ON THE ACCURACY OF THE BACKWARD DIFFERENCE FORMULA OF ORDER FOUR AND RUNGE-KUTTA OF ORDER FOUR IN SOLVING A FIRST ORDER NON-LINEAR ORDINARY DIFFERENTIAL EQUATION

A Thesis
Presented to the College of
Higher Degrees and Research
Kampala International University
Kampala, Uganda

In Partial Fulfillment of the Requirements for the Degree of
Master of science in Applied Mathematics (MSC-AM)

BY

NWEKE IJEOMA ANIE
MSC-AM/45105/143/DF

July, 2016

DECLARATION A

"This Thesis is my original work and has not been presented for a Degree or any other academic award in any University or Institution of learning."

Name and Signature of Candidate Ijeoma
NWEKE IJEOMA ANIE
MSC-AM/45105/143/DF

Date 20/7/2016

DECLARATION B

"I confirm that the work reported in this Thesis was carried out by the candidate under my supervision".

Dr. Godwin Kakuba

Name and Signature of Supervisor

19/7/2016

Date

DEDICATION

This work is dedicated to the Almighty God. And to my children Raisat, Nabih and Zaimat.

ACKNOWLEDGEMENT

I am indeed grateful to my supervisor Dr. Godwin Kakuba, who tirelessly went through my work and inspired me to dig deeper into the core of the matter. His positive criticism, patience and understanding greatly assisted me.

I wish to thank Dr. Sudi Balimutajjo, Dr. Mustapha Mundu, Dr. Maria Mbatudde and Engr. A.E. Talabi for their advice and guidance throughout the course.

Thanks also goes to the Federal College of Education, Kano, Nigeria for their sponsorship throughout my Masters Programme.

Contents

List of Tables

List of Figures

ABSTRACT

In this study, the researcher investigated if the Runge-Kutta of order 4 method (RK_4) is as accurate as the Backward Difference Formula of order 4 (BDF_4) in solving a 1st order non-linear Ordinary Differential Equation? Two 1st order non-linear ODE were tested to analyse this problem. The RK_4 proved to be more accurate than the BDF_4 because the RK_4 can generate it's values without depending on the Analytic method and it showed very slight deviation from the Analytic method when it was rounded off to 9 decimal places, otherwise it was equal to the Analytic method when rounded off to 7 decimal places in equation 1 and 6 decimal places in equation 2. It shows very slight deviation from the Analytic method while the BDF_4, despite it's using starting values from the Analytic method showed large deviation from the Analytic method even when applied to the two tested problems. Even then, BDF_4 is more prefered than the RK_4 in solving stiff problems because it is A-Stable and converges easily.

1 Introduction

1.1 Background

The nature of problems commonly occurring in different fields such as engineering, marketing and medical are modeled using differential equations that require either analytical or numerical solutions for better interpretation or understanding of such problems.

Analytical solutions are very accurate and efficient. In many instances however, most of the mathematical models, i.e., ordinary differential equations (ODEs) lack analytical solutions. In these cases Numerical Analysis helps us to derive forms of approximations to the solution of such problems, which practically suffices in fields such as engineering see [11, 17].

Slightly over One Hundred years ago, in 1895, Carl Runge published the first Runge-Kutta method, which extended the approximation method of Euler to a more robust scheme of greater accuracy. Euler's idea followed a Riemann step-wise approach where a solution within each step is related to a constant rate of change determined at the beginning of each step. The idea of Runge was not based on this unsymmetrical and relatively inaccurate Riemann rule, but on such improved formulas as the midpoint and trapezoidal rules. In 1905, Martin Kutta described the popular fourth-order Runge-Kutta method inorder to increase accuracy and efficiency.

On the other hand, there are other numerical methods that also extend the Euler idea to approximate the rate of change of a function basing on previously computed values. Such methods include the linear multi-step backward differentiation formula (BDF). With this formula each subsequent iteration corresponds to a higher order BDF.

There exist the RK_1, RK_2, RK_3 and BDF_1, BDF_2, BDF_3. But in this study, emphasis was made on the RK_4, and BDF_4, inorder to acquire better accuracies, because it is known that the higher the order of any numerical method, the better the accuracy.

Generally, higher order methods compared to lower order methods, produce more accurate approximations of ODE solutions at lower computational costs. Additionally, the working of RK and BDFs methods are different, in this context, it is appropriate to carry out a comparative analysis of the accuracy of results from both methods at some high order.

1.2 Research Question

Are the Backward Difference Formula of order 4 and Runge-Kutta of order 4 method equally acccurate in solving a 1st order non-linear Ordinary Differential Equation?

1.3 Objectives

1.3.1 Main objective

To compare and contrast the accuracies of a Backward Difference Formula (BDF) of order 4 and Runge Kutta of order 4

1.3.2 Specific objectives

(i) to analyse the accuracy of a Runge Kutta of order 4.

(ii) to analyse the accuracy of a BDF of order 4.

(iii) To compare and contrast the accuracies of the solutions of BDF of order 4 and Runge-Kutta of order 4.

1.4 Siginificance of the study

As science and engineering becomes more and more sophisticated, there is an increasing need for practitioners to develop skills using numerical analysis which is a branch in mathematics.

The study highlights issues regarding solving ODEs that maybe useful to researchers, students and professions that apply numerical analysis in their field of interest.

Numerical analysis, despite it's being time-consuming and stressful, can be used to solve problems that cannot be solved analytically. Thereby deriving some form of accuracies to the solution of such problems. This study can help in clarifying accuracy of BDF and Runge-Kutta methods.

2 Literature Review

2.1 Ordinary Differential Equations

An ODE is a differential equation containing a function or functions of one independent variable and its derivatives.

There are two types of ordinary differential equations, namely, linear and non-linear. Linear equations are equations in which the variable in an equation appears only with a power of one. For example $x'' + x = 0$, $x'' + 2x' + x = 0$ and so on. But this study deals with non-linear equations, which are complicated equations to solve i.e they can not be solved exactly and are the subject of much on-going research. For example, $x'' + \sin(x) = 0$, $x' + x^2 = 0$ and so on.

In Mathematics, a stiff equation is a differential equation for which certain numerical methods for solving the equation are numerically unstable, unless the stepsize is taken to be extremely small see [12, 13, 14].

The Phenomenon of Stiffness is not precisely defined in the literature. An attempt at describing a Stiff problem is: A problem is Stiff if explicit methods don't work, or work only extremely slowly as in [21].

In this dissertation, the first order nonlinear differential equation is considered.

$$\frac{dy}{dx} = f(x, y);$$

$$y(x_0) = y_0$$

$$a \leq x \leq b$$

as the test problem. This equation can be nonlinear or even a system of nonlinear equations (in which cases y is a vector and f is a vector of n different functions).

2.2 Ways of solving differential equations

The solutions of first order differential equations are premised on two known tools, namely, the Fundamental Theorem of Calculus (FTC) and the Fundamental Theorem of Numerical Analysis (FTN); FTC is a theoretical tool whereas FTN is experimentally applied.

The Fundamental Theorem of Calculus relates differentiation and integration of functions, showing that these two operations are essentially inverses of one another. In this study numerical integration methods as means of solving a nonlinear ODE are applied.

The theorem is usually stated in two parts, where the first part deals with the derivative of an antiderivative, while the second part deals with the relationship between antiderivatives and definite integrals as in [6, 22].

The first part asserts that if f is a continuous function and a is a number in the domain of f and we define the function g by

$$g(x) = \int_a^x f(t)\,\mathrm{d}t$$

then

$$g'(x) = f(x)$$

In order words,

$$\frac{d}{dx}\left(\int_a^x f(t)\,\mathrm{d}t\right) = f(x)$$

On the other hand, the second part asserts that if f is continuous function and F is an antiderivative of f on the interval [a,b], then

$$\int_a^b f(t)\,\mathrm{d}t = F(b) - F(a)$$

The Fundamental Theorem of Numerical Analysis states that for a numerical method, Consistency plus Stability implies Convergence see [1, 19].

2.2.1 Numerical methods

Numerical methods for ordinary differential equations approximate solutions to initial value problems of the form

$$\frac{dy}{dx} = f(x, y)$$

where

$$y(x_0) = y_0$$

The result is approximations for the value of $y(x)$ at discrete times x_n:

$$y_n \approx y(x_n)$$

where

$$x_n = x_0 + nh$$

for $n = 0, 1, ...N + 1$

2.2.2 Types of Numerical Methods

Numerical analysis is not only the design of numerical methods, but also their analysis. Three central concepts in this analysis are:

*Convergence: whether the method approximates the solution,
*Order: how well it approximates the solution, and
*Stability: whether errors are damped out see [12].

Numerical methods fall into two major categories namely the Single-Step methods and the Multi-Step methods.

Single-Step methods include the Taylor's series method, Euler's method, Runge-Kutta method and they refer to only one previous point and its derivative to determine the current value. Although methods such as Runge-Kutta take some intermediate steps (for example, a half-step) to obtain a higher order method, they still discard all previous information before taking a second step.

Multi-Step methods include the Adams-Bashforth method, Adams-Moulton method, Backward Difference Formula. These methods attempt to gain efficiency by keeping and using the information from previous steps rather than discarding it. Consequently, multistep method refer to several previous points and derivative values.

One major problem with the multi-step is the question of how to start the process off. The usual method is to use a one-step method of at least the same order as the desired multi-step method to generate the initial values or to solve the problem analytically to get the initial values necessary to get the process started.

For some differential equations, application of standard methods-such as the Euler method, explicit Runge-Kutta method, or multistep methods (e.g. Adams-Bashforth methods) - exhibit instabilty in the solutions, though other methods may produce stable solutions. This "difficult behaviour" in the equation (which may not necessarily be complex itself) is described as stiffness (as mentioned under 1.5.5), and is often caused by the presence of different time scales in the underlying problem. For example, a collision in a mechanical system like in an impact oscillator typically occurs at much smaller time scale than the time for the motion of objects; this discrepancy makes for very "sharp turns" in the curves of the state parameters.

The consistence of a numerical method is important because it analyses the order of accuracy inorder to ensure convergence.

A related concept is the global (truncation) error, the error sustained in all the steps one needs to reach a fixed time t. Explicitly, the global error at time t is $y_N - y(t)$ where $N = (t - t_0)/h$. The global error of a pth order one-step method is $0(h^p)$; in particular, such a method is convergent. This statement is not necessarily true for multi-step methods.

3 Preliminaries: Consistency, Convergence and Stability of BDF, RK

3.0.3 Explicit and Implicit methods

Explicit methods means a new value y_{n+1} is defined in terms of already known values like y_n. While Implicit Methods means we have to solve an equation to find y_{n+1}. It costs more time to solve equations using Implicit methods than explicit methods. This cost must be taken into consideration when one selects the method to use see [2, 12, 13].

3.0.4 Local truncation error

Local truncation error is defined to be the difference between the result y_{n+s} of the method, assuming that all the previous values $y_{n+s-1},\ldots,y_n$ are exact, and the exact solution of the equation at time t_{n+s} as in [12, 13].

The local (truncation) error of the method is the error committed by one step of the method. That is, it is the difference between the result given by the method, assuming that no error was made in earlier steps, and the exact solution see [2, 12, 13, 14, 15]:

$$\delta_{n+k}^{h} = \Psi(x_{n+k}; y_n, y_{n+1}, \ldots, y_{n+k-1}; h) - y_{n+k}$$

3.0.5 Order

The method has order p if

$$\delta_{n+k}^{h} = 0(h^{p+1})$$

as $h \to 0$ [2, 12, 13, 14, 15]:

3.0.6 Consistency

Consistency is a connection between the problem and the numerical method. A numerical method is said to be consistent with an IVP if the error commuted by the numerical algorithm over a single time step is small. Where single time step refers to step size. Consistency is a local property and hence easy to verify using a local truncation error. That is

The method is consistent see [12] if

$$\lim_{h \to 0} \delta^h_{n+k} = 0$$

A one-step difference equation with local truncation error $\tau_i(h)$ is said to be consistent if

$$\lim_{h \to 0} \max_{1 \leq i \leq N} |\tau_i(h)| = 0$$

A multi-step method is consistent if the <u>local truncation error</u> goes to zero faster than the step size h see [12, 13, 15]. Therefore an m-step multi-step is consistent if $\lim_{h\to 0} |\tau_i(h)| = 0$, for all $i = m, m+1, ..., N$ and $\lim_{h\to 0} |\alpha_i - y(t_i)| = 0$, for all i = 1,2,...,m-1. Where,

$\tau_i(h)$ is the local truncation error
α is the initial value
$y(t_i)$ is the exact solution [4, 24].

3.0.7 Stability

A Numerical method must be stable to converge. This means

A Linear multi-step method is zero-stable for a certain differential equation on a given time interval, if a perturbation in the starting values of size ε causes the numerical solution over that time interval to change by no more than $K\varepsilon$ for some value of K which does not depend on the step size h.

This is called "zero-stability" because it is enough to check the condition for the differential equation $y' = 0$ [23].

Now suppose that a consistent linear multistep method is applied to a sufficiently smooth differential equation and that the starting values y_1,...,y_{s-1} all converge to the initial value y_0 as $h \to 0$. Then, the numerical solution converges to the exact solution as $h \to 0$ if and only if the method is zero-stable. This result is known as the Dahlquist Equivalence Theorem, named after Germund Dahlquist as in [2, 7, 8, 23].

The stability of a multi-step method with respect to round-off error is dictated by magnitudes of zeros of the characteristic polynomial. If $|\lambda_j| > 1$ for any of $\lambda_2, \lambda_3, ..., \lambda_m$, the round-off error grows exponentially as in [24].

Definition of stability of multi-step method.

(1) Methods that satisfy the root condition and have $\lambda = 1$ as the only root of the characteristic equation with magnitude one are called strongly stable.

(2) Methods that satisfy the root condition and have more than one distinct roots with magnitude one are called weakly stable (relatively stable).

(3) Methods that do not satisfy the root condition are called unstable see [24].

3.0.8 Characteristic Polynomials

Let A be a square matrix (n x n). The characteristic polynomial of A is the determinant of the n x n matrix $\lambda l_n - A$. This is a polynomial of degree n in λ see [18].

The following are characteristic polynomials or generating polynomials as in [9].

$$\rho(\omega) = \sum_{m=0}^{s} a_m \omega^m$$
$$\sigma(\omega) = \sum_{m=0}^{s} b_m \omega^m$$

where

* ρ and σ are the two consistency conditions above corresponding to
$\rho(1) = 0$ and $\rho'(1) = \sigma(1)$.

* a_m and b_m are the coefficients.

Note that different choices of the coefficients a_m and b_m yield different numerical methods.

* ω is the root.

For example, if the characteristic equation has $\omega = 1$, then $\omega = 1$ is the only root of modulus 1, which makes the method strongly stable.

3.0.9 Root Condition

A (complex) polynomial p obeys the root condition if

* all its zeros, i.e., all z such that $p(z) = 0$, lie in the unit disk, i.e., $|z| \leq 1$, and

* all zeros on the unit circle are simple, i.e., if $|z| = 1$ then $p'(z) \neq 0$ [9].

3.0.10 Convergence

A numerical method is said to converge to a solution if the distance between the numerical solution and the exact solution goes to zero as the method parameter approaches some limit. Convergence is the desired property of a numerical method. That is

A one-step equation is said to be convergent if

$$\lim_{h \to 0} \max_{1 \le i \le N} |\omega_i - y(t_i)| = 0$$

where $y(t_i)$ is the exact solution
and ω_i is the approximate solution as in [24].

A multi-step method is convergent if and only if it is consistent and the polynomial p obeys the root condition [9].

3.1 Backward Difference Formula (BDF)

Backward Difference Formula (BDF) A BDF is used to solve the initial value problem:

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

where

$$y(x_0) = y_0$$

The general formula for a BDF can be written as

$$\sum_{k=0}^{s} a_k y_{n+k} = h\beta f(x_{n+s}, y_{n+s})$$

where h denotes the step size and $x_n = x_0 + nh$. The coefficients a_k and β are chosen so that the method achieves order s, which is the maximum possible. BDF methods are implicit and as such, require the solution of nonlinear equations at each step. Typically modified Newton's method is used to solve these nonlinear equations.

The Backward Difference Formulas (BDF) were proposed by Celaya, E.A., Anza, J.J. and Gear, W.C. as in [5, 10]. They are linear multi-step methods useful to solve ODEs of order 1, such as equation 1. Since they were introduced, the Backward Difference Formula have been widely used due to their good stability properties for solving stiff problems.

The BDfs, the BDF_1 and BDF_2 are unconditionally stable. The stability properties of the BDFs and their error constants can be calculated following the formula given as in [2, 5, 10, 12]. The $A(\alpha)$-stability of the BDFs [5, 13, 20] and their error constants can be seen in the Table below

Order	1	2	3	4	5	6
α	90	90	86.03	73.35	51.84	17.84
C	-1/2	-1/3	-1/4	-1/5	-1/6	-1/7

Table 1: The Stability properties of BDFs

where

α shows the region of absolute stability contains large wedges symmetric about the negative real axis in the left half plane. It can be calculated from the formula given as in [2, 5, 10, 12].

C is the order condition in local truncation error. That is the constants α_j are chosen in order to verify the order condition, that says the multistep method is of order p if the next condition is satisfied see [2, 16]:

C_i, for $1 \leq q \leq p$ and $C_{p+1} \neq 0$

$$C_0 = \sum_{i=0}^{k} \alpha_i$$

$C_1 = \sum_{i=0}^{k} \alpha_i$ - $\sum_{i=0}^{k} \beta_i$

$C_1 = \frac{1}{q!}(\sum_{i=0}^{k} i^q \alpha_i)$ - $\frac{1}{q!}(\sum_{i=0}^{k} i^{q-1} \beta_i)$ $q \geq 2$

While the other methods like Adams methods were constructed so that they satisfy the root condition, this is no longer automatically true for the BDFs. In fact, we have that the characteristic polynomial ρ for a BDF satisfies the root condition and the underlying BDF method is convergent if and only if $1 \leq s \leq 6$ [9]. Methods with $s > 6$ are not zero-stable so they cannot be used.

BDF_1 and BDF_2 are consistent and convergent because they obey the root condition, since all their zeros lie inside the unit disk. They are A-stable, because their region of absolute stability contains the entire left half-plane (all $\omega \epsilon$C such that Re(ω) ≤ 0).

While BDF_3 and BDF_4, apart from being consistent and convergent because they obey the root condition,since all their zeros lie inside the unit disk. They are A(α)-Stable, which means that their region of absolute stability contains large wedges symmetric about the negative real axis in the left half plane.

BDF_s are implicit in nature because they need several approximations to obtain the next approximations. For example if we have obtained values for $y(x_n)$ say $y_0, y_1, \ldots$. Then it is reasonable to have a formula that uses more than just $y(x_n)$ determine $y(x_{n+1})$.

In summary the properties of BDFs are shown in Table 2 [9].

BDFs are implicit methods which are suitable for solving Stiff problems. It is convergent if and only if $p \leq 6$ and it is unstable when $p > 6$. From Table 2, $A(\alpha)$-Stable means it is weaker than A-Stable.

BDFs	LTE	Order	Consistence	Stability	Convergence
BDF_1	$-\frac{h^2}{2}y^2(\eta)$	1	Consistent	A-Stable	Convergent
BDF_2	$-\frac{2h^3}{9}y^3(\eta)$	2	Consistent	A-Stable	Convergent
BDF_3	$-\frac{3h^4}{22}y^4(\eta)$	3	Consistent	$A(\alpha)$-Stable	Convergent
BDF_4	$-\frac{12h^5}{125}y^5(\eta)$	4	Consistent	$A(\alpha)$-Stable	Convergent

Table 2: The properties of BDFs

3.2 Runge-Kutta methods

This is a group of methods due to two German Mathematicians Carl Runge and Martin Kutta. The most widely used of the Runge-Kutta methods is the fourth order Runge-Kutta method. It is widely used in solving differential equations. These are one-step methods, but they depend on estimates of the solution at different points.

The formula of a Runge-Kutta of order 4 is

$$y_{n+1} = y_n + \frac{1}{6}[k_1 + 2k_2 + 2k_3 + k_4]$$

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

$$k_2 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)$$

$$k_3 = hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

Here y_{n+1} is the RK_4 approximation of $y(x_{n+1})$, this next value (y_{n+1}) is determined by the present value (y_n) plus the weighted average of four increments, where each increment is the product of the size of the interval, h, and an estimated slope specified by function f on the right-hand side of the differential equation.

* k_1 is the increment based on the slope at the beginning of the interval, using y_1 [Euler's method];
* k_2 is the increment based on the slope at the midpoint of the interval, using $y + \frac{h}{2}k_1$;
* k_3 is again the increment based on the slope at the midpoint, but now using $y + \frac{h}{2}k_2$;
* k_4 is the increment based on the slope at the end of the interval, using $y + hk_3$.

In averaging the four increments, greater weight is given to the increments at the midpoint. If f is independent of y, so that the differential equation is equivalent to a simple integral, then RK_4 is Simpson's rule.

The RK_4 method is a fourth- order method, meaning that the local truncation error is of the order $0(h^5)$, while the total accumulated error is order $0(h^4)$.

The advantages of Runge-Kutta Methods are:
(1) It is a One-step method where the global error is of the same order as local error.
(2) There is no need to know the derivatives of f.
(3) It is easy for Automatic Error Control.

The Runge-Kutta methods fall into two categories: The Explicit and the Implicit methods. From the formula used in this study, the RK_4 is an explicit Runge-Kutta method. These methods are convergent for $p \leq 4$, and they become unstable when $p > 4$ see [3]. It is noted that they can never be A-Stable [13]. Additionally the relatively stability of these methods gets weaker as the order increases. As for the Implicit Runge-Kutta method, they are all A-Stable and are suitable for solving Stiff problems see [12].

The advantage of implicit Runge-Kutta methods over explicit ones is their greater stability, especially when applied to stiff equations. All (implicit) Runge-Kutta are all A-Stable. And moreover, as the order increases the stability becomes better.

In summary the properties of Runge-kutta are shown in Table 3 as in [13].

RKs	LTE	Order	Consistence	Stability	Convergence
RK_1	$0(h)^2$	p = 1	Consistent	Stable	Convergent
RK_2	$0(h)^3$	p = 2	Consistent	Stable	Convergent
RK_3	$0(h)^4$	p = 3	Consistent	Weakly Stable	Convergent
RK_4	$0(h)^5$	p = 4	Consistent	Weakly Stable	Convergent

Table 3: The properties of Runge-Kutta

4 Numerical Experiments, Results and Discussions

In order to achieve the stated objectives, the following two test problems were considered and solved.

$$\frac{dy}{dx} + xy = xy^2 \tag{1}$$

with

$$y(0) = \frac{1}{2}$$

the analytic solution is given as

$$y(x) = \frac{1}{e^{\frac{x^2}{2}} + 1}$$

n	x_n	Analytic (y_n)
0	0	0.500000000
1	0.05	0.499687500
2	0.1	0.498750002
3	0.15	0.497187529
4	0.2	0.495000166
5	0.25	0.492188135
6	0.3	0.488751898
7	0.35	0.484692285
8	0.4	0.480010659
9	0.45	0.474709102
10	0.5	0.468790626

Table 4: Analytic solutions of equation 1

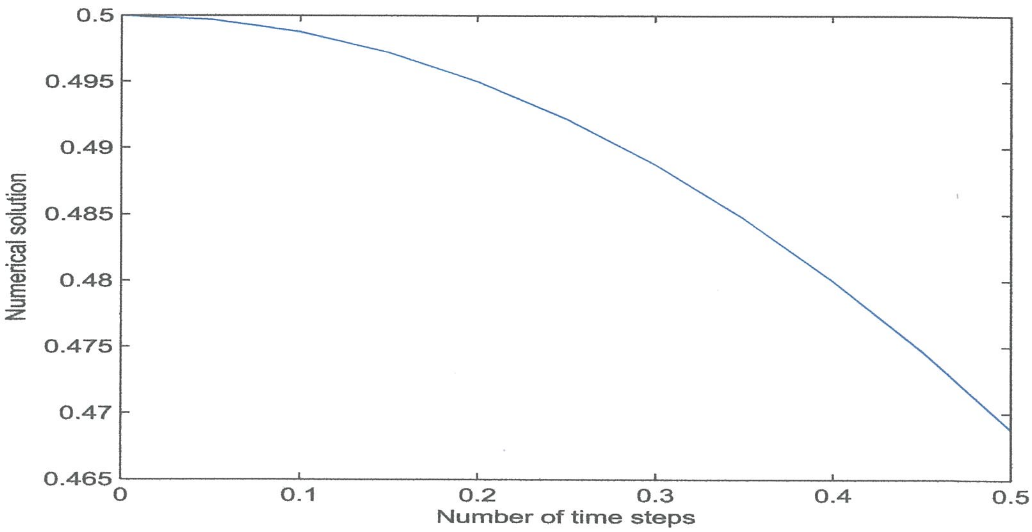

Figure 1: The graph of Analytic solution of equation 1

and

$$2\frac{dy}{dx} = \frac{y}{x} + \frac{x^2}{y} \tag{2}$$

with

$$y(1) = 2$$

the analytic solution is given as

$$y(x) = \sqrt{\frac{x^3 + 7x}{2}}$$

n	x_n	Analytic (y_n)
0	1.0	2.000000000
1	1.1	2.124970588
2	1.2	2.250333309
3	1.3	2.376657316
4	1.4	2.504396135
5	1.5	2.633913438
6	1.6	2.765501763
7	1.7	2.899396489
8	1.8	3.035786554
9	1.9	3.174822830
10	2.0	3.316624790

Table 5: Analytic solutions of equation 2

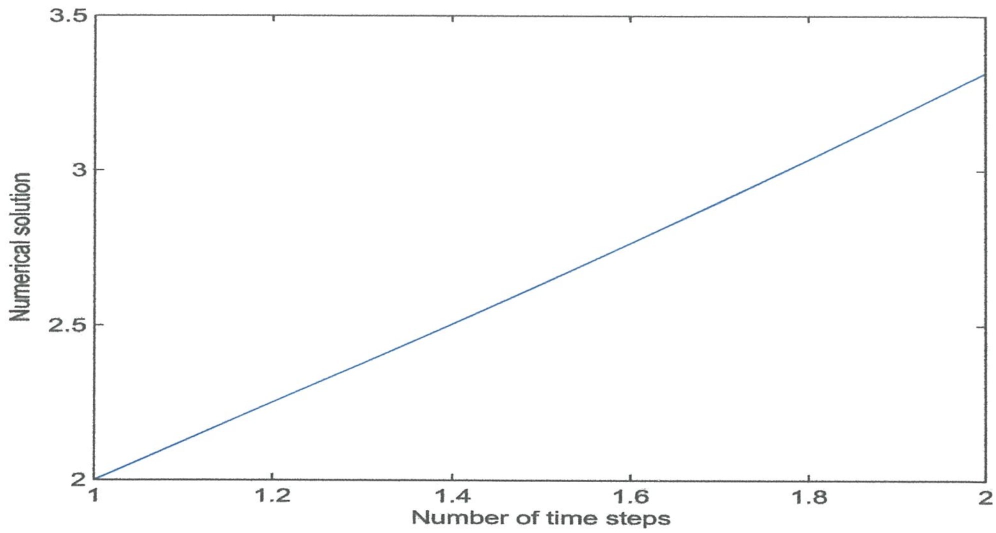

Figure 2: The graph of Analytic solution of equation 2

The Runge-Kutta method and Backward Difference Formula both of the same order will be used to perform numerical experiments and compare numerical results.

4.1 Results for Runge-Kutta method

4.1.1 Specific objective 1

The first objective was achieved by considering the Runge-Kutta method of order four to test the problems in equation (1) and equation (2).They were further compared with the Analytic solution to extract errors.

n	x_n	Analytic (y_n)	RK_4 (y_n)	Error
0	0	0.500000000	0.500000000	0.000000000
1	0.05	0.499687500	0.499687500	0.000000000
2	0.1	0.498750002	0.498750018	-0.000000016
3	0.15	0.497187529	0.497187545	-0.000000016
4	0.2	0.495000166	0.495000182	-0.000000016
5	0.25	0.492188135	0.492188151	-0.000000016
6	0.3	0.488751898	0.488751913	-0.000000015
7	0.35	0.484692285	0.484692300	-0.000000015
8	0.4	0.480010659	0.480010674	-0.000000015
9	0.45	0.474709102	0.474709116	-0.000000014
10	0.5	0.468790626	0.468790640	-0.000000014

Table 6: Errors of RK_4 solutions of equation 1

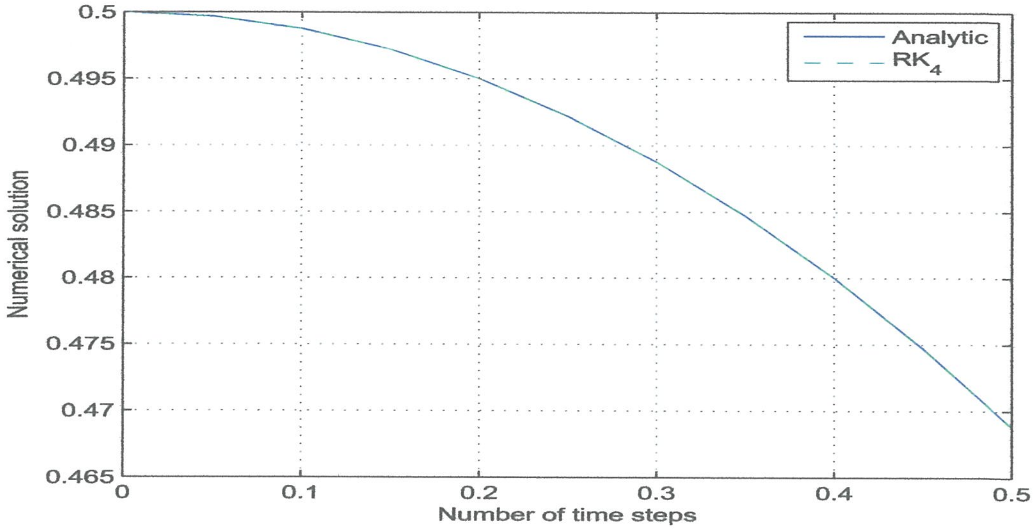

Figure 3: The graph of the Analytic and RK_4 solutions of equation 1

n	x_n	Analytic (y_n)	RK_4 (y_n)	Error
0	1.0	2.000000000	2.000000000	0.000000000
1	1.1	2.124970588	2.124970633	-0.000000045
2	1.2	2.250333309	2.250333384	-0.000000075
3	1.3	2.376657316	2.376657412	-0.000000096
4	1.4	2.504396135	2.504396245	-0.000000110
5	1.5	2.633913438	2.633913559	-0.000000121
6	1.6	2.765501763	2.765501892	-0.000000129
7	1.7	2.899396489	2.899396625	-0.000000136
8	1.8	3.035786554	3.035786695	-0.000000141
9	1.9	3.174822830	3.174822975	-0.000000145
10	2.0	3.316624790	3.316624939	-0.000000149

Table 7: Errors of RK_4 solutions of equation 2

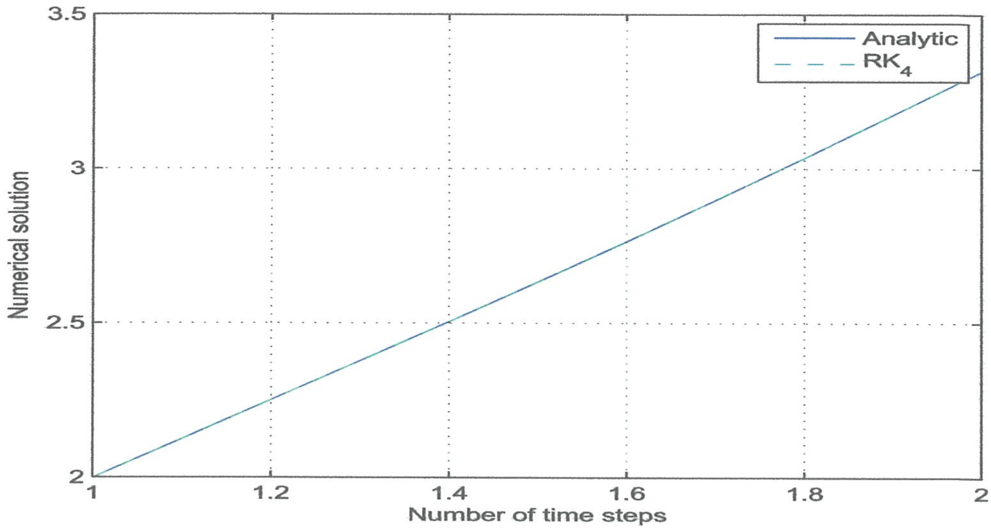

Figure 4: The graph of the Analytic and RK_4 solutions of equation 2

4.2 Results for Backward Difference Formula method

4.2.1 Specific objective 2

The second objective was achieved by considering the BDF method of order four to test the problems in equation (1) and equation (2). They were further compared with the Analytic solution to extract errors.

n	x_n	Analytic (y_n)	BDF_4 (y_n)	Error
0	0	0.500000000	0.500000000	0.000000000
1	0.05	0.499687500	0.499687500	0.000000000
2	0.1	0.498750002	0.498750002	0.000000000
3	0.15	0.497187529	0.497187529	0.000000000
4	0.2	0.495000166	0.496200052	-0.001199886
5	0.25	0.492188135	0.495991559	-0.003803424
6	0.3	0.488751898	0.496125736	-0.007373838
7	0.35	0.484692285	0.496239099	-0.011546814
8	0.4	0.480010659	0.496248601	-0.016237942
9	0.45	0.474709102	0.496214495	-0.021505393
10	0.5	0.468790626	0.496191779	-0.027401153

Table 8: Errors of BDF_4 solutions of equation 1

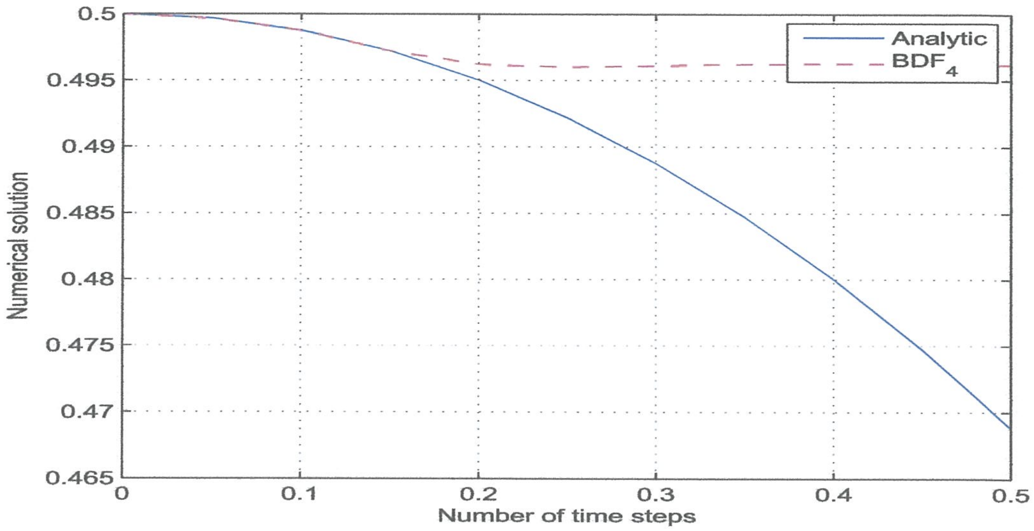

Figure 5: The graph of the Analytic and BDF_4 solutions of equation 1

n	x_n	Analytic (y_n)	BDF_4 (y_n)	Error
0	1.0	2.000000000	2.000000000	0.000000000
1	1.1	2.124970588	2.124970588	0.000000000
2	1.2	2.250333309	2.250333309	0.000000000
3	1.3	2.376657316	2.376657316	0.000000000
4	1.4	2.504396135	2.533148666	-0.028752531
5	1.5	2.633913438	2.724060724	-0.090147286
6	1.6	2.765501763	2.939008322	-0.173506559
7	1.7	2.899396489	3.170383156	-0.270986667
8	1.8	3.035786554	3.417714222	-0.381927668
9	1.9	3.174822830	3.683911219	-0.509088389
10	2.0	3.316624790	3.971624778	-0.654999988

Table 9: Errors of BDF_4 solutions of equation 2

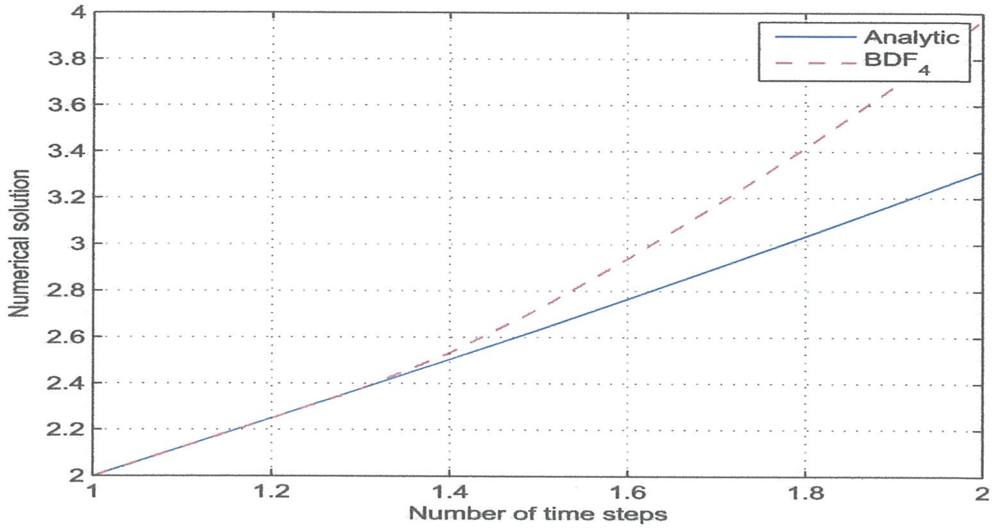

Figure 6: The graph of the Analytic and BDF_4 solutions of equation 2

4.3 Accuracies of BDF_4 and RK_4

4.3.1 Specific objective 3

The third objective was achieved by comparing and contrasting the accuracies of the solutions of BDF of order 4 and Runge-Kutta of order 4

n	x_n	Analytic (y_n)	RK_4 (y_n)	Error RK_4	BDF_4 (y_n)	Error BDF_4
0	0	0.500000000	0.500000000	0.000000000	0.500000000	0.000000000
1	0.05	0.499687500	0.499687500	0.000000000	0.499687500	0.000000000
2	0.1	0.498750002	0.498750018	-0.000000016	0.498750002	0.000000000
3	0.15	0.497187529	0.497187545	-0.000000016	0.497187529	0.000000000
4	0.2	0.495000166	0.495000182	-0.000000016	0.496200052	-0.001199886
5	0.25	0.492188135	0.492188151	-0.000000016	0.495991559	-0.003803424
6	0.3	0.488751898	0.488751913	-0.000000015	0.496125736	-0.007373838
7	0.35	0.484692285	0.484692300	-0.000000015	0.496239099	-0.011546814
8	0.4	0.480010659	0.480010674	-0.000000015	0.496248601	-0.016237942
9	0.45	0.474709102	0.474709116	-0.000000014	0.496214495	-0.021505393
10	0.5	0.468790626	0.468790640	-0.000000014	0.496191779	-0.027401153

Table 10: Errors of RK_4 and BDF_4 of equation 1

where Error = Exact - Numerical

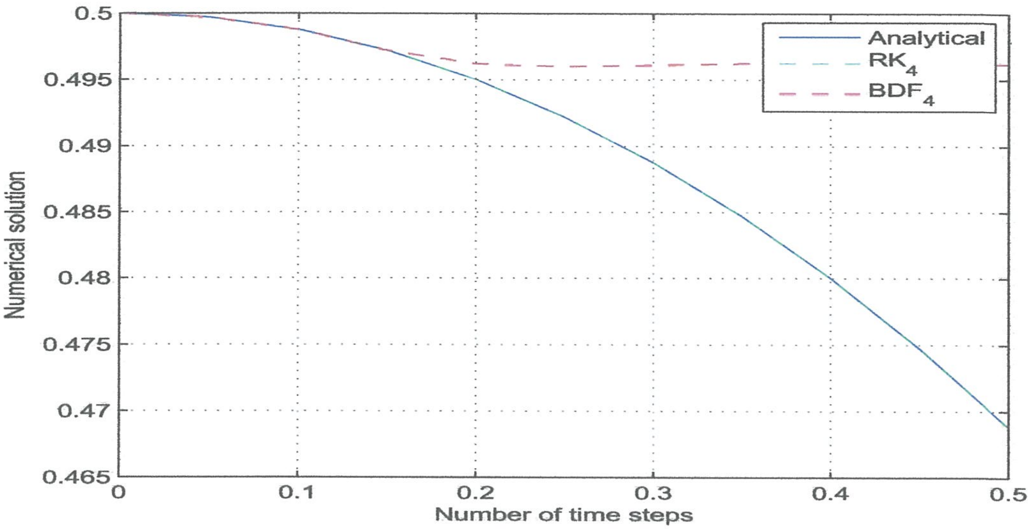

Figure 7: The graph of the Analytic, RK_4 and BDF_4 solutions of equation 1

n	x_n	Analytic (y_n)	RK_4 (y_n)	Error	BDF_4 (y_n)	Error
0	1.0	2.000000000	2.000000000	0.000000000	2.000000000	0.000000000
1	1.1	2.124970588	2.124970633	-0.000000045	2.124970588	0.000000000
2	1.2	2.250333309	2.250333384	-0.000000075	2.250333309	0.000000000
3	1.3	2.376657316	2.376657412	-0.000000096	2.376657316	0.000000000
4	1.4	2.504396135	2.504396245	-0.000000110	2.533148666	-0.028752531
5	1.5	2.633913438	2.633913559	-0.000000121	2.724060724	-0.090147286
6	1.6	2.765501763	2.765501892	-0.000000129	2.939008322	-0.173506559
7	1.7	2.899396489	2.899396625	-0.000000136	3.170383156	-0.270986667
8	1.8	3.035786554	3.035786695	-0.000000141	3.417714222	-0.381927668
9	1.9	3.174822830	3.174822975	-0.000000145	3.683911219	-0.509088389
10	2.0	3.316624790	3.316624939	-0.000000149	3.971624778	-0.654999988

Table 11: Errors of RK_4 and BDF_4 of equation 2

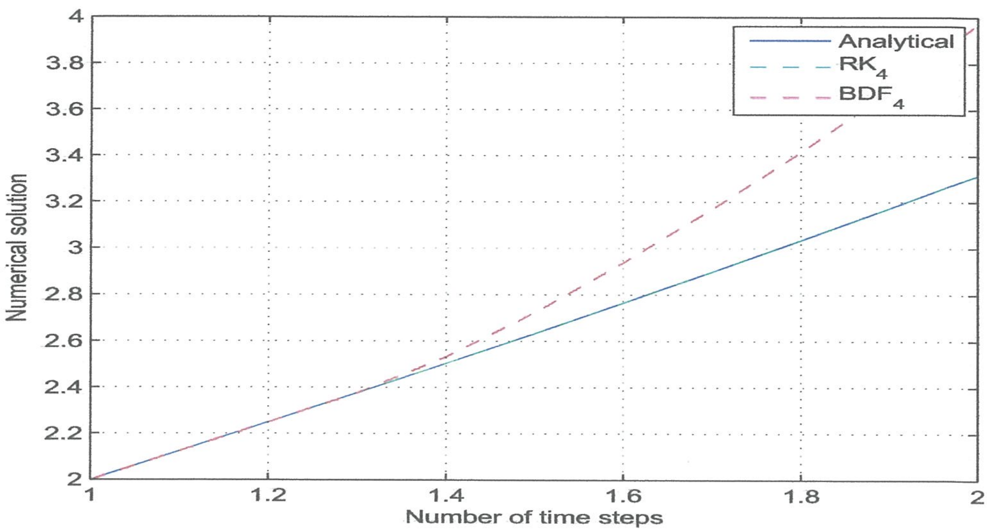

Figure 8: The graph of the Analytic, RK_4 and BDF_4 solutions of equation 2

4.4 Discussions of results

4.4.1 Specific Objective one

In this study, the solution of two 1st order non-linear ODE were chosen with their Analytical method. The graph of their resultant equations were plotted see Figure 1 and 2. The purpose of this activity was to provide the basis of discussion of the other two methods regarding their accuracy.

The solutions of two 1st order non-linear ODE were computed using the Runge-Kutta of order 4 method in comparison with the Analytic method. Their graph of the resultant equation were also plotted. The purpose was to generate values that were used in investigating it's accuracy, by analysing it's deviations from the analytic solutions see Table 6,7 and Figure 3,4. The RK_4 method showed very slight deviation from the Analytical method when it is rounded off to 9 decimal places, otherwise it was equal to the Analytic method when rounded off to 7 decimal places in equation 1 and 6 decimal places in equation 2.

4.4.2 Specific Objective two

The solutions of two 1st order non-linear ODE were computed using the BDF of order 4 method in comparison with the Analytic method. Their graph of the resultant equation were also plotted. The purpose was to generate values that were used in investigating it's accuracy, by analysing it's deviations from the analytic solutions see Table 8,9 and Figure 5,6. The BDF_4 method showed large deviation from the Analytical method despite using the first four starting values from the Analytic method.

4.4.3 Specific Objective three

The accuracies using BDF of order 4 method and the Runge-Kutta of order 4 method were compared and contrasted. The RK_4 proved to be more accurate than the BDF_4 because the RK_4 can generate it's values without depending on the Analytic method (see Table 6 and 7) and it showed very slight deviation from the Analytic method when it is rounded off to 9 decimal places, otherwise it was equal to the Analytic method when rounded off to 7 decimal places in equation 1 (see Table 6 and Figure 3) and 6 decimal places in equation 2 (see Table 7 and Figure 4), while the BDF_4, despite it's using starting values from the Analytic method showed large deviation from the Analytic method (see Figure 5 and 6) even when applied to the two tested problems (see Table 8 and 9). The BDF_4 has the disadvantage of requiring starting values, since initially there was information at one point only.

5 Conclusions and Recommendation

5.1 Introduction

In this chapter, conclusions were made and recommendations as well as areas for further research are outlined.

5.2 Conclusions

Two first order non-linear ODEs were chosen with their Analytic method. They were further tested using the BDF_4 and the RK_4, each in comparison with their analytic method and some results were gotten. These results led to comparing the level of accuracy between the BDF_4 and RK_4. The RK_4 method showed very slight deviation from the Exact method when it is rounded off to 9 decimal places, otherwise it was equal to the Exact method when rounded off to 7 decimal places in equation 1 (see Table 6 and Figure 3) and 6 decimal places in equation 2 (see Table 7 and Figure 4), unlike the BDF_4 which showed large deviation from the Analytic solution, even when applied to the two tested problems.The RK_4 showed more accuracy over BDF_4 because it had less errors as compared to the BDF_4 despite the BDF_4 using the first four starting values from the Analytic method. But on the other hand, the BDF_4 is more prefered than the RK_4 in solving stiff problems because it is A-Stable and converges easily see Table 2 and 3.

5.3 Recommendations

The Analytical method should be used with the numerical methods in analysing the errors in solving 1st order non-linear Ordinary differential equations.

The RK_4 method showed very slight deviation from the Analytical method when it is rounded up to 9 decimal places, otherwise it was equal to the Analytical method when rounded up to 6 or 7 decimal places. So RK_4 is highly recommended for solving 1st order non-linear ordinary differential equation.

The BDF_4 method showed much deviation from the Exact solution, even with the two different problems used for testing.

Both the RK_4 and the BDF_4 were compared, and I accept from the results that RK_4 is more prefered in solving 1st order non-linear ordinary differential equation because it has very slight errors, and it is almost as accurate as the Analytic solution.

Numerical methods are time-consuming, and generally, BDF_4 works better than the RK_4 in solving stiff problems because it is A-Stable and converges easily. Although it costs more time to solve these equations, if only this cost is taken into consideration. It will be recommendable to carry out investigations on how these costs can be reduced.

Finally, I suggest to my colleagues, students and all academicians to continue to pursue this area more vigorously for more expansion and advancement because Mathematics is an important and interesting aspect of our Education and general technological advancements. One of the branches of Mathematics called the Numerical Analysis is very important in understanding the theories underlying some particular areas of science and technology being practiced. This study was just to compare the accuracies of a BDF_4 and RK_4 in solving 1st order nonlinear Ordinary Differential Equations.

6 Appendices

6.1 MATLAB codes

6.1.1 For plotting Figure 1

```
clc
clear
xn = 0:0.05:0.5;
y = [0.500000000  0.499687500 0.498750002 0.497187529
0.495000166 0.492188135 0.488751898 0.484692285
0.480010659 0.474709102 0.468790626]
plot (xn , y)
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.2 For plotting Figure 2

```
clc
clear
xn = 1:0.1:2;
y = [2.000000000  2.124970588 2.250333309 2.376657316
2.504396135 2.633913438 2.765501763 2.899396489
3.035786554 3.174822830 3.316624790]
plot (xn , y)
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.3 For plotting Figure 3

```
clc
clear
xn = 0:0.05:0.5;
a = [0.500000000  0.499687500 0.498750002 0.497187529
0.495000166 0.492188135 0.488751898 0.484692285
```

```
0.480010659 0.474709102 0.468790626]
b = [0.500000000  0.499687500 0.498750018 0.497187545
0.495000182 0.492188151 0.488751913 0.484692300
0.480010674 0.474709116 0.468790640]
plot (xn , a, 'b')
hold on
plot (xn , b, 'g--')
hold on
grid on
legend ('Analytic','RK_4')
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.4 For plotting Figure 4

```
clc
clear
xn = 1:0.1:2;
a = [2.000000000  2.124970588 2.250333309 2.376657316
2.504396135 2.633913438 2.765501763 2.899396489
3.035786554 3.174822830 3.316624790]
b = [2.000000000  2.124970633 2.250333384 2.376657412
2.504396245 2.633913559 2.765501892 2.899396625
3.035786695 3.174822975  3.316624939]
plot (xn , a, 'b')
hold on
plot (xn , b, 'g--')
hold on
grid on
legend ('Analytic','RK_4')
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.5 For plotting Figure 5

```
clc
clear
xn = 0:0.05:0.5;
a = [0.500000000  0.499687500 0.498750002 0.497187529
0.495000166 0.492188135 0.488751898 0.484692285
0.480010659 0.474709102 0.468790626]
b = [0.500000000  0.499687500 0.498750002 0.497187529
0.496200052 0.495991559 0.496125736 0.496239099
0.496248601 0.496214495 0.496191779]
plot (xn , a, 'b')
hold on
plot (xn , b, 'r--')
hold on
grid on
legend ('Analytic','BDF_4')
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.6 For plotting Figure 6

```
clc
clear
xn = 1:0.1:2;
a = [2.000000000  2.124970588 2.250333309 2.376657316
2.504396135 2.633913438 2.765501763 2.899396489
3.035786554 3.174822830 3.316624790]
b = [2.000000000  2.124970588 2.250333309 2.376657316
2.533148666 2.724060724 2.939008322 3.170383156
3.417714222 3.683911219  3.971624778]
plot (xn , a, 'b')
hold on
```

```
plot (xn , b, 'r--')
hold on
grid on
legend ('Analytic','BDF_4')
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.7 For plotting Figure 7

```
clc
clear
xn = 0:0.05:0.5;
a = [0.500000000  0.499687500 0.498750002 0.497187529
0.495000166 0.492188135 0.488751898 0.484692285
0.480010659 0.474709102 0.468790626]
b = [0.500000000  0.499687500 0.498750018 0.497187545
0.495000182 0.492188151 0.488751913 0.484692300
0.480010674 0.474709116 0.468790640]
c = [0.500000000  0.499687500 0.498750002 0.497187529
0.496200052 0.495991559 0.496125736 0.496239099
0.496248601 0.496214495 0.496191779]
plot (xn , a, 'b')
hold on
plot (xn , b, 'g--')
hold on
plot (xn , c, 'r--')
hold on
grid on
legend ('Analytical','RK_4','BDF_4')
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

6.1.8 For plotting Figure 8

```
clc
clear
xn = 1:0.1:2;
a = [2.000000000  2.124970588 2.250333309 2.376657316
2.504396135 2.633913438 2.765501763 2.899396489
3.035786554 3.174822830 3.316624790]
b = [2.000000000  2.124970633 2.250333384 2.376657412
2.504396245 2.633913559 2.765501892 2.899396625
3.035786695 3.174822975  3.316624939]
c = [2.000000000  2.124970588 2.250333309 2.376657316
2.533148666 2.724060724 2.939008322 3.170383156
3.417714222 3.683911219  3.971624778]
plot (xn , a, 'b')
hold on
plot (xn , b, 'g--')
hold on
plot (xn , c, 'r--')
hold on
grid on
legend ('Analytical','RK_4','BDF_4')
xlabel ('Number of time steps')
ylabel ('Numerical solution')
```

7 List of References

References

[1] Alec, J. *The Fundamaental Theorem of Numerical Analysis*, (2006).

[2] Butcher, J.C. *Numerical methods for ordinary differential equations in the 20th century.* J.Comput. Appl. Math., Vol. 125 (2000), 1-29.

[3] Butcher, J. *Runge-Kutta methods.* Scholarpedia (2007), 2(9):3147

[4] Burden, R.L. & Faires, J.D. *Numerical Analysis.* Brooks Cole (2015), 10th Edition.

[5] Celaya, E.A. & Anza, J.J. *BDF-α: A Multi-step Method with Numerical Damping Control.* Universal Journal of Computational Mathematics 1(3):96-108. DOI:10.13189/ujcmj.2013.010305, (2013).

[6] Courant, R. & Fritz, J. *Introduction to Calculus and Analysis.* Springer, (1965).

[7] Dahlquist, D. *Convergence and Stability in the numerical Integration of ordinary differential equations.* Mathematica Scandinavica Vol.4 (1956), pp.33-53

[8] Dahlquist, G. *A special stability problem for linear multistep methods.*BIT 3: 27-43, doi:10.1007/BF01963532, ISSN 0006-3835, (1963).

[9] Fasshauer, G. *Multi-step methods.* Retrieved November 23, 2015, from *math.iit.edu/ fass : http : //www.math.iit.edu/ fass/*478578_chapter_2.pdf, (2007).

[10] Gear, W.C. *Numerical initial value problems in Ordinary Differential equations.* Prentice Hall, New Jersey, (1971).

[11] Gerald, C.F. *Applied Numerical Analysis.* Second Edition. Addison-Wesley Publication Company, (1980).

[12] Hairer, E., Nørsett, S.P. & Wanner G. *Solving ordinary differential equations, I, Nonstiff problems.* Springer, Berlin, (1993).

[13] Hairer, E. & Wanner, G. *Solving Ordinary Diferential Equations, II, Stiff and Differential Algebraic Problems.* Springer, Berlin, (1996).

[14] Hochbruck, M & Ostermann, A. *Exponential Intergrators.* 209-286 doi:10.1017/S0962492910000048, (2010).

[15] Iserles, A. *A First Course in the Numerical Analysis of Differential Equations.* Cambridge University Press ISBN 0-521-55376-8 (hardback) ISBN 0-521-55655-4 (paperback), (1996).

[16] Lambert, J.D. *Numerical methods for Ordinary differential systems: the initial value problem.* John Wiley & sons, Chichester, (1991).

[17] Magaji, S. *Computer Problem Solving for Differential Equations (D.E).* Bsc Mathematics thesis, (1995).

[18] Quinlan, R. *The Characteristic Polynomial and Characteristic Equation.* Section 7.2, (2004-2005).

[19] Richtmyer & Morton. *Difference Methods for Initial Value Problems.* Second Edition, (1967).

[20] Shampine, L.F. & Reichelt, M.W. *The MATLAB ODE Suite.* SIAM J. Sci. Comput., Vol.18 (1997), No.1, 1-22.

[21] Shampine, L. *Numerical Solution of Ordinary Differential Equation* (1994).

[22] Steward, J. *Fundamental Theorem of Calculus, Calculus: early transcendentals.* Belmont, california, (2003): Thomson / Brooks / Cole.

[23] Süli & Mayers. *An Introduction to Numerical analysis.* Cambridge University Press, ISBN 0-521-00794-1, (2003).

[24] Zhiliang Xu. *Lec-5.10.pdf.* Retrieved December 19, 2015, from $ww3.nd.edu/\ zxu2\ :\ http\ ://www.ww3.nd.edu/\ zxu2$/acms40390F13/lec-5.10.pdf, (2015).