Journal of the Nigerian Association of Mathematical Physics
***Volume 52* (*July & Sept., 2019 Issue*), pp197 – 206**
© J. of NAMP

THE ACCURACY OF FOURTH ORDER RUNGE KUTTA METHOD IN SOLVING FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS (ODE) WITH INITIAL CONDITIONS

Anthony Anya Okeke

Department of Mathematics, Federal University Gashua, Yobe State, Nigeria

Abstract

In this paper, we present the fourth-order Runge Kutta Method (RK4) in solving first order Ordinary Differential Equations (ODE) with initial conditions. The proposed methods are quite efficient and practically well suited for solving these problems. Many examples are considered to validate the accuracy and easy application of the proposed methods. We compared the approximate solutions with the analytical solution. We found out that there is good agreement between the exact and approximate solutions. We also compared the performance and the computational effort of the methods. Besides, to achieve more accuracy in the solutions, the step size needs to be very, very small. Finally, the error terms have been analyzed for the proposed methods for different step sizes and compared also by appropriate examples to demonstrate reliability and efficiency. All results were obtained by MATLAB programing language.

Keywords**:** *Ordinary Differential Equations (ODE), Initial value Problems (IVP), Runge Kutta Method, Error Analysis*

1. Introduction

It is known generally that Differential Equations are among the most important Mathematical tools used in producing models in the engineering, mathematics, physics, aeronautics, elasticity, astronomy, dynamics, biology, chemistry, medicine, environmental sciences, economics, social sciences, banking and many other areas [1]. Many researchers have studied the nature of Differential Equations and many complicated systems that can be described quite precisely with mathematical expressions. Although there are many analytic methods for finding the solution of differential equations, there exist quite many differential equations that cannot be solved analytically [2]. This means that the solution cannot be expressed as the sum of a finite number of elementary functions (polynomials, exponentials, trigonometric, and hyperbolic functions). For simple differential equations, it is possible to find closed-form solutions [3]. But many differential equations arising in applications are so complicated that it is sometimes impractical to have solution formulas; even when a solution formula is available, it may involve integrals that can be calculated only by using a numerical quadrature formula. In either case, numerical methods provide a powerful alternative tool for solving the differential equations under the prescribed initial condition or conditions [3].

There are many types of practical numerical methods for solving initial value problems for ordinary differential equations. From our findings, the ancestor of all numerical methods in use today was developed by Leonhard Euler between 1768 and 1770 [4], improved Euler's method and Runge Kutta methods described by Carl Runge and Martin Kutta in 1895 and 1905 respectively [5]. We have excellent and exhaustive books which can be consulted, such as [1-3, 6-12].

From the literature review, we found that many authors have worked on numerical solutions of initial value problems using the fourth-order Runge Kutta method and many researchers have tried to solve initial value problems to get a higher accurate solution by applying numerous methods, such as the Euler method, the Runge Kutta method, the Adomian Decomposition Method, Hybrid method, Extrapolation method, and many other methods. See [13-21]. In this paper, we applied the fourth-order Runge Kutta method for solving initial value problems in ordinary differential equations.

A rigorous and elaborate numerical technique is the Runge Kutta method. This technique is the most widely used one since it gives starting values and is particularly suitable when the compilation of higher derivatives is complicated [15]**.** We used two examples of different kinds of ordinary differential equations to illustrate the proposed formulae. The results obtained from each of the numerical examples show that the convergence and error analysis which we presented clearly

Corresponding Author**:** Okeke A.A., Email: anyaokeke@gmail.com, Tel: +2348025388355

illustrate the efficiency of the method. However, the Runge Kutta method also has its advantages and disadvantages to use. It has the advantage of being the most widely used numerical weapon, since it gives reliable values, starting values and particularly suitable when the computation of higher-order derivatives are complicated. It also possesses the advantage of requiring only the function values at some selected points on the sub-intervals. It is easy to change step-length for special procedures necessary for starting, which minimize the computing time. However, the method is very laborious. It is a lengthy procedure and needs to check back the values computed earlier. The inherent error in the Runge Kutta method is hard to be estimated and the method has its limitation in solving certain types of differential equations only and the step-length is the key factor of the computation.

Lastly, this paper is structured as follows: Section 2: definitions and the general concept of differential equations,; Section 3: problem formulations; Section 4: error analysis, Section 5: numerical examples; Section 6: discussion of results; and the last section, the conclusion of the paper.

2. Definitions and General Concepts of Differential Equations

A differential equation is a mathematical equation for an unknown function of one or more variables that relates the values of the function itself and its derivatives of various orders.

2.1. **Definition**: The general form of a differential equation is as follows:

$$a_0(t)\frac{d^n y}{dt^n} + a_1(t)\frac{d^{n-1}y}{dt^{n-1}} + \cdots\cdots\cdots + a_{n-1}(t)\frac{dy}{dt} + a_n(t)y = f(t) \tag{2.1}$$

Here $a_i(t);\quad n = 1,2,3,\cdots\cdots\cdots$ and $f(t)$ is the function of t and $y = y(t)$ is an unknown function in terms of t.

2.2. **Definition:** An equation involving a function $y(t)$ of one independent variable (t) and its derivatives $y(t)\cdots\cdots\cdots y^n(t)$ is called an ordinary differential equation (ODE) of order n.

2.3. **Definition**: An implicit ODE of order n depending on $y^{(n)}$ has the form $F(t,y,y'(t),y"(t),\cdots y^{(n)}(t)) = 0$. In a special form, $F(t,y,y',y",\cdots y^{(n-1)}) = y^{(n)}$ is called an ODE in explicit form.

2.4. **Definition:** A partial differential equation for the function $u = u(t_1,t_2,t_3,\cdots\cdots\cdots,t_n)$ is of the form $F\left(t_1,t_2,t_3,\cdots,t_n,\frac{\partial u}{\partial t_1},\frac{\partial u}{\partial t_2}.\cdots,\frac{\partial u}{\partial t_n},\frac{\partial^2 u}{\partial t_1 \partial t_2},\frac{\partial^3 u}{\partial t_2 \partial t_3},\cdots\right) = 0$, where F is a linear function of u and its derivatives.

2.5. **Definition:** An initial value problem (IVP) is a differential equation such as $y'(t) = f(t,y(t))$, satisfies the following initial conditions (IC) $y(t_0) = y_0$; $y'(t_0) = y'_0$, where $t_0 \in I$, for some open interval $I \in \mathbb{R}$.

2.6. **Definition:** A boundary value problem is a differential equation together with a set of additional restraints, called boundary conditions. A solution to a boundary value problem is a solution to the differential equation which also satisfies given boundary conditions. The basic two point boundary value problem is given by$y'(t) = f(t,y(t))$ with $g(y(a),y(b)) = 0$.

2.7. **Definition:** A set of first order differential equation:

$y = f(t,y)$

$y_1 = f_1(t,y_1\cdots\cdots\cdots y_n)$

$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$

$y_n = f_n(t,y_1\cdots\cdots\cdots y_n)$

is called a first order system of ordinary differential equations.

2.8. **Definition**: A set of first order differential equation is called autonomous if all functions f_k on the right-hand side do not depend on t.

$y = f(y)$or

$y_1 = f_1(y_1\cdots\cdots\cdots y_n)$

$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$

$y_n = f_n(y_1\cdots\cdots\cdots y_n)$

2.9. **Definition**: A solution of a system of ordinary differential equation is the form $y(t) = y_0$ with $y_0 \in \mathbb{R}^2$ is called an equilibrium solution. $y(t) = y(t+T)$with$T > 0$ is called a period. The parameter T denote period.

2.10. **Definition:** An equilibrium solution $y(t) = y_0$of an autonomous system is called stable if any solution in the neighborhood of y_0 will always approach this equilibrium solution as $t \longrightarrow \infty$. In this case $y(t) \longrightarrow y_0$ for $t \rightarrow \infty$. Otherwise the equilibrium solution is called unstable.

3. Problem Formulation

Runge Kutta method is a technique for approximating the solution of ODEs. This technique was developed by two German Mathematicians, Karl Runge by 1894 and extended by Wilhelm Kutta a few years later. The technique is popular because it is efficient, quite accurate, stable, and used in most computer programmes for differential equations. The Runge Kutta methods are distinguished by their order in the sense that they agree with Taylor's series solution up to terms of h^r, where r is the order of the method. It does not demand prior computational of higher derivatives of $y(x)$ as in Taylor's series method. The under-listed are the order of the Runge Kutta Method:

(i) Runge Kutta Method of order one is called Euler's Method,
(ii) Runge Kutta Method of order two is the same as modified Euler's or Heun's Method,
(iii) The fourth-order Runge Kutta Method is called Classical Runge Kutta Method.

In this paper, we shall only focus on the fourth-order Runge Kutta Method.

3.1. The Derivative of the Fourth Order Runge Kutta Method

We shall derive the formula of fourth-order Runge Kutta method to obtain an approximate numerical solution of the first order differential equation $y' = f(x,y)$ with the initial condition $y(x_0) = y_0$ and it is assumed that is not a singular point. Let us take the first-order differential equation

$$y' = \frac{dy}{dx} = f(x,y); \quad y(x_0) = y_0 \qquad (3.1)$$

Let $h = x_1 - x_0$, from Taylor's series expansion, we have

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

or$y(x+h) - y(x) = hy'(x) + \frac{h^2}{2!}y''(x) + \frac{h^3}{3!}y''{}'(x) + \cdots$ (3.2)

Differentiating (3.1) partially with respect to variables x & y , we get

$$y' = f(x,y) = f$$
$$y'' = f'(x,y) = f_x + ff_y$$
$$y'' = f''(x,y) = f_{xx} + 2ff_{xy} + f^2f_{yy} + f_xf_y + ff_y{}^2$$
$$y^{iv} = f'''(x,y) = (f_{xxx} + 3ff_{xxy} + f^2f_{xyy} + f^3f_{yyy} + f_y\left(f_{xx} + 2ff_{xy} + f^2f_{yy}\right) + 3\left(f_x + ff_y\right)\left(f_{xy} + ff_{yy}\right) + f_y{}^2(f_x + ff_y)$$

Let us introduce the following convenient form

$$F_1 = f_x + ff_y, \quad F_2 = f_{xx} + 2ff_{xy} + f^2f_{xy}, \quad F_3 = f_{xxx} + 3ff_{xxy} + f^2f_{xxy} + f^2f_{yyy}$$

Then we get as

$$y' = f, \quad y'' = F_1, \quad y''' = F_2 + f_yF_1$$
$$y^{i4} = F_3 + f_yF_2 + 3F_1\left(f_{xy} + ff_{yy}\right) + F_1\left(f_{xy} + ff_{yy}\right) + F_1f_y{}^2$$

If we now put them into (3.2), we obtain

$$y(x+h) - y(x)$$
$$= hf + \frac{h^2}{2!}F_1 + \frac{h^3}{3!}\left(F_1 + f_yF_1\right)$$
$$+ \frac{h^4}{4!}\left\{F_3 + f_yF_2 + 3F_1\left(f_{xy} + ff_{xy}\right) + F_1\left(f_{xy} + ff_{yy}\right) + F_1f_y{}^2\right\} \qquad (3.3)$$

Now, we shall develop a fourth-order formula. In order to develop the Runge Kutta formula to find the co-efficient a, b, c, d, m, n & p from below

$$k_1 = hf(x,y) = hf,$$
$$k_2 = hf(x + mh, y + mk_1)$$
$$k_3 = hf(x + nh, y + nk_2)$$
$$k_4 = hf(x + ph, y + pk_3) \qquad (3.4)$$

Our aim then is Δy will be expressed in the form

$$\Delta y = y(x,y) - y(x) = ak_1 + bk_2 + ck_3 + dk_4 \qquad (3.5)$$

At this stage, we may use Taylor's series expansion for two variables as the followings

$$k_1 = hf,$$
$$k_2 = h[f + mhF_1 + \tfrac{1}{2}m^2h^2F_2 + \tfrac{1}{6}m^3h^3F_3 + \cdots\cdots\cdots$$
$$k_3 = h[f + nhF_1 + \frac{1}{2}h^2\left(n^2F_2 + 2mnf_yF_1\right)$$
$$+ \tfrac{1}{6}h^3\left\{n^3F_3 + 3m^2nf_yF_2 + 6mn^2F_1\left(F_{xy} + ff_{yy}\right)\right\} + \cdots\cdots\cdots]$$

$k_4 = h[f + phF_1 + \frac{1}{2}h^2(p^2F_2 + 2npf_yF_1)$

$+\frac{1}{6}h^3\{p^3F_3 + 3n^2pf_yF_2 + 6np^2F_1(F_{xy} + ff_{yy}) + 6mnpF_1{f_y}^2\} + \cdots\cdots\cdots]$

Substituting the values of $k_1, k_2, k_3 \& k_4$ in (3.5), we obtain

$y(x+h) - y(x) = ahf + bh[f + mhF_1 + \frac{1}{2}m^2h^2F_2 + \frac{1}{6}m^3h^3F_3 + \cdots\cdots\cdots +$

$ch[f + nhF_1 + \frac{1}{2}h^2(n^2F_2 + 2mnf_yF_1) + \frac{1}{6}h^3\{n^3F_3 + 3m^2nf_yF_2 + 6mn^2F_1(F_{xy} + ff_{yy})\} + \cdots\cdots\cdots]$

$+dh[f + phF_1 + \frac{1}{2}h^2(p^2F_2 + 2npf_yF_1)$

$+\frac{1}{6}h^3\{p^3F_3 + 3n^2pf_yF_2 + 6np^2F_1(F_{xy} + ff_{yy}) + 6mnpF_1{f_y}^2\} + \cdots\cdots\cdots]$

This can be represented as

$y(x+h) - y(x) = (a+b+c+d)hf + (bm+cn+dp)h^2F_2 +$

$(bm^2 + cn^2 + dp^2)\frac{h^3F_1}{2} + (bm^3 + cn^3 + dp^3)\frac{h^4F_2}{6} +$

$(cmn + dnp)h^3f_yF_1 + (cm^2n + dn^2p)h^4f_yF_2 +$

$(cmn^2 + dnp^2)h^4F_1(f_{xy} + ff_{yy}) + dmnph^4{f_y}^2F_1 + \cdots\cdots\cdots$ (3.6)

When we compare (3.3) and (3.6), we get

$a+b+c+d = 1, bm+cn+dp = \frac{1}{2}, bm^2+cn^2+dp^2 = \frac{1}{3}, bm^3+cn^3+dp^3 = \frac{1}{4},$

$cmn + dnp = \frac{1}{6},\ cm^2n + dn^2p = \frac{1}{12}, \qquad cmn^2 + dnp^2 = \frac{1}{8},\ dmnp = \frac{1}{24}$

By solving the above equations, we obtain

$m = n = \frac{1}{2},\ p = 1,\ a = d = \frac{1}{6}, \quad b = c = \frac{1}{3}$

Now we put these values in (3.4) and (3.5), we get the fourth-order Runge Kutta formulae as follows:

$k_1 = hf(x,y) = hf,\ k_2 = hf\left(x + \frac{h}{2}, y + \frac{k_1}{2}\right), k_3 = hf(x + \frac{h}{2}, y + \frac{k_2}{2}), k_4 = hf(x+h, y+k_3)$

$\Delta y = y(x+h) - y(x) = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

When the initial values are (x_0, y_0), then, the first increment in y is computed from the given formulae below

$k_1 = hf(x_0,y_0) = hf, k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right), k_3 = hf(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}),$

$k_4 = hf(x_0 + h, y_0 + k_3)$

$\Delta y = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

or, $y(x_0 + h) = y(x_0) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

or, $y_1 = y_0 + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$

Hence, the general fourth-order Runge Kutta formulae for the n^{th} interval is given by the followings:

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

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

$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ (3.7)

4. Error Analysis

Numerical stability and errors are well discussed in depth in [10-12]. There are two types of errors in the numerical solution of ODEs: Round-off errors and truncation errors. *Round-off error* occurs because computers use a fixed number of bits and hence fixed the number of binary digits to represent numbers. In a numerical computation round-off errors are introduced at every stage of computation. Hence though an individual round-off error due to a given number at a given numerical step may be small but the cumulative effect can be significant. When the number of bits required for representing a number is less than the number is usually rounded to fit the available number of bits. This is done either by chopping or by symmetric rounding. Also, *truncation error*arises when you use an approximation in place of an exact expression in a mathematical procedure.

One of the serious drawbacks of the Runge Kutta method is error estimation [10]. The direct methods of estimating the error of higher-order Runge Kutta formulae are very complicated and time-consuming. Moreover, it is possible to compute the errors in laborious ways, which are very hard, involving higher-order partial derivatives. We shall first estimate the error in second-order Runge Kutta formulae and the errors for higher orders can be obtained by generalizing the computed error. We get the second-order Runge Kutta formulae as follows:

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2)$$
$$k_1 = hf(x_n, y_n)$$
$$k_2 = hf(x_n + h, y_n + k_1) \qquad (4.1)$$

Now, the truncated error is given by the following formula

$$E_r = y(x_{n+1}) - y_{n+1} \qquad (4.2)$$

Now expanding $y(x_{n+1})$ by Taylor's series expansion, we get

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + hy'(x_n) + \frac{h^2}{2!}y''(x_n) + \frac{h^3}{3!}y'''(x_n) + \frac{h^4}{4!}y'''(x_n) \cdots\cdots\cdots$$
$$= y_n + hf + \frac{h^2}{2!}(f_x + ff_y) + \frac{h^3}{3!}(f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_x f_y + ff_y{}^2) + o(h^4) \qquad (4.3)$$

We may use Taylor's series expansion in (4.1), we get

$$k_1 = hf$$
$$k_2 = h\left[f + h(f_x + ff_y) + \frac{h^2}{2}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \cdots\cdots\cdots\right]$$
$$= hf + h^2(f_x + ff_y) + \frac{h^3}{2}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \cdots\cdots\cdots$$
$$y_{n+1} = y_n + \frac{1}{2}\left[hf + hf + h^2(f_x + ff_y) + \frac{h^3}{2}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \cdots\cdots\cdots\right]$$
$$or\ \ y_{n+1} = y_n + hf + \frac{h^2}{2}(f_x + ff_y) + \frac{h^3}{4}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \qquad (4.4)$$

Now, substituting (4.3) and (4.4) in (4.2), we get

$$E_r = \left[y_n + hf + \frac{h^2}{2}(f_x + ff_y) + \frac{h^3}{6}(f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_x f_y + ff_y{}^2) + \cdots\right] -$$
$$\left[y_n + hf + \frac{h^2}{2}(f_x + ff_y) + \frac{h^3}{4}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \cdots\right]$$
$$= \left(\frac{h^3}{6} - \frac{h^2}{4}\right)(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \frac{h^3}{6}(f_x f_y + ff_y{}^2 + \cdots$$
$$= -\frac{h^3}{12}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + \frac{h^3}{6}(f_x f_y + ff_y{}^2) + \cdots$$
$$= -\frac{h^3}{12}(f_{xx} + 2ff_{xy} + f^2 f_{yy}) + 2f_y - 2ff_y{}^2 \cdots \qquad (4.5)$$

Hence, (4.5) shows that the truncation error of the second-order Runge Kutta formula is of order h^3. Similarly, we can show that the truncation errors in the third-order, fourth-order Runge Kutta formula are of $h^4 \& h^5$ respectively.
Hence, by applying Taylor's series expansion as above manner, we get the truncation error of the nth -order Runge Kutta formulae of order h^{n+1} as follows

$$E_r = ch^{n+1}y^{n+1} \qquad (4.6)$$

5. Numerical examples

In this section, we present two numerical examples to verify the accuracy of the proposed method. The numerical results and errors are computed and the findings are represented graphically. The computations were done using MATLAB programing language. The convergence of the IVP is calculated $e_n = |y(x_n) - y_n| < \delta$, where $y(x_n)$ represents the exact solution and δ depends on the problem which varies from 10^{-7} while the absolute error is computed by $|y(x_n) - y_n|$.

Example 1: We consider the initial value problem $y' = 2y + 4 - x,\ y(0) = 0.5$, on the interval $0 \le x \le 1$. The exact solution of the given problem is given by $y(x) = -\frac{7}{4} + \frac{1}{2}x + \frac{9}{4}e^{2x}$. The results obtained are shown in Tables 1(a) and Table 1(b) and graphically displayed in Figures 1-3.

Table 1. (a) Numerical approximations for different step size.

n	x_n	Exact Solution y_n	Approximation		
			$h = 0.1$	$h = 0.05$	$h = 0.025$
0	0.0	0.500000000000000	0.500000000000000	0.500000000000000	0.500000000000000
1	0.1	0.482991259254641	0.482991276041667	0.482991260282179	0.482991259318196
2	0.2	0.481443217233737	0.481443249170363	0.481443219188586	0.481443217354649
3	0.3	0.494601846762876	0.494601892331463	0.494601849552140	0.494601846935397
4	0.4	0.521749895006882	0.521749952801790	0.521749898544523	0.521749895225692
5	0.5	0.562205089964132	0.562205158684403	0.562205094170517	0.562205090224305
6	0.6	0.615318434431166	0.615318512873659	0.615318439232651	0.615318434728148
7	0.7	0.680472583171637	0.680472670224579	0.680472588500170	0.680472583501218
8	0.8	0.757080299231656	0.757080393868592	0.757080305024406	0.757080299589949
9	0.9	0.844582985541526	0.844583086815646	0.844582991740541	0.844582985924948
10	1.0	0.942449288132117	0.942449395170920	0.942449294683989	0.942449288537364

Table 1(b) Observed absolute errors for example 1.

n	x_n	Errors		
		$h = 0.1$	$h = 0.05$	$h = 0.025$
0	0.0	0.0000000000000000	0.0000000000000000	0.0000000000000000
1	0.1	0.0000000167870257	0.0000000010275381	0.0000000000635554
2	0.2	0.0000000319366255	0.0000000019548488	0.0000000001209114
3	0.3	0.0000000455685865	0.0000000027892643	0.0000000001725214
4	0.4	0.0000000577949080	0.0000000035376411	0.0000000002188106
5	0.5	0.0000000687202708	0.0000000042063848	0.0000000002601734
6	0.6	0.0000000784424928	0.0000000048014847	0.0000000002969818
7	0.7	0.0000000870529420	0.0000000053285324	0.0000000003295807
8	0.8	0.0000000946369367	0.0000000057927503	0.0000000003582932
9	0.9	0.0000001012741199	0.0000000061990149	0.0000000003834224
10	1.0	0.0000001070388032	0.0000000065518719	0.0000000004052469

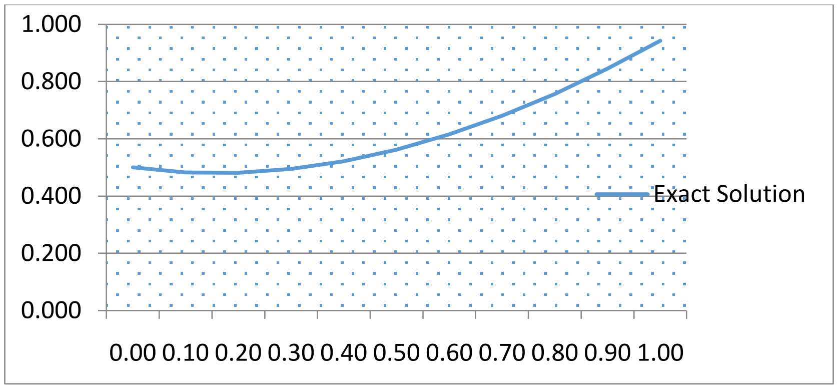

Figure 1: Exact Numerical Solution

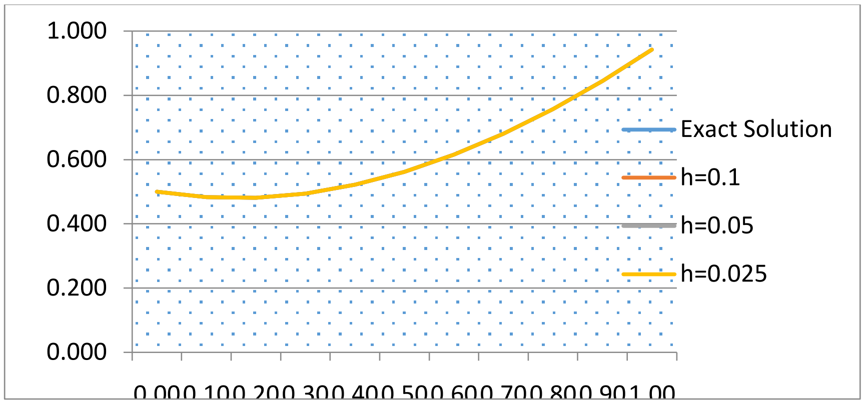

Figure 2: Exact Solution and Numerical Approximation for different step sizes

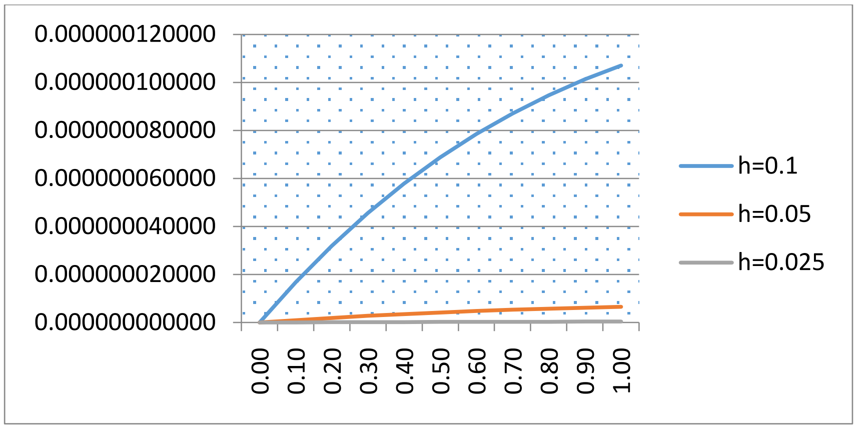

Figure 3: Error for different step sizes

Example 2: We consider the initial value problem $y^{'} = \left(\frac{2x}{1+x^2}\right)y, \quad y(0) = 2.15$, on the interval $0 \le x \le 1$. The exact solution of the given problem is given by $y(x) = 2.15(1 + x^2)$. The results obtained are shown in Tables 2(a) and Table 2(b) and graphically displayed in Figures 4-6.

Table 2. (a) Numerical approximations for different step size.

n	x_n	Exact Solution y_n	Approximation		
			$h = 0.1$	$h = 0.05$	$h = 0.0125$
0	0.0	2.1500000000000000	2.1499999999999999	2.1499999999999999	2.1499999999999999
1	0.1	2.1715000000000000	2.2189261331225589	2.1953422717100741	2.1774571783101435
2	0.2	2.2360000000000000	2.3334262590434469	2.2848168625939533	2.2481613547284356
3	0.3	2.3435000000000000	2.4953349088988657	2.4193352905262206	2.3623386200456209
4	0.4	2.4940000000000000	2.7063977618557740	2.5997528348958383	2.5201991763896565
5	0.5	2.6875000000000000	2.9682198057208375	2.8268457712415058	2.7219321270457231
6	0.6	2.9240000000000000	3.2822334942455038	3.1012989059616269	2.9677029158303689
7	0.7	3.2035000000000000	3.6496866641324219	3.4237025865589228	3.2576530871066636
8	0.8	3.5260000000000000	4.0716461444112806	3.7945568275012476	3.5919017080978404
9	0.9	3.8915000000000000	4.5490116345173126	4.2142797983894926	3.9705477490335759
10	1.0	4.3000000000000000	5.0825348832800321	4.6832183031819152	4.3936728449178357

Table 2. (b) Observed absolute errors for example 1.

n	x_n	Errors		
		$h = 0.1$	$h = 0.05$	$h = 0.0125$
0	0.0	0.0000000000000000	0.0000000000000000	0.0000000000000000
1	0.1	0.0474261331225589	0.0238422717100741	0.0059571783101435
2	0.2	0.0974262590434472	0.0488168625939536	0.0121613547284358
3	0.3	0.1518349088988655	0.0758352905262205	0.0188386200456208
4	0.4	0.2123977618557738	0.1057528348958381	0.0261991763896563
5	0.5	0.2807198057208376	0.1393457712415058	0.0344321270457231
6	0.6	0.3582334942455043	0.1772989059616275	0.0437029158303694
7	0.7	0.4461866641324219	0.2202025865589228	0.0541530871066636
8	0.8	0.5456461444112803	0.2685568275012473	0.0659017080978401
9	0.9	0.6575116345173129	0.3227797983894929	0.0790477490335761
10	1.0	0.7825348832800320	0.3832183031819150	0.0936728449178359

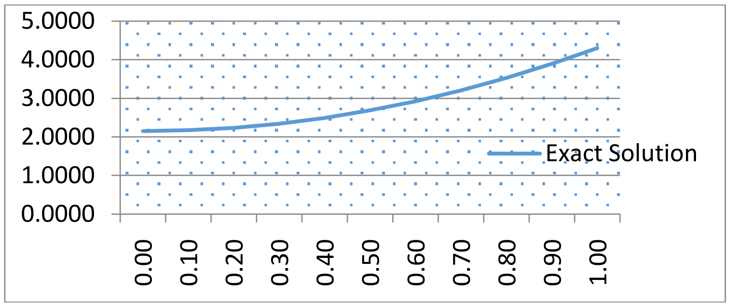

Figure 4: Exact Numerical Solution

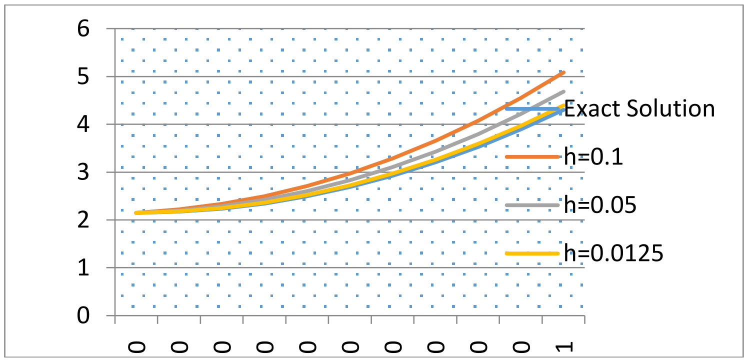

Figure 5: Exact Solution and Numerical Approximation for different step sizes

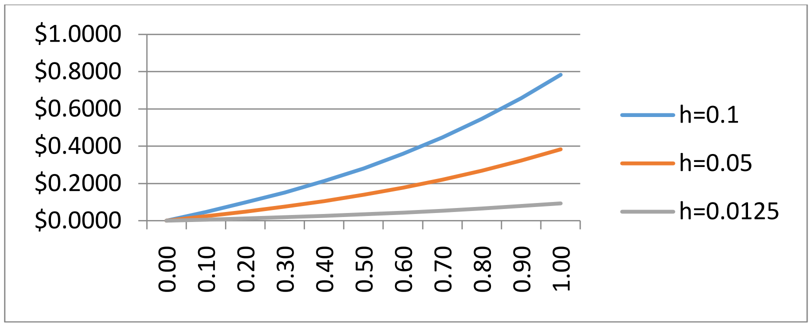

Figure 6: Error for different step sizes

6. Discussion and Results

The obtained results are displayed in Table 1(a), (b) and Table 2(a), (b) and graphically represented in Figures (1-3) and Figures (4-6) respectively. The approximate solutions and absolute errors are calculated using MATLAB programming language with different step sizes and also computed with the exact solution. From the tables, we observed that good results are obtained even for a reasonably large step size but the approximations enhanced by reducing the step size. Our proposed method gives very good results when compared with the exact solutions. Hence, we say that a numerical solution converges to the exact solution if reducing the step size leads to decrease errors such that in the limit when the step size reduces to zero the errors go to zero.

7. Conclusion

In this paper, the fourth-order Runge Kutta has been presented for solving first order Ordinary Differential Equations (ODE) with initial conditions. To find more accurate results of the numerical solution, we reduced the step size to very, very small. From our tables and figures, we analyzed that the solution for the proposed method converges to the exact solution for decreasing the step size h. The numerical solutions obtained are in good agreement with the exact solutions. The results of the two problems guarantee consistency, convergence, and stability. Thus, the accuracy increase with decrease step size, we may conclude that this method is applied to solve first-order ODEs with initial conditions to find the anticipated accuracy.In our subsequent research, we shall examine the comparison of the fourth-order Runge Kutta method with other existing methods like the Adomian decomposition.

References

[1] Lambert J. D., (2000). *Computational Methods in Ordinary Differential Equations*. New York: Wiley & Sons: 21-205.

[2] Butcher J. C., (2003).*Numerical Methods for Ordinary Differential Equations.* West Sussex: John Wiley & Sons Ltd. 45-95.

[3] Atkinson K, Han W, Stewart D, (2009). *Numerical Solution of Ordinary Differential Equations.* New Jersey: John Wiley & Sons, Hoboken: 70-87.

[4] Euler L, (1768) Institutions Calculi Integralis Volumen Primum, Opera Omnia. Vol. XI, B. G. Teubneri Lipsiaeet Berolini MCMXIII: 21-228.

[5] Euler L, (1913) *De integration aequationum di erentialium per approximationem,* In Opera Omnia, 1st series, Vol II, Institutiones Calculi Integralis, Teubner, Leipzig and Berlin, 424434.

[6] Brenan K. E., Campbell S. L., Petzold L. R. (1989).*Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* New York: Society for Industrial and Applied Mathematics: 76-127.

[7] Lambert J. D., (1999).*Numerical Methods for Ordinary Differential Systems: The Initial value Problem.* New York: John Wiley &Sons: 149-205.

[8] Boyce W. E. and DiPrima R. C. (2000). Elementary Differential Equations and Boundary Value Problems. New York: John Wiley & Sons, Inc.: 419-471.

[9] Carnahan B., Luther H. A.,Wikes J. O. (1990). Applied Numerical Methods, Florida: Krieger Publishing Company: 341-386.

[10] Fatunla S. O. (1988).*Numerical Methods for Initial Value Problems in Ordinary Differential equations*. Boston: Academic Press, INC: 41-62.

[11] Conte S. D. and Boor C. D. (1980).*Elementary Numerical Analysis: Algorithmic Approach*. New York: McGraw-Hill Book Company: 356-387.

[12] Bosede O, Emmanuel F, Temitayo O, (2012) On Some Numerical Methods for Solving Initial Value Problems in Ordinary Differential Equations. *IOSR Journal of Mathematics (IOSRJM)* Vol. 1 (3): 25-31.

[13] Okeke A. A., Tumba P., Gambo J. J. (2019). The Use of Adomian Decomposition Method in Solving Second Order Autonomous and Non-autonomous Ordinary Differential Equations, (2019).*International Journal of Mathematics and Statistics Invention (IJMSI).* Vo. 7(1): 91-97. Available from: http://www.ijmsi.org/Papers/Volume.7.Issue.1/L07019197.pdf.

[14] Okeke A. A., Tumba P., Anorue O. F., Dauda A. A (2019).Analysis and Comparative Study of Numerical Solutions of Initial Value Problems (IVP) in Ordinary Differential Equations (ODE) With Euler and Runge Kutta Methods, (2019).*American Journal of Engineering Research (AJER),* vol. 8, no. 8, pp. 40-53. Available from: http://www.ajer.org/papers/Vol-8-issue-8/F0808014053.pdf.

[15] Fadugba S. E., Ogunrinde B, Okunlola T, (2012). Euler's Method for Solving Initial Value Problems in Ordinary Differential Equations. The Pacific Journal of Science and Technology, Vol. 13 (2): 152-158.

[16] Islam Md. A. (2015). Accurate Solutions of Initial Value Problems for Ordinary Differential Equations with the Fourth Order Runge Kutta Method. *Journal of Mathematics Research*, Vol. 7 (3): 41-45.

[18] Islam Md, (2015). Accurate Analysis of Numerical Solutions of Initial Value Problems (IVP) for ordinary differential equations (ODE). *IOSR Journal of Mathematics (IOSR-JM)*, Vol. 11 (3): 18- 23.

[19] Jamali N, (2019). Analysis and Comparative Study of Numerical Methods to Solve Ordinary Differential Equation with Initial Value Problem. *International Journal of Advanced Research (IJAR)*. Vol. 7(5): 117-128: Available from: http://www.journalijar.com/uploads/536_IJAR-27303.pdf.

[20] Hossain B. B.,Hossain M. J., MiahMd, et al. (2017). A Comparative Study on Fourth Order and Butcher′s Fifth Order Runge Kutta Methods with Third Order Initial Value Problem (IVP). *Applied and Computational Mathematics*, Vol. 6(6): 243-253. Available from: doi:10.11648/j.acm.20170606.12.

[21] Fadugba S. E., Olaosebikan T. E. (2018) Comparative Study of a Class of One-Step Methods for the Numerical Solutions of Some Initial Value Problems in ordinary Differential Equations. *Research Journal of Mathematics and Computer Science:* 2-9: Available from:https://escipub.com/Articles/RJMCS/RJMCS-2017-12-1801.

[22] Hamed A. B., Alrhaman I. Y, Sani I. (2017). The accuracy of Euler and modified Euler technique for First Order Ordinary Differential Equations with initial conditions. American journal of Engineering Research (AJER), Vol. 6(9): 334-338.