

A Study on Numerical Solutions of Second Order Initial Value Problems (IVP) for Ordinary Differential Equations with Fourth Order and Butcher's Fifth Order Runge-Kutta Methods

Md. Jahangir Hossain¹, Md. Shah Alam², Md. Babul Hossain^{2,*}

¹Department of Basic Science, World University of Bangladesh, Dhaka, Bangladesh

²Department of Mathematics, Mawlana Bhashani Science and Technology University, Tangail, Bangladesh

Abstract In this paper, we mainly present fourth order Runge-Kutta (RK4) and Butcher's fifth order Runge-Kutta (RK5) Methods for solving second order initial value problems (IVP) for ordinary differential equations (ODE). These two proposed methods are quite proficient and practically well suited for solving engineering problems based on such problems. To obtain the accuracy of the numerical solutions for this study, we have compared the approximate solutions with the exact solutions and originate a good agreement. Numerical and graphical comparisons between fourth order Runge-Kutta method and Butcher's fifth order Runge-Kutta method have been presented. In order to, achieve more accuracy in the solution; the step size needs to be very small. Moreover, the error terms have been analyzed of these two proposed methods for different step sizes to scrutinize supremacy. A numerical example is given to exhibit the reliability and efficiency of these two methods.

Keywords Differential Equation, Initial Value Problem, Error analysis, Butcher's Method, Runge-Kutta Method

1. Introduction

Most of the problems in science, mathematical physics and engineering are formulated by differential equations. The solution of differential equations is a significant part to develop the various modeling in science and engineering. There are many analytical methods for finding the solution of ordinary differential equations. But a few numbers of differential equations have analytic solutions where a large numbers of differential equations have no analytic solutions. In this case we use the numerical methods to get the approximate solution of a differential equation under the prescribed initial condition or conditions. Numerical methods are widely used for solving differential equations where it is difficult to obtain the exact solutions. There are many types of numerical methods for solving initial value problems for ordinary differential equations such as Eulers method, Runge-Kutta fourth order method (RK4). Runge-Kutta method is the powerful numerical technique to solve the initial value problems (IVP). This method widely used one since it gives reliable starting values and is particularly suitable when the computation of higher

derivatives is complicated. Also this method gives more accuracy of the numerical results and used in most computer programs for a differential equation. Many authors try to spread these methods to get more accurate solution of Initial Value Problems (IVP).

In [1] the author presented fifth order improved Runge-Kutta method for solving ordinary differential equation, in [2] the author presented on fifth order Runge-Kutta methods, also in [3] the author presented a comparative study on numerical solutions of Initial Value Problems (IVP) for ordinary differential equations (ODE) with Euler and Runge-Kutta methods. Also in [4-15] presented various numerical methods for finding approximations of Initial Value Problems (IVP) in ordinary differential equations (ODE). In this paper, we converted the coefficients of Butcher's RK5 table to the general fifth order Runge-Kutta method. Also we use fourth order Runge-Kutta (RK4) and fifth order Runge-Kutta (RK5) methods for solving second order initial value problems in ordinary differential equations. Approximate solutions are compared to the exact solutions by calculate the error terms.

2. Numerical Methods

In this section we will represent two methods Runge-Kutta fourth order (RK4) and Butcher's fifth order Runge-Kutta (RK5) methods for solving initial value problem (IVP) for

* Corresponding author:

babulhossainh@yahoo.com (Md. Babul Hossain)

Published online at <http://journal.sapub.org/ajcam>

Copyright © 2017 Scientific & Academic Publishing. All Rights Reserved

ordinary differential equations (ODEs). These methods are distinguished by their order in the sense that agrees with Taylor's series solution up to terms of h^r where r is the order of the method.

Next we will apply these two methods for solving second order initial value problem.

2.1. Runge-Kutta Fourth Order Method

We consider the following initial value problem:

$$\frac{dy}{dx} = f(x, y) \quad (1)$$

with initial condition

$$y(x_0) = y_0 \quad (2)$$

Now the method is based on computing y_{k+1} as follows:

$$y_{k+1} = y_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4) \quad (3)$$

where

$$x_{k+1} = x_k + h$$

$$f_1 = f(x_k, y_k)$$

$$f_2 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f_1\right)$$

$$f_3 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f_2\right)$$

$$f_4 = f(x_k + h, y_k + hf_3)$$

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

2.2. Fourth Order Runge-Kutta Method for Solving Initial Value Problem (IVP) for the System of Two Differential Equation

We consider the following system of differential equations

$$\frac{dx}{dt} = p(t, x, y) \quad (4)$$

$$\frac{dy}{dt} = Q(t, x, y) \quad (5)$$

with the initial conditions

$$x(t_0) = x_0, y(t_0) = y_0 \quad (6)$$

The fourth order RungeKutta method will become as:

$$x_{i+1} = x_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (7)$$

$$y_{i+1} = y_i + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4) \quad (8)$$

where

$$t_{i+1} = t_i + h$$

$$k_1 = P(t_i, x_i, y_i)$$

$$l_1 = Q(t_i, x_i, y_i)$$

$$k_2 = P\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1, y_i + \frac{h}{2}l_1\right)$$

$$l_2 = Q\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_1, y_i + \frac{h}{2}l_1\right)$$

$$k_3 = P\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_2, y_i + \frac{h}{2}l_2\right)$$

$$l_3 = Q\left(t_i + \frac{h}{2}, x_i + \frac{h}{2}k_2, y_i + \frac{h}{2}l_2\right)$$

$$k_4 = P(t_i + h, x_i + hk_3, y_i + hl_3)$$

$$l_4 = Q(t_i + h, x_i + hk_3, y_i + hl_3)$$

2.3. Butcher's Fifth Order Runge-Kutta Method

We consider the following initial value problem

$$\frac{dy}{dx} = f(x, y) \quad (9)$$

With initial condition

$$y(x_0) = y_0 \quad (10)$$

Now the method is based on computing y_{k+1} as follows

$$y_{k+1} = y_k + \frac{h}{90}(7f_1 + 32f_3 + 12f_4 + 32f_5 + 7f_6) \quad (11)$$

where

$$x_{k+1} = x_k + h$$

$$f_1 = f(x_k, y_k)$$

$$f_2 = f\left(x_k + \frac{1}{4}h, y_k + \frac{1}{4}hf_1\right)$$

$$f_3 = f\left(x_k + \frac{1}{4}h, y_k + \frac{1}{8}hf_1 + \frac{1}{8}hf_2\right)$$

$$f_4 = f\left(x_k + \frac{1}{2}h, y_k - \frac{1}{2}hf_2 + hf_3\right)$$

$$f_5 = f\left(x_k + \frac{3}{4}h, y_k - \frac{3}{4}hf_1 + \frac{2}{4}hf_2 + \frac{12}{4}hf_3 - \frac{12}{4}hf_4 + \frac{8}{4}hf_5\right)$$

$$k = 0, 1, 2, \dots$$

2.4. Butcher's Fifth Order Runge-Kutta Method for Solving Initial Value Problem (IVP) for the System of Two Differential Equations

We consider the following system of differential equations:

$$\frac{dx}{dt} = P(t, x, y) \quad (12)$$

$$\frac{dy}{dt} = Q(t, x, y) \quad (13)$$

with the initial conditions

$$x(t_0) = x_0, y(t_0) = y_0 \quad (14)$$

The Butcher's fifth order Runge-Kutta method will become as:

$$x_{i+1} = x_i + \frac{h}{90}(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6) \quad (15)$$

$$y_{i+1} = y_i + \frac{h}{90}(7l_1 + 32l_3 + 12l_4 + 32l_5 + 7l_6) \quad (16)$$

Where

$$t_{i+1} = t_i + h$$

$$k_1 = P(t_i, x_i, y_i)$$

$$l_1 = Q(t_i, x_i, y_i)$$

$$k_2 = P\left(t_i + \frac{1}{4}h, x_i + \frac{1}{4}hk_1, y_i + \frac{1}{4}hl_1\right)$$

$$l_2 = Q\left(t_i + \frac{1}{4}h, x_i + \frac{1}{4}hk_1, y_i + \frac{1}{4}hl_1\right)$$

$$k_3 = P\left(t_i + \frac{1}{4}h, x_i + \frac{1}{8}hk_1 + \frac{1}{8}hk_2, y_i + \frac{1}{8}hl_1 + \frac{1}{8}hl_2\right)$$

$$l_3 = Q\left(t_i + \frac{1}{4}h, x_i + \frac{1}{8}hk_1 + \frac{1}{8}hk_2, y_i + \frac{1}{8}hl_1 + \frac{1}{8}hl_2\right)$$

$$k_4 = P\left(t_i + \frac{1}{2}h, x_i - \frac{1}{2}hk_2 + hk_3, y_i - \frac{1}{2}hl_2 + hl_3\right)$$

$$l_4 = Q\left(t_i + \frac{1}{2}h, x_i - \frac{1}{2}hk_2 + hk_3, y_i - \frac{1}{2}hl_2 + hl_3\right)$$

$$k_5 = P\left(t_i + \frac{3}{4}h, x_i + \frac{3}{16}hk_1 + \frac{9}{16}hk_4, y_i + \frac{3}{16}hl_1 + \frac{9}{16}hl_4\right)$$

$$l_5 = Q\left(t_i + \frac{3}{4}h, x_i + \frac{3}{16}hk_1 + \frac{9}{16}hk_4, y_i + \frac{3}{16}hl_1 + \frac{9}{16}hl_4\right)$$

$$l_6 = Q\left(t_i + h, x_i - \frac{3}{7}hk_1 + \frac{2}{7}hk_2 + \frac{12}{7}hk_3 - \frac{12}{7}hk_4 + \frac{8}{7}hk_5, y_i - \frac{3}{7}hl_1 + \frac{2}{7}hl_2 + \frac{12}{7}hl_3 - \frac{12}{7}hl_4 + \frac{8}{7}hl_5\right)$$

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

3. Error Analysis

Our main goal is to find the more accurate results in numerical solutions of ordinary differential equations. There are two types of errors occurs (Truncation errors and Round-off errors). Truncation error in numerical analysis arises when approximations are used to estimate some quantity. Rounding errors originate from the fact that computer can only represent a limited number of significant figures. Thus, such numbers cannot be represented exactly in computer memory. The discrepancy introduced by this limitation is called Round-off error. The accuracy of the solution will depend on how small we take the step size, h . A numerical method is said to be convergent if the numerical solution approaches the exact solution as the step size h tends to zero. The convergence of initial value problem is calculated by $e_n = |y(t_n) - y_n| < \delta$ where $y(t_n)$ denotes the approximate solution and y_n denotes the exact solution and δ depends on the problem which varies from 10^{-5} . The errors for these two formulas are defined by $errors = |y(t_n) - y_n|$.

4. Illustrative Example

In this section, we illustrate a second order initial value problem as numerical example to compare between these two methods. All the computations are performed by MATLAB software. Numerical results and errors are computed and the outcomes are represented by graphically.

Example:

Consider the initial value problem

$$\frac{d^2 y}{dx^2} + 4y = \sin 3t \quad (17)$$

with the initial conditions $y(0) = 1, y'(0) = 2$ on the interval $0 \leq t \leq 2$. The exact solution of this initial value problem is given by $y(t) = \cos 2t + \frac{13}{10} \sin 2t - \frac{1}{5} \sin 3t$.

Now this second order initial value problem can be written in the form of system of first order differential equations by

putting $x = \frac{dy}{dt}$ as

$$\frac{dy}{dt} = x \quad (18)$$

$$\frac{dx}{dt} = \sin 3t - 4y \quad (19)$$

with the initial conditions

$$y(0) = 1, x(0) = 2 \quad (20)$$

The approximate results, exact results and maximum errors for step sizes 0.1, 0.05, 0.025 and 0.0125 are shown in

Tables 1-4 and the graphs of the numerical solutions are displayed in Figures 1-8.

Table 1. Numerical approximations and maximum errors for step size $h=0.1$

t_n	RK4 Method $h=0.1$		RK5 Method $h=0.1$		Exact value
	$y(t_n)$	Errors	$y(t_n)$	Errors	y_n
0.00	1.000000000E+00	0.0000E+00	1.000000000E+00	0.0000E+00	1.000000000E+00
0.10	1.179231460E+00	1.2061E-06	1.179232656E+00	1.0865E-08	1.179232667E+00
0.20	1.314374988E+00	1.3563E-06	1.314376314E+00	3.0529E-08	1.314376344E+00
0.30	1.402705024E+00	4.2447E-07	1.402705391E+00	5.6906E-08	1.402705448E+00
0.40	1.442863358E+00	1.5472E-06	1.442861723E+00	8.7232E-08	1.442861810E+00
0.50	1.434720049E+00	4.4600E-06	1.434715471E+00	1.1828E-07	1.434715589E+00
0.60	1.379247210E+00	8.1704E-06	1.379238893E+00	1.4660E-07	1.379239040E+00
0.70	1.278422420E+00	1.2502E-05	1.278409750E+00	1.6881E-07	1.278409919E+00
0.80	1.135170779E+00	1.7253E-05	1.135153344E+00	1.8183E-07	1.135153526E+00
0.90	9.533460568E-01	2.2207E-05	9.533236662E-01	1.8316E-07	9.533238494E-01
1.00	7.377429496E-01	2.7133E-05	7.377156457E-01	1.7105E-07	7.377158167E-01
1.10	4.941251350E-01	3.1788E-05	4.940932018E-01	1.4470E-07	4.940933465E-01
1.20	2.292484306E-01	3.5923E-05	2.292124036E-01	1.0426E-07	2.292125078E-01
1.30	-4.914446172E-02	3.9276E-05	-4.918378913E-02	5.0969E-08	-4.918373816E-02
1.40	-3.323810064E-01	4.1585E-05	-3.324225780E-01	1.3018E-08	-3.324225910E-01
1.50	-6.109878809E-01	4.2582E-05	-6.110303779E-01	8.4674E-08	-6.110304626E-01
1.60	-8.749062292E-01	4.2011E-05	-8.749480802E-01	1.6031E-07	-8.749482405E-01
1.70	-1.113799050E+00	3.9638E-05	-1.113838453E+00	2.3577E-07	-1.113838689E+00
1.80	-1.317446829E+00	3.5266E-05	-1.317481788E+00	3.0674E-07	-1.317482095E+00
1.90	-1.476217104E+00	2.8757E-05	-1.476245493E+00	3.6895E-07	-1.476245862E+00
2.00	-1.581583710E+00	2.0055E-05	-1.581603347E+00	4.1844E-07	-1.581603765E+00

Table 2. Numerical approximations and maximum errors for step size $h=0.05$

t_n	RK4 Method $h=0.05$		RK5 Method $h=0.05$		Exact value
	$y(t_n)$	Errors	$y(t_n)$	Errors	y_n
0.00	1.000000000E+00	0.0000E+00	1.000000000E+00	0.0000E+00	1.000000000E+00
0.10	1.179232605E+00	6.1381E-08	1.179232666E+00	4.1046E-10	1.179232667E+00
0.20	1.314376290E+00	5.4799E-08	1.314376343E+00	1.0629E-09	1.314376344E+00
0.30	1.402705468E+00	1.9794E-08	1.402705447E+00	1.8867E-09	1.402705448E+00
0.40	1.442861969E+00	1.5829E-07	1.442861808E+00	2.7932E-09	1.442861810E+00
0.50	1.434715942E+00	3.5321E-07	1.434715585E+00	3.6816E-09	1.434715589E+00
0.60	1.379239635E+00	5.9448E-07	1.379239036E+00	4.4475E-09	1.379239040E+00
0.70	1.278410789E+00	8.7017E-07	1.278409914E+00	4.9904E-09	1.278409919E+00
0.80	1.135154693E+00	1.1670E-06	1.135153520E+00	5.2220E-09	1.135153526E+00
0.90	9.533253203E-01	1.4709E-06	9.533238443E-01	5.0737E-09	9.533238494E-01
1.00	7.377175838E-01	1.7670E-06	7.377158122E-01	4.5019E-09	7.377158167E-01
1.10	4.940953867E-01	2.0402E-06	4.940933430E-01	3.4927E-09	4.940933465E-01
1.20	2.292147825E-01	2.2746E-06	2.292125058E-01	2.0643E-09	2.292125078E-01
1.30	-4.918128384E-02	2.4543E-06	-4.918373843E-02	2.6653E-10	-4.918373816E-02
1.40	-3.324200278E-01	2.5632E-06	-3.324225892E-01	1.8213E-09	-3.324225910E-01
1.50	-6.110278774E-01	2.5852E-06	-6.110304585E-01	4.0943E-09	-6.110304626E-01
1.60	-8.749457350E-01	2.5055E-06	-8.749482341E-01	6.4287E-09	-8.749482405E-01
1.70	-1.113836378E+00	2.3106E-06	-1.113838680E+00	8.6887E-09	-1.113838689E+00
1.80	-1.317480105E+00	1.9900E-06	-1.317482084E+00	1.0735E-08	-1.317482095E+00
1.90	-1.476244324E+00	1.5371E-06	-1.476245849E+00	1.2435E-08	-1.476245862E+00
2.00	-1.581602814E+00	9.5096E-07	-1.581603751E+00	1.3667E-08	-1.581603765E+00

Table 3. Numerical approximations and maximum errors for step size $h=0.025$

t_n	RK4 Method $h=0.025$		RK5 Method $h=0.025$		Exact value y_n
	$y(t_n)$	Errors	$y(t_n)$	Errors	
0.00	1.000000000E+00	0.0000E+00	1.000000000E+00	0.0000E+00	1.000000000E+00
0.10	1.179232663E+00	3.3814E-09	1.179232667E+00	1.3810E-11	1.179232667E+00
0.20	1.314376342E+00	2.4649E-09	1.314376344E+00	3.4642E-11	1.314376344E+00
0.30	1.402705451E+00	2.7029E-09	1.402705448E+00	6.0227E-11	1.402705448E+00
0.40	1.442861822E+00	1.1820E-08	1.442861810E+00	8.7762E-11	1.442861810E+00
0.50	1.434715613E+00	2.4380E-08	1.434715589E+00	1.1412E-10	1.434715589E+00
0.60	1.379239080E+00	3.9720E-08	1.379239040E+00	1.3610E-10	1.379239040E+00
0.70	1.278409976E+00	5.7067E-08	1.278409918E+00	1.5065E-10	1.278409919E+00
0.80	1.135153601E+00	7.5574E-08	1.135153525E+00	1.5515E-10	1.135153526E+00
0.90	9.533239437E-01	9.4342E-08	9.533238493E-01	1.4759E-10	9.533238494E-01
1.00	7.377159292E-01	1.1244E-07	7.377158166E-01	1.2681E-10	7.377158167E-01
1.10	4.940934755E-01	1.2891E-07	4.940933464E-01	9.2548E-11	4.940933465E-01
1.20	2.292126506E-01	1.4278E-07	2.292125078E-01	4.5563E-11	2.292125078E-01
1.30	-4.918358513E-02	1.5304E-07	-4.918373815E-02	1.2399E-11	-4.918373816E-02
1.40	-3.324224323E-01	1.5871E-07	-3.324225909E-01	7.8682E-11	-3.324225910E-01
1.50	-6.110303038E-01	1.5881E-07	-6.110304624E-01	1.4986E-10	-6.110304626E-01
1.60	-8.749480880E-01	1.5245E-07	-8.749482403E-01	2.2194E-10	-8.749482405E-01
1.70	-1.113838550E+00	1.3882E-07	-1.113838688E+00	2.9060E-10	-1.113838689E+00
1.80	-1.317481978E+00	1.1734E-07	-1.317482095E+00	3.5145E-10	-1.317482095E+00
1.90	-1.476245774E+00	8.7653E-08	-1.476245861E+00	4.0033E-10	-1.476245862E+00
2.00	-1.581603715E+00	4.9773E-08	-1.581603765E+00	4.3354E-10	-1.581603765E+00

Table 4. Numerical approximations and maximum errors for step size $h=0.0125$

t_n	RK4 Method $h=0.0125$		RK5 Method $h=0.0125$		Exact value y_n
	$y(t_n)$	Errors	$y(t_n)$	Errors	
0.00	1.000000000E+00	0.0000E+00	1.000000000E+00	0.0000E+00	1.000000000E+00
0.10	1.179232666E+00	1.9687E-10	1.179232667E+00	4.4609E-13	1.179232667E+00
0.20	1.314376344E+00	1.2374E-10	1.314376344E+00	1.1031E-12	1.314376344E+00
0.30	1.402705449E+00	2.1496E-10	1.402705448E+00	1.8983E-12	1.402705448E+00
0.40	1.442861811E+00	7.9895E-10	1.442861810E+00	2.7456E-12	1.442861810E+00
0.50	1.434715590E+00	1.5953E-09	1.434715589E+00	3.5467E-12	1.434715589E+00
0.60	1.379239043E+00	2.5617E-09	1.379239040E+00	4.2026E-12	1.379239040E+00
0.70	1.278409922E+00	3.6489E-09	1.278409919E+00	4.6194E-12	1.278409919E+00
0.80	1.135153530E+00	4.8035E-09	1.135153526E+00	4.7180E-12	1.135153526E+00
0.90	9.533238554E-01	5.9688E-09	9.533238494E-01	4.4376E-12	9.533238494E-01
1.00	7.377158238E-01	7.0865E-09	7.377158167E-01	3.7447E-12	7.377158167E-01
1.10	4.940933546E-01	8.0964E-09	4.940933465E-01	2.6338E-12	4.940933465E-01
1.20	2.292125168E-01	8.9377E-09	2.292125078E-01	1.1319E-12	2.292125078E-01
1.30	-4.918372862E-02	9.5484E-09	-4.918373816E-02	7.0401E-13	-4.918373816E-02
1.40	-3.324225811E-01	9.8670E-09	-3.324225910E-01	2.7883E-12	-3.324225910E-01
1.50	-6.110304528E-01	9.8336E-09	-6.110304626E-01	5.0115E-12	-6.110304626E-01
1.60	-8.749482311E-01	9.3927E-09	-8.749482405E-01	7.2475E-12	-8.749482405E-01
1.70	-1.113838680E+00	8.4964E-09	-1.113838689E+00	9.3594E-12	-1.113838689E+00
1.80	-1.317482088E+00	7.1089E-09	-1.317482095E+00	1.1210E-11	-1.317482095E+00
1.90	-1.476245856E+00	5.2111E-09	-1.476245862E+00	1.2670E-11	-1.476245862E+00
2.00	-1.581603762E+00	2.8059E-09	-1.581603765E+00	1.3625E-11	-1.581603765E+00

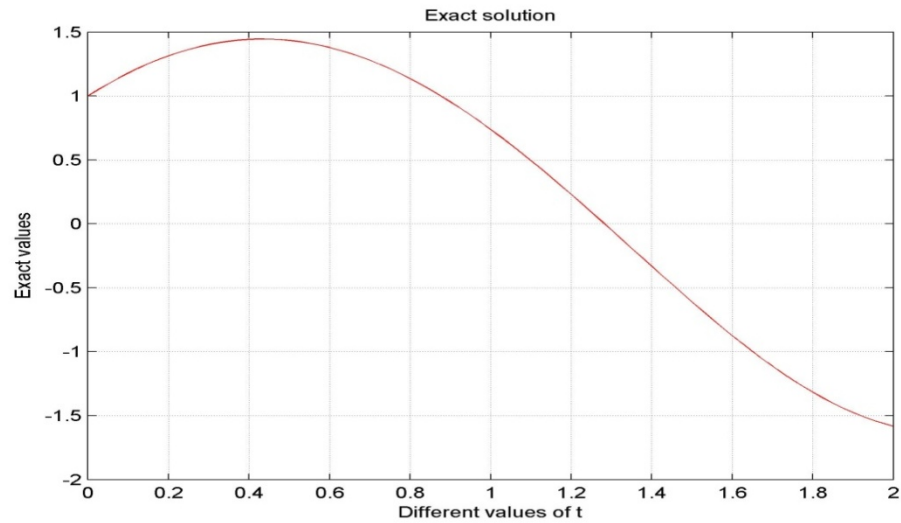


Figure 1. Exact numerical solutions

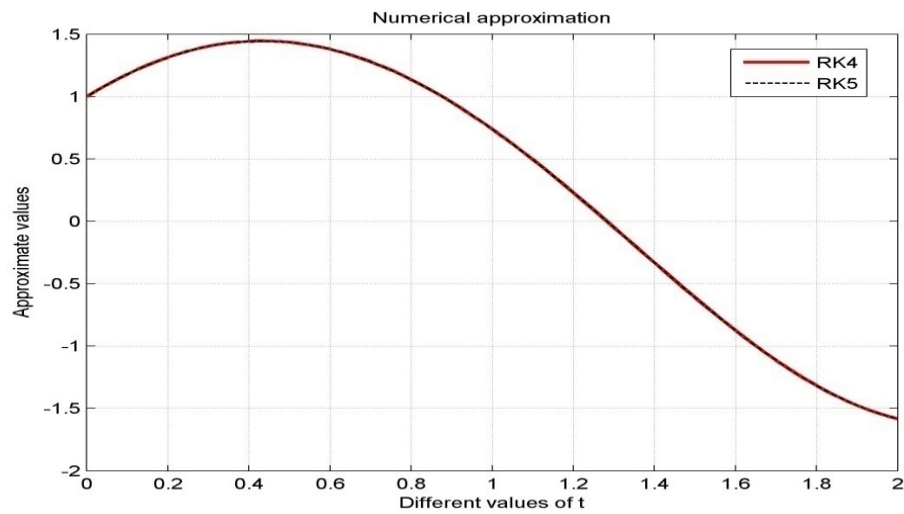
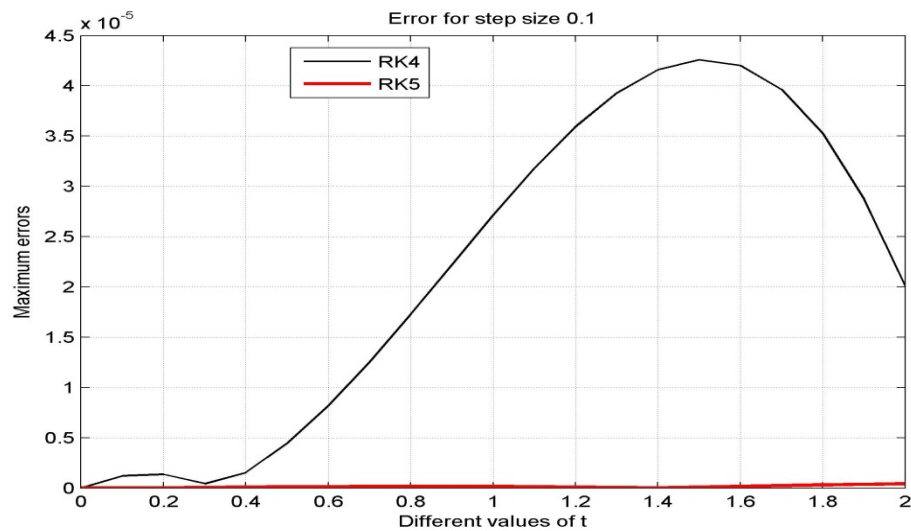


Figure 2. Approximate numerical solutions

Figure 3. Error for step size $h=0.1$

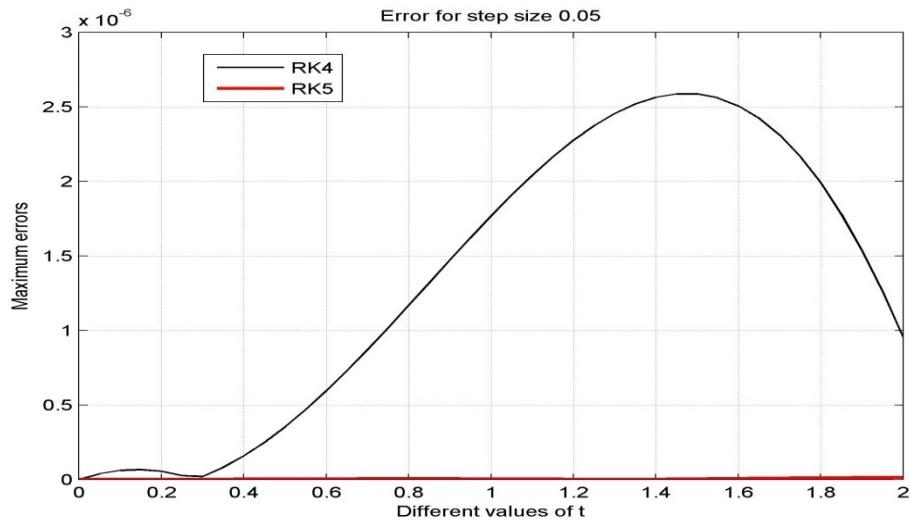


Figure 4. Error for step size $h=0.05$

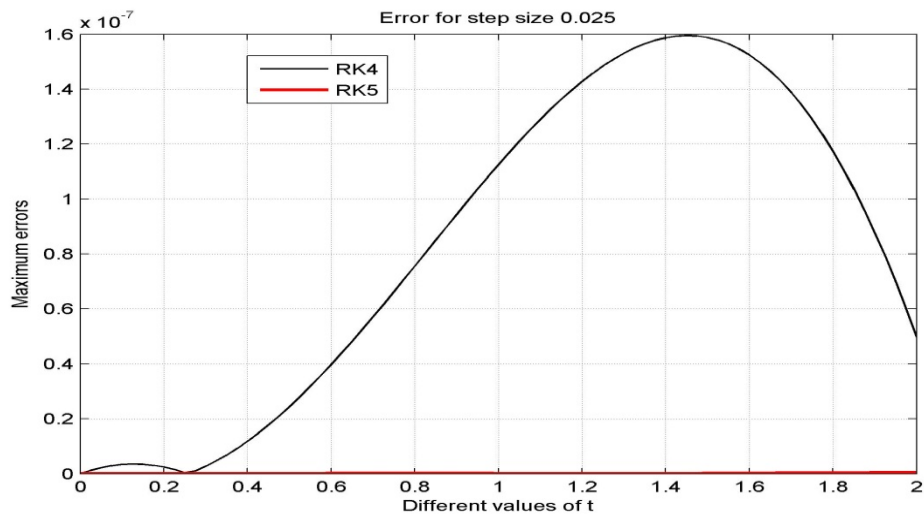


Figure 5. Error for step size $h=0.025$

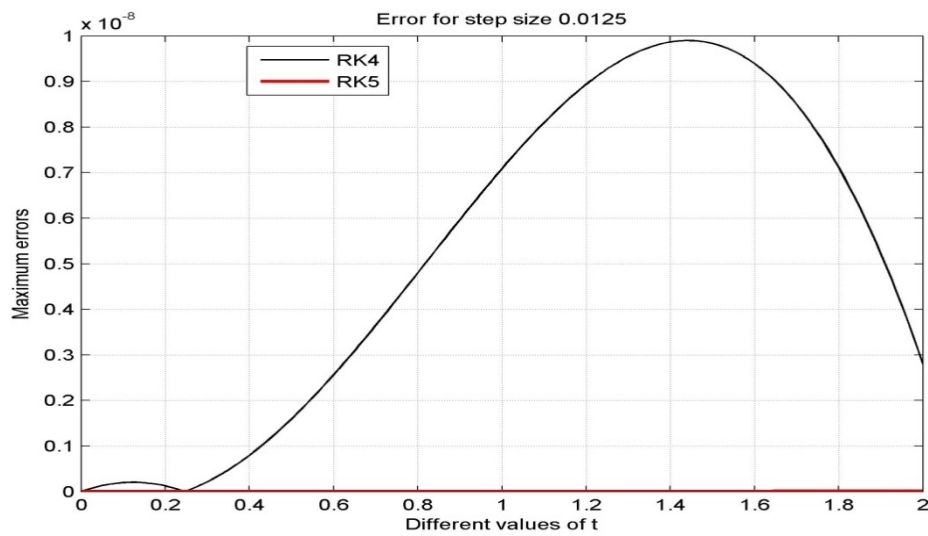


Figure 6. Error for step size $h=0.0125$

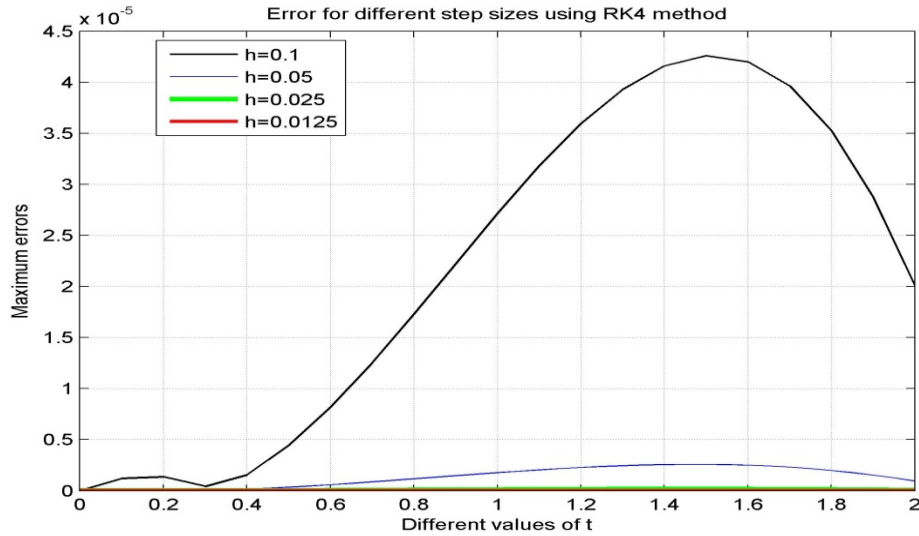


Figure 7. Error for different step sizes using RK4 method

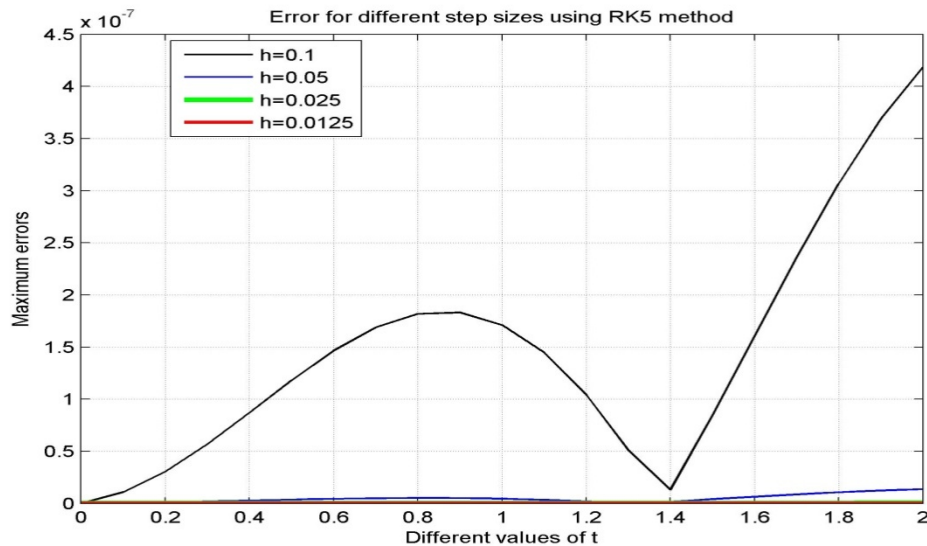


Figure 8. Error for different step sizes using RK5 method

5. Discussion

The acquired results are displayed in Tables (1-4) and graphically presented in Figures (1-8). The approximate solutions and errors are calculated with the step sizes 0.1, 0.05, 0.025 and 0.0125 by using MATLAB. The approximate solutions are compared to the exact solutions. The Runge-Kutta fifth order (RK5) approximations for same step size converge to exact solution. It is also mention that the small step size provides the better approximation. We observe that from Tables (1-4) Runge-Kutta fifth order (RK5) method gives more accurate results and better than Runge-Kutta fourth order (RK4) method. Also from figures 7 and 8 we conclude that if the step size tends to zero then the errors also tends to zero.

6. Conclusions

In this paper, fourth order Runge-Kutta method and Butcher's fifth order Runge-Kutta method are applied to solve second order initial value problems (IVP) of ordinary differential equation (ODE). To find more accurate results we need to reduce the step size for both methods. From the resulting tables and figures we analyzed that both methods converge to the exact solutions for very small values of the step size, h . From the figure 2 we also observe that both methods give almost same results but from Figures 3-6 it is clear that RK5 method gives more accurate results than RK4 method. Also we state that the Butcher's fifth order Runge-Kutta method is more appropriate and proficient for finding the numerical solutions of initial value problems

(IVP) than fourth order Runge-Kutta method. Hence from this study we conclude that to find more accurate result higher order numerical method is appropriate than lower order methods.

REFERENCES

- [1] Rabiei, F., & Ismail, F. (2012). Fifth-order Improved Runge-Kutta method for solving ordinary differential equation. *Australian Journal of Basic and Applied Sciences*, 6(3), 97-105.
- [2] Butcher, J. C. (1995). On fifth order Runge-Kutta methods. *BIT Numerical Mathematics*, 35(2), 202-209.
- [3] Islam, M. A. (2015). A Comparative Study on Numerical Solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE) with Euler and RungeKutta Methods. *American Journal of Computational Mathematics*, 5(03), 393.
- [4] Butcher, J. C. (1964). On Runge-Kutta processes of high order. *Journal of the Australian Mathematical Society*, 4(02), 179-194.
- [5] Islam, M. A. (2015). Accuracy Analysis of Numerical solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE). *IOSR Journal of Mathematics*, 11, 18-23.
- [6] Butcher, J. C. (1996). A history of Runge-Kutta methods. *Applied numerical mathematics*, 20(3), 247-260.
- [7] Goeken, D., & Johnson, O. (2000). Runge-Kutta with higher order derivative approximations. *Applied numerical mathematics*, 34(2-3), 207-218.
- [8] Lambert, J. D. (1973). Computational methods in ordinary differential equations. Wiley, New York.
- [9] Mathews, J.H. (2005). Numerical Methods for Mathematics, Science and Engineering. Prentice-Hall, India.
- [10] Hall, G. and Watt, J.M. (1976). Modern Numerical Methods for Ordinary Differential Equations. Oxford University Press, Oxford.
- [11] Burden, R.L. and Faires, J.D. (2002). Numerical Analysis. Bangalore, India.
- [12] Gerald, C.F. and Wheatley, P.O. (2002). Applied Numerical Analysis. Pearson Education, India.
- [13] Hossain, Md.S., Bhattacharjee, P.K. and Hossain, Md.E. (2013). Numerical Analysis. Titus Publications, Dhaka.
- [14] Sastry, S.S. (2000). Introductory Methods of Numerical Analysis. Prentice-Hall, India.
- [15] Balagurusamy, E. (2006). Numerical Methods. Tata McGraw-Hill, New Delhi.