IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 16, Issue 4 Ser. IV (Jul.–Aug. 2020), PP 29-39
www.iosrjournals.org

On The Derivation and Implementation of a Fifth-Stage Fourth-Order Runge–Kutta Formula for Solving Initial Value Problems in Ordinary Differential Equations

Ononogbo Chibuike Benjamin[1], Aziegbemi B. Okisamen[2]
[1](Department of Mathematics, Ambrose Alli University, Nigeria)
[2](Department of Mathematics and Computer Science, Indiana University of Pennsylvania, USA)

Abstract:
In this paper, a new fifth-stage fourth-order Runge–Kutta formula was derived for solving initial value problems (IVPs) in Ordinary Differential Equation, which was implemented and compared with classical Runge-Kutta formula through the computation of some tested initial value problemsin other to determine the level of performance, consistency and accuracy.We discovered that errors are minimal with our new method.Errors in the new method and that of classical Runge-Kutta method were plotted with MATLAB to determine their curves.Our new method was tested further on the Kermack-McKendrick SIR model for the course of an epidemic in a population which computes the number of Susceptible, Infected, and Recovered people in a population at any time. The codes and plots were built in Python. We called this method ***BO4 method****.*

Keywords:
Initial value problems, Taylor series, consistency, convergence, stability, and error analysis

---

Date of Submission: 11-08-2020 Date of Acceptance: 27-08-2020

I. Introduction

Ordinary differential equations arise frequently in several models of mathematical physics, biological Sciences, engineering and applied mathematics. Unfortunately, many cannot be solved exactly. This is why numerical treatment is very important and provides a powerful alternative tool for solving the differential equations, which are modeled as initial value problems (IVPs) [6]. Runge-Kutta methods provide a unique way for solving initial value problems for system of ordinary differential equation of the form:

$$y' = f(x, y), \quad a \le x \le b, \qquad y(a) = y_0$$

With a given step length h through the interval $[a, b]$, successively producing approximations y_n to y_{n+1} [9].

We deal broadly with the step by step derivation and stability of our new method. Recent work on Runge-Kutta methods includes [1], [4] and more recent work are [2] and [7]. More on the general disparity of the Runge-Kutta method can be seen in [3] and [5].

This paper has the following structure: Section 2 presents derivation of the method. Section 3, presents the consistency and convergence of the method. Section 4, presents the stability of the method. Section 5 presents the implementation in comparison with classical Runge-Kutta method with tested examples, a display of solution tables will be provided by way of comparison with some existing methodsand the error analysis of the method.Finally, section 6 presents the summaryand conclusion.

II. Derivation of BO4 Method

In this section we will introduce the basic tools required for the derivation of fifth- stage fourth- order Runge-Kutta method. Consider the general initial value problem in ordinary differential equations of the form;

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0 \qquad \dots (2.1)$$

The main aim in this section is to obtain approximate solution of y(x), the approach of the fifth- stage fourth-order Runge-Kutta method is proceed to evaluate y_{n+1} as an approximation to $y(x_{n+1}) = y(x_n + h)$, an important special case "without the loss of generality" if the function f does not depend on x but y, by setting $x' = 1$ then equation (2.1) reduces to so called "autonomous" and written in the form

$$y'(x) = f(y(x)), y(x_0) = y_0 \dots (2.2)$$

The proposed explicit Runge-Kutta method of order four with fifth- stages, denoted by
$k_1, \dots, k_5$ for one step, according to [7], the solution of equation (2.2) can be derive from the general Runge-Kutta formula given by;

$$y_{n+1} - y_n = h\,\Phi(x_n, y_n, h) \quad \dots (2.3)$$

$$\Phi(x_n, y_n, h) = \sum_{i=1}^{R} b_i k_i \text{ for } i = 1,2,\dots,5 \quad \dots (2.4)$$

$$k_1 = f(x_n, y_n) = f \quad \dots (2.5)$$

$$k_i = f\left(x_n + hc_i, y_n + h\sum_{j=1}^{i-1} a_{ij} k_j\right) \text{for} i = 2,3,\dots,5 \quad \dots (2.6)$$

$$c_i = \sum_{j=1}^{i-1} a_{ij} \quad , i = 2,3,\dots,5 \quad \dots (2.7)$$

Where the coefficients a_{ij}, b_i, c_i $(c_i \in [0,1])$are the constants, h is the step size.
From equation (2.6) we have:

$k_2 = f(x_n + \hbar c_2, y_n + \hbar(a_{21}k_1))$
$k_3 = f(x_n + \hbar c_3, y_n + \hbar(a_{31}k_1 + a_{32}k_2))$
$k_4 = f(x_n + \hbar c_4, y_n + \hbar(a_{41}k_1 + a_{42}k_2 + a_{43}k_3))$
$k_5 = f(x_n + \hbar c_5, y_n + \hbar(a_{51}k_1 + a_{52}k_2 + a_{53}k_3 + a_{54}k_4))$

For the purpose of linearity, the above parameters will be modified as follows:

$$a_{21} = a_1, \quad a_{31} = a_2, \quad a_{32} = a_3, \quad a_{41} = a_4, \quad a_{42} = a_5,$$
$$a_{43} = a_6, \quad a_{51} = a_7, \quad a_{52} = a_8, \quad a_{53} = a_9, \quad a_{54} = a_{10},$$

Substituting we have:

$$k_2 = f\big(x_n + hc_2, y_n + h(a_1k_1)\big) \quad \dots (2.8)$$
$$k_3 = f(x_n + hc_3, y_n + h(a_2k_1 + a_3k_2)) \quad \dots (2.9)$$
$$k_4 = f(x_n + hc_4, y_n + h(a_4k_1 + a_5k_2 + +a_6k_3)) \quad \dots (2.10)$$
$$k_5 = f(x_n + hc_5, y_n + h(a_7k_1 + a_8k_2 + a_9k_3 + +a_{10}k_4)) \quad \dots (2.11)$$

$\sum_{i=1}^{5} b_i = 1$ which the final result depend on the derivatives, the approach of R-K method is expanding $k_{i's}$in Taylor's series expansion, after some algebraic simplification this expansion is equated to the exact solution $y(x_0 + \hbar)$ that is given by the Taylor's series:

$$y(x_0 + h) = y(x_0) + hy^{(1)}(x_0) + \frac{1}{2!}h^2y^{(2)}(x_0) + \frac{1}{3!}h^3y^{(3)}(x_0) + \frac{1}{4!}h^4y^{(4)}(x_0) + O(h^5) \quad \dots (2.12)$$

The first step is to calculate the successive derivatives of $y^{(1)}, y^{(2)}$, ...up to fourth order for equation (2.2) as follows:

$y^{(1)} = f(x, y) = f = k_1$
$y^{(2)} = ff_y{=}k_1f_y$
$y^{(3)} = ff_y^2 + f^2f_{yy} \;= k_1f_y^2 + k_1^2f_{yy}$
$y^{(4)} = ff_y^3{+}4f^2f_yf_{yy} + f^3f_{yyy} \;= k_1f_y^3 + 4k_1^2f_yf_{yy} + k_1^3f_{yyy}$

Where f_y represent the partial derivation of f with respect to$y$. substituting $y^{(1)}, y^{(2)}, y^{(3)}, y^{(4)}$ into equation (2 .12) we have the Fourth order Taylor's expansion as:

$$y_{n+1} - y_n = hk_1 + \frac{1}{2}h^2k_1f_y + \frac{1}{6}h^3\big(k_1f_y^2 + k_1^2f_{yy}\big) + \frac{1}{24}h^4\big(k_1f_y^3 + 4k_1^2f_yf_{yy} + k_1^3f_{yyy}\big) \quad \dots (2.13)$$

Equation (2.13) will be use to compare with Taylor expansions ofk_is.

For easy computation and convenience, we set

$$c_3 = (a_2 + a_3), \quad c_4 = (a_4 + a_5 + a_6), \quad c_5 = (a_7 + a_8 + a_9 + a_{10})$$

Taking the Taylor series expansion of equation (2.8), to (2.11) about the point (y_n), i.e., neglecting all the derivatives of x and leaving those of $y$alone, we have:

$$k_2 = \sum_{i=0}^{\infty}\frac{1}{i!}\left(ha_1k_1\frac{d}{dy}\right)^i f(y) \qquad \dots(2.14a)$$

$$k_3 = \sum_{i=0}^{\infty}\frac{1}{i!}\left(h(a_2k_1 + a_3k_2)\frac{d}{dy}\right)^i f(y) \qquad \dots(2.14b)$$

$$k_4 = \sum_{i=0}^{\infty}\frac{1}{i!}\left(h(a_4k_1 + a_5k_2 + a_6k_3)\frac{d}{dy}\right)^i f(y) \qquad \dots(2.14c)$$

$$k_5 = \sum_{i=0}^{\infty}\frac{1}{i!}\left(h(a_7k_1 + a_8k_2 + a_9k_3 + a_{10}k_4)\frac{d}{dy}\right)^i f(y) \qquad \dots(2.14d)$$

Expanding equation (2.14a) to (2.14d) with Maple we have:

$$k_2 = k_1 + ha_1k_1f_y + \frac{1}{2}h^2a_1^2k_1^2f_{yy} + \frac{1}{6}h^3a_1^3k_1^3f_{yyy}$$

$$k_3 = k_1 + k_1c_3f_yh + \frac{1}{2}h^2k_1^2c_3^2f_{yy} + h^2a_3a_1k_1f_y^2 + \frac{1}{2}h^3a_3a_1k_1^2(a_1 + 2c_3)f_{yy}f_y + \frac{1}{6}h^3k_1^3c_3^3f_{yyy}$$

$$k_4 = k_1 + k_1c_4f_yh + h^2k_1(a_1a_5 + a_6c_3)f_y^2 + \frac{1}{2}h^2k_1^2c_4^2f_{yy} + h^3a_6a_3a_1k_1f_y^2 + \frac{1}{2}h^3k_1^2(a_1^2a_5 + 2a_1a_5c_4 + a_6c_3^2 + 2a_6c_3c_4)f_{yy}f_y + \frac{1}{6}h^3k_1^3c_4^3f_{yyy}$$

$$k_5 = k_1 + k_1c_5f_yh + h^2k_1(a_1a_8 + a_9c_3 + a_{10}c_4)f_y^2 + \frac{1}{2}h^2k_1^2c_5^2f_{yy} + h^3k_1(a_1a_3a_9 + a_1a_5a_{10} + a_6a_{10}c_3)f_y^3 + \frac{1}{2}h^3k_1^2(a_1^2a_8 + 2a_1a_8c_5 + a_9c_3^2 + 2a_9c_3c_5 + a_{10}c_4^2 + 2a_{10}c_4c_5)f_{yy}f_y + \frac{1}{6}h^3k_1^3c_5^3f_{yyy}$$

Substituting all $k_1, \dots, k_5$ into equation (2.4) and simplifying we have:

$$\begin{aligned}\Phi(y_n,h) = {} & b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4 + b_5k_5 \\ = {} & k_1(b_1 + b_2 + b_3 + b_4 + b_5) + hk_1(a_1b_2 + b_3c_3 + b_4c_4 + b_5c_5)f_y \\ & + h^2k_1(a_1a_3b_3 + a_1a_5b_4 + a_1a_8b_5 + a_6b_4c_3 + a_9b_5c_3 + a_{10}b_5c_4)f_y^2 \\ & + \frac{1}{2}h^2k_1^2(a_1^2b_2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2)f_{yy} \\ & + h^3k_1(a_1a_3a_6b_4 + a_1a_3a_9b_5 + a_1a_5a_{10}b_5 + a_6a_{10}b_5c_3)f_y^3 \\ & + \frac{1}{2}h^3k_1^2(a_1^2a_3b_3 + a_1^2a_5b_4 + a_1^2a_8b_5 + 2a_1a_3b_3c_3 + 2a_1a_5b_4c_4 + 2a_1a_8b_5c_5 + a_6b_4c_3^2 \\ & + 2a_6b_4c_3c_4 + a_9b_5c_3^2 + 2a_9b_5c_3c_5 + a_{10}b_5c_4^2 + 2a_{10}b_5c_4c_5)f_{yy}f_y \\ & + \frac{1}{6}h^3k_1^3(a_1^3b_2 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3)f_{yyy}\end{aligned}$$

Substitutinginto equation (2.3) we have

$$\begin{aligned}y_{n+1} - y_n = {} & hk_1(b_1 + b_2 + b_3 + b_4 + b_5) + h^2k_1(a_1b_2 + b_3c_3 + b_4c_4 + b_5c_5)f_y \\ & + h^3k_1(a_1a_3b_3 + a_1a_5b_4 + a_1a_8b_5 + a_6b_4c_3 + a_9b_5c_3 + a_{10}b_5c_4)f_y^2 \\ & + \frac{1}{2}h^3k_1^2(a_1^2b_2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2)f_{yy} \\ & + h^4k_1(a_1a_3a_6b_4 + a_1a_3a_9b_5 + a_1a_5a_{10}b_5 + a_6a_{10}b_5c_3)f_y^3 \\ & + \frac{1}{2}h^4k_1^2(a_1^2a_3b_3 + a_1^2a_5b_4 + a_1^2a_8b_5 + 2a_1a_3b_3c_3 + 2a_1a_5b_4c_4 + 2a_1a_8b_5c_5 + a_6b_4c_3^2 \\ & + 2a_6b_4c_3c_4 + a_9b_5c_3^2 + 2a_9b_5c_3c_5 + a_{10}b_5c_4^2 + 2a_{10}b_5c_4c_5)f_{yy}f_y \\ & + \frac{1}{6}h^4k_1^3(a_1^3b_2 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3)f_{yyy}\end{aligned}$$

Comparing with equation (2.13) by matching the coefficients of, h^i for (i = 1, 2, 3, 4) we have the following equations:

$$b_1 + b_2 + b_3 + b_4 + b_5 = 1 \qquad \dots(2.15a)$$

$$a_1b_2 + b_3c_3 + b_4c_4 + b_5c_5 = \frac{1}{2} \qquad \dots(2.15b)$$

$$a_1a_3b_3 + a_1a_5b_4 + a_1a_8b_5 + a_6b_4c_3 + a_9b_5c_3 + a_{10}b_5c_4 = \frac{1}{6} \quad \ldots (2.15c)$$

$$a_1^2b_2 + b_3c_3^2 + b_4c_4^2 + b_5c_5^2 = \frac{1}{3} \quad \ldots (2.15d)$$

$$a_1a_3a_6b_4 + a_1a_3a_9b_5 + a_1a_5a_{10}b_5 + a_6a_{10}b_5c_3 = \frac{1}{24} \quad \ldots (2.15e)$$

$$a_1^2a_3b_3 + a_1^2a_5b_4 + a_1^2a_8b_5 + 2a_1a_3b_3c_3 + 2a_1a_5b_4c_4 + 2a_1a_8b_5c_5 + a_6b_4c_3^2 + 2a_6b_4c_3c_4 + a_9b_5c_3^2 + 2a_9b_5c_3c_5 + a_{10}b_5c_4^2 + 2a_{10}b_5c_4c_5 = \frac{1}{3} \ldots (2.15f)$$

$$a_1^3b_2 + b_3c_3^3 + b_4c_4^3 + b_5c_5^3 = \frac{1}{4} \ldots (2.15g)$$

Collecting four out of the 7 equations (i.e. 2.15a, 2.15b,2.15d and 2. 15g), for easy computation and convenience, we set:

$$\left.\begin{aligned} b_4 &= 0 \\ c_1 &= 0 \\ c_2 &= a_1 = \frac{1}{4} \\ c_3 &= (a_2 + a_3) = \frac{1}{2} \ldots (2.16) \\ c_4 &= (a_4 + a_5 + a_6) = \frac{3}{4} \\ c_5 &= (a_7 + a_8 + a_9 + a_{10}) = 1 \end{aligned}\right\}$$

Solving the equations we have;

$$b_1 = \frac{1}{6}, \quad b_2 = 0, \quad b_3 = \frac{2}{3}, \quad b_4 = 0, \quad b_5 = \frac{1}{6}$$

By substituting $b_1, b_2, b_3, b_4, b_5, c_3, c_4, c_5$ and a_1 into the remaining equations (i.e. 2.15c, 2.15e and 2.15f), we observed that they are three equations with six unknown, this implies that we have three“free” parameters to assign value in other to solve the equations. Hence, setting

$$a_3 = \frac{1}{4}, \quad a_5 = \frac{1}{2}, \quad a_9 = \frac{3}{4}$$

Solving the equations with maple 18 we have:

$$a_6 = -\frac{15}{8}, \quad a_8 = \frac{9}{4}, \quad a_{10} = -\frac{1}{4}$$

Now to obtain our $a_2, a_4, and a_7$, we substitute into equation (2.16) we have:

$$a_2 = \frac{1}{4}, \quad a_4 = \frac{17}{8}, \quad a_7 = -\frac{7}{4}$$

Substituting all these values into equation (2.3) we have:

$$y_{n+1} - y_n = \frac{h}{6}(k_1 + 4k_3 + k_5) \ldots (2.17)$$

Where

$$\left.\begin{aligned}
k_1 &= f(y_n) \\
k_2 &= f(y_n + \frac{1}{4} h k_1) \\
k_3 &= f\left(y_n + h\left(\frac{1}{4}(k_1 + k_2)\right)\right) \\
k_4 &= f\left(y_n + h\left(\frac{1}{8}(17k_1 + 4k_2 - 15k_3)\right)\right) \\
k_5 &= f\left(y_n + h\left(\frac{1}{4}(-7k_1 + 9k_2 + 3k_3 - k_4)\right)\right)
\end{aligned}\right\} \ldots (2.18)$$

The Butcher's table [8] will be:

0	0				
$\frac{1}{4}$	$\frac{1}{4}$				
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$			
$\frac{3}{4}$	$\frac{17}{8}$	$\frac{1}{2}$	$-\frac{15}{8}$		
1	$-\frac{7}{4}$	$\frac{9}{4}$	$\frac{3}{4}$	$-\frac{1}{4}$	
	$\frac{1}{6}$	0	$\frac{2}{3}$	0	$\frac{1}{6}$

III. Consistency and Convergency of BO4 Method

From equation (2.17) with $k_{i's}$ in equation (2.8) to (2.11).The equation (2.17) and equation (2.8) to (2.11)is consistence and convergence to a known function if

$$y' = f(x, y), y(x_0) = y_0,$$

i.e$\Phi(x, y, 0) = f(x, y)$.This implies that we are substituting all $k_{i's}$ into Equation (2.17) where$a_{i's}$are given in our derivation.

$$T_n(h^5) = y_{n+1} - y_n$$

$$= \frac{h}{6}\Bigg[f(y_n) + 4\left(f\left(y_n + h\left(a_2\left(f(y_n) + a_3\left(f(y_n + a_1 h f(y_n))\right)\right)\right)\right)\right)$$

$$+ \left(f\left(y_n + h\left(a_7 f(y_n) + a_8(f(y_n + a_1 h f(y_n)))\right)\right)\right)$$

$$+ a_9\left(f\left(y_n + h\left(a_2 f(y_n) + a_3\left(f(y_n + a_1 h f(y_n))\right)\right)\right)\right)$$

$$+ a_{10}\left(f\left(y_n + h\left(a_4 f(y_n) + a_5\left(f(y_n + a_1 h f(y_n))\right)\right)\right)\right)$$

$$+ a_6\left(f\left(y_n + h\left(a_2 f(y_n) + a_3\left(f(y_n + a_1 h f(y_n))\right)\right)\right)\right)\Bigg]$$

Dividing all through by h and taking the limit ofboth sides as $h \rightarrow 0$we have ;

$$\lim_{\hbar\to 0} T_n(\hbar^5) = \lim_{\hbar\to 0}\left(\frac{y_{n+1}-y_n}{\hbar}\right) = \frac{1}{6}[f(y_n)+3f(y_n)+4f(y_n)-3f(y_n)+f(y_n)]$$

$$\lim_{\hbar\to 0} T_n(\hbar^5) = \lim_{\hbar\to 0}\left(\frac{y_{n+1}-y_n}{\hbar}\right) = \frac{1}{6}[6f(y_n)]$$

$$y'(y_n) = f(y_n)$$

Hence, the method is consistent and convergences [10].

IV. Stability Region of BO4 Method

In this section we discuss the stability region for the BO4 method of 4th order. The stability largely depends on the initial value problem (IVP). It should be noted that condition $\left|\frac{y_{n+1}}{y_n}\right| \angle 1$ must be satisfied in order to determine the stability region of the new 4th order Runge-Kutta method formula in the complex plane [11]. With the help of stability polynomials, the stability regions for the BO4 method can be obtained.

To get the area of region, the differential equation $y' = \lambda y$ can be use as a test of the equation, substituting into equation (2.18) we have:

$k_1 = y' = \lambda y_n$

$k_2 = \lambda y_n\left(1+\frac{1}{4}\hbar\lambda\right)$

$k_3 = \lambda y_n\left(1+\frac{1}{2}\hbar\lambda+\frac{1}{16}\hbar^2\lambda^2\right)$

$k_4 = \lambda y_n\left(1+\frac{3}{4}\hbar\lambda-\frac{13}{16}\hbar^2\lambda^2-\frac{15}{128}\hbar^3\lambda^3\right)$

$k_5 = \lambda y_n\left(1+\hbar\lambda+\frac{3}{4}\hbar^2\lambda^2+\frac{1}{4}\hbar^3\lambda^3+\frac{15}{512}\hbar^4\lambda^4\right)$

Substituting all k_1, k_2, k_3, k_4, k_5 into equation (2.17), simplifying the expression and letting $\mu = h\lambda$. Also dividing both side by y_n we have:

$$\frac{y_{n+1}}{y_n} = \left(1+\mu+\frac{1}{2}\mu^2+\frac{1}{6}\mu^3+\frac{1}{24}\mu^4\right) \dots (4.1)$$

Using MATLAB packge , we obtaine the following results for the stability region for the BO4 method as :

$\mu_1 =$ -0.2706 + 2.5048i
$\mu_2 =$ -0.2706 - 2.5048i
$\mu_3 =$ -1.7294 + 0.8890i
$\mu_4 =$ -1.7294 - 0.8890i

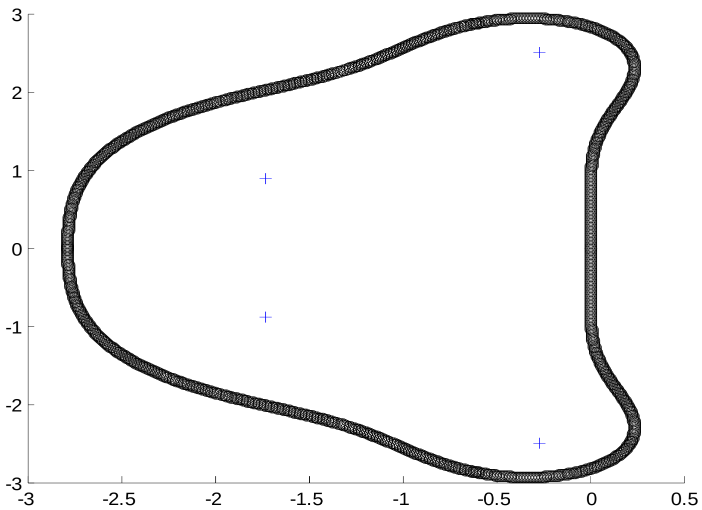

Figure 1: Region of Absolute Stability of BO4 Method

V. Implementation of BO4 Method

We selectedsome tested initial value problems from the corresponding cited publications and solved them by using BO4 method in comparison with classical Runge-Kutta method. The numerical solutions to these Initial Value Problems were generated by a well thought out MATLABpackage

Problem 1. $y' = y$; $y(0) = 1$, $0 \le x \le 1$ With theoretical solution $y(x) = e^x$, h = 0.1[12]
Problem 2.$y' = -y$; $y(0) = 1$, $0 \le x \le 1$ With theoretical solution$y(x) = e^{-x}$, h = 0.1[13]
Problem 3.$y' = y + 1$; $y(0) = 1$, $0 \le x \le 1$ With theoretical solution $y(x) = -1 + 2e^x, h = 0.1$[13]
Problem 4.The Kermack-McKendrick SIR model for the course of an epidemic in a population is given by the system of ODEs
$y_1' = -cy_1y_2$,
$y_2' = cy_1y_2 - dy_2$
$y_3' = dy_2$
Wherey_1 represents susceptible, y_2 represents infectives in circulation, and y_3 represents infectives removed by isolation, death or recovery and immunity. The parameters c and d represent the infection rate and the removal rate, respectively. [14]

Table 1: Numerical Result for Problem 1

		BO4 METHOD		CLASSICAL RUNGE-KUTTA METHOD	
XN	TSOL	YN	ERROR	YN	ERROR
0.1	1.105170918076	1.105170882161	3.5914189400E-08	1.105170833333	8.4742314499E-08
0.2	1.221402758160	1.221402678778	7.9382633800E-08	1.221402570851	1.8730947549E-07
0.3	1.349858807576	1.349858675979	1.3159706524E-07	1.349858497063	3.1051346561E-07
0.4	1.491824697641	1.491824503725	1.9391632944E-07	1.491824240081	4.5756058475E-07
0.5	1.648721270700	1.648721002812	2.6788835550E-07	1.648720638597	6.3210329015E-07
0.6	1.822118800391	1.822118445116	3.5527489772E-07	1.822117962092	8.3829857589E-07
0.7	2.013752707470	2.013752249391	4.5807939175E-07	2.013751626597	1.0808736999E-06
0.8	2.225540928492	2.225540349914	5.7857830127E-07	2.225539563292	1.3652001520E-06
0.9	2.459603111157	2.459602391801	7.1935638957E-07	2.459601413780	1.6973768786E-06
1	2.718281828459	2.718280945113	8.8334638759E-07	2.718279744135	2.0843238793E-06

The result in problem 1 indicates that the new method performed very well when compared with the ClassicalRunge-Kutta method as seen from the error columns. This is expected from the fact that the new method has order five, while that of classical method is order four.

Table 2: Numerical Result for Problem 2

		BO4 METHOD		CLASSICAL RUNGE-KUTTA METHOD	
XN	TSOL	YN	ERROR	YN	ERROR
0.1	0.90483741804	0.904837451172	3.3135915456E-08	0.904837500000	8.1964040444E-08
0.2	0.81873075308	0.818730813043	5.9965233445E-08	0.818730901406	1.4832826811E-07
0.3	0.74081822068	0.740818302070	8.1388182083E-08	0.740818422001	2.0131945988E-07
0.4	0.67032004604	0.670320144226	9.8190765074E-08	0.670320288917	2.4288185130E-07
0.5	0.60653065971	0.606530770771	1.1105834996E-07	0.606530934423	2.7471074648E-07
0.6	0.54881163609	0.548811756682	1.2058770293E-07	0.548811934376	2.9828228854E-07
0.7	0.49658530379	0.496585431089	1.2729764576E-07	0.496585618671	3.1487981944E-07
0.8	0.44932896412	0.449329095756	1.3163848595E-07	0.449329289734	3.2561720653E-07
0.9	0.40656965974	0.406569793741	1.3400035864E-07	0.406569991200	3.3145947648E-07
1	0.36787944117	0.367879575892	1.3472060090E-07	0.367879774412	3.3324105608E-07

Again, the result in problem 2 also indicates that the new method performed well when compared with the Classical Runge-Kuttamethod as seen from the error columns. This is also expected from the fact that the new method is of higher order.

Table 3: Numerical Result for Problem 3

		BO4 METHOD		CLASSICAL RUNGE-KUTTA METHOD	
XN	TSOL	YN	ERROR	YN	ERROR
0.1	1.210341836151	1.210341764323	7.1828378800E-08	1.210341666667	1.6948462878E-07
0.2	1.442805516320	1.442805357555	1.5876526760E-07	1.442805141701	3.7461895075E-07
0.3	1.699717615152	1.699717351958	2.6319413071E-07	1.699716994125	6.2102693077E-07
0.4	1.983649395283	1.983649007450	3.8783265888E-07	1.983648480161	9.1512116907E-07
0.5	2.297442541400	2.297442005624	5.3577671100E-07	2.297441277194	1.2642065799E-06
0.6	2.644237600781	2.644236890231	7.1054979545E-07	2.644235924184	1.6765971513E-06
0.7	3.027505414941	3.027504498782	9.1615878350E-07	3.027503253194	2.1617473993E-06
0.8	3.451081856985	3.451080699828	1.1571566021E-06	3.451079126585	2.7304003036E-06
0.9	3.919206222314	3.919204783601	1.4387127791E-06	3.919202827560	3.3947537563E-06
1	4.436563656918	4.436561890225	1.7666927752E-06	4.436559488270	4.1686477577E-06

Finally, the above resultfrom problem 3 shows that the new method is consistent and converges faster and gives a favorable result when compared with that of Classical Runge-Kutta method. Hence the rate and time of convergence is very encouraging.

VI. Result of Problem

The BO4 Method was tested further on the Kermack-McKendrick SIR model which computes the number of Susceptible, Infected, and Recovered people in a population at any time.The code was built in Python.William Kermack and Anderson McKendrick searched for a mathematical answer as to when the epidemic would terminate and observed that, in general whenever the population of susceptible individuals falls below a threshold value, which depends on several parameters, the epidemic terminates [15].Weused the parameter values $c = 1$ and $d = 5$, and initial values $y_1(0) = 95, y_2(0) = 5$ and $y_3(0) = 0$ and integrated from $t = 0$ to $t = 1$. Thus, we plot each solution component on the same graph as a function of time t. As expected with an epidemic, we see the *susceptible* people are contracting the disease faster than the *infected* people are recovering. This we observed in the figure below through the steepness of y_1 (susceptibles) against y_3 (removed/recovered). The number of infected individuals (y_2) initially increased to a threshold and then begins to decrease down to zero as a result the progressive lack of people to infect either due to death or recovery of the infected people.

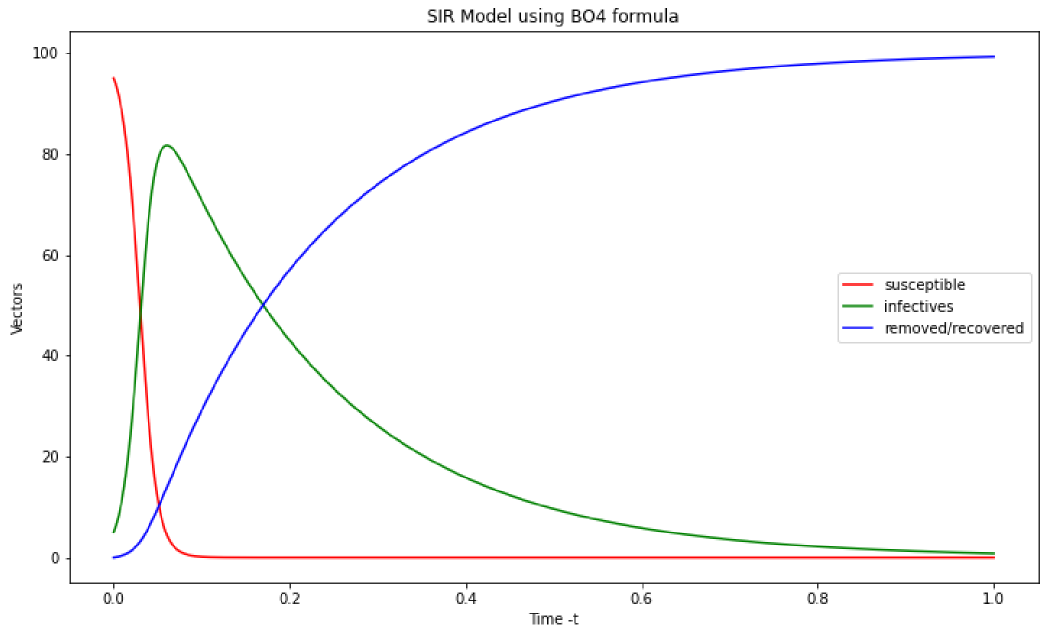

Figure 2: SIR Model using the BO4 Method

VII. Error Analysis of Classical Runge-Kutta Methodand BO4 Method

Below are the figures that show the error analysis of Classical Runge-Kutta method and BO4 method for Problem 1, Problem 2 and Problem 3. This was done with the aid of MATLAB

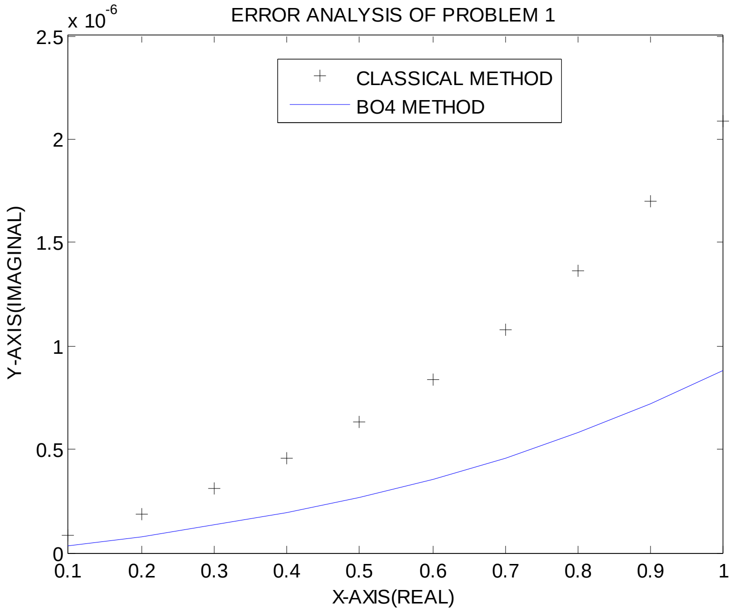

Figure3. Error analysis of Problem 1

From Figure 3 above, theBO4method performs better in terms of accuracy than classical Runge-Kutta method for Problem 1

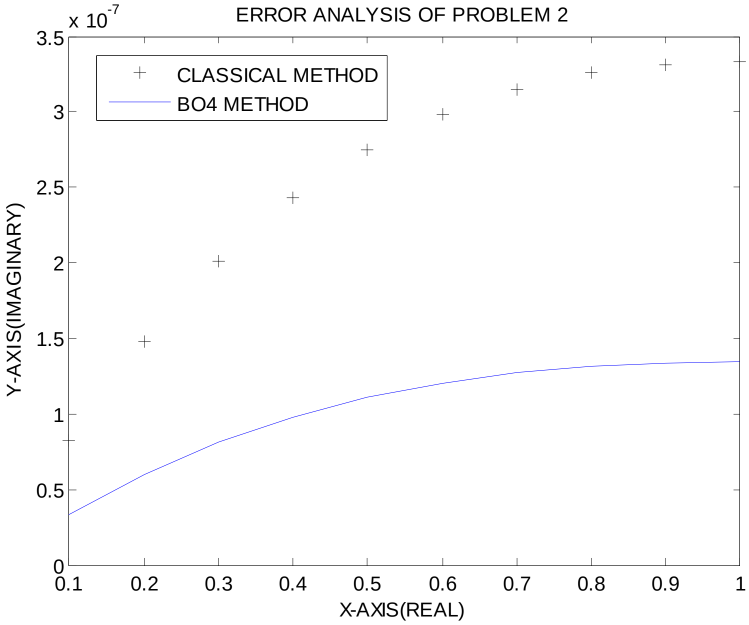

Figure4. Error analysis of Problem 2

The BO4 method clearly shows superiority over classical Runge-Kuttamethod from the above graph, it can be shown that the BO4 method competes favorably with other existing methods in terms of accuracy for Problem 2.

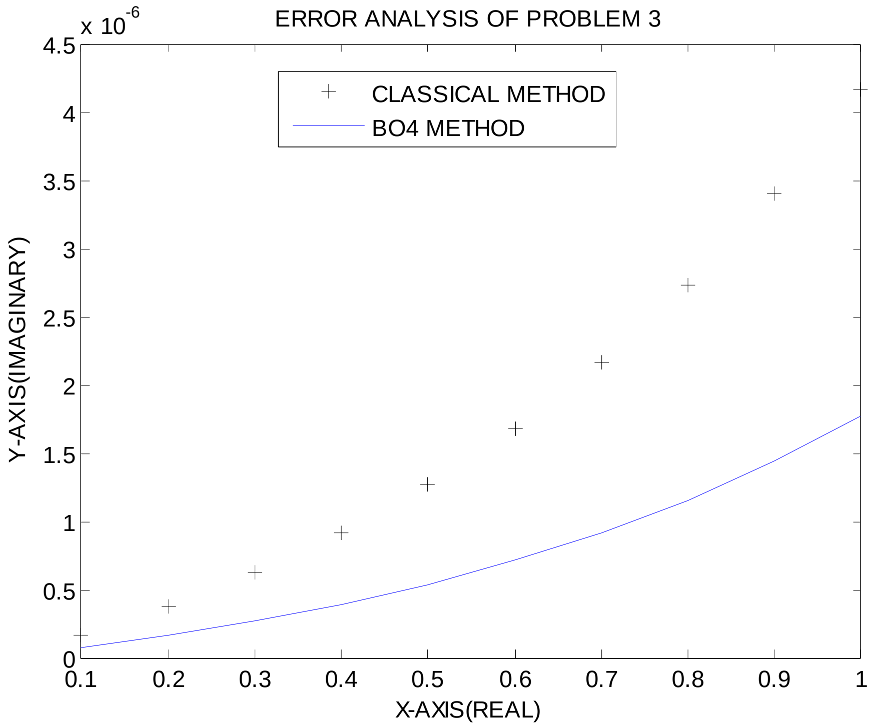

Figure 5. Error Analysis of Problem 3.

Finally, from figure 5 above, it can be shown that the BO4 method competes favorably with other existing methods in terms of accuracy for Problem 3

VIII. Conclusion

In this paper, we derived and implemented the Fifth-Stage Fourth-Order Runge-Kutta formula. Numerical results illustrate that the BO4 method (new method) is more efficient in solving ordinary differential equations via minimal errors that occurred. We also found out that the computer time required was smaller when using BO4 method than classical Runge-Kutta method. This BO4 method maintain a high degree of accuracy in handling first order initial value problems and maybe extended to second order initial value problems with the hope of getting good results. Furthermore, the numerical result shows that the BO4 method is appropriate for the solution of non-stiff initial value problems in ordinary differential equation. Finally, we observed that the BO4 formula can be extended to solving the SIR model for the course of an epidemic in any population with the hope of getting a reliable mathematical solution.

References

[1]. Agbeboh And Ehielmua (2012): The Modified Fourth Order Kutta's Algorithm for Solving Initial Value Problems in Ordinary Differential Equations. A synopsis of master's theses of School of Post Graduate Studies, Ambrose Alli University Ekpoma, Edo State Nigeria.

[2]. Agbebohand Ehielmua (2019): On the derivation of a new fifth order implicit Runge-Kuttascheme for stiff problems in Ordinary Differential Equation. Journal of the Nigeria mathematics Society. Vol.38, 1ssue.247-258,2019.

[3]. AgbebohG.U,Ukpebor, L.A. And Esehkaigbe(2009): "A modified sixth stage fourth-orderRunge-Kutta method for solving initial value problems in ordinary differential equations" Journal of mathematical science Vil.20,pp,97-100.2009

[4]. Agbeboh G.U AndEhielmua.M. (2014): Modified Kutta's algorithm" JNAMP, Vol.25,pp.105-114, 2014

[5]. Agbeboh G.U. (2013): "On the stability analysis of a geometric 4th order Runge-Kutta formula" mathematical theory and modeling Vol. 3, pp. 90-105, 2013.

[6]. Al-Shimmary, Abbas. (2017). Solving initial value problem using Runge-Kutta 6th order method. Journal of Engineering and Applied Sciences. 12. 3953-3961.

[7]. Bazuaye Frank Etin-Osa(2019). A New 4th Order Hybrid Runge-Kutta Method for Solving Initial Value Problems (IVP), Pure and Applied Mathematics Journal. Vol.7, No.6,2018

[8]. Butcher J.C. (2000): The numerical Analysis of Ordinary Differential Equations. Runge-Kutta Methods. New York John Wiley & Sons Ltd.

[9]. H., Musa &Saidu, Ibrahim & Waziri, Marianus (2010). A Simplified Derivation and Analysis of Fourth Order Runge Kutta Method. International Journal of Computer Applications. 9. 10.5120/1402-1891.

[10]. Lambert.J.D.(1973): Computational Methods in Ordinary Differential Equations, John Wiley &Sons London New York Sydney Toronto

[11]. Bazuaye Frank Etin-Osa (2019). A New 4th Order Hybrid Runge-Kutta Method for Solving Initial Value Problems (IVP), Pure and Applied Mathematics Journal. Vol.7, No.6,2018.

[12]. Lambert J.D (2000): Numerical Methods for Ordinary Differential Systems, The Initial Value Problem. New York. John Wiley and Sons Ltd

[13]. G.U. Agbeboh, U.S.U. Aashikpelokhai and I. Aigbedion (2007): Implementation of a new 4^{th} order Runge-Kutta formula for solving Initial Value Problems (I.V.Ps). International Journal of Physical Sciences, 2(4) pp. 089-098

[14]. Jacquez, J. A., & O'Neill, P. (1991). Reproduction numbers and thresholds in stochastic epidemic models. I. Homogeneous populations. *Mathematical biosciences*, *107*(2), 161–186. https://doi.org/10.1016/0025-5564(91)90003-2

[15]. Beckley, R., Weatherspoon, C., Alexander, M., Chandler, M., Johnson, A.J., & Bhatt, G.S. (2013). Modeling epidemics with differential equations.

Ononogbo Chibuike Benjamin, et. al. "On The Derivation and Implementation of a Fifth-Stage Fourth-Order Runge–Kutta Formula for Solving Initial Value Problems in Ordinary Differential Equations." *IOSR Journal of Mathematics (IOSR-JM)*, 16(4), (2020): pp. 29-39.