A CLASS OF INVERSE MONO-IMPLICIT RUNGE-KUTTA SCHEMES

BY

SIMON STEPHEN

M.Sc./MA/05/041

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY YOLA IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF MASTER DEGREE OF SCIENCE IN MATHEMATICS

DECEMBER 2009

DECLARATION

I declare that this work was carried out in its original form by Simon Stephen (M.Sc/MA/05/0041) of the department of Mathematics and Computer Science, Federal University of Technology, Yola. Nigeria

……………………………… ………………………………

Sign Date

APPROVAL PAGE

This thesis entitled "A Class of Inverse Mono-implicit Runge-Kutta Schemes" written by Simon Stephen (M.Sc/MA/05/041) meets the regulations governing the award of master degree of science in Mathematics. Federal University of Technology, Yola and is approved for its contribution to knowledge and literary presentation

………………………… ………………………..

Prof.M.R.Odekunle Date

(Supervisor)

………………………... …………………………

Prof. G.Wajiga Date

(Internal Examiner)

………………………… ……………..........

Prof.S.U. Momoh Date

(Internal Examiner)

Mathematics Department

University of Jos, Nigeria

……………………………… …………………………

Dr.M.O.Egwurube Date

(Head of Department

………………………… ………..................

Prof.A.Nur Date

(Dean School of Post Graduate studies)

DEDICATION

This work is dedicated to God Almighty who is my heavenly father and my all, to Him be glory and honor now and forever more. Amen

ACKNOWLEDGEMENT

First and foremost, I want to thank my heavenly Father for providing all that I needed to undertake this study. His presence has provided a sweet companion all through my years of studies and even now.

I also have to thank several people for their contributions in making this work a success. Prominent among them is my supervisor, Prof.M.R Odekunle, without whose guidance, comments and constructive criticism, the accomplishment of this work would have been an illusion. He played a dual role both as a supervisor and a model and has greatly shaped my life to face the challenges that lies ahead. To him I owe my thanks for giving me the benefit of his expertise.

My special thanks also go to the following persons Dr. M.O.Egwurube (H.O.D. Mathematics and Computer Science), Prof.G.Wajiga, Mr. Paul Atsen (PG-coordinator), and all the staff of Mathematics and Computer Science department, Federal University of Technology, Yola (FUTY). Your contributions in making this work a reality is well acknowledged.

Heartfelt gratitude also goes to my beloved wife, Mrs.Ednah and our children for their love, care and concern; above all for their unflinching determination to support me. They stood by me during this period of research.

I also owe debt of gratitude to my friends and colleagues,especially Abubakar Fotta, Gabriel and Shelleng. Finally, I want to thank Mal Kabir M. Ali (HOD Mathematics and Statistics) Federal Polytechnic, Mubi for his moral support and encouragement.

ABSTRACT

A class of inverse Mono-Implicit Runge-Kutta methods used for the numerical solution of initial value problems whose solution have singular point(s) shall be discussed. In this project, Inverse Mono-implicit Runge-Kutta schemes are developed and are found to be consistent, A-stable and able to perform well when faced with singularity problem but poorly when applied to non-singular problems. The results obtained when compared with those by Classical Runge- Kutta methods are better.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.0 INTRODUCTION

Numerical analysis is the branch of mathematics involved with scientific computation for obtaining numerical values (or solutions) to various mathematical problems for which there are always physical (real life) applications. It involves the study, development and analysis of algorithms.

The development of class of inverse Mono- implicit Runge – Kutta Schemes has its bases in Runge – Kutta Schemes as the name implies. According to Cooper and Vignesvaran (1993), considerable attention has been given to the development of the Runge – Kutta methods which allow a more efficient implementation.

Let us consider the initial value problem (IVP)

$$y'(x) = f(x, y),\ y(x_0) = y_0 \tag{1.1}$$

The numerical solution of (1.1) at (x_n, y_n) is given by

$$y_{n+1} = y_n + h\varphi\ (x_n, y_n, h) \tag{1.2}$$

where

$$\varphi\ (x, y, h) = \sum_{r=1}^{R} c_r k_r$$

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

..

$$k_r = f(x_n + ha_r, y_n + h\sum_{s=1}^{r-1} b_{rj} k_s) \tag{1.3}$$

and

$a_r = \sum_{j=1}^{r-1} b_{rj}$, r = 2,3, … ,R , the values above are for explicit case

To specify a particular method, one needs to provide the integer s (the number of stages), and the coefficients a_r (for $1 \le j \le s$), b_{rj} (for r = 1, 2, 3,…,s) and c_r (for r =2, 3,…,s). The Runge – Kutta method is consistent if

$c_r = \sum_{r=1}^{r-1} a_r$ for r= 2,3,…, S

The method could also be said to have certain order say p if it satisfies certain requirements i.e the truncation error is $o\left(h^{p+1}\right)$. This can be derived from the truncation error itself. For example a two – stage method has order 2 if

$c_1 + c_2 = 1$, $c_2 a_2 = \frac{1}{2}$ and $a_2 b_{21} = \frac{1}{2}$ (Wikipedia 2007)

Runge – Kutta Schemes involve tedious manipulations but it is a good price to pay for accuracy of result. Cash and Singhal (1982) proposed the mono implicit Runge – Kutta formulae for the numerical integration of stiff differential equation. The standard form of this class is given by,

$$y_{n+1} = y_n + h\sum_{i=1}^{s} b_i k_i \tag{1.4}$$

where

$$k_i = f\left(x_n + c_i h, (1-v)y_n + v_i y_{n+1} + h\sum x_{ij} k_j\right), \text{ i= 2 (1) s}$$

and

$$c_i = v_i + \sum_{j=1}^{i=1} x_{ij}$$

Muir and Adams(2001) also developed Mono-implicit Runge-Kutta-Nystron method with application to boundary value ordinary differential equation, while Muir and Enright (1987) obtained relationships among some classes of implicit Runge-Kutta method and their stability functions.

1.1 STATEMENT OF THE PROBLEM

Okunbor(1987) developed implicit rational Runge – Kutta Schemes of ODE with large response characteristics. The schemes were found to be A-stable, $A(\alpha)$ - stable and L – stable. In the same vein, Odekunle (2001) discovered a set of semi-implicit rational Runge – Kutta schemes that were used in the solution of some stiff initial value problems. This idea was further expanded by Ademiluyi et al. (2002) to develop the class of implicit rational Runge – Kutta schemes which were found to perform very well in stiff systems though highly inadequate when applied to the problems with points of singularity. The challenge motivated Odekunle et al (2004) to come up with a class of inverse Runge – Kutta schemes which were found to efficiently solve problems with singularity point. However, poor performance in solving problems with non-singularity point was a quick fall of the scheme.

The need to have scheme(s) with high stability, few function evaluations motivated the study of developing a class of inverse mono-implicit Runge – Kutta Schemes. There are also some basic reasons why implicit Runge-Kutta methods are preferred to explicit ones. They are as follows:

(a) Higher orders of accuracy can be obtained than for explicit methods

(b) For linear systems of differential equations, implicit method can be implemented explicitly.

(c) For stiff problems explicit methods are never satisfactory whereas some implicit methods are (d)Implicit methods are closely related to Rosenbrock methods

(e)The structure of certain high-order explicit methods can be derived directly from some related implicit methods

(f) Implicit methods have an algebraic nicety not possessed by explicit methods to a certain group whereas the subset corresponding to explicit methods is only a semi group (Butcher, 1985).

1.2 OBJECTIVES OF THE STUDY

As long as life continues problems will never cease to occur and mathematics provides solution to the fast rate of changes in the globe. We can therefore summarize the aims and objectives of the work as follows.

1) To develop a class of inverse mono-implicit Runge – Kutta Schemes for singular problems
2) To investigate the consistency, convergence, error constant and stability of the schemes.
3) To illustrate the applicability of the numerical experiments with some specific examples.
4) To compare results with some existing schemes
5) To suggest any further research work for improvement where necessary.

1.3 DEFINITION OF SOME BASIC TERMS AND NOTATIONS

Definition 1.3.1

The method (1.2) is said to have order p if p is the largest integer for which

$$\mathrm{y}\left(x+h\right) - \mathrm{y}\left(x\right) - \mathrm{h}\,\phi\left(x, y(x), h\right) = \mathrm{O}\left(h^{p+1}\right) \quad (1.5)$$

Definition 1.3.2

The method (1.2) is said to be consistent with the IVP if

$\phi\ (x, y, o) \equiv \mathrm{f}(x, y)$

If the method (1.2) is consistent with the IVP then

$$\mathrm{y}(x+h) - \mathrm{y}(x) - \mathrm{h}\phi\ (x, y(x), h) = \mathrm{y}(x) + \mathrm{h}\, y'(x) + \mathrm{o}(h^2)$$

$$= \mathrm{y}(x) - \mathrm{h}\phi\ (x, y(x), 0) \quad (1.6)$$

$$= \mathrm{h}\, y'\ (x_n) - \mathrm{h}\, y'(x_n) + \mathrm{o}(h^2) = \mathrm{o}(h^2)$$

Since $y'\ (x) = \mathrm{f}\ (x, y(x)) = \phi(x, y(x), 0)$ by definition (1.3.2). Thus a consistent method has order of at least one.

Definition 1.3.3

The Taylor algorithm of order p is

$$\mathrm{y}(x+h) = \mathrm{y}(x) + \mathrm{h}\, y'(x) + \frac{h^2}{2!}\mathrm{y}^2(x) + \ldots + \frac{h^{\ p}}{p!}\mathrm{y}^p(x) \quad (1.7)$$

Definition 1.3.4

An R-stage Runge – Kutta method involves 'R' function evaluation per step. Each of the functions $\mathrm{k}_r\ (x, y, h)$, r = 1,2,…, R may be integrated as an approximation to the derivative $\mathrm{y}^1(x)$ and the function as a weighted mean of these approximations.

Definition 1.3.5

Differential equations whose exact solution has a term of the form e^{-ct}, where c is a large positive constant are referred to as stiff differential equations. This is usually only a part of the solution called the transient solution; the more important part of the solution is called the steady state solution. A transient portion of a stiff equation will rapidly decay to zero as t increases (Lambert,1973)

Definition 1.3.6

A differential equation or system of ODE is said to be autonomous if it does not explicitly contain the independent variable (usually denoted t)

Defination 1.3.7

The numerical scheme (3.1) is said to be convergent if it satisfies theorem (1.1) also if the numerical approximation y_{n+1} tends to exact solution $y(x_{n+1})$ of ODEs (1.1) as the step length h tends to zero.That is

$$\lim_{n\to\infty}\left[y(x_{n+1})-y_{n+1}\right]=0$$

Notations 1.3.8

Runge- Kutta method for higher orders involve tedious manipulations, as such, to reduce the cumbersome nature of the work, the following notations are adopted (Lambert, 1973)

$$f = f(x,y), \qquad f_x = \frac{\partial f(x,y)}{\partial x}, \qquad f_{xx} = \frac{\partial^2 f(x,y)}{\partial x^2} \tag{1.8}$$

$$f_{xy} = \frac{\partial^2 f(x,y)}{\partial x \partial y}, \qquad f_{yy} = \frac{\partial^2 f(x,y)}{\partial y^2}$$

for a function of two variables, f(x,y) the rate of change of the function can be due to change in either x or y. The derivative of f can be expressed in terms of partial derivatives (1.8) for the expression in the neighborhood of the point (a,b)

$$f(x,y) = f(a,b) + f_x(a,b)(x-a) + f_y(a,b)(y-b) + \frac{1}{2^1}\{f_{xx}(a,b)(x-a)^2 + 2f_{xy}(a,b)(x-a)(y-b) + f_{yy}(a,b)(y-b)^2\} + \ldots \tag{1.8.1}$$

Now expanding k's in Taylor series about the point (x,y), we have

$$k_1 = f$$

$$k_2 = f + ha_2 f + \frac{1}{2}h^2 a_z^2 G + o(h^3). \tag{1.8.2}$$

$$k_3 = f + ha_3 f + h\,h^2\left(a_2 b_{32} F f_y + \frac{1}{2}a_3^2 G\right) + o(h^3)\ _i \tag{1.8.3}$$

where

$F = f_x + ff_y$ and $G = f_{xx} + 2ff_{xy} + f^2 f_{yy}$

1.4 BASIC THEOREMS

Theorem 1.1 Henrici's theorem of consistency and convergence (Lambert, 1979)

i. Let the function $\varphi(x, y, h)$ be continuous jointly as a function of its three arguments in the region D defined by $x \in (a, b)$, $y \in (-\infty, \infty)$, $h \in \{0, h_0\}$, $h_0 > 0$

ii. Let $\varphi(x, y, h)$ satisfy a Lipschitz condition of the form $|\varphi(x, y^*, h) - \varphi(x, y, h)| \leq M|y^* - y|$ for all points (x, y^*, h), (x, y, h) in D then the method (1.2) is convergent if and only if it is consistent.

Theorem 1.2: Henrici's Existence theorem (Lambert 1979)

Let $f(x, y)$ be defined and continuous for all points (x, y) in the region D defined by $a \leq x \leq b$, $-\infty < y < \infty$, a and b are finite, and let there exist a constant L such that, for every x, y, y^* such that (x, y) and (x, y^*) are both in D,then $|f(x, y) - f(x, y^*)| \leq L|y - y^*|$

For all the methods, we shall consider conditions (i) and (ii) in theorem 1.1 as being satisfied if $f(x, y)$ satisfies the condition stated in theorem 1.2. For such methods, consistency is necessary and sufficient for convergence.

CHAPTER TWO

LITERATURE REVIEW

2.0 INTRODUCTION

Numerical methods for differential equations are vast in areas of applications such as kinetic theory, heat and matter transfer engineering, physics, electric transmission network, environmental, social sciences and many others.

Ademiluyi and Babatola (2000) developed implicit rational Runge-Kutta schemes that are capable of handling numerous problems with large response characteristics. They discovered that the classes of such schemes were stable but often are difficult to solve numerically because of their fast responding components which tends to control the total stability of the system. According to them, this behavior was first discovered by Curtis and Hirschfelder in 1952 when studying the motion governing mass spring system with different stiffness.

Burrage et al. (1994) developed order result for mono-implicit Runge – Kutta methods (MIRK) which concluded that the order of s-stage mono-implicit Runge – Kutta method is s+1. Cash et al (2000)developed Mono-implicit Runge-Kutta formulae for the numerical solution of second order non linear two point boundary value problem .This is implemented to avoid the costly matrix multiplication associated with other methods. The schemes hereby discussed in this research work should be used when encountering problems with singularity point.

2.1 Inverse Runge – Kutta schemes

In Odekunle et al (2004), the general s-stage inverse Runge – Kutta (R-K) scheme is defined as

$$Y_{n+1} = \frac{A}{1 + y_n \sum_{i=1}^{s} B_i Q_i} \qquad (2.1)$$

where A is a constant to be determined and

$$Q_i = hg\left(x_n + d_i h_i, Z_n + \sum_{j=1}^{s} b_{ij} Q_i \right), \quad i = 1(1)s \tag{2.2}$$

with

$$\mathrm{d}_i = \sum_{j=1}^{i} b_{ij} \; , i = 1(1)s \tag{2.3}$$

The Scheme (2.1) is said to be

i. Explicit if $b_{ij} = 0$, $j \geq i$

ii. Semi-implicit if $b_{ij} = 0$, $j > i$

iii. Implicit b $_{ij \neq 0}$ for at least one j>i

h is the step size and constraints (2.3) are imposed to ensure consistency of the method.

2.2 ERROR ESTIMATION FOR THE SCHEME

The local truncation error at x $_{n+1}$ of the general explicit one- step method (1.2) is defined to be T $_{n+1}$

where

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\phi[x_n, y(x_n), h] \tag{2.4}$$

And y(x) is the theoretical solution of the initial value problem (Lambert, 1979). According to Turner (1994) the local truncation error results from the use of first order Taylor expansion of the local solution passing through (x $_{n,}$ y_n). The error generated is given by $\frac{h^2 y''(\xi_n)}{2}$ for some point $\xi_n \in (x_{n,} x_{n-1})$ so that Euler's method is local. If we have the bound $|y''(x)| \leq M$ then the local truncation error is bounded by $\frac{Mh^2}{2}$.In the same spirit, Lotkin (1951) proposed the following error bound for the partial derivatives of $f(x, y)$ which would be used in obtaining an upper boundary of the local truncation error of the Runge – Kutta schemes. Suppose the differential system (1.1) satisfies

$$|f(x, y)| \leq M \tag{2.5}$$

and

$$\left|\frac{\partial f}{\partial y}\right| \leq L \tag{2.6}$$

For $a \leq x \leq b$ and $y < \infty$

Lotkin(1951) assumed the inequality

$$\left|\frac{\partial^{i+j} f(x, y)}{\partial x^i \partial y^j}\right| < L^{i+j} M^{i-j} \tag{2.7}$$

in order to get a tidy bound for various methods. The Lotkin error is merely a theoretical tool for comparing methods. According to Fatunla (1988), the global error e_{n+1} at the point x_{n+k} is the difference between the theoretical solution y (x_{n+k}) and the numerical solution y_{n+k}. That is,

$$e_{n+1}=y(x_{n+1})-y_{n+1} \tag{2.8}$$

The best check for the convergence of a numerical solution is to compare it with the exact solution of the equations involved. If the exact solution of our problem is unknown, we have to use other methods to convince ourselves about the convergence of the numerical solution.

A nice check for the numerical solution is to verify that this function converges for smaller time steps. If this function does converge to anything, we cannot trust the numerical solution nor the solution found by the finest grid mesh. Fatunla (1988), observed that the properties of the increment function Φ of the one step scheme are in general very crucial to its stability and convergence characteristics.

CHAPTER THREE

DERIVAYION OF A CLASS OF INVERSE MONO- IMPLICIT RUNGE-KUTTA SCHEMES

3.0 INTRODUCTION

We shall define the general s-stage inverse mono-implicit Runge-Kutta scheme as

$$y_{n+1} = \frac{A}{1 + y_n \sum_{i=1}^{s} B_i H_i} \tag{3.1}$$

Where A is a constant to be determined while

$$H_i = \left\{ x_n + d_i h, (1 - v_i) Z_n + V_i Z_{n+1} + h \sum_{j=i}^{i} x_{ij} H_j \right\} \tag{3.2i}$$

with

$$\left.\begin{aligned} d_i &= V_i + \sum_{j=1}^{i} x_{ij},\ i = i(1)s \\ g(x_n, Z_n) &= -Z_n^2 f(x_n, y_n) \\ Z_n &= -\frac{1}{y_n} \end{aligned}\right\} \tag{3.2ii}$$

where h is the step size or mesh size (or grid spacing) with (3.2ii) as constraints to ensure consistency of the scheme.

3.1 DERIVATION OF ONE-STAGE METHOD

Set s=1 in equation (3.1) and (3.2, i&ii) to have

$$y_{n+1} = \frac{A}{1 + y_n B_1 H_1} \tag{3.3}$$

where $H_1 = hg\{x_n + d_1 h, (1 - v_1) Z_n + v_1 Z_{n+1} + h x_{11} H_1\}$ (3.4)

$$= hg\{x_n + d_1 h, Z_n + v_1 (Z_{n+1} - Z_n) + h x_{11} H_1\}$$

with

$$g(x_{n,}Z_n) = -Z^2 f(x_{n,}y_n)$$

$$Z_n = \frac{-1}{y_n} \qquad (3.5)$$

We shall adopt binomial expansion on the right hand side of (3.3) and ignore any order higher than one to obtain.

$$y_{n+1} = A - Ay_n\beta_1 H_1 + \text{(higher order terms)} \qquad (3.6)$$

Applying Taylor's series of y_{n+1} about x_n gives

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2!}y''_n + \frac{h^3}{3!}y_n''' + 0(h^4) \qquad (3.6.1)$$

using equation (1.8) and adoption of the following notation

$$\Delta f = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}(y')$$

$$= \frac{\partial f}{\partial x} + f\frac{\partial f}{\partial y} = f_x + ff_y$$

$$y'_n = f(x_{n,}y_n) = f_n$$

$$y''_n = f_x + f_n f_y = \Delta f_n$$

$$y'''_n = \frac{\partial}{\partial x}(f_x + f_n f_y) + \frac{\partial}{\partial y}(f_x + f_n f_y)f_n \qquad (3.6.2)$$

$$= f_{xx} + f_n f_{xy} + f_x f_y + (f_{xy} + f_n f_{yy} + f_y^2)f_n$$

$$y'''_n = f_{xx} + 2f_n f_{xy} + f_x f_y + (f_{xy} + f_n f_{yy} + f_y^2)f_n$$

$$= \Delta^2 f_n + f_y \Delta f_n$$

Using (3.6.2) in (3.6.1) we have

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2!}\Delta f_n + \frac{h^2}{3!}(\Delta^2 f_n + f_y \Delta f_n) + 0(h^4) \qquad (3.7)$$

Now the Taylor series expansion of H_1 about (x_n, z_n) yields

$$H_1 = h\left\{ g_n + d_1 h g_n + \{v_1(Z_{n+1} - Z_n) + hx_{11}H_1\} g_n g_z + \frac{1}{2}\{(dh_1)^2 g_{xx} 2d_1 h((\{(v_1(Z_{n+1} - Z_n) + hx_{11}H_1)\} g_n g_{xz}))\} \right\}$$

$$+ \{V_1(Z_{n+1} - Z_n) + hx_{11}H_1\}^2 g_n^2 g_{zz} + \frac{1}{3!}\{(d_1 h)^3 g_{xxx} + 3(d_1 h)^2 \{(v_1(Z_{n+1} - Z_n) + hx_{11}H_1)\} g_n g_{xxz}\}$$

$$+3(d_1 h)\ \{(v_1(Z_{n+1} - Z_n) + hx_{11}H_1)^2\} g_n{}^2 g_{xzz} + \{(v_1(Z_{n+1} - Z_n) + hx_{11}H_1)^3 g_n{}^3 g_{xzz}\} + 0(h^4) \qquad (3.8)$$

Setting $d_1 = x_{11} + H_1$ in equation (3.8) yields

$$H_1 = hA_1 + h^2 A_2 \quad + h^3 A_3 + 0(h^4) \qquad (3.9)$$

where

$$A_1 = g_n$$

$$A_2 = d_1 \Delta g_n$$

$$A_3 = \frac{d_1^2}{2} \Delta^2 g_n \qquad (3.9.1)$$

$$\Delta g_n = g_x + g_n g_z$$

$$\Delta^2 g_n = g_{xx} + 2g_n g_{xx} + g_n{}^2 g_{zz}$$

$$\Delta^3 g_n = g_{xxx} + 3g_n g_{xxz} + 3g_n^2 g_{xzz} + g_n^3 g_{zzz}$$

In this analysis, we have assumed that $y(x) \in c^{\alpha}(a,b)$ *and* $g(z) \in c^n[a,b]$ and that

$$g_n = \frac{-f_n}{y_n^2}, \qquad g_n = \frac{-f_y}{y_n^2}, \qquad f_{y_y} = \frac{\partial f}{\partial y_y}, \qquad g_{xxx} = \frac{-f_{xxx}}{y_n^2}$$

$$g_x = \frac{-f_x}{y_n^2}, \qquad g_{xx} = \frac{-f_{xx}}{y_n^2}, \qquad g_{xz} = \frac{-2f_x}{y_n} + f_{xy}, \qquad g_{xzz} = \frac{-2f_{xx}}{y_n} + f_{xyy} \qquad (3.9.2)$$

$$g_z = \frac{-2f_n}{y_n} + f_y \quad , \quad g_{xz} = -2f_n - y_n^2 - y_n^2 f_{yy} \quad , \quad g_{zzz} = 4y_n^2 f_y + 6y_n^2 f_{yy} + 4y_n f_{yyy}$$

Substituting (3.9) into (3.6) we have

y_{n+1}= A-A$y_n \beta_1 hA_1$ -A$y_n h^2 \beta_1 A_2$ ……(neglecting higher terms) (3.10)

Comparing (3.10) with Taylor expansion of y_{n+1} about (x_n,y_n) for the powers of h

$$\begin{aligned}
&A = y_n \\
&-Ay_n\beta_1 A_1 = f_n \\
&-Ay_n\beta_1 g_n = f_n \, where A_1 = g_n \\
&-Ay_n\beta_1 A_2 = \frac{\Delta f_n}{2} \\
&-Ay_n\beta_1 d_1 \Delta g_n = \frac{\Delta f_n}{2} \, where A_2 = d_1 \Delta g_n
\end{aligned} \tag{3.11}$$

Solving (3.11) simultaneously and using the values ofA_1, A_2 we arrive at

A=y_n , β_1=1, d_1=1/2

Therefore equation (3.1) becomes

$$\begin{aligned}
&y_{n+1} = \frac{y_n}{1 + y_n H_1} \\
&where \\
&H_1 = hg\left(x_n + \frac{h}{2}, z_n + \frac{1}{4}(z_{n+1} - z_n) + \frac{1}{4}hH_1\right); d_1 = v_1 + x_{11} \quad when \quad d_1 = 1/2, v_1 = 1/4, x_{11} = 1/4
\end{aligned} \tag{3.12}$$

3.2 TWO-STAGE METHODS

Set s=r=2 in equations (3.1), (3.2i and ii) we have

$$y_{n+1} = \frac{A}{1 + y_n \sum_{i=1}^{2} \beta_i H_i}. \tag{3.2.1}$$

$$= \frac{A}{1 + y_n(\beta_1 H_1 + \beta_2 H_2)} \tag{3.2.2}$$

So that

H_1=gh(x_n+d_1h,z_n+v_1(z_{n+1}-z_n)+h$x_{11}$$H_1$) and

$H_2=gh(x_n+d_2h,z_n+v_2(z_{n+1}-z_n)+h(x_{21}H_1+x_{22}H_2)$

where A is a constant to be determined and (3.2.2) can be written in the form

$$y_{n+1} = A[1 + y_n(\beta_1 H_1 + \beta_2 H_2)]^{-1}$$

Expanding using binomial expansion method and considering only two terms

$$y_{n+1} = A - Ay_n\beta_1 H_1 - Ay_n\beta_2 H_2 \qquad (3.2.3)$$

Adopting Taylor series expansion for H_2 about (x_n,z_n) and application of notations

$$H_2 = h\begin{bmatrix} g_n + d_2hg_x + h(x_{21}H_1 + x_{22}H_2)g_ng_z + \frac{1}{2}\left(d_2^2h^2g_{xx} + 2d_2h^2(x_{21}H_1 + x_{22}H_2)g_ng_{xz} + h^2(x_{21}H_1 + x_{22}H_2)^2g_n^2g_{zz}\right) \\ + \frac{1}{3!}\left(d_2^3h^3g_{xxx} + 3d_2^2h^3(x_{21}H_1 + x_{22}H_2)g_ng_{xxz} + 3d_2h^3(x_{21}H_1 + x_{22}H_2)^2g_n^2g_{xzz} + h^3(x_{21}H_1 + x_{22}H_2)^3g_n^3g_{zzz}\right) \end{bmatrix}$$

Let $d_2=x_{21}H_1+x_{22}H_2$ (3.2.3) becomes

$$H_2 = h\begin{bmatrix} g_n + d_2hg_n + hd_2g_ng_z + \frac{1}{2}\left(d_2^2h^2g_{xx} + 2d_2^2h^2g_ng_{xz} + d_2^2h^2g_n^2g_{zz}\right) \\ + \frac{1}{6}\left(d_2^3h^3g_{xxx} + 3d_2^3h^3g_ng_{xxz} + 3d_2^3h^3g_n^2 + d_2^3h^3g_n^3g_{zzz}\right) \end{bmatrix} \qquad (3.2.4)$$

$$+ 0(h^4)$$

$$H_2 = hg_n + h^2d_2\Delta g_n + \frac{1}{2}d_2^2h^3\Delta^2 g_n + \frac{1}{6}h^4d_2^3\Delta^3 g_n \qquad (3.2.5)$$

Substituting (3.2.5) and (3.8) into (3.2.3) gives the following results

$$y_{n+1} = A - hAy_ng_n(\beta_1 + \beta_2) - h^2Ay_n\Delta g_n(\beta_1 d_1 + \beta_2 d_2) - \frac{h^3}{2}Ay_n\Delta^2 g_n(\beta_1 d_1^2 + \beta_2 d_2^2) \qquad (3.2.6)$$

Comparing the coefficients of h in equations (3.2.6) with the Taylor's expansion of y_{n+1} about

(x_n, y_n) in equation (3.7) we obtain

$$\begin{aligned} &A = y_n \\ &-Ay_ng_n(\beta_1 + \beta_2) = f_n \\ &-Ay_n\Delta g_n(\beta_1 d_1 + \beta_2 d_2) = \frac{\Delta f_n}{2} \end{aligned} \qquad (3.2.7)$$

These equations can be solved by adopting the same steps as for one stage schemes, that is

$$\beta_1 + \beta_2 = 1$$

$$\beta_1 d_1 + \beta_2 d_2 = \frac{1}{2}$$

We have two equations in four unknown's i.e. $\beta_{1,}\beta_2, d_1$ *and* $.d_2$

(i) choosing $\beta_1 = \beta_2 = \frac{1}{2}, d_1 = d_2 = \frac{1}{2}$

Substituting this into equation (3.2.2) we have

$$y_{n+1} = \frac{y_n}{1 + \frac{1}{2} y_n (H_1 + H_2)} \tag{3.2.8}$$

This is a two stage inverse mono-implicit Runge-Kutta scheme where

$$H_1 = hg(x_n + \frac{1}{2}h, z_n + \frac{1}{4}(z_{n-1} - z_n) + \frac{1}{4}hH_1)$$
$$H_2 = hg(x_n + \frac{1}{2}h, z_n + \frac{1}{4}(z_{n+1} - z_n) + \frac{1}{8}h(H_1 + H_2))$$

(ii) Choosing $\beta_1 = \frac{1}{4}, \beta_2 = \frac{3}{4}, d_1 = 1, d_2 = \frac{1}{3}$ so that

$$y_{n+1} = \frac{y_n}{1 + \frac{1}{4} y_n (H_1 + 3H_2)} \tag{3.2.9}$$

where

$$H_1 = hg(x_n + h, z_n + \frac{1}{2}(z_{n+1} - z_n) + \frac{1}{2}hH_1)$$
$$H_2 = hg(x_n + \frac{1}{3}h, z_n + \frac{1}{9}(z_{n+1} - z_n) + \frac{1}{9}h(H_1 + H_2))$$

(iii) $\beta_1 = \frac{1}{3}, \beta_2 = \frac{2}{3}, d_1 = \frac{1}{2}, d_2 = \frac{1}{2}$

So that

$$y_{n+1} = \frac{y_n}{1 + \frac{1}{3} y_n (H_1 + 2H_2)}$$

where (3.2.10)

$$H_1 = hg(x_n + \frac{1}{2}h, z_n + \frac{1}{3}(z_{n+1} - z_n) + \frac{1}{6}hH_1)$$
$$H_2 = hg(x_n + \frac{1}{2}h, z_n + \frac{1}{6}(z_{n+1} - z_n) + \frac{1}{6}h(H_1 + H_2))$$

Where $d_1 = v_1 + x_{11}$ *and*. $d_2 = v_2 + x_{21} + x_{22}$

3.3 ERROR ANALYSIS

There are basically three major classes of errors of numerical approximation techniques for ODEs, these are: discretization, truncation and round off.

Round off error is the class of error introduced by the computing devices. It is expressed mathematically as:

$$\gamma_{n+1} = y_{n+1} - p_{n+1} \tag{3.2.11}$$

where y_{n+1} is the expected solution of the difference equation equivalent of (1.1) and p_{n+1} is the computer output at the (n+1)th iteration This class of error is not amendable to analysis due to inevitable loss of accuracy associated with it but can be controlled by employing double precision arithmetic (Ademiluyi, et al. 2000)

Secondly truncation error is the error introduced as a result of ignoring some of the higher terms of the power series expansion during the development of the new scheme. Mathematically, truncation error is defined as the amount by which the true solution $y(x_{n+1})$ of the differential equation (1.1) fails to satisfy the difference equation (3.1). This can be defined in the form

$$T_{n+1} = y(x_{n+1}) - y_{n+1} \tag{3.2.12}$$

$$T_{n+1} = y(x_{n+1}) - \frac{A}{1 + y(x_n)\sum_{r=1}^{s}\beta_r H_r} \tag{3.2.13}$$

By the Taylor series expansion of $y(x_{n+1})$ and H_r about $(x_n, y(x_n))$, the local truncation error for the one stage schemes (3.12) of order two is by Lambert (1979)

$$T_{n+1} = \left[(\Delta^2 f_n + f_y \Delta f_n)\left(\frac{1}{6} - \frac{1}{2}\beta_i d_i^2\right) - \frac{1}{y_n}\beta_i d_i \left(\frac{2 f_n}{y_n} - f_y f_n\right) \right] h^3 \tag{3.2.14}$$

Using Lotkin (1951) technique, the bound of the local truncation error of (3.2.14) is found to be

$$|T_{n+1}| = \left[6|L_1|\left|N^2M - 4|K_2|\right|M\left(\frac{N^2}{2} - MN + M^3\right)\right| \right] h^3 \tag{3.2.15}$$

Where

$$\frac{\partial^{i+j}}{\partial x^i \partial y^j} f(x,y) < \frac{L^{i+j}}{M^{j-1}} \quad \text{i,j=0(1)n} \tag{3.2.16}$$

$$K_1 = \frac{1}{6} - \frac{1}{2}\beta_1 d_1^2$$

and

$$K_2 = \frac{\beta_1 d_1}{y_n}$$

Lastly, discretization error is the error introduced by the conversion of the differential equation (1.1) into the equivalent difference equation (3.1). Mathematically, the discretization error associated with the formula (3.1) is the difference between the exact solution $y(x_{n+1})$ and the numerical solution y_{n+1} generated by solving at point x_{n+1}. This can be estimated from the equation

$$e_{n+1} = y_{n+1} - y(x_{n+1}) \tag{3.2.17}$$

3.4 STABILITY ANALYSIS

In exploring the suitability and stability of the schemes, we subjected equation (3.1) to Dahlquist (1963) Stability test using the initial value problem

$y' = \lambda y$, $y(x_0) = y_0$

to obtain the difference equation

$$y_{n+1} = \frac{y_n}{1 - h\beta^T (1 + hB)^{-1} e} \tag{3.2.19}$$

where

$$\beta^T = (\beta_1, \beta_2, \ldots, \beta_r)$$

and

$$e = [1,1,\ldots,1]^T$$

The formula (3.2.19) is the first order difference equation which can be written in the form

$$y_{n+1} = \mu(z) y_n$$

where

$$z = \lambda h$$

and

$$\mu(z) = \frac{1}{1 - h\beta^T (1 + hB)^{-1} e} \qquad (3.2.20)$$

The parameters (b_{ij}, d_i ,H_i) in the scheme (3.1) are chosen to ensure that $\mu(z)$ is Pades approximation to e^h .The resulting schemes were A-stable because $|\mu(z)| < 1, A(\alpha)$ -stable because arg($\mu(z)$) lies between $[0, \pi/2]$ and L-stable because $\lim_{n\to\infty} |\mu(z)| = 0$ (Ehle,1969).

CHAPTER FOUR

NUMERICAL EXPERIMENTS AND RESULTS

4.0 INTRODUCTION

In this chapter, we shall by way of illustration and computations, test the efficiency and stability of our methods by comparing it with classical Runge-Kutta method (R-K) and rational mono-implicit method using the following example

4.1 Numerical example

$y' = 1 + y^2$, y (0) =1, $0 \le x \le 1$

whose theoretical solution $y(x) = \tan(x + \frac{\pi}{4})$ is unbounded at $x = \frac{\pi}{4}$.In the neighborhood of this singularity, the solution becomes unbounded

4.2 DISCUSSION OF RESULTS

TABLE 4.1.1 ERRORS OF EXAMPLE 4.1 at x=0.1

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Eqn.3.2.8)	Error using Inverse Mono-implicit scheme(Eqn3.2.9)
0.100000	$2.2304887\text{x}10^{-1}$	$3.3031975\text{x}10^{-3}$	$3.3031975\text{x}10^{-3}$	$2.5228596\text{x}10^{-1}$
0.050000	$1.1769326\text{x}10^{-1}$	$1.7629233\text{x}10^{-3}$	$2.0415904\text{x}10^{-1}$	$2.4558547\text{x}10^{-3}$
0.025000	$6.0557170\text{x}10^{-2}$	$9.1120508\text{x}10^{-4}$	$1.8555033\text{x}10^{-1}$	$1.0875925\text{x}10^{-3}$
0.012500	$3.0729853\text{x}10^{-2}$	$4.6331403\text{x}10^{-4}$	$1.7723542\text{x}10^{-1}$	$5.0780108\text{x}10^{-4}$
0.006250	$1.5480864\text{x}10^{-2}$	$2.3363777\text{x}10^{-4}$	$1.7329201\text{x}10^{-1}$	$2.4474757\text{x}10^{-4}$
0.003125	$7.7698013\text{x}10^{-3}$	$1.1733708\text{x}10^{-3}$	$1.7137026\text{x}10^{-1}$	$1.2005361\text{x}10^{-4}$

TABLE 4.1.2 ERRORS OF EXAMPLE 4.1 at x=0.2

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Equation3.2.8)	Error using Inverse Mono-implicit scheme(Eqn.3.2.9)
0.100000	2.8544723×10^{-1}	7.9658416×10^{-3}	7.9658416×10^{-3}	5.9348592×10^{-1}
0.050000	1.5240971×10^{-1}	4.2245829×10^{-3}	4.8540290×10^{-1}	5.7501831×10^{-3}
0.025000	7.8928903×10^{-2}	2.1768758×10^{-3}	4.4242561×10^{-1}	2.5651669×10^{-3}
0.012500	4.0188779×10^{-2}	1.1051648×10^{-3}	4.2301965×10^{-1}	1.2031412×10^{-3}
0.006250	2.0281261×10^{-2}	5.5686090×10^{-4}	4.1377420×10^{-1}	5.8139820×10^{-4}
0.003125	1.0188076×10^{-2}	2.7953199×10^{-4}	4.0925901×10^{-1}	2.8560907×10^{-4}

TABLE 4.1.3 ERRORS OF EXAMPLE 4.1 at x=0.3

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Equation3.2.8)	Error using Inverse Mono-implicit scheme(Equation3.2.9)
0.100000	3.8725644×10^{-1}	1.5355481×10^{-2}	1.5355481×10^{-2}	1.1693962×10^{0}
0.050000	2.0996810×10^{-1}	8.1029324×10^{-3}	9.5042304×10^{-1}	1.0826984×10^{-2}
0.025000	1.0967301×10^{-1}	4.1663069×10^{-3}	8.6343045×10^{-1}	4.8582177×10^{-3}
0.012500	5.6097543×10^{-2}	2.1121497×10^{-3}	8.2416277×10^{-1}	2.2869636×10^{-3}
0.006250	2.8376089×10^{-2}	1.0636120×10^{-3}	8.0545806×10^{-1}	1.1074332×10^{-3}
0.003125	1.4271416×10^{-2}	5.3373656×10^{-4}	7.9632413×10^{-1}	5.4464576×10^{-4}

TABLE 4.1.4 ERRORS OF EXAMPLE 4.1 at x=0.4

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Equation3.2.8)	Error using Inverse Mono-implicit scheme(Equation3.2.9)
0.100000	$5.6913446x10^{-1}$	$2.8775163x10^{-2}$	$2.8775163x10^{-2}$	$2.4415845x10^{0}$
0.050000	$3.1521136x10^{-1}$	$1.5128718x10^{-2}$	$1.9263915x10^{0}$	$1.9938283x10^{-2}$
0.025000	$1.6664830x10^{-1}$	$7.7634405x10^{-3}$	$1.7287842x10^{0}$	$8.9854629x10^{-3}$
0.012500	$8.5810818x10^{-2}$	$3.9333716x10^{-3}$	$1.6408202x10^{0}$	$4.2415630x10^{-3}$
0.006250	$4.3554669x10^{-2}$	$1.9798838x10^{-3}$	$1.5991829x10^{0}$	$2.0571521x10^{-3}$
0.003125	$2.194351x10^{-2}$	$9.9331135x10^{-4}$	$1.5789116x10^{0}$	$1.0126054x10^{-3}$

TABLE 4.1.5 ERRORS OF SOLUTION TO EXAMPLE 4.1 at x=0.5

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Eqn3.2.8)	Error using Inverse Mono-implicit scheme(equation3.2.9)
0.100000	$9.4285751x10^{-1}$	$5.8131648x10^{-2}$	$5.8131648x10^{-2}$	$7.0337613x10^{0}$
0.050000	$5.3331324x10^{-1}$	$3.0503066x10^{-2}$	$5.0135774x10^{0}$	$3.9880081x10^{-2}$
0.025000	$2.9068986x10^{-1}$	$1.5639537x10^{-2}$	$4.3350201x10^{0}$	$1.8016920x10^{-2}$
0.012500	$1.5126312x10^{-1}$	$7.9206719x10^{-3}$	$4.0476437x10^{0}$	$8.5195927x10^{-3}$
0.006250	$7.7206051x10^{-2}$	$3.9861352x10^{-3}$	$3.9145825x10^{0}$	$4.1363095x10^{-3}$
0.003125	$3.9009337x10^{-2}$	$1.9996376x10^{-3}$	$3.8504725x10^{0}$	$2.0372406x10^{-3}$

TABLE 4.1.6 ERRORS OF SOLUTION TO EXAMPLE **4.1 at x=0.6**

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Eqn3.2.8)	Error using Inverse Mono-implicit scheme(Eqn3.2.9)
0.100000	1.9196731×10^{0}	1.4582976×10^{-1}	1.4582976×10^{-1}	1.8570690×10^{2}
0.050000	1.1621993×10^{0}	7.6612277×10^{-2}	5.3755019×10^{1}	1.0033832×10^{-1}
0.025000	6.4928314×10^{-1}	$3.93122455 \times 10^{-2}$	3.1502374×10^{1}	4.5299381×10^{-2}
0.012500	3.4488841×10^{-1}	1.9919009×10^{-2}	2.5866552×10^{1}	2.1424019×10^{-2}
0.006250	1.7799921×10^{-1}	1.0026806×10^{-2}	2.3689450×10^{1}	1.0403912×10^{-2}
0.003125	9.0457082×10^{-2}	5.0305223×10^{-3}	2.2719983×10^{1}	5.1251107×10^{-3}

TABLE 4.1.7 ERRORS OF SOLUTION TO EXAMPLE 4.1 at x=0.7

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Eqn3.2.8)	Error using Inverse Mono-implicit scheme(Eqn.3.2.9)
0.100000	6.2220762×10^{0}	6.9035756×10^{-1}	6.9035756×10^{-1}	2.1136806×10^{1}
0.050000	4.3275766×10^{0}	3.6728775×10^{-1}	2.3681636×10^{1}	4.9760563×10^{-1}
0.025000	2.6590535×10^{0}	1.8979121×10^{-1}	2.5358759×10^{1}	2.2218702×10^{-1}
0.012500	1.4992603×10^{0}	9.6519501×10^{-2}	2.6342695×10^{1}	1.0460896×10^{-1}
0.006250	8.0063994×10^{-1}	4.8677872×10^{-2}	2.6879096×10^{1}	5.0699153×10^{-2}
0.003125	4.1441551×10^{-1}	2.4445325×10^{-2}	2.7159652×10^{1}	2.4952530×10^{-2}

TABLE 4.1.8 ERRORS OF SOLUTION TO EXAMPLE 4.1 at x=0.75

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Eqn.3.2.8)	Error using Inverse Mono-implicit scheme(Eqn.3.2.9)
0.050000	1.6013326×10^{1}	2.0622140×10^{0}	3.5675168×10^{1}	3.1028397×10^{0}
0.025000	1.1637436×10^{1}	1.0875780×10^{0}	3.6290739×10^{1}	1.3379067×10^{0}
0.012500	7.3713184×10^{0}	5.5915548×10^{-1}	3.6624056×10^{1}	6.2082586×10^{-1}
0.006250	4.2410997×10^{0}	2.8359472×10^{-1}	3.6797901×10^{1}	2.9891337×10^{-1}
0.003125	2.2926489×10^{0}	1.4282608×10^{-1}	3.6886732×10^{1}	1.4665891×10^{-1}

TABLE 4.1.9 ERRORS OF SOLUTION TO EXAMPLE 4.1 at x=8

h	Error using classical R-K scheme	Error using Mono-implicit Rational R-K scheme	Error using Inverse Mono-implicit scheme(Eqn3.2.8)	Error using Inverse Mono-implicit scheme(Eqn.3.2.9)
0.100000	1.8048855×10^{2}	4.0393499×10^{1}	4.0393499×10^{1}	6.3717115×10^{1}
0.050000	1.2954353×10^{3}	1.6201348×10^{1}	6.3136263×10^{1}	1.3271623×10^{1}
0.025000	2.2650890×10^{5}	7.3754891×10^{0}	6.2822564×10^{1}	6.7751491×10^{0}
0.012500	2.6236124×10^{19}	3.5301724×10^{0}	6.2658520×10^{1}	3.3976176×10^{0}

DISCUSSION OF RESULTS

The results displayed in table 4.1.1to 4.1.9 above, compares the results of our newly developed schemes (3.2.8) and (3.2.9) with that obtained by Abubakar (2009) of order two was applied to solve our example and the classical R-K scheme of order four. The new scheme (3.2.8) was not good enough when compared with the classical R-K and the Mono-implicit rational Runge-Kutta scheme, but the new scheme (3.2.9) was at an advantage when compared to the Classical R-K and can compute favorably with the Mono-implicit rational R-k scheme. We could also clearly see that the new scheme (3.2.9) showed weakness with higher grid size but at a lower grid size ,the Mono-implicit R-K scheme and the new scheme (3.2.9) were similar in all manners. The Mono-implicit rational R-K scheme and the new scheme (3.2.9) were competing with high degree of accuracy and convergence till after 7^{th} iteration when scheme (3.2.9) picked up slowly after a jump at the singularity point while the rational Mono-implicit R-K could not pick up again. Classical R-K showed greater improvement with smaller mesh size. At higher step size the Mono-implicit rational R-K scheme performs better than the two new schemes (3.2. 8) and (3.2.9).

It can be seen from tables (4.1.1) to (4.1.9) that the error of equations (3.2.8) and (3.2.9) approaches zero as h increases which imply that the schemes are consistent and convergent having also satisfied the necessary and sufficient conditions in theorem (1.1).

CHAPTER 5

SUMMARYAND CONCLUSION

5.0 INTRODUCTION

The project work developed a class of Inverse Mono-implicit Runge-Kutta schemes for first order differential equations. The work used Inverse Runge-Kutta method and the general Mono-implicit Runge-Kutta methods. The results obtained from the schemes when used to solve numerical problem were found to compete favorably with Mono-implicit rational Runge-Kutta and does better than classical Runge-Kutta schemes

5.1. SUMMARY

The analysis carried out from the results above shows that the new scheme (3.2.9) and rational mono-implicit Runge-Kutta scheme was found to perform very well with high level of accuracy and convergence even with large step size. On the other hand Mono-implicit rational Runge –kutta scheme was also found to be very good even with large step sizes while classical R-K was found to be good with small step size. Both mono-implicit rational R-K schemes and the new scheme (3.2.9) were found to compete favorably with classical R-K before the point of singularity when used to solve the same numerical problem.

5.2 CONCLUSION

The new scheme (3.2.9) recovered sluggishly at point of singularity, the results obtained from the scheme when used to solve some numerical problem were found to compete favorably with rational mono-implicit Runge-Kutta scheme and classical R-k schemes. Classical R-K scheme and rational Mono-implicit R-K scheme are weak at the point of singularity while the new scheme (3.2.9) showed a sign of recovery at this same point, thus being capable of handling autonomous and non-autonomous differential equations be it linear or non-linear with or without singularity point. However, Mono-implicit Rational Runge-Kutta and classical R-k schemes can only handle non-singularity cases very well. Hence the new scheme (3.2.9) is of advantage in this regard.

5.3 RECOMMENDATION

Based on this research, the schemes (Inverse Mono-implicit Runge-Kutta for first order differential equation) was found to be A-stable and effective in handling singular problems. Thus we recommend that the schemes developed be tried on second order differentia equation

REFERENCES

Abubakar,F.U (2009): *Rational Mono-implicit Runge- Kutta Methods*. M.Sc. Thesis, Federal University of Technology, Yola. (Unpublished)

Ademiliyu, R.A.and Babalola.P.O. (2000): *Implicit rational Runge-Kutta Schemes* for approximation of *Ordinary Differential equation with large response characteristics*. Akure: Federal University of Technology, Industrial Mathematics and Computer science Department.

Ademiluyi, R.A, Babatola P.O and Odekunle,M.R (2002): *Rationalized implicit Runge-Kutta schemes for the integration of stiff ordinary differential equations*. BJPAS 2(2) pg16-23

Burrage, K.Chipman, F. &Muir, P. (1994): *Order results for Mono-Implicit Runge-kutta Methods*. SIAM Journal of Numerical analysis vol .3.pp 877-895

Butcher, J.C. (1985): *The Numerical Analysis of Ordinary Differential Equations. Runge-Kutta and General Linear Methods*. A Wiley-interscience NewYork

Cash J.R, Garcia M.P and Moore D.R. (2000): *Mono-implicit Runge-Kutta Formulae for the Numerical Solution of Second Order Non-linear Two point Boundary Value Problem.* Oxford: Clarendon Press 107-125

Cash, J.R.and Singhal, A. (1982*): Mono-Implicit Runge-Kutta Formulae for the numerical Integration of stiff Differential system.* IMA journal of numerical analysis.Vol.2. pp 213-240

Cooper, G.J.and Vignesvaran,R (1993): *Some schemes for implementation of implicit Runge-Kutta methods,*Journal of computational and applied mathematics **45**, 213-225

Dahlquist, G. (1963): *A special Stability Problem for Linear Multistep Methods,* BIT **3** pp27-43

Ehle B.L (1969): *On Pade's Approximation, the Exponential Function and A-stable methods for the Numerical Solution of initial Value Problems.* University of Waterloo, Department of Applied Analysis and Computer Science. Research Rep.CSRR 2010

Fatunla, O.S. (1988): *Numerical methods for initial value problems in ODEs* New York Academic Press Inc

Lambert, J.D (1973): *Computational methods in ordinary differential equations*, NewYork; John Wiley.

Lambert, J.D. (1979): *Safe point Methods for Separably Stiff Systems of ODEs*. Report No. **NA/31.** Department of Mathematics, University of Dundee. UK

Lotkin, M.(1951): *On the Accuracy of R-K Methods*.MTAC 5,pg128-132

Muir,P.H.& Adams,M.(2001): *Mono-Implicit Runge-Kutta Nystrom Method with Application to Boundry Value ODEs*. BIT 4, pg776-779.

Muir, P.H. & Enright, W.H. (1987): *Relationship Among some classes of implicit Runge-Kutta method and Their stability functions.* BIT 27, pg 405-425

Odekunle, M.R., Oye, N.D., Adee, S.O& Ademiluyi, R.A. (2004): *A class of inverse Runge-Kutta schemes for the numerical integration of singular Problems,* ELSEVIER. Journal of Applied Mathematics and computing. Vol.158 pp 149-158

Odekunle,M.R (2001):*Some Semi-Implicit Rational Runge-Kutta Schemes*.Bagale Journal Of Pure and Applied Sciences, Vol 1.No.1, pg 11-14

Okunbor D.I. (1987) *Explicit Rational R-K Schemes for Stiff System of Ordinary Differential Equations* M.Sc Thesis, University of Benin, Benin City, Nigeria, (unpublished).

Turner,R.B (1994): A New Approach to Error Estimation for General Linear Methods Numerical Mathematics pp314-333

Voss, D.A. & Muir, P.H. (1999): *Mono-Implicit Runge-Kutta schemes for the parallel solution of initial value ODEs.* Journal of computational and applied Mathematics vol.102. pp235-53

Wikipedia (2007): Runge-Kutta methods.[Internet] retrieved June26, 2007 from http:wikipedia.org/wiki/Runge-Kutta methods