



Publisher homepage: [www.universepg.com](http://www.universepg.com), ISSN: 2707-4625 (Online) & 2707-4617 (Print)

<https://doi.org/10.34104/ijmms.021.08021>

## International Journal of Material and Mathematical Sciences

Journal homepage: [www.universepg.com/journal/ijmms](http://www.universepg.com/journal/ijmms)

International Journal of  
Material and  
Mathematical Sciences



# A Comparative Study on Classical Fourth Order and Butcher Sixth Order Runge-Kutta Methods with Initial and Boundary Value Problems

Mst. Sharmin Banu<sup>1\*</sup>, Irfan Raju<sup>2</sup>, and Samiran Mondal<sup>1</sup>

<sup>1</sup>Department of Mathematics, Jashore University of Science and Technology, Jashore, Bangladesh; and <sup>2</sup>Department of Arts and Sciences, Bangladesh Army University of Science and Technology, Saidpur, Bangladesh.

\*Correspondence: [msbsharmin\\_math@just.edu.bd](mailto:msbsharmin_math@just.edu.bd) (Mst. Sharmin Banu, Lecturer, Department of Mathematics, Jashore University of Science and Technology, Jashore, Bangladesh).

## ABSTRACT

In this paper, it is discussed about Runge-Kutta fourth order method and Butcher Sixth order Runge-Kutta method for approximating a numerical solution of higher order initial value and boundary value ordinary differential equations. The proposed methods are most efficient and extolled practically for solving these problems arising indifferent sector of science and engineering. Also, the shooting method is applied to convert the boundary value problems to initial value problems. Illustrative examples are provided to verify the accuracy of the numerical outcome and compared the approximated result with the exact result. The approximated results are found in good agreement with the result of exact solution and firstly converge to more accuracy in the solution when step size is very small. Finally, the error with different step sizes is analyzed and compared to these two methods.

**Keywords:** Runge-Kutta fourth order method, Shooting method, Initial value, and boundary value problem.

## 1. INTRODUCTION:

Differential equations (DEs) are of great use in modeling different real life problems arising in science and engineering (Arora, 2019). Model equations formed by using DEs get complicated and several times it becomes quite difficult to find its exact solution (Ahmed and Iqbal, 2020). However, to find the exact solution of a complicated model equation a practice is to simplify the model equation and then find the exact solution of the simplified equation, after then they obtained result is used to approximate the original equation (Islam, 2015).

In this circumstance, the approximate result differs from the real one. To avoid such inconvenience researchers, find their faith in numerical techniques to

find out the approximate solution of a complicated model equation. In the recent era as a result of the technological revolution, numerical techniques have become the most desirable technique in searching the solution to such a problem that cannot be solved analytically. There are several numerical methods to solve DEs. However, the Euler method is one of the basic methods for solving initial value problems of ordinary differential equations (ODEs). Problems arising in DEs are of two types based on the condition given at the endpoints namely initial value problems (IVPs) and boundary value problems (BVPs), and it was introduced in 1768 by British mathematician Leonhard Euler (Hossain *et al.*, 2017). After then several numerical methods developed for solving DEs namely Higher-order Taylor methods, Runge-Kutta

methods, Multistep methods, Extrapolation methods, shooting method, Finite difference methods, Finite element methods, Finite volume methods, and so on, which may find any standard textbook (Burden & Faires, 1985; Balagurusamy, 2006; Mathews, 1992; Gerald, 2004; Equb, 2006; Lambert, 1973; Sastry, 2012; Iqbal *et al.*, 2020).

However, the researcher paid a little more attention to Runge-Kutta (RK) method to solve ODEs due to its simplicity to apply and able to provide accuracy. The method was introduced around 1900 by German mathematicians Carl Runge and Martin Wilhelm Kutta. There are several types of Runge-Kutta method (Gowri *et al.*, 2017; Ababneh *et al.*, 2009; Ademiluyi & Babatola, 2001; Akanbi, 2011; Agam & Yahaya, 2014; Chauhan & Srivastava, 2018; Dormand & Prince, 1980; Paul *et al.*, 2014; Ralston, 1962; Senthikumar, 2009; Paul *et al.*, 2016; Senthikumar & Paul, 2012; Yaakub & Evans, 1999; Paul & Senthil Kumar, 2015; Fehlberg, 1964; Fehlberg, 1966). Among them, RK fourth-order method is more popular. Nowadays it is widely used for solving DEs (Arora, 2019).

In this article, we have used RK fourth order and RK sixth order method to solve initial and boundary value DEs. Also, we have to use the shooting method to convert the boundary value DE to the initial value DE to employ the RK method. The rest of the paper is organized as follows: Section 2 deals with a brief discussion of the RK method, in section 3 our selected problem is introduced, application of RK fourth order and sixth order method to the selected problem is presented in section 4 and 5, respectively, section 6 consists of result and discussion and conclusion is placed in section 7. Finally, the paper ends with a list of references.

## 2. RUNGE-KUTTA METHOD

Runge-Kutta method is a universal method that is widely used for solving ODE. The Runge-Kutta techniques are designed in such a way to achieve greater accuracy in approximation where functional values are required to know at some certain point of the subinterval. A brief discussion of the RK 4<sup>th</sup> order and 6<sup>th</sup> order method is presented in the adjacent subsections.

### A. Runge-Kutta 4<sup>th</sup> order method

Runge-Kutta 4<sup>th</sup> order (RK4) method is the most powerful method to solve ODE. The classical RK method of 4<sup>th</sup> order requires four evaluations per step and it gives more accurate results. A widely used technique of classical RK fourth order method was used in this study whose derivation is also presented here.

The Runge-Kutta method finds approximate value  $y$  for a given  $x$ . Only first-order ODEs can be solved by using the Runge-Kutta 4<sup>th</sup> order method. The formula used to compute the next value  $y_{n+1}$  from the previous value  $y_n$  given below. The value of  $n$  are  $0, 1, 2, 3, \dots, s(x - x_0)/h$ . Here,  $h$  is step height and  $x_{n+1} = x_n + h$

$$y(x+h) - y(x) \cong ak_1 + bk_2 + ck_3 + dk_4$$

$$k_1 = hf(x, y)$$

$$k_2 = hf(x + mh, y + mk_1)$$

$$k_3 = hf(x + mh, y + mk_2)$$

$$k_4 = hf(x + ph, y + pk_1)$$

- $k_1$  is the increment based on the slope at the beginning of the interval, using  $y$
- $k_2$  is the increment based on the slope at the midpoint of the interval, using  $y + \frac{hk_1}{2}$
- $k_3$  is again the increment based on the slope at the midpoint, using  $y + \frac{hk_2}{2}$
- $k_4$  is the increment based on the slope at the end of the interval, using  $y + hk_3$

To find the coefficients  $a, b, c, d, m, n, p$  from Runge-Kutta formulas, re-product the Taylor series in term of  $h$ . The last formula is not a polynomial approximation.

$$F_1 = f_x + f f_y$$

$$F_2 = f_{xx} + 2f f_{xy} + f^2 f_{yy}$$

$$F_3 = f_{xxx} + f f_{xxy} + 3f^2 f_{xyy} + f^3 f_{yyy}$$

Then differentiating the equation  $y' = f(x, y)$ , we get

$$y^2 = f_x + f_y y' = f_x + f_y f = F_1$$

$$y^3 = f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_y (f_x + f f_y)$$

$$= F_2 + f_y F_1$$

$$y^4 = f_{xxx} + 3f f_{xxy} + 3f^2 f_{xyy} + f^3 f_{yyy}$$

$$+ f_y (f_{xx} + 2f f_{xy} + f^2 f_{yy})$$

$$+ 3(f_x + f f_y)(f_{xx} + f f_{yy})$$

$$+ f_y^2 (f_x + f f_y)$$

$$= F_3 + f_y F_2 + 3F_1 (f_{xy} + f f_{yy} + f_y^2 F_1)$$

Which follows the Taylors series is to be written in the form

$$\begin{aligned} y(x+h) - y(x) &= hf + \frac{1}{2}h^2 F_1 + \frac{1}{6}h^3 (F_2 + f_y F_1) \\ &+ \frac{1}{24}h^4 [F_3 + f_y F_2 \\ &+ 3(f_{xx} + f f_{yy})F_1 + f_x^2 F_1] \end{aligned}$$

Turning now to the various k values similar computations produce

$$\begin{aligned} k_1 &= hf \\ k_2 &= h \left[ f + mhF_1 + \frac{1}{2}m^2 h^2 F_2 + \frac{1}{6}m^3 h^3 F_3 \right. \\ &\quad \left. + \dots \dots \dots \right] \\ k_3 &= \left[ \left\{ f + nhF_1 + \frac{1}{2}h^2 (n^2 F_2 + 2mnf_y F_1) \right. \right. \\ &\quad \left. \left. + \frac{1}{6}h^3 (n^2 F_3 + 3m^2 n f_y F_2 \right. \right. \\ &\quad \left. \left. + 6mn^2 (f_{xy} + f f_{yy}) F_1 \right\} \right. \\ &\quad \left. + 6mnp f_y^2 F_1 + \dots \dots \dots \right] \\ k_4 &= \left[ \left\{ f + phF_1 + \frac{1}{2}h^2 (p^2 F_2 + 2pnf_y F_1) \right. \right. \\ &\quad \left. \left. + \frac{1}{6}h^3 (p^3 F_3 + 3n^2 p f_y F_2 + 6np^2 (f_{xy} \right. \right. \\ &\quad \left. \left. + f f_{yy}) F_1 \right\} + 6mnp f_y^2 F_1 \right. \\ &\quad \left. + \dots \dots \dots \right] \end{aligned}$$

Combining these according to the final Runge-Kutta formula,

$$\begin{aligned} y(x+h) - y(x) &= (a + b + c + d)hf \\ &+ (bm + cn + dp)h^2 F_1 \\ &+ \frac{1}{2}(bm^2 + cn^2 + dp^2)h^3 F_2 \\ &+ \frac{1}{6}(bm^3 + cn^3 + dp^3)h^4 F_3 \\ &+ (cmn + dnp)h^3 f_y F_1 \\ &+ \frac{1}{2}(cm^2 n + dn^2 p)h^4 f_y F_2 \\ &+ (cmn^2 + dnp^2)h^4 (f_{xy} + f f_{yy}) F_1 \\ &+ dmnph^4 f_y^2 F_1 + \dots \dots \dots \end{aligned}$$

Comparison with the Taylor series suggests the eight conditions

$$\begin{aligned} a + b + c + d &= 1 \\ bm + cn + dp &= \frac{1}{2} \end{aligned}$$

$$\begin{aligned} bm^2 + cn^2 + dp^2 &= \frac{1}{3} \\ bm^3 + cn^3 + dp^3 &= \frac{1}{4} \\ cmn + dnp &= \frac{1}{6} \\ cmn^2 + dnp^2 &= \frac{1}{8} \\ cm^2 n + dn^2 p &= \frac{1}{12} \\ dmn p &= \frac{1}{3} \end{aligned}$$

The eight equations in seven unknowns are actually excessive. The classical solution set is

$$\begin{aligned} m = n &= \frac{1}{2} \\ p &= 1 \\ a = d &= \frac{1}{6} \\ b = c &= \frac{1}{3} \end{aligned}$$

Leading to the Runge-Kutta formulas

$$\begin{aligned} k_1 &= hf(x, y) \\ k_2 &= hf \left( x + \frac{1}{2}h, y + \frac{1}{2}k_1 \right) \\ k_3 &= hf \left( x + \frac{1}{2}h, y + \frac{1}{2}k_2 \right) \\ k_4 &= hf(x + h, y + k_3) \end{aligned}$$

From equation (1) we get

$$\begin{aligned} y(x+h) &= y(x) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ y_{r+1}(x+h) &= y_r(x) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

Where  $r = 1, 2, 3, \dots, n$ , which is the general 4<sup>th</sup> order RK method.

**B. Runge-Kutta 6<sup>th</sup> order method**

Runge-Kutta Sixth order (RK6) method is better for solving an ODE as well as a partial differential equation (PDE) with an IVP. Though in general, higher-order Runge-Kutta gives a better solution than the lower order ones (Hossain et al., 2017). But it is not true for all purposes and differs from case to case. However, it is a well-known result that the Runge-Kutta method of order  $\rho$  needs at least  $\rho$  stages to the process. It is true for  $\rho = 1, 2, 3, 4$  but suddenly for order  $\rho = 5$  there exist a process with stages 6 due to (Kutta, 1901) but corrected by (Nyström, 1925). Again, in the study of (Huia, 1956) there are 8 stages processed for  $\rho = 6$ . To mitigate, these varieties and

complexity Butcher (Butcher, 1963) provide a common form to all Runge-Kutta method by a matrix and two vectors and it is known as Butcher tableau. This method approximates the solution to an ODE of the form  $y' = f(t, y)$  with  $s$  stages is

$$y_{n+1} = y_n + h \sum_{i=1}^s b_i k_{ni}$$

With  $k_i = hf(x_n + c_i h, y_n + \sum_{j=1}^{i-1} a_{ij} k_j)$ , Where,  $n$  is the node's running index ( $n = 0, 1, 2, \dots$ ),  $i$  is the index used to label stages ( $1 \leq i \leq s$ ), the nodes  $c_i = \sum_{j=1}^{i-1} a_{ij}$  and for the consistency the weights  $b_i$  are defined as  $\sum_{i=1}^s b_i = 1$ . And the Butcher tableau is

0						
$c_1$	$a_{21}$					
$c_2$	$a_{31}$	$a_{32}$				
$c_3$	$a_{41}$	$a_{42}$	$a_{43}$			
$\vdots$						
$c_s$	$a_{s1}$	$a_{s2}$	$a_{s3}$	$\dots$	$a_{s,(s-1)}$	
	$b_1$	$b_2$	$b_3$	$\dots$	$b_{s-1}$	$b_s$

Derivation of RK6 method from Butcher tableau can find in any standard numerical analysis book. Due to the paucity of space, we are not presenting the derivation here. For details of derivation please see (Butcher, 1963). And the Butcher tableau for RK6 method is -

0						
$\frac{1}{3}$	$\frac{1}{3}$					
$\frac{2}{3}$	0	$\frac{2}{3}$				
$\frac{1}{3}$	$\frac{1}{12}$	$\frac{1}{3}$	$-\frac{1}{12}$			
$\frac{1}{2}$	$-\frac{1}{16}$	$\frac{9}{8}$	$-\frac{3}{16}$	$-\frac{3}{8}$		
$\frac{1}{2}$	0	$\frac{9}{8}$	$-\frac{3}{8}$	$-\frac{3}{4}$	$\frac{1}{2}$	
1	$\frac{9}{44}$	$-\frac{9}{11}$	$\frac{63}{44}$	$\frac{18}{11}$	0	$-\frac{16}{11}$
	$\frac{11}{120}$	0	$\frac{27}{40}$	$\frac{27}{40}$	$-\frac{4}{15}$	$-\frac{4}{15}$
						$\frac{11}{120}$

Leading to the RK6 formulas

$$y_{n+1} = y_n + \frac{h}{120} (11k_1 + 81k_3 + 81k_4 - 32k_5 - 32k_6 + 11k_7),$$

Where,

$$\begin{aligned} k_1 &= f(x_n, y_n, z_n) \\ k_2 &= f\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{3}, z_n + \frac{l_1}{3}\right) \\ k_3 &= f\left(x_n + \frac{2h}{3}, y_n + \frac{2k_2}{3}\right) \\ k_4 &= f\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{12} + \frac{k_2}{3} - \frac{k_3}{12}\right) \\ k_5 &= f\left(x_n + \frac{h}{2}, y_n - \frac{k_1}{16} + \frac{9k_2}{8} - \frac{3k_3}{16} - \frac{3k_4}{8}\right) \\ k_6 &= f\left(x_n + \frac{h}{2}, y_n + \frac{9k_2}{8} - \frac{3k_3}{16} - \frac{3k_4}{4} + \frac{k_5}{2}\right) \\ k_7 &= f\left(x_n + h, y_n + \frac{9k_1}{44} - \frac{9k_2}{11} + \frac{63k_3}{44} + \frac{18k_4}{11} - \frac{16k_6}{11}\right) \end{aligned}$$

### 3. Problem statement

There are two parts of differential equations, IVP and BVP on the basics of the given conditions indicated at the endpoints. In our present work, second order initial and boundary value ODEs are solved by RK4 and RK6 methods. Among many different formulas, one most common form of fourth ordered which derived and Butcher's sixth ordered Runge-Kutta methods are used. The BVP problems are firstly converted to a system of IVP through the medium of shooting method.

#### A. The numerical solution of IVPs

Consider the following second order IVP of ODE:

$$\frac{d^2y}{dx^2} = f(x, y, y'), \tag{1}$$

With the initial condition  $y(x_0) = y_0, y'(x_0) = \alpha$ . Now this second order IVP can be written in a system of first order differential equations by considering

$$\frac{dy}{dx} = y' = z.$$

Then the new problem will be,

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

Where,  $g(x, y, z) = z$

With the initial condition  $y(x_0) = y_0, z(x_0) = z_0 = \alpha$ .

Then the RK4 method will use the following formula:

$$z(x_{n+1}) = z_n + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4),$$

$$y(x_{n+1}) = y_n + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4),$$

Where,

$$k_1 = f(x_n, y_n, z_n)$$

$$l_1 = f(x_n, y_n, z_n)$$

$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}\right)$$

$$l_2 = g\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}, z_n + \frac{l_1}{2}\right)$$

$$k_3 = f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}\right)$$

$$l_3 = g\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}, z_n + \frac{l_2}{2}\right)$$

$$k_4 = f\left(x_n + \frac{h}{2}, y_n + k_3, z_n + l_3\right)$$

$$l_4 = g\left(x_n + \frac{h}{2}, y_n + k_3, z_n + l_3\right)$$

For  $n = 0, 1, 2, \dots, \dots, \dots, \dots$

And the RK6 method will use the following formula

$$z_{n+1} = z_n + \frac{h}{12^6} (11k_1 + 81k_3 + 81k_4 - 32k_5 - 32k_6 + 11k_7),$$

$$y_{n+1} = y_n + \frac{h}{12^6} (11l_1 + 81l_3 + 81l_4 - 32l_5 - 32l_6 + 11l_7),$$

Where,

$$k_1 = f(x_n, y_n, z_n)$$

$$l_1 = f(x_n, y_n, z_n)$$

$$k_2 = f\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{3}, z_n + \frac{l_1}{3}\right)$$

$$l_2 = g\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{3}, z_n + \frac{l_1}{3}\right)$$

$$k_3 = f\left(x_n + \frac{2h}{3}, y_n + \frac{2k_2}{3}, z_n + \frac{2l_2}{3}\right)$$

$$l_3 = g\left(x_n + \frac{2h}{3}, y_n + \frac{2k_2}{3}, z_n + \frac{2l_2}{3}\right)$$

$$k_4 = f\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{12} + \frac{k_2}{3} - \frac{k_3}{12}, z_n + \frac{l_1}{12} + \frac{l_2}{3} - \frac{l_3}{12}\right)$$

$$l_4 = g\left(x_n + \frac{h}{3}, y_n + \frac{k_1}{12} + \frac{k_2}{3} - \frac{k_3}{12}, z_n + \frac{l_1}{12} + \frac{l_2}{3} - \frac{l_3}{12}\right)$$

$$k_5 = f\left(x_n + \frac{h}{2}, y_n - \frac{k_1}{16} + \frac{9k_2}{8} - \frac{3k_3}{16} - \frac{3k_4}{8}, z_n - \frac{l_1}{16} + \frac{9l_2}{8} - \frac{3l_3}{16} - \frac{3l_4}{8}\right)$$

$$l_5 = g\left(x_n + \frac{h}{2}, y_n - \frac{k_1}{16} + \frac{9k_2}{8} - \frac{3k_3}{16} - \frac{3k_4}{8}, z_n - \frac{l_1}{16} + \frac{9l_2}{8} - \frac{3l_3}{16} - \frac{3l_4}{8}\right)$$

$$k_6 = f\left(x_n + \frac{h}{2}, y_n + \frac{9k_2}{8} - \frac{3k_3}{16} - \frac{3k_4}{4} + \frac{k_5}{2}, z_n + \frac{9l_2}{8} - \frac{3l_3}{16} - \frac{3l_4}{4} + \frac{l_5}{2}\right)$$

$$l_6 = g\left(x_n + \frac{h}{2}, y_n + \frac{9k_2}{8} - \frac{3k_3}{16} - \frac{3k_4}{4} + \frac{k_5}{2}, z_n + \frac{9l_2}{8} - \frac{3l_3}{16} - \frac{3l_4}{4} + \frac{l_5}{2}\right)$$

$$k_7 = f\left(x_n + h, y_n + \frac{9k_1}{44} - \frac{9k_2}{11} + \frac{63k_3}{44} + \frac{18k_4}{11} - \frac{16k_6}{11}, z_n + \frac{9l_1}{44} - \frac{9l_2}{11} + \frac{63l_3}{44} + \frac{18l_4}{11} - \frac{16l_6}{11}\right)$$

$$l_7 = g\left(x_n + h, y_n + \frac{9k_1}{44} - \frac{9k_2}{11} + \frac{63k_3}{44} + \frac{18k_4}{11} - \frac{16k_6}{11}, z_n + \frac{9l_1}{44} - \frac{9l_2}{11} + \frac{63l_3}{44} + \frac{18l_4}{11} - \frac{16l_6}{11}\right)$$

For  $n = 1, 2, 3, \dots, \dots, \dots, \dots$

### B. The numerical solution of BVPs

Consider the following second order boundary value ODEs:

$$\frac{d^2y}{dx^2} = f(x, y, y'), \quad a \leq x \leq b \quad (2)$$

With the boundary condition  $y(a) = \alpha, y(b) = \beta$ .

The same formulas of RK4 and RK6 methods that are used in solving IVP are used along with the shooting method to solve this BVP. Mainly, in the Shooting method, BVP has been transferred under consideration by an IVP with unknown parameter  $t$  as of the form,

$$\frac{d^2y}{dx^2} = f(x, y, y'); \quad a \leq x \leq b \quad (3)$$

$$y(a) = \alpha, y'(a) = t.$$

The parameter  $t$  is to be chosen in a manner to ensure that  $\lim_{k \rightarrow \infty} y(b, t_k) = y(b) = \beta$ , where  $y(x, t_k)$  denotes the solution to the initial-value problem with  $t = t_k$ . Thus RK4 and RK6 methods are used for solving the IVPs of Eq. (3) for the repeated value of  $t$  until we carry out the desired level of tolerance and accuracy and the solution  $y(x)$  of the BVP achieved.

#### 4. Application of Runge-Kutta 4<sup>th</sup> order and 6<sup>th</sup> order method to the initial value problem

In this section, we consider the following examples of IVP to test the efficiency and convergence of numerical solution along with actual solution where Runge-Kutta fourth order and sixth order methods were applied separately. The tables are containing the approximated result, actual result, and computed absolute error; the outcomes are displayed in the graph also. For better comparison, three different step sizes are used as 0.1, 0.05, and 0.025, respectively. The computations programming language, associated with the examples, is performed by MATLAB.

**Example 1.** Consider the IVP

$$y'' = \frac{12 \log x}{x^2} - \frac{y'}{x}, 1 \leq x \leq 2,$$

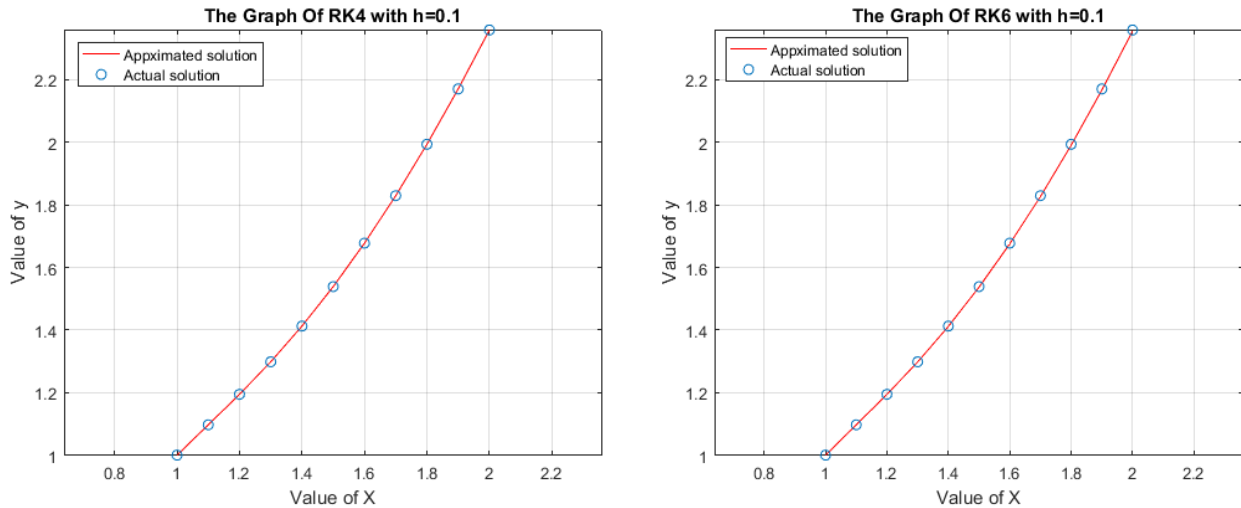
$$y(1) = y'(1) = 1.$$

The actual solution of this equation is  $y = 1 + \log x + 2(\log x)^3$ .

**Table 1 to 3** and **Fig 1 to 3** show the computed result and maximum absolute error relative to the actual solution. Where the line curve represents the approximated curve and the circle is for actual results.

**Table 1:** Computed result and absolute maximum error of IVP with step size 0.1

$x_n$	Approximation of RK4		Approximation of RK6		Actual solution
	Computed $y_n$	Error $e_n$	Computed $y_n$	Error $e_n$	
1.000000	1.000000	0.00000e+00	1.000000	0.00000e+00	1.000000
1.100000	1.0970379	3.86129e-06	1.0971472	1.05414e-04	1.0970418
1.200000	1.1944366	6.09001e-06	1.1942902	1.52473e-04	1.1944427
1.300000	1.2984765	7.46833e-06	1.2978944	5.89531e-04	1.2984840
1.400000	1.4126503	8.38106e-06	1.4115575	1.10119e-03	1.4126587
1.500000	1.5387746	9.02728e-06	1.5371557	1.62791e-03	1.5387836
1.600000	1.6776449	9.51450e-06	1.6755183	2.13619e-03	1.6776544
1.700000	1.8294325	9.90299e-06	1.8268344	2.60793e-03	1.8294424
1.800000	1.9939290	1.02277e-05	1.9909050	3.03416e-03	1.9939392
1.900000	2.1707007	1.05096e-05	2.1672998	3.41138e-03	2.1707112
2.000000	2.3591857	1.07614e-05	2.3554572	3.73927e-03	2.3591965



**Fig 1:** Graphical Representation of Approximated and Exact Solution of IVP with h=0.1.

**Table 2:** Computed result and absolute maximum error of IVP with step size 0.05

$x_n$	Approximation of RK4		Approximation of RK6		Actual solution
	Computed $y_n$	Error $e_n$	Computed $y_n$	Error $e_n$	
1.000000	1.000000	0.00000e+00	1.000000	0.00000e+00	1.000000
1.050000	1.0490223	1.35883e-07	1.0490925	7.00280e-05	1.0490225

1.1000000	1.0970415	2.37595e-07	1.0971166	7.48028e-05	1.0970418
1.1500000	1.1452217	3.15069e-07	1.1452544	3.24255e-05	1.1452220
1.2000000	1.1944423	3.75109e-07	1.1943992	4.35531e-05	1.1944427
1.2500000	1.2453651	4.22442e-07	1.2452226	1.42942e-04	1.2453655
1.3000000	1.2984835	4.60403e-07	1.2982259	2.58054e-04	1.2984840
1.3500000	1.3541606	4.91369e-07	1.3537780	3.83078e-04	1.3541611
1.4000000	1.4126582	5.17059e-07	1.4121451	5.13627e-04	1.4126587
1.4500000	1.4741588	5.38724e-07	1.4735129	6.46388e-04	1.4741593
1.5000000	1.5387831	5.57291e-07	1.5380047	7.78876e-04	1.5387836
1.5500000	1.6066033	5.73446e-07	1.6056946	9.09239e-04	1.6066039
1.6000000	1.6776539	5.87706e-07	1.6766183	1.03611e-03	1.6776544
1.6500000	1.7519394	6.00462e-07	1.7507815	1.15849e-03	1.7519400
1.7000000	1.8294417	6.12013e-07	1.8281667	1.27569e-03	1.8294424
1.7500000	1.9101247	6.22588e-07	1.9087381	1.38723e-03	1.9101254
1.8000000	1.9939386	6.32365e-07	1.9924464	1.49279e-03	1.9939392
1.8500000	2.0808232	6.41484e-07	2.0792316	1.59223e-03	2.0808238
1.9000000	2.1707106	6.50054e-07	2.1690258	1.68545e-03	2.1707112
1.9500000	2.2635273	6.58161e-07	2.2617554	1.77248e-03	2.2635279
2.0000000	2.3591958	6.65873e-07	2.3573431	1.85339e-03	2.3591965

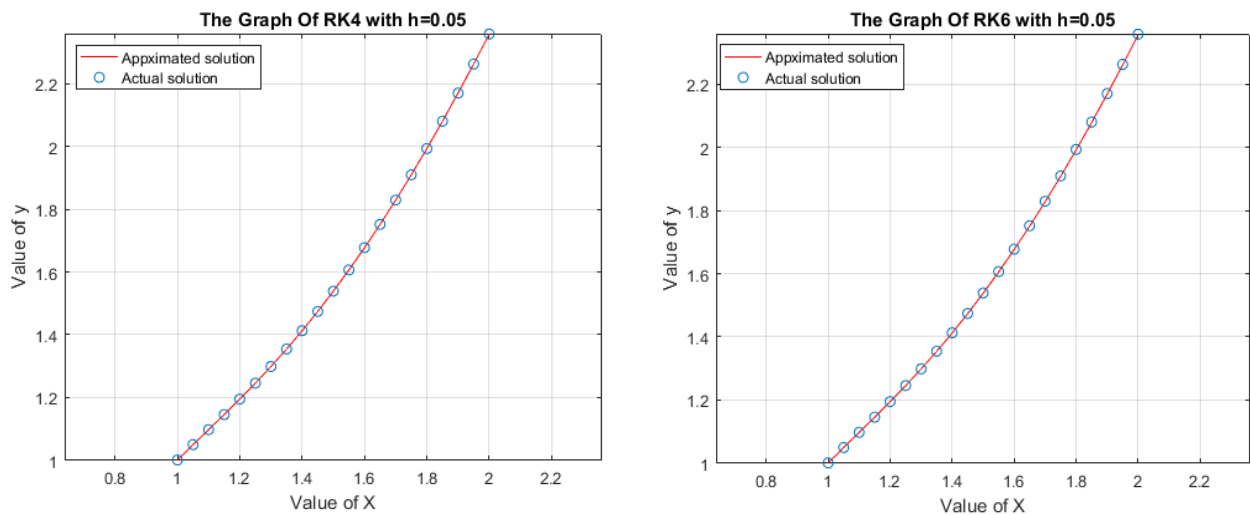


Fig 2: Graphical Representation of Approximated and Exact Solution of IVP with h=0.05.

Table 3: Computed result and absolute maximum error of IVP with step size 0.025.

$x_n$	Approximation of RK4		Approximation of RK6		Actual solution
	Computed $y_n$	Error $e_n$	Computed $y_n$	Error $e_n$	
1.000000	1.000000	0.000000	1.000000	0.000000	1.000000
1.025000	1.0247227	4.51298e-09	1.0247467	2.39873e-05	1.0247227
1.050000	1.0490224	8.40261e-09	1.0490607	3.82013e-05	1.0490225
1.075000	1.0730772	1.17698e-08	1.0731213	4.40768e-05	1.0730772
1.100000	1.0970418	1.46976e-08	1.0970846	4.28514e-05	1.0970418
1.125000	1.1210510	1.72543e-08	1.1210866	3.55947e-05	1.1210510
1.150000	1.1452220	1.94966e-08	1.1452452	2.32321e-05	1.1452220
1.175000	1.1696565	2.14717e-08	1.1696630	6.56510e-06	1.1696565
1.200000	1.1944427	2.32189e-08	1.1944290	1.37108e-05	1.1944427
1.225000	1.2196571	2.47711e-08	1.2196201	3.69921e-05	1.2196571
1.250000	1.2453655	2.61560e-08	1.2453028	6.27545e-05	1.2453655
1.275000	1.2716249	2.73969e-08	1.2715344	9.05423e-05	1.2716249
1.300000	1.2984839	2.85135e-08	1.2983640	1.19959e-04	1.2984840
1.325000	1.3259842	2.95226e-08	1.3258336	1.50661e-04	1.3259842
1.350000	1.3541611	3.04383e-08	1.3539787	1.82348e-04	1.3541611
1.375000	1.3830443	3.12728e-08	1.3828295	2.14760e-04	1.3830443

1.400000	1.4126586	3.20365e-08	1.4124110	2.47671e-04	1.4126587
1.425000	1.4430248	3.27381e-08	1.4427439	2.80884e-04	1.4430248
1.450000	1.4741593	3.33854e-08	1.4738451	3.14230e-04	1.4741593
1.475000	1.5060754	3.39849e-08	1.5057279	3.47561e-04	1.5060755
1.500000	1.5387836	3.45422e-08	1.5384029	3.80750e-04	1.5387836
1.525000	1.5722913	3.50624e-08	1.5718776	4.13687e-04	1.5722913
1.550000	1.6066039	3.55495e-08	1.6061576	4.46278e-04	1.6066039
1.575000	1.6417244	3.60074e-08	1.6412460	4.78443e-04	1.6417245
1.600000	1.6776544	3.64392e-08	1.6771443	5.10113e-04	1.6776544
1.625000	1.7143935	3.68478e-08	1.7138523	5.41229e-04	1.7143935
1.650000	1.7519400	3.72356e-08	1.7513683	5.71743e-04	1.7519400
1.675000	1.7902910	3.76047e-08	1.7896894	6.01613e-04	1.7902910
1.700000	1.8294423	3.79570e-08	1.8288116	6.30805e-04	1.8294424
1.725000	1.8693891	3.82943e-08	1.8687298	6.59292e-04	1.8693891
1.750000	1.9101253	3.86179e-08	1.9094383	6.87052e-04	1.9101254
1.775000	1.9516444	3.89291e-08	1.9509304	7.14067e-04	1.9516445
1.800000	1.9939392	3.92291e-08	1.9931989	7.40325e-04	1.9939392
1.825000	2.0370017	3.95188e-08	2.0362359	7.65815e-04	2.0370018
1.850000	2.0808238	3.97993e-08	2.0800333	7.90532e-04	2.0808238
1.875000	2.1253966	4.00712e-08	2.1245822	8.14473e-04	2.1253966
1.900000	2.1707112	4.03353e-08	2.1698736	8.37636e-04	2.1707112
1.925000	2.2167581	4.05922e-08	2.2158981	8.60024e-04	2.2167582
1.950000	2.2635279	4.08425e-08	2.2626463	8.81639e-04	2.2635279
1.975000	2.3110106	4.10866e-08	2.3101082	9.02486e-04	2.3110107
2.000000	2.3591964	4.13251e-08	2.3582739	9.22573e-04	2.3591965

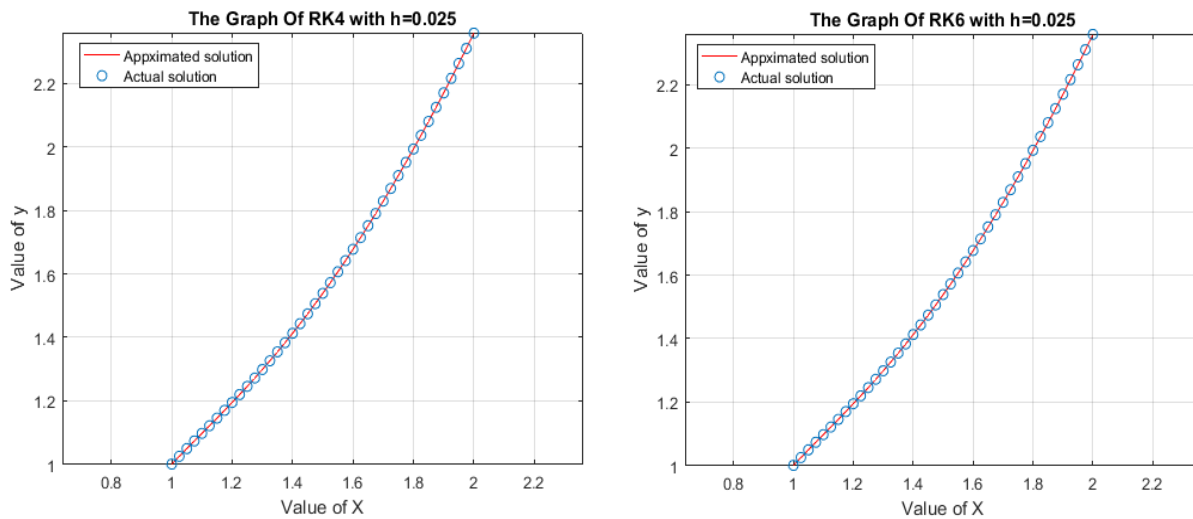


Fig 3: Graphical Representation of Approximated and Exact Solution of IVP with h=0.025.

### 5. Application of Runge-Kutta 4<sup>th</sup> order and 6<sup>th</sup> order method to the boundary value problem

The following BVP was taken for our study and test the efficiency and convergence of Runge-Kutta fourth order and sixth order methods separately and compared with actual solution. The tables are containing the approximated result, actual result, and computed absolute error; also the outcomes are displayed in the graph. For better comparison, three different step sizes are used as 0.1, 0.05, and 0.025, respectively. The computations programming lang-

uage, for the affiliated examples, are performed by MATLAB.

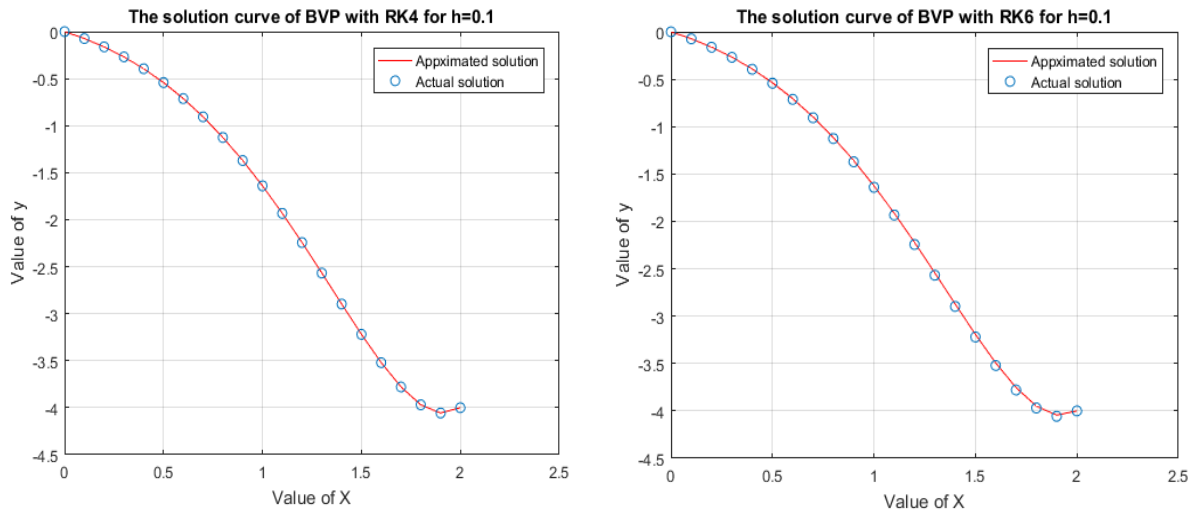
**Example 2:** Consider the BVP  $y'' = 2y' - y + xe^x - x$ ,  $0 \leq x \leq 2$ ,  $y(0) = 0$ ,  $y(2) = -4$  and compare the result to the actual solution  $y(x) = \frac{1}{6}x^3e^x - \frac{5}{3}xe^x + 2e^x - x - 2$ .

Table 4-6, and Fig 4-6 show the computed result and maximum absolute error with actual solution. As earlier the line curved represents the approximated curve and circle for the actual results.



**Table 4:** Computed result and the absolute maximum error of BVP with step size 0.1

$x_n$	Approximation of RK4		Approximation of RK6		Actual solution
	Computed $y_n$	Error $e_n$	Computed $y_n$	Error $e_n$	
0.000000	0.000000	0.00000e+00	0.000000	0.00000e+00	0.000000
0.100000	-0.0736052	6.39298e-05	-0.0730929	5.76240e-04	-0.0736691
0.200000	-0.1625683	1.31897e-04	-0.1613603	1.33989e-03	-0.1627002
0.300000	-0.2689348	2.02621e-04	-0.2668235	2.31393e-03	-0.2691374
0.400000	-0.3947131	2.74492e-04	-0.3914685	3.51910e-03	-0.3949876
0.500000	-0.5417979	3.45559e-04	-0.5371713	4.97218e-03	-0.5421435
0.600000	-0.7118714	4.13522e-04	-0.7056013	6.68358e-03	-0.7122849
0.700000	-0.9062775	4.75759e-04	-0.8980989	8.65435e-03	-0.9067532
0.800000	-1.1258639	5.29362e-04	-1.1155209	1.08723e-02	-1.1263932
0.900000	-1.3707854	5.71225e-04	-1.3580493	1.33074e-02	-1.3713567
1.000000	-1.6402609	5.98163e-04	-1.6249539	1.59052e-02	-1.6408591
1.100000	-1.9322744	6.07107e-04	-1.9143014	1.85801e-02	-1.9328815
1.200000	-2.2432110	5.95369e-04	-2.2226010	2.12053e-02	-2.2438063
1.300000	-2.5674143	5.61010e-04	-2.5443731	2.36022e-02	-2.5679753
1.400000	-2.8966519	5.03338e-04	-2.8716293	2.55259e-02	-2.8971552
1.500000	-3.2194709	4.23563e-04	-3.1932458	2.66486e-02	-3.2198944
1.600000	-3.5204258	3.25658e-04	-3.4942122	2.65392e-02	-3.5207515
1.700000	-3.7791548	2.17462e-04	-3.7547339	2.46384e-02	-3.7793722
1.800000	-3.9692780	1.12098e-04	-3.9491616	2.02286e-02	-3.9693901
1.900000	-4.0570888	2.97665e-05	-4.0447200	1.23985e-02	-4.0571185
2.000000	-4.0000000	0.00000e+00	-4.0000000	0.00000e+00	-4.0000000



**Fig 4:** Graphical Representation of Approximated and Exact Solution of BVP with h=0.1.

**Table 5:** Computed result and absolute maximum error of BVP with step size 0.05.

$x_n$	Approximation of RK4		Approximation of RK6		Actual solution
	Computed $y_n$	Error $e_n$	Computed $y_n$	Error $e_n$	
0.000000	0.000000	0.00000e+00	0.000000	0.00000e+00	0.000000
0.050000	-0.0350340	7.79325e-06	-0.0349139	1.27939e-04	-0.0350418
0.100000	-0.0736533	1.58318e-05	-0.0733917	2.77396e-04	-0.0736691
0.150000	-0.1161125	2.40787e-05	-0.1156867	4.49847e-04	-0.1161365
0.200000	-0.1626677	3.24921e-05	-0.1620535	6.46748e-04	-0.1627002
0.250000	-0.2135749	4.10253e-05	-0.2127464	8.69511e-04	-0.2136159
0.300000	-0.2690878	4.96261e-05	-0.2680179	1.11948e-03	-0.2691374
0.350000	-0.3294556	5.82373e-05	-0.3281160	1.39793e-03	-0.3295139
0.400000	-0.3949208	6.67963e-05	-0.3932816	1.70598e-03	-0.3949876
0.450000	-0.4657158	7.52351e-05	-0.4637464	2.04462e-03	-0.4657910
0.500000	-0.5420600	8.34809e-05	-0.5397288	2.41466e-03	-0.5421435

0.5500000	-0.6241561	9.14555e-05	-0.6214309	2.81664e-03	-0.6242476
0.6000000	-0.7121858	9.90764e-05	-0.7090341	3.25085e-03	-0.7122849
0.6500000	-0.8063054	1.06257e-04	-0.8026945	3.71722e-03	-0.8064117
0.7000000	-0.9066403	1.12905e-04	-0.9025379	4.21530e-03	-0.9067532
0.7500000	-1.0132795	1.18929e-04	-1.0086543	4.74414e-03	-1.0133984
0.8000000	-1.1262690	1.24232e-04	-1.1210910	5.30227e-03	-1.1263932
0.8500000	-1.2456047	1.28718e-04	-1.2398458	5.88756e-03	-1.2457334
0.9000000	-1.3712244	1.32289e-04	-1.3648595	6.49716e-03	-1.3713567
0.9500000	-1.5029990	1.34852e-04	-1.4960065	7.12738e-03	-1.5031338
1.0000000	-1.6407228	1.36316e-04	-1.6330855	7.77355e-03	-1.6408591
1.0500000	-1.7841026	1.36597e-04	-1.7758093	8.42990e-03	-1.7842392
1.1000000	-1.9327459	1.35617e-04	-1.9237921	9.08943e-03	-1.9328815
1.1500000	-2.0861478	1.33315e-04	-2.0765374	9.74370e-03	-2.0862812
1.2000000	-2.2436767	1.29641e-04	-2.2334236	1.03827e-02	-2.2438063
1.2500000	-2.4045580	1.24567e-04	-2.3936880	1.09946e-02	-2.4046826
1.3000000	-2.5678572	1.18088e-04	-2.5564098	1.15655e-02	-2.5679753
1.3500000	-2.7324606	1.10229e-04	-2.7204915	1.20794e-02	-2.7325708
1.4000000	-2.8970542	1.01053e-04	-2.8846378	1.25174e-02	-2.8971552
1.4500000	-3.0601008	9.06630e-05	-3.0473332	1.28582e-02	-3.0601915
1.5000000	-3.2198152	7.92162e-05	-3.2068174	1.30770e-02	-3.2198944
1.5500000	-3.3741363	6.69289e-05	-3.3610577	1.31456e-02	-3.3742033
1.6000000	-3.5206974	5.40882e-05	-3.5077197	1.30317e-02	-3.5207515
1.6500000	-3.6567931	4.10637e-05	-3.6441353	1.26989e-02	-3.6568342
1.7000000	-3.7793439	2.83210e-05	-3.7672666	1.21056e-02	-3.7793722
1.7500000	-3.8848571	1.64354e-05	-3.8736684	1.12051e-02	-3.8848735
1.8000000	-3.9693840	6.10969e-06	-3.9594456	9.94455e-03	-3.9693901
1.8500000	-4.0284742	1.80821e-06	-4.0202079	8.26457e-03	-4.0284724
1.9000000	-4.0571248	-6.30406e-06	-4.0510201	6.09846e-03	-4.0571185
1.9500000	-4.0497257	-6.17426e-06	-4.0463480	3.37148e-03	-4.0497195
2.0000000	-4.0000000	-8.88178e-16	-4.0000000	8.88178e-16	-4.0000000

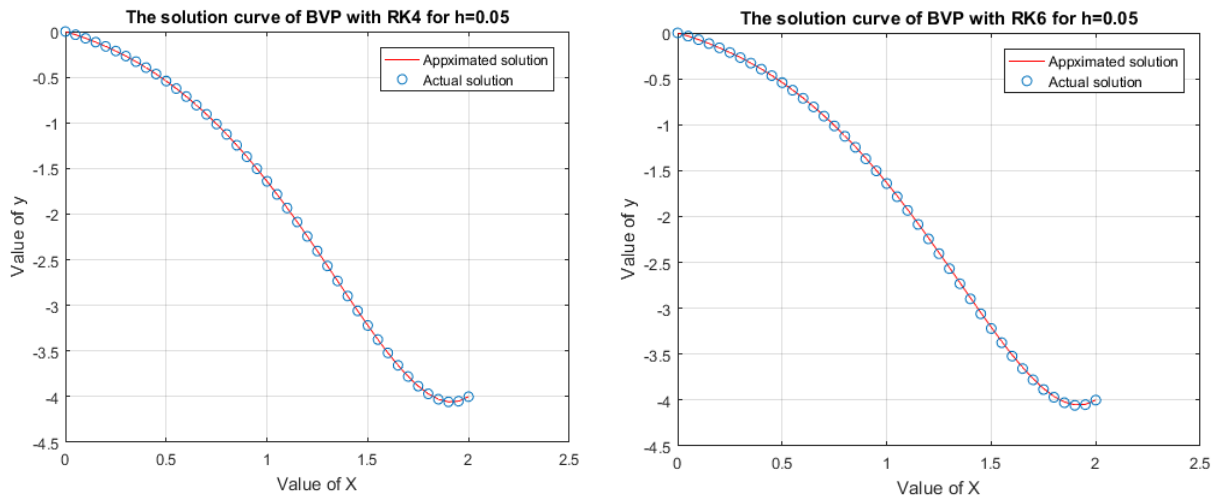


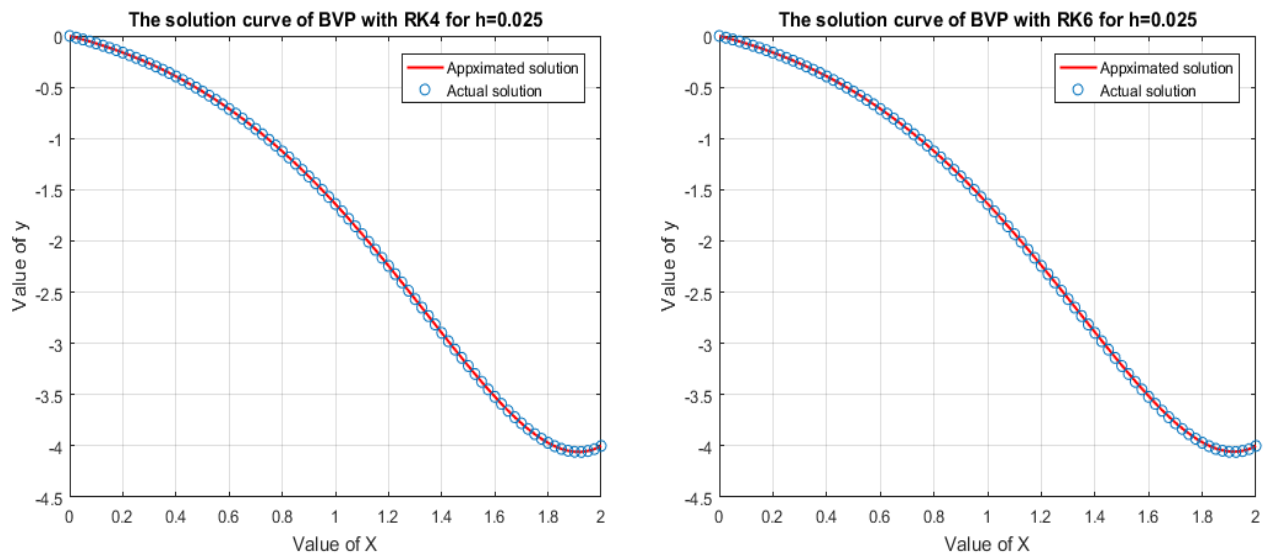
Fig 5: Graphical Representation of Approximated and Exact Solution of BVP with h=0.05.

Table 6: Computed result and absolute maximum error of BVP with step size 0.025.

$x_n$	Approximation of RK4		Approximation of RK6		Actual solution
	Computed $y_n$	Error $e_n$	Computed $y_n$	Error $e_n$	
0.0000000	0.0000000	0.00000e+00	0.0000000	0.00000e+00	0.0000000
0.0250000	-0.0170876	9.63739e-07	-0.0170585	3.00443e-05	-0.0170886
0.0500000	-0.0350399	1.94270e-06	-0.0349792	6.26581e-05	-0.0350418
0.0750000	-0.0538885	2.93577e-06	-0.0537935	9.79337e-05	-0.0538914
0.1000000	-0.0736652	3.94177e-06	-0.0735332	1.35964e-04	-0.0736691
0.1250000	-0.0944019	4.95944e-06	-0.0942300	1.76840e-04	-0.0944068

0.1500000	-0.1161306	5.98746e-06	-0.1159159	2.20653e-04	-0.1161365
0.1750000	-0.1388833	7.02441e-06	-0.1386228	2.67493e-04	-0.1388903
0.2000000	-0.1626921	8.06879e-06	-0.1623827	3.17451e-04	-0.1627002
0.2250000	-0.1875890	9.11904e-06	-0.1872275	3.70612e-04	-0.1875981
0.2500000	-0.2136058	1.01735e-05	-0.2131889	4.27062e-04	-0.2136159
0.2750000	-0.2407740	1.12304e-05	-0.2402984	4.86884e-04	-0.2407853
0.3000000	-0.2691251	1.22880e-05	-0.2685873	5.50160e-04	-0.2691374
0.3250000	-0.2986901	1.33444e-05	-0.2980865	6.16965e-04	-0.2987034
0.3500000	-0.3294995	1.43975e-05	-0.3288265	6.87373e-04	-0.3295139
0.3750000	-0.3615833	1.54454e-05	-0.3608373	7.61454e-04	-0.3615988
0.4000000	-0.3949711	1.64858e-05	-0.3941483	8.39272e-04	-0.3949876
0.4250000	-0.4296915	1.75166e-05	-0.4287882	9.20887e-04	-0.4297090
0.4500000	-0.4657725	1.85354e-05	-0.4647847	1.00635e-03	-0.4657910
0.4750000	-0.5032410	1.95399e-05	-0.5021648	1.09571e-03	-0.5032605
0.5000000	-0.5421230	2.05276e-05	-0.5409545	1.18901e-03	-0.5421435
0.5250000	-0.5824432	2.14960e-05	-0.5811784	1.28628e-03	-0.5824647
0.5500000	-0.6242251	2.24424e-05	-0.6228600	1.38754e-03	-0.6242476
0.5750000	-0.6674908	2.33643e-05	-0.6660214	1.49280e-03	-0.6675142
0.6000000	-0.7122607	2.42588e-05	-0.7106828	1.60208e-03	-0.7122849
0.6250000	-0.7585534	2.51232e-05	-0.7568632	1.71535e-03	-0.7585785
0.6500000	-0.8063857	2.59548e-05	-0.8045791	1.83260e-03	-0.8064117
0.6750000	-0.8557724	2.67505e-05	-0.8538453	1.95380e-03	-0.8557991
0.7000000	-0.9067257	2.75074e-05	-0.9046743	2.07889e-03	-0.9067532
0.7250000	-0.9592557	2.82227e-05	-0.9570761	2.20781e-03	-0.9592839
0.7500000	-1.0133695	2.88932e-05	-1.0110579	2.34049e-03	-1.0133984
0.7750000	-1.0690716	2.95161e-05	-1.0666243	2.47682e-03	-1.0691011
0.8000000	-1.1263631	3.00884e-05	-1.1237765	2.61667e-03	-1.1263932
0.8250000	-1.1852420	3.06069e-05	-1.1825127	2.75992e-03	-1.1852726
0.8500000	-1.2457023	3.10688e-05	-1.2428270	2.90639e-03	-1.2457334
0.8750000	-1.3077346	3.14710e-05	-1.3047101	3.05591e-03	-1.3077660
0.9000000	-1.3713249	3.18107e-05	-1.3681484	3.20825e-03	-1.3713567
0.9250000	-1.4364549	3.20851e-05	-1.4331238	3.36318e-03	-1.4364870
0.9500000	-1.5031015	3.22913e-05	-1.4996134	3.52041e-03	-1.5031338
0.9750000	-1.5712367	3.24267e-05	-1.5675894	3.67966e-03	-1.5712691
1.0000000	-1.6408266	3.24888e-05	-1.6370185	3.84058e-03	-1.6408591
1.0250000	-1.7118319	3.24750e-05	-1.7078615	4.00280e-03	-1.7118643
1.0500000	-1.7842068	3.23831e-05	-1.7800733	4.16590e-03	-1.7842392
1.0750000	-1.8578991	3.22111e-05	-1.8536019	4.32944e-03	-1.8579313
1.1000000	-1.9328495	3.19571e-05	-1.9283886	4.49290e-03	-1.9328815
1.1250000	-2.0089914	3.16194e-05	-2.0043672	4.65576e-03	-2.0090230
1.1500000	-2.0862500	3.11967e-05	-2.0814637	4.81742e-03	-2.0862812
1.1750000	-2.1645422	3.06879e-05	-2.1595957	4.97723e-03	-2.1645729
1.2000000	-2.2437762	3.00924e-05	-2.2386718	5.13448e-03	-2.2438063
1.2250000	-2.3238506	2.94097e-05	-2.3185916	5.28842e-03	-2.3238800
1.2500000	-2.4046539	2.86400e-05	-2.3992443	5.43823e-03	-2.4046826
1.2750000	-2.4860643	2.77837e-05	-2.4805090	5.58300e-03	-2.4860920
1.3000000	-2.5679485	2.68421e-05	-2.5622535	5.72179e-03	-2.5679753
1.3250000	-2.6501616	2.58166e-05	-2.6443338	5.85355e-03	-2.6501874
1.3500000	-2.7325461	2.47095e-05	-2.7265937	5.97717e-03	-2.7325708
1.3750000	-2.8149314	2.35237e-05	-2.8088635	6.09145e-03	-2.8149550
1.4000000	-2.8971329	2.22628e-05	-2.8909601	6.19510e-03	-2.8971552
1.4250000	-2.9789514	2.09313e-05	-2.9726855	6.28675e-03	-2.9789723
1.4500000	-3.0601719	1.95345e-05	-3.0538265	6.36492e-03	-3.0601915
1.4750000	-3.1405635	1.80785e-05	-3.1341536	6.42802e-03	-3.1405816
1.5000000	-3.2198779	1.65707e-05	-3.2134201	6.47436e-03	-3.2198944
1.5250000	-3.2978485	1.50194e-05	-3.2913614	6.50213e-03	-3.2978635
1.5500000	-3.3741898	1.34341e-05	-3.3676939	6.50941e-03	-3.3742033
1.5750000	-3.4485963	1.18256e-05	-3.4421141	6.49412e-03	-3.4486082
1.6000000	-3.5207413	1.02062e-05	-3.5142974	6.45408e-03	-3.5207515

1.6250000	-3.5902757	8.58945e-06	-3.5838973	6.38695e-03	-3.5902842
1.6500000	-3.6568272	6.99069e-06	-3.6505439	6.29023e-03	-3.6568342
1.6750000	-3.7199989	5.42689e-06	-3.7138431	6.16128e-03	-3.7200044
1.7000000	-3.7793683	3.91689e-06	-3.7733750	5.99728e-03	-3.7793722
1.7250000	-3.8344855	2.48149e-06	-3.8286928	5.79523e-03	-3.8344880
1.7500000	-3.8848724	1.14358e-06	-3.8793215	5.55197e-03	-3.8848735
1.7750000	-3.9300207	7.16596e-08	-3.9247565	5.26412e-03	-3.9300207
1.8000000	-3.9693913	1.13665e-06	-3.9644620	4.92809e-03	-3.9693901
1.8250000	-4.0024117	2.02122e-06	-3.9978695	4.54011e-03	-4.0024096
1.8500000	-4.0284751	2.69245e-06	-4.0243763	4.09614e-03	-4.0284724
1.8750000	-4.0469388	3.11449e-06	-4.0433437	3.59193e-03	-4.0469357
1.9000000	-4.0571218	3.24832e-06	-4.0540956	3.02297e-03	-4.0571185
1.9250000	-4.0583036	3.05161e-06	-4.0559160	2.38449e-03	-4.0583005
1.9500000	-4.0497220	2.47845e-06	-4.0480481	1.67144e-03	-4.0497195
1.9750000	-4.0305713	1.47916e-06	-4.0296913	8.78492e-04	-4.0305698
2.0000000	-4.0000000	7.10543e-15	-4.0000000	6.21725e-15	-4.0000000



**Fig 6:** Graphical Representation of Approximated and Exact Solution of BVP with  $h=0.025$ .

### 6. Comparison between the obtained results

The obtained result of our model examples of IVP and BVP is expressed through **Table 1-3** and **Table 4-6** and graphical representations are in **Fig 1-3** and **Fig 4-6**, respectively. From the tables and figures for both examples and both methods, we observed that if step size leads to deterioration the estimated solution converges faster to the exact solution such that in the limit when step size tends to zero the errors go to zero. Also, we observed from **Table 1-3** that in solving IVP the method RK4 gives a more accurate and better result than the method RK6 and it has been clarified with the adjacent graphs **Fig 1-3**.

We also see from **Table 4-6** and **Fig 4-6** that in solving BVP the integrator RK4 gives better approximation and converges faster to the actual solution than RK6.

### 7. CONCLUSION:

In this paper, we have applied Runge-Kutta fourth order and Butcher’s sixth ordered methods to solve IVP and BVP of ODEs and the results are found in good agreement with exact solutions. We assume three different step sizes for each problem to arrive at a more accurate result with competition between the exact and approximated solutions in tables and figures. From the above investigation of these methods, observed that the convergence of approximated solutions to exact solutions increases with decreasing step size in both cases and the convergence rate of RK4 is superior to RK6 in comparison to the exact solutions. Hence it is cleared from this study that to find the more accurate result the lower order RK4 method is appropriate than the higher order RK6 method.

## 8. ACKNOWLEDGEMENT:

We would like to express our gratitude to our dear students Rokhsana Yeasmin, Sumaiya Pervin, Sayantani Biswas for their effortless support in completing this paper. We would also like to thank our colleagues for their advice and support.

## 9. CONFLICTS OF INTEREST:

The authors declare that there is no conflict of interest concerning the publication of this paper.

## 10. REFERENCES:

- 1) Ababneh, O.Y., Ahmad, R. and Ismail, E.S. (2009). New multi-step Runge-Kutta method. *Appl. Mathematical Sciences*, **3**(45), 2255-2262. <http://www.m-hikari.com/ams/ams-password-2009/ams-password45-48-2009/ababnehAMS45-48-2009-2.pdf>
- 2) Ademiluyi, R.A., Babatola, P.O. and Kayode, S.J. (2001). Semi implicit Rational Runge-Kutta formulas of approximation of stiff initial value problems in ODEs. *Journal of Mathematical Science and Education*, **3**, pp.1-25.
- 3) Agam, S.A. and Yahaya, Y.A. (2014). A highly efficient implicit Runge-Kutta method for first order ordinary differential equations. *African j. of mathematics and Computer Science Research*, **7**(5), pp.55-60. <https://academicjournals.org/journal/AJMCSR/article-abstract/9D7ECC646284>
- 4) Ahmed M, and Iqbal MA. (2020). An execution of a mathematical example using Euler's Phi-function in Hill Chiper cryptosystem, *Int. J. Mat. Math. Sci.*, **2**(6), 99-103. <https://doi.org/10.34104/ijmms.020.0990103>
- 5) Akanbi, M.A. (2011). On 3-stage geometric explicit Runge-Kutta method for singular autonomous initial value problems in ordinary differential equations. *Computing*, **92**(3), pp.243-263.
- 6) Arora, G. (2019). Pratiksha. A cumulative study on differential transforms method. *International J. of Mathematical, Engineering and Management Sciences*, **4**(1), pp.170-181.
- 7) Balagurusamy, E. (2006). *Numerical Methods*. Tata McGraw-Hill, New Delhi. [https://books.google.co.in/books?id=anwHfrXY8\\_wC&printsec=frontcover#v=onepage&q&f=false](https://books.google.co.in/books?id=anwHfrXY8_wC&printsec=frontcover#v=onepage&q&f=false)
- 8) Burden, R.L. and Faires, D.J. (1985). *Numerical analysis*. 3<sup>rd</sup> edn Boston. MA: PWS Publishing Company.
- 9) Butcher, J.C. (1963). Coefficients for the study of Runge-Kutta integration processes. *Journal of the Australian Mathematical Society*, **3**(2), pp.185-201. <https://doi.org/10.1017/S1446788700027932>
- 10) Butcher, J.C. (1964). On Runge-Kutta processes of high order. *Journal of the Australian Mathematical Society*, **4**(2), pp.179-194. <https://doi.org/10.1017/S1446788700023387>
- 11) Chauhan, V. and Srivastava, P.K. (2018). Trio-Geometric mean-based three-stage Runge-Kutta algorithm to solve initial value problem arising in autonomous systems. *Intern. J. of Model., Simul., and Scienti. Comput.*, **9**(4), p.1850026. <https://doi.org/10.1142/S1793962318500265>
- 12) Dormand, J.R. and Prince, P.J. (1980). A family of embedded Runge-Kutta formulae. *J. of computational and applied mathematics*, **6**(1), 19-26. [https://doi.org/10.1016/0771-050X\(80\)90013-3](https://doi.org/10.1016/0771-050X(80)90013-3)
- 13) Eaqub Ali, S.M. (2006). *A Text Book of Numerical Methods with Computer Programming*. Beauty Publication, Khulna.
- 14) Fehlberg, E. (1964). New high-order Runge-Kutta formulas with step size control for systems of first-and second-order differential equations. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, **44**(S1 S1), pp.T17-T29.
- 15) Fehlberg, E. (1966). New High-Order Runge-Kutta Formulas with an Arbitrarily Small Truncation Error. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, **46**(1), pp.1-16.
- 16) Gerald, C.F. (2004). *Applied numerical analysis*. Pearson Education India. <http://www.cse.iitm.ac.in/~vplab/downloads/opt/Applied%20Numerical%20Analysis.pdf>
- 17) Gowri, P., Priyadharsini, S., Maheswari, T. (2017). A Case Study on Runge-Kutta 4th Order Differential Equations and Its Application. *Imperial J. of Interdisciplinary Research*, **3**, 2.
- 18) Hossain, B., Hossain, J., Miah, M. and Alam, S. (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*, **6**(6), pp.243-253. <https://doi.org/10.11648/j.acm.20170606.12>
- 19) Iqbal MA, Hossain MR, and Hassan MK. (2020). The metric of space-time curvature in a weak gravitational field and its consequence in Newtonian approximation, *Int. J. Mat. Math. Sci.*, **2**(5), 87-92. <https://doi.org/10.34104/ijmms.020.087092>
- 20) Islam, M.A. (2015). A comparative study on numerical solutions of initial value problems (IVP) for ordinary differential equations (ODE)

- with Euler and Runge-Kutta Methods. *American J. of computational mathematics*, **5**(3), p.393.
- 21) Kutta, W. (1901). Beitrag zur näherungsweise Integration totaler Differentialgleichungen. *Z. Math. Phys.*, **46**, pp.435-453.
- 22) Lambert, J.D. (1973). Computational methods in ordinary differential equations. <https://doi.org/10.1002/zamm.19740540726>
- 23) Mathews, J.H. (1992). *Numerical methods for mathematics, science and engineering* (Vol. 10). Prentice-Hall International. <https://go-pdf.online/numerical-methods-mathematics-science-engineering-mathews.pdf>
- 24) Nyström, E.J. (1925). *Über die numerische Integration von Differentialgleichungen: (Mitgeteilt am 23. Sept. 1925 von E. Lindelöf und KF Sundman)*. Societ asscientiarum Fennica.
- 25) Paul, G.C., Ismail, A.I.M. and Karim, M.F. (2014). Implementation of method of lines to predict water levels due to a storm along the coastal region of Bangladesh. *Journal of Oceanography*, **70**(3), pp.199-210. <https://link.springer.com/article/10.1007%2Fs10872-014-0224-x>
- 26) Paul, G.C. and Senthil Kumar, S. (2015). Exploration on initial structures of extrasolar proto planets via new explicit RKAHeM (4, 4) method. *The Egyptian Journal of Remote Sensing and Space Science*, **18**(1), pp.1-8.
- 27) Paul, G.C., Ismail, A.I.M., Rahman, A., Karim, M.F. and Hoque, A. (2016). Development of tide-surge interaction model for the coastal region of Bangladesh. *Estuaries and coasts*, **39**(6), pp. 1582-1599. <https://www.jstor.org/stable/44857792?seq=1>
- 28) Ralston, A. (1962). Runge-Kutta methods with minimum error bounds. *Mathematics of computation*, **16**(80), pp.431-437.
- 29) Sastry, S.S. (2012). *Introductory methods of numerical analysis*. PHI Learning Pvt. Ltd.
- 30) Senthilkumar, S. (2009). New embedded Runge-Kutta fourth order four stage algorithms for raster and time-multiplexing cellular neural networks simulation. *Tiruchirappalli, Tamilnadu, India: National Institute of Technology*.
- 31) Senthilkumar, S. and Paul, G.C. (2012). Application of new RKAHeM (4, 4) technique to analyze the structure of initial extrasolar giant proto planets. *Earth Science Informatics*, **5**(1), pp.23-31. <https://link.springer.com/article/10.1007/s12145-011-0093-4>
- 32) Yaakub, A.R. and Evans, D.J. (1999). A fourth order Runge-Kutta RK (4, 4) method with error control. *International journal of computer mathematics*, **71**(3), pp. 383-411. <https://doi.org/10.1080/00207169908804817>

**Citation:** Banu MS, Raju I, and Mondal S. (2021). A comparative study on classical fourth order and butcher sixth order Runge-Kutta methods with initial and boundary value problems, *Int. J. Mat. Math. Sci.*, **3**(1), 8-21. <https://doi.org/10.34104/ijmms.021.08021> 