2.3.1-22), we obtain

$$\text{trace}(A) = \sum_{j=1}^n \lambda_j \quad \det(A) = \prod_{j=1}^n \lambda_j \quad (2.3.1-24)$$

where λ_j ; $1 \leq j \leq n$ are the eigenvalues of A and they form the diagonal elements of T .

The theorem 2.3.1-3 is more of a theoretical tool, rather than a computational one. Theorem 2.3.1-4 to follow has a more important application.

Theorem 2.3.1-4 (Principal axes theorem) : Let A be a Hermitian matrix of order n . That is $A^* = A$. Then A has real eigenvalues λ_j ; $1 \leq j \leq n$, not necessarily distinct, and corresponding eigenvectors u_j ; $1 \leq j \leq n$, which form an orthonormal basis for C^n . If A is also real, the eigenvectors u_j ; $1 \leq j \leq n$, can be considered as real; and they form an orthonormal basis for R^n . Finally there exists a unitary matrix U for which

$$U^*AU = D = \text{diag}[\lambda_1, \dots, \lambda_n] \quad (2.3.1-25)$$

where D is a diagonal matrix with diagonal elements λ_j ; for $1 \leq j \leq n$. For the case of a real matrix A , then U is considered as orthogonal and the result of (2.3.1-25) follows with U^* replaced by U^T .

$$J_n(\lambda) = \begin{bmatrix} \lambda & 1 & 0 & \dots & 0 \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & 1 \\ 0 & & & & \lambda \end{bmatrix} \quad n \geq 1.$$

$J_n(\lambda)$ has the single value eigenvalue λ , of algebraic multiplicity n and geometric multiplicity 1. The algebraic multiplicity of an eigenvalue λ of a matrix A is its multiplicity as a root of the characteristic polynomial $f_A(\lambda)$; while its geometric multiplicity is the maximum number of linearly independent eigenvectors associated with the eigenvalue. The algebraic and geometric multiplicities of an eigenvalue need not be equal.

Theorem 2.3.1-6 (Jordan canonical form) : Let A be of order n . Then there exists a nonsingular matrix P such that

$$P^{-1}AP = \begin{bmatrix} J_{n_1}(\lambda_1) & & & \\ & J_{n_2}(\lambda_2) & & 0 \\ & & \ddots & \\ 0 & & & J_{n_r}(\lambda_r) \end{bmatrix}. \quad (2.3.1-28)$$

The eigenvalues λ_i ; $1 \leq i \leq r$, need not be distinct. For A Hermitian, Theorem 2.3.1-4 implies that $n_i = 1$; $1 \leq i \leq r$. Therefore, the sum of the geometric multiplicities is n , the order of the matrix A .

2.3.2 VECTOR AND MATRIX NORMS

The introduction of the concept of the norm of a vector allows one to measure the size of a vector.

Definition 2.3.2-1 : Let V be a vector space and let $\|\cdot\|$ be a real-valued function defined on V . Then $\|\mathbf{x}\|$ is a norm if

- 1) $\|\mathbf{x}\| \geq 0$ for all $\mathbf{x} \in V$; and $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$.
- 2) $\|\alpha\mathbf{x}\| = |\alpha| \|\mathbf{x}\|$, for all $\mathbf{x} \in V$ and all scalars α .
- 3) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, for all $\mathbf{x}, \mathbf{y} \in V$.

Simple consequences of the definition of the norm are the triangle inequality

$$\|\mathbf{x} - \mathbf{y}\| \leq \|\mathbf{x} - \mathbf{z}\| + \|\mathbf{z} - \mathbf{y}\| \quad (2.3.2-1)$$

and the reverse triangle inequality,

$$|\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x} - \mathbf{y}\| \quad (2.3.2-2)$$

for all $\mathbf{x}, \mathbf{y} \in V$.

For $1 \leq p < \infty$ and $\mathbf{x} \in V$, we define the p -norm by

$$\|\mathbf{x}\|_p = \left(\sum_{j=1}^n |x_j|^p \right)^{1/p}. \quad (2.3.2-3)$$

The maximum norm is defined by

$$\|\mathbf{x}\|_\infty = \max_{1 \leq j \leq n} |x_j| \quad (2.3.2-4)$$

for $\mathbf{x} \in V$.

Note that the norm of \mathbf{x} on C^n or R^n is a continuous function of the components of \mathbf{x} .

Theorem 2.3.2-1 (Equivalence of norms) : Let N and M be two norms on a finite dimensional space V . Then, for all $\mathbf{x} \in V$, there are constants $c_1, c_2 > 0$ such that

$$c_1 M(\mathbf{x}) \leq N(\mathbf{x}) \leq c_2 M(\mathbf{x}). \quad (2.3.2-5)$$

Note that this theorem does not generalize to infinite dimensional spaces.

Many numerical methods for problems involving linear systems result in a sequence of vectors $\{\mathbf{x}_i; i \geq 0\}$, and the concept of convergence is of prime importance.

Definition 2.3.2-2 : A sequence of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m, \dots\}$ in V (real or complex) is said to converge to a vector \mathbf{x} if and only if

$$\|\mathbf{x} - \mathbf{x}_m\| \rightarrow 0 \quad \text{as} \quad m \rightarrow \infty. \quad (2.3.2-6)$$

By using Theorem 2.3.2-1 and Definition 2.3.2-2, we may conclude that $\mathbf{x}_m \rightarrow \mathbf{x}$ with the M norm if and only if it converges with the N norm. We should emphasize that this result is only true for the finite dimensional spaces.

We shall now extend Definition 2.3.2-1 to cope with the matrix norm. As we have noted earlier in Section 2.3.1 that a vector is a special case of a matrix, thus a matrix norm should satisfy the usual three requirements of a vector norm listed in Definition 2.3.2-1. By using the same notation of norm, we define the matrix norm as follows:

Definition 2.3.2-3 : The matrix norm satisfies the requirements 1, 2 and 3 of Definition 2.3.2-1 in addition to two other conditions set as follows:

$$4) \quad \|AB\| \leq \|A\| \|B\| \quad (2.3.2-7)$$

5) Usually the vector space, V , which we shall be working with will have some vector norm, say $\|\mathbf{x}\|_V$, for

all $\mathbf{x} \in V$. We require that the matrix and vector norms to be compatible,

$$\|\mathbf{Ax}\|_v \leq \|A\| \|\mathbf{x}\|_v, \quad (2.3.2-8)$$

for all $\mathbf{x} \in V$, and for all A .

Usually when given a vector space with a norm $\|\cdot\|_v$, an associated matrix norm is defined by

$$\|A\| = \sup_{\mathbf{x} \neq 0} \frac{\|\mathbf{Ax}\|_v}{\|\mathbf{x}\|_v}. \quad (2.3.2-9)$$

It is often known as the operator norm. Table(2.3.2) gives some of the most important operator norms $\|A\|_p$ of a matrix A induced by the vector norms $\|\mathbf{x}\|_p$.

Definition 2.3.2-4 : The spectral radius of a matrix A is denoted by $\rho(A)$ and is defined by $\rho(A) = \max_i |\lambda_i|$ where $\lambda_1, \lambda_2, \dots, \lambda_n$ are the eigenvalues of the matrix A . The spectrum of A is the set of all eigenvalues of A .

Vector norm	Matrix norm
$\ \mathbf{x}\ _1 = \sum_{i=1}^n x_i $	$\ A\ _1 = \max_{1 \leq j \leq n} \sum_{i=1}^n a_{ij} $
$\ \mathbf{x}\ _2 = \left[\sum_{i=1}^n x_i ^2 \right]^{1/2}$	$\ A\ _2 = \sqrt{\rho(A^*A)}$
$\ \mathbf{x}\ _\infty = \max_{1 \leq i \leq n} x_i $	$\ A\ _\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n a_{ij} $

Table(2.3.2): Vector norms and the associated operator matrix norms

The operator norm of the matrix A defined by $\|A\|_1$ is called the column norm, while that defined by $\|A\|_\infty$ is called the row norm of A .

Theorem 2.3.2-2 : Let A be any square matrix. Then for an arbitrary matrix norm,

$$\rho(A) \leq \|A\|. \quad (2.3.2-10)$$

Moreover, if $\varepsilon > 0$ is given, then there exists an operator matrix norm $\|\cdot\|_\varepsilon$, such that

$$\|A\|_\varepsilon \leq \rho(A) + \varepsilon. \quad (2.3.2-10a)$$

This theorem can be used to analyse the rate of convergence of some of the iteration methods considered in chapter 3. As a consequence of Theorem 2.3.2-2, an important corollary follows.

Corollary 2.3.2-1 : For a square matrix A , $\rho(A) < 1$ if and only if $\|A\| < 1$ for some operator matrix norm.

2.3.3 CONVERGENCE AND PERTURBATION THEOREMS

The results to follow form the theoretical framework from which we can construct error analysis for numerical methods for linear systems of equations.

Theorem 2.3.3-1 : Let A be a square matrix of order n . Then A^m converges to the zero matrix as $m \rightarrow \infty$ if and only if $\rho(A) < 1$.

Theorem 2.3.3-2 (Geometric series) : Let A be a square matrix. If $\rho(A) < 1$, then $(I - A)^{-1}$ exists which can be expressed as a convergent series,

$$(I - A)^{-1} = I + A + A^2 + \dots + A^m + \dots \quad (2.3.3-1)$$

Conversely, if the series in (2.3.3-1) is convergent, then $\rho(A) < 1$.

Theorem 2.3.3-3 : Let A be a square matrix. If for some operator matrix norm, $\|A\| \leq 1$, then $(I - A)^{-1}$ exists and has the geometric series expansion (2.3.3-1). Moreover,

$$\|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|}. \quad (2.3.3-2)$$

We shall illustrate Theorem 2.3.3-3 by considering the invertibility of the matrix

$$A = \begin{bmatrix} 4 & 1 & 0 & 0 & \dots & 0 \\ 1 & 4 & 1 & 0 & & \\ 0 & 1 & 4 & 1 & & \\ \cdot & & \cdot & \cdot & \cdot & \\ \cdot & & \cdot & \cdot & \cdot & \\ \cdot & & & 1 & 4 & 1 \\ 0 & & \dots & 0 & 1 & 4 \end{bmatrix}.$$

We rewrite A as

$$A = 4(I + B)$$

where

$$B = \begin{bmatrix} 0 & \frac{1}{4} & 0 & 0 & \dots & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \dots & \cdot \\ 0 & & \cdot & & & \cdot \\ \cdot & & & \cdot & & \cdot \\ \cdot & & & & \cdot & \\ \cdot & & & & & 0 \\ 0 & & \dots & & & \frac{1}{4} \\ & & & & & \frac{1}{4} \\ & & & & & 0 \end{bmatrix}.$$

By taking the row norm of B , we obtain $\|B\|_{\infty} = \frac{1}{2}$. Thus by Theorem 2.3.3-3, $(I + B)^{-1}$ exists and from (2.3.3-2),

$$\|(I + B)^{-1}\|_{\infty} \leq \frac{1}{1 - \frac{1}{2}} = 2.$$

Therefore $A^{-1} = \frac{1}{4} (I + B)^{-1}$, and $\|A^{-1}\|_{\infty} \leq \frac{1}{2}$.

By using the definition of row norm and inequality (2.3.2-10), we obtain

$$\rho(A) \leq 6 \quad \text{and} \quad \rho(A^{-1}) \leq \frac{1}{2} .$$

Since A is nonsingular, we can easily show that the eigenvalues of A^{-1} are the reciprocal of those of A . Furthermore since A is Hermitian, all eigenvalues of A are real. Hence we have, for all $\lambda \in \sigma(A)$, the bound

$$2 \leq |\lambda| \leq 6$$

where $\sigma(A)$ is the set of all eigenvalues of A .

Theorem 2.3.3-4 : Let A and B be square matrices of the same order. Assume that A is nonsingular and suppose that

$$\|A - B\| < \frac{1}{\|A^{-1}\|} \tag{2.3.3-3}$$

then B is also nonsingular,

$$\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \|A - B\|} \tag{2.3.3-4}$$

and

$$\|A^{-1} - B^{-1}\| \leq \frac{\|A^{-1}\|^2 \|A - B\|}{1 - \|A^{-1}\| \|A - B\|} . \tag{2.3.3-5}$$

This theorem states that all sufficiently close perturbations of a nonsingular matrix are nonsingular.

The proofs of the theorems quoted in this chapter may be found in Atkinson[1978], and Isaacson and Keller[1966] or any other standard books on linear algebra.

CHAPTER 3 SURVEYS OF ODE AND PDE SOLVERS

3.1 INTRODUCTION

This chapter consists of two sections. In the first section we describe the ODE solvers and in the second section we outline the various direct and iterative methods of solving the elliptic PDE problems, which form the basis of the work accounted in chapter 6.

3.2 ODE SOLVERS

The ODE solvers described in this section consist of the standard basic single-step methods and the multistep methods for the special problems $y^{(n)} = f(x,y)$. These methods will be modified to form the GM variations which we shall describe in detail in chapters 4 and 5.

3.2.1 BASIC DEFINITIONS AND THEOREMS

We shall first state the following fairly elementary results which are important tools used throughout numerical analysis.

Theorem 3.2.1-1 (Mean Value Theorem) : Let $f(x)$ be a continuous and differentiable function in $[a,b]$. Then there exists at least one point $\zeta \in [a,b]$ such that

$$f(b) - f(a) = f^{(1)}(\zeta)(b - a). \quad (3.2.1-1)$$

Theorem 3.2.1-2 (Taylor's Theorem) : Let $f(x)$ have $n + 1$ continuous derivatives on $[a,b]$ for some $n \geq 0$, and let $x, x_0 \in [a,b]$. Then

$$f(x) = P_n(x) + R_{n+1}(x) \quad (3.2.1-2)$$

where

$$P_n(x) = \sum_{i=0}^n \left\{ \frac{(x - x_0)^i}{i!} f^{(i)}(x_0) \right\} \quad (3.2.1-3)$$

$$\begin{aligned} R_{n+1}(x) &= \frac{1}{n!} \int_{x_0}^x (x - t)^n f^{(n+1)}(t) dt \\ &= \frac{(x - x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\zeta) \end{aligned} \quad (3.2.1-4)$$

for some ζ between x_0 and x ; $0! = 1$ and $f^{(0)}(x_0) = f(x_0)$.

The Taylor series of a function $f(x)$ can be calculated directly from the definition 3.2.1-3 with as many terms included as desired. However, due to the complexity of the differentiation of some functions, they are often obtained indirectly.

The Taylor's Theorem 3.2.1-2 may be extended to several dimensions. We shall now state the Taylor's Theorem for functions of two variables.

Theorem 3.2.1-3 (Taylor's Theorem of two dimensions) : Let $f(x, y)$ be a given function of two independent variables x and y . Let $L(x_0, y_0; x_1, y_1)$ denote the set of all points (x, y) on the straight line segment joining (x_0, y_0) and (x_1, y_1) . Let (x_0, y_0) and $(x_0 + \zeta, y_0 + \eta)$ be two given points and assume that $f(x, y)$ is $n + 1$ times continuously differentiable for all (x, y) in some neighbourhood of $L(x_0, y_0; x_0 + \zeta, y_0 + \eta)$. Then

$$\begin{aligned} f(x_0 + \zeta, y_0 + \eta) &= f(x_0, y_0) + \sum_{j=1}^n \left\{ \frac{1}{j!} \left[\zeta \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} \right]^j f(x, y) \Big|_{\substack{x=x_0 \\ y=y_0}} \right\} \\ &\quad + \frac{1}{(n+1)!} \left[\zeta \frac{\partial}{\partial x} + \eta \frac{\partial}{\partial y} \right]^{n+1} f(x, y) \Big|_{\substack{x=x_0 + \theta\zeta \\ y=y_0 + \theta\eta}} \end{aligned}$$

for some $0 \leq \theta \leq 1$. The point $(x_0 + \theta\zeta, y_0 + \theta\eta)$ is an arbitrary point on the line $L(x_0, y_0; x_0 + \zeta, y_0 + \eta)$.

Consider the initial value problem for a single-first order ODE

$$y^{(1)} = f(x, y), \quad y(x_0) = y_0. \quad (3.2.1-5)$$

Let $x \in [a, b]$ be an arbitrary point where a and b are finite. We assume that the exact solution of (3.2.1-5) exists and is unique in the closed interval $[a, b]$. Consider the sequence of points defined by $x_n = a + nh$; for $n = 0, 1, 2, \dots$. The parameter h is assumed to be constant and is called the steplength, stepwidth, stepsize or simply the step of the method. We seek an approximate solution denoted by y_n on the discrete set of points $\{x_n; n = 0, 1, \dots, (b-a)/h\}$. Let $f_n = f(x_n, y_n)$, be the corresponding value of f at the discrete point (x_n, y_n) . A computational method for determining the sequence of approximate solutions $\{y_n\}$ is called a linear multistep method of stepnumber k or a linear k -step method if the relationship between y_{n+j} and f_{n+j} , for $j = 0, 1, \dots, k$ is linear. In a multistep method, the computation of y_{n+1} requires the explicit knowledge of some or all of the values of $y_n, y_{n-1}, \dots, y_{n+1-k}$. In addition, multistep methods require a special starting procedure and a special computation may also be required at points where there is a change in stepsize. In contrast, in a single-step method, the value of y_{n+1} can be found if only y_n is known; knowledge of any of the values of y_{n-1}, y_{n-2}, \dots is not required. On the other hand, in single-step methods, every lattice point may be considered as a new starting point, since the starting point does not play any special role.

In the following subsections we shall survey some relevant basic theories of both the single-step and multistep methods which are essential for the discussion in chapters 4 and 5 where the new GM methods are formulated. All the basic work is quoted without proofs, which may be found in standard texts such as

Henrici[1962], Isaacson and Keller[1966], Dahlquist, Bjorck and Anderson[1974] and Butcher[1987].

Theorem 3.2.1-4 states the condition on $f(x,y)$ which guarantee the existence of a unique solution of the initial value problem (3.2.1-5). The proof may be found in Henrici[1962] which we omit.

Theorem 3.2.1-4 : Let $f(x,y)$ be defined and continuous for all points (x,y) in the region $R_1 = \{(x,y); a \leq x \leq b, -\infty < y < \infty\}$, where a and b are finite. Let there exist a constant L_c such that, for every x, y, y^* such that (x,y) and (x,y^*) are both in R_1 ,

$$|f(x,y) - f(x,y^*)| \leq L_c |y - y^*|. \quad (3.2.1-6)$$

Let η be any given number. Then, there exists a unique solution $y(x)$ of the initial value problem (3.2.1-5) such that $y(x)$ is continuous and differentiable for all (x,y) in R_1 .

The condition (3.2.1-6) is known as a Lipschitz condition and L_c is called the Lipschitz constant. Note that in particular, if $f(x,y)$ is continuously differentiable with respect to y for all (x,y) in R_1 , then by the mean value theorem

$$F(x,y) - F(x,y^*) = \frac{\partial f}{\partial y}(x,\bar{y})(y - y^*),$$

where \bar{y} is an interior point of the interval with endpoints y and y^* , and (x,y) and (x,y^*) are both in R_1 .

Thus by choosing

$$L_c = \sup_{(x,y) \in R_1} \left| \frac{\partial f}{\partial y}(x,y) \right| \quad (3.2.1-7)$$

the condition (3.2.1-6) is guaranteed.

In cases where we have a system of m simultaneous first order equations in m dependent variables z_1, z_2, \dots, z_m and assuming that each of these variables satisfies a given condition at the same initial point a , then we have an initial value problem for a first-order system, which may be written as

$$\left. \begin{aligned} \mathbf{z}^{(1)} &= \mathbf{f}(x, \mathbf{z}), \\ \mathbf{z}(a) &= \boldsymbol{\eta}, \end{aligned} \right\} \quad (3.2.1-8)$$

where the respective vectors \mathbf{z} , \mathbf{f} and $\boldsymbol{\eta}$ are as follows:

$$\left. \begin{aligned} \mathbf{z} &= [z_1, z_2, \dots, z_m]^T, \\ \mathbf{f} &= [f_1, f_2, \dots, f_m]^T, \\ \boldsymbol{\eta} &= [\eta_1, \eta_2, \dots, \eta_m]^T. \end{aligned} \right\} \quad (3.2.1-9)$$

If the z_i $i = 1, 2, \dots, m$ satisfy the conditions at different points a, b, c, \dots of x , then we have a multi-point boundary-value problem; if there are only two different points of x , then we have a two-point boundary-value problem.

Theorem 3.2.1-4 may then have to be generalized to cater for the necessary conditions of the unique existence of the solution of (3.2.1-8). Thus the region R_1 reads as

$$R_1 = \{(x, \mathbf{z}); a \leq x \leq b, -\infty < z_i < \infty; i = 1, 2, \dots, m\}, \quad (3.2.1-10)$$

and the condition (3.2.1-6) becomes

$$\|\mathbf{f}(x, \mathbf{z}) - \mathbf{f}(x, \mathbf{z}^*)\| \leq L_c \|\mathbf{z} - \mathbf{z}^*\|, \quad (3.2.1-11)$$

where (x, \mathbf{z}) and (x, \mathbf{z}^*) are interior points of R_1 and $\|\cdot\|$ denotes a vector norm. In the case when each of the functions $f_i(x, z_1, z_2, \dots, z_m)$; $i = 1, 2, \dots, m$ is continuously differentiable with respect to each of the z_i $i = 1, 2, \dots, m$ then we may choose

$$L_c = \sup_{(x, z) \in R_1} \left\| \frac{\partial f}{\partial z} \right\|, \quad (3.2.1-12)$$

where $\frac{\partial f}{\partial z}$ is known as the Jacobian of f with respect to z and

$$\frac{\partial f}{\partial z} = \left(\frac{\partial f_i}{\partial z_j} (x, z_1, \dots, z_m) \right)_{m \times m} \quad (3.2.1-13)$$

is an $m \times m$ matrix. The norm $\| \cdot \|$ in (3.2.1-12) is the subordinate matrix norm to the vector norm used in (3.2.1-11) (see Mitchell[1969]).

3.2.2 SINGLE-STEP AND RK METHODS

Single-step methods for solving (3.2.1-5) 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 best known single-step methods are the RK methods while the simplest single-step method is based on using the Taylor series.

Let $y(x)$ be the solution of (3.2.1-5) and be $r + 1$ times continuously differentiable. By expanding $y(x_0+h)$ about x_0 using Taylor expansion, we obtain

$$y(x_0+h) = y(x_0) + hy^{(1)}(x_0) + \dots + \frac{h^r}{r!} y^{(r)}(x_0) + \frac{h^{r+1}}{(r+1)!} y^{(r+1)}(\zeta), \quad (3.2.2-1)$$

for some $x_0 \leq \zeta \leq x_0 + h$.

The Taylor series method is obtained by neglecting the remainder term in (3.2.2-1). Thus an approximation for $y(x_0+h)$ may be obtained provided we can calculate $y^{(2)}(x_0), \dots, y^{(r)}(x_0)$. This can be done by differentiating $y^{(1)}(x) = f(x, y(x))$ to obtain $y^{(2)}(x) = f_x(x, y(x)) + f(x, y(x))f_y(x, y(x))$ and higher order derivatives of $y(x)$. However, in most cases we avoid the

differentiation of $f(x,y)$. Therefore we turn to the RK methods which are closely related to the Taylor series expansion of $y(x)$ in (3.2.2-1), but do not involve the differentiation of $f(x,y)$.

We shall write the general RK methods in the form

$$y_{n+1} = y_n + h\Phi(x_n, y_n, h; f) \quad (3.2.2-2)$$

for $n \geq 0$. In the case of $\Phi(x, y, h; f) = f(x, y)$, we obtain a method of the form

$$y_{n+1} = y_n + hf(x_n, y_n); \quad n \geq 0 \quad (3.2.2-3)$$

which is called the Euler method.

We assume the arithmetic to be exact and neglect rounding errors, because they are often negligible. With these hypotheses we define the following quantity.

Definition 3.2.2-1 : The local truncation error, T_{n+1} for (3.2.2-2) at x_{n+1} is defined by

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\Phi(x_n, y(x_n), h; f), \quad (3.2.2-4)$$

for $n \geq 0$.

The local truncation error, T_{n+1} measures how well the exact solution fits the formula (3.2.2-2).

Now define τ_{n+1} such that

$$T_{n+1} = h\tau_{n+1}. \quad (3.2.2-5)$$

Therefore by combining (3.2.2-4) and (3.2.2-5), we obtain

$$y(x_{n+1}) = y(x_n) + h\Phi(x_n, y(x_n), h; f) + h\tau_{n+1}, \quad (3.2.2-6)$$

for $n \geq 0$.

Now, in order to obtain convergence of (3.2.2-2), we need to have $\tau_{n+1} \rightarrow 0$ as $h \rightarrow 0$. Since

$$\tau_{n+1} = \frac{y(x_{n+1}) - y(x_n)}{h} - \Phi(x_n, y(x_n), h; f),$$

therefore we require that

$$\Phi(x, y(x), h; f) \rightarrow f(x, y(x)) \text{ as } h \rightarrow 0.$$

By defining,

$$\delta(h) = \max_{\substack{x \in [a, b] \\ y \in (-\infty, \infty)}} |f(x, y) - \Phi(x, y, h; f)|$$

and assume that $\delta(h) \rightarrow 0$ as $h \rightarrow 0$, we have the consistency condition for (3.2.2-2) defined as follows:

Definition 3.2.2-2 : A single-step method (3.2.2-2) is said to be consistent with the differential equation $y^{(1)} = f(x, y(x))$ if

$$\lim_{h \rightarrow 0} \delta(h) = 0. \quad (3.2.2-7)$$

For practical purposes the total error between the computed approximation and the exact solution, after several integration steps, is of interest. Hence we define the following quantity.

Definition 3.2.2-3 : The global truncation error g_n at x_n is defined as

$$g_n = y(x_n) - y_n. \quad (3.2.2-8)$$

In order to be able to estimate the global error g_n , we assume that the function $\Phi(x, y, h; f)$ satisfies a Lipschitz condition, with respect to the dependent variable y , for all $x_0 \in [a, b]$, $y \in (-\infty, \infty)$ and sufficiently small $h > 0$,

$$|\Phi(x, y, h; f) - \Phi(x, z, h; f)| < L_c |y - z| \quad (3.2.2-9)$$

where $0 < L_c < \infty$.

From the definition of the local truncation error (3.2.2-1), it follows that

$$y(x_{n+1}) = y(x_n) + h\Phi(x_n, y(x_n), h; f) + T_{n+1}. \quad (3.2.2-10)$$

Now subtract (3.2.2-2) from (3.2.2-10), we obtain

$$g_{n+1} = g_n + h[\Phi(x_n, y(x_n), h; f) - \Phi(x_n, y_n, h; f)] + T_{n+1}.$$

By assumption (3.2.2-9), we obtain the following estimate

$$\begin{aligned} |g_{n+1}| &\leq |g_n| + h|\Phi(x_n, y(x_n), h; f) - \Phi(x_n, y_n, h; f)| + |T_{n+1}| \\ &\leq |g_n| + hL_c|y(x_n) - y_n| + |T_{n+1}| \\ &= [1 + hL_c]|g_n| + |T_{n+1}|. \end{aligned} \quad (3.2.2-11)$$

If we assume that the absolute value of the local truncation error is bounded, say,

$$\max_n |T_n| \leq D, \quad (3.2.2-12)$$

then the absolute value of the global truncation error g_n for $n = 0, 1, 2, \dots$, satisfies the inequality

$$|g_{n+1}| \leq [1 + hL_c]|g_n| + D. \quad (3.2.2-13)$$

By repeated application of (3.2.2-13), we can show that

$$|g_n| \leq \frac{(1 + hL_c)^n - 1}{hL_c} D + (1 + hL_c)^n |g_0|. \quad (3.2.2-14)$$

Furthermore, from the fact that the function e^t is convex, so that the tangent at $t = 0$ is below the curve and $1 + t \leq e^t$ for all real values of t , then it follows that

$$(1 + hL_c) \leq e^{hL_c} \text{ and } (1 + hL_c)^n \leq e^{nhL_c}. \quad (3.2.2-15)$$

From (3.2.2-14) and (3.2.2-15), we deduce the following results.

Lemma 3.2.2-1 : If the global error satisfies (3.2.2-13), then

$$|g_n| \leq \frac{(1+hL_c)^n - 1}{hL_c} D + (1+hL_c)^n |g_0| \leq \frac{D}{hL_c} (e^{nhL_c} - 1) + e^{nhL_c} |g_0|. \quad (3.2.2-16)$$

Consequently, from Lemma 3.2.2-1 and since the global truncation error satisfies $g_0 = y(x_0) - y_0 = 0$, we deduce the following theorem.

Theorem 3.2.2-1 : The global truncation error g_n , at the fixed abscissa $x_n = x_0 + nh$ is bounded by

$$|g_n| \leq \frac{D}{hL_c} (e^{nhL_c} - 1) \leq \frac{D}{hL_c} e^{nhL_c}. \quad (3.2.2-17)$$

From (3.2.2-17), we note that there are two decisive quantities to be considered; namely the Lipschitz constant L_c , of the function $\Phi(x, y, h; f)$ and the local truncation error T_{n+1} , in the qualitative judgement of a single-step method. Thus, if we assume that the function $f(x, y)$ and the solution $y(x)$ are sufficiently many times continuously differentiable, the local truncation error may be determined by means of Taylor series.

We further note that if the condition (3.2.2-7) is also satisfied in Theorem 3.2.2-1, then the numerical solution $\{y_n\}$ converges to $y(x)$. For proof of Theorem (3.2.2-1), see Atkinson[1978], pg.374.

Definition 3.2.2-4 : Let $E_n(f)$ be an exact error formula, and let $\tilde{E}_n(f)$ be an estimate of it. We say that $\tilde{E}_n(f)$ is an asymptotic error estimate for $E_n(f)$ if

$$\lim_{n \rightarrow \infty} \frac{\tilde{E}_n(f)}{E_n(f)} = 1$$

or equivalently,

$$\lim_{n \rightarrow \infty} \frac{E_n(f) - \tilde{E}_n(f)}{E_n(f)} = 0.$$

Next, we define a special notation which we shall be using very frequently in the discussion throughout the thesis.

Definition 3.2.2-5 : Suppose $B(x,h)$ is a function defined for $x \in [x_0, b]$ and for all sufficiently small h , then the notation

$$B(x,h) = O(h^p)$$

for some $p > 0$ means that there is a constant c such that

$$|B(x,h)| \leq ch^p$$

for all $x \in [x_0, b]$ and for all sufficiently small h . If B depends on h only, the same kind of bound is implied.

Definition 3.2.2-6 : The method (3.2.2-2) is said to have order p if p is the largest integer for which $y(x+h) - y(x) - h\Phi(x, y(x), h; f) = O(h^{p+1})$ holds, where $y(x)$ is the exact solution of the initial-value problem (3.2.1-5).

Now as a consequence of Theorem 3.2.2-1, we have the following result.

Corollary 3.2.2-1 : If the single-step method (3.2.2-2) has a local truncation error $T_{n+1} = O(h^{p+1})$, then the rate of convergence of $\{y_n\}$ to $y(x_n)$ is $O(h^p)$.

The asymptotic error formula for (3.2.2-2) may be derived by assuming that

$$T_{n+1} = \psi(x_n)h^{p+1} + O(h^{p+2}) \quad (3.2.2-18)$$

with $\psi(x)$ determined by $y(x)$ and $f(x, y(x))$.

In chapter 4, we shall define a new RK method using the modified form of (3.2.2-2) where $\Phi(x_n, y_n, h; f)$ is nonlinear which we shall call the RK-GM methods. This work was first studied by Sanugi[1986]. We shall then extend this and show that the classical RK methods as well as those of Sanugi[1986] are special cases of these nonlinear forms.

3.2.3 STABILITY ANALYSIS FOR EXPLICIT RK METHODS

The first analysis of instability phenomena and stepsize restrictions for hyperbolic equations was reported by Courant, Friedrichs and Lewy in 1928. It was later followed by many authors independently, notably Guillou and Lago in 1961.

a) EULER METHOD

Suppose $\phi(x)$ is a smooth function of $y^{(1)} = f(x, y)$. Now linearize f in its neighbourhood as follows:

$$y^{(1)}(x) = f(x, \phi(x)) + f_y(x, \phi(x)) [y(x) - \phi(x)] + \dots \quad (3.2.3-1)$$

By rearranging the terms, we have

$$y^{(1)}(x) - f(x, \phi(x)) = f_y(x, \phi(x)) [y(x) - \phi(x)] + \dots \quad (3.2.3-2)$$

By letting $\bar{y}(x) = y(x) - \phi(x)$, (3.2.3-2) becomes

$$\begin{aligned} \bar{y}^{(1)}(x) &= f_y(x, \phi(x))\bar{y}(x) + \dots \\ &= J(x)\bar{y}(x) + \dots \end{aligned} \quad (3.2.3-3)$$

By neglecting the error terms, we may obtain as a first approximation by treating the Jacobian $J(x)$ as constant and arrive at a general representation given as

$$y^{(1)} = Jy. \quad (3.2.3-4)$$

Now applying Euler method to (3.2.3-4), we obtain

$$y_{n+1} = R(hJ)y_n \quad (3.2.3-5)$$

with

$$R(z) = 1 + z. \quad (3.2.3-6)$$

The plot of (3.2.3-6) is a circle of radius 1 and centre $(-1,0)$.

b) EXPLICIT RK METHODS

Following the notation of Hairer et al.[1987], we redefine the RK method (3.2.2-2) as follows:

Let k_{it} be such that

$$k_{it} = f_t(v_{i1}, \dots, v_{im}), \quad (3.2.3-7)$$

for $i = 1, \dots, s$. Then the RK method is defined by

$$y_{n+1t} = y_{nt} + h \sum_{j=1}^s \{b_j f_t(v_{j1}, \dots, v_{jm})\} \quad (3.2.3-8)$$

with

$$v_{it} = y_{nt} + h \sum_{j=1}^{i-1} \{a_{ij} f_t(v_{j1}, \dots, v_{jm})\} \quad (3.2.3-9)$$

for $i = 1, 2, \dots, s$.

On applying the RK method (3.2.3-8) to (3.2.3-4), we have

$$v_i = y_n + hJ \sum_{j=1}^s a_{ij} v_j \quad (3.2.3-10)$$

and

$$y_{n+1} = y_n + hJ \sum_{j=1}^s b_j v_j. \quad (3.2.3-11)$$

On combining (3.2.3-11) and (3.2.3-10), we obtain

$$y_{n+1} = R(hJ)y_n \quad (3.2.3-12)$$

where

$$R(z) = 1 + z \sum_j b_j + z^2 \sum_{j,k} b_j a_{jk} + z^3 \sum_{j,k,l} \{b_j a_{jk} a_{kl}\} \quad (3.2.3-13)$$

is a polynomial of degree $\leq s$ with $z = hJ$. Thus we have the following definition.

Definition 3.2.3-1 : The polynomial function $R(z)$ of (3.2.3-13) is called the stability function of the method (3.2.2-2). The set

$$S = \{z \in \mathbb{C} ; |R(z)| \leq 1\} \quad (3.2.3-14)$$

is called the stability region of the method (3.2.2-2).

The stability function $R(z)$ may be interpreted as the numerical solution after one step of the Dalquist test equation,

$$y^{(1)} = \lambda y, \quad y_0 = 1, \quad z = h\lambda. \quad (3.2.3-15)$$

The theorem below relates the stability region and the order of the RK method.

Theorem 3.2.3-1 : If the RK method is of order p , then

$$R(z) = \sum_{j=0}^p \frac{z^j}{j!} + O(z^{p+1}). \quad (3.2.3-16)$$

Its proof follows directly by considering the difference between the exact and numerical solution of (3.2.3-15).

The stability regions of the explicit RK methods with $s = 1, 2, 3, 4$ are plotted in Figure(4.3.4b) where they are compared with those of the RK-GM methods.

3.2.4 LINEAR MULTISTEP METHODS FOR THE SPECIAL CLASS OF ODE PROBLEMS

In this section and in chapter 5, we shall be concentrating on the study of the methods for problems of the type

$$y^{(2)} = f(x, y), \quad (3.2.4-1)$$

that is, no derivatives appear in the right-hand side of the differential equation. Equations of the form $y^{(n)} = f(x, y)$; for any integer $n \geq 2$ belong to a special class of differential equations. Such equations or systems of such equations occur frequently, for example, in mechanical problems without dissipation. Although we can reformulate (3.2.4-1) into a system of first-order equations, it may seem unnatural to introduce the first derivatives when their values are irrelevant to the problem. In fact, astronomers have for more than a century been integrating such problems using methods which work without first derivatives.

3.2.5 GENERAL OPERATORS FOR SPECIAL SECOND-ORDER EQUATIONS

Consider a linear k-step method of the form

$$\sum_{j=0}^k \alpha_j y_{n+j} = h^2 \sum_{j=0}^k \beta_j f_{n+j}, \quad k \geq 2 \quad (3.2.5-1)$$

where $\alpha_k \neq 0$, and α_0 and β_0 do not vanish simultaneously. Without loss of generality, we may assume that $\alpha_k = 1$. If $\beta_k = 0$, then (3.2.5-1) is called an explicit k-step method, otherwise it is known as an implicit k-step method. The direct application of (3.2.5-1) to problems of the form (3.2.4-1) finds theoretical evidence in Ash[1969].

Now associated to the linear multistep method (3.2.5-1), we define the linear difference operator L as follows:

$$L[y(x);h] = \sum_{j=0}^k [\alpha_j y(x+jh) - h^2 \beta_j y^{(2)}(x+jh)], \quad (3.2.5-2)$$

where $y(x)$ is an arbitrary and continuously differentiable function on an interval $[a,b]$. Assume that $y(x)$ has sufficiently many higher derivatives. Then by Taylor expansion about x , we have

$$L[y(x);h] = \sum_{i=0}^q h^i C_i y^{(i)}(x) + \dots, \quad (3.2.5-3)$$

where the coefficients C_i ; $i = 0, 1, \dots$ are constants and independent of the stepsize h and the function $y(x)$. By simple manipulation, we obtain these coefficients as listed below.

$$\left. \begin{aligned} C_0 &= \sum_{i=0}^k \alpha_i, & C_1 &= \sum_{i=1}^k i \alpha_i, \\ C_2 &= \frac{1}{2!} \sum_{i=1}^k i^2 \alpha_i - \sum_{i=0}^k \beta_i, \\ &\vdots \\ C_q &= \frac{1}{q!} \sum_{i=1}^k i^q \alpha_i - \frac{1}{(q-2)!} \sum_{i=1}^k i^{q-2} \beta_i, \end{aligned} \right\} \quad (3.2.5-4)$$

for $q = 3, 4, \dots$

Definition 3.2.5-1 : The difference operator (3.2.5-2) and the associated multistep method (3.2.5-1) are said to be of order p if, in (3.2.5-3); $C_q = 0$, for $q = 0, 1, \dots, p+1$; and $C_{p+2} \neq 0$.

From the definition above, it is clear that only the first of the nonvanishing coefficients in the expression (3.2.5-3), namely C_{p+2} , has any significance. Thus we define C_{p+2} as the error constant.

Definition 3.2.5-2 : The local truncation error T_{n+k} at x_{n+k} of the method (3.2.5-1) is defined as the expression $L[y(x_n);h]$ given by (3.2.5-2), when $y(x)$ is the exact solution of the problem (3.2.4-1).

Consider the application of (3.2.5-1) to yield y_{n+k} under the assumption that no previous truncation errors have been made. In particular, assume that $y_{n+j} = y(x_{n+j})$ $j = 0, 1, \dots, k-1$. From (3.2.5-2), we obtain

$$\begin{aligned} \sum_{j=0}^k \alpha_j y(x_n + jh) &= h^2 \sum_{j=0}^k \beta_j y^{(2)}(x_n + jh) + L[y(x_n); h] \\ &= h^2 \sum_{j=0}^k \beta_j f(x_n + jh, y(x_n + jh)) + L[y(x_n); h], \end{aligned} \quad (3.2.5-5)$$

since in this context, $y(x)$ is assumed to be the exact solution of (3.2.4-1). The value of y_{n+k} given by (3.2.5-1) satisfies

$$\sum_{j=0}^k \alpha_j y_{n+j} = h^2 \sum_{j=0}^k \beta_j f(x_{n+j}, y_{n+j}). \quad (3.2.5-6)$$

Subtract (3.2.5-6) from (3.2.5-5) and use the assumption stated above, to obtain

$$\begin{aligned} y(x_{n+k}) - y_{n+k} &= h^2 \beta_k [f(x_{n+k}, y(x_{n+k})) - f(x_{n+k}, y_{n+k})] + L[y(x_n); h]. \end{aligned}$$

By the mean value theorem,

$$f(x_{n+k}, y(x_{n+k})) - f(x_{n+k}, y_{n+k}) = [y(x_{n+k}) - y_{n+k}] \frac{\partial f(x_{n+k}, \eta_{n+k})}{\partial y}$$

where η_{n+k} is an interior point of the interval with end-points y_{n+k} and $y(x_{n+k})$. Therefore, we obtain

$$T_{n+k} = \left[1 - h^2 \beta_k \frac{\partial f(x_{n+k}, \eta_{n+k})}{\partial y} \right] [y(x_{n+k}) - y_{n+k}]. \quad (3.2.5-7)$$

Hence the local truncation error of an explicit method is the difference between the exact solution and the numerical solution generated by the method under the assumption stated. On the other hand, for an implicit method, the local truncation error is approximately proportional to the difference between the two solutions mentioned. Next, if we make a further assumption that

the exact solution $y(x)$ has continuous derivatives of sufficiently high order, then, both the explicit and the implicit methods satisfy the following result,

$$y(x_{n+k}) - Y_{n+k} = C_{p+2}h^{p+2}y^{(p+2)}(x_n) + O(h^{p+3}), \quad (3.2.5-8)$$

where p is the order of the method. The term $C_{p+2}h^{p+2}y^{(p+2)}(x_n)$ is the principal local truncation error at the point x_n .

We note that the results (3.2.5-7) and (3.2.5-8) are true only under the assumption that no previous truncation errors have been made, which could be unrealistic. Thus if we make no such assumption, then the error $g_{n+k} = y(x_{n+k}) - Y_{n+k}$ is the global or accumulated truncation error. This error involves all the truncation errors made at each application of the method. It is this error which should tend to zero as $h \rightarrow 0$, $n \rightarrow \infty$ for $nh = x_n - a$ remains fixed as a criterion for convergence.

The coefficients C_i ; $i = 0, 1, 2, \dots$ defined in (3.2.5-4) can be used to derive a linear multistep method of any structure and order. For a given k , the parameters α_i and β_i can be determined such that the order is optimal. By doing so we prescribe the conditions for the desired structure of the multistep method.

3.2.6 BASIC PROPERTIES OF LINEAR MULTISTEP METHODS

An important basic property of a linear multistep method to be of any value is that the sequence of solutions $\{y_n\}$ generated by the method converges to the exact solution $y(x)$ as the stepsize h tends to zero. Thus we have the following definition of convergence (Lambert[1973]).

Definition 3.2.6-1 : The linear multistep method (3.2.5-1) is said to be convergent if, for all initial value problems (3.2.4-1) subject to the hypothesis of Theorem 3.2.1-4, we have that

$$\lim_{\substack{h \rightarrow 0 \\ nh = x - a}} y_n = y(x_n) \quad (3.2.6-1)$$

holds for all $x \in [a, b]$, and for all solutions $\{y_n\}$ of the difference equation (3.2.5-1) satisfying starting conditions $y_\mu = \eta_\mu(h)$ for which $\lim_{h \rightarrow 0} \eta_\mu(h) = \eta$ and

$$\lim_{h \rightarrow 0} \frac{\eta_\mu(h) - \eta_0(h)}{h} = \hat{\eta}; \text{ for } \mu = 0, 1, 2, \dots, k-1.$$

Definition 3.2.6-2 : The linear multistep method (3.2.5-1) is said to be consistent if it has order $p \geq 1$.

From (3.2.5-4) it follows that the method (3.2.5-1) is consistent if and only if

$$\left. \begin{aligned} \sum_{i=0}^k \alpha_i &= 0; \quad \sum_{i=1}^k i\alpha_i = 0; \\ \frac{1}{2!} \sum_{i=1}^k i^2\alpha_i &= \sum_{i=0}^k \beta_i. \end{aligned} \right\} \quad (3.2.6-2)$$

Let

$$\lim_{\substack{h \rightarrow 0 \\ n \rightarrow \infty \\ nh = x - a}} y_n = y(x)$$

and for $i = 0, 1, \dots, k$,

$$\lim_{\substack{h \rightarrow 0 \\ n \rightarrow \infty \\ nh = x - a}} y_{n+i} = y(x).$$

Write for $i = 0, 1, \dots, k$,

$$y(x) = y_{n+i} + \theta_{in}(h);$$

where

$$\lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} \theta_{in}(h) = 0.$$

Hence

$$\sum_{i=0}^k \alpha_i y(x) = \sum_{i=0}^k \alpha_i y_{n+i} + \sum_{i=0}^k \alpha_i \theta_{in}(h),$$

$$y(x) \sum_{i=0}^k \alpha_i = h^2 \sum_{i=0}^k \beta_i f_{n+i} + \sum_{i=0}^k \alpha_i \theta_{in}(h), \quad (3.2.6-3)$$

by using (3.2.5-1).

Now as $h \rightarrow 0$ and $n \rightarrow \infty$, both terms on the right-hand side of (3.2.6-3) vanish, whereas the left-hand side term is unaffected. Therefore it must be zero. Since $y(x)$ is in general not zero, therefore $\sum_{i=0}^k \alpha_i$ must be zero, which is the first condition of (3.2.6-2). We can easily show that the second condition of (3.2.6-2) follows directly from the situation of the problem that the right-hand side of (3.2.4-1) is independent of the first derivative.

Next, under the limiting conditions of $h \rightarrow 0$ and $n \rightarrow \infty$ we have for $i = 1, 2, \dots, k$,

$$\lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} \frac{y_{n+i} - 2y_n + y_{n-i}}{(ih)^2} = y^{(2)}(x);$$

or

$$y_{n+i} - 2y_n + y_{n-i} = (ih)^2 y^{(2)}(x) + (ih)^2 \kappa_{in}(h);$$

where $\lim_{\substack{n \rightarrow \infty \\ h \rightarrow 0}} \kappa_{in}(h) = 0$.

Hence

$$\sum_{i=0}^k \alpha_i y_{n+i} - 2y_n \sum_{i=0}^k \alpha_i + \sum_{i=0}^k \alpha_i y_{n-i}$$

$$= h^2 y^{(2)}(x) \sum_{i=0}^k i^2 \alpha_i + h^2 \sum_{i=0}^k i^2 \alpha_i \kappa_{in}(h).$$

Since $\sum_{i=0}^k \alpha_i = 0$, we have on dividing by $2h^2$,

$$\frac{1}{2} \sum_{i=0}^k \beta_i [f_{n+i} + f_{n-i}] = \frac{1}{2} y^{(2)}(x) \sum_{i=0}^k i^2 \alpha_i + \frac{1}{2} \sum_{i=0}^k i^2 \alpha_i \kappa_{in}(h).$$

But $\lim_{\substack{h \rightarrow 0 \\ n \rightarrow \infty}} f_{n+1} = f(x, y(x))$ and $\lim_{\substack{h \rightarrow 0 \\ n \rightarrow \infty}} \kappa_{1n}(h) = 0$.

Hence we obtain

$$f(x, y(x)) \sum_{i=0}^k \beta_i = y^{(2)}(x) \frac{1}{2!} \sum_{i=0}^k i^2 \alpha_i. \quad (3.2.6-4)$$

Therefore $y(x)$ satisfies the differential equation (3.2.4-1) if and only if $\sum_{i=0}^k \beta_i = \frac{1}{2!} \sum_{i=1}^k i^2 \alpha_i$. Thus we have

shown that a convergent linear multistep method is necessarily consistent. However, consistency alone is not sufficient for convergence (Lambert[1973]). By defining the first and second characteristic polynomials

$$\rho(\zeta) = \sum_{i=0}^k \alpha_i \zeta^i, \quad \sigma(\zeta) = \sum_{i=0}^k \beta_i \zeta^i,$$

we can easily verify that method(3.2.5-1) is consistent if and only if

$$\rho(1) = \rho^{(1)}(1) = 0, \quad \rho^{(2)}(1) = 2\sigma(1).$$

We should note that for a consistent method to be meaningful, $k \geq 2$. The polynomial $\rho(\zeta)$ associated with the consistent method has a double root at 1; which is called the principal root while the other roots are spurious. Thus we have zero-stability defined as follows:

Definition 3.2.6-3 : The linear multistep method (3.2.5-1) is said to be zero-stable if no root of the first characteristic polynomial $\rho(\zeta)$ has modulus greater than one, and if every root of modulus one has multiplicity not greater than two.

We now state the theorem which gives the necessary and sufficient conditions for convergence of method (3.2.5-1).

Theorem 3.2.6-1 : The necessary and sufficient conditions for a linear multistep method to be convergent are that it be consistent and zero-stable.

Another important theorem which limits the order of a zero-stable linear k-step method depending on whether k is odd or even is stated as follows:

Theorem 3.2.6-2 : No zero-stable linear multistep method of stepnumber k can have order exceeding k + 1 when k is odd, or exceeding k + 2 when k is even.

The proofs of Theorem 3.2.6-1 and Theorem 3.2.6-2 are given in Henrici[1962].

Definition 3.2.6-4 : A zero-stable linear k-step method which has order k + 2 is called an optimal method.

Thus from the results above, we can conclude that a necessary condition for optimality is that k be even and that all the roots of $\rho(\zeta)$ have modulus unity .

3.2.7 STÖRMER-COWELL METHODS

In the pre computer-oriented methods, the right-hand side of a linear multistep method is written in terms of a power series in a difference operator. A typical example is

$$y_{n+1} - y_n = h(1 - \frac{1}{2} \nabla - \frac{1}{12} \nabla^2 - \frac{1}{24} \nabla^3 - \dots) f_{n+1}. \quad (3.2.7-1)$$

By truncating the series (3.2.7-1), we can obtain the following methods

$$y_{n+1} - y_n = \frac{1}{2} h(f_{n+1} + f_n),$$

and

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

The existence of formulae similar to (3.2.7-1) has resulted in family names associated to classes of linear multistep methods, of different stepnumber, in which the first characteristic polynomial $\rho(\zeta)$ is common. Thus the methods of the form (3.2.4-1), which equate $y_{n+2} - 2y_{n+1} + y_n$ to power series and with first characteristic polynomial $\rho(\zeta) = \zeta^k - 2\zeta^{k-1} + \zeta^{k-2}$ are often known as Störmer-Cowell methods. The most well known such method is the optimal two-step method of Numerov given as

$$y_{n+2} - 2y_{n+1} + y_n = \frac{h^2}{12} (f_{n+2} + 10f_{n+1} + f_n). \quad (3.2.7-2)$$

In chapter 5 we shall modify (3.2.7-2) to obtain its GM version.

3.2.8 BOUNDS FOR THE LOCAL AND GLOBAL TRUNCATION ERRORS OF METHODS (3.2.5-1)

Let

$$Y = \max_{x \in [a,b]} |y^{(p+1)}| \quad (3.2.8-1)$$

and

$$G = |C_{p+2}|, \quad (3.2.8-2)$$

where C_{p+2} is the error constant of (3.2.5-1).

A bound for the global error when (3.2.5-1) is applied to (3.2.4-1) is given in Henrici[1962] page 314. This bound reflects the different influences of starting error, local truncation error and local round-off error. However, if the local truncation error is bounded by GYh^{p+2} , then the global truncation error is proportionately bounded by GYh^p (Lambert[1973]).

3.2.9 ABSOLUTE AND RELATIVE STABILITY OF LINEAR MULTISTEP METHODS

The theory of weak stability attempts to provide a criterion, involving h , for the global error to be damped out as the computation progresses. Thus it

enables one to choose h small enough for the criterion to be fulfilled while the local truncation error is kept acceptably small. By assuming that $\frac{\partial f}{\partial y} = \lambda$ constant, and the local error equals ϕ_c , a constant, then the linearized error equation is given by

$$\sum_{i=0}^k (\alpha_i - h^2 \lambda \beta_i) \tilde{e}_{n+i} = \phi_c, \quad (3.2.9-1)$$

where $\tilde{e}_n = y(x_n) - \tilde{y}_n$, for (\tilde{y}_n) is the sequence of solutions of (3.2.5-1) where the round-off error has been included. The general solution (see Lambert[1973], page 268) of (3.2.9-1) is

$$\tilde{e}_n = \sum_{s=1}^k d_s r_s^n - \phi_c / (h^2 \lambda \sum_{i=0}^k \beta_i),$$

where d_s are arbitrary constants and r_s are the roots, assumed constant, of the stability polynomial

$$\pi(r, \bar{h}) = \rho(r) - \bar{h} \sigma(r), \quad (3.2.9-2)$$

with $\bar{h} = h^2 \lambda$.

Let r_1 and r_2 be the two roots of (3.2.9-2) that tend to the double principal root $\xi_1 = +1$. Then (Lambert[1973]),

$$r_1 = \exp(h\lambda^{1/2}) + O(h^{p+2}) \text{ and } r_2 = \exp(-h\lambda^{1/2}) + O(h^{p+2}).$$

Since for many methods r_1 and r_2 lie on the unit circle when \bar{h} is small and negative, we have the following definition of absolute and relative stability for the linear multistep methods for the special second order problems (Lambert[1973]).

Definition 3.2.9-1 : The linear multistep method (3.2.5-1) is said to be absolutely stable for a given \bar{h} if, for that \bar{h} , all the roots r_s of (3.2.9-2) satisfy $|r_s| \leq 1$, $s = 1, 2, \dots, k$; and to be relatively stable if, for that \bar{h} , $|r_s| \leq \min(|r_1|, |r_2|)$, $s = 3, 4, \dots, k$. An

interval $[\alpha, \beta]$ of the real line is said to be an interval of absolute or relative stability if the method is absolutely or relatively stable for all $\bar{h} \in [\alpha, \beta]$.

We also have that every zero-stable consistent linear multistep method of class (3.2.5-1) is absolutely unstable for small positive \bar{h} .

For a system of equations $\mathbf{y}^{(2)} = \mathbf{f}(x, \mathbf{y})$, we then consider λ the ^{largest} eigenvalue, assumed constant, of the Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}$, which may be complex. The intervals of the absolute or relative stability discussed in the previous paragraph are replaced by regions of absolute or relative stability.

3.3 ELLIPTIC PDE SOLVERS

Many physical or engineering systems can be described by means of functions of several independent variables which from natural energy principles must satisfy certain PDEs. There is an enormous variety of PDEs and systems of PDEs that occur in these applications. Their appropriate numerical solution often requires special procedures. In this section, we shall survey some of the current elliptic PDE solvers and highlight some of their characteristics. We shall restrict the discussion to the treatment of the second-order PDEs for an unknown function with two independent variables of the form

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G \quad (3.3-1)$$

where $u(x,y)$ is the function that we are looking for in the region $R_1 \subset \mathbb{R}^2$ and satisfies (3.3-1). The given coefficients A, B, C, D, E, F and G in (3.3-1) may be piecewise continuous functions of x and y .

3.3.1 CLASSIFICATION OF PDES AND TYPES OF ELLIPTIC PROBLEMS

Analogously to the classification of conic sections

$$Ax^2 + 2Bxy + Cy^2 + Dx + Ey + F = 0 \quad (3.3.1-1)$$

the PDEs (3.3-1) are divided into three classes according to the Definition 3.3.1-1.

Definition 3.3.1-1 : In a region R_1 , a second-order partial differential equation of the form (3.3-1) with $A^2 + B^2 + C^2 \neq 0$ is called

- (1) elliptic if $AC - B^2 > 0$ for all $(x,y) \in R_1$,
- (2) hyperbolic if $AC - B^2 < 0$ for all $(x,y) \in R_1$,
- (3) parabolic if $AC - B^2 = 0$ for all $(x,y) \in R_1$.

However the classification defined by Definition 3.3.1-1 above depends, in general, on the region of the (x,y) plane under consideration. The differential equation

$$xu_{xx}(x, y) + u_{yy}(x, y) = 0, \quad (3.3.1-1a)$$

for instance, is elliptic for $x > 0$, hyperbolic for $x < 0$ and parabolic for $x = 0$.

The well known special cases of elliptic differential equations are

$$(1) \text{ Laplace equation: } \nabla^2 u = 0, \quad (3.3.1-2a)$$

$$(2) \text{ Poisson equation: } \nabla^2 u = f(x, y), \quad (3.3.1-2b)$$

$$(2) \text{ Helmholtz equation: } \nabla^2 u = r(x, y)u, \quad (3.3.1-2b)$$

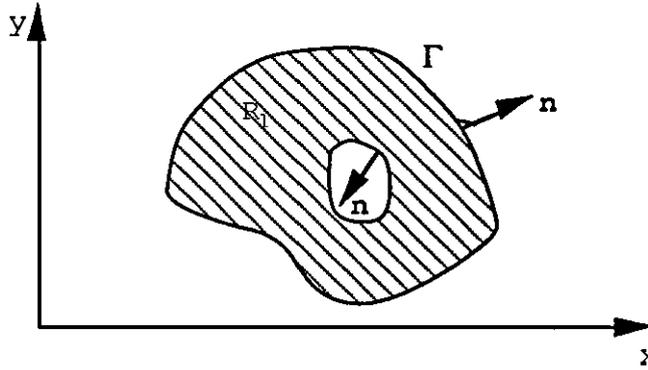
$$\text{where } \nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}.$$

The Laplace equation occurs in problems of elasticity and hydrodynamics. The solution of the Poisson equation can describe the static temperature distribution in a homogeneous medium or the stress in some torsion problems. The Helmholtz equation or sometimes called the 'reduced wave' equation arises in the theories of sound, electromagnetic waves and tidal waves.

In order to define the desired solution of an elliptic differential equation uniquely, certain boundary conditions must be imposed on the boundary of the region R_1 . To simplify the problem, we assume that the region R_1 is bounded and that its boundary is a constituent of several curves, as shown in Figure(3.3.1). We denote the synthesis of all boundary curves by Γ . The boundary should consist of piecewise continuously differentiable curves on which the normal vector \mathbf{n} pointing outward from the region R_1 can be defined. The boundary Γ is assumed to consist of three disjoint components Γ_1 , Γ_2 and Γ_3 such that

$$\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 = \Gamma. \quad (3.3.1-3)$$

It is possible that $\Gamma_i = \emptyset$, for $i = 1, 2$, or 3 , where \emptyset denotes the empty set.



Figure(3.3.1):Region R_1 and boundary Γ .

There are four distinct problems involving (3.3-1) depending on the boundary conditions prescribed on Γ as follows:

- (1) Dirichlet condition: $u = \Phi$ on Γ_1 (3.3.1-4a)
- (2) Neumann condition: $\frac{\partial u}{\partial n} = \Psi$ on Γ_2 (3.3.1-4b)
- (3) Cauchy condition: $\frac{\partial u}{\partial n} + \alpha u = \beta$ on Γ_3 (3.3.1-4c)
- (4) Periodic condition: $u_0 = u_T$ (3.3.1-4d)

where α , β , Ψ and Φ are given functions on the respective boundary components and T is the period of the function u in the case of condition (3.3.1-4d). If the elliptic differential equation is only subjected to condition (3.3.1-4a), then the problem is called a Dirichlet boundary value problem. If we have $\Gamma = \Gamma_2$, then it is the Neumann boundary value problem. Similarly, if $\Gamma = \Gamma_3$, the problem is known as the Cauchy boundary-value problem, while the periodic problem has no boundary conditions. This latter problem will be dealt with in more detail in chapter 6.

Before developing finite difference methods for solving elliptic equations, we shall state an analytical tool in the study of elliptic PDEs - The Maximum (Minimum) Modulus Theorem - that is, the solution of, for example, the Laplace equation (3.3.1-2a) has no maxima or minima at interior points of the domain of integration. Alternatively, it states that every solution of the

elliptic equation achieves its maximum or minimum values on the boundary. Thus, a problem of mathematical physics is well-posed if its solution exists, is unique, and depends continuously on the data. Therefore we maintain that an elliptic problem is well-posed provided the boundary is closed.

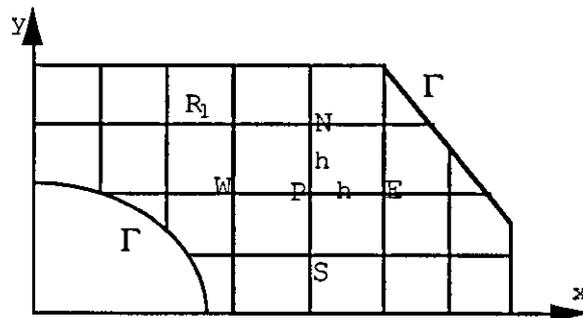
3.3.2 DISCRETIZATION OF THE ELLIPTIC BOUNDARY - VALUE PROBLEMS

The problem (3.3-1) or specifically (3.3.1-2a), (3.3.1-2b) or (3.3.1-2c) can be solved approximately in a given region R_1 subject to boundary conditions (3.3.1-4a) to (3.3.1-4d) through the application of the finite difference method as illustrated below.

Step 1: Discretization of the region

The desired function $u(x,y)$ is substituted by its values at the discrete points of the region R_1 and the boundary Γ . For convenience, the function $u(x,y)$ is discretized using a regular square net with mesh size h in the region R_1 as shown in Figure(3.3.2a).

The values of u are those obtained at the grid points unless they are already known from the boundary conditions. In the case of curved boundary components, we introduce points as the intersection of grid lines with the boundary. In Figure(3.3.2a), the grid points are those given by the intersections of the lines such as the points marked P, W, N, S and E.



Figure(3.3.2a):Region with mesh and grid points.

Let the value of the exact solution be $u(x_i, y_j)$ and the corresponding approximate value be u_{ij} . We note that in certain problems it is necessary to use variable mesh sizes in the x and y directions in order to accommodate the shape of the region or the nature of the desired solution (Marsal[1976]). Moreover, regular triangular or hexagonal nets may be useful (Collatz[1966], Marsal[1976]) because a regular hexagonal net easily admits to locally finer discretization. In practical problems, the use of triangular grids arises for curved boundaries as used in Finite Element method.

Step 2: Finite difference approximation of the dependent variable and its derivatives

For the chosen discretization of the function the partial differential equation is approximated at the grid points by means of the discrete function values u_{ij} . In the case of a regular square net, the first and second partial derivatives may be approximated by means of difference quotients. We can use the central difference quotient to approximate the first partial derivatives as it is convenient and simple and furthermore it gives a good approximation of the first derivative since it represents the slope of the interpolating parabola at the mid point. For a regular interior point $P(x, y)$ which has four neighbouring grid points at a distance h away, we adopt the approximations

$$\left. \begin{aligned} u_x(x_i, Y_j) &\approx \frac{u_{i+1j} - u_{i-1j}}{2h} \\ u_y(x_i, Y_j) &\approx \frac{u_{ij+1} - u_{ij-1}}{2h} \end{aligned} \right\} \quad (3.3.2-1a)$$

$$\left. \begin{aligned} u_{xx}(x_i, Y_j) &\approx \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h^2} \\ u_{yy}(x_i, Y_j) &\approx \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{h^2} \end{aligned} \right\} \quad (3.3.2-1b)$$

where the difference quotients are defined by means of the approximate values at the grid points.

In Figure(3.3.2a), the four neighbouring grid points are labelled as N, S, E and W. Thus we define

$$\left. \begin{aligned} u_P &= u_{ij}, \quad u_N = u_{i,j+1}, \quad u_W = u_{i-1,j}, \\ u_S &= u_{i,j-1}, \quad u_E = u_{i+1,j}. \end{aligned} \right\} \quad (3.3.2-2)$$

We can now approximate the Poisson equation (3.3.1-2b) at the grid point P(x,y) by the difference equation

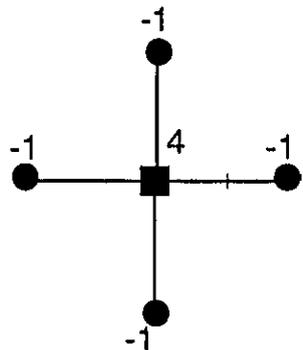
$$\frac{u_E - 2u_P + u_W}{h^2} + \frac{u_N - 2u_P + u_S}{h^2} \approx f_P, \quad (3.3.2-3)$$

where f_P denotes $f(x_i, y_j)$.

Multiply (3.3.2-3) throughout by $-h^2$ and rearrange, we obtain

$$-u_E - u_W + 4u_P - u_N - u_S + h^2 f_P = 0. \quad (3.3.2-4)$$

It is often written in operator form as



$$Ou + h^2 f_P = 0. \quad (3.3.2-5)$$

Step 3:Adaptation of the difference approximation to the boundary conditions

The prescribed boundary conditions of the problem must be taken into account, and the difference approximation of the differential equation may have to be adapted to the boundary conditions.

If there is only a Dirichlet boundary condition to be satisfied, then the net can be such as to generate regular interior grid points. In such a case, the operator form of (3.3.2-5) can be superimposed on all interior grid points with unknown function values, while the known boundary values can simply be substituted. If there exist irregular grid points as in Figure(3.3.2a), then appropriate difference equations must be derived for them. We shall consider such cases in Section (3.3.3).

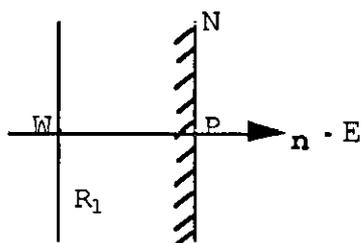
In this preliminary discussion, we assume that the boundary coincides with a net line parallel to the y axis, and that the Neumann boundary condition requires the normal derivative to vanish (see Figure(3.3.2b)).

The outward normal vector \mathbf{n} points in the positive direction of the x axis. Let E be a fictitious point with value u_E . The derivative in the normal direction can be approximated by means of the central difference quotient as

$$\left. \frac{\partial u}{\partial n} \right|_P \approx \frac{u_E - u_W}{2h}. \quad (3.3.2-6)$$

Since it vanishes, therefore, to second order,

$$u_E = u_W. \quad (3.3.2-7)$$



Figure(3.3.2b):Neumann boundary condition

Thus we have $u(x,y)$ is symmetric with respect to the boundary. Hence by using (3.3.2-4), it follows that

$$-u_w + 2u_p - \frac{1}{2}u_N - \frac{1}{2}u_s + \frac{1}{2}h^2 f_p = 0. \quad (3.3.2-8)$$

Therefore its corresponding operator form is

$$\left. \begin{array}{l} \begin{array}{c} -1/2 \\ \bullet \\ | \\ 2 \\ \blacksquare \\ | \\ -1/2 \\ \bullet \end{array} \quad \bigcirc u + \frac{1}{2}h^2 f_p = 0 \\ \\ \frac{\partial u}{\partial n} \Big|_P = \frac{\partial u}{\partial x} \Big|_P = 0 \end{array} \right\} \quad (3.3.2-9)$$

for a regular boundary point $P(x,y)$ with three neighbouring grid points N, W and S .

Step 4: Representation of the problem as a system of linear equations

In order to determine the numerical approximation of the unknown values at the grid points, we need an equation at each point. These we can obtain from the preceding two steps where we have a linear difference equation for each grid point. Upon applying it on all the grid points we are able to give a system of linear equations for the unknown values. The grid points of the net in which the function values are unknown are numbered. It is done in a suitable way so that the appropriate structure for the resulting system of equations will permit an efficient solution procedure. Thus the system of linear equations represents the discrete form of the given boundary-value problem. This system of linear equations can be written in a compact form as

$$Mu = s \quad (3.3.2-10)$$

where M is the matrix derived from applying the operator forms (3.3.2-5) and (3.3.2-9) on the grid points, u is the vector of the unknown values at the grid points and s is the constant vector obtained from applying the operator forms (3.3.2-5) and (3.3.2-9) on the grid points.

We shall now describe in detail how the discretization process can be carried out. There are three well known methods of obtaining finite difference approximation to partial derivatives. These methods are based upon variational formulations, Taylor series expansion and integral equations. All these approximations may introduce truncation errors; their presence will be denoted by the asymptotic O notation. We shall derive the finite difference approximations based on finite Taylor series expansion of the solution vector over the domain of integration.

As previously outlined in Step 1, in order to apply the method of finite differences to obtain an approximate solution for the problem defined by (3.3-1), grid lines parallel to the coordinate axes are super-imposed on the region so that for any grid point (i, j)

$$\left. \begin{array}{l} x = ih \quad i=0,1,2\dots n-1 \\ y = jh \quad j=0,1,2\dots m-1 \end{array} \right\} \quad (3.3.2-11)$$

where for simplicity we have chosen an equal mesh size h . We may choose $x = ih_1$ and $y = jh_2$ such that $h_1 = ah$ and $h_2 = bh$ for $a \neq b$ to obtain unequal mesh sizes.

Assume that $u(x,y)$ has continuous partial derivatives at least of fourth order in the neighbourhood of (x,y) . By using the Taylor series expansions, we obtain

$$\begin{aligned}
u(x \pm h, y) &= u(x, y) \pm hu_x(x, y) \\
&+ \frac{h^2}{2!} u_{xx}(x, y) \pm \frac{h^3}{3!} u_{xxx}(x, y) + \frac{h^4}{4!} u_{xxxx}(x, y) \\
&\pm \frac{h^5}{5!} u_{xxxxx}(x, y) + \frac{h^6}{6!} u_{xxxxxx}(x, y) \\
&+ O(h^7)
\end{aligned} \tag{3.3.2-12a}$$

$$\begin{aligned}
u(x, y \pm h) &= u(x, y) \pm hu_y(x, y) \\
&+ \frac{h^2}{2!} u_{yy}(x, y) \pm \frac{h^3}{3!} u_{yyy}(x, y) + \frac{h^4}{4!} u_{yyyy}(x, y) \\
&\pm \frac{h^5}{5!} u_{yyyyy}(x, y) + \frac{h^6}{6!} u_{yyyyyy}(x, y) \\
&+ O(h^7).
\end{aligned} \tag{3.3.2-12b}$$

By subtracting $u(x-h, y)$ from $u(x+h, y)$, we obtain

$$u_x(x, y) = \frac{1}{2h} [u(x+h, y) - u(x-h, y)] + O(h^2). \tag{3.3.2-13}$$

By adding $u(x-h, y)$ to $u(x+h, y)$, we have

$$u_{xx}(x, y) = \frac{1}{h^2} [u(x+h, y) - 2u(x, y) + u(x-h, y)] + O(h^2). \tag{3.3.2-14}$$

Similarly, we may obtain

$$u_y(x, y) = \frac{1}{2h} [u(x, y+h) - u(x, y-h)] + O(h^2) \tag{3.3.2-15}$$

and

$$u_{yy}(x, y) = \frac{1}{h^2} [u(x, y+h) - 2u(x, y) + u(x, y-h)] + O(h^2). \tag{3.3.2-16}$$

By a similar application of the Taylor series expansion in two dimensions, we may obtain

$$\begin{aligned}
u(x \pm h, y \pm h) &= u(x, y) \pm hu_x(x, y) + hu_y(x, y) \\
&+ \frac{h^2}{2!} [u_{xx}(x, y) \pm 2u_{xy}(x, y) + u_{yy}(x, y)] \\
&+ \frac{h^3}{3!} [\pm u_{xxx}(x, y) + 3u_{xxy}(x, y) \\
&\quad \pm 3u_{xyy}(x, y) + u_{yyy}(x, y)] \\
&+ \frac{h^4}{4!} [u_{xxxx}(x, y) \pm 4u_{xxxxy}(x, y) \\
&\quad + 6u_{xxyy}(x, y) \pm 4u_{xyyy}(x, y) + u_{yyyy}(x, y)] \\
&+ O(h^5).
\end{aligned} \tag{3.3.2-17}$$

Correspondingly, we may obtain the expressions for $u(x\pm h, y-h)$. Hence we have the result

$$u_{xy}(x, y) = \frac{1}{4h^2} [u(x+h, y+h) - u(x-h, y+h) - u(x+h, y-h) + u(x-h, y-h)] + O(h^2). \quad (3.3.2-18)$$

Now for the case when $B = 0$ in (3.3-1), and neglecting terms of order h^2 and higher in (3.3.2-12) to (3.3.2-16), we arrive at the five-point finite difference approximation to the partial differential equation (3.3-1) at a grid point (i, j) ,

$$-\alpha_0 u_{i,j} + \alpha_1 u_{i+1,j} + \alpha_2 u_{i-1,j} + \alpha_3 u_{i,j+1} + \alpha_4 u_{i,j-1} = s_{i,j} \quad (3.3.2-19)$$

where the α_i , $i = 0, 1, 2, 3$ and 4 are functions of x and y and are given by

$$\left. \begin{aligned} \alpha_0 &= \sum_{k=1}^4 \alpha_k - h^2 F_{i,j} \\ \alpha_1 &= A_{i,j} + \frac{1}{2} h D_{i,j} \\ \alpha_2 &= A_{i,j} - \frac{1}{2} h D_{i,j} \\ \alpha_3 &= C_{i,j} + \frac{1}{2} h E_{i,j} \\ \alpha_4 &= C_{i,j} - \frac{1}{2} h E_{i,j} \\ s_{i,j} &= h^2 G_{i,j} \end{aligned} \right\} \quad (3.3.2-20)$$

and we have $u(x, y) = u(ih, jh)$ denoted by $u_{i,j}$. Similarly with the $A_{i,j} = A(ih, jh)$, $D_{i,j} = D(ih, jh)$, $C_{i,j} = C(ih, jh)$, $E_{i,j} = E(ih, jh)$, $F_{i,j} = F(ih, jh)$ and $G_{i,j} = G(ih, jh)$ for all $(i, j) \in \Gamma$.

Clearly, all the α_i will be positive provided that h satisfies the condition

$$0 < h < \min \left\{ \frac{2A_{i,j}}{|D_{i,j}|}, \frac{2C_{i,j}}{|E_{i,j}|} \right\}, \quad (3.3.2-21)$$

where the minimum is taken over all points of $R_1 \cup \Gamma$. Since $A, C > 0$ and $F \leq 0$ and are bounded, there exists a positive minimum and for that h

$$\alpha_0 \geq \sum_{k=1}^4 \alpha_k. \quad (3.3.2-22)$$

This relation will be important in the discussion of iterative methods of solving the resulting equations obtained from Step 4.

Now suppose the number of interior grid points is n . By applying the five-point computational operator (3.3.2-19) at every interior grid point yields a set of n simultaneous linear equations whose matrix form is

$$Mu = s. \quad (3.3.2-23)$$

The vectors u and s consist of n unknowns and the quantities $-s_{i,j}$ together with boundary values respectively. The matrix M is normally square and sparse (but with real coefficients) in which the main diagonal entries are the α_0 of (3.3.2-20) and the off diagonal entries are the negatives of the α_i of (3.3.2-20) which do not correspond to boundary points. Before proceeding to the discussion of the various methods available for solving the system given by (3.3.2-23), we shall note the following properties of the matrix M .

Let $M = (m_{i,j})$,

$$\left. \begin{aligned} (1) \quad & m_{i,j} > 0 \quad \forall i = j \\ & m_{i,j} \leq 0 \quad \forall i \neq j \\ (2) \quad & m_{i,i} \geq \sum_{\substack{j=1 \\ j \neq i}}^m |m_{i,j}| \quad \text{with strict inequality} \\ & \qquad \qquad \qquad \text{for some } i \\ (3) \quad & M \text{ is irreducible} \end{aligned} \right\} (3.3.2-24)$$

Conditions (1) and (2) follow from (3.3.2-21) and its preceding argument. The fact that the inequality holds for some i can be observed by applying the five-point

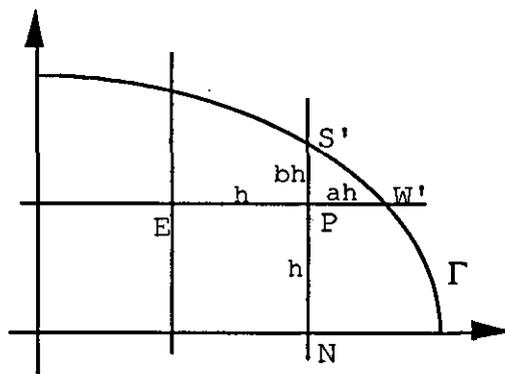
operator at an interior grid point adjacent to a boundary.

Now (3.3.2-23) and in particular (3.3.2-10) can be solved for the unknown vector \mathbf{u} either by a direct method or an iterative method which is described in Section (3.3.5)

3.3.3 GRID POINTS NEAR THE BOUNDARY AND GENERAL BOUNDARY CONDITIONS

In this section we shall illustrate the treatment of the situations when we have grid points near the boundary and general boundary conditions have to be satisfied. We consider a typical example which can be obviously generalized. The basic problem consists of constructing an appropriate difference approximation of a given differential operator, say that of $u_{xx} + u_{yy}$.

Suppose $P(x,y)$ be an irregular interior grid point which lies near the boundary Γ as shown in Figure(3.3.3a). Let W' and S' be the points where the boundary curve Γ and the net lines intersect at ah and bh distances away from $P(x,y)$ respectively, with $0 < a,b < 1$.



Figure(3.3.3a): Irregular grid point near the boundary

We aim to obtain the approximations of u_{xx} and u_{yy} at the grid point $P(x,y)$ such that they are linear combinations of the values of u_P, u_E and $u_{W'}$ and u_P, u_N and $u_{S'}$ respectively. We assume that $u(x,y)$ is sufficiently

continuously differentiable. By means of the Taylor series expansion of the function u about x we may obtain the following expressions:

$$\begin{aligned} u(x+h, y) &= u(x, y) + hu_x(x, y) + \frac{1}{2} h^2 u_{xx}(x, y) \\ &\quad + \frac{1}{6} h^3 u_{xxx}(x, y) + \dots \\ u(x-ah, y) &= u(x, y) - ah u_x(x, y) + \frac{1}{2} a^2 h^2 u_{xx}(x, y) \\ &\quad - \frac{1}{6} a^3 h^3 u_{xxx}(x, y) + \dots \\ u(x, y) &= u(x, y) \end{aligned}$$

Suppose that c_1 , c_2 and c_3 are some constants such that

$$\begin{aligned} c_1 u(x+h, y) + c_2 u(x-ah, y) + c_3 u(x, y) \\ = (c_1 + c_2 + c_3) u(x, y) + (c_1 - ac_2) h u_x(x, y) \\ + (c_1 + a^2 c_2) \frac{1}{2} h^2 u_{xx}(x, y) + \dots \end{aligned}$$

Since this linear combination has to be an approximation of u_{xx} at the point $P(x, y)$, therefore we obtain the necessary conditions

$$\begin{aligned} c_1 + c_2 + c_3 &= 0 \\ (c_1 - ac_2) h &= 0 \\ (c_1 + a^2 c_2) \frac{1}{2} h^2 &= 1 \end{aligned}$$

Hence we obtain

$$c_1 = \frac{2}{h^2(1+a)}, \quad c_2 = \frac{2}{h^2 a(1+a)}, \quad c_3 = \frac{2}{h^2 a}.$$

Since the approximate function values of u are known at the points $P(x, y)$, E and W' , we can use this information to approximate u_{xx} . Thus we have at the point $P(x, y)$

$$u_{xx} \approx \frac{2}{h^2} \left\{ \frac{u_E}{1+a} + \frac{u_{W'}}{a(1+a)} - \frac{u_P}{a} \right\}. \quad (3.3.3-1a)$$

Similarly, we can obtain an approximation of u_{yy} at the point $P(x, y)$ as

$$u_{yy} \approx \frac{2}{h^2} \left\{ \frac{u_N}{1+b} + \frac{u_{S'}}{b(1+b)} - \frac{u_P}{b} \right\}. \quad (3.3.3-1b)$$

Thus by multiplying both equations (3.3.3-1a) and (3.3.3-1b) by $-h^2$ and adding the results, we may obtain the difference equation for the Poisson equation (3.3.1-2b) as

$$2\left[\frac{1}{a} + \frac{1}{b}\right]u_P - \frac{2}{1+b}u_N - \frac{2}{a(1+a)}u_W - \frac{2}{b(1+b)}u_S - \frac{2}{1+a}u_E + h^2f_P = 0 \quad (3.3.3-2)$$

Note that in general, the symmetricity of M is dependent on the equality between a and b since for $a \neq b$, we have the coefficients of u_N and u_E in (3.3.3-2) are different. Even by scaling the difference equations we may not in general attain symmetry. In the particular case $a = b$ we may multiply (3.3.3-2) by the factor $\frac{1+a}{2}$ so that M is symmetric with respect to the grid point $P(x,y)$. The difference equation is then modified to

$$2\left[\frac{1+a}{a}\right]u_P - u_N - \frac{1}{a}u_W - \frac{1}{a}u_S - u_E + \frac{1}{2}(1+a)h^2f_P = 0. \quad (3.3.3-3)$$

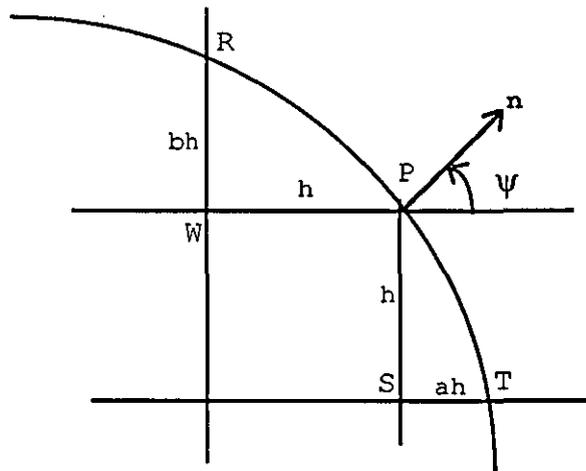
Next we suppose that a Neumann condition is to be satisfied on the boundary part Γ_2 and we assume that the boundary point $P(x,y)$ is a grid point and the boundary is as shown in Figure(3.3.3b). Let the angle between the outer normal vector \mathbf{n} at the point $P(x,y)$ and the positive x axis be ψ , measured anti-clockwise. By using an appropriate linear combination of the expression of the normal derivative and the function values of u at the neighbouring grid points W, S, R and T of the grid point $P(x,y)$, it can be shown that the resulting difference equation of the Poisson equation for the grid point $P(x,y)$ is given by

$$-C_P u_P - C_W u_W - C_S u_S - C_R u_R - C_T u_T - C_\perp \left. \frac{\partial u}{\partial n} \right|_P + f_P = 0, \quad (3.3.3-4)$$

with

$$\left. \begin{aligned}
 c_L &= \frac{2(a+b+2)}{hN} \\
 c_T &= \frac{2(\sin\psi - \cos\psi)}{ah^2N} \\
 c_R &= \frac{2(\cos\psi - \sin\psi)}{bh^2N}
 \end{aligned} \right\} \quad (3.3.3-5)$$

where N is defined by $N = (a+1)\sin\psi + (b+1)\cos\psi$ and a, b, h and ψ are given. The known value of the Neumann boundary condition is then substituted into (3.3.3-4).



Figure(3.3.3b):Neumann condition at the boundary point

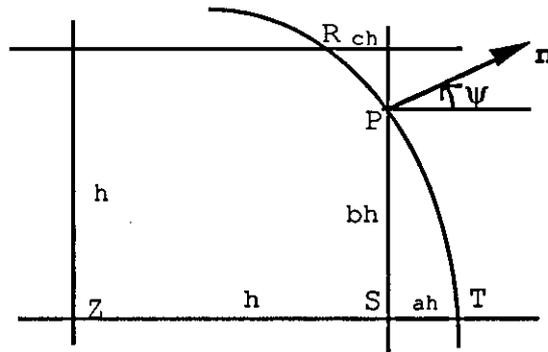
The treatment of a Cauchy boundary condition at a general point is similar to that of the Neumann condition. Consider the situation as shown in Figure(3.3.3c). Let $P(x,y)$ be a boundary point which is not the intersection point of grid lines. Let the direction of the outer normal vector be defined by the angle ψ .

Then it can be shown that the resulting difference equation of the Poisson equation is given by

$$c_P u_P + c_S u_S + c_Z u_Z + c_R u_R + c_T u_T + c_L \gamma - f_P = 0, \quad (3.3.3-6)$$

where γ is a known value of the Cauchy condition. The boundary condition at the point $P(x,y)$ is expressed

implicitly in terms of the coefficients of the u values of (3.3.3-6) and the explicit constant $c_{1\gamma}$.



Figure(3.3.3c):Cauchy condition at the boundary point

Generally, c_z is nonzero and even if Z is a regular interior grid point, in which case we can use the five-point difference equation (3.3.2-4) for Z ; the matrix M of the system of equations is in any event nonsymmetric.

The derivation of difference equations for boundary points with Neumann or Cauchy condition is tedious and susceptible to errors. However, the coefficients can be easily determined by means of a computer program where it only needs the information about points neighbouring to the boundary point, the type of boundary condition including the numerical values of ψ , α , β and γ and the elliptic differential equation to be approximated. Alternatively, we may use the computer to generate the complete system of difference equations corresponding to a given boundary-value problem such as in the form of the operator equations.

3.3.4 DISCRETIZATION ERRORS

The solution of the system of linear difference equations represent the approximation of the function values of the solutions $u(x,y)$ of the given boundary problem. Therefore in order to have an idea of (at least) qualitative estimates, we need to determine the local truncation error of the difference approximation. We shall consider the Poisson equation and the

difference equations used so far to illustrate our discussion.

The local truncation error of a difference equation is defined as the value that results when the exact solution of the differential equation is substituted into the difference equation.

Therefore in the case of the five-point difference equation (3.3.2-4) valid for a regular interior grid point $P(x,y)$, the local truncation error T_p is given by

$$T_p = \frac{1}{h^2} [u(x+h,y) + u(x-h,y) + u(x,y-h) + u(x,y+h) - 4u(x,y)] - f(x,y). \quad (3.3.4-1)$$

By substituting (3.3.2-12a) and (3.3.2-12b) in (3.3.4-1) we obtain for function values at $P(x,y)$

$$T_p = u_{xx}(x,y) + u_{yy}(x,y) - f(x,y) + \frac{1}{12} h^2 [u_{xxxx}(x,y) + u_{yyyy}(x,y)] + \frac{1}{360} h^4 [u_{xxxxxx}(x,y) + u_{yyyyyy}(x,y)] + O(h^6) \quad (3.3.4-2)$$

By the assumption that $u(x,y)$ is a solution of the Poisson equation, the local truncation error of the five-point difference equation at a regular interior grid point $P(x,y)$ is given by

$$T_p = \frac{1}{12} h^2 [u_{xxxx}(x,y) + u_{yyyy}(x,y)] + \frac{1}{360} h^4 [u_{xxxxxx}(x,y) + u_{yyyyyy}(x,y)] + O(h^6) \quad (3.3.4-3)$$

Thus we deduce that T_p is $O(h^2)$. This result holds true even for boundary points with the Neumann condition $\frac{\partial u}{\partial n} = 0$ whenever the boundary coincides either with a mesh line or a diagonal of it.

For an irregular grid point $P(x,y)$ near the boundary (see Figure(3.3.3a)), it can be shown that the local

truncation error is proportional to the mesh size h . Thus, we have the local truncation error is $O(h)$.

Next we shall investigate the relationship between the local truncation error T_p and the error $\tau_p = u(x,y) - u_p$ of the numerical approximation at the point $P(x,y)$. First, we assume that at all grid points, the difference equation used has a local truncation error of the form (3.3.4-3). Consider a typical regular interior grid point $P(x,y)$. Thus, from (3.3.4-1), we have

$$\frac{1}{h^2} [u(x,y+h) + u(x-h,y) + u(x,y-h) + u(x+h,y) - 4u(x,y)] - f(x,y) - T_p = 0. \quad (3.3.4-4)$$

From (3.3.2-4), we have the approximations satisfy the difference equation

$$\frac{1}{h^2} [u_N + u_W + u_S + u_E - 4u_P] - f(x,y) = 0. \quad (3.3.4-5)$$

By subtracting (3.3.4-5) from (3.3.4-4), we obtain the error equation as

$$\frac{1}{h^2} [\tau_N + \tau_W + \tau_S + \tau_E - 4\tau_P] - T_p = 0. \quad (3.3.4-6)$$

By taking (3.3.4-4) into account and multiplying (3.3.4-6) by $-h^2$, we obtain for each regular interior grid point $P(x,y)$,

$$4\tau_P - \tau_N - \tau_W - \tau_S - \tau_E + C_p h^4 + D_p h^6 + O(h^8) = 0 \quad (3.3.4-7)$$

where C_p, D_p, \dots are constants that depend on the point $P(x,y)$ and on the solution $u(x,y)$. The discrete errors satisfy a system of linear equations in which the matrix is identical to that of the system of difference equations. The components of the constant vectors of (3.3.4-7) are $O(h^4)$. Let τ denote the error vector at the grid points, ζ and ξ denote the vectors of the constants C_p and D_p respectively. Therefore from (3.3.4-7), we obtain the system

$$M\tau + h^4\zeta + h^6\xi + \dots = 0. \quad (3.3.4-8)$$

Since M is nonsingular, the existence of τ is guaranteed. Hence from (3.3.4-8), we obtain

$$\tau = -M^{-1}[h^4\zeta + h^6\xi + \dots]. \quad (3.3.4-9)$$

By taking the Euclidean vector norm and the subordinate spectral norm, from (3.3.4-9), the error estimate is obtained as

$$\|\tau\|_2 \leq \|M^{-1}\|_2 \{h^4\|\zeta\|_2 + h^6\|\xi\|_2 + \dots\}. \quad (3.3.4-10)$$

It can be shown that the spectral norm of the inverse of M satisfies $\|M^{-1}\|_2 \leq Kh^{-2}$ for some constant K (see Schwarz[1989], pp.453-455). Consequently, from (3.3.4-10), the following error estimate is obtained

$$\|\tau\|_2 \leq K \{h^2\|\zeta\|_2 + h^4\|\xi\|_2 + \dots\}. \quad (3.3.4-11)$$

The Euclidean norm of the error decreases as h^2 . Thus the order of convergence of the five-point formula (3.3.2-4) is two. It also follows that the approximate solutions at the grid points converge, as h tends to zero, to the exact solutions of the boundary-value problem. We have implicitly assumed that the solution $u(x,y)$ is sufficiently continuously differentiable on the closed domain. The inequality (3.3.4-11) can also be derived by other methods (Collatz[1966]; Finkenstein[1977]).

However, if difference equations with a local truncation error $O(h)$ are present, then (3.3.4-10) will involve an h^3 term. Consequently, the bound for the norm of the error is only proportional to h .

In the situations where the assumption that the solution $u(x,y)$ has to be at least four times continuously differentiable is not fulfilled; special analytical techniques are required to describe the convergence

behaviour correctly (Bramble et al.[1968]). Such instances occur, for example, if Dirichlet boundary conditions are discontinuous, or if the domain has obtuse corners. Then the low-order partial derivatives of the solution are singular at the points concerned. Thus it is quite advantageous in the numerical solution of such problems to incorporate analytical tools adequately in order to treat the points of singularity (Gladwell and Wait[1979]; Mitchell and Griffiths[1980]).

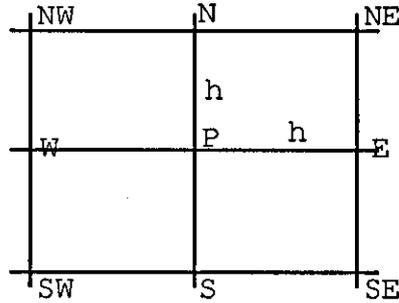
The accuracy of the approximate solution of the difference equations may be increased by decreasing the mesh size h . The error estimate (3.3.4-11) will correspondingly reduce the error at the expense of a considerable increase in the order of the system of the linear equations. Moreover, the condition number of the matrix M of the system of difference equations becomes large, because of the increase of the spectral norm $\|M^{-1}\|_2$.

Another possibility of improving the accuracy of the approximate solution is to increase the order of the difference equations. This is achieved by taking more function values into the approximations. The second partial derivative may be approximated by means of the difference approximation

$$u_{xx}(x,y) \approx \frac{1}{12h^2} [-u(x-2h,y) + 16u(x-h,y) - 30u(x,y) + 16u(x+h,y) - u(x+2h,y)], \quad (3.3.4-12)$$

with truncation error of $O(h^4)$. However, this approach will result in a difference equation which involves grid points further from the central point of interest. Thus it has the disadvantage of deriving special difference approximations for too many grid points near the boundary.

A better approach is to approximate the differential expression $\Delta u = u_{xx} + u_{yy}$ at the point $P(x,y)$ as an entity by a linear combination of function values. Consider eight neighbouring grid points of $P(x,y)$ as depicted in Figure(3.3.4).



Figure(3.3.4):Eight neighbouring points of a grid point

From (3.3.2-12a) and (3.3.2-12b) we have the expressions for $u(x\pm h,y)$ and $u(x,y\pm h)$ respectively. While from (3.3.2-17) we have the expression for $u(x\pm h,y+h)$ and then $u(x\pm h,y-h)$ can be deduced from it.

Next we combine the function values of four grid points to form the following expressions:

$$\begin{aligned}\Sigma_1 &= u(x,y+h) + u(x-h,y) + u(x,y-h) + u(x+h,y) \\ &= 4u(x,y) + h^2[u_{xx}(x,y) + u_{yy}(x,y)] \\ &\quad + \frac{h^4}{12}[u_{xxxx}(x,y) + u_{yyyy}(x,y)] + O(h^6) \quad (3.3.4-13a)\end{aligned}$$

$$\begin{aligned}\Sigma_2 &= u(x+h,y+h) + u(x-h,y+h) + u(x-h,y-h) + u(x+h,y-h) \\ &= 4u(x,y) + 2h^2[u_{xx}(x,y) + u_{yy}(x,y)] \\ &\quad + \frac{h^4}{6}[u_{xxxx}(x,y) + 6u_{xxyy}(x,y) + u_{yyyy}(x,y)] + O(h^6). \quad (3.3.4-13b)\end{aligned}$$

So by some combination of (3.3.4-13a) and (3.3.4-13b), we obtain

$$\begin{aligned}4\Sigma_1 + \Sigma_2 - 20u(x,y) &= 6h^2[u_{xx}(x,y) + u_{yy}(x,y)] \\ &\quad + \frac{h^4}{2}[u_{xxxx}(x,y) + 2u_{xxyy}(x,y) + u_{yyyy}(x,y)] \\ &\quad + O(h^6). \quad (3.3.4-13c)\end{aligned}$$

However, we can write that

$$\begin{aligned} u_{xxxx}(x,y) + 2u_{xxyy}(x,y) + u_{yyyy}(x,y) \\ = [u_{xx}(x,y) + u_{yy}(x,y)]_{xx} + [u_{xx}(x,y) + u_{yy}(x,y)]_{yy} \\ = \Delta(\Delta u). \end{aligned}$$

Using the fact that $u(x,y)$ should be the solution of the Poisson equation so that we have $\Delta u = f$.

Hence

$$\Delta(\Delta u)(x,y) = \Delta f(x,y) = f_{xx}(x,y) + f_{yy}(x,y)$$

where all the values are evaluated at the grid point $P(x,y)$.

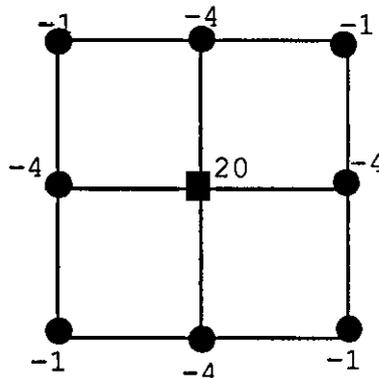
Therefore by replacing the expression within the second parentheses of (3.3.4-13c) without error from the value of $\Delta f(x,y)$, we obtain

$$[u_{xx}(x,y) + u_{yy}(x,y)] \approx \frac{1}{6h^2} [4\Sigma_1 + \Sigma_2 - 20u(x,y)] - \frac{h^2}{12} \Delta f(x,y).$$

Next we substitute the exact values of the solution with the approximations at the corresponding grid points in (3.3.4-13a) and (3.3.4-13b) and multiply the result by $-6h^2$ to obtain the following difference equation for a regular interior grid point $P(x,y)$

$$\begin{aligned} 20u_p - 4[u_N + u_W + u_S + u_E] - u_{NE} - u_{NW} - u_{SW} - u_{SE} \\ + \frac{h^4}{2} [\Delta f]_p + 6h^2 f_p = 0. \end{aligned} \quad (3.3.4-14)$$

In operator form the representation is given as



$$\Delta u + 6h^2 f_p + \frac{h^4}{2} [\Delta f]_p = 0. \quad (3.3.4-15)$$

The constant term in (3.3.4-15) constitutes the function value and the sum of $f_{xx}(x,y) + f_{yy}(x,y)$ evaluated at the point $P(x,y)$. If $f(x,y)$ is a constant function, then the last term of (3.3.4-15) is not present. For the Laplace equation the constant term is completely absent from (3.3.4-15). It is clear that from the derivation, the local truncation error of the difference approximation is $O(h^4)$.

Another strategy of increasing the order of the difference approximation for the Poisson equation without enlarging the number of grid points can be done as described in the following paragraphs.

The differential expression itself is used in some of the neighbouring grid points beside the function values. The value of the differential expression at the corresponding grid points are replaced by the known function on the right-hand side of the given differential equation.

The Taylor series expansion of $u_{xx} + u_{yy}$ at the four neighbouring grid points N,W,S and E can be obtained as

$$\begin{aligned} u_{xx}(x\pm h, y) + u_{yy}(x\pm h, y) \\ = u_{xx}(x, y) + u_{yy}(x, y) \pm hu_{xxx}(x, y) \pm hu_{xyy}(x, y) \\ + \frac{1}{2} h^2 [u_{xxyy}(x, y) + u_{xxxx}(x, y)] + \dots \quad (3.3.4-16a) \end{aligned}$$

$$\begin{aligned} u_{xx}(x, y\pm h) + u_{yy}(x, y\pm h) \\ = u_{xx}(x, y) + u_{yy}(x, y) \pm hu_{xxy}(x, y) \pm hu_{yyy}(x, y) \\ + \frac{1}{2} h^2 [u_{xxyy}(x, y) + u_{yyyy}(x, y)] + \dots \quad (3.3.4-16b) \end{aligned}$$

By adding (3.3.4-16a) and (3.3.4-16b) we obtain

$$\begin{aligned} \Sigma_3 = u_{xx}(x, y+h) + u_{yy}(x, y+h) + u_{xx}(x-h, y) + u_{yy}(x-h, y) \\ + u_{xx}(x, y-h) + u_{yy}(x, y-h) + u_{xx}(x+h, y) + u_{yy}(x+h, y) \\ = 4 [u_{xx}(x, y) + u_{yy}(x, y)] \\ + h^2 [u_{xxxx}(x, y) + 2u_{xxyy}(x, y) + u_{yyyy}(x, y)] + \dots \quad (3.3.4-17) \end{aligned}$$

By taking a suitable linear combination of Σ_1 , Σ_2 and Σ_3 , we may eliminate the fourth partial derivatives and obtain the following relationship

$$8\Sigma_1 + 2\Sigma_2 - h^2\Sigma_3 - 40u = 8h^2[u_{xx}(x,y) + u_{yy}(x,y)] + O(h^6). \quad (3.3.4-18)$$

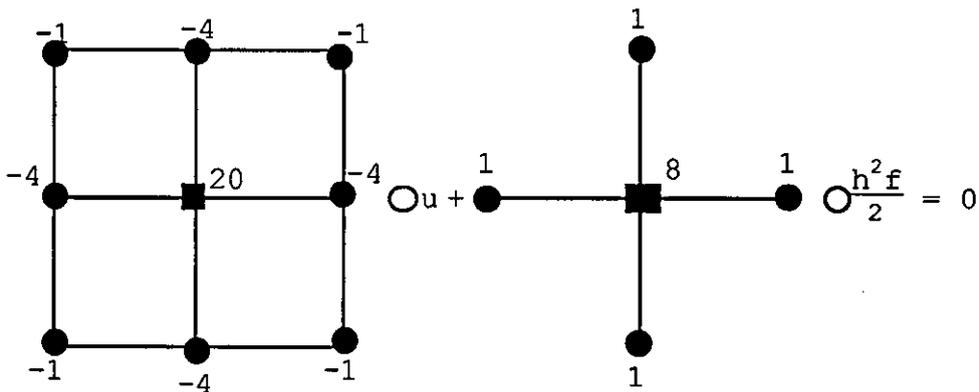
Thus we obtain the approximation

$$u_{xx}(x,y) + u_{yy}(x,y) \approx \frac{1}{8h^2} [8\Sigma_1 + 2\Sigma_2 - h^2\Sigma_3 - 40u],$$

the quantities $u_{xx} + u_{yy}$ occurring in Σ_3 is substituted by the values of f at the corresponding four grid points. Hence by substituting the function values of u with their respective approximations, for the Poisson equation (3.3.1-2b), we obtain the difference equation valid for a regular interior grid point $P(x,y)$

$$20u_P - 4[u_N + u_W + u_S + u_E] - u_{NE} - u_{NW} - u_{SW} - u_{SE} + \frac{h^2}{2}[8f_P + f_N + f_W + f_S + f_E] = 0. \quad (3.3.4-19)$$

Written in operator form, we have



$$(3.3.4-20)$$

which is called the multiple-point operator or the Hermitian operator (Collatz[1966]). It is clear that the local truncation error of (3.3.4-19) is at least $O(h^4)$. However, a careful analysis shows that it is $O(h^6)$ accurate and thus for some problems it admits a high accuracy of the approximate solutions.

3.3.5 METHODS OF SOLUTION OF $Mu = s$

The conditions in (3.3.2-24) are sufficient to prove the existence of a unique solution to the system (3.3.2-23). We remark here that the matrix M is nonsingular and hence the unique solution of (3.3.2-23) exists and is given by

$$u = M^{-1}s. \quad (3.3.5-1)$$

Methods of solution for the system (3.3.2-23) can be classified into two groups: direct and iterative methods.

In the discussion of the various basic methods of solving the system of linear equations (3.3.2-23), we assume the entries of the matrix M and the right-hand side vector s are given by (3.3.2-20).

3.3.6 DIRECT METHODS OF SOLVING (3.3.2-23)

Broadly speaking, direct methods obtain the solution (neglecting round-off errors) of (3.3.2-23) in a finite number of steps. These methods depend on the fact that a closed-form solution to the discretized problem exists. There is a class of fast direct methods that are often used to solve certain class of linear systems. Such systems arise from the discretization of linear elliptic PDEs with constant coefficients over rectangular domains.

During the pre 1965 large-scale computers, direct methods were seldom used for solving large scale linear systems (3.3.2-23) arising from elliptic difference equations^{in 2- or 3-Dimensions}. However, especially for the finite element approximations widely used since 1965 in structural mechanics, direct methods have become increasingly adopted for solving ^{2-Dimension} problems.

We shall now describe the outline of the basic direct methods for solving linear systems of equations (3.3.2-23). We may state some standard theorems without proofs.

Consider the linear system of equations (3.3.2-23) where M is a nonsingular $n \times n$ matrix. For the purpose of the discussion, we assume that M is a full dense matrix.

a) GAUSSIAN ELIMINATION AND LU DECOMPOSITION

The most common form of Gaussian elimination subtracts multiples of rows of M from other rows so as to reduce (3.3.2-23) to an upper triangular system. The unknown vector \mathbf{u} is then solved by back substitution. Mathematically, this is equivalent to first forming the decomposition

$$M = LU, \quad (3.3.6-1)$$

where L is a lower triangular matrix with unity on the main diagonal and U is an upper triangular matrix. Then, the solutions are obtained through solving the triangular systems

$$L\tilde{\mathbf{v}} = \mathbf{s}, \quad U\mathbf{u} = \tilde{\mathbf{v}} \quad (3.3.6-2)$$

which are known as the forward and backward substitutions.

Now if the matrix M is symmetric positive definite, then one has the alternative of using the Choleski decomposition

$$M = LL^T \quad (3.3.6-3)$$

where L is a lower triangular matrix, followed by the forward and backward substitutions

$$L\tilde{\mathbf{v}} = \mathbf{s}, \quad L^T\mathbf{u} = \tilde{\mathbf{v}} \quad (3.3.6-4)$$

to solve the linear system $M\mathbf{u} = \mathbf{s}$.

b) ORTHOGONAL REDUCTION

An alternative to LU decomposition is the reduction

$$M = QR, \quad (3.3.6-5)$$

where Q is an orthogonal matrix and R is upper triangular.

There are two usual approaches to the factorization (3.3.6-5), namely, the Householder transformations and Givens transformations. However, they are both slower than the LU decomposition, even though they are numerically stable without any row interchanges; but this does not overcome the operation count advantage of LU decomposition, even with pivoting. Consequently, they are rarely used for nonsingular systems of equations. Instead, they are widely used such as for eigenvalues determination, least-square problems and orthogonalization of vectors.

A Householder transformation is a matrix of the form $I - \mathbf{w}\mathbf{w}^T$ where \mathbf{w} is a real column vector such that $\mathbf{w}^T\mathbf{w} = 2$. Thus a Householder transformation is symmetric and orthogonal. The reduction is done as follows. Let \mathbf{m}_1 be the first column of M and define

$$\mathbf{u}^T = (m_{11} - \sigma, m_{21}, \dots, m_{n1}), \quad \mathbf{w} = \mu\mathbf{u} \quad (3.3.6-6)$$

where

$$\sigma = \pm(m_1^T m_1)^{1/2}, \quad \gamma = (\sigma^2 - m_{11}\sigma)^{-1}, \quad \mu = \gamma^{1/2} \quad (3.3.6-7)$$

and the sign of σ is always opposite to that of m_{11} for numerical stability.

Therefore the effect of the Householder transformations applied to M successively is to produce a matrix M_1 which has zero elements below the main diagonal such that

The remaining block equations can be written in the same manner as above.

We observe that the equations

$$\left. \begin{aligned} \mathbf{u}_{13} + W_{13}\mathbf{u}_{21} &= \mathbf{d}_{13} \\ V_{21}\mathbf{u}_{13} + \mathbf{u}_{21} + W_{21}\mathbf{u}_{31} &= \mathbf{d}_{21} \\ V_{23}\mathbf{u}_{13} + \mathbf{u}_{23} + W_{23}\mathbf{u}_{31} &= \mathbf{d}_{23} \\ &\vdots \\ &\vdots \end{aligned} \right\} \quad (3.3.6-18)$$

are independent of \mathbf{u}_{i2} . Therefore, we can solve the reduced system (3.3.6-18) independently of \mathbf{u}_{i2} . Once the system (3.3.6-18) has been solved, the vectors \mathbf{u}_{i2} can be obtained from the second equations of (3.3.6-16) and (3.3.6-17) as

$$\left. \begin{aligned} \mathbf{u}_{i2} &= \mathbf{d}_{i2} - W_{i2}\mathbf{u}_{21} \\ \text{and} \\ \mathbf{u}_{i2} &= \mathbf{d}_{i2} - V_{i2}\mathbf{u}_{i-13} - W_{i2}\mathbf{u}_{i+11} \end{aligned} \right\} \quad (3.3.6-19)$$

for $i \geq 2$.

The first and the last vectors, namely, \mathbf{u}_{11} and \mathbf{u}_{p3} do not appear in the reduced system. Thus we may obtain \mathbf{u}_{11} from the first equation of (3.3.6-16) as

$$\mathbf{u}_{11} = \mathbf{d}_1 - W_{11}\mathbf{u}_{21}. \quad (3.3.6-20)$$

Similarly, \mathbf{u}_{p3} may be obtained from the last block equation of (3.3.6-15) as

$$\mathbf{u}_{p3} = \mathbf{d}_p - V_{p1}\mathbf{u}_{p-13}. \quad (3.3.6-21)$$

The method described above is called the Lawrie-Sameh partitioning algorithm. It is summarized as follows:

Step 1: Do LU decomposition (3.3.6-14)

and solve (3.3.6-13).

Step 2: Solve (3.3.6-18) to obtain \mathbf{u}_{i1} and \mathbf{u}_{i3} .

Step 3: Obtain \mathbf{u}_{i2} from (3.3.6-19),

then obtain \mathbf{u}_{11} and \mathbf{u}_{p3} .

Next, multiply (3.3.6-36a) by $-CA_I^{-1}$ and add to (3.3.6-36b); we ^{thus} obtain the equation

$$\bar{A}u_T = \bar{s}, \quad (3.3.6-37)$$

where

$$\left. \begin{aligned} \bar{A} &= A_T - CA_I^{-1}B \\ \bar{s} &= u_T - CA_I^{-1}u_I \end{aligned} \right\}. \quad (3.3.6-37a)$$

The matrix \bar{A} is known as the Gauss transform or Schur complement. Once the system (3.3.6-37) is solved for u_T , the remaining unknown vectors u_i can be determined by solving the systems

$$A_i u_i = s_i - B_i u_T, \quad (3.3.6-38)$$

for $i = 1, \dots, p$ using the LU decomposition method as follows.

Let A_i , for $i = 1, \dots, p$ have stable LU decomposition

$$A_i = L_i U_i. \quad (3.3.6-39)$$

We then solve the systems

$$\left. \begin{aligned} L_i Y_i &= B_i, \quad L_i y_i = s_i, \\ U_i Z_i &= Y_i, \quad U_i z_i = y_i, \end{aligned} \right\} \quad (3.3.6-40)$$

for $i = 1, \dots, p$.

Now

$$C_i A_i^{-1} B_i = C_i (L_i U_i)^{-1} B_i = C_i U_i^{-1} Y_i = C_i Z_i \quad (3.3.6-41a)$$

and,

$$C_i A_i^{-1} s_i = C_i (L_i U_i)^{-1} L_i y_i = C_i U_i^{-1} y_i = C_i z_i. \quad (3.3.6-41b)$$

Therefore using (3.3.6-41a) and (3.3.6-41b) in (3.3.6-37a), we obtain

$$\bar{A} = A_T - \sum_{i=1}^p C_i A_i^{-1} B_i = A_T - \sum_{i=1}^p C_i Z_i \quad (3.3.6-42a)$$

and

$$\bar{s} = s_T - \sum_{i=1}^p C_i A_i^{-1} s_i = s_T - \sum_{i=1}^p C_i z_i. \quad (3.3.6-42b)$$

Thus we have the domain decomposition algorithm summarized as follows.

Step 1: Form the LU decomposition (3.3.6-39) and solve the systems (3.3.6-40).

Step 2: Form $C_i Z_i$ and $C_i z_i$; $i = 1, \dots, p$.

Step 3: Form \bar{A} and \bar{s} and solve the system $\bar{A} u_T = \bar{s}$.

Step 4: Form $c_i = s_i - B_i u_T$; $i = 1, \dots, p$.

Step 5: Solve the systems $A_i u_i = c_i$; $i = 1, \dots, p$ using the LU decompositions (3.3.6-39).

Next, if M is symmetric positive definite and since the matrix of (3.3.6-34), call it \bar{M} , arises from M by interchanges of equations and unknowns, it is related to M by $\bar{M} = PMP^T$, where P is a permutation matrix. Hence, all the matrices \bar{M} , A_i and A_T are symmetric positive definite. Therefore, we may use the Choleski decomposition:

$$A_i = L_i L_i^T \quad (3.3.6-43)$$

instead of the factorization (3.3.6-39) in step 1 of the domain decomposition algorithm above. Then, we solve the systems

$$L_i Y_i = B_i, \quad L_i y_i = s_i, \quad (3.3.6-44)$$

for $i = 1, \dots, p$.

By symmetry, $C_i = B_i^T$, therefore, we have

$$\bar{A} = A_T - \sum_{i=1}^p B_i^T A_i^{-1} B_i, \quad \bar{s} = s_T - \sum_{i=1}^p B_i^T A_i^{-1} s_i. \quad (3.3.6-45)$$

Since

$$B_i^T A_i^{-1} B_i = B^T (L_i L_i^T)^{-1} B_i = Y_i^T Y_i, \quad (3.3.6-46)$$

therefore substituting (3.3.6-44) and (3.3.6-46) into (3.3.6-45), we obtain

$$\bar{A} = A_T - \sum_{i=1}^p Y_i^T Y_i, \quad \bar{s} = s_T - \sum_{i=1}^p Y_i^T y_i. \quad (3.3.6-47)$$

Clearly, \bar{A} is symmetric. Furthermore it is positive definite since for any nonzero vector u_2 of length $p-1$ and set $u_1 = -A_1^{-1} B u_2$, then, by the positive definiteness of M , we obtain

$$0 < (u_1^T, u_2^T) \begin{bmatrix} A_1 & B \\ B^T & A_T \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}.$$

Therefore, we have

$$0 < u_1^T A_1 u_1 + 2u_1^T B u_2 + u_2^T A_T u_2 = u_2^T \bar{A} u_2,$$

which shows that \bar{A} is positive definite. The algorithm for the symmetric positive definite domain decomposition can be summarized as follows.

Step 1: Form the Choleski decomposition. (3.3.6-43) and solve the systems (3.3.6-44).

Step 2: Form \bar{A} and \bar{s} by (3.3.6-47) and solve $\bar{A} u_T = \bar{s}$

Step 3: For $i = 1, \dots, p$; form $c_i = s_i - B_i u_T$ and solve the systems $A_i u_i = c_i$.

e) ODD-EVEN/CYCLIC REDUCTION METHOD

The partitioning and domain decomposition methods all apply in principle to tridiagonal systems. Another approach to the solution of tridiagonal systems is known as the cyclic/odd-even reduction method. With the advent of parallel and vector computers, this algorithm, or one of its variants, has probably been the most popular method for tridiagonal systems. It was first proposed by G. Golub and R. Hockney for special block tridiagonal systems (see Hockney[1965]), but it soon became apparent (Hockney[1970]) that it could be applied to general tridiagonal systems.

The details of this algorithm are described in section 6.5.1. Therefore in this section, we shall only give a brief outline of the method. We write the tridiagonal systems in the form

$$\left. \begin{aligned} a_1 u_1 + b_1 u_2 &= d_1 \\ c_2 u_1 + a_2 u_2 + b_2 u_3 &= d_2 \\ c_3 u_2 + a_3 u_3 + b_3 u_4 &= d_3 \\ &\vdots \\ c_n u_{n-1} + a_n u_n &= d_n \end{aligned} \right\} \quad (3.3.6-48)$$

By multiplying the first equation of (3.3.6-48) with $\frac{-c_2}{a_1}$ and adding the results to the second equation, we eliminate the term involving u_1 from the second equation. Next multiply the third equation by $\frac{-b_2}{a_3}$ and add to the second equation to eliminate the term involving u_3 from the second equation. Thus we obtain a new second equation of the form

$$a_2(1)u_2 + b_2(1)u_4 = d_2(1).$$

The same thing is done with the other equations; every time working with overlapping groups of three equations to produce new middle equations in which the odd-indexed

unknowns have been eliminated. Thus, if n is odd, we obtain at the end of the process a modified system

$$\left. \begin{aligned} a_2(1)u_2 + b_2(1)u_4 &= d_2(1) \\ c_2(1)u_2 + a_4(1)u_4 + b_4(1)u_6 &= d_4(1) \\ \cdot & \\ \cdot & \\ \cdot & \end{aligned} \right\} \quad (3.3.6-49)$$

which involves only the unknowns u_2, u_4, \dots, u_{n-1} . The system (3.3.6-49) is tridiagonal in the unknowns u_2, u_4, \dots . The process of obtaining a new system which contains only the even-indexed unknowns is continued until no further reduction is possible. We note that, by this process, we have at each stage of the reduction, the initial system is split up into two disjoint subsystems; one contains the even-indexed unknowns and the other contains the odd-indexed unknowns. We call the even-indexed ones the reduced system and the odd-indexed ones the eliminated system. In case of $n = 2^q - 1$, the algorithm will terminate with a single final equation. The final equation is then solved and the other vectors of the unknowns can be computed by back substitution.

If $n \neq 2^q - 1$, the process may be terminated in a system with *fewer* unknowns which can be solved separately prior to the back substitution process. Alternatively, we may add a dummy equation of the form $u_1 = 1$ to the system so that the total number of unknowns is $2^q - 1$, for some q . Heller[1976] showed that during the cyclic reduction process the off-diagonal elements decrease quadratically in size relative to the diagonal elements; thus allowing termination before the full $\log n$ steps have been performed. Lambiotte and Voigt[1975] showed that cyclic reduction is a Gaussian elimination applied to the matrix PMP^T for some permutation matrix P . Therefore, if M is symmetric positive definite, so is PMP^T and cyclic reduction is

numerically stable. However, we still need to handle the right-hand side carefully to ensure numerical stability (see Golub and van Loan[1983]). Since Gaussian elimination will cause fill-in when applied to PMP^T , the arithmetic operation count of cyclic reduction is roughly twice that of Gaussian elimination applied to the tridiagonal system.

3.3.7 BASIC ITERATIVE METHODS FOR LINEAR EQUATIONS

Consider the system of linear equations given by (3.3.2-23), where M is a $n \times n$ nonsingular matrix, $\mathbf{u} = (u_1, \dots, u_n)^T$ and $\mathbf{s} = (s_1, \dots, s_n)^T$. In general, the coefficient matrix M is sparse, that is, most of its elements are zeros.

Alternative to the class of methods discussed previously, is another class of methods called iterative methods which obtain the solution of the problem by successive approximations. In the use of iterative methods, one starts with an arbitrary initial guess to the solution and then successively improves the approximation. The iterations will be stopped after some prescribed criteria are met.

This is equivalent to finding a sequence of vectors $\mathbf{u}^{(r)}$, $r = 0, 1, 2, \dots$ such that,

$$\lim_{r \rightarrow \infty} \mathbf{u}^{(r)} = M^{-1} \mathbf{s}. \quad (3.3.7-1)$$

Therefore, we can express the vector $\mathbf{u}^{(r)}$ as a function of M , \mathbf{s} , $\mathbf{u}^{(r-1)}$, \dots , $\mathbf{u}^{(r-k)}$, where k is called the degree or order of the iterative method. Usually we choose a first degree method; that is, $k = 1$. Thus we can write the first degree iterative method as

$$\mathbf{u}^{(r)} = F_r(M, \mathbf{s}, \mathbf{u}^{(r-1)}). \quad (3.3.7-2)$$

The iteration method is said to be stationary if for some $i > 0$, F_r is independent of r for all $r \geq i$, otherwise it is nonstationary.

The iterative method (3.3.7-2) is said to be linear, if for each r , F_r is a linear function of $\mathbf{u}^{(r-1)}$, otherwise it is nonlinear.

The most general linear, stationary iterative method of first degree is of the form,

$$\mathbf{u}^{(r+1)} = G\mathbf{u}^{(r)} + \mathbf{g}, \quad (3.3.7-3)$$

where G is called the iteration matrix, which depends on M and \mathbf{s} ; \mathbf{g} is a column vector.

If \mathbf{u} is the exact solution, then from (3.3.7-1) and (3.3.7-3) we obtain,

$$M^{-1}\mathbf{s} = GM^{-1}\mathbf{s} + \mathbf{g}.$$

Hence, we have

$$\mathbf{g} = (I - G)M^{-1}\mathbf{s}. \quad (3.3.7-4)$$

Now (3.3.7-4) is called the consistency condition. If this consistency condition applies, then there is an r , say r_0 , such that,

$$\mathbf{u}^{(r_0+1)} = G\mathbf{u}^{(r_0)} + \mathbf{g} = G\mathbf{u} + \mathbf{g} = \mathbf{u}. \quad (3.3.7-5)$$

This means that as soon as the solution is obtained, further iterations do not modify the successive iterates.

Now suppose that the matrix M is partitioned as

$$M = H - T,$$

where H and T are square matrices and H is nonsingular. Then (3.3.2-23) becomes,

$$Hu = Tu + s. \quad (3.3.7-6)$$

By introducing the iteration count r in (3.3.7-6), we obtain the iterative method,

$$Hu^{(r+1)} = Tu^{(r)} + s. \quad (3.3.7-7)$$

Thus, it is clear that we may have different iterative methods for various splittings of the matrix M .

Let the coefficient matrix M be split into the form,

$$M = D - L - U, \quad (3.3.7-8)$$

where D is a positive diagonal matrix in which the elements are the diagonal elements of M , and L and U are lower and upper triangular matrices with null diagonals respectively. Equation (3.3.2-23) then becomes

$$(D - L - U)u = s. \quad (3.3.7-9)$$

In this discussion, we shall briefly describe the basic iterative methods for solving systems of linear equations. These methods may be grouped into two classes; namely, the point-iterative methods and the block-iterative methods. In the point-iterative methods, each component of successive approximations to the solution is computed explicitly, while in the block methods, several systems of linear equations are solved at each stage of the computation. However, each of these systems is smaller than the original system derived directly from the problem.

a) JACOBI METHOD

Now consider (3.3.7-9). Since D is a positive diagonal matrix, D^{-1} exists. By letting

$$G_B = D^{-1}(L + U) \text{ and } g_B = D^{-1}s, \quad (3.3.7-10)$$

then (3.3.7-9) can be written in the form

$$u = G_B u + g_B. \quad (3.3.7-11)$$

By introducing the iteration count in (3.3.7-11), we obtain the Jacobi iterative method as

$$\mathbf{u}^{(r+1)} = G_B \mathbf{u}^{(r)} + \mathbf{g}_B, \quad (3.3.7-12)$$

where G_B and \mathbf{g}_B are defined in (3.3.7-10). The matrix G_B is known as the Jacobi iteration matrix.

b) JACOBI OVERRELAXATION (JOR) METHOD

This is a modified Jacobi method where the convergence of the approximation to the solution is accelerated using a real parameter $\omega > 1$.

Accordingly, we have from (3.3.7-12),

$$\mathbf{u}^{(r+1)} = \omega(G_B \mathbf{u}^{(r)} + \mathbf{g}_B) + (1 - \omega) \mathbf{I} \mathbf{u}^{(r)},$$

which upon simplifying gives,

$$\mathbf{u}^{(r+1)} = [\omega G_B + (1 - \omega) \mathbf{I}] \mathbf{u}^{(r)} + \omega \mathbf{g}_B. \quad (3.3.7-13)$$

where $G_\omega = [\omega G_B + (1 - \omega) \mathbf{I}]$ is the JOR iteration matrix and G_B as already defined in (3.3.7-10). Note that if $\omega = 1$, the JOR method reduces to the Jacobi method.

c) GAUSS-SEIDEL METHOD (GS)

Consider the matrix M given in (3.3.7-8), where again D is the diagonal of M and $-L$ and $-U$ are the strictly lower and upper triangular parts of M . Now it is reasonable to use the most current updates of the sequence of the approximations in the computation of the subsequent vectors of the unknowns. Thus we may write the iterative method as

$$D \mathbf{u}^{(r+1)} = L \mathbf{u}^{(r+1)} + U \mathbf{u}^{(r)} + \mathbf{s} \quad (3.3.7-14)$$

for $r = 0, 1, 2, \dots$ and $\mathbf{u}^{(0)}$ is the initial guess vector.

Since $D - L$ is just the lower triangular part of M , its inverse exists by assuming that the matrix M has nonzero

diagonal elements. Hence we can write (3.3.7-14) in the form (3.3.7-15) for $r = 0, 1, 2, \dots$,

$$\mathbf{u}^{(r+1)} = \mathbf{H}\mathbf{u}^{(r)} + \mathbf{d}, \quad (3.3.7-15)$$

where

$$\left. \begin{aligned} \mathbf{H} &= (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U} \\ \mathbf{d} &= (\mathbf{D} - \mathbf{L})^{-1}\mathbf{s}. \end{aligned} \right\} \quad (3.3.7-15a)$$

The iterative method defined by (3.3.7-15) is called the Gauss-Seidel iterative method and \mathbf{H} as defined in (3.3.7-15a) is known as the Gauss-Seidel (GS) iteration matrix.

d) SUCCESSIVE OVERRELAXATION (SOR) METHOD

The GS method defined in (3.3.7-15) can be modified by introducing an acceleration parameter ω as follows.

Let the GS iteration vector given by (3.3.7-14) be denoted by $\mathbf{v}^{(r+1)}$. Define

$$\mathbf{u}^{(r+1)} = \mathbf{u}^{(r)} + \omega(\mathbf{v}^{(r+1)} - \mathbf{u}^{(r)}). \quad (3.3.7-16)$$

Substituting the representation of $\mathbf{v}^{(r+1)}$ as given by (3.3.7-14) into (3.3.7-16), we obtain

$$\mathbf{u}^{(r+1)} = (1 - \omega)\mathbf{u}^{(r)} + \omega\mathbf{D}^{-1}[\mathbf{L}\mathbf{u}^{(r+1)} + \mathbf{U}\mathbf{u}^{(r)} + \mathbf{s}]$$

or

$$(\mathbf{D} - \omega\mathbf{L})\mathbf{u}^{(r+1)} = [\omega\mathbf{U} + (1 - \omega)\mathbf{D}]\mathbf{u}^{(r)} + \omega\mathbf{s}. \quad (3.3.7-17)$$

We assume that \mathbf{D} is nonsingular, then $(\mathbf{D} - \omega\mathbf{L})^{-1}$ exists. Hence, we can write (3.3.7-17) in the form,

$$\mathbf{u}^{(r+1)} = \mathcal{L}_\omega\mathbf{u}^{(r)} + (\mathbf{D} - \omega\mathbf{L})^{-1}\omega\mathbf{s}, \quad (3.3.7-18)$$

where $\mathcal{L}_\omega = (\mathbf{D} - \omega\mathbf{L})^{-1}[\omega\mathbf{U} + (1 - \omega)\mathbf{D}]$ denotes the SOR iteration matrix. Notice that if $\omega = 1$, the method

becomes the GS method and for $\omega < 1$ we call the method as successive underrelaxation (SUR).

Table (3.3.7) summarizes all the iterative methods we have just discussed, which can be described by the general formula (3.3.7-3).

METHOD	ITERATION MATRIX G	VECTOR e
JACOBI	$D^{-1}[L + U]$	$D^{-1}s$
JOR	$\omega D^{-1}[L + U] + [1 - \omega]I$	$\omega D^{-1}s$
GS	$[D - L]^{-1}U$	$[D - L]^{-1}s$
SOR	$[D - \omega L]^{-1}[\omega U + (1 - \omega)D]$	$[D - \omega L]^{-1}\omega s$

Table(3.3.7): Iterative Methods

3.3.8 CONVERGENCE OF ITERATIVE METHODS

In this section, we discuss some of the classical results of the convergence theorems for the basic iterative methods. These results and their proofs may be found in Varga[1962], Young[1971] and Ortega[1987].

Now consider the iterative method (3.3.7-3), that is,

$$\mathbf{u}^{(r+1)} = G\mathbf{u}^{(r)} + \mathbf{g} \quad (3.3.8-1)$$

for $r = 0, 1, 2, \dots$, which gives the approximate solutions of the linear system (3.3.2-23), that is,

$$M\mathbf{u} = \mathbf{s}. \quad (3.3.8-2)$$

By assuming that M is a nonsingular matrix and that $\bar{\mathbf{u}}$ is the exact solution of (3.3.8-2), then the iterative method (3.3.8-1) is consistent with (3.3.8-2) if

$$\bar{\mathbf{u}} = G\bar{\mathbf{u}} + \mathbf{g}. \quad (3.3.8-3)$$

Consequently, by subtracting (3.3.8-3) from (3.3.8-1), we obtain the basic error vector equation,

$$\mathbf{e}^{(r+1)} = G\mathbf{e}^{(r)}, \quad (3.3.8-4)$$

for $r = 0, 1, 2, \dots$, where $\mathbf{e}^{(1)} = \mathbf{u}^{(1)} - \bar{\mathbf{u}}$ is the error vector at the i^{th} iteration step. Hence by introducing an initial error vector $\mathbf{e}^{(0)}$, we have by induction, the error vector equation (3.3.8-4) written as

$$\mathbf{e}^{(r)} = G^r \mathbf{e}^{(0)}, \quad (3.3.8-5)$$

for $r = 1, 2, \dots$.

Now the error $\mathbf{e}^{(r)}$ converges to a null vector for any arbitrary $\mathbf{e}^{(0)}$ if and only if G^r converges to a null matrix as r increases. G^r converges to a null matrix if the spectral radius $\mu(G)$ of the matrix G is less than unity. Thus the convergence rate of the iterative scheme depends on how fast G^r converges to the null matrix. For proofs of these classical results, we refer to Varga[1962], Young[1971] and Ortega[1987].

If there is no eigenvector deficiency with G , then the eigenvectors form a complete set. Accordingly, we may expand the initial error vector, $\mathbf{e}^{(0)}$ in terms of the eigenvectors of G ,

$$\mathbf{e}^{(0)} = \sum_j a_j \mathbf{v}_j \quad (3.3.8-6)$$

where \mathbf{v}_j is an eigenvector satisfying

$$G\mathbf{v}_j = \lambda_j \mathbf{v}_j. \quad (3.3.8-7)$$

Substitute (3.3.8-6) into (3.3.8-5) and using (3.3.8-7), we obtain

$$\mathbf{e}^{(r)} = \sum_j a_j \lambda_j^r \mathbf{v}_j. \quad (3.3.8-8)$$

By taking norms, we have

$$\|\mathbf{e}^{(r)}\| = \left\| \sum_j a_j \lambda_j^r \mathbf{v}_j \right\| \leq \mu^r \|\mathbf{e}^{(0)}\|, \quad (3.3.8-9)$$

where μ is the spectral radius of G and is given by

$$\mu = \max_j |\lambda_j|. \quad (3.3.8-10)$$

In practical problems, the initial error vector, $\mathbf{e}^{(0)}$, is arbitrary. Therefore, we may use μ^r as a basis for comparing the convergence rate of various iterative methods.

Now define the convergence rate as

$$R = -\frac{d}{dr} \ln(\mu^r) = -\ln \mu. \quad (3.3.8-11)$$

Suppose that the eigenvalue corresponding to μ is a double root and an eigenvector deficiency occurs, then the above discussion is not valid (Wachspress[1966]). So we assume that G has only one double root equal to μ . By using the similarity transformation of G , we obtain its Jordan canonical form,

$$\underline{M} = P^{-1}MP$$

where \underline{M} may be written as

$$\underline{M} = \left[\begin{array}{cc|ccc} \mu & 1 & & & \\ 0 & \mu & & & \\ \hline & & \lambda_3 & & \\ & & & \lambda_4 & \\ & & & & \ddots \\ & & & & & \ddots \end{array} \right]. \quad (3.3.8-12)$$

Premultiplying (3.3.8-5) by P^{-1} , we obtain

$$P^{-1}\mathbf{e}^{(r)} = (P^{-1}MP)P^{-1}\mathbf{e}^{(r-1)} = \underline{M}^r P^{-1}\mathbf{e}^{(0)}. \quad (3.3.8-13)$$

By letting

$$\zeta^{(r)} = P^{-1}\mathbf{e}^{(r)} \quad (3.3.8-14)$$

we can write (3.3.8-13) as

$$\zeta^{(r)} = \underline{M}^r \zeta^{(0)}. \quad (3.3.8-15)$$

The initial vector $\zeta^{(0)}$ may be expressed as

$$\zeta^{(0)} = \sum_{j=1}^n a_j \bar{v}_j \quad (3.3.8-16)$$

where the \bar{v}_j 's with $j \neq 2$ are normalized eigenvectors of \underline{M} and \bar{v}_2 is the auxiliary vector. Since \underline{M} is the Jordan canonical form of M , the direction of each of the \bar{v}_j including for $j = 2$, is a unit vector.

Hence by combining (3.3.8-15) and (3.3.8-16), we obtain

$$\begin{aligned} \zeta^{(r)} &= \sum_{j=1}^n a_j \underline{M}^r \bar{v}_j \\ &= a_1 \mu^r \bar{v}_1 + a_2 (r\mu^{r-1} \bar{v}_1 + \mu^r \bar{v}_2) + \sum_{j=3}^n a_j \lambda_j^r \bar{v}_j. \end{aligned} \quad (3.3.8-17)$$

Now consider the terms in (3.3.8-17). The last term approaches zero faster than the first two terms. The first term in the parentheses occurs due to the eigenvector deficiency and is the dominant term as r approaches infinity. Thus we have the limit,

$$\zeta^{(r)} \rightarrow a_2 r \mu^{r-1} \bar{v}_1, \quad (3.3.8-18)$$

as r increases. We note that $r\mu^{r-1}$ is an increasing function until r approaches $-1/\ln \mu$, when it then tends to decrease as r increases further. Thus the decay rate of $r\mu^{r-1}$ may be defined by

$$R(r\mu^{r-1}) \equiv -\frac{d}{dr} \ln (r\mu^{r-1}) = -\left(\frac{1}{r} + \ln \mu\right). \quad (3.3.8-19)$$

Hence we see that as r becomes larger, the decay rate approaches (3.3.8-11).

3.3.9 THE OPTIMUM SOR PARAMETER

We shall now restrict the discussion on the SOR iterative method. Consider the SOR iterative method (3.3.7-18). It is not only restricted to the finite difference equation for the elliptic differential equation. A sufficient condition for (3.3.7-18) to be convergent is that M is positive definite (see Varga[1962], pg.77). This condition is useful if the SOR method is applied to the finite element method (see Nakamura[1977], chapter 7). However, even when M is not symmetric nor positive definite, the SOR method is convergent provided all the diagonal elements of M are positive and M is irreducibly diagonally dominant (see Nakamura[1977], section 8.3).

In the remaining parts of this section, we shall study the eigenvectors-and-eigenvalues relationship of the SOR iteration matrix and then derive the optimum SOR parameter ω_{opt} .

From the SOR method (3.3.7-18), we have

$$L_{\omega} = (D - \omega L)^{-1}[\omega U + (1 - \omega)D] \quad (3.3.9-1)$$

$$g_{\omega} = (D - \omega L)^{-1}\omega s. \quad (3.3.9-2)$$

Let η_j be the eigenvectors of L_{ω} and γ_j the corresponding eigenvalues, then we have

$$L_{\omega}\eta_j = \gamma_j\eta_j. \quad (3.3.9-3)$$

Assume that the matrix M is consistently ordered, and partitioned into the form (3.3.7-8).

$$\begin{aligned}
D_{\alpha} \mathbf{u}_{\alpha}^{(r+1)} - \omega L_{\alpha} \mathbf{u}_{\alpha-1}^{(r+1)} \\
= (1 - \omega) D_{\alpha} \mathbf{u}_{\alpha}^{(r)} + \omega U_{\alpha} \mathbf{u}_{\alpha+1}^{(r)} + \omega \mathbf{s}_{\alpha}, \quad (3.3.9-7)
\end{aligned}$$

where (3.3.7-18) is premultiplied by $(D - \omega L)$.

Similarly, (3.3.9-3) can be written in the form

$$\gamma_j [D_{\alpha} \eta_{j\alpha} - \omega L_{\alpha} \eta_{j\alpha-1}] = (1 - \omega) D_{\alpha} \eta_{j\alpha} + \omega U_{\alpha} \eta_{j\alpha+1} \quad (3.3.9-8)$$

where $\eta_{j\alpha}$ is the α^{th} block subvector of η_j .

Now define the eigenvectors ϕ_j of the Jacobi iteration matrix G_B by

$$G_B \phi_j = \lambda_j \phi_j \quad (3.3.9-9)$$

where λ_j are the corresponding eigenvalues. These eigenvalues are such that

$$\lambda_n < \lambda_{n-1} < \dots < \lambda_2 < \lambda_1. \quad (3.3.9-10)$$

The eigenvectors η_j of the SOR iterative method are related to the eigenvectors ϕ_j of the Jacobi iterative method by

$$\eta_{j\alpha} = \gamma_j^{\alpha/2} \phi_{j\alpha}. \quad (3.3.9-11)$$

Substitute (3.3.9-11) into (3.3.9-8), we obtain

$$\begin{aligned}
\gamma_j [D_{\alpha} \gamma_j^{\alpha/2} \phi_{j\alpha} - \omega L_{\alpha} \gamma_j^{\alpha/2-1/2} \phi_{j\alpha-1}] \\
= (1 - \omega) D_{\alpha} \gamma_j^{\alpha/2} \phi_{j\alpha} + \omega U_{\alpha} \gamma_j^{\alpha/2+1/2} \phi_{j\alpha+1}. \quad (3.3.9-12)
\end{aligned}$$

Divide throughout by $\gamma_j^{\alpha/2}$ and simplify, (3.3.9-12)

becomes

$$(\gamma_j + \omega - 1) D_{\alpha} \phi_{j\alpha} = \omega \gamma_j^{1/2} [L_{\alpha} \phi_{j\alpha-1} + U_{\alpha} \phi_{j\alpha+1}]. \quad (3.3.9-13)$$

Since we may express the Jacobi iteration matrix G_B in terms of the submatrices D_α , L_α and U_α as

$$G_B = D_\alpha^{-1} [L_\alpha + U_\alpha], \quad (3.3.9-14)$$

and (3.3.9-9) in the form

$$D_\alpha^{-1} L_\alpha \phi_{j\alpha-1} + D_\alpha^{-1} U_\alpha \phi_{j\alpha+1} = \lambda_j \phi_{j\alpha} \quad (3.3.9-15)$$

where $\phi_{j\alpha}$ is the α^{th} subvector of ϕ_j , therefore by applying (3.3.9-15) to the right-hand side of (3.3.9-13), we obtain

$$(\gamma_j + \omega - 1) D_\alpha \phi_{j\alpha} = \omega \gamma_j^{1/2} \lambda_j D_\alpha \phi_{j\alpha}. \quad (3.3.9-16)$$

Thus by equating coefficients of both the left and right-hand sides of (3.3.9-16) we have

$$\gamma_j + \omega - 1 = \omega \gamma_j^{1/2} \lambda_j. \quad (3.3.9-17)$$

Hence γ_j is the root of (3.3.9-17) and (3.3.9-11) is proved.

Now the eigenvalues γ_j of the SOR iterative matrix are obtained by solving (3.3.9-17) for $\gamma_j^{1/2}$,

$$\gamma_j^{1/2} = \frac{\omega \lambda_j}{2} \pm \left[\left(\frac{\omega \lambda_j}{2} \right)^2 - \omega + 1 \right]^{1/2}. \quad (3.3.9-18)$$

By squaring both sides, (3.3.9-18) becomes

$$\gamma_j^\pm = \frac{(\omega \lambda_j)^2}{2} - \omega + 1 \pm \omega |\lambda_j| \left[\left(\frac{\omega \lambda_j}{2} \right)^2 - \omega + 1 \right]^{1/2}. \quad (3.3.9-19)$$

We have for the Jacobi iteration matrix; $|\lambda_j|$ and $-|\lambda_j|$ are both eigenvalues. Therefore each λ_j yields an identical pair of γ_j 's.

Now γ_j^\pm become complex if

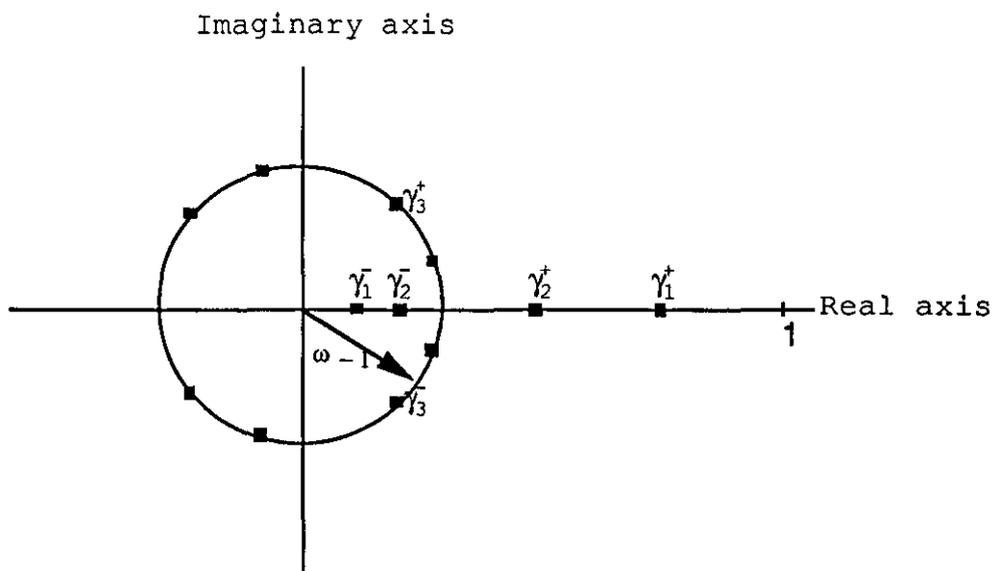
$$\lambda_j^2 < \frac{4(\omega - 1)}{\omega^2}, \quad (3.3.9-20)$$

and this results in

$$|\lambda_j| = \sqrt{\gamma_j^+ \gamma_j^-} = \omega - 1, \quad (3.3.9-21)$$

which is a constant.

Thus (3.3.9-21) states that for all λ_j satisfying (3.3.9-20), the corresponding γ_j 's lie on the circumference of a circle of radius $\omega - 1$. For those λ_j not satisfying (3.3.9-20), γ_j^+ is greater than $\omega - 1$ and γ_j^- is less than $\omega - 1$. The distribution of the SOR eigenvalues are illustrated in Figure(3.3.9), where γ_j^+ and γ_j^- are assumed to correspond to λ_j . Notice that the positions of the SOR eigenvalues change as ω is changed.



Figure(3.3.9): Distribution of SOR eigenvalues

As ω is gradually increased, the radius of the circle is increased; γ_2^+ and γ_2^- meet at $\gamma = \omega - 1$ and then split into

two complex eigenvalues on the circle. If ω is further increased, γ_1^+ is decreased and γ_1^- increased until γ_1^+ and γ_1^- meet at $\omega - 1$, when ω satisfies the relationship (3.3.9-22) and (3.3.9-23),

$$\left(\frac{\omega\mu(G_B)}{4}\right)^2 - \omega + 1 = 0, \quad (3.3.9-22)$$

and

$$\gamma_1^+ = \gamma_1^- = \omega - 1, \quad (3.3.9-23)$$

where $\mu(G_B) = \lambda_1$ is the spectral radius of the Jacobi iteration matrix G_B . The minimum spectral radius is attained when (3.3.9-22) is satisfied, thus giving the optimum SOR relaxation factor ω_{opt} as

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - (\mu(G_B))^2}}, \quad (3.3.9-24)$$

where we have chosen the smaller root of (3.3.9-22). Hence the minimum spectral radius of the SOR method is given by

$$\mu(L_\omega) = \omega_{opt} - 1. \quad (3.3.9-25)$$

Thus if a sufficiently large number of iteration is allowed, it can be shown that the use of ω_{opt} results in the fastest convergence. However, in practice, the iterations are stopped when a certain criterion is met or a prescribed limit of iteration count is attained. It must be recognized that when $\omega = \omega_{opt}$ is used, at least one eigenvector deficiency occurs with the double eigenvalue equal to μ as given by (3.3.9-23). The error decay is governed by $r\mu^{r-1}$ instead of μ^r as discussed in section 3.3.8. So by restricting the number of iterations, the effect of eigenvector deficiency is a serious drawback with ω_{opt} . In fact, if the maximum iteration count is restricted, an ω slightly larger than

ω_{opt} should result in a faster convergence even though the spectral radius is larger.

In chapter 6 we shall study the eigenvectors and eigenvalues relationship for the SOR iterative method applied to a periodic problem and show that the standard SOR parameter is not applicable for this type of problem. Instead we shall derive the SOR parameter for such a problem.

CHAPTER 4

NUMERICAL SOLUTION OF PROBLEMS INVOLVING ODES BY USING THE GM SINGLE STEP METHODS

4.1 INTRODUCTION

Suppose that the solution of the initial value problem,

$$y^{(1)} = f(x, y), \quad y(x_0) = y_0, \quad (4.1-1)$$

is approximated by an explicit single step method of the form,

$$Y_{n+1} = Y_n + h\Phi(x_n, Y_n; h), \quad (4.1-2)$$

where Y_{n+1} approximates $y(x_n + h)$ and let Φ be the increment function of the method.

The general explicit s-stage RK method, which is in fact a special case of (4.1-2) may be defined as

$$Y_{n+1} = Y_n + h\Phi(x, y; h), \quad (4.1-3a)$$

$$\left. \begin{aligned} \Phi(x, y; h) &= \sum_{i=1}^s w_i k_i \\ k_1 &= f(x, y) \\ &\cdot \\ &\cdot \\ k_r &= f\left(x + c_r h, y + h \sum_{i=1}^{r-1} a_{ri} k_i\right) \end{aligned} \right\} \quad (4.1-3b)$$

with

$$c_r = \sum_{i=1}^{r-1} a_{ri}, \quad \text{for } r = 2, 3, \dots, s. \quad (4.1-3c)$$

We observe that the s-stage RK method involves s function evaluations per step. Each of the functions $k_r(x, y; h)$, $r = 1, 2, \dots, s$, may be interpreted as an approximation to the derivative $y^{(1)}(x)$, and the increment function $\Phi(x, y; h)$ as a weighted average of

these approximations. Note that consistency constrains the conditions to $\sum_{i=1}^s w_i = 1$. By choosing values for the constants a_{ri} , w_i , c_r such that the expansion of the function $\Phi(x, y; h)$ defined by (4.1-3b) in powers of h differs from the truncated Taylor series expansion for $\Phi(x, y; h)$ only in the p^{th} and higher powers of h , then the method clearly has order $p-1$. We have assumed that $y(x)$ has sufficient continuous derivatives of at least order p on the closed interval $[a, b]$.

Similarly, in the light of the discussion above, we may now define the s -stage RK-GM method by y_{n+1} as given in (4.1-3a), k_r , $r = 1, 2, \dots, s$ as given in (4.1-3b), c_r , $r = 2, 3, \dots, s$ as given in (4.1-3c) but the increment function $\Phi(x, y; h)$ is defined by

$$\Phi(x, y; h) = \sum_{i,j=1}^s w_{i,j} \sqrt{k_i k_j}, \quad (4.1-3d)$$

where w_i is now replaced by $w_{i,j}$.

In the following sections we shall derive the GM single step methods which correspond to (4.1-2) and then the 3-stage third-order and 4-stage fourth-order RK-GM methods. As a further extension to the latter case, we shall develop a new adaptive strategy based on the combination of the new RK-GM method with the classical RK method. Finally we show how the RK-GM methods may be extended to systems of ODEs.

4.2 DERIVATION OF COMPOSITE SINGLE STEP GM METHODS

Recall the initial-value problem given in (4.1-1) of the first order ODE

$$y^{(1)} = f(x, y), \quad y(x_0) = y_0.$$

Lambert[1973] defines the class of linear single step methods of order one as

$$y_{n+1} = y_n + h\{\theta f_n + (1 - \theta) f_{n+1}\}. \quad (4.2-1)$$

This method has the truncation error of (in terms of θ)

$$T_{n+1} = (\theta - \frac{1}{2}) h^2 y_n^{(2)} + (\frac{\theta}{2} - \frac{1}{3}) h^3 y_{n+1}^{(3)}. \quad (4.2-2)$$

This error is smallest when $\theta = \frac{1}{2}$ and hence the method is of order 2. It is A-stable if and only if $\theta \leq \frac{1}{2}$ and the truncation error is $\frac{-1}{12} h^3 y_{n+1}^{(3)}$ when $\theta = \frac{1}{2}$. The method given by (4.2-1) is also known as the θ -method.

We define the general nonlinear single step GM method or the composite single step GM method by

$$y_{n+1} = y_n + h\{\alpha_1 f_n + \alpha_2 f_{n+1} + \alpha_3 \sqrt{f_n f_{n+1}}\}. \quad (4.2-3)$$

where the constant coefficients α_i ; $i = 1, 2, 3$ are to be determined depending on the order of the method. We shall now derive the explicit form of (4.2-3) by determining the values of the α_i , $i = 1, 2, 3$.

First we introduce the notation

$$f = f(x, y), \quad f^{(1)} = \frac{df}{dx}, \quad f^{(2)} = \frac{d^2 f}{dx^2}, \quad f^{(3)} = \frac{d^3 f}{dx^3} \quad \text{etc.}$$

By using the Taylor series expansion of f_{n+1} about x_n and with the help of the REDUCE program for symbolic manipulation, we obtain

$$f_{n+1} = f_n + h f_n^{(1)} + \frac{h^2}{2!} f_n^{(2)} + \frac{h^3}{3!} f_n^{(3)} + \frac{h^4}{4!} f_n^{(4)} + O(h^5) \quad (4.2-4)$$

and

$$f_n f_{n+1} = f_n^2 \{1 + F\} + O(h^5) \quad (4.2-5)$$

where

$$F = \frac{h}{f_n} [f_n^{(1)} + \frac{h}{2!} f_n^{(2)} + \frac{h^2}{3!} f_n^{(3)} + \frac{h^3}{4!} f_n^{(4)}]. \quad (4.2-6)$$

On substituting (4.2-6) into (4.2-5) and neglecting $O(h^4)$ approximations, we obtain

$$\sqrt{f_n f_{n+1}} \approx f_n \sqrt{1 + F} . \quad (4.2-7)$$

On expanding $\sqrt{1 + F}$, we have

$$\begin{aligned} \sqrt{1 + F} \approx f_n \left\{ 1 + \frac{h}{2f_n} [f_n^{(1)} + \frac{h}{2!} f_n^{(2)} + \frac{h^2}{3!} f_n^{(3)}] \right. \\ \left. + \frac{(1/2)(-1/2)}{2} \left(\frac{h}{f_n}\right)^2 [f_n^{(1)} + \frac{h}{2!} f_n^{(2)}]^2 \right. \\ \left. + \frac{(1/2)(-1/2)(-3/2)}{6} \left(\frac{h}{f_n}\right)^3 [f_n^{(1)} + \frac{h}{2!} f_n^{(2)}]^3 \right\} . \end{aligned}$$

Therefore

$$\begin{aligned} \sqrt{1 + F} \approx f_n \left\{ 1 + \frac{h f_n^{(1)}}{2 f_n} + \frac{h^2}{8} \left[2 \frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n}\right)^2 \right] \right. \\ \left. + \frac{h^3}{48} \left[4 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n}\right)^3 \right] \right\} . \quad (4.2-8) \end{aligned}$$

On substituting (4.2-4) and (4.2-8) into (4.2-3), we obtain

$$\begin{aligned} Y_{n+1} \approx Y_n + (\alpha_1 + \alpha_2 + \alpha_3) h f_n + \left(\alpha_2 + \frac{\alpha_3}{2}\right) h^2 f_n^{(1)} \\ + \left(\frac{\alpha_2}{2} + \frac{\alpha_3}{4}\right) h^3 f_n^{(2)} + \left(\frac{\alpha_2}{6} + \frac{\alpha_3}{12}\right) h^4 f_n^{(3)} \\ + \left(-\frac{\alpha_3}{16}\right) h^3 \left\{ 2 \frac{(f_n^{(1)})^2}{f_n} + 2h \frac{f_n^{(1)} f_n^{(2)}}{f_n} - h \frac{(f_n^{(1)})^3}{f_n} \right\} . \quad (4.2-9) \end{aligned}$$

But by expanding Y_{n+1} about x_n , we have

$$Y_{n+1} \approx Y_n + h f_n + \frac{1}{2} h^2 f_n^{(1)} + \frac{1}{6} h^3 f_n^{(2)} + \frac{1}{24} h^4 f_n^{(3)} . \quad (4.2-10)$$

Now by equating the coefficients of like terms in (4.2-9) and (4.2-10) we arrive at the following consistency conditions :

$$hf_n : \quad \alpha_1 + \alpha_2 + \alpha_3 = 1 \quad (4.2-11a)$$

$$h^2 f_n^{(1)} : \quad \alpha_2 + \frac{\alpha_3}{2} = \frac{1}{2} \quad (4.2-11b)$$

for the method to be of order 2 accurate.

There are now two equations in three unknowns. Thus there is one arbitrary parameter to choose. Therefore the parameters of the method are obtained as

$$\left. \begin{aligned} \alpha_1 &= \alpha_2 = \alpha \\ \alpha_3 &= 1 - 2\alpha \end{aligned} \right\} \quad (4.2-12)$$

and the resulting method may be written in the form

$$Y_{n+1} = Y_n + h \left[\alpha(f_n + f_{n+1}) + (1 - 2\alpha)\sqrt{f_n f_{n+1}} \right] \quad (4.2-13)$$

where α is an arbitrary constant.

We note that the Trapezoidal method can be deduced from (4.2-13) by letting $\alpha = \frac{1}{2}$; that is

$$Y_{n+1} = Y_n + \frac{h}{2} [f_n + f_{n+1}]. \quad (4.2-14)$$

If $\alpha = 0$ we obtain the original GM method as

$$Y_{n+1} = Y_n + h\sqrt{f_n f_{n+1}}. \quad (4.2-15)$$

The method defined in (4.2-13) has the truncation error given by

$$T_{n+1}^{GM} = y(x_{n+1}) - Y_{n+1}. \quad (4.2-16)$$

The Taylor series expansion of $y(x_{n+1})$ about x_n is given by

$$y(x_{n+1}) = Y_n + hf_n + \frac{h^2}{2!} f_n^{(1)} + \frac{h^3}{3!} f_n^{(2)} + O(h^4). \quad (4.2-17)$$

We have from (4.2-9), (4.2-11a), (4.2-11b) and (4.2-12),

$$Y_{n+1} = Y_n + hf_n + \frac{h^2}{2!} f_n^{(1)} + \frac{h^3}{8} \left[2f_n^{(2)} - (1 - 2\alpha) \frac{(f_n^{(1)})^2}{f_n} \right] + O(h^4). \quad (4.2-18)$$

Therefore, by subtracting (4.2-18) from (4.2-17) and assuming that the method is of order 2 accurate, we obtain the truncation error as

$$T_{n+1}^{GM} = h^3 \left[-\frac{1}{12} f_n^{(2)} + \frac{1-2\alpha}{8} \frac{(f_n^{(1)})^2}{f_n} \right]. \quad (4.2-19)$$

Now write $2\alpha = \theta$, $\frac{1}{2}(f_n + f_{n+1}) = F_n$ and $\sqrt{f_n + f_{n+1}} = G_n$ in (4.2-13), then (4.2-13) becomes

$$Y_{n+1} = Y_n + h \{ \theta F_n + (1 - \theta) G_{n+1} \}. \quad (4.2-20)$$

By comparing the forms of (4.2-1) and (4.2-20), the composite GM method given by (4.2-20) is nonlinear since G_n is nonlinear. Furthermore, (4.2-20) contains as special cases, the Trapezoidal ($\theta = 1$) and the original GM ($\theta = 0$) methods.

4.2.1 ACCURACY AND STABILITY ANALYSIS OF EQUATION (4.2-20)

To recapitulate, the truncation error of (4.2-20) is given as

$$T_{n+1}^{GM} = h^3 \left[-\frac{f_n^{(2)}}{12} + \frac{(1-\theta)(f_n^{(1)})^2}{8f_n} \right]. \quad (4.2.1-1)$$

Note that when $\theta = 1$, $T_{n+1}^{GM} = -\frac{h^3}{12} f_n^{(2)}$, which is the

truncation error of the Trapezoidal method. If it is possible to make $\theta = 1 - \frac{2}{3} \frac{f_n f_n^{(2)}}{(f_n^{(1)})^2}$, then the truncation

error T_{n+1}^{GM} given in (4.2.1-1) will vanish.

Now assume that

$$\theta = 1 - \frac{2}{3} \frac{f_n f_n^{(2)}}{(f_n^{(1)})^2}. \quad (4.2.1-2)$$

Next we write

$$\begin{aligned} Y_n^{(1)} &= f_n, \\ Y_n^{(2)} &= f_n^{(1)}, \\ &\approx \frac{(f_{n+1} - f_n)}{h}, \\ Y_n^{(3)} &\approx \frac{(Y_{n+1}^{(2)} - Y_n^{(2)})}{h}, \\ &\approx \frac{\frac{f_{n+2} - f_{n+1}}{h} - \frac{f_{n+1} - f_n}{h}}{h}, \\ &\approx \frac{f_{n+2} - 2f_{n+1} + f_n}{h^2}. \end{aligned}$$

Then

$$\begin{aligned} \theta &= 1 - \frac{2}{3} f_n \frac{\left[\frac{f_{n+2} - 2f_{n+1} + f_n}{h^2} \right]}{\left[\frac{f_{n+1} - f_n}{h} \right]^2}, \\ &= \frac{1}{3} \end{aligned}$$

provided

$$3 - 2f_n \frac{[f_{n+2} - 2f_{n+1} + f_n]}{[f_{n+1} - f_n]^2} = 1,$$

$$\text{or } f_n [f_{n+2} - 2f_{n+1} + f_n] = [f_{n+1} - f_n]^2,$$

$$\text{i.e. } f_{n+1}^2 = f_n f_{n+2}. \quad (4.2.1-3)$$

Now if $\theta = \frac{1}{3}$, (4.2-20) becomes

$$Y_{n+1} = Y_n + \frac{h}{3} [F_n + 2G_n]. \quad (4.2.1-4)$$

By substituting $F_n = \frac{1}{2}(f_n + f_{n+1})$ and $G_n = \sqrt{f_n f_{n+1}}$, we obtain

$$Y_{n+1} = Y_n + \frac{h}{6} [f_n + f_{n+1} + 4\sqrt{f_n f_{n+1}}]. \quad (4.2.1-5)$$

4.2.2 NUMERICAL RESULTS FROM USING (4.2.1-5)

Problem 1 $y^{(1)} = e^{-x}$

Initial condition $x_0 = 0, y(0) = 1.$

The exact solution is $y = -e^{-x} + 2.$

x_n	exact solution		computed solution	absolute error
.10	.10951625819640E+01	GM	.10951625852673E+01	.30162400221854E-08
		TR	.10952418709018E+01	.72399239220941E-04
.20	.11812692469220E+01	GM	.11812692532142E+01	.53266419441053E-08
		TR	.11814202794575E+01	.12785614784293E-03
.30	.12591817793183E+01	GM	.12591817883150E+01	.71448695996741E-08
		TR	.12593977281455E+01	.17149932658229E-03
.40	.13296799539644E+01	GM	.13296799654082E+01	.86064428726028E-08
		TR	.13299546414813E+01	.20658167867241E-03
.50	.13934693402874E+01	GM	.13934693539454E+01	.98014821757484E-08
		TR	.13937971767688E+01	.23526637574057E-03
.60	.14511883639060E+01	GM	.14511883795676E+01	.10792259460366E-07
		TR	.14515642915591E+01	.25904814459039E-03
.70	.15034146962086E+01	GM	.15034147136831E+01	.11623190800500E-07
		TR	.15038341385534E+01	.27899311203584E-03
.80	.15506710358828E+01	GM	.15506710549976E+01	.12326813608665E-07
		TR	.15511298519488E+01	.29588226993488E-03
.90	.15934303402594E+01	GM	.15934303608585E+01	.12927511672595E-07
		TR	.15939247831417E+01	.31030090854644E-03
1.0	.16321205588286E+01	GM	.16321205807707E+01	.13443921036412E-07
		TR	.16326472381873E+01	.32269635713130E-03
=====				
euclidean norm of the error :.317811E-07(GM)				
.762847E-03 (TR)				
=====				

Table (4.2.2a)

Problem 2 $y^{(1)} = 3x^2$

Initial condition $x_0 = 0, y(0) = 1.$

The exact solution is $y = x^3 + 1.$

x_n	exact solution	computed solution	absolute error
.10	.100100000000000E+01	GM .100050000000000E+01	.49950049950044E-03
		TR .100150000000000E+01	.49950049950067E-03
.20	.100800000000000E+01	GM .100700000000000E+01	.99206349206360E-03
		TR .100900000000000E+01	.99206349206360E-03
.30	.102700000000000E+01	GM .102550000000000E+01	.14605647517040E-02
		TR .102850000000000E+01	.14605647517043E-02
.40	.106400000000000E+01	GM .106200000000000E+01	.18796992481205E-02
		TR .106600000000000E+01	.18796992481205E-02
.50	.112500000000000E+01	GM .112250000000000E+01	.22222222222224E-02
		TR .112750000000000E+01	.22222222222226E-02
.60	.121600000000000E+01	GM .121300000000000E+01	.24671052631580E-02
		TR .121900000000000E+01	.24671052631582E-02
.70	.134300000000000E+01	GM .133950000000000E+01	.26061057334327E-02
		TR .134650000000000E+01	.26061057334328E-02
.80	.151200000000000E+01	GM .150800000000000E+01	.26455026455026E-02
		TR .151600000000000E+01	.26455026455028E-02
.90	.172900000000000E+01	GM .172450000000000E+01	.26026604973972E-02
		TR .173350000000000E+01	.26026604973977E-02
1.0	.200000000000000E+01	GM .199500000000000E+01	.24999999999998E-02
		TR .200500000000000E+01	.25000000000004E-02
=====			
euclidean norm of the error : .668875E-02 (GM)			.668875E-02 (TR)
=====			

Table (4.2.2b)

Notation

GM denotes the method (4.2.1-5)

TR denotes the Trapezoidal method (4.2-14).

We note that in problem 1,

$$Y_{n+1}^{(1)} = e^{-x_n - h},$$

$$Y_n^{(1)} = e^{-x_n},$$

and

$$Y_{n+2}^{(1)} = e^{-x_n - 2h},$$

therefore

$$\begin{aligned} (Y_{n+1}^{(1)})^2 &= e^{-2x_n-2h}, \\ &= (e^{-x_n})(e^{-x_n-2h}), \\ &= Y_n^{(1)} Y_{n+2}^{(1)}, \end{aligned}$$

which is the condition to be satisfied for method (4.2.1-5) to be competitive.

In problem 2, we have

$$Y_{n+1}^{(1)} = 3(x_n + h)^2.$$

Therefore

$$(Y_{n+1}^{(1)})^2 = 9(x_n + h)^4.$$

However,

$$Y_n^{(1)} = 3x_n^2$$

and

$$Y_{n+2}^{(1)} = 3(x_n + 2h)^2.$$

Therefore

$$Y_n^{(1)} Y_{n+2}^{(1)} = 9x_n^2(x_n + 2h)^2.$$

Hence for this problem, the condition (4.2.1-3) is satisfied only for $h \ll 1$.

We observe that in both problems, the GM method (4.2.1-5) gives better results. This is because the condition (4.2.1-3) is satisfied by the problems; especially problem 1. Thus we have shown that the method (4.2.1-5) is favourable for problems having the properties defined by (4.2.1-3).

4.2.3 STABILITY ANALYSIS OF (4.2-20)

Consider the test equation $y^{(1)} = \lambda y$ and the application of the composite GM method (4.2-20) to this problem. The following difference equation will be obtained.

$$Y_{n+1} = Y_n + h\lambda \left[\frac{\theta}{2}(Y_n + Y_{n+1}) + (1-\theta)\sqrt{Y_n Y_{n+1}} \right]. \quad (4.2.3-1)$$

Note that (4.2.3-1) is dependent on θ . If $\theta = 1$, we have the Trapezoidal rule and (4.2.3-1) reduces to

$$Y_{n+1} = Y_n + \frac{h\lambda}{2} [Y_n + Y_{n+1}]. \quad (4.2.3-2)$$

Write $\frac{Y_{n+1}}{Y_n} = Q_n$, we obtain

$$Q_n = 1 + \frac{h\lambda}{2} [1 + Q_n]. \quad (4.2.3-3)$$

On solving for Q_n in terms of h and λ , we have

$$Q_n = \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}. \quad (4.2.3-4)$$

Absolute stability requires that

$$\left| \frac{Y_{n+1}}{Y_n} \right| = |Q_n| = \left| \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}} \right| < 1. \quad (4.2.3-5)$$

Next we consider $\theta=0$. This reduces (4.2.3-1) to

$$Y_{n+1} = Y_n + h\lambda \sqrt{Y_n Y_{n+1}}. \quad (4.2.3-6)$$

By writing $\frac{Y_{n+1}}{Y_n} = Q_n$, we obtain

$$Q_n = 1 + h\lambda \sqrt{Q_n}. \quad (4.2.3-7)$$

Substitute $Q_n = P_n^2$, we obtain a quadratic in P_n ,

$$P_n^2 - h\lambda P_n - 1 = 0 \quad (4.2.3-8)$$

having two roots given by

$$P_{in} = \frac{h\lambda \pm \sqrt{(h\lambda)^2 + 4}}{2}, \quad i = 1, 2. \quad (4.2.3-9)$$

The conditions $|P_{in}| < 1$, $i = 1, 2$, imply that

$$\left| h\lambda \pm \sqrt{(h\lambda)^2 + 4} \right| < 2.$$

We shall now consider the cases corresponding to the root

$$P_{1n} = \frac{h\lambda + \sqrt{(h\lambda)^2 + 4}}{2}. \quad (4.2.3-10)$$

There are two cases to be considered. Firstly $h\lambda$ is real and secondly $h\lambda$ is purely imaginary.

Let $h\lambda = z$, where z can be real or imaginary.

Case(1) : $h\lambda$ is real.

Thus we obtain the function in z as

$$\tilde{f}(z) = z + \sqrt{z^2 + 4}. \quad (4.2.3-11)$$

We observe that

$$\tilde{f}(z) = z + \sqrt{(2+z)^2 - 4z}$$

and

$$|\tilde{f}(z)| = \left| z + \sqrt{(2+z)^2 - 4z} \right|.$$

Let $z = -x$ for $z < 0$ and $x > 0$. Then

$$\begin{aligned} |\tilde{f}(-x)| &= \left| -x + \sqrt{(2-x)^2 + 4x} \right| \\ &\leq \left| -x + \sqrt{(2+x)^2} \right| \\ &\leq |-x + 2 + x|. \end{aligned}$$

Hence

$$|\bar{f}(z)| < 2 \text{ for all } z < 0. \quad (4.2.3-12)$$

Similarly, we obtain

$$|\bar{f}(z)| > 2 \text{ for all } z > 0. \quad (4.2.3-13)$$

Case(2) : z is purely imaginary.

Let $z = ix$ where x is real. Then we obtain

$$\bar{f}(z) = ix + \sqrt{(ix)^2 + 4}$$

and

$$\begin{aligned} |\bar{f}(z)| &= \left| ix + \sqrt{4 - x^2} \right| \\ &= \sqrt{x^2 + 4 - x^2} \\ &= 2. \end{aligned}$$

Hence the method is absolutely stable for $h\lambda$ lying on the left half of the complex plane.

By considering the root $P_{2n} = \frac{h\lambda - \sqrt{(h\lambda)^2 + 4}}{2}$ and follow a similar discussion above, we can easily show that the method is absolutely stable for $h\lambda$ lying on the right half of the complex plane.

Thus the imaginary axis of the complex plane is the boundary for the region of absolute stability of the method.

Next, we consider the stability of (4.2.3-1) when $\theta = \frac{2}{3}$.

This reduces (4.2.3-1) to

$$Y_{n+1} = Y_n + \frac{h\lambda}{3} \left[Y_{n+1} + Y_n + \sqrt{Y_{n+1}Y_n} \right]. \quad (4.2.3-14)$$

On writing $\frac{Y_{n+1}}{Y_n} = Q_n^2$, we obtain after some rearrangement,

$$Q_n^2 \left(1 - \frac{h\lambda}{3}\right) - \frac{h\lambda}{3} Q_n - \left(1 + \frac{h\lambda}{3}\right) = 0. \quad (4.2.3-15)$$

The roots of (4.2.3-15), for $i = 1, 2$, are given as

$$Q_{in} = \frac{\frac{h\lambda}{3} \pm \sqrt{4 - \frac{(h\lambda)^2}{3}}}{2(1 - \frac{h\lambda}{3})}. \quad (4.2.3-16)$$

The conditions $|Q_{in}| < 1$, for $i = 1, 2$, imply that

$$\left| \frac{\frac{h\lambda}{3} \pm \sqrt{4 - \frac{(h\lambda)^2}{3}}}{1 - \frac{h\lambda}{3}} \right| < 2. \quad (4.2.3-17)$$

We shall now consider the root

$$Q_{1n} = \frac{\frac{h\lambda}{3} + \sqrt{4 - \frac{(h\lambda)^2}{3}}}{2(1 - \frac{h\lambda}{3})}. \quad (4.2.3-18)$$

As before, we need to consider two cases, firstly if $h\lambda$ is real and secondly if $h\lambda$ is purely imaginary.

Let $h\lambda = z$, where z can be real or imaginary.

Case(1) z is real.

Thus we have a function in z defined as

$$\bar{f}(z) = \frac{\frac{z}{3} + \sqrt{4 - \frac{z^2}{3}}}{1 - \frac{z}{3}}. \quad (4.2.3-19)$$

Now since $z^2 > 0$ for all z real, we notice that

$$|\bar{f}(z)| < \left| \frac{2 + \frac{z}{3}}{1 - \frac{z}{3}} \right|. \quad (4.2.3-20)$$

Let $h\lambda = -x$ for $h\lambda < 0$ and $x > 0$. Then from (4.2.3-20)

$$|\bar{f}(-x)| < \left| \frac{2 - \frac{x}{3}}{1 + \frac{x}{3}} \right| < 2. \quad (4.2.3-21)$$

Thus

$$|f(z)| < 2 \quad \text{for all } z < 0. \quad (4.2.3-22)$$

Similarly,

$$|f(z)| > 2 \quad \text{for all } z > 0. \quad (4.2.3-23)$$

Case(2) z is purely imaginary.

Let $z = ix$ where x is real. Therefore from (4.2.3-18), we obtain

$$f(z) = \frac{\frac{ix}{3} + \left\{\frac{x^2}{3} + 4\right\}^{1/2}}{1 - \frac{ix}{3}}. \quad (4.2.3-24)$$

By taking the modulus on both sides, we have

$$\begin{aligned} |f(z)| &= \left| \frac{\frac{ix}{3} + \left\{\frac{x^2}{3} + 4\right\}^{1/2}}{1 - \frac{ix}{3}} \right| \\ &= \left\{ \frac{4 + \frac{4}{9}x^2}{1 + \frac{x^2}{9}} \right\}^{1/2} \\ &= 2. \end{aligned}$$

A similar conclusion as in the case of $\theta=0$ follows. That is, the imaginary axis of the complex plane is the boundary for the region of absolute stability of the method. The method is absolutely stable for $h\lambda$ lying on the left half and right half of the complex plane depending on which root is being taken as given by (4.2.3-26).

4.3 DERIVATION OF COMPOSITE RK-GM METHODS

At the beginning of this chapter we have defined the s -stage RK-GM method as that given by (4.1-3a), (4.1-3b), (4.1-3c) and (4.1-3d). From this we can derive the RK-GM methods of various orders depending on the accuracy of

the Taylor series being used in the evaluation of the various terms involved.

We note that as in the case of the standard RK method, the first-order formula involves only a single function evaluation. Thus, the first-order RK-GM method is identical to the Euler method, i.e.

$$\left. \begin{aligned} Y_{n+1} &= Y_n + hk_1 \\ k_1 &= f(x_n, Y_n). \end{aligned} \right\} \quad (4.3-1)$$

4.3.1 SECOND-ORDER RK-GM METHOD

The second-order two-stage RK method is given by

$$Y_{n+1} = Y_n + \frac{h}{2} [w_1 k_1 + w_2 k_2] \quad (4.3.1-1)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + c_1 h, Y_n + a_{21} h k_1). \end{aligned} \right\} \quad (4.3.1-2)$$

Typical parameters for (4.3.1-1) and (4.3.1-2) are

$$\begin{aligned} c_1 &= 1, \quad a_{21} = 1, \\ w_1 &= w_2 = 1. \end{aligned}$$

The corresponding RK-GM method is of the form

$$Y_{n+1} = Y_n + h [w_1 k_1 + w_2 k_2 + w_3 \sqrt{k_1 k_2}] \quad (4.3.1-3)$$

with k_1 and k_2 given by (4.3.1-2). The coefficients c_1 , a_{21} , and w_i , $i = 1, 2, 3$ are to be determined so that (4.3.1-3) will have the highest accuracy possible.

By using the Taylor series expansion of k_2 about (x_n, Y_n) , we obtain

$$\begin{aligned}
k_2 &= f + c_1 h f_x + a_{21} h f f_y \\
&\quad + \frac{1}{2} h^2 [c_1^2 f_{xx} + 2c_1 a_{21} f f_{xy} + (a_{21} f)^2 f_{yy}] + \dots \\
&= f \left\{ 1 + h \left[c_1 \frac{f_x}{f} + a_{21} f_y \right. \right. \\
&\quad \left. \left. + \frac{1}{2} h \left(c_1^2 \frac{f_{xx}}{f} + 2c_1 a_{21} f_{xy} + a_{21}^2 f f_{yy} \right) \right] \right\} + \dots \quad (4.3.1-4)
\end{aligned}$$

where for simplicity, we have denoted $f=f(x,y)$, $f_x=f_x(x,y)$, $f_y=f_y(x,y)$, $f_{xx}=f_{xx}(x,y)$, $f_{xy}=f_{xy}(x,y)$ and $f_{yy}=f_{yy}(x,y)$ for all (x,y) in the domain of integration.

Now

$$\begin{aligned}
k_1 k_2 &= f^2 \left\{ 1 + h \left[c_1 \frac{f_x}{f} + a_{21} f_y + \frac{1}{2} h \left(c_1^2 \frac{f_{xx}}{f} \right. \right. \right. \\
&\quad \left. \left. \left. + 2c_1 a_{21} f_{xy} + a_{21}^2 f f_{yy} \right) \right] \right\} + \dots \quad (4.3.1-5)
\end{aligned}$$

By letting

$$\begin{aligned}
g_1 &= c_1 h \frac{f_x}{f} + a_{21} h f_y \\
&\quad + \frac{1}{2} h^2 \left[c_1^2 \frac{f_{xx}}{f} + 2c_1 a_{21} f_{xy} + a_{21}^2 f f_{yy} \right], \quad (4.3.1-6)
\end{aligned}$$

we obtain

$$\sqrt{k_1 k_2} \approx f \sqrt{1 + g_1}. \quad (4.3.1-7)$$

Therefore after simplifying and rearranging (4.3.1-7), we have

$$g_1 = \frac{k_1 k_2}{f^2} - 1. \quad (4.3.1-8)$$

Now reconsider the RK-GM method (4.3.1-3) and after substituting (4.3.1-7), we obtain

$$y_{n+1} \approx y_n + h \left[w_1 k_1 + w_2 k_2 + w_3 f \sqrt{1 + g_1} \right]. \quad (4.3.1-9)$$

Note that by substituting (4.3.1-6) into (4.3.1-4), we have

$$k_2 \approx f [1 + g_1]. \quad (4.3.1-10)$$

From (4.3.1-7), we obtain

$$\sqrt{k_1 k_2} \approx f \left[1 + \frac{1}{2} g_1 \right] \quad (4.3.1-11)$$

or

$$\begin{aligned} \sqrt{k_1 k_2} \approx f \left\{ 1 + \frac{h}{2} \left[c_1 \frac{f_x}{f} + a_{21} f_y \right. \right. \\ \left. \left. + \frac{1}{2} h \left(c_1^2 \frac{f_{xx}}{f} + 2c_1 a_{21} f_{xy} + a_{21}^2 f f_{yy} \right) \right] \right\}. \end{aligned} \quad (4.3.1-12)$$

Hence on substituting (4.3.1-4) and (4.3.1-12) into (4.3.1-3) and after some simplification and rearrangement we arrive at

$$\begin{aligned} Y_{n+1} \approx Y_n + (w_1 + w_2 + w_3) h f \\ + (w_2 + \frac{1}{2} w_3) h^2 f \left\{ c_1 \frac{f_x}{f} + a_{21} f_y \right. \\ \left. + \frac{1}{2} h \left(c_1^2 \frac{f_{xx}}{f} + 2c_1 a_{21} f_{xy} + a_{21}^2 f f_{yy} \right) \right\}. \end{aligned} \quad (4.3.1-13)$$

Now the Taylor series expansion of $y(x_{n+1})$ about x_n is given by

$$Y_{n+1} = Y_n + h Y_n^{(1)} + \frac{h^2}{2} Y_n^{(2)} + o(h^3). \quad (4.3.1-14)$$

Since

$$Y^{(1)} = f(x, y), \quad (4.3.1-15)$$

then

$$\begin{aligned} Y_{n+1} \approx Y_n + h f + \frac{h^2}{2} [f_x + f f_y] \\ + \frac{h^3}{6} [f_{xx} + 2f f_{xy} + f_x f_y + f^2 f_{yy} + f f_y^2]. \end{aligned} \quad (4.3.1-16)$$

By equating (4.3.1-13) and (4.3.1-16) we obtain,

$$\text{coefficient of } h f : w_1 + w_2 + w_3 = 1 \quad (4.3.1-17a)$$

$$\text{coefficient of } h^2 f_x : (w_2 + \frac{1}{2} w_3) c_1 = \frac{1}{2} \quad (4.3.1-17b)$$

$$\text{coefficient of } h^2 f f_y : (w_2 + \frac{1}{2} w_3) a_{21} = \frac{1}{2} \quad (4.3.1-17c)$$

Assume $w_2 \neq w_3 \neq 0$, then we obtain

$$c_1 = a_{21} = \frac{1}{\beta}, \quad (4.3.1-17d)$$

for an arbitrary constant β .

By substituting $c_1 = \frac{1}{\beta}$ into (4.3.1-17b) and rearranging we obtain

$$2w_2 + w_3 = \beta. \quad (4.3.1-17e)$$

Now solve the simultaneous equations (4.3.1-17a) and (4.3.1-17e) for w_1 , w_2 and w_3 , to obtain

$$\left. \begin{aligned} w_1 &= 1 - \alpha \\ w_2 &= \beta - \alpha \\ w_3 &= 2\alpha - \beta \end{aligned} \right\} \quad (4.3.1-18)$$

for some arbitrary constants α and β .

Thus, the general second-order 2-stage RK-GM method for some arbitrary parameters α and β is given by

$$Y_{n+1} = Y_n + h[(1 - \alpha)k_1 + (\beta - \alpha)k_2 + (2\alpha - \beta)\sqrt{k_1 k_2}], \quad (4.3.1-19a)$$

where

$$\left. \begin{aligned} k_1 &= f(x, y) \\ k_2 &= f(x_n + \rho h, Y_n + \rho h k_1) \end{aligned} \right\} \quad (4.3.1-19b)$$

and $\rho = \frac{1}{\beta}$.

As α and β are arbitrary constants, there are infinitely many formulae that can be derived from (4.3.1-19a) and (4.3.1-19b). One choice of β which tends to involve less work is 2. This results in the RK-GM formula of the form

$$Y_{n+1} = Y_n + h[(1 - \alpha)k_1 + (2 - \alpha)k_2 + 2(\alpha - 1)\sqrt{k_1 k_2}], \quad (4.3.1-20a)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + \frac{1}{2} h, Y_n + \frac{1}{2} h k_1) \end{aligned} \right\}. \quad (4.3.1-20b)$$

Now (4.3.1-20a) can be written in a more compact form to give

$$Y_{n+1} = Y_n + h \left[(1 - \alpha) (\sqrt{k_1} - \sqrt{k_2})^2 + k_2 \right]. \quad (4.3.1-20c)$$

Thus, if $\alpha=1$, we obtain the formula

$$Y_{n+1} = Y_n + h k_2, \quad (4.3.1-20d)$$

where k_2 is given in (4.3.1-20b) above.

Another reasonable choice of β is to set $\beta=1$. Hence from (4.3.1-19a) and (4.3.1-19b) we obtain another form of the RK-GM formula, which is given by

$$Y_{n+1} = Y_n + h \left[(1 - \alpha) (k_1 + k_2) + (2\alpha - 1) \sqrt{k_1 k_2} \right], \quad (4.3.1-20e)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + h, Y_n + h k_1) \end{aligned} \right\} \quad (4.3.1-20f)$$

Note that the classical RK method of order 2 can be deduced from (4.3.1-20e) by setting $\alpha=\frac{1}{2}$ to obtain the formula

$$Y_{n+1} = Y_n + \frac{h}{2} (k_1 + k_2). \quad (4.3.1-20g)$$

If we set $\alpha=1$, we shall obtain the original RK-GM method of order 2 given as

$$Y_{n+1} = Y_n + h \sqrt{k_1 k_2}. \quad (4.3.1-20h)$$

with k_1 and k_2 as defined in (4.3.1-20f) above.

Table (4.3.1) below lists some of the formulae that can be derived from (4.3.1-19a) and (4.3.1-19b) for various values of α and β .

$\beta = 1$ (4.3.1-20e)	$\alpha = 0$	$Y_{n+1} = Y_n + h[(\sqrt{k_1} - \sqrt{k_2})^2 + \sqrt{k_1 k_2}]$, $k_1 = f(x_n, Y_n), k_2 = f(x_n+h, Y_n+hk_1)$.
	$\alpha = 1$ (4.3.1-20h)	$Y_{n+1} = Y_n + h\sqrt{k_1 k_2}$, $k_1 = f(x_n, Y_n), k_2 = f(x_n+h, Y_n+hk_1)$.
	$\alpha = \frac{1}{2}$ (4.3.1-20g)	$Y_{n+1} = Y_n + \frac{h}{2} [k_1 + k_2]$, $k_1 = f(x_n, Y_n), k_2 = f(x_n+h, Y_n+hk_1)$.
$\beta = 2$ (4.3.1-20a)	$\alpha = 0$	$Y_{n+1} = Y_n + h[(\sqrt{k_1} - \sqrt{k_2})^2 + k_2]$, $k_1 = f(x_n, Y_n), k_2 = f(x_n+\bar{h}, Y_n+\bar{h}k_1)$.
	$\alpha = \frac{1}{2}$	$Y_{n+1} = Y_n + \bar{h}[(\sqrt{k_1} - \sqrt{k_2})^2 + 2k_2]$, $k_1 = f(x_n, Y_n), k_2 = f(x_n+\bar{h}, Y_n+\bar{h}k_1)$.
$\bar{h} = \frac{h}{2}$	$\alpha = 1$ (4.3.1-20d)	$Y_{n+1} = Y_n + hk_2$, $k_1 = f(x_n, Y_n), k_2 = f(x_n+\bar{h}, Y_n+\bar{h}k_1)$.

Table(4.3.1):Two-stage second-order RK-GM formulae.

4.3.1.1 ERROR ANALYSIS FOR THE SECOND-ORDER METHODS

Consider the second-order RK-GM methods derived in section 4.3.1. By using the definition of the local truncation error of a method, we therefore obtain the local truncation error of the second-order RK-GM methods as given by the difference between (4.3.1-16) and (4.3.1-13). Thus we have the desired local truncation error T_{n+1} as

$$T_{n+1} = (2w_2 + w_3) \frac{h^3}{12} \{ (3c_1^2 - 2)f_{xx} + 2(3c_1a_{21} - 2)ff_{xy} + (3a_{21}^2 - 2)f^2f_{yy} - 2(f_x + ff_y)f_y \}. \quad (4.3.1.1-1a)$$

We note that given a specific problem, the truncation error T_{n+1} is dependent on the method used. In other words, for a particular function f and its derivatives,

T_{n+1} is totally dependent on the parameters of the method used, namely the w_i , $i = 1, 2, 3$; c_1 and a_{21} .

Now, by using (4.3.1-17d) and (4.3.1-17e), we may simplify (4.3.1.1-1a) to obtain

$$T_{n+1}^{(\beta)} = \frac{h^3}{12} \{ (3\beta^{-1} - 2)G - 2Ff_y \}, \quad (4.3.1.1-1b)$$

where

$$\left. \begin{aligned} F &= f_x + ff_y \\ \text{and} \\ G &= f_{xx} + 2ff_{xy} + f^2f_{yy}. \end{aligned} \right\} \quad (4.3.1.1-2)$$

In section 4.3.1, we have derived two classes of the RK-GM methods. In this section we shall discuss their respective local truncation errors. First, we shall consider the local truncation error of (4.3.1-20a) which can be deduced from (4.3.1.1-1b) by substituting $\beta=2$ and is therefore given by

$$T_{n+1}^{(2)} = -\frac{h^3}{24} \{ G + 4Ff_y \}. \quad (4.3.1.1-3a)$$

Next, if we substitute $\beta=1$ in (4.3.1.1-1b), we shall obtain the truncation error of (4.3.1-20b), which is given as

$$T_{n+1}^{(1)} = \frac{h^3}{12} \{ G - 2Ff_y \}. \quad (4.3.1.1-3b)$$

We observe that the difference between the two local truncation errors is

$$\tau = T_{n+1}^{(1)} - T_{n+1}^{(2)}$$

$$\text{or} \quad \tau = \frac{h^3}{24} G, \quad (4.3.1.1-4)$$

where G is given in (4.3.1.1-2).

Now τ is positive provided $G = f_{xx} + 2ff_{xy} + f^2f_{yy}$ is positive. Thus $T_{n+1}^{(1)}$ is always greater than $T_{n+1}^{(2)}$ for positive values of G . In other words, formula (4.3.1-20e) will be less accurate than formula (4.3.1-20a) when applied to a function f such that G is always positive in the interval of integration.

From (4.3.1.1-1b), we can deduce the principal error function for the general second-order RK-GM method as

$$\Psi(x, y) = \frac{1}{12} \{ (3\beta^{-1} - 2)G - 2Ff_y \}. \quad (4.3.1.1-5)$$

By following an argument originally suggested by Lotkin[1951], we can find a bound for $\Psi(x, y)$, if we assume that the following bounds for f and its partial derivatives hold for $x \in [a, b]$, $y \in (-\infty, \infty)$:

$$|f(x, y)| < Q, \quad \left| \frac{\partial^{i+j} f(x, y)}{\partial x^i \partial y^j} \right| < \frac{P^{i+j}}{Q^{j-1}}, \quad i+j \leq p, \quad (4.3.1.1-6)$$

where P and Q are positive constants, and p is the order of the method. In this case, we have $p = 2$. Hence using (4.3.1.1-6), we obtain the following:

$$\left. \begin{aligned} |f_y| &< P \\ |F| = |f_x + ff_y| &< 2PQ \\ |G| = |f_{xx} + 2ff_{xy} + f^2f_{yy}| &< 4P^2Q. \end{aligned} \right\} \quad (4.3.1.1-7)$$

By substituting (4.3.1.1-7) into (4.3.1.1-5), we obtain

$$|\Psi(x, y)| < \frac{1}{3} [|3\beta^{-1} - 2| + 1] P^2Q, \quad (4.3.1.1-8) \quad \leftarrow$$

and the bound for the principal local truncation error as

$$|\Psi(x, y)h^3| < \frac{1}{3} [1 + |3\beta^{-1} - 2|] h^3 P^2Q. \quad (4.3.1.1-9)$$

However, Henrici[1962], shows that the bound for the principal local truncation error is also a bound for the whole local truncation error T_{n+1} , even though the assumptions on the bounds for f and its partial derivatives are different from those of (4.3.1.1-6). This is a consequence of the fact that the RK method is a single-step explicit method (Lambert[1973]). Thus we may write instead of (4.3.1.1-9),

$$|T_{n+1}| < \frac{1}{3} [1 + |3\beta^{-1} - 2|] h^3 P^2 Q. \quad (4.3.1.1-10)$$

Hence the bounds for the methods for $\beta=1$ and $\beta=2$ are respectively obtained from (4.3.1.1-10) as

$$|T_{n+1}^{(1)}| < \frac{2}{3} h^3 P^2 Q \quad (4.3.1.1-10a)$$

and

$$|T_{n+1}^{(2)}| < \frac{1}{2} h^3 P^2 Q. \quad (4.3.1.1-10b)$$

Alternatively, the bound for the local truncation error can be found by the well known approach of Bieberbach; where, in the neighbourhood of

$$|x - x_0| < A, \quad |y - y_0| < B$$

we have

$$\left. \begin{aligned} |f(x,y)| < Q, \quad \left| \frac{\partial^{i+j} f(x,y)}{\partial x^i \partial y^j} \right| < \frac{N}{Q^{j-1}}, \quad i+j \leq 4 \\ |x - x_0| N < 1 \quad \text{and} \quad A Q < B. \end{aligned} \right\} \quad (4.3.1.1-11)$$

Thus the bound for the local truncation error of (4.3.1.-19a) is given by

$$|T_{n+1}| < \frac{h^3}{3} N Q (1 + N). \quad (4.3.1.1-12)$$

However, Lotkin showed numerically that the approach adopted in Lotkin[1951] gave a sharper bound than that adopted by Bieberbach (Lambert[1973]).

4.3.1.2 NUMERICAL RESULTS

Problem: $y^{(1)}(x) = e^{-x}$.

Initial condition $x_0 = 0, y_0 = 1$.

Exact solution $y(x) = -e^{-x} + 2$.

The three formulae used are as follows:

A) $(\alpha = -1, \beta = \frac{3}{20}) \quad y_{n+1} = y_n + \frac{h}{20}\{40k_1 + 23k_2 - 43\sqrt{k_1k_2}\}$

B) $(\alpha = 0, \beta = 1 \text{ (RK Method)}) \quad y_{n+1} = y_n + \frac{h}{2}\{k_1 + k_2\}$

C) $(\alpha = 1, \beta = 1 \text{ (Original GM)}) \quad y_{n+1} = y_n + \sqrt{k_1k_2}$.

In Table (4.3.1.2a) we list the errors at the end point of the computation for each formula derived from (4.3.1-19a). Table (4.3.1.2b) compares the results of using the second-order two-stage RK-GM method(A), the classical second-order two-stage RK method(B) and the original second-order two-stage RK-GM method(C). For the given problem, the RK-GM method(A) has the least error as compared with the other two methods despite it being computationally expensive to use as we can see from its form. However, as the results show, the RK-GM method(A) may be useful if we require good accuracy by using only a low-order method.

Parameters of formula (4.3.1-19a)		Results at $x_n = 1.00$	
α	β	Numerical solution at x_n	Error at x_n $e_n = y(x_n) - y_n $
0	1	.1573158E+01	.5896286E-01
0	2	.1572094E+01	.6002608E-01
1/2	1	.1572443E+01	.5967767E-01
1/2	2	.1571911E+01	.6020928E-01
1	1	.1571728E+01	.6039248E-01
1	2	.1571728E+01	.6039248E-01
-1	1	.1574587E+01	.5753324E-01
-1	3/4	.1576344E+01	.5577667E-01
-1	1/2	.1580607E+01	.5151320E-01
-1	1/4	.1598490E+01	.3363025E-01
-1	3/20	.1631026E+01	.1094983E-02
-1	1/8	.1649765E+01	.1764405E-01

Table(4.3.1.2a): Numerical results from two-stage second-order RK-GM Formulae

x_n	Exact Solution	Numerical Solution	Error
.10	.1095163E+01	.1094998E+01 (A) .1086286E+01 (B) .1086071E+01 (C)	.1648442E-03 .8876563E-02 .9091785E-02
.20	.1181269E+01	.1180955E+01 (A) .1164361E+01 (B) .1163951E+01 (C)	.3140015E-03 .1690841E-01 .1731837E-01
.30	.1259182E+01	.1258733E+01 (A) .1235006E+01 (B) .1234420E+01 (C)	.4489645E-03 .2417592E-01 .2476210E-01
.40	.1329680E+01	.1329109E+01 (A) .1298928E+01 (B) .1298183E+01 (C)	.5710842E-03 .3075184E-01 .3149745E-01
.50	.1393469E+01	.1392788E+01 (A) .1356767E+01 (B) .1355877E+01 (C)	.6815826E-03 .3670198E-01 .3759186E-01
.60	.1451188E+01	.1450407E+01 (A) .1409102E+01 (B) .1408082E+01 (C)	.7815656E-03 .4208589E-01 .4310631E-01
.70	.1503415E+01	.1502543E+01 (A) .1456457E+01 (B) .1455319E+01 (C)	.8720341E-03 .4695745E-01 .4809598E-01
.80	.1550671E+01	.1549717E+01 (A) .1499306E+01 (B) .1498060E+01 (C)	.9538933E-03 .5136542E-01 .5261083E-01
.90	.1593430E+01	.1592402E+01 (A) .1538076E+01 (B) .1536734E+01 (C)	.1027963E-02 .5535392E-01 .5669603E-01
1.00	.1632121E+01	.1631026E+01 (A) .1573158E+01 (B) .1571728E+01 (C)	.1094983E-02 .5896286E-01 .6039248E-01

Table(4.3.1.2b):Results obtained from selected RK-GM formulae

4.3.2 THIRD-ORDER RK-GM METHOD

The standard third-order RK method for the problem (4.1-1) may be given by

$$Y_{n+1} = Y_n + h \sum_{i=1}^3 w_i k_i, \quad (4.3.2-1)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + c_1 h, Y_n + a_{21} h k_1) \\ k_3 &= f(x_n + c_2 h, Y_n + a_{31} h k_1 + a_{32} h k_2) \end{aligned} \right\} \quad (4.3.2-2)$$

A typical set of parameters used for the standard third-order RK method is

$$\left. \begin{aligned} c_1 &= \frac{1}{2}, & a_{21} &= \frac{1}{2}, \\ c_2 &= 1, & a_{31} &= -1, & a_{32} &= 2, \\ w_1 &= \frac{1}{6}, & w_2 &= \frac{2}{3}, & w_3 &= \frac{1}{6}. \end{aligned} \right\} \quad (4.3.2-3)$$

Thus we may write (4.3.2-1) as

$$Y_{n+1} = Y_n + \frac{h}{6} [k_1 + 4k_2 + k_3]. \quad (4.3.2-4)$$

The corresponding third-order composite RK-GM method may be defined by the formula

$$Y_{n+1} = Y_n + h [w_1 \sqrt{k_1 k_2} + w_2 \sqrt{k_2 k_3} + w_3 \sqrt{k_3 k_1} + w_4 k_1 + w_5 k_2 + w_6 k_3]. \quad (4.3.2-5)$$

where w_i ; $i = 1, 2, \dots, 6$ are to be determined so that the method is third-order accurate and the k_i ; $i = 1, 2, 3$ are as specified in (4.3.2-2) above.

The Taylor series expansion of Y_{n+1} about x_n is given by

$$Y_{n+1} = Y_n + h y_n^{(1)} + \frac{h^2}{2} y_n^{(2)} + \frac{h^3}{6} y_n^{(3)} + O(h^4), \quad (4.3.2-6)$$

where

$$\left. \begin{aligned} y^{(1)} &= f, & y^{(2)} &= f_x + f f_y, \\ y^{(3)} &= f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_x f_y + f f_y^2 \\ \text{and} \\ y^{(4)} &= f_{xxx} + 3f f_{xxy} \\ &\quad + 3f_x f_{xy} + 5f f_y f_{xy} + 3f f_x f_{yy} \\ &\quad + 3f^2 f_{xyy} + 4f^2 f_y f_{yy} + f_y f_{xx} \\ &\quad + f_x f_y^2 + f f_y^3 + f^3 f_{yyy}. \end{aligned} \right\} \quad (4.3.2-7)$$

Now expand k_2 and k_3 in (4.3.2-2) using the Taylor series expansion of a function of two variables and substitute the results in (4.3.2-5). Then use the REDUCE program to expand and obtain the right-hand side of (4.3.2-5). The left-hand side of (4.3.2-5) is given by (4.3.2-6). Hence, by equating the corresponding terms of the left- and right-hand sides of (4.3.2-5), we obtain the following results:

$$\text{coefficient of } hf: \quad \sum_{i=1}^6 w_i = 1,$$

$$\text{coefficient of } h^2 f_x: \quad \frac{1}{2} [c_1(w_1 + w_2 + 2w_5) + c_2(w_2 + w_3 + 2w_6)] = \frac{1}{2}$$

$$\text{coefficient of } h^2 ff_y: \quad \frac{1}{2} [a_{21}(w_1 + w_2 + 2w_5) + (a_{31} + a_{32})(w_2 + w_3 + 2w_6)] = \frac{1}{2}$$

$$\text{coefficient of } h^3 f_{xx}: \quad \frac{1}{4} [c_1^2(w_1 + w_2 + 2w_5) + c_2^2(w_2 + w_3 + 2w_6)] = \frac{1}{6}$$

$$\text{coefficient of } h^3 ff_{xy}: \quad \frac{1}{2} [c_1 a_{21}(w_1 + w_2 + 2w_5) + c_2(a_{31} + a_{32})(w_2 + w_3 + 2w_6)] = \frac{1}{3}$$

$$\text{coefficient of } h^3 ff_y^2: \quad -\frac{1}{8} [a_{21}^2(w_1 + w_2) - 2a_{21}a_{31}w_2 - 2a_{21}a_{32}(3w_2 + 2w_3 + 4w_6) + (a_{31} + a_{32})^2(w_2 + w_3)] = \frac{1}{6}$$

$$\text{coefficient of } h^3 f^2 f_{yy}: \quad \frac{1}{4} [a_{21}^2(w_1 + w_2 + 2w_5) + (a_{31} + a_{32})^2(w_2 + w_3 + 2w_6)] = \frac{1}{6}$$

$$\text{coefficient of } h^3 f_x f_y: \quad -\frac{1}{4} [c_1 a_{21}(w_1 + w_2) - c_1 a_{31} w_2 - c_2 a_{21} w_2 - c_1 a_{32}(3w_2 + 2w_3 + 4w_6) + c_2(a_{31} + a_{32})(w_2 + w_3)] = \frac{1}{6}$$

$$\text{coefficient of } h^3 \frac{f_x^2}{f}: \quad -\frac{1}{8} [c_1^2(w_1 + w_2) - 2c_1 c_2 w_2 + c_2^2(w_2 + w_3)] = 0$$

Hence the resulting equations of condition are obtained as follows:

$$\begin{aligned} \sum_{i=1}^6 w_i &= 1 \\ c_1(w_1 + w_2 + 2w_5) + c_2(w_2 + w_3 + 2w_6) &= 1 \\ a_{21}(w_1 + w_2 + 2w_5) + (a_{31} + a_{32})(w_2 + w_3 + 2w_6) &= 1 \\ c_1^2(w_1 + w_2 + 2w_5) + c_2^2(w_2 + w_3 + 2w_6) &= \frac{2}{3} \\ c_1 a_{21}(w_1 + w_2 + 2w_5) + c_2(a_{31} + a_{32})(w_2 + w_3 + 2w_6) &= \frac{2}{3} \\ a_{21}^2(w_1 + w_2) - 2a_{21}a_{31}w_2 - 2a_{21}a_{32}(3w_2 + 2w_3 + 4w_6) \\ &\quad + (a_{31} + a_{32})^2(w_2 + w_3) = \frac{4}{3} \\ a_{21}^2(w_1 + w_2 + 2w_5) + (a_{31} + a_{32})^2(w_2 + w_3 + 2w_6) &= \frac{2}{3} \\ c_1 a_{21}(w_1 + w_2) - c_1 a_{31}w_2 - c_2 a_{21}w_2 \\ &\quad - c_1 a_{32}(3w_2 + 2w_3 + 4w_6) + c_2(a_{31} + a_{32})(w_2 + w_3) = -\frac{2}{3} \\ c_1^2(w_1 + w_2) - 2c_1 c_2 w_2 + c_2^2(w_2 + w_3) &= 0 \end{aligned}$$

Recall the general s -stage RK method defined by (4.1-3a), (4.1-3b) and (4.1-3c). We may choose the c_i to cover the step interval while the a_{ij} 's be some simple (preferably linear) combination of the c_i 's. Now setting $c_1 = a_{21}$ and $c_2 = a_{31} + a_{32}$ in the preceding set of equations reduces those equations to a set of linear equations of the form

$$Mw = b, \quad (4.3.2-8)$$

where

$$\begin{aligned} \mathbf{w}^T &= (w_1, w_2, w_3, w_4, w_5, w_6), \\ \mathbf{b}^T &= (1, 1, \frac{2}{3}, -\frac{4}{3}, -\frac{2}{3}, 0) \end{aligned}$$

and

$$M = (m_{ij}), \text{ for } i, j = 1, 2, \dots, 6.$$

Hence the system of equations (4.3.2-8) may be solved for w in terms of the a_{ij} and c_i 's.

To simplify further the method to be derived and hence obtain a method which is computationally competitive, we

may choose the a_{ij} and c_i such that $a_{21} = c_1 = \frac{1}{2}$, $-a_{31} = c_2 = 1$ and $a_{32} = 2$. This leads to solving the system of equations of the form

$$M_1 \mathbf{w} = \mathbf{b}, \quad (4.3.2-9)$$

where

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{1}{2} & \frac{3}{2} & 1 & 0 & 1 & 2 \\ \frac{1}{4} & \frac{5}{4} & 1 & 0 & \frac{1}{2} & 2 \\ \frac{1}{4} & -\frac{15}{4} & -3 & 0 & 0 & -8 \\ -\frac{1}{4} & -\frac{7}{4} & -3 & 0 & 0 & -4 \\ \frac{1}{4} & \frac{1}{4} & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Using the REDUCE program for algebraic manipulation the solutions of (4.3.2-9) are obtained as

$$\left. \begin{aligned} w_5 &= \alpha_1, & w_6 &= \alpha_2 \\ w_1 &= \frac{1}{6}[4 - 9\alpha_1 + 12\alpha_2] \\ w_2 &= \frac{1}{6}[4 - 3\alpha_1 - 12\alpha_2] \\ w_3 &= \frac{1}{6}[3\alpha_1 - 2] \\ w_4 &= \frac{1}{2}[\alpha_1 - 2\alpha_2] \end{aligned} \right\} \quad (4.3.2-10)$$

where α_1 and α_2 are some arbitrary constants.

By letting $w_1 = w_2 = w_3 = 0$, that is $\alpha_1 = \frac{2}{3}$ and $\alpha_2 = \frac{1}{6}$, we obtain the standard RK method of order 3. The general composite RK-GM method for the particular choice of the a_{ij} and c_i is obtained as

$$Y_{n+1} = Y_n + h \left[w_1 \sqrt{k_1 k_2} + w_2 \sqrt{k_2 k_3} + w_3 \sqrt{k_3 k_1} + w_4 k_1 + w_5 k_2 + w_6 k_3 \right] \quad (4.3.2-11)$$

where w_i , $i = 1, 2, \dots, 6$ are given by (4.3.2-10) and

$$\left. \begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + \frac{1}{2}hk_1\right) \\ k_3 &= f\left(x_n + h, y_n + h(2k_2 - k_1)\right) \end{aligned} \right\} \quad (4.3.2-11a)$$

For $w_5 = w_6 = 0$, that is $\alpha_1 = \alpha_2 = 0$, we have the RK-GM method given by the formula

$$y_{n+1} = y_n + \frac{h}{3} \left[2(\sqrt{k_1 k_2} + \sqrt{k_2 k_3}) - \sqrt{k_1 k_3} \right], \quad (4.3.2-12)$$

and the k_i ' are as specified in (4.3.2-11a).

4.3.2.1 ERROR ANALYSIS OF THE THIRD-ORDER RK-GM METHOD

Consider the third-order RK-GM methods derived in section 4.3.2. By definition, the local truncation error of the method is given by the difference between (4.3.2-5) and (4.3.2-6) such that the method is third-order accurate. By using the REDUCE program, we obtain the local truncation error of the RK-GM method (4.3.2-12) as

$$T_{n+1} = \frac{h^4}{24} \left[f^2 f_y f_{yy} + f f_x f_{yy} - f f_y^3 + f f_y f_{xy} - f_x f_y^2 + f_x f_{xy} \right].$$

Thus the method defined by (4.3.2-12) is third-order accurate. This is shown numerically by the results given in section(4.3.2.2).

4.3.2.2 NUMERICAL RESULTS

Problem $y^{(1)} = -e^{-x}$.

Initial condition $x_0 = 0, y_0 = 1$.

For values of x in the interval $0 \leq x \leq 1$.

Table (4.3.2.2) compares the results of the numerical solutions of the problem obtained by using the RK-GM method (4.3.2-12) and the classical RK method (4.3.2-4). Both formulae are of the third-order, three-stage, Runge-Kutta class type of method. The numerical results show that the RK-GM method may give better results than the standard RK method for a certain class of problems.

x_n	Exact Solution	Numerical Solution	Error
.10	.9048374E+00	.9048337E+00 (A) .9048333E+00 (B)	.3735045E-05 .4084703E-05
.20	.8187308E+00	.8187240E+00 (A) .8187234E+00 (B)	.6759204E-05 .7391967E-05
.30	.7408182E+00	.7408090E+00 (A) .7408082E+00 (B)	.9173952E-05 .1003277E-04
.40	.6703200E+00	.6703090E+00 (A) .6703079E+00 (B)	.1106789E-04 .1210401E-04
.50	.6065307E+00	.6065181E+00 (A) .6065170E+00 (B)	.1251828E-04 .1369017E-04
.60	.5488116E+00	.5487980E+00 (A) .5487968E+00 (B)	.1359238E-04 .1486482E-04
.70	.4965853E+00	.4965709E+00 (A) .4965696E+00 (B)	.1434868E-04 .1569191E-04
.80	.4493290E+00	.4493141E+00 (A) .4493127E+00 (B)	.1483793E-04 .1622697E-04
.90	.4065697E+00	.4065546E+00 (A) .4065531E+00 (B)	.1510413E-04 .1651808E-04
1.00	.3678794E+00	.3678643E+00 (A) .3678628E+00 (B)	.1518528E-04 .1660682E-04

Table (4.3.2.2): Comparison of (A) RK-GM formula (4.3.2-12) and (B) RK formula (4.3.2-4).

4.3.3 FOURTH-ORDER METHOD

The standard fourth-order RK method for the problem (4.1-1) may be given by

$$Y_{n+1} = Y_n + h \sum_{i=1}^4 w_i k_i, \quad (4.3.3-1)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + c_1 h, Y_n + a_{21} h k_1) \\ k_3 &= f(x_n + c_2 h, Y_n + a_{31} h k_1 + a_{32} h k_2) \\ k_4 &= f(x_n + c_3 h, Y_n + a_{41} h k_1 + a_{42} h k_2 + a_{43} h k_3) \end{aligned} \right\} \quad (4.3.3-2)$$

A typical set of parameter values for the standard fourth-order RK method is

$$\begin{aligned} c_1 &= c_2 = a_{21} = a_{32} = \frac{1}{2}, \\ a_{31} &= a_{41} = a_{42} = 0, \end{aligned}$$

$$\begin{aligned}
c_3 &= a_{43} = 1, \\
w_1 &= w_4 = \frac{1}{6}, \\
w_2 &= w_3 = \frac{1}{3}.
\end{aligned}$$

Thus, we obtain the standard RK method of the form

$$Y_n = Y_n + \frac{h}{6} [k_1 + 2(k_2 + k_3) + k_4] \quad (4.3.3-3)$$

where

$$\left. \begin{aligned}
k_1 &= f(x_n, Y_n) \\
k_2 &= f(x_n + \frac{1}{2}h, Y_n + \frac{1}{2}hk_1) \\
k_3 &= f(x_n + \frac{1}{2}h, Y_n + \frac{1}{2}hk_2) \\
k_4 &= f(x_n + h, Y_n + hk_3).
\end{aligned} \right\} \quad (4.3.3-4)$$

We shall define a fourth-order composite RK-GM method by the formula

$$\begin{aligned}
Y_{n+1} = Y_n + h [&w_1\sqrt{k_1k_2} + w_2\sqrt{k_2k_3} + w_3\sqrt{k_3k_4} \\
&+ w_4\sqrt{k_4k_1} + w_5\sqrt{k_4k_2} + w_6\sqrt{k_4k_3} \\
&+ w_7k_1 + w_8k_2 + w_9k_3 + w_{10}k_4], \quad (4.3.3-5)
\end{aligned}$$

where the k_i , $i = 1, 2, 3, 4$ are specified in (4.3.3-2). The w_i , $i = 1, \dots, 10$ are to be determined so that the method is fourth-order accurate.

Next set the following parameters

$$\left. \begin{aligned}
c_1 &= c_2 = a_{21} = a_{32} = \frac{1}{2} \\
a_{31} &= a_{41} = a_{42} = 0 \\
c_3 &= a_{43} = 1.
\end{aligned} \right\} \quad (4.3.3-6)$$

so that the k_i , $i = 1, 2, 3, 4$ of the RK-GM formula are the same as those of the AM formula (4.3.3-3).

Then use the Taylor series expansion of the k_i ; $i = 1, 2, 3, 4$ and the REDUCE program to expand the square root terms of (4.3.3-5). By equating like terms of the left- and right-hand sides of (4.3.3-5), we obtain a set of equations which can be written in matrix form as

$$\mathbf{Aw} = \mathbf{b}, \quad (4.3.3-7)$$

where A is a 16 by 10 matrix of the form,

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 2 & 3 & 3 & 0 & 2 & 2 & 4 \\ 1 & 2 & 1 & 4 & 5 & 5 & 0 & 2 & 2 & 8 \\ -1 & 4 & 3 & 4 & 7 & 11 & 0 & 0 & 8 & 16 \\ -1 & 2 & 1 & 0 & 3 & 5 & 0 & 0 & 4 & 8 \\ 3 & 0 & -5 & 8 & 9 & 9 & 0 & 0 & 0 & 32 \\ 1 & 2 & 1 & 8 & 9 & 9 & 0 & 2 & 2 & 16 \\ 1 & 0 & -3 & 8 & 11 & 15 & 0 & 0 & 0 & 32 \\ -1 & 2 & 1 & 0 & 5 & 7 & 0 & 0 & 4 & 16 \\ -1 & 4 & 3 & 4 & 9 & 13 & 0 & 0 & 8 & 24 \\ -1 & 4 & 3 & 8 & 13 & 17 & 0 & 0 & 8 & 32 \\ -1 & 6 & 5 & 12 & 17 & 23 & 0 & 0 & 12 & 40 \\ -1 & 2 & 1 & -4 & 1 & 3 & 0 & 0 & 4 & 8 \\ 3 & 0 & -1 & 8 & 1 & 5 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 4 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 8 & 3 & 3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

\mathbf{b} is a column vector of the form,

$$\mathbf{b}^T = \left(1, 2, \frac{8}{3}, \frac{16}{3}, \frac{8}{3}, \frac{16}{3}, 4, \frac{16}{3}, 4, \frac{20}{3}, 8, \frac{32}{3}, \frac{8}{3}, 0, 0, 0 \right)$$

and

$$\mathbf{w}^T = \left(w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10} \right).$$

Premultiplying both sides of (4.3.3-7) by A^T gives

$$A^T A \mathbf{w} = A^T \mathbf{b} = \mathbf{d}, \quad (4.3.3-8)$$

where $B = A^T A$ is now a 10×10 square matrix. Hence the set of equations is now of the form

$$B \mathbf{w} = \mathbf{d}, \quad (4.3.3-9)$$

which, upon using the REDUCE program, produces the following results

$$\left. \begin{aligned}
 w_8 &= \alpha_0, \quad w_9 = \alpha_1, \quad w_7 = \alpha_2, \quad w_{10} = w_7 \\
 w_1 &= \frac{1}{6}[-12\alpha_2 + 3(\alpha_1 - \alpha_0) + 2] \\
 w_2 &= 4\alpha_2 - \alpha_1 - \alpha_0 \\
 w_3 &= \frac{1}{6}[-12\alpha_2 - 3(\alpha_1 - \alpha_0) + 2] \\
 w_4 &= \frac{1}{3}[6\alpha_2 - 1], \quad w_5 = w_1, \quad w_6 = w_3
 \end{aligned} \right\} \quad (4.3.3-10)$$

where α_0 , α_1 and α_2 are arbitrary constants. The choice of α_0 , α_1 and α_2 will now determine the method.

Thus the standard RK method (4.3.3-3) can be easily deduced by setting $w_1 = w_2 = w_3 = w_4 = 0$ in (4.3.3-10), that is when $\alpha_0 = \alpha_1 = \frac{1}{3}$ and $\alpha_2 = \frac{1}{6}$.

The general composite RK-GM method may now be defined by (4.3.3-5) with the k_i , $i = 1, 2, 3, 4$ given by (4.3.3-4) and the w_i , $i = 1, 2, \dots, 10$ given by (4.3.3-10).

By letting $w_7 = w_8 = w_9 = 0$ in (4.3.3-10), that is when $\alpha_1 = 0$, $i = 0, 1, 2$ we obtain a typical RK-GM method which corresponds to the standard RK method (4.3.3-3). This particular RK-GM method is given as

$$Y_{n+1} = Y_n + \frac{h}{3} [(\sqrt{k_1} + \sqrt{k_4})(\sqrt{k_2} + \sqrt{k_3}) - \sqrt{k_1 k_4}], \quad (4.3.3-12)$$

where the k_i , $i = 1, 2, 3, 4$ are defined in (4.3.3-4).

Note that we are free to set the constants c_i and a_{ij} in (4.3.3-6) and hence obtain other forms of the composite RK-GM methods. The aim here is to obtain a RK-GM method which corresponds to the standard RK method for which the application will be shown to be of some importance.

4.3.3.1 ERROR ANALYSIS OF (4.3.3-12)

In the derivation of the fourth-order RK-GM method (4.3.3-5) we have used the Taylor series expansion of Y_{n+1} about x_n up to and including the terms of $O(h^5)$. Thus we may write

$$Y_{n+1} \approx Y_n + hY_n^{(1)} + \frac{h^2}{2} Y_n^{(2)} + \frac{h^3}{6} Y_n^{(3)} + \frac{h^4}{24} Y_n^{(4)} + \frac{h^5}{120} Y_n^{(5)}. \quad (4.3.3-13)$$

By differentiating (4.3.2-6) again, we obtain

$$\begin{aligned} Y^{(5)} &= f_{xxxx} + ff_{xxx} + 3[f_x f_{xxy} + ff_y f_{xxy} + ff_{xxx} + f^2 f_{xxy}] \\ &+ 3[f_{xx} f_{xy} + f(f_{xy})^2 + f_x f_{xxy} + ff_x f_{xy}] \\ &+ 5[f_x f_y f_{xy} + f(f_y)^2 f_{xy} + f(f_{xy})^2 \\ &\quad + f^2 f_{xy} f_{yy} + ff_y f_{xxy} + f^2 f_y f_{xy}] \\ &+ 3[(f_x)^2 f_{yy} + ff_x f_y f_{yy} + ff_{xx} f_{yy} \\ &\quad + f^2 f_{xy} f_{yy} + ff_x f_{xy} + f^2 f_x f_{yyy}] \\ &+ 3[2ff_x f_{xy} + 2f^2 f_y f_{xy} + f^2 f_{xxy} + f^3 f_{xy}] \\ &+ 4[2ff_x f_y f_{yy} + f^2 f_{xy} f_{yy} + f^2 f_y f_{xy} \\ &\quad + 2(ff_y)^2 f_{yy} + f^3 (f_{yy})^2 + f^3 f_y f_{yyy}] \\ &+ [f_{xx} f_{xy} + ff_{xx} f_{yy} + f_{xxx} f_y + ff_y f_{xxy}] \\ &+ [f_{xx} (f_y)^2 + ff_{xy} (f_y)^2 + 2f_x f_y f_{xy} + 2ff_x f_y f_{yy}] \\ &+ [f_x (f_y)^3 + f(f_y)^4 + 3f(f_y)^2 f_{xy} + 3(ff_y)^2 f_{yy}] \\ &+ [3f^2 f_x f_{yyy} + 3f^3 f_y f_{yyy} + f^3 f_{xy} + f^4 f_{yyy}], \\ &= f_{xxxx} + 4ff_{xxx} + 6f_x f_{xxy} + 9ff_y f_{xxy} + 6f^2 f_{xxy} \\ &\quad + 4f_{xx} f_{xy} + 8f(f_{xy})^2 + 12ff_x f_{xy} + 7f_x f_y f_{xy} \\ &\quad + 9f(f_y)^2 f_{xy} + 12f^2 f_{xy} f_{yy} + 15f^2 f_y f_{xy} + 3(f_x)^2 f_{yy} \\ &\quad + 13ff_x f_y f_{yy} + 4ff_{xx} f_{yy} + 6f^2 f_x f_{yyy} + 4f^3 f_{xy} \\ &\quad + 11(f f_y)^2 f_{yy} + 4f^3 (f_{yy})^2 + 7f^3 f_y f_{yyy} + f_y f_{xxx} \\ &\quad + f_{xx} (f_y)^2 + f_x (f_y)^3 + f(f_y)^4 + f^4 f_{yyy}. \end{aligned} \quad (4.3.3-14)$$

Hence by using the REDUCE program, the local truncation error of (4.3.3-12) is obtained as

$$T_{n+1} = \frac{h^5}{23040f^3} H \quad (4.3.3-15)$$

where

$$\begin{aligned}
H = & [8f^7 f_{yyyy} + 16f^6 f_y f_{yyy} - 108f^6 f_{yy}^2 + 32f^6 f_{xyyy} \\
& + 48f^5 f_x f_{yyy} + 228f^5 f_y^2 f_{yy} - 384f^5 f_{xy} f_{yy} + 48f^5 f_{xxyy} \\
& + 384f^4 f_x f_{yy} - 576f^4 f_x f_{xxy} + 96f^4 f_x f_{xyy} - 267f^4 f_y^4 \\
& + 72f^4 f_y^2 f_{xy} + 528f^4 f_y f_{xxy} - 168f^4 f_{xx} f_{yy} \\
& - 336f^4 f_{xy}^2 + 32f^4 f_{xxyy} + 204f^3 f_x^2 f_{yy} \\
& - 312f^3 f_x f_y^3 + 96f^3 f_x f_y f_{xy} + 48f^3 f_x f_{xxy} \\
& - 12f^3 f_y^2 f_{xx} - 32f^3 f_y f_{xxx} - 288f^3 f_{xx} f_{xy} \\
& + 8f^3 f_{xxxx} - 30f^2 f_x^2 f_y^2 + 120f^2 f_x^2 f_{xy} \\
& - 60f^2 f_{xx}^2 + 60f f_x^2 f_{xx} - 15f_x^4].
\end{aligned}$$

4.3.3.2 NUMERICAL RESULTS

Problem : $y^{(1)} + y = 0$.

Initial condition $x_0 = 0, y_0 = 1$.

Exact solution $y = e^{-x}$.

Solution domain $[0,1]$.

x_n	Exact Solution	Numerical solution	Relative Error
.00	.1000000E+01	.1000000E+01	0
.10	.9048374E+00	(A).9048375E+00 (B).9048375E+00	.5998097E-07 .4302973E-07
.20	.8187308E+00	(A).8187310E+00 (B).8187309E+00	.1136851E-06 .8155651E-07
.30	.7408182E+00	(A).7408185E+00 (B).7408184E+00	.1612057E-06 .1156473E-06
.40	.6703200E+00	(A).6703204E+00 (B).6703203E+00	.2026952E-06 .1454114E-06
.50	.6065307E+00	(A).6065310E+00 (B).6065309E+00	.2383607E-06 .1709975E-06
.60	.5488116E+00	(A).5488121E+00 (B).5488119E+00	.2684584E-06 .1925892E-06
.70	.4965853E+00	(A).4965857E+00 (B).4965856E+00	.2932861E-06 .2104004E-06
.80	.4493290E+00	(A).4493294E+00 (B).4493293E+00	.3131760E-06 .2246692E-06
.90	.4065697E+00	(A).4065701E+00 (B).4065700E+00	.3284863E-06 .2356527E-06
1.00	.3678794E+00	(A).3678799E+00 (B).3678798E+00	.3395930E-06 .2436205E-06

Table (4.3.3.2): Results from (A) RK-GM formula (4.3.3-12) and (B) RK formula (4.3.3-3).

From the numerical results we observe that both the RK-GM method (4.3.3-12) and the RK method (4.3.3-3) are comparable; they have the same order of accuracy. However, the RK-GM method involves more work which is required for the evaluation of the terms containing square roots. Nevertheless, we may combine the two methods to form an embedded formula with error control strategy. This is discussed and illustrated in Evans and Jayes[1990] and in section 4.4.

4.3.4 STABILITY ANALYSIS FOR THE RK-GM METHODS

In this section we shall endeavour to investigate the absolute stability property of the RK-GM methods developed in the preceding sections. To accomplish this objective we shall determine the region of stability of the various RK-GM methods.

The first-order method in the class of RK-GM formulae is identical to the Euler method. We shall therefore omit the discussion of its stability property since it has been given earlier in section 3.2.3.

We shall now consider the stability property of the second-order, two-stage RK-GM method. By applying the standard test problem $y^{(1)} = \lambda y$ to (4.3.1-19a) and (4.3.1-19b), we obtain

$$\left. \begin{aligned} k_1 &= \lambda y_n \\ k_2 &= \lambda(1 + \rho z) y_n \end{aligned} \right\} \quad (4.3.4-1)$$

$$Y_{n+1} = Y_n \{ 1 + z [w_1 + w_2(1 + \rho z) + w_3 \sqrt{1 + \rho z}] \} \quad (4.3.4-2)$$

where $z = h\lambda$, $\rho = \frac{1}{\beta}$ and β is an arbitrary constant.

From (4.3.4-2), we obtain the polynomial $Q(z)$ as

$$\begin{aligned}
Q(z) &= \frac{Y_{n+1}}{Y_n} \\
&= 1 + z[w_1 + w_2(1 + \rho z) + w_3\sqrt{1 + \rho z}]. \quad (4.3.4-3)
\end{aligned}$$

Now by substituting the values of w_i for $i = 1, 2$ and 3 given in (4.3.1-18) into (4.3.4-3), we obtain

$$\begin{aligned}
Q(z) &= 1 + [1 + \beta - 2\alpha]z + \left[1 - \frac{\alpha}{\beta}\right]z^2 \\
&\quad + [2\alpha - \beta]z\left[1 + \frac{z}{\beta}\right]^{1/2}. \quad (4.3.4-4)
\end{aligned}$$

Note that we may write (4.3.4-3) in the form

$$Q(z) = 1 + z + O(z^2). \quad (4.3.4-5)$$

Hence for sufficiently small positive z , $Q(z) > 1$ ^{and} we may conclude that the interval of absolute stability has the form $(\delta, 0)$. Moreover, if the two-stage method possesses order two, then (4.3.1-18) holds and (4.3.4-3) yields (4.3.4-4).

We observe that if $w_3 = 0$, $\alpha = \frac{1}{2}$ and $\beta = 1$, then the RK-GM method reduces to the classical RK method of order two. The stability polynomial $Q(z)$ is now independent of the coefficients of the method. Otherwise, we will always have $Q(z)$ determined by the values of the parameters α and β as indicated in (4.3.4-4). Thus the plot of $Q(z)$ against z is always established by the parameters α and β of the method. However, in every case the plot of $Q(z)$ against z reveals that $|Q(z)| < 1$ whenever $z \in (\delta(\alpha, \beta), 0)$. Hence in general, the interval of absolute stability of the second-order, two-stage RK-GM methods is governed by the parameters α and β .

To illustrate this we shall consider the stability property of two second-order, two-stage RK-GM methods derived from the set of parameters $\alpha = -1$, $\beta = \frac{3}{20}$ and $\alpha = 1$, $\beta = 1$.

For $\alpha = 1$ and $\beta = 1$, we have the RK-GM method given by

$$Y_{n+1} = Y_n + h\sqrt{k_1 k_2} \quad (4.3.4-6)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n+h, Y_n+hk_1) \end{aligned} \right\} \quad (4.3.4-6a)$$

and for $\alpha = -1$ and $\beta = \frac{3}{20}$ the corresponding RK-GM method is

$$Y_{n+1} = Y_n + \frac{h}{20} \{ 40k_1 + 23k_2 - 43\sqrt{k_1 k_2} \} \quad (4.3.4-7)$$

where

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + \frac{20}{3}h, Y_n + \frac{20}{3}hk_1) \end{aligned} \right\} \quad (4.3.4-7a)$$

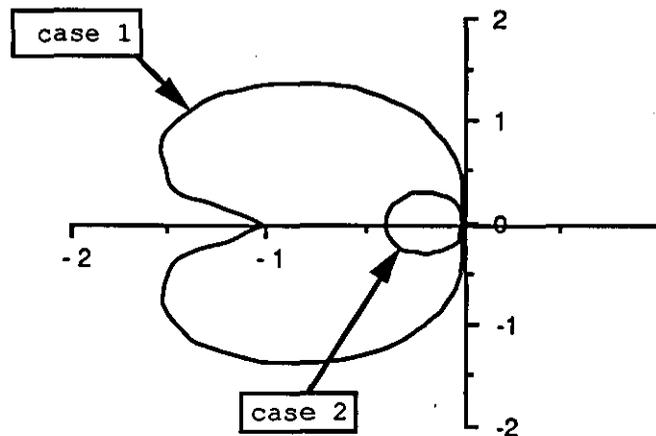
Now by applying the test function $y^{(1)} = \lambda y$ to (4.3.4-6) and (4.3.4-7), we obtain the respective stability polynomials $Q_1(z)$ and $Q_2(z)$ as

$$\text{case 1 : } Q_1(z) = 1 + z\sqrt{1+z} \quad (4.3.4-8a)$$

and

$$\text{case 2 : } Q_2(z) = 1 + \frac{63}{20}z + \frac{460}{3}z^2 - 43z\sqrt{1 + \frac{20}{3}z} \quad (4.3.4-8b)$$

The plots of $Q_1(z)$ and $Q_2(z)$ are given in Figure(4.3.4a).



Figure(4.3.4a): Stability regions of the second-order, two-stage RK-GM methods for different set of parameters.

Next we shall discuss the stability property of the third-order, three-stage RK-GM methods. By applying the standard test problem $y^{(1)} = \lambda y$ to (4.3.2-11) we have

$$\begin{aligned} k_1 &= \lambda y_n, \\ k_2 &= \lambda \left[y_n + \frac{1}{2} z y_n \right], \\ &= \lambda y_n \left[1 + \frac{z}{2} \right], \\ k_3 &= \lambda y_n \left[1 + z \left(2 \left(1 + \frac{z}{2} \right) - 1 \right) \right], \\ &= \lambda y_n \left[1 + z(1 + z) \right]. \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} y_{n+1} = y_n \left\{ 1 + z \left[w_1 \left(1 + \frac{z}{2} \right)^{1/2} \right. \right. \\ \quad + w_2 \left(\left(1 + \frac{z}{2} \right) \left(1 + z(1 + z) \right) \right)^{1/2} \\ \quad + w_3 \left(1 + z(1 + z) \right)^{1/2} + w_4 + w_5 \left(1 + \frac{z}{2} \right) \\ \quad \left. \left. + w_6 (1 + z(1 + z)) \right] \right\}. \end{aligned}$$

Hence the polynomial function $Q(z)$ is obtained as

$$\begin{aligned} Q(z) &= \frac{y_{n+1}}{y_n} \\ &= 1 + z \left\{ w_1 \left(1 + \frac{z}{2} \right)^{1/2} \right. \\ &\quad + w_2 \left(\left(1 + \frac{z}{2} \right) \left(1 + z(1 + z) \right) \right)^{1/2} \\ &\quad + w_3 \left(1 + z(1 + z) \right)^{1/2} + w_4 + w_5 + w_6 \\ &\quad \left. + \left(\frac{w_5}{2} + w_6 \right) z + w_6 z^2 \right\}. \end{aligned} \tag{4.3.4-8}$$

From (4.3.4-8) it is easy to see that if the RK-GM method is consistent, then $\sum_{i=1}^6 w_i = 1$. Moreover, if the three stage RK-GM method is of order three, then equations (4.3.2-7a) to (4.3.2-7i) hold and the polynomial $Q(z)$ is given by (4.3.4-8). Hence for sufficiently small positive z , $Q(z) > 1$, we may conclude that the interval of absolute stability of this method has the form $(\delta, 0)$. However, we note that from (4.3.4-8), the polynomial $Q(z)$ is governed by the coefficients

of the method as it is clearly indicated in the form of the expression of $Q(z)$. Therefore, the interval of absolute stability of each member within the same class of three-stage third-order RK-GM methods may not be the same.

By substituting $w_i = 0, i = 1, 2, 3$ and $w_4 = \frac{1}{6}, w_5 = \frac{2}{3},$ and $w_6 = \frac{1}{6},$ we have the classical RK method of order three. Its stability region defined by the polynomial $Q(z)$ is given by

$$Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} . \quad (4.3.4-9)$$

Now by letting $w_5 = w_6 = 0,$ we obtain the RK-GM method (4.3.2-12). The corresponding polynomial $Q(z)$ which determines its stability region can be deduced from (4.3.4-8) for $w_1 = w_2 = \frac{2}{3}, w_3 = -\frac{1}{3}$ and $w_4 = 0.$ Hence we have the stability region of the third-order RK-GM method (4.3.2-12) is given by

$$Q(z) = 1 + \frac{1}{3} z \left\{ 2 \left[\left(1 + \frac{z}{2} \right)^{1/2} + \left(\left(1 + \frac{z}{2} \right) (1 + z(1 + z)) \right)^{1/2} \right] - \left(1 + z(1 + z) \right)^{1/2} \right\} . \quad (4.3.4-10)$$

Finally we consider the four-stage fourth-order RK-GM methods. Again, because of the complicated and lengthy derivation, we shall only concentrate on a particular case of the class of four-stage fourth-order RK-GM methods derived in section 4.3.3. Specifically, we shall discuss the stability region of (4.3.3-12).

By applying the standard test problem $y^{(1)} = \lambda y$ in (4.3.3-11) and (4.3.3-12), we obtain

$$\left. \begin{aligned}
 k_1 &= \lambda y_n \\
 k_2 &= \lambda y_n \left[1 + \frac{z}{2} \right] \\
 k_3 &= \lambda y_n \left[1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right] \\
 k_4 &= \lambda y_n \left[1 + z \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right) \right] \\
 Y_{n+1} &= Y_n \left\{ 1 + \frac{z}{3} \left[1 + \left[\left(1 + z \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right) \right)^{1/2} \right] \right. \right. \\
 &\quad \times \left[\left(1 + \frac{z}{2} \right)^{1/2} + \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right)^{1/2} \right] \\
 &\quad \left. \left. - \left(1 + z \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right) \right)^{1/2} \right] \right\}
 \end{aligned} \right\} (4.3.4-11)$$

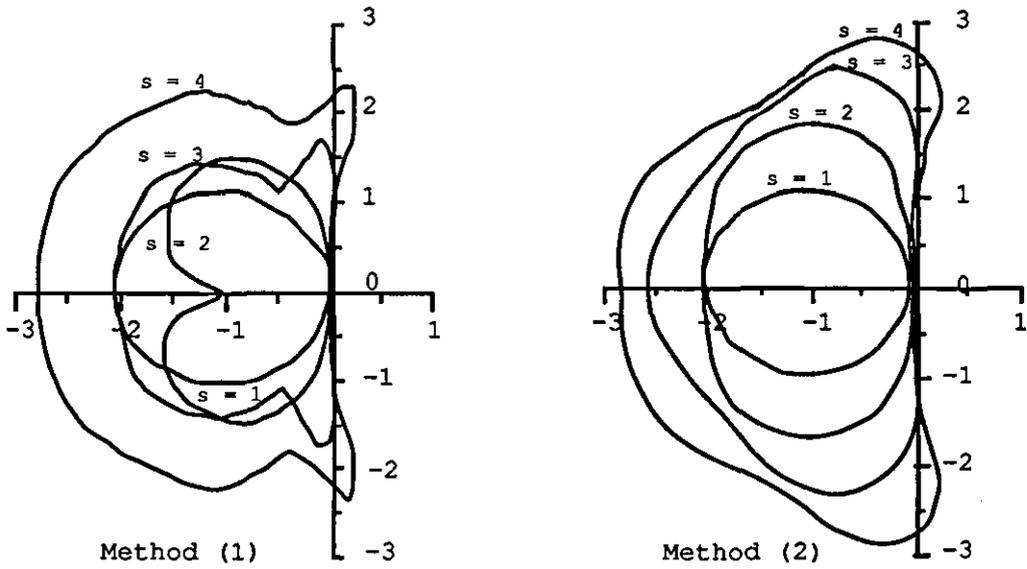
So we obtain the polynomial $Q(z)$ which describes the stability region of the fourth-order RK-GM method (4.3.3-12) as given by

$$\begin{aligned}
 Q(z) = 1 + \frac{z}{3} \{ & 1 + \left[\left(1 + z \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right) \right)^{1/2} \right] \times \\
 & \left[\left(1 + \frac{z}{2} \right)^{1/2} + \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right)^{1/2} \right] \\
 & - \left(1 + z \left(1 + \frac{z}{2} \left(1 + \frac{z}{2} \right) \right) \right)^{1/2} \}. \quad (4.3.4-12)
 \end{aligned}$$

The stability region of the classical fourth-order RK method can be easily obtained by applying the test problem $y^{(1)} = \lambda y$ in (4.3.3-3) and (4.3.3-4). Thus we have the polynomial

$$Q(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}. \quad (4.3.4-13)$$

Figure(4.3.4b) shows the stability regions of both the classical RK method and the RK-GM method for orders $s = 1$ to 4. Except for the order one method (which is the Euler method) the RK-GM methods appear to have smaller regions of stability.



Figure(4.3.4b): Stability regions of the RK-GM methods (Method (1)) and the RK methods (Method (2)) of orders $s = 1, 2, 3$ and 4.

4.3.5 OPTIMAL EXPLICIT TWO-STAGE RK PROCESS

Consider the explicit two-stage RK-GM process defined by

$$Y_{n+1} = Y_n + h\Phi_{GM2}(x_n, Y_n; h), \quad (4.3.5-1)$$

where

$$\Phi_{GM2}(x_n, Y_n; h) = w_1 k_1 + w_2 k_2 + w_3 \sqrt{k_1 k_2} \quad (4.3.5-2)$$

with constraint

$$w_1 + w_2 + w_3 = 1. \quad (4.3.5-3)$$

Furthermore, we have

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f(x_n + c_1 h, Y_n + a_{21} h k_1) \end{aligned} \right\}. \quad (4.3.5-4)$$

with constraint

$$c_1 = a_{21}. \quad (4.3.5-5)$$

Therefore using (4.3.1-4) and (4.3.5-5) the Taylor series expansion of k_2 about (x_n, y_n) in the solution space is given as

$$k_2 = f \left\{ 1 + hc_1 \left[\frac{f_x}{f} + f_y + \frac{1}{2} hc_1 \left(\frac{f_{xx}}{f} + 2f_{xy} + ff_{yy} \right) \right] \right\} + O(h^3) \quad (4.3.5-6)$$

with all terms evaluated at (x_n, y_n) .

Hence we obtain the expansion of $\sqrt{k_1 k_2}$ as

$$\sqrt{k_1 k_2} = f \left\{ 1 + \frac{1}{2} hc_1 \left[\frac{f_x}{f} + f_y + \frac{1}{2} hc_1 \left(\frac{f_{xx}}{f} + 2f_{xy} + ff_{yy} \right) \right] - \frac{1}{8} (hc_1)^2 \left[\left(\frac{f_x}{f} \right)^2 + 2 \frac{f_x f_y}{f} + f_y^2 \right] \right\} + O(h^3). \quad (4.3.5-7)$$

This yields the expression for Φ_{GM2} as

$$\begin{aligned} \Phi_{GM2} = & (w_1 + w_2 + w_3) f + hc_1 (w_2 + \frac{1}{2} w_3) (f_x + ff_y) \\ & + (hc_1)^2 \left(\frac{w_2}{2} + \frac{w_3}{4} \right) f \left(\frac{f_{xx}}{f} + 2f_{xy} + ff_{yy} \right) \\ & - \frac{1}{8} (hc_1)^2 w_3 f \left[\left(\frac{f_x}{f} \right)^2 + 2 \frac{f_x f_y}{f} + f_y^2 \right] + O(h^3). \end{aligned} \quad (4.3.5-8)$$

The equivalent increment function Φ_{T2} for the Taylor series expansion method is obtained as

$$\begin{aligned} \Phi_{T2} = & f + \frac{h}{2} [f_x + ff_y] + \frac{h^2}{6} [f_{xx} + 2ff_{xy} + f_x f_y + f^2 f_{yy} + ff_y^2] \\ & + O(h^3). \end{aligned} \quad (4.3.5-9)$$

By equating like terms of (4.3.5-8) and (4.3.5-9), we obtain

$$\begin{aligned} \text{coefficient of } hf: & \quad w_1 + w_2 + w_3 = 1 \\ \text{coefficient of } h(f_x + ff_y): & \quad (w_2 + \frac{1}{2} w_3) c_1 = \frac{1}{2} \\ \text{coefficient of } h^2 f \left(\frac{f_{xx}}{f} + 2f_{xy} + ff_{yy} \right): & \quad c_1^2 \left(\frac{w_2}{2} + \frac{w_3}{4} \right) = \frac{1}{6} \end{aligned}$$

Thus we have a set of three equations in four unknowns. Hence by setting $c_1 = a_{21} = \alpha \neq 0$ as the free parameter, we obtain

$$\begin{aligned} w_1 + w_2 + w_3 &= 1 \\ 2w_2 + w_3 &= \frac{1}{\alpha} \\ 6w_2 + 3w_3 &= \frac{2}{\alpha^2} \end{aligned}$$

Therefore $\alpha = \frac{2}{3}$ and by setting $w_1 = \beta$, we obtain $w_3 = \frac{1}{2} - 2\beta$. Now the error function Ψ_2 for the explicit two-stage RK-GM process is obtained as

$$\begin{aligned} \Psi_2 &= \Phi_{GM2} - \Phi_{T2} \\ &= -\frac{c_1^2 w_3}{8} f \left[\left(\frac{f_x}{f} \right)^2 + 2 \frac{f_x f_y}{f} + f_y^2 \right] - \frac{1}{6} [f_x f_y + f f_y^2] \\ &= -\frac{1-4\beta}{36} f \left[\left(\frac{f_x}{f} \right)^2 + 2 \frac{f_x f_y}{f} + f_y^2 \right] - \frac{1}{6} [f_x f_y + f f_y^2]. \end{aligned}$$

By assuming that the Lotkin[1951] inequality (4.3.1.1-6) holds, this leads to the following bound on the error function Ψ_2 ,

$$|\Psi_2(x_n, Y_n; h)| < \left[\frac{|1-4\beta|}{9} + \frac{1}{3} \right] PQ^2. \quad (4.3.5-10)$$

Now (4.3.5-10) attains its minimum value of $\frac{1}{3} PQ^2$ when the free parameter $\beta = \frac{1}{4}$. Hence the coefficients of an optimal two-stage RK-GM process are obtained as

$$c_1 = a_{21} = \frac{2}{3}, \quad w_1 = \frac{1}{4}, \quad w_2 = \frac{3}{4}, \quad w_3 = 0.$$

The resulting method is

$$Y_{n+1} = Y_n + \frac{h}{4} [k_1 + 3k_2]$$

which is called the Heun's two-stage scheme and

$$\begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f\left(x_n + \frac{2}{3}h, Y_n + \frac{2}{3}hk_1\right). \end{aligned}$$

4.4 ERROR CONTROL AND ADAPTIVE METHODS

In Section 4.3 we have developed the composite RK-GM methods of orders $s = 2, 3$ and 4. It is well known that any single RK method has a fixed order. Moreover, a method involving only one equation has no way of monitoring the discretization error on its own. Hence it cannot select the appropriate step-size to maintain accuracy throughout the integration range. Thus it is necessary to combine two different methods to achieve automatic step-size control. Several adaptive RK methods have been derived recently. However, they are developed by the combination of two distinct RK methods of different orders. We propose here a method which is the combination of two different RK methods but of the same order. Nevertheless, this combined method is still of a fixed order and hence its effectiveness may be restricted.

A composite Runge-Kutta arith-geometric mean method (AGM) was developed in Evans and Jayes [1990]. The combination of the classical RK method with the RK-GM method to form an adaptive error control strategy suggests an alternative Runge-Kutta AGM method to existing techniques. This is tested with the various library routines, namely the NAG (subroutine D02YAF), the IMSL (Subroutine DVERK) and the subroutine RKF45. Some interesting numerical results are obtained and commented upon.

4.4.1 ERROR ESTIMATION FOR RK PROCESSES

The work of Lagrange in 1797 and more importantly of Cauchy has set the trend that every numerical method should be accompanied by a reliable error estimate.

Runge in 1905 also found the importance of error estimates for the RK methods. Thus a theorem on the rigorous error bounds can be expressed as:

Theorem 4.4.1-1 (Hairer et al.[1987]) : Given an RK method (4.1-3a, -3b, -3c) of order p . Suppose all partial derivatives of $f(x,y)$ up to order p exist and are continuous, then the local error of (4.1-3a) satisfies the following,

$$\|y(x_n+h)-y_{n+1}\| \leq h^{p+1} \left\{ \frac{1}{(p+1)!} \max_{t \in [0,1]} \|y^{(p+1)}(x_n+th)\| + \frac{1}{p!} \sum_{i=1}^s |w_i| \max_{t \in [0,1]} \|k_i^p(t)\| \right\} \quad (4.4.1-1)$$

and hence also,

$$\|y(x_n+h)-y_{n+1}\| \leq Ch^{p+1} \quad , \quad (4.4.1-2)$$

where C is a constant.

However for higher order methods, (4.4.1-1) seems impractical. Alternatively, it is more realistic to consider the principal error term of the method. For autonomous equations, this error term is obtained by subtracting the Taylor series expansion of the numerical solution from the Taylor series expansion of the exact solution.

Another alternative way of obtaining an error estimate is to consider the global error of the method. A global error is the error of the computed solution after some integration steps. Suppose that we have a single-step method (4.1-3a) and the initial-value problem (4.1-1).

The numerical solution of (4.1-3a) at a point $X > a$ is then obtained by a single-stepwise procedure,

$$\left. \begin{aligned} Y_{n+1} &= Y_n + h_n \Phi(x_n, Y_n; h_n), \\ h_n &= x_{n+1} - x_n, \\ x_N &= X. \end{aligned} \right\} \quad (4.4.1-3)$$

The global error is therefore given by,

$$E = y(X) - Y_N \quad , \quad (4.4.1-4)$$

which is found simply by transporting the local errors to the final point x_N and then summing up. This can be done in any of the two following ways:

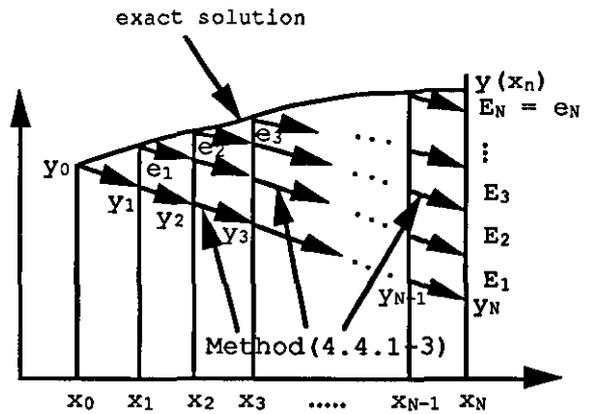
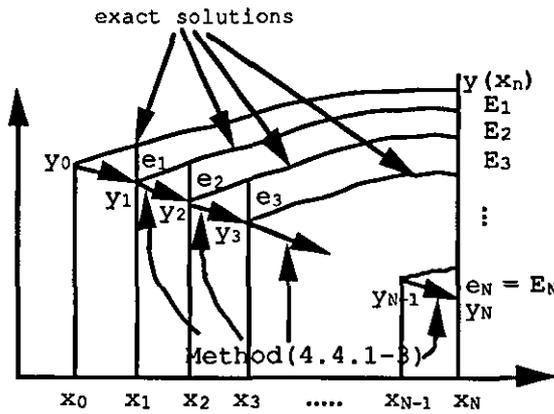
Either (a) along the exact solution curves (Figure (4.4.1a)); which can give distinct results when well defined estimates of error propagation for the exact solutions are known,

or (b) along $N-n$ steps of the numerical method (Figure (4.4.1b)); which was used by Cauchy in 1824 and Runge in 1905.

In either case we first estimate the local errors e_i using Theorem 4.4.1-1 giving,

$$\|e_i\| \leq C h_{n-1}^{p+1} . \quad (4.4.1-5)$$

Then, the transported errors E_i are estimated accordingly.



Figure(4.4.1a):Global error estimation [method (a)]

Figure(4.4.1b):Global error estimation [method (b)]

(These figures are adapted from Hairer et al.[1987])

a) For method(a), the following theorem follows:

Theorem 4.4.1-2 (Hairer, et al.[1987]) : Let U be a neighbourhood of $\{(x, y(x)); a \leq x \leq X\}$ where $y(x)$ is the exact solution of the problem (4.1-1). Suppose that in U ,

$$\left\| \frac{df}{dy} \right\| \leq L \quad \text{or} \quad \mu \left(\frac{df}{dy} \right) \leq L, \quad (4.4.1-6)$$

and that the local error estimates (4.4.1-5) are true in U . Then the global error (4.4.1-4) satisfies,

$$\|E\| \leq h^p \frac{C'}{L} \{ \exp(L(x_N - a)) - 1 \}, \quad (4.4.1-7)$$

where $h = \max_1 h_1$,

$$C' = \begin{cases} C, & L \geq 0 \\ C \exp(-Lh), & L < 0 \end{cases}$$

and h is sufficiently small for the numerical solution to be within U . If $L \rightarrow 0$, then $\|E\| \rightarrow h^p C(x_N - a)$.

b) For the second method, we need to estimate $\|z_{n+1} - y_{n+1}\|$ in terms of $\|z_n - y_n\|$, where besides (4.4.1-3),

$$z_{n+1} = z_n + h_n \Phi(x_n, y_n; h_n), \quad (4.4.1-8)$$

is another numerical solution. Hence the theorem below follows:

Theorem 4.4.1-3 (Hairer et al.[1987]) : Suppose that the local error satisfies, for initial values on the exact solution,

$$\|y(x+h) - y(x) - h\Phi(x, y(x); h)\| \leq Ch^{p+1} \quad (4.4.1-9)$$

and suppose that in a neighbourhood of the solution the increment function Φ satisfies,

$$\|\Phi(x, z; h) - \Phi(x, y; h)\| \leq L \|z - y\|. \quad (4.4.1-10)$$

Then the global error (4.4.1-4) can be estimated by,

$$\|E\| \leq h^p \frac{C}{L} \{ \exp(L(x_N - a)) - 1 \} \quad (4.4.1-11)$$

where $h = \max_i h_i$.

Theorem 4.4.1-2 and Theorem 4.4.1-3 above discuss the convergence as a function of h as it tends to zero, where h is now interpreted as the maximum step size and assume that there is a function $\theta(t)$ such that $0 < D \leq \theta(t) \leq 1$ for $t \in [a, x_N]$ and that,

$$\left. \begin{aligned} h_n &= h\theta(x_n) \\ x_{n+1} &= x_n + h_n. \end{aligned} \right\} \quad (4.4.1-12)$$

We see that if $h > 0$, a finite number of steps will cover the interval $[a, x_N]$ since $h_n \geq \Delta h > 0$.

4.4.2 ADAPTIVE ERROR CONTROL STRATEGY

The basic idea of an adaptive error control strategy is to compare two approximations and thereby obtain an estimate of their accuracy. If the accuracy is acceptable, one of them is taken as the numerical approximation at the mesh point. Otherwise, the step-size is modified and the process repeated on the mesh points using the new step-size.

To reduce the total number of function evaluations, the two methods should use the same points of evaluation. Since the standard Runge-Kutta (AM) and the RK-GM methods use the same k_i , for $i = 1, 2, 3, 4$, consequently the adaptive error control strategy which employs the AM and RK-GM pair requires only four function evaluations per step.

We shall now pursue the analysis of the basic technique. Consider the solution of (4.1-1) and the approximation (4.1-2). The 'best' method would be that, whenever a

tolerance $TOL > 0$ was given, the number of mesh points used to ensure that the global error, i.e. $|y(x_i) - y_i|$, did not exceed TOL for any $i = 1, 2, \dots, n-1$ is minimal. In other words, the amount of computational work is minimized for the given error tolerance if h is chosen by $h = h_n \theta_n(t_n)$ in the n^{th} step of the iteration process.

Generally, the global error of a difference method cannot be determined explicitly. Alternatively, we can work with the local truncation error of the method. That is we try to find an estimate of the global error in terms of the local truncation error.

To recapitulate Theorems 4.4.1-2 and 4.4.1-3, we see that from Theorem 4.4.1-3, the global error is one order less than that of the local truncation error. Therefore it is justified to consider the control of the local error if we are interested in the control of the global error.

Thus, an alternative approach to step-size selection is to exploit the determination of the local truncation error of a method as best we can while controlling its global error.

Suppose two single-step methods (4.4.1-3) and (4.4.1-8) are used to approximate the solution of the initial-value problem (4.1-1). Denote the local truncation of (4.4.1-3) by τ_{n+1} and that of (4.4.1-8) by δ_{n+1} . Let them be of orders r and s accuracy respectively.

Assume $0 < r \leq s$. Suppose that $y_n \approx y(x_n) \approx z_n$, then,

$$\begin{aligned}
 y(x_{n+1}) - Y_{n+1} &= y(x_{n+1}) - y_n - h_n \Phi(x_n, y_n; h_n) \\
 &\approx y(x_{n+1}) - y(x_n) - h_n \Phi(x_n, y(x_n); h_n) \\
 &= h_n \tau_{n+1} .
 \end{aligned}
 \tag{4.4.2-1}$$

Since a method with local truncation error of order r is given by

$$\tau_{n+1} = \frac{y(x_{n+1}) - Y(x_n)}{h_n} - \Phi(x_n, Y(x_n); h_n).$$

So, the local truncation error of (4.4.1-3) can be written as

$$\begin{aligned} \tau_{n+1} &\approx \frac{1}{h_n} [y(x_{n+1}) - Y_{n+1}] \\ &= \frac{1}{h_n} [y(x_{n+1}) - z_{n+1}] + \frac{1}{h_n} [z_{n+1} - Y_{n+1}] \\ &\approx \delta_{n+1} + \frac{1}{h_n} [z_{n+1} - Y_{n+1}]. \end{aligned} \quad (4.4.2-2)$$

Consequently,

$$\Delta = \frac{1}{h_n} [z_{n+1} - Y_{n+1}], \quad (4.4.2-3)$$

can be used to approximate the optimal step-size to control the global error. Now τ_{n+1} is of order r , so a constant p exists such that,

$$ph_n^r \approx \frac{1}{h_n} [z_{n+1} - Y_{n+1}]. \quad (4.4.2-4)$$

An approximate step-size can now be chosen by considering the truncation error with h_n replaced by qh_n where q is a positive number bounded above and away from zero.

Hence, we may write

$$\begin{aligned} \Delta(qh_n) &\approx p(qh_n)^r \\ &\approx q^r (ph_n^r) \\ &\approx \frac{q^r}{h_n} [z_{n+1} - Y_{n+1}]. \end{aligned} \quad (4.4.2-5)$$

To bound $\Delta(qh_n)$ by ϵ , choose q such that,

$$\frac{q^r}{h_n} |z_{n+1} - Y_{n+1}| \approx \Delta(qh_n) \leq \epsilon, \quad (4.4.2-6)$$

that is, such that

$$q \leq \left[\frac{\epsilon h_n}{|z_{n+1} - Y_{n+1}|} \right]^{1/r} \quad (4.4.2-7)$$

In practice, the usual choice of q is such that,

$$q = \left[\frac{\epsilon h_n}{2|z_{n+1} - Y_{n+1}|} \right]^{1/r} \quad (4.4.2-8)$$

The technique adopted that utilizes (4.4.2-7) for error control consists of using the RK-GM method,

$$z_{n+1} = Y_n + \frac{h}{3} [\sqrt{k_1} (\sqrt{k_2} + \sqrt{k_3} - \sqrt{k_4}) + \sqrt{k_4} (\sqrt{k_2} + \sqrt{k_3})], \quad (4.4.2-9)$$

with local truncation error of order 4 to estimate the local error in the AM method,

$$Y_{n+1} = Y_n + \frac{h}{6} [k_1 + 2(k_2 + k_3) + k_4], \quad (4.4.2-10)$$

of order 4. We call this combined method the AM-GM scheme.

Now by taking the absolute difference between (4.4.2-9) and (4.4.2-10) we have,

$$\begin{aligned} & |Y_{n+1} - z_{n+1}| \\ &= \frac{h}{6} |2 [\sqrt{k_1} (\sqrt{k_2} + \sqrt{k_3} - \sqrt{k_4}) + \sqrt{k_4} (\sqrt{k_2} + \sqrt{k_3})] \\ &\quad - [k_1 + 2(k_2 + k_3) + k_4]| \end{aligned} \quad (4.4.2-11)$$

which can be used to control the error.

Numerical results are obtained and compared with those from the RK-Fehlberg method with error control. The AM-GM method above requires only four function evaluations per step as compared with the RK-Fehlberg which needs six function evaluations per step. Arbitrary RK methods of order four and five used together would require ten function evaluations per step.

4.4.3 ERROR CONTROL AND STEP SIZE SELECTION IN THE AM-GM METHOD

Ralston[1962] provided an error bound (following the Lotkin[1951] technique) for the classical fourth-order Runge-Kutta scheme as

$$\|\psi(x_n, y_n; h)\| \leq \frac{73}{720} ML^4 . \quad (4.4.3-1)$$

Similarly, we can obtain an error bound for the four-stage explicit AM-GM scheme of order four by considering the local truncation errors of the AM and RK-GM methods. We have for the AM method,

$$y_{n+1}^{AM} = y_n + LTE_{AM} , \quad (4.4.3-2)$$

and for the RK-GM method,

$$y_{n+1}^{GM} = y_n + LTE_{GM} , \quad (4.4.3-3)$$

where y_{n+1}^{AM} and y_{n+1}^{GM} are the numerical approximations at x_{n+1} obtained by the AM and RK-GM methods respectively and LTE_{AM} and LTE_{GM} are the corresponding local truncation errors of the AM and RK-GM methods.

It follows that the difference between the AM and RK-GM numerical approximations at x_{n+1} is given by

$$y_{n+1}^{AM} - y_{n+1}^{GM} = LTE_{AM} - LTE_{GM} . \quad (4.4.3-4)$$

The local truncation error of the AM method is given by,

$$\begin{aligned} LTE_{AM} = \frac{h^5}{2880} \{ & f^4 f_{yyyy} + 2f^3 f_y f_{yyy} - 6f^3 f_{yy}^2 + 4f^3 f_{xyyy} + 6f^2 f_x f_{yyy} \\ & + 36f^2 f_y^2 f_{yy} - 18f^2 f_{xy} f_{yy} + 6f^2 f_{xxyy} + 12ff_x f_{xyy} - 4f_y f_{xxx} \\ & + 48ff_x f_y f_{yy} - 72ff_x f_{xxy} - 24ff_y^4 + 24ff_y^2 f_{xy} + 66ff_y f_{xxy} \\ & - 6ff_{xx} f_{yy} - 12ff_{xy}^2 + 4ff_{xxyy} + f_{xxxx} + 18f_x^2 f_{yy} - 24f_x f_y^3 \\ & + 12f_x f_y f_{xy} + 6f_x f_{xxy} + 6f_y^2 f_{xx} - 6f_{xx} f_{xy} \} . \quad (4.4.3-5) \end{aligned}$$

The local truncation error of the RK-GM method is given by,

$$\begin{aligned}
 \text{LTE}_{\text{GM}} = & \frac{h^5}{23040f^3} \{ 8f^7 f_{\text{yyyy}} + 16f^6 f_y f_{\text{yyy}} - 108f^6 f_{\text{yy}}^2 + 32f^6 f_{\text{xyyy}} \\
 & + 48f^5 f_x f_{\text{yyy}} + 228f^5 f_y^2 f_{\text{yy}} - 384f^5 f_{\text{xy}} f_{\text{yy}} + 48f^5 f_{\text{xyyy}} \\
 & + 384f^4 f_x f_y f_{\text{yy}} - 576f^4 f_x f_{\text{xy}} + 96f^4 f_x f_{\text{xyy}} - 267f^4 f_y^4 \\
 & + 72f^4 f_y^2 f_{\text{xy}} + 528f^4 f_y f_{\text{xyy}} - 168f^4 f_{\text{xx}} f_{\text{yy}} - 336f^4 f_{\text{xy}}^2 \\
 & + 32f^4 f_{\text{xyyy}} + 204f^3 f_x^2 f_{\text{yy}} - 312f^3 f_x f_y^3 + 96f^3 f_x f_y f_{\text{xy}} \\
 & + 48f^3 f_x f_{\text{xyy}} - 12f^3 f_y^2 f_{\text{xx}} - 32f^3 f_y f_{\text{xxx}} - 288f^3 f_{\text{xx}} f_{\text{xy}} \\
 & + 8f^3 f_{\text{xxxx}} - 30f^2 f_x^2 f_y^2 + 120f^2 f_x^2 f_{\text{xy}} - 60f^2 f_{\text{xx}}^2 \\
 & + 60f f_x^2 f_{\text{xx}} - 15f_x^4 \}. \tag{4.4.3-6}
 \end{aligned}$$

Therefore the absolute difference between LTE_{AM} and LTE_{GM} is given by

$$\begin{aligned}
 & |\text{LTE}_{\text{AM}} - \text{LTE}_{\text{GM}}| \\
 & = \frac{h^5}{1536f^3} \{ 4f^6 f_{\text{yy}}^2 + 4f^5 f_y^2 f_{\text{yy}} + 16f^5 f_{\text{xy}} f_{\text{yy}} + 5f^4 f_y^4 \\
 & \quad + 8f^4 f_y^2 f_{\text{xy}} + 8f^4 f_{\text{xx}} f_{\text{yy}} + 16f^4 f_{\text{xy}}^2 - 4f^3 f_x^2 f_{\text{yy}} \\
 & \quad + 8f^3 f_x f_y^3 + 4f^3 f_y^2 f_{\text{xx}} + 16f^3 f_{\text{xx}} f_{\text{xy}} + 2f^2 f_x^2 f_y^2 \\
 & \quad - 8f^2 f_x^2 f_{\text{xy}} + 4f^2 f_{\text{xx}}^2 - 4f f_x^2 f_{\text{xx}} + f_x^4 \} \tag{4.4.3-7}
 \end{aligned}$$

From Lotkin[1951] we have the inequalities (4.3.1.1-6) and by direct substitution of (4.3.1.1-6) into (4.4.3-7) we obtain

$$|\text{LTE}_{\text{AM}} - \text{LTE}_{\text{GM}}| < \frac{78}{1536} h^5 \overset{PQ}{\text{PQ}^4} = \frac{13}{256} \overset{PQ}{\text{PQ}^4} h^5. \tag{4.4.3-8}$$

Hence,

$$|Y_{n+1}^{\text{AM}} - Y_{n+1}^{\text{GM}}| \leq \frac{13}{256} \overset{PQ}{\text{PQ}^4} h^5. \tag{4.4.3-9}$$

Now suppose TOL is the user-set tolerance, then by setting,

$$|Y_{n+1}^{AM} - Y_{n+1}^{GM}| \leq \text{TOL} , \quad (4.4.3-10)$$

the error control and step-size selection can be determined by (4.4.3-9).

Example

Consider the initial-value problem,

$$y^{(1)} = y , y(0) = 1 , 0 \leq x \leq 1 , ?$$

$0 < Q \leq 1$

and we set TOL = 10^{-4} , $P = \exp(1) \approx 2.72$ and $Q = 1.00$.

Then,

$$|\frac{13}{256} PQ^4 h^5| < 10^{-4} ,$$

provided $h \leq 0.2355$.

However the bound (4.4.3-9) by itself is of theoretical interest only. Perhaps by combining it with some other information it can be used for choosing the stepsize in practical problems.

4.4.4 PRACTICAL ERROR CONTROL

The error estimates given in section 4.4.1 are of little practical importance because they require the computation of several higher order derivatives. There are various alternative methods for error control, namely:

- (i) Richardson extrapolation,
- (ii) Automatic step-size control,
- (iii) Embedded RK formulae.

Method (i) is basically summarized by the following theorem.

Theorem 4.4.4-1 (Hairer et al.[1987]) : Suppose that y_2 is the numerical result of two steps with step-size h of a RK method of order p , and w is the result of one larger step with step-size $2h$. Then the error of y_2 can be extrapolated (following Richardson) as,

$$y(x_0 + 2h) - y_2 = \frac{y_2 - w}{2^p - 1} + O(h^{p+2}) , \quad (4.4.4-1)$$

and

$$\hat{y}_2 = y_2 + \frac{y_2 - w}{2^p - 1} , \quad (4.4.4-2)$$

is an approximation of order $p+1$ to $y(x_0 + 2h)$.

In method (ii), the error used is obtained from (4.4.4-2) as

$$\text{err} = \frac{1}{2^p - 1} \max_i \frac{|y_{2,i} - w_i|}{d_i} , \quad (4.4.4-3)$$

where d_i is the scaling factor.

Then the error, err , is compared with TOL to obtain the optimal step size from the error behaviour Ch^{p+1} as follows,

$$h \left(\frac{\text{TOL}}{\text{err}} \right)^{1/p+1} . \quad (4.4.4-4)$$

The new step-size is computed from

$$h_{\text{new}} = h \times \min\{\text{fac max}, \max[\text{fac min}, \text{fac} \left(\frac{\text{TOL}}{\text{err}} \right)^{1/p+1}]\}, \quad (4.4.4-5)$$

where fac is the safety factor.

If $\text{err} \leq \text{TOL}$, then the two computed steps are accepted and the solution y_2 or \hat{y}_2 is taken. A new step is tried with h_{new} as step-size. Otherwise, both steps are rejected and the computations repeated using h_{new} . The usual choice of fac is $0.8, 0.9, (0.25)^{1/p+1}$ or $(0.38)^{1/p+1}$.
(Hairer et al. 1987)
 The maximal step-size increase 'fac max' is usually taken between 1.5 and 5. It prevents the program from having very

large step increases and contributes to its safety. If it is too small it increases the computational work unnecessarily. In cases after a step rejection the parameter fac max is set the value one is recommended (Shampine and Watt[1979]).

In the third method, the aim is to develop a RK formula which contains two approximations y_1 and y_2 . The latter can be of the same or higher order than the former. This can then serve for error and step-size control at every step and thus make step rejections economical.

The embedded RK method was first proposed by Merson[1957], Ceschino[1962] and Zonneveld[1963]. However, Merson's method is of order 5 for only linear equations with constant coefficients. Thus, the method over-estimates the error for small h . Similarly, Zonneveld's second formula does not estimate the truncation error. Ceschino's method is uneconomic because the error estimate is too precise (Hairer et al.[1987]).

Sarafyan[1966], England[1969] and Fehlberg[1968, 1969] derived some other formulae of different orders. Fehlberg attempted to minimize the error coefficient of the lower order result y_1 in order to make his method optimal. Consequently, the difference between the two approximations might under-estimate the local error. Dormand and Prince[1980] developed a method for which the error terms of the higher order result are minimized and the lower order result is computed just for the step-size mechanism. This is claimed to give excellent results (Hairer et al.[1987]).

4.4.5 MATRIX REPRESENTATION OF THE RK PROCESSES

An s-stage RK process can be described by the matrix notation (see Fatunla[1988]),

$$\begin{bmatrix} A & C \\ B & 0 \end{bmatrix} \quad (4.4.5-1)$$

where,

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1s} \\ a_{21} & a_{22} & \dots & a_{2s} \\ & & \cdot & \\ & & \cdot & \\ & & \cdot & \\ a_{s-1,1} & a_{s-1,2} & \dots & a_{s-1,s} \\ a_{s,1} & a_{s,2} & \dots & a_{s,s} \end{bmatrix}$$

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1s} \\ & b_{22} & \dots & b_{2s} \\ & 0 & \cdot & \\ & & & \cdot \\ & & & \cdot \\ & & & b_{s,s} \end{bmatrix}$$

and

$$C = [c_{1s+1}, c_{2s+1}, \dots, c_{ss+1}]^T.$$

The RK process is written as,

$$y_{n+1} = y_n + h \sum_{i,j=1}^s b_{ij} (k_i k_j)^{1/2}, \quad (4.4.5-2)$$

where

$$\left. \begin{aligned}
 k_1 &= f(x_n, Y_n) \\
 k_2 &= f(x_n + c_{2s+1}h, Y_n + ha_{21}k_1) \\
 &\vdots \\
 &\vdots \\
 k_s &= f(x_n + c_{s,s+1}h, Y_n + h \sum_{j=1}^{s-1} a_{sj}k_j)
 \end{aligned} \right\} \quad (4.4.5-2a)$$

Here we shall extend the matrix representation to the GM formulae. Some explicit RK schemes are represented as follows:

Four-stage schemes:

Classical:
$$\left[\begin{array}{cccc|c}
 0 & 0 & 0 & 0 & 0 \\
 1/2 & 0 & 0 & 0 & 1/2 \\
 0 & 1/2 & 0 & 0 & 1/2 \\
 0 & 0 & 1 & 0 & 1 \\
 \hline
 1/6 & 0 & 0 & 0 & 0 \\
 0 & 2/6 & 0 & 0 & 0 \\
 0 & 0 & 2/6 & 0 & 0 \\
 0 & 0 & 0 & 1/6 & 0
 \end{array} \right], \quad (4.4.5-3)$$

Kutta:
$$\left[\begin{array}{cccc|c}
 0 & 0 & 0 & 0 & 0 \\
 1/3 & 0 & 0 & 0 & 1/3 \\
 -1/3 & 1 & 0 & 0 & 2/3 \\
 1 & -1 & 1 & 0 & 1 \\
 \hline
 1/8 & 0 & 0 & 0 & 0 \\
 0 & 3/8 & 0 & 0 & 0 \\
 0 & 0 & 3/8 & 0 & 0 \\
 0 & 0 & 0 & 1/8 & 0
 \end{array} \right], \quad (4.4.5-4)$$

RK-GM:
$$\left[\begin{array}{cccc|c}
 0 & 0 & 0 & 0 & 0 \\
 1/2 & 0 & 0 & 0 & 1/2 \\
 0 & 1/2 & 0 & 0 & 1/2 \\
 0 & 0 & 1 & 0 & 1 \\
 \hline
 0 & 1/3 & 1/3 & -1/3 & 0 \\
 0 & 0 & 0 & 1/3 & 0 \\
 0 & 0 & 0 & 1/3 & 0 \\
 0 & 0 & 0 & 0 & 0
 \end{array} \right]. \quad (4.4.5-5)$$

Matrix representation of the RK-processes provides a brief description of such schemes. For the embedded RK schemes we propose a similar representation of the form (4.4.5-1) but with the diagonal elements of A representing the coefficients of the first approximation and the elements of B representing the coefficients of the second approximation. Thus we have the following matrix notation.

$$\text{AM-GM:} \quad \left[\begin{array}{cccc|c} 1/6 & 0 & 0 & 0 & 1/0 \\ 1/2 & 2/6 & 0 & 0 & 1/1/2 \\ 0 & 1/2 & 2/6 & 0 & 1/1/2 \\ 0 & 0 & 1 & 1/6 & 1 \\ \hline 0 & 1/3 & 1/3 & -1/3 & 1/0 \\ 0 & 0 & 0 & 1/3 & 1/0 \\ 0 & 0 & 0 & 1/3 & 1/0 \\ 0 & 0 & 0 & 0 & 1/0 \end{array} \right], \quad (4.4.5-6)$$

$$\text{Kutta Merson:} \quad \left[\begin{array}{cccc|c} 1/2 & 0 & 0 & 0 & 1/0 \\ 1/3 & 0 & 0 & 0 & 1/1/3 \\ 1/6 & 1/6 & -3/2 & 0 & 1/1/3 \\ 1/8 & 0 & 3/8 & 2 & 1/1/2 \\ 1/2 & 0 & -2/3 & 2 & 1 \\ \hline 1/6 & 0 & 0 & 0 & 1/0 \\ 0 & 0 & 0 & 0 & 1/0 \\ 0 & 0 & 0 & 0 & 1/0 \\ 0 & 0 & 0 & 4/6 & 1/0 \\ 0 & 0 & 0 & 0 & 1/1/6 \end{array} \right] \quad (4.4.5-7)$$

and Fehlberg 4(5) scheme:

$\frac{25}{216}$	0	0	0	0	0	0	0
$\frac{1}{4}$	0	0	0	0	0	0	$\frac{1}{4}$
$\frac{3}{32}$	$\frac{9}{32}$	$\frac{1408}{2565}$	0	0	0	0	$\frac{3}{8}$
$\frac{1932}{2197}$	$\frac{7200}{2197}$	$\frac{7296}{2197}$	$\frac{2197}{4104}$	0	0	0	$\frac{12}{13}$
$\frac{439}{216}$	-8	$\frac{3680}{513}$	$\frac{845}{4104}$	$-\frac{1}{5}$	0	0	1
$\frac{8}{27}$	2	$\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	0	0	$\frac{1}{2}$
$\frac{16}{135}$	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	$\frac{6656}{12825}$	0	0	0	0	0
0	0	0	$\frac{28561}{56430}$	0	0	0	0
0	0	0	0	$-\frac{9}{50}$	0	0	0
0	0	0	0	0	$\frac{2}{55}$	0	0

(4.4.5-8)

4.4.6 DISCUSSION OF THE IMSL, NAG AND RKF45 ERROR CONTROL STRATEGIES

The subroutine used in the IMSL routine DVERK is based on a code designed by T.E. Hull, W.H. Enright and K.R. Jackson. It uses RK formulae of orders 5 and 6 that were developed by J.H. Verner.

DVERK attempts to keep the global error proportional to a user-set tolerance TOL. The proportionality depends on the type of error control used, the differential equation and the range of integration. The proportionality is expected to be steady for the smaller values of TOL. Thus making TOL smaller improves the accuracy.

The routine adopts a measure of the 'scale' of a problem, from which each method can calculate its appropriate maximum step-size HMAX. The SCALE parameter

provides such an attempt. The use of SCALE is based on a theoretical study of the application of DVERK's formula to homogeneous linear equations with constant coefficients. Thus SCALE is exactly the Lipschitz constant.

Consequently DVERK works efficiently with non-stiff systems where derivative evaluations are inexpensive and the solutions are required at only a small number of spaced points.

RKF45 is a subroutine for solving initial-value problems in ordinary differential equations. It is based on RK formulae developed by E. Fehlberg in 1970 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. Comparison of the two values yields an error estimate which is used for step-size control. As in many other popular RK routines, the RKF45 adopts error per step criterion and local extrapolation.

RKF45 is primarily designed to solve non-stiff and mildly stiff differential equations when derivative evaluations are inexpensive. It should generally not be used when high accuracy is required.

The NAG subroutine D02YAF is based on the RK-Merson formula. This subroutine is used by all the NAG routines D02BAF, B02BBF, and D02BDF. The D02BAF and D02BBF routines integrate a system of first-order differential equations over an interval. The D02BDF integrates a system of first-order differential equations over a range and computes a global error estimate.

The accuracy of the integration is governed by the parameter TOL. The routines have been designed so that for most problems, the error in the solution at XEND is approximately proportional to TOL. However, the actual relation between TOL and the accuracy achieved cannot be given precisely, but can be estimated from the global error estimates obtained by D02BDF.

The global error estimates used in D02BDF are computed using a method similar to that in Shampine and Watt[1976]. D02BDF has an option of a stiffness check because the explicit RK-Merson method is not suitable for integrating stiff equations. The check used is an extension of a scheme described in Hall and Watt[1976].

In each routine we have substituted the standard formula with the new AM-GM pair and some interesting results were obtained.

4.4.7 EXPERIMENTAL RESULTS

The following is a list of sample problems used in the numerical experiments. The numerical results of testing the AM-GM and other methods are then obtained.

Problem 1 $y^{(1)} + 3x^2y = 0$

Initial condition $x = 0, y = 1$

Exact solution $y = e^{-x^3}$

Problem 2 $y^{(1)} + y = 0$

Initial condition $x = 0, y = 1$

Exact solution $y = e^{-x}$

Problem 3 $y^{(1)} - y = 0$

Initial condition $x = 0, y = 1$

Exact solution $y = e^x$

Problem 4 $y^{(1)} + y - x - 1 = 0$

Initial condition $x = 0, y = 1$

Exact solution $y = x + e^{-x}$

Problem 5 $y^{(1)} - 20(2 - x)^{-11} + 1 = 0$

Initial condition $x = 0, y = 2^{-9} - 1$

Exact solution $y = 2(2 - x)^{-10} - x - 1$

Problem 6 $y_1^{(1)} - y_3 = 0,$

$$y_2^{(1)} - y_4 = 0,$$

$$y_3^{(1)} + \frac{y_1}{z} = 0,$$

$$y_4^{(1)} + \frac{y_2}{z} = 0.$$

Initial conditions $x = 0, y_1 = 1 - e,$

$$y_2 = y_3 = 0, y_4 = a \left\{ \frac{1+e}{1-e} \right\}^{1/2},$$

where $z = \frac{1}{a^2} \{y_1^2 + y_2^2\}^{3/2}$ and $a = \pi.$

Problem 7 $y_1^{(1)} = \tan(y_3),$

$$y_2^{(1)} = -0.032 \tan(y_3)/y - 0.02 y \sec(y_3),$$

$$y_3^{(1)} = -0.032/y_2^2.$$

Initial conditions $x = 0, y_1 = 0, y_2 = 0.5$ and $y_3 = \frac{\pi}{5}.$

Problem 8 $y_1^{(1)} = -0.04y_1 + 1 \times 10^4 y_2 y_3,$

$$y_2^{(1)} = 0.04y_1 - 1 \times 10^4 y_2 y_3 - 3 \times 10^7 y_2^2,$$

$$y_3^{(1)} = 3 \times 10^7 y_2^2.$$

Initial conditions $x = 0, y_1 = 1, y_2 = y_3 = 0.$

Given is a list of the numerical experiments performed and the corresponding numerical results. The notation NFC denotes the number of function evaluations.

Experiment 1 : Comparison between the AM-GM method with error control and the RK-Fehlberg method with error control. Problems 1,2,3,4 and 5 were used in this experiment. The following parameters were set:

tolerance, TOL = 5×10^{-5} ,
 initial minimum step-size, HMIN = 0.02
 maximum step-size, HMAX = 0.1

Problem 1

Results from RK-Fehlberg method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.999406E+00	.999406E+00	.407979E-08	6
.18409	.1000000	.993781E+00	.993781E+00	.225594E-07	12
.28409	.1000000	.977333E+00	.977333E+00	.492812E-07	18
.38409	.1000000	.944913E+00	.944913E+00	.788102E-07	24
.48409	.1000000	.892755E+00	.892755E+00	.938627E-07	30
.58409	.1000000	.819330E+00	.819330E+00	.610187E-07	36
.68409	.1000000	.726048E+00	.726048E+00	.612959E-07	42
.78409	.1000000	.617513E+00	.617513E+00	.286069E-06	48
.88409	.1000000	.501067E+00	.501066E+00	.521689E-06	54
.98409	.1000000	.385574E+00	.385573E+00	.478684E-06	60
1.08409	.1000000	.279689E+00	.279689E+00	.363005E-06	66

Table(4.4.7a)

Results from RK-GM method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.999406E+00	.999406E+00	.110431E-07	4
.18409	.1000000	.993781E+00	.993781E+00	.420574E-07	8
.28409	.1000000	.977333E+00	.977333E+00	.723510E-07	12
.38409	.1000000	.944913E+00	.944913E+00	.102927E-06	16
.48409	.1000000	.892755E+00	.892755E+00	.138257E-06	20
.58409	.1000000	.819330E+00	.819330E+00	.176468E-06	24
.68409	.1000000	.726048E+00	.726048E+00	.155914E-06	28
.78409	.1000000	.617513E+00	.617513E+00	.184546E-06	32
.88409	.1000000	.501068E+00	.501066E+00	.153076E-05	36
.98409	.1000000	.385578E+00	.385573E+00	.516917E-05	40
1.08409	.1000000	.279702E+00	.279689E+00	.128152E-04	44

Table(4.4.7b)

Problem 2

Results from RK-Fehlberg method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.919349E+00	.919349E+00	.587557E-08	6
.18409	.1000000	.831861E+00	.831861E+00	.183618E-07	12
.28409	.1000000	.752699E+00	.752699E+00	.284184E-07	18
.38409	.1000000	.681070E+00	.681070E+00	.363947E-07	24
.48409	.1000000	.616258E+00	.616258E+00	.425955E-07	30
.58409	.1000000	.557613E+00	.557613E+00	.472866E-07	36
.68409	.1000000	.504549E+00	.504549E+00	.506991E-07	42
.78409	.1000000	.456535E+00	.456535E+00	.530339E-07	48
.88409	.1000000	.413090E+00	.413090E+00	.544652E-07	54
.98409	.1000000	.373779E+00	.373779E+00	.551438E-07	60
1.08409	.1000000	.338209E+00	.338210E+00	.552001E-07	66

Table (4.4.7c)

Results from RK-GM method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.919349E+00	.919349E+00	.345521E-07	4
.18409	.1000000	.831861E+00	.831861E+00	.106618E-06	8
.28409	.1000000	.752699E+00	.752699E+00	.164654E-06	12
.38409	.1000000	.681071E+00	.681070E+00	.210680E-06	16
.48409	.1000000	.616258E+00	.616258E+00	.246454E-06	20
.58409	.1000000	.557614E+00	.557613E+00	.273512E-06	24
.68409	.1000000	.504550E+00	.504549E+00	.293188E-06	28
.78409	.1000000	.456535E+00	.456535E+00	.306643E-06	32
.88409	.1000000	.413090E+00	.413090E+00	.314881E-06	36
.98409	.1000000	.373780E+00	.373779E+00	.318775E-06	40
1.08409	.1000000	.338210E+00	.338210E+00	.319076E-06	44

Table (4.4.7d)

Problem 3

Results from RK-Fehlberg method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.108773E+01	.108773E+01	.489335E-08	6
.18409	.1000000	.120212E+01	.120212E+01	.178206E-07	12
.28409	.1000000	.132855E+01	.132855E+01	.334129E-07	18
.38409	.1000000	.146828E+01	.146828E+01	.520878E-07	24
.48409	.1000000	.162270E+01	.162270E+01	.743212E-07	30
.58409	.1000000	.179336E+01	.179336E+01	.100655E-06	36
.68409	.1000000	.198197E+01	.198197E+01	.131706E-06	42
.78409	.1000000	.219041E+01	.219041E+01	.168175E-06	48
.88409	.1000000	.242078E+01	.242078E+01	.210858E-06	54
.98409	.1000000	.267538E+01	.267538E+01	.260659E-06	60
1.08409	.1000000	.295675E+01	.295675E+01	.318603E-06	66

Table (4.4.7e)

Results from RK-GM method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.108773E+01	.108773E+01	.355344E-07	4
.18409	.1000000	.120212E+01	.120212E+01	.131448E-06	8
.28409	.1000000	.132855E+01	.132855E+01	.247143E-06	12
.38409	.1000000	.146828E+01	.146828E+01	.385720E-06	16
.48409	.1000000	.162270E+01	.162270E+01	.550712E-06	20
.58409	.1000000	.179336E+01	.179336E+01	.746142E-06	24
.68409	.1000000	.198197E+01	.198197E+01	.976587E-06	28
.78409	.1000000	.219041E+01	.219041E+01	.124725E-05	32
.88409	.1000000	.242078E+01	.242078E+01	.156405E-05	36
.98409	.1000000	.267537E+01	.267538E+01	.193368E-05	40
1.08409	.1000000	.295674E+01	.295675E+01	.236377E-05	44

Table (4.4.7f)

Problem 4

Results from RK-GM method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.00005	.0000512	.100000E+01	.100000E+01	.177636E-13	40
.00009	.0000430	.100000E+01	.100000E+01	.177636E-13	44
.00026	.0001721	.100000E+01	.100000E+01	.175415E-13	48
.00092	.0006616	.100000E+01	.100000E+01	.175415E-13	52
.00252	.0015992	.100000E+01	.100000E+01	.175415E-13	56
.00613	.0036106	.100002E+01	.100002E+01	.124345E-13	60
.01337	.0072422	.100009E+01	.100009E+01	.152545E-12	64
.02675	.0133744	.100035E+01	.100035E+01	.366129E-11	68
.04973	.0229842	.100122E+01	.100122E+01	.554201E-10	72
.08689	.0371598	.100367E+01	.100367E+01	.611742E-09	76
.14381	.0569201	.100986E+01	.100986E+01	.509958E-08	80
.22682	.0830057	.102389E+01	.102389E+01	.327426E-07	84
.32682	.1000000	.104803E+01	.104803E+01	.949571E-07	88
.42682	.1000000	.107940E+01	.107940E+01	.145034E-06	92
.52682	.1000000	.111730E+01	.111730E+01	.184720E-06	96
.62682	.1000000	.116111E+01	.116111E+01	.215540E-06	100
.72682	.1000000	.121027E+01	.121027E+01	.238821E-06	104
.82682	.1000000	.126426E+01	.126426E+01	.255719E-06	108
.92682	.1000000	.132263E+01	.132263E+01	.267238E-06	112
1.02682	.1000000	.138497E+01	.138497E+01	.274249E-06	116

Table (4.4.7g)

Results from RK-Fehlberg method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	.100344E+01	.100344E+01	.587557E-08	6
.18409	.1000000	.101595E+01	.101595E+01	.183618E-07	12
.28409	.1000000	.103679E+01	.103679E+01	.284184E-07	18
.38409	.1000000	.106516E+01	.106516E+01	.363947E-07	24
.48409	.1000000	.110035E+01	.110035E+01	.425955E-07	30
.58409	.1000000	.114170E+01	.114170E+01	.472866E-07	36
.68409	.1000000	.118864E+01	.118864E+01	.506991E-07	42
.78409	.1000000	.124062E+01	.124062E+01	.530339E-07	48
.88409	.1000000	.129718E+01	.129718E+01	.544652E-07	54
.98409	.1000000	.135787E+01	.135787E+01	.551438E-07	60
1.08409	.1000000	.142230E+01	.142230E+01	.552001E-07	66

Table (4.4.7h)

Problem 5

Results from RK-Fehlberg method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	-.108109E+01	-.108109E+01	.292578E-08	6
.18409	.1000000	-.117896E+01	-.117896E+01	.188044E-07	12
.28409	.1000000	-.127505E+01	-.127505E+01	.561042E-07	18
.38409	.1000000	-.136761E+01	-.136761E+01	.148290E-06	24
.48409	.1000000	-.145288E+01	-.145288E+01	.389534E-06	30
.58409	.1000000	-.152234E+01	-.152233E+01	.106314E-05	36
.68409	.1000000	-.155563E+01	-.155563E+01	.308877E-05	42
.78409	.1000000	-.150095E+01	-.150094E+01	.972445E-05	48
.85437	.0702760	-.134084E+01	-.134083E+01	.124725E-04	60
.91240	.0580304	-.104879E+01	-.104878E+01	.147492E-04	72
.96116	.0487602	-.594932E+00	-.594915E+00	.166247E-04	84
1.00286	.0417026	.552076E-01	.552258E-01	.181931E-04	96

Table(4.4.7i)

Results from RK-GM method with error control

x	h	Numerical Solution	Exact Solution	Absolute Error	NFC
.08409	.0840896	-.108109E+01	-.108109E+01	.295967E-07	4
.18409	.1000000	-.117896E+01	-.117896E+01	.174971E-06	8
.28409	.1000000	-.127505E+01	-.127505E+01	.507427E-06	12
.38409	.1000000	-.136761E+01	-.136761E+01	.130539E-05	16
.48409	.1000000	-.145288E+01	-.145288E+01	.332768E-05	20
.54406	.0599730	-.149733E+01	-.149733E+01	.366976E-05	28
.58875	.0446882	-.152492E+01	-.152493E+01	.380406E-05	36
.62162	.0328730	-.154083E+01	-.154083E+01	.384765E-05	44
.64531	.0236821	-.154923E+01	-.154923E+01	.385915E-05	52
.66165	.0163399	-.155317E+01	-.155317E+01	.386139E-05	60
.67217	.0105249	-.155478E+01	-.155479E+01	.386168E-05	68
.67831	.0061345	-.155535E+01	-.155536E+01	.386170E-05	76
.68345	.0051405	-.155561E+01	-.155561E+01	.386171E-05	84
.68516	.0017166	-.155565E+01	-.155565E+01	.386171E-05	92
.68636	.0011961	-.155566E+01	-.155566E+01	.386171E-05	100
.68681	.0004558	-.155566E+01	-.155566E+01	.386171E-05	108
.68695	.0001395	-.155566E+01	-.155566E+01	.386171E-05	120
.68698	.0000268	-.155566E+01	-.155566E+01	.386171E-05	136
.68701	.0000310	-.155566E+01	-.155566E+01	.386171E-05	140
.68711	.0000961	-.155566E+01	-.155566E+01	.386171E-05	144
.68734	.0002328	-.155566E+01	-.155566E+01	.386171E-05	148
.68785	.0005090	-.155566E+01	-.155566E+01	.386171E-05	152
.68885	.0010045	-.155565E+01	-.155565E+01	.386171E-05	156
.69069	.0018337	-.155560E+01	-.155561E+01	.386171E-05	160
.69383	.0031422	-.155546E+01	-.155546E+01	.386171E-05	164
.69895	.0051240	-.155504E+01	-.155504E+01	.386172E-05	168
.70701	.0080562	-.155387E+01	-.155387E+01	.386184E-05	172
.71941	.0124032	-.155078E+01	-.155078E+01	.386295E-05	176
.73860	.0191830	-.154249E+01	-.154249E+01	.387484E-05	180
.77053	.0319317	-.151710E+01	-.151711E+01	.408108E-05	184
.80810	.0375717	-.146245E+01	-.146245E+01	.479280E-05	192
.84306	.0349575	-.137756E+01	-.137757E+01	.557572E-05	200
.87350	.0304464	-.126574E+01	-.126574E+01	.617476E-05	208
.90436	.0308547	-.110204E+01	-.110205E+01	.713799E-05	212
.93147	.0271094	-.900711E+00	-.900719E+00	.788741E-05	216
.95566	.0241886	-.659678E+00	-.659687E+00	.849461E-05	220
.97758	.0219218	-.375328E+00	-.375337E+00	.901139E-05	224
.99765	.0200747	-.439909E-01	-.440004E-01	.946410E-05	228
1.01617	.0185216	.338061E+00	.338051E+00	.986803E-05	232

Table(4.4.7j)

Experiment 2 : Investigation of integrating $y^{(1)} = -y$ from $y(0) = 1$ to $x=20$ for different values of the tolerance, TOL using the AM-GM method with error control.

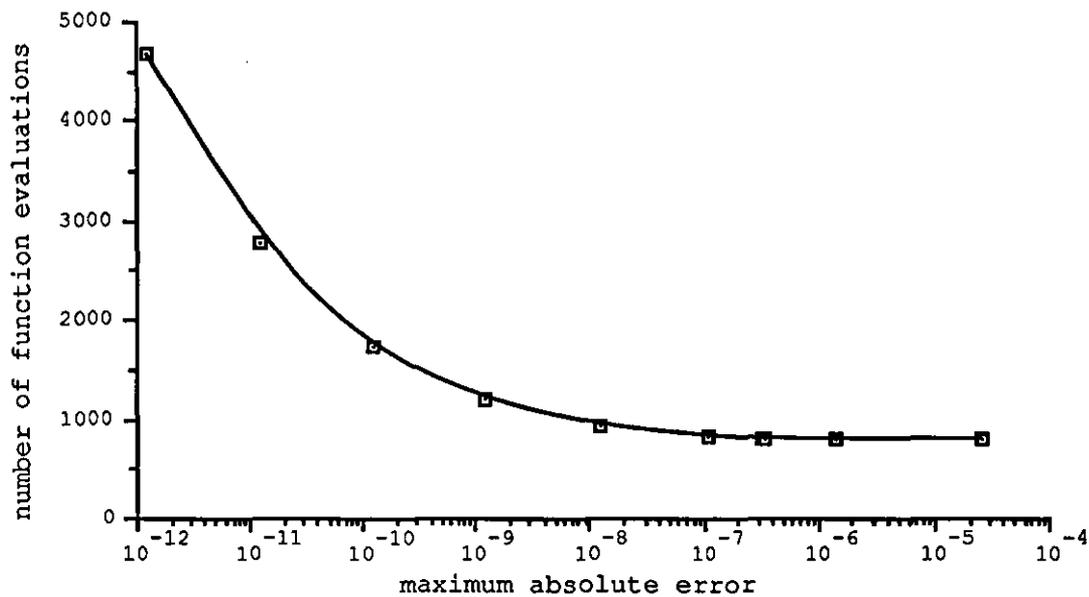


Figure (4.4.7) : Integrating the test function from $x = 0$ to $x = 20$ for different values of tolerance using the RK-GM method with error control

Experiment 3 : Substitution of the Kutta Verner standard formula in DVERK with the Kutta RK-GM formula. Problems 2,4,6 and 7 were used. The tolerance TOL was set at 0.1×10^{-3} .

Problem 2

1: Results from the classical RK & Kutta 4-stage 4th order method with error control

x	Numerical Solution	Exact Solution	Absolute Error	Time	FNC
1.00	.367879E+00	.367879E+00	.539633E-08	.12	157
2.00	.135335E+00	.135335E+00	.156107E-07	.06	247
3.00	.497871E-01	.497871E-01	.429597E-07	.04	302
4.00	.183157E-01	.183156E-01	.950234E-07	.02	337
5.00	.673811E-02	.673795E-02	.167268E-06	.02	362
6.00	.247925E-02	.247875E-02	.500959E-06	.01	377
7.00	.912789E-03	.911882E-03	.906754E-06	.01	387
8.00	.336062E-03	.335463E-03	.599566E-06	.00	397
9.00	.126023E-03	.123410E-03	.261352E-05	.00	402
10.00	.472587E-04	.453999E-04	.185882E-05	.00	407

Table (4.4.7k)

2: Results from the classical RK & RK-GM 4-stage 4th order method with error control

x	Numerical Solution	Exact Solution	Absolute Error	Time	FNC
1.00	.367930E+00	.367879E+00	.506931E-04	.01	15
2.00	.135461E+00	.135335E+00	.125865E-03	.01	25
3.00	.498728E-01	.497871E-01	.857769E-04	.01	35
4.00	.183617E-01	.183156E-01	.460887E-04	.01	45
5.00	.688565E-02	.673795E-02	.147701E-03	.00	50
6.00	.253509E-02	.247875E-02	.563426E-04	.01	60
7.00	.950661E-03	.911882E-03	.387786E-04	.01	65
8.00	.356498E-03	.335463E-03	.210351E-04	.01	70
9.00	.133687E-03	.123410E-03	.102768E-04	.01	75
10.00	.501325E-04	.453999E-04	.473256E-05	.00	80

Table (4.4.7l)

3: Results from the IMSL Routine

x	Numerical Solution	Exact Solution	Absolute Error	Time	FNC
1.00	.367879E+00	.367879E+00	.418952E-08	.01	16
2.00	.135335E+00	.135335E+00	.671057E-08	.01	32
3.00	.497871E-01	.497871E-01	.437037E-08	.01	48
4.00	.183152E-01	.183156E-01	.450001E-06	.00	56
5.00	.673762E-02	.673795E-02	.331680E-06	.01	64
6.00	.247857E-02	.247875E-02	.183134E-06	.01	72
7.00	.911792E-03	.911882E-03	.898537E-07	.01	80
8.00	.335421E-03	.335463E-03	.413260E-07	.01	88
9.00	.123392E-03	.123410E-03	.182455E-07	.01	96
10.00	.453921E-04	.453999E-04	.783142E-08	.00	104

Table (4.4.7m)

Problem 4

1: Results from the classical RK & Kutta 4-stage 4th order method with error control

x	Numerical Solution	Exact Solution	Absolute Error	Time	FNC
1.00	.136788E+01	.136788E+01	.247152E-07	.09	113
2.00	.213534E+01	.213534E+01	.121996E-07	.09	238
3.00	.304979E+01	.304979E+01	.638528E-08	.09	348
4.00	.401832E+01	.401832E+01	.348820E-08	.08	448
5.00	.500674E+01	.500674E+01	.189779E-08	.07	538
6.00	.600248E+01	.600248E+01	.104384E-08	.06	618
7.00	.700091E+01	.700091E+01	.559230E-09	.06	693
8.00	.800034E+01	.800034E+01	.288841E-09	.05	763
9.00	.900012E+01	.900012E+01	.146544E-09	.05	828
10.00	.100000E+02	.100000E+02	.737526E-10	.05	888

Table (4.4.7n)

2: Results from the classical RK & RK-GM 4-stage 4th order method with error control

x	Numerical Solution	Exact Solution	Absolute Error	Time	FNC
1.00	.136790E+01	.136788E+01	.159371E-04	.03	33
2.00	.213545E+01	.213534E+01	.113069E-03	.01	43
3.00	.304987E+01	.304979E+01	.810657E-04	.01	53
4.00	.401870E+01	.401832E+01	.384911E-03	.00	58
5.00	.500701E+01	.500674E+01	.274759E-03	.01	63
6.00	.600263E+01	.600248E+01	.151013E-03	.01	68
7.00	.700099E+01	.700091E+01	.742799E-04	.00	73
8.00	.800037E+01	.800034E+01	.343481E-04	.00	78
9.00	.900014E+01	.900012E+01	.152692E-04	.01	83
10.00	.100001E+02	.100000E+02	.660470E-05	.00	88

Table (4.4.7o)

3: Results from the IMSL Routine

x	Numerical Solution	Exact Solution	Absolute Error	Time	FNC
1.00	.136788E+01	.136788E+01	.418952E-08	.01	16
2.00	.213534E+01	.213534E+01	.671057E-08	.01	32
3.00	.304979E+01	.304979E+01	.437037E-08	.01	48
4.00	.401832E+01	.401832E+01	.450001E-06	.01	56
5.00	.500674E+01	.500674E+01	.331680E-06	.01	64
6.00	.600248E+01	.600248E+01	.183134E-06	.00	72
7.00	.700091E+01	.700091E+01	.898537E-07	.01	80
8.00	.800034E+01	.800034E+01	.413260E-07	.01	88
9.00	.900012E+01	.900012E+01	.182455E-07	.00	96
10.00	.100000E+02	.100000E+02	.783142E-08	.01	104

Table (4.4.7p)

Problem 6

1: Results from the classical RK & Kutta 4-stage 4th order method with error control

x	Numerical Solution				Time	NFC
	Y ₁	Y ₂	Y ₃	Y ₄		
1.00	.294418E+00	.812179E+00	-.762596E+00	.479233E+00	.36	182
2.00	-.490300E+00	.939875E+00	-.719180E+00	-.172382E+00	.32	342
3.00	-.105403E+01	.575706E+00	-.388829E+00	-.509100E+00	.28	482
4.00	-.125000E+01	-.210655E-06	.118433E-06	-.608367E+00	.28	622
5.00	-.105403E+01	-.575706E+00	.388830E+00	-.509100E+00	.27	762
6.00	-.490299E+00	-.939875E+00	.719180E+00	-.172382E+00	.27	902
7.00	.294418E+00	-.812178E+00	.762596E+00	.479233E+00	.31	1062
8.00	.750000E+00	.462146E-06	-.538027E-06	.101394E+01	.33	1232
9.00	.294417E+00	.812179E+00	-.762596E+00	.479232E+00	.34	1402
10.00	-.490300E+00	.939875E+00	-.719180E+00	-.172383E+00	.32	1562

Table (4.4.7q)

2: Results from the classical RK & RK-GM 4-stage 4th order method with error control

x	Numerical Solution				Time	NFC
	Y ₁	Y ₂	Y ₃	Y ₄		
1.00	.294416E+00	.812087E+00	-.762683E+00	.479130E+00	.12	51
2.00	-.490376E+00	.939571E+00	-.719202E+00	-.172694E+00	.15	115
3.00	-.105398E+01	.575037E+00	-.388564E+00	-.509480E+00	.03	130
4.00	-.124950E+01	-.977888E-03	.603705E-03	-.608577E+00	.04	149
5.00	-.105278E+01	-.576665E+00	.389689E+00	-.508829E+00	.04	168
6.00	-.488217E+00	-.940125E+00	.719918E+00	-.171187E+00	.03	178
7.00	.296660E+00	-.810653E+00	.761798E+00	.481478E+00	.20	260
8.00	.749806E+00	.308763E-02	-.354989E-02	.101403E+01	.07	288
9.00	.291466E+00	.813415E+00	-.764045E+00	.476280E+00	.09	321
10.00	-.493434E+00	.938159E+00	-.718334E+00	-.175081E+00	.16	385

Table (4.4.7r)

3: Results from the IMSL Routine

x	Numerical Solution				Time	NFC
	Y ₁	Y ₂	Y ₃	Y ₄		
1.00	.294430E+00	.812194E+00	-.762571E+00	.479277E+00	.04	24
2.00	-.490250E+00	.939978E+00	-.719158E+00	-.172255E+00	.03	40
3.00	-.105400E+01	.575948E+00	-.388897E+00	-.508965E+00	.03	56
4.00	-.125008E+01	.345204E-03	-.145958E-03	-.608308E+00	.03	72
5.00	-.105429E+01	-.575332E+00	.388635E+00	-.509186E+00	.02	80
6.00	-.490732E+00	-.939623E+00	.719159E+00	-.172690E+00	.01	88
7.00	.294100E+00	-.812237E+00	.762871E+00	.478969E+00	.03	104
8.00	.750025E+00	-.176106E-03	.448063E-03	.101408E+01	.03	120
9.00	.294973E+00	.812375E+00	-.762019E+00	.479978E+00	.04	144
10.00	-.489401E+00	.941219E+00	-.719227E+00	-.170916E+00	.03	160

Table (4.4.7s)

Problem 7

1: Results from the classical RK & Kutta 4-stage 4th order method with error control

Numerical Solution					
x	Y ₁	Y ₂	Y ₃	Time	NFC
1.00	.627515E+00	.446394E+00	.484640E+00	.38	157
2.00	.104995E+01	.405429E+00	.306828E+00	.28	277
3.00	.125818E+01	.380484E+00	.978218E-01	.21	367
4.00	.124323E+01	.374252E+00	-.128851E+00	.21	457
5.00	.995912E+00	.387238E+00	-.351464E+00	.22	552
6.00	.506401E+00	.417250E+00	-.550776E+00	.31	682
7.00	-.236407E+00	.460480E+00	-.718036E+00	.38	842
8.00	-.124552E+01	.512973E+00	-.853854E+00	.54	1027
9.00	-.253658E+01	.571416E+00	-.963178E+00	.53	1187
10.00	-.412869E+01	.633281E+00	-.105167E+01	.44	1317

Table (4.4.7t)

2: Results from the classical RK & RK-GM 4-stage 4th order method with error control

Numerical Solution					
x	Y ₁	Y ₂	Y ₃	Time	NFC
1.00	.627514E+00	.446394E+00	.484640E+00	.04	15
2.00	.104995E+01	.405429E+00	.306829E+00	.02	25
3.00	.125818E+01	.380484E+00	.978233E-01	.04	39
4.00	.124322E+01	.374252E+00	-.128849E+00	.09	76
5.00	.995910E+00	.387237E+00	-.351462E+00	.03	91
6.00	.506407E+00	.417249E+00	-.550774E+00	.02	101
7.00	-.236295E+00	.460477E+00	-.718025E+00	.02	106
8.00	-.124531E+01	.512966E+00	-.853839E+00	.01	111
9.00	-.253627E+01	.571408E+00	-.963161E+00	.01	116
10.00	-.412829E+01	.633272E+00	-.105165E+01	.02	121

Table (4.4.7u)

3: Results from the IMSL Routine

Numerical Solution					
x	Y ₁	Y ₂	Y ₃	Time	NFC
1.00	.627515E+00	.446394E+00	.484640E+00	.03	16
2.00	.104995E+01	.405429E+00	.306828E+00	.02	24
3.00	.125818E+01	.380484E+00	.978213E-01	.01	32
4.00	.124322E+01	.374252E+00	-.128851E+00	.02	40
5.00	.995910E+00	.387238E+00	-.351464E+00	.02	48
6.00	.506398E+00	.417250E+00	-.550776E+00	.02	56
7.00	-.236411E+00	.460481E+00	-.718036E+00	.02	64
8.00	-.124552E+01	.512973E+00	-.853854E+00	.02	72
9.00	-.253658E+01	.571416E+00	-.963178E+00	.02	80
10.00	-.412869E+01	.633281E+00	-.105167E+01	.02	88

Table (4.4.7v)

Experiment 4 : Substitution of the Fehlberg standard formula in RKF45 with the RK-GM formula. Problems 2 and 6 were used.

Problem 2

RELERR = .10E-08

ABSERR = .00E+00

Results from RKF45

x	Numerical Solution	Exact Solution	Absolute Error	NFC
.0	.100000000E+01	.100000000E+01	.000000000E+00	1
.5	.606530660E+00	.606530660E+00	.109230291E-09	67
1.0	.367879441E+00	.367879441E+00	.133512368E-09	127
1.5	.223130160E+00	.223130160E+00	.121738647E-09	187
2.0	.135335283E+00	.135335283E+00	.985598270E-10	247
2.5	.820849985E-01	.820849986E-01	.747739926E-10	307
3.0	.497870683E-01	.497870684E-01	.544473217E-10	367
3.5	.301973834E-01	.301973834E-01	.385401329E-10	427
4.0	.183156389E-01	.183156389E-01	.267214827E-10	487
4.5	.111089965E-01	.111089965E-01	.182366813E-10	547
5.0	.673794699E-02	.673794700E-02	.122919166E-10	607
5.5	.408677143E-02	.408677144E-02	.820195734E-11	667
6.0	.247875217E-02	.247875218E-02	.542753256E-11	727
6.5	.150343919E-02	.150343919E-02	.356659580E-11	787
7.0	.911881963E-03	.911881966E-03	.232982383E-11	847
7.5	.553084369E-03	.553084370E-03	.151414103E-11	907
8.0	.335462627E-03	.335462628E-03	.979652069E-12	967
8.5	.203468368E-03	.203468369E-03	.631356729E-12	1027
9.0	.123409804E-03	.123409804E-03	.405480960E-12	1087
9.5	.748518296E-04	.748518299E-04	.259609866E-12	1147
10.0	.453999296E-04	.453999298E-04	.165754549E-12	1207
10.5	.275364492E-04	.275364493E-04	.105565245E-12	1267
11.0	.167017007E-04	.167017008E-04	.670794359E-13	1327
11.5	.101300936E-04	.101300936E-04	.425361938E-13	1387
12.0	.614421233E-05	.614421235E-05	.269218555E-13	1447
12.5	.372665316E-05	.372665317E-05	.170096731E-13	1507
13.0	.226032940E-05	.226032941E-05	.107297793E-13	1567
13.5	.137095908E-05	.137095909E-05	.675837212E-14	1627
14.0	.831528715E-06	.831528719E-06	.425105624E-14	1687
14.5	.504347660E-06	.504347663E-06	.267052463E-14	1747
15.0	.305902319E-06	.305902321E-06	.167563440E-14	1807
15.5	.185539135E-06	.185539136E-06	.105021595E-14	1867
16.0	.112535174E-06	.112535175E-06	.657544702E-15	1927
16.5	.682560334E-07	.682560338E-07	.411288772E-15	1987
17.0	.413993769E-07	.413993772E-07	.257021864E-15	2047
17.5	.251099914E-07	.251099916E-07	.160478541E-15	2107
18.0	.152299796E-07	.152299797E-07	.100117226E-15	2167
18.5	.923744960E-08	.923744966E-08	.624115266E-16	2227
19.0	.560279640E-08	.560279644E-08	.388779454E-16	2287
19.5	.339826780E-08	.339826782E-08	.242014221E-16	2347
20.0	.206115361E-08	.206115362E-08	.150554196E-16	2407

Table (4.4.7w)

Results from RKGM44

x	Numerical Solution	Exact Solution	Absolute Error	NFC
.0	.100000000E+01	.100000000E+01	.000000000E+00	1
.5	.606530672E+00	.606530660E+00	.119927582E-07	49
1.0	.367879459E+00	.367879441E+00	.174246161E-07	89
1.5	.223130177E+00	.223130160E+00	.167640901E-07	129
2.0	.135335297E+00	.135335283E+00	.139058159E-07	169
2.5	.820850093E-01	.820849986E-01	.107108639E-07	209
3.0	.497870762E-01	.497870684E-01	.787252005E-08	249
3.5	.301973890E-01	.301973834E-01	.561181949E-08	289
4.0	.183156428E-01	.183156389E-01	.391020614E-08	329
4.5	.111089992E-01	.111089965E-01	.267939583E-08	369
5.0	.673794881E-02	.673794700E-02	.181151460E-08	409
5.5	.408677265E-02	.408677144E-02	.121191589E-08	449
6.0	.247875298E-02	.247875218E-02	.803643984E-09	489
6.5	.150343972E-02	.150343919E-02	.529062311E-09	529
7.0	.911882312E-03	.911881966E-03	.346125250E-09	569
7.5	.553084595E-03	.553084370E-03	.225247657E-09	609
8.0	.335462774E-03	.335462628E-03	.145903093E-09	649
8.5	.203468463E-03	.203468369E-03	.941272558E-10	689
9.0	.123409865E-03	.123409804E-03	.605064813E-10	729
9.5	.748518687E-04	.748518299E-04	.387710322E-10	769
10.0	.453999545E-04	.453999298E-04	.247723319E-10	809
10.5	.275364651E-04	.275364493E-04	.157873981E-10	849
11.0	.167017108E-04	.167017008E-04	.100377983E-10	889
11.5	.101301000E-04	.101300936E-04	.636863125E-11	929
12.0	.614421639E-05	.614421235E-05	.403282805E-11	969
12.5	.372665572E-05	.372665317E-05	.254918535E-11	1009
13.0	.226033102E-05	.226032941E-05	.160872061E-11	1049
13.5	.137096010E-05	.137095909E-05	.101368534E-11	1089
14.0	.831529357E-06	.831528719E-06	.637846514E-12	1129
14.5	.504348063E-06	.504347663E-06	.400833291E-12	1169
15.0	.305902572E-06	.305902321E-06	.251584569E-12	1209
15.5	.185539294E-06	.185539136E-06	.157729268E-12	1249
16.0	.112535274E-06	.112535175E-06	.987824414E-13	1289
16.5	.682560956E-07	.682560338E-07	.618038243E-13	1329
17.0	.413994158E-07	.413993772E-07	.386317890E-13	1369
17.5	.251100157E-07	.251099916E-07	.241263777E-13	1409
18.0	.152299948E-07	.152299797E-07	.150549321E-13	1449
18.5	.923745905E-08	.923744966E-08	.938695870E-14	1489
19.0	.560280229E-08	.560279644E-08	.584855591E-14	1529
19.5	.339827146E-08	.339826782E-08	.364138811E-14	1569
20.0	.206115589E-08	.206115362E-08	.226566345E-14	1609

Table (4.4.7x)

Problem 6

RELERR = .10E-04, ABSERR = .00E+00

Results from RKF45

x	Y ₁	Y ₂	NFC
.0	.750000000E+00	.000000000E+00	1
.5	.619767068E+00	.477791194E+00	29
1.0	.294414200E+00	.812176162E+00	41
1.5	-.105182402E+00	.958032935E+00	53
2.0	-.490307436E+00	.939865968E+00	65
2.5	-.813950377E+00	.799577611E+00	77
3.0	-.105403733E+01	.575689270E+00	89
3.5	-.120073792E+01	.300141410E+00	101
4.0	-.124999925E+01	-.203220177E-04	113
4.5	-.120073054E+01	-.300180438E+00	125
5.0	-.105402372E+01	-.575723620E+00	137
5.5	-.813932812E+00	-.799604771E+00	149
6.0	-.490289189E+00	-.939884469E+00	161
6.5	-.105167252E+00	-.958042932E+00	173
7.0	.294423177E+00	-.812179506E+00	185
7.5	.619768334E+00	-.477789287E+00	197
8.0	.749992743E+00	.519256401E-05	215
8.5	.619749395E+00	.477797051E+00	233
9.0	.294384015E+00	.812175304E+00	245
9.5	-.105220979E+00	.958018385E+00	257
10.0	-.490347442E+00	.939834876E+00	269
10.5	-.813985082E+00	.799530758E+00	281
11.0	-.105406135E+01	.575629678E+00	293
11.5	-.120074734E+01	.300073386E+00	305
12.0	-.124999155E+01	-.916976293E-04	317

Table (4.4.7y)

Results from RKGM44

x	Y ₁	Y ₂	NFC
.0	.750000000E+00	.000000000E+00	1
.5	.619889683E+00	.476891715E+00	5
1.0	.294371048E+00	.810342931E+00	9
1.5	-.105820308E+00	.954515800E+00	13
2.0	-.491257114E+00	.933523687E+00	17
2.5	-.814175266E+00	.789461516E+00	21
3.0	-.105198408E+01	.561354343E+00	25
3.5	-.119456309E+01	.281781047E+00	29
4.0	-.123774775E+01	-.214859845E-01	33
4.5	-.118050702E+01	-.322990982E+00	37
5.0	-.102422819E+01	-.597104235E+00	41
5.5	-.773678543E+00	-.815524725E+00	45
6.0	-.440221688E+00	-.944820297E+00	49
6.5	-.490553995E-01	-.945112714E+00	53
7.0	.347092196E+00	-.774931255E+00	57
7.5	.652291257E+00	-.416360037E+00	61
8.0	.746470302E+00	.702246288E-01	65
8.5	.581257348E+00	.532921413E+00	69
9.0	.236039898E+00	.838300231E+00	73
9.5	-.168397143E+00	.952932064E+00	77
10.0	-.547661917E+00	.906117222E+00	81
10.5	-.857639536E+00	.741077913E+00	85
11.0	-.107769430E+01	.496996055E+00	89
11.5	-.119875599E+01	.206688531E+00	93
12.0	-.121736269E+01	-.101479029E+00	97

Table (4.4.7z)

Experiment 5 : Substitution of the Kutta Merson standard formula in D02YAF with the AM-GM formula. Problems 2,7 and 8 were used.

Problem 2

Results from D02BBF

Calculation with TOL = .1E-02
x and solution at equally spaced points

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01	.10000000E+01	.00000000E+00
1.00	.36796495E+00	.36787940E+00	.85499190E-04
2.00	.13538523E+00	.13533530E+00	.49951740E-04
3.00	.49811389E-01	.49787070E-01	.24322440E-04
4.00	.18325715E-01	.18315640E-01	.10075480E-04
5.00	.67261683E-02	.67379470E-02	.11778730E-04
6.00	.24306972E-02	.24787520E-02	.48055080E-04
7.00	.87849704E-03	.91188210E-03	.33384940E-04
8.00	.32162751E-03	.33546260E-03	.13835120E-04

IFAIL = 0

Table (4.4.7aa)

Calculation with TOL = .1E-03
x and solution at equally spaced points

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01	.10000000E+01	.00000000E+00
1.00	.36790186E+00	.36787940E+00	.22406850E-04
2.00	.13536003E+00	.13533530E+00	.24752720E-04
3.00	.49804650E-01	.49787070E-01	.17583530E-04
4.00	.18325208E-01	.18315640E-01	.95683550E-05
5.00	.67422989E-02	.67379470E-02	.43518610E-05
6.00	.24800357E-02	.24787520E-02	.12833940E-05
7.00	.91227415E-03	.91188210E-03	.39217260E-06
8.00	.33545629E-03	.33546260E-03	.63371990E-08

IFAIL = 0

Table (4.4.7ab)

Calculation with TOL = .1E-04
x and solution at equally spaced points

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01	.10000000E+01	.00000000E+00
1.00	.36788436E+00	.36787940E+00	.49120900E-05
2.00	.13534067E+00	.13533530E+00	.53886090E-05
3.00	.49792005E-01	.49787070E-01	.49377700E-05
4.00	.18319082E-01	.18315640E-01	.34423050E-05
5.00	.67402237E-02	.67379470E-02	.22767450E-05
6.00	.24800295E-02	.24787520E-02	.12772290E-05
7.00	.91256152E-03	.91188210E-03	.67955050E-06
8.00	.33596529E-03	.33546260E-03	.50266210E-06

IFAIL = 0

Table (4.4.7ac)

Results from RKGM44

Calculation with TOL = .1E-02
x and solution at equally spaced points

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01	.10000000E+01	.00000000E+00
1.00	.36787944E+00	.36787940E+00	.91496190E-08
2.00	.13533528E+00	.13533530E+00	.17479400E-08
3.00	.49787068E-01	.49787070E-01	.15345870E-08
4.00	.18315639E-01	.18315640E-01	.14437950E-09
5.00	.67379535E-02	.67379470E-02	.65383120E-08
6.00	.24788007E-02	.24787520E-02	.48380780E-07
7.00	.91213465E-03	.91188210E-03	.25267300E-06
8.00	.33600717E-03	.33546260E-03	.54454480E-06

IFAIL = 0

Table (4.4.7ad)

Calculation with TOL = .1E-03
x and solution at equally spaced points

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01	.10000000E+01	.00000000E+00
1.00	.36787944E+00	.36787940E+00	.91497550E-08
2.00	.13533528E+00	.13533530E+00	.17453330E-08
3.00	.49787068E-01	.49787070E-01	.14909810E-08
4.00	.18315639E-01	.18315640E-01	.45799620E-09
5.00	.67379470E-02	.67379470E-02	.26804160E-11
6.00	.24787522E-02	.24787520E-02	.80731060E-10
7.00	.91188231E-03	.91188210E-03	.33250880E-09
8.00	.33546482E-03	.33546260E-03	.21973290E-08

IFAIL = 0

Table (4.4.7ae)

Calculation with TOL = .1E-04
x and solution at equally spaced points

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01	.10000000E+01	.00000000E+00
1.00	.36787944E+00	.36787940E+00	.91497550E-08
2.00	.13533528E+00	.13533530E+00	.17453330E-08
3.00	.49787068E-01	.49787070E-01	.14909770E-08
4.00	.18315639E-01	.18315640E-01	.45810380E-09
5.00	.67379470E-02	.67379470E-02	.75775670E-12
6.00	.24787522E-02	.24787520E-02	.10857130E-09
7.00	.91188197E-03	.91188210E-03	.69348740E-11
8.00	.33546263E-03	.33546260E-03	.49373470E-11

IFAIL = 0

Table (4.4.7af)

Results from D02BAF

Calculation with TOL = .1E-02

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01		
8.00	.32162751E-03	.33546260E-03	.13835120E-04

IFAIL = 0

Calculation with TOL = .1E-03

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01		
8.00	.33545629E-03	.33546260E-03	.63371990E-08

IFAIL = 0

Calculation with TOL = .1E-04

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01		
8.00	.33596529E-03	.33546260E-03	.50266210E-06

Table (4.4.7ag)

Results from RKGM44

Calculation with TOL= .1E-02

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01		
8.00	.33600717E-03	.33546260E-03	.54454480E-06

IFAIL = 0

Calculation with TOL= .1E-03

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01		
8.00	.33546482E-03	.33546260E-03	.21973240E-08

IFAIL = 0

Calculation with TOL= .1E-04

x	Computed Solution	Exact Solution	Absolute Error
.00	.10000000E+01		
8.00	.33546263E-03	.33546260E-03	.49373540E-11

IFAIL = 0

Table (4.4.7ah)

Problem 7

Results from D02BBF

Calculation with TOL = .1E-02

x	Y ₁	Y ₂	Y ₃
.00	.00000E+00	.50000E+00	.62857E+00
1.00	.62692E+00	.44644E+00	.48496E+00
2.00	.10490E+01	.40548E+00	.30698E+00
3.00	.12573E+01	.38053E+00	.98060E-01
4.00	.12423E+01	.37428E+00	-.12849E+00
5.00	.99526E+00	.38724E+00	-.35122E+00
6.00	.50661E+00	.41720E+00	-.55049E+00
7.00	-.23564E+00	.46041E+00	-.71775E+00
8.00	-.12441E+01	.51288E+00	-.85360E+00

Table (4.4.7ai)

Calculation with TOL = .1E-03

x	Y ₁	Y ₂	Y ₃
.00	.00000E+00	.50000E+00	.62857E+00
1.00	.62751E+00	.44639E+00	.48464E+00
2.00	.10499E+01	.40543E+00	.30683E+00
3.00	.12581E+01	.38049E+00	.97830E-01
4.00	.12432E+01	.37425E+00	-.12884E+00
5.00	.99589E+00	.38724E+00	-.35145E+00
6.00	.50643E+00	.41725E+00	-.55076E+00

Table (4.4.7aj)

Results from RKGM44

Calculation with TOL = .1E-02

x	Y ₁	Y ₂	Y ₃
.00	.00000E+00	.50000E+00	.62857E+00
1.00	.62751E+00	.44639E+00	.48464E+00
2.00	.10500E+01	.40543E+00	.30683E+00
3.00	.12582E+01	.38048E+00	.97822E-01
4.00	.12432E+01	.37425E+00	-.12885E+00
5.00	.99591E+00	.38724E+00	-.35146E+00
6.00	.50640E+00	.41725E+00	-.55078E+00
7.00	-.23641E+00	.46048E+00	-.71804E+00
8.00	-.12455E+01	.51297E+00	-.85385E+00

Table (4.4.7ak)

Calculation with TOL = .1E-03

x	Y ₁	Y ₂	Y ₃
.00	.00000E+00	.50000E+00	.62857E+00
1.00	.62751E+00	.44639E+00	.48464E+00
2.00	.10500E+01	.40543E+00	.30683E+00
4.00	.12432E+01	.37425E+00	-.12885E+00
5.00	.99591E+00	.38724E+00	-.35146E+00
6.00	.50640E+00	.41725E+00	-.55078E+00
7.00	-.23641E+00	.46048E+00	-.71804E+00
8.00	-.12455E+01	.51297E+00	-.85385E+00

Table (4.4.7al)

Results from D02BAF

Tolerance used, TOL = .1E-02

x	Y ₁	Y ₂	Y ₃
.00	.00000000E+00	.50000000E+00	.62857144E+00
8.00	-.12440529E+01	.51287614E+00	-.85360339E+00

IFAIL = 0

Tolerance used, TOL = .1E-03

x	Y ₁	Y ₂	Y ₃
.00	.00000000E+00	.50000000E+00	.62857144E+00
8.00	-.12454177E+01	.51296657E+00	-.85384173E+00

IFAIL = 0

Tolerance used, TOL = .1E-04

x	Y ₁	Y ₂	Y ₃
.00	.00000000E+00	.50000000E+00	.62857144E+00
8.00	-.12455092E+01	.51297191E+00	-.85385267E+00

IFAIL = 0

Table (4.4.7am)

Results from RKGM44

Tolerance used, TOL = .1E-02

x	Y ₁	Y ₂	Y ₃
.00	.00000000E+00	.50000000E+00	.62857144E+00
8.00	-.12455199E+01	.51297257E+00	-.85385403E+00

IFAIL = 0

Tolerance used, TOL = .1E-03

x	Y ₁	Y ₂	Y ₃
.00	.00000000E+00	.50000000E+00	.62857144E+00
8.00	-.12455199E+01	.51297257E+00	-.85385403E+00

IFAIL = 0

Tolerance used, TOL = .1E-04

x	Y ₁	Y ₂	Y ₃
.00	.00000000E+00	.50000000E+00	.62857144E+00
8.00	-.12455199E+01	.51297257E+00	-.85385403E+00

IFAIL = 0

Table (4.4.7an)

Problem 8

Tolerance used, TOL= .1E-05

Results from D02BDF

X AND SOLUTION	.13478	.99475	.00004	.00521
CURRENT ERROR ESTIMATES	.45E-06	.13E-05	-.18E-05	
MAXIMUM ERROR ESTIMATES	.48E-06	.15E-05	-.18E-05	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		1.	73.	53.
STIFFNESS FACTOR	1.0000			

X AND SOLUTION	.27038	.98974	.00003	.01023
CURRENT ERROR ESTIMATES	.19E-05	-.29E-06	-.16E-05	
MAXIMUM ERROR ESTIMATES	.19E-05	.15E-05	-.31E-05	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		1.	148.	53.
STIFFNESS FACTOR	1.0000			

X AND SOLUTION	.30000	.98867	.00003	.01129
CURRENT ERROR ESTIMATES	.23E-05	-.71E-07	-.22E-05	
MAXIMUM ERROR ESTIMATES	.23E-05	.15E-05	-.35E-05	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		1.	164.	53.
STIFFNESS FACTOR	.9126			

Table (4.4.7ao)

Results from RKGM44

X AND SOLUTION	.03665	.99854	.00004	.00142
CURRENT ERROR ESTIMATES	.10E-12	-.60E-15	-.10E-12	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.07852	.99691	.00004	.00306
CURRENT ERROR ESTIMATES	.10E-12	-.63E-15	-.99E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.12108	.99527	.00004	.00469
CURRENT ERROR ESTIMATES	.98E-13	-.66E-15	-.97E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.16437	.99363	.00004	.00633
CURRENT ERROR ESTIMATES	.96E-13	-.69E-15	-.96E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.20838	.99200	.00004	.00797
CURRENT ERROR ESTIMATES	.95E-13	-.72E-15	-.94E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.25313	.99036	.00003	.00961
CURRENT ERROR ESTIMATES	.93E-13	-.76E-15	-.93E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.29864	.98872	.00003	.01124
CURRENT ERROR ESTIMATES	.92E-13	-.79E-15	-.91E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	1.0000			
X AND SOLUTION	.30000	.98867	.00003	.01129
CURRENT ERROR ESTIMATES	.92E-13	-.59E-15	-.91E-13	
MAXIMUM ERROR ESTIMATES	.10E-12	.92E-09	-.92E-09	
NUMBER OF SIGN CHANGES FOR EACH ESTIMATE		5.	1.	0.
STIFFNESS FACTOR	.5000			

Table (4.4.7ap)

4.5 RK-GM METHOD FOR SYSTEM OF ODES

In this section we shall investigate the feasibility of extending the RK-GM methods to systems of ODEs. Before we deal with the RK-GM method, we shall first discuss how can the classical RK method^{can} be used to solve a system of ODEs.

Consider the classical RK method of order four given by

$$Y_{n+1} = Y_n + \frac{h}{6} [k_1 + 2(k_2 + k_3) + k_4] \quad (4.5-1)$$

where,

$$\left. \begin{aligned} k_1 &= f(x_n, Y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, Y_n + \frac{hk_1}{2}\right) \\ k_3 &= f\left(x_n + \frac{h}{2}, Y_n + \frac{hk_2}{2}\right) \\ k_4 &= f(x_n + h, Y_n + hk_3) \end{aligned} \right\} \quad (4.5-1a)$$

is used to solve the first-order initial-value problem (4.1-1). We shall now extend (4.5-1) and (5.4-1a) to solve the system of first-order differential equations (3.2.1-8) as follows.

Choose an integer $N > 0$ and as usual set $h = (b-a)/N$. The interval $[a, b]$ is partitioned into N subintervals with mesh points such that for each $j = 0, 1, \dots, N$

$$x_j = a + jh. \quad (4.5-2)$$

Let $y_{i,j}$ denote the approximation of $z_i(x_j)$ for each $j = 0, 1, \dots, N$ and $i = 1, 2, \dots, m$; that is $y_{i,j}$ approximates the i^{th} solution $z_i(x)$ of (3.2.1-8) at the j^{th} mesh point x_j . For the initial conditions, set

$$\left. \begin{aligned} Y_{10} &= \alpha_1 \\ Y_{20} &= \alpha_2 \\ &\vdots \\ &\vdots \\ Y_{m0} &= \alpha_m \end{aligned} \right\} \quad (4.5-3)$$

Suppose the values $Y_{1j}, Y_{2j}, \dots, Y_{mj}$ have been computed, then $Y_{1,j+1}, Y_{2,j+1}, \dots, Y_{m,j+1}$ can be obtained by first calculating the following quantities

$$k_{1i} = f_i(x_j; Y_{1j}, Y_{2j}, \dots, Y_{mj}) \quad (4.5-4a)$$

for each $i = 1, 2, \dots, m$;

$$k_{2i} = f_i\left(x_j + \frac{h}{2}; Y_{1j} + \frac{hk_{11}}{2}, Y_{2j} + \frac{hk_{12}}{2}, \dots, Y_{mj} + \frac{hk_{1m}}{2}\right) \quad (4.5-4b)$$

for each $i = 1, 2, \dots, m$;

$$k_{3i} = f_i\left(x_j + \frac{h}{2}; Y_{1j} + \frac{hk_{21}}{2}, Y_{2j} + \frac{hk_{22}}{2}, \dots, Y_{mj} + \frac{hk_{2m}}{2}\right) \quad (4.5-4c)$$

for each $i = 1, 2, \dots, m$;

$$k_{4i} = f_i\left(x_j + \frac{h}{2}; Y_{1j} + \frac{hk_{31}}{2}, Y_{2j} + \frac{hk_{32}}{2}, \dots, Y_{mj} + \frac{hk_{3m}}{2}\right) \quad (4.5-4d)$$

for each $i = 1, 2, \dots, m$;

and then

$$Y_{i,j+1} = Y_{ij} + \frac{h}{6} [k_{1i} + 2(k_{2i} + k_{3i}) + k_{4i}] \quad (4.5-4e)$$

for each $i = 1, 2, \dots, m$.

We note that $k_{11}, k_{12}, \dots, k_{1m}$ must all be computed before k_{21} can be calculated. In general, each $k_{s1}, k_{s2}, \dots, k_{sm}$ must be determined before any of the expression k_{s+1i} .

We shall now adopt the technique to extend the RK-GM methods to solve systems of first-order initial-value problems defined by (3.2.1-8). We shall first consider

the extension of the second-order RK-GM method (4.3.1-20h) to deal with the system of equations defined by (3.2.1-8). By using the same notation given earlier in this section, for the second-order RK-GM method (4.3.1-20h) when applied to (3.2.1-8), we obtain the following algorithm.

Let $y_{i,j}$ be the approximation of $z_i(x_j)$ for each $j = 0, 1, \dots, N$ and $i = 1, 2, \dots, m$. Let the initial conditions be given by (4.5-3).

Suppose the values of $Y_{1,j}, Y_{2,j}, \dots, Y_{m,j}$ have been computed, then the values of $Y_{1,j+1}, Y_{2,j+1}, \dots, Y_{m,j+1}$ can be obtained by first calculating

$$k_{1i} = f_i(x_j; Y_{1,j}, Y_{2,j}, \dots, Y_{m,j}) \quad (4.5-5a)$$

for each $i = 1, 2, \dots, m$;

$$k_{2i} = f_i(x_j + h; Y_{1,j} + hk_{11}, Y_{2,j} + hk_{12}, \dots, Y_{m,j} + hk_{1m}) \quad (4.5-4b)$$

for each $i = 1, 2, \dots, m$;

and then

$$Y_{i,j+1} = Y_{i,j} + h\sqrt{k_{1i}k_{2i}} \quad (4.5-5c)$$

for each $i = 1, 2, \dots, m$.

As noted earlier, we have to compute each $k_{s1}, k_{s2}, \dots, k_{sm}$ before any of the expression $k_{s+1,1}, k_{s+1,2}, \dots, k_{s+1,m}$ can be obtained.

Next we consider the extension of the third-order RK-GM method (4.3.2-4) to deal with systems of equations. To illustrate the idea, we shall consider the system of two equations of the form

$$\left. \begin{aligned} y^{(1)} &= f_1(x, y, z) \\ z^{(1)} &= f_2(x, y, z). \end{aligned} \right\} \quad (4.5-6)$$

By the application of (4.3.2-4) to (4.5-6) we therefore obtain

$$Y_{i+1} = Y_i + h \left\{ \sum_{s=1}^3 w_s k_s + w_4 \sqrt{k_1 k_2} + w_5 \sqrt{k_3 k_2} + w_6 \sqrt{k_1 k_3} \right\}, \quad (4.5-7a)$$

and

$$z_{i+1} = z_i + h \left\{ \sum_{s=1}^3 w_s l_s + w_4 \sqrt{l_1 l_2} + w_5 \sqrt{l_3 l_2} + w_6 \sqrt{l_1 l_3} \right\}, \quad (4.5-7b)$$

where

$$\left. \begin{aligned} k_1 &= f_1(x_i, Y_i, z_i) \\ l_1 &= f_2(x_i, Y_i, z_i) \\ k_2 &= f_1(x_i + c_1 h, Y_i + a_{21} h k_1, z_i + a_{21} h l_1) \\ l_2 &= f_2(x_i + c_1 h, Y_i + a_{21} h k_1, z_i + a_{21} h l_1) \\ k_3 &= f_1(x_i + c_1 h, Y_i + a_{31} h k_1 + a_{32} h k_2, z_i + a_{31} h l_1 + a_{32} h l_2) \\ l_3 &= f_2(x_i + c_2 h, Y_i + a_{31} h k_1 + a_{32} h k_2, z_i + a_{31} h l_1 + a_{32} h l_2) \end{aligned} \right\} \quad (4.5-7c)$$

Now (4.5-7a) and (4.5-7b) may be written in vector form as

$$Y_{i+1,2} = Y_{i,2} + h K^T w \quad (4.5-7d)$$

where

$$Y_{i+1,2} = \begin{pmatrix} Y_{i+1} \\ z_{i+1} \end{pmatrix}, \quad Y_{i,2} = \begin{pmatrix} Y_i \\ z_i \end{pmatrix}, \quad w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{pmatrix}$$

and

$$\mathbf{K}_{6 \times 2} = \begin{bmatrix} k_1 & l_1 \\ k_2 & l_2 \\ k_3 & l_3 \\ \sqrt{k_1 k_2} & \sqrt{l_1 l_2} \\ \sqrt{k_3 k_2} & \sqrt{l_3 l_2} \\ \sqrt{k_1 k_3} & \sqrt{l_1 l_3} \end{bmatrix} .$$

Hence we can write (4.5-7c) in vector notation as

$$\left. \begin{aligned} \mathbf{k}_{12} &= \mathbf{f}(x_1, \mathbf{Y}_{12}^T) \\ \mathbf{k}_{22} &= \mathbf{f}(x_1 + c_1 h, \mathbf{Y}_{12}^T + a_{21} h \mathbf{k}_{12}^T) \\ \mathbf{k}_{32} &= \mathbf{f}(x_1 + c_2 h, \mathbf{Y}_{12}^T + a_{31} h \mathbf{k}_{12}^T + a_{32} h \mathbf{k}_{22}^T) . \end{aligned} \right\} \quad (4.5-7e)$$

Now we can easily extend the idea above to a system of m equations as follows. By deduction, we have from (4.5-7d)

$$\mathbf{Y}_{i+1,m} = \mathbf{Y}_{i,m} + h \mathbf{K}^T \mathbf{w} \quad (4.5-8)$$

where

$$\mathbf{Y}_{i+1,m} = \begin{pmatrix} Y_{i+1,1} \\ Y_{i+1,2} \\ \vdots \\ Y_{i+1,m} \end{pmatrix}, \quad \mathbf{Y}_{i,m} = \begin{pmatrix} Y_{i,1} \\ Y_{i,2} \\ \vdots \\ Y_{i,m} \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{pmatrix},$$

$$\mathbf{K}_{6 \times m} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ k_{31} & k_{32} & \dots & k_{3m} \\ \sqrt{k_{11}k_{21}} & \sqrt{k_{12}k_{22}} & \dots & \sqrt{k_{1m}k_{2m}} \\ \sqrt{k_{21}k_{31}} & \sqrt{k_{22}k_{32}} & \dots & \sqrt{k_{2m}k_{3m}} \\ \sqrt{k_{31}k_{11}} & \sqrt{k_{32}k_{12}} & \dots & \sqrt{k_{3m}k_{1m}} \end{bmatrix}$$

and

$$\left. \begin{aligned} \mathbf{k}_{1m} &= \mathbf{f}(x_1, \mathbf{Y}_{1m}^T) \\ \mathbf{k}_{2m} &= \mathbf{f}(x_1 + c_1 h, \mathbf{Y}_{1m}^T + a_{21} h \mathbf{k}_{1m}^T) \\ \mathbf{k}_{3m} &= \mathbf{f}(x_1 + c_2 h, \mathbf{Y}_{1m}^T + a_{31} h \mathbf{k}_{1m}^T + a_{32} h \mathbf{k}_{2m}^T) \end{aligned} \right\} \quad (4.5-9)$$

The remaining RK-GM methods can similarly be extended to ODE systems. However to maintain accurate results, we have to examine the accuracy of every component of the numerical solution \mathbf{Y}_{1m} . If any of the components fail to be sufficiently accurate, the entire numerical solution \mathbf{Y}_{1m} must be recomputed. Since the RK-GM methods involve the computation of square roots, the truncation error introduced may be amplified. We therefore recommend for the method to be suitable only for systems of small orders and perhaps those which involve squared terms. We should also note that the function \mathbf{f} should always maintain the same sign in its interval of evaluation in order for the method to be valid, otherwise there will be an evaluation of the square root of a negative term. Thus some form of error control needs to be incorporated into the algorithm. One possible and easy way of ensuring that the function is always of the same sign is to maintain the function to be on the same side of the x-axis. If there is a change in sign of the function,

then the step size is reduced until the two neighbouring values of the function are of the same sign again.

The convergence theorems and error estimates for systems of ODEs are similar to those for the single equation, except that the bounds are given in terms of vector norms (Gear[1971]).

4.5.1 NUMERICAL RESULTS FOR SYSTEMS

Problem $f_1(x) = -u_1 u_2,$
 $f_2(x) = 1.00.$

Exact solution $u_1(x) = e^{-x^2/2}.$
 $u_2 = x.$

Initial conditions $x = 0, u_1(0) = 1, u_2(0) = 0.$

	x	w(1,j+1)	w(2,j+1)	u(1,x)	absolute error
	.00	.1000000000E+01	.0000000000E+00	.1000000000E+01	
GM	.10	.1000000000E+01	.1000000000E+00	.9950124792E+00	.49875E-02
RK		.9950000000E+00	.1000000000E+00		.12479E-04
GM	.20	.9859287527E+00	.2000000000E+00	.9801986733E+00	.57301E-02
RK		.9801579167E+00	.2000000000E+00		.40757E-04
GM	.30	.9620212510E+00	.3000000000E+00	.9559974818E+00	.60238E-02
RK		.9559153442E+00	.3000000000E+00		.82138E-04
GM	.40	.9291995447E+00	.4000000000E+00	.9231163464E+00	.60832E-02
RK		.9229840606E+00	.4000000000E+00		.13229E-03
GM	.50	.8884840606E+00	.5000000000E+00	.8824969026E+00	.59872E-02
RK		.8823112297E+00	.5000000000E+00		.18567E-03
GM	.60	.8410519914E+00	.6000000000E+00	.8352702114E+00	.57818E-02
RK		.8350340529E+00	.6000000000E+00		.23616E-03
GM	.70	.7882060772E+00	.7000000000E+00	.7827045382E+00	.55015E-02
RK		.7824269076E+00	.7000000000E+00		.27763E-03
GM	.80	.7313240236E+00	.8000000000E+00	.7261490371E+00	.51750E-02
RK		.7258444017E+00	.8000000000E+00		.30464E-03
GM	.90	.6718030534E+00	.9000000000E+00	.6669768109E+00	.48262E-02
RK		.6666638882E+00	.9000000000E+00		.31292E-03
GM	1.0	.6110058050E+00	.1000000000E+01	.6065306597E+00	.44751E-02
RK		.6062308067E+00	.1000000000E+01		.29985E-03

Table (4.5.1a): Comparison of the RK and RK-GM methods (both of second-order) to solve systems of first-order ODEs.

	x	w(1, j+1)	w(2, j+1)	u(1, x)	absolute error
	.00	.1000000000E+01	.0000000000E+00	.1000000000E+01	
GM	.10	.995988587E+00	.1000000000E+00	.995012479E+00	.9761E-03
RK		.995012479E+00	.1000000000E+00		.2602E-10
GM	.20	.981301132E+00	.2000000000E+00	.980198673E+00	.1102E-02
RK		.980198673E+00	.2000000000E+00		.2526E-09
GM	.30	.957150973E+00	.3000000000E+00	.955997482E+00	.1153E-02
RK		.955997481E+00	.3000000000E+00		.8050E-09
GM	.40	.924281083E+00	.4000000000E+00	.923116346E+00	.1165E-02
RK		.923116345E+00	.4000000000E+00		.1581E-08
GM	.50	.883645264E+00	.5000000000E+00	.882496903E+00	.1148E-02
RK		.882496901E+00	.5000000000E+00		.2050E-08
GM	.60	.836381164E+00	.6000000000E+00	.835270211E+00	.1111E-02
RK		.835270210E+00	.6000000000E+00		.1053E-08
GM	.70	.783761749E+00	.7000000000E+00	.782704538E+00	.1057E-02
RK		.782704542E+00	.7000000000E+00		.3326E-08
GM	.80	.727140084E+00	.8000000000E+00	.726149037E+00	.9910E-03
RK		.726149051E+00	.8000000000E+00		.1379E-07
GM	.90	.667892765E+00	.9000000000E+00	.666976811E+00	.9160E-03
RK		.666976845E+00	.9000000000E+00		.3367E-07
GM	1.00	.607365774E+00	.1000000000E+01	.606530660E+00	.8351E-03
RK		.606530726E+00	.1000000000E+01		.6669E-07

Table (4.5.1b): Comparison of the RK and RK-GM methods (both of fourth-order) to solve systems of first-order ODEs for $h = 0.1$.

4.6 CONCLUSIONS AND RECOMMENDATIONS

In this chapter we have derived the second-, third- and fourth-order RK-GM methods. Numerical results show that they are worthy of further investigations. Thus we proceed with the study of the error control strategy using the RK-GM fourth-order method.

From the numerical results obtained, the error control strategy implementation of the AM-GM method appears to be less accurate compared with its RK-Fehlberg counterpart. This is expected because the RK-Fehlberg method is of one order higher. However, the AM-GM method requires only four function evaluations per step in contrast to the six function evaluations per step in the

case of the RK-Fehlberg method. This may have some advantages for equations with complicated functions and large order systems.

The investigation of integrating $y^{(1)} = -y$ for various values of the tolerance using the RK-GM method shows that the method is good for low accuracy only. It could form the basis of an integrating strategy in conjunction with extrapolation methods (Sanugi[1986]).

The use of classical Runge-Kutta and Kutta fourth-order four-stage formulae in an embedded method incorporating error control involves a substantial amount of work in function evaluations. However, the accuracy of the results shows the method to be competitive. This may be due to the fact that the two formulae use different values of the k_i ; hence the increased work.

The numerical results obtained from experiment 5 indicate that the AM-GM method could be more accurate than the Kutta-Merson method. The reason may be that the AM-GM method uses a smaller step-size than the Kutta Merson method. However, this may incur excessive work. Results of problem 8 in this experiment are self-explanatory.

The RK-GM formulae have been investigated as to their suitability for consideration to be included in embedded ordinary differential equation methods. The idea appears to be attractive because of their simplicity and ease of programmability. However the results are not so convincing when compared with the more well known methods. Nevertheless, the investigation is worthwhile as confirmed by the experimental results.

CHAPTER 5

NUMERICAL SOLUTION OF PROBLEMS INVOLVING ODES - GM MULTISTEP METHODS

5.1 INTRODUCTION

Suppose that we are given a p^{th} -order initial-value problem,

$$y^{(p)} = f(x, y, y^{(1)}, \dots, y^{(p-1)}), \quad (5.1-1)$$

with initial conditions $x_0 = \alpha_0$, $y(x_0) = y_0$, $y^{(1)}(x_0) = y_0^{(1)}$,
 $\dots, y^{(p-1)}(x_0) = y_0^{(p-1)}$.

In this chapter we shall seek to obtain an approximation of $y(x_{n+1})$ as a GM combination of the values y_n, \dots, y_{n-k-1} and of the derivatives computed at $y_{n+1}, y_n, \dots, y_{n-k-1}$. Thus,

$$y_{n+1} = \alpha_1 y_n + \alpha_2 y_{n-1} + \dots + \alpha_k y_{n-k-1} + h \left\{ \sum_{i,j=0}^{k-1} \beta_{i,j} [f(y_{n-1}) f(y_{n-j})]^{\frac{1}{2}} \right\}, \quad (5.1-2)$$

for some fixed numbers $\alpha_1, \alpha_2, \dots, \alpha_k, \beta_{0,0}, \beta_{0,1}, \beta_{0,2}, \dots, \beta_{0,k-1}$ and $\beta_{1,1}, \dots, \beta_{k-1,k-1}$.

It may happen, of course, that α_k or $\beta_{i,k-1}$ is zero, but we assume that this is not the case for methods of order k . Under the assumption that k cannot be so reduced, the method given by (5.1-2) is known as a nonlinear k -step method or a GM multistep method.

If all the cross coefficients $\beta_{i,j}$ are equal to zero for $i \neq j$, then (5.1-2) will be reduced to the linear k -step method. On the other hand, if all the coefficients $\beta_{i,j}$ are zero for $i = j$, then (5.1-2) will be reduced to the GM multistep method of the form

$$\begin{aligned}
 Y_{n+1} = & \alpha_1 Y_n + \alpha_2 Y_{n-1} + \dots + \alpha_k Y_{n-k-1} \\
 & + h \sum_{\substack{i,j=0 \\ i \neq j}}^{k-1} \beta_{i,j} [f(Y_{n-i}) f(Y_{n-j})]^{\frac{1}{2}}.
 \end{aligned} \tag{5.1-2a}$$

Note that (5.1-2) may give rise to either an explicit or implicit method depending on the value of $\beta_{0,j}$, $j = 0, 1, \dots, k-1$. If all $\beta_{0,j}$, $j = 0, 1, \dots, k-1$ are zero, then the method is said to be explicit; otherwise it is implicit. For an explicit method y_n can be computed directly from (5.1-2), while in the implicit case, we have to solve an equation of the form

$$y_n - h\beta_{00}f(y_n) = v, \tag{5.1-3}$$

where v is independent of y_n and is given by

$$v = \sum_{i=1}^{k-1} [\alpha_i Y_{n-i} + h\beta_{i,i}f(Y_{n-i})].$$

For a nonlinear function f , (5.1-3) requires the solution of a polynomial equation. If f has a small Lipschitz constant, then (5.1-3) can be solved by fixed-point iteration as the limit of the sequence $z^{(0)}$, $z^{(1)}$, \dots , where, for $j \geq 1$, $z^{(j)}$ is obtained as

$$z^{(j)} = v + h\beta_{00}f(z^{(j-1)}).$$

In the case of stiff problems, where the Lipschitz constant is necessarily large, this iterative method is not convergent and it is necessary to use some variant of the Newton method (Butcher[1987]). Another possible approach to using implicit methods is in the context of predictor-corrector pairs.

5.2 NUMERICAL METHODS FOR FIRST-ORDER ODES

Consider (5.1-1) for the case of $p = 1$, then we have the first-order initial-value problem (4.1-1). Let the general form of the formula which approximates y_{n+1} be defined by

$$Y_{n+1} - Y_n = h \{ \alpha_1 f_n + \alpha_2 f_{n+1} + \alpha_3 \sqrt{f_n f_{n+1}} \}. \tag{5.2-1}$$

The values of α_1 , α_2 and α_3 are to be determined so as to give the highest accuracy possible.

Consider the Taylor series expansion of y_{n+1} about x_n and take the difference $y_{n+1} - y_n$. We obtain

$$Y_{n+1} - Y_n = hf_n + \frac{h^2}{2} f_n^{(1)} + \frac{h^3}{6} f_n^{(2)} + \frac{h^4}{24} f_n^{(3)} + O(h^5). \quad (5.2-2a)$$

$= hf_n' + \frac{h^2}{2} f_n''$

Next we consider the right-hand side of (5.2-1). By the Taylor series expansion of f_{n+1} about x_n , we have

$$f_{n+1} = f_n + hf_n^{(1)} + \frac{h^2}{2} f_n^{(2)} + \frac{h^3}{6} f_n^{(3)} + \frac{h^4}{24} f_n^{(4)} + O(h^5). \quad (5.2-2b)$$

If we multiply (5.2-2b) throughout by f_n and take the square root, we obtain

$$\begin{aligned} \sqrt{f_n f_{n+1}} &= f_n + \frac{h}{2} f_n^{(1)} + \frac{h^2}{8} \left[2f_n^{(2)} - \frac{(f_n^{(1)})^2}{f_n} \right] \\ &\quad + \frac{h^3}{48} \left[4f_n^{(3)} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n} + 3 \frac{(f_n^{(1)})^3}{f_n^2} \right] \\ &\quad + \frac{h^4}{384} \left[8f_n^{(4)} - 16 \frac{f_n^{(1)} f_n^{(3)}}{f_n} - 12 \frac{(f_n^{(2)})^2}{f_n} \right. \\ &\quad \left. + 36 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^2} - 15 \frac{(f_n^{(1)})^4}{f_n^3} \right] + O(h^5). \quad (5.2-2c) \end{aligned}$$

By using (5.2-2b) and (5.2-2c), we obtain the right-hand side of (5.2-1) as

$$\begin{aligned}
& (\alpha_1 + \alpha_2 + \alpha_3)hf_n + (2\alpha_2 + \alpha_3)\frac{h^2}{2}f_n^{(1)} + (2\alpha_2 + \alpha_3)\frac{h^3}{4}f_n^{(2)} \\
& - \alpha_3\frac{h^3}{8}\frac{(f_n^{(1)})^2}{f_n} + \frac{h^4}{48}\left[4(2\alpha_2 + \alpha_3)f_n^{(3)}\right. \\
& \left. - 6\alpha_3\frac{f_n^{(1)}f_n^{(2)}}{f_n} + 3\alpha_3\frac{(f_n^{(1)})^3}{f_n^2}\right] + O(h^5). \quad (5.2-2d)
\end{aligned}$$

By equating the coefficients of like terms in (5.2-2a) and (5.2-2d), the following results are obtained:

$$\left. \begin{aligned}
\text{coefficient of } hf_n: \quad \alpha_1 + \alpha_2 + \alpha_3 &= 1 \\
\text{coefficient of } h^2f_n^{(1)}: \quad 2\alpha_2 + \alpha_3 &= 1 \\
\text{coefficient of } h^3f_n^{(2)}: \quad 2\alpha_2 + \alpha_3 &= \frac{2}{3}
\end{aligned} \right\}. \quad (5.2-2e)$$

Now (5.2-2e) is inconsistent because of the second and third equations. Therefore we solve only two equations in three unknowns. Since the third equation in (5.2-2e) is the coefficient of the term involving h^3 , we may choose to solve the set of two equations, namely the first and second equations of (5.2-2e) in order to maintain the highest accuracy as possible. Hence we have a system of simultaneous equations

$$\left. \begin{aligned}
\alpha_2 + \alpha_3 &= 1 - \alpha_1 \\
2\alpha_2 + \alpha_3 &= 1
\end{aligned} \right\}. \quad (5.2-2f)$$

On solving (5.2-2f) we obtain the solutions as

$$\text{and } \left. \begin{aligned}
\alpha_1 = \alpha_2 = \beta \\
\alpha_3 = 1 - 2\beta
\end{aligned} \right\} \quad (5.2-2g)$$

for some arbitrary constant β .

Therefore the general GM formula can be written as

$$y_{n+1} = y_n + h \left\{ \beta (f_n + f_{n+1}) + (1 - 2\beta) \sqrt{f_n f_{n+1}} \right\} \quad (5.2-3)$$

where β is the parameter defined in (5.2-2g).

Now the well known Trapezoidal formula (AM) can be deduced from (5.2-3) by substituting $\beta = \frac{1}{2}$ and obtain

$$y_{n+1} = y_n + \frac{h}{2} (f_n + f_{n+1}). \quad (5.2-3a)$$

Similarly, the original GM formula can be derived from (5.2-3) by replacing $\beta = 0$ to obtain

$$y_{n+1} = y_n + h \sqrt{f_n f_{n+1}}. \quad (5.2-3b)$$

The formula given by (5.2-3) is accurate up to order $O(h^2)$. Therefore, the local truncation error of (5.2-3) is obtained as

$$T_2^{GM} = \frac{h^3}{4} f_n^{(2)} - \alpha_3 \frac{h^3}{8} \frac{(f_n^{(1)})^2}{f_n} - \frac{h^3}{6} f_n^{(2)},$$

or

$$T_2^{GM} = \frac{h^3}{24} \left\{ 2f_n^{(2)} - 3\alpha_3 \frac{(f_n^{(1)})^2}{f_n} \right\}. \quad (5.2-4)$$

Hence we deduce that, the local truncation error of the AM formula (5.2-3a) is

$$T_2^{AM} = \frac{h^3}{12} f_n^{(2)} \quad (5.2-4a)$$

and that of the original GM formula (5.2-3b) is given by

$$T_2^{GM} = \frac{h^3}{24} \left\{ 2f_n^{(2)} - 3 \frac{(f_n^{(1)})^2}{f_n} \right\}. \quad (5.2-4b)$$

Table(5.2) compares the computational complexity of the original GM formula with the AM formula

Formula	Multiplication	Addition	Square Root
AM	1	2	0
GM	2	1	1
Ratio:AM/GM	0.5	2	0

Table(5.2): Computational complexity of the AM and GM formulae

We observe that the GM formula involves an extra amount of work in the evaluation of a square root; assuming that multiplication and addition ^{will shortly} require an equivalent amount of work. However, if the problems to be solved contain the evaluation of a square of a function, then naturally the GM formula has an advantage over the AM formula.

5.2.1 CONDITIONS UNDER WHICH THE GM FORMULA IS MORE ACCURATE THAN THE AM FORMULA

We now compare the local truncation error of the GM formula (5.2-3) given by (5.2-4) with that of the AM formula (5.2-3a) given by (5.2-4a). We also note that if $\alpha_3 = 0$, then (5.2-3) reduces to (5.2-3a). Therefore, we shall discuss the error given by (5.2-4) for non-zero values of α_3 and observe whether it is less or greater than (5.2-4a).

We have from (5.2-4), the local truncation error of the GM formula as

$$T_2^{GM} = \frac{h^3}{24} \left\{ 2f_n^{(2)} - 3\alpha_3 \frac{(f_n^{(1)})^2}{f_n} \right\}. \quad (5.2.1-1)$$

For the GM formula to be better than the AM formula (5.2-3a), we need $|T_2^{GM}| < |T_2^{AM}|$. Therefore, we have

$$\left| \frac{h^3}{24} \left\{ 2f_n^{(2)} - 3\alpha_3 \frac{[f_n^{(1)}]^2}{f_n} \right\} \right| < \left| \frac{h^3}{12} f_n^{(2)} \right|. \quad (5.2.1-2)$$

If we assume that $f_n^{(2)} > 0$, then the inequality (5.2.1-2) becomes

$$-\frac{h^3}{12} f_n^{(2)} < \frac{h^3}{12} f_n^{(2)} - \frac{\alpha_3 h^3}{8} \frac{(f_n^{(1)})^2}{f_n} < \frac{h^3}{12} f_n^{(2)},$$

or

$$-\frac{h^3}{6} f_n^{(2)} < -\frac{\alpha_3 h^3}{8} \frac{(f_n^{(1)})^2}{f_n} < 0. \quad (5.2.1-3)$$

Let $\alpha_3 > 0$, then from the inequality (5.2.1-3), we obtain

$$f_n > 0 \text{ and } f_n f_n^{(2)} > \frac{3\alpha_3}{4} (f_n^{(1)})^2. \quad (5.2.1-4)$$

Next, if $f_n^{(2)} < 0$ and let $-f_n^{(2)} = G > 0$, then inequality (5.2.1-2) becomes,

$$-\frac{h^3}{12} G < \frac{h^3}{a} G - \frac{\alpha_3 h^3}{a} \frac{(f_n^{(1)})^2}{f_n} < \frac{h^3}{12} G,$$

or

$$0 < -\frac{\alpha_3 h^3}{8} \frac{(f_n^{(1)})^2}{f_n} < \frac{h^3}{6} G. \quad (5.2.1-5)$$

Since we have assumed that $\alpha_3 > 0$, therefore from the left-hand side of inequality (5.2.1-5), we obtain $f_n < 0$.

Let $-f_n = H > 0$, then the right-hand side of inequality (5.2.1-5) becomes,

$$\frac{\alpha_3 h^3}{8} \frac{(f_n^{(1)})^2}{H} < \frac{h^3}{6} G,$$

or

$$GH > \frac{3\alpha_3}{4} (f_n^{(1)})^2,$$

i.e.

$$f_n f_n^{(2)} > \frac{3\alpha_3}{4} (f_n^{(1)})^2, \quad (5.2.1-6)$$

which is similar to the result obtained in the case of $f_n^{(2)} > 0$.

Therefore, the necessary and sufficient condition for the GM formula (5.2-3) to be more accurate than the AM formula (5.2-3a) is that

$$f_n f_n^{(2)} > \frac{3\alpha_3}{4} (f_n^{(1)})^2, \quad (5.2.1-7)$$

for $\alpha_3 > 0$.

Similarly, by following the same lines of argument as above, we can establish the condition under which the GM formula (5.2-3) is better than the AM formula (5.2-3a) for the case of $\alpha_3 < 0$ as

$$f_n f_n^{(2)} < \frac{3\alpha_3}{4} (f_n^{(1)})^2. \quad (5.2.1-8)$$

5.2.2 STABILITY ANALYSIS OF (5.2-3)

We shall now consider the stability regions of the class of methods defined by (5.2-3), namely

$$Y_{n+1} = Y_n + h \left\{ \beta(f_n + f_{n+1}) + (1 - 2\beta) \sqrt{f_n f_{n+1}} \right\}.$$

By applying formula (5.2-3) to the test problem $y^{(1)} = \lambda y$, we obtain

$$Y_{n+1} = Y_n + h\lambda \left\{ \beta(Y_n + Y_{n+1}) + (1 - 2\beta) \sqrt{Y_n Y_{n+1}} \right\}. \quad (5.2.2-1)$$

Now dividing (5.2.2-1) throughout by y_n , gives the result

$$\frac{Y_{n+1}}{Y_n} = 1 + h\lambda \left\{ \beta \left(1 + \frac{Y_{n+1}}{Y_n} \right) + (1 - 2\beta) \left\{ \frac{Y_{n+1}}{Y_n} \right\}^{1/2} \right\}. \quad (5.2.2-1a)$$

Now substitute $\frac{Y_{n+1}}{Y_n} = P_n^2$ in (5.2.2-1a), and after some rearrangement, we have

$$P_n^2(1 - h\lambda\beta) + h\lambda(2\beta - 1)P_n - (1 + h\lambda\beta) = 0. \quad (5.2.2-1b)$$

Now (5.2.2-1b) is a quadratic in P_n which upon solving for P_n , gives

$$P_n = \frac{h\lambda(1-2\beta) \pm \{ [h\lambda(1-2\beta)]^2 + 4(1-h\lambda\beta)(1+h\lambda\beta) \}^{1/2}}{2(1-h\lambda\beta)} \quad (5.2.2-2a)$$

Absolute stability requires that

$$\left| \frac{Y_{n+1}}{Y_n} \right| = |P_n| < 1$$

or, similarly,

$$|P_n| < 1.$$

Next consider the roots of the quadratic equation (5.2.2-1b) given in (5.2.2-2a). We shall consider the root

$$P_n = \frac{h\lambda(1-2\beta) + \{ [h\lambda(1-2\beta)]^2 + 4(1-h\lambda\beta)(1+h\lambda\beta) \}^{1/2}}{2(1-h\lambda\beta)} \quad (5.2.2-2b)$$

and discuss the condition under which $|P_n| < 1$.

Now $|P_n| < 1$ implies that

$$\begin{aligned} \hat{F}(h\lambda; \beta) &= |h\lambda(1-2\beta) + \{ [h\lambda(1-2\beta)]^2 + 4(1-h\lambda\beta)(1+h\lambda\beta) \}^{1/2}|, \\ &< 2|1-h\lambda\beta|. \end{aligned}$$

Let $h\lambda = z$, then we have

$$\begin{aligned} \hat{F}(z; \beta) &= |z(1-2\beta) + \{ [z(1-2\beta)]^2 + 4(1-z\beta)(1+z\beta) \}^{1/2}|, \\ &< 2|1-z\beta|. \end{aligned} \quad (5.2.2-3)$$

Assume $\beta > 0$, we have two cases to be considered, ^{sufficient to determine the stability region.} They are

- (a) z is real
- (b) z is purely imaginary.

Consider case (a), we have

$$\begin{aligned}\hat{f}(z;\beta) &= z(1-2\beta) + \{ [z(1-2\beta)]^2 + 4(1-z\beta)(1+z\beta) \}^{1/2}, \\ &< 2(1 - z\beta).\end{aligned}\tag{5.2.2-3a}$$

or on rearranging the terms in the inequality (5.2.2-3a), we obtain for real z ,

$$\begin{aligned}\hat{g}(z;\beta) &= f(z;\beta) + 2z\beta, \\ &= z + \{ [z(1-2\beta)]^2 + 4(1-z\beta)(1+z\beta) \}^{1/2} < 2.\end{aligned}$$

If we plot the function $g(z;\beta)$ against z , we notice that $|\hat{g}(z;\beta)| < 2$ for all $z < 0$ and $z > \frac{1}{\beta}$, for $\beta > 0$ and $|\hat{g}(z;\beta)| > 2$ for $0 < z < \frac{1}{\beta}$ for $\beta > 0$.

Next consider case(b). Let $z = iy$ where y is real. Then from inequality (5.2.2-3) we obtain

$$\begin{aligned}\text{L.H.S.} &= \left| iy(1-2\beta) + \{ -[y(1-2\beta)]^2 + 4(1+(y\beta))^2 \}^{1/2} \right|, \\ &= \{ [y(1-2\beta)]^2 + 4(1+(y\beta))^2 + [-[(1-2\beta)y]^2] \}^{1/2}, \\ &= 2\sqrt{1 + (y\beta)^2}\end{aligned}$$

$$\text{R.H.S.} = 2 |1 - \beta iy| = 2\sqrt{1 + (\beta y)^2}.$$

Hence (5.2.2-3) is an equality statement. This suggests that the imaginary axis of the complex plane is the boundary for the absolute stability region of the method, irrespective of the value of the parameter β . The method is therefore absolutely stable for $h\lambda$ lying on the left half of the complex plane.

5.3 NUMERICAL METHODS FOR A SPECIAL CLASS OF SECOND-ORDER ODES

By using the approach described in Section 4.5, it is clearly possible to express the second-order differential equation

$$y^{(2)} = f(x, y, y^{(1)}),\tag{5.3-1a}$$

in the form of a first order system of ODEs, that is $v^{(1)} = f(x, u, v)$, where $u = y$ and $v = y^{(1)}$. If, however, (5.3-1a) has the special form

$$y^{(2)} = f(x, y), \quad (5.3-1b)$$

that is, the function f is independent of $y^{(1)}$, then it is natural to enquire whether there exist direct methods which do not require the explicit introduction of the first derivative into an equation in which it does not already appear. We might ask the same sort of question about special higher order ODEs of the form $y^{(m)} = f(x, y)$. In section 5.3.1 we shall derive the GM method to deal with equations of the form (5.3-1b); while in section 5.4 we shall consider the special class of fourth-order ODEs of the form $y^{(4)} = f(x, y)$.

5.3.1 DERIVATION OF THE GM METHOD FOR PROBLEMS OF THE TYPE $y^{(2)} = f(x, y)$

Consider (5.1-1a) for the case of $p = 2$ and that the function f is independent of $y^{(1)}$. Then, we have a special class of second-order, initial-value problems defined by the form (5.3-1b) with initial conditions $y(x_0) = y_0$ and $y^{(1)}(x_0) = y_0^{(1)}$.

Let the general form of the formula which approximates y_{n+1} be defined as

$$\begin{aligned} y_{n+1} - 2y_n + y_{n-1} &= h^2 \left\{ \alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n+1} \right. \\ &\quad \left. + \alpha_4 \sqrt{f_n f_{n-1}} + \alpha_5 \sqrt{f_n f_{n+1}} + \alpha_6 \sqrt{f_{n-1} f_{n+1}} \right\}. \end{aligned} \quad (5.3.1-1)$$

Now consider the expression for the right-hand side of (5.3.1-1). By using the REDUCE program for algebraic manipulation we obtain the following results:

$$\alpha_2 f_{n-1} = \alpha_2 f_n \left\{ 1 - h \frac{f_n^{(1)}}{f_n} + \frac{h^2}{2!} \frac{f_n^{(2)}}{f_n} - \frac{h^3}{3!} \frac{f_n^{(3)}}{f_n} + \frac{h^4}{4!} \frac{f_n^{(4)}}{f_n} - \frac{h^5}{5!} \frac{f_n^{(5)}}{f_n} \right\} + O(h^6). \quad (5.3.1-2a)$$

$$\alpha_3 f_{n+1} = \alpha_3 f_n \left\{ 1 + h \frac{f_n^{(1)}}{f_n} + \frac{h^2}{2!} \frac{f_n^{(2)}}{f_n} + \frac{h^3}{3!} \frac{f_n^{(3)}}{f_n} + \frac{h^4}{4!} \frac{f_n^{(4)}}{f_n} + \frac{h^5}{5!} \frac{f_n^{(5)}}{f_n} \right\} + O(h^6). \quad (5.3.1-2b)$$

$$\begin{aligned} \alpha_4 \sqrt{f_n f_{n-1}} = \alpha_4 f_n \left\{ 1 - \frac{h}{2} \frac{f_n^{(1)}}{f_n} + \frac{h^2}{8} \left[2 \frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\ - \frac{h^3}{48} \left[4 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \\ + \frac{h^4}{384} \left[8 \frac{f_n^{(4)}}{f_n} - 16 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} - 12 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\ \left. + 36 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 15 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\ - \frac{h^5}{3840} \left[16 \frac{f_n^{(5)}}{f_n} - 40 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 80 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\ \left. + 180 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 120 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \right. \\ \left. - 300 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 105 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] \left. \right\} \\ + O(h^6). \quad (5.3.1-2c) \end{aligned}$$

$$\begin{aligned}
\alpha_5 \sqrt{f_n f_{n+1}} &= \alpha_5 f_n \left\{ 1 + \frac{h}{2} \frac{f_n^{(1)}}{f_n} + \frac{h^2}{8} \left[2 \frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
&+ \frac{h^3}{48} \left[4 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \\
&+ \frac{h^4}{384} \left[8 \frac{f_n^{(4)}}{f_n} - 16 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} - 12 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\
&\quad \left. + 36 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 15 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\
&+ \frac{h^5}{3840} \left[16 \frac{f_n^{(5)}}{f_n} - 40 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 80 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
&\quad + 180 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 120 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \\
&\quad \left. - 300 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 105 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] \left. \right\} + o(h^6).
\end{aligned}$$

(5.3.1-2d)

$$\begin{aligned}
\alpha_6 \sqrt{f_{n-1} f_{n+1}} &= \alpha_6 f_n \left\{ 1 + \frac{h^2}{2} \left[\frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
&+ \frac{h^4}{24} \left[\frac{f_n^{(4)}}{f_n} - 4 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} + 6 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} \right. \\
&\quad \left. \left. - 3 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \right\} + o(h^6).
\end{aligned}$$

(5.3.1-2e)

Now substituting (5.3.1-2a), (5.3.1-2b), (5.3.1-2c), (5.3.1-2d) and (5.3.1-2e) into the right-hand side of (5.3.1-1), we obtain

$$\begin{aligned}
& h^2 f_n \left\{ \sum_{i=1}^6 \alpha_i + (-2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5) \frac{h f_n^{(1)}}{2 f_n} \right. \\
& \quad + (2\alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_5 + 2\alpha_6) \frac{h^2 f_n^{(2)}}{4 f_n} \\
& \quad + (-2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5) \frac{h^3 f_n^{(3)}}{12 f_n} \\
& \quad + (-\alpha_4 - \alpha_5 - 4\alpha_6) \frac{h^2}{8} \left[\frac{f_n^{(1)}}{f_n} \right]^2 \\
& \quad \left. + (\alpha_4 - \alpha_5) \frac{h^3}{8} \frac{f_n^{(1)} f_n^{(2)}}{f_n} + (-\alpha_4 + \alpha_5) \frac{h^3}{16} \left[\frac{f_n^{(1)}}{f_n} \right]^3 \right\} + O(h^6).
\end{aligned} \tag{5.3.1-3}$$

Next, we have the left-hand side of (5.3.1-1) given by

$$h^2 f_n + \frac{1}{12} h^4 f_n^{(2)} + \frac{1}{360} h^6 f_n^{(4)} + O(h^8), \tag{5.3.1-4}$$

Therefore, by equating (5.3.1-3) and (5.3.1-4) we obtain the following results:

$$\text{coeff. of } h^2 f_n: \quad \sum_{i=1}^6 \alpha_i = 1, \tag{5.3.1-5a}$$

$$\text{coeff. of } h^3 f_n^{(1)}: \quad -2\alpha_2 + 2\alpha_3 - \alpha_4 + \alpha_5 = 0 \tag{5.3.1-5b}$$

$$\text{coeff. of } h^4 f_n^{(2)}: \quad 2\alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_5 + 2\alpha_6 = \frac{1}{3}, \tag{5.3.1-5c}$$

$$\text{coeff. of } h^4 \frac{[f_n^{(1)}]^2}{f_n}: \quad \alpha_4 + \alpha_5 + 4\alpha_6 = 0, \tag{5.3.1-5d}$$

$$\text{coeff. of } h^5 \frac{f_n^{(1)} f_n^{(2)}}{f_n}: \quad \alpha_4 - \alpha_5 = 0. \tag{5.3.1-5e}$$

Now for (5.3.1-1) to be symmetric, we impose the condition

$$\alpha_2 - \alpha_3 = 0. \tag{5.3.1-6}$$

Thus, we have a system of simultaneous equation of the form

$$A\bar{\alpha} = \mathbf{b}, \quad (5.3.1-7) \quad \times$$

to be solved, where A is a 6 by 6 matrix,

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -2 & 2 & -1 & 1 & 0 \\ 0 & 2 & 2 & 1 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 4 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix},$$

$$\bar{\alpha}^T = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) \text{ and } \mathbf{b}^T = (1, 0, \frac{1}{3}, 0, 0, 0).$$

By using the REDUCE program, we solve the system of equations (5.3.1-7) to obtain the solutions as

$$\left. \begin{aligned} \alpha_6 &= \alpha, \quad \alpha_4 = \alpha_5 = -2\alpha \\ \alpha_2 &= \alpha_3 = \frac{1}{12} (1 + 6\alpha) \\ \alpha_1 &= \frac{1}{6} (5 + 12\alpha) \end{aligned} \right\} \quad (5.3.1-8)$$

where α is an arbitrary constant.

By substituting (5.3.1-8) into (5.3.1-1), we obtain the general form of the GM formula as

$$\begin{aligned} &Y_{n-1} - 2Y_n + Y_{n+1} \\ &= \frac{h^2}{12} \{ 2(5 + 12\alpha)f_n + (1 + 6\alpha)(f_{n-1} + f_{n+1}) \\ &\quad - 12\alpha [2(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}) - \sqrt{f_{n-1} f_{n+1}}] \}, \quad (5.3.1-9a) \end{aligned}$$

where the arbitrary parameter α is given in (5.3.1-8).

We may deduce the Numerov (AM) method from (5.3.1-9a) by substituting $\alpha = 0$ to give

$$Y_{n-1} = 2Y_n - Y_{n+1} + \frac{h^2}{12} \{ f_{n-1} + 10f_n + f_{n+1} \}, \quad (5.3.1-9b)$$

and the GM formula is obtained by substituting $\alpha = -\frac{1}{6}$ to obtain

$$Y_{n-1} = 2Y_n - Y_{n+1} + \frac{h^2}{6} \left\{ 3f_n + 2 \left[\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}} \right] - \sqrt{f_{n-1} f_{n+1}} \right\}. \quad (5.3.1-9c)$$

Another form of the GM formula may be obtained by substituting $\alpha = -\frac{5}{12}$ in (5.3.1-9a) to give

$$Y_{n-1} = 2Y_n - Y_{n+1} + \frac{h^2}{2} \left\{ 10 \left[2 \left(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}} \right) - \sqrt{f_{n-1} f_{n+1}} \right] - 3 (f_{n-1} + f_{n+1}) \right\}. \quad (5.3.1-9d)$$

5.3.2 ERROR ANALYSIS OF (5.3-9a)

Next, we consider the derivation of the formula (5.3.1-9a) to decide on the magnitude of its local truncation error. We observe that (5.3.1-9a) is accurate to order $O(h^4)$. Therefore, the local truncation error of (5.3.1-9a) can be obtained from (5.3.1-2a), (5.3.1-2b), (5.3.1-2c), (5.3.1-2d), (5.3.1-2e) and (5.3.1-4) as

$$\begin{aligned} T_{n+1}^{GM} = & h^6 \left\{ \frac{1}{720} \left[30(\alpha_2 + \alpha_3 + \alpha_6) + 15(\alpha_4 + \alpha_5) - 2 \right] f_n^{(4)} \right. \\ & - \frac{1}{24} \left[\alpha_4 + \alpha_5 + 4\alpha_6 \right] \frac{f_n^{(1)} f_n^{(3)}}{f_n} - \frac{1}{32} \left[\alpha_4 + \alpha_5 \right] \frac{[f_n^{(2)}]^2}{f_n} \\ & + \frac{1}{32} \left[3(\alpha_4 + \alpha_5) + 8\alpha_6 \right] \frac{f_n^{(2)} [f_n^{(1)}]^2}{f_n^2} \\ & \left. - \frac{1}{128} \left[5(\alpha_4 + \alpha_5) + 16\alpha_6 \right] \frac{[f_n^{(1)}]^4}{f_n^3} \right\}, \end{aligned}$$

$$\begin{aligned}
&= \frac{h^6}{5760} \left\{ 8 \left[30(\alpha_2 + \alpha_3 + \alpha_6) + 15(\alpha_4 + \alpha_5) - 2 \right] f_n^{(4)} \right. \\
&- 240 \left[\alpha_4 + \alpha_5 + 4\alpha_6 \right] \frac{f_n^{(1)} f_n^{(3)}}{f_n} - 180 \left[\alpha_4 + \alpha_5 \right] \frac{[f_n^{(2)}]^2}{f_n} \\
&+ 180 \left[3(\alpha_4 + \alpha_5) + 8\alpha_6 \right] \frac{f_n^{(2)} [f_n^{(1)}]^2}{f_n^2} \\
&\left. - 45 \left[5(\alpha_4 + \alpha_5) + 16\alpha_6 \right] \frac{[f_n^{(1)}]^4}{f_n^3} \right\}. \quad (5.3.2-1)
\end{aligned}$$

By substituting the values of α_i ; $i = 1, 2, \dots, 6$ given in (5.3.1-8) into (5.3.2-1), we obtain the local truncation error of (5.3.1-9a) as

$$\begin{aligned}
T_{n+1}^{GM} &= \frac{h^6}{5760} \left\{ 8 \left[5(1 + 12\alpha) - 2(1 + 30\alpha) \right] f_n^{(4)} \right. \\
&+ 180\alpha \left[4 \frac{[f_n^{(2)}]^2}{f_n} - 4 \frac{f_n^{(2)} [f_n^{(1)}]^2}{f_n^2} + \frac{[f_n^{(1)}]^4}{f_n^3} \right] \left. \right\}, \\
&= \frac{h^6}{480} \left\{ 2f_n^{(4)} + 15\alpha \left[4 \frac{[f_n^{(2)}]^2}{f_n} - 4 \frac{f_n^{(2)} [f_n^{(1)}]^2}{f_n^2} + \frac{[f_n^{(1)}]^4}{f_n^3} \right] \right\}.
\end{aligned}$$

Define

$$M_n = \frac{\left[(f_n^{(1)})^2 - 2f_n f_n^{(2)} \right]^2}{f_n^3}, \quad (5.3.2-2)$$

then we can write

$$T_{n+1}^{GM} = \frac{h^6}{480} \left\{ 2f_n^{(4)} + 15\alpha M_n \right\}, \quad (5.3.2-3a)$$

where α is the parameter of the formula.

For $\alpha = 0$, we obtain the local truncation error of the AM formula and is given as

$$T_{n+1}^{AM} = \frac{h^6}{240} f_n^{(4)}. \quad (5.3.2-3b)$$

Next we seek to find the condition under which the GM method results in a local truncation error smaller in modulus than the AM method. By comparing (5.3.2-3a) and (5.3.2-3b), we observe that assuming $\alpha > 0$, then $T_{n+1}^{GM} < T_{n+1}^{AM}$ provided $M_n < 0$.

Therefore by definition of M_n , it follows that

$$M_n = \frac{[(f_n^{(1)})^2 - 2f_n f_n^{(2)}]^2}{f_n^3} < 0. \quad (5.3.2-4a)$$

Since the numerator is always positive, we therefore have the condition that f_n is negative for (5.3.2-4a) to be true. Thus we claim that for f negative in the interval of the integration and $\alpha > 0$, then $|T_{n+1}^{GM}| < |T_n^{AM}|$.

The condition of $T_{n+1}^{GM} < T_{n+1}^{AM}$ which requires that $\alpha M_n < 0$, for the case $\alpha < 0$, implies that $M_n > 0$. Hence we have

$$M_n = \frac{[(f_n^{(1)})^2 - 2f_n f_n^{(2)}]^2}{f_n^3} > 0. \quad (5.3.2-4b)$$

Since the numerator is always positive, therefore f_n^3 must be always positive in the interval of integration. Thus f_n must be positive throughout the interval of integration and for $\alpha < 0$, the GM formula has a smaller modulus value of local truncation error than the AM counterpart.

Therefore there are possible functions f for which the local truncation error of the GM formula is smaller in modulus than the error introduced by the AM formula.

Now consider two particular cases of the GM formula given in (5.3.1-9c) and (5.3.1-9d). In the first case, we have $\alpha = -\frac{1}{6}$. Therefore its local truncation error is given as

$$T_{n+1}^{GM1} = \frac{h^6}{960} \{ 4f_n^{(4)} - 5M_n \}. \quad (5.3.2-5a)$$

In the second case, we have $\alpha = -\frac{5}{12}$. Hence its local truncation error is obtained as

$$T_{n+1}^{GM2} = \frac{h^6}{1920} \{ 8f_n^{(4)} - 25M_n \}. \quad (5.3.2-5b)$$

Table(5.3.2) compares the three formulae, namely the AM formula (5.3.1-9b) and the two GM formulae (5.3.1-9c) and (5.3.1-9d). The GM formulae are found to involve two times more work than the AM formula as indicated in Table(5.3.2). However they are all $O(h^4)$ accuracy as confirmed by the numerical results given in section(5.3.3)

Formula	Addition	Multiplication	Square Root	Local Truncation Error
AM Formula (5.3.1-9b)	4	3	0	$\frac{h^6}{240} f_n$
GM Formula (5.3.1-9c)	5	6	4	$\frac{h^6}{240} f_n - \frac{h^6}{192} M_n$
GM Formula (5.3.1-9d)	6	8	3	$\frac{h^6}{240} f_n - \frac{5h^6}{384} M_n$

Table(5.3.2): Computational complexity of the AM and GM formulae

5.3.3 NUMERICAL RESULTS

Problem 1 $y^{(2)} + xy = 0$.

Initial conditions $x_0 = 0, y_0 = 1, y_0^{(1)} = 2$.

Exact solution $y = (1 - x^3/3 + x^6/180 - \dots)$
 $+ 2(x - x^4/12 + x^7/504 - \dots)$

x_n	Numerical Solution	Exact Solution	Relative Error
.10	.119965000595E+01	.119965000595E+01	0
.20	.139806708690E+01 (A) .139808941867E+01 (B) .139812291632E+01 (C)	.139706707302E+01	.715795183538E-03 .731779934094E-03 .755757059928E-03
.30	.159365496416E+01 (A) .159369988491E+01 (B) .159376726590E+01 (C)	.158965491786E+01	.251629850849E-02 .254455669036E-02 .258694387026E-02
.40	.178442927854E+01 (A) .178449670465E+01 (B) .178459784349E+01 (C)	.177442925714E+01	.563562698202E-02 .567362574206E-02 .573062369611E-02
.50	.196803408010E+01 (A) .196812375478E+01 (B) .196825826628E+01 (C)	.194803447421E+01	.102665564462E-01 .103125898631E-01 .103816397197E-01
.60	.214176811045E+01 (A) .214187958857E+01 (B) .214204680503E+01 (C)	.210677028571E+01	.166120744014E-01 .166649886290E-01 .167443596286E-01
.70	.230262257646E+01 (A) .230275518922E+01 (B) .230295410745E+01 (C)	.224663040833E+01	.249227322500E-01 .249817596535E-01 .250703003513E-01
.80	.244733187537E+01 (A) .244748469339E+01 (B) .244771391933E+01 (C)	.236335522540E+01	.355328090634E-01 .355974705342E-01 .356944622725E-01
.90	.257243840106E+01 (A) .257261020039E+01 (B) .257286789810E+01 (C)	.245250045357E+01	.489043528245E-01 .489744035077E-01 .490794790066E-01
1.00	.267437213004E+01 (A) .267456136319E+01 (B) .267484521144E+01 (C)	.250952380952E+01	.656890840780E-01 .657644900745E-01 .658775984861E-01

Table (5.3.3a)

Problem 2 $y^{(2)} + 2x^2y = 0$

Initial conditions $x_0 = 0, y_0 = 1, y_0^{(1)} = 1$

Exact solution $y = (1 - x^4/6 + x^8/168 - \dots)$
 $+ (-x^5/10 + x^9/360 - \dots)$

x_n	Numerical Solution	Exact Solution	Relative Error
.10	.109998233340E+01	.109998233340E+01	0
.20	.119970133876E+01 (A) .119970148371E+01 (B) .119970170113E+01 (C)	.119970134999E+01	.936544166655E-08 .111454780902E-06 .292685114570E-06
.30	.129840729714E+01 (A) .129840771400E+01 (B) .129840833929E+01 (C)	.129840744521E+01	.114042419673E-06 .207014078941E-06 .688599808733E-06
.40	.139471310095E+01 (A) .139471390026E+01 (B) .139471509923E+01 (C)	.139471396246E+01	.617701420252E-06 .446004673666E-07 .815054255502E-06
.50	.148648365085E+01 (A) .148648492608E+01 (B) .148648683894E+01 (C)	.148648701017E+01	.225990528824E-05 .140202151487E-05 .115188875393E-06
.60	.157074164286E+01 (A) .157074346792E+01 (B) .157074620553E+01 (C)	.157075197074E+01	.657512117010E-05 .541321910915E-05 .367035407630E-05
.70	.164360423379E+01 (A) .164360665899E+01 (B) .164361029682E+01 (C)	.164363156960E+01	.166313519292E-04 .151558381446E-04 .129425494473E-04
.80	.170027071388E+01 (A) .170027376092E+01 (B) .170027833152E+01 (C)	.170033680417E+01	.388689400721E-04 .370769185203E-04 .343888611584E-04
.90	.173508683101E+01 (A) .173509048788E+01 (B) .173509597325E+01 (C)	.173523947285E+01	.879658605241E-04 .858584445949E-04 .826972879105E-04
1.00	.174171581559E+01 (A) .174172003318E+01 (B) .174172635964E+01 (C)	.174206349206E+01	.199577383509E-03 .197156351997E-03 .193524763655E-03

Table (5.3.3b)

Problem 3 $y^{(2)} + x^2y = 1 + x + x^2$

Initial conditions $x_0 = 0, y_0 = 2, Y_0^{(1)} = 2$

Exact solution $y = 2(1 - x^4/12 + x^8/672 - \dots)$
 $+ 2(x - x^5/20 + x^9/1440 - \dots)$

x_n	Numerical Solution	Exact Solution	Relative Error
.10	.220515731629E+01	.220515731629E+01	0
.20	.242116719231E+01 (A)	.242116688706E+01	.126074661035E-06
	.242116695965E+01 (B)		.299825750827E-07
	.242116661067E+01 (C)		.114155553755E-06
.30	.264857051467E+01 (A)	.264856910711E+01	.531441912139E-06
	.264856965700E+01 (B)		.207614850185E-06
	.264856837048E+01 (C)		.278125189599E-06
.40	.288743990707E+01 (A)	.288743590441E+01	.138623499026E-05
	.288743775492E+01 (B)		.640885393397E-06
	.288743452671E+01 (C)		.477135192944E-06
.50	.313722608903E+01 (A)	.313721710689E+01	.286309062579E-05
	.313722141120E+01 (B)		.137201393477E-05
	.313721439451E+01 (C)		.864583432560E-06
.60	.339659195430E+01 (A)	.339657430537E+01	.519609809508E-05
	.339658206413E+01 (B)		.228428857071E-05
	.339656722913E+01 (C)		.208334600216E-05
.70	.366323827217E+01 (A)	.366320554629E+01	.893367162527E-05
	.366321433013E+01 (B)		.239785523519E-05
	.366317841905E+01 (C)		.740533036675E-05
.80	.393372811772E+01 (A)	.393366680462E+01	.155867549444E-04
	.393380252836E+01 (B)		.345031097912E-04
	.393391394862E+01 (C)		.628278940337E-04
.90	.420332107723E+01 (A)	.420319930665E+01	.289709273196E-04
	.420351786622E+01 (B)		.757897843078E-04
	.420381214595E+01 (C)		.145803056907E-03
1.00	.446583286944E+01 (A)	.446557539683E+01	.576572081016E-04
	.446617585261E+01 (B)		.134463249736E-03
	.446668869506E+01 (C)		.249306781711E-03

Table (5.3.3c)

Problem 4 $y^{(2)} - y = 0$.

Initial conditions $x_0 = 0, y_0 = 1, y_0^{(1)} = -1$.

Exact solution $y = e^{-x}$.

x_n	Numerical Solution	Exact Solution	Relative Error
.10	.904837418036E+00	.904837418036E+00	0
.20	.818723210252E+00 (A)	.818730753078E+00	.921282811814E-05
	.818723204446E+00 (B)		.921991941253E-05
	.818723195738E+00 (C)		.923055635404E-05
.30	.740796234571E+00 (A)	.740818220682E+00	.296781450339E-04
	.740796217646E+00 (B)		.297009915767E-04
	.740796192258E+00 (C)		.297352613983E-04
.40	.670277221236E+00 (A)	.670320046036E+00	.638870942600E-04
	.670277188265E+00 (B)		.639362813079E-04
	.670277138808E+00 (C)		.640100619134E-04
.50	.606460980113E+00 (A)	.606530659713E+00	.114882238736E-03
	.606460926459E+00 (B)		.114970698783E-03
	.606460845978E+00 (C)		.115103388947E-03
.60	.548709348791E+00 (A)	.548811636094E+00	.186379618579E-03
	.548709270018E+00 (B)		.186523151717E-03
	.548709151859E+00 (C)		.186738451633E-03
.70	.496444810957E+00 (A)	.496585303791E+00	.282917826556E-03
	.496444702746E+00 (B)		.283135736194E-03
	.496444540430E+00 (C)		.283462601060E-03
.80	.449144721232E+00 (A)	.449328964117E+00	.410040081691E-03
	.449144579303E+00 (B)		.410355951213E-03
	.449144366408E+00 (C)		.410829756226E-03
.90	.406336078720E+00 (A)	.406569659741E+00	.574516605226E-03
	.406335898755E+00 (B)		.574959246825E-03
	.406335628808E+00 (C)		.575623210446E-03
1.00	.367590796995E+00 (A)	.367879441171E+00	.784616219919E-03
	.367590574570E+00 (B)		.785220833784E-03
	.367590240932E+00 (C)		.786127756540E-03

Table (5.3.3d)

Problem 5 $y^{(2)} - y\{[(Q + Bx)/x]^2 - Q/x^2\} = 0.$

Initial conditions $x_0 = 1, y_0 = 10e, y_0^{(1)} = 10e(Q + B).$

Exact solution $y = Cx^Q e^{Bx}.$

Set $B = 1, C = 10, Q = 3/2$

x_n	Numerical Solution	Exact Solution	Relative Error
1.10	.346587549802E+02	.346587549802E+02	0
1.20	.436390152385E+02 (A) .436389911205E+02 (B) .436389549437E+02 (C)	.436440703713E+02	.115826337471E-03 .116378942430E-03 .117207849867E-03
1.30	.543715401009E+02 (A) .543714673415E+02 (B) .543713582034E+02 (C)	.543873445416E+02	.290590410221E-03 .291928209623E-03 .293934891700E-03
1.40	.671413377291E+02 (A) .671411902473E+02 (B) .671409690280E+02 (C)	.671744823128E+02	.493410333302E-03 .495605835950E-03 .498899041178E-03
1.50	.822756510828E+02 (A) .822754002360E+02 (B) .822750239731E+02 (C)	.823338855610E+02	.707296611036E-03 .710343313008E-03 .714913276563E-03
1.60	.100149815184E+03 (A) .100149428824E+03 (B) .100148849297E+03 (C)	.100242328229E+03	.922894012738E-03 .926748276880E-03 .932529539024E-03
1.70	.121193894010E+03 (A) .121193335539E+03 (B) .121192497854E+03 (C)	.121331621407E+03	.113513192929E-02 .113973477670E-02 .114663886846E-02
1.80	.145900197789E+03 (A) .145899425179E+03 (B) .145898266297E+03 (C)	.146096168080E+03	.134137871608E-02 .134666708162E-02 .135459940673E-02
1.90	.174831793908E+03 (A) .174830758642E+03 (B) .174829205788E+03 (C)	.175101520393E+03	.154040058834E-02 .154631296640E-02 .155518126903E-02
2.00	.208632138690E+03 (A) .208630784600E+03 (B) .208628753528E+03 (C)	.208994066965E+03	.173176339644E-02 .173824248020E-02 .174796080318E-02

Table(5.3.3e)

Notations

The following notations are used in Table(5.3.3a)-Table(5.3.3e).

A denotes AM Formula (5.3.1-9b)

B denotes GM Formula (5.3.1-9c)

C denotes GM Formula (5.3.1-9d)

5.3.4 DERIVATION OF THE NEW GM STRATEGY WITH ERROR CONTROL

Consider the new GM formula given by (5.3.1-9c) written as

$$Y_{n+1}^* = 2Y_n - Y_{n-1} + \frac{h^2}{12} \left\{ 6f_n + 4\sqrt{f_n} [\sqrt{f_{n+1}} + \sqrt{f_{n-1}}] - 2\sqrt{f_{n+1}f_{n-1}} \right\}. \quad (5.3.4-1)$$

Its local truncation error is obtained as

$$T_1^{\text{GM}} = h^6 \left[\frac{f_n^{(4)}}{240} - \frac{M_n}{192} \right],$$

where M_n is given by (5.3.2-2).

The AM formula given by (5.3.1-9b) is now written as

$$Y_{n+1}^{**} = 2Y_n - Y_{n-1} + \frac{h^2}{12} \left\{ f_{n-1} + 10f_n + f_{n+1} \right\}, \quad (5.3.4-2)$$

and its local truncation error is

$$T^{\text{AM}} = \frac{h^6}{240} f_n^{(4)}.$$

Thus we can write (5.3.4-1) and (5.3.4-2) respectively as

$$Y_{n+1}^* \approx Y(x_{n+1}) - h^6 \left[\frac{f_n^{(4)}}{240} - \frac{M_n}{192} \right], \quad (5.3.4-3a)$$

and

$$Y_{n+1}^{**} \approx Y(x_{n+1}) - \frac{h^6}{240} f_n^{(4)}. \quad (5.3.4-3b)$$

By subtracting (5.3.4-3b) from (5.3.4-3a), we obtain

$$Y_{n+1}^* - Y_{n+1}^{**} \approx \frac{h^6}{192} M_n. \quad (5.3.4-4)$$

Therefore the error in the final result is approximated by

$$y(x_{n+1}) - Y_{n+1}^{**} \approx \frac{4}{5} (Y_{n+1}^* - Y_{n+1}^{**}) \frac{f_n^{(4)}}{M_n}. \quad (5.3.4-5)$$

Now $f_n^{(4)}$ can be obtained from (4.3.3-14) and by using the bounds for f and its derivatives given in (4.3.1.1-6), we may approximate the bound for (5.3.4-5) as

$$\left| y(x_{n+1}) - Y_{n+1}^{**} \right| < \frac{4}{5} \left| Y_{n+1}^* - Y_{n+1}^{**} \right| S, \quad (5.3.4-6)$$

where it can be shown that

$$S = \left| \frac{f_n^{(4)}}{M_n} \right| < \frac{75}{32}.$$

Therefore the estimate for the local truncation error EST, of Y_{n+1}^{**} is given by

$$\text{EST} = \frac{4}{5} \left| Y_{n+1}^* - Y_{n+1}^{**} \right| S,$$

i.e.

$$\text{EST} = \frac{15}{8} \left| Y_{n+1}^* - Y_{n+1}^{**} \right|. \quad (5.3.4-7)$$

We note that the method is 4th-order accurate when used to solve any problem of the type $y^{(2)} = f(x, y)$. The error estimate EST, is obtained by assuming that the function $f(x, y)$ and its derivatives satisfy the bounds set by Lotkin[1951] as stated in (4.3.1.1-6).

5.3.5 NUMERICAL RESULTS

Problem 1 $y^{(2)} + 2(1 + 2x^2)y = 0$

Initial conditions $x_0 = 0, y_0 = 1, y_0^{(1)} = 0$

Exact solution $y = e^{-x^2}$

Error tolerance = .5E-04

End value of x is 1.00

Initial value of h is .10

x_n	Numerical Solution	Exact Solution	Estimated Error	Relative Error
=====				
stepsize = .200000				
.40	.838355985316E+00	.852143788966E+00	.273514E-05	.744424E-02
.60	.654098199936E+00	.697676326071E+00	.223660E-04	.256693E-01
stepsize = .100000				
stepsize = .200000				
stepsize = .100000				
.60	.682157705837E+00	.697676326071E+00	.100610E-05	.914109E-02
.70	.592649391481E+00	.612626394184E+00	.225991E-05	.123879E-01
.80	.503447183285E+00	.527292424043E+00	.405816E-05	.156128E-01
.90	.418074783906E+00	.444858066223E+00	.637544E-05	.185370E-01
1.0	.339298962571E+00	.367879441171E+00	.914598E-05	.208940E-01
1.1	.269031977289E+00	.298197279430E+00	.122611E-04	.224660E-01

Table(5.3.5a)

Problem 2 $y^{(2)} + 3(2 - 3x^3)xy = 0$

Initial conditions $x_0 = 0, y_0 = 1, y_0^{(1)} = 0$

Exact solution $y = e^{-x^3}$

Error tolerance = .5E-05

End value of x is 1.00

Initial value of h is .10

x_n	Numerical Solution	Exact Solution	Estimated Error	Relative Error
=====				
stepsize = .050000				
stepsize = .025000				
stepsize = .012500				
.0250	.999984375128E+00	.999984375122E+00	.628367E-06	.286085E-11
stepsize = .025000				
.0625	.999755889541E+00	.999755889175E+00	.382381E-06	.182909E-09
stepsize = .050000				
.1375	.997403789770E+00	.997403766683E+00	.182059E-05	.115584E-07
.1875	.993429903457E+00	.993429881359E+00	.473009E-06	.110856E-07
.2375	.986692869220E+00	.986692849158E+00	.238859E-06	.100978E-07
stepsize = .100000				
.2375	.986692869220E+00	.986692849158E+00	.238859E-06	.100978E-07
stepsize = .100000				
stepsize = .050000				
.2875	.976516477444E+00	.976516460795E+00	.174163E-06	.842382E-08
stepsize = .100000				
stepsize = .050000				
.3375	.962286219322E+00	.962286207688E+00	.159438E-06	.592838E-08
stepsize = .100000				
stepsize = .050000				
.3875	.943474877305E+00	.943474872381E+00	.167036E-06	.253368E-08
stepsize = .100000				
stepsize = .050000				
.4375	.919670120386E+00	.919670123753E+00	.188399E-06	.175437E-08
.4875	.890602181746E+00	.890602194637E+00	.220495E-06	.681832E-08
.5375	.856169304930E+00	.856169327952E+00	.262675E-06	.124027E-07
.5875	.816458383600E+00	.816458416466E+00	.316626E-06	.180936E-07
.6375	.771758163392E+00	.771758204691E+00	.388667E-06	.233099E-07
.6875	.722562587107E+00	.722562634159E+00	.497253E-06	.273153E-07
.7375	.669562387936E+00	.669562436783E+00	.699821E-06	.292575E-07
.7875	.613623874279E+00	.613623919855E+00	.122749E-05	.282440E-07
.8375	.555754960699E+00	.555754997195E+00	.380470E-05	.234583E-07
stepsize = .025000				
stepsize = .050000				
stepsize = .025000				
stepsize = .050000				
stepsize = .025000				
.8375	.555754996736E+00	.555754997195E+00	.128263E-06	.294733E-09
.8625	.526439353933E+00	.526439354264E+00	.719990E-06	.217185E-09
stepsize = .012500				
stepsize = .025000				
stepsize = .012500				
.8625	.526439354260E+00	.526439354264E+00	.328206E-07	.265009E-11
.8750	.511748556578E+00	.511748556581E+00	.555681E-06	.189540E-11
stepsize = .025000				
stepsize = .012500				
.8875	.497059808109E+00	.497059808111E+00	.672323E-07	.107948E-11
.9000	.482391140115E+00	.482391140115E+00	.290194E-06	.203450E-12
stepsize = .025000				
.9375	.438684585146E+00	.438684584982E+00	.544515E-06	.113566E-09
stepsize = .050000				
1.0125	.354172711588E+00	.354172674704E+00	.352268E-05	.272375E-07

Table (5.3.5b)

Problem 3 $y^{(2)} - y = 0$.

Initial conditions $x_0 = 0$, $y_0 = 1$, $y_0^{(1)} = -1$.

Exact solution $y = e^{-x}$.

Error tolerance = $.5E-09$

End value of x is 1.00

Initial value of h is .20

x_n	Numerical Solution	Exact Solution	Estimated Error	Relative Error
=====				
stepsize = .100000				
stepsize = .050000				
.100	.904837417995E+00	.904837418036E+00	.145037E-09	.216752E-10
.150	.860707976386E+00	.860707976425E+00	.137964E-09	.211072E-10
.200	.818730753041E+00	.818730753078E+00	.131235E-09	.205411E-10
.250	.778800783036E+00	.778800783071E+00	.124835E-09	.199780E-10
.300	.740818220648E+00	.740818220682E+00	.118746E-09	.194183E-10
.350	.704688089687E+00	.704688089719E+00	.112955E-09	.188625E-10
.400	.670320046005E+00	.670320046036E+00	.107446E-09	.183120E-10
.450	.637628151593E+00	.637628151622E+00	.102206E-09	.177666E-10
.500	.606530659685E+00	.606530659713E+00	.972214E-10	.172272E-10
.550	.576949810354E+00	.576949810380E+00	.924798E-10	.166945E-10
.600	.548811636069E+00	.548811636094E+00	.879694E-10	.161686E-10
.550	.576949810354E+00	.576949810380E+00	.924798E-10	.166945E-10
.600	.548811636069E+00	.548811636094E+00	.879694E-10	.161686E-10
.650	.522045776737E+00	.522045776761E+00	.836791E-10	.156508E-10
.700	.496585303769E+00	.496585303791E+00	.795981E-10	.151406E-10
.750	.472366552719E+00	.472366552741E+00	.757161E-10	.146391E-10
.800	.449328964097E+00	.449328964117E+00	.720234E-10	.141465E-10
.850	.427414931929E+00	.427414931949E+00	.685107E-10	.136632E-10
.900	.406569659722E+00	.406569659741E+00	.651694E-10	.131894E-10
.950	.386741023437E+00	.386741023455E+00	.619910E-10	.127256E-10
1.000	.367879441155E+00	.367879441171E+00	.589677E-10	.122718E-10

Table (5.3.5c)

Problem 4 $y^{(2)} - y = 0$

Initial conditions $x_0 = 0, y_0 = 1, y_0^{(1)} = 1.$

Exact solution $y = e^x.$

Error tolerance = $.5E-09$

End value of x is 1.00

Initial value of h is .20

x_n	Numerical Solution	Exact Solution	Estimated Error	Relative Error
=====				
stepsize = .100000				
stepsize = .050000				
.100	.110517091803E+01	.110517091808E+01	.160297E-09	.216753E-10
.150	.116183424268E+01	.116183424273E+01	.168516E-09	.221892E-10
.200	.122140275811E+01	.122140275816E+01	.177156E-09	.227015E-10
.250	.128402541663E+01	.128402541669E+01	.186239E-09	.232110E-10
.300	.134985880752E+01	.134985880758E+01	.195788E-09	.237176E-10
.350	.141906754853E+01	.141906754859E+01	.205826E-09	.242200E-10
.400	.149182469758E+01	.149182469764E+01	.216379E-09	.247186E-10
.450	.156831218543E+01	.156831218549E+01	.227473E-09	.252119E-10
.500	.164872127063E+01	.164872127070E+01	.239136E-09	.257001E-10
.550	.173325301780E+01	.173325301787E+01	.251397E-09	.261819E-10
.600	.182211880032E+01	.182211880039E+01	.264286E-09	.266578E-10
.650	.191554082893E+01	.191554082901E+01	.277836E-09	.271266E-10
.700	.201375270739E+01	.201375270747E+01	.292081E-09	.275880E-10
.750	.211700001653E+01	.211700001661E+01	.307056E-09	.280419E-10
.800	.222554092840E+01	.222554092849E+01	.322800E-09	.284876E-10
.850	.233964685183E+01	.233964685193E+01	.339350E-09	.289248E-10
.900	.245960311106E+01	.245960311116E+01	.356749E-09	.293537E-10
.950	.258570965921E+01	.258570965932E+01	.375039E-09	.297732E-10
1.000	.271828182835E+01	.271828182846E+01	.394269E-09	.301837E-10

Table (5.3.5d)

5.4 IMPLICIT FORMULA FOR A SPECIAL CLASS OF FOURTH-ORDER ODES.

Consider the special fourth-order ODEs problems of the type

$$y^{(4)} = f(x, y) \quad (5.4-1a)$$

with the initial conditions at $x = x_0, y(x_0) = y_0, y^{(1)}(x_0) = y_0^{(1)}, y^{(2)}(x_0) = y_0^{(2)}, y^{(3)}(x_0) = y_0^{(3)}$. Such problems occur in the vibration analysis of beams.

Let the general form of the formula which approximates the solution of (5.4-1a) be defined by

$$\begin{aligned}
& Y_{n-2} - 4Y_{n-1} + 6Y_n - 4Y_{n+1} + Y_{n+2} \\
& = h^4 \left\{ \alpha_1 f_n + \alpha_2 f_{n-1} + \alpha_3 f_{n-2} + \alpha_4 f_{n+1} + \alpha_5 f_{n+2} \right. \\
& + \alpha_6 \sqrt{f_n f_{n-1}} + \alpha_7 \sqrt{f_n f_{n+1}} + \alpha_8 \sqrt{f_n f_{n-2}} \\
& + \alpha_9 \sqrt{f_n f_{n+2}} + \alpha_{10} \sqrt{f_{n-1} f_{n+1}} + \alpha_{11} \sqrt{f_{n-1} f_{n-2}} \\
& + \alpha_{12} \sqrt{f_{n-1} f_{n+2}} + \alpha_{13} \sqrt{f_{n+1} f_{n-2}} + \alpha_{14} \sqrt{f_{n+1} f_{n+2}} \\
& \left. + \alpha_{15} \sqrt{f_{n-2} f_{n+2}} \right\}. \tag{5.4-2}
\end{aligned}$$

The Taylor series expansion of $f_{n\pm 1}$ about x_n is given by

$$\begin{aligned}
f_{n\pm 1} = f_n \left\{ 1 \pm h \frac{f_n^{(1)}}{f_n} + \frac{h^2}{2!} \frac{f_n^{(2)}}{f_n} \pm \frac{h^3}{3!} \frac{f_n^{(3)}}{f_n} + \frac{h^4}{4!} \frac{f_n^{(4)}}{f_n} \right. \\
\pm \frac{h^5}{5!} \frac{f_n^{(5)}}{f_n} + \frac{h^6}{6!} \frac{f_n^{(6)}}{f_n} \pm \frac{h^7}{7!} \frac{f_n^{(7)}}{f_n} + \frac{h^8}{8!} \frac{f_n^{(8)}}{f_n} \\
\left. \pm \frac{h^9}{9!} \frac{f_n^{(9)}}{f_n} + \frac{h^{10}}{10!} \frac{f_n^{(10)}}{f_n} + \dots \right\}. \tag{5.4-3a}
\end{aligned}$$

Similarly, we can deduce the expansion of $f_{n\pm 2}$ from (5.4-3a) above by substituting h with $2h$.

Hence using the expansion of $f_{n\pm 1}$ and $f_{n\pm 2}$, we list the following results which were obtained from the REDUCE program for symbolic manipulation,

$$\begin{aligned}
\sqrt{f_n f_{n\pm 1}} = f_n \left\{ 1 \pm \frac{h}{2} \frac{f_n^{(1)}}{f_n} + \frac{h^2}{8} \left[2 \frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
\pm \frac{h^3}{48} \left[4 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \\
+ \frac{h^4}{384} \left[8 \frac{f_n^{(4)}}{f_n} - 16 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} - 12 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\
\left. \left. + 36 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 15 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \pm \dots \right.
\end{aligned}$$

$$\begin{aligned}
& \pm \frac{h^5}{3840} \left[16 \frac{f_n^{(5)}}{f_n} - 40 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 80 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
& \quad + 180 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 120 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \\
& \quad \left. - 300 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 105 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] \\
& + \frac{h^6}{46080} \left[32 \frac{f_n^{(6)}}{f_n} - 96 \frac{f_n^{(1)} f_n^{(5)}}{f_n^2} + 360 \left(\frac{f_n^{(2)}}{f_n} \right)^3 \right. \\
& \quad - 240 \frac{f_n^{(2)} f_n^{(4)}}{f_n^2} - 160 \left(\frac{f_n^{(3)}}{f_n} \right)^2 \\
& \quad + 360 \frac{f_n^{(4)} (f_n^{(1)})^2}{f_n^3} + 1440 \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^3} \\
& \quad - 1200 \frac{f_n^{(3)} (f_n^{(1)})^3}{f_n^4} - 2700 \frac{(f_n^{(1)} f_n^{(2)})^2}{f_n^4} \\
& \quad \left. + 3150 \frac{(f_n^{(1)})^4 f_n^{(2)}}{f_n^5} - 945 \left(\frac{f_n^{(1)}}{f_n} \right)^6 \right] + \dots \}.
\end{aligned}$$

(5.4-4)

$$\begin{aligned}
\sqrt{f_{n-1} f_{n+1}} &= f_n \left\{ 1 + \frac{h^2}{2} \left[\frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
& + \frac{h^4}{24} \left[\frac{f_n^{(4)}}{f_n} - 4 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} + 6 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 3 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\
& \left. + \frac{h^6}{720} \left[\frac{f_n^{(6)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(5)}}{f_n^2} - 10 \left(\frac{f_n^{(3)}}{f_n} \right)^2 + 15 \frac{f_n^{(4)} (f_n^{(1)})^2}{f_n^3} + \dots \right] \right\}
\end{aligned}$$

$$\begin{aligned}
& + 60 \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^3} - 60 \frac{f_n^{(3)} (f_n^{(1)})^3}{f_n^4} - 90 \frac{(f_n^{(1)} f_n^{(2)})^2}{f_n^4} \\
& + 135 \frac{(f_n^{(1)})^4 f_n^{(2)}}{f_n^5} - 45 \left(\frac{f_n^{(1)}}{f_n} \right)^6 \Big] + \dots \Big\}. \quad (5.4-5)
\end{aligned}$$

$$\begin{aligned}
\sqrt{f_{n+2} f_{n+1}} &= f_n \left\{ 1 \pm \frac{3}{2} h \frac{f_n^{(1)}}{f_n} + \frac{h^2}{8} \left[10 \frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
&\pm \frac{h^3}{16} \left[12 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \\
&+ \frac{h^4}{384} \left[136 \frac{f_n^{(4)}}{f_n} - 112 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} - 108 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\
&\quad \left. + 276 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 111 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\
&\pm \frac{h^5}{3840} \left[176 \frac{f_n^{(5)}}{f_n} - 200 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 560 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
&\quad + 1140 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 680 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \\
&\quad \left. - 1740 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 585 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] \\
&+ \frac{h^6}{46080} \left[2080 \frac{f_n^{(6)}}{f_n} - 2976 \frac{f_n^{(1)} f_n^{(5)}}{f_n^2} + 16200 \left(\frac{f_n^{(2)}}{f_n} \right)^3 \right. \\
&\quad \left. - 10800 \frac{f_n^{(2)} f_n^{(4)}}{f_n^2} - 7840 \left(\frac{f_n^{(3)}}{f_n} \right)^2 \right. \\
&\quad \left. + 12840 \frac{f_n^{(4)} (f_n^{(1)})^2}{f_n^3} + 60000 \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^3} - \dots \right]
\end{aligned}$$

$$\begin{aligned}
& - 44880 \frac{f_n^{(3)} (f_n^{(1)})^3}{f_n^4} - 107820 \frac{(f_n^{(1)} f_n^{(2)})^2}{f_n^4} \\
& + 117990 \frac{(f_n^{(1)})^4 f_n^{(2)}}{f_n^5} - 34065 \left(\frac{f_n^{(1)}}{f_n} \right)^6 \Big] + \dots \Big\} .
\end{aligned}
\tag{5.4-6}$$

$$\begin{aligned}
\sqrt{f_{n\mp 1} f_{n\pm 2}} &= f_n \left\{ 1 \pm \frac{1}{2} h \frac{f_n^{(1)}}{f_n} + \frac{1}{8} h^2 \left[10 \frac{f_n^{(2)}}{f_n} - 9 \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
&\quad \pm \frac{h^3}{16} \left[16 \frac{f_n^{(3)}}{f_n} - 18 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 9 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \\
&\quad + \frac{h^4}{384} \left[136 \frac{f_n^{(4)}}{f_n} - 432 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} - 108 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\
&\quad \quad \left. + 756 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 351 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\
&\quad \pm \frac{h^5}{3840} \left[496 \frac{f_n^{(5)}}{f_n} - 1800 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 2160 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
&\quad \quad + 5940 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 4920 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \\
&\quad \quad \left. - 11340 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 4185 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] \\
&\quad + \frac{h^6}{46080} \left[2080 \frac{f_n^{(6)}}{f_n} - 9504 \frac{f_n^{(1)} f_n^{(5)}}{f_n^2} + 16200 \left(\frac{f_n^{(2)}}{f_n} \right)^3 \right. \\
&\quad \quad - 10800 \frac{f_n^{(2)} f_n^{(4)}}{f_n^2} - 12960 \left(\frac{f_n^{(3)}}{f_n} \right)^2 + 29160 \frac{f_n^{(4)} (f_n^{(1)})^2}{f_n^3} \\
&\quad \quad \left. + 108000 \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^3} - 101520 \frac{f_n^{(3)} (f_n^{(1)})^3}{f_n^4} - \dots \right]
\end{aligned}$$

$$\begin{aligned}
& - 192780 \frac{(f_n^{(1)} f_n^{(2)})^2}{f_n^4} + 251910 \frac{(f_n^{(1)})^4 f_n^{(2)}}{f_n^5} \\
& - 79785 \left(\frac{f_n^{(1)}}{f_n} \right)^6 \Big] + \dots \Big\} . \tag{5.4-7}
\end{aligned}$$

Next, we may deduce the expansion of $\sqrt{f_n f_{n+2}}$ and $\sqrt{f_{n-2} f_{n+2}}$ respectively from (5.4-4) and (5.4-5) by substituting h with $2h$.

Now consider the parameters α_i , $i = 1, 2, 3, \dots, 15$ of (5.4-2). If all of the α_i , $i = 1, 2, 3, \dots, 15$ are nonzero and there is no cancellation of terms on the right-hand side of (5.4-2), then obviously the method given by (5.4-2) will involve more work. To minimize this work we can introduce some simplifying properties into the parameters α_i , $i = 1, 2, 3, \dots, 15$. This may automatically reduce the computational complexity, truncation error and rounding errors. Thus we set the guide-lines in selecting the parameters α_i , $i = 1, 2, 3, \dots, 15$ based on those criteria. Moreover, it is natural to set the parameters α_i , $i = 1, 2, 3, \dots, 15$ such that the formula given by (5.4-2) is symmetric.

Hence by letting

$$\begin{aligned}
\alpha_2 &= \alpha_4, \quad \alpha_3 = \alpha_5, \quad \alpha_6 = \alpha_7, \\
\alpha_8 &= \alpha_9, \quad \alpha_{11} = \alpha_{14}, \quad \alpha_{12} = \alpha_{13},
\end{aligned}$$

we obtain the right-hand side of (5.4-2) as

$$\begin{aligned}
\text{RHS (1)} &= h^4 \left\{ \alpha_1 f_n + \alpha_2 [f_{n-1} + f_{n+1}] + \alpha_3 [f_{n-2} + f_{n+2}] \right. \\
&+ \alpha_6 [\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}] + \alpha_8 [\sqrt{f_n f_{n-2}} + \sqrt{f_n f_{n+2}}] \\
&+ \alpha_{10} \sqrt{f_{n+1} f_{n-1}} + \alpha_{11} [\sqrt{f_{n-2} f_{n-1}} + \sqrt{f_{n+2} f_{n+1}}] \\
&\left. + \alpha_{12} [\sqrt{f_{n+1} f_{n-2}} + \sqrt{f_{n-1} f_{n+2}}] + \alpha_{15} \sqrt{f_{n+2} f_{n-2}} \right\} \tag{5.4-8}
\end{aligned}$$

Again using the REDUCE program, we obtain (5.4-8) as

$$\begin{aligned}
 \text{RHS}(1) = h^4 \{ & [\alpha_1 + 2(\alpha_2 + \alpha_3 + \alpha_6 + \alpha_8 + \alpha_{11} + \alpha_{12}) \\
 & + \alpha_{10} + \alpha_{15}] f_n \\
 & + \frac{1}{2} [2(\alpha_2 + 2(2\alpha_3 + \alpha_8 + \alpha_{15})) \\
 & + \alpha_6 + \alpha_{10} + 5(\alpha_{11} + \alpha_{12})] h^2 f_n^{(2)} \\
 & + \frac{1}{24} [2(\alpha_2 + 8(2\alpha_3 + \alpha_8 + \alpha_{15})) \\
 & + \alpha_6 + \alpha_{10} + 17(\alpha_{11} + \alpha_{12})] h^4 f_n^{(4)} \\
 & + \frac{1}{720} [2(\alpha_2 + 32(2\alpha_3 + \alpha_8 + \alpha_{15})) \\
 & + \alpha_6 + \alpha_{10} + 65(\alpha_{11} + \alpha_{12})] h^6 f_n^{(6)} \\
 & + \frac{1}{4} [\alpha_6 + 2(2(\alpha_8 + 2\alpha_{15}) + \alpha_{10}) \\
 & + \alpha_{11} + 9\alpha_{12}] h^2 \frac{(f_n^{(1)})^2}{f_n} \\
 & + \frac{1}{12} [\alpha_6 + 4(4((\alpha_8 + 2\alpha_{15}) + \alpha_{10}) \\
 & + 7\alpha_{11} + 27\alpha_{12})] h^4 \frac{f_n^{(1)} f_n^{(3)}}{f_n} \\
 & + \frac{1}{16} [\alpha_6 + 16\alpha_8 + 9(\alpha_{11} + \alpha_{12})] h^4 \frac{(f_n^{(1)})^2}{f_n} \\
 & + \frac{1}{16} [3(\alpha_6 + 21\alpha_{12}) + 4(4(3\alpha_8 + 4\alpha_{15}) + \alpha_{10}) \\
 & + 23\alpha_{11}] h^4 f_n^{(2)} \left(\frac{f_n^{(1)}}{f_n}\right)^2 \\
 & + \frac{1}{64} [5\alpha_6 + 8(2(5\alpha_8 + 8\alpha_{15}) + \alpha_{10}) \\
 & + 37\alpha_{11} + 117\alpha_{12}] h^4 \frac{(f_n^{(1)})^4}{f_n^3} \\
 & + \frac{1}{23040} h^6 \left\{ -96 [\alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) \right. \\
 & \left. + 31\alpha_{11} + 99\alpha_{12}] \frac{f_n^{(1)} f_n^{(5)}}{f_n} \right. \\
 & - \dots
 \end{aligned}$$

$$\begin{aligned}
& - 240 [\alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12})] \frac{f_n^{(2)} f_n^{(4)}}{f_n} \\
& - 160 [\alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 49\alpha_{11} + 27\alpha_{12}] \frac{(f_n^{(3)})^2}{f_n} \\
& + 480 [3(\alpha_6 + 75\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) \\
& \quad + 125\alpha_{11}] \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^2} \\
& + 120 [3(\alpha_6 + 81\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) \\
& \quad + 107\alpha_{11}] \frac{f_n^{(4)} (f_n^{(1)})^2}{f_n^2} \\
& - 180 [15\alpha_6 + 16(30\alpha_8 + \alpha_{10} + \alpha_{15}) \\
& \quad + 599\alpha_{11} + 1071\alpha_{12}] \frac{(f_n^{(1)} f_n^{(2)})^2}{f_n^3} \\
& + 360 [\alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12})] \frac{(f_n^{(2)})^3}{f_n^2} \\
& - 240 [5\alpha_6 + 8(20\alpha_8 + \alpha_{10} + \alpha_{15}) \\
& \quad + 187\alpha_{11} + 423\alpha_{12}] \frac{f_n^{(3)} (f_n^{(1)})^3}{f_n^3} \\
& + 90 [35\alpha_6 + 16(70\alpha_8 + 3(\alpha_{10} + \alpha_{15})) \\
& \quad + 1311\alpha_{11} + 2799\alpha_{12}] \frac{f_n^{(2)} (f_n^{(1)})^4}{f_n^{(4)}} \\
& - 45 [21\alpha_6 + 32(21\alpha_8 + \alpha_{10} + \alpha_{15}) \\
& \quad + 757\alpha_{11} + 1773\alpha_{12}] \frac{(f_n^{(1)})^6}{f_n^5} \} \} \\
& + \dots \qquad \qquad \qquad (5.4-9)
\end{aligned}$$

Now the Taylor series expansion of y_{n+1} and y_{n+2} about x_n are respectively given by

$$\begin{aligned}
Y_{n\pm 1} = Y_n \pm h y_n^{(1)} + \frac{1}{2} h^2 Y_n^{(2)} \pm \frac{1}{3!} h^3 Y_n^{(3)} + \frac{1}{4!} h^4 Y_n^{(4)} \\
\pm \frac{1}{5!} h^5 Y_n^{(5)} + \frac{1}{6!} h^6 Y_n^{(6)} \pm \frac{1}{7!} h^7 Y_n^{(7)} \\
+ \frac{1}{8!} h^8 Y_n^{(8)} \pm \frac{1}{9!} h^9 Y_n^{(9)} + \frac{1}{10!} h^{10} Y_n^{(10)} \\
+ \dots
\end{aligned} \tag{5.4-10a}$$

and

$$\begin{aligned}
Y_{n\pm 2} = Y_n \pm 2h y_n^{(1)} + 2 h^2 Y_n^{(2)} \pm \frac{8}{3!} h^3 Y_n^{(3)} + \frac{16}{4!} h^4 Y_n^{(4)} \\
\pm \frac{32}{5!} h^5 Y_n^{(5)} + \frac{64}{6!} h^6 Y_n^{(6)} \pm \frac{128}{7!} h^7 Y_n^{(7)} \\
+ \frac{256}{8!} h^8 Y_n^{(8)} \pm \frac{512}{9!} h^9 Y_n^{(9)} + \frac{1024}{10!} h^{10} Y_n^{(10)} \\
+ \dots
\end{aligned} \tag{5.4-10b}$$

Therefore the left-hand side of (5.4-2) is given as

$$\begin{aligned}
\text{LHS}(1) = h^4 f_n + \frac{1}{6} h^6 f_n^{(2)} + \frac{1}{80} h^8 f_n^{(4)} \\
+ \frac{17}{30240} h^{10} f_n^{(6)} + \dots
\end{aligned} \tag{5.4-11}$$

since $y_n^{(4)} = f_n$.

Hence, by equating the coefficients of like terms in (5.4-9) and (5.4-11) we obtain the results given in Table(5.4).

Thus we have a system of nine equations involving nine unknowns to be solved. By using the REDUCE program, we obtain the solutions given in (5.4-12),

$$\left. \begin{aligned}
\alpha_{15} = \alpha, \quad \alpha_{12} = \beta, \quad \alpha_{11} = \delta \\
\alpha_1 = \frac{1}{120} (79 - 3600\alpha - 1620\beta + 180\delta) \\
\alpha_2 = \frac{1}{180} (31 - 1440\alpha - 630\beta - 90\delta) \\
\alpha_3 = \frac{1}{720} (-1 + 360\alpha + 180\beta - 180\delta) \\
\alpha_6 = 32\alpha + 15\beta - \delta \\
\alpha_8 = \frac{1}{2} (-4\alpha - 3\beta - \delta) \\
\alpha_{10} = -16\alpha - 9\beta + \delta
\end{aligned} \right\} \tag{5.4-12}$$

where α , β and δ are arbitrary parameters.

TERM	COEFFICIENT OF RHS (1)	LHS (1)
$h^4 f_n$	$\alpha_1 + 2(\alpha_2 + \alpha_3 + \alpha_6 + \alpha_8 + \alpha_{11} + \alpha_{12}) + \alpha_{10} + \alpha_{15}$	1
$h^6 f_n^{(2)}$	$\frac{1}{2} \{ 2[\alpha_2 + 2(2\alpha_3 + \alpha_8 + \alpha_{15})] + \alpha_6 + \alpha_{10} + 5(\alpha_{11} + \alpha_{12}) \}$	$\frac{1}{6}$
$h^8 f_n^{(4)}$	$\frac{1}{24} \{ 2[\alpha_2 + 8(2\alpha_3 + \alpha_8 + \alpha_{15})] + \alpha_6 + \alpha_{10} + 17(\alpha_{11} + \alpha_{12}) \}$	$\frac{1}{80}$
$h^{10} f_n^{(6)}$	$\frac{1}{720} \{ 2[\alpha_2 + 32(2\alpha_3 + \alpha_8 + \alpha_{15})] + \alpha_6 + \alpha_{10} + 65(\alpha_{11} + \alpha_{12}) \}$	$\frac{17}{30240}$
$h^6 [f_n^{(1)}]^2 / f_n$	$\frac{1}{4} \{ \alpha_6 + 2[2(\alpha_8 + 2\alpha_{15}) + \alpha_{10}] + \alpha_{11} + 9\alpha_{12} \}$	0
$h^8 f_n^{(1)} f_n^{(3)} / f_n$	$\frac{1}{12} \{ \alpha_6 + 4[\alpha_{10} + 4(\alpha_8 + 2\alpha_{15})] + 7\alpha_{11} + 27\alpha_{12} \}$	0
$h^8 [f_n^{(1)}]^2 / f_n$	$\frac{1}{16} \{ \alpha_6 + 16\alpha_8 + 9(\alpha_{11} + \alpha_{12}) \}$	0
$h^8 f_n^{(2)} \frac{f_n^{(1)}}{f_n}^2$	$\frac{1}{16} \{ 3(\alpha_6 + 21\alpha_{12}) + 4[\alpha_{10} + 4(3\alpha_8 + 4\alpha_{15})] + 23\alpha_{11} \}$	0
$h^8 [f_n^{(1)}]^4 / f_n^3$	$\frac{1}{64} \{ 5\alpha_6 + 8[\alpha_{10} + 2(5\alpha_8 + 8\alpha_{15})] + 37\alpha_{11} + 117\alpha_{12} \}$	0

Table(5.4) : Coefficients of GM4 Formula

Thus there are infinitely many formulae which can be deduced from (5.4-2) depending on the parameters α , β and δ .

Now for $\alpha = \beta = \delta = 0$, (5.4-2) reduces to the AM formula given as

$$Y_{n+2} = 4Y_{n-1} - 6Y_n + 4Y_{n+1} - Y_{n-2} + \frac{1}{720} h^4 \{ 474f_n + 124[f_{n-1} + f_{n+1}] - [f_{n-2} + f_{n+2}] \}, \quad (5.4-13a)$$

while for $\alpha = \frac{1}{144}$, $\beta = \frac{1}{30}$ and $\delta = 0$, (5.4-2) reduces to the new GM4 formula of the form

$$Y_{n+2} = 4Y_{n-1} - 6Y_n + 4Y_{n+1} - Y_{n-2} + \frac{1}{1440} h^4 \{ 15[f_{n-2} + f_{n+2}] + 1040[\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}}] + 10\sqrt{f_{n+2} f_{n-2}} - 592\sqrt{f_{n+1} f_{n-1}} - 92[\sqrt{f_n f_{n-2}} + \sqrt{f_n f_{n+2}}] + 48[\sqrt{f_{n+1} f_{n-2}} + \sqrt{f_{n-1} f_{n+2}}] \}. \quad (5.4-13b)$$

Another form of the GM4 formula which can be deduced from (5.4-12) is by setting $\alpha_1 = 1$ and $\alpha = \beta = 0$. Thus we obtain the GM4 formula as

$$\begin{aligned}
 Y_{n+2} = & 4Y_{n-1} - 6Y_n + 4Y_{n+1} - Y_{n-2} \\
 & + \frac{1}{720} h^4 \left\{ 720f_n + 42 \left[(f_{n-1} + f_{n+1}) - (f_{n-2} + f_{n+2}) \right] \right. \\
 & \quad - 82 \left[2 \left(\sqrt{f_n f_{n-1}} + \sqrt{f_n f_{n+1}} \right. \right. \\
 & \quad \quad \left. \left. - \left(\sqrt{f_{n-1} f_{n+1}} + \sqrt{f_{n-1} f_{n-2}} + \sqrt{f_{n+1} f_{n+2}} \right) \right) \right. \\
 & \quad \left. \left. - \left(\sqrt{f_n f_{n-2}} + \sqrt{f_n f_{n+2}} \right) \right] \right\}. \quad (5.4-13c)
 \end{aligned}$$

5.4.1 VARIANTS OF GM4 FORMULA

We shall now investigate several alternative approaches of determining the values of the parameters α_i , $i = 1, 2, \dots, 15$ of (5.4-2). Given below is a list of some of the possible ways:

$$\begin{aligned}
 \text{case(1): set } & \alpha_2 = \alpha_4, \alpha_3 = \alpha_5, \alpha_6 = \alpha_7, \\
 & \alpha_8 = \alpha_9, \alpha_{11} = \alpha_{14}, \alpha_{12} = \alpha_{13},
 \end{aligned}$$

$$\begin{aligned}
 \text{case(2): set } & \alpha_2 = \alpha_4, \alpha_3 = \alpha_5, \alpha_6 = -\alpha_7, \\
 & \alpha_8 = -\alpha_9, \alpha_{11} = \alpha_{14}, \alpha_{12} = \alpha_{13},
 \end{aligned}$$

$$\begin{aligned}
 \text{case(3): set } & \alpha_2 = \alpha_4, \alpha_3 = \alpha_5, \alpha_6 = -\alpha_7, \\
 & \alpha_8 = -\alpha_9, \alpha_{11} = -\alpha_{14}, \alpha_{12} = -\alpha_{13},
 \end{aligned}$$

$$\begin{aligned}
 \text{case(4): set } & \alpha_2 = \alpha_4, \alpha_3 = \alpha_5, \alpha_6 = \alpha_7, \\
 & \alpha_8 = \alpha_9, \alpha_{11} = \alpha_{12} = \alpha_{13} = \alpha_{14} = 0.
 \end{aligned}$$

Case(1) has been dealt with previously in Table(5.4) and the parameters are given in (5.4-12).

Now consider case(2) above. Then the right-hand side of (5.4-2) becomes

$$\begin{aligned}
\text{RHS (2)} = h^4 \{ & \alpha_1 f_n + \alpha_2 [f_{n-1} + f_{n+1}] + \alpha_3 [f_{n-2} + f_{n+2}] \\
& + \alpha_6 [\sqrt{f_n f_{n-1}} - \sqrt{f_n f_{n+1}}] + \alpha_8 [\sqrt{f_n f_{n-2}} - \sqrt{f_n f_{n+2}}] \\
& + \alpha_{10} \sqrt{f_{n+1} f_{n-1}} + \alpha_{11} [\sqrt{f_{n-2} f_{n-1}} + \sqrt{f_{n+2} f_{n+1}}] \\
& + \alpha_{12} [\sqrt{f_{n+1} f_{n-2}} + \sqrt{f_{n-1} f_{n+2}}] + \alpha_{15} \sqrt{f_{n+2} f_{n-2}} \} \\
& (5.4.1-1)
\end{aligned}$$

By using (5.4-4), we obtain

$$\begin{aligned}
& \alpha_6 [\sqrt{f_n f_{n-1}} - \sqrt{f_n f_{n+1}}] \\
& = -2\alpha_6 f_n \left\{ \frac{h}{2} \frac{f_n^{(1)}}{f_n} + \frac{h^3}{48} \left[4 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \right. \\
& \quad + \frac{h^5}{3840} \left[16 \frac{f_n^{(5)}}{f_n} - 40 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 80 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
& \quad \quad \quad \left. + 180 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 120 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \right. \\
& \quad \quad \quad \left. - 300 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 105 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] \\
& \quad \left. + \dots \right\}. \tag{5.4.1-2a}
\end{aligned}$$

Next, $\alpha_8 [\sqrt{f_n f_{n-2}} - \sqrt{f_n f_{n+2}}]$ is deduced from (5.4.1-2a) by substituting h with $2h$. Thus we have

$$\begin{aligned}
& \alpha_8 [\sqrt{f_n f_{n-2}} - \sqrt{f_n f_{n+2}}] \\
& = -2\alpha_8 f_n \left\{ h \frac{f_n^{(1)}}{f_n} + \frac{h^3}{6} \left[4 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)} f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \right. \\
& \quad + \frac{h^5}{120} \left[16 \frac{f_n^{(5)}}{f_n} - 40 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 80 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
& \quad \quad \quad \left. + 180 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 120 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} - \dots \right. \\
& \quad \left. \left. \right\}
\end{aligned}$$

$$- 300 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 105 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \Big] + \dots \Big\} . \quad (5.4.1-2b)$$

By using (5.4-6), we obtain

$$\begin{aligned} & \alpha_{11} \left[\sqrt{f_{n-1}f_{n-2}} + \sqrt{f_{n+1}f_{n+2}} \right] \\ &= 2\alpha_{11}f_n \left\{ 1 + \frac{h^2}{8} \left[10 \frac{f_n^{(2)}}{f_n} - \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\ & \quad + \frac{h^4}{384} \left[136 \frac{f_n^{(4)}}{f_n} - 112 \frac{f_n^{(1)}f_n^{(3)}}{f_n^2} - 108 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\ & \quad \quad \left. + 276 \frac{f_n^{(2)}(f_n^{(1)})^2}{f_n^3} - 111 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\ & \quad + \frac{h^6}{46080} \left[2080 \frac{f_n^{(6)}}{f_n} - 2976 \frac{f_n^{(1)}f_n^{(5)}}{f_n^2} \right. \\ & \quad \quad - 10800 \frac{f_n^{(2)}f_n^{(4)}}{f_n^2} - 7840 \left(\frac{f_n^{(3)}}{f_n} \right)^2 + 16200 \left(\frac{f_n^{(2)}}{f_n} \right)^3 \\ & \quad \quad + 12840 \frac{f_n^{(4)}(f_n^{(1)})^2}{f_n^3} + 60000 \frac{f_n^{(1)}f_n^{(2)}f_n^{(3)}}{f_n^3} \\ & \quad \quad - 44880 \frac{f_n^{(3)}(f_n^{(1)})^3}{f_n^4} - 107820 \frac{(f_n^{(1)}f_n^{(2)})^2}{f_n^4} \\ & \quad \quad \left. \left. + 117990 \frac{(f_n^{(1)})^4 f_n^{(2)}}{f_n^5} - 34065 \left(\frac{f_n^{(1)}}{f_n} \right)^6 \right] + \dots \right\} . \end{aligned} \quad (5.4.1-2c)$$

$$\begin{aligned} & \alpha_{11} \left[\sqrt{f_{n-1}f_{n-2}} - \sqrt{f_{n+1}f_{n+2}} \right] \\ &= -2\alpha_{11}f_n \left\{ \frac{3}{2}h \frac{f_n^{(1)}}{f_n} + \frac{h^3}{16} \left[12 \frac{f_n^{(3)}}{f_n} - 6 \frac{f_n^{(1)}f_n^{(2)}}{f_n^2} + 3 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \right. \\ & \quad \left. + \dots \right\} \end{aligned}$$

$$\begin{aligned}
& + \frac{h^5}{3840} \left[176 \frac{f_n^{(5)}}{f_n} - 200 \frac{f_n^{(1)} f_n^{(4)}}{f_n^2} - 560 \frac{f_n^{(2)} f_n^{(3)}}{f_n^2} \right. \\
& \quad + 1140 \frac{f_n^{(1)} (f_n^{(2)})^2}{f_n^3} + 680 \frac{f_n^{(3)} (f_n^{(1)})^2}{f_n^3} \\
& \quad \left. - 1740 \frac{f_n^{(2)} (f_n^{(1)})^3}{f_n^4} + 585 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] + \dots \}.
\end{aligned}
\tag{5.4.1-2d}$$

By using (5.4-7), we obtain

$$\begin{aligned}
& \alpha_{12} \left[\sqrt{f_{n-1} f_{n+2}} + \sqrt{f_{n+1} f_{n-2}} \right] \\
& = 2\alpha_{12} f_n \left\{ 1 + \frac{1}{8} h^2 \left[10 \frac{f_n^{(2)}}{f_n} - 9 \left(\frac{f_n^{(1)}}{f_n} \right)^2 \right] \right. \\
& \quad + \frac{h^4}{384} \left[136 \frac{f_n^{(4)}}{f_n} - 432 \frac{f_n^{(1)} f_n^{(3)}}{f_n^2} - 108 \left(\frac{f_n^{(2)}}{f_n} \right)^2 \right. \\
& \quad \quad \left. + 756 \frac{f_n^{(2)} (f_n^{(1)})^2}{f_n^3} - 351 \left(\frac{f_n^{(1)}}{f_n} \right)^4 \right] \\
& \quad + \frac{h^6}{46080} \left[2080 \frac{f_n^{(6)}}{f_n} - 9504 \frac{f_n^{(1)} f_n^{(5)}}{f_n^2} + 16200 \left(\frac{f_n^{(2)}}{f_n} \right)^3 \right. \\
& \quad \quad - 10800 \frac{f_n^{(2)} f_n^{(4)}}{f_n^2} - 12960 \left(\frac{f_n^{(3)}}{f_n} \right)^2 \\
& \quad \quad + 29160 \frac{f_n^{(4)} (f_n^{(1)})^2}{f_n^3} + 108000 \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^3} \\
& \quad \quad - 101520 \frac{f_n^{(3)} (f_n^{(1)})^3}{f_n^4} - 192780 \frac{(f_n^{(1)} f_n^{(2)})^2}{f_n^4} \\
& \quad \quad \left. \left. + 251910 \frac{(f_n^{(1)})^4 f_n^{(2)}}{f_n^5} - 79785 \left(\frac{f_n^{(1)}}{f_n} \right)^6 \right] + \dots \right\}.
\end{aligned}
\tag{5.4.1-2e}$$

and

$$\begin{aligned}
 & \alpha_{12} \left[\sqrt{f_{n-1}f_{n+2}} - \sqrt{f_{n+1}f_{n-2}} \right] \\
 &= 2\alpha_{12}f_n \left\{ \frac{1}{2}h \frac{f_n^{(1)}}{f_n} + \frac{h^3}{16} \left[16 \frac{f_n^{(3)}}{f_n} - 18 \frac{f_n^{(1)}f_n^{(2)}}{f_n^2} + 9 \left(\frac{f_n^{(1)}}{f_n} \right)^3 \right] \right. \\
 & \quad + \frac{h^5}{3840} \left[496 \frac{f_n^{(5)}}{f_n} - 1800 \frac{f_n^{(1)}f_n^{(4)}}{f_n^2} - 2160 \frac{f_n^{(2)}f_n^{(3)}}{f_n^2} \right. \\
 & \quad \quad \quad \left. + 5940 \frac{f_n^{(1)}(f_n^{(2)})^2}{f_n^3} + 4920 \frac{f_n^{(3)}(f_n^{(1)})^2}{f_n^3} \right. \\
 & \quad \quad \quad \left. \left. - 11340 \frac{f_n^{(2)}(f_n^{(1)})^3}{f_n^4} + 4185 \left(\frac{f_n^{(1)}}{f_n} \right)^5 \right] + \dots \right\}. \tag{5.4.1-2f}
 \end{aligned}$$

Hence, by using (5.4-3a), (5.4-3b), (5.4-5), (5.4.1-2a), (5.4.1-2b), (5.4.1-2c) and (5.4.1-2e); (5.4.1-1) becomes

RHS (2)

$$\begin{aligned}
 &= h^4 \left\{ [\alpha_1 + 2(\alpha_2 + \alpha_3 + \alpha_{11} + \alpha_{12}) + \alpha_{10} + \alpha_{15}] f_n \right. \\
 & \quad - [\alpha_6 + 2\alpha_8] h f_n^{(1)} \\
 & \quad + \frac{1}{2} [2(\alpha_2 + 2(2\alpha_3 + \alpha_{15})) + \alpha_{10} + 5(\alpha_{11} + \alpha_{12})] h^2 f_n^{(2)} \\
 & \quad - \frac{1}{6} [\alpha_6 + 8\alpha_8] h^3 f_n^{(3)} - \frac{1}{120} [\alpha_6 + 32\alpha_8] h^5 f_n^{(5)} \\
 & \quad + \frac{1}{24} [2(\alpha_2 + 8(2\alpha_3 + \alpha_{15})) + \alpha_{10} + 17(\alpha_{11} + \alpha_{12})] h^4 f_n^{(4)} \\
 & \quad + \frac{1}{720} [2(\alpha_2 + 32(2\alpha_3 + \alpha_{15})) + \alpha_{10} + 65(\alpha_{11} + \alpha_{12})] h^6 f_n^{(6)} \\
 & \quad + \frac{1}{4} [2(\alpha_{10} + 4\alpha_{15}) + \alpha_{11} + 9\alpha_{12}] h^2 \frac{(f_n^{(1)})^2}{f_n} \\
 & \quad \left. - \frac{1}{8} [\alpha_6 + 8\alpha_8] h^3 \frac{(f_n^{(1)})^3}{f_n} \right\} + \dots \tag{5.4.1-3}
 \end{aligned}$$

By equating (5.4.1-3) and (5.4-11) we obtain the results given in Table(5.4.1a)

TERM	COEFFICIENT OF RHS (2)	LHS (2)
$h^4 f_n$	$\alpha_1 + 2(\alpha_2 + \alpha_3 + \alpha_{11} + \alpha_{12}) + \alpha_{10} + \alpha_{15}$	1
$h^6 f_n^{(2)}$	$\frac{1}{2}\{2[\alpha_2 + 2(2\alpha_3 + \alpha_{15})] + \alpha_{10} + 5(\alpha_{11} + \alpha_{12})\}$	$\frac{1}{6}$
$h^8 f_n^{(4)}$	$\frac{1}{24}\{2[\alpha_2 + 8(2\alpha_3 + \alpha_{15})] + \alpha_{10} + 17(\alpha_{11} + \alpha_{12})\}$	$\frac{1}{80}$
$h^{10} f_n^{(6)}$	$\frac{1}{720}\{2[\alpha_2 + 32(2\alpha_3 + \alpha_{15})] + \alpha_{10} + 65(\alpha_{11} + \alpha_{12})\}$	$\frac{17}{30240}$
$h \frac{[f_n^{(1)}]^2}{f_n}$	$\frac{1}{4}\{2(\alpha_{10} + 4\alpha_{15}) + \alpha_{11} + 9\alpha_{12}\}$	0
$h^5 f_n^{(1)}$	$-\{\alpha_6 + 2\alpha_8\}$	0
$h^7 f_n^{(3)}$	$-\frac{1}{6}\{\alpha_6 + 8\alpha_8\}$	0
$h^9 f_n^{(5)}$	$-\frac{1}{120}\{\alpha_6 + 32\alpha_8\}$	0
$h \frac{[f_n^{(1)}]^3}{f_n^2}$	$-\frac{1}{8}\{\alpha_6 + 8\alpha_8\}$	0

Table(5.4.1a): Coefficients of GM4 Formula case(2)

Thus by using the REDUCE program, the solutions of the equations obtained from Table(5.4.1a) above are given in (5.4.1-6), where the α , β and δ are some arbitrary constants.

Therefore the corresponding GM4 formula for case(2) is given by

$$\begin{aligned}
 Y_{n+2} = & 4Y_{n-1} - 6Y_n + 4Y_{n+1} - Y_{n-2} \\
 & + \frac{1}{120} h^4 \{ 79f_n + 120[\alpha_2(f_{n-1} + f_{n+1}) + \alpha_3(f_{n-2} + f_{n+2}) \\
 & + \alpha_6(\sqrt{f_n f_{n-1}} - \sqrt{f_n f_{n+1}}) + \alpha_8(\sqrt{f_n f_{n-2}} - \sqrt{f_n f_{n+2}}) \\
 & + \alpha_{10} \sqrt{f_{n+1} f_{n-1}} + \alpha_{11}(\sqrt{f_{n+1} f_{n-2}} + \sqrt{f_{n-1} f_{n+2}})] \} \quad (5.4.1-5)
 \end{aligned}$$

where the parameters of the formula are given in (5.4.1-6).

$$\left. \begin{aligned}
 \alpha_{15} &= 0, \alpha_{12} = 0, \alpha_{11} = \alpha, \alpha_{10} = \beta \\
 \alpha_8 &= \delta, \alpha_6 = -\frac{1}{2} (9\beta + \delta + 8\alpha) \\
 \alpha_3 &= -\frac{1}{720} [360(\alpha + \beta + \delta) + 1] \\
 \alpha_2 &= \frac{1}{180} [45(7\beta - \delta + 8\alpha) + 31] \\
 \alpha_1 &= \frac{79}{120}
 \end{aligned} \right\}. \quad (5.4.1-6)$$

Similarly, by following the same lines of discussions as in the cases(1) and (2), we obtain the results for case(3) and case(4) as in Table(5.4.1b) and Table(5.4.1c) respectively.

However the resulting coefficient matrix obtained from case (3) is singular as can be easily checked by computing its determinant. Thus we do not proceed further with this case.

We shall now consider case(4). The results of equating the right-hand side of (5.4-2) and (5.4-11) are tabulated in Table(5.4.1c)

By using the REDUCE program, the solutions obtained by solving the consistency equations derived for case(4) are as follows:

$$\left. \begin{aligned}
 \alpha_{15} &= \alpha, \alpha_{10} = -16\alpha, \alpha_8 = -2\alpha \\
 \alpha_6 &= 32\alpha, \alpha_3 = \frac{1}{720} [360\alpha - 1] \\
 \alpha_2 &= \frac{1}{180} [-1440\alpha + 31] \\
 \alpha_1 &= \frac{1}{120} [-3600\alpha + 79]
 \end{aligned} \right\}. \quad (5.4.1-7)$$

Thus an investigation of cases (1) to (4) resulted in formulae which are similar in many respects except in the combination of the parameters α_1 .

COEFFICIENT OF:	α_1	α_2	α_3	α_6	α_8	α_{10}	α_{11}	α_{12}	α_{15}	RHS
$h^4 f_n$	1	2	2	0	0	1	0	0	1	1
$h^5 f_n^{(1)}$	0	0	0	1	2	0	3	1	0	0
$h^6 f_n^{(2)}$	0	1	4	0	0	$\frac{1}{2}$	0	0	2	$\frac{1}{6}$
$h^7 f_n^{(3)}$	0	0	0	$\frac{1}{6}$	$\frac{4}{3}$	0	$\frac{3}{2}$	$\frac{7}{6}$	0	0
$h^8 f_n^{(4)}$	0	$\frac{1}{12}$	$\frac{4}{3}$	0	0	$\frac{1}{24}$	0	0	$\frac{2}{3}$	$\frac{1}{80}$
$h^9 f_n^{(5)}$	0	0	0	$\frac{1}{120}$	$\frac{4}{15}$	0	$\frac{11}{40}$	$\frac{31}{120}$	0	0
$h^{10} f_n^{(6)}$	0	$\frac{1}{360}$	$\frac{8}{45}$	0	0	$\frac{1}{720}$	0	0	$\frac{4}{45}$	$\frac{17}{30240}$
$h^6 [f_n^{(1)}]^3 / f_n$	0	0	0	0	0	$\frac{1}{2}$	0	0	2	0
$h^7 [f_n^{(1)}]^3 / f_n^2$	0	0	0	$\frac{1}{8}$	1	0	$\frac{3}{8}$	$\frac{9}{8}$	0	0

Table(5.4.1b): Coefficients of GM4 Formula for case(3)

COEFFICIENT OF:	α_1	α_2	α_3	α_6	α_8	α_{10}	α_{15}	RHS
$h^4 f_n$	1	2	2	2	2	1	1	1
$h^6 f_n^{(2)}$	0	1	4	$\frac{1}{2}$	2	$\frac{1}{2}$	2	$\frac{1}{6}$
$h^8 f_n^{(4)}$	0	$\frac{1}{2}$	$\frac{4}{3}$	$\frac{1}{24}$	$\frac{2}{3}$	$\frac{1}{24}$	$\frac{2}{3}$	$\frac{1}{80}$
$h^{10} f_n^{(6)}$	0	$\frac{1}{360}$	$\frac{8}{45}$	$\frac{1}{720}$	$\frac{4}{45}$	$\frac{1}{720}$	$\frac{4}{45}$	$\frac{17}{30240}$
$h^6 \frac{[f_n^{(1)}]^2}{f_n}$	0	0	0	$\frac{1}{4}$	1	$\frac{1}{2}$	2	0
$h^8 \frac{[f_n^{(2)}]^2}{f_n}$	0	0	0	$\frac{1}{16}$	1	0	0	0
$h^8 f_n^{(2)} \left[\frac{f_n^{(1)}}{f_n} \right]^2$	0	0	0	$\frac{3}{16}$	3	$\frac{1}{4}$	4	0
$h^8 \frac{f_n^{(1)} f_n^{(3)}}{f_n}$	0	0	0	$\frac{1}{12}$	$\frac{4}{3}$	$\frac{1}{6}$	$\frac{8}{3}$	0
$h^8 \frac{[f_n^{(1)}]^4}{f_n^3}$	0	0	0	$\frac{5}{64}$	$\frac{5}{4}$	$\frac{1}{8}$	2	0

Table(5.4.1c): Coefficients of GM4 formula for case(4)

5.4.2 ERROR ANALYSIS OF THE GM4 FORMULAE

By using case(1) to illustrate the error analysis, we obtain the local truncation error of the GM4 formula as

$$T_{n+1}^{GM} = \frac{h^{10}}{720} \left\{ f_n^{(6)} [2(\alpha_2 + 32(2\alpha_3 + \alpha_8 + \alpha_{15})) + \alpha_6 + \alpha_{10} + 65(\alpha_{11} + \alpha_{12})] + 720R \right\} - \frac{17h^{10}}{30240} f_n^{(6)} \quad (5.4.2-1)$$

where

$$R = \frac{1}{23040} \left\{ -96 [\alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 31\alpha_{11} + 99\alpha_{12}] \frac{f_n^{(1)} f_n^{(5)}}{f_n} - 240 [\alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12})] \frac{f_n^{(2)} f_n^{(4)}}{f_n} - 160 [\alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 49\alpha_{11} + 27\alpha_{12}] \frac{[f_n^{(3)}]^2}{f_n} + 120 [3(\alpha_6 + 81\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) + 107\alpha_{11}] \frac{f_n^{(4)} [f_n^{(1)}]^2}{f_n^2} + 480 [3(\alpha_6 + 75\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) + 125\alpha_{11}] \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^2} - 180 [3(5\alpha_6 + 357\alpha_{12}) + 16(30\alpha_8 + \alpha_{10} + \alpha_{15}) + 599\alpha_{11}] \frac{[f_n^{(2)} f_n^{(1)}]^2}{f_n^3} + 360 [\alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12})] \frac{[f_n^{(2)}]^3}{f_n^2} - 240 [5\alpha_6 + 423\alpha_{12} + 8(20\alpha_8 + \alpha_{10} + \alpha_{15}) + 187\alpha_{11}] \frac{f_n^{(3)} [f_n^{(1)}]^3}{f_n^3} + 90 [35\alpha_6 + 2799\alpha_{12} + 16(70\alpha_8 + 3(\alpha_{10} + \alpha_{15})) + 1311\alpha_{11}] \frac{f_n^{(2)} [f_n^{(1)}]^4}{f_n^4} - 45 [3(7\alpha_6 + 591\alpha_{12}) + 32(21\alpha_8 + \alpha_{10} + \alpha_{15}) + 757\alpha_{11}] \frac{[f_n^{(1)}]^6}{f_n^5} \right\}. \quad (5.4.2-1a)$$

For the case(1), we have the parameters of the formula as given in (5.4-12). Since there are infinitely many values of α , β and δ which may satisfy (5.4-12), we shall therefore consider only some values which will meet the criteria stated previously.

First we shall rewrite (5.4.2-1a) in the form of

$$R = \frac{1}{23040} \sum_{i=1}^{10} a_i x_i, \quad (5.4.2-2)$$

where

$$\left. \begin{aligned} a_1 &= -96\{\alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 31\alpha_{11} + 99\alpha_{12}\} \\ a_2 &= -240\{\alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12})\} \\ a_3 &= -160\{\alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 49\alpha_{11} + 27\alpha_{12}\} \\ a_4 &= 120\{3(\alpha_6 + 81\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) + 107\alpha_{11}\} \\ a_5 &= 480\{3(\alpha_6 + 75\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) + 125\alpha_{11}\} \\ a_6 &= -180\{3(5\alpha_6 + 357\alpha_{12}) + 16(30\alpha_8 + \alpha_{10} + \alpha_{15}) + 599\alpha_{11}\} \\ a_7 &= 360\{\alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12})\} \\ a_8 &= -240\{5\alpha_6 + 423\alpha_{12} + 8(20\alpha_8 + \alpha_{10} + \alpha_{15}) + 187\alpha_{11}\} \\ a_9 &= 90\{35\alpha_6 + 2799\alpha_{12} + 16(70\alpha_8 + 3(\alpha_{10} + \alpha_{15})) + 1311\alpha_{11}\} \\ a_{10} &= -45\{3(7\alpha_6 + 591\alpha_{12}) + 32(21\alpha_8 + \alpha_{10} + \alpha_{15}) + 757\alpha_{11}\} \end{aligned} \right\} (5.4.2-2a)$$

and

$$\left. \begin{aligned} x_1 &= \frac{f_n^{(2)} f_n^{(4)}}{f_n} & , & & x_2 &= \frac{f_n^{(1)} f_n^{(5)}}{f_n} \\ x_3 &= \frac{[f_n^{(3)}]^2}{f_n} & , & & x_4 &= \frac{f_n^{(4)} [f_n^{(1)}]^2}{f_n^2} \\ x_5 &= \frac{f_n^{(1)} f_n^{(2)} f_n^{(3)}}{f_n^2} & , & & x_6 &= \frac{[f_n^{(2)} f_n^{(1)}]^2}{f_n^3} \\ x_7 &= \frac{[f_n^{(2)}]^3}{f_n^2} & , & & x_8 &= \frac{f_n^{(3)} [f_n^{(1)}]^3}{f_n^3} \\ x_9 &= \frac{f_n^{(2)} [f_n^{(1)}]^4}{f_n^4} & , & & x_{10} &= \frac{[f_n^{(1)}]^6}{f_n^5} \end{aligned} \right\} (5.4.2-2b)$$

By letting α be arbitrary, and $\beta = \delta = 0$ in (5.4-12), then by using the REDUCE program, we obtain the local truncation error as

$$T_{n+1}^{GM} = \frac{h^{10}}{75600} \{ 19530x_1\alpha + 32550x_2\alpha - 245700x_3\alpha + 425250x_4\alpha \\ - 37800x_5\alpha + 220500x_6\alpha - 543375x_7\alpha + 170100x_8\alpha \\ + 25200x_9\alpha - 61425x_{10}\alpha - 6048\alpha f_n^{(6)} - 25f_n^{(6)} \}. \quad (5.4.2-3)$$

Now we consider (5.4.2-2) with $\mathbf{a} = [a_1, \dots, a_{10}]^T$ and $\mathbf{x} = [x_1, \dots, x_{10}]^T$. Then R will vanish if and only if $\mathbf{a} \mathbf{x}^T = 0$. If we assume that \mathbf{x} is not a zero vector, that is, not all of the components of \mathbf{x} will vanish at the same time, then we should have $\mathbf{a} = \mathbf{0}$. From (5.4.2-2a), for $\mathbf{a} = \mathbf{0}$, this is equivalent to solving the system of simultaneous equations formed by the components of \mathbf{a} . Thus we are required to solve the system of equations

$$\left. \begin{aligned} \alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 31\alpha_{11} + 99\alpha_{12} &= 0 \\ \alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12}) &= 0 \\ \alpha_6 + 2(16\alpha_8 + \alpha_{10} + \alpha_{15}) + 49\alpha_{11} + 27\alpha_{12} &= 0 \\ 3(\alpha_6 + 81\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) + 107\alpha_{11} &= 0 \\ 3(\alpha_6 + 75\alpha_{12}) + 4(24\alpha_8 + \alpha_{10} + \alpha_{15}) + 125\alpha_{11} &= 0 \\ 3(5\alpha_6 + 357\alpha_{12}) + 16(30\alpha_8 + \alpha_{10} + \alpha_{15}) + 599\alpha_{11} &= 0 \\ \alpha_6 + 32\alpha_8 + 45(\alpha_{11} + \alpha_{12}) &= 0 \\ 5\alpha_6 + 423\alpha_{12} + 8(20\alpha_8 + \alpha_{10} + \alpha_{15}) + 187\alpha_{11} &= 0 \\ 35\alpha_6 + 2799\alpha_{12} + 16(70\alpha_8 + 3(\alpha_{10} + \alpha_{15})) + 1311\alpha_{11} &= 0 \\ 3(7\alpha_6 + 591\alpha_{12}) + 32(21\alpha_8 + \alpha_{10} + \alpha_{15}) + 757\alpha_{11} &= 0 \end{aligned} \right\} (5.4.2-4)$$

By using the REDUCE program, we obtain the solutions as

$$\left. \begin{aligned} \alpha_8 &= \gamma \\ \alpha_6 &= -32\gamma \\ \alpha_{15} &= \alpha \\ \alpha_{10} &= -\alpha \\ \alpha_{11} &= \alpha_{12} = 0. \end{aligned} \right\} (5.4.2-4a)$$

By substituting the values of the parameters given by (5.4.2-4a) in (5.4.2-1), we obtain the local truncation error of the GM4 formula for case(1) as

$$T_{n+1}^{GM} = - \frac{1113\alpha + 25}{75600} h^{10} f_n^{(6)}. \quad (5.4.2-5)$$

For the AM formula we have the local truncation error as

$$T_{n+1}^{AM} = - \frac{1}{3024} h^{10} f_n^{(6)}.$$

Hence we have

$$\begin{aligned} \left| \frac{T_{n+1}^{GM}}{T_{n+1}^{AM}} \right| &= \frac{1113\alpha + 25}{75600} 3024, \\ &= \frac{1113\alpha + 25}{25}. \end{aligned}$$

For

$$\left| \frac{T_{n+1}^{GM}}{T_{n+1}^{AM}} \right| < 1,$$

we therefore obtain the result $-\frac{50}{1113} < \alpha < 0$. Hence we can establish the condition under which the GM4 formula has a smaller local truncation error than the AM formula.

5.4.3 COMPUTATIONAL COMPLEXITY OF GM4 FORMULAE

Again in this section we shall make use of the GM4 formulae derived for case(1) to illustrate our discussion. Consider the three formulae given by (5.4-13a), (5.4-13b) and (5.4-13c). Table(5.4.3) compares their computational complexity.

Formula	Addition	Multiplication	Square Root
(5.4-13a)	8	7	0
(5.4-13b)	13	17	8
(5.4-13c)	15	18	7

Table(5.4.3): Computational complexity of GM4 formulae

Thus in any case (that is either formula (5.4-13b) or (5.4-13c)), the GM4 formula involves twice as much work as the AM formula (5.4-13a). However, if the problem to be solved contains the computation of a squared function, then the GM4 formulae would have a better advantage.

5.4.4 NUMERICAL RESULTS

Problem $y^{(4)} = y$

Initial conditions $x = 0, y = 1, y^{(1)} = 1, y^{(2)} = 1, y^{(3)} = 1$

Exact solution $y = e^x$.

The values of y_1, y_2 and y_3 were given by the values of the exact solution of the problem.

Parameters of the equation

CASE(1) $\alpha = 0, \beta = 0, \delta = 0$ (Formula(5.4-13a))

CASE(2) $\alpha = \frac{1}{144}, \beta = \frac{1}{30}, \delta = 0$ (Formula(5.4-13b))

CASE(3) $\alpha = 0, \beta = 0, \delta = \frac{41}{180}$ (Formula(5.4-13c))

x_n	Numerical Solution	Exact Solution	Relative Error
.10	.110517091807565E+01	.110517091807565E+01	0
.20	.122140275816017E+01	.122140275816017E+01	0
.30	.134985880757600E+01	.134985880757600E+01	0
.40	.149182469764126E+01 (1) .149182469764144E+01 (2) .149182469764083E+01 (3)	.149182469764127E+01	.803741129E-14 .111630712E-12 .297979581E-12
.50	.164872127070007E+01 (1) .164872127070098E+01 (2) .164872127069786E+01 (3)	.164872127070013E+01	.369014598E-13 .515947053E-12 .137626284E-11
.60	.182211880039032E+01 (1) .182211880039312E+01 (2) .182211880038356E+01 (3)	.182211880039051E+01	.102484818E-12 .143137535E-11 .381472619E-11
.70	.201375270747003E+01 (1) .201375270747670E+01 (2) .201375270745391E+01 (3)	.201375270747048E+01	.222292401E-12 .308849709E-11 .822790622E-11
.80	.222554092849155E+01 (1) .222554092850518E+01 (2) .222554092845860E+01 (3)	.222554092849247E+01	.412254071E-12 .571309114E-11 .152162858E-10
.90	.245960311115526E+01 (1) .245960311118035E+01 (2) .245960311109463E+01 (3)	.245960311115695E+01	.687546582E-12 .951262584E-11 .253355860E-10
1.00	.271828182845616E+01 (1) .271828182849892E+01 (2) .271828182835283E+01 (3)	.271828182845904E+01	.106109653E-11 .146700884E-10 .390735115E-10

Table(5.4.4)

5.5 EXPLICIT FORMULA FOR A SPECIAL CLASS OF FOURTH-ORDER ODES

In section 5.4, we have derived the implicit formulae for the special fourth-order ODEs problems of the type (5.4-1a). Now it is characteristic of an implicit formula that it requires a predictor to start with. Therefore in this section, we shall derive an explicit formula which could be combined with the implicit formula (5.4-13a) to form a predictor-corrector pair to be used with the special fourth-order ODEs problems (5.4-1a).

Consider the explicit formula defined by

$$Y_{n-2} - 4Y_{n-1} + 6Y_n - 4Y_{n+1} + Y_{n+2} = h^4 \{ \beta_1 f_{n-2} + \beta_2 f_{n-1} + \beta_3 f_n + \beta_4 f_{n+1} \} \quad (5.5-1)$$

where β_i , $i = 1, 2, 3, 4$ are the free parameters to be determined.

By using the Taylor series expansion of f_{n-2} and f_{n+1} given in (5.4-3a), we can write the right-hand side of (5.5-1) as

$$\begin{aligned} & h^4 \left\{ [\beta_1 + \beta_2 + \beta_3 + \beta_4] f_n + h [-2\beta_1 - \beta_2 + \beta_4] f_n^{(1)} \right. \\ & + \frac{h^2}{2} [4\beta_1 + \beta_2 + \beta_4] f_n^{(2)} + \frac{h^3}{6} [-8\beta_1 - \beta_2 + \beta_4] f_n^{(3)} \\ & + \frac{h^4}{24} [16\beta_1 + \beta_2 + \beta_4] f_n^{(4)} + \frac{h^5}{120} [-32\beta_1 - \beta_2 + \beta_4] f_n^{(5)} \\ & \left. + \frac{h^6}{720} [64\beta_1 + \beta_2 + \beta_4] f_n^{(6)} + \dots \right\}. \end{aligned} \quad (5.5-2)$$

Now the left-hand side of (5.5-1) is given by (5.4-11). Therefore by equating (5.5-2) and (5.4-11), we obtain the following results:

$$\left. \begin{aligned}
 \text{coeff. of } h^4 f_n: & \quad \beta_1 + \beta_2 + \beta_3 + \beta_4 = 1 \\
 \text{coeff. of } h^5 f_n^{(1)}: & \quad -2\beta_1 - \beta_2 + \beta_4 = 0 \\
 \text{coeff. of } h^6 f_n^{(2)}: & \quad \frac{1}{2}[4\beta_1 + \beta_2 + \beta_4] = \frac{1}{6} \\
 \text{coeff. of } h^7 f_n^{(3)}: & \quad \frac{1}{6}[-8\beta_1 - \beta_2 + \beta_4] = 0
 \end{aligned} \right\} \quad (5.5-3)$$

On solving (5.5-3) we obtain the solutions as:

$$\beta_1 = 0, \quad \beta_2 = \beta_4 = \frac{1}{6}, \quad \beta_3 = \frac{2}{3}. \quad (5.5-4)$$

Thus the desired explicit formula is

$$\begin{aligned}
 Y_{n+2} = & \quad 4Y_{n-1} - Y_{n-2} - 6Y_n + 4Y_{n+1} \\
 & \quad + \frac{h^4}{6} [f_{n-1} + 4f_n + f_{n+1}]. \quad (5.5-5)
 \end{aligned}$$

Its local truncation error is obtained as

$$\begin{aligned}
 T_{n+1} = & \quad h^8 \left[\frac{1}{24}(16\beta_1 + \beta_2 + \beta_4) - \frac{1}{80} \right] f_n^{(4)}, \\
 = & \quad \frac{h^8}{720} f_n^{(4)}. \quad (5.5-6)
 \end{aligned}$$

Hence the explicit formula given by (5.5-5) is $O(h^8)$. This is confirmed by the numerical results given in section 5.5.1 that follows.

5.5.1 NUMERICAL RESULTS

The numerical results obtained by applying the formula (5.5-5) to some selected problems confirm that it is $O(h^8)$ accurate.

Problem 1 $y^{(4)} = 24 + e^x$.

Initial conditions $x_0 = 0, y_0 = y_0^{(1)} = y_0^{(2)} = y_0^{(3)} = 1$.

Exact solution $y = x^4 + e^x$.

x_n	Numerical Solution	Exact Solution	Relative Error
.2	.122300275816E+01	.122300275816E+01	0
.4	.151742469764E+01	.151742469764E+01	0
.6	.195171880039E+01	.195171880039E+01	0
.8	.263514093378E+01	.263514092849E+01	.2007125152E-08
1.0	.371828185608E+01	.371828182846E+01	.7427166474E-08
1.2	.539371700936E+01	.539371692274E+01	.1605964691E-07
1.4	.789680017843E+01	.789679996684E+01	.2679319322E-07
1.6	.115066328679E+02	.115066324244E+02	.3854663472E-07
1.8	.165472483023E+02	.165472474644E+02	.5063866395E-07
2.0	.233890575667E+02	.233890560989E+02	.6275290991E-07

Table (5.5.1a)

Problem 2 $y^{(4)} = y$.

Initial conditions $x_0 = 0, y_0 = y_0^{(1)} = y_0^{(2)} = y_0^{(3)} = 1$.

Exact solution $y = e^x$.

x_n	Numerical Solution	Exact Solution	Relative Error
.05	.105127109638E+01	.105127109638E+01	0
.10	.110517091808E+01	.110517091808E+01	0
.15	.116183424273E+01	.116183424273E+01	0
.20	.122140275816E+01	.122140275816E+01	.4799381310E-13
.25	.128402541669E+01	.128402541669E+01	.2315512778E-12
.30	.134985880758E+01	.134985880758E+01	.6691643954E-12
.35	.141906754860E+01	.141906754859E+01	.1504480092E-11
.40	.149182469765E+01	.149182469764E+01	.2897040247E-11
.45	.156831218550E+01	.156831218549E+01	.5019644364E-11
.50	.164872127071E+01	.164872127070E+01	.8050848056E-11
.55	.173325301789E+01	.173325301787E+01	.1217313470E-10
.60	.182211880042E+01	.182211880039E+01	.1756804247E-10
.65	.191554082906E+01	.191554082901E+01	.2441430112E-10
.70	.201375270754E+01	.201375270747E+01	.3288494096E-10
.75	.211700001670E+01	.211700001661E+01	.4314482921E-10
.80	.222554092862E+01	.222554092849E+01	.5534959871E-10
.85	.233964685209E+01	.233964685193E+01	.6964445783E-10
.90	.245960311137E+01	.245960311116E+01	.8616269483E-10
.95	.258570965959E+01	.258570965932E+01	.1050258385E-09
1.00	.271828182880E+01	.271828182846E+01	.1263430240E-09

Table (5.5.1b)

Problem 3 $y^{(4)} = 34320(2 - x)^{-14}$.
 Initial conditions $x_0 = -2, y_0 = 2^{-19} + 1$.
 Exact solution $y = 2(2 - x)^{-10} - x - 1$.

x_n	Numerical Solution	Exact Solution	Relative Error
-1.90	.900002456880E+00	.900002456880E+00	0
-1.80	.800003185620E+00	.800003185620E+00	0
-1.70	.700004159228E+00	.700004159228E+00	0
-1.60	.600005470223E+00	.600005470222E+00	.1632753109E-11
-1.50	.500007250198E+00	.500007250193E+00	.1100348289E-10
-1.40	.400009688168E+00	.400009688149E+00	.4679199161E-10
-1.30	.300013058469E+00	.300013058419E+00	.1668920077E-09
-1.20	.200017763685E+00	.200017763568E+00	.5810853334E-09
-1.10	.100024401551E+00	.100024401305E+00	.2461445220E-08
-1.00	.338706660818E-04	.338701756162E-04	.1448075280E-04
-0.90	-.999524600654E-01	-.999524610028E-01	.9378402242E-08
-0.80	-.199932475575E+00	-.199932477318E+00	.8721265620E-08
-0.70	-.299902858094E+00	-.299902861285E+00	.1064142170E-07
-0.60	-.399858318127E+00	-.399858323925E+00	.1450027747E-07
-0.50	-.499790274267E+00	-.499790284800E+00	.2107453149E-07
-0.40	-.599684540155E+00	-.599684559408E+00	.3210568660E-07
-0.30	-.699517181761E+00	-.699517217371E+00	.5090699434E-07
-0.20	-.799246918783E+00	-.799246985763E+00	.8380342085E-07
-0.10	-.898800821963E+00	-.898800950677E+00	.1432061575E-06
0 0	-.998046621172E+00	-.998046875000E+00	.2543249241E-06

Table (5.5.1c)

Problem 4 $y^{(4)} = \cos(x)$.
 Initial conditions $x_0 = 0, y_0 = -y_0^{(2)} = 1, y_0^{(1)} = y_0^{(3)} = 0$.
 Exact solution $y = \cos(x)$.

x_n	Numerical Solution	Exact Solution	Relative Error
.15	.988771077936E+00	.988771077936E+00	0
.30	.955336489126E+00	.955336489126E+00	0
.45	.900447102353E+00	.900447102353E+00	0
.60	.825335615250E+00	.825335614910E+00	.4126850341E-09
.75	.731688870557E+00	.731688868874E+00	.2300772161E-08
.90	.621609973255E+00	.621609968271E+00	.8018581516E-08
1.05	.497571059352E+00	.497571047892E+00	.2303250344E-07
1.20	.362357777026E+00	.362357754477E+00	.6222982817E-07
1.35	.219006726961E+00	.219006687093E+00	.1820386036E-06
1.50	.707372668276E-01	.707372016677E-01	.9211547398E-06
1.65	-.791207885575E-01	-.791208888067E-01	.1267038648E-05
1.80	-.227201947709E+00	-.227202094693E+00	.6469301671E-06
1.95	-.370180624167E+00	-.370180831351E+00	.5596831312E-06
2.10	-.504845822011E+00	-.504846104600E+00	.5597518007E-06
2.25	-.628173247918E+00	-.628173622723E+00	.5966576789E-06
2.40	-.737393230282E+00	-.737393715541E+00	.6580736222E-06
2.55	-.830052920079E+00	-.830053535235E+00	.7411038314E-06
2.70	-.904071376583E+00	-.904072142017E+00	.8466518442E-06
2.85	-.957786300814E+00	-.957787237553E+00	.9780242184E-06
3.00	-.989991367208E+00	-.989992496600E+00	.1140808767E-05

Table (5.5.1d)

5.5.2 PREDICTOR-CORRECTOR PAIR USING (5.4-13a) AND (5.5-5)

By using (5.5-5) as the predictor and (5.4-13a) as the corrector, we form a predictor-corrector pair and investigate numerically its suitability to solve some special fourth-order initial-value ODEs problems. Some numerical results are given in section 5.5.3. The $P(EC)^mE$ algorithm is used in this case. Thus the algorithm is as follows:

Step 1: Initialization of variables.

Step 2: Predict $y_{n+2}^{(P)} = 4y_{n-1} - y_{n-2} - 6y_n + 4y_{n+1}$

$$+ \frac{h^4}{6} [f_{n-1} + 4f_n + f_{n+1}]$$

Step 3: Correct $y_{n+2}^{(C)} = 4y_{n-1} - y_{n-2} - 6y_n + 4y_{n+1}$

$$+ \frac{h^4}{720} \{ 474f_n + 124 [f_{n-1} + f_{n+1}] \\ - [f_{n-2} + f_{n+2}^{(P)}] \}$$

Step 4: If $\left| y_{n+2}^{(P)} - y_{n+2}^{(C)} \right| > \epsilon$ repeat step 3,

else accept $y_{n+2}^{(C)}$ as the numerical solution,

advance to another step interval

and repeat step 3.

5.5.3 NUMERICAL RESULTS

In each problem, the first three solutions were obtained using the exact solution of the problem.

Problem 1 $y^{(4)} = y$

Initial conditions $x_0 = 0, y_0 = y_0^{(1)} = y_0^{(2)} = y_0^{(3)} = 1.$

Exact solution is $y(x) = e^x.$

The value of eps was set at 5E-11.

x_n	Numerical Solution.	Exact Solution.	Relative Error
0.10	0.11051709180756E+01	0.11051709180756E+01	0
0.20	0.12214027581602E+01	0.12214027581602E+01	0
0.30	0.13498588075760E+01	0.13498588075760E+01	0
0.40	0.14918246976412E+01	0.14918246976413E+01	0.270890529E-13
0.50	0.16487212706999E+01	0.16487212707001E+01	0.124845450E-12
0.60	0.18221188003899E+01	0.18221188003905E+01	0.346206143E-12
0.70	0.20137527074690E+01	0.20137527074705E+01	0.747811042E-12
0.80	0.22255409284894E+01	0.22255409284925E+01	0.138382477E-11
0.90	0.24596031111513E+01	0.24596031111569E+01	0.230494214E-11
1.00	0.27182818284494E+01	0.27182818284494E+01	0.355561276E-11

Table(5.5.3a)

Problem 2 $y^{(4)} = \cos(x).$

Initial conditions $x_0 = 0, y_0 = -y_0^{(2)} = 1, y_0^{(1)} = y_0^{(3)} = 0.$

Exact solution $y = \cos(x).$

The value of eps was set to 5e-10.

x_n	Numerical Solution	Exact Solution	Relative Error
.15	.988771077936042E+00	.988771077936042E+00	0
.30	.955336489125606E+00	.955336489125606E+00	0
.45	.900447102352677E+00	.900447102352677E+00	0
.60	.825335614911496E+00	.825335614909678E+00	.181754611E-11
.75	.731688868882805E+00	.731688868873821E+00	.898436880E-11
.90	.621609968297267E+00	.621609968270664E+00	.266026090E-10
1.05	.497571047952894E+00	.497571047891727E+00	.611673490E-10
1.20	.362357754597030E+00	.362357754476674E+00	.120356725E-09
1.35	.219006687305838E+00	.219006687093042E+00	.212796558E-09
1.50	.707372020155056E-01	.707372016677031E-01	.347802426E-09
1.50	.707372020155056E-01	.707372016677031E-01	.347802426E-09
1.65	-.791208882716267E-01	-.791208888067337E-01	.535106959E-09
1.80	-.227202093908509E+00	-.227202094693087E+00	.784577264E-09
1.95	-.370180830245356E+00	-.370180831351286E+00	.110592996E-08
2.10	-.504846103091408E+00	-.504846104599857E+00	.150844937E-08
2.25	-.628173620722024E+00	-.628173622722739E+00	.200071459E-08
2.40	-.737393712950901E+00	-.737393715541245E+00	.259034449E-08
2.55	-.830053531951460E+00	-.830053535235222E+00	.328376171E-08
2.70	-.904072137931075E+00	-.904072142017061E+00	.408598544E-08
2.85	-.957787232552635E+00	-.957787237553090E+00	.500045516E-08
3.00	-.989992490571556E+00	-.989992496600445E+00	.602888928E-08

Table(5.5.3b)

Problem 3 $y^{(4)} = 24 + e^x$.

Initial conditions $x_0 = 0, y_0 = y_0^{(1)} = y_0^{(2)} = y_0^{(3)} = 1$.

Exact solution $y = x^4 + e^x$.

The value of eps was set to 5e-07.

x_n	Numerical Solution.	Exact Solution.	Relative Error
.20	.122300275816017E+01	.122300275816017E+01	0
.40	.151742469764127E+01	.151742469764127E+01	0
.60	.195171880039051E+01	.195171880039051E+01	0
.80	.263514092844179E+01	.263514092849247E+01	.506772402E-10
1.00	.371828182819444E+01	.371828182845904E+01	.264606115E-09
1.20	.539371692190658E+01	.539371692273655E+01	.829963653E-09
1.40	.789679996481741E+01	.789679996684467E+01	.202726458E-08
1.60	.115066324201453E+02	.115066324243951E+02	.424980584E-08
1.80	.165472474563843E+02	.165472474644129E+02	.802864619E-08
2.00	.233890560848676E+02	.233890560989306E+02	.140630867E-07

Table (5.5.3c)

Problem 4 $y^{(4)} = 34320(2 - x)^{-14}$.

Initial conditions $x_0 = -2, y_0 = 2^{-19} + 1$.

Exact solution $y = 2(2 - x)^{-10} - x - 1$.

The value of eps was set to 5e-07.

x_n	Numerical Solution.	Exact Solution.	Relative Error
-1.90	.900002456879889E+00	.900002456879889E+00	0
-1.80	.800003185620442E+00	.800003185620442E+00	0
-1.70	.700004159228257E+00	.700004159228257E+00	0
-1.60	.600005470222397E+00	.600005470222455E+00	.580646642E-13
-1.50	.500007250192412E+00	.500007250192741E+00	.329736238E-12
-1.40	.400009688148210E+00	.400009688149351E+00	.114108722E-11
-1.30	.300013058415803E+00	.300013058418920E+00	.311745074E-11
-1.30	.300013058415803E+00	.300013058418920E+00	.311745074E-11
-1.20	.200017763560974E+00	.200017763568393E+00	.741962047E-11
-1.10	.100024401289039E+00	.100024401305222E+00	.161830410E-10
-1.00	.338701422661497E-04	.338701756161797E-04	.333500300E-10
-.90	-.999524610690470E-01	-.999524610027672E-01	.662797595E-10
-.80	-.199932477447225E+00	-.199932477318294E+00	.128930755E-09
-.70	-.299902861533303E+00	-.299902861285008E+00	.248294774E-09
-.60	-.399858324402958E+00	-.399858323925223E+00	.477735129E-09
-.50	-.499790285725370E+00	-.499790284800001E+00	.925369781E-09
-.40	-.599684561224631E+00	-.599684559408419E+00	.181621251E-08
-.30	-.699517221003311E+00	-.699517217370856E+00	.363245412E-08
-.20	-.799246993203639E+00	-.799246985762834E+00	.744080486E-08
-.10	-.898800966360997E+00	-.898800950676679E+00	.156843185E-07
.00	-.998046909174331E+00	-.998046875000001E+00	.341743300E-07

Table (5.5.3d)

5.6 CONCLUSIONS AND RECOMMENDATIONS

In this chapter we have investigated the feasibility of extending the geometric mean (GM) approach of deriving numerical methods for the solution of problems involving ODEs. For the second-order ODE problems of special type, we have obtained the GM modified form (GM2) of the Numerov method which may be combined together with the Numerov method (AM) to form a new adaptive error control method. Numerical results are presented for the simplified prototype adaptive error control method formed by the combination of the GM2 and the AM methods. Therefore a carefully designed adaptive error control method utilizing these strategies may give some promising results. This could be an area of further research.

We have also investigated special type problems involving fourth-order ODEs. New implicit and explicit methods for solving these problems are derived. Numerical results for some selected problems show that the predictor-corrector pair formed by these formulae are encouraging.

The GM methods may give encouraging results if some criteria are satisfied despite the fact that they may involve more computational work. However, the main contribution of the GM approach is in the derivation of alternative methods which are of the same order as the AM methods. These could then be combined together to form an adaptive error control method with fewer function evaluations than the more conventional approach of combining two methods of different order.

CHAPTER 6

NUMERICAL SOLUTIONS OF PERIODIC BOUNDARY-VALUE PROBLEMS IN ELLIPTIC PDES

6.1 INTRODUCTION

Boundary-value problems involving elliptic PDEs arise naturally as descriptions of processes or equilibrium states in many physical or engineering systems. In this chapter, we shall concentrate the discussion on periodic boundary-value problems of the elliptic PDE type. This type of problem occurs naturally in circular domains, periodic-time and torroidal-space structures. The elliptic PDE can be solved numerically through the applications of finite difference/element methods as described in chapter 3. This often leads to a large system of algebraic equations and their solution is a major numerical problem by itself. There are two alternative methods of solutions; namely direct and iterative methods. For the periodic problems with which we are concerned, we shall show that the standard optimum SOR formula is not applicable and next we derive the optimum parameter for this problem. Finally, we shall develop a new direct method, analogous to the odd-even reduction method, the modified form^{of which} is numerically stable.

6.2 FORMULATION OF THE PROBLEM

Consider the solution of the self-adjoint second-order elliptic PDE of the form

$$\begin{aligned} - [A(x,y)u_x(x,y)]_x - [C(x,y)u_y(x,y)]_y + F(x,y)u(x,y) \\ = G(x,y) \end{aligned} \tag{6.2-1}$$

for $(x,y) \in R_1$ and $R_1 = \{(x,y) | 0 \leq x \leq l_1, 0 \leq y \leq l_2\}$ with $u(x,y)$ periodic both in the x and y directions. According to Mikhlin[1964], this problem belongs to a

class of elliptic fourth^{order} boundary-value problems. The boundary conditions imposed are

$$\left. \begin{aligned} u(x, y) &= u(x \pm l_1, y) \\ u(x, y) &= u(x, y \pm l_2) \end{aligned} \right\} \quad (6.2-2)$$

for all $(x, y) \in R_1$.

The functions $A(x, y)$, $C(x, y)$, $F(x, y)$ and $G(x, y)$ are assumed to be continuous in R_1 and satisfy the following conditions:

$$\left. \begin{aligned} A(x, y) > 0, \quad A(x, y) &= A(x \pm l_1, y) = A(x, y \pm l_2) \\ C(x, y) > 0, \quad C(x, y) &= C(x \pm l_1, y) = C(x, y \pm l_2) \\ F(x, y) \geq 0, \quad F(x, y) &= F(x \pm l_1, y) = F(x, y \pm l_2) \\ G(x, y) &= G(x \pm l_1, y) = G(x, y \pm l_2) \end{aligned} \right\} \quad (6.2-3)$$

for all $(x, y) \in R_1$.

6.3 THE DIFFERENCE EQUATIONS

In chapter 3, we have formulated the difference equations for the linear, second-order PDE (3.3-1) and thus obtain the system of simultaneous linear equations (3.3.2-10) to be solved. Similarly, in this section, we shall develop the corresponding difference equations which approximate the self-adjoint second-order PDE (6.2-1) with the prescribed boundary conditions (6.2-2).

Thus, following the same technique as described in chapter 3, and at each grid point (i, j) we substitute the derivatives in (6.2-1) by their equivalent weighted difference representations, to the following approximations:

$$[A(x, y) u_x(x, y)]_x \approx \frac{A_{i+\frac{1}{2}j} (u_{i+1,j} - u_{i,j}) - A_{i-\frac{1}{2}j} (u_{i,j} - u_{i-1,j})}{h^2} \quad (6.3-1a)$$

$$B_j = \begin{bmatrix} b_{0j} & c_{0j} & & & & a_{0j} \\ a_{1j} & b_{1j} & c_{1j} & & & \\ & & \cdot & \cdot & \cdot & 0 \\ & & & \cdot & \cdot & \\ & 0 & & & \cdot & \\ & & & a_{n-2j} & b_{n-2j} & c_{n-2j} \\ c_{n-1j} & & & & a_{n-1j} & b_{n-1j} \end{bmatrix} \quad (6.3-4b)$$

and

$$\left. \begin{aligned} A_j &= \text{diag}[d_{0j}, d_{1j}, \dots, d_{n-1j}] \\ C_j &= \text{diag}[e_{0j}, e_{1j}, \dots, e_{n-1j}] \end{aligned} \right\} \quad (6.3-4c)$$

for all $j = 0, 1, \dots, m-1$. The vectors \mathbf{u} and \mathbf{s} of (3.3.2-10) are then partitioned relative to the matrix M of (6.3-4a).

6.4 SPECTRAL DECOMPOSITION METHOD

We consider the matrix equation,

$$M\mathbf{u} = \mathbf{s} \quad (6.4-1)$$

where M is the block matrix of order mn of the form,

$$M = \begin{bmatrix} T & S & & & & S \\ S & T & S & & & \\ & & \cdot & \cdot & \cdot & 0 \\ & & & \cdot & \cdot & \\ & 0 & & & \cdot & \\ & & & & S & T & S \\ S & & & & S & T \end{bmatrix} \quad (6.4-1a)$$

and the submatrices S and T are symmetric matrices of order n . We assume that S and T commute, that is,

$$TS = ST. \quad (6.4-1b)$$

The vectors \mathbf{u} and \mathbf{s} are likewise partitioned. Thus we have

$$\left. \begin{aligned} \mathbf{u} &= [\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_m^T]^T \\ \mathbf{s} &= [\mathbf{s}_1^T, \mathbf{s}_2^T, \dots, \mathbf{s}_m^T]^T \end{aligned} \right\} \quad (6.4-1c)$$

where

$$\left. \begin{aligned} \mathbf{u}_j &= [u_{1j}, u_{2j}, \dots, u_{nj}]^T \\ \mathbf{s}_j &= [s_{1j}, s_{2j}, \dots, s_{nj}]^T \end{aligned} \right\} \quad (6.4-1d)$$

for $j = 1, 2, \dots, m$.

Now T and S are commutative; therefore they have a common basis of eigenvectors.

Then, by the well-known theorem of Frobenius (see Varga[1962], Bellman[1960]) there exists an orthogonal matrix Q (i.e., $Q^T = Q^{-1}$) the columns ^{of which} are the set of eigenvectors of T and S such that,

$$\left. \begin{aligned} Q^T T Q &= \Lambda \\ Q^T S Q &= \Omega \end{aligned} \right\} \quad (6.4-2)$$

where Λ and Ω are the diagonal matrices the elements of which $\lambda_i, \omega_i; i = 1, 2, \dots, n$ are the eigenvalues of T and S respectively.

The system (6.4-1) together with (6.4-1a) and (6.4-1b), may be written as,

$$\left. \begin{aligned} T\mathbf{u}_1 + S\mathbf{u}_2 + S\mathbf{u}_m &= \mathbf{s}_1 \\ S\mathbf{u}_{j-1} + T\mathbf{u}_j + S\mathbf{u}_{j+1} &= \mathbf{s}_j ; j=1, 2, \dots, m-1 \\ S\mathbf{u}_1 + S\mathbf{u}_{m-1} + T\mathbf{u}_m &= \mathbf{s}_m \end{aligned} \right\} \quad (6.4-3)$$

By using (6.4-2), we have

$$T = Q\Lambda Q^T$$

$$S = Q\Omega Q^T$$

which upon substitution into (6.4-3), give the following equations,

$$\left. \begin{aligned} \Lambda v_1 + \Omega v_2 + \Omega v_m &= t_1 \\ \Omega v_{j-1} + \Lambda v_j + \Omega v_{j+1} &= t_j ; j=2,3,\dots,m-1 \\ \Omega v_1 + \Omega v_{m-1} + \Lambda v_m &= t_m \end{aligned} \right\} \quad (6.4-4)$$

where

$$\left. \begin{aligned} v_j &= Q^T u_j \\ t_j &= Q^T s_j \end{aligned} \right\} \quad (6.4-4a)$$

for $j = 1, 2, \dots, m$. The components of v_j and t_j ; $j = 1, 2, \dots, m$ are labelled as in (6.4-1c).

Further, (6.4-4) may be resolved by writing them, for $i=1, 2, \dots, n$, as

$$\left. \begin{aligned} \lambda_1 v_{i1} + \omega_1 v_{i2} + \omega_1 v_{im} &= t_{i1} \\ \omega_1 v_{ij-1} + \lambda_1 v_{ij} + \omega_1 v_{ij+1} &= t_{ij} ; j = 2, 3, \dots, m-1 \\ \omega_1 v_{i1} + \omega_1 v_{im-1} + \lambda_1 v_{im} &= t_{im} \end{aligned} \right\} \quad (6.4-5)$$

Now, if we write

$$T_i = \begin{bmatrix} \lambda_1 & \omega_1 & & & & & & & \omega_1 \\ \omega_1 & \lambda_1 & \omega_1 & & & & & & \\ & & & & 0 & & & & \\ & & & \cdot & \cdot & \cdot & & & \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & & \cdot & \cdot & & \\ & & 0 & & & & & & \\ & & & & & & \omega_1 & \lambda_1 & \omega_1 \\ \omega_1 & & & & & & \omega_1 & \lambda_1 & \end{bmatrix} \quad (6.4-6a)$$

where T_i is a matrix of order m and correspondingly define the vectors w_i and y_i such that

$$\left. \begin{aligned} w_i &= [v_{i1}, v_{i2}, \dots, v_{im}]^T \\ \text{and} \\ y_i &= [t_{i1}, t_{i2}, \dots, t_{im}]^T \end{aligned} \right\} \quad (6.4-6b)$$

then (6.4-5) is equivalent to the system

$$T_i w_i = y_i, \quad i=1, 2, \dots, n. \quad (6.4-7)$$

Hence, the vector w_i satisfies a symmetric tridiagonal matrix system of equations that has constant elements along the diagonal, the super- and sub-diagonals as in (6.4-6a) which can be solved efficiently (see Evans[1985]). After (6.4-7) has been solved, it is then possible to solve for $u_j = Qv_j$ for $j = 1, 2, \dots, m$ (Buzbee et al.[1970]).

The above matrix decomposition algorithm is due to Buzbee et al.[1970]. The algorithm for solving (6.4-1) proceeds as follows:

- (1) Compute or determine the eigenvectors of T and eigenvalues of T and S .
 - (2) Compute $t_j = Q^T s_j$, $j = 1, 2, \dots, m$.
 - (3) Solve $T_i w_i = y_i$, $i = 1, 2, \dots, n$.
 - (4) Compute $u_j = Qv_j$, $j = 1, 2, \dots, m$.
- (Buzbee et al.[1970]).

Hockney[1965] has analysed this algorithm for the solution of Poisson's equation in a square, where he has taken into consideration the fact that the matrix Q is known and uses the fast Fourier Transform (Cooley and Tukey[1965]) to perform steps (2) and (4). Shintani[1968] has given methods of solving for the eigenvalues and eigenvectors in several special cases.

which we shall outline in Section 6.5.1. Evans and Li[1990] develop a new direct method, called the recursive tri-reduction method for tridiagonal systems. In Section 6.5.2, we extend this method to be applied to a symmetric constant diagonal periodic linear system of equations (6.4-1). Then a stable version of this method is derived.

6.5.1 BLOCK ODD-EVEN/CYCLIC REDUCTION ALGORITHM

Consider the system of matrix equation (6.4-1) with all the assumptions about T and S and the partitioning of the matrices still being adopted. Furthermore, we assume that there are m blocks of matrices T and S along the principal diagonal of M , and $m = 2^k - 1$, for some $k \geq 2$. Such equations are known to arise in the discretization of a certain class of elliptic PDEs, using the method of separation of variables (Buzbee et al.[1970]).

Now consider a set of three consecutive neighbouring equations about u_j for $j = 2, 4, 6, \dots, m-1$. We assume that u_j and s_j are 0 for $j < 1$ and $j > mn$. Thus for (6.4-3) we obtain from the second equation, the following set of equations

$$\left. \begin{aligned} Su_{j-2} + Tu_{j-1} + Su_j &= s_{j-1} \\ Su_{j-1} + Tu_j + Su_{j+1} &= s_j \\ Su_j + Tu_{j+1} + Su_{j+2} &= s_{j+1} \end{aligned} \right\}. \quad (6.5.1-1)$$

By multiplying the first and the third equations by S , and the second by $-T$ and adding them together, we obtain the result as

$$S(1)u_{j-2} + T(1)u_j + S(1)u_{j+2} = s_j(1) \quad (6.5.1-2)$$

where

$$\left. \begin{aligned} \mathbf{s}_1(1) &= \mathbf{s}_1 - S[\mathbf{u}_2 + \mathbf{u}_{m-1}] \\ \mathbf{s}_j(1) &= \mathbf{s}_j - S[\mathbf{u}_{j-1} + \mathbf{u}_{j+1}], j=3, 5, \dots, m-2 \\ \mathbf{s}_m(1) &= \mathbf{s}_m - S[\mathbf{u}_{m-1} + \mathbf{u}_2] \end{aligned} \right\}. \quad (6.5.1-4a)$$

The system (6.5.1-3) is known as the reduced system, while that of (6.5.1-4) is known as the eliminated system. The process of reduction is called the odd-even/cyclic reduction. Since we are manipulating with blocks of matrices, it is specifically called the block odd-even/cyclic reduction.

The reduced system (6.5.1-3) has the same structure as the original system of matrix equation (6.4-1). Therefore we can repeat the reduction process to obtain another reduced and eliminated system, which will eventually terminate with only a single block of matrix equations. The unknowns can then be solved by means of Gaussian elimination and the whole set of unknowns are then obtained by back-substitution. However, as noted by Buzbee et al.[1970], we may stop the reduction process after any step and use the algorithm of section 6.4 to solve the resulting equations.

To proceed with the reduction process of (6.5.1-3), we shall first introduce the following notations.

Let for $j=1, 2, \dots, m$,

$$\left. \begin{aligned} T(0) &= T \\ S(0) &= S \\ \text{and} \\ \mathbf{s}_j(0) &= \mathbf{s}_j \end{aligned} \right\}. \quad (6.5.1-5)$$

Next define, for $j = 2, 4, 6, \dots, m-1$,

$$\left. \begin{aligned} T(1) &= 2[S(0)]^2 - [T(0)]^2 \\ S(1) &= [S(0)]^2 \\ \mathbf{s}_j(1) &= S(0)[\mathbf{s}_{j-1}(0) + \mathbf{s}_{j+1}(0)] - T(0)\mathbf{s}_j(0) \end{aligned} \right\} \quad (6.5.1-6)$$

Now, from the reduced system (6.5.1-3), consider a set of three consecutive neighbouring equations about \mathbf{u}_j for $j = 4, 8, 12, \dots, m-3$,

$$\left. \begin{aligned} S(1)\mathbf{u}_{j-4} + T(1)\mathbf{u}_{j-2} + S(1)\mathbf{u}_j &= \mathbf{s}_{j-2}(1) \\ S(1)\mathbf{u}_{j-2} + T(1)\mathbf{u}_j + S(1)\mathbf{u}_{j+2} &= \mathbf{s}_j(1) \\ S(1)\mathbf{u}_j + T(1)\mathbf{u}_{j+2} + S(1)\mathbf{u}_{j+4} &= \mathbf{s}_{j+2}(1) \end{aligned} \right\} \quad (6.5.1-7)$$

By multiplying the first and the third equations by $S(1)$, the second by $-T(1)$, and adding the three resulting equations, we have eliminated the subvectors with indices that are odd multiples of two, that is the subvectors \mathbf{u}_{j-2} and \mathbf{u}_{j+2} . Thus we obtain

$$S(2)\mathbf{u}_{j-2} + T(2)\mathbf{u}_j + S(2)\mathbf{u}_{j+2} = \mathbf{s}_j(2) \quad (6.5.1-8)$$

where

$$\left. \begin{aligned} T(2) &= 2[S(1)]^2 - [T(1)]^2 \\ S(2) &= [S(1)]^2 \\ \mathbf{s}_j(2) &= S(1)[\mathbf{s}_{j-2}(1) + \mathbf{s}_{j+2}(1)] - T(1)\mathbf{s}_j(1) \end{aligned} \right\} \quad (6.5.1-9)$$

for $j = 4, 8, 12, \dots, m-3$.

Therefore after two steps of the reduction process, we obtain the reduced system as

Now (6.5.1-16) can be solved for $u_{2^{k-1}}$. The other vectors u_j 's are solved repeatedly in a binary tree pattern using the eliminated system. While (6.5.1-16) can be solved by the LU factorization, Buzbee et al.[1970] and Lakshmivarahan and Dhall[1990] describe another method based on the factorization of the matrix $T(k-1)$.

6.5.2 THE BLOCK TRI-REDUCTION (TR3) ALGORITHM

The periodic block-tridiagonal system (6.4-1) can be written as

$$\left. \begin{aligned} Tu_1 + Su_2 + Su_m &= s_1 \\ Su_{j-1} + Tu_j + Su_{j+1} &= s_j, \quad j=2,3,\dots,m-1 \\ Su_1 + Su_{m-1} + Tu_m &= s_m \end{aligned} \right\} \quad (6.5.2-1)$$

where we assume that $m = 3^k - 1$ for some $k \geq 2$.

Consider the reduced block odd-even reduction algorithm deduced from section 6.5.1. Thus, after the first step of the odd-even reduction process, we have a set of three consecutive neighbouring blocks of equations centred around u_j as

$$\left. \begin{aligned} S(1)u_{j-3} + T(1)u_{j-1} &+ S(1)u_{j+1} &= s_{j-1}(1) \\ S(1)u_{j-1} + T(1)u_j + S(1)u_{j+1} & &= s_j \\ S(1)u_{j-1} &+ T(1)u_{j+1} + S(1)u_{j+3} &= s_{j+1}(1) \end{aligned} \right\} \quad (6.5.2-2)$$

By multiplying the first and the third equations by S , the second by $-[S(1) + T(1)]$ and adding them together, we obtain

$$\begin{aligned} SS(1)u_{j-3} - T[S(1) + T(1)]u_j + SS(1)u_{j+3} \\ = S[s_{j-1}(1) + s_{j+1}(1)] - [S(1) + T(1)]s_j. \end{aligned} \quad (6.5.2-3)$$

Now define for $j = 3, 6, 9, \dots, 3(3^{k-1}-1)$,

$$\left. \begin{aligned} Q(1) &= S^3, \quad Q(0) = S \\ P(1) &= T[T^2 - 3S^2], \quad P(0) = T \\ D_1(1) &= S[\mathbf{s}_{j-1}(1) + \mathbf{s}_{j+1}(1)] + [T^2 - 3S^2]\mathbf{s}_j \\ D_j(0) &= \mathbf{s}_j \end{aligned} \right\}. \quad (6.5.2-4)$$

We have

$$SS(1) = S^3$$

and

$$\begin{aligned} T[S(1) + T(1)] &= T[S^2 + 2S^2 - T^2] \\ &= -T[T^2 - 3S^2] \end{aligned}$$

by the definitions (6.5.1-5) and (6.5.1-6).

Hence substituting (6.5.2-4) into (6.5.2-3), we have

$$Q(1)\mathbf{u}_{j-3} + P(1)\mathbf{u}_j + Q(1)\mathbf{u}_{j+3} = D_j(1) \quad (6.5.2-5)$$

for $j = 3, 6, 9, \dots, 3(3^{k-1}-1)$. We assume that \mathbf{u}_j and D_j are $\mathbf{0}$ for $j < 1$ and $j > mn$.

We notice that, in arriving at (6.5.2-5), we have eliminated the immediate neighbouring vectors \mathbf{u}_{j-1} and \mathbf{u}_{j+1} of the vector \mathbf{u}_j . This constitutes the first step of the TR3 reduction process. The next step proceeds by splitting the system derived from the second step of the odd-even reduction algorithm into two sub-systems using the TR3 reduction algorithm as follows:

Consider a set of three consecutive blocks of equations centred around \mathbf{u}_j after the second step of the odd-even reduction process. Thus we have the following set of equations.

$$\left. \begin{aligned} S(2)\mathbf{u}_{j-6} + T(2)\mathbf{u}_{j-3} &+ S(2)\mathbf{u}_{j+3} &= \mathbf{s}_{j-3}(2) \\ S(1)\mathbf{u}_{j-3} + T(1)\mathbf{u}_j + S(1)\mathbf{u}_{j+3} & &= \mathbf{s}_j(1) \\ S(2)\mathbf{u}_{j-3} &+ T(2)\mathbf{u}_{j+3} + S(2)\mathbf{u}_{j+6} &= \mathbf{s}_{j+3}(2) \end{aligned} \right\} \quad (6.5.2-6)$$

Now multiply the first and the last equations by $S(1)$, the second by $-[S(2) + T(2)]$ and add them together, we have eliminated the subvectors \mathbf{u}_{j-3} and \mathbf{u}_{j+3} . Thus we have the resulting block equation as

$$Q(2)\mathbf{u}_{j-6} + P(2)\mathbf{u}_j + Q(2)\mathbf{u}_{j+6} = \mathbf{D}_j(2) \quad (6.5.2-7)$$

where

$$\left. \begin{aligned} Q(2) &= S(1)S(2) \\ P(2) &= T(1) \{ [T(1)]^2 - 3[S(1)]^2 \} \\ \mathbf{D}_j(2) &= S(1) [\mathbf{s}_{j-3}(2) + \mathbf{s}_{j+3}(2)] \\ &\quad + \{ [T(1)]^2 - 3[S(1)]^2 \} \mathbf{s}_j(1) \end{aligned} \right\} \quad (6.5.2-8)$$

for $j = 9, 18, 27, \dots, 9(3^{k-2}-1)$.

By continuing in this way, we have after r^{th} step of the TR3 reduction process,

$$Q(r)\mathbf{u}_{j-h} + P(r)\mathbf{u}_j + Q(r)\mathbf{u}_{j+h} = \mathbf{D}_j(r) \quad (6.5.2-9)$$

where

$$\left. \begin{aligned} Q(r) &= S(r-1)S(r) \\ P(r) &= T(r-1) \{ [T(r-1)]^2 - 3[S(r-1)]^2 \} \\ \mathbf{D}_j(r) &= S(r-1) [\mathbf{s}_{j-h}(r) + \mathbf{s}_{j+h}(r)] \\ &\quad + \{ [T(r-1)]^2 - 3[S(r-1)]^2 \} \mathbf{s}_j(r-1) \end{aligned} \right\} \quad (6.5.2-10)$$

for $j = 3^r, 2 \times 3^r, 3 \times 3^r, \dots, (3^{k-r}-1) \times 3^r$, $r = 1, 2, \dots, k$ and $h = 2 \times 3^{r-1}$.

After the r^{th} step of the TR3 reduction process, the reduced system for the TR3 algorithm becomes

$$R(r)\zeta(r) = \mathbf{w}(r), \quad (6.5.2-11)$$

where

instead of the odd-even reduction algorithm is given by the ratio

$$\eta = \frac{\log_3 N}{\log_2 N} \cdot \quad (6.5.2-16)$$

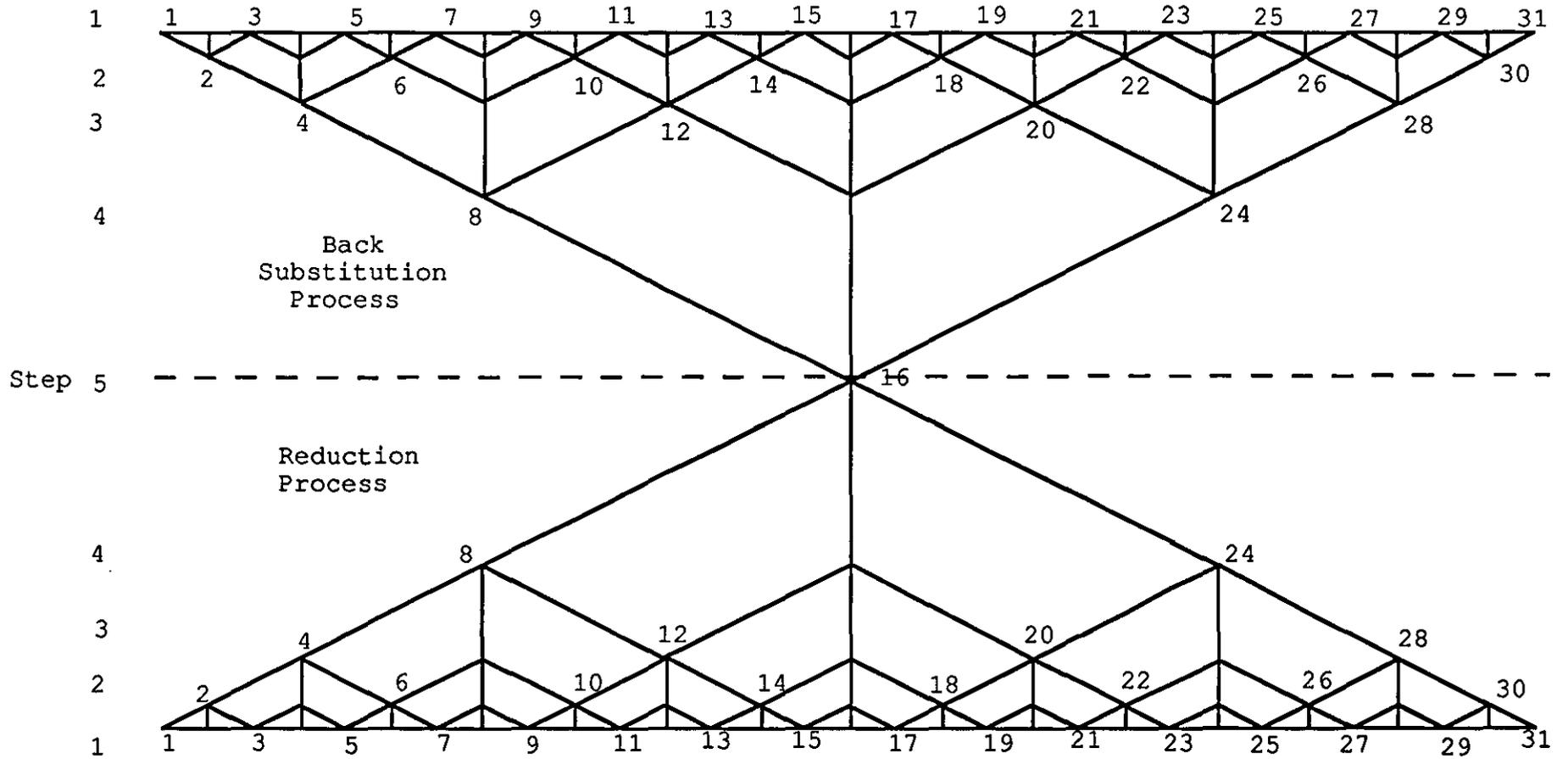
For the case illustrated in Figure(6.5.2a) and Figure(6.5.2b), the gain obtained is

$$\eta = \frac{\log_3 26}{\log_2 31} \approx 0.6.$$

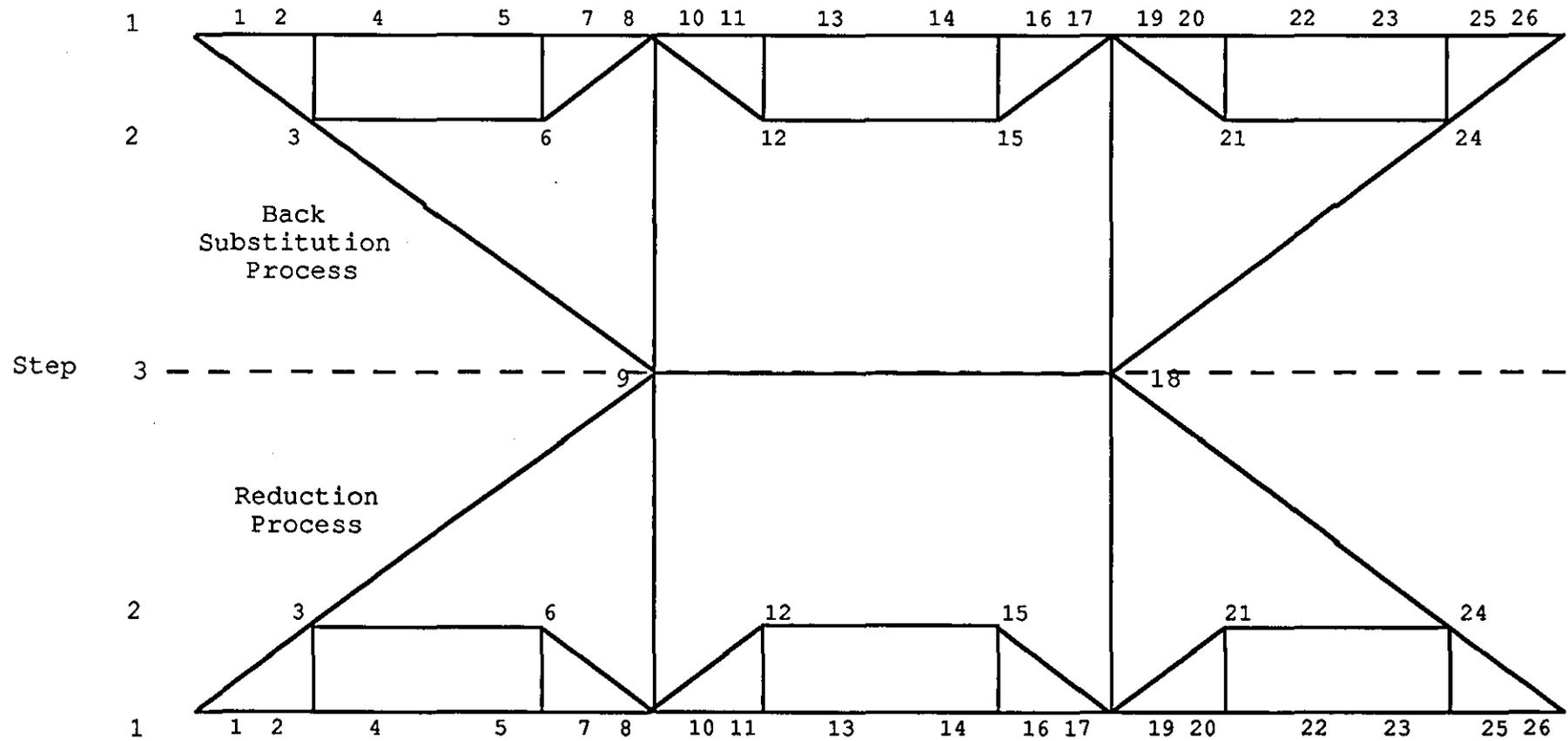
However the ratio given by (6.5.2-16) is only a comparison of the number of reduction stages. For an efficient comparison of the two algorithms, we should include a comparison of the work to be completed in the odd-even and the TR3 reduction stages.

Further illustration is given in the next section where we compare ^{the} reduction times of both algorithms. As expected, for a given matrix, the TR3 reduction algorithm spends less time in its reduction stages than the odd-even reduction algorithm. This is illustrated in Figure(6.5.3). However, the amount of work in the TR3 reduction stage is slightly more than that in the odd-even reduction stage because at each stage, the TR3 algorithm has to solve a system of 2x2 equations instead of a single equation as in the case of the odd-even algorithm.

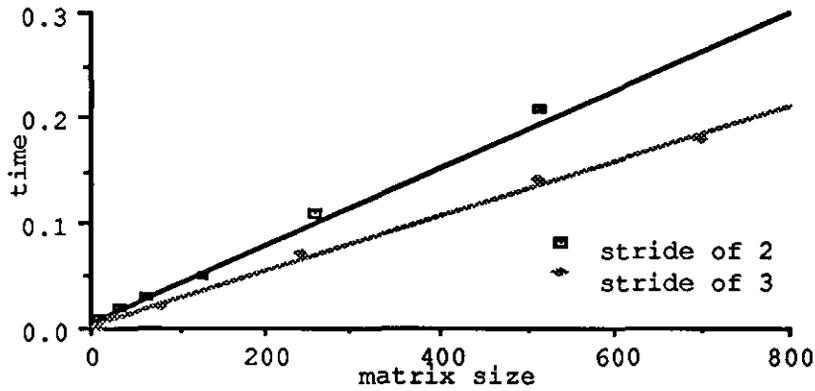
Now, as the matrix size increases, round-off errors affect the numerical solutions of the problem causing instability. This could be overcome by deriving a stable version of the algorithm. This is discussed in section 6.5.5.



Figure(6.5.2a):Odd-Even Reduction Algorithm for a matrix of order $N=2^5-1$



Figure(6.5.2b):TR3 Reduction Algorithm
for a matrix of order $N=3 \times 3 - 1$



Figure(6.5.3) : Comparison of reduction time between TR3(stride of 3) and odd-even(stride of 2) algorithms for various matrix sizes.

6.5.4 STABILITY ANALYSIS OF THE BLOCK TR3 REDUCTION ALGORITHM

In (Lakshmivarahan and Dhall[1990]), it is shown that the odd-even reduction suffers from a severe effect of round-off errors in the numerical processes. The main source of these round-off errors lies in the computation of the right-hand side of (6.5.1-2). Similarly, the TR3 reduction algorithm does not possess good control over round-off errors.

In this section, we shall analyse the stability of the block TR3 algorithm, and in the next section we shall exhibit a modified form of the block TR3 algorithm with an improved error-control property.

As is evident from the odd-even reduction case, the main source of round-off errors in the TR3 reduction algorithm also lies in the computation of the right-hand side term of the equation. Thus, in this case, we shall analyse the term $\{[T(r-1)]^2 + 3[S(r-1)]^2\} \mathbf{s}_{j(r-1)}$ of (6.5.2-9).

First, we introduce the sequence of polynomials defined by

$$p_k(a, t) = \begin{cases} -2t^k \cos(k\theta), & \text{when } |a| < |2t| \\ -2t^k \cosh(k\phi), & \text{when } |a| > |2t| \end{cases} \quad (6.5.4-1)$$

with initial conditions

$$\left. \begin{aligned} p_0(a, t) &= -2p_1(a, t) = -2t \cos\theta \\ p_2(a, t) &= -a^2 + 2t^2. \end{aligned} \right\} \quad (6.5.4-1a)$$

Let

$$p_3(a, t) = a(a^2 - 3t^2). \quad (6.5.4-1b)$$

Then for $t \neq 0$, we use the transformation

$$a = \begin{cases} -2t \cos\theta, & \text{when } |a| < |2t| \\ -2t \cosh\phi, & \text{when } |a| > |2t| \end{cases}. \quad (6.5.4-2)$$

It is evident that by a short manipulation, the polynomial can be recursively computed as follows

$$p_k(a, t) = -ap_{k-1}(a, t) - t^2 p_{k-2}(a, t). \quad (6.5.4-3)$$

Now hyperbolic functions satisfy identities which are very similar to those satisfied by trigonometric functions. Therefore, we shall only discuss the conditions of the case related to the trigonometric functions; that is, the case when $|a| < |2t|$. The case when $|a| > |2t|$ can then be similarly deduced.

The case when $|a| < |2t|$.

By substituting $a = -2t \cos\theta$ into (6.5.4-1b), we obtain

$$p_3(a, t) = -2t^3 \cos\theta [4\cos^2\theta - 3]. \quad (6.5.4-4)$$

Consequently, for $p_3(a, t) = 0$, we obtain the conditions that either $\cos\theta = 0$ or $\cos\theta = \pm \frac{\sqrt{3}}{2}$, that is $\theta = \frac{\pi}{2}, \frac{\pi}{6},$ or $\frac{5\pi}{6}$.

Hence, we have

$$\frac{a}{2t} = -\cos \left[\frac{2j-1}{2 \times 3^{r-1}} \right] \pi, \quad (6.5.4-5)$$

for $j=1, 2, 3, \dots, 3^{r-1}$ and $r \geq 2$.

Therefore the general sequence of polynomials may be factorized as,

$$p_{3^{r-1}}(a, t) = \prod_{j=1}^{3^{r-1}} \left\{ a + 2t \cos \left[\frac{2j-1}{2 \times 3^{r-1}} \right] \pi \right\}, \quad (6.5.4-6)$$

for $r \geq 2$.

Now we can use this recursive definition to compute the term

$$\{ [T(r-1)]^2 + 3[S(r-1)]^2 \} \mathbf{s}_j(r-1) = p_{3^{r-1}}(T, S) \mathbf{s}_j(r-1) \quad (6.5.4-7)$$

as follows.

Let

$$\left. \begin{aligned} \eta_0 &= -2\mathbf{s}_j(r-1) \\ \eta_1 &= T\mathbf{s}_j(r-1), \quad \eta_2 = T(1)\mathbf{s}_j(r-1) \\ \eta_k &= -T\eta_{k-1} - S^2\eta_{k-2} \end{aligned} \right\} \quad (6.5.4-8)$$

for $k = 3, 4, \dots, 3^{r-1}$ and $r \geq 2$.

Therefore we obtain

$$\begin{aligned} \eta_{3^{r-1}} &= p_{3^{r-1}}(T, S) \mathbf{s}_j(r-1) \\ &= \{ [T(r-1)]^2 + 3[S(r-1)]^2 \} \mathbf{s}_j(r-1). \end{aligned} \quad (6.5.4-9)$$

However, due to the presence of round-off errors, the true computed sequence is given by

$$\left. \begin{aligned} \zeta_0 &= \eta_0 \\ \zeta_1 &= T\mathbf{s}_j(r-1) + \delta_0, \quad \zeta_2 = T(1)\mathbf{s}_j(r-1) + \delta_1 \\ \zeta_k &= -T\zeta_{k-1} - S^2\zeta_{k-2} + \delta_{k-1} \end{aligned} \right\} \quad (6.5.4-10)$$

where δ_i ; $i = 0, 1, 2, \dots, k-1$ is the perturbation induced by the round-off errors.

Now, the matrices T and S are symmetric. Furthermore, they are commutative. Therefore from the well-known theory of matrices (see Bellman[1960] or Varga[1962]), there exists an orthogonal matrix Q which diagonalizes T and S simultaneously. Hence we obtain

$$\left. \begin{aligned} T &= Q\Lambda Q^T \\ S &= Q\Omega Q^T \end{aligned} \right\} \quad (6.5.4-11)$$

where

$$\left. \begin{aligned} \Lambda &= \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n] \\ \Omega &= \text{diag}[\omega_1, \omega_2, \dots, \omega_n] \end{aligned} \right\} \quad (6.5.4-12)$$

are diagonal matrices of eigenvalues of T and S , respectively.

We may assume that the eigenvalues of T and S are distinct and that the columns of Q are normalized eigenvectors of T and S .

We also know that $Q^T = Q^{-1}$ and $QS^2Q^T = \Omega^2$ where $\Omega^2 = \text{diag}[\omega_1^2, \omega_2^2, \dots, \omega_n^2]$.

Therefore on combining these with (6.5.4-10), (6.5.4-11) and (6.5.4-12), we have

$$\left. \begin{aligned} \rho_0 &= -2\mathbf{g}_j(r-1) \\ \rho_1 &= -\frac{1}{2} \Lambda \rho_0 + \Psi_0 \\ \rho_k &= -\Lambda \rho_{k-1} - \Omega^2 \rho_{k-2} + \Psi_{k-1} \end{aligned} \right\} \quad (6.5.4-13)$$

where

$$\left. \begin{aligned} \rho_k &= Q^T \zeta_k \\ \mathbf{g}_j(r-1) &= Q^T \mathbf{s}_j(r-1) \\ \psi_k &= Q^T \delta_k \end{aligned} \right\}. \quad (6.5.4-14)$$

Now Λ and Ω are diagonal matrices as given in (6.5.4-12). Therefore we can write (6.5.4-13) component-wise as

$$\rho_{jk+1} + \lambda_j \rho_{jk} + \omega_j^2 \rho_{jk-1} = \psi_{jk}, \quad (6.5.4-15)$$

for $j = 1, 2, \dots, n$.

The solution of the recursive relation (6.5.4-15) may be obtained by considering its characteristic equation

$$v^2 + \lambda_j v + \omega_j^2 = 0. \quad (6.5.4-16)$$

Next, we write the two roots of (6.5.4-16) as

$$\mu_j = \frac{[-\lambda_j + \sqrt{(\lambda_j^2 - 4\omega_j^2)}]}{2} \quad \text{and} \quad \nu_j = \frac{[-\lambda_j - \sqrt{(\lambda_j^2 - 4\omega_j^2)}]}{2}.$$

Now there are two cases to be considered.

Case (1): $\mu_j = \nu_j$.

It can be shown from first principles that

$$\begin{aligned} \rho_{jk} &= \left[\frac{\mu_j^k - \nu_j^k}{\mu_j - \nu_j} \right] \rho_{j1} - \mu_j \nu_j \left[\frac{\mu_j^{k-1} - \nu_j^{k-1}}{\mu_j - \nu_j} \right] \rho_{j0} \\ &\quad + \sum_{t=1}^{k-1} \left[\frac{\mu_j^{k-t} - \nu_j^{k-t}}{\mu_j - \nu_j} \right] \psi_{jt}. \end{aligned} \quad (6.5.4-17)$$

From (6.5.4-13) we have

$$\left. \begin{aligned} \rho_{j0} &= -2\mathbf{g}_{j0}(r-1) \\ \rho_{j1} &= -\frac{1}{2}\lambda_j \rho_{j0} + \psi_{j0} \end{aligned} \right\}. \quad (6.5.4-18)$$

Therefore on substituting (6.5.4-18) into (6.5.4-17), we obtain

$$\begin{aligned} \rho_{jk} = & \left[\frac{\mu_j^k - v_j^k}{\mu_j - v_j} \right] [\lambda_j g_{j0}^{(r-1)} + \psi_{j0}] \\ & + 2\mu_j v_j \left[\frac{\mu_j^{k-1} - v_j^{k-1}}{\mu_j - v_j} \right] g_{j0}^{(r-1)} + \sum_{t=1}^{k-1} \left[\frac{\mu_j^{k-t} - v_j^{k-t}}{\mu_j - v_j} \right] \psi_{jt} \end{aligned}$$

which on simplifying, becomes

$$\begin{aligned} \rho_{jk} = & \left\{ \left[\frac{\mu_j^k - v_j^k}{\mu_j - v_j} \right] \lambda_j + 2\mu_j v_j \left[\frac{\mu_j^{k-1} - v_j^{k-1}}{\mu_j - v_j} \right] \right\} g_{j0}^{(r-1)} \\ & + \sum_{t=0}^{k-1} \left[\frac{\mu_j^{k-t} - v_j^{k-t}}{\mu_j - v_j} \right] \psi_{jt}. \end{aligned} \quad (6.5.4-19)$$

Next, from the expressions for μ_j and v_j , it follows that μ_j and v_j are complex conjugates if $|\lambda_j| < |2\omega_j|$ and real if $|\lambda_j| > |2\omega_j|$.

Let

$$\frac{\lambda_j}{2\omega_j} = \begin{cases} \cos\theta_j, & \text{if } |\lambda_j| < |2\omega_j| \\ \cosh\phi_j, & \text{if } |\lambda_j| > |2\omega_j|. \end{cases} \quad (6.5.4-20)$$

Now there are two subcases of case (1) to be considered.

Case (1a): $|\lambda_j| < |2\omega_j|$ and the roots μ_j and v_j are complex conjugates.

Thus $\mu_j \neq v_j$ and

$$\left. \begin{aligned} \mu_j &= \omega_j (\cos\theta_j + i\sin\theta_j) = \omega_j e^{i\theta_j} \\ v_j &= \omega_j (\cos\theta_j - i\sin\theta_j) = \omega_j e^{-i\theta_j} \end{aligned} \right\} \quad (6.5.4-21)$$

where $i = \sqrt{-1}$.

By substituting (6.5.4-21) into (6.5.4-19), we obtain

$$\begin{aligned} \rho_{jk} = & \left\{ \omega_j^{k-1} \lambda_j \left[\frac{e^{ik\theta_j} - e^{-ik\theta_j}}{e^{i\theta_j} - e^{-i\theta_j}} \right] \right. \\ & + 2\omega_j^k \left[\frac{e^{i(k-1)\theta_j} - e^{-i(k-1)\theta_j}}{e^{i\theta_j} - e^{-i\theta_j}} \right] \left. \right\} g_{j0}^{(r-1)} \\ & + \sum_{t=0}^{k-1} \left\{ \omega_j^{k-t-1} \left[\frac{e^{i(k-t)\theta_j} - e^{-i(k-t)\theta_j}}{e^{i\theta_j} - e^{-i\theta_j}} \right] \psi_{jt} \right\}. \end{aligned}$$

Thus we have

$$\begin{aligned} \rho_{jk} = & \left\{ \omega_j^{k-1} \lambda_j \left[\frac{\sin(k\theta_j)}{\sin\theta_j} \right] + 2\omega_j^k \left[\frac{\sin((k-1)\theta_j)}{\sin\theta_j} \right] \right\} g_{j0}^{(r-1)} \\ & + \sum_{t=0}^{k-1} \left\{ \omega_j^{k-t-1} \left[\frac{\sin((k-t)\theta_j)}{\sin\theta_j} \right] \psi_{jt} \right\}. \quad (6.5.4-22) \end{aligned}$$

When $|\lambda_j| < |2\omega_j|$, we have from (6.5.4-20) $\lambda_j = -2\omega_j \cos\theta_j$. Therefore (6.5.4-22) can be simplified to obtain

$$\begin{aligned} \rho_{jk} = & -2g_{j0}^{(r-1)} \omega_j^k \cos(k\theta_j) + \sum_{t=0}^{k-1} \left[\omega_j^{k-t-1} \frac{\sin((k-t)\theta_j)}{\sin\theta_j} \psi_{jt} \right]. \quad (6.5.4-23) \end{aligned}$$

Case (1b): $|\lambda_j| > |2\omega_j|$ and the roots μ_j and ν_j are real and distinct.

Therefore we can write

$$\left. \begin{aligned} \mu_j &= \omega_j (\cosh\phi_j + \sinh\phi_j) = \omega_j e^{\phi_j} \\ \nu_j &= \omega_j (\cosh\phi_j - \sinh\phi_j) = \omega_j e^{-\phi_j} \end{aligned} \right\}. \quad (6.5.4-24)$$

By substituting (6.5.4-24) into (6.5.4-19) and simplifying, we obtain

$$\begin{aligned} \rho_{jk} = & \left\{ \omega_j^{k-1} \lambda_j \left[\frac{\sinh(k\phi_j)}{\sinh\phi_j} \right] + 2\omega_j^k \left[\frac{\sinh((k-1)\phi_j)}{\sinh\phi_j} \right] \right\} g_{j0}^{(r-1)} \\ & + \sum_{t=0}^{k-1} \left\{ \omega_j^{k-t-1} \left[\frac{\sinh((k-t)\phi_j)}{\sinh\phi_j} \right] \psi_{jt} \right\}. \quad (6.5.4-25) \end{aligned}$$

From (6.5.4-20), since $|\lambda_j| > |2\omega_j|$, we have $\lambda_j = -2\omega_j \cosh \phi_j$. Therefore by combining this with (6.5.4-22), we obtain

$$\begin{aligned} \rho_{jk} &= -2g_{j0}^{(r-1)} \omega_j^k \cosh(k\phi_j) \\ &+ \sum_{t=0}^{k-1} \left[\omega_j^{k-t-1} \frac{\sinh((k-t)\phi_j)}{\sinh\phi_j} \psi_{jt} \right]. \end{aligned} \quad (6.5.4-26)$$

Now, let $p_k(\Lambda, \Omega)$ be a diagonal matrix such that

$$[p_k(\Lambda, \Omega)]_{j1} = \begin{cases} p_k(\lambda_j, \omega_j), & \text{for } j = 1 \\ 0, & \text{otherwise} \end{cases} \quad (6.5.4-27)$$

where from (6.5.4-1), we have

$$p_k(\lambda_j, \omega_j) = -2\omega_j^k \begin{cases} \cos(k\theta_j), & \text{if } |\lambda_j| < |2\omega_j| \\ \cosh(k\phi_j), & \text{if } |\lambda_j| > |2\omega_j|. \end{cases} \quad (6.5.4-28)$$

From (6.5.4-11), we obtain

$$p_k(T, S) = Q p_k(\Lambda, \Omega) Q^T. \quad (6.5.4-29)$$

Similarly, we define a diagonal matrix $R(q)$ such that

$$[R(q)]_{j1} = \begin{cases} \omega_j^{q-1} \begin{cases} \frac{\sin(q\theta_j)}{\sin\theta_j}, & \text{for } j = 1 \text{ and } |\lambda_j| < |2\omega_j| \\ \frac{\sinh(q\phi_j)}{\sinh\phi_j}, & \text{for } j = 1 \text{ and } |\lambda_j| > |2\omega_j| \end{cases} \\ 0, & \text{otherwise.} \end{cases} \quad (6.5.4-30)$$

Now, by using (6.5.4-27) and (6.5.4-30), ρ_k can be expressed in matrix-vector form as

$$\rho_k = p_k(\Lambda, \Omega) g_j^{(r-1)} + \sum_{t=0}^{k-1} R(k-t) \psi_t. \quad (6.5.4-31)$$

Hence, by using (6.5.4-27) and (6.5.4-29); we can rewrite (6.5.4-31) to give

$$\zeta_k = Q\rho_k = p_k(T, S) \mathbf{s}_j(r-1) + \sum_{t=0}^{k-1} QR(k-t) Q^T \delta_t. \quad (6.5.4-32)$$

Therefore, in the case $\left| \frac{\lambda_j}{2\omega_j} \right| > 1$; if $\phi_n > \phi_1$, then it is evident that for large k , ρ_{nk} may become very large relative to ρ_{1k} . Since $\zeta_k = Q\rho_k$, the effect of ρ_{1k} will be insignificant due to the round-off errors.

Case (2): $\mu_j = v_j = -\frac{\lambda_j}{2}$.

This happens only if $\lambda_j^2 = 4\omega_j^2$. It can be shown from first principles that

$$\rho_{jk} = k\mu_j^{k-1} \rho_{j1} - (k-1)\mu_j^k \rho_{j0} + \sum_{t=1}^{k-1} [(k-t)\mu_j^{k-t-1} \psi_{jt}]. \quad (6.5.4-33)$$

By substituting $\mu_j = v_j = -\frac{\lambda_j}{2}$ and (6.5.4-13) into (6.5.4-33), we have

$$\begin{aligned} \rho_{jk} = & k \left[-\frac{\lambda_j}{2} \right]^{k-1} \left[\lambda_j g_{j0}(r-1) + \psi_{j0} \right] \\ & - (k-1) \left[-\frac{\lambda_j}{2} \right]^k \left[-2g_{j0}(r-1) \right] \\ & + \sum_{t=1}^{k-1} \left\{ (k-t) \left[-\frac{\lambda_j}{2} \right]^{k-t-1} \psi_{jt} \right\}, \end{aligned}$$

i.e.

$$\rho_{jk} = -2 \left[-\frac{\lambda_j}{2} \right]^k g_{j0}(r-1) + \sum_{t=0}^{k-1} \left\{ (k-t) \left[-\frac{\lambda_j}{2} \right]^{k-t-1} \psi_{jt} \right\}. \quad (6.5.4-34)$$

Now, let $p_k(\Lambda, \Omega)$ be a diagonal matrix defined as

$$[p_k(\Lambda, \Omega)]_{jj} = \begin{cases} p_k(\lambda_j, \omega_j), & \text{for } j = 1 \\ 0, & \text{otherwise} \end{cases} \quad (6.5.4-35)$$

where

$$p_k(\lambda_j, \omega_j) = -2 \left[-\frac{\lambda_j}{2} \right]^k. \quad (6.5.4-36)$$

From (6.5.4-11), we have

$$p_k(T, S) = Q p_k(\Lambda, \Omega) Q^T. \quad (6.5.4-37)$$

Similarly, we define a diagonal matrix $R(q)$ such that

$$[R(q)]_{j_1}^{j_1} = \begin{cases} q \left[-\frac{\lambda_j}{2} \right]^q, & \text{for } j = 1 \\ 0, & \text{otherwise.} \end{cases} \quad (6.5.4-38)$$

Therefore, by using (6.5.4-35) and (6.5.4-38), we may express p_k of (6.5.4-34) in matrix-vector form as

$$p_k = p_k(\Lambda, \Omega) g_{j(r-1)} + \sum_{t=0}^{k-1} R(k-t) \psi_t. \quad (6.5.4-39)$$

Now, by using (6.5.4-35) and (6.5.4-37), we can rewrite (6.5.4-39) in the form

$$\zeta_k = Q p_k = p_k(T, S) s_{j(r-1)} + \sum_{t=0}^{k-1} [Q R(k-t) Q^T \delta_t]. \quad (6.5.4-40)$$

In the case $\left| \frac{\lambda_j}{2\omega_j} \right| = 1$, on combining (6.5.4-34) with $\lambda_j = -2\omega_j$ and simplifying, we obtain

$$p_{jk} = -2g_{j\rho(r-1)} \omega_j^k + \sum_{t=0}^{k-1} [(k-t) \omega_j^{k-t-1} \psi_{jt}]. \quad (6.5.4-41)$$

Therefore we have

$$p_{1k} = -2g_{j\rho(r-1)} \omega_1^k + \sum_{t=0}^{k-1} [(k-t) \omega_1^{k-t-1} \psi_{1t}] \quad (6.5.4-41a)$$

and

$$p_{nk} = -2g_{n\rho(r-1)} \omega_n^k + \sum_{t=0}^{k-1} [(k-t) \omega_n^{k-t-1} \psi_{nt}] \quad (6.5.4-41b)$$

where ω_j ; $j = 1, 2, \dots, n$ are the eigenvalues of S .

Since $\zeta_k = Q p_k$, therefore it is evident that the vector ζ_k will decay as k increases if the moduli of all the eigenvalues ω_j ; $j = 1, 2, \dots, n$ are less than unity and hence will remain bounded.

6.5.5 STABLE VERSION OF THE TR3 REDUCTION ALGORITHM

It has been shown in the previous section that the TR3 reduction algorithm is numerically unstable. In this section we shall derive a stable version of the TR3 reduction algorithm. Basically, this is done by modifying the computation of the right-hand side term.

From (6.5.2-4), we note that $P(1)$ is a polynomial of degree 3 in T and S . By induction, it is easy to show that $P(k)$ is a polynomial of degree 3^k in the matrices T and S .

Therefore we obtain the factorization for $P(1)$ as

$$P(1) = \prod_{j=1}^3 \left\{ T + 2S \cos \left[\frac{2j-1}{2 \times 3} \right] \pi \right\}. \quad (6.5.5-1)$$

Thus we obtain

$$\begin{aligned} P(1) &= T(T + \sqrt{3} S)(T - \sqrt{3} S) \\ &= T(T^2 - 3S^2) \end{aligned} \quad (6.5.5-1a)$$

since, by (6.4-1b), T and S are commutative.

Now by using (6.5.4-6) we obtain the factorization of $P(k-1)$ as

$$P(k-1) = \prod_{j=1}^{3^{k-1}} H_j(k-1), \quad (6.5.5-2)$$

where the matrix $H_j(k-1)$ is defined by

$$H_j(k-1) = [T + 2S \cos \theta_j(k-1)] \quad (6.5.5-2a)$$

and

$$\theta_j(k-1) = \left[\frac{2j-1}{2 \times 3^{k-1}} \right] \pi, \quad (6.5.5-2b)$$

for $j = 1, 2, 3, \dots, 3^{k-1}$ and $k \geq 2$.

Therefore the single block matrix equation (6.5.2-14) can now be solved as follows.

Substitute (6.5.5-2) into (6.5.2-14); we then obtain

$$[H_1^{(k-1)}H_2^{(k-1)}\dots H_{3^{k-1}}^{(k-1)}]u_{3^{k-1}} = D_{3^{k-1}}^{(k-1)}. \quad (6.5.5-3)$$

Define

$$z_0 = D_{3^{k-1}}^{(k-1)},$$

and for $j = 1, 2, \dots, 3^{k-1}$, we solve repeatedly for z_j using

$$H_j^{(k-1)}z_j = z_{j-1}. \quad (6.5.5-4)$$

Clearly, we have

$$u_{3^{k-1}} = z_{3^{k-1}}. \quad (6.5.5-5)$$

Similarly, by using the same factorization we can solve the eliminated system (6.5.2-12), since the matrix $E(r)$ depends on $P(r-1)$ only. This solution process is motivated by Lakshmivarahan and Dhall[1990], and Buzbee et al.[1970].

As already stated in the previous section, the main source of round-off error lies in the computation of the right-hand side term. Therefore, in order to induce stability, we shall reorganize the computation of the right-hand side term. This idea was first motivated by Buneman[1969], then followed by Buzbee et al.[1970] and Lakshmivarahan and Dhall[1990]. Henceforth, we shall consider the special case of (6.4-1a), when $S = I$, the identity matrix of order n .

We shall now consider the reorganization of the right-hand side of (6.5.2-5) so as to obtain a stable version of the TR3 reduction algorithm.

Now, after the first step of the TR3 reduction stage, we obtain from (6.5.2-5),

$$\mathbf{u}_{j-3} + P(1)\mathbf{u}_j + \mathbf{u}_{j+3} = \mathbf{D}_j(1), \quad (6.5.5-6)$$

where

$$P(1) = -T[T^2 - 3I] \quad (6.5.5-6a)$$

$$\mathbf{D}_j(1) = \mathbf{s}_{j-1}(1) + \mathbf{s}_{j+1}(1) + [T^2 - 3I]\mathbf{s}_j \quad (6.5.5-6b)$$

for $j = 3, 6, 9, \dots, 3(3^{k-1}-1)$.

From the odd-even reduction algorithm, we have the definition,

$$\alpha_j(0) = 0, \quad \beta_j(0) = \mathbf{s}_j \quad (6.5.5-7a)$$

$$\left. \begin{aligned} \alpha_j(r+1) &= \alpha_j(r) - [T(r)]^{-1}[\alpha_{j-h}(r) + \alpha_{j+h}(r) - \beta_j(r)] \\ \beta_j(r+1) &= \beta_{j-h}(r) + \beta_{j+h}(r) - 2\alpha_j(r+1) \end{aligned} \right\} \quad (6.5.5-7b)$$

and

$$\mathbf{s}_j(r) = T(r)\alpha_j(r) + \beta_j(r). \quad (6.5.5-7c)$$

Therefore we shall now write (6.5.5-6b) in the form

$$\begin{aligned} \mathbf{D}_j(1) &= T(1)\alpha_{j-1}(1) + \beta_{j-1}(1) + T(1)\alpha_{j+1}(1) + \beta_{j+1}(1) \\ &\quad + T[T^2 - 3I]T^{-1}\mathbf{s}_j, \\ &= T(1)[\alpha_{j-1}(1) + \alpha_{j+1}(1)] + \beta_{j-1}(1) + \beta_{j+1}(1) \\ &\quad + T[T^2 - 3I]T^{-1}\mathbf{s}_j. \end{aligned} \quad (6.5.5-8)$$

By adding and subtracting $-\alpha_{j-1}(1) + \alpha_{j+1}(1)$, we obtain

$$\begin{aligned} \mathbf{D}_j(1) &= P(1)T^{-1}\mathbf{s}_j + [T(1) + I][\alpha_{j-1}(1) + \alpha_{j+1}(1)] \\ &\quad + \beta_{j-1}(1) + \beta_{j+1}(1) - [\alpha_{j-1}(1) + \alpha_{j+1}(1)], \\ &= P(1)T^{-1}\{\mathbf{s}_j - [\alpha_{j-1}(1) + \alpha_{j+1}(1)]\} \\ &\quad + \beta_{j-1}(1) + \beta_{j+1}(1) - [\alpha_{j-1}(1) + \alpha_{j+1}(1)]. \end{aligned} \quad (6.5.5-9)$$

Hence, we write (6.5.5-9) in the form

$$\mathbf{D}_j(1) = \mathbf{P}(1)\mathbf{v}_j(1) + \boldsymbol{\kappa}_j(1) \quad (6.5.5-10)$$

where

$$\mathbf{v}_j(1) = \mathbf{T}^{-1} \{ \mathbf{s}_j - [\boldsymbol{\alpha}_{j-1}(1) + \boldsymbol{\alpha}_{j+1}(1)] \} \quad (6.5.5-10a)$$

$$\boldsymbol{\kappa}_j(1) = \boldsymbol{\beta}_{j-1}(1) + \boldsymbol{\beta}_{j+1}(1) - [\boldsymbol{\alpha}_{j-1}(1) + \boldsymbol{\alpha}_{j+1}(1)] \quad (6.5.5-10b)$$

for $j = 3, 6, 9, \dots, 3(3^{k-1}-1)$.

Next consider

$$\mathbf{D}_j(2) = \mathbf{s}_{j-3}(2) + \mathbf{s}_{j+3}(2) + \{ [\mathbf{T}(1)]^2 - 3\mathbf{I} \} \mathbf{s}_j(1) \quad (6.5.5-11)$$

which upon using (6.5.5-7c) can be written as

$$\begin{aligned} \mathbf{D}_j(2) &= \mathbf{T}(2) [\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)] + \boldsymbol{\beta}_{j-3}(2) + \boldsymbol{\beta}_{j+3}(2) \\ &\quad + \{ [\mathbf{T}(1)]^2 - 3\mathbf{I} \} \{ \mathbf{T}(1)\boldsymbol{\alpha}_j(1) + \boldsymbol{\beta}_j(1) \}, \\ &= \mathbf{P}(2)\boldsymbol{\alpha}_j(1) + \mathbf{T}(2) [\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)] \\ &\quad + \{ [\mathbf{T}(1)]^2 - 3\mathbf{I} \} \boldsymbol{\beta}_j(1) + \boldsymbol{\beta}_{j-3}(2) + \boldsymbol{\beta}_{j+3}(2), \\ &= \mathbf{P}(2) \{ \boldsymbol{\alpha}_j(1) + [\mathbf{T}(1)]^{-1}\boldsymbol{\beta}_j(1) \} \\ &\quad + \mathbf{T}(2) [\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)] \\ &\quad + \boldsymbol{\beta}_{j-3}(2) + \boldsymbol{\beta}_{j+3}(2). \end{aligned} \quad (6.5.5-12)$$

Now by adding and subtracting $[\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)]$, we obtain

$$\begin{aligned} \mathbf{D}_j(2) &= \mathbf{P}(2) \{ \boldsymbol{\alpha}_j(1) + [\mathbf{T}(1)]^{-1}\boldsymbol{\beta}_j(1) \} \\ &\quad + [\mathbf{T}(2) + \mathbf{I}] [\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)] \\ &\quad + \boldsymbol{\beta}_{j-3}(2) + \boldsymbol{\beta}_{j+3}(2) - [\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)]. \end{aligned}$$

Hence, we obtain

$$\mathbf{D}_j(2) = \mathbf{P}(2)\mathbf{v}_j(2) + \boldsymbol{\kappa}_j(2) \quad (6.5.5-13)$$

where

$$\mathbf{v}_j(2) = \boldsymbol{\alpha}_j(1) - [\mathbf{T}(1)]^{-1} \{ \boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2) - \boldsymbol{\beta}_j(1) \} \quad (6.5.5-13a)$$

$$\boldsymbol{\kappa}_j(2) = \boldsymbol{\beta}_{j-3}(2) + \boldsymbol{\beta}_{j+3}(2) - [\boldsymbol{\alpha}_{j-3}(2) + \boldsymbol{\alpha}_{j+3}(2)] \quad (6.5.5-13b)$$

for $j = 9, 18, 27, \dots, 9(3^{k-2}-1)$.

Thus inductively, after $r+1$ steps for $r = 0, 1, 2, \dots$, of the reduction stages, we obtain the following results:

$$D_j(r+1) = P(r+1)U_j(r+1) + K_j(r+1) \quad (6.5.5-14)$$

where

$$U_j(r+1) = \alpha_j(r) - [T(r)]^{-1} \{ \alpha_{j-h}(r+1) + \alpha_{j+h}(r+1) - \beta_j(r) \} \quad (6.5.5-14a)$$

$$K_j(r+1) = \beta_{j-h}(r+1) + \beta_{j+h}(r+1) - [\alpha_{j-h}(r+1) + \alpha_{j+h}(r+1)] \quad (6.5.5-14b)$$

for $j = 3^{r+1}, 2 \times 3^{r+1}, 3 \times 3^{r+1}, \dots, (3^{k-r-1} - 1) \times 3^{r+1}$; $r = 0, 1, 2, \dots$ and $h = 2 \times 3^{r-1}$.

Now (6.5.5-14a) can be rewritten in the form

$$T(r) [\alpha_j(r) - U_j(r+1)] = \alpha_{j-h}(r+1) + \alpha_{j+h}(r+1) - \beta_j(r). \quad (6.5.5-15)$$

Since the $\alpha_j(r)$, $\alpha_{j \pm h}(r+1)$ and $\beta_j(r)$ are known quantities, then $U_j(r+1)$ may be obtained by solving (6.5.5-15).

It can be shown that the following results hold for the odd-even reduction stage

$$T(r) [\alpha_j(r) - \alpha_j(r+1)] = \alpha_{j-h}(r) + \alpha_{j+h}(r) - \beta_j(r) \quad (6.5.5-16)$$

and $T(r)$ is factorized as

$$T(r) = -\prod_{j=1}^{2^r} J_j(r) \quad (6.5.5-17)$$

where the matrix

$$J(r) = [T + 2I \cos \bar{\theta}_j(r)] \quad (6.5.5-17a)$$

and

$$\bar{\theta}_j(r) = \frac{2j-1}{2^{r+1}} \pi \quad (6.5.5-17b)$$

(see Lakshmivarahan and Dhall[1990]), where the information can be used to solve for $\alpha_j(r+1)$ prior to solving the $v_j(r+1)$.

After $k-1$ steps of the TR3 reduction stage, we obtain a reduced system of the form given by (6.5.2-13),

$$P(k-1)u_{3^{k-1}} = P(k-1)v_{3^{k-1}}(k-1) + K_{3^{k-1}}(k-1) \quad (6.5.5-18)$$

by using (6.5.5-14) and since $m=3^k-1$.

Therefore we obtain the solution

$$u_{3^{k-1}} = v_{3^{k-1}}(k-1) + [P(k-1)]^{-1}K_{3^{k-1}}(k-1) \quad (6.5.5-19)$$

where we can substitute the factorization of $P(k-1)$ as given in (6.5.5-2) to compute the second term of the right-hand side of (6.5.5-19).

The other solution vectors of (6.4-1) can be computed as follows.

Consider the equations (6.5.2-8) and (6.5.5-14); thus for the appropriate r , we have

$$u_{j-h} + P(r)u_j + u_{j+h} = P(r)v_j(r) + K_j(r) \quad (6.5.5-20)$$

from which we deduce

$$P(r)[u_j - v_j(r)] = K_j(r) - [u_{j-h} + u_{j+h}] \quad (6.5.5-21)$$

for $j = 3^r, 2 \times 3^r, 4 \times 3^r, 5 \times 3^r, \dots, (3^{k-r}-1) \times 3^r$; $r = k-2, k-3, \dots, 2, 1, 0$ and $h = 2 \times 3^{r-1}$.

Therefore (6.5.5-21) can be solved for u_j ; $j = 3^r, 2 \times 3^r, 4 \times 3^r, 5 \times 3^r, \dots, (3^{k-r}-1) \times 3^r$ and $r = k-2, k-3, \dots, 2, 1, 0$ using the factorization of $P(r)$ given in (6.5.5-2).

Next we express the $v_j(r)$ and $K_j(r)$ in terms of the u_j as follows.

Consider the original governing equation,

$$\mathbf{u}_{j-1} + T\mathbf{u}_j + \mathbf{u}_{j+1} = \mathbf{s}_j. \quad (6.5.5-22)$$

From (6.5.5-10a), we can write

$$\mathbf{s}_j = T\mathbf{v}_j(1) + \alpha_{j-1}(1) + \alpha_{j+1}(1). \quad (6.5.5-23)$$

Now, it can be shown that from the odd-even reduction stage, we have

$$\alpha_j(1) = \mathbf{u}_j + T^{-1}[\mathbf{u}_{j-1} + \mathbf{u}_{j+1}] \quad (6.5.5-24a)$$

and

$$\beta_j(1) = \mathbf{u}_{j-2} + \mathbf{u}_{j+2} - T^{-1}T(1)[\mathbf{u}_{j-1} + \mathbf{u}_{j+1}]. \quad (6.5.5-24b)$$

By combining (6.5.5-23) and (6.5.5-24a) we obtain

$$\begin{aligned} \mathbf{s}_j &= T\mathbf{v}_j(1) + \mathbf{u}_{j-1} + T^{-1}[\mathbf{u}_{j-2} + \mathbf{u}_j] + \mathbf{u}_{j+1} + T^{-1}[\mathbf{u}_j + \mathbf{u}_{j+2}] \\ &= T\mathbf{v}_j(1) + \mathbf{u}_{j-1} + \mathbf{u}_{j+1} + T^{-1}[\mathbf{u}_{j-2} + 2\mathbf{u}_j + \mathbf{u}_{j+2}]. \end{aligned} \quad (6.5.5-25)$$

By substituting (6.5.5-25) into (6.5.5-22), we obtain

$$\begin{aligned} \mathbf{u}_{j-1} + T\mathbf{u}_j + \mathbf{u}_{j+1} \\ = T\mathbf{v}_j(1) + \mathbf{u}_{j-1} + \mathbf{u}_{j+1} + T^{-1}[\mathbf{u}_{j-2} + 2\mathbf{u}_j + \mathbf{u}_{j+2}]. \end{aligned}$$

Therefore, we have

$$\mathbf{v}_j(1) = \mathbf{u}_j - [T^{-1}]^2[\mathbf{u}_{j-2} + 2\mathbf{u}_j + \mathbf{u}_{j+2}]. \quad (6.5.5-26)$$

Next, from (6.5.5-10b) we have

$$\kappa_j(1) = \beta_{j-1}(1) + \beta_{j+1}(1) - [\alpha_{j-1}(1) + \alpha_{j+1}(1)]$$

which upon substituting (6.5.5-24a) and (6.5.5-24b), gives

$$\begin{aligned} \kappa_j(1) &= \mathbf{u}_{j-3} + \mathbf{u}_{j-1} - T^{-1}T(1)[\mathbf{u}_{j-2} + \mathbf{u}_j] \\ &\quad + \mathbf{u}_{j+3} + \mathbf{u}_{j+1} - T^{-1}T(1)[\mathbf{u}_{j+2} + \mathbf{u}_j] \\ &\quad - \{\mathbf{u}_{j+1} + \mathbf{u}_{j-1} + T^{-1}[\mathbf{u}_{j-2} + 2\mathbf{u}_j + \mathbf{u}_{j+2}]\} \\ &= \mathbf{u}_{j-3} + \mathbf{u}_{j+3} - T^{-1}[T(1) + I][\mathbf{u}_{j-2} + 2\mathbf{u}_j + \mathbf{u}_{j+2}] \\ &= \mathbf{u}_{j-3} + \mathbf{u}_{j+3} - T^{-1}P(1)[\mathbf{u}_{j-2} + 2\mathbf{u}_j + \mathbf{u}_{j+2}]. \end{aligned} \quad (6.5.5-27)$$

Next, from the odd-even reduction stage, we know that

$$\alpha_j(2) = \mathbf{u}_j - \mathfrak{S}(2) \sum_{l=1}^2 [\mathbf{u}_{j-(2l-1)} + \mathbf{u}_{j+(2l-1)}] \quad (6.5.5-28a)$$

and

$$\beta_j(2) = \mathbf{u}_{j-4} + \mathbf{u}_{j+4} + \mathfrak{S}(2) T(2) \sum_{l=1}^2 [\mathbf{u}_{j-(2l-1)} + \mathbf{u}_{j+(2l-1)}] \quad (6.5.5-28b)$$

where

$$\mathfrak{S}(2) = [T(0)T(1)]^{-1} \quad (6.5.5-28c)$$

$$T(0) = T \quad (6.5.5-28d)$$

$$T(1) = 2I - T^2. \quad (6.5.5-28e)$$

Hence using (6.5.5-24a), (6.5.5-24b) and (6.5.5-28a), we can simplify (6.5.5-13a) to obtain $\mathbf{v}_j(2)$ in terms of the \mathbf{u}_j as

$$\begin{aligned} \mathbf{v}_j(2) &= \mathbf{u}_j + T^{-1}[\mathbf{u}_{j-1} + \mathbf{u}_{j+1}] \\ &\quad - [T(1)]^{-1} \left\{ \mathbf{u}_{j-3} - \mathfrak{S}(2) \sum_{l=1}^2 [\mathbf{u}_{j-3-(2l-1)} + \mathbf{u}_{j-3+(2l-1)}] \right. \\ &\quad \left. + \mathbf{u}_{j+3} - \mathfrak{S}(2) \sum_{l=1}^2 [\mathbf{u}_{j+3-(2l-1)} + \mathbf{u}_{j+3+(2l-1)}] \right. \\ &\quad \left. - [\mathbf{u}_{j-2} + \mathbf{u}_{j+2} - T^{-1}T(1)[\mathbf{u}_{j-1} + \mathbf{u}_{j+1}]] \right\} \\ &= \mathbf{u}_j - [T(1)]^{-1} \left\{ [\mathbf{u}_{j-3} + \mathbf{u}_{j+3}] - [\mathbf{u}_{j-2} + \mathbf{u}_{j+2}] \right. \\ &\quad \left. - \mathfrak{S}(2) \sum_{l=1}^2 [\mathbf{u}_{j-2(1+l)} + \mathbf{u}_{j+2(1+l)} \right. \\ &\quad \left. + \mathbf{u}_{j-2(1-2)} + \mathbf{u}_{j+2(1-2)}] \right\}. \quad (6.5.5-29a) \end{aligned}$$

Likewise, by using (6.5.5-28a), (6.5.5-28b) in (6.5.5-13b), we can obtain the expression for $\kappa_j(2)$ in terms of the \mathbf{u}_j as

$$\begin{aligned} \kappa_j(2) &= \mathbf{u}_{j-7} + \mathbf{u}_{j+7} + \mathbf{u}_{j-1} + \mathbf{u}_{j+1} - [\mathbf{u}_{j-3} + \mathbf{u}_{j+3}] \\ &\quad + \mathfrak{S}(2) [T(2) + I] \left\{ \sum_{l=1}^2 [\mathbf{u}_{j-3-(2l-1)} + \mathbf{u}_{j-3+(2l-1)} \right. \\ &\quad \left. + \mathbf{u}_{j+3-(2l-1)} + \mathbf{u}_{j+3+(2l-1)}] \right\} \\ &= \mathbf{u}_{j-7} + \mathbf{u}_{j+7} + \mathbf{u}_{j-1} + \mathbf{u}_{j+1} - [\mathbf{u}_{j-3} + \mathbf{u}_{j+3}] \\ &\quad + \mathfrak{S}(2) P(2) [T(1)]^{-1} \left\{ \sum_{l=1}^2 [\mathbf{u}_{j-2(1-2)} + \mathbf{u}_{j+2(1-2)} \right. \\ &\quad \left. + \mathbf{u}_{j-2(1+1)} + \mathbf{u}_{j+2(1+1)}] \right\}. \quad (6.5.5-29b) \end{aligned}$$

Now (6.5.5-29a) can be generalized to give

$$\begin{aligned} \mathbf{v}_j(r+1) = \mathbf{u}_j - [\mathbf{T}(r)]^{-1} \left\{ \sum_{l=1}^{2^r} (-1)^l [\mathbf{u}_{j-(l+1)} + \mathbf{u}_{j+(l+1)}] \right. \\ \left. + (-1)^r \mathfrak{S}(r+1) \sum_{l=1}^{2^r} [\mathbf{u}_{j-2(l+1)} + \mathbf{u}_{j+2(l+1)} \right. \\ \left. + \mathbf{u}_{j-2(l-2)} + \mathbf{u}_{j+2(l-2)}] \right\}. \end{aligned}$$

Therefore, we obtain for $r \geq 1$

$$\begin{aligned} \mathbf{v}_j(r+1) = \mathbf{u}_j + [\mathbf{T}(r)]^{-1} \left\{ \sum_{l=1}^{2^r} (-1)^{l+1} [\mathbf{u}_{j-(l+1)} + \mathbf{u}_{j+(l+1)}] \right\} \\ + (-1)^{r+1} \mathfrak{S}(r+1) [\mathbf{T}(r)]^{-1} \left\{ \sum_{l=1}^{2^r} [\mathbf{u}_{j-2(l+1)} + \mathbf{u}_{j+2(l+1)} \right. \\ \left. + \mathbf{u}_{j-2(l-2)} + \mathbf{u}_{j+2(l-2)}] \right\}. \end{aligned} \quad (6.5.5-30a)$$

Similarly, we have for $r \geq 1$

$$\begin{aligned} \mathbf{\kappa}_j(r+1) = \sum_{l=1}^{3^r} (-1)^{l+1} [\mathbf{u}_{j-(2^l-1)} + \mathbf{u}_{j+(2^l-1)}] \\ + (-1)^{r+1} \mathfrak{S}(r+1) \mathbf{P}(r+1) [\mathbf{T}(r)]^{-1} \left\{ \sum_{l=1}^{2^r} [\mathbf{u}_{j-2(l+1)} + \mathbf{u}_{j+2(l+1)} \right. \\ \left. + \mathbf{u}_{j-2(l-2)} + \mathbf{u}_{j+2(l-2)}] \right\}. \end{aligned} \quad (6.5.5-30b)$$

Therefore on writing

$$\Delta_j(r+1) = \mathfrak{S}(r+1) \sum_{l=1}^{2^r} [\mathbf{u}_{j-2(l+1)} + \mathbf{u}_{j+2(l+1)} + \mathbf{u}_{j-2(l-2)} + \mathbf{u}_{j+2(l-2)}] \quad (6.5.5-31)$$

for $r \geq 1$, we obtain

$$\begin{aligned} \mathbf{v}_j(r+1) = \mathbf{u}_j + [\mathbf{T}(r)]^{-1} \left\{ \sum_{l=1}^{2^r} (-1)^{l+1} [\mathbf{u}_{j-(l+1)} + \mathbf{u}_{j+(l+1)}] \right. \\ \left. + (-1)^{r+1} \Delta_j(r+1) \right\}, \end{aligned} \quad (6.5.5-32a)$$

$$\begin{aligned} \mathbf{\kappa}_j(r+1) = \sum_{l=1}^{3^r} (-1)^{l+1} [\mathbf{u}_{j-(2^l-1)} + \mathbf{u}_{j+(2^l-1)}] \\ + (-1)^{r+1} \mathbf{P}(r+1) [\mathbf{T}(r)]^{-1} \Delta_j(r+1), \end{aligned} \quad (6.5.5-32b)$$

and

$$\mathfrak{S}(r+1) = \left[\prod_{l=0}^r \mathbf{T}(l) \right]^{-1}. \quad (6.5.5-32c)$$

We note that the matrix T is obtained from the discretized Poisson equation and is of the form

$$T = \begin{bmatrix} 4 & -1 & & & & -1 \\ -1 & 4 & -1 & & & \\ & & \cdot & \vdots & \vdots & 0 \\ & & & \cdot & \vdots & \cdot \\ & & & 0 & \cdot & \cdot \\ -1 & & & & -1 & 4 \end{bmatrix}. \quad (6.5.5-33)$$

Now let

$$\|u\| = \sum_{j=1}^m \|u_j\|. \quad (6.5.5-34)$$

Then, from (6.5.5-32a), we obtain

$$\|v_{j(r+1)} - u_j\| \leq \| [T(r)]^{-1} \| \{ 2 \|u\| + \|\Delta_{j(r+1)}\| \}.$$

Thus, we have

$$\|v_{j(r+1)} - u_j\| \leq \| [T(r)]^{-1} \| \{ 2 + 4 \|\mathfrak{S}_{(r+1)}\| \} \|u\|. \quad (6.5.5-35)$$

By using (6.5.5-17), (6.5.5-17a) and (6.5.5-33) it is easy to show that

$$\| [T(r)]^{-1} \| \leq 2^{-2^r} \leq 2. \quad (6.5.5-36)$$

Furthermore, we have (Lakshmivarahan and Dhall[1990], pp. 403-405) that

$$\|\mathfrak{S}_{(r+1)}\| \leq e^{-(2^{r+1}-1)\phi^*}, \quad (6.5.5-37)$$

where ϕ^* is defined such that

$$[\cosh(2^j \phi^*)]^{-1} = \max_{\phi_i} \{ [\cosh(2^j \phi_i)]^{-1} \} \quad (6.5.5-38)$$

and ϕ_i is given by

$$\phi_i = \cosh^{-1} \left(-\frac{\lambda_i}{2} \right). \quad (6.5.5-39)$$

This happens when T is of the form given in (6.5.5-33) where $|\lambda_i| \geq 2$ since $\lambda_i = -4 + 2\cos\left(\frac{i\pi}{n+1}\right)$; $i = 1, 2, \dots, n$.

Therefore by combining (6.5.5-36) and (6.5.5-37) with (6.5.5-35), we obtain

$$\|v_{j(r+1)} - u_j\| \leq 2^{-2^{r+1}} \{1 + 2e^{-(2^{r+1}-1)\phi^*}\} \|u\|. \quad (6.5.5-40)$$

Now (6.5.5-40) states that as r tends to infinity, $v_{j(r+1)}$ converges to the solution u_j . In other words, for large r , $v_{j(r+1)}$ is a good approximation to u_j .

Next, we also have that, when $|\lambda_i| \geq 2$,

$$\|P_{(r+1)} \mathfrak{S}_{(r+1)}\| \leq 2e^{\phi_n}. \quad (6.5.5-41)$$

From (6.5.5-32b), we obtain

$$\begin{aligned} \|\kappa_{j(r+1)} - \sum_{l=1}^{3^r} (-1)^{l+1} [u_{j-(2^l-1)} + u_{j+(2^l-1)}]\| \\ \leq 4 \|T(r)\|^{-1} \|P_{(r+1)} \mathfrak{S}_{(r+1)}\| \|u\|. \end{aligned} \quad (6.5.5-42)$$

Hence, by using (6.5.5-36) and (6.5.5-41) in (6.5.5-42), we have

$$\begin{aligned} \|\kappa_{j(r+1)} - \sum_{l=1}^{3^r} (-1)^{l+1} [u_{j-(2^l-1)} + u_{j+(2^l-1)}]\| \\ \leq 4 \times 2 \times 2e^{\phi_n} \|u\| = 16e^{\phi_n} \|u\|. \end{aligned} \quad (6.5.5-43)$$

Thus $\kappa_{j(r+1)}$ remains bounded throughout the computation. This proves that the modified form of the TR3 reduction algorithm is numerically stable.

6.6 ITERATIVE METHOD FOR SOLVING (6.4.1)

In chapter 3 we have outlined the basic iterative methods for solving the linear system of equations. In the following sections, we shall use the successive line overrelaxation (SLOR) technique to solve the system of linear equations (6.4-1) derived from the discretization of the Poisson equation with periodic boundary conditions (6.2.2) using the five-point finite difference approximation (6.3-2). Next, we shall prove numerically that both the periodic and non-periodic problems share a common optimum parameter yet their spectral radii are different. Thus we conclude the chapter by deriving the relationship between the optimum parameter and the spectral radius for the elliptic periodic boundary-value problem.

6.6.1 OUTLINE OF SLOR ITERATIVE METHOD FOR SOLVING (6.4-1)

Consider the solutions of a system of linear equations derived from the discretization of the second-order elliptic PDE with periodic boundary conditions

$$\mathbf{M}\mathbf{u} = \mathbf{s} \quad (6.6.1-1)$$

where

$$\mathbf{M} = \begin{bmatrix} A_{1,1} & -A_{1,2} & & & -A_{1,n} \\ -A_{2,1} & A_{2,2} & -A_{2,3} & & \\ & \cdot & \cdot & \cdot & 0 \\ & 0 & \cdot & \cdot & \cdot \\ & & & -A_{n-1,n-2} & A_{n-1,n-1} & -A_{n-1,n} \\ -A_{n,1} & & & -A_{n,n-1} & A_{n,n} \end{bmatrix} \quad (6.6.1-1a)$$

$$\left. \begin{aligned} \mathbf{u} &= [\mathbf{u}_1^T, \mathbf{u}_2^T, \dots, \mathbf{u}_n^T]^T \\ \mathbf{s} &= [\mathbf{s}_1^T, \mathbf{s}_2^T, \dots, \mathbf{s}_n^T]^T \end{aligned} \right\} \quad (6.6.1-1b)$$

The SLOR method may be written in the form

$$\left. \begin{aligned}
 A_{1,1} \mathbf{u}_1^{(r+1)} &= A_{1,1} \mathbf{u}_1^{(r)} + \omega \{ A_{1,2} \mathbf{u}_2^{(r)} + A_{1,n} \mathbf{u}_n^{(r)} + \mathbf{s}_1 - A_{1,1} \mathbf{u}_1^{(r)} \} \\
 A_{i,i} \mathbf{u}_i^{(r+1)} &= A_{i,i} \mathbf{u}_i^{(r)} + \omega \{ A_{i,i-1} \mathbf{u}_{i-1}^{(r+1)} + A_{i,i+1} \mathbf{u}_{i+1}^{(r)} + \mathbf{s}_i - A_{i,i} \mathbf{u}_i^{(r)} \} \\
 \text{for } i &= 2, 3, \dots, n-1 \\
 A_{n,n} \mathbf{u}_n^{(r+1)} &= A_{n,n} \mathbf{u}_n^{(r)} + \omega \{ A_{n,n-1} \mathbf{u}_{n-1}^{(r+1)} + A_{n,1} \mathbf{u}_1^{(r+1)} + \mathbf{s}_n - A_{n,n} \mathbf{u}_n^{(r)} \}
 \end{aligned} \right\} \quad (6.6.1-2)$$

for $r = 0, 1, 2, \dots$

We can further simplify (6.6.1-2) to give

$$\left. \begin{aligned}
 \mathbf{u}_1^{(r+1)} &= (1 - \omega) \mathbf{u}_1^{(r)} + \omega A_{1,1}^{-1} \{ A_{1,2} \mathbf{u}_2^{(r)} + A_{1,n} \mathbf{u}_n^{(r)} + \mathbf{s}_1 \} \\
 \mathbf{u}_i^{(r+1)} &= (1 - \omega) \mathbf{u}_i^{(r)} + \omega A_{i,i}^{-1} \{ A_{i,i-1} \mathbf{u}_{i-1}^{(r+1)} + A_{i,i+1} \mathbf{u}_{i+1}^{(r)} + \mathbf{s}_i \} \\
 \text{for } i &= 2, 3, \dots, n-1 \\
 \mathbf{u}_n^{(r+1)} &= (1 - \omega) \mathbf{u}_n^{(r)} + \omega A_{n,n}^{-1} \{ A_{n,n-1} \mathbf{u}_{n-1}^{(r+1)} + A_{n,1} \mathbf{u}_1^{(r+1)} + \mathbf{s}_n \}
 \end{aligned} \right\} \quad (6.6.1-3)$$

for $r = 0, 1, 2, \dots$

Now write

$$\left. \begin{aligned}
 \mathbf{v}_1^{(r)} &= A_{1,1}^{-1} \{ A_{1,2} \mathbf{u}_2^{(r)} + A_{1,n} \mathbf{u}_n^{(r)} + \mathbf{s}_1 \} \\
 \mathbf{v}_i^{(r+1)} &= A_{i,i}^{-1} \{ A_{i,i-1} \mathbf{u}_{i-1}^{(r+1)} + A_{i,i+1} \mathbf{u}_{i+1}^{(r)} + \mathbf{s}_i \} \\
 \text{for } i &= 2, 3, \dots, n-1 \\
 \mathbf{v}_n^{(r+1)} &= A_{n,n}^{-1} \{ A_{n,n-1} \mathbf{u}_{n-1}^{(r+1)} + A_{n,1} \mathbf{u}_1^{(r+1)} + \mathbf{s}_n \}
 \end{aligned} \right\} \quad (6.6.1-4)$$

for $r = 0, 1, 2, \dots$

By combining (6.6.1-4) and (6.6.1-3), we obtain

general case of $i = 2, 3, \dots, n-1$ by taking into consideration the periodicity of the problem.

Henceforth, we shall concentrate on the lines $i = 2, 3, \dots, n-1$. Therefore from (6.6.1-5), (6.6.1-6), (6.6.1-6a) and (6.6.1-6b), we have

$$\mathbf{u}_i^{(r+1)} = (1 - \omega)\mathbf{u}_i^{(r)} + \omega\mathbf{v}_i^{(r+1)} \quad (6.6.1-7)$$

for $i = 2, 3, \dots, n-1$ and the matrix $A_{i,i}$ is of the form

$$A_{i,i} = \begin{bmatrix} \alpha & -\beta & & & & -\gamma \\ -\gamma & \alpha & -\beta & & & \\ & \cdot & \cdot & \cdot & & 0 \\ & 0 & \cdot & \cdot & \cdot & \cdot \\ & & & -\gamma & \alpha & -\beta \\ -\beta & & & & -\gamma & \alpha \end{bmatrix} \quad (6.6.1-7a)$$

for $i = 1, 2, \dots, n$.

The system of linear equations (6.6.1-6) can either be solved directly (Benson[1969]) or iteratively. Thus we have n (including the two systems of equations derived from the first and the last lines) such systems of n linear equations to be solved.

In the next section we shall derive the relationship between the spectral radius and overrelaxation factor for the periodic problem that we are now considering.

$$L = D^{-1}E$$

and

$$U = D^{-1}F,$$

we can write the system (6.6.1-1) in the form

$$[I - L - U]u = D^{-1}s \quad (6.6.2-4)$$

and the respective Jacobi and SOR iteration matrices are given as

$$G_B = L + U \quad (6.6.2-4a)$$

$$L_\omega = [I - \omega L]^{-1}[\omega U - (\omega - 1)I]. \quad (6.6.2-4b)$$

Assume that the submatrices $A_{i,i}$ of (6.6.1-7a) for $i = 1, 2, \dots, n$ are in normalized form. Therefore we may write

$$A_{i,i} = T = \begin{bmatrix} 1 & -\delta & & & & -\sigma \\ -\sigma & 1 & -\delta & & & \\ & \cdot & \cdot & \cdot & & 0 \\ & 0 & & \cdot & \cdot & \cdot \\ & & & & -\sigma & 1 & -\delta \\ -\delta & & & & -\sigma & 1 & \end{bmatrix} \quad (6.6.2-5)$$

for $i = 1, 2, \dots, n$.

The block matrix M of (6.6.1-1) from the discretization of the problem is then given by

$$M = \begin{bmatrix} T & -\delta I & & & & -\sigma I \\ -\sigma I & T & -\delta I & & & \\ & \cdot & \cdot & \cdot & & 0 \\ & 0 & & \cdot & \cdot & \cdot \\ & & & & -\sigma I & T & -\delta I \\ -\delta I & & & & -\sigma I & T & \end{bmatrix}. \quad (6.6.2-6)$$

Since the submatrices T and I are commutative, then by the spectral decomposition method and writing

$$T\Psi = \Gamma\Psi \quad (6.6.2-7)$$

where Ψ and Γ are the respective eigenvectors and eigenvalues of T , deduced from section 6.4, then we may resolve the linear system of (6.6.1-1) as

$$T_i w_i = y_i, \quad i=1,2,\dots,n. \quad (6.6.2-8)$$

There are n such systems of n linear equations to be solved where $T_i, i=1,2,\dots,n$ are of the form of the original block matrix M . Since the submatrices T_i are all the same, we shall now denote them by T as given by (6.6.2-5). Henceforth, we shall consider the point form given by (6.6.2-8) in the analysis.

Now the Jacobi iteration matrix G_B of the point equivalent is of the form

$$G_B = \begin{bmatrix} 0 & \delta & & & & & & & \sigma \\ \sigma & 0 & \delta & & & & & & \\ & & \cdot & \cdot & \cdot & & 0 & & \\ & 0 & & \cdot & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & & & \sigma & 0 & \delta \\ \delta & & & & & & & \sigma & 0 \end{bmatrix}. \quad (6.6.2-9)$$

Thus we have

$$G_B = Q_1 + Q_2, \quad (6.6.2-10)$$

where

$$Q_1 = \begin{bmatrix} 0 & & & & & & & & \sigma \\ \sigma & 0 & & & & & & & \\ & & \cdot & \cdot & \cdot & & 0 & & \\ & 0 & & \cdot & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & & \\ & & & & & & \sigma & 0 & \\ & & & & & & & \sigma & 0 \end{bmatrix} \quad (6.6.2-10a)$$

and

$$Q_2 = \begin{bmatrix} 0 & \delta & & & & & \\ & 0 & \delta & & & & \\ & & \cdot & \cdot & & & 0 \\ & 0 & & \cdot & \cdot & & \\ & & & & \cdot & & \\ & & & & & 0 & \delta \\ \delta & & & & & & 0 \end{bmatrix}. \quad (6.6.2-10b)$$

That is, we can express Q_1 and Q_2 in the form

$$\left. \begin{aligned} Q_1 &= L_1 + U_1 \\ Q_2 &= L_2 + U_2 \end{aligned} \right\} \quad (6.6.2-11)$$

where L_i and U_i , $i=1,2$ are lower and upper triangular matrices respectively.

Now suppose λ is an eigenvalue of the SOR matrix L_ω and let \mathbf{z} be the corresponding eigenvector, then

$$[I - \omega L]^{-1}[\omega U - (\omega - 1)I]\mathbf{z} = \lambda \mathbf{z}. \quad (6.6.2-12)$$

Premultiplying both sides of (6.6.2-12) by $[I - \omega L]$, we obtain

$$[\omega U - (\omega - 1)I]\mathbf{z} = \lambda[I - \omega L]\mathbf{z}.$$

Thus on simplifying, we obtain

$$[U + \lambda L]\mathbf{z} = \left[\frac{\lambda + \omega - 1}{\omega} \right] I \mathbf{z} \quad (6.6.2-13)$$

where the matrix $[U + \lambda L]$ has the form

$$[U + \lambda L] = \begin{bmatrix} 0 & \delta & & & & & \sigma \\ \lambda\sigma & 0 & \delta & & & & \\ & & \cdot & \cdot & \cdot & \cdot & 0 \\ & 0 & & \cdot & \cdot & \cdot & \cdot \\ & & & & \lambda\sigma & 0 & \delta \\ \lambda\delta & & & & & \lambda\sigma & 0 \end{bmatrix}. \quad (6.6.2-14)$$

Now write

$$[U + \lambda L] = P_1 + P_2, \quad (6.6.2-15)$$

with

$$P_1 = \begin{bmatrix} 0 & & & & & & \sigma \\ \lambda\sigma & 0 & & & & & \\ & \cdot & \cdot & \cdot & & & 0 \\ & & \cdot & \cdot & \cdot & & \cdot \\ 0 & & & \cdot & \cdot & & \cdot \\ & & & & \lambda\sigma & 0 & \\ & & & & & & 0 \end{bmatrix} \quad (6.6.2-15a)$$

and

$$P_2 = \begin{bmatrix} 0 & \delta & & & & & \\ & 0 & \delta & & & & \\ & & \cdot & \cdot & \cdot & & 0 \\ & & & \cdot & \cdot & \cdot & \cdot \\ 0 & & & & & & \cdot \\ & & & & & 0 & \delta \\ \lambda\delta & & & & & & 0 \end{bmatrix}. \quad (6.6.2-15b)$$

That is, P_1 and P_2 can be expressed in the form

$$\left. \begin{aligned} P_1 &= U_1 + \lambda L_1 \\ P_2 &= U_2 + \lambda L_2 \end{aligned} \right\}. \quad (6.6.2-16)$$

We shall now establish a similarity transformation which relates the matrices $[U_i + \lambda L_i]$ and $[U_i + L_i]$ for $i=1,2$.

Let

$$Q = \text{diag}[1, \lambda^{-1/n}, \lambda^{-2/n}, \dots, \lambda^{-(n-1)/n}], \quad (6.6.2-17)$$

then, we have the inverse of Q as

$$Q^{-1} = \text{diag}[1, \lambda^{1/n}, \lambda^{2/n}, \dots, \lambda^{(n-1)/n}]. \quad (6.6.2-18)$$

By virtue of this similarity transformation,

$$QP_1Q^{-1} = P_1. \quad (6.6.2-19)$$

But by substituting (6.6.2-17), (6.6.2-18) and P_1 as given in (6.6.2-15a), we can easily show that

$$QP_1Q^{-1} = \lambda^{(n-1)/n}Q_1. \quad (6.6.2-20)$$

Therefore, by equating (6.6.2-19) and (6.6.2-20), we obtain

$$P_1 = \lambda^{(n-1)/n}Q_1. \quad (6.6.2-21)$$

Similarly, we can show that

$$P_2 = \lambda^{1/n}Q_2. \quad (6.6.2-22)$$

Next, we note that (6.6.2-13) can also be expressed as

$$\det \left\{ [U + \lambda L] - \left[\frac{\lambda + \omega - 1}{\omega} \right] I \right\} = 0,$$

or

$$\det \left\{ [P_1 + P_2] - \left[\frac{\lambda + \omega - 1}{\omega} \right] I \right\} = 0. \quad (6.6.2-23)$$

Thus

$$\det \left\{ Q [P_1 + P_2 - \left(\frac{\lambda + \omega - 1}{\omega} \right) I] Q^{-1} \right\} = 0,$$

or

$$\det \left\{ QP_1Q^{-1} + QP_2Q^{-1} - \left(\frac{\lambda + \omega - 1}{\omega} \right) I \right\} = 0. \quad (6.6.2-24)$$

Recall that

$$\det \{ B - \lambda I \} = 0,$$

may be written as

$$\prod_{k=0}^n \{\mu_k(B) - \Lambda\} = 0,$$

where $\mu_k(B)$ are the eigenvalues of B.

Therefore, (6.6.2-24) becomes,

$$\prod_{k=0}^n \left\{ \mu_k(QP_1Q^{-1} + QP_2Q^{-1}) - \left(\frac{\lambda + \omega - 1}{\omega} \right) \right\} = 0. \quad (6.6.2-25)$$

Now the general circulant matrix,

$$\begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \dots & a_{n-2} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ a_2 & a_3 & a_4 & \dots & a_1 \\ a_1 & a_2 & a_3 & \dots & a_0 \end{bmatrix} \quad (6.6.2-26)$$

has eigenvalues of the form

$$\mu_j = \sum_{k=0}^{n-1} a_k \zeta_j^k, \quad 0 \leq j \leq n-1 \quad (6.6.2-26a)$$

with the corresponding eigenvectors as

$$v_j = [1, \zeta_j, \zeta_j^2, \dots, \zeta_j^{n-1}]^T, \quad (6.6.2-26b)$$

where

$$\zeta_j = \exp\left(\frac{2\pi i j}{n}\right), \quad (6.6.2-26c)$$

for $0 \leq j \leq n-1$ and $i = \sqrt{-1}$.

Thus, for the matrix Q_1 , the eigenvalues are

$$\sigma \zeta_j^{n-1} \quad (6.6.2-27a)$$

and for the matrix Q_2 , the eigenvalues are

$$\delta \zeta_j \quad (6.6.2-27b)$$

for $0 \leq j \leq n-1$.

Their corresponding eigenvectors are

$$(1, \zeta_j, \zeta_j^2, \dots, \zeta_j^{n-1})^T, \quad (6.6.2-27c)$$

for $0 \leq j \leq n-1$.

We shall now concentrate on the coincident eigenvectors of Q_1 and Q_2 . Then, by using (6.6.2-21) and (6.6.2-22), we can simplify (6.6.2-25) to give

$$\prod_{k=0}^n \left\{ \mu_k (\lambda^{(n-1)/n} Q_1 + \lambda^{1/n} Q_2) - \left(\frac{\lambda + \omega - 1}{\omega} \right) \right\} = 0. \quad (6.6.2-28)$$

Since Q_1 and Q_2 have coincident eigenvectors, then there exists some k such that (6.6.2-28) reduces to

$$\lambda^{(n-1)/n} \mu(Q_1) + \lambda^{1/n} \mu(Q_2) = \frac{\lambda + \omega - 1}{\omega}. \quad (6.6.2-29)$$

Now let $\Lambda = \lambda^{1/n} \zeta_j$ and substitute $\mu(Q_1) = \sigma \zeta_j^{n-1}$, $\mu(Q_2) = \delta \zeta_j$ and, since $\zeta_j^n = 1$, then

$$\Lambda^n - \Lambda^{n-1} \omega \sigma - \Lambda \omega \delta + \omega - 1 = 0. \quad (6.6.2-30)$$

By adding and subtracting $\omega^2 \sigma \delta$ in (6.6.2-30), we obtain

$$[\Lambda^n - \Lambda^{n-1} \omega \sigma - \Lambda \omega \delta + \omega^2 \sigma \delta] - [\omega^2 \sigma \delta - \omega + 1] = 0. \quad (6.6.2-31)$$

Therefore, given a system of linear equations of the form (6.6.1-1), (6.6.2-31) describes the relation between λ , the eigenvalue of the SOR matrix L_ω of (6.6.2-4), and ω , the overrelaxation factor.

In order to determine the optimum ω which satisfies (6.6.2-31), we let

$$G_n(\Lambda) = \Lambda^n - \Lambda^{n-1} \omega \sigma - \Lambda \omega \delta + \omega^2 \sigma \delta, \quad (6.6.2-32)$$

$$C_\omega = \omega^2 \sigma \delta - \omega + 1. \quad (6.6.2-33)$$

Now, for (6.6.2-31) to be true, we should have

$$G_n(\Lambda) = C_\omega, \quad (6.6.2-34)$$

for any ω satisfying (6.6.2-32) and (6.6.2-33). But (6.6.2-33) is independent of Λ , therefore we may now write (6.6.2-31) as

$$\Lambda^n - \Lambda^{n-1}\omega\sigma - \Lambda\omega\delta + \omega^2\sigma\delta = C_\omega, \quad (6.6.2-35)$$

where

$$\begin{aligned} C_\omega &= \omega^2\sigma\delta - \omega + 1, \\ &= \sigma\delta \left\{ \omega - \frac{1}{2\sigma\delta} + \frac{\sqrt{1 - 4\sigma\delta}}{2\sigma\delta} \right\} \left\{ \omega - \frac{1}{2\sigma\delta} - \frac{\sqrt{1 - 4\sigma\delta}}{2\sigma\delta} \right\} \end{aligned}$$

Therefore we may write C_ω in the form

$$C_\omega = \sigma\delta \left\{ \omega - \frac{2}{1 + \sqrt{1 - 4\sigma\delta}} \right\} \left\{ \omega - \frac{2}{1 - \sqrt{1 - 4\sigma\delta}} \right\} \quad (6.6.2-36)$$

We observe that, for any ω satisfying (6.6.2-32) and (6.6.2-36), the relation (6.6.2-35) is true. Thus it is necessary and sufficient to discuss the magnitude of C_ω in order to determine the optimum Λ which is true for (6.6.2-31). Note that C_ω is only determined by ω and it is independent of Λ , since σ and δ are constants. Now $0 < \omega < 2$. If $\omega = 0$, then $C_\omega = 1$ and $\Lambda^n = 1$. Thus Λ has a unity root of multiplicity n . This is obvious and expected. The numerical results confirm this. As ω moves away from zero towards unity, $C_\omega = \omega^2\sigma\delta + 1 - \omega$ moves away from unity towards $\sigma\delta$ which is always positive but less than unity since $0 < \sigma, \delta < \frac{1}{2}$. But now as ω moves further away from unity towards two, $C_\omega = \omega^2\sigma\delta + (1 - \omega)$ is governed by the sign of $1 - \omega$. Clearly, C_ω will first be positive as long as $1 - \omega > 0$. Then at $\omega = \omega_{opt}$, $C_{\omega_{opt}} = 0$ and then stays negative for $\omega > \omega_{opt}$, when at $\omega = 2$, $C_\omega = 4\sigma\delta - 1 < 0$ since $0 < \sigma, \delta < \frac{1}{2}$.

In the accompanying Figure(6.6.2), the optimization strategy is illustrated. The graphs depict the roots of the characteristic polynomial described by (6.6.2-35) for different values of C_ω as ω progresses from zero towards two. In Figure(6.6.2a) where $\omega = 0$ and $C_\omega = 1$, all the eigenvalues are unity. As ω increases from zero but C_ω decreases from unity, some of the eigenvalues become complex pairs, Figure(6.6.2b) to Figure(6.6.2d). As ω increases further, more eigenvalues become complex until eventually all of the eigenvalues lie on a circle of radius $|\lambda_{opt}|$ at $\omega = \omega_{opt}$ in Figure(6.6.2e) where $|\lambda_{opt}|$ is the minimum of the maximum value of the moduli of the eigenvalues which is less than unity. At this point $C_{\omega_{opt}}$ is numerically equal to zero. For values of ω beyond ω_{opt} , C_ω moves away from zero; the eigenvalues increase in moduli as shown in Figure(6.6.2f) and Figure(6.6.2g).

Therefore, the optimum Λ is obtained corresponding to setting $C_\omega = 0$. Hence the optimum relaxation parameter ω is determined by solving the equation

$$\omega^2 \sigma \delta + (1 - \omega) = 0. \quad (6.6.2-37)$$

Let ω_{opt} be the solution of (6.6.2-37), therefore we have

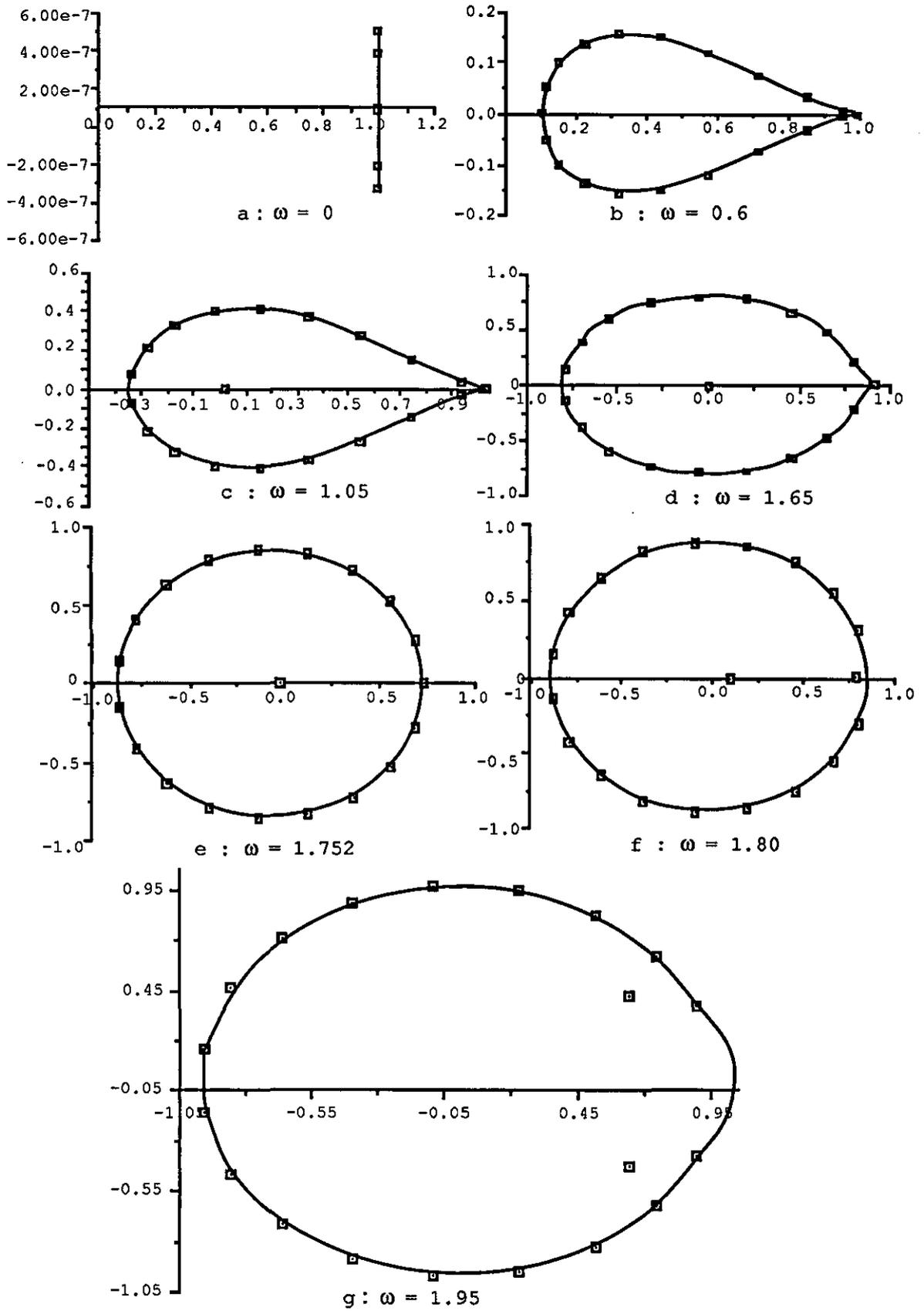
$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - 4\sigma\delta}} \quad (6.6.2-38)$$

since the other root $\frac{2}{1 - \sqrt{1 - 4\sigma\delta}} > 2$ for $0 < \sigma, \delta < \frac{1}{2}$.

Similarly, if σ and δ are of opposite sign, we can easily show that (6.6.2-38) ^{is} still valid. Thus the optimum relaxation factor is given by (6.6.2-38).

Finally the spectral radius λ_{opt} can be simply derived from (6.6.2-35) by substituting $C_\omega = 0$. That is,

$$\{\Lambda^{n-1} - \omega\sigma\}\{\Lambda - \omega\delta\} = 0. \quad (6.6.2-39)$$



Figure(6.6.2) : Optimization process of the relaxation parameter for periodic problem.

Hence we obtain the solutions of (6.6.2-39) as

$$\Lambda = [\omega\sigma]^{1/(n-1)} \quad \text{or} \quad \Lambda = \omega\delta.$$

Since $\Lambda = \lambda^{1/n}$, therefore we have

$$\lambda = [\omega\sigma]^{n/(n-1)} \quad \text{or} \quad \lambda = [\omega\delta]^n.$$

Now for M symmetric and both $|\sigma|, |\delta| < \frac{1}{2}$, so that

$$[\omega\delta]^n < [\omega\sigma]^{n/(n-1)},$$

therefore the optimum spectral radius λ_{opt} , is given by

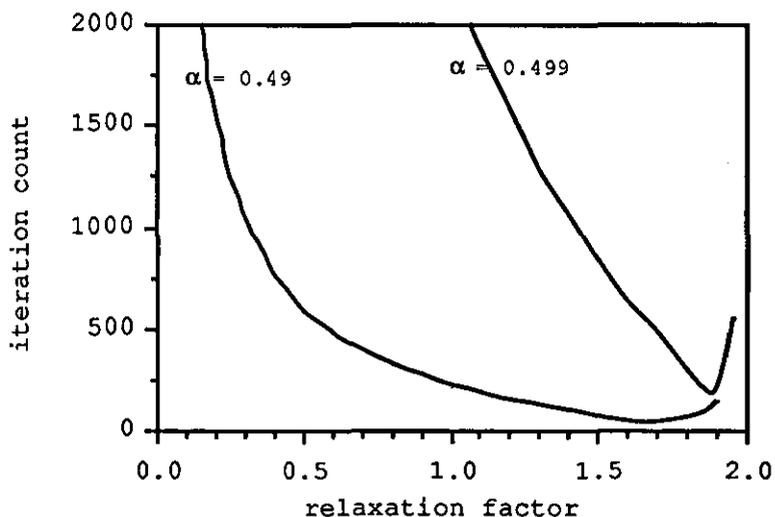
$$\lambda_{\text{opt}} = [\omega\rho]^{n/(n-1)}, \quad (6.6.2-40)$$

where $|\sigma| = |\delta| = \rho$.

Thus we observe that, for a periodic problem, the standard SOR formula is not applicable, although the optimum relaxation factors of both cases coincide ^{asymptotically, when n is large}. This phenomenon is illustrated in the numerical results given in the following section.

6.6.3 NUMERICAL RESULTS

In this experiment, we have used the Laplace equation with Dirichlet boundary conditions (for the SOR method) and with periodic boundary conditions (for the confirmation of the theoretical results derived in the previous section).



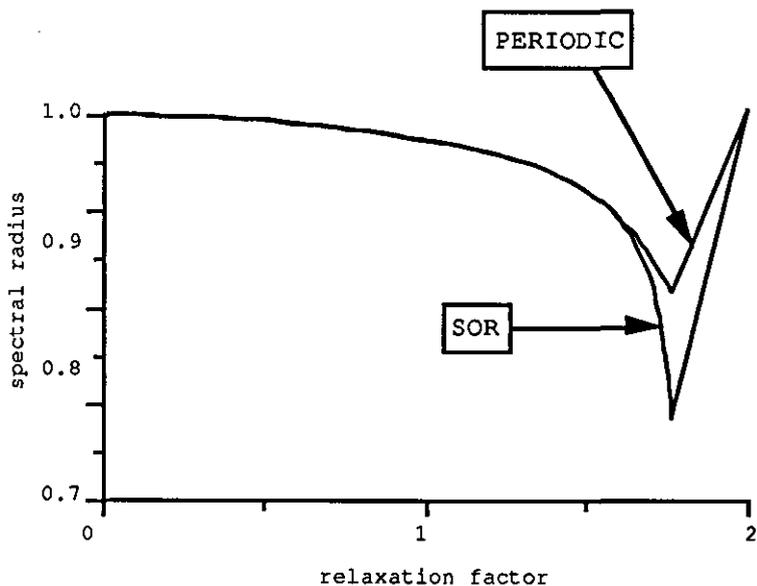
Figure(6.6.3a) : Number of iterations against relaxation parameter for periodic problems. Matrix size 20 and tolerance set at $5e-06$. α denotes the matrix element.

In Table(6.6.3) we present the comparison of the theoretical values of the spectral radius for periodic problem and the spectral radius of the standard SOR method for different matrix elements α . The matrix used is of size 20 and symmetric. The results confirm that the standard SOR is not applicable for the periodic case.

Matrix size 20
Tolerance used 5e-06

Matrix element α	Relaxation parameter ω	Number of iterations	Theoretical values of spectral radius(periodic)	radius(sor)
.10	1.01021	5	.10102E+00	.10205E-01
.15	1.02357	6	.15354E+00	.23573E-01
.20	1.04356	7	.20871E+00	.43561E-01
.25	1.07180	9	.26795E+00	.71797E-01
.30	1.11111	11	.33333E+00	.11111E+00
.35	1.16676	13	.40837E+00	.16676E+00
.40	1.25000	17	.50000E+00	.25000E+00
.42	1.29652	20	.54454E+00	.29652E+00
.44	1.35596	23	.59662E+00	.35596E+00
.45	1.39286	26	.62679E+00	.39286E+00
.48	1.56250	42	.75000E+00	.56250E+00
.49	1.66806	60	.81735E+00	.66806E+00
.499	1.88109	192	.93866E+00	.88109E+00

Table(6.6.3)



Figure(6.6.3b) : Spectral radius against relaxation factor for SOR and periodic problems (for $\alpha = 0.495$).

6.7 CONCLUSIONS AND RECOMMENDATIONS

In this chapter, the iterative and direct methods of solving linear systems of equations derived from the discretization of the periodic boundary-value problems for the elliptic PDE were investigated.

The investigation of the iterative method of the problem leads us to conclude that the standard optimum SOR parameter cannot be applied to the problem. Instead, we derive a new formula for the optimum parameter and hence deduce the rate of convergence of the method for the problem. Numerical experiments are presented to confirm the results.

In the investigation of the direct method of solving the problem, a new tri-reduction (TR3) algorithm is derived the modified form^{of which} is shown to be numerically stable. The reduction stage of the TR3 algorithm seems to be faster than that of the standard cyclic reduction algorithm. Furthermore, this algorithm may also be adapted for parallel computation. This could be an area of further research.

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK

The thesis can be mainly divided into two parts. The first part is concerned with the numerical solution of problems involving ODE. This part is discussed in chapters 4 and 5 after the introductory chapters 1-3. The second part is concerned with the numerical solution of periodic problems involving PDEs which is described in chapter 6.

In chapter 4 we have considered a modified RK method using the geometric mean (GM) principle since we can regard the classical RK method as an arithmetic mean of approximations at various points. If certain conditions of the problems are met, we show that the RK-GM method may give better results compared with the corresponding classical RK method. We illustrate this for the case of a second-order RK-GM method. However, the RK-GM method may be more computationally complex and the region of absolute stability is smaller than the classical RK method. The applicability of the RK-GM method is further investigated in an imbedded form with the classical RK method to develop an adaptive strategy. This seems to be promising since we have two methods of the same order and the combined formula requires only a small number of function evaluations. This is illustrated for the case of an $O(4)$ method. However, the numerical results obtained are less encouraging when compared with the more established methods. Nevertheless, the investigation is worthwhile as confirmed by the comparable numerical results.

In chapter 5, the suitability of the geometric mean approach is further investigated on the multistep

(specifically, the two-step) methods for the numerical solution of special types of problems in ODEs. The geometric mean version of the Numerov method is found to be comparable in accuracy with the classical Numerov method. The two methods are then combined to form an adaptive formula. The prototype adaptive strategy involving the two methods however do not give convincing results. This could be due to the error control strategy used which is too simplistic. Further work in this area is warranted.

The investigation on the multistep method for the fourth-order special type problems results in both new implicit and explicit formulae. The numerical results obtained show good accuracy. Thus they may be combined together to form a predictor-corrector pair for the fourth-order special type ODE problems that occur in celestial mechanics, etc.

Finally, in chapter 6 we are concerned with the numerical solution of periodic problems involving PDEs. The investigation of the optimum parameter of the SOR method applied to periodic problems shows that the standard optimal SOR parameter is not applicable for these problems. A new formula is introduced for the optimal SOR parameter and its asymptotic rate of convergence established which is confirmed by numerical experiments.

The study of a suitable direct method for solving the special systems of linear equations which occur for periodic problems leads to the derivation of a new strategy called the tri-reduction algorithm (TR3). The number of reduction stages of the TR3 algorithm is found to be less than that of the cyclic reduction algorithm. An efficient software implementation of the TR3 algorithm would give better timing results than the cyclic reduction algorithm especially on a parallel

computer. We further study the stability of the TR3 algorithm and derive a stabilized form of this algorithm.

The idea of the TR3 reduction can be further extended on quarto- and quin-reduction algorithms. However, we have to be aware of rounding errors which may grow exponentially. The TR3 algorithm can also be investigated for its suitability as a parallel algorithm.

REFERENCES

- Adams, L.M. [1982], *Iterative algorithms for large sparse linear systems on parallel computers*, Doctoral thesis, Dept. of Applied Maths. and Computer Science, Univ. of Virginia, Charlottesville, Virginia.
- Aiken, R.C.(ed.) [1985], *Stiff computation*, Oxford University Press, Oxford.
- Ames, W.F. [1977], *Numerical methods for partial differential equations*, 2nd edn. Academic Press, New York.
- Arms, R.J., Gates, L. D., and Zondek, B. [1956], *A method of block iteration*, J. Soc. Ind. Appl. Maths., 4.
- Ash, J.H. [1969], *Analysis of multistep methods for special second-order ordinary differential equations*, Ph.D. thesis, University of Toronto.
- Atkinson, K.E. [1978], *An introduction to numerical analysis*, John Wiley & Sons, New York.
- Barwise, J. (ed.) [1988], *Computers and mathematics*, Not. Amer. Math. Soc., 35(9), pp. 1333-1349.
- Bekakos, M.P. [1985], *Design and analysis of parallel algorithms*, M.Sc. thesis, Loughborough University of Technology.
- Bellman, R. [1960], *Introduction to matrix analysis*, McGraw-Hill Book Co. Inc., New York.
- Benson, A. [1969], *The numerical solution of partial differential equations by finite difference methods*, Ph.D. thesis, University of Sheffield, U. K.

- Birkhoff, G., and Lynch, R.E. [1984], *Numerical solution of elliptic problems*, SIAM Studies in Applied Math., Philadelphia.
- Bramble, J.H., Hubbard, B.E., and Zlamal, M. [1968], , SIAM J. Numer. Anal., 5, pp. 1-25.
- Brown, W.S. and Hearn, A.C. [1978], *Application of symbolic algebraic computation*, Proc. Computational Atomic and Molecular Physics Conf., Nottingham, England.
- Bui, T.D., Oppenheim, A.K., and Pratt, D.T. [1984], *Recent advances in methods for numerical solution of ordinary differential equation initial value problems*, J. Comp. Appl. Maths., 11, pp. 283-296.
- Buneman, O. [1969], *A compact non-iterative Poisson solver*, Rep.294, Stanford University Institute for Plasma Research, Stanford, California.
- Burden, R.L., Faires, J.D., and Reynolds, A.C. [1989], *Numerical analysis*, PWS Publishers, Boston, Massachusetts.
- Butcher, J.C. [1987], *The numerical analysis of ordinary differential equations : Runge-Kutta and general linear methods*, John Wiley & Sons, Chichester. .
- Buzbee, B.L. [1973], *A fast Poisson solver amenable to parallel computation*, IEEE Trans.Comp. C-22, pp.793-796.
- Buzbee, B.L., Golub, G.H., and Neilson, C.W. [1970], *On direct methods for solving Poisson's equations*. SIAM J. Numer. Anal., 7(4), pp. 627-656.
- Cash, J.R. [1989], *A block 6(4) Runge-Kutta formula for nonstiff initial value problems*, ACM Trans. Math. Softwr., 15(1), pp. 15-28.

- Cash, J.R., and Karp, A.H. [1990], *A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides*, ACM Trans. Math. Softwr., 16(3), pp. 201-222.
- Ceschino, F. [1962], *Evaluation de l'erreur par pas dans les problèmes différentiels*, Chiffres, 5, pp. 223-229.
- Chan, T.F., and Elman, H.C. [1989], *Fourier analysis of iterative methods for elliptic problems*, SIAM Review, 31(1), pp. 20-49.
- Cloutman, L.D. and Fullerton, L.W. [1977], *Some applications of automated heuristic stability analysis*, report LA-6885-MS, Los Alamos Scientific Laboratory, Los Alamos, N. M.
- Collatz, L. [1966], *The numerical treatment of differential equations*, 3rd edn., Springer, Berlin.
- Cooley, J.W., and Tukey, J.W. [1965], *An algorithm for machine calculation of complex Fourier series*, Math. Comput., 19, pp. 297-301.
- Dalquist, G., Bjorck, A., and Anderson, A. [1974], *Numerical methods*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Dormand, J.R., and Prince, P.J. [1980], *A family of embedded RK formulae*, J. Comp. Appl. Math., 6, pp.19-26.
- England, R. [1969], *Error estimates for Runge-Kutta type solutions to systems of ordinary differential equations*, Comput. J., 12, pp. 166-170.
- Enright, W.H. [1986], *A new error-control for initial value solvers*, Appl. Math. Comput., 31, pp. 288-301.

- Enright, W.H. [1989], *Analysis of error control strategies for continuous Runge-Kutta methods*, SIAM J. Numer. Anal., 26(3), pp. 588-599.
- Eriksson, K., and Johnson, C. [1987], *Error estimates and automatic time control for nonlinear parabolic problems*, SIAM J. Numer. Anal., 24(1), pp.12-23.
- Evans, D.J. [1985], *The solution of an elliptic PDE with periodic boundary conditions in a rectangular region*, Computer Studies 239, University of Technology, Loughborough, U. K.
- Evans, D.J. [1991], *The strides reduction algorithms for solving tridiagonal linear systems*, Intern. J. Computer Math., 41, pp. 237-250.
- Evans, D.J., and Benson, A. [1980], *Iterative solution of the nonlinear parabolic periodic boundary value problem*, Maths. and Comp. in Simul., 22, pp. 113-117.
- Evans, D.J., and Jayes, M.I. [1990], *A new fourth order Runge-Kutta method for the step-by-step integration of ordinary differential equations*, Dept. of Comp. Studies, Int. Rep. 556, University of Loughborough.
- Evans, D.J., and Li, C. [1991], *The recursive tri-reduction method for tridiagonal linear systems*, Dept. of Comp. Studies, Int. Rep. 581, University of Loughborough.
- Fatunla, S.O. [1988], *Numerical methods for initial value problems in ordinary differential equations*, Academic Press, San Diego.
- Fehlberg, E. [1968], *Classical fifth-, sixth-, seventh-, and eight-order Runge-Kutta formulas with step-size control*, NASA Tech. Report 287(1968), extract published in Computing, 4, pp. 93-106(1969).

- 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. 61-71(1970).
- Finkenstein, Graf Finck von, K. [1977, 1978], *Einführung in die numerische Mathematik I + II*, Carl Hanser, Munich.
- Forsythe, G.E., Malcolm, M.A., and Moler, C.D. [1977], *Computer methods for mathematical computations*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Fujishiro, I., Ikebe, Y., Harashima, A., and Watanabae, M. [1989], *A note on cyclic reduction Poisson solvers with application to bioconvection phenomena problems*, *Computers & Fluids*, 17(3), pp. 419-435.
- Gear, C.W. [1971], *Numerical initial-value problems in ordinary differential equations*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Gear, C.W., and Østerby, O. [1984], *Solving ordinary differential equations with discontinuities*, *ACM Trans. Math. Softwr.*, 10(1), pp. 23-44.
- Gekeler, E. [1984], *Discretization methods for stable initial value problems* (Lecture notes in mathematics, 1044), Springer, Berlin.
- George, A. [1973], *Nested dissection of a finite element mesh*, *SIAM. J. Numer. Anal.*, 10, pp. 345-363.
- Gladwell, I., and Wait, R. (eds.) [1979], *A survey of numerical methods for partial differential equations*, Clarendon, Oxford.
- Golub, G.H., and Van Loan, C.F. [1989], *Matrix computations*, 2nd edn., John Hopkins Univ. Press, Baltimore and London.

- Gustafsson, K. [1990], *Control theoretic techniques for stepsize selection in explicit RK methods*, ACM TOMS (to appear).
- Gustafsson, K., Lundh, M., and Soderlind, G. [1988], *A PI stepsize control for the numerical solution of ordinary differential equations*, BIT, 28, pp.270-287.
- Hackbush, W. [1986], *Theorie und numerik elliptischer differentialgleichungen*, Teubner, Stuttgart.
- Hageman, L.A., and Young, D.M. [1981], *Applied iterative methods*, Academic Press, New York.
- Hairer, E., and Warner, G. [1991], *Solving ordinary differential equations II : Stiff and differential-algebraic problems*, Springer-verlag, Berlin, Heidelberg.
- Hairer, E., Norsett, S.P., and Wanner, G. [1987], *Solving ordinary differential equations I, nonstiff problems*, Springer-Verlag, Berlin.
- Hall, G., and Watt, J.M. (eds.), [1976], *Modern numerical methods for ordinary differential equations*, Clarendon Press, Oxford.
- Hearn, A.C. [1991], *REDUCE user's manual version 3.4*, RAND.
- Hegland, M. [1991], *On the parallel solution of tridiagonal systems by wrap-around partitioning and incomplete LU factorization*, Numer. Math., 59, pp. 453-472.
- Heller, D. [1976], *Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems*, SIAM J. Numer. Anal., 13, pp. 484-496.

- Henrici, P. [1962], *Discrete variable methods in ordinary differential equations*, John Wiley & Sons, New York.
- Henrici, P. [1977], *Error propagation for difference methods*, Robert E. Kieger Pub. Co., Huntington, New York.
- Higham, D.S. [1991], *Global error versus tolerance for explicit Runge-Kutta methods*, IMA J. Numer. Anal., 11, pp. 457-480.
- Hockney, R.W. [1965], *A fast direct solution of Poisson's equation using Fourier analysis*, JACM, 12(1), pp.95-113.
- Hockney, R.W. [1970], *The potential calculation and some applications*, Math. Comput. Phys., 9, pp. 135-211.
- Hockney, R.W., and Jesshope, C.R. [1981], *Parallel Computers*, Adam Hilger.
- Hull, T.E., Enright, W.H., and Jackson, K.R. [1976], *User's guide for DVERK - a subroutine for solving non-stiff ODE's*, Dept. of Comp. Sci. Tech. Rep. No. 100, University of Toronto.
- Irons, B. [1970], *A frontal solution program of finite element analysis*, Int. J. Numer. Methods Eng., 2, pp. 5-32.
- Isaacson, E., and Keller, H.B. [1966], *Analysis of numerical methods*, John Wiley & Sons Inc., New York.
- Jain, M.K. [1984], *Numerical solution of differential equations*, 2nd edn., John Wiley, New York.
- Jayes, M.I., and Evans, D.J. [1991], *A new adaptive RK method*, Computer Studies 592, PARC, Univ. of Tech., Loughborough, Leics., U.K.

- Jayes, M.I.B. [1975], *Numerical solution of the fourth boundary value problem for elliptic partial differential equations*, M.Sc. Dissertation, Loughborough University of Technology, U.K.
- Jennings, A. [1977], *Matrix computation for engineers and scientists*, John Wiley & Sons, Chichester.
- Jenson, J., and Niordson, F. [1977], *Symbolic and algebraic manipulation languages and their application in mechanics*, in *Structural Mechanics Software Series*, 1, pp. 541-576, University Press of Virginia.
- Johnson, C. [1988], *Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations*, *SIAM J. Numer. Anal.* 25, pp.908-926.
- Johnson, C., Nie, Y.Y. and Thomee, V. [1990], *An a posteriori error estimate and adaptive timestep control for a backward Euler discretization of a parabolic problem*, *SIAM J. Numer. Anal.*, 27(2), pp. 277-291.
- Johnsson, S.L. [1984], *Odd-even cyclic reduction on ensemble architectures and the solution of tridiagonal systems of equations*, Dept. of Comp. Sci. Tech. Rep. YALE/CSD/RR-339, University of Yale, New Haven.
- Krogh, F.T. [1973], *Algorithms for changing the step size*, *SIAM J. Numer. Anal.*, 10(5), pp. 949-965.
- Lakshmivarahan, L., and Dhall, S.K. [1990], *Analysis and design of parallel algorithms: Arithmetic and matrix problems*, McGraw-Hill Inc., New York.
- 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.

- Lambiotte, J., and Voigt, R. [1975], *The solution of tridiagonal linear systems on the CDC STAR-100 computer*, ACM Trans. Math. Softwr., 1, pp. 308-329.
- Lapidus, L., and Seinfeld, J.H. [1971], *Numerical solution of ordinary differential equations*, John Wiley, New York.
- Li, C. [1989], *Iterative methods for a class of large, sparse, nonsymmetric linear systems*, Ph.D. Thesis, Loughborough University of Technology, U. K.
- Lindberg, B. [1989], *Variable stepsize in Cowell's method*, BIT, 29, pp. 369-371.
- Lotkin, M. [1951], *On the accuracy of RK methods*, MTAC, 5, pp. 128-132.
- Maron, M.J. [1987], *Numerical analysis : A practical approach*, Macmillan Pub. Co., New York.
- Mcree, J.D. [1989], *Two lectures on REDUCE*, Dept. of Math. Phys., Univ. College Dublin.
- Meneguette, M. [1991], *Chawla-Numerov method revisited*, J. Comput. Appl. Math., 36, pp. 247-250.
- Merson, R.H. [1957], *An operational method for the study of integrating processes*, Proc. Symp. Data Processing, Weapon Research Establishment, Salisbury, S. Australia.
- Mikhlin, S.G. [1964], *Linear equations of mathematical physics*, Holt, Rinehart & Winston, New York.
- Miranker, W.L. [1981], *Numerical methods for stiff equations and singular perturbation problems*, Reidel Pub. Co., Dordrecht, Boston, London.
- Mitchell, A.R. [1969], *Computational methods in partial differential equations*, John Wiley & Sons, Chichester.

- Mitchell, A.R., and Griffiths, D.F. [1980], *The finite difference method in partial differential equations*, John Wiley & Sons, Chichester.
- Nakamura, S. [1977], *Computational methods in engineering and science with applications to fluid dynamics and nuclear systems*, John Wiley & Sons, New York, London, Sydney, Toronto.
- Nakashima, M. [1991], *Embedded psuedo-Runge-Kutta methods*, SIAM J. Numer. Anal., 28(6), pp. 1790-1802.
- Ortega, M.J. [1972], *Numerical analysis : A second course*, Academic Press, New York and London.
- Ortega, M.J. [1987], *Matrix theory : A second course*, Plenum Press, New York.
- Ortega, M.J. [1988], *Introduction to parallel and vector solution of linear systems*, Plenum Press, New York and London.
- Parter, S.V. [1965], *On estimating the 'rate of convergence' of iterative methods for elliptic differential equations*, Trans. Amer. Math. Soc., 114, pp.320-354.
- Parter, S.V. (ed.) [1979], *Numerical methods for partial differential equations*, Academic Pres, New York.
- Press, W.H., et al [1986], *Numerical recipes*, Cambridge University Pres, New York.
- Ralston, A. [1962], *Runge-Kutta methods with minimum error bounds*, Math. Comput., 16, pp. 431-437, corr. 17, pg. 488.
- Reid, J.K. [1977], *Sparse matrices*. In *The state of the art in numerical analysis*, (D.Jacobs, ed.), pp. 85-146, Academic Press, New York.

- Rice, J.R. [1977], *Mathematical software 111*, Academic Press Inc., New York.
- Ryna, G. [1987], *REDUCE: Software for algebraic computation*, Springer.
- Sanugi, B.B. [1986], *New numerical strategies for initial value type ordinary differential equations*, Ph.D. thesis, Loughborough University of Technology.
- Sarafyan, D. [1965], *Multistep methods for the numerical solution of ordinary differential equations made self-starting*, Tech. Report, 49, Math. Res. Center, Madison.
- Schumann, U., and Sweet, R. [1976], *A direct method for the solution of Poisson's equation with Neumann boundary conditions on a staggered grid of arbitrary size*, J. Computat. Phys., 20, pp. 171-181.
- Schwarz, H.R. [1989], *Numerical analysis : A comprehensive introduction*, John Wiley & Sons, Chichester.
- Shampine, L. F., Watts, H. A., and Davenport, S.M. [1976], *Solving nonstiff ordinary differential equations - The state of the art*, SIAM Rev., 19(3), pp. 376-411.
- Shampine, L.F. [1977], *Stiffness and nonstiff differential equation solvers, II: Detecting stiffness with Runge-Kutta methods*, ACM Trans. Math. Softwr., 3, pp.44-53.
- Shampine, L.F., and Gordon, M.K. [1975], *Computer solution of ordinary differential equations*, W. H. Freeman and Co., San Francisco.

- Shampine, L.F., and Watts, H.A. [1976], *Global error estimation for ordinary differential equations*, ACM Trans. Math. Softwr., 2, pp. 172-186.
- Shampine, L.F., and Watts, H.A. [1979], *The art of writing a Runge-Kutta code II*, Appl. Math. Comput., Vol.5, pp.93-121.
- Shintani, H. [1968], *Direct solution of partial difference equations for a rectangle*, J. Sci., Hiroshima Univ. Ser. A-I, 32, pp. 17-53.
- Simon, B. [1989], *Wolfram's Mathematica: wonderful, revolutionary, and bug-ridden*, PC Mag., 8(6), pp. 33-36.
- Skeel, R.D. [1986], *Construction of variable-stepsize multistep formulas*, Math. Comp., 47(176), pp. 503-510.
- Smith, G.D. [1985], *Numerical solution of partial differential equations: Finite difference methods*, Oxford Univ. Press.
- Stauffer, et al [1988], *Computer simulation and computer algebra*, Springer.
- Stewart, G. [1973], *Introduction to matrix computations*, Academic Press, New York.
- Swarztrauber, P. [1974], *A direct method for the discrete solution of separable elliptic equations*, SIAM J. Numer. Anal., 11, pp. 1136-1150.
- Swarztrauber, P., and Sweet, R. [1979], *Algorithm 541: Efficient fortran subprograms for the solution of elliptic PDE's*, ACM Trans. Math. Softwr., 5, pp. 352-364.
- Swarztrauber, P.N. [1977], *The methods of cyclic reduction, Fourier analysis, and the facr algorithm*

- for the discrete solution of Poisson's equation on a rectangle, *SIAM Review*, 1978, pp.490-501.
- Swarztrauber, P.N. [1984], *Fast Poisson solvers*, MAA Studies in Numer. Anal., 24, Math. Assoc. America.
- Swarztrauber, P.N. [1987], *Approximate cyclic reduction for solving Poisson's equation*, *SIAM J. Sci. Stat. Comput.*, 8(3), pp. 199-209.
- Sweet, R.A. [1973], *Direct methods for the solution of Poisson's equation on a staggered grid*, *J. Computat. Phys.*, 12, pp. 422-428.
- Sweet, R.A. [1974], *A generalized cyclic reduction algorithm*, *SIAM J. Numer. Anal.*, 11, pp. 506-520.
- Sweet, R.A. [1977], *A cyclic reduction algorithm for solving block tridiagonal systems of arbitrary dimension*, *SIAM J. Numer. Anal.*, 14, pp. 706-720.
- Sweet, R.A. [1988], *A parallel and vector variant of the cyclic reduction algorithm*, *SIAM J. Sci. Stat. Comput.*, 9(4), pp. 761-765.
- Taubes, G.A. [1988], *Physics whiz goes into biz*, *Fortune*, 117(8), pp. 90-93.
- Temperton, C. [1979], *Direct methods for the solution of the discrete Poisson equation: some comparisons*, *J. Computat. Phys.*, 31, pp.1-20.
- Temperton, C. [1980], *On the facr(1) algorithm for the discrete Poisson equation*, *J. Computat. Phys.*, 34, pp.314-329.
- Twizell, E.H. [1984], *Computational methods for partial differential equations*, John Wiley, New York.
- Varga, R.S. [1962], *Matrix iterative analysis*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.

- Vemuri, V., and Karplus, W.J. [1981], *Digital computer treatment of partial differential equations*, Prentice-Hall, Englewood Cliffs.
- Verner, J.H. [1978], *Explicit Runge-Kutta methods with estimates of the local truncation error*, SIAM J. Numer. Anal., 15(4), pp. 772-790.
- Verner, J.H. [1979], *Families of embedded Runge-Kutta methods*, SIAM J. Numer. Anal., 16(5), pp. 857-875.
- Wachspress, E.L. [1966], *Iterative solutions of elliptic systems and applications to the neutron diffusion equations on reactor physics*, Prentice-Hall Inc., Englewood Cliffs, New Jersey.
- Wallace, J. [1991], *A comparison of the cyclic odd-even reduction, tri-reduction and quin-reduction algorithms for the solution of tridiagonal systems of linear equations*, M.Sc. thesis, Loughborough University of Technology, U. K.
- Wayner, P. [1989], *Symbolic math on the Mac*, Byte, 14(1), pp. 239-244.
- Wazwaz, A.M. [1990], *A modified third order Runge-Kutta method*, Appl. Math. Lett., 3(3), pp. 123-125.
- Wolfram, S. [1991], *Mathematica: a system for doing mathematics by computer*, Addison-Wesley Publ. Co., Inc., Reading, MA.
- Young, D.M. [1971], *Iterative solution of large linear systems*, Academic Press, New York.
- Zonneveld, J.A. [1963], *Automatic numerical integration*, Math. Centre Tracts, 8, Mathematisch Centrum, Amsterdam.
- Zwillinger, D. [1989], *Handbook of differential equations*, Academic Press, Inc., Boston.

APPENDICES

Appendix 1

```
C      THIS PROGRAM SOLVES  $Y' = F(X,Y)$  USING RK-GM METHOD
C      OF ORDER 4
      program rkgm4
C      THE FILE prob.f contains the subroutine for the model problems.
$INCLUDE prob.f
C      THE FILE parm4.f contains the subroutine that computes the residual
C      vectors  $fc(1) \dots fc(n)$  for the parameters of the equation.
$INCLUDE parm4.f
      implicit double precision (a-h,o-z)
      double precision x(30),fc(30),par(30),alpha0,alpha1,alpha2
      integer*4 nn(19),nd(19)
      external parm4
      common/blk2/x,par/blk1/n,n1
      common/blk3/x0,xend,y0,npb,nsteps
      open(6,file='data10')
C      set number of equations
      print*,'number of equations n1=23
~      number of unknowns n =19 '
      read*, n1,n
      m = n-10
      read(6,*) (par(i),i = 1,n1),(nn(i),nd(i),i = 1,m)
~      ,alpha0,alpha1,alpha2
      do 10 i = 1,m
          x(i)=dble(nn(i))/nd(i)
10      continue
      write(*,1) (par(i),i = 1,n1),(nn(j),nd(j),j = 1,m)
~      ,alpha0,alpha1,alpha2
1      format(2(2x,10(f4.2,2x)),2x,3(f4.2,2x)//'a1=',i2,'/',i2,
~ 2x,'a2=',i2,'/',i2,2x,'a3=',i2,'/',i2,2x//'b1=',i2,
~ '/',i2,2x,'b2=',i2,'/',i2,2x,'b3=',i2,'/',i2,2x/
~ 'b4=',i2,'/',i2,2x,'b5=',i2,'/',i2,2x,'b6=',i2,'/'
~ ,i2,2x//'alpha0=',f5.2,2x,'alpha1 ='
~ ,f5.2,2x,'alpha2=',f5.2,///)
C      set the parameters of the equation
      a1 = x(1)
      a2 = x(2)
      a3 = x(3)
      b1 = x(4)
      b2 = x(5)
      b3 = x(6)
      b4 = x(7)
      b5 = x(8)
      b6 = x(9)
      x(10) = (-12.d0*alpha2 + 3.d0*alpha1 - 3.d0*alpha0+ 2.d0)/6.d0
      x(11) = 4.d0*alpha2 - alpha1 - alpha0
      x(12) = (-12.d0*alpha2 - 3.d0*alpha1 + 3.d0*alpha0 + 2.d0)/6.d0
      x(13) = (6.d0*alpha2 - 1.d0)/3.d0
      x(14) = (-12.d0*alpha2+3.d0*alpha1-3.d0*alpha0+2.d0)/6.d0
      x(15) = (-12.d0*alpha2-3.d0*alpha1+3.d0*alpha0+2.d0)/6.d0
      x(16) = alpha2
      x(17) = alpha0
      x(18) = alpha1
      x(19) = alpha2
      w1 = x(10)
      w2 = x(11)
      w3 = x(12)
      w4 = x(13)
      w5 = x(14)
      w6 = x(15)
      w7 = x(16)
      w8 = x(17)
```

```

w9 = x(18)
w10 = x(19)
call parm4(fc)
print*, 'CHECK FOR CONSISTENCY'
do 999 i = 1, n1
    write(*, 100) i, fc(i)
100    format(10x, 'f(', i2, ') = ', e20.12)
        print*,
        print*,
999    continue
C    set the number of points
    print*, 'number of points :10'
    read*, nsteps
C    call routine that generates the model problem
250    print*, 'type problem number'
        read*, npb
        if(npb.lt.1) stop 'end of problem'
        call problem
        h = (xend-x0)/nsteps
        xn = x0
        yn = y0
        print*, '      x          exact          computed
~      rel error '
11    hk1 = h*f(npb, xn, yn)
        hk2 = h*f(npb, xn+h*a1, yn+hk1*b1)
        hk3 = h*f(npb, xn+h*a2, yn+hk1*b2+hk2*b3)
        hk4 = h*f(npb, xn+h*a3, yn+hk1*b4+hk2*b5+hk3*b6)
        yn1 = yn+DSIGN(1.d0, f(npb, xn, yn)) * (w1*dsqrt(dabs(hk1*hk2))
~ +w2*dsqrt(dabs(hk2*hk3))+w3*dsqrt(dabs(hk3*hk1))
~ + w4*dsqrt(dabs(hk4*hk1))+w5*dsqrt(dabs(hk4*hk2))
~ +w6*dsqrt(dabs(hk3*hk4)))+w7*hk1+w8*hk2+w9*hk3+w10*hk4
        err = dabs(yn - exact(npb, xn))/(1.d0+dabs(exact(npb, xn)))
        write(*, 12) xn, exact(npb, xn), yn, err
12    format(1x, f6.3, 3(2x, e16.7))
        xn = xn + h
        yn = yn1
        if(xn.gt.xend) then
            go to 250
        else
            go to 11
        endif
        stop
        end

```

Appendix 2

```

program tri-redn
C   The file tr3.f contains the subroutine symp3(a,m,n,b,w,x)
C   algorithm for the TR3 reduction
$INCLUDE tr3.f
      implicit double precision (a-h,o-z)
      parameter (lim = 4000)
      dimension w(lim),b(lim),x(lim)
      external symp3
C   a denotes diagonal element of the matrix A
C   n denotes the size of the matrix A, n = 3**m
C   bb denotes the right-hand side of the equation,
C       Ax = b
      print*, 'enter a,n,bb'
      read*, a,n,bb
      do 100 j = 1,n
         b(j) = bb
100    continue
      m = int(log(real(n))/log(3.))
      print*, 'The number of reduction steps m = ',m
C   Start timing
      call symp3(a,m,n,b,w,x)
C   Stop timing
C   Print final results
C   print*, (x(j), j = 1,n)
      stop
      end

      subroutine symp3(a,m,n,b,w,x)
C   This subroutine solves the simultaneous equation, A*x = b
C   where,
C   a is the diagonal element of A,
C   n is the size of the matrix A,
C   m is any integer number such that n = 3**m,
C   b is the right hand side vector in the equation A*x = b,
C   w is the vector of the multipliers of the diagonal elements of A,
C   during the reduction process,
C   t is the depth of recursion,
C   x is the vector of the unknown in the equation A*x = b.
      implicit double precision (a-h,o-z)
      parameter (lim = 200)
      dimension w(m+1),b(n),x(n),aa(lim,lim)
      integer t,m
      integer time1,time2
      double precision time
C   Start timing
      call _clock_time(time1)
      w(1) = a
      do 10 t = 1,m-1
         w(t+1) = w(t)*(w(t)**2 - 3.d0)
         do 9 j = 3**t,n,3**t
            if (j .ne. (3**m)) then
               b(j) = b(j-2*3**(t-1)) - w(t)*(b(j-3**(t-1))
~                  + b(j+3**(t-1)))
~                  + (w(t)**2 - 1.d0)*b(j) + b(j+2*3**(t-1))
            else
               b(j) = b(j-2*3**(t-1)) - w(t)*(b(j-3**(t-1)) + b(1))
~                  + (w(t)**2 - 1.d0)*b(j) + b(2)
            endif
          9      continue
         10     continue
C   Compute the inverse of the matrix B

```

```

C      where      B*x = b such that
C              |      bb      1      1      |
C              |      |      |      |      |
C      B =      |      1      bb      1      |
C              |      |      |      |      |
C              |      1      1      bb      |
C      and
C              | d*e              -1              -1      |
C              |      |      |      |      |
C      B-1 = (1/d)*| -1              d*e              -1      |
C              |      |      |      |      |
C              | -1              -1              d*e      |
C              d = w(m)**2 + w(m) - 2.d0
C              e = (w(m)**2 - 1.d0)/(w(m)**3 - 3.d0*w(m) + 2.d0)
C      Stop timing
C      call _clock_time(time2)
C      time = (time2 - time1)/100.d0
C      write(*,200)time
200      format(5x,'elapsed time is ',e10.5)
C      Solution process
C      Obtain the values of u(3**(m-1)),
C      u(2*3**(m-1)), and u(3**m) from u = A*b
C      do 12 i = 3**(m-1),n,3**(m-1)
C          x(i) = 0.d0
C          do 11 j = 3**(m-1),n,3**(m-1)
C              if (j .eq. i) then
C                  x(i) = (x(i) + e*b(j))
C              else
C                  x(i) = (x(i) - (1.d0/d)*b(j))
C              endif
11          continue
12      continue
C      Obtain the rest of the u's from
C
C      | c      1 || u1 |      | b1 |
C      |      . ||      | = |      |
C      | 1      c || u2 |      | b2 |
C
C      do 20 i = 1,m-1
C          t = m - i
C          ww = 1.d0/(w(t)**2 - 1.d0)
C          k = 3**t
C          do 15 j = 3**(t-1),n - 3**(t-1),k
C              if (j .ne. 3**(t-1)) then
C                  x(j) = (w(t)*(b(j) - x(j-3**(t-1)))
~                  - (b(j+3**(t-1)) - x(j+2*3**(t-1))))*ww
C                  x(j+3**(t-1)) = (-b(j) - x(j-3**(t-1)))
~                  + w(t)*(b(j+3**(t-1))
~                  - x(j+2*3**(t-1))))*ww
C              else
C                  x(j) = (w(t)*(b(j) - x(n))
C              else
C                  x(j) = (w(t)*(b(j) - x(n))
~                  - (b(j+3**(t-1)) - x(j+2*3**(t-1))))*ww
C                  x(j+3**(t-1)) = (-b(j) - x(n))
~                  + w(t)*(b(j+3**(t-1))
~                  - x(j+2*3**(t-1))))*ww
C              endif
15          continue
20      continue
C      return
C      end

```

Appendix 3

```

program pred_crrtr
C THIS PROGRAM IMPLEMENTS THE PREDICTOR CORRECTOR
C PAIR USING
C 
$$Y_{n+2} = -6*Y_n + 4*Y_{n-1} + 4*Y_{n+1} - Y_{n-2}$$

C 
$$+ (h**4/6)*\{4*F(X_n, Y_n) + F(X_{n-1}, Y_{n-1})$$

C 
$$+ F(X_{n+1}, Y_{n+1})\}$$

C AS THE PREDICTOR AND
C 
$$Y_{n+2} = -6*Y_n + 4*Y_{n-1} + 4*Y_{n+1} - Y_{n-2}$$

C 
$$+ (h**4/720)*\{474*F(X_n, Y_n)$$

C 
$$+ 124[F(X_{n-1}, Y_{n-1}) + F(X_{n+1}, Y_{n+1})]$$

C 
$$- [F(X_{n-2}, Y_{n-2}) + F(X_{n+2}, Y_{n+2})]\}$$

C AS THE CORRECTOR.
C THE PROBLEM SOLVED IS OF THE TYPE
C 
$$D^4Y = F(X, Y)$$

C GIVEN THE INITIAL CONDITIONS  $X_0, Y_0,$  AND  $DY_i, i=1, 2, 3.$ 
implicit double precision (a-h, o-z)
write(*,*) 'INITIAL VALUES OF eps, x0 xend nsteps'
read(*,*) eps, x0, xend, nsteps
write(*, 5) eps, x0, xend, nsteps
5 format (e10.4, 2f6.2, 3x, i3)
xn0 = x0
yn0 = exact(xn0)
h = dabs(xend - x0)/nsteps
xn1 = xn0 + h
xn2 = xn1 + h
xn3 = xn2 + h
yn1 = exact(xn1)
yn2 = exact(xn2)
yn3 = exact(xn3)
write(*, 6) xn1, yn1, xn2, yn2, xn3, yn3
6 format (1x, 3(2(e25.15, 5x)/))
write(*,*) ' xn4 computed
~ exact ',
~' relative error'
C call predictor to obtain yn4
do 10 j = 1, nsteps-3
8 call predic(h, xn1, xn2, xn3, yn1, yn2, yn3, yn0, yn4p)
xn4 = xn3 + h
yn4c = 4.*yn3-6.*yn2+4.*yn1-yn0
~ + (h**4./720.)* (474.*f(xn2, yn2)
~ + 124.*(f(xn3, yn3)+f(xn1, yn1))
~ - (f(xn0, yn0)+f(xn4, yn4p)))
exct = exact(xn4)
abserr = dabs(exct-yn4c)
C test if the required accuracy is satisfied
if (dabs(yn4c-yn4p) .le. eps) then
go to 9
else
go to 8
endif
9 write(*, 100) xn4, yn4c, exct, abserr
100 format (f7.2, 2(e22.15, 2x), e15.9)
write(*,*)
xn0 = xn1
xn1 = xn2
xn2 = xn3
xn3 = xn4
yn0 = yn1
yn1 = yn2
yn2 = yn3
yn3 = yn4c

```

```

10  continue
    stop
    end
C   Define the subroutine to predict the `solution
    subroutine predic(h,xn1,xn2,xn3,yn1,yn2,yn3,yn0,yn4p)
    implicit double precision (a-h,o-z)
    yn4p = 4.d0*yn3 - 6.d0*yn2 + 4.d0*yn1 - yn0
    ~      + (h**4./6.)*(f(xn1,yn1)
    ~      + 4.d0*f(xn2,yn2) + f(xn3,yn3))
    return
    end
C   Define the function f(x,y)
    function f(x,y)
    implicit double precision (a-h,o-z)
    f = y
    return
    end
C   Define the exact solution
    function exact(x)
    implicit double precision (a-h,o-z)
    exact = dexp(x)
    return
    end

```

Appendix 4

```
PROGRAM GM44
C THIS PROGRAM SOLVES AN ODE FIRST ORDER PROBLEM USING
C THE NEW R-K-GM METHOD WITH ERROR CONTROL.
C THE TWO R-K FORMULAE ARE OF THE SAME ORDER 4.
C THE FOLLOWING FILES CONTAIN THE SUBROUTINES
C THAT DEFINE THE MODEL PROBLEMS
$INCLUDE ../PROB.F
$INCLUDE ../ERROR.F
$INCLUDE ../RHS.F
  IMPLICIT DOUBLE PRECISION (A-H,O-Z)
  LOGICAL EXSOL
  INTEGER FOUR, NPB, NEQN
  PARAMETER (LIM = 10)
  DOUBLE PRECISION K(4), Y(LIM), EXACT(LIM), ABSERR(LIM)
  ~ , YPRIME(LIM), TOL, W(LIM),
  ~ X, XEND, ALFASQ, HMIN, HMAX
  COMMON/BLK3/X, XEND, Y
  COMMON/BLK4/ALFASQ
  COMMON/BLK5/NPB
  COMMON/BLK6/NEQN
  COMMON/BLK7/EXACT, ABSERR
  COMMON/BLK8/EXSOL
  EXTERNAL FCN, PROBLEM, ERROR
  READ*, XEND, NPB, NEQN
  RK = .5D0
  FOUR = 4
  TOL = 5.0E-05
  HMAX = 0.1
  HMIN = 0.02
  CALL PROBLEM
  H = (TOL)**(0.25)
  WRITE(*,10) X, Y(1), HMAX, HMIN, TOL
10  FORMAT(17X, 'INITIAL CONDITIONS'/23X
  ~, 'X = ', F5.2/23X, 'Y = ', E10.4
  ~/17X, 'MAXIMUM STEP SIZE IS HMAX = ', E10.4
  ~/17X, 'MINIMUM STEP SIZE IS HMIN = ', E10.4
  ~/25X, 'TOLERANCE', E10.4)
  NFC = 0
  WRITE(*,123)
123  FORMAT(T6, 'X', T15, 'H', T30, 'Y', T43, 'EXACT', T57,
  ~'ABS. ERROR', T74, 'NFC'/T1, 80('-'))
  W(1) = Y(1)
5    IF (X .LE. XEND) THEN
      CALL FCN (NEQN, X, W(1), YPRIME(1))
      K(1) = H*YPRIME(1)
      W(1) = Y(1) + K(1)*RK
      CALL FCN (NEQN, X+H*RK, W(1), YPRIME(1))
      K(2) = H*YPRIME(1)
      W(1) = Y(1) + K(2)*RK
      CALL FCN (NEQN, X+H*RK, W(1), YPRIME(1))
      K(3) = H*YPRIME(1)
      W(1) = Y(1) + K(3)
      CALL FCN (NEQN, X+H, W(1), YPRIME(1))
      K(4) = H*YPRIME(1)
      WAM = (K(1) + 2.D0*(K(2) + K(3)) + K(4))/6.D0
      WGM = DSIGN(1.D0, K(1)) * (DSQRT(DABS(K(1))*DABS(K(2)))
  ~ + DSQRT(DABS(K(1))*DABS(K(3)))
  ~ - DSQRT(DABS(K(1))*DABS(K(4)))
  ~ + DSQRT(DABS(K(2))*DABS(K(4)))
  ~ + DSQRT(DABS(K(3))*DABS(K(4))))/3.D0
      NFC = NFC + FOUR
```

```

ERR = DABS(WAM - WGM)
R = ERR/H
DELTA = 0.84*(TOL/R)**(0.25)
IF (R .LE. TOL) THEN
  X = X + H
  Y(1) = Y(1) + WAM
  W(1) = Y(1)
  CALL ERROR
  WRITE(*,20) X,H,Y(1),EXACT(1),ABSERR(1),NFC
  FORMAT(T2,F7.5,T10,F10.7,T24,E12.6
  ,T38,E12.6,T55,E12.6,T70,I6)
ENDIF
IF (DELTA .LE. 0.1) THEN
  H = 0.1*H
ELSE
  IF (DELTA .GE. 4.D0) THEN
    H = 4.D0*H
  ELSE
    H = DELTA*H
  ENDIF
ENDIF
IF (H .GT. HMAX) THEN
  H = HMAX
ENDIF
IF (H .LT. HMIN) THEN
  HMIN = HMIN/2.D0
  GO TO 5
ENDIF
GO TO 5
ENDIF
STOP
END

```

Appendix 5

```

program num_am_exp
C Numerov type method for the special fourth order ODE.
C (EXPLICIT METHOD)
C
C -----
C THE PROBLEM SOLVED IS OF THE TYPE
C D4Y = F(X,Y)
C GIVEN THE INITIAL CONDITIONS X0 , Y0 ,AND DYi0 ,i = 1,2,3.
C -----
implicit double precision (a-h,o-z)
write(*,*)'INITIAL VALUES OF x0 xend nsteps'
read(*,*)x0,xend,nsteps
y0 = exact(x0)
xn0 = x0
yn0 = y0
h = dabs(xend - x0)/nsteps
write(*,*)
write(*,*)' xn      computed solution
~ exact solution  relative error'
write(*,*)
C calculate y1,y2,y3 using the exact solution
xn1 = xn0 + h
xn2 = xn1 + h
xn3 = xn2 + h
yn1 = exact(xn1)
yn2 = exact(xn2)
yn3 = exact(xn3)
do 10 j = 1 , nsteps
  xn4 = xn3 + h
  yn4 = 4.*(yn1 + yn3) - 6.*yn2 - yn0
  ~      + ((h**4.)/6.)*(f(xn1,yn1)
  ~      + 4.*f(xn2,yn2) + f(xn3,yn3))
  exct = exact(xn1)
  if(exct .ne. 0)then
    abserr = dabs(exct - yn1)/dabs(exct)
  else
    abserr = dabs(exct - yn1)
  endif
  write(*,100)xn1,yn1,exct,abserr
100 format(f6.3,2e20.12,e18.10)
C reset appropriate values of xn0,xn1,xn2,xn3,yn0,yn1,yn2,yn3
  xn0 = xn1
  xn1 = xn2
  xn2 = xn3
  xn3 = xn4
  yn0 = yn1
  yn1 = yn2
  yn2 = yn3
  yn3 = yn4
10 continue
stop
end

```

```

C   Define the function f(x,y)
function f(x,y)
implicit double precision (a-h,o-z)
C1   f = 24.d0 + dexp(x)
C2   f = y
C3   f = 34320.d0*(2.d0 - x)**(-14)
      f = cos(x)
      return
end

C   Define the exact solution
function exact(x)
implicit double precision (a-h,o-z)
C1   exact = x**4 + dexp(x)
C2   exact = dexp(x)
C3   exact = 2.d0*(2.d0 - x)**(-10) - x - 1.d0
      exact = cos(x)
      return
end

```

Appendix 6

```

        program ode2_cases
C
C -----
C THIS PROGRAM INVESTIGATES ALL POSSIBLE CASES OF THE PARAMETERS
C FOR THE NEW GM FORMULA FOR SOLVING THE SPECIAL SECOND ORDER ODE
C PROBLEMS.
C -----
        implicit double precision (a-h,o-z)
        character answer,Y,N
C choose problem number
        write(*,*)
1111 write(*,*) 'PLEASE TYPE THE CORRECT PROBLEM NUMBER,
~ANY OF 1 TO 6'
1 read*,num
  call problem(num,x0,y0,xend,nsteps)
C choose the parameters of the formula, say a = -1/6 or a = -5/12
C for a = 0 gives the Numerov formula
2 do 9999 ll = 1 , 3
  print*, 'INPUT THE NUMERATOR AND DENOMINATOR OF a '
  read*, an, ad
  a = an/ad
  a1 = (12.*a + 5.)/6.
  a2 = (6.*a + 1.)/12.
C a3 = a2 and a4 = a5
  a4 = -2.*a
  a6 = a
  print*,
  print*, 'PARAMETER OF THE EQUATION '
  write(*,3) an, ad, a1, a2, a4, a6
3 format(1x,'a = ',f5.2,'/',f5.2//1x,4(f8.3,2x))
  print*,
  xn0 = x0
  yn0 = y0
  h = dabs(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)+f(num,xn1,yn1))
~ + dsign(1.d0,f(num,xn,yn))*(a4*(dsqrt(dabs(f(num,xn1,yn1))))
~ + dsqrt(dabs(f(num,xn0,yn0))))*dsqrt(dabs(f(num,xn,yn)))
~ + a6*dsqrt(dabs(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
C and computed solutions
  if(exct.ne.0)then
    err = dabs(exct - yn)/dabs(exct)
  else
    err = dabs(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
    write(*,*)'DO YOU WANT TO HAVE ANOTHER TRY,TYPE "Y"
~      IF YES AND "N" IF NO'
    read(*,*)answer
    if (answer.eq.'Y') go to 1111
    stop
    end

C
    subroutine predic(h,xn,yn1,yn,yn0)
    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)
    implicit double precision (a-h,o-z)
    common/blk1/b,c,q
    if(num.eq.1)then
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
    elseif(num.eq.2)then
C      PROBLEM:2  Y'' + 2*X**2*Y = 0
C      INITIAL CONDITIONS X0=0,Y0=1,Y'=1
C      EXACT SOLUTION Y=(1 - X**4/6 + X**8/168 - ...)
C                      + (X - X**5/10 + X**9/360 - ...)
C      CHOOSE SOLUTION DOMAIN [0,1]
    write(*,*)' PROBLEM:2  Y'''' + 2*X**2*Y = 0'
    write(*,*)'  INITIAL CONDITIONS X0=0,Y0=1,Y''=1'
    write(*,*)'  EXACT SOLUTION Y=(1 - X**4/6 + X**8/168 - ...)'
    write(*,*)'          + (X - X**5/10 + X**9/360 - ...)'
    write(*,*)'          CHOOSE SOLUTION DOMAIN [0,1]'
    write(*,*)'INPUT VALUES OF x0 y0  xend nsteps'
    read*,x0,y0,xend,nsteps
    return
    elseif(num.eq.3)then
C      PROBLEM:3  Y'' + X**2*Y = 1 + X + X**2
C      INITIAL CONDITIONS X0=0,Y0=2,Y'=2
C      EXACT SOLUTION Y=2*(1 - X**4/12 + X**8/672 - ...)
C                      + 2*(X - X**5/20 + X**9/1440 - ...)
C      CHOOSE SOLUTION DOMAIN [0,1]
    write(*,*)' PROBLEM:3  Y'''' + X**2*Y = 1 + X + X**2'
    write(*,*)'  INITIAL CONDITIONS X0=0,Y0=2,Y''=2'
    write(*,*)'  EXACT SOLUTION Y=2*(1 - X**4/12
~  + X**8/672 - ...)'
    write(*,*)'          + 2*(X - X**5/20 + X**9/1440 - ...)'
    write(*,*)'          CHOOSE SOLUTION DOMAIN [0,1]'
    write(*,*)'INPUT VALUES OF x0 y0  xend nsteps'
    read*,x0,y0,xend,nsteps

```

```

    return
elseif(num.eq.4)then
C   PROBLEM:4  Y'' - Y = 0
C   INITIAL CONDITIONS X0=0,Y0=1,Y'0=-1
C   EXACT SOLUTION Y=exp(-X)
C   CHOOSE SOLUTION DOMAIN [0,1]
    write(*,*)' PROBLEM:4  Y'''' - Y = 0'
    write(*,*)'   INITIAL CONDITIONS X0=0,Y0=1,Y''0=-1'
    write(*,*)'       EXACT SOLUTION Y=exp(-X)'
    write(*,*)'       CHOOSE SOLUTION DOMAIN [0,1]'
    write(*,*)'INPUT VALUES OF x0 xend nsteps'
    read*,x0,y0,xend,nsteps
    return
elseif(num.eq.5)then
C   PROBLEM:5  Y'' - 220.*(2.-x)**(-12) = 0
C   INITIAL CONDITIONS X0=1,Y0=-2,Y'0=-1
C   EXACT SOLUTION Y=2*(2-X)**(-10)-X-1
C   CHOOSE SOLUTION DOMAIN [0,1]
    write(*,*)' PROBLEM:5  Y'''' - 220*(2-X)**(-12) = 0'
    write(*,*)'   INITIAL CONDITIONS X0=1,Y0=0,Y''0=19'
    write(*,*)'       EXACT SOLUTION Y=2*(2-X)**(-10)-X-1'
    write(*,*)'       CHOOSE SOLUTION DOMAIN [0,1]'
    write(*,*)'INPUT VALUES OF x0 xend nsteps'
    read*,x0,y0,xend,nsteps
    return
elseif(num.eq.6)then
C   PROBLEM:6  Y'' - Y*((Q + B*X)/X)**2 - Q/(X**2) = 0
C   INITIAL CONDITIONS X0=1,Y0=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
function f(num,x,y)
implicit double precision(a-h,o-z)
common/blk1/b,c,q
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

function exact(num,x)
implicit double precision(a-h,o-z)
common/blk1/b,c,q
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.)
~          + 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

```

