SOLUTION OF SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS,USING PREDICTOR - CORRECTOR METHODS

B Y

DAVID ONI OLASEINDE AWOYEMI
B.Sc(HONS)LAGOS, PGDE, ABU

A thesis submitted to the Post–Graduate School, Ahmadu Bello university, Zaria in partial fulfilment of the requirements for the: degree of Master of Science (M.Sc.) in Mathematics.

Department of Mathematics,
Faculty of Science,
Ahmadu Bello University,
Zaria, Nigeria.

1987

DECLARATION

This project is not a published or unpublished work of another researcher. No part of it has been submitted for a degree in any other University. Information obtained from the published and unpublished works of others has been carefully acknowledge in the text.

Signature [redacted]

Date 15/9/87

CERTIFICATION

This thesis titled SOLUTION OF SECOND ORDER ORDINARY DIFFERENTIAL EQUATIONS USING PREDICTOR-CORRECTOR METHODS by David Oni Olaseinde Awoyemi meets the regulation governing the award of the degree of Master of Science (M.Sc.) of Ahmadu Bello University, and is approved for its contribution to scientific knowledge and literary presentation.

DR G. A. OLAYI
Name and Signature
Chairman,
Supervisory Committee

Date: 15/9/87

DR. S.C. MATHUR
Name and Signature of
Supervisor

Date: 15-9-87

Name and Signature
Head, Department of
Mathematics

Date: 17/9/87

Name and Signature
Dean,
Post-Graduate School

Date: _____

DEDICATION

This thesis is dedicated to the memory of my late father, Chief Aluko Awoyemi.

ACKNOWLEDGEMENT

First and foremost, I am very grateful to God Almighty who has made my admission to this course not only possible but in addition provided me all the financial aids and good health necessary for the course.

Secondly, I wish to express my sincere appreciation to my supervisor, Dr. G.A. Olayi for patiently reading through this project many times and coming out each time with useful corrections and new ideas for improvement. His criticisms, suggestions and availability for discussions at every stage of the work were highly cherished. I am equally grateful to Dr. Mathur for reading through the thesis and for his pieces of advice for improvement. My thanks also go to the Head of Department, Dr. U.M. Makarfi, my able lecturers, Professor M.A. Rashid whose method of teaching I cherish so much, Dr. L.M. Strivastava and Dr. (Miss) V.W. Inyang who gave me the necessary basic concepts in Functional Analysis. My gratitude also goes to my friends, Dr. R. Bello, Dr. E.T. Jolayemi, Dr. A.S.S. Dorvlo, Dr. Abayomi Abe, Mr. S.A. Olajide and many others including my colleagues whose names are numerous to mention for lack of space.

Thirdly, my grateful thanks go to the authority of Kaduna Polytechnic for giving me the full opportunity for the course. In particular I wish to express my sincere appreciation to the Rector, Alhaji Y. Aboki, my Head of Department, Mr. R.E. Olaofe, Mr. J.F. Ogundare, Mrs. D.K. Ariko, Mr. Y.O. Aransiola, Mr. J.A. Ojenike, Engr. C.O. Oroge and many others for their support in various ways.

This acknowledgement will not be complete without making a special reference to my 'academic father' the late Dr. Richard Forshaw who, having noticed in me my love for higher education in his Mathematics Education classes for the PGDE, 1980/81 academic session, persistently encouraged me to go for higher course either in Mathematics or Mathematics Education. I am indeed very grateful to him for his fatherly advice. May his soul rest in perfect peace.

Finally, I am gratefully indebted to my mother, wife and children for their understanding during the period of the course. I am also grateful to Mrs. E. Suleiman for patiently typing the thesis.

TABLE OF CONTENTS

	PAGE
Title	i
Declaration	ii
Certification	iii
Dedication	iv
Acknowledgement	v&vi
Tables of Contents	vii&viii
Abstract	ix
List of Flow charts	x
List of Tables	xi&xii
List of Appendices	xiii&xiv
CHAPTER 1. ORDINARY DIFFERENTIAL EQUATIONS ..	1
1.1 Aim of the Project	1
1.2 Definitions of Basic Terms	2
1.3 Historical Development	7
1.4 Application of Ordinary Differential Equations	13
1.5 Steps in Problem Investigation ..	22
CHAPTER 2. CLASSICAL NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS ..	26
2.1 Why Numerical Methods of solutions.	26
2.2 The Classical Methods	29
2.3 Derivation of some of the Basic Methods	35

			PAGE
CHAPTER 3.	SOLUTION OF SECOND ORDER ORDINARY DIFFERFNTIAL EQUATION AS A SYSTEM OF TWO FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS		46
	3.0	Introduction	46
	3.1	Improved Euler Method	54
	3.2	The Classical Runge-Kutta Method	61
	3.3	Adams-Moulton Method	72
	3.4	The Fourth-order Milne Method..	82
	3.5	Taylor series expansion Methods 1 and 2	92
	3.6	The Classical Numerov's Method	116
CHAPTER 4.	COMPARISON OF METHODS AND ANALYSIS OF RESULTS		127
	4.1	Programs for systems 3 and 4 ..	127
	4.2	Comparison of Taylor Series Methods 1 and 2	140
	4.3	Comparison of Execution Times..	150
	4.4	Comparison of Memory Allocations	154
	4.5	Discussion of Numerical Accuracy, Execution Times and Memory Allocations	156
CHAPTER 5.	CONCLUSION AND RECOMMENDATION ..		158
	5.1	Conclusion	158
	5.2	Recommendation..	163
REFERENCES			165
APPENDICES			171

ABSTRACT

in this project, two classes of second order ordinary differential equations, $y = f(x,y,y')$ with initial conditions $y(x_0) = y_0, y'(x_0) = y'_0$ and $y = f(x,y)$ with $y(x_0) = y_0, y'(x_0) = y'_0$ were solved directly or as a system of two first order equations using the classical methods of Euler, Runge-Kutta, Adams-Moulton, Milne, Numerov and Taylor. Comparison of the methods and analysis of results were made.

It was discovered that for the system $y = f(x,y,y')$, Milne's method is better than Adams-Moulton's method in terms of numerical accuracies, execution times and memory allocations for all the step-sizes H considered, while Taylor series methods are found unsuitable for the test problem and similar problems. For the system $y' = f(x,y)$, however, Numerov's method is better than Milne's method, which in turn is still better than Adams-Moulton's method also in terms of numerical accuracies, execution times and memory allocations for all our H especially in problems where y_{K+1} can be made a subject in the Numerov's formula. Here, Taylor series method is found completely unsuitable for the test problem and similar problems.

LIST OF FLOW CHARTS

	PAGE
Flow Charting	48
Flow-Chart i	54
Flow-Chart ii	63
Flow-Chart iii	73
Flow-Chart iv	84
Flow-Chart v	93
Flow-Chart vi	105
Flow-Chart vii	118
Flow-Chart viii	125
Flow-Chart ix	128

LIST OF TABLES

	PAGE
Table 3.1	57
Table 3.2	60
Table 3.3	68
Table 3.4	70
Table 3.5	77
Table 3.6	80
Table 3.7	88
Table 3.8	90
Table 3.9	97
Table 3.10	98
Table 3.11	99
Table 3.12	101
Table 3.13	101
Table 3.14	102
Table 3.15	109
Table 3.16	111
Table 3.17	111
Table 3.18	113
Table 3.19	114
Table 3.20	115
Table 3.21	123
Table 3.22	126

LIST OF TABLES

	PAGE
Table 4.1	131
Table 4.2	131
Table 4.3	133
Table 4.4	133
Table 4.5	135
Table 4.6	136
Table 4.7	137
Table 4.8	138
Table 4.9	140
Table 4.10	142
Table 4.11	143
Table 4.12	145
Table 4.13	148
Table 4.14	148
Table 4.15	152
Table 4.16	153
Table 4.17	154
Table 4.18	155

LIST OF APPENDICES

		PAGE
APPENDIX A		171
	AWO1A	171
	AWO2A	172
	AWO3A	178
	AWO4A	182
	AWO5(R4A)	186
	AWO5(R3A)	189
	AWO5(R2A)	192
	AWO6(D4A)	196
	AWO6(D3A)	199
	AWO6(D2A)	202
	AWO8(EA)	205
	AWO8(RA)	210
OUTPUTS OF	AWO9(EA)	215
" "	AWO9(RA)	217

LIST OF APPENDICES

		PAGE
APPENDIX B		219
OUTPUTS OF	AWO1B	219
" "	AWO2B	221
" "	AWO3B	223
" "	AWO4B	225
" "	AWO5(R4B)	227
" "	AWO5(R3B)	229
" "	AWO5(R2B)	231
" "	AWO6(D4B)	234
" "	AWO6(D3B)	236
" "	AWO6(D2B)	238
	AWO7A	240
	AWO7B	243
OUTPUTS OF	AWO8(EB)	246
" "	AWO8(RB)	248
" "	AWO9(EB)	250
" "	AWO9(RB)	252

CHAPTER 1

ORDINARY DIFFERENTIAL EQUATIONS

1.1 Aim of the Project

As would be observed in section 1.4, problems involving the solution of a system of simultaneous first order differential equations are very common in practice. They occur in economics, ecology, epidermology, pharmacology etc. Besides, any ordinary differential equation of order n can be reduced to a system of n simultaneous first order ordinary differential equations. Unfortunately, the classical methods of solution such as Euler, Runge-Kutta, Adams-Moulton and Milnes could not be used to solve second and higher order ordinary differential equations directly unless they are reduced to system of first order equations.

Despite the importance of simultaneous first order ordinary differential equations, however, there is a dearth of pedagogical materials on the subject. Most text books concentrate on methods of solution of a single first order ordinary differential equation and mention systems of simultaneous first order equations in passing or not at all.

The aim of this project, therefore, is to provide a study of the methods of solving a system of two simultaneous first order ordinary differential equations or the solution of second order ordinary differential equations by reduction to first order systems.

1.2 Definition of Terms:

Differential Equation

An equation involving a relation between an unknown function and one or more of its derivatives is defined by Conte and deBoor(8) as a differential equation. Carnahan, Luther and Wilkes(6), also defined an ordinary differential equation of order n as an equation whose highest derivative is of order n, and ordinary because only total derivatives appear (no partial derivative). Alternatively, the equation is defined to be ordinary if there is only one independent variable, say x. Thus an ordinary differential equation of order n has the form

$$y^{(n)} = f(x,y,y',y'',\ldots,y^{(n-1)}) \quad \ldots (1)$$

Initial value problem

A function y(x) that satisfies (1) implying that it is at least n times differentiable, is a solution of the equation (1). Generally we have many y(x) that satisfy (1). Thus to obtain a unique solution, values of y(x) and/or of its derivatives at some values of x must be given. For an nth-order equation, n such values are normally sufficient to determine a unique solution. If all n conditions are specified at the same value of x, then the problem is termed and initial value problem; see (19) for example. On the other hand when initial and final values of x are given we have boundary-value problems.

Linear and Nonlinear Differential Equations

Differential equations are further classified as linear and nonlinear. The differential equation (1) is said to be linear if f is a linear function of the variables $y, y', \ldots, y^{(n)}$. Linear differential equations possess the important property that if $y_1(x), y_2(x), \ldots, y_n(x)$ are any solution of (1) then so is $c_1y_1(x) + C_2y_2(x) + \cdots + c_ny_n(x)$ for arbitrary constants c_i.

Two solutions y_1 and y_2 of a second order linear differential equation are said to be <u>linearly independent</u>, if the Wronskian of the solution does not vanish, the Wronskian being defined by Conte and deBoor(8) as:

$$W(y_1,y_2) = y_1y_2^1 - y_2y_1^1 = \begin{vmatrix} y_1 & y_1^1 \\ y_2 & y_2^1 \end{vmatrix}$$

The concept of linear independence can be extended to the solutions of equations of order n. If $y_2(x)$, $y_2(x)$, ..., $y_n(x)$ are n linearly independent solutions of a homogeneous differential equation of order n, then $y(x) = c_1y_1(x) + \ldots + c_ny_n(x)$ is called the general solution.

Among linear equations are those with constant coefficients. Thus the n^{th}-order linear differential equation with constant coefficients is of the form

$$L_y = y^{(n)} + a_{n-1}y^{(n-1)} + \ldots + a_oy^{(o)} = 0 \quad \ldots (2)$$

where the a_i are assumed to be real.
If we seek solutions of (2) in the form e^{Bx}, then direct substitution shows that B must satisfy the polynomial equation

$$B^n + a_{n-1}B^{n-1} + \ldots + a_o = 0 \quad \ldots\ldots\ldots\ldots\ldots(3)$$

Equation (3) is called the characteristic equation of the n^{th}-order differential equation (2). If equation (3) has n distinct roots B_i ($i = 1,2, \ldots, n$), then it can be shown that $y(x) = c_1e^{B_1x} + c_2e^{B_2x} + \ldots + c_ne^{B_nx}$ is the general solution of (2) where c_i are arbitrary constants. If $B_1 = a + iB$ is a complex root of (3) so is its conjugate $B_2 = a - iB$. Corresponding to such a pair of conjugate - complex roots are two solutions $y_1 = e^{ax}\cos Bx$ and $y_2 = e^{ax}\sin Bx$ which are linearly independent.

If B_1 is a double root of (3) then $y_1 = e^{B_1x}$ and $y_2 = xe^{B_1x}$ are linearly independent solutions of (2).

Finally if equation (2) is non-homogeneous, i.e. if $Ly = g(x)$ ... (4) and if $h(x)$ is a particular solution of (4) i.e. of $Lh = g(x)$ then the general solution of ( ), assuming that the roots of (3) are distinct, is

$$y = h(x) + c_1e^{B_1x} + c_2e^{B_2x} + \ldots + c_ne^{B_nx}$$

An ordinary differential equation different from the form discussed above is said to be nonlinear.

An example of a nonlinear ordinary differential equation is given by

$$\frac{d^3y}{dx^3} + 2e^x \frac{d^2y}{dx^2} + y\frac{dy}{dx} = x^4 \quad . \quad . \quad . \quad (5)$$

The term $y\frac{dy}{dx}$ in equation (5) makes the equation nonlinear.

Boyce and Diprima (3) also said that a nonlinear equation is more difficult to solve analytically than a linear equation since analytic methods of solution of nonlinear equations are limited. In some cases analytic solution of a nonlinear equation may not exist. However, most of the nonlinear ordinary differential equations could be made linear where possible so that analytic solutions could easily be obtained; and where analytic solutions are not possible numerical solutions are employed.

Stiff and Nonstiff system of Ordinary Differential Equations

A system of ordinary differential equation is also subdivided into stiff and nonstiff ordinary differential equations. Deuflhard (13) defined a stiff ODE system as one whose error propagation along the solution trajectories cannot be described in a realistic way in terms of local Lipschitz constant L.

Thus for two solutions u and v of the autonomous system, $y' = f(y)$, $y(o) = y_o$ ...(6) one has $//u(t) - v(t)// \leq //u(o)-v(o)//$ in some norm independent of L.

On the other hand an ODE system is understood to be nonstiff by (13) if the error propagation along the solution trajectories can be reasonably determined in terms of local Lipschitz constants of the right-hand side f. Thus using (6) $//f(u) - f(v)// \leq L//u - v//$. $u,v \in D$, $f \in C^1$, implies that $//f_y(y)// \leq L$ for some neighbourhood D of the solution graph. If u and v are two solutions of the same ODE system then the fundamental lemma yields $//u(t) - v(t)// \leq //u(o)- v(o)//\exp(Lt)$.

1.3 Historical Development

The development of the theory of differential equations came with the general development of Mathematics and dated back to the beginnings of the Calculus(16) with Isaac Newton (1642-1727) and Gottfried Wilhelm Leibniz (1646-1716) in the seventeenth century. Kline(25) said that Newton did little in the theory of differential equation, but his development of the calculus and the basic principles

of mechanics provided a basis for the development of their applications in the eighteenth century most notably by Euler. The classification of first order equations according to the forms $dy/dx = f(x)$, $dy/dx = f(y)$ and $dy/dx = f(x,y)$ was credited to Newton by Edward(16).

He also said that Newton developed a method of solution for the later equation using infinite series when $f(x,y)$ is a polynomial in x and y, while Leibniz got the fundamental results of the calculus independently a little later than Newton. Scott(32) credited the popular notation dy/dx and integral sign to Leibniz. Leibniz also discovered the method of separation of variable for solving first order homogeneous equations and linear first order equations. Eve(17) also said:

> "Leibniz's notation in the calculus is unquestionably more convenient and flexible than the fluxiona notation of Newton. Many of the elementary rules for differentiation that a student learns early in a beginning course in the calculus were derived by Leibniz. The rule for finding the nth derivative of the product of two functions is still referred to as Leibniz' rule".

Following Newton and Leibniz came the Bernoulli brothers, Jacob(1654-1705), Johann (1667-1748) and Johann's son, Daniel(1700-1782). With the aid of calculus, they formulated as differential equations and solved many problems in Mechanics. One of such problems credited to them by Boyce and DiPrima(3) was the Brachistochrone problem.

By the end of the seventeenth century most of the elementary methods of solving first order ordinary differential equations were known and attention was focussed on higher order equations and partial differential equations. Kline(25) claimed that the search for general method of integrating ordinary differential equation was completed about 1775. Boyce and DiPrima(3) said that Jacopo Riccati (1676-1754), an Italian Mathematician, considered equations of the form $f(y,y',y'') = 0$. He also considered an important nonlinear equation known as the Riccati equation:

$$dy/dx = a_0(x) + a_1(x)y + a_2(x)y^2$$

Leonard Euler (1707-1783), one of the greatest mathematicians of all times made a tremendous contribution in ordinary differential equations. His work on the formulation of problems in Mechanics in Mathematical Language and his development of methods of solving these problems were highly commended by Kline(25).

Euler also considered the possibility of reducing second order equation to a system of first order equations by a suitable change of variables. He also introduced the concept of an integrating factor, gave a general treatment of linear ordinary differential equations with constant coefficients, contributed to the method of power series solutions and gave a numerical procedure for solving differential equations. He also made important contributions to the theory of Fourier series and gave the first systematic discussion of the calculus of variation.

Later in the eighteenth century the great French mathematicians Joseph-Louis Lagrange (1736-1813) and Pierre Simon Laplace (1749-1827) made their own important contributions to the theory of ordinary differential equations. Kline (25) credited to them the first scientific treatment of partial differential equations.

In recent years parts of the effort of mathematicians in the areas of ordinary and partial differential equations has been to develop a rigorous, systematic but general theory. The goal is not so much to construct solutions of particular differential equation, but rather to develop techniques suitable for treating classes of equations.

Apart from analytic methods of solution of ordinary differential equations, other useful and popular methods are the Numerical Methods of solution. These include Picard's iteration method, Taylor's expansion method (1715), Euler's method, Runge-Kutta method which started with the work of Carl Runge (1856-1927) in 1895 and was continued by M.W. Kutta (1867-1944) in 1901, and Adams-Moulton's method to mention a few.

The important step in the development of numerical solutions of ordinary differential equations according to Deuflhard(13) was made by Gragg(23),(24) who found the missing starting step for the central difference discretization (called explicit Euler or explicit mid-point rule). He first proved the existence of two separate h^2 - expansion, for even and odd grid points, respectively. Shortly after that Bulirsch and Stoer(5) implemented a code, which in those days was regarded as a general-purpose ordinary differential equations (ODE) solver. This code, however, could not solve stiff ODE's. Attempts to construct a successful stiff ODE solver of extrapolation type were made by Dahlquist and Lindberg(10),(11). However, due to intrinsic difficulties of the applied implicit trapezoidal rule and the implicit mid-point rule, those attempts did not lead to codes

that were competitive to the Backward Differentiation Formula (BDF) integrators developed by Gear(20). Moreover, because of dificiencies in the stepsize control, the reputation of the nonstiff extrapolation codes also gradually decayed in comparison with the rise of Adams methods according to (34). However, general interest in extrapolation methods once more arose with the suggestion of a new A-stable discretization based on the semi-implicit mid-point rule by Bader and Deuflhard(1). This new discretization allowed the construction of a stiff integrator of extrapolation type which turned out to be a serious competitor for BDF integrators. Moreover, Deuflhard(12) proposed a new order and stepsize control for extrapolation methods, which also brought nonstiff extrapolation integrators back to the international stage.

1.4 Applications of Ordinary Differential Equations

Gellert and others(21) claimed that differential equations are important to both mathematicians and non-mathematicians because of their uses in investigating a wide variety of problems arising in the physical, biological and social sciences. For example the calculation of pendulum oscillations, satellite trajectories, load carrying wings, dams, earthquake tremors, heat propagation, speed of chemical reactions, radio-active decay as well as calculations in electro-technology and ship building use ordinary differential equations. In fact many problems of higher analysis presuppose a knowledge of ordinary differential equations. Such problems include the potential theory, the calculus of variations, the theoretical physics and the partial differential equations. The following examples are typical of some phenomena in which applications of first and second order differential equations arise.

The equations reproduced in (i) - (viii) and (x) are due to Boyce and DiPrima(3) while the one reproduced in (ix) is due to Daniel and Moore(9) and those in (xii) and (xiii) are from Bochner(2).

(i) Radioactive Decay

The simplest equation is of the form:

$\frac{d\theta}{dt} = K\theta$, $\theta(0) = \theta_o$ where $\theta(t)$ is the amount of radioactive substance present at time t, k is a nagative constant of proportionality and $\theta(0)$ is the original amount of the substance present at $t = 0$.

(ii) Chemistry

In Chemistry, a second order chemical reaction involves the interraction (collision) of one molecule of a substance P with one molecule of a substance Q in order to produce one molecule of a new substance X such that P+Q = X.
Such chemical reaction could be described by the differential equation:-

$\frac{dx}{dt} = C(p-x)(q-x)$ where

p and q are the initial concentrations of P and Q respectively.
Also x(t), p-x(t) and q-x(t) are the concentrations of X, P and Q respectively at time t. C is a possitive constant.

(iii) Orthogonal Trajectory

A common geometrical problem is to find the family of curves that intersects a given family of curves orthogonally at each point. The equation describing the slope of each of the curves passing through a given point (x,y) is given by:

$\frac{dy}{dx} = 2kx$ or $\frac{dy}{dx} = cx$

where $c = 2k$ and c is a constant.

(iv) Mechanics

The motion of an object of mass m in a medium that offers a resistance can be described by the equation:

$m \frac{dv}{dt} = mg - kv$

where kv is the resistance to the motion and g is the acc leration due to gravity.

Similarly, the motion of a body of mass m projected upward from the earth's surface with an initial velocity v_o, and assuming there is no air resistance, but the earth gravitational field which varies with

the altitude, can be described by the differential equation:

$$m \frac{dv}{dt} = - \frac{mgR^2}{(x+R)^2}$$

$$\text{or } v \frac{dv}{dx} = - \frac{gR^2}{(x+R)^2} ,$$

$$v(o) = v_o$$

where R is the radius of the earth and x is the altitude above sea level.

Also the differential equation describing the motion of a space vehicle of initial mass m_o projected upward from the surface of the earth with an initial velocity v_o is given by:

$\frac{dv}{dt} = -g + \frac{sB}{m_o - Bt}$ where B is the constant rate of fuel consumption, s is the exhaust speed, g is the acceleration due to gravity and t is the time of motion.

(v) Brachistochrone problem

The differential equation describing the famous problem in the history of mathematics - Brachistochrone problem which is to find the curve along which a particle will slide without friction in the minimum

time from one given point to another is given by

$\left[1 + \left(\frac{dy}{dx}\right)^2\right] y = k^2$, where K is a positive constant to be determined.

(vi) Medicine

Epidemic e.g. Typhoid

A simple mathematical model that could be used to describe epidemic phenomonon is given by $\frac{dx}{dt} = Bxy$, $x(o) = x_o$ where x and y denote the proportions of sick and well members respectively such that $x + y = 1$. x and y are continuous and B is a positive constant of proportionality.

Alternatively, if x and y represent the proportion of carriers and of well, but susceptible, people in the population at time t, and the carriers are removed from the population at a rate B, then the differential equation describing that aspect is given by

$\frac{dx}{dt} = -Bx$, $x(o) = x_o$

If the disease spreads at a rate proportional to the product x and y, we have

$\frac{dy}{dt} = -axy$, $y(o) = Y_o$. a is a positive constant.

(vii) Social Sciences

Differential equations are also used in social science, e.g. Business studies and Banking, to describe important physical phenomena. For example in Banking a simple mathe - matical model that can be used to compute amount accruing from an initial investment, S_o over a period of time t is given by

$\frac{dS}{dt} = RS$, $S(o) = S_o$, where

S_o is the sum of money deposited in the Bank at the rate R percent compound interest for t years. S(t) is the investment at the end of t years.

(viii) Engineering

Important areas of application of the theory of second-order linear differential equation are in the field of Engineering and Physical Sciences. In these areas a large number of physical phenomena are governed by equations of the forms

$$m \frac{d^2y}{dx^2} + c \frac{dy}{dx} + ky = f(x) \quad \ldots (6)$$

where m, c and k are constants and

f is a given furction of x, and

$$p(x)\frac{d^2y}{dx^2} + q(x)\frac{dy}{dx} + r(x)y = S(x) \quad \ldots(7)$$

where the coefficients P(x), q(x), r(x) and S(x) are functions of the independent variable x. The Bessel equation

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2-a^2)y = 0$$

where a is a constant and the Legendre equation

$$(1-x^2)\frac{d^2y}{dx^2} - 2x\frac{dy}{dx} + r(r+1)y = 0$$

where r is a constant belonging to the class of equation (7).

Similarly, the test problems for the project appearing in Chapter 3 under Section 3.0 belong to these classes of equations. Specifically, the test problem (5) belongs to the class of equation (7) above while the test problem (6) belongs to the class of equation (6) also above when c = o

(ix) Weather Forecasting

The study of weather forecasting can be described by the equation given by

$$\frac{d^2y}{dt^2} = -\frac{\partial P(y)}{\partial T}, \quad y \text{ in } S_2CE^3$$

where P is a non-constant pressure distribution over the earth's surface assumed to be the sphere

$$S_2 = \{(y_1,y_2,y_3);\ y_1^2 + y_2^2 + y_3^2 = 1 \text{ in } E^3,$$

$\frac{P(y)}{T}$ is a continuous time-independent vector field on S_2.

(x) Attraction of Spheroid

The differential equations describing this phenomenon include the Legendre's equation of order r given by

$$(1-x^2)\frac{d^2y}{dx^2} - 2x\frac{dy}{dx} + r(r+1)y = 0.$$

The Legendre functions which are solutions of the differential equation bearing his name arose from his study of the attraction of spheroids.

(xi) Problems involving Circular Geometry

The Bessel's equation of order k is given by

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - k^2)y = 0$$

where k is a constant, often an integer, arises in many different physical

situations, particularly in problems involving circular geometry such as the determination of the temperature distribution in a circular plate. The solutions to Bessel's equations are known as Bessel functions.

(xii) Algebraic Topology

Algebraic Topology was stimulated by a formular due to L. Kronecker (1823 - 1893). The so called Kronecker Integral is given by the joint solutions of the following three independent functions: $U = f(x,y,z)$, $V = g(x,y,z)$, $W = h(x,y,z)$ in a region of three-dimensional Euclidean space, and similarly for higher dimentional space.

(xiii) Tensorial Function

Tensorial functions make use of partial differential equations for their solutions. For example if we let $g(u,v)$ denote the height of a building erected over a domain D in the (u,v) - plane as base and we introduce a transformation of co-ordinates.

$$U = \phi(x,y), \quad V = \psi(x,y) \quad \ldots \quad (8)$$

then the height becomes

$$f(x,y) = g(\phi(x,y), \psi(x,y)) \quad \ldots \quad (9)$$

Now the shape of the roof is approximately described by the two functions.

$$\frac{\partial g}{\partial u}, \quad \frac{\partial g}{\partial v} \qquad \ldots \quad (10)$$

In the co-ordinate x,y the corresponding functions are

$$\frac{\partial f}{\partial x}, \quad \frac{\partial f}{\partial y} \qquad \ldots\ldots \quad (11)$$

and linked to the functions (10) by the relations

$$\left.\begin{aligned} \frac{\partial f}{\partial x} &= \frac{\partial g}{\partial u}\frac{\partial \phi}{\partial x} + \frac{\partial g}{\partial v}\frac{\partial \psi}{\partial x} \\ \frac{\partial f}{\partial y} &= \frac{\partial g}{\partial u}\frac{\partial \phi}{\partial y} + \frac{\partial g}{\partial v}\frac{\partial \psi}{\partial y} \end{aligned}\right\} \quad \ldots \quad (12)$$

Hence in order to obtain either one of the two functions(11), both functions (10) must be known. In this sense the pair (10) and (11) cannot be broken up into singles. Such groupings of functions are called tensors.

1.4 Steps in Problem Investigation

To investigate a problem using differential equation Boyce(3) and Maki(26) suggested the following steps:
First it is necessary to translate the physical situation into mathematical model-defined by Olayi(28) as a

"set of assumptions about the data/phenomenon together with the rules of mathematical reasoning which permits us to deduce implications from the assumptions". To do this we need to make suitable assumptions that are relevant to the observed phenomena. For example it has been observed that radioactive materials decay at a rate proportional to the amount of the material present, that object move about in accordance with Newton's law of motion and that isolated insect populations grow at a rate proportional to the current population. Each of these statements concerns a rate of change or derivative and consequently when expressed in mathematical terms takes the form of a differential equation.

Secondly, it is important to realise that the mathematical model so formulated is merely an approximate descriptions of the actual situation since the model is ased on observations which are themselves approximations. For example a body moving at the speed comparable to the speed of light is not governed by Newton's law, insect populations do not grow indefinitely as stated because of limitations of their food supply.

Moreover, the process of formulating a physical problem mathematically often involves the conceptual replacement of a discrete process by a continuous one. For instance, the number of members in an insect population changes by discrete amounts which is contrary to our assumption. Although if the population is large, it is reasonable to consider it as continuous variable and thus think of its derivative.

Thirdly, we have to solve our mathematical equations or at least investigate as much as possible about the properties of the solution - whether a solution exists and it is unique and depends continuously on the data of the problem. The initial condition must also be formulated so that a unique solution to the problem could be obtained. If the equation is difficult to solve, the mathematical model may need to be simplified further. For example, if the equation is nonlinear it may be approximated to a linear one. Although any such approximations must be thoroughly examined to make sure that the resulting mathematical model still reflects the essential features of the physical process under investigation. A good knowledge of the physics of the problem will be an asset at this stage.

Also Scheid(31) claimed that a reduction of an ordinary differential equation of order n ($n \geq 1$) to a system of first order equations may simplify the problem. For example, consider the second-order ordinary differential equation (Bessel's equation)

$$x^2 \frac{d^2y}{dx^2} + x \frac{dy}{dx} + (x^2 - a^2)y = 0$$

where a is a constant. By defining one new variable $p = \frac{dy}{dx}$, the second-order equation becomes a pair of first-order equations.

$$\frac{dy}{dx} = p$$

$$x^2 \frac{dp}{dx} + xp + (x^2 - a^2)y = 0$$

which may be simpler to solve than the original second order equation.

Finally, after a solution has been obtained, it must be interpreted in relation to the context in which the problem arose.

CHAPTER 2

CLASSICAL NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS

2.1 Why Numerical Methods of solution

Although many important differential equations can be solved by well known analytical techniques, a greater number of physically significant differential equations cannot be so solved. Example of such equations include nonlinear differential equations. The free oscillation of a pendulum given by the equation

$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin\theta = 0$ (1) is a nonlinear differential equation whose exact solution cannot be obtained analytically except θ is very small so that $\sin\theta = \theta$. In this case equation (1) becomes linear. Another example is a problem in Civil Engineering where an approximation to the rise-time for a reservoir is expressed as

$\frac{dR}{dh} = \frac{4000}{10-20h^{3/2}}$ for h = 0.25 to 0.5 . Here

$$R = \int_{2.5}^{0.5} \frac{4000}{10-20h^{3/2}}\, dh \qquad \ldots \qquad (2)$$

The integral on the right hand side of (2) cannot be evaluated easily analytically. Also the differential equation $\frac{dy}{dx} = \sqrt{\frac{x}{x^2-a^2}}$

where a is a constant, cannot be solved without recourse to numerical method. Even if some of the equations can be integrated directly, it may be more difficult to evaluate the analytic solutions than to solve the original initial value problems numerically. For example, the analytic solution may be in the form of a complicated implicit relation $F(x,y) = 0$ that cannot be evaluated explicitly. Another situation is that in which the mathematical model of a system is known but exact solution cannot be obtained. An example is the Blasius problem - the boundary layer on a flat plate - in Fluid Mechanics. After the desired simplification of the problem it reduces to a single nonlinear ordinary differential equation given by Carnahan and others(6) as

$2\frac{d^3f}{ds^3} + f\frac{d^2f}{ds^2} = 0$, the exact solution of

which cannot be obtained analytically. Hence a numerical solution of such problems must be preferred. Another situation where a numerical solution is necessary is a situation where a problem is known or defined only at discrete point. Even when analytic solution can be easily found the advent of the computer has made numerical solution a lot easier than analytic solutions. For instance, it is better and faster to employ numerical methods for differential equations whose solutions are in infinite series forms. A typical example of such equation is the Bessel equation.

$x^2 \frac{d^2y}{dx^2} + x \frac{dy}{dx} + (x^2-a^2)y = 0$ where

$y(x) = x^p \sum_{i=0}^{?} Ki\ Xi$ may be used to represent the entire solution family analytically.

It should be noted that a numerical approximation is useless unless there is some way to estimate the error of the method.

2.2 The Classical Methods

2.2.1 Taylor's Method

A procedure for stepping from one value of z to another, that is, from x_i to x_{i+1} is based on Taylor's expansion and is given by Carnahan and others(6) as

$$y(x_{i+1}) = y(x_i+h)=y(x_i)+hf(x_i,y(x_i)) + \frac{h^2 f'(x_i,y(x_i))}{2i} + \ldots\ldots\ldots +$$

$$\frac{h^n f^{n-1}(xi,\ y(xi))}{ni} + \frac{h^{n+1} f^n(k,y(k))}{(n+1)i} \quad \ldots \quad (3)$$

K in (x_i,x_{i+1}).

Equation (3) is a Taylor's expansion of order n with a truncation error of order h^{n+1}.

To evaluate the higher order derivatives of (3) we have to differentiate f(x,y) by chain rule since f is a function of both x and y. Thus if $y' = f(x,y)$, then $y'' = f'(x,y) = f_x(x,y)+ f_y(x,y)f(x,y)$

y''' could be obtained similarly.

2.2.2 Euler's Method

If in equation (3) we let y_i be the approximate value of $y(x_i)$ and we retain the first two terms plus an error term, then

$$y_{i+1} = y_i + hf(x_i,y_i) + \frac{h^2 f(K,\ y(K))}{2!} \quad \ldots(4)$$

which Carnahan and others(6) also give as Euler's formula. The error term is called truncation error and is of order two, that is the product of h^2 and a constant. The improved Euler's formula is given by $y_{i+1} = y_i + \frac{h}{2}\,(f(x_i,y_i) + f(x_i+h,\bar{y}_{i+1}))$ where $\bar{y}_{i+1} = y_i + hf(x_i,y_i)$. The error term is of order three. That is $e_T = kh^3$ where k is a constant. The one-step algorithm called the improved polygon method or the modified Euler's method is given by

$y_{i+1} = y_i + hf(x_i+\frac{h}{2},\ \bar{y}_{i+\frac{1}{2}})$ where

$\bar{y}_{i+\frac{1}{2}} = y_i + \frac{h}{2}f(x_i,y_i)$. The error term is also of order three.

2.2.3 Runge-Kutta Method

In this method one-step procedure which involves only first-order derivative evaluations is developed, but which also produces results equivalent in accuracy to the higher-order Taylor's formulas. Approximations with accuracies equivalent to Taylor's expansion of y(x) retaining terms in h^2,h^3 and h^4 respectively require the estimation of f(x,y) at two, three and four values respectively of x on the interval $x_i \leq x \leq x_{i+1}$

Second order Runge-Kutta method is given by

$$y_{i+1} = y_i + \frac{h}{2}(k_1+k_2) \quad \ldots \quad (5)$$

where $k_1 = f(x_i,y_i)$

$$k_2 = f(x_i+h, y_i + hf(x_i,y_i))$$

$$= f(x_i+h, y_i + hk_1)$$

which is the same as the improved Euler's formula or Heun's formula with the truncation error $e_T = kh^3$.

Similarly, the fourth-order Runge-Kutta method is given by

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad \ldots \quad (6)$$

where $k_1 = f(x_i, y_i)$

$$k_2 = f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}h\, k_1)$$

$$k_3 = f(x_i + \tfrac{1}{2}h,\ y_i + \tfrac{1}{2}h\, k_2)$$

$$k_4 = f(x_i + h,\ y_i + hk_3)$$

Equation (6) is the classical - Runge-Kutta formular. It is equivalent to a five-term Taylor's formula. Hence the truncation error is of order 5 and is given by $e_T = kh^5$.

It should be noted that without some measure of the truncation error, it will be difficult to choose the proper step size h. A rough but useful rule by Collatz(15) and Ralston(29) is as follows:

If $\left|\frac{k_2 - k_3}{k_1 - k_2}\right|$ becomes large (more than a a few hundredths), h should be decreased usually by half.

Scraton(15) also has shown that an estimate of the truncation error can be obtained by one additional evaluation of f(x,y). This may however, be costly if f(x,y) is a complicated function.

Other numerical methods given by Carnahan and others(6) for solving differential equations $y' = f(x,y)$ and higher orders are the multistep methods.

These methods make use of the knowledge of y_{i-3} at x_{i-3}, y_{i-2} at x_{i-2}, y_{i-1} at x_{i-1} and y_i at x_i to determine y_{i+1} at x_{i+1}.

Listed below are some of the most popular multistep or predictor-corrector methods.

2.2.4(a) Fourth-order Milne Method

Predictor:

$$y_{i+1} = y_{i-3} + \frac{4h}{3}(2f_i - f_{i-1} + 2f_{i-2}) \quad \ldots (7)$$

The error term is of order five and is given by $e_T = kh^5$

Corrector:

$$y_{i+1} = y_{i-1} + \frac{h}{3}(f_{i+1} + 4f_i + f_{i-1}) \quad \ldots (8)$$

Similarly $e_T = kh^5$

(b) Sixth-order Milne Method

Predictor:

$$y_{i+1} = y_{i-5} + \frac{3h}{10}(11f_i - 14f_{i-1} + 26f_{i-2} - 14f_{i-3} + 11f_{i-4}) \quad \ldots (9)$$

$e_T = kh^7$

Corrector:

$$y_{i+1} = y_{i-3} + \frac{2h}{45}(7f_{i+1} + 32f_i + 12f_{i-1} + 32f_{i-2} + 7f_{i-3}) \quad \ldots (10)$$

$e_T = kh^7$

(c) Fourth-order Hamming Method

Corrector:

$$y_{i+1} = \frac{9}{8}y_i - \frac{1}{8}y_{i-2} + \frac{3}{8}h\,(f_{i+1} + 2f_i - f_{i-1}) \quad \ldots \quad (11)$$

Predictor:

This algorithm uses the fourth order Milne's method predictor listed in equation (7) above.

2.2.5 Adam - Moulton Method

Predictor

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \quad \ldots \quad (12)$$

$$e^T = kh^5$$

Corrector:

$$y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}) \quad \ldots \quad (13)$$

$$e_T = kh^5$$

2.2.6 Numerov's Method

This is a method for solving second order ordinary differential equations of the form $\frac{d^2y}{dx^2} = F(x,y)$ in which the first derivative does not appear explicitly.

Numerov's formula is given by

$$y_{i+1} = 2y_i - y_{i-1} + \frac{h^2}{12}(F_{i+1} + 10F_i + F_{i-1}) \ldots (14)$$

2.3 Derivation of some of the Basic Methods

2.3.1 Runge-Kutta's Method

All Runge-Kutta's methods are of the form $y_{i+1} = y_i + h\phi(x_i, y_i, h)$ ... (15) where ϕ is the increment function. ϕ is a suitably chosen approximation to $f(x,y)$ on the interval $x_i \leq x \leq x_{i+1}$.

Let ϕ be a weighted average of two derivative evaluations k_1 and k_2 on the interval $x_i \leq x \leq x_{i+1}$, that is

$$\phi = ak_1 + bk_2 \ldots (16)$$

so that (15) becomes

$$y_{1+1} = y_i + h(ak_1 + bk_2) \ldots (17)$$

let $k_1 = f(x_i, y_i)$

$$k_2 = f(x_i + ph, y_i + qhf(x_i, y_i))$$

$$= f(x_i + ph, y_i + qhk_1) \ldots (18)$$

where p and q are constants to be determined later. First we expand k_2 in Taylor's series for a function of two variables and drop all terms with the power of h greater than one.

Thus $k_2 = f(x_i+ph,\ y_i+qhf(x_i,y_i)) + O(h^2)$...(19)

From equations (18) and (19),(17) becomes

$$y_{i+1} = y_i + h(af(x_i,y_i) + bf(x_i,y_i)) + h^2(bpf_x(x_i,y_i) + bqf(x_i,y_i)f_y(x_i,y_i)) + O(h^3) \quad \ldots \quad (20)$$

Next we expand the object function y(x) about x_i using Taylor's series.

i.e. $y(x_i+h) = y(x_{i+1}) = y(x_i) + hf(x_i,y(x_i)) + \frac{h^2}{2!}f'(x_i,y(x_i)) + \frac{h^3}{3!}f''(k,y(k))$,

k in (x_i, x_{i+1}) ... (21)

By chain rule

$f'(x_i,y(x_i)) = f_x(x_i,y(x_i)) + f_y(x_i,$

$y(x_i))\ f(x_i,y(x_i))$ ... (22)

putting (22) in (21) and equating terms of like powers of h in (20) and (21), we have:

Power of h	Expansion of y(x)	Runge-Kutta Method
0	$y(x_i)$	Y_i
1	$f(x_i,y(x_i))$	$(a+b)\ f(x_i,y_i)$
2	$\frac{1}{2}(f_x(x_i,y(x_i)) + f_y(x_i,y(x_i))f(x_i,y(x_i)))$	$(bpf_x(x_i,y_i) + bqf_y(x_i,y_i)f(x_i,y_i))$

Assuming $y_i = y(x_i)$ and coefficients of terms in h^2 are equal, then

$$\left.\begin{aligned} a + b &= 1 \\ bp &= \tfrac{1}{2} \\ bq &= \tfrac{1}{2} \end{aligned}\right\} \quad \ldots \quad (23)$$

Equation (23) contains four variables and therefore cannot be solved exactly.

$\therefore$ b must be given an arbitrary value. The two common choices for b are $b = \frac{1}{2}$ and 1 respectively.

For $b = \frac{1}{2}$, $a = \frac{1}{2}$, $p = 1$ and $q = 1$

Then (17) becomes

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i,y_i) + f(x_i + h, y_i + hf(x_i,y_i))\right) \ldots (24)$$

Or

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i,y_i) + f(x_{i+1}, \bar{y}_{i+1}\right) \quad \ldots \quad (25)$$

where

$$\bar{y}_{i+1} = y_i + hf(x_i,y_i) \quad \ldots \quad (26)$$

The one-step algorithm of (25) and (26) is known as the improved Euler's method or Heun's method.

Again for b = 1, a = 0, P = ½ and q = ½

Then (17) becomes

$$Y_{i+1} = Y_i + hf(x_i + \frac{h}{2}, \bar{Y}_{i+\frac{1}{2}}) \quad \ldots \quad (27)$$

where

$$\bar{Y}_{i+\frac{1}{2}} = Y_i + \frac{h}{2}f(x_i, Y_i) \quad \ldots \quad (28)$$

The one-step algorithm of (27) and (28) is called the improved Polygon method or the modified Euler's method.

The higher order Runge-Kutta methods are similarly developed.

2.3.2 Adams-Moulton's Method: Predictor

Adams-Moulton Predictor formula is given by Dodes (14) as

$$PreY_{i+1} = a_o Y_i + h(b_o f_i + b_1 f_{i-1} + b_2 f_{i-2} + b_3 f_{i-3}) \ldots (29)$$

while the corrector has the form:

$$CorY_{i+1} = a_o Y_i + h(b_{-1} f_{i+1} + b_o f_i + b_1 f_{i-1} + b_2 f_{i-2}) \ldots (30)$$

The coefficients are not the same in (29) and (30) We use the method of undetermined coefficients to find the coefficients of (29). Since there are five coefficients, we need five equations.

Step 1. Make (29) exact for $y = 1$. Then

$y' = f = 0$.

Substituting in (29)

$1 = a$. ... (31)

Step 2. Make (29) exact for $y = x$. Then

$y' = f = 1$. For ease of computation, we assume that f has been translated so that $x_i = 0$. Then $y_i = x_i = 0$, $y_{i+1} = x_i + h = h$, and so on.

substituting in (29)

$$h = h(b_o + b_1 + b_2 + b_3) \quad \ldots \quad (32)$$

$$1 = b_o + b_1 + b_2 + b_3 \quad \ldots \quad (33)$$

Step 3. Make (29) exact for $y = x^2$. Then

$y' = f = 2x$. Also, $y_i = (x_i)^2 = 0$

$y_{i+1} = (x_i + h)^2 = h^2$

$y'_i = f_i = 2x_i = 0$. $y'_{i-1} = f_{i-1} = 2(x_i - h) = -2h$ and so on. Then

$$h^2 = h(-2hb_1 - 4hb_2 - 6hb_3) \quad \ldots \quad (34)$$

$$1 = -2b_1 - 4b_2 - 6b_3 \quad \ldots \quad (35)$$

Step 4. Make (29) exact for $y = x^3$. Then

$y' = f = 3x^2$. Also, $y_{i+1} = (x_i+h)^3 = h^3$

$y'_i = f_i = 0$, $y'_{i-1} = f_{i-1} = 3(x_i-h)^2 = 3h^2$

and so on. Then

$$h^3 = h(3h^2b_1 + 12h^2b_2 + 27h^2b_3) \quad \ldots(36)$$

$$1 = 3b_1 + 12b_2 + 27b_3 \quad \ldots \quad (37)$$

Step 5. Make (29) exact for $y = x^4$. The result is

$$1 = -4b_1 - 32b_2 - 108b_3 \quad \ldots \quad (38)$$

Collecting equations (31),(33),(35), (37) and (38) and solving them simultaneously we have

$$a_o = 1,\ b_o = \frac{55}{24},\ b_1 = -\frac{59}{24},$$

$$b_2 = \frac{37}{24},\ b_3 = -\frac{3}{8},$$

leading to the predictor formula:

$$\text{Pre } y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \quad \ldots \quad (39)$$

Using the same procedure the corrector formular given by

$$\text{Cor}y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}) \quad \ldots (40)$$

could be similarly obtained.

Calculation of Error in Adams-Moulton's Method

The error can be calculated in several ways (see Hamming, pp.144-151,196). For our purpose we shall use Taylor series method to estimate the error.

Thus if y_{i+1} is the true value, the error in the Predictor formula is given by

$$E = y_{i+1} - (y_i + \frac{h}{24}(55y'_i - 59y'_{i-1} + 37y'_{i-2} - 9y'_{i-3})) \quad \ldots \quad (41)$$

$$E = (y_{i+1} - y_i) - \frac{h}{24}(55y'_i - 59y'_{i-1} + 37y'_{i-2} - 9y'_{i-3})$$

By Taylor's theorem, expansion about $x_i = 0$ gives

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2}y'' + \frac{1}{6}h^3y'''_i + \frac{1}{24}h^4y^{iv} + \frac{1}{120}h^5y^v_i + \ldots\ldots \quad \ldots \quad (42)$$

Also

$$y'_{i-1} = y'_i - hy'' + \tfrac{1}{2}h^2y''' - \frac{1}{6}h^3y^{iv} + \frac{1}{24}h^4y^v_i \quad \ldots(43)$$

$$y'_{i-2} = y'_i - (2h)y''_i + \tfrac{1}{2}(2h)^2y'''_i - \frac{1}{6}(2h)^3y^{iv}_i + \frac{1}{24}(2h)^4y^v_i + \ldots \quad \ldots \quad (44)$$

$$y'_{i-3} = y'_i - (3h)y''_i - \frac{(3h)^2}{2}y'''_i - \frac{(3h)^3}{6}y^{iv}_i$$

$$+ \frac{(3h)^4}{24}y^{v}_i + \ldots \qquad \ldots (45)$$

We line up the coefficient as follows:

hy'_i	$h^2y''_i$	$h^3y'''_i$	$h^4y^{iv}_i$	$h^5y^{v}_i$
1	$\frac{1}{2}$	$\frac{1}{6}$	$\frac{1}{24}$	$\frac{1}{120}$
$-\frac{55}{24}$	0	0	0	0
$\frac{59}{24}$	$-\frac{59}{24}$	$\frac{59}{48}$	$-\frac{59}{144}$	$\frac{59}{576}$
$-\frac{37}{24}$	$\frac{74}{24}$	$-\frac{74}{24}$	$\frac{296}{144}$	$-\frac{592}{596}$
$\frac{9}{24}$	$-\frac{27}{24}$	$\frac{81}{48}$	$-\frac{243}{144}$	$\frac{729}{576}$
0	0	0	0	$\frac{251}{720}$

We conclude that the Adams-Moulton's predictor equation agrees with a Taylor series up to h^4, the error being

$$\frac{251}{720}h^5y^{v}(\theta_1), \quad \theta_1 \text{ in } (x_i, x_{i+1}) \ldots (46)$$

The error in the corrector equation is derived similarly and is given by

$$-\frac{19}{720}h^5y^{v}(\theta_2), \quad \theta_2 \text{ in } (x_i, x_{i+1}) \ldots (47)$$

The Adams-Moulton "Mop-Up"

Let y be the exact numerical solution

From (39) and (46) we have

$$E_p = y - Pre = \frac{251}{720}h^5 y^v(\theta_1) \quad \ldots \quad (48)$$

Similarly from (40) and (47) we have

$$E_c = y - Cor = -\frac{19}{720}h^5 y^v(\theta_2) \quad \ldots \quad (49)$$

Assuming that $y^v(\theta_1)$ and $y^v(\theta_2)$ are equal

$$\frac{EP}{E_c} = -\frac{251}{19}$$

$$E_p = -\frac{251}{19}E_c$$

$$E_p - E_c = -\frac{251}{19}E_c - E_c = -\frac{270}{19}E_c$$

$$E_c = -\frac{19}{270}(E_p - E_c)$$

But $E_c = y - Cor$ and

$$E_p - E_c = (y - Pre) - (y - Cor) = Cor - Pre$$

Thus $y = Cor + E_c = Cor + \frac{19}{270}(Pre - Cor)$

is called the Adams-Moulton "Mop-Up".

2.3.3 Numerov's Method

The problems are of the form

$y'' = F(x,y)$

The basic equation is of the form

$$y_{k+1} = Ay_k + by_{k-1} + h^2(CF_{k+1} + DF_k + EF_{k-1}) \quad \ldots (50)$$

where $F = y''$

Using Taylor series to express

y_{k+1}, F_{k+1}, y_{k-1} and F_{k-1} we have

$$y_{k+1} = y_k + hy'_k + \frac{h^2}{2!}y''_k + \frac{h^3}{3!}y''' + \frac{h^6}{6!}y^{vi}(\theta_1) \ldots (51)$$

$$y_{k-1} = y_k - hy'_k + \frac{h^2}{2!}y''_k - \frac{h^3}{3!} + \ldots + \frac{h^6}{6!}y^{iv}(\theta_2) \ldots (52)$$

$$F_{k+1} = y''_k + hy'''_k + \ldots + \frac{h^6}{4!}y^{vi}(\theta_3) \quad \ldots (53)$$

$$F_{k-1} = y''_k - hy'''_k + \ldots + \frac{h^6}{4!}y^{vi}(\theta_4) \quad \ldots (54)$$

Substituting (51), (52), (53) and (54) in (50) and matching coefficients of like powers of h, we find that if y^{vi} can be considered equal for $\theta_1, \theta_2, \theta_3, \theta$, then

$$y_k = Ay_k + By_k \Rightarrow A+B = 1 \quad \ldots \quad (55)$$

$$y_k' = -By_k' \Rightarrow B = -1 \quad \ldots \quad (56)$$

$$\frac{1}{2!}y_k'' = \frac{1}{2!}By_k'' + Cy_k'' + Dy_k'' + Ey_k'' \Rightarrow \tfrac{1}{2}B+C+D+E=\tfrac{1}{2} \quad \ldots (57)$$

$$\frac{1}{3!}y_k''' = -\frac{1}{3!}By_k''' + Cy_k''' - Ey_k''' \Rightarrow \frac{1}{6} = -\frac{1}{6}B+C-E \quad \ldots (58)$$

$$\frac{1}{4!}y_k^{iv} = \frac{1}{4!}By_k^{iv} + \frac{1}{2!}Cy_k^{iv} + \frac{1}{2!}Ey_k^{iv} \Rightarrow$$

$$\frac{1}{24} = \frac{1}{24}B + \tfrac{1}{2}C + \tfrac{1}{2}E \quad \ldots \quad (59)$$

Thus from (55) - (59) we have

$A = 2$, $B = -1$, $C = \frac{1}{12}$, $D = \frac{5}{6}$ and $E = \frac{1}{12}$

To obtain an estimate of the error in (50), we assume that y^{vi} is the same at the four values of θ, and then add the coefficients of $h^6 y^{vi}(\theta)$:

$$-\frac{1}{6!} + B\cdot\frac{1}{6!} + C\cdot\frac{1}{4!} + E\cdot\frac{1}{4!} = -\frac{1}{240} \quad \ldots (60)$$

Thus we have the Numerov's formula for

$y'' = F(x,y)$ as $y_{k+1} = 2y_k - y_{k-1} +$

$$\frac{h^2}{12}(F_{k+1} + 10F_k + F_{k-1}) + R$$

where $R = -\frac{h^2}{240}y^{vi}(\theta)$, $\theta \in (x_{k-1}, x_{k+1})$ $\ldots$ (61)

CHAPTER 3

SOLUTION OF SECOND ORDER ORDINARY DIFFERENTIAL EQUATION AS A SYSTEM OF TWO FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS

3.0 Introduction

We shall consider two classes of second order ordinary differential equations: those of the form

$y'' = f(x,y,y')$ ... (1)

subject to initial conditions

$y(x_o) = y_o,\ y'(x_o) = y_o'$

and those of the form

$y'' = f(x,y)$ ... (2)

subject to initial conditions

$y(x_o) = y_o,\ y'(x_o) = y_o'$

The system of equations corresponding to (1) is

$y' = P$ ..... (3a)

$P' = f(x,y,P)$ ... (3b)

with initial conditions

$y(x_o) = y_o, y'(x_o) = y_o'$ while that for (2) is

$y' = q$ ... (4a)

$q' = f(x,y)$... (4b)

with initial conditions

$y(x_o) = y_o,\ y'(x_o) = y_o'$

The systems (3) and (4) are solved using the following classical numerical methods

(a) Euler

(b) Runge-Kutta

(c) Adams-Moulton

(d) Milne and

(e) Taylor.

Equation (2) is also solved by Numerov's method. For each system, the results of each method are compared with each other and with other methods, of solving (1) or (2) where possible. For example the results obtained from using system (4) in solving equation (2) will be compared with the classical Numerov's method for (2).

The following second order equations will be used as test problems.

(I) $y'' = 3y'/x - 3y/x^2 + 2x - 1$ ... (5)

$y(1) = 2,\ y'(1) = 10$

(II) $y'' = y + \exp(x)$ ... (6)

$y(o) = 0,\ y'(o) = 0.5$

The regions of comparison of results for (5) and (6) are $DI = \{x/1.1 \leq x \leq 2.0\}$ ... (7a)

and $D2 = \{x/0.1 \leq x \leq 1.0\}$ ... (7b)

However, the set of error values obtained for (5) are compared at R1=$\{11,13,15,17,19 \text{ and } 20\}$ ... (7c)

and those for (6) atR2 =$\{0.1,.3,.5,.7,.9 \text{ and } 1.0\}$ ... (7d)

The step-sizes H = $\{0.1,0.05,0.025 \text{ and } 0.0125\}$ ... (7e)

are the same for (5) and (6). In the detailed outputs under Appendices A and B results for (5) and (6) are recorded at points

p1 = $\{1.1, 1.2, 1.3, 1.4, \ldots 2.0\}$ ... (7f)

and

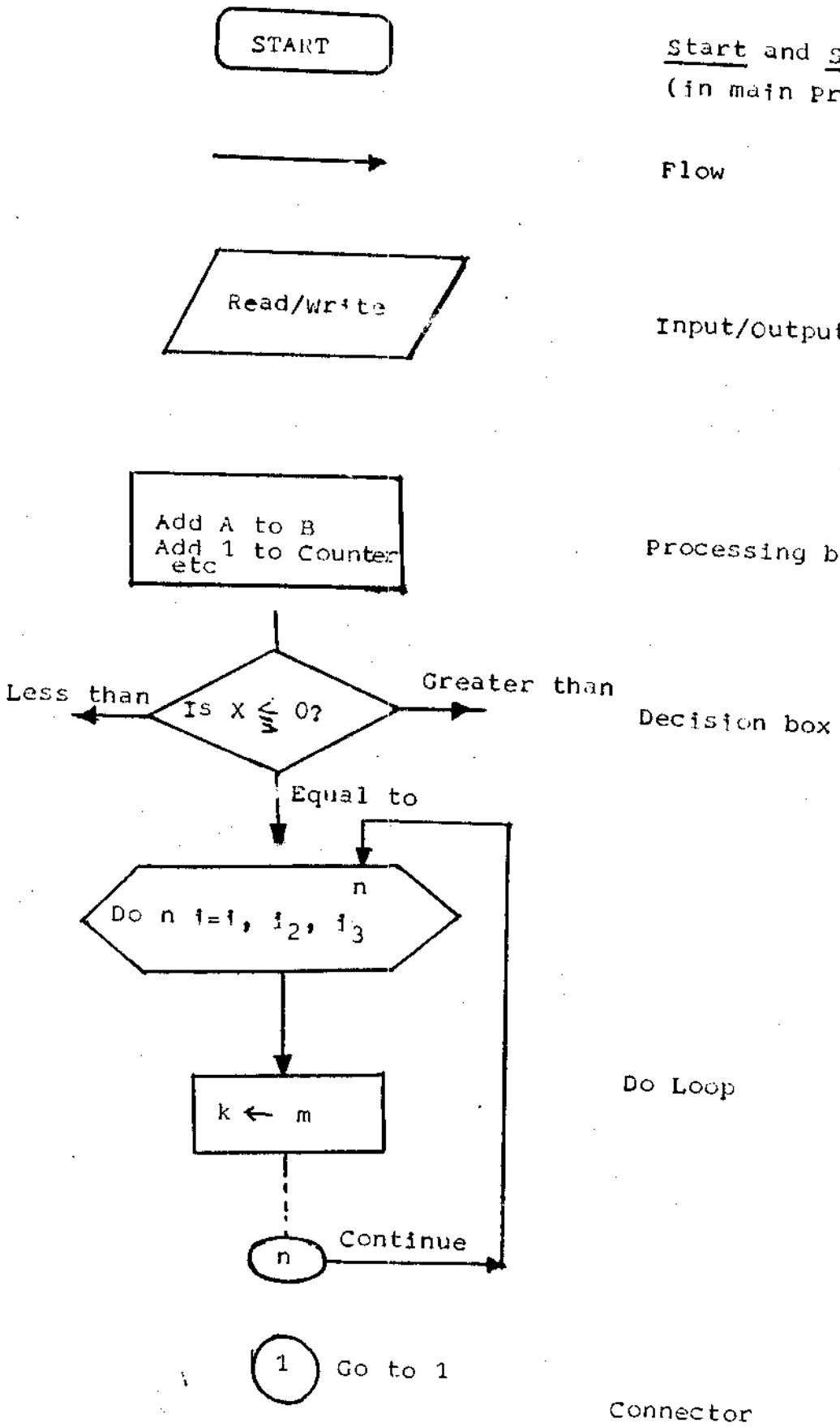
p2 = $\{0.1, 0.2, 0.3, 0.4, \ldots 1.0\}$ ... (7g)

respectively in D1 and D2. Also in the outputs the headings X-VAL, EX.SOL, C.SOL, C.Y' and ERROR - represent X-value, Exact Solution, Calculated Solution, Calculated first derivative of y with respect to x and (Ex.Sol - C.sol) respectively.

Flow Charting

A flow chart is defined by Chopra and Kumar (7) as "a depiction of an algorithm which describes the steps of the algorithm and the order in which they are to be carried out". The symbols adopted in the flow charts which are due to Chopra, Kumar and Organic(27) are as follow:

SYMBOLS


START
Start and Stop
(in main Program)
Flow
Read/Write
Input/Output
Add A to B
Add 1 to Counter
etc
Processing box
Less than
Is X ≤ 0?
Greater than
Decision box
Equal to
n
Do n i=i1, i2, i3
k ← m
Do Loop
n
Continue
1
Go to 1
Connector

The same flow charts I - VII are used for the system 3 and 4. Even though the flow-charts were basically prepared for system 3, they are equally suitable for system 4 with the following modifications.

1. Replacing f(x,y,p) with f(x,y) in the flow charts wherever it appears in the equations.
2. Replacing the test problem (5) with the test problem (6) in the flow charts.

Programs

The flow charts in the project were coded into appropriate programs named AWO1A, AWO1B, AWO2A, AWO2B and so on to produce in the regions D1 and D2 approximate solutions as it is described by Gladwell and Sayers (22) at the points P1 and P2 respectively.

To ensure the outputs at P1 and P2 respectively when $H \neq 0.1$, the following programming statements were added into each program immediately after the programming statement -- "ERR = DABS(ERROR)" before the WRITE statement. They were also shown in some of the Flow Charts:

N1 = K

M1 = I/N1

J1 = n1 * M1

```
      IF (I.NE.J1) GO TO 106
106   XO    =    X1
```

where

K = 2 for H = 0.05 ... (i)

K = 4 for H = 0.025 ... (ii)

K = 8 for H = 0.0125 ... (iii)

I is an integer variable name and is usually referred to as the index of DO loop or a DO variable.

(i) shows there is a print out at P1 and P2 in the second iteration since 0.05 x 2 = 0.1

(ii) shows there is a print out at P1 and P2 respectively in the fourth iteration since 0.025 x 4 = 0.1 and

(iii) shows there is a print out at P1 and P2 respectively in the eighth iteration since 0.0125 x 8 = 0.1

For H = 0.1 the modification is not necessary since the step-size is also H = 0.1

The statement "106 XO = X1" is an example of the leading statement for the change of locations for the next iteration.

Outputs

Four sets of outputs for H = 0.1 to H=0.0125 are produced by each program except in AWO5(R2A) and AWO5(R2B) where the values of H are reduced beyond 0.0125 for further deductions.

Tables

Each table gives the outputs of each program at R1 and R2 for the systems 3 and 4 respectively.

Appendices

The Appendices are grouped into Appendix A and Appendix B to cater for the program listings and the detailed outputs which are based on the appropriate flow charts for the various methods considered under systems 3 and 4.

Each program is given a name as indicated earlier and the outputs are properly labelled so that references are easily made to any of the outputs when necessary.

Under Appendix A the program listings of AWO9(EA) and AWO9(RA) are ommitted since they are almost identical with the corresponding program listings of AWO8(EA) and AWO8(RA). The only difference is that while AWO9(EA) and AWO9(RA) include the given initial values as part of the starting values, AWO8(EA) and AWO8(RA) do not include them.

Under Appendix B both the program listings and outputs of AW07A and AW07B are given. On the other hand only the outputs of other programs are given. This is because their program listings are basically the same as the corresponding program listings under Appendix A except for minor changes which have been clarified under Flow Charting.

3.1 Improved Euler Methods

3.1.1 System 3

The improved Euler method for solving system 3 is given by Dorn and McCracken (15) as:

$$y_{k+1} = y_k + \frac{h}{2}(P_k + P_{k+1}) \quad \cdots \quad (8)$$

where

$$P_{k+1} = P_k + \frac{h}{2}(F(x_k, y_k, P_k) + F(x_{k+1}, y_{k+1}, P_{k+1}))$$

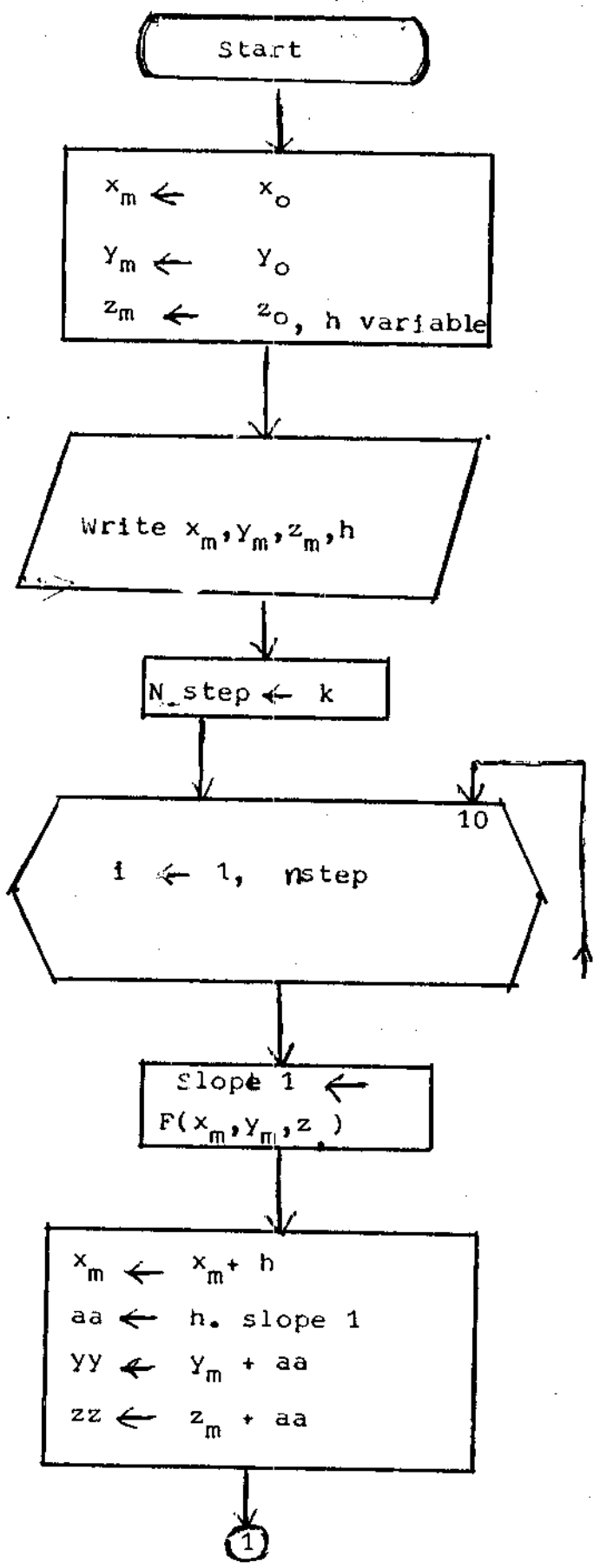
$$x_{k+1} = k_k + h$$

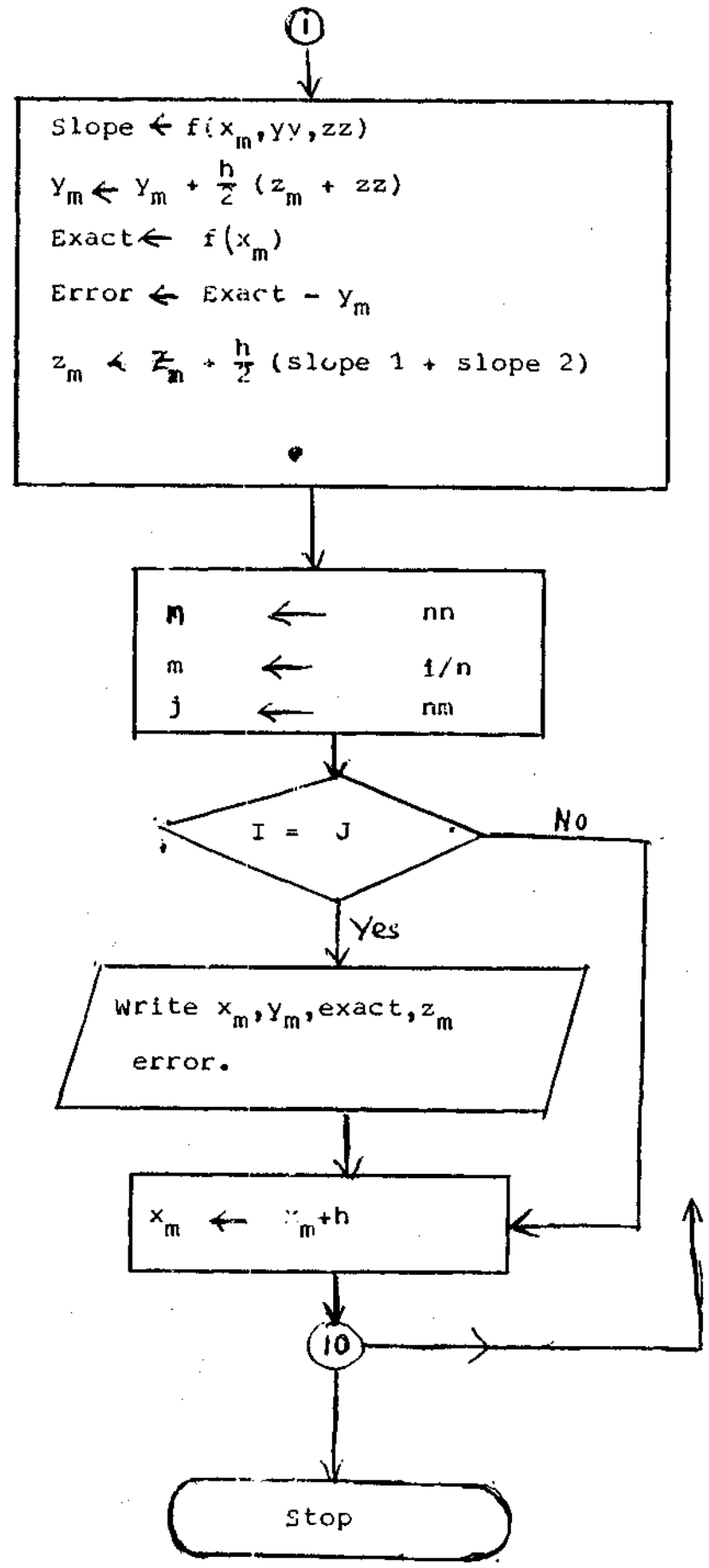
$$y_{k+1} = y_k + h(F(x_k, y_k, P_k))$$

$$P_{k+1} = P_k + h(F(x_k, y_k, p_k))$$

Flow Chart I

Flow chart I is a pictorial representation of the improved Euler algorithm which solves systems 3 and 4 numerically in the regions D1 and D2 respectively. Furthermore, it records exact solutions, calculated solutions and error-values at each point in R1 and R2 respectively.


Start
$x_m \leftarrow x_o$
$y_m \leftarrow y_o$
$z_m \leftarrow z_o$, h variable
Write x_m, y_m, z_m, h
N_step ← k
10
i ← 1, nstep
Slope 1 ← F(x_m, y_m, z_m)
$x_m \leftarrow x_m + h$
aa ← h. slope 1
yy ← y_m + aa
zz ← z_m + aa
1


Slope ← f(xm,yy,zz)
ym ← ym + h/2 (zm + zz)
Exact ← f(xm)
Error ← Exact - ym
zm ← zm + h/2 (slope 1 + slope 2)
n ← nn
m ← 1/n
j ← nm
I = J
No
Yes
Write xm,ym,exact,zm
error.
xm ← xm+h
10
Stop

Program

The program named AW01A uses Euler Method (8) based on the flow chart I, to solve the test problem (5) in the region D1 using the step - size H given by (7e). The table below shows the exact solutions and the error values at selected x - values in (7c) and for all the step sizes. Detailed outputs are found in AW01A under Appendix A.

Table 3.1

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	0.00486	0.00397	0.00243	0.00134
1.3	6.25741	0.08108	0.04772	0.02593	0.01352
1.5	10.74344	0.25776	0.14573	0.07771	0.04015
1.7	16.83598	0.54710	0.30462	0.16121	0.08299
1.9	24.78947	0.95944	0.52994	0.27933	0.14351
2.0	29.54517	1.21456	0.66902	0.35215	0.18081

Observation

1. Compared with the exact solutions the magnitudes of the error-values under each value of H are tolerable. It is observed however, on the error column of each set of output from the first to the fourth set that the error is growing as we move down from x = 1.1 to x = 2.0. Also the error value reduces as the value of H reduces.

2. It could be deduced from 1 that if H is further decreased within the region D1, error values will further reduce correspondingly.

3. Finally, if we take any value of x say x = 1.1 and we look at the error-values from H = 0.1 to H = 0.0125 it is observed that the error value under H = 0.05 is about ½ of the corresponding error-value under H = 0.1. This feature continues to H = 0.0125.

3.1.2 System 4

The improved Euler method for solving system 4 is a modification of (8) and is given by

$$y_{k+1} = y_k + \frac{h}{2}(p_k + p_{k+1}) \quad \ldots \quad (9)$$

where

$$p_{k+1} = p_k + \frac{h}{2}(f(x_k,y_k)\ f(x_{k+1},y_{k+1}))$$

$$x_{k+1} = x_k + h$$

$$y_{k+1} = y_k + hf(x_k, y_k)$$

Program:

The program named AWO1B uses the improved Euler method given by equation (9) based on the modified flow-chart I to solve the test problem (6).

Outputs:

The outputs at the selected points of x in (7d) are as shown in Table 3.2. The detailed outputs are obtained in AWO1B under Appendix B.

Table 3.2

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.00026	.00004	.00001	.00001
0.3	.20248	.00015	.00027	.00022	.00013
0.5	.41218	.00094	.00116	.00076	.00043
0.7	.70481	.00324	.00280	.00171	.00094
0.9	1.10682	.00707	.00541	.00320	.00173
1.0	1.35914	.00969	.00716	.00419	.00225

Observations

1. Error-values are tolerable for each value of H when compared with the corresponding exact solutions in column 2. Hence the method is stable. However, there is general growth of error values for H = 0.05 to H= 0.0125 in the region D2. For H = 0.1 error-value decreases from x = 0.1 to x= 0.3. Beyond this value of x it grows up to x = 1.0 as could be seen in Table 3.2.
2. This method shows a great degree of accuracy which is obvious from the magnitudes of error values. Accuracy increases also as we move from H=0.1 to H=0.0125. It is also likely that a high degree of accuracy will be obtained in our region of observation if other values of H less than 0.0125 are used.

3.2 The Classical Runge-Kutta Method

3.2.1 System 3

The classical Runge-Kutta method for solving system 3 is given by Dodes(14) as

$$Y_{n+1} = Y_n + l_1 \quad \ldots \quad (10a)$$

$$P_{n+1} = P_n + k_1 \quad \ldots \quad (10b)$$

where

$$l_1 = \frac{1}{6}(k_{11} + 2l_{12} + 2l_{13} + l_{14}) \ldots (10c)$$

$$k_1 = \frac{1}{6}(k_{11} + 2k_{12} + 2k_{13} + k_{14}) \ldots (10d)$$

$$l_{11} = hf_1(x_n, y_n, p_n)$$

$$k_{11} = hf_2(x_n, y_n, p_n)$$

$$l_{12} = hf_1(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_{11}, p_n + \tfrac{1}{2}l_{11})$$

$$k_{12} = hf_2(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_{11}, p_n + \tfrac{1}{2}l_{11})$$

$$l_{13} = hf_1(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_{12}, p_n + \tfrac{1}{2}l_{12})$$

$$k_{13} = hf_2(x_n + \tfrac{1}{2}h, y_n + \tfrac{1}{2}k_{12}, p_n + \tfrac{1}{2}l_{12})$$

$$l_{14} = hf_1(x_n + h, y_n + k_{13}, p_n + l_{13})$$

$$k_{14} = hf_2(x_n + h, y_n + k_{13}, p_n + l_{13})$$

If x and y are not involved explicitly in l_{11}, l_{12}, l_{13} and l_{14} the equations can be simplified as follows:

Step 1. $x = x_n$, $y = y_n$, $p = p_n$

$l_{11} = hp$

$k_{11} = hf_2(x,y,p)$

Step 2. $x = x_n + \frac{1}{2}h$, $y = y_n + \frac{1}{2}k_{11}$, $p = p_n + \frac{1}{2}l_{11}$

$l_{12} = hp$

$k_{12} = hf_2(x,y,p)$

Step 3. x is unchanged, $y = y_n + \frac{1}{2}k_{12}$, $p = p_n + \frac{1}{2}l_{12}$

$l_{13} = hp$

$k_{13} = hf_2(x,y,p)$

Step 4. $x = x_n + h$, $y = y_n + k_{13}$, $p = p_n + l_{13}$

$l_{14} = hp$

$k_{14} = hf_2(x,y,p)$

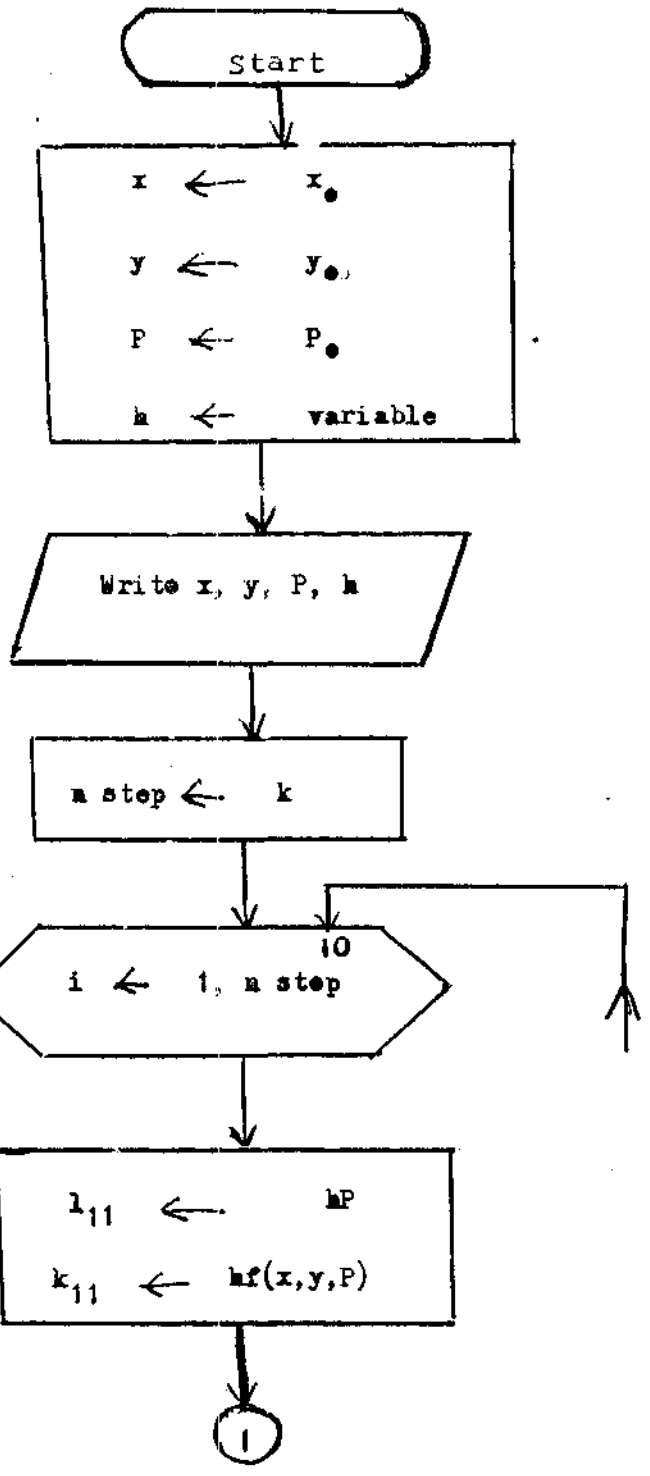
Step 5. Calculate l_1 and k_1 using (10c) and (10d).

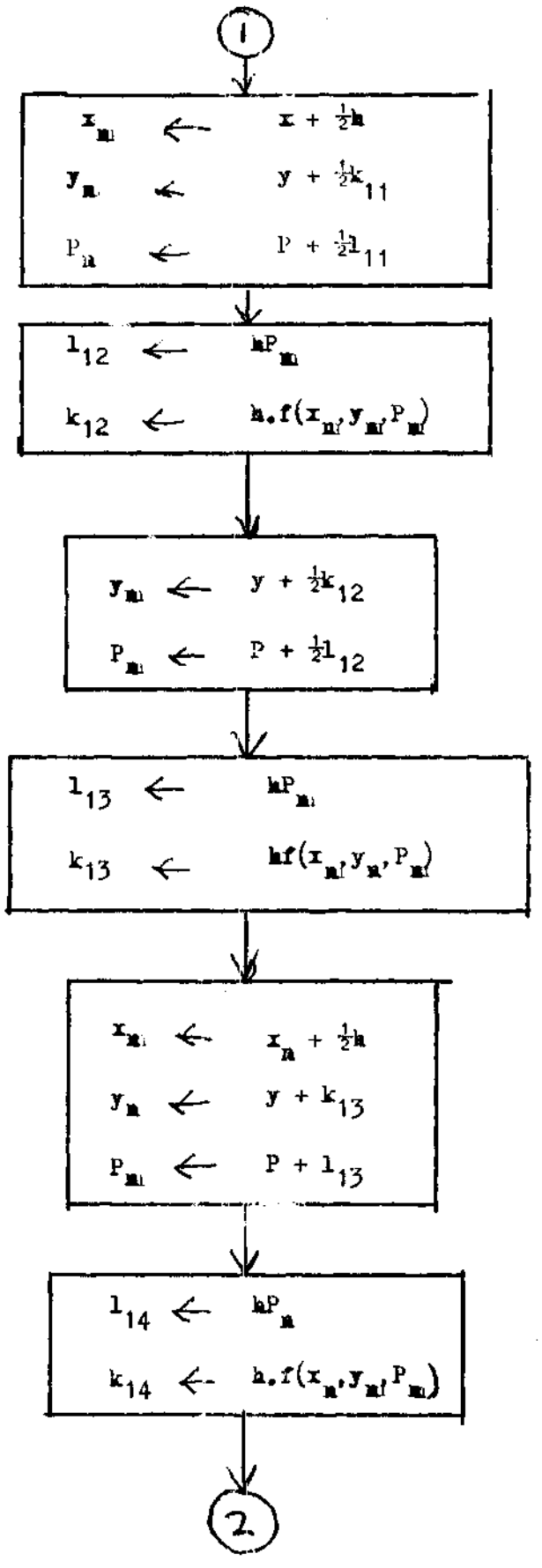
Step 6. Calculate y_{n+1} and p_{n+1} using (10a) and (10b)

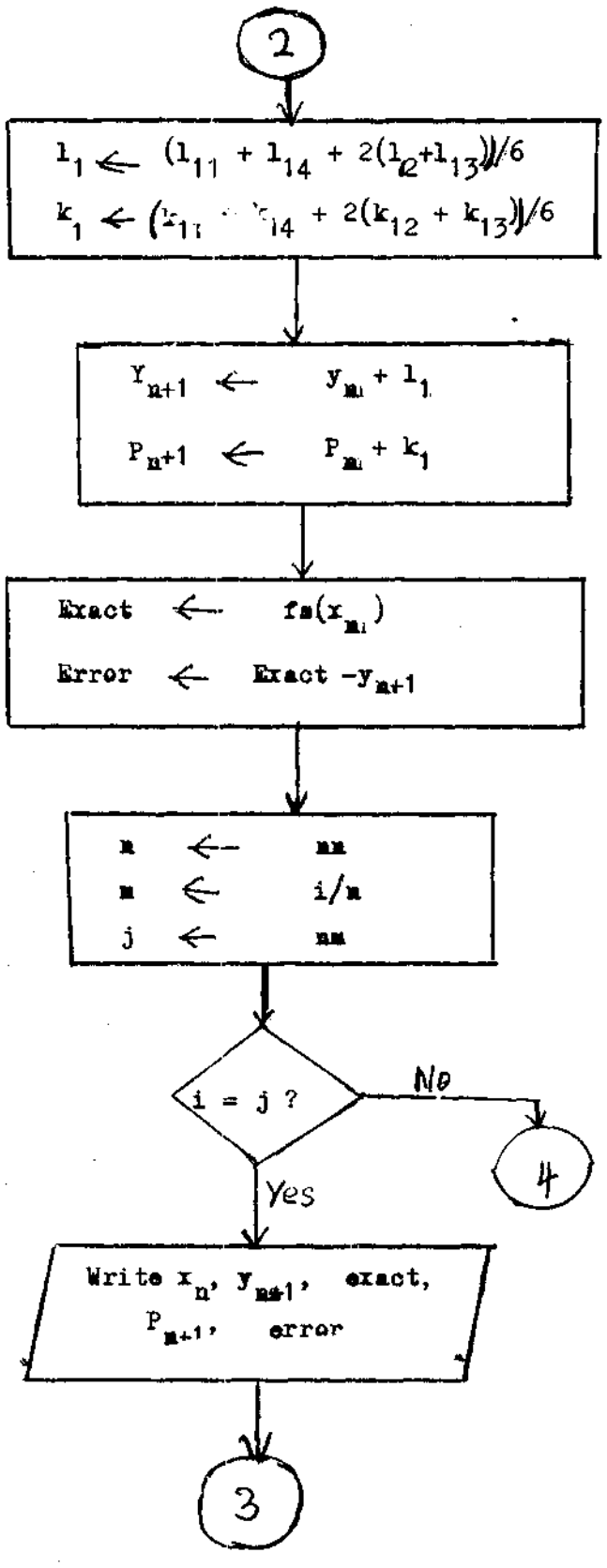
Step 7. Calculate and record exact solutions, numerical solutions and error values at selected points in the regions of interest.

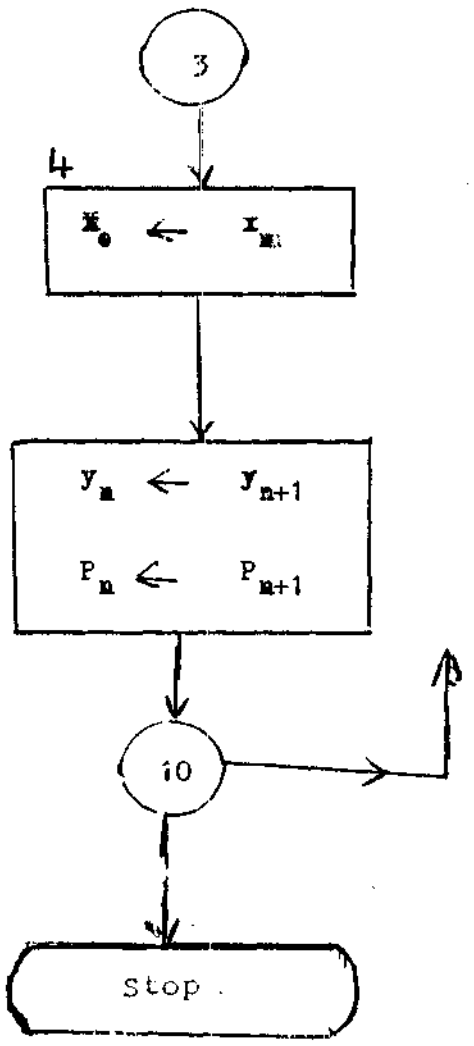
Flow Chart II

The flow chart II represents pictorially the classical **Runge-Kutta** algorithm steps 1 - 7 under section 3.2.1. It solves systems 3 and 4 numerically in the region D1 and D2 and records exact solutions, calculated solutions and error values at each point in R1 and R2 respectively.


Start
x ← x•
y ← y•
P ← P•
h ← variable
Write x, y, P, h
n step ← k
10
i ← 1, n step
l11 ← hP
k11 ← hf(x,y,P)
1


1
$x_m \leftarrow x + \frac{1}{2}h$
$y_m \leftarrow y + \frac{1}{2}k_{11}$
$P_m \leftarrow P + \frac{1}{2}l_{11}$
$l_{12} \leftarrow hP_m$
$k_{12} \leftarrow h.f(x_m, y_m, P_m)$
$y_m \leftarrow y + \frac{1}{2}k_{12}$
$P_m \leftarrow P + \frac{1}{2}l_{12}$
$l_{13} \leftarrow hP_m$
$k_{13} \leftarrow hf(x_m, y_m, P_m)$
$x_m \leftarrow x_n + \frac{1}{2}h$
$y_m \leftarrow y + k_{13}$
$P_m \leftarrow P + l_{13}$
$l_{14} \leftarrow hP_m$
$k_{14} \leftarrow h.f(x_m, y_m, P_m)$
2


2
$l_1 \leftarrow (l_{11} + l_{14} + 2(l_{12}+l_{13}))/6$
$k_1 \leftarrow (k_{11} + k_{14} + 2(k_{12} + k_{13}))/6$
$Y_{n+1} \leftarrow y_n + l_1$
$P_{n+1} \leftarrow P_n + k_1$
Exact $\leftarrow$ $fn(x_n)$
Error $\leftarrow$ Exact $-y_{n+1}$
n $\leftarrow$ nn
n $\leftarrow$ i/n
j $\leftarrow$ nn
i = j ?
No
4
Yes
Write x_n, y_{n+1}, exact,
P_{n+1}, error
3


3
4
$x_0 \leftarrow x_m$
$y_m \leftarrow y_{m+1}$
$P_m \leftarrow P_{m+1}$
10
Stop

Program

The program named AWO2A uses the classical Runge-Kutta method (3) which is based on the flow-chart II to solve the test problem (5).

Outputs

The Table 3.3 below shows the exact solutions and the error values at the selected x-values in (7c) while detailed outputs are obtained in AWO2A in Appendix A.

Table 3.3

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	0.07815	0.04379	0.02327	0.01201
1.3	6.25741	0.36205	0.20175	0.10690	0.05508
1.5	10.74344	0.83396	0.46300	0.24486	0.12604
1.7	16.83598	1.50915	0.83549	0.44119	0.22694
1.9	24.78947	2.39651	1.32377	0.69820	0.35891
2.0	29.54517	2.92095	1.61191	0.84975	0.43670

Observation

The above Table reveals the same feature as obtained in Table 3.1 except that accuracy is generally better in Table 3.1. This could be observed by the lower values of errors generally when error outputs of AWO1A and AWO2A in Appendix A are compared.

3.2.2 System 4

The classical Runge-Kutta method for solving system 4 is a modification of (10) and is given by:

$$Y_{n+1} = Y_n + L_1 \quad \ldots \quad (11a)$$

$$P_{n+1} = P_n + k_1 \quad \ldots \quad (11b)$$

where

$$L_1 = \frac{1}{6}(L_{11} + 2L_{12} + 2L_{13} + L_{14}) \ldots (11c)$$

$$k_1 = \frac{1}{6}(k_{11} + 2k_{12} + 2k_{13} + k_{14}) \ldots (11d)$$

Step 1: $x = x_n, y = y_n$ and $P = P_n$

$$L_{11} = hP$$

$$k_{11} = hf(x,y)$$

Step 2: $x = x_n + \frac{1}{2}h$, $y = y_n + \frac{1}{2}k_{11}$, $P = P_n + \frac{1}{2}L_{11}$

$$L_{12} = hp$$

$$k_{12} = hf(x,y)$$

Step 3: x is unchanged, $y = y_n + \frac{1}{2}k_{12}$, $P = P_n + \frac{1}{2}L_{12}$

$$L_{13} = hp$$

$$k_{13} = hf(x,y)$$

Step 4: $x = x_n + h$, $y = y_n + k_{13}$, $P = P_n + L_{13}$

$$L_{14} = hP$$

$$k_{14} = hf(x,y)$$

Step 5: Calculate L_1 and k_1 using (11c) and and (11d)

Step 6: Calculate Y_{n+1} and P_{n+1} using (11a) and (11b)

Step 7: Calculate and record exact solutions, numerical solutions and error-values at the required points in the region D2.

Program:

The Program called AWO2B uses the classical Runge-Kutta method (11) based on the modified flow-chart II to solve our test problem (6).

Outputs:

The selected outputs are as shown in table 3.4 below.

The detailed outputs are also obtained in AWO2B in Appendix B.

Table 3.4

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES.			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.00267	.00129	.00064	.00032
0.3	.20248	.00805	.00391	.00193	.00096
0.5	.41218	.01366	.00667	.00329	.00164
0.7	.70481	.01975	.00967	.00479	.00238
0.9	1.10682	.02658	.01305	.00647	.00322
1.0	1.35914	.03036	.01493	.00740	.00369

As in Table 3.2 error values in Table 3.4 are seen to be clearly growing, though acceptable when compared with the exact solutions in column 2 of Table 3.4, from H=0.1 to H= 0.0125, while they are decreasing rowise at about ½ the value of the previous error value.

For example looking at row x = 0.1 we observe that the error value under H = 0.05 is about $\frac{1}{5}$ the value of the error under H= 0.1. Similarly, the error under H = 0.025 is about $\frac{1}{2}$ the value of the error under H = 0.05 while the error under H = 0.0125 is exactly $\frac{1}{2}$ the value of the error under H = 0.025. This feature is true of other rows as well. It is also seen that better accuracy is achieved with decreasing values of H. Hence if H is further decreased beyond H = 0.0125 much better accuracy will be recorded in our region of observation, D2. Also if Table 3.4 is compared with Table 3.2, we discover that Runge-Kutta method is less accurate than improved Euler method. Each of the error value in Table 3.4 is greater than the corresponding error value in Table 3.2.

3.3 Adams-Moulton's Methods

3.3.1 System 3

Adams-Moulton method for solving system 3 is given by Dodes (14) as:

$$\text{Pre } Y_{k+1} = Y_k + \frac{h}{24}(55P(x_k,y_k,P_k) - 59P(x_{k-1},y_{k-1},P_{k-1}) + 37P(x_{k-2},y_{k-2},P_{k-2}) - 9P(x_{k-3},y_{k-3},P_{k-3})) \quad \ldots (12a)$$

$$\text{Pre } P_{k+1} = P_k + \frac{h}{24}(55F(x_k,P_k,y_k) - 59F(x_{k-1},y_{k-1},P_{k-1}) + 37F(x_{k-2},y_{k-2},P_{k-2}) - 9F(x_{k-3},y_{k-3},P_{k-3})) \quad \ldots (12b)$$

Followed by:

$$\text{Cor } Y_{k+1} = Y_k + \frac{h}{24}(9P(x_{k+1},y_{k+1},P_{k+1}) + 19P(x_k,y_k,P_k) - 5P(x_{k-1},y_{k-1},P_{k-1}) + P(x_{k-2},y_{k-2},P_{k-2})) \quad \ldots (12c)$$

$$\text{Cor } P_{k+1} = P_k + \frac{h}{24}(9F(x_{k+1},y_{k+1},P_{k+1}) 19F(x_k,y_k,P_k) - 5F(x_{k-1},y_{k-1},P_{k-1}) + F(x_{k-2},y_{k-2},P_{k-2})) \quad \ldots (12d)$$

and completed by the mop-up given by:

$$y \Rightarrow \text{Cor } y + \frac{19}{270}(\text{Pre } y - \text{Cor } y) \quad \ldots (12e)$$

$$P \Rightarrow \text{Cor } P + \frac{19}{270}(\text{Pre } P - \text{Cor } P) \quad \ldots (12f)$$

Program Planning

Step 1. Read in starting values obtained from the classical Runge-Kutta method in AWO2A.

Step 2. Calculate Predictor values Y_{k+1} and P_{k+1} using equations (12a) and (12b).

Step 3. Set k = 4. Calculate the corrector values of Y_5 using (12c).
i.e. $CorY_5 = Y_4 + \frac{h}{24}(9P_5 + 19P_4 - 5P_3 + P_2)$
where P_5 = Cor P_5 from equation (12b)

Step 4. Calculate $x_5 = x_4 + h$ and set Y_5 = Cor Y_5, $Y'' = F_5 = F(x_5, Y_5, P_5)$

Step 5. Calculate the corrector value of P_5 using (12d).

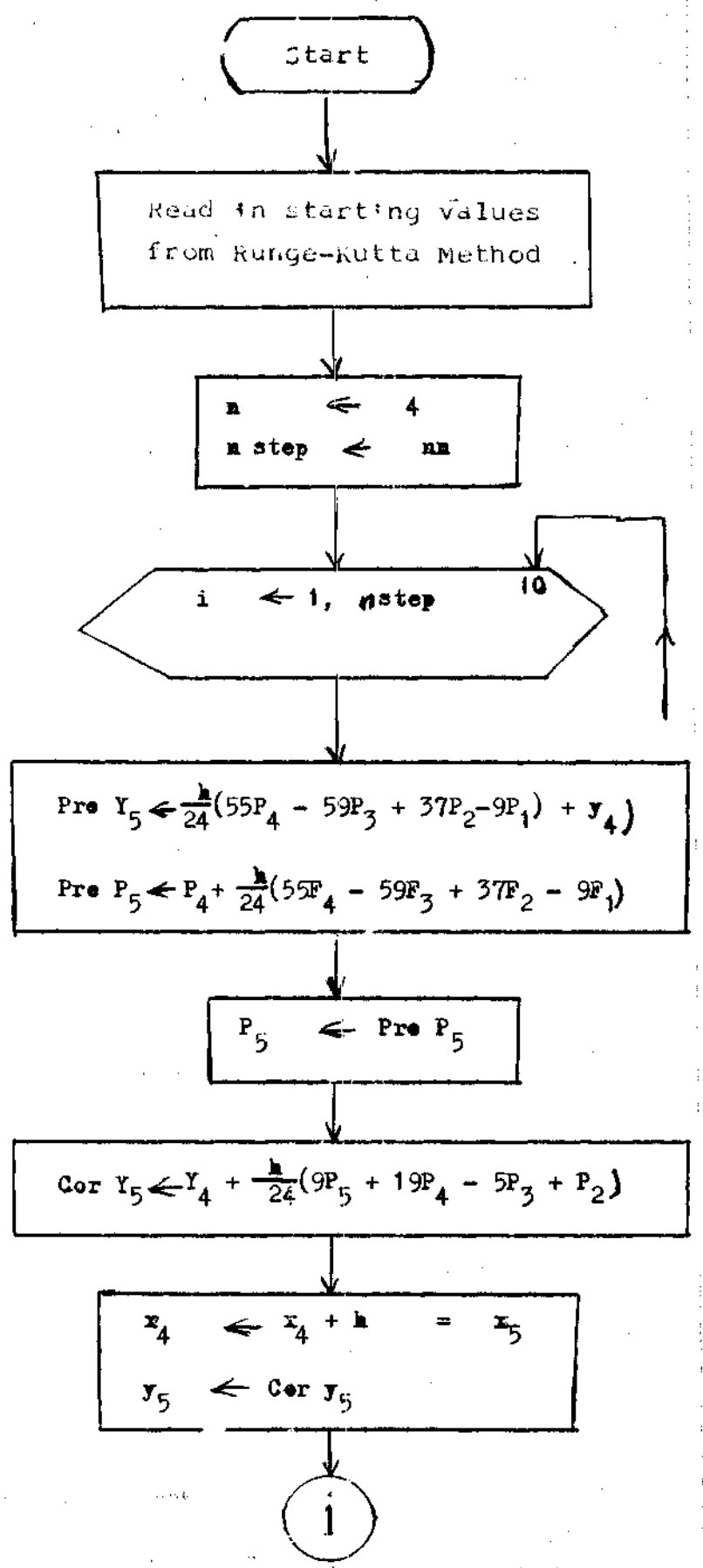
Step 6. Calculate the "mop-ups" using equations 12e and 12f.

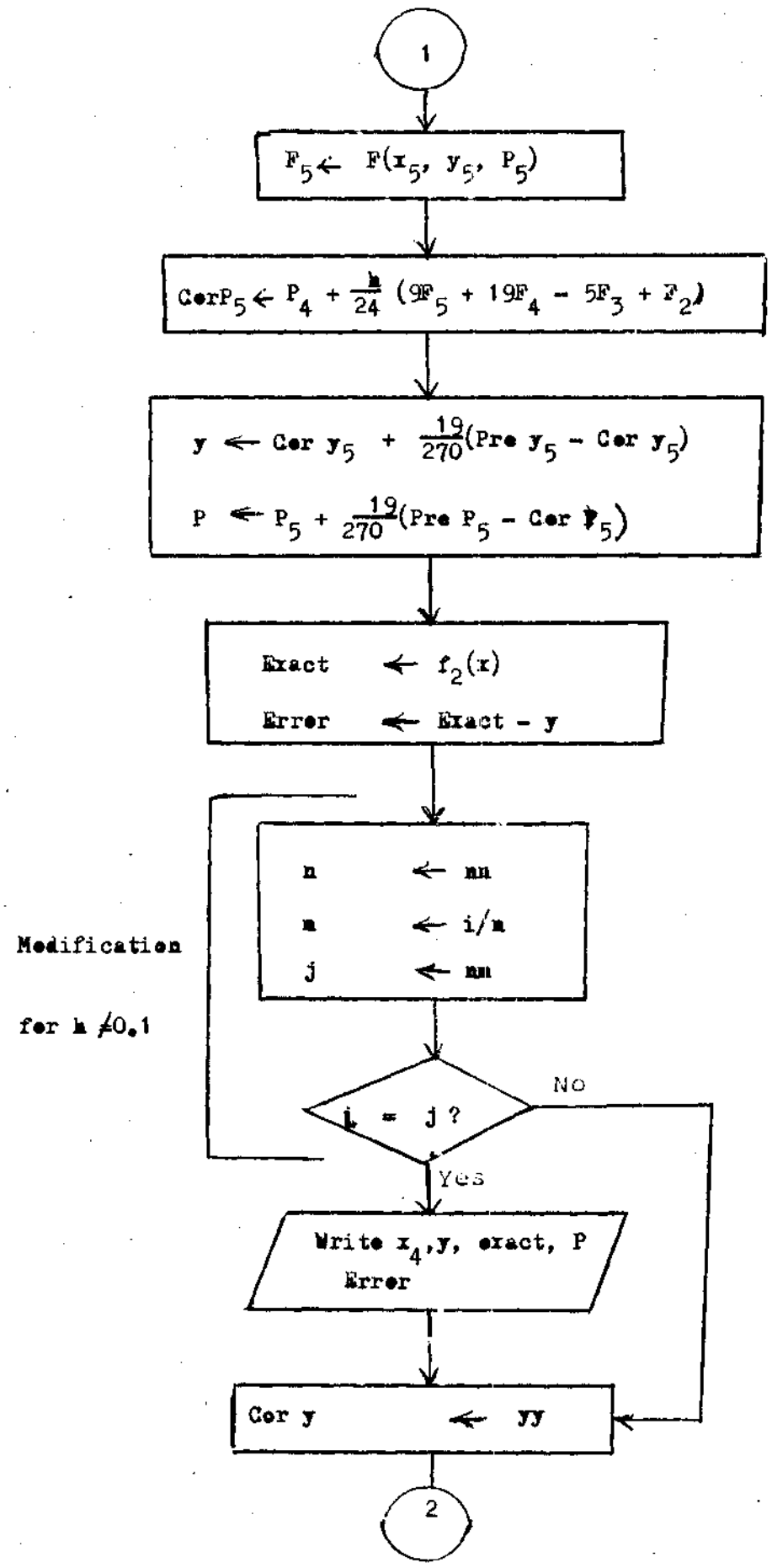
Step 7. Calculate exact solutions and error values i.e. Exact = $f_2(x_n)$ and Error = exact - y. Then print out results

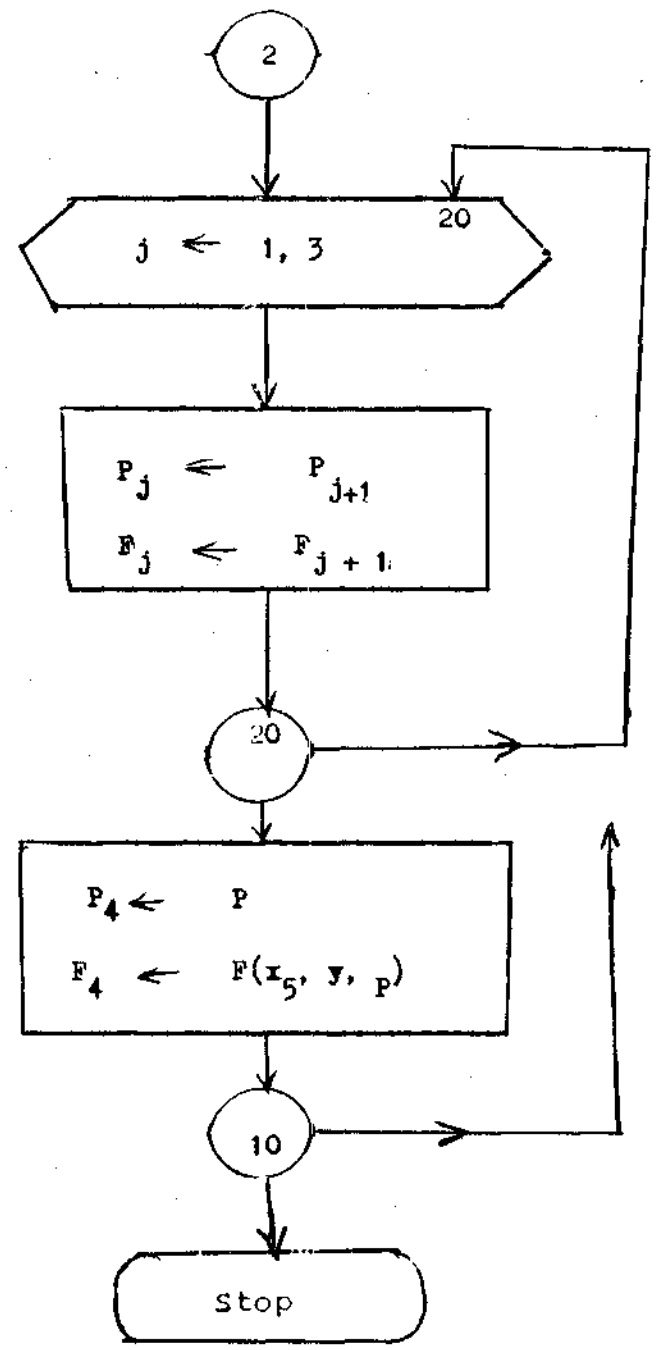
Step 8. Inter-change locations and continue iterations.

Flow Chart III

The flow chart III is a pictorial representation of the program planning steps 1-8 above. It solves numerically systems 3 and 4 in D1 and D2 and records the required solutions at each point in R1 and R2 respectively.


Start
Read in starting values from Runge-Kutta Method
n ⇐ 4
n step ⇐ nm
i ⇐ 1, nstep
10
Pre $Y_5 \Leftarrow \frac{h}{24}(55P_4 - 59P_3 + 37P_2 - 9P_1) + y_4$
Pre $P_5 \Leftarrow P_4 + \frac{h}{24}(55F_4 - 59F_3 + 37F_2 - 9F_1)$
P_5 ⇐ Pre P_5
Cor $Y_5 \Leftarrow Y_4 + \frac{h}{24}(9P_5 + 19P_4 - 5P_3 + P_2)$
$x_4 \Leftarrow x_4 + h = x_5$
y_5 ⇐ Cor y_5
1


1
$F_5 \leftarrow F(x_5, y_5, P_5)$
$CorP_5 \leftarrow P_4 + \frac{h}{24}(9F_5 + 19F_4 - 5F_3 + F_2)$
$y \leftarrow Cor\ y_5 + \frac{19}{270}(Pre\ y_5 - Cor\ y_5)$
$P \leftarrow P_5 + \frac{19}{270}(Pre\ P_5 - Cor\ P_5)$
Exact $\leftarrow f_2(x)$
Error $\leftarrow$ Exact - y
n $\leftarrow$ nn
m $\leftarrow$ i/n
j $\leftarrow$ nn
Modification
for h $\neq$0.1
i = j ?
No
Yes
Write x_4,y, exact, P
Error
Cor y $\leftarrow$ yy
2


2
20
j ← 1, 3
P_j ← P_j+1
F_j ← F_j + 1
20
P_4 ← P
F_4 ← F(x_5, y, p)
10
Stop

Program:

The Program called AWC3A uses Adams-Moulton's method (12) based on the flow-chart III to solve the test problem (5).

Since this method cannot start by itself, part of the starting values is obtained from the classical Runge-Kutta method (AWC2A) in Appendix A. The other part is the set of given initial values for the test problem (5). These starting values are printed along with the outputs in Appendix A.

Outputs

As usual outputs are recorded at selected points in (7c). Detailed outputs are obtained in AWO3A under Appendix A.

Table 3.5

X-Val	EX. SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3,12986	-	-	.01860	.00522
1.3	6.25741	-	.13196	.04032	.01119
1.5	10.74344	.66698	.23619	.07129	.01966
1.7	16.83598	1.07778	.37638	.11286	.03103
1.9	24.78947	1.60792	.55733	.16646	.04568
2.0	29.54517	1.92255	.66458	.19822	.05435

Observation

1. It is observed in column 3 under H=0.1 that we could not record error-values for x=1.1 and x=1.3 and also in column 4 for x=1.1. This is because the first three sets of outputs in R1 for x=1.1 to x=1.3 are parts of the starting values.

2. It is also observed that a growing error-value exists, though tolerable when the error values are compared with the corresponding exact solutions in column 2, for each value of H from x=1.5 to 2.0 in column 3,from x=1.3 to x=2.0 in column 4 and from x=1.1 to x=2.0 in columns 5 and 6 respectively. We can see also that in columns 6 under H=0.0125 we have the least values of errors increasing of course from .00522 at x=1.1 to .05435 at x=2.0. Hence the highest degree of accuracy is obtained when H=0.0125 in our region D1.

3. Finally, we can deduce that choosing a different region below x=1.1, say $0.9 \geq x \geq 0.0$ with $x_o = 1.0$ and with $H \leq 0.0125$ or less error values will reduce further, which implies greater accuracy for this method in the new region.

3.3.2 System 4

The Adams-Moulton equations for solving system 4, where P and F represent y' and y'', are modification of (12) and are given as follow:

$$\text{Pre } y_{k+1} = \frac{h}{24}(55P(x_k,y_k)-59(x_{k-1},y_{k-1}) + 37(x_{k-2},y_{k-2},)-9P(x_{k-3},y_{k-3}))\ldots(13a)$$

$$\text{Pre } P_{k+1} = P_k + \frac{h}{24}(55F(x_k,y_k)-59(x_{k-1},y_{k-1}) + 37F(x_{k-2},y_{k-2})-9F(x_{k-3},y_{k-3}))\ldots(13b)$$

followed by:

$$\text{Cor } y_{k+1} = y_k + \frac{h}{24}(9P(x_{k+1},y_{k+1})+19P(x_k,y_k) -5P(x_{k-1},y_{k-1}) + P(x_{k-2},y_{k-2}))\ldots(13c)$$

$$\text{Cor } P_{k+1} = P_k + \frac{h}{24}(9F(x_{k+1},y_{k+1})+19F(x_k,y_k) -5F(x_{k-1},y_{k-1})+F(x_{k-2},y_{k-2}))\ldots(13d)$$

and completed by the mop-up:

$$y = \text{Cor } y + \frac{19}{270}(\text{Prey}-\text{Cor } y) \ldots (13e)$$

$$P = \text{Cor } P + \frac{19}{270}(\text{Prep}-\text{Corp}) \ldots (13f)$$

Program:

The program called AWO3B uses Adams-Moulton's method given by equations (13) above and based on the modified Flow-chart III to solve the test problem (6) in our region D2 using the same step-sizes (7e).

Since this method cannot start by itself, part of the starting values is obtained from the classical Runge-Kutta method while the other part is the set of given initial values for the test problem (6). As usual the starting values are printed along with outputs in Appendix B.

Outputs:

The outputs for discusion are given in Table 3.6 at our selected values of x in (7d) while the detailed outputs are obtained as usual in A/03B under Appendix B.

Table 3.6

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526			.00047	.00011
0.3	.20248		.00166	.00038	.00009
0.5	.41218	.00657	.00136	.00032	.00007
0.7	.70481	.00538	.00111	.00026	.00006
0.9	1.10682	.00440	.00091	.00021	.00005
1.0	1.35914	.00398	.00082	.00019	.00004

Observation:

1. Nothing is recorded in column 3, under H=0.1 for the values of x = 0.1 and x = 0.3. Also in column 4 under H= =0.5 nothing is recorded for x = 0.1. The same reasons given under Table 3.5 are valid here.

2. From H = 0.1 to H = 0.0125 error values are decreasing. The decrease in values is becoming more and more as we move from H=0.1 to H=0.0125 with the least error value of .0004 recorded at row x = 1.0 under H= 0.0125.

3. From our observations one could rightly deduce that further reduction in H values than 0.0125 will produce better accuracy.

3.4 The fourth order Milne's Method

3.4.1 System 3

The fourth order Milne's equation for solving system 3 is given by Conte and deBoor (8) as

$$\text{Pre } y_{k+1} = y_{k-3} + \frac{4h}{3}(2P(x_k,y_k,P_k) - P(x_{k-1},y_{k-1}, P_{k-1} + 2P(x_{k-2},y_{k-2},P_{k-2})) \quad \ldots (14a)$$

$$\text{Pre } P_{k+1} = P_{k-3} + \frac{4h}{3}(2F(x_k,y_k,P_k) - F(x_{k-1}\ y_{k-1},P_{k-1}) + 2F(x_{k-2},y_{k-2},P_{k-2})) \quad \ldots (14b)$$

Followed by:

$$\text{Cor } y_{k+1} = y_{k-1} + \frac{h}{3}(P(x_{k+1},y_{k+1},P_{k+1}) + 4P(x_k,y_k,P_k) + P(x_{k-1},y_{k-1}, P_{k-1})) \quad \ldots (14c)$$

$$\text{Cor } P_{k+1} = P_{k-1} + \frac{h}{3}(F(x_{k+1},y_{k+1},P_{k+1}) + 4F(x_k,y_k,P_k) + F(x_{k-1},y_{k-1},P_{k-1}) \quad \ldots (7d)$$

and completed by the mop-up given by:

$$Y = \text{Cor } y_{k+1} + \frac{1}{29}(\text{Pre } y_{k+1} - \text{Cor } y_{k+1}) \quad \ldots (14e)$$

$$P = \text{Cor } P_{k+1} + \frac{1}{29}(\text{Pre } P_{k+1} - \text{Cor } P_{k+1}) \quad \ldots (14f)$$

Program Planning

Step 1. Read in the same set of starting values obtained from Runge-Kutta method for Adams-Moulton's method.

ie. $A_{kj}(j = 1,3;\ k = 1,4)$ in Appendix A

Thus we have:

$A_{11} - A_{41}$ representing $y_1 - y_4$

$A_{12} - A_{42}$ " $P_1 - P_4 = y'_k$

$A_{13} - A_{43}$ " $F_1 - F_4 = y''_k$

$k = 1,4)$

Also in the Milne's formulae we have

A_{52} representing $PreP_5$ = prP

A_{53} " F_{k+1} $= F(x_{k+1}, y_{k+1}, P_{k+1})$

$= y''_{k+1}$

Step 2. Calculate the predictor values of y_{k+1} and P_{k+1} using step 1 and equations (14a) and (14b).

i.e. Pre $y_{k+1} = y_{k-3} + \frac{4h}{3}(2P_k - P_{k-1} + 2P_{k-2})$

Pre $P_{k+1} = P_{k-3} + \frac{4h}{3}(2F_k - F_{k-1} + 2F_{k-2})$

Step 3. Calculate Cor y_{k+1} using equation(14c)

i.e. Cor $y_{k+1} = y_{k-1} + \frac{h}{3}(P_{k+1} + 4P_k + P_{k-1})$

Here set P_{k+1} = Pre P_{k+1}

Now to calculate Cor P_{k+1}, we need to calculate F_{k+1} first. Thus $x_{k+1}=x_k + h$. Since this increment occurs at each iteration put it as $x_k = x_k + h$. Also Cor $y_{k+1} = y_{k+1}$ and Pre $P_{k+1} = P_{k+1}$. Thus $F_{k+1}= F(x_k, \text{Cor } y_{k+1}, \text{Pre}P_{k+1})$ and Cor $P_{k+1} = P_{k-1} + \frac{h}{3}(F_{k+1} + 4F_k + F_{k-1})$ using equation (14d)

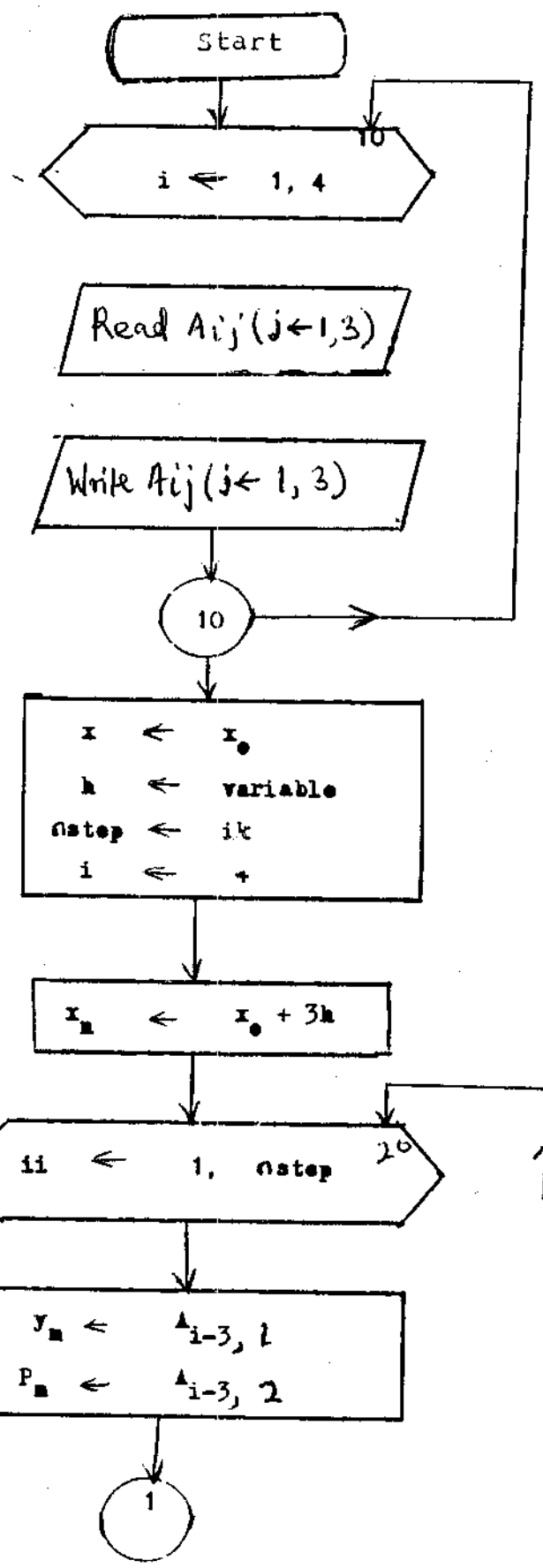
Step 4. Calculate the mop-ups using equation (14e) and(14f)

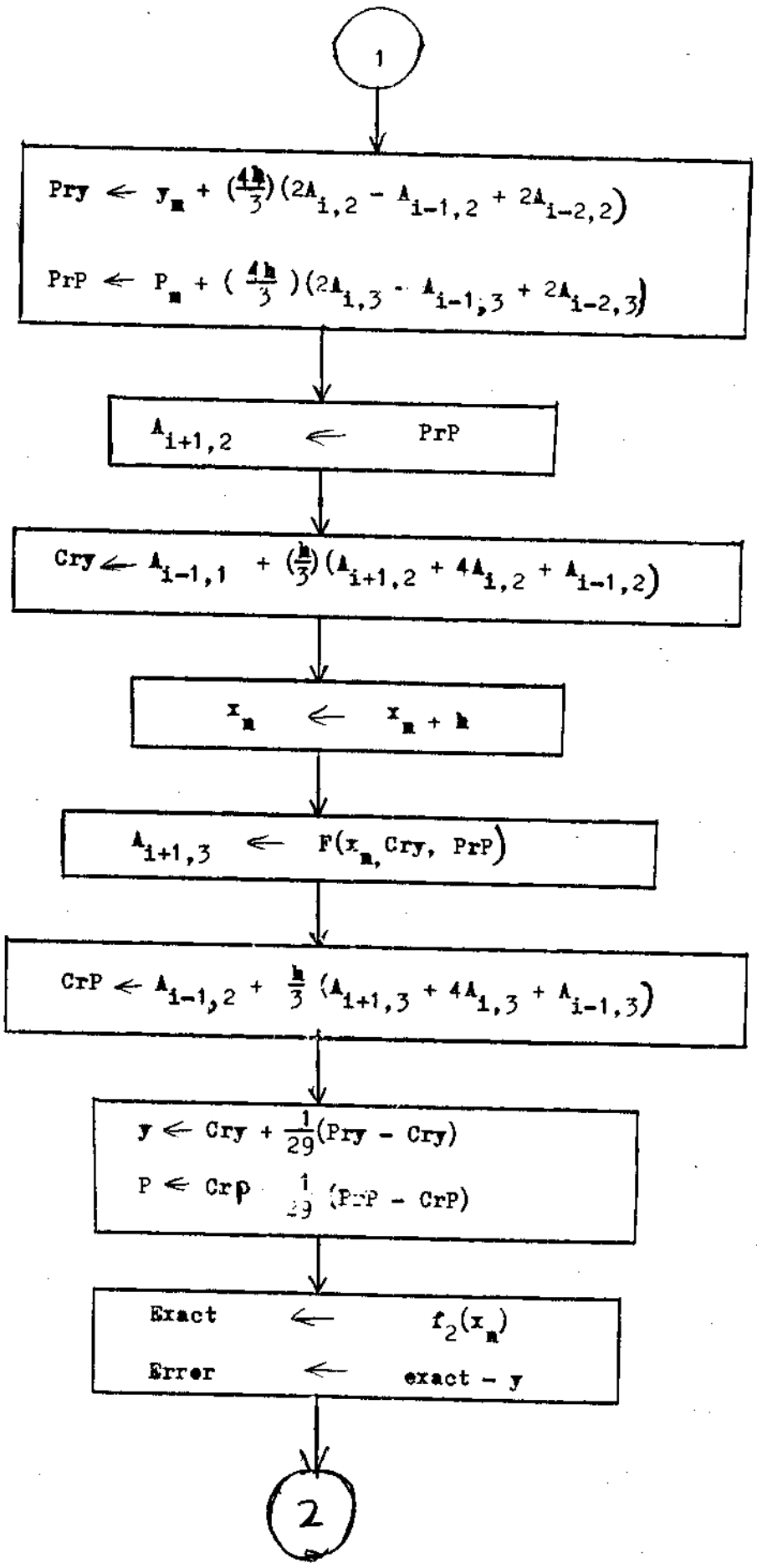
Step 5. Calculate exact solutions and errors.

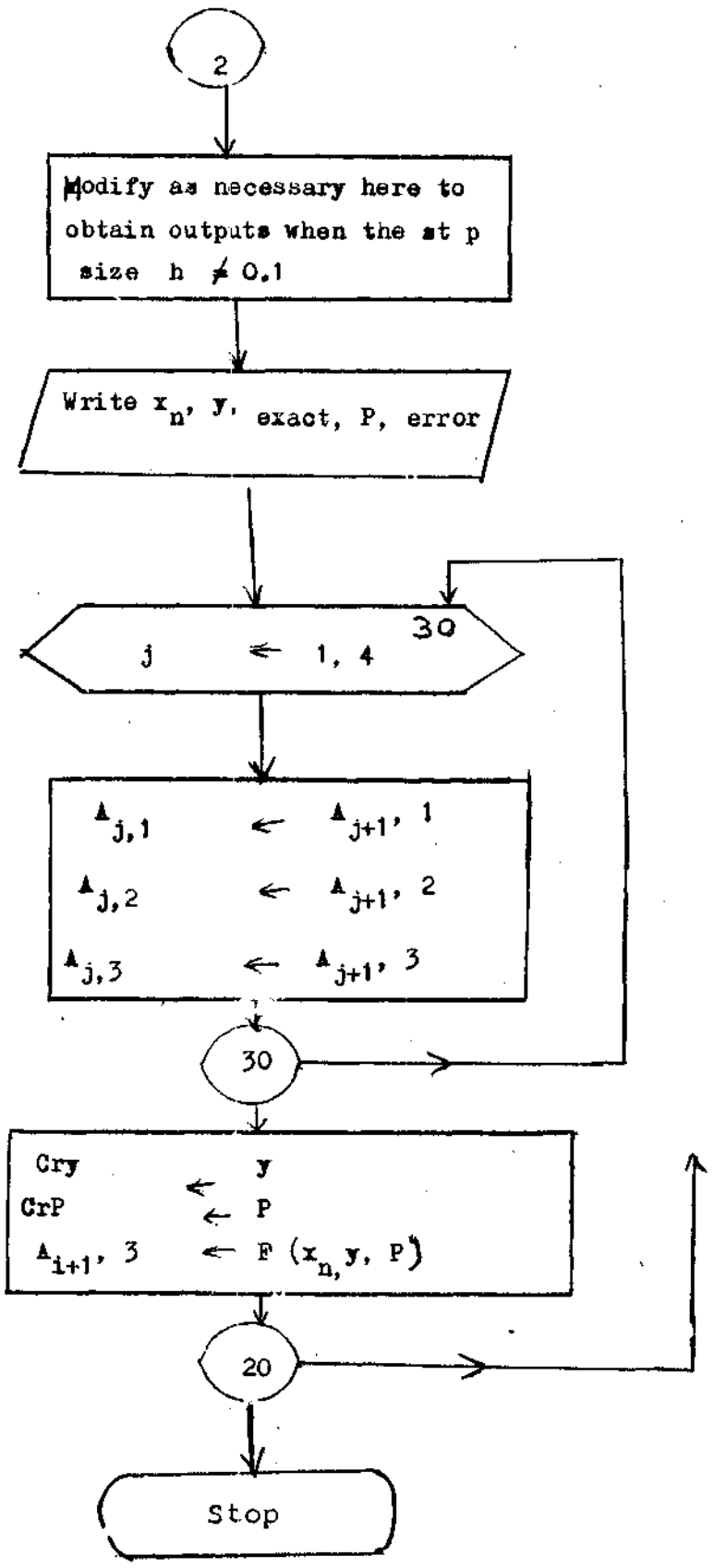
Step 6. Inter-change locations for the subsequent iterations.

Flow Chart IV

The Flow chart IV represents pictorially the algorithm steps 1-6 under program planning above based on the fourth-order Milne's method. It solves numerically systems 3 and 4 in the region D1 and D2 and records required solutions at each point in R1 and R2 respectively.


Start
10
i ← 1, 4
Read Aij (j←1,3)
Write Aij (j← 1, 3)
10
x ← x0
h ← variable
nstep ← ik
i ← +
xn ← x0 + 3h
ii ← 1, nstep
20
yn ← Ai-3, 1
Pn ← Ai-3, 2
1


1
Pry ← $y_n + (\frac{4h}{3})(2A_{i,2} - A_{i-1,2} + 2A_{i-2,2})$
PrP ← $P_n + (\frac{4h}{3})(2A_{i,3} - A_{i-1,3} + 2A_{i-2,3})$
$A_{i+1,2}$ ← PrP
Cry ← $A_{i-1,1} + (\frac{h}{3})(A_{i+1,2} + 4A_{i,2} + A_{i-1,2})$
x_n ← $x_n + h$
$A_{i+1,3}$ ← $F(x_n, Cry, PrP)$
CrP ← $A_{i-1,2} + \frac{h}{3}(A_{i+1,3} + 4A_{i,3} + A_{i-1,3})$
y ← $Cry + \frac{1}{29}(Pry - Cry)$
P ← $CrP + \frac{1}{29}(PrP - CrP)$
Exact ← $f_2(x_n)$
Error ← exact - y
2


2
Modify as necessary here to obtain outputs when the st p size h ≠ 0.1
Write x_n, y, exact, P, error
30
j ⇐ 1, 4
$A_{j,1}$ ← A_{j+1}, 1
$A_{j,2}$ ← A_{j+1}, 2
$A_{j,3}$ ← A_{j+1}, 3
30
Cry ← y
CrP ← P
A_{i+1}, 3 ← F (x_n, y, P)
20
Stop

Program:

The program named AWO4A uses the fourth order Milne's method given by equation (14) above which is based on the flow chart IV to solve the test problem (5).

Since this method cannot start by itself, the same starting values used in AWO3A are used to start it.

Outputs:

Error-values are obtained at the same selected values of x as could be seen in the following Table. Detailed outputs are obtained in AWO4A under Appendix A.

Table 3.7

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	-	-	.01325	.00385
1.3	6.25741	-	.10463	.03159	.00886
1.5	10.74344	.58353	.18653	.05657	.01585
1.7	16.83598	.88828	.29510	.08969	.02514
1.9	24.78947	1.27285	.43372	.13207	.03703
2.0	29.54517	1.51437	.51542	.15708	.04405

Observation:

The same features revealed in Table 3.5 are also revealed in Table 3.7, except that Milne's method gives better accuracy in the region considered as could be seen from the error-values in Table 3.7 when compared with Table 3.5. This is also true when the error outputs of AWO3A and AWO4A under Appendix A are compared. Also the errors are tolerable when compared with the exact solution in column 2 of Table 3.7.

3.4.2 System 4

The fourth order Milne's method for solving system 4 is also a modification of (14) and is given by:

$$\text{Pre } y_{k+1} = y_{k-3} + \frac{4h}{3}\,(2p(x_k, y_k) - p(x_{k-1}, y_{k-1}) + 2P(x_{k-2}, y_{k-2})) \quad \ldots \quad (15a)$$

$$\text{Pre } P_{k+1} = P_{k-3} + \frac{4h}{3}(2F\,(x_k, y_k) - F(x_{k-1}, y_{k-1}) + 2F(x_{k-2}, y_{k-2})) \quad \ldots\ldots \quad (15b)$$

Followed by:

$$\text{Cor } y_{k+1} = y_{k-1} + \frac{h}{3}(P(x_{k+1}, y_{k+1}) + 4P(x_k, y_k) + P(x_{k-1}, y_{k-1})) \quad \ldots (15c)$$

$$\text{Cor } P_{k+1} = P_{k-1} + \frac{h}{3}(F(x_{k+1},y_{k+1}) + 4F(x_k,y_k) + F(x_{k-1},y_{k-1})) \quad \ldots \quad (15d)$$

and completed by the mop-ups:

$$y = \text{Cor } y + \frac{1}{29}(\text{pre } y - \text{Cor } y) \quad \ldots \quad (15e)$$

$$P = \text{cor } p + \frac{1}{29}(\text{pre } p - \text{cor } p) \quad \ldots \quad (15f)$$

Program:

The program named AWO4B uses Milne's method (15) stated above which is based on the modified Flow Chart IV to solve the test problem (6).

As in Adams-Moulton method, Milne method cannot start by itself. So we have to use the same starting values used in AWO3B to start it.

Outputs:

Selected outputs are as given in Table 3.8. Detailed outputs are obtained in AWO4B.

Table 3.8

X-VAL	EX.SOL	ABSOLUTE ERROR-VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526			.00029	.00007
0.3	.20248		.00094	.00022	.00005
0.5	.41218	.00703	.00067	.00015	.00003
0.7	.70481	.00626	.00044	.00010	.00002
0.9	1.10682	.00568	.00025	.00006	.00001
1.0	1.35914	.00108	.00016	.00004	.00000

Observation:

The same features as in Adams-Moulton's method, Table 3.6 are revealed by Table 3.8 except that this method is more accurate than Adams-Moulton's method. This is obvious from the smaller error-values recorded in each column of Table 3.8. For example in column 6 of Table 3.6 under H = 0.0125 the last error value recorded is .00004 whereas in the corresponding column of Table 3.8 the last error value recorded is .00000 which confirms that Milne's method is more accurate than Adams-Moulton's method. The error values in both Table 3.6 and Table 3.8 are, however, acceptable when compared with the corresponding exact solutions in column 2 of both tables.

3.5 Taylor Series Expansion Methods 1 and 2

In this section the Taylor series expansion method is divided into two parts called Method 1 and Method 2 respectively. Method 1 solves the ordinary differential equations(5) and (6) after reduction to systems 3 and 4 respectively, while Method 2 solves the same equations directly without reducing them to system 3 or 4.

3.5.1 Method 1

The Taylor series expansion method 1 for solving system 3 is given by:

$$y_k = y_{k-1} + hy'_{k-1} + \frac{h^2}{2!}y''_{k-1} + \frac{h^3}{3!}y'''_{k-1} + \frac{h^4}{4!}y^{iv}_{k-1} \quad \ldots (16a)$$

$$p_k = p_{k-1} + hp'_{k-1} + \frac{h^2}{2!}p''_{k-1} + \frac{h^3}{3!}p'''_{k-1} \quad \ldots (16b)$$

Thus for k = 1, Equations (16a) and (16b) become

$$y_1 = y_o + hy'_o + \frac{h^2}{2!}y''_o + \ldots \frac{h^4}{4!}y^{iv}_o$$

$$p_1 = p_o + hp'_o + \frac{h^2}{2!}p''_o + \frac{h^3}{3!}p'''_o \quad \ldots (16c)$$

Using System 3, we have

$$\left.\begin{aligned} y'_o &= p_o \\ y''_o &= f(x_o, y_o, p_o) = p'_o \\ y'''_o &= f'(x_o, y_o, p_o) = p''_o \end{aligned}\right\} \quad \ldots (16d)$$

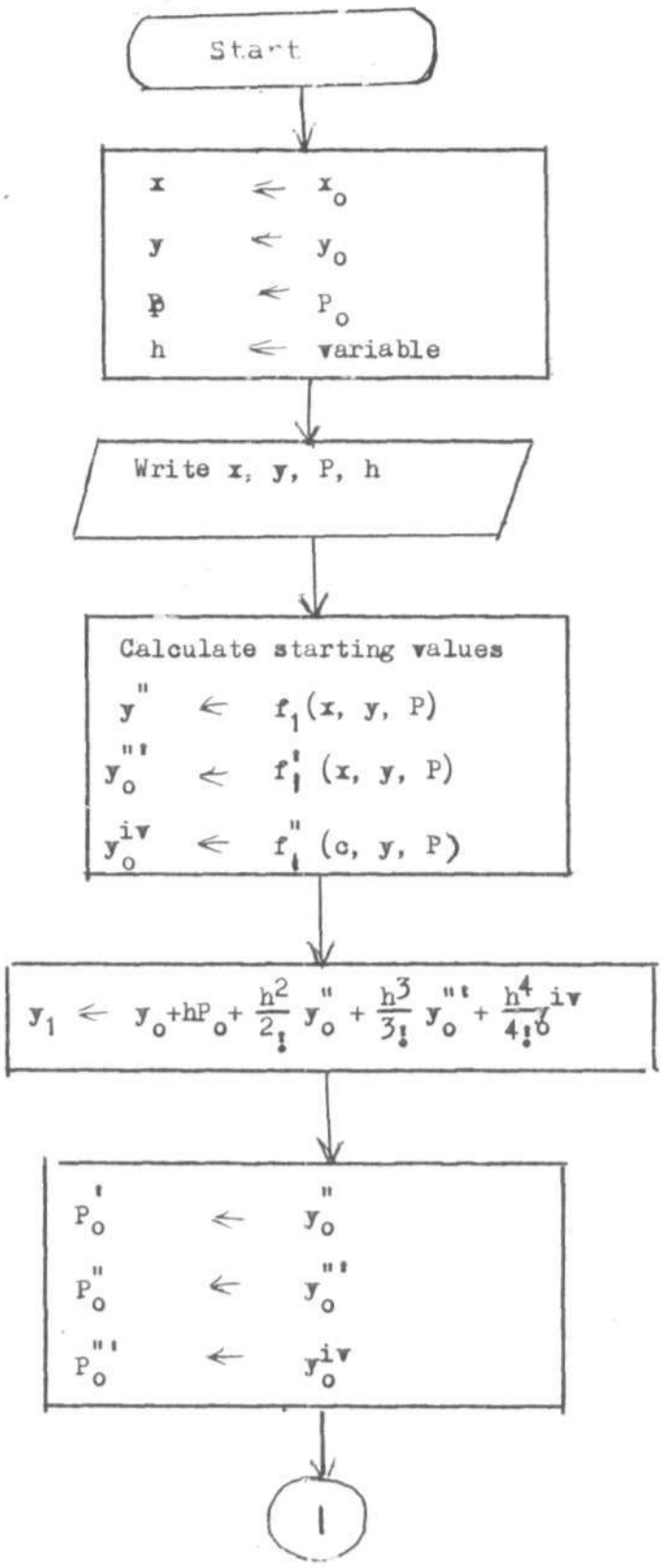
and using system 4, we have

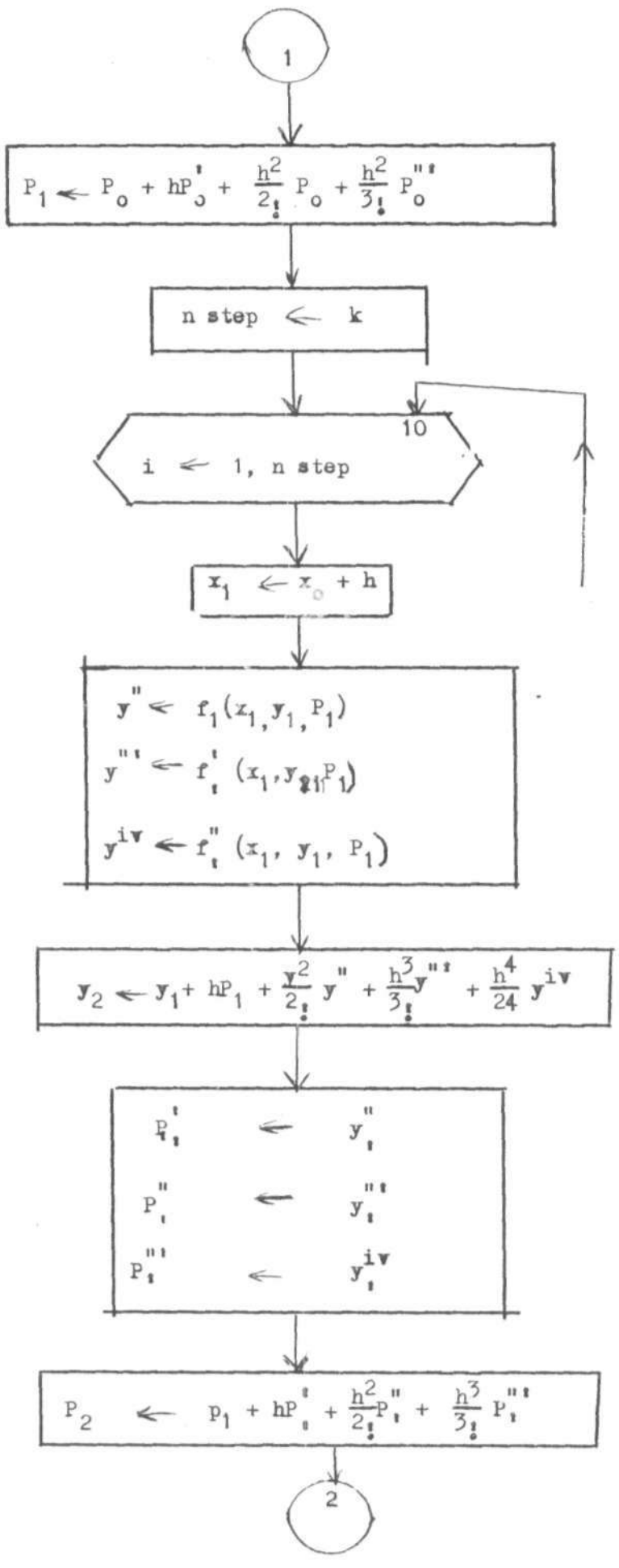
$$\left.\begin{aligned} y_o' &= p_o \\ y_o'' &= f(x_o,y_o) = p_o' \\ y''' &= f'(x_o,y_o) = p_o'' \end{aligned}\right] \quad \ldots \quad (16e)$$

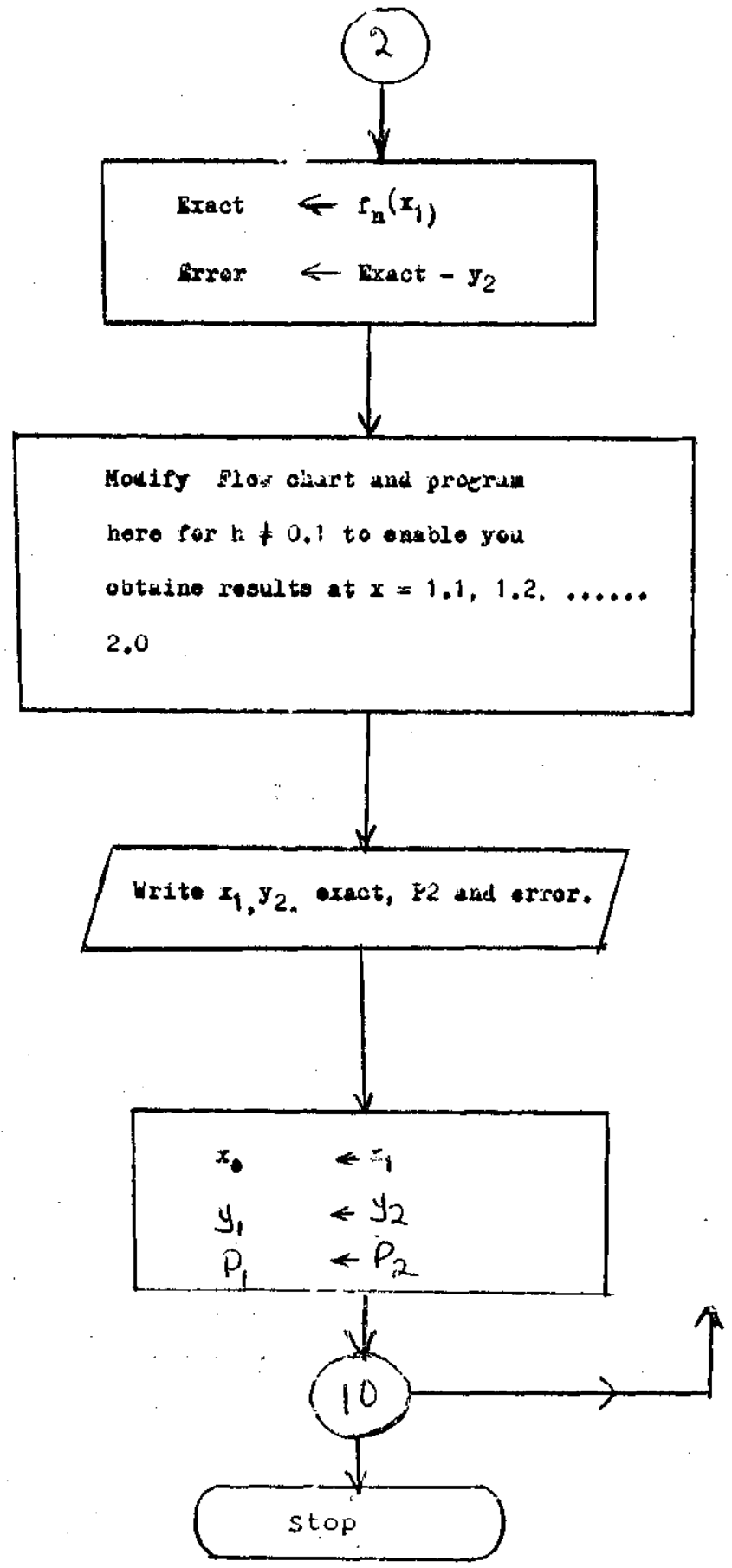
Equations (16c) and (16d) give the starting values for system 3 while equations (16c) and (16e) give the starting values for system 4.

Flow-Chart V

The Taylor series expansion Method 1 for systems 3 and 4 uses the Flow-chart V. As usual it is modified for system 4 with p = 0. This Flow chart represents algorithmically equations 16a-16d and solves numerically systems 3 and 4 respectively in the regions D1 and D2. It also records exact solutions, calculated solutions and error-values at each point in R1 and R2 respectively.


Start
x ← x_o
y ← y_o
P ← P_o
h ← variable
Write x, y, P, h
Calculate starting values
y'' ← f_1(x, y, P)
y_o''' ← f_1' (x, y, P)
y_o^iv ← f_1'' (c, y, P)
y_1 ← y_o+hP_o+ h^2/2! y_o'' + h^3/3! y_o''' + h^4/4! y_o^iv
P_o' ← y_o''
P_o'' ← y_o'''
P_o''' ← y_o^iv
1


1
$P_1 \leftarrow P_o + hP'_o + \frac{h^2}{2!} P_o + \frac{h^2}{3!} P''_o$
n step ← k
10
i ← 1, n step
$x_1 \leftarrow x_o + h$
$y'' \leftarrow f_1(x_1, y_1, P_1)$
$y''' \leftarrow f'_1(x_1, y_1, P_1)$
$y^{iv} \leftarrow f''_1(x_1, y_1, P_1)$
$y_2 \leftarrow y_1 + hP_1 + \frac{h^2}{2!} y'' + \frac{h^3}{3!} y''' + \frac{h^4}{24} y^{iv}$
$P'_1 \leftarrow y''_1$
$P''_1 \leftarrow y'''_1$
$P'''_1 \leftarrow y^{iv}_1$
$P_2 \leftarrow p_1 + hP'_1 + \frac{h^2}{2!} P''_1 + \frac{h^3}{3!} P'''_1$
2


2
Exact ← $f_n(x_1)$
Error ← Exact - y_2
Modify Flow chart and program here for h ≠ 0.1 to enable you obtaine results at x = 1.1, 1.2, ... 2.0
Write x_1, y_2, exact, P2 and error.
x_0 ← x_1
y_1 ← y_2
P_1 ← P_2
10
Stop

Program:

The program named AWO5(R4A) uses the fourth-order Taylor series expansion method 1 given by equation (16) which is based on the flow chart V to solve the test problem (5).

Outputs:

The outputs are as given in Table 3.9. Error values are compared at our selected values of x. Detailed outputs are got in AWO5(R4A) under Appendix A.

Table 3.9

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H-0.0125
1.1	3.12986	1.40920	.66782	.32495	.16027
1.3	6.25741	2.05756	.98448	.48141	.23803
1.5	10.74344	2.82856	1.36234	.66846	.33108
1.7	16.83598	3.72541	1.00308	.88693	.43985
1.9	24.78947	4.75093	2.30815	1.13757	.56470
2.0	29.54517	5.31273	2.58521	1.27516	.63326

Observation

1. Table 3.9 reveals a growing error-value for each value of H in the selected region D1. However, as H decreases, erro-values decreases correspondingly.

2. If we compare the error-values with the corresponding exact solutions in column 2 of Table 3.9, we discover that error-values under H=0.1 and H=0.05 are two large to be accepted when compared with the exact solutions. Error-values under H=0.025 and H=0.0125 could be tolerated. Hence this method should not be used when the step-size H $\geq$0.025.

3. The Table also reveals that in a region below x=1.1 say 0.9 $\leq$ x $\geq$ 0.0 with initial value x_o = 1.0, error value for each value of H will reduce which means better accuracy for the method.

Outputs of Third and Second orders Taylor Series Method 1.

Table 3.10

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	1.40904	.66780	.32495	.16027
1.3	6.25741	2.05704	.98441	.48140	.23803
1.5	10.74344	2.82756	1.36222	.66844	.33108
1.7	16.83598	3.72379	1.80287	.88690	.43984
1.9	24.78947	4.74852	2.30783	1.13753	.56469
2.0	29.54517	5.30985	2.58484	1.27511	.63325

Table 3.11

X-VAL	EX.SOL	ABSOLUTE ERROR-VALUES			
		H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	1.38286	.65988	.32204	.15905
1.3	6.25741	1.92202	.92968	.45700	.22654
1.5	10.74344	2.46748	1.20290	.59375	.29495
1.7	16.83598	2.98841	1.46314	.72381	.35997
1.9	24.78947	3.45327	1.69358	.83849	.41716
2.0	29.54517	3.65464	1.79228	.88730	.44143

Observation:

1. The same features as obtained in Table 3.9 repeated themselves in Tables 3.10 and 3.11. Hence the same arguments are valid here. Looking at the three tables, however, it is observed that accuracy in Table 3.10 is better than accuracy in Table 3.9. Similarly, accuracy in Table 3.11 is better than accuracy in Table 3.10. For example if we look at Column 6 under H=0.0125 of Tables 3.9 and 3.10 we see that error-values at x = 1.1 to x = 1.5 are the same. But from x = 1.7 to x = 2.0 each error-value in Table 3.9 is slightly greater than the corresponding error-value in Table 3.10 by 0.00001.

2. Table 3.11 gives the best accuracy for this method as could be seen from the least error-values. One expects that accuracy should be best in Table 3.9 followed by Table 3.10 and lastly Table 3.11 but it is the other way round. This may be due partly to the behaviour of the test problem in the region of observation and partly due to the accumulated round off error of Taylor series method.

Detailed outputs from which Tables 3.10 and 3.11 were extracted are in AWO5(R3A) and AWO5(R2A) respectively under Appendix A.

Programs for System 4

The Program named AWO5(R4B) uses the fourth-order Taylor series expansion Method 1 to solve the test problem (6) in the form of system 4.

Outputs

The selected outputs are given in Table 3.12. As usual we could obtain detailed outputs in AWO5(R4B) under Appendix B.

Table 3.12

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.06688	.03188	.01556	.00769
0.3	.20248	.09588	.04586	.02243	.01109
0.5	.41218	.13443	.06446	.03156	.01562
0.7	.70481	.18536	.08906	.04365	.02161
0.9	1.10682	.25224	.12138	.05954	.02949
1.0	1.35914	.29305	.14111	.06924	.03430

The description of the third and second orders Taylor series expansion method 1 are the same same as that of fourth order Taylor series described above. Hence their selected outputs will be given only for discussion. Detailed outputs can be obtained in AWO5(R3B) and AWO5(R2B) respectively under Appendix B.

307241

Table 3.13

X-VAL	EX-SOL	ABSOLUTE ERROR VALUE			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.06683	.03187	.01556	.00769
0.3	.20248	.09562	.04580	.02241	.01109
0.5	.41218	.13375	.06431	.03153	.01561
0.7	.70481	.18397	.08874	.04358	.02159
0.9	1.10682	.24976	.12080	.05940	.02945
1.0	1.35914	.28984	.14036	.06906	.03425

Table 3.14

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.06554	.03148	.01542	.00763
0.3	.20248	.08940	.04331	.02131	.01057
0.5	.41218	.11773	.05733	.02828	.01405
0.7	.70481	.15156	.07404	.03659	.01819
0.9	1.10682	.19201	.09402	.04652	.02314
1.0	1.35914	.21511	.10542	.05218	.02596

Observation

1. As usual a growing error values occurs in each column of Table 3.12, Table 3.13 and Table 3.14 respectively from H=0.1 to H=0.0125. As H reduces in value, however, accuracy also increases as could be observed from the reducing error-values with the least values of errors in the column H=0.0125 when compared with other columns.

2. A better accuracy will result generally if H< 0.0125 in our region of interest. This is obvious from the last two sets of the detailed error outputs in AWO5(R2B) under Appendix B.

3. We could also observe that the rate of decrease in error values in each row is

approximately the same as the rate at which H is decreasing. H is chosen to decrease by half its previous value for succeeding set of outputs. The outputs in each row also give nearly the same picture. For example in row 1 of Table 3.12 at x = 0.1, .03188 is approximately ½(.06688) and so on. This is true for all other rows.

4. It is also observed that the best accuracy is recorded for the second order Taylor series expansion method followed by the third order Taylor series and lastly by the fourth order Taylor series expansion method. This is confirmed by the decreasing error value outputs as we move from the fourth order to the third order and then to the second order Taylor series, although some of the output for the fourth and third orders Taylor series are correspondingly the same.

For example in Table 3.12 which is the selected output for the fourth-order Taylor series method, the first two outputs under H=0.0125 are equal to the corresponding values recorded in Table 3.13. But the third order Taylor series method still produces better accuracy than the fourth-order and the second order more accurate than the third order. The reason for this is due to the growing round off error of the Taylor series expansion method.

3.5.2 Taylor Series Expansion Method 2

The Taylor series method 2 for solving ordinary differential equations of the forms (1) and (2) directly is still given by (16a). The starting value is also given by (16_c) in section (3.5.1) as

$$y_1 = y_o + hy_o' + \frac{h^2}{2!}y_o'' + \frac{h^3}{3!}y_o''' + \frac{h^4}{4!}y_o^{iv}$$

$$\therefore y_2 = y_1 + hy_1' + \frac{h^2}{2}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1^{iv} \quad \ldots(17a)$$

where y_1' is defined by Spiegel (35) as

$$y_1' = \underset{h \to 0}{Lt} \quad \frac{y_1 - y_o}{h} \doteq \frac{y_1 - y_o}{h} \quad \ldots \quad (17b)$$

Putting (17b) into (17a) we have

$$y_2 = 2y_1 - y_o + \frac{h^2}{2!}y_1'' + \frac{h^3}{3!}y_1''' + \frac{h^4}{4!}y_1^{iv}$$

Also similarly

$$y_3 = 2y_2 - y_1 + \frac{h^2}{2!}y_2'' + \frac{h^3}{3!}y_2''' + \frac{h^4}{4!}y_2^{iv}$$

$$\vdots$$

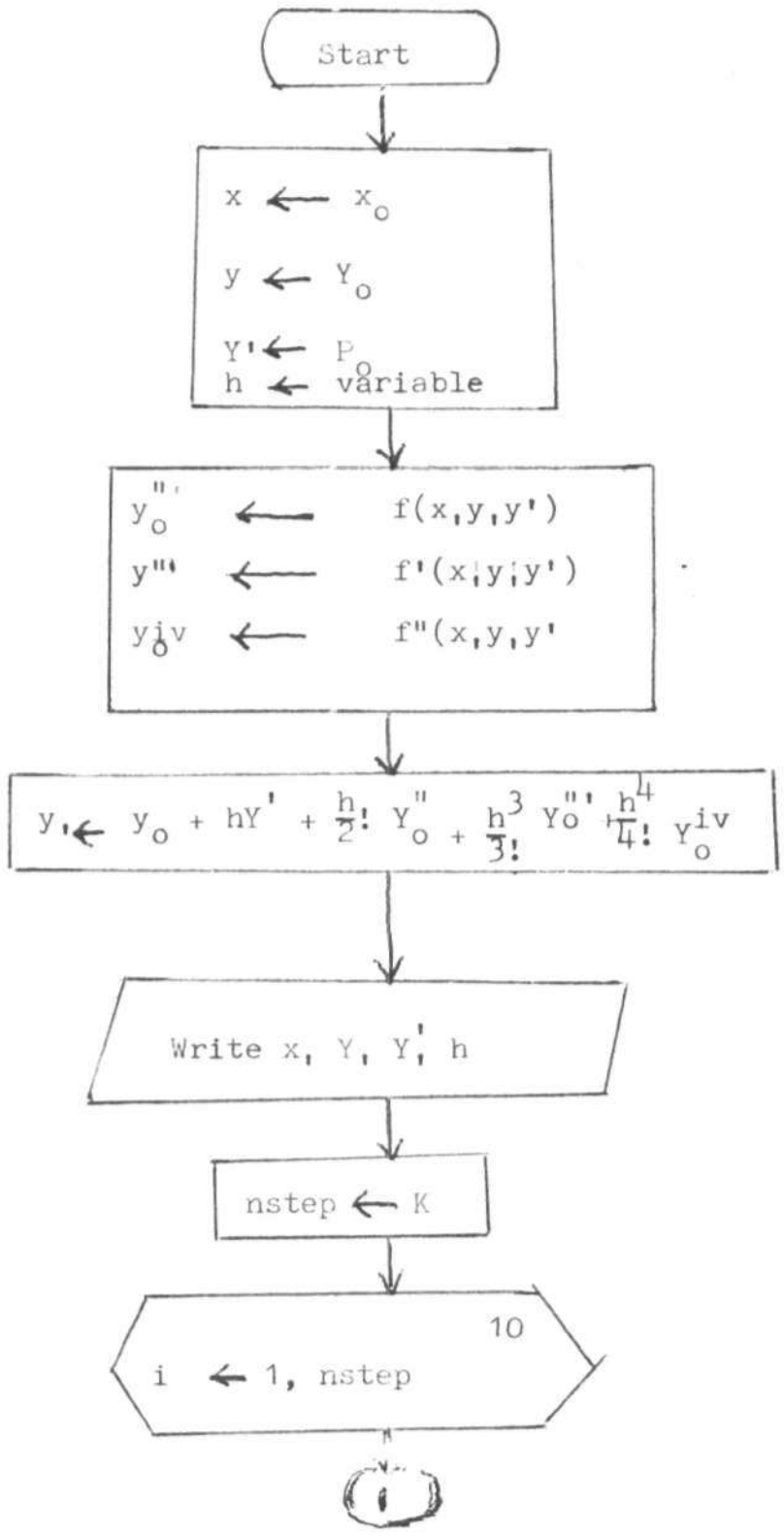
$$y_{k+1} = 2y_k - y_{k-1} + \frac{h^2}{2!}y_2'' + \frac{h^3}{3!}y_2''' + \frac{h^4}{4!}y_2^{iv}$$

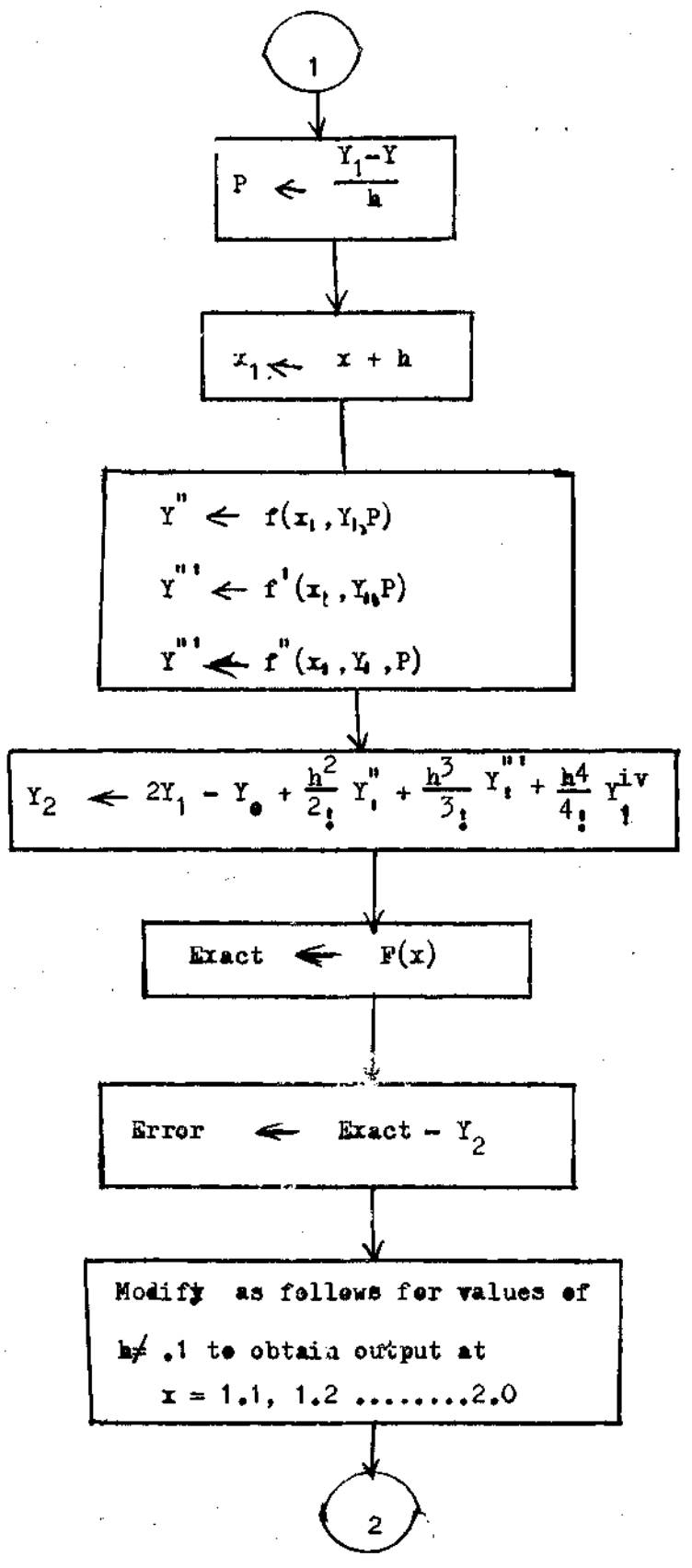
To calculate the starting value y_1 we use (16e) while y_1' is obtained from (17b).

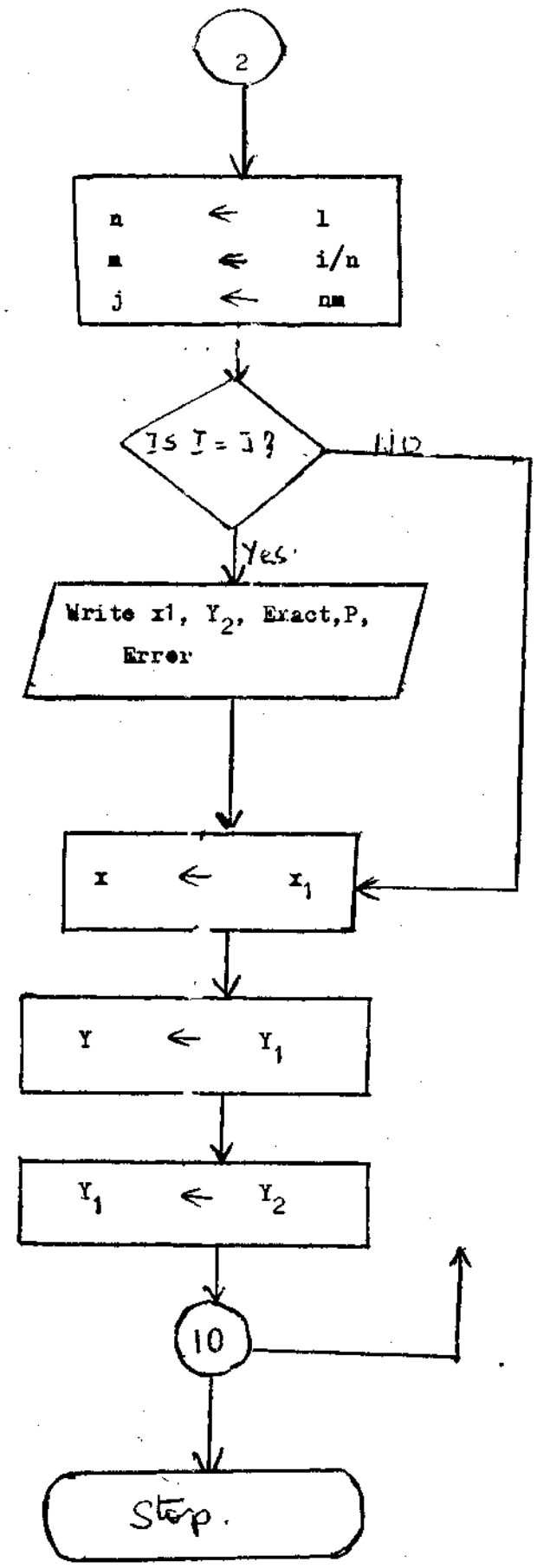
The flow chart gives the details.

Flow-Chart VI

The Flow chart VI is based on equations (16a), (16c), (17a) and (17b). As usual it solves second order ordinary differential equations (1) and (2) directly in the region D1 and D2 and records solutions at each point in R1 and R2 respectively.


Start
x ← x_o
y ← Y_o
Y' ← P_o
h ← variable
y"_o ← f(x,y,y')
y"' ← f'(x,y,y')
y^iv_o ← f"(x,y,y'
y, ← y_o + hY' + h/2! Y"_o + h^3/3! Yo"' + h^4/4! Y^iv_o
Write x, Y, Y', h
nstep ← K
10
i ← 1, nstep


1
P ← (Y1-Y)/h
x1 ← x + h
Y" ← f(x1, Y1, P)
Y"' ← f'(x1, Y1, P)
Y"' ← f"(x1, Y1, P)
Y2 ← 2Y1 - Y0 + h^2/2! Y1" + h^3/3! Y1"' + h^4/4! Y1^iv
Exact ← F(x)
Error ← Exact - Y2
Modify as follows for values of
h = .1 to obtain output at
x = 1.1, 1.2 .....2.0
2


2
n ← 1
m ← i/n
j ← nm
Is I = J ?
No
Yes
Write xi, Y2, Exact, P,
Error
x ← x1
Y ← Y1
Y1 ← Y2
10
Stop.

Program for System 3

The program named AWO6(D4A) uses the Taylor series expansion method 2 given by Egs.(16) and (17) based on the Flow-chart VI to solve the test problem (5) which is in the form system 3.

Outputs

The outputs at our selected values of x are as given in Table 3.15 below while the detailed outputs are in AWO6(D4A).

Table 3.15

X-VAL	EX.SOL	ABSOLUTE ERROR VALUE			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	1.25553	.55922	.23719	.08251
1.3	6.25741	1.00315	.10849	.31053	.51316
1.5	10.74344	.13783	1.23420	1.75188	2.00329
1.7	16.83598	2.42821	3.73050	4.34892	4.65016
1.9	24.78947	6.13018	7.64316	8.36473	8.71699
2.0	29.54517	8.59296	10.21322	10.98733	11.36558

Observation

1. In column 3 under H=0.1, the error-value decreases starting from x=1.1 to x=1.5 and start to grow from x=1.7 to x=2.0

2. The same picture as revealed in column 3 under H=0.1 of Table 3.15 is revealed in column 4 and 5 under H=0.05 and H=0.025 in

the detailed output of AWO6(D4A) under Appendix A, although error is growing from x=1.1 to x=2.0 in column 5 under H=0.025 in Table 3.15 above. The reason is that x=1.2 is not considered in Table 3.15.

3. In the last column however the error value starts to grow from x=1.1 to x=2.0. Further observation of the table reveals that in row 1 at x=1.1 the error value decreases from 1.25553 for H=0.1 to 0.08251 for H=0.0125. But from the rows at x=1.7 to x=2.0 error value increases from H=0.1 to H=0.0125 with the largest value of error in the last row.

4. If the error values under each value of H are compared with the corresponding values of the exact solutions in column 2 we discover that apart from the instability of the error values under H=0.1 to H=0.025, the values become so high generally from x=1.7 to x=2.0 and also increasing with decreasing values of H. Hence this method is highly unstable for the test problem and should not be used to solve such problems.

Outputs of Third and Second order Taylor Series Method 2

Table 3.16

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	1.25545	.55921	.23719	.08251
1.3	6.25741	1.00289	.10844	.31054	.51316
1.5	10.74344	.13838	1.23431	1.75190	2.00329
1.7	16.83598	2.42916	3.73070	4.34897	4.65017
1.9	24.78947	6.13164	7.64348	8.36480	8.71701
2.0	29.54517	8.59472	10.21361	10.98742	11.36561

Table 3.17

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
1.1	3.12986	1.24081	.55554	.23604	.08208
1.3	6.25741	.95332	.09102	.31763	.51633
1.5	10.74344	.24385	1.27500	1.77020	2.01183
1.7	16.83598	2.61194	3.80745	4.38389	4.66680
1.9	24.78947	6.41349	7.76629	8.42188	8.74449
2.0	29.54517	8.93435	10.36362	11.05767	11.39956

Table 3.16 and 3.17 reveal similar features as obtained in Table 3.15. Therefore the same comments are maintained. It is observed, however, that each error values in Table 3.15 is very close to its corresponding values in 3.16 and 3.17. For example in column 6 under H=0.0125 we have each error value in Table 3.15 equal to its corresponding values in Table 3.16 from x=1.1 to x=1.3

From the results given in Tables 3.15 - 3.17 we could see that the solutions are not stable. Hence, this method cannot be used to solve the test problem.

Programs for System 4

The programs named AWO6(D4B), AWO6(D3B) and AWO6(D2B) use the Taylor series expansion method 2 based on the modified flow-chart vi to solve the test problem (6) in the form of system 4.

Outputs

The outputs of the programs in the order specified above at p2 are as given below. Detailed outputs are in AWO6(D4B), AWO6(D3B) and AWO6(D2B) respectively under Appendix B.

Table 3.18

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.06135	.02783	.01223	.00471
0.3	.20248	.05886	.01431	.00648	.01652
0.5	.41218	.03063	.02811	.05559	.06887
0.7	.70481	.03315	.11004	.14606	.16350
0.9	1.10682	.14579	.24578	.29269	.31541
1.0	1.35914	.22599	.33956	.39297	.41884

Observation

1. It is observed that in each column from H=0.1 to H=0.025 error-value is decreasing up to a point and then starts to increase up to the last row. However, the last column under H=0.0125 shows a different picture. Here error value grows from .00471 to .41884.

2. It could be seen also that as H is decreasing in value, error value is seen to be magnifying correspondingly, especially from row x=0.7 to row x=1.0. In row x=0.1, however, error value is observed to decrease with decreasing value of H.

3. From (1) and (2) above it could be concluded that the Taylor series method 2 is unstable for system 4.

Outputs for AW06(D3B)

Table 3.19

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.06132	.02783	.01223	.00471
0.3	.20248	.05876	.01429	.00649	.01652
0.5	.41218	.03041	.02816	.05560	.06887
0.7	.70481	.03357	.11013	.14608	.16350
0.9	1.10682	.14651	.24593	.29272	.31541
1.0	1.35914	.22670	.33975	.39301	.41885

Observation

1. The same features of instability revealed in Table 3.18 repeated themselves here. Therefore the same comments for Table 3.18 is valid for Table 3.19 as well.

2. It should be observed however that each error value under H=0.0125 in Table 3.18 is the same as the corresponding error value under H=0.0125 in Table 3.19 except for the last error value of each with a difference of .00001 in their error values.

Outputs for AW06(D2B)

Table 3.20

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.06054	.02763	.01217	.00469
0.3	.20248	.05596	.01331	.00688	.01670
0.5	.41218	.02402	.03065	.05669	.06938
0.7	.70481	.04553	.11508	.14832	.16456
0.9	1.10682	.16658	.25453	.29669	.31731
1.0	1.35914	.25200	.35073	.39810	.42130

Observation

The same comments for Table 3.18 and 3.19 will be maintained here for Table 3.20 as the same features are similarly revealed.

3.6 The Classical Numerov's Method

The classical Numerov's method for solving the test problem (6) directly is given by Dodes(8) as

$$Y_{k+1} = 2Y_k - Y_{k-1} + \frac{h^2}{12}(F_{k+1} + 10F_k + F_{k-1}) \quad \ldots(18)$$

We shall apply this method to two cases. Firstly we shall use the formula as we put it above, but find Y_1 and Y_2 by Taylor series method. Secondly we shall transfer $\frac{h^2}{12}F_{k+1}$ to the left hand side and make Y_{k+1} the subject of the equation. Here we don't need to expand Y_2 by Taylor series.

3.6.1 Case 1

In Eq.(18) if we make k=1 it is possible to find the starting value Y_1 using Taylor series. To improve the accuracy of Y_1 we set h to $\frac{1}{2}h$, i.e. $h = \frac{1}{2}h$. Thus:

$$Y_1 = Y(x_o+h) = Y_o + hy'_o + \tfrac{1}{2}hy''_o + \frac{1}{3!}h^3Y''_o \quad \ldots (19)$$

where the values of Y_o and Y'_o are calculated from the initial condition x_o. Also from Eq.(19) above:

$$\left.\begin{aligned} Y''_o &= F(x_o, Y_o) \\ Y'''_o &= F'(x_k, Y_o) \end{aligned}\right\} \quad \ldots\ldots\ldots (20)$$

Substituting (20) in (19) we obtain the starting value Y_1.

Next we calculate the predictor value of $Y_{k+1} = Y_2$ at $x_2 = x_o + 2h$ using a portion of equation (18).

Thus Pre $Y_2 = 2Y_1 - Y_o + \frac{h^2}{12}(10F_1 + F_o)$ ... (21)

i.e. $\text{Pre}Y(x_2) = 2Y(x_1) - Y(x_o) + \frac{h^2}{12}(10Y''(x_1) + Y''(x_o))$

where Y'' is calculated from the original equation. To correct (21) we add $\frac{h^2}{12}F_{k+1}$

where $F_{k+1} = F(x_{k+1}, Y_{k+1})$. $Y_{k+1} = Y_2$ for $k=1$

using the Taylor series expansion again.

$$Y_2 = Y_1 + hy_1' + \frac{h^2}{2!}y'' + \quad \ldots \qquad \ldots (22)$$

Next we use the definition of a derivative to evaluate Y_1'. Thus:

$$f'(x_o) = \underset{h \to 0}{Lt} \frac{f(x_o+h) - f(x_o)}{h}$$

for small h, $f(x_o+h) = f(x_o)$

$$f'(x_o + h) = \underset{h \to 0}{Lt} \frac{f(x_o + h) - f(x_o)}{h}$$

$$\text{i.e.} \quad Y_1' \doteq \frac{Y_1 - Y_o}{h} \qquad \ldots (23)$$

Similarly $Y_2' \doteq \frac{Y_2 - Y_1}{h}$

$$\vdots$$

$$Y_k' \doteq \frac{Y_k - Y_{k-1}}{h}$$

Also $Y'' = F(x_1, Y_1)$ ... (24)

Now substituting Eqs (23 and (27) in Eq. (22) we have Y_2

Thus we can continue to calculate Y_2 iteratively up to the desired limit.

Flow-Chart VII

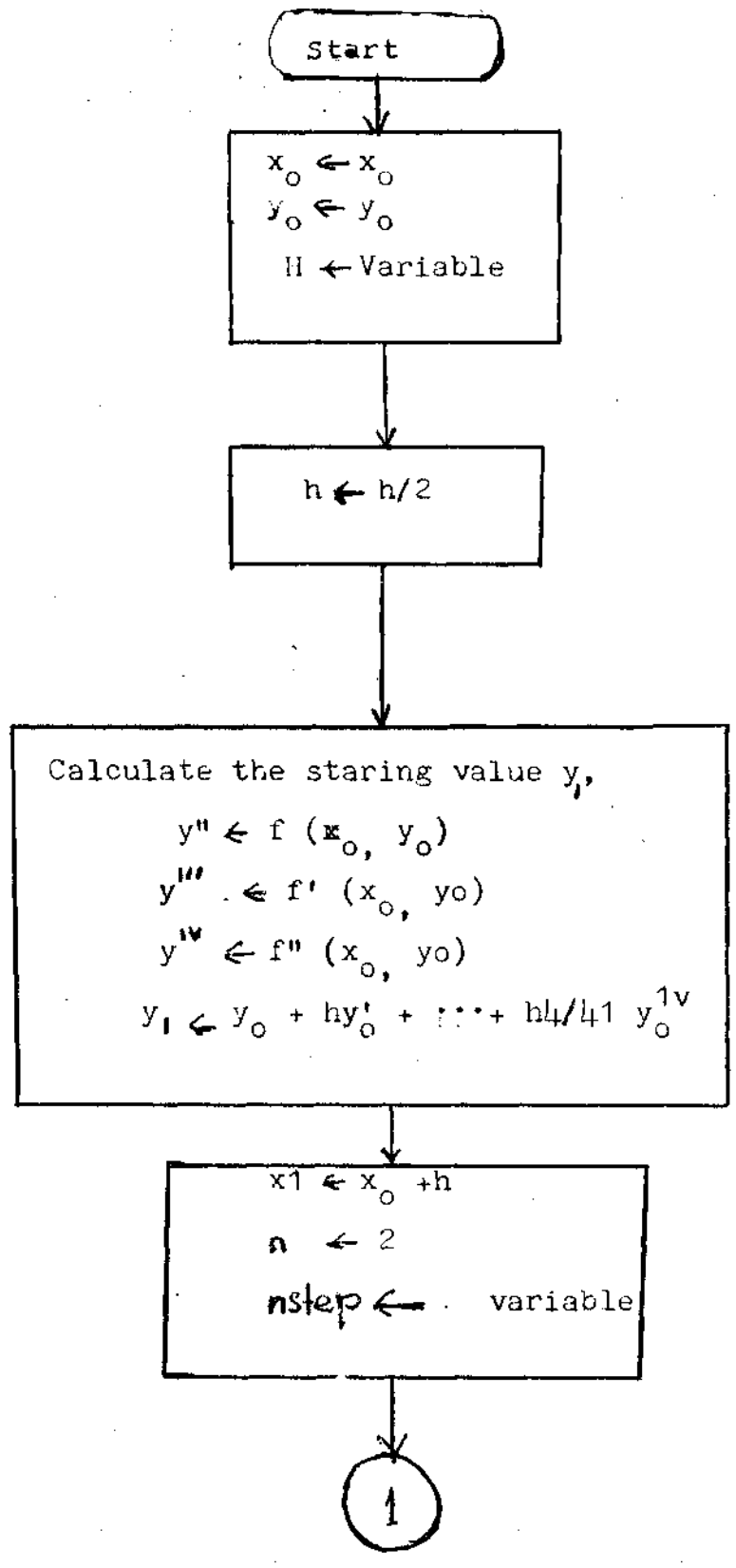
The flow-chart VII on the next page solves system 4 numerically in the region D2 and records exact solutions, calculated solutions and error-values at each point in R2.

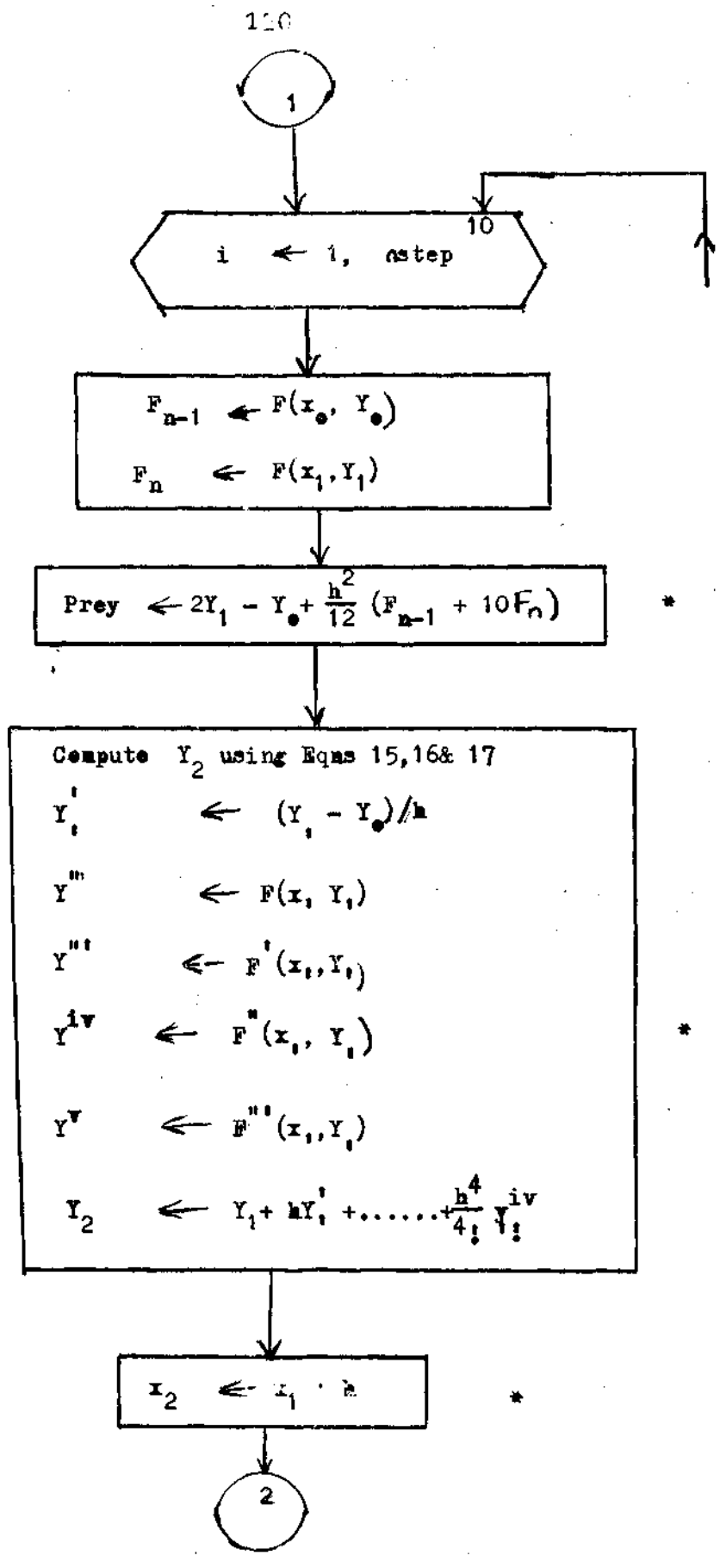
Program:

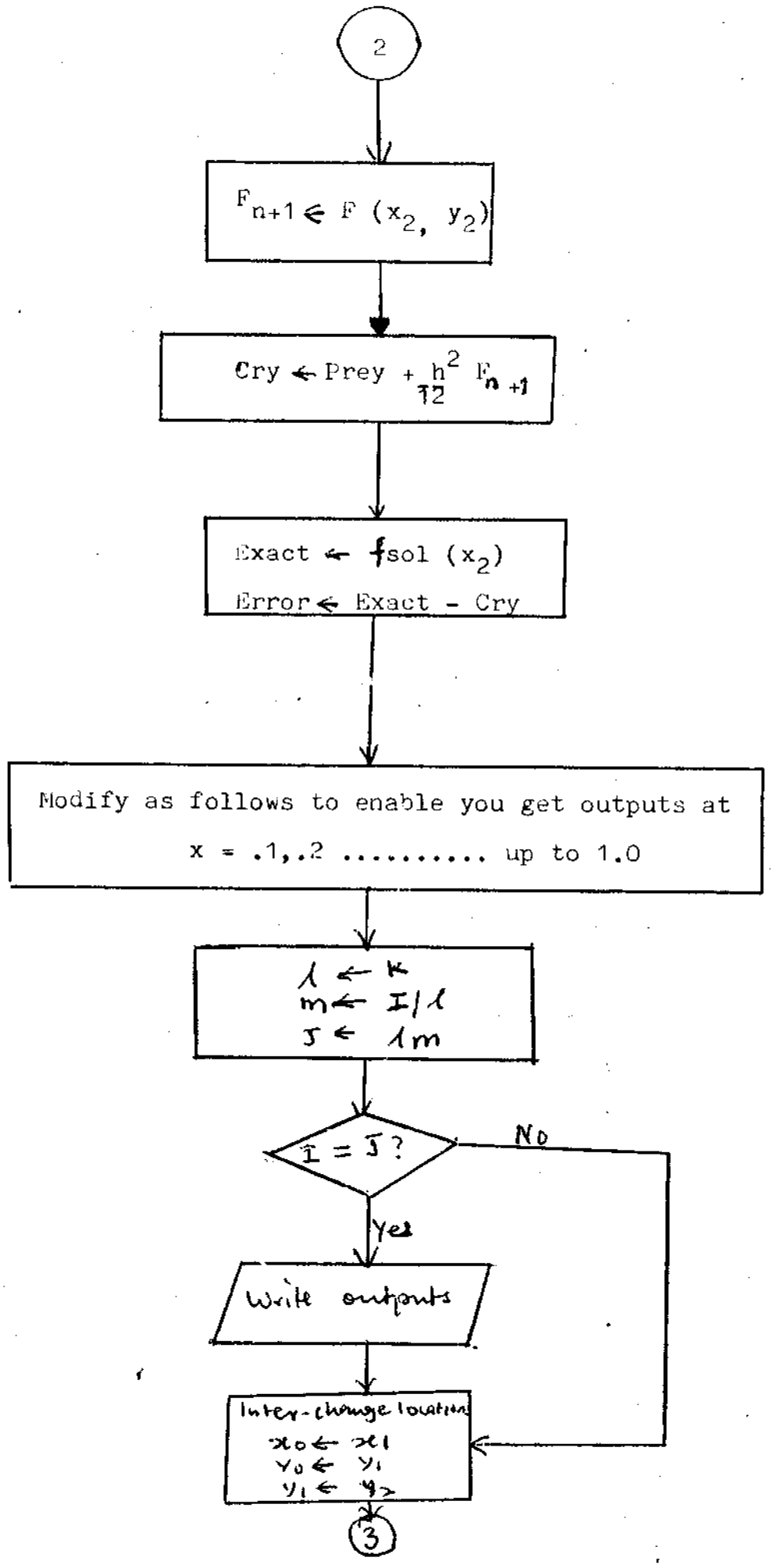
The program named AW07A uses the classical Numerov's method given by equation (18) based on the flow chart VII to solve the test problem (6) directly.

Outputs:

Table 3.21 below shows the error values at the selected values of x. Detailed outputs are as usual found in AW07A in Appendix B.


Start
$x_o \leftarrow x_o$
$y_o \leftarrow y_o$
H ← Variable
h ← h/2
Calculate the staring value y_1,
$y'' \leftarrow f(x_o, y_o)$
$y''' \leftarrow f'(x_o, yo)$
$y^{iv} \leftarrow f''(x_o, yo)$
$y_1 \leftarrow y_o + hy'_o + \cdots + h4/41\ y_o^{1v}$
x1 ← x_o +h
n ← 2
nstep ← variable
1


120
1
10
i ← 1, nstep
$F_{n-1} \leftarrow F(x_0, Y_0)$
$F_n \leftarrow F(x_1, Y_1)$
Prey $\leftarrow 2Y_1 - Y_0 + \frac{h^2}{12}(F_{n-1} + 10F_n)$
Compute Y_2 using Eqns 15,16& 17
$Y_1' \leftarrow (Y_1 - Y_0)/h$
$Y''' \leftarrow F(x_1, Y_1)$
$Y''' \leftarrow F'(x_1, Y_1)$
$Y^{iv} \leftarrow F''(x_1, Y_1)$
$Y^{v} \leftarrow F'''(x_1, Y_1)$
$Y_2 \leftarrow Y_1 + hY_1' + \ldots\ldots + \frac{h^4}{4!} Y_1^{iv}$
$x_2 \leftarrow x_1 \cdot h$
2


2
$F_{n+1} \Leftarrow F(x_2, y_2)$
Cry ← Prey + $\frac{h^2}{12} F_{n+1}$
Exact ← fsol (x_2)
Error ← Exact - Cry
Modify as follows to enable you get outputs at
x = .1,.2 ..... up to 1.0
I ← K
m ← I/l
J ← lm
I = J ?
No
Yes
Write outputs
Inter-change locations
$x_0 \leftarrow x_1$
$y_0 \leftarrow y_1$
$y_1 \leftarrow y_2$
3

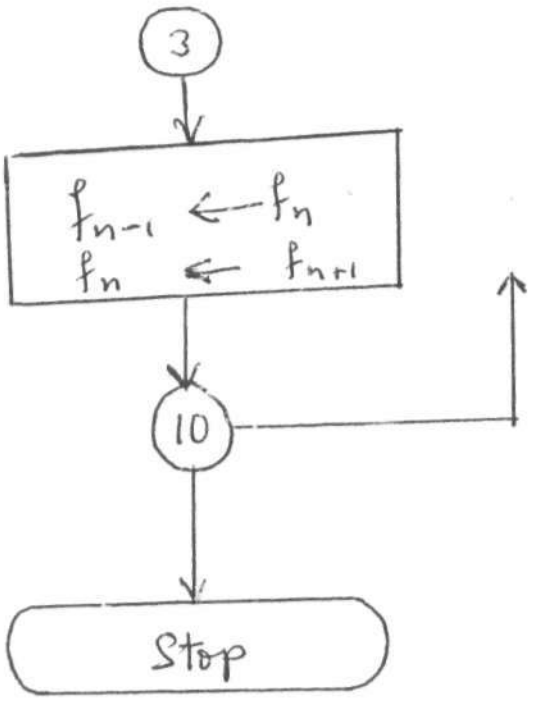

3
$f_{n-1} \leftarrow f_n$
$f_n \leftarrow f_{n+1}$
10
Stop

Table 3.21

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.00000	.00163	.00221	.00244
0.3	.20248	.02018	.02367	.02511	.02575
0.5	.41218	.07130	.07713	.07964	.08078
0.7	.70481	.16447	.17322	.17707	.17886
0.9	1.10682	.31457	.32701	.33257	.33517
1.0	1.35914	.41693	.43158	.43814	.44123

Observation

1. As usual the error values for each set of outputs in Table 3.21 from H=0.1 to H=0.0125 grow from x=0.1 to x=1.0 with the least error values in row x=0.1 and largest error values in row x=1.0. Also as H increases in values, error values also increase.

2. One of the major reasons for this increase in error values apart from the round off error arising from Taylor series expansion, is the introduction of definition of derivative:

$$f'(x_o) = \lim_{h \to o} \frac{x_o+h)-f(x_o)}{h} \doteq (y_1 - y_o)/h$$

to calculate y_1' in equation (22).

As h is decreasing, y'_1 is increasing and thus y_2 is increasing much greater than the exact solution reproduced in column 2 of Table 3.21. This is obvious in row 1 of Table 3.21 and in the detailed out- in AWO7A.

3. It could also be observed from x=0.7 to x=1.0 that error-values are generally too large when compared with the corresponding exact solutions. Hence this method is not suitable for the test problem.

3.6.2 Case 2:

Next we consider case 2. As pointed out earlier, equation (18) is given by Froberg (18) as

$$Y_{k+1} - \frac{h^2}{12}F_{k+1} = 2Y_k - Y_{k-1} + \frac{h^2}{12}(10F_k + F_{k-1}) \quad \ldots (25)$$

where $F_{k+1} = Y''_{k+1} = F(x_{k+1}, Y_{k+1})$. In terms of our test problem (6)

$$Y''_{k+1} = Y_{k+1} + e^{x_{k+1}} \quad \ldots (26)$$

Putting equation (26) in (25), we have:

$$Y_{k+1}(1 - \frac{h^2}{12}) = \frac{h^2}{12}e^{x_{k+1}} + 2Y_k - Y_{k-1} + \frac{h^2}{12}(10F_k + F_{k-1})$$

$$Y_{k+1} = \frac{12}{12-h^2}\left\{ \frac{h^2}{12}e^{x_{k+1}} + 2Y_k - Y_{k-1} + \frac{h^2}{12}(10F_k + F_{k-1})\right\} \quad \ldots \quad (27)$$

which is free from Y_i

Flow Chart VIII

The flow chart VII is modified slightly to produce the flow-chart VIII based on equation (27) by replacing the boxes marked *with the following single box.

$b1 \leftarrow 12/(12 - h^2)$
$b2 \leftarrow 2y_1 - y_o + (h^2/12)(10F_k + F_{k-1})$
$pry \leftarrow b1 \cdot b2$
$x^2 \leftarrow x1 + h$
$y2 \leftarrow pry + b1 \cdot \frac{h^2}{12}e^{x2}$

As usual the flow-chart VIII solves system 4 numerically in the region D2 and records necessary outputs at each point in R2.

Outputs

The outputs are as given by Table 3.22. Detailed outputs are in AWO7B in Appendix B.

Table 3.22:

X- VAL	EX.SOL	ABSOLUTE ERROR VALUES.			
-	-	H=0.1	H=0.05	H=0.025	H=0.0125
0.1	.05526	.00000	.000000	.00000	.00000
0.3	.20248	.00000	.000000	.00000	.00000
0.5	.41218	.00000	.000000	.00000	.00000
0.7	.70481	.00000	.000000	.00000	.00000
0.9	1.10682	.00000	.000000	.00000	.00000
1.0	1.35914	.00000	.000000	.00000	.00000

Observation:

The table shows all zero error values for all values of H considered. Thus this method is very accurate for the test problem (6) when compared with first method with y' present and y_2 expanded by Taylor series. Table 3.22 further confirms that expansion of y_2 by Taylor series and introduction of y_1', which is in fact an approximation of the actual definition of derivative, increase the error-values in 'Case 1'.

CHAPTER 4

COMPARISON OF METHODS AND ANALYSIS OF RESULTS

4.1 (i) Programs for System 3

The programs named AWO1A, AWO2A, AWO3A and AWO4A are put together to produce four programs named AWO8(EA), AWO8(RA), AWO9(EA) and AWO9(RA) respectively. AWO8(EA) and AWO9(EA) use the improved Euler method to start Adams-Moulton and Milne's methods, while AWO8(RA) and AWO9(RA) use the classical Runge-Kutta method to do the same thing. However, AWO9(EA) and AWO9(RA) include as part of starting values for Adams-Moulton and Milne's methods the given initial conditions.

Each program computes x - values, calculated solutions, exact solutions, first derivatives of y with respect to x and error-values, but only exact solutions and error-values are printed at p1 = (1.1 , 1.2, ..., 2.0).

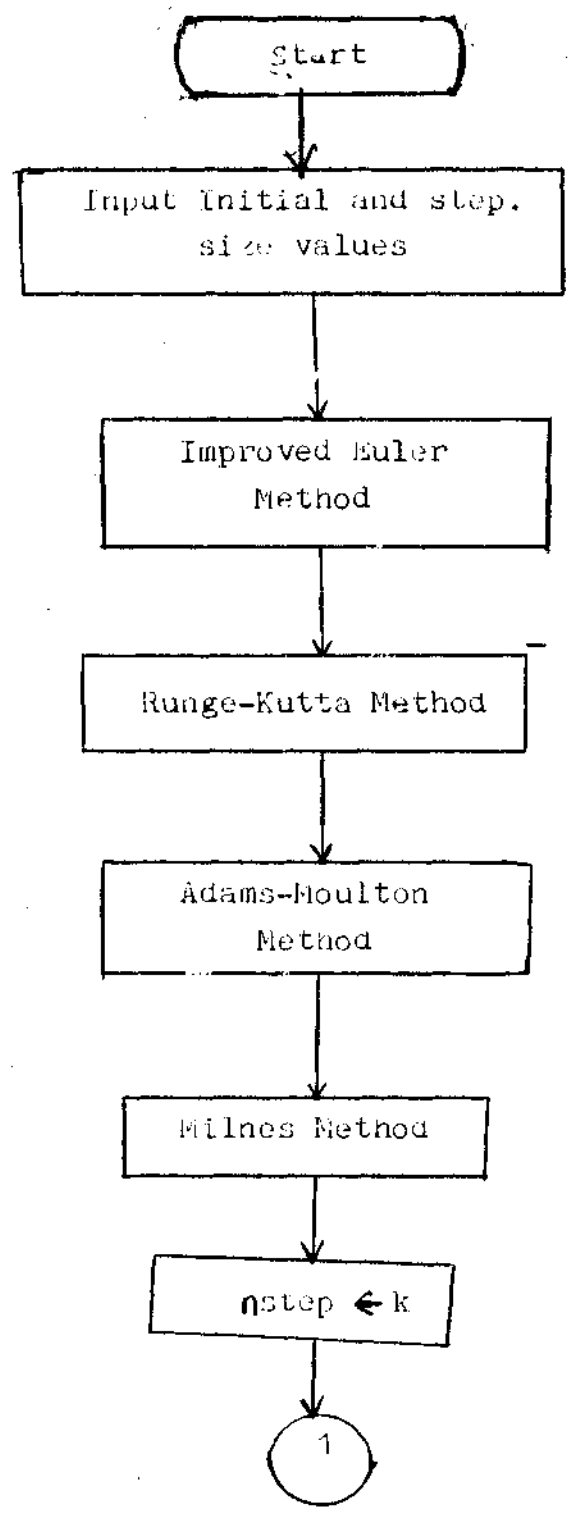
In the outputs ER1, ER2, ER3 and ER4 represent error values for Euler, Runge-Kutta, Adams-Moulton and Milne's methods respectively in that order, while Ex. represents exact solution for each method.

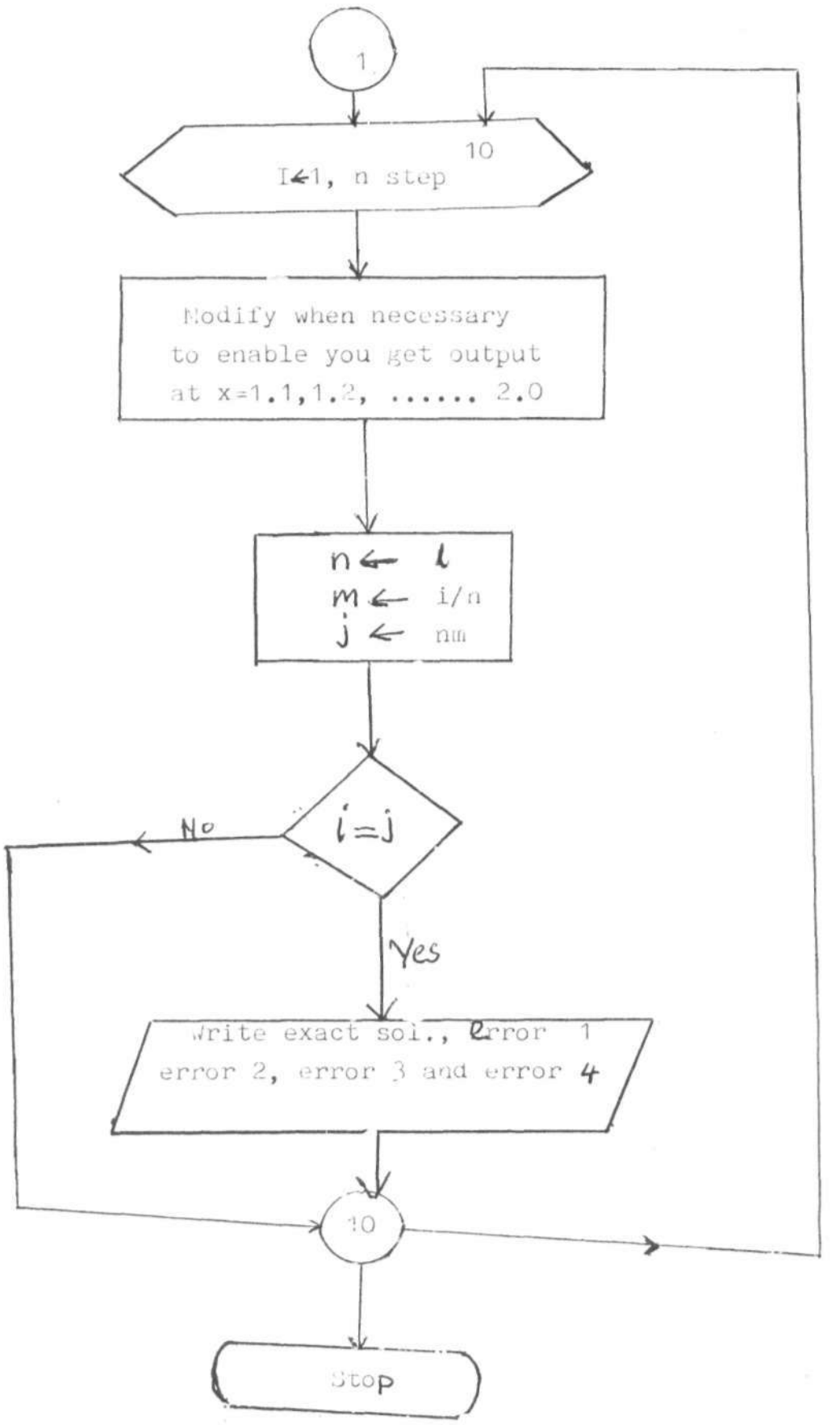
(ii) Program for System 4

Apart from the changes in the names of the programs from AW01A to AW01B, AW02A to AW02B and so forth, and the printed outputs at p2 =(0.1, 0.2, ..., 1.0), the description of system 3 still holds for system 4.

Flow-Chart IX

The flow-chart on the next page caters for system 3 and system 4. It records exact solutions and error-values of Euler, Runge-Kutta, Adams-Moulton and Milne's method for comparison at each point in R1 and R2 respectively.


Start
Input Initial and step. size values
Improved Euler Method
Runge-Kutta Method
Adams-Moulton Method
Milnes Method
nstep ← k
1


1
10
I←1, n step
Modify when necessary
to enable you get output
at x=1.1,1.2, ... 2.0
n← l
m← i/n
j ← nm
i=j
No
Yes
Write exact sol., error 1
error 2, error 3 and error 4
10
Stop

4.1.1 Outputs for System 3

Only the outputs for H=0.0125 are shown in Tables 4.1 - 4.4 at the selected values of x for easy comparison. Corresponding detailed outputs are however in AWO8(EA), AWO8(RA), AWO9(EA) and AWO9(RA) respectively under Appendix A.

Table 4.1

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ERI	ER2	ER3	ER4
1.1	3.12986	.00134	.01201	.00106	.00090
1.3	6.25741	.01352	.05508	.00515	.00433
1.5	10.74344	.04015	.12604	.01119	.00948
1.7	16.83598	.08299	.22694	.01948	.01659
1.9	24.78947	.14351	.35891	.03032	.02592
2.0	29.54517	.18081	.43670	.03678	.03150

Table 4.2

X-VAL	EX.SOL	ABSOLUTU ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
1.1	3.12986	.00134	.01201	.00676	.00636
1.3	6.25741	.01352	.05508	.01454	.01285
1.5	10.74344	.04015	.12604	.02560	.02221
1.7	16.83598	.08299	.22694	.04044	.03487
1.9	24.78947	.14351	.35891	.05956	.05127
2.0	29.54517	.18081	.43670	.07089	.06100

Observation

1. Table 4.1 and Table 4.2 on page 131 reveal a growing value of error generally in each column, though acceptable when compared with the corresponding exact solution in column 2.

2. In Table 4.1, Milne's method is seen to be the most accurate of all the methods considered. This is obvious from the least error values recorded for this method. Next to Milne's method in accuracy is the Adams-Moulton method followed by the improved Euler method and least in accuracy is the classical Runge-Kutta method.

3. The picture is a bit different in Table 4.2 where in row x=1.1, the improved Euler method with 0.00134 error value takes the lead in accuracy. After that row, we have the same picture as revealed in Table 4.1.

4. It could be observed also that starting values affect the accuracies of Adams-Moulton and Milne's methods. For when the improved Euler method, which is seen to be more accurate than Runge-Kutta method, is used to rpoduce the starting values accuracy in Adams-Moulton and Milne's methods improves greatly. This is obvious when Table 4.1 is compared with Table 4.2.

Table 4.3

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
1.1	3.12986	.00134	.01201	.00088	.00078
1.3	6.25741	.01352	.05508	.00403	.00348
1.5	10.74344	.04015	.12604	.00867	.00737
1.7	16.83598	.08299	.22694	.01504	.01266
1.9	24.78947	.14351	.35891	.02337	.01953
2.0	29.54517	.18081	.43670	.02833	.02362

Table 4.4

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
1.1	3.12986	.00134	.01201	.00522	.00380
1.3	6.25741	.01352	.05508	.01119	.00891
1.5	10.74344	.04015	.12604	.01966	.01603
1.7	16.83598	.08299	.22694	.03103	.02547
1.9	24.78947	.14351	.35891	.04567	.03756
2.0	29.54517	.18081	.43670	.05435	.04470

Observation

The features revealed in Table 4.1 and 4.2 repeated themselves in Table 4.3 and 4.4. Therefore, the same comments will be maintained for Tables 4.3 and 4.4 as far as those features are concerned. It should be noted, however, that accuracy in Table 4.3 and 4.4 respectively for Adams-Moulton and improved Euler methods where the starting values includes the given initial values is much better than accuracy in the corresponding Tables 4.1 and 4.2 respectively where the starting values do not include the given initial values.

4.1.2 Outputs for System 4

As usual we shall discuss only the outputs in Tables 4.5 - 4.8 for H=0.0125 in this section. The detailed outputs are however obtained in AWO8(EB), AWO8(RB), AWO9(EB) and AWO9(RB) in Appendix B. Also as usual AWO9(EB) and AWO9(RB) include the given initial values as part of the starting values for Adams-Moulton and the Fourth order Milne's methods while AWO8(EB) and AWO8(RB) do not include them.

Table 4.5

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
0.1	.05526	.00001	.00032	.00001	.00001
0.3	.20248	.00013	.00096	.00004	.00003
0.5	.41218	.00043	.00164	.00007	.00006
0.7	.70481	.00094	.00238	.00011	.00009
0.9	1.10682	.00173	.00322	.00015	.00012
1.0	1.35914	.00225	.00369	.00018	.00014

Table 4.6

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
0.1	.05526	.00001	.00032	.00015	.00015
0.3	.20248	.00013	.00096	.00012	.00013
0.5	.41218	.00043	.00164	.00010	.00011
0.7	.70481	.00094	.00238	.00008	.00010
0.9	1.10682	.00173	.00322	.00007	.00009
1.0	1.35914	.00225	.00369	.00006	.00008

Observation

1. Looking at Table 4.5 from AWO8(EB) for which the starting values for Adams-Moulton and Milne's methods are from the improved Euler's method, we could see that error values grow generally in columns 3,4,5 and 6 from rows x=0.1 to x=1.0. As usual the fourth order Milne's method gives the best accuracy which is obvious from the least error values recorded for the method when compared with the corresponding error values of other methods. Adams-Moulton method is still second, followed by the improved Euler method and lastly the classical Runge-Kutta method. The accuracies of the methods are also acceptable when compared with the corresponding exact solutions in column 2.

2. Table 4.6 which arises from AWO8(RB) for which the starting values for Adams-Moulton and Milne's methods are from the classical Runge-Kutta method, gives a different picture. Here we could observe in columns 5 and 6 that for Adams-Moulton and Milne's methods the error-values are decreasing from rows x=0.1 to x=1.0 More importantly, we observed that Adams-Moulton's method takes the lead for the first time in accuracy followed of course by the fourth order Milne's method and then Euler method and lastly Runge-Kutta method. But the accuracy of each method is accepted when the errors are compared with the corresponding exact solutions in column 2 of Table 4.6

Table 4.7

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
0.1	.05526	.00001	.00032	.00001	.00001
0.3	.20248	.00013	.00096	.00003	.00003
0.5	.41218	.00043	.00164	.00006	.00005
0.7	.70481	.00094	.00238	.00008	.00007
0.9	1.10682	.00173	.00322	.00012	.00010
1.0	1.35914	.00225	.00369	.00013	.00012

Table 4.8

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES FOR H=0.0125			
-	-	ER1	ER2	ER3	ER4
0.1	.05526	.00001	.00032	.00011	.00007
0.3	.20248	.00013	.00096	.00009	.00005
0.5	.41218	.00043	.00164	.00007	.00004
0.7	.70481	.00094	.00238	.00006	.00002
0.9	1.10682	.00173	.00322	.00005	.00001
1.0	1.35914	.00225	.00369	.00005	.00001

Observation

1. Table 4.7 is produced from AW09(EB) which includes the given initial values to start Adams-Moulton and Milne's methods. As a result accuracy of these methods as seen in Table 4.7 is better than accuracy for them in Table 4.5. Similarly accuracy of these methods is also better in Table 4.8 than what obtains in Table 4.6. This is obvious from the reduced error values of these methods when Tables 4.5 and 4.7 and Tables 4.6 and 4.8 are compared. Thus to guarantee a better accuracy for both methods the starting values must include the given initial values irrespective of whether the rest of the starting values are produced by the improved Euler method or the classical Runge-Kutta method.

2. Furthermore a quick look at Tables 4.7 and 4.8 reveals that accuracy of Milne's method is better than that of Adams-Moulton's method in both Tables. This is obvious from the smalle error-values in Milne's method when compared wit the corresponding error-values in Adams-Moulton' method in the Tables. Also Table 4.8 arising fr AWO9(RB), which uses the classical Runge-Kutta method to start both Adams-Moulton and Milne's methods, gives a better accurate results for these methods than what obtains in Table 4.7 for which the starting values for Adams-Moulton and Milne's methods are from the improved Euler method.

3. Accuracy for each of the methods including the improved Euler and the classical Runge-Kutta methods are acceptable when we compare the error values of each method with the corresponding exact solutions in column 2 of each Table.

4.2 Comparison of Taylor Series Expansion Method 1 with Method 2 for Systems 3 and 4

4.2.1 System 3

In this section comparison of AWO5(R4A) with AWO6(D4A), AWO5(R3A) with AWO6(D3A) and AWO5(R2A) with AWO6(D2A) will be made in that order for H=0.1, 0.05, 0.025 and 0.0125 at our point P_1.

Comparison of AWO5(R4A) with AWO6(D4A)

Table 4.9

X-VAL	EX.SOL	ABSOLUTE ERROR VALUES							
-	-	H = 0.1		H = 0.05		H = 0.025		H = 0.0125	
-	-	AWO5 (R4A)	AWO6 (D4A)	AWO5 (R4A)	AWO6 (D4A)	AWO5 (R4A)	AWO6 (D4A)	AWO5 (R4A)	AWO6 (D4A)
1.1	3.12986	1.40920	1.25553	.66783	.55922	.32495	.23719	.16027	.08251
1.3	6.25741	2.05788	1.00315	.98458	.10849	.48144	.31053	.23804	.51316
1.5	10.74344	2.82974	.13783	1.36269	1.23420	.66855	1.75188	.33111	2.00329
1.7	16.83598	3.72809	2.42821	1.80385	3.73050	.88713	4.34892	.43990	4.65016
1.9	24.78947	4.75585	6.13018	2.30954	7.64316	1.13794	8.36473	.56479	8.71699
2.0	29.54517	5.31909	8.59296	2.58700	10.21322	1.27563	10.98733	.63338	11.36558

Observation

1. Looking at columns 3 and 4 under H=0.1 of Table 4.9, error value keeps on growing from x=1.1 to x=2.0 in column 3 while in column 4 it is reducing from x=1.1 to x=1.5 and then starts to grow from x=1.7 to x=2.0. Also each value of error in column 3 is greater than its corresponding value in column 4 from x=1.1 to x x=1.7 and from x=1.9 and beyond that value of x, reverse could be seen.

   This facts suggest that with H=0.1 and with a lower region say $0.9 \leq x \leq 0.0$ with $x_o=1.0$, a better accuracy will be produced for AWO6(D4A) than for AWO5(R4A).

2. The same feature is observed in other columns under H=0.05 to H=0.0125 up to a particular value of x which varies according to the H value and then the reverse follows up to x = 2.0

3. Finally, a critical look at the table shows that Method 1 which produces AWO5(R4A) gives a better accurate results especially when $H < 0.025$ than method 2 which produces AWO6(D4A). Infact when we compare the error values of methods 1 and 2 with the exact solution in column 2 we could observe that method 2 is very unstable and thus unsuitable for the test problem (5). On the other hand method 1 is tolerable if $H < 0.025$.

Comparison of AWO5(R3A) with AWO6(D3A)

Table 4.10

X-VAL	EX.SOL	A B S O L U T E E R R O R V A L U E S							
-	-	H = 0.1		H = 0.05		H = 0.025		H = 0.0125	
-	-	AWO5 (R3A)	AWO6 (D3A)	AWO5 (R3A)	AWO6 (D3A)	AWO5 (R3A)	AWO6 (D3A)	AWO5 (R3A)	AWO6 (D3A)
1.1	3.12986	1.40904	1.25545	.66780	.55921	.32495	.23719	.16027	.08251
1.3	6.25741	2.05704	1.00289	.98441	.10844	.48140	.31054	.23803	.51316
1.5	10.74344	2.82756	.13838	1.36222	1.23431	.66844	1.75190	.33108	2.00329
1.7	16.83598	3.72379	2.42916	1.80287	3.73070	.88690	4.34897	.43984	4.65017
1.9	24.78947	4.74852	6.13164	2.30783	7.64348	1.13753	8.36480	.56469	8.71701
2.0	29.54517	5.30985	8.59472	2.58484	10.21361	1.27511	10.98742	.63325	11.36561

Comparison of AWO5(2A) with AWO6(D2A)

Table 4.11

X-VAL	EX.SOL	A B S O L U T E E R R O R V A L U E S							
-	-	H = 0.1		H = 0.05		H = 0.025		H = 0.0125	
-	-	AWO5 (R2A)	AWO6 (D2A)	AWO5 (R2A)	AWO6 (D2A)	AWO5 (R2A)	AWO6 (D2A)	AWO5 (R2A)	AWO6 (D2A)
1.1	3.12986	1.38286	1.24081	.65988	.55554	.32204	.23604	.15905	.08208
1.3	6.25741	1.92202	.95332	.92968	.09102	.45700	.31763	.22654	.51633
1.5	10.74344	2.46748	.24385	1.20290	1.27600	.59375	1.77020	.29495	2.01183
1.7	16.83598	2.98841	2.61194	1.46314	3.80745	.72381	4.38389	.35997	4.66680
1.9	24.78947	3.45327	6.41349	1.69358	7.76629	.83849	8.42188	.41716	8.74449
2.C	29.54517	3.65464	8.93435	1.79228	10.36362	.88730	11.05767	.44143	11.39956

Observation

The same features as revealed in Table 4.9 repeated themselves in Tables 4.10 and 4.11. Hence the same comments will be maintained.. We therefore conclude this section by saying that Taylor series expansion method 1 is preferred to Taylor series expansion method 2 in terms of accuracy and stability. Infact method 2 should not be used for the test and similar problems.

4.2.2 System 4

In this section we shall compare AWO5(R4B) with AWO6(D4B), AWO5(R3B) with AWO6(D3B) and AWO5(R2B) with AWO6(D2B) for H=0.1, 0.05, 0.025 and 0.0125 at points P2.

Comparison of AWO5(R4B) with AWO6(D4B)

Table 4.12

X-VAL	EX.SOL	A B S O L U T E E R R O R V A L U E S							
-	-	H=0.1		H=0.05		H=0.025		H=0.0125	
-	-	AWO5 (R4B)	AWO6 (D4B)	AWO5 (R4B)	AWO6 (D4B)	AWO5 (R4B)	AWO6 (D4B)	AWO5 (R4B)	AWO6 (D4B)
0.1	.05526	.06688	.06135	.03188	.02783	.01556	.01223	.00769	.00471
0.3	.20248	.09588	.05886	.04586	.01431	.02243	.00648	.01109	.01652
0.5	.44218	.13443	.03063	.06446	.02811	.03156	.05559	.01562	.06887
0.7	.70481	.18536	.03315	.08906	.11004	.04365	.14606	.02161	.16350
0.9	1.10682	.25224	.14579	.12138	.24578	.05954	.29269	.02949	.31541
1.0	1.35914	.29305	.22579	.14111	.33956	.06924	.39297	.03430	.41884

Observation

1. In columns 3, 5, 7 and 9 under H=0.1, 0.05, 0.025 and 0.0125, the error-values are growing from rows x=0.1 to x= .0. Also when they are compared with the corresponding exact solutions in column 2, it is discovered that the errors are generally large to be accepted for method 1. In some cases error-values are greater than the corresponding exact solutions. For example in column 3 of row x=0.1 under H=0.1 the error values .06688 is greater than the corresponding exact solution .05526.

2. Looking at columns 4, 6, 8 and 10, the same features are observed; but in addition the method 2 of the solution is unstable. This is seen from the unstable manner the error-values decreases from 0.06135 at row x=0.1 to 0.03063 at row x=0.5 and starts to grow from 0.03315 at x=0.7 to 0.22579 at x=1.0. The same features are evident in columns 6 and 8. Column 10 shows a growing value of errors in Table 4.12, but looking at the detailed outputs in AWO6(D4B) under Appendix B, the observation becomes similar to what we observe in columns 4, 6 and 8. Here also error value starts to decrease from 0.00471 at x=0.1 to 0.00259 at x=0.2 and there after starts to grow from 0.01652 to 0.41884.

3. Comparing the accuracy of method 1 with method 2, it is discovered that for H=0.1, each error-value in column 3 is greater than its corresponding value in column 4. Hence for H=0.1, Method 2 is better than Method 1. For H=0.0125, however, Method 1 is better than Method 2, although in rows x=0.1 and x=0.2 we have error values in column 10 larger than the corresponding error-values in column 9. This is evident in AWO5(R4B) and AWO6(D4B). For H=0.05 and H=0.025, it is not easy to say categorically which of the two methods is better than the other in terms of accuracy as could be seen in Table 4.12 and in AWO5(R4B) and AWO6(D4B).

For example in columns 5 and 6 under H=0.05 Method 2 shows a better accuracy from rows x=0.1 to x=0.5 while from rows x=0.7 to x=1.0, Method 1 is better than Method 2. The same feature is revealed in columns 7 and 8 under H=0.025.

Comparison of AW05(R3B) with AW06(D3B)

Table 4.13

X-VAL	EX.SOL	A B S O L U T E E R R O R V A L U E S							
-	-	H=0.1		H=0.05		H=0.025		H=0.0125	
-	-	AW05 (R3B)	AW06 (D3B)	AW05 (R3B)	AW06 (D3B)	AW05 (R3B)	AW06 (D3B)	AW05 (R3B)	AW06 (D3B)
0.1	.05526	.06683	.06132	.03187	.02783	.01556	.01223	.00769	.00471
0.3	.20248	.09562	.05876	.04580	.01429	.02241	.00649	.01109	.01652
0.5	.41218	.13375	.03041	.06431	.02816	.03153	.05560	.01561	.06887
0.7	.70481	.18397	.03357	.08874	.11013	.04358	.14608	.02159	.16350
0.9	1.10682	.24976	.14651	.12080	.24593	.05940	.29272	.02945	.31541
1.0	1.35914	.28984	.22670	.14036	.33975	.06906	.39301	.03425	.41885

Comparison of Aw05(R2B) with AW06(D2B)

Table 4.14

X-VAL	EX.SOL	A B S O L U T E E R R O R V A L U E S							
-	-	H=0.1		H=0.05		H=0.025		H=0.0125	
-	-	AW05 (R2B)	AW06 (D2B)	AW05 (R2B)	AW06 (D2B)	AW05 (R2B)	AW06 (D2B)	AW05 (R2B)	AW06 D2B)
0.1	.05526	.06554	.06054	.03148	.02763	.01542	.01217	.00763	.00469
0.3	.20248	.08940	.05596	.04331	.01331	.02131	.00688	.01057	.01670
0.5	.41218	.11773	.02402	.05733	.03065	.02828	.05669	.01405	.06938
0.7	.70481	.15156	.04553	.07404	.11508	.03659	.14832	.01819	.16456
0.9	1.10682	.19201	.16658	.09402	.25453	.04652	.29669	.02314	.31731
1.0	1.35914	.21511	.25200	.10542	.35073	.05218	.39810	.02596	.42130

Observation

1. The same features as revealed in Table 4.12 repeated themselves in Tables 4.13 and 4.14 respectively. Hence the same comments will be maintained for these observations.

2. In addition to 1, if we look through at the three Tables, however, it will be discovered that for Method 1 accuracy of the object function y(x) expanded to second order Taylor series is better than accuracy of y(x) expanded to third order and accuracy of y(x) expanded to third order is better than when it is expanded to fourth order. Hence to use method 1, it is enough to limit the expansion of y(x) to second order Taylor series for a fairly good accuracy. Furthermore, it is also discovered that as H is reducing in value for this method, accuracy improves correspondingly. This is evident from the last two sets of error outputs in AWO5(R2B) under Appendix 2.

3. The instability of the error-values of method 2 makes it difficult to say whether accuracy is better when the solution function y(x) is expanded to second, third or fourth order Taylor series. All that could be said is that apart from the instability

of method 2, error-value increases very sharply from rows x=0.7 to x=1.0 for all values of H considered and worst for H=0.0125 as revealed in Tables 4.12, 4.13 and 4.14 respectively.

4.3 Comparison of Execution Times

4.3.0 Introduction

The Execution times in seconds are obtained from Cyber 72 (Main Frame) at Iya Abubakar Computer Centre, Ahmadu Bello University, Zaria for the various methods. The C.P.U. times are not considered because they were found to be highly unreliable, for instance if you get a C.P.U. time for a particular problem as 0.400 seconds, if the problem is repeated almost immediately, the C.P.U. time will alter. The reason may be due to the fact that the system is a Multi-user.

The calculated results are not recorded because the same accuracy to 5 decimal places are maintained on Cyber 72 as obtained on TRS 80 - Micro Computer at Kaduna Polytechnic which I used principally for the project.

Consideration of execution times of Taylor Series methods is limited to the test problems reduced to systems of first-order equations since the accuracy of this method is at least comparable with Numerov's method 'Case 1' with y' computed. Even though Taylor series method 2 was used to solve the test problems directly without reducing them respectively to systems of two first order equations the method was found unstable for the two test problems (5) and (6).

4.3.1 Test problem (5): $y'' = 3y'/x - 3y/x^2 + 2x - 1$

Initial conditions: $x(1) = 2, y'(1) = 10$

Solution in the region D1 = $(x/1.1 \leq x \leq 2.0)$

Table 4.15

Methods	Execution Times in Seconds					
-	H=0.1	H=0.05	H=0.025	H=0.0125	TOTAL TIME	RELATIVE ORDER OF ACCURACY
Improved Euler Method	.054	.059	.067	.089	0.269	3rd
Runge-Kutta	.064	.066	.084	.117	0.331	4th
Adams-Moulton	.085	.105	.121	.149	0.460	2nd
Milne	.084	.100	.114	.141	0.439	1st
Taylor series order R2	.051	.059	.067	.080	0.257	5th
Taylor series order R3	.055	.059	.071	.086	0.271	6th
Taylor series order R4	.056	.062	.077	.103	0.298	7th

R2 - R4 . = Second, Third and Fourth order Taylor series method of solving the test problem (5) reduced to a system of two first order ordinary differential equations.

4.3.2 Test problem 6: $y'' = y+e^x$

Initial Conditions: $y(o) = 0, y'(o) = 0.5$

Solution in the Region D2 = $(x/0.1 \leq x \leq 1.0)$

Table 4.16

Methods	Execution Times in Seconds					
-	H=0.1	H=0.05	H=0.025	H=0.0125	TOTAL TIME	RELATIVE ORDER OF ACCURACY
Improved Euler method	.053	.058	.068	.095	0.274	4th
Runge-Kutta	.056	.067	.091	0.128	0.342	5th
Adams-Moulton	.087	.100	.127	.155	0.469	3rd
Milne	.084	.102	.114	.148	0.448	2nd
Taylor series order R2	.051	.055	.059	.077	0.242	6th
Taylor series order R3	.052	.056	.071	.092	0.271	7th
Taylor series order R4	.054	.063	.078	.113	0.308	8th
Numerov Method Case 1	.076	.098	.156	.266	0.596	9th
Numerov Method Case 2	.062	.080	.107	.190	0.439	1st

Comparison of Memory Allocations

The memory allocations for our values of H for each method are obtained on TRS 80 - Micro Computer at Kaduna Polytechnic. However, consideration of memory allocations for Taylor series Method 2 is not considered since the method is found unstable for both test problems (5) and (6).

4.4.1 Test Problem (5): See Section 4.3.1

Table 4.17

Methods	Memory Allocations in Bytes				
	H=0.1	H=0.05	H=0.025	H = 0.0125	Total Memory Allocation
Improved Euler Method	9250	9414	9414	9414	37492
Classical Runge-Kutta Method	9633	9797	9797	9797	39024
Adams-Moulton	10772	10837	10837	10837	43283
Milne	10668	10733	10733	10733	42867
Taylor Series (R2)	9365	9529	9529	9529	37952
Taylor Series (R3)	9842	10006	10006	10006	39860
Taylor Series (R4)	10371	10535	10535	10535	41976

4.4.2 Test Problem (6): See Section 4.3.2

Table 4.18

Methods	Memory Allocations in Bytes				
	H=0.1	H=0.05	H = 0.025	H = 0.0125	Total Memory Allocations
Improved Euler Method	8795	8843	8843	8843	35324
Classical Runge-Kutta Method	9104	9207	9207	9207	36725
Adams-Moulton	10258	10262	10262	10262	41044
Milne	10145	10149	10149	10149	40592
Taylor Series (R2)	8793	8890	8890	8890	35463
Taylor Series (R3)	9413	9512	9512	9512	37949
Taylor Series (R4)	9665	9764	9764	9764	38957
Numerov's Method 'Case 1'	10011	9950	9950	9950	39861
Numerov's Method 'Case 2'	9858	9797	9797	9797	39249

4.5 Discussion of Numerical Accuracy, Execution Time and Memory Allocation.

Tables 4.15 and 4.17 show the various execution times in seconds and the memory allocations in bytes for all values of H considered for each method to solve the test problem (5) reproduced above under section 4.3.1 for convenience. It is revealed in Table 4.17 that, except for H=0.1, the memory allocations for each method for all other values of H considered are respectively the same and greater than what we obtain for H=0.1. This is so because of the modification of the program for the method which enables us to obtain our outputs at $x = 1.1, 1.2, 1.3, \ldots\ldots, 2.0$. Relative orders of numerical accuracy are also shown on Table 4.15. A look at the two Tables reveals that in terms of numerical accuracy, computer execution times and memory allocations, the Fourth Order Milne's method is better than Adams-Moulton method (both of them being Predictor-Corrector methods) for our test problem and similar problems for all values of H. However, the least execution times and memory allocations are

recorded for the Improved Euler methods for all values of H when compared with the other classical methods of Runge Kutta, Adams-Moulton and Milne as revealed in both tables.

Similarly, Tables 4.16 and 4.18 for the test problem (6) also reproduced above under section 4.3.2 reveal that Numerov's method 'case 2' takes the lead in numerical accuracy. In fact absolute accuracy is recorded for this method as shown in Table 3.22 and the detailed outputs in AWO7B. Furthermore, its execution times and memory allocations are much less than those of Predictor-Corrector Methods of Adams-Moulton and Milne. Here the memory allocation for H=0.1 is larger than what they are for all other values of H considered. The reason may be that the modification of the program mentioned above is done for all values of H in this case. Clearly, Numerov's method 'case 2' is the best for this type of problem, although it is not suitable for a problem that does not allow y_{k+1} to be made a subject after $\frac{h^2}{12}F_{k+1}$, where $F_{k+1}=F(x_{k+1},y_{k+1})$ has been moved to the left of equality sign. Here again, of all the four classical methods referred to above, the Improved Euler method has the least execution times and memory allocations for our values of H as revealed also in Tables 4.16 and 4.18 respectively. Numerov's method 'Case 1' in which y'_1 is computed is the least accurate method also with the largest execution time.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

5.1.1 System 3: $y''=f(x,y,y'),y(x_o)=y_o,y'(x_o)=y_o'$

1. Of all the methods considered for this system, the Fourth Order Milne's method is found to be the most accurate method followed by Adams-Moulton, Improved Euler and Runge-Kutta methods as we could see in AWO8(EA), AWO8(RA), AW09(EA) and AWO9(RA) under Appendix A. The total execution time and memory allocation for the Milne's method are also smaller than what we obtain for Adams-Moulton's method.

2. It is shown that when the Improved Euler Method is used as starters for Adams-Moulton and the Fourth order Milne's methods,accuracy is better in both methods than when the classical Runge-Kutta method is used as starters for them. This is obvious when we compare AWO8(EA) with AWO8(RA) and AWO9(EA) with AWO9(RA).

3. Accuracy in Adams-Moulton and the Fourth order Milne's methods is still better when the given initial values are used as part of the starting values. This is obvious when AWO8(EA) is compared with AWO9(EA) and AWO8(RA) is compared with AWO9(RA).

4. It is shown that a higher degree of accuracy is obtained ror the Taylor Series method 1 if H< 0.0125. This is confirmed in the last wo sets of outputs in AWO5(R2A).

5. It is also shown that Taylor Series Method 2 is found to be highly unstable for the test problem under this system. This is obvious from its outputs in AWO6(D4A), AWO6(D3A) and AWO6(D2A).

5.1.2 System 4: $y''=f(x,y), y(x_o)= y_o, y'(x_o) = y_o'$

1. Numerov's method 'Case 2' is found to be the most accurate method for the test problem (6) and similar problems under this system with zero error values for all values of H considered as confirmed in AWO7B. Its total execution time and memory space are also the least

when compared with those of Adams-Moulton and Milne's methods.

2. In numerical accuracy, the fourth order Milne's method comes next to Numerov's method followed by Adams-Moulton's method, the Improved Euler method and the classical Runge-Kutta method in that order. This is obvious from the outputs of AWO8(EB), AWO9(EB) and AWO9(RB). Even though in AWO8(RB) Adams-Moulton method is better in accuracy than the fourth order Milne's method, we could still claim in general that the fourth order Milne's method is still preferred in terms of accuracy, execution times and Memory allocations. Of all the classical methods considered, the Improved Euler Method has the least total execution times and Memoery allocations. This is confirmed in Tables 4.15, 4.16, 4.17 and 4.18 respectively.

3. It is shown that when the given initial values are unused as part of the starting values, the improved Euler method should be used as starter for the fourth order Milne's method while the Classical Runge-Kutta method is used as starter for Adams-Moulton's method. This is obvious when the

error outputs of AWO8(EB) and AWO8(RB) are compared.

4. It is shown that when the given initial values are used as part of the starting values, while the remaining part is from the classical Runge-Kutta method, accuracies of Adams-Moulton and the fourth-order Milne's method are better. This is obvious when the error outputs in AWO9(EB) are compared with the corresponding error outputs in AWO9(RB).

5. Numerov's method 'Case 1' in which y_1 and y_2 are found by Taylor series shows a very high values of errors generally when compared with the exact solutions. The error values are also seen to be increasing with the decreasing values of H. Thus the method is not suitable for the test problem.

6. Error-values in Taylor series method 1 are also high when compared with the exact solution in column 2. Better accuracy is however obtained when $H < 0.0125$ as could be seen from the last two sets of error outputs in AWO5(R2B) under Appendix B.

7. It is discovered that Taylor series method 2 is found to be highly unstable for this system as could be seen from the error outputs in AWO6(D4B), AWO6(D3B) and AWO6(D2B). Further investigation of the method for $H < 0.0125$ will be a worthwhile exercise since error-values for $H=0.0125$ starts to be stable from row $x=0.2$ in the detailed outputs of the programs.

8. In Tables 4.9, 4.10, 4.11, 4.15 and 4.17 for the test problem (5) under system 3 using the Taylor series method 1, it is discovered that the numerical accuracy, total execution time and memory allocation of the object function y(x) expanded to second order is the best followed by y(x) expanded to third order and lastly y(x) expanded to fourth order. The same features are obvious in Tables 4.12, 4.13, 4.14, 4.16 and 4.18 respectively for the test problem (6) still using the Taylor series method 1. This observation is likely to be due to accumulation of round off error of Taylor series from the second order up to the fourth order expansion of y(x).

5.2 Recommendation

1. It is observed from Tables 4.15, 4.16, 4.17 and 4.18 for our test problem (5) and (6) that of all the classical methods of Runge-Kutta, Adams-Moulton, Milne and Numerov 'case 2' considered, the least execution times and memory allocations for our values of H are recorded for the improved Euler method. The numerical accuracy of the method is also encouraging when compared with the accuracies of the other classical methods. I will therefore recommend that improved Euler method be used for similar problems. However, if the desired accuracy cannot be achieved by the improved Euler method, the Numerov's method 'case 2' where applicable or the fourth order Milne's method should be used.

2. For research purposes, however, where numerical accuracy is very important, it is recommended that Numerov's method 'case 2' be used. Otherwise the fourth order Milne's method should be considered since it is more accurate, and the execution times and memory

allocations are much less than what we obtain for Adams-Moulton's method for the test problems.

3. Although the classical Runge-Kutta method considered does not produce as accurate results as improved Euler method for instance, it is recommended to be used as part of the starters under system 4 for the Multi-step methods, while the other part of the starting values is the set of the given initial values.

4. Even though Taylor's series method 1 and method 2 are not recommended to be used ror the test problems and similar problems because of inaccuracy and instability, further investigation of their behaviours on other systems of second order ordinary differential equations is necessary especially when the step-size $H < 0.0125$.

5. Table 4.16 also reveals Numerov's method 'Case 1' as very bad in accuracy and also with the highest execution time. Hence the method is not recommended for the test problem and similar problems. It may however, be used for teaching purposes for comparison with 'Case 2'.

REFERENCES

1. Bader, G and Deuflhard, P.
A semi-implicit mid-point rule for stiff systems of ordinary differential equations, Numer.Math., 41(1983), , pp. 373 - 398

2. Bochner, S
The role of Mathematics in the rise of Science, Princeton, New Jarsey, 1966, pp. 298 - 300.

3. Boyce, E.W. and DiPrima, C.R.
Boundary Value Problems, John Wiley & Sons, New York, 1977, pp1 - 8, 24, 359, 51-83.

4. Brauer, R., and Nohel, J.A.
Ordinary Differential Equations: A first Course, Second Edition, Menlo Park, California, 1973, p.360.

5. Bulirsch, R and Stoer, J.
Numerical treatment of Ordinary differential equations by extrapolation methods, Numer. Math., 8(1966), pp. 1 - 13.

6. Carnahan, B., Luther, H.A. and Wilkes J.O.
Applied Numerical Methods John Wiley & Sons, New York, Chichester, Brisbane, Toronto, 1969 pp. 341 - 345, 407, 384 - 392.

7. Chopra, M.G. and Kumar Ram.
Fortran IV Programming, Vikas Publishing. House PVT Ltd., New Delhi, 1980. pp. 11-16.

8. Conte, S.D. and deBoor, C.
Elementary Numerical Analysis, 2nd ed., Mc Graur-Hill Kogakusha, Ltd. New Delhi, London, Mexico, 1972, pp. 320 - 321, 353.

9. Daniel, J.W. & Moore, R.E.
Computation and Theory in ordinary Differential Equations, W.H. Freeman & Company San Francisco, 1970, pp. 29 - 32.

10. Dahlquist, G
A special stability problem for linear multistep methods, BIT, 3(1963), pp.27-43.

11. Dahlquist, G and Lindberg, B
On some implicit one-step methods for stiff differential equations, Roy.Inst. Techn. Stockholm: Tech. Rep. TRITA-NA-7302, Stockholm, 1973.

12. Deuflhard, G
Order and stepsize control in extrapolation methods, Numer.Math., 41(1983), pp.399-422.

13. Deuflhard, P
On Recent Progress in Extrapolation methods for Ordinary Differential Equations, SIAM REVIEW, Dec. 1985, pp. 505-533.

14. Dodes, I.A.,
Numerical Analysis for Computer Science, North-Holland, New York, Oxford,1978, pp. 514-518, 552-558, 564-566, 574-577.

15. Dorn, W.S. and McCracken, D.D.
Numerical Methods with Fortran IV case studies, New York, Wiley, 1972, pp. 376 - 393.

16. Edwards, C.H.
The Historical Development of Calculus, Springer-Verlag, New York, Berlin, 1970. pp. 189 - 267.

17. Eve, H
An Introduction to the History of Mathematics, Fourth edition, New York, Holt, 1976 pp. 327 - 330.

18. Froberg, C.E.
Introduction to Numerical Analysis,
Addison-Wesley Publishing Co.,
Reading etc. 1966, pp. 251 - 252.

19. Forsythe, G.E. Malcolm, M.A. Moler, C.B.
Computer Methods for Mathematical
Computations, 1977, pp.110, 126-127

20. Gear, C.W.
Numerical Initial Value Problems in
Ordinary Differential Equations,
Prentice-Hall, Englewood Cliffs NJ, 1971.

21. Gellert, W., Kustner, H., Hellwich,
M and Kastner, H.
The VNR Concise Encyclopedia of Mathematics,
Van Nostrand Reinhold Company, 1977, pp.500.

22. Gladwell, I and Sayers, D.K.
Computational Techniques for Ordinary
Differential Equations - Proceeding of
the Conference held at the University
of Manchester on Dec. 18-20, 1978,pp.1-6

23. Gragg, W.B.
Repeated extrapolation to the limit in the
numerical solution of ordinary differential
equations, Thesis, Univ. California, Los
Angeles, 1963.

24. Gragg, W.B.
On extrapolation algorithms for ordinary initial value problems, SIAM J, Numer. Anal., 2 (1965), pp. 384 - 404.

25. Kline, M.
Mathematical Thought from Ancient to Modern Times, Oxford University Press, New York, 1972. pp. 500, 479-480, 528-534, 717-721.

26. Maki, D.P. and Thomson M.
Mathematical Models and Applications, Euglewood Cliff, New Jersey, 1973, pp. 1-4.

27. Organick, E.I. and Meissner, L.P.,
Fortran IV, 2nd Ed., Addison-Wesley Publishing Company, Reading, 1974, pp 4-8.

28. Olayi, G.A.
The construction and use of Mathematical model. A paper presented at the workshop on Mathematical Modelling at the University of Benin on Jan. 26-27, 1986, p.1.

29. Ralston, A and Hebert S. Wilf,
Mathematical Methods for Digital Computers, New Yorkm Wiley 1960, vol. 1 ch.9.

30. Redsh, K.A.
An Introduction to Computational Methods.
The English University Press Ltd., 102
Newgate Street London E.C.1., 1960,
pp. 114 - 115.

31. Scheid, F.
Theory and Problems of Numerical Analysis,
Schaum's Outline series. Mc Graw -Hill,
New York, 1968, pp. 223-226.

32. Scott, J.F.
History of Mathematics, Taylor and Francis
Ltd., London, 1969, pp. 153 - 161.

33. Scraton R.E.
"Estimation of the Truncation Error in
Runge-Kutta and Allied Processes"
Computer Journal (October,1964)
pp. 246 - 248.

34. Shampine, L.F. Watts, H.A. and Davenport, S.M.
Solving nonstiff ordinary differential
equations - The state of the art, this
Review, 18(1976), pp.376-411.

35. Spiegel, M.R.
Theory and Problems of Advanced calculus,
Schaum Publishing Co., New York, 1963, p.57.

```
00020    C        ********** APPENDIX A **********
00040    C
00100    C        DATE:14/08/1987




00200    C        NAME OF FILE: AW01A
00300    C        NUMERICAL SOLUTION OF THE SECOND ORDER DIFF. EQUAT.
00400    C        Y''=3Y'/X-3Y/X.X+2Y-1 SUBJECT TO THE INITIAL CONDI-
00500    C        TIONS XM=1,YM=2,YM'=ZM=10.
00600    C        THE IMPROVED EULER METHOD IS USED AFTER THE ORIGINAL
00700    C        EQUATION HAS BEEN REDUCED TO A SYSTEM OF TWO FIRST
00800    C        EQUATIONS
00900            IMPLICIT REAL*8(A-H,O-Z)
01000    C       DEFINE STATEMENT FUNCTIONS FOR EXACT AND CAL.SOLUTIONS.
01100           FSOL(X)=3.*X**3-2.*X+X*X*(1.+X*DLOG(X))
01200           FUN(X,Y,Z)=3.*Z/X-3.*Y/(X*X)+2.*X-1.
01300    C       INITIALIZE X,Y,Z,H
01400           XM=1.0
01500           YM=2.0
01600           ZM=10.0
01700           H=0.1
01800           WRITE(2,100)XM,YM,ZM,H
01900      100  FORMAT(10X,2HXM,F5.0,3X,2HYM,F5.0,3X,2HZM,F5.0,4X,1HH,F8.4/)
02000    C       WRITE COLUMN HEADINGS
02100           WRITE(2,101)
02200      101  FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
02300    C       START ITERATION
02400           NSTEP=10
02500           DO 10 I=1,NSTEP
02600           SLOPE1=FUN(XM,YM,ZM)
02700           XM=XM+H
02800           AA=H*SLOPE1
02900           YY=YM+AA
03000           ZZ=ZM+AA
03100           SLOPE2=FUN(XM,YY,ZZ)
03200           YM=YM+(H/2.)*(ZM+ZZ)
03300           ZM=ZM+(H/2.)*(SLOPE1+SLOPE2)
03400           EXACT=FSOL(XM)
03500           ERROR=EXACT-YM
03600           ERR=DABS(ERROR)
03700           WRITE(2,102)XM,YM,EXACT,ZM,ERR
03800      102  FORMAT(4X,5F13.5)
03900       10  CONTINUE
04000           STOP
04100           END
```

XM 1. YM 2. ZM 10. H .1000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.12500	3.12986	12.45669	.00486
1.20000	4.50779	4.53905	15.18664	.03126
1.30000	6.17633	6.25741	18.19743	.08108
1.40000	8.15923	8.31528	21.49587	.15605
1.50000	10.48568	10.74344	25.08812	.25776
1.60000	13.18547	13.57313	28.97978	.38766
1.70000	16.28888	16.83598	33.17599	.54710
1.80000	19.82666	20.56397	37.68149	.73731
1.90000	23.83003	24.78947	42.50068	.95944
2.00000	28.33061	29.54517	47.63764	1.21456

XM 1. YM 2. ZM 10. H .0500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.12589	3.12986	12.54007	.00397
1.20000	4.51953	4.53905	15.36403	.01953
1.30000	6.20969	6.25741	18.47839	.04772
1.40000	8.22578	8.31528	21.88903	.08950
1.50000	10.59771	10.74344	25.60130	.14573
1.60000	13.35594	13.57313	29.62009	.21720
1.70000	16.53136	16.83598	33.94994	.30462
1.80000	20.15531	20.56397	38.59503	.40866
1.90000	24.25954	24.78947	43.55929	.52994
2.00000	28.87615	29.54517	48.84640	.66902

XM 1. YM 2. ZM 10. H .0250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.12743	3.12986	12.58982	.00243
1.20000	4.52816	4.53905	15.46926	.01089
1.30000	6.23148	6.25741	18.64423	.02593
1.40000	8.26722	8.31528	22.12007	.04805
1.50000	10.66574	10.74344	25.90171	.07771
1.60000	13.45783	13.57313	29.99367	.11530
1.70000	16.67477	16.83598	34.40016	.16121
1.80000	20.34819	20.56397	39.12509	.21578
1.90000	24.51015	24.78947	44.17215	.27933
2.00000	29.19302	29.54517	49.54480	.35215

XM 1. YM 2. ZM 10. H .0125

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.12852	3.12986	12.61704	.00134
1.20000	4.53330	4.53905	15.52669	.00575
1.30000	6.24389	6.25741	18.73452	.01352
1.40000	8.29036	8.31528	22.24561	.02492
1.50000	10.70329	10.74344	26.06465	.04015
1.60000	13.51368	13.57313	30.19598	.05945
1.70000	16.75298	16.83598	34.64365	.08299
1.80000	20.45301	20.56397	39.41142	.11096
1.90000	24.64596	24.78947	44.50286	.14351
2.00000	29.36437	29.54517	49.92132	.18081

```
00100    C        DATE:16/08/1987
00200    C        NAME OF FILE: AW02A
00300    C        NUMERICAL SOLUTION OF SEC. ORDER DIFF. EQUATION
00400    C        Y''=3Y'/X-3Y/X.X+2X-1 WITH INITIAL CONDITIONS
00500    C        X0=1.0, Y0=2.0, P0=Y0'=10.0
00600    C        THE CLASSICAL FOURTH ORDER RUNGE KUTTA METHOD IS USED
00700             IMPLICIT REAL*8(A-H,K,L,O-Z)
00800    C        DEFINE STATEMENT FUNCTIONS FOR EXACT & CALC. SOLUTIONS
00900             FSOL(X)=3.*X**3-2.*X+X*X*(1.+X*DLOG(X))
01000             FUN(X,Y,P)=3.*P/X-3.*Y/(X*X)+2.*X-1.
01100    C        INITIALIZE X,Y,P,H
01200             X0=1.0

01300             Y0=2.0
01400             P0=10.0
01500             H=0.1
01600    C        PRINT INITIAL VALUES
01700             WRITE(2,100)X0,Y0,P0,H
01800       100   FORMAT(10X,'X0',F5.0,3X,'Y0',F5.0,3X,'P0',F6.1,4X,'H',F8.4/)
01900    C        PRINT COLUMN HEADINGS
02000             WRITE(2,101)
02100       101   FORMAT(10X,5HX-VAL,8X,6HEX.SOL,7X,5HC.SOL,8X,4HC.Y',12X,3HERR/)
02200    C        START ITERATION
02300             NSTEP=10
02400             DO 10 I=1,NSTEP
02500             L11=H*P0
02600             K11=H*FUN(X0,Y0,P0)
02700             XN=X0+0.5*H
02800             YN=Y0+0.5*K11
02900             PN=P0+0.5*L11
03000             L12=H*PN
03100             K12=H*FUN(XN,YN,PN)
03200             YN=Y0+0.5*K12
03300             PN=P0+0.5*L12
03400             L13=H*PN
03500             K13=H*FUN(XN,YN,PN)
03600             XN=XN+0.5*H
03700             YN=Y0+K13
03800             PN=P0+L13
03900             L14=H*PN
04000             K14=H*FUN(XN,YN,PN)
04100             L1=(L11+L14+2.*(L12+L13))/6.0
04
```

```
  900      102   FORMAT(4X,5F13.5)
05000    C       INTERCHANGE LOCATIONS
05100      103   XO=XN
05200            YO=YN1
05300            PO=PN1
05400       10   CONTINUE
05500            STOP
05600            END
```

XO 1. YO 2. PO 10.0 H .1000

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
1.10000	3.12986	3.05171	12.26119	.07815
1.20000	4.53905	4.34123	14.77723	.19782
1.30000	6.25741	5.89536	17.55738	.36205
1.40000	8.31528	7.74189	20.61025	.57339
1.50000	10.74344	9.90948	23.94394	.83396
1.60000	13.57313	12.42769	27.56602	1.14545
1.70000	16.83598	15.32683	31.48364	1.50915
1.80000	20.56397	18.63799	35.70357	1.92598
1.90000	24.78947	22.39296	40.23224	2.39651
2.00000	29.54517	26.62422	45.07575	2.92095

XO 1. YO 2. PO 10.0 H .0500

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
1.10000	3.12986	3.08606	12.43284	.04379
1.20000	4.53905	4.42855	15.13886	.11050
1.30000	6.25741	6.05567	18.12572	.20175
1.40000	8.31528	7.99640	21.40049	.31888
1.50000	10.74344	10.28045	24.96969	.46300
1.60000	13.57313	12.93815	28.83942	.63498
1.70000	16.83598	16.00048	33.01535	.83549
1.80000	20.56397	19.49897	37.50283	1.06500
1.90000	24.78947	23.46570	42.30689	1.32377
2.00000	29.54517	27.93326	47.43230	1.61191

XO 1. YO 2. PO 10.0 H .0250

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
1.10000	3.12986	3.10659	12.53338	.02327
1.20000	4.53905	4.48043	15.35057	.05863
1.30000	6.25741	6.15051	18.45818	.10690
1.40000	8.31528	8.14649	21.86222	.16879
1.50000	10.74344	10.49859	25.56828	.24486
1.60000	13.57313	13.23759	29.58151	.33555
1.70000	16.83598	16.39478	33.90677	.44119
1.80000	20.56397	20.00194	38.54856	.56203
1.90000	24.78947	24.09127	43.51116	.69820
2.00000	29.54517	28.69542	48.79859	.84975

XO 1. YO 2. PO 10.0 H .0125

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
1.10000	3.12986	3.11785	12.58804	.01201
1.20000	4.53905	4.50882	15.46567	.03023
1.30000	6.25741	6.20233	18.63883	.05508
1.40000	8.31528	8.22835	22.11296	.08693
1.50000	10.74344	10.61740	25.89311	.12604
1.60000	13.57313	13.40048	29.98394	.17266
1.70000	16.83598	16.60904	34.38982	.22694
1.80000	20.56397	20.27497	39.11485	.28900
1.90000	24.78947	24.43056	44.16287	.35891
2.00000	29.54517	29.10847	49.53754	.43670

```
00100    C        DATE: 16/08/1987
00200    C        NAME OF FILE: AW03A
00300    C        NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
00400    C        Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT TO INITIAL CONDITIONS
00500    C        XO=1.0,YO=2.0,PO=Y'=10.0 USING ADAMS MOULTON METHOD.
00600    C        THE CLASSICAL RUNGE KUTTA METHOD IS USED TO OBTAIN
00700    C        THE STARTING VALUES.
00800             IMPLICIT REAL*8(A-H,O-Z)
00900             DIMENSION A(5,4)
01000    C        DEFINE STATEMENT FUNCTIONS FOR EXACT & CAL. SOLUTIONS
01100             FSOL(X)=3.*X**3-2.*X+X*X*(1.+X*DLOG(X))
01200             FUN(X,Y,P)=3.*P/X-3.*Y/(X*X)+2.*X-1.
01300             H=0.1
01400             WRITE(2,50)H
01500       50    FORMAT(10X,2HH=,F8.4//)
01600    C        PRINT COLUMN HEADINGS
01700             WRITE(2,100)
01800      100    FORMAT(10X,1HX,12X,1HY,11X,4HY'=P,9X,5HY''=F/)
01900             DO 10 I=1,4
02000             READ(1,101)(A(I,L),L=1,4)
02100      101    FORMAT(4F7.5)
02200             WRITE(2,102)(A(I,L),L=1,4)
02300      102    FORMAT(4X,4F13.5)
02400       10    CONTINUE
02500    C        WRITE COLUMN HEADINGS
02600             WRITE(2,103)
02700      103    FORMAT(/)
02800             WRITE(2,104)
02900      104    FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
03000    C        START ITERATION
03100             NSTEP=10
03200             N=4
03300             XN=A(N,1)
03400             DO 20 J=4,NSTEP
03500             P1=55.*A(N,3)-59.*A(N-1,3)
03600             P2=37.*A(N-2,3)-9.*A(N-3,3)
03700             PRY=A(N,2)+(H/24.)*(P1+P2)
03800             P3=55.*A(N,4)-59.*A(N-1,4)
03900             P4=37.*A(N-2,4)-9.*A(N-3,4)
04000             PRP=A(N,3)+(H/24.)*(P3+P4)
04100             A(N+1,3)=PRP
04200             C1=9.*A(N+1,3)+19.*A(N,3)
04300             C2=A(N-2,3)-5.*A(N-1,3)
04400             CRY=A(N,2)+(H/24.)*(C1+C2)
04500    C        MODIFY SOLUTION BY ''MOP-UP'' TECHNIQUE
04
```

```
04600             YY=CRY+(19./270.)*(PRY-CRY)
04700             XN=XN+H
04800             A(N+1,4)=FUN(XN,CRY,PRP)
04900             C3=9.*A(N+1,4)+19.*A(N,4)
05000             C4=A(N-2,4)-5.*A(N-1,4)
05100             CRP=A(N,3)+(H/24.)*(C3+C4)
05200             PP=CRP+(19./270.)*(PRP-CRP)
05300             EXACT=FSOL(XN)
05400             ERROR=EXACT-YY
05500             ERR=DABS(ERROR)
05600             WRITE(2,105)XN,YY,EXACT,PP,ERR
05700       105   FORMAT(4X,5F13.5)
05800       C     INTERCHANGE LOCATIONS
05900             A(N,2)=YY
06000             DO 30 I=1,3
06100             A(I,3)=A(I+1,3)
06200             A(I,4)=A(I+1,4)
06300        30   CONTINUE
06400             A(N,3)=PP
06500             A(N,4)=FUN(XN,YY,PP)
06600        20   CONTINUE
06700             STOP
06800             END
```

H= .1000

X	Y	Y'=P	Y''=F
1.00000	2.00000	10.00000	25.00000
1.10000	3.05171	12.26119	27.07339
1.20000	4.34123	14.77723	29.29885
1.30000	5.89536	17.55738	31.65189

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.40000	7.80960	8.31528	20.85872	.50567
1.50000	10.07647	10.74344	24.46669	.66698
1.60000	12.71462	13.57313	28.36343	.85852
1.70000	15.75819	16.83598	32.56216	1.07778
1.80000	19.23706	20.56397	37.06535	1.32691
1.90000	23.18155	24.78947	41.87603	1.60792
2.00000	27.62263	29.54518	46.99751	1.92255

H= .0500

X	Y	Y'=P	Y''=F
1.00000	2.00000	10.00000	25.00000
1.05000	2.51271	11.18278	26.21350
1.10000	3.08606	12.43284	27.45636
1.15000	3.72351	13.75121	28.72620

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.20000	4.44688	4.53905	15.22265	.09217
1.30000	6.12545	6.25741	18.39102	.13196
1.40000	8.13539	8.31528	21.85835	.17989
1.50000	10.50725	10.74344	25.62987	.23619
1.60000	13.27164	13.57313	29.70969	.30149
1.70000	16.45960	16.83598	34.10171	.37638
1.80000	20.10251	20.56397	38.80955	.46146
1.90000	24.23215	24.78947	43.83665	.55733
2.00000	28.88059	29.54517	49.18625	.66458

H= .0250

X	Y	Y'=P	Y''=F
1.00000	2.00000	10.00000	25.00000
1.02500	2.25315	10.60712	25.66149
1.05000	2.52167	11.23165	26.32873
1.07500	2.80600	11.87370	27.00152

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.11126	3.12986	12.55767	.01860
1.20000	4.51067	4.53905	15.47909	.02838
1.30000	6.21709	6.25741	18.69954	.04032
1.40000	8.26071	8.31528	22.22393	.05457
1.50000	10.67216	10.74344	26.05672	.07129
1.60000	13.48248	13.57313	30.20204	.09066
1.70000	16.72311	16.83598	34.66378	.11286
1.80000	20.42590	20.56397	39.44557	.13807
1.90000	24.62301	24.78947	44.55084	.16646
2.00000	29.34696	29.54517	49.98284	.19822

H= .0125

X	Y	Y'=P	Y''=F
1.00000	2.00000	10.00000	25.00000
1.01250	2.12578	10.30789	25.34602
1.02500	2.25544	10.62021	25.69325
1.03750	2.38903	10.93698	26.04166

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.12464	3.12986	12.62177	.00522
1.20000	4.53114	4.53905	15.55788	.00791
1.30000	6.24623	6.25741	18.79441	.01119
1.40000	8.30019	8.31528	22.33617	.01509
1.50000	10.72378	10.74344	26.18761	.01966
1.60000	13.54818	13.57313	30.35287	.02496
1.70000	16.80495	16.83598	34.83583	.03103
1.80000	20.52605	20.56397	39.64013	.03792
1.90000	24.74380	24.78947	44.76919	.04568
2.00000	29.49082	29.54517	50.22628	.05435

```
00100   C       DATE: 16/08/1987
00200   C       NAME OF FILE: AW04A
00300   C       NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
00400   C       Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT TO INITIAL CONDITIONS
00500   C       XO=1.0,ZO=2.0,PO=ZO'=10.0 USING THE 4TH ORDER MILNE'S
00600   C       METHOD. THE CLASSICAL RUNGE KUTTA METHOD IS USED TO
00700   C       OBTAIN THE STARTING VALUES.
00800           IMPLICIT REAL*8(A-H,O-Z)
00900           DIMENSION X(4,1),A(5,4)
01000   C       DEFINE STATEMENT FUNCTIONS FOR EXACT & CAL. SOLUTIONS
01100           FSOL(X)=3.*X**3-2.*X+X*X*(1.+X*DLOG(X))
01200           FUN(X,Z,P)=3.*P/X-3.*Z/(X*X)+2.*X-1.0
01300   C       INITIALIZE X,Z,P,H
01400           XO=1.0
01500           H=0.1
01600           WRITE(2,50)H
01700        50 FORMAT(10X,2HH=,F8.4//)
01800   C       PRINT COLUMN HEADINGS
01900           WRITE(2,100)
02000       100 FORMAT(10X,1HX,12X,1HZ,11X,4HZ'=P,10X,5HZ''=F/)
02100   C       INPUT STARTING VALUES
02200           DO 10 I=1,4
02300           READ(1,101)X(I,1),(A(I,J),J=1,3)
02400       101 FORMAT(4F7.5)
02500           WRITE(2,102)X(I,1),(A(I,J),J=1,3)
02600       102 FORMAT(4X,4F13.5)
02700        10 CONTINUE
02800           WRITE(2,103)
02900       103 FORMAT(/)
03000   C       WRITE COLUMN HEADINGS
03100           WRITE(2,104)
03200       104 FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Z',11X,3HERR/)
03300   C       START ITERATION
03400           NSTEP=10
03500           N=4
03600           XN=XO+3.*H
03700           DO 20 J=4,NSTEP
03800           ZM=A(N-3,1)
03900           PM=A(N-3,2)
04000           PRZ=ZM+(4.*H/3.)*(2.*A(N,2)-A(N-1,2)+2.*A(N-2,2))
04100           PRP=PM+(4.*H/3.)*(2.*A(N,3)-A(N-1,3)+2.*A(N-2,3))
04200           XN=XN+H
```

```
04300            A(N+1,3)=FUN(XN,PRZ,PRP)
04400            A(N+1,1)=A(N-1,1)+(H/3.)*(PRP+4.*A(N,2)+A(N-1,2))
04500            A(N+1,2)=A(N-1,2)+(H/3.)*(A(N+1,3)+4.*A(N,3)+A(N-1,3))
04600    C       MODIFY SOLUTIONS BY "MOP-UP" TECHNIQUE
04700            ZZ=A(N+1,1)+(1./29.)*(PRZ-A(N+1,1))
04800            PP=A(N+1,2)+(1./29.)*(PRP-A(N+1,2))
04900            EXACT=FSOL(XN)
05000            ERROR=EXACT-ZZ
05100            ERR=DABS(ERROR)
05200            WRITE(2,105)XN,ZZ,EXACT,PP,ERR
05300      105   FORMAT(4X,5F13.5)
05400    C       INTERCHANGE LOCATIONS
05500            DO 30 M=1,5
05600            A(M,1)=A(M+1,1)
05700            A(M,2)=A(M+1,2)
05800            A(M,3)=A(M+1,3)
05900       30   CONTINUE
06000            A(N+1,1)=ZZ
06100            A(N+1,2)=PP
06200            A(N+1,3)=FUN(XN,ZZ,PP)
06300       20   CONTINUE
06400            STOP
06500            END
```

H= .1000

X	Z	Z'=P	Z''=F
1.00000	2.00000	10.00000	25.00000
1.10000	3.05171	12.26119	27.07339
1.20000	4.34123	14.77723	29.29885
1.30000	5.89536	17.55738	31.65189

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
1.40000	7.90271	8.31528	21.20048	.41257
1.50000	10.15992	10.74344	24.77443	.58353
1.60000	12.86552	13.57313	28.99765	.70761
1.70000	15.94769	16.83598	33.03418	.88828
1.80000	19.50028	20.56397	37.84460	1.06369
1.90000	23.51662	24.78947	42.55623	1.27285
2.00000	28.03080	29.54517	47.95938	1.51437

H= .0500

X	Z	Z'=P	Z''=F
1.00000	2.00000	10.00000	25.00000
1.05000	2.51271	11.18278	26.21350
1.10000	3.08606	12.43284	27.45636
1.15000	3.72351	13.75121	28.72620

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
1.20000	4.46921	4.53905	15.32446	.06984
1.30000	6.15279	6.25741	18.53409	.10463
1.40000	8.17297	8.31528	22.01955	.14231
1.50000	10.55691	10.74344	25.81113	.18653
1.60000	13.33586	13.57313	29.91368	.23728
1.70000	16.54088	16.83598	34.32953	.29510
1.80000	20.20354	20.56397	39.06299	.36043
1.90000	24.35575	24.78947	44.11733	.43372
2.00000	29.02975	29.54517	49.49579	.51542

H= .0250

X	Z	Z'=P	Z''=F
1.00000	2.00000	10.00000	25.00000
1.02500	2.25315	10.60712	25.66149
1.05000	2.52167	11.23165	26.32873
1.07500	2.80600	11.87370	27.00152

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
1.10000	3.11661	3.12986	12.58538	.01325
1.20000	4.51724	4.53905	15.51349	.02181
1.30000	6.22582	6.25741	18.73818	.03159
1.40000	8.27215	8.31528	22.26735	.04313
1.50000	10.68688	10.74344	26.10536	.05657
1.60000	13.50109	13.57313	30.25636	.07204
1.70000	16.74628	16.83598	34.72420	.08969
1.80000	20.45431	20.56397	39.51253	.10966
1.90000	24.65740	24.78947	44.62477	.13207
2.00000	29.38809	29.54517	50.06416	.15708

H= .0125

X	Z	Z'=P	Z''=F
1.00000	2.00000	10.00000	25.00000
1.01250	2.12578	10.30789	25.34602
1.02500	2.25544	10.62021	25.69325
1.03750	2.38903	10.93698	26.04166

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
1.10000	3.12600	3.12986	12.62959	.00385
1.20000	4.53292	4.53905	15.56658	.00613
1.30000	6.24855	6.25741	18.80413	.00886
1.40000	8.30319	8.31528	22.34702	.01209
1.50000	10.72759	10.74344	26.19971	.01585
1.60000	13.55294	13.57313	30.36633	.02019
1.70000	16.81084	16.83598	34.85075	.02514
1.80000	20.53323	20.56397	39.65663	.03074
1.90000	24.75244	24.78947	44.78737	.03703
2.00000	29.50112	29.54517	50.24624	.04405

```
00100    C        DATE:16/08/1987
00200    C        NAME OF FILE: AW05(R4A)
00300    C        NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
00400    C        Y''=3Y'/X-3Y/X.X+2X-1 USING TAYLOR SERIES METHOD 1
00500    C        SUBJECT TO INITIAL CONDITIONS XO=1.0,YO=2.0,PO=YO'=10.0
00600    C        THE OBJECT FUNCTION Y(X) IS EXPANDED TO FOURTH ORDER.
00700            IMPLICIT REAL*8(A-H,O-Z)
00800    C        INPUT INITIAL VALUES
00900            XO=1.0
01000            YO=2.0
01100            PO=10.0
01150            H=0.1
01200            WRITE(2,100)XO,YO,PO,H
01300      100   FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F9.5/)
01400    C        PRINT COLUMN HEADINGS
01500            WRITE(2,101)
01600      101   FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
01700    C        CALCULATE THE STARTING VALUES
01800            YOP2=3.*PO/XO-3.*YO/(XO*XO)+2.*XO-1.
01900            YOP3=3.*YOP2/XO-6.*PO/(XO*XO)+6.*YO/XO**3+2.
02000            YOP4=3.*YOP3/XO-9.*YOP2/(XO*XO)+18.*PO/(XO**3)-18.*YO/(XO**4)
02100            Y1=YO+H*PO+(H*H/2.)*YOP2+((H**3)/6.)*YOP3+((H**4)/24.)*YOP4
02200            POP1=YOP2
02300            POP2=YOP3
02400            POP3=YOP4
02500            P=PO+H*POP1+(H*H/2.)*POP2+((H**3)/6.)*POP3
02600    C        START ITERATION
02700            NSTEP=10
02800            DO 10 I=1,NSTEP
02900            X1=XO+H
03000            YP2=3.*P/X1-3.*Y1/(X1*X1)+2.*X1-1.
03100            YP3=3.*YP2/X1-6.*P/(X1*X1)+6.*Y1/(X1**3)+2.
03200            YP4=3.*YP3/X1-9.*YP2/(X1*X1)+18.*P/(X1**3)-18.*Y1/(X1**4)
03300            Y2=Y1+H*P+(H*H/2.)*YP2+((H**3)/6.)*YP3+((H**4)/24.)*YP4
03400            PP1=YP2
03500            PP2=YP3
03600            PP3=YP4
03700            P1=P+H*PP1+(H*H/2.)*PP2+((H**3)/6.)*PP3
03800            EXACT=3.*X1**3-2.*X1+X1*X1*(1.+X1*DLOG(X1))
03900            ERROR=EXACT-Y2
04000            ERR=DABS(ERROR)
04100            WRITE(2,102)X1,Y2,EXACT,P1,ERR
04200      102   FORMAT(4X,5F13.5)
04300    C        INTERCHANGE LOCATIONS
04400            XO=X1
04500            Y1=Y2
04600            P=P1
04700       10   CONTINUE
04800            STOP
04900            END
```

XO 1. YO 2. PO 10.0 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	4.53906	3.12986	15.58768	1.40920
1.20000	6.25742	4.53905	18.83027	1.71837
1.30000	8.31530	6.25741	22.37857	2.05788
1.40000	10.74348	8.31528	26.23703	2.42820
1.50000	13.57318	10.74344	30.40981	2.82974
1.60000	16.83604	13.57313	34.90077	3.26291
1.70000	20.56406	16.83598	39.71355	3.72809
1.80000	24.78959	20.56397	44.85158	4.22562
1.90000	29.54533	24.78947	50.31812	4.75585
2.00000	34.86427	29.54517	56.11623	5.31909

XO 1. YO 2. PO 10.0 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.79769	3.12986	14.07951	.66783
1.20000	5.35770	4.53905	17.17099	.81865
1.30000	7.24200	6.25741	20.56583	.98458
1.40000	9.48114	8.31528	24.26865	1.16586
1.50000	12.10614	10.74344	28.28374	1.36269
1.60000	15.14843	13.57313	32.61511	1.57529
1.70000	18.63982	16.83598	37.26650	1.80385
1.80000	22.61251	20.56397	42.24145	2.04854
1.90000	27.09901	24.78947	47.54330	2.30954
2.00000	32.13217	29.54517	53.17521	2.58700

XO 1. YO 2. PO 10.0 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.45481	3.12986	13.35346	.32495
1.20000	4.93848	4.53905	16.36986	.39943
1.30000	6.73886	6.25741	19.68842	.48144
1.40000	8.88639	8.31528	23.31382	.57111
1.50000	11.41200	10.74344	27.25045	.66855
1.60000	14.34699	13.57313	31.50238	.77386
1.70000	17.72311	16.83598	36.07343	.88713
1.80000	21.57243	20.56397	40.96716	1.00846
1.90000	25.92741	24.78947	46.18698	1.13794
2.00000	30.82081	29.54517	51.73608	1.27563

XO 1. YO 2. PO 10.0 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.29012	3.12986	12.99740	.16027
1.20000	4.73632	4.53905	15.97639	.19727
1.30000	6.49545	6.25741	19.25691	.23804
1.40000	8.59791	8.31528	22.84371	.28263
1.50000	11.07455	10.74344	26.74121	.33111
1.60000	13.95665	13.57313	30.95351	.38351
1.70000	17.27588	16.83598	35.48447	.43990
1.80000	21.06428	20.56397	40.33768	.50031
1.90000	25.35427	24.78947	45.51656	.56479
2.00000	30.17856	29.54517	51.02433	.63338

```
00100    C        DATE:16/08/1987
00200    C        NAME OF FILE: AW05(R3A)
00300    C        NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
00400    C        Y''=3Y'/X-3Y/X.X+2X-1 USING TAYLOR SERIES METHOD 1
00500    C        SUBJECT TO INITIAL CONDITIONS XO=1.0,YO=2.0,PO=YO'=10.0
00600    C        THE OBJECT FUNCTION Y(X) IS EXPANDED TO THIRD  ORDER.
00700             IMPLICIT REAL*8(A-H,O-Z)
00800    C        INPUT INITIAL VALUES
00900             XO=1.0
01000             YO=2.0
01100             PO=10.0
01200             H=0.1
01300             WRITE(2,100)XO,YO,PO,H
01400       100   FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F9.5/)
01500    C        PRINT COLUMN HEADINGS
01600             WRITE(2,101)
01700       101   FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
01800    C        CALCULATE THE STARTING VALUES
01900             YOP2=3.*PO/XO-3.*YO/(XO*XO)+2.*XO-1.
02000             YOP3=3.*YOP2/XO-6.*PO/(XO*XO)+6.*YO/XO**3+2.
02100             Y1=YO+H*PO+(H*H/2.)*YOP2+((H**3)/6.)*YOP3
02200             POP1=YOP2
02300             POP2=YOP3
02400             P=PO+H*POP1+(H*H/2.)*POP2
02500    C        START ITERATION
02600             NSTEP=10
02700             DO 10 I=1,NSTEP
02800             X1=XO+H
02900             YP2=3.*P/X1-3.*Y1/(X1*X1)+2.*X1-1.
03000             YP3=3.*YP2/X1-6.*P/(X1*X1)+6.*Y1/(X1**3)+2.
03100             Y2=Y1+H*P+(H*H/2.)*YP2+((H**3)/6.)*YP3
03200             PP1=YP2
03300             PP2=YP3
03400             P1=P+H*PP1+(H*H/2.)*PP2
03500             EXACT=3.*X1**3-2.*X1+X1*X1*(1.+X1*DLOG(X1))
03600             ERROR=EXACT-Y2
03700             ERR=DABS(ERROR)
03800             WRITE(2,102)X1,Y2,EXACT,P1,ERR
03900       102   FORMAT(4X,5F13.5)
04000    C        INTERCHANGE LOCATIONS
04100             XO=X1
04200             Y1=Y2
04300             P=P1
04400        10   CONTINUE
04500             STOP
04600             END
```

XO 1. YO 2. PO 10.0 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	4.53889	3.12986	15.58549	1.40904
1.20000	6.25700	4.53905	18.82671	1.71794
1.30000	8.31446	6.25741	22.37347	2.05704
1.40000	10.74206	8.31528	26.23022	2.42678
1.50000	13.57101	10.74344	30.40112	2.82756
1.60000	16.83291	13.57313	34.89003	3.25978
1.70000	20.55976	16.83598	39.70061	3.72379
1.80000	24.78390	20.56397	44.83628	4.21993
1.90000	29.53799	24.78947	50.30030	4.74852
2.00000	34.85503	29.54517	56.09574	5.30985

XO 1. YO 2. PO 10.0 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.79766	3.12986	14.07910	.66780
1.20000	5.35762	4.53905	17.17025	.81857
1.30000	7.24183	6.25741	20.56471	.98441
1.40000	9.48083	8.31528	24.26711	1.16556
1.50000	12.10566	10.74344	28.28174	1.36222
1.60000	15.14773	13.57313	32.61259	1.57459
1.70000	18.63885	16.83598	37.26344	1.80287
1.80000	22.61120	20.56397	42.23781	2.04723
1.90000	27.09731	24.78947	47.53903	2.30783
2.00000	32.13001	29.54517	53.17027	2.58484

XO 1. YO 2. PO 10.0 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.45481	3.12986	13.35337	.32495
1.20000	4.93846	4.53905	16.36970	.39941
1.30000	6.73882	6.25741	19.68815	.48140
1.40000	8.88632	8.31528	23.31346	.57104
1.50000	11.41188	10.74344	27.24997	.66844
1.60000	14.34683	13.57313	31.50178	.77369
1.70000	17.72288	16.83598	36.07268	.88690
1.80000	21.57212	20.56397	40.96627	1.00815
1.90000	25.92700	24.78947	46.18594	1.13753
2.00000	30.82028	29.54517	51.73487	1.27511

XO 1. YO 2. PO 10.0 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.29012	3.12986	12.99738	.16027
1.20000	4.73632	4.53905	15.97635	.19727
1.30000	6.49544	6.25741	19.25684	.23803
1.40000	8.59790	8.31528	22.84362	.28262
1.50000	11.07453	10.74344	26.74109	.33108
1.60000	13.95661	13.57313	30.95337	.38347
1.70000	17.27582	16.83598	35.48428	.43984
1.80000	21.06421	20.56397	40.33746	.50024
1.90000	25.35417	24.78947	45.51630	.56469
2.00000	30.17843	29.54517	51.02403	.63325

```
00100    C        DATE:16/08/1987
00200    C        NAME OF FILE: AW05(R2A)
00300    C        NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
00400    C        Y''=3Y'/X-3Y/X.X+2X-1 USING TAYLOR SERIES METHOD 1
00500    C        SUBJECT TO INITIAL CONDITIONS XO=1.0,YO=2.0,PO=YO'=10.0
00600    C        THE OBJECT FUNCTION Y(X) IS EXPANDED TO SECOND ORDER.
00700             IMPLICIT REAL*8(A-H,O-Z)
00800    C        INPUT INITIAL VALUES
00900             XO=1.0
01000             YO=2.0
01100             PO=10.0
01200             H=0.1
01300             WRITE(2,100)XO,YO,PO,H
01400       100   FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F9.6/)
01500    C        PRINT COLUMN HEADINGS
01600             WRITE(2,101)
01700       101   FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
01800    C        CALCULATE THE STARTING VALUES
01900             YOP2=3.*PO/XO-3.*YO/(XO*XO)+2.*XO-1.
02100             Y1=YO+H*PO+(H*H/2.)*YOP2
02200             POP1=YOP2
02400             P=PO+H*POP1
02500    C        START ITERATION
02600             NSTEP=10
02700             DO 10 I=1,NSTEP
02800             X1=XO+H
02900             YP2=3.*P/X1-3.*Y1/(X1*X1)+2.*X1-1.
03100             Y2=Y1+H*P+(H*H/2.)*YP2
03200             PP1=YP2
03400             P1=P+H*PP1

03500             EXACT=3.*X1**3-2.*X1+X1*X1*(1.+X1*DLOG(X1))
03600             ERROR=EXACT-Y2
03700             ERR=DABS(ERROR)
03800             WRITE(2,102)X1,Y2,EXACT,P1,ERR
03900       102   FORMAT(4X,5F13.5)
04000    C        INTERCHANGE LOCATIONS
04100             XO=X1
04200             Y1=Y2
04300             P=P1
04400        10   CONTINUE
04500             STOP
04600             END
```

XO 1. YO 2. PO 10.0 H .100000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	4.51271	3.12986	15.25430	1.38286
1.20000	6.18882	4.53905	18.26772	1.64976
1.30000	8.17944	6.25741	21.54474	1.92202
1.40000	10.51115	8.31528	25.08952	2.19587
1.50000	13.21092	10.74344	28.90594	2.46748
1.60000	16.30610	13.57313	32.99764	2.73297
1.70000	19.82439	16.83598	37.36808	2.98841
1.80000	23.79382	20.56397	42.02051	3.22985
1.90000	28.24274	24.78947	46.95800	3.45327
2.00000	33.19982	29.54517	52.18349	3.65464

XO 1. YO 2. PO 10.0 H .050000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.78974	3.12986	13.95393	.65988
1.20000	5.33240	4.53905	16.93409	.79334
1.30000	7.18709	6.25741	20.19528	.92968
1.40000	9.38216	8.31528	23.74197	1.06688
1.50000	11.94635	10.74344	27.57829	1.20290
1.60000	14.90882	13.57313	31.70813	1.33569
1.70000	18.29912	16.83598	36.13512	1.46314
1.80000	22.14712	20.56397	40.86271	1.58315
1.90000	26.48305	24.78947	45.89414	1.69358
2.00000	31.33746	29.54517	51.23249	1.79228

XO 1. YO 2. PO 10.0 H .025000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.45190	3.12986	13.30100	.32204
1.20000	4.92783	4.53905	16.26263	.38878
1.30000	6.71442	6.25741	19.51489	.45700
1.40000	8.84096	8.31528	23.06241	.52568
1.50000	11.33720	10.74344	26.90949	.59375
1.60000	14.23329	13.57313	31.06013	.66016
1.70000	17.55979	16.83598	35.51808	.72381
1.80000	21.34759	20.56397	40.28687	.78362
1.90000	25.62796	24.78947	45.36985	.83849
2.00000	30.43247	29.54517	50.77017	.88730

XO 1. YO 2. PO 10.0 H .012500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.28890	3.12986	12.97376	.15905
1.20000	4.73147	4.53905	15.92559	.19242
1.30000	6.48396	6.25741	19.17310	.22654
1.40000	8.57618	8.31528	22.72101	.26090
1.50000	11.03840	10.74344	26.57369	.29495
1.60000	13.90129	13.57313	30.73521	.32816
1.70000	17.19594	16.83598	35.20938	.35997
1.80000	20.95379	20.56397	39.99979	.38982
1.90000	25.20664	24.78947	45.10982	.41716
2.00000	29.98660	29.54517	50.54267	.44143

XO 1. YO 2. PO 10.0 H .006250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.20889	3.12986	12.80994	.07903
1.20000	4.63477	4.53905	15.75673	.09571
1.30000	6.37020	6.25741	19.00179	.11278
1.40000	8.44524	8.31528	22.54988	.12997
1.50000	10.89044	10.74344	26.40542	.14700
1.60000	13.73673	13.57313	30.57250	.16360
1.70000	17.01547	16.83598	35.05498	.17950
1.80000	20.75838	20.56397	39.85646	.19441
1.90000	24.99753	24.78947	44.98035	.20806
2.00000	29.76533	29.54517	50.42988	.22016

XO 1. YO 2. PO 10.0 H .003125

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.16925	3.12986	12.72798	.03939
1.20000	4.58679	4.53905	15.67221	.04773
1.30000	6.31368	6.25741	18.91603	.05627
1.40000	8.38014	8.31528	22.46421	.06486
1.50000	10.81682	10.74344	26.32119	.07338
1.60000	13.65481	13.57313	30.49109	.08168
1.70000	16.92560	16.83598	34.97777	.08963
1.80000	20.66105	20.56397	39.78485	.09708
1.90000	24.89337	24.78947	44.91577	.10390
2.00000	29.65511	29.54517	50.37374	.10994

```
00100    C        DATE: 17/08/1987
00200    C        NAME OF FILE: AW06(D4A)
00300    C        NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
00400    C        Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT TO INITIAL CONDITIONS
00500    C        X0=1.0,Y0=2.0,P0=Y0'=10.0 USING THE TAYLOR SERIES
00600    C        METHOD DIRECTLY.
00700             IMPLICIT REAL*8(A-H,O-Z)
00800    C        INPUT INITIAL VALUES
00900             X0=1.0
01000             Y0=2.0
01100             P0=10.0
01200             H=0.1
01300             WRITE(2,100)X0,Y0,P0,H
01400      100    FORMAT(10X,'X0',F5.0,3X,'Y0',F5.0,3X,'P0',F5.1,4X,'H',F9.5/)
01500    C        PRINT COLUMN HEADINGS
01600             WRITE(2,101)
01700      101    FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
01800    C        CALCULATE THE STARTING VALUES
01900             Y0P2=3.*P0/X0-3.*Y0/(X0*X0)+2.*X0-1.
02000             Y0P3=3.*Y0P2/X0-6.*P0/(X0*X0)+6.*Y0/(X0**3)+2.
02100             Y0P4=3.*Y0P3/X0-9.*Y0P2/(X0*X0)+18.*P0/(X0**3)-18.*Y0/(X0**4)
02200             Y1=Y0+H*P0+(H*H/2.)*Y0P2+((H**3)/6.)*Y0P3+((H**4)/24.)*Y0P4
02300    C        START ITERATION
02400             NSTEP=10
02500             DO 10 I=1,NSTEP
02600             P=(Y1-Y0)/H
02700             X1=X0+H
02800             YP2=3.*P/X1-3.*Y1/(X1*X1)+2.*X1-1.
02900             YP3=3.*YP2/X1-6.*P/(X1*X1)+6.*Y1/(X1**3)+2.
03000             YP4=3.*YP3/X1-9.*YP2/(X1*X1)+18.*P/(X1**3)-18.*Y1/(X1**4)
03100             Y2=2.*Y1-Y0+(H*H/2.)*YP2+((H**3)/6.)*YP3+((H**4)/24.)*YP4
03200             EXACT=3.*X1**3-2.*X1+X1*X1*(1.+X1*DLOG(X1))
03300             ERROR=EXACT-Y2
03400             ERR=DABS(ERROR)
03500             WRITE(2,102)X1,Y2,EXACT,P,ERR
03600      102    FORMAT(4X,5F13.5)
03700    C        INTERCHANGE LOCATIONS
03800             X0=X1
03900             Y0=Y1
04000             Y1=Y2
04100       10    CONTINUE
04200             STOP
04300             END
```

XO 1. YO 2. PO 10.0 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	4.38538	3.12986	11.29858	1.25553
1.20000	5.76319	4.53905	12.55525	1.22414
1.30000	7.26057	6.25741	13.77812	1.00315
1.40000	8.87529	8.31528	14.97371	.56001
1.50000	10.60562	10.74344	16.14728	.13783
1.60000	12.45014	13.57313	17.30323	1.12300
1.70000	14.40776	16.83598	18.44520	2.42821
1.80000	16.47768	20.56397	19.57629	4.08629
1.90000	18.65930	24.78947	20.69918	6.13018
2.00000	20.95222	29.54517	21.81614	8.59296

XO 1. YO 2. PO 10.0 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.68908	3.12986	11.26367	.55922
1.20000	4.96806	4.53905	12.48936	.42901
1.30000	6.36590	6.25741	13.68442	.10849
1.40000	7.88025	8.31528	14.85479	.43503
1.50000	9.50924	10.74344	16.00532	1.23420
1.60000	11.25137	13.57313	17.14004	2.32176
1.70000	13.10548	16.83598	18.26236	3.73050
1.80000	15.07067	20.56397	19.37516	5.49330
1.90000	17.14631	24.78947	20.48093	7.64316
2.00000	19.33196	29.54517	21.58181	10.21322

XO 1. YO 2. PO 10.0 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.36705	3.12986	11.24677	.23719
1.20000	4.59771	4.53905	12.45741	.05866
1.30000	5.94688	6.25741	13.63895	.31053
1.40000	7.41215	8.31528	14.79704	.90313
1.50000	8.99157	10.74344	15.93632	1.75188
1.60000	10.68360	13.57313	17.06068	2.88953
1.70000	12.48705	16.83598	18.17339	4.34892
1.80000	14.40099	20.56397	19.27723	6.16298
1.90000	16.42474	24.78947	20.37461	8.36473
2.00000	18.55785	29.54517	21.46761	10.98733

XO 1. YO 2. PO 10.0 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.21237	3.12986	11.23845	.08251
1.20000	4.41914	4.53905	12.44169	.11991
1.30000	5.74425	6.25741	13.61656	.51316
1.40000	7.18525	8.31528	14.76858	1.13003
1.50000	8.74016	10.74344	15.90231	2.00329
1.60000	10.40742	13.57313	17.02155	3.16571
1.70000	12.18581	16.83598	18.12950	4.65016
1.80000	14.07440	20.56397	19.22891	6.48957
1.90000	16.07248	24.78947	20.32214	8.71699
2.00000	18.17959	29.54517	21.41123	11.36558

```
      C         DATE: 17/08/1987
      C         NAME OF FILE: AW06(D3A)
      C         NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
      C         Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT TO INITIAL CONDITIONS
      C         XO=1.0,YO=2.0,PO=YO'=10.0 USING THE TAYLOR SERIES
      C         METHOD DIRECTLY.
      C         Y1 AND Y2 ARE EXPANDED TO THIRD ORDER RESPECTIVELY.
                IMPLICIT REAL*8(A-H,O-Z)
      C         INPUT INITIAL VALUES
                XO=1.0
                YO=2.0
                PO=10.0
                H=0.1
                WRITE(2,100)XO,YO,PO,H
          100   FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F9.5/)
      C         PRINT COLUMN HEADINGS
                WRITE(2,101)
          101   FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
      C         CALCULATE THE STARTING VALUES
                YOP2=3.*PO/XO-3.*YO/(XO*XO)+2.*XO-1.
                YOP3=3.*YOP2/XO-6.*PO/(XO*XO)+6.*YO/(XO**3)+2.
                Y1=YO+H*PO+(H*H/2.)*YOP2+((H**3)/6.)*YOP3
      C         START ITERATION
                NSTEP=10
                DO 10 I=1,NSTEP
                P=(Y1-YO)/H
                X1=XO+H
                YP2=3.*P/X1-3.*Y1/(X1*X1)+2.*X1-1.
                YP3=3.*YP2/X1-6.*P/(X1*X1)+6.*Y1/(X1**3)+2.
                Y2=2.*Y1-YO+(H*H/2.)*YP2+((H**3)/6.)*YP3
                EXACT=3.*X1**3-2.*X1+X1*X1*(1.+X1*DLOG(X1))
                ERROR=EXACT-Y2
                ERR=DABS(ERROR)
                WRITE(2,102)X1,Y2,EXACT,P,ERR
          102   FORMAT(4X,5F13.5)
      C         INTERCHANGE LOCATIONS
                XO=X1
                YO=Y1
                Y1=Y2
           10   CONTINUE
                STOP
                END
```

XO 1. YO 2. PO 10.0 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	4.38531	3.12986	11.29833	1.25545
1.20000	5.76304	4.53905	12.55474	1.22399
1.30000	7.26031	6.25741	13.77735	1.00289
1.40000	8.87490	8.31528	14.97266	.55962
1.50000	10.60507	10.74344	16.14597	.13838
1.60000	12.44940	13.57313	17.30163	1.12373
1.70000	14.40681	16.83598	18.44333	2.42916
1.80000	16.47649	20.56397	19.57415	4.08748
1.90000	18.65783	24.78947	20.69676	6.13164
2.00000	20.95046	29.54517	21.81345	8.59472

XO 1. YO 2. PO 10.0 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.68907	3.12986	11.26361	.55921
1.20000	4.96804	4.53905	12.48923	.42899
1.30000	6.36585	6.25741	13.68422	.10844
1.40000	7.88018	8.31528	14.85452	.43510
1.50000	9.50913	10.74344	16.00499	1.23431
1.60000	11.25122	13.57313	17.13964	2.32191
1.70000	13.10528	16.83598	18.26189	3.73070
1.80000	15.07042	20.56397	19.37462	5.49355
1.90000	17.14600	24.78947	20.48032	7.64348
2.00000	19.33157	29.54517	21.58113	10.21361

XO 1. YO 2. PO 10.0 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.36705	3.12986	11.24675	.23719
1.20000	4.59771	4.53905	12.45738	.05866
1.30000	5.94687	6.25741	13.63890	.31054
1.40000	7.41213	8.31528	14.79697	.90315
1.50000	8.99154	10.74344	15.93624	1.75190
1.60000	10.68357	13.57313	17.06058	2.88956
1.70000	12.48701	16.83598	18.17327	4.34897
1.80000	14.40093	20.56397	19.27709	6.16304
1.90000	16.42467	24.78947	20.37445	8.36480
2.00000	18.55776	29.54517	21.46744	10.98742

XO 1. YO 2. PO 10.0 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.21237	3.12986	11.23845	.08251
1.20000	4.41914	4.53905	12.44168	.11991
1.30000	5.74425	6.25741	13.61654	.51316
1.40000	7.18524	8.31528	14.76856	1.13004
1.50000	8.74015	10.74344	15.90229	2.00329
1.60000	10.40741	13.57313	17.02152	3.16572
1.70000	12.18580	16.83598	18.12947	4.65017
1.80000	14.07438	20.56397	19.22888	6.48959
1.90000	16.07246	24.78947	20.32210	8.71701
2.00000	18.17957	29.54517	21.41119	11.36561

```
0100    C       DATE: 17/08/1987
0200    C       NAME OF FILE: AW06(D2A)
0300    C       NUMERICAL SOL. OF THE SEC. ORDER DIFF. EQUATION
0400    C       Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT TO INITIAL CONDITIONS
0500    C       XO=1.0,YO=2.0,PO=YO'=10.0 USING THE TAYLOR SERIES
0600    C       METHOD DIRECTLY.
0650    C       Y1 AND Y2 ARE EXPANDED TO SECOND ORDER RESPECTIVELY.
0700            IMPLICIT REAL*8(A-H,O-Z)
0800    C       INPUT INITIAL VALUES
0900            XO=1.0
1000            YO=2.0
1100            PO=10.0
1200            H=0.1
1300            WRITE(2,100)XO,YO,PO,H
1400      100   FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F9.5/)
1500    C       PRINT COLUMN HEADINGS
1600            WRITE(2,101)
1700      101   FORMAT(10X,5HX-VAL,8X,5HC.SOL,8X,6HEX.SOL,8X,4HC.Y',11X,3HERR/)
1800    C       CALCULATE THE STARTING VALUES
1900            YOP2=3.*PO/XO-3.*YO/(XO*XO)+2.*XO-1.
2200            Y1=YO+H*PO+(H*H/2.)*YOP2
2300    C       START ITERATION
2400            NSTEP=10
2500            DO 10 I=1,NSTEP
2600            P=(Y1-YO)/H
2            X1=XO+H
2800            YP2=3.*P/X1-3.*Y1/(X1*X1)+2.*X1-1.
3100            Y2=2.*Y1-YO+(H*H/2.)*YP2
3200            EXACT=3.*X1**3-2.*X1+X1*X1*(1.+X1*DLOG(X1))
3300            ERROR=EXACT-Y2
3400            ERR=DABS(ERROR)



3500            WRITE(2,102)X1,Y2,EXACT,P,ERR
3600      102   FORMAT(4X,5F13.5)
3700    C       INTERCHANGE LOCATIONS
3800            XO=X1
3900            YO=Y1
4000            Y1=Y2
4100       10   CONTINUE
4200            STOP
4300            END
```

XO 1. YO 2. PO 10.0 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	4.37067	3.12986	11.25000	1.24081
1.20000	5.73352	4.53905	12.45669	1.19447
1.30000	7.21073	6.25741	13.62850	.95332
1.40000	8.80003	8.31528	14.77213	.48475
1.50000	10.49960	10.74344	15.89302	.24385
1.60000	12.30798	13.57313	16.99565	1.26516
1.70000	14.22404	16.83598	18.08378	2.61194
1.80000	16.24691	20.56397	19.16059	4.31706
1.90000	18.37598	24.78947	20.22878	6.41349
2.00000	20.61083	29.54517	21.29071	8.93435

XO 1. YO 2. PO 10.0 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.68540	3.12986	11.23923	.55554
1.20000	4.95878	4.53905	12.43974	.41973
1.30000	6.34843	6.25741	13.60903	.09102
1.40000	7.85195	8.31528	14.75315	.46332
1.50000	9.46744	10.74344	15.87703	1.27600
1.60000	11.19336	13.57313	16.98476	2.37977
1.70000	13.02853	16.83598	18.07981	3.80745
1.80000	14.97202	20.56397	19.16508	5.59195
1.90000	17.02318	24.78947	20.24311	7.76629
2.00000	19.18155	29.54517	21.31607	10.36362

XO 1. YO 2. PO 10.0 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.36590	3.12986	11.23450	.23604
1.20000	4.59423	4.53905	12.43250	.05518
1.30000	5.93978	6.25741	13.60109	.31763
1.40000	7.40010	8.31528	14.74597	.91518
1.50000	8.97324	10.74344	15.87185	1.77020
1.60000	10.65764	13.57313	16.98262	2.91549
1.70000	12.45209	16.83598	18.08160	4.38389
1.80000	14.35563	20.56397	19.17158	6.20834
1.90000	16.36760	24.78947	20.25498	8.42188
2.00000	18.48751	29.54517	21.33391	11.05767

XO 1. YO 2. PO 10.0 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
1.10000	3.21194	3.12986	11.23231	.08208
1.20000	4.41766	4.53905	12.42921	.12139
1.30000	5.74108	6.25741	13.59758	.51633
1.40000	7.17972	8.31528	14.74298	1.13556
1.50000	8.73161	10.74344	15.86999	2.01183
1.60000	10.39517	13.57313	16.98241	3.17796
1.70000	12.16918	16.83598	18.08347	4.66680
1.80000	14.05268	20.56397	19.17592	6.51129
1.90000	16.04499	24.78947	20.26214	8.74449
2.00000	18.14561	29.54517	21.34417	11.39956

```
00100    C        DATE 19/06/1987
00200    C        FILE NAME: AW08(EA)
00300    C        COMPARISON OF ERROR VALUES OF THE SECOND ORDER DIFF.
00400    C        EQUATION Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT INITIAL CONDITIONS
00500    C        XO=1.0,YO=2.0,YO'=10.0.
00600    C        METHODS OF SOLUTION ARE 1. IMPROVED EULER METHOD
00700    C        2. RUNGE KUTTA METHOD 3.ADAMS-MOULTON METHO AND 4.
00800    C        THE FOURTH ORDER MILNE'S METHOD.
00900    C        STARTING VALUES DO NOT INCLUDE THE INITIAL CONDITIONS
01000    C        X=STEP-SIZE
01100    C        EX=EXACT SOLUTION
01200    C        ER1=ERROR IN IMPROVED EULER METHOD
01300    C        ER2=ERROR IN THE FOURTH ORDER RUNGE-KUTTA METHOD
01400    C        ER3=ERROR IN ADAMS-MOULTON METHOD
01500    C        ER4=ERROR IN THE FOURTH ORDER MILNE'S METHOD
01600             IMPLICIT REAL*8(A-H,K,L,O-Z)
01700             DIMENSION A(5,3),B(5,3),D(80,6)
01800    C        DEFINE STATEMENT FUNTIONS FOR EXACT AND CALCULATED SOLS.
01900             FSOL(X)=3.*X**3-2.*X+X*X*(1.+X*DLOG(X))
02000             FUN(X,Y,P)=3.*P/X-3.*Y/(X*X)+2.*X-1.
02100    C        INITIALIZE X,Y,P,H
02200             XO=1.0
02300             YO=2.0
02400             PO=10.0
02500             H=0.1
02600    C        PRINT INITIAL VALUES
02700             WRITE(2,100)XO,YO,PO,H
02800      100    FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F8.4/)



02900    C        EULER ALGORITHM FOLLOWS
03000             NSTEP=10
03100             DO 10 I=1,NSTEP
03200             SLOPE1=FUN(XO,YO,PO)
03300             XO=XO+H
03400             AA=H*SLOPE1
03500             YY=YO+AA
03600             PP=PO+AA
03700             SLOPE2=FUN(XO,YY,PP)
03800             YO=YO+(H/2.)*(PO+PP)
03900             PO=PO+(H/2.)*(SLOPE1+SLOPE2)
04000             D(I,1)=XO
04100             EXACT=FSOL(XO)
04200             D(I,2)=EXACT
04300             ERROR=EXACT-YO
04400             D(I,3)=DABS(ERROR)
04500             IF(I.GT.4) GO TO 10
04600             A(I,1)=YO
04700             A(I,2)=PO
```

```
 800            A(I,3)=FUN(XO,YO,PO)
4900            B(I,1)=A(I,1)
5000            B(I,2)=A(I,2)
5100            B(I,3)=A(I,3)
5200        10  CONTINUE
5[illegible]0   C       RESET INTIAL VALUES FOR RUNGE-KUTTA METHOD
5[illegible]0           XO=1.0
5500            YO=2.0
5600            PO=10.0
5700            DO 20 I=1,NSTEP
5800            L11=H*PO
5900            K11=H*FUN(XO,YO,PO)
6000            XN=XO+0.5*H
6100            YN=YO+0.5*K11
6200            PN=PO+0.5*L11
6300            L12=H*PN
6400            K12=H*FUN(XN,YN,PN)
6500            YN=YO+0.5*K12
6600            PN=PO+0.5*L12
6700            L13=H*PN
6800            K13=H*FUN(XN,YN,PN)
6900            XN=XN+0.5*H
7000            YN=YO+K13
7100            PN=PO+L13
7200            L14=H*PN
7300            K14=H*FUN(XN,YN,PN)
7400            L1=(L11+L14+2.*(L12+L13))/6.
7500            K1=(K11+K14+2.*(K12+K13))/6.
7600            YN1=YO+L1
7700            PN1=PO+K1
7[illegible]0           EXACT=FSOL(XN)
7[illegible]0           ERROR=EXACT-YN1
8000            D(I,4)=DABS(ERROR)
8100    C       INTERCHANGE LOCATIONS
8200            XO=XN
8300            YO=YN1
8400            PO=PN1
8500        20  CONTINUE
8600    C       ADAMS-MOULTON'S PREDICTOR-CORRECTOR ALGORITHM
8700    C       STARTING VALUES ARE FROM EULER'S ALGORITHM.
8800            XN=1.0+4.*H


8900            N=4
9000            DO 30 J=5,NSTEP
9100            P1=55.*A(N,2)-59.*A(N-1,2)
9200            P2=37.*A(N-2,2)-9.*A(N-3,2)
9300            PRY1=A(N,1)+(H/24.)*(P1+P2)
9400            P3=55.*A(N,3)-59.*A(N-1,3)
9500            P4=37.*A(N-2,3)-9.*A(N-3,3)
9[illegible]0           PRP1=A(N,2)+(H/24.)*(P3+P4)
[illegible]             A(N+1,2)=PRP1
[illegible]0            C1=9.*A(N+1,2)+19.*A(N,2)
9900            C2=A(N-2,2)-5.*A(N-1,2)
0000            CRY=A(N,1)+(H/24.)*(C1+C2)
0100    C       MODIFY SOLUTION BY MOP-UP TECHNIQUE
0200            YK=CRY+(19./270.)*(PRY1-CRY)
0
```

```
      XN=XN+H
      A(N+1,3)=FUN(XN,CRY,PRP1)
      C3=9.*A(N+1,3)+19.*A(N,3)
      C4=A(N-2,3)-5.*A(N-1,3)
      CRP1=A(N,2)+(H/24.)*(C3+C4)
C     MODIFY SOLUTION BY MOP-UP TECHNIQUE
      PP=CRP1+(19./270.)*(PRP1-CRP1)
      EXACT=FSOL(XN)
      ERROR=EXACT-YK
      D(J,5)=DABS(ERROR)
C     INTERCHANGE LOCATIONS
      A(N,1)=YK
      DO 25 M=1,3
      A(M,2)=A(M+1,2)
      A(M,3)=A(M+1,3)
   25 CONTINUE
      A(N,2)=PP
      A(N,3)=FUN(XN,YK,PP)
   30 CONTINUE
C     FOURTH ORDER MILNE'S ALGORITHM
C     STARTING VALUES ARE FROM EULER'S ALGORITHM
      XN=1.0+4.0*H
      DO 40 I=5,NSTEP
      PRZ=B(N-3,1)+(4.*H/3.)*(2.*B(N,2)-B(N-1,2)+2.*B(N-2,2))
      PRP=B(N-3,2)+(4.*H/3.)*(2.*B(N,3)-B(N-1,3)+2.*B(N-2,3))
      XN=XN+H
      B(N+1,2)=PRP
      CRZ=B(N-1,1)+(H/3.)*(B(N+1,2)+4.*B(N,2)+B(N-1,2))
      B(N+1,3)=FUN(XN,CRZ,PRP)
      CRP=B(N-1,2)+(H/3.)*(B(N+1,3)+4.*B(N,3)+B(N-1,3))
C     ADJUST SOLUTION BY MOP-UP TECHNIQUE
      ZZ=CRZ+(1./29.)*(PRZ-CRZ)
      PP=CRP+(1./29.)*(PRP-CRP)
      EXACT=FSOL(XN)
      ERROR=EXACT-ZZ
      D(I,6)=DABS(ERROR)
C     INTERCHANGE LOCATIONS
      DO 35 NI=1,3
      B(NI,1)=B(NI+1,1)
      B(NI,2)=B(NI+1,2)
      B(NI,3)=B(NI+1,3)
   35 CONTINUE
      B(N,1)=ZZ
      B(N,2)=PP
      B(N,3)=FUN(XN,ZZ,PP)
   40 CONTINUE



C     PROVIDE SPACING FOR ERROR VALUES
      WRITE(2,102)
  102 FORMAT(/)
C     WRITE COLUMN HEADINGS FOR ERROR VALUES
      WRITE(2,103)
  103 FORMAT(10X,1HX,10X,2HEX,12X,3HER1,9X,3HER2,10X,3HER3,9X,3HER4/)
C     PRINT OUT ERROR VALUES
      DO 60 I=5,NSTEP
```

```
 800      104    FORMAT(4X,6F12.5)
5900       60    CONTINUE
6000             STOP
6100             END
```

XO 1. YO 2. PO 10.0 H .1000

X	EX	ER1	ER2	ER3	ER4
1.50000	10.74344	.25776	.83396	.25451	.24400
1.60000	13.57313	.38766	1.14545	.36801	.33973
1.70000	16.83598	.54710	1.50915	.50185	.47802
1.80000	20.56397	.73731	1.92598	.65556	.60026
1.90000	24.78947	.95944	2.39651	.83045	.78343
2.00000	29.54517	1.21456	2.92095	1.02782	.93875

XO 1. YO 2. PO 10.0 H .0500

X	EX	ER1	ER2	ER3	ER4
1.30000	6.25741	.04772	.20175	.04468	.04000
1.40000	8.31528	.08950	.31888	.07543	.06672
1.50000	10.74344	.14573	.46300	.11217	.09880
1.60000	13.57313	.21720	.63498	.15539	.13674
1.70000	16.83598	.30462	.83549	.20554	.18090
1.80000	20.56397	.40866	1.06500	.26304	.23166
1.90000	24.78947	.52994	1.32377	.32832	.28939
2.00000	29.54517	.66902	1.61191	.40182	.35447

XO 1. YO 2. PO 10.0 H .0250

X	EX	ER1	ER2	ER3	ER4
1.20000	4.53905	.01089	.05863	.00883	.00763
1.30000	6.25741	.02593	.10690	.01682	.01441
1.40000	8.31528	.04805	.16879	.02655	.02273
1.50000	10.74344	.07771	.24486	.03815	.03272
1.60000	13.57313	.11530	.33555	.05176	.04447
1.70000	16.83598	.16121	.44119	.06750	.05812
1.80000	20.56397	.21578	.56203	.08553	.07377
1.90000	24.78947	.27933	.69820	.10596	.09154
2.00000	29.54517	.35215	.84975	.12894	.11154

XO 1. YO 2. PO 10.0 H .0125

X	EX	ER1	ER2	ER3	ER4
1.10000	3.12986	.00134	.01201	.00106	.00090
1.20000	4.53905	.00575	.03023	.00288	.00242
1.30000	6.25741	.01352	.05508	.00515	.00433
1.40000	8.31528	.02492	.08693	.00791	.00667
1.50000	10.74344	.04015	.12604	.01119	.00948
1.60000	13.57313	.05945	.17266	.01503	.01277
1.70000	16.83598	.08299	.22694	.01948	.01659
1.80000	20.56397	.11096	.28900	.02456	.02096
1.90000	24.78947	.14351	.35891	.03032	.02592
2.00000	29.54517	.18081	.43670	.03678	.03150

```
0100 C         DATE 19/06/1987
0200   C         FILE NAME: AW08(RA)
0300   C         COMPARISON OF ERROR VALUES OF THE SECOND ORDER DIFF.
0400   C         EQUATION Y''=3Y'/X-3Y/X.X+2X-1 SUBJECT INITIAL CONDITIONS
0500   C         XO=1.0,YO=2.0,YO'=10.0.
0600   C         METHODS OF SOLUTION ARE 1. IMPROVED EULER METHOD
0700   C         2. RUNGE KUTTA METHOD 3.ADAMS-MOULTON METHO AND 4.
0800   C         THE FOURTH ORDER MILNE'S METHOD.
0900   C         STARTING VALUES DO NOT INCLUDE THE INITIAL CONDITIONS
1000   C         X=STEP-SIZE
1100   C         EX=EXACT SOLUTION
1200   C         ER1=ERROR IN IMPROVED EULER METHOD
1300   C         ER2=ERROR IN THE FOURTH ORDER RUNGE-KUTTA METHOD
1400   C         ER3=ERROR IN ADAMS-MOULTON METHOD
1500   C         ER4=ERROR IN THE FOURTH ORDER MILNE'S METHOD
1600              IMPLICIT REAL*8(A-H,K,L,O-Z)
1700              DIMENSION A(5,3),B(5,3),D(80,6)
1800   C          DEFINE STATEMENT FUNTIONS FOR EXACT AND CALCULATED SOLS.

1900               FSOL(X)=3.*X**3-2.*X+X*X*(1.+X*DLOG(X))
2000               FUN(X,Y,P)=3.*P/X-3.*Y/(X*X)+2.*X-1.
2100   C           INITIALIZE X,Y,P,H
2200               XO=1.0
2300               YO=2.0
2400               PO=10.0
2500               H=0.1
2600   C           PRINT INITIAL VALUES
2700               WRITE(2,100)XO,YO,PO,H
2800        100    FORMAT(10X,'XO',F5.0,3X,'YO',F5.0,3X,'PO',F5.1,4X,'H',F8.4/)
2900   C           EULER ALGORITHM FOLLOWS
3000               NSTEP=10
3100               DO 10 I=1,NSTEP
3200               SLOPE1=FUN(XO,YO,PO)
3300               XO=XO+H
3400               AA=H*SLOPE1
3500               YY=YO+AA
3600               PP=PO+AA
3700               SLOPE2=FUN(XO,YY,PP)
3800               YO=YO+(H/2.)*(PO+PP)
3900               PO=PO+(H/2.)*(SLOPE1+SLOPE2)
4000               D(I,1)=XO
4100               EXACT=FSOL(XO)
4200               D(I,2)=EXACT
4300               ERROR=EXACT-YO
4400               D(I,3)=DABS(ERROR)
4500        10     CONTINUE
4600   C           RESET INTIAL VALUES FOR RUNGE-KUTTA METHOD
4700               XO=1.0
4800               YO=2.0
```

```
                PO=10.0
                DO 20 I=1,NSTEP
                L11=H*PO
                K11=H*FUN(XO,YO,PO)
                XN=XO+0.5*H
                YN=YO+0.5*K11
                PN=PO+0.5*L11
                L12=H*PN
                K12=H*FUN(XN,YN,PN)
                YN=YO+0.5*K12
                PN=PO+0.5*L12
                L13=H*PN
                K13=H*FUN(XN,YN,PN)
                XN=XN+0.5*H
                YN=YO+K13
                PN=PO+L13
                L14=H*PN
                K14=H*FUN(XN,YN,PN)
                L1=(L11+L14+2.*(L12+L13))/6.
                K1=(K11+K14+2.*(K12+K13))/6.
                YN1=YO+L1
                PN1=PO+K1
                EXACT=FSOL(XN)
                ERROR=EXACT-YN1
                D(I,4)=DABS(ERROR)
      C         INTERCHANGE LOCATIONS
                IF(I.GT.4) GO TO 15
                A(I,1)=YN1
                A(I,2)=PN1
                A(I,3)=FUN(XN,YN1,PN1)




                B(I,1)=A(I,1)
                B(I,2)=A(I,2)
                B(I,3)=A(I,3)
      C         INTERCHANGE LOCATIONS
           15   XO=XN
                YO=YN1
                PO=PN1
           20   CONTINUE
      C         ADAMS-MOULTON'S PREDICTOR-CORRECTOR ALGORITHM
      C         STARTING VALUES ARE FROM RUNGE-KUTTA METHOD
                XN=1.0+4.*H
                N=4
                DO 30 J=5,NSTEP
                P1=55.*A(N,2)-59.*A(N-1,2)
```

```
 300            P2=37.*A(N-2,2)-9.*A(N-3,2)
9400            PRY1=A(N,1)+(H/24.)*(P1+P2)
9500            P3=55.*A(N,3)-59.*A(N-1,3)
                P4=37.*A(N-2,3)-9.*A(N-3,3)
                PRP1=A(N,2)+(H/24.)*(P3+P4)
9800            A(N+1,2)=PRP1
9900            C1=9.*A(N+1,2)+19.*A(N,2)
0000            C2=A(N-2,2)-5.*A(N-1,2)
0100            CRY=A(N,1)+(H/24.)*(C1+C2)
0200    C       MODIFY SOLUTION BY MOP-UP TECHNIQUE
0300            YK=CRY+(19./270.)*(PRY1-CRY)
0400            XN=XN+H
0500            A(N+1,3)=FUN(XN,CRY,PRP1)
0600            C3=9.*A(N+1,3)+19.*A(N,3)
0700            C4=A(N-2,3)-5.*A(N-1,3)
0800            CRP1=A(N,2)+(H/24.)*(C3+C4)
0900    C       MODIFY SOLUTION BY MOP-UP TECHNIQUE
1000            PP=CRP1+(19./270.)*(PRP1-CRP1)
1100            EXACT=FSOL(XN)
1200            ERROR=EXACT-YK
1300            D(J,5)=DABS(ERROR)
1400    C       INTERCHANGE LOCATIONS
1500            A(N,1)=YK
1600            DO 25 M=1,3
1700            A(M,2)=A(M+1,2)
1800            A(M,3)=A(M+1,3)
1900       25   CONTINUE
2000            A(N,2)=PP
                A(N,3)=FUN(XN,YK,PP)
       30   CONTINUE
2300    C       FOURTH ORDER MILNE'S ALGORITHM
2400    C       STARTING VALUES ARE FROM RUNGE-KUTTA METHOD
2500            XN=1.0+4.0*H
2600            DO 40 I=5,NSTEP
2700            PRZ=B(N-3,1)+(4.*H/3.)*(2.*B(N,2)-B(N-1,2)+2.*B(N-2,2))
2800            PRP=B(N-3,2)+(4.*H/3.)*(2.*B(N,3)-B(N-1,3)+2.*B(N-2,3))
2900            XN=XN+H
3000            B(N+1,2)=PRP
3100            CRZ=B(N-1,1)+(H/3.)*(B(N+1,2)+4.*B(N,2)+B(N-1,2))
3200            B(N+1,3)=FUN(XN,CRZ,PRP)
3300            CRP=B(N-1,2)+(H/3.)*(B(N+1,3)+4.*B(N,3)+B(N-1,3))
3400    C       ADJUST SOLUTION BY MOP-UP TECHNIQUE
3500            ZZ=CRZ+(1./29.)*(PRZ-CRZ)
3600            PP=CRP+(1./29.)*(PRP-CRP)
3700            EXACT=FSOL(XN)
3800            ERROR=EXACT-ZZ




                D(I,6)=DABS(ERROR)
    C           INTERCHANGE LOCATIONS
                DO 35 NI=1,3
4200            B(NI,1)=B(NI+1,1)
4300            B(NI,2)=B(NI+1,2)
```

```
  400           B(NI,3)=B(NI+1,3)
14500       35  CONTINUE
14600           B(N,1)=ZZ
14700           B(N,2)=PP
14800           B(N,3)=FUN(XN,ZZ,PP)
14900       40  CONTINUE
15000   C       PROVIDE SPACING FOR ERROR VALUES
15100           WRITE(2,102)
15200     102   FORMAT(/)
15300   C       WRITE COLUMN HEADINGS FOR ERROR VALUES
15400           WRITE(2,103)
15500     103   FORMAT(10X,1HX,10X,2HEX,12X,3HER1,9X,3HER2,10X,3HER3,9X,3HER4/
15600   C       PRINT OUT ERROR VALUES
15700           DO 60 I=5,NSTEP
15800           WRITE (2,104)(D(I,J),J=1,6)
15900     104   FORMAT(4X,6F12.5)
16000      60   CONTINUE
16100           STOP
16200           END
```

XO 1. YO 2. PO 10.0 H .1000

X	EX	ER1	ER2	ER3	ER4
1.50000	10.74344	.25776	.83396	.76908	.68009
1.60000	13.57313	.38766	1.14545	.98982	.92177
1.70000	16.83598	.54710	1.50915	1.24711	1.12952
1.80000	20.56397	.73731	1.92598	1.53928	1.41247
1.90000	24.78947	.95944	2.39651	1.86875	1.70282
2.00000	29.54517	1.21456	2.92095	2.23782	2.03887

XO 1. YO 2. PO 10.0 H .0500

X	EX	ER1	ER2	ER3	ER4
1.30000	6.25741	.04772	.20175	.16023	.14899
1.40000	8.31528	.08950	.31888	.21957	.19996
1.50000	10.74344	.14573	.46300	.28918	.26000
1.60000	13.57313	.21720	.63498	.36996	.33007
1.70000	16.83598	.30462	.83549	.46264	.41078
1.80000	20.56397	.40866	1.06500	.56796	.50277
1.90000	24.78947	.52994	1.32377	.68668	.60670
2.00000	29.54517	.66902	1.61191	.81953	.72320

XO 1. YO 2. PO 10.0 H .0250

X	EX	ER1	ER2	ER3	ER4
1.20000	4.53905	.01089	.05863	.03588	.03309
1.30000	6.25741	.02593	.10690	.05116	.04594
1.40000	8.31528	.04805	.16879	.06938	.06140
1.50000	10.74344	.07771	.24486	.09079	.07967
1.60000	13.57313	.11530	.33555	.11560	.10094
1.70000	16.83598	.16121	.44119	.14403	.12540
1.80000	20.56397	.21578	.56203	.17633	.15324
1.90000	24.78947	.27933	.69820	.21271	.18467
2.00000	29.54517	.35215	.84975	.25340	.21987

XO 1. YO 2. PO 10.0 H .0125

X	EX	ER1	ER2	ER3	ER4
1.10000	3.12986	.00134	.01201	.00676	.00636
1.20000	4.53905	.00575	.03023	.01027	.00927
1.30000	6.25741	.01352	.05508	.01454	.01285
1.40000	8.31528	.02492	.08693	.01963	.01714
1.50000	10.74344	.04015	.12604	.02560	.02221
1.60000	13.57313	.05945	.17266	.03251	.02810
1.70000	16.83598	.08299	.22694	.04044	.03487
1.80000	20.56397	.11096	.28900	.04943	.04258
1.90000	24.78947	.14351	.35891	.05956	.05127
2.00000	29.54517	.18081	.43670	.07089	.06100

OUTPUTS OF AW09(EA) FOR H=0.1 TO H=0.0125

XO 1. YO 2. PO 10.0 H .1000

X	EX	ER1	ER2	ER3	ER4
1.40000	8.31528	.15605	.57339	.15319	.14186
1.50000	10.74344	.25776	.83396	.23632	.20269
1.60000	13.57313	.38766	1.14545	.33636	.30863
1.70000	16.83598	.54710	1.50915	.45240	.38592
1.80000	20.56397	.73731	1.92598	.58557	.52884
1.90000	24.78947	.95944	2.39651	.73704	.62893
2.00000	29.54517	1.21456	2.92095	.90779	.81144

XO 1. YO 2. PO 10.0 H .0500

X	EX	ER1	ER2	ER3	ER4
1.20000	4.53905	.01953	.11050	.01925	.01767
1.30000	6.25741	.04772	.20175	.03969	.03609
1.40000	8.31528	.08950	.31888	.06487	.05806
1.50000	10.74344	.14573	.46300	.09496	.08405
1.60000	13.57313	.21720	.63498	.13031	.11438
1.70000	16.83598	.30462	.83549	.17127	.14935
1.80000	20.56397	.40866	1.06500	.21821	.18926
1.90000	24.78947	.52994	1.32377	.27146	.23441
2.00000	29.54517	.66902	1.61191	.33139	.28511

XO 1. YO 2. PO 10.0 H .0250

X	EX	ER1	ER2	ER3	ER4
1.10000	3.12986	.00243	.02327	.00240	.00219
1.20000	4.53905	.01089	.05863	.00742	.00663
1.30000	6.25741	.02593	.10690	.01371	.01205
1.40000	8.31528	.04805	.16879	.02137	.01857
1.50000	10.74344	.07771	.24486	.03050	.02628
1.60000	13.57313	.11530	.33555	.04119	.03527
1.70000	16.83598	.16121	.44119	.05357	.04562
1.80000	20.56397	.21578	.56203	.06772	.05743
1.90000	24.78947	.27933	.69820	.08376	.07078
2.00000	29.54517	.35215	.84975	.10179	.08576

XO 1. YO 2. PO 10.0 H .0125

X	EX	ER1	ER2	ER3	ER4
1.10000	3.12986	.00134	.01201	.00088	.00078
1.20000	4.53905	.00575	.03023	.00228	.00200
1.30000	6.25741	.01352	.05508	.00403	.00348
1.40000	8.31528	.02492	.08693	.00615	.00527
1.50000	10.74344	.04015	.12604	.00867	.00737
1.60000	13.57313	.05945	.17266	.01163	.00983
1.70000	16.83598	.08299	.22694	.01504	.01266
1.80000	20.56397	.11096	.28900	.01895	.01589
1.90000	24.78947	.14351	.35891	.02337	.01953
2.00000	29.54517	.18081	.43670	.02833	.02362

OUTPUTS OF AW09(RA) FOR H=0.1 TO H=0.0125

XO 1. YO 2. PO 10.0 H .1000

X	EX	ER1	ER2	ER3	ER4
1.40000	8.31528	.15605	.57339	.50567	.41257
1.50000	10.74344	.25776	.83396	.66698	.58931
1.60000	13.57313	.38766	1.14545	.85852	.73142
1.70000	16.83598	.54710	1.50915	1.07778	.93121
1.80000	20.56397	.73731	1.92598	1.32691	1.14152
1.90000	24.78947	.95944	2.39651	1.60792	1.37562
2.00000	29.54517	1.21456	2.92095	1.92255	1.65894

XO 1. YO 2. PO 10.0 H .0500

X	EX	ER1	ER2	ER3	ER4
1.20000	4.53905	.01953	.11050	.09217	.06983
1.30000	6.25741	.04772	.20175	.13196	.10555
1.40000	8.31528	.08950	.31888	.17989	.14717
1.50000	10.74344	.14573	.46300	.23619	.19562
1.60000	13.57313	.21720	.63498	.30149	.25139
1.70000	16.83598	.30462	.83549	.37638	.31500
1.80000	20.56397	.40866	1.06500	.46146	.38698
1.90000	24.78947	.52994	1.32377	.55733	.46783
2.00000	29.54517	.66902	1.61191	.66458	.55806

XO 1. YO 2. PO 10.0 H .0250

X	EX	ER1	ER2	ER3	ER4
1.10000	3.12986	.00243	.02327	.01860	.01325
1.20000	4.53905	.01089	.05863	.02838	.02189
1.30000	6.25741	.02593	.10690	.04032	.03217
1.40000	8.31528	.04805	.16879	.05457	.04430
1.50000	10.74344	.07771	.24486	.07129	.05841
1.60000	13.57313	.11530	.33555	.09066	.07467
1.70000	16.83598	.16121	.44119	.11286	.09322
1.80000	20.56397	.21578	.56203	.13808	.11422
1.90000	24.78947	.27933	.69820	.16647	.13780
2.00000	29.54517	.35215	.84975	.19823	.16411

XO 1. YO 2. PO 10.0 H .0125

X	EX	ER1	ER2	ER3	ER4
1.10000	3.12986	.00134	.01201	.00522	.00380
1.20000	4.53905	.00575	.03023	.00792	.00612
1.30000	6.25741	.01352	.05508	.01119	.00891
1.40000	8.31528	.02492	.08693	.01509	.01220
1.50000	10.74344	.04015	.12604	.01966	.01603
1.60000	13.57313	.05945	.17266	.02496	.02044
1.70000	16.83598	.08299	.22694	.03103	.02547
1.80000	20.56397	.11096	.28900	.03792	.03116
1.90000	24.78947	.14351	.35891	.04567	.03756
2.00000	29.54517	.18081	.43670	.05435	.04470

********** APPENDIX B **********

OUTPUTS OF AW01B FOR H=0.1 TO H=0.0125

XM 0. YM 0. ZM .5 H .1000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05500	.05526	.61026	.00026
.20000	.12183	.12214	.73789	.00031
.30000	.20233	.20248	.88535	.00015
.40000	.29863	.29836	1.05543	.00026
.50000	.41312	.41218	1.25127	.00094
.60000	.54856	.54664	1.47643	.00192
.70000	.70806	.70481	1.73494	.00324
.80000	.89516	.89022	2.03132	.00494
.90000	1.11389	1.10682	2.37069	.00707
1.00000	1.36883	1.35914	2.75884	.00969

XM 0. YM 0. ZM .5 H .0500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05522	.05526	.60910	.00004
.20000	.12219	.12214	.73549	.00005
.30000	.20275	.20248	.88160	.00027
.40000	.29900	.29836	1.05019	.00063
.50000	.41334	.41218	1.24440	.00116
.60000	.54851	.54664	1.46775	.00188
.70000	.70762	.70481	1.72423	.00280
.80000	.89419	.89022	2.01835	.00397
.90000	1.11223	1.10682	2.35519	.00541
1.00000	1.36630	1.35914	2.74050	.00716

XM 0. YM 0. ZM .5 H .0250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05527	.05526	.60849	.00001
.20000	.12222	.12214	.73420	.00008
.30000	.20270	.20248	.87956	.00022
.40000	.29881	.29836	1.04733	.00045
.50000	.41294	.41218	1.24060	.00076
.60000	.54781	.54664	1.46291	.00118
.70000	.70653	.70481	1.71821	.00171
.80000	.89260	.89022	2.01099	.00238
.90000	1.11002	1.10682	2.34632	.00320
1.00000	1.36333	1.35914	2.72991	.00419

XM 0. YM 0. ZM .5 H .0125

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05527	.05526	.60817	.00001
.20000	.12219	.12214	.73353	.00005
.30000	.20261	.20248	.87850	.00013
.40000	.29862	.29836	1.04583	.00026
.50000	.41261	.41218	1.23861	.00043
.60000	.54729	.54664	1.46035	.00065
.70000	.70575	.70481	1.71501	.00094
.80000	.89151	.89022	2.00707	.00129
.90000	1.10855	1.10682	2.34158	.00173
1.00000	1.36139	1.35914	2.72423	.00225

OUTPUTS OF AW02B FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .1000

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
.10000	.05526	.05259	.61052	.00267
.20000	.12214	.11679	.73819	.00535
.30000	.20248	.19443	.88546	.00805
.40000	.29836	.28755	1.05509	.01081
.50000	.41218	.39852	1.25020	.01366
.60000	.54664	.53000	1.47433	.01663
.70000	.70481	.68506	1.73144	.01975
.80000	.89022	.86716	2.02604	.02306
.90000	1.10682	1.08024	2.36320	.02658
1.00000	1.35914	1.32878	2.74864	.03036

XO 0. YO 0. PO .5 H .0500

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
.10000	.05526	.05397	.60914	.00129
.20000	.12214	.11955	.73543	.00259
.30000	.20248	.19857	.88132	.00391
.40000	.29836	.29310	1.04954	.00527
.50000	.41218	.40551	1.24321	.00667
.60000	.54664	.53850	1.46583	.00813
.70000	.70481	.69514	1.72136	.00967
.80000	.89022	.87891	2.01430	.01131
.90000	1.10682	1.09376	2.34968	.01306
1.00000	1.35914	1.34421	2.73321	.01493

XO 0. YO 0. PO .5 H .0250

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
.10000	.05526	.05462	.60848	.00064
.20000	.12214	.12086	.73412	.00128
.30000	.20248	.20055	.87934	.00193
.40000	.29836	.29577	1.04688	.00260
.50000	.41218	.40889	1.23984	.00329
.60000	.54664	.54261	1.46172	.00402
.70000	.70481	.70002	1.71648	.00479
.80000	.89022	.88461	2.00859	.00560
.90000	1.10682	1.10035	2.34309	.00647
1.00000	1.35914	1.35174	2.72569	.00740

XO 0. YO 0. PO .5 H .0125

X-VAL	EX.SOL	C.SOL	C.Y'	ERR
.10000	.05526	.05494	.60816	.00032
.20000	.12214	.12151	.73348	.00063
.30000	.20248	.20152	.87837	.00096
.40000	.29836	.29707	1.04557	.00129
.50000	.41218	.41054	1.23818	.00164
.60000	.54664	.54464	1.45970	.00200
.70000	.70481	.70243	1.71407	.00238
.80000	.89022	.88743	2.00577	.00279
.90000	1.10682	1.10360	2.33984	.00322
1.00000	1.35914	1.35545	2.72197	.00369

H= .1000

OUTPUTS OF AW03B FOR H=0.1 TO H=0.0125

X	Y	Y'=P	Y''=F
0.00000	0.00000	.50000	1.00000
.10000	.05259	.61052	1.15776
.20000	.11679	.73819	1.33820
.30000	.19443	.88546	1.54429

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.40000	.29113	.29836	1.05151	.00724
.50000	.40561	.41218	1.24312	.00657
.60000	.54069	.54664	1.46364	.00594
.70000	.69944	.70481	1.71707	.00538
.80000	.88535	.89022	2.00785	.00486
.90000	1.10242	1.10682	2.34103	.00440
1.00000	1.35516	1.35914	2.72227	.00398

H= .0500

X	Y	Y'=P	Y''=F
0.00000	0.00000	.50000	1.00000
.05000	.02564	.55256	1.07691
.10000	.05397	.60914	1.15914
.15000	.08520	.67000	1.24703

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.20000	.12030	.12214	.73469	.00184
.30000	.20081	.20248	.87908	.00166
.40000	.29686	.29836	1.04579	.00150
.50000	.41082	.41218	1.23791	.00136
.60000	.54541	.54664	1.45894	.00123
.70000	.70370	.70481	1.71281	.00111
.80000	.88921	.89022	2.00401	.00100
.90000	1.10591	1.10682	2.33755	.00091
1.00000	1.35832	1.35914	2.71912	.00082

H= .0250

X	Y	Y'=P	Y''=F
0.00000	0.00000	.50000	1.00000
.02500	.01266	.52563	1.03797
.05000	.02596	.55224	1.07724
.07500	.03994	.57984	1.11783

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05479	.05526	.60831	.00047
.20000	.12172	.12214	.73326	.00042
.30000	.20209	.20248	.87779	.00038
.40000	.29802	.29836	1.04462	.00035
.50000	.41186	.41218	1.23685	.00032
.60000	.54635	.54664	1.45798	.00029
.70000	.70455	.70481	1.71194	.00026
.80000	.88998	.89022	2.00322	.00024
.90000	1.10661	1.10682	2.33683	.00021
1.00000	1.35895	1.35914	2.71847	.00019

H= .0125

X	Y	Y'=P	Y''=F
0.00000	0.00000	.50000	1.00000
.01250	.00629	.51266	1.01887
.02500	.01274	.52555	1.03805
.03750	.01935	.53869	1.05756

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05515	.05526	.60795	.00011
.20000	.12204	.12214	.73294	.00010
.30000	.20239	.20248	.87750	.00009
.40000	.29828	.29836	1.04436	.00008
.50000	.41211	.41218	1.23662	.00007
.60000	.54657	.54664	1.45776	.00007
.70000	.70475	.70481	1.71175	.00006
.80000	.89016	.89022	2.00304	.00005
.90000	1.10677	1.10682	2.33667	.00005
1.00000	1.35910	1.35914	2.71833	.00004

OUTPUTS OF AW04B FOR H=0.1 TO H=0.0125

H= .1000

X	Z	Z'=P	Z''=F
0.00000	0.00000	.50000	1.00000
.10000	.05259	.61052	1.15776
.20000	.11679	.73819	1.33820
.30000	.19443	.88546	1.54429

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
.40000	.29441	.29836	1.04803	.00395
.50000	.40515	.41218	1.24336	.00703
.60000	.54381	.54664	1.46025	.00282
.70000	.69855	.70481	1.71773	.00626
.80000	.88834	.89022	2.00452	.00188
.90000	1.10114	1.10682	2.34205	.00568
1.00000	1.35806	1.35914	2.71892	.00108

H= .0500

X	Z	Z'=P	Z''=F
0.00000	0.00000	.50000	1.00000
.05000	.02564	.55256	1.07691
.10000	.05397	.60914	1.15914
.15000	.08520	.67000	1.24703

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
.20000	.12105	.12214	.73392	.00109
.30000	.20154	.20248	.87833	.00094
.40000	.29756	.29836	1.04506	.00080
.50000	.41151	.41218	1.23719	.00067
.60000	.54608	.54664	1.45822	.00055
.70000	.70437	.70481	1.71210	.00044
.80000	.88987	.89022	2.00329	.00034
.90000	1.10657	1.10682	2.33683	.00025
1.00000	1.35898	1.35914	2.71840	.00016

H= .0250

X	Z	Z'=P	Z''=F
0.00000	0.00000	.50000	1.00000
.02500	.01266	.52563	1.03797
.05000	.02596	.55224	1.07724
.07500	.03994	.57984	1.11783

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
.10000	.05497	.05526	.60813	.00029
.20000	.12189	.12214	.73309	.00025
.30000	.20226	.20248	.87762	.00022
.40000	.29818	.29836	1.04446	.00018
.50000	.41203	.41218	1.23669	.00015
.60000	.54651	.54664	1.45782	.00013
.70000	.70471	.70481	1.71178	.00010
.80000	.89014	.89022	2.00306	.00008
.90000	1.10677	1.10682	2.33667	.00006
1.00000	1.35911	1.35914	2.71831	.00004

H= .0125

X	Z	Z'=P	Z''=F
0.00000	0.00000	.50000	1.00000
.01250	.00629	.51266	1.01887
.02500	.01274	.52555	1.03805
.03750	.01935	.53869	1.05756

X-VAL	C.SOL	EX.SOL	C.Z'	ERR
.10000	.05519	.05526	.60791	.00007
.20000	.12208	.12214	.73290	.00006
.30000	.20243	.20248	.87746	.00005
.40000	.29832	.29836	1.04432	.00004
.50000	.41215	.41218	1.23657	.00003
.60000	.54661	.54664	1.45772	.00003
.70000	.70479	.70481	1.71171	.00002
.80000	.89020	.89022	2.00300	.00002
.90000	1.10681	1.10682	2.33663	.00001
1.00000	1.35914	1.35914	2.71829	.00000

OUTPUTS OF AW05(R4B) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .100000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.12214	.05526	.73282	.06688
.20000	.20247	.12214	.87737	.08033
.30000	.29836	.20248	1.04422	.09588
.40000	.41217	.29836	1.23647	.11380
.50000	.54661	.41218	1.45760	.13443
.60000	.70478	.54664	1.71157	.15815
.70000	.89017	.70481	2.00284	.18536
.80000	1.10676	.89022	2.33645	.21655
.90000	1.35906	1.10682	2.71807	.25224
1.00000	1.65219	1.35914	3.15412	.29305

XO 0. YO 0. PO .5 H .050000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.08714	.05526	.66805	.03188
.20000	.16050	.12214	.80251	.03836
.30000	.24834	.20248	.95786	.04586
.40000	.35287	.29836	1.13702	.05450
.50000	.47664	.41218	1.34326	.06446
.60000	.62255	.54664	1.58031	.07591
.70000	.79387	.70481	1.85236	.08906
.80000	.99434	.89022	2.16415	.10413
.90000	1.22820	1.10682	2.52104	.12138
1.00000	1.50026	1.35914	2.92906	.14111

XO 0. YO 0. PO .5 H .025000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.07082	.05526	.63740	.01556
.20000	.14089	.12214	.76705	.01875
.30000	.22490	.20248	.91692	.02243
.40000	.32504	.29836	1.08983	.02667
.50000	.44375	.41218	1.28897	.03156
.60000	.58383	.54664	1.51795	.03719
.70000	.74846	.70481	1.78083	.04365
.80000	.94127	.89022	2.08221	.05106
.90000	1.16636	1.10682	2.42729	.05954
1.00000	1.42838	1.35914	2.82193	.06924

XO 0. YO 0. PO .5 H .012500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.06295	.05526	.62248	.00769
.20000	.13141	.12214	.74979	.00927
.30000	.21357	.20248	.89699	.01109
.40000	.31156	.29836	1.06685	.01319
.50000	.42780	.41218	1.26253	.01562
.60000	.56504	.54664	1.48756	.01841
.70000	.72642	.70481	1.74596	.02161
.80000	.91550	.89022	2.04227	.02528
.90000	1.13631	1.10682	2.38158	.02949
1.00000	1.39344	1.35914	2.76968	.03430

OUTPUTS OF AW05(R3B) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .100000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.12209	.05526	.73211	.06683
.20000	.20234	.12214	.87622	.08020
.30000	.29810	.20248	1.04257	.09562
.40000	.41172	.29836	1.23424	.11336
.50000	.54593	.41218	1.45470	.13375
.60000	.70379	.54664	1.70788	.15715
.70000	.88878	.70481	1.99825	.18397
.80000	1.10488	.89022	2.33080	.21467
.90000	1.35658	1.10682	2.71120	.24976
1.00000	1.64898	1.35914	3.14583	.28984

XO 0. YO 0. PO .5 H .050000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.08713	.05526	.66792	.03187
.20000	.16048	.12214	.80227	.03834
.30000	.24828	.20248	.95750	.04580
.40000	.35277	.29836	1.13652	.05441
.50000	.47649	.41219	1.34260	.06431
.60000	.62232	.54664	1.57946	.07569
.70000	.79355	.70481	1.85129	.08874
.80000	.99391	.89022	2.16283	.10369
.90000	1.22763	1.10682	2.51943	.12080
1.00000	1.49950	1.35914	2.92711	.14036

XO 0. YO 0. PO .5 H .025000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.07082	.05526	.63737	.01556
.20000	.14088	.12214	.76699	.01874
.30000	.22489	.20248	.91684	.02241
.40000	.32502	.29836	1.08971	.02665
.50000	.44371	.41218	1.28882	.03153
.60000	.58377	.54664	1.51774	.03714
.70000	.74839	.70481	1.78057	.04358
.80000	.94117	.89022	2.08189	.05095
.90000	1.16622	1.10682	2.42690	.05940
1.00000	1.42820	1.35914	2.82146	.06906

XO 0. YO 0. PO .5 H .012500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.06295	.05526	.62248	.00769
.20000	.13141	.12214	.74978	.00926
.30000	.21357	.20248	.89697	.01109
.40000	.31155	.29836	1.06683	.01319
.50000	.42779	.41218	1.26249	.01561
.60000	.56503	.54664	1.48751	.01839
.70000	.72640	.70481	1.74590	.02159
.80000	.91547	.89022	2.04219	.02526
.90000	1.13627	1.10682	2.38148	.02945
1.00000	1.39339	1.35914	2.76956	.03425

OUTPUTS OF AW05(R2B) FOR H=0.1 TO H=0.003125

XO 0. YO 0. PO .5 H .100000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.12080	.05526	.71602	.06554
.20000	.19911	.12214	.85024	.07697
.30000	.29188	.20248	1.00513	.08940
.40000	.40131	.29836	1.18351	.10295
.50000	.52991	.41218	1.39851	.11773
.60000	.68053	.54664	1.62371	.13389
.70000	.85637	.70481	1.89314	.15156
.80000	1.06109	.89022	2.20133	.17088
.90000	1.29883	1.10682	2.55340	.19201
1.00000	1.57425	1.35914	2.95511	.21511

XO 0. YO 0. PO .5 H .050000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.08674	.05526	.66189	.03148
.20000	.15929	.12214	.79146	.03715
.30000	.24579	.20248	.94115	.04331
.40000	.34838	.29836	1.11373	.05001
.50000	.46951	.41218	1.31232	.05733
.60000	.61195	.54664	1.54046	.06532
.70000	.77886	.70481	1.80213	.07404
.80000	.97380	.89022	2.10183	.08359
.90000	1.20084	1.10682	2.44463	.09402
1.00000	1.46456	1.35914	2.83627	.10542

XO 0. YO 0. PO .5 H .025000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.07068	.05526	.63486	.01542
.20000	.14038	.12214	.76213	.01824
.30000	.22379	.20248	.90925	.02131
.40000	.32301	.29836	1.07896	.02464
.50000	.44046	.41218	1.27438	.02828
.60000	.57889	.54664	1.49902	.03225
.70000	.74141	.70481	1.75684	.03659
.80000	.93155	.89022	2.05233	.04133
.90000	1.15334	1.10682	2.39053	.04652
1.00000	1.41132	1.35914	2.77717	.05218

XO 0. YO 0. PO .5 H .012500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.06289	.05526	.62135	.00763
.20000	.13118	.12214	.74748	.00904
.30000	.21305	.20248	.89332	.01057
.40000	.31060	.29836	1.06161	.01223
.50000	.42623	.41218	1.25545	.01405
.60000	.56266	.54664	1.47834	.01603
.70000	.72300	.70481	1.73425	.01819
.80000	.91077	.89022	2.02764	.02055
.90000	1.12996	1.10682	2.36355	.02314
1.00000	1.38510	1.35914	2.74770	.02596

XO 0. YO 0. PO .5 H .006250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05905	.05526	.61460	.00379
.20000	.12664	.12214	.74016	.00450
.30000	.20774	.20248	.88536	.00526
.40000	.30446	.29836	1.05294	.00609
.50000	.41918	.41218	1.24599	.00700
.60000	.55462	.54664	1.46801	.00799
.70000	.71388	.70481	1.72296	.00907
.80000	.90046	.89022	2.01531	.01025
.90000	1.11836	1.10682	2.35008	.01154
1.00000	1.37209	1.35914	2.73298	.01295

XO 0. YO 0. PO .5 H .003125

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05715	.05526	.61122	.00189
.20000	.12438	.12214	.73650	.00224
.30000	.20510	.20248	.88138	.00263
.40000	.30141	.29836	1.04861	.00304
.50000	.41567	.41218	1.24127	.00349
.60000	.55062	.54664	1.46285	.00399
.70000	.70934	.70481	1.71733	.00453
.80000	.89533	.89022	2.00915	.00512
.90000	1.11258	1.10682	2.34335	.00576
1.00000	1.36561	1.35914	2.72563	.00646

OUTPUTS OF AW06(D4B) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.11660	.05526	.55258	.06135
.20000	.18496	.12214	.61346	.06282
.30000	.26133	.20248	.68353	.05886
.40000	.34687	.29836	.76378	.04850
.50000	.44281	.41218	.85533	.03063
.60000	.55056	.54664	.95943	.00392
.70000	.67166	.70481	1.07749	.03315
.80000	.80785	.89022	1.21105	.08237
.90000	.96103	1.10682	1.36185	.14579
1.00000	1.13335	1.35914	1.53183	.22579

XO 0. YO 0. PO .5 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.08309	.05526	.55324	.02783
.20000	.14628	.12214	.61479	.02414
.30000	.21678	.20248	.68552	.01431
.40000	.29566	.29836	.76645	.00270
.50000	.38407	.41218	.85869	.02811
.60000	.48329	.54664	.96349	.06334
.70000	.59477	.70481	1.08225	.11004
.80000	.72010	.89022	1.21652	.17012
.90000	.86105	1.10682	1.36804	.24578
1.00000	1.01958	1.35914	1.53875	.33956

XO 0. YO 0. PO .5 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.06749	.05526	.55356	.01223
.20000	.12825	.12214	.61543	.00611
.30000	.19599	.20248	.68649	.00648
.40000	.27174	.29836	.76775	.02663
.50000	.35659	.41218	.86032	.05559
.60000	.45181	.54664	.96545	.09483
.70000	.55875	.70481	1.08454	.14606
.80000	.67896	.89022	1.21915	.21126
.90000	.81413	1.10682	1.37101	.29269
1.00000	.96617	1.35914	1.54206	.39297

XO 0. YO 0. PO .5 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05997	.05526	.55372	.00471
.20000	.11955	.12214	.61575	.00259
.30000	.18596	.20248	.68697	.01652
.40000	.26018	.29836	.76838	.03819
.50000	.34331	.41218	.86111	.06887
.60000	.43658	.54664	.96641	.11006
.70000	.54132	.70481	1.08566	.16350
.80000	.65904	.89022	1.22044	.23118
.90000	.79142	1.10682	1.37246	.31541
1.00000	.94030	1.35914	1.54368	.41884

OUTPUTS OF AW06(D3B) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.11658	.05526	.55250	.06132
.20000	.18490	.12214	.61328	.06276
.30000	.26124	.20248	.68324	.05876
.40000	.34672	.29836	.76337	.04835
.50000	.44259	.41218	.85478	.03041
.60000	.55025	.54664	.95872	.00361
.70000	.67124	.70481	1.07659	.03357
.80000	.80729	.89022	1.20994	.08292
.90000	.96032	1.10682	1.36051	.14651
1.00000	1.13244	1.35914	1.53022	.22670

XO 0. YO 0. PO .5 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.08309	.05526	.55322	.02783
.20000	.14627	.12214	.61474	.02413
.30000	.21677	.20248	.68545	.01429
.40000	.29563	.29836	.76635	.00273
.50000	.38402	.41218	.85855	.02816
.60000	.48323	.54664	.96331	.06340
.70000	.59469	.70481	1.08202	.11013
.80000	.71998	.89022	1.21624	.17023
.90000	.86089	1.10682	1.36770	.24593
1.00000	1.01939	1.35914	1.53834	.33975

XO 0. YO 0. PO .5 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.06749	.05526	.55356	.01223
.20000	.12825	.12214	.61542	.00611
.30000	.19599	.20248	.68647	.00649
.40000	.27173	.29836	.76772	.02663
.50000	.35658	.41218	.86028	.05560
.60000	.45179	.54664	.96540	.09484
.70000	.55873	.70481	1.08448	.14608
.80000	.67893	.89022	1.21908	.21129
.90000	.81410	1.10682	1.37093	.29272
1.00000	.96613	1.35914	1.54196	.39301

XO 0. YO 0. PO .5 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05997	.05526	.55372	.00471
.20000	.11955	.12214	.61574	.00259
.30000	.18595	.20248	.68696	.01652
.40000	.26018	.29836	.76838	.03819
.50000	.34331	.41218	.86110	.06887
.60000	.43657	.54664	.96640	.11006
.70000	.54131	.70481	1.08565	.16350
.80000	.65903	.89022	1.22042	.23118
.90000	.79141	1.10682	1.37244	.31541
1.00000	.94029	1.35914	1.54365	.41885

OUTPUTS OF AW06(D2B) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .10000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.11580	.05526	.55000	.06054
.20000	.18329	.12214	.60801	.06115
.30000	.25844	.20248	.67487	.05596
.40000	.34234	.29836	.75153	.04398
.50000	.43620	.41218	.83904	.02402
.60000	.54135	.54664	.93859	.00528
.70000	.65928	.70481	1.05151	.04553
.80000	.79163	.89022	1.17926	.09859
.90000	.94024	1.10682	1.32351	.16658
1.00000	1.10714	1.35914	1.48607	.25200

XO 0. YO 0. PO .5 H .05000

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.08289	.05526	.55194	.02763
.20000	.14576	.12214	.61203	.02362
.30000	.21579	.20248	.68115	.01331
.40000	.29400	.29836	.76026	.00437
.50000	.38153	.41218	.85046	.03065
.60000	.47964	.54664	.95296	.06700
.70000	.58973	.70481	1.06912	.11508
.80000	.71337	.89022	1.20046	.17684
.90000	.85229	1.10682	1.34867	.25453
1.00000	1.00841	1.35914	1.51563	.35073

XO 0. YO 0. PO .5 H .02500

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.06743	.05526	.55291	.01217
.20000	.12806	.12214	.61405	.00592
.30000	.19559	.20248	.68429	.00688
.40000	.27104	.29836	.76464	.02733
.50000	.35549	.41218	.85618	.05669
.60000	.45019	.54664	.96016	.09644
.70000	.55649	.70481	1.07795	.14832
.80000	.67591	.89022	1.21108	.21430
.90000	.81014	1.10682	1.36128	.29669
1.00000	.96104	1.35914	1.53044	.39810

XO 0. YO 0. PO .5 H .01250

X-VAL	C.SOL	EX.SOL	C.Y'	ERR
.10000	.05995	.05526	.55339	.00469
.20000	.11947	.12214	.61505	.00267
.30000	.18578	.20248	.68587	.01670
.40000	.25986	.29836	.76682	.03851
.50000	.34280	.41218	.85904	.06938
.60000	.43582	.54664	.96376	.11082
.70000	.54025	.70481	1.08236	.16456
.80000	.65759	.89022	1.21639	.23262
.90000	.78951	1.10682	1.36758	.31731
1.00000	.93784	1.35914	1.53785	.42130

```
100    C        DATE: 23/06/1987
200    C        NAME OF FILE: AW07A
300    C        NUMERICAL SOLUTION OF ORDINARY DIFF. EQUATION
400    C        Y''=Y+EXP(X) WITH INITIAL CONDITIONS BELOW
500    C        CLASSICAL NUMEROV'S METHOD IS USED
600    C        FORMULA: Yk+1=2Yk-Yk-1+(h.h/12)(Fk+1+10Fk+Fk-1)



700              IMPLICIT REAL*8(A-H,O-Z)
800              DIMENSION F(3)
900    C         DEFINE STATEMENT FUNCTIONS
000              FSOL(X)=0.5*X*DEXP(X)
100              FUN(X,Y)=Y+DEXP(X)
200              WRITE(2,77)
300        77    FORMAT(10X,38HOUTPUTS OF AW07A FOR H=0.1 TO H=0.0125//)
400    C         INITIALIZE X,Y,H
500              XO=0.0
600              YO=0.0
700              DY1=0.5
800              H=0.1
900    C         PRINT INITIAL VALUES
000              WRITE(2,100)XO,YO,DY1,H
100      100     FORMAT(10X,'XO',F8.5,3X,'YO',F8.5,3X,'DY1',F8.5,3X,'H',F8.5//)
200    C         PRINT COLUMN HEADINGS
                 WRITE(2,101)
      101        FORMAT(10X,'X',12X,'EX.SOL',9X,'C.SOL',8X,'DY1',10X,'ERROR'/)
                 H=H/2.
0                DY2=YO+DEXP(XO)
700              DY3=DY1+DEXP(XO)
800              DY4=DY2+DEXP(XO)
900              Y1=YO+H*DY1+(H*H/2.)*DY2+((H**3)/6.)*DY3+((H**4)/24.)*DY4
000              X1=XO+H
100              N=2
200              NSTEP=20
300              DO 10 I=2,NSTEP
400              F(N-1)=FUN(XO,YO)
500              F(N)=FUN(X1,Y1)
600              X2=X1+H
700              PRY=2.*Y1-YO+(H*H/12.)*(F(N-1)+10.*F(N))
800              DP1=(Y1-YO)/H
900              DP2=Y1+DEXP(X1)
000              DP3=DP1+DEXP(X1)
100              DP4=DP2+DEXP(X1)
200
300              Y2=Y1+H*DP1+0.5*H*H*DP2+((H**3)/6.)*DP3+((H**4)/24.)*DP4
400              X2=X1+H
500              F(N+1)=FUN(X2,Y2)
600              CRY=PRY+(H*H/12.)*(F(N+1))
70               EXACT=FSOL(X2)
                 ERROR=EXACT-CRY
                 ERR=DABS(ERROR)
000              N1=2
100              N2=I/N1
200              N3=N1*N2
300              IF(I.NE.N3) GO TO 103
```

```
400             WRITE(2,102)X2,EXACT,CRY,DP1,ERR
500       102   FORMAT(4X,5F13.5)
600     C       INTER CHANGE LOCATIONS
700       103   XO=X1
800             X1=X2
900             YO=Y1
000             Y1=Y2
100             F(N-1)=F(N)
200             F(N)=F(N+1)
300        10   CONTINUE
400             STOP
500             END
```

OUTPUTS OF AW07A FOR H=0.1 TO H=0.0125

XO 0.00000 YO 0.00000 DY1 .50000 H .10000

X	EX.SOL	C.SOL	DY1	ERROR
.10000	.05526	.05526	.52564	.00000
.20000	.12214	.11535	.58292	.00679
.30000	.20248	.18230	.64894	.02018
.40000	.29836	.25711	.72465	.04126
.50000	.41218	.34088	.81108	.07130
.60000	.54664	.43483	.90944	.11180
.70000	.70481	.54034	1.02103	.16447
.80000	.89022	.65892	1.14734	.23129
.90000	1.10682	.79225	1.29001	.31457
1.00000	1.35914	.94221	1.45087	.41693

XO 0.00000 YO 0.00000 DY1 .50000 H .05000

X	EX.SOL	C.SOL	DY1	ERROR
.10000	.05526	.05363	.53943	.00163
.20000	.12214	.11284	.59914	.00930
.30000	.20248	.17881	.66781	.02367
.40000	.29836	.25251	.74642	.04585
.50000	.41218	.33505	.83605	.07713
.60000	.54664	.42763	.93791	.11900
.70000	.70481	.53160	1.05338	.17322
.80000	.89022	.64843	1.18395	.24178
.90000	1.10682	.77981	1.33133	.32701
1.00000	1.35914	.92757	1.49739	.43158

XO 0.00000 YO 0.00000 DY1 .50000 H .02500

X	EX.SOL	C.SOL	DY1	ERROR
.10000	.05526	.05305	.54657	.00221
.20000	.12214	.11185	.60751	.01029
.30000	.20248	.17736	.67753	.02511
.40000	.29836	.25057	.75761	.04780
.50000	.41218	.33254	.84886	.07964
.60000	.54664	.42449	.95251	.12215
.70000	.70481	.52774	1.06994	.17707
.80000	.89022	.64378	1.20268	.24644
.90000	1.10682	.77425	1.35245	.33257
1.00000	1.35914	.92100	1.52115	.43814

XO 0.00000 YO 0.00000 DY1 .50000 H .01250

X	EX.SOL	C.SOL	DY1	ERROR
.10000	.05526	.05282	.55020	.00244
.20000	.12214	.11143	.61176	.01072
.30000	.20248	.17673	.68246	.02575
.40000	.29836	.24969	.76329	.04867
.50000	.41218	.33140	.85535	.08078
.60000	.54664	.42304	.95991	.12359
.70000	.70481	.52595	1.07833	.17886
.80000	.89022	.64161	1.21216	.24861
.90000	1.10682	.77166	1.36313	.33517
1.00000	1.35914	.91791	1.53317	.44123

```
0100    C       DATE: 23/06/1987
0200    C       NAME OF FILE: AW07B
0300    C       NUMERICAL SOLUTION OF ORDINARY DIFF. EQUATION
0400    C       Y''=Y+EXP(X) WITH INITIAL CONDITIONS BELOW
0500    C       CLASSICAL NUMEROV'S METHOD IS USED
0600    C       FORMULA: Yk+1=2Yk-Yk-1+(h.h/12)(Fk+1+10Fk+Fk-1)
1000            IMPLICIT REAL*8(A-H,O-Z)
1100            DIMENSION F(3)
1200    C       DEFINE STATEMENT FUNCTIONS
1300            FSOL(X)=0.5*X*DEXP(X)
1400            FUN(X,Y)=Y+DEXP(X)
1450            WRITE(2,77)
1460       77   FORMAT(10X,38HOUTPUTS OF AW07B FOR H=0.1 TO H=0.0125//)
1500    C       INITIALIZE X,Y,H
1600            XO=0.0
1700            YO=0.0
1800            DY1=0.5
1900            H=0.1
2100    C       PRINT INITIAL VALUES
2200            WRITE(2,100)XO,YO,DY1,H
2300      100   FORMAT(10X,'XO',F8.5,3X,'YO',F8.5,3X,'DY1',F8.5,3X,'H',F8.5//)
2400    C       PRINT COLUMN HEADINGS
2500            WRITE(2,101)
2600      101   FORMAT(10X,'X',12X,'EX.SOL',9X,'C.SOL',8X,'ERROR'/)
2700            H=H/2.
2800            DY2=YO+DEXP(XO)
2900            DY3=DY1+DEXP(XO)
3000            DY4=DY2+DEXP(XO)
                Y1=YO+H*DY1+(H*H/2.)*DY2+((H**3)/6.)*DY3+((H**4)/24.)*DY4
3500            X1=XO+H
3600            N=2
3700            NSTEP=20
3800            DO 10 I=2,NSTEP
3900            F(N-1)=FUN(XO,YO)
4000            F(N)=FUN(X1,Y1)
4100            X2=X1+H
4200            B1=12./(12.-(H*H))
4300            B2=2.*Y1-YO+(H*H/12.)*(F(N-1)+10.*F(N))
4350            PRY=B1*B2
4400            CRY=PRY+(B1*H*H*DEXP(X2))/12.
4500            B2=2.*Y1-YO+(H*H/12.)*(F(N-1)+10.*F(N))
4700            EXACT=FSOL(X2)
4800            ERROR=EXACT-CRY
4900            ERR=DABS(ERROR)
4950            N1=2
                N2=I/N1
                N3=N1*N2
                IF(I.NE.N3) GO TO 103
                WRITE(2,102)X2,EXACT,CRY,ERR
5100      102   FORMAT(4X,4F13.5)
5200    C       INTER CHANGE LOCATIONS
5300      103   XO=X1
```

```
 400            X1=X2
 500            YO=Y1
 600            Y1=CRY
 700            F(N-1)=F(N)
 800            F(N)=F(N+1)
 900       10   CONTINUE
 000            STOP
 100            END
```

OUTPUTS OF AW07B FOR H=0.1 TO H=0.0125

XO 0.00000 YO 0.00000 DY1 .50000 H .10000

X	EX.SOL	C.SOL	ERROR
.10000	.05526	.05526	.00000
.20000	.12214	.12214	.00000
.30000	.20248	.20248	.00000
.40000	.29836	.29836	.00000
.50000	.41218	.41218	.00000
.60000	.54664	.54664	.00000
.70000	.70481	.70481	.00000
.80000	.89022	.89022	.00000
.90000	1.10682	1.10682	.00000
1.00000	1.35914	1.35914	.00000

XO 0.00000 YO 0.00000 DY1 .50000 H .05000

X	EX.SOL	C.SOL	ERROR
.10000	.05526	.05526	.00000
.20000	.12214	.12214	.00000
.30000	.20248	.20248	.00000
.40000	.29836	.29836	.00000
.50000	.41218	.41218	.00000
.60000	.54664	.54664	.00000
.70000	.70481	.70481	.00000
.80000	.89022	.89022	.00000
.90000	1.10682	1.10682	.00000
1.00000	1.35914	1.35914	.00000

XO 0.00000 YO 0.00000 DY1 .50000 H .02500

X	EX.SOL	C.SOL	ERROR
.10000	.05526	.05526	.00000
.20000	.12214	.12214	.00000
.30000	.20248	.20248	.00000
.40000	.29836	.29836	.00000
.50000	.41218	.41218	.00000
.60000	.54664	.54664	.00000
.70000	.70481	.70481	.00000
.80000	.89022	.89022	.00000
.90000	1.10682	1.10682	.00000
1.00000	1.35914	1.35914	.00000

XO 0.00000 YO 0.00000 DY1 .50000 H .01250

X	EX.SOL	C.SOL	ERROR
.10000	.05526	.05526	.00000
.20000	.12214	.12214	.00000
.30000	.20248	.20248	.00000
.40000	.29836	.29836	.00000
.50000	.41218	.41218	.00000
.60000	.54664	.54664	.00000
.70000	.70481	.70481	.00000
.80000	.89022	.89022	.00000
.90000	1.10682	1.10682	.00000
1.00000	1.35914	1.35914	.00000

OUTPUTS OF AW08(EB) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .1000

X	EX	ER1	ER2	ER3	ER4
.50000	.41218	.00094	.01366	.00143	.00173
.60000	.54664	.00192	.01663	.00253	.00188
.70000	.70481	.00324	.01975	.00370	.00380
.80000	.89022	.00494	.02306	.00489	.00374
.90000	1.10682	.00707	.02658	.00613	.00599
1.00000	1.35914	.00969	.03036	.00744	.00577

XO 0. YO 0. PO .5 H .0500

X	EX	ER1	ER2	ER3	ER4
.30000	.20248	.00027	.00391	.00032	.00024
.40000	.29836	.00063	.00527	.00059	.00046
.50000	.41219	.00116	.00667	.00086	.00068
.60000	.54664	.00188	.00813	.00114	.00090
.70000	.70481	.00280	.00967	.00144	.00114
.80000	.89022	.00397	.01131	.00175	.00139
.90000	1.10682	.00541	.01306	.00207	.00165
1.00000	1.35914	.00716	.01493	.00242	.00193

XO 0. YO 0. PO .5 H .0250

X	EX	ER1	ER2	ER3	ER4
.20000	.12214	.00008	.00128	.00007	.00006
.30000	.20248	.00022	.00193	.00014	.00011
.40000	.29836	.00045	.00260	.00020	.00016
.50000	.41218	.00076	.00329	.00027	.00022
.60000	.54664	.00118	.00402	.00034	.00027
.70000	.70481	.00171	.00479	.00042	.00034
.80000	.89022	.00238	.00560	.00050	.00040
.90000	1.10682	.00320	.00647	.00058	.00047
1.00000	1.35914	.00419	.00740	.00067	.00054

XO 0. YO 0. PO .5 H .0125

X	EX	ER1	ER2	ER3	ER4
.10000	.05525	.00001	.00032	.00001	.00001
.20000	.12214	.00005	.00063	.00002	.00002
.30000	.20248	.00013	.00096	.00004	.00003
.40000	.29835	.00026	.00129	.00006	.00005
.50000	.41218	.00043	.00164	.00007	.00006
.60000	.54664	.00065	.00200	.00009	.00007
.70000	.70481	.00094	.00238	.00011	.00009
.80000	.89022	.00129	.00279	.00013	.00011
.90000	1.10682	.00173	.00322	.00015	.00012

OUTPUTS OF AWOB(RB) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .1000

X	EX	ER1	ER2	ER3	ER4
.50000	.41219	.00094	.01366	.00973	.00612
.60000	.54664	.00192	.01663	.00883	.00934
.70000	.70481	.00324	.01975	.00799	.00456
.80000	.89022	.00494	.02306	.00723	.00820
.90000	1.10682	.00707	.02658	.00654	.00326
1.00000	1.35914	.00969	.03036	.00592	.00733

XO 0. YO 0. PO .5 H .0500

X	EX	ER1	ER2	ER3	ER4
.30000	.20248	.00027	.00391	.00234	.00237
.40000	.29836	.00063	.00527	.00212	.00220
.50000	.41218	.00116	.00667	.00192	.00205
.60000	.54664	.00188	.00813	.00174	.00192
.70000	.70481	.00280	.00967	.00157	.00180
.80000	.89022	.00397	.01131	.00142	.00169
.90000	1.10682	.00541	.01306	.00129	.00160
1.00000	1.35914	.00716	.01493	.00116	.00152

XO 0. YO 0. PO .5 H .0250

X	EX	ER1	ER2	ER3	ER4
.20000	.12214	.00008	.00128	.00058	.00058
.30000	.20248	.00022	.00193	.00052	.00054
.40000	.29836	.00045	.00260	.00047	.00050
.50000	.41218	.00076	.00329	.00043	.00047
.60000	.54664	.00118	.00402	.00039	.00044
.70000	.70481	.00171	.00479	.00035	.00041
.80000	.89022	.00238	.00560	.00032	.00039
.90000	1.10682	.00320	.00647	.00029	.00036
1.00000	1.35914	.00419	.00740	.00026	.00035

XO 0. YO 0. PO .5 H .0125

X	EX	ER1	ER2	ER3	ER4
.10000	.05526	.00001	.00032	.00015	.00015
.20000	.12214	.00005	.00063	.00014	.00014
.30000	.20248	.00013	.00096	.00012	.00013
.40000	.29836	.00026	.00129	.00011	.00012
.50000	.41218	.00043	.00164	.00010	.00011
.60000	.54664	.00065	.00200	.00009	.00010
.70000	.70481	.00094	.00238	.00008	.00010
.80000	.89022	.00129	.00279	.00007	.00009
.90000	1.10682	.00173	.00322	.00007	.00009
1.00000	1.35914	.00225	.00369	.00006	.00008

OUTPUTS OF AW09(EB) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .1000

X	EX	ER1	ER2	ER3	ER4
.40000	.29836	.00026	.01081	.00069	.00095
.50000	.41218	.00094	.01366	.00147	.00087
.60000	.54664	.00192	.01663	.00229	.00237
.70000	.70481	.00324	.01975	.00313	.00208
.80000	.89022	.00494	.02306	.00400	.00385
.90000	1.10682	.00707	.02658	.00491	.00338
1.00000	1.35914	.00969	.03036	.00586	.00547

XO 0. YO 0. PO .5 H .0500

X	EX	ER1	ER2	ER3	ER4
.20000	.12214	.00005	.00259	.00009	.00012
.30000	.20248	.00027	.00391	.00029	.00029
.40000	.29836	.00063	.00527	.00048	.00046
.50000	.41218	.00116	.00667	.00068	.00064
.60000	.54664	.00188	.00813	.00089	.00082
.70000	.70481	.00280	.00967	.00111	.00101
.80000	.89022	.00397	.01131	.00134	.00121
.90000	1.10682	.00541	.01306	.00158	.00142
1.00000	1.35914	.00716	.01493	.00184	.00165

XO 0. YO 0. PO .5 H .0250

X	EX	ER1	ER2	ER3	ER4
.10000	.05526	.00001	.00064	.00001	.00001
.20000	.12214	.00008	.00128	.00006	.00006
.30000	.20249	.00022	.00193	.00011	.00010
.40000	.29834	.00045	.00260	.00016	.00014
.50000	.41218	.00076	.00329	.00021	.00019
.60000	.54664	.00118	.00402	.00026	.00023
.70000	.70481	.00171	.00479	.00032	.00028
.80000	.89022	.00238	.00560	.00038	.00033
.90000	1.10682	.00320	.00647	.00044	.00039
1.00000	1.35914	.00419	.00740	.00051	.00045

XO 0. YO 0. PO .5 H .0125

X	EX	ER1	ER2	ER3	ER4
.10000	.05526	.00001	.00033	.00001	.00001
.20000	.12214	.00005	.00063	.00002	.00002
.30000	.20249	.00013	.00096	.00003	.00003
.40000	.29834	.00025	.00129	.00004	.00004
.50000	.41218	.00043	.00164	.00006	.00005
.60000	.54664	.00065	.00200	.00007	.00006
.70000	.70481	.00094	.00238	.00009	.00007
.80000	.89022	.00129	.00279	.00010	.00009
.90000	1.10682	.00173	.00322	.00012	.00010
1.00000	1.35914	.00225	.00369	.00013	.00012

OUTPUTS OF AW09(RB) FOR H=0.1 TO H=0.0125

XO 0. YO 0. PO .5 H .1000

X	EX	ER1	ER2	ER3	ER4
.40000	.29836	.00026	.01081	.00724	.00395
.50000	.41218	.00094	.01366	.00657	.00704
.60000	.54664	.00192	.01663	.00594	.00282
.70000	.70481	.00324	.01975	.00538	.00626
.80000	.89022	.00494	.02306	.00487	.00188
.90000	1.10682	.00707	.02653	.00440	.00569
1.00000	1.35914	.00969	.03036	.00399	.00108

XO 0. YO 0. PO .5 H .0500

X	EX	ER1	ER2	ER3	ER4
.20000	.12214	.00005	.00259	.00184	.00110
.30000	.20246	.00027	.00391	.00167	.00095
.40000	.29836	.00063	.00527	.00151	.00081
.50000	.41218	.00116	.00667	.00137	.00068
.60000	.54664	.00188	.00813	.00124	.00056
.70000	.70481	.00280	.00967	.00112	.00045
.80000	.89022	.00397	.01131	.00101	.00035
.90000	1.10682	.00541	.01306	.00092	.00026
1.00000	1.35914	.00716	.01493	.00083	.00017

XO 0. YO 0. PO .5 H .0250

X	EX	ER1	ER2	ER3	ER4
.10000	.05526	.00001	.00064	.00046	.00029
.20000	.12214	.00008	.00128	.00042	.00025
.30000	.20248	.00022	.00193	.00038	.00021
.40000	.29836	.00045	.00260	.00034	.00018
.50000	.41218	.00076	.00329	.00031	.00015
.60000	.54664	.00118	.00402	.00028	.00012
.70000	.70481	.00171	.00479	.00025	.00010
.80000	.89022	.00238	.00560	.00023	.00007
.90000	1.10682	.00320	.00647	.00021	.00005
1.00000	1.35914	.00419	.00740	.00019	.00003

XO 0. YO 0. PO .5 H .0125

X	EX	ER1	ER2	ER3	ER4
.10000	.05526	.00001	.00032	.00011	.00007
.20000	.12214	.00005	.00063	.00010	.00006
.30000	.20248	.00013	.00096	.00009	.00005
.40000	.29836	.00026	.00129	.00008	.00004
.50000	.41218	.00043	.00164	.00007	.00004
.60000	.54664	.00065	.00200	.00007	.00003
.70000	.70481	.00094	.00238	.00006	.00002
.80000	.89022	.00129	.00279	.00006	.00002
.90000	1.10682	.00173	.00322	.00005	.00001
1.00000	1.35914	.00225	.00369	.00005	.00001