International Annals of Science
ISSN: 2456-7132
Volume 10, Issue 1, pp. 118-133, 2021
DOI: https://doi.org/10.21467/ias.10.1.118-133

RESEARCH ARTICLE

Accuracy Analysis on Solution of Initial Value Problems of Ordinary Differential Equations for Some Numerical Methods with Different Step Sizes

Mohammad Asif Arefin[1*], Biswajit Gain[1], Rezaul Karim[2]

[1] Department of Mathematics, Jashore University of Science and Technology, Bangladesh
[2] Department of Mathematics, Mawlana Bhashani Science and Technology University, Bangladesh
[*] Corresponding author email: asif.math@just.edu.bd

Received: 04 December 2020 / Revised: 06 February 2021 / Accepted: 26 February 2021 / Published: 07 March 2021

ABSTRACT

In this article, three numerical methods namely Euler's, Modified Euler, and Runge-Kutta method have been discussed, to solve the initial value problem of ordinary differential equations. The main goal of this research paper is to find out the accurate results of the initial value problem (IVP) of ordinary differential equations (ODE) by applying the proposed methods. To achieve this goal, solutions of some IVPs of ODEs have been done with the different step sizes by using the proposed three methods, and solutions for each step size are analyzed very sharply. To ensure the accuracy of the proposed methods and to determine the accurate results, numerical solutions are compared with the exact solutions. It is observed that numerical solutions are best fitted with exact solutions when the taken step size is very much small. Consequently, all the proposed three methods are quite efficient and accurate for solving the IVPs of ODEs. Error estimation plays a significant role in the establishment of a comparison among the proposed three methods. On the subject of accuracy and efficiency, comparison is successfully implemented among the proposed three methods.

Keywords: Euler's method; Modified Euler method; Fourth-order Runge-Kutta method.

1 Introduction

Differential Equations play a significant role in solving complex mathematical problems in almost every section of Science and Engineering. In mathematics, many real problems emerge in the form of differential equations, either in the form of ordinary differential equations or partial differential equations. Numerical approximation methods are frequently used to solve mathematical problems where as it is so much difficult or even impossible to find out exact results. Though there are many analytical methods for solving ODEs, a large number of ODEs cannot be solved by using analytical methods. In those cases, we must have to use numerical approximate methods for solving the IVPs of ODEs. There are several numerical methods to solve the IVPs, where the exactness and effectiveness of all numerical methods are not the same. In day-to-day life, it is very important to know the process before attempting a problem in which the problem can be solved easily with less error and less computational time. From the literature review analysis, we can understand that many authors have worked on numerical solutions of the ODEs using numerous numerical methods namely Euler's method, Modified Euler method, and RK-4 method, etc. Many authors have taken initiative, to maintain the high correctness for the solution of IVPs of ODEs. In [1], the author represents the accuracy analysis of IVPs for ODEs using the Euler method, and also in[2] , the author tries to find out accurate solutions by using the RK-4 method of IVPs of ODEs. Various numerical methods are discussed and provided an order of accuracy in [3], for solving IVPs. A comparison between fourth-order and butcher's fifth-order RK methods is discussed in [4] and an extension of the RK method is discussed in[5]. Improving the modified Euler method, embedded modified Euler method, modified Euler method for dynamic analyses, modified Euler method for finding numerical solution of the intuitionistic fuzzy DEs, enhanced Euler's method, the accuracy of Euler and modified Euler technique all of these topics are elaborately analyzed in [6]–[11]. Authors try to established comparison [12], [13] among various numerical methods for solving the IVPs of ODEs. In [14] authors discussed Multistep Iterative Methods for Solving ODEs. The author explains the application of Matlab in [15].

The rest of the part of the article is arranged as: In segment 2, we have presented the formulation of the problem. in segment 3, some numerical examples are solved using the proposed method. In segment 4, we have discussed our finding results elaborately. In the lattermost part, the conclusion is given.

2 Formulation of the Problem

For obtaining the approximate solutions of the IVPs of ODEs, we consider three numerical methods which consisting of the form

$$y' = f\big(x, y(x)\big), \quad x\epsilon(x_0, x_n), \quad y(x_0) = y_0 \qquad (1)$$

Here, $y' = \frac{dy}{dx}$ and $f\big(x, y(x)\big)$ is a given function whereas the solution of the equation (1) is $y(x)$. A continuous approximation to the solution $y(x)$ may not be found; instead, approximation result to y will be produced at different values, with in the interval (x_0, x_n). Numerical methods engage the equation (1) to obtain the approximate solutions to the values of the solution corresponding to various selected vales of $x = x_n = x_0 + nh$, $n = 1,2\,,3$ .... and "h" denotes the step size. Solution of equation (1) is introduced by a set of points $\{(x_n, y_n): n = 0,1,2,3, \ldots, n\}$ and each point (x_n, y_n) is an approximation to the corresponding point $(x_n, y(x_n))$ on the curve of the solution.

2.1 Euler's Method

In 1768, Euler introduced this method for solving the IVPs of the ordinary differential equation. It is the least difficult one-venture strategy. Euler Method is the most elementary approximation procedure for obtaining the result of the initial value problem. Error analysis can be easily understood by studying this method. Euler Approximation is generally denoted by:

$$y_{n+1}(x) = y_n(x) + hf(x_{n,}y_n)\ , \text{ where } n = 0,1,2,...$$

2.2 Modified Euler Method

In this method, the curve in the interval (x_0, x_1) where $x_1 = x_0 + h$ is approximated by the line through (x_0, y_0) with the slope $f(x_0 + \frac{h}{2}, y_0 + \frac{h}{2} f(x_0, y_0))$ which is the slope as the middle point whose abscissa is average of x_0 and x_1. Generalized Modified Euler method is

$$y_{n+1}(x) = y_n(x) + h\, f(x_n + \frac{h}{2}, \qquad y_n + \frac{h}{2}\, f(x_n, y_n))$$

2.3 Runge-Kutta Method

In 1894, the Runge-Kutta Method was first introduced by Runge and then extended by Kutta. Both of them were German mathematicians. RK method is the most well-known method because it is stable, the accuracy level is high and so much easy to codding. In the Runge-Kutta method, it is not required to find the derivatives of the superior order and the results found in the RK-4 method converge closer to the analytical solution. The general solution for the RK-4 method is

$$y_{n+1}(x) = y_n(x) + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad n = 0,1,2, \ldots.$$

Where,

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

3 Numerical Example Solved by the Proposed Methods

In this section, two numerical problems are solved by the proposed three numerical methods with different step sizes. Numerical results and errors are computed along with the exact solution using Matlab programming language. Here tables represented the result analysis and figures represent the outcomes, graphically.

Example 1: Taking the IVP $y' = xe^{3x} - 2y$, $y(0) = 0$ with the interval $0 \leq x \leq 1$. Where exact solution of this problem is $y(x) = \frac{1}{5}xe^{3x} - \frac{1}{25}e^{3x} + \frac{1}{25}e^{-2x}$. [16]

Approximate solutions and errors are represented in Tables: 1(a)-(d) and the curves of the numerical solutions are displayed in Figures: 1(a)-(e) and the error estimation are shown in figure: 2-7

Table: 1(a), 1(b), 1(c), 1(d) represents the numerical approximations and errors of step sizes $0.1, 0.05, 0.025, 0.0125$ respectively.

Table 1(a): *Numerical approximation and Error analysis for step size 0.1*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000
0.1	0.000000000000	5.75E-03	0.006749294038	9.97E-04	0.005754631312	2.58E-06	0.005752053972
0.2	0.013498588076	1.33E-02	0.029155044345	2.34E-03	0.026818770597	5.97E-06	0.026812801841
0.3	0.047241246468	2.39E-02	0.075378133434	4.23E-03	0.071155164515	1.06E-05	0.071144527667
0.4	0.111581090509	3.92E-02	0.157727645204	6.95E-03	0.150795060680	1.72E-05	0.150777835474
0.5	0.222069549317	6.15E-02	0.294500766590	1.09E-02	0.283643159044	2.66E-05	0.283616521867
0.6	0.401740092971	9.43E-02	0.512613833943	1.66E-02	0.496059711487	4.01E-05	0.496019565630
0.7	0.684370922241	1.42E-01	0.851350829919	2.49E-02	0.826540417642	5.95E-05	0.826480869814
0.8	1.119128631673	2.12E-01	1.367687493311	3.68E-02	1.330944404474	8.74E-05	1.330857026397
0.9	1.777157015789	3.13E-01	2.143833316315	5.41E-02	2.089901607342	1.27E-04	2.089774397011
1.0	2.760901467870	4.58E-01	3.297890507633	7.88E-02	3.219283395463	1.84E-04	3.219099319039

Table 1(b): *Numerical approximation and Error analysis for step size 0.05*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution (y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000
0.1	0.002904585607	2.85E-03	0.005996035529	2.44E-04	0.005752215632	1.62E-07	0.005752053972
0.2	0.020189420367	6.62E-03	0.027385660104	5.73E-04	0.026813176258	3.74E-07	0.026812801841
0.3	0.059214999908	1.19E-02	0.072179548852	1.04E-03	0.071145194819	6.67E-07	0.071144527667
0.4	0.131177686492	1.96E-02	0.152476152077	1.70E-03	0.150778915532	1.08E-06	0.150777835474
0.5	0.252808105109	3.08E-02	0.286275093353	2.66E-03	0.283618191417	1.67E-06	0.283616521867
0.6	0.448804514468	4.72E-02	0.500071097732	4.05E-03	0.496022080755	2.52E-06	0.496019565630
0.7	0.755304484628	7.12E-02	0.832550464785	6.07E-03	0.826484598769	3.73E-06	0.826480869814
0.8	1.224821078658	1.06E-01	1.339842276136	8.99E-03	1.330862495679	5.47E-06	1.330857026397
0.9	1.933241334179	1.57E-01	2.102958301727	1.32E-02	2.089782356165	7.96E-06	2.089774397011
1.0	2.989724252969	2.29E-01	3.238309127452	1.92E-02	3.219110831606	1.15E-05	3.219099319039

Mohammad Asif Arefin *et al.*, *Int. Ann. Sci.;* Vol. 10, Issue 1, pp: 118-133, 2021

Table 1(c): *Numerical approximation and Error analysis for step size 0.025*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000
0.1	0.004335777285	1.42E-03	0.005812269509	6.02E-05	0.005752064070	1.01E-08	0.005752053972
0.2	0.023511255782	3.30E-03	0.026954180136	1.41E-04	0.026812825231	2.34E-08	0.026812801841
0.3	0.065189414327	5.96E-03	0.071399940965	2.55E-04	0.071144569342	4.17E-08	0.071144527667
0.4	0.140984971975	9.79E-03	0.151196875876	4.19E-04	0.150777902934	6.75E-08	0.150777835474
0.5	0.268216182029	1.54E-02	0.284272396357	6.56E-04	0.283616626128	1.04E-07	0.283616521867
0.6	0.472412666565	2.36E-02	0.497018932082	9.99E-04	0.496019722666	1.57E-07	0.496019565630
0.7	0.790891292011	3.56E-02	0.827977794459	1.50E-03	0.826481102592	2.33E-07	0.826480869814
0.8	1.277838339234	5.30E-02	1.333072715991	2.22E-03	1.330857367747	3.41E-07	1.330857026397
0.9	2.011512537525	7.83E-02	2.093025016734	3.25E-03	2.089774893669	4.97E-07	2.089774397011
1.0	3.104429503703	1.15E-01	3.223835121649	4.74E-03	3.219100037310	7.18E-07	3.219099319039

Table 1(d): *Numerical approximation and Error analysis for step size 0.0125*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000	0.00E+00	0.000000000000
0.1	0.005045831795	7.06E-04	0.005767003879	1.49E-05	0.005752054602	6.31E-10	0.005752053972
0.2	0.025164790521	1.65E-03	0.026847902665	3.51E-05	0.026812803302	1.46E-09	0.026812801841
0.3	0.068169901843	2.97E-03	0.071207939211	6.34E-05	0.071144530269	2.60E-09	0.071144527667
0.4	0.145884152643	4.89E-03	0.150881866166	1.04E-04	0.150777839687	4.21E-09	0.150777835474
0..5	0.275918865492	7.70E-03	0.283779339607	1.63E-04	0.283616528377	6.51E-09	0.283616521867
0.6	0.484218652898	1.18E-02	0.496267638266	2.48E-04	0.496019575434	9.80E-09	0.283616521867
0.7	0.808689273993	1.78E-02	0.826852428092	3.72E-04	0.826480884346	1.45E-08	0.826480869814
0.8	1.304352661078	2.65E-02	1.331406959518	5.50E-04	1.330857047705	2.13E-08	1.330857026397
0.9	2.050652054747	3.91E-02	2.090581154110	8.07E-04	2.089774428011	3.10E-08	2.089774397011
1.0	3.161779432515	5.73E-02	3.220274616732	1.18E-03	3.219099363868	4.48E-08	3.219099319039

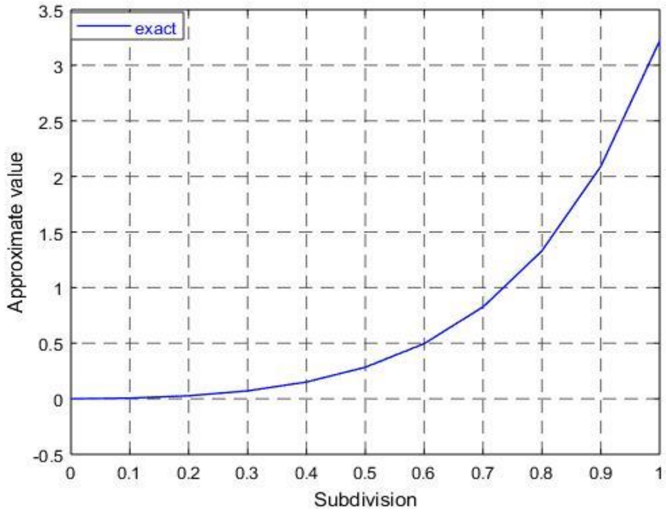

Figure 1(a). *Exact solutions curve*

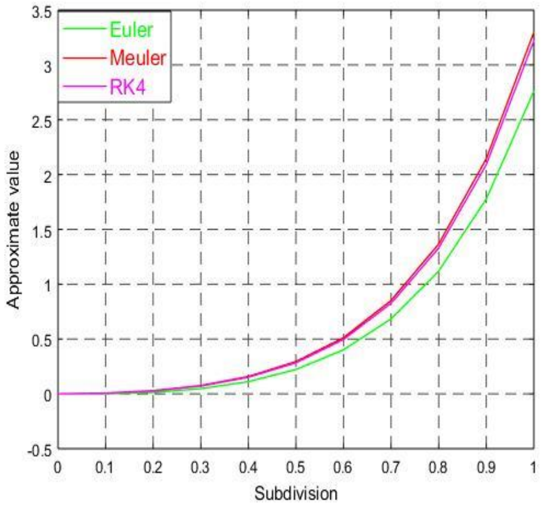

Figure 1(b). *Approximate solution curve for step size* 0.1

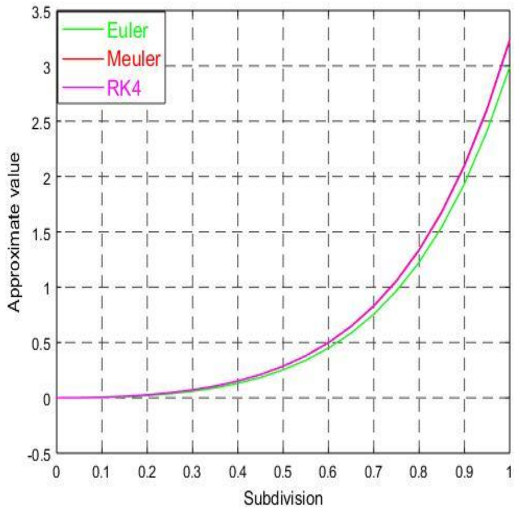

Figure 1(c). *Approximate solution curve for step size* 0.05

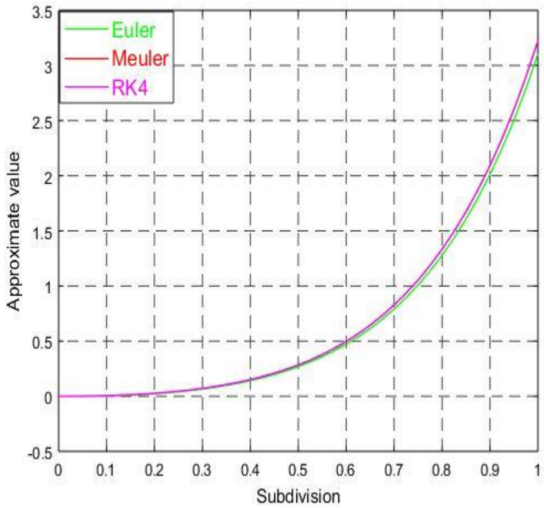

Figure 1(d). *Approximate solution curve for step size* 0.025

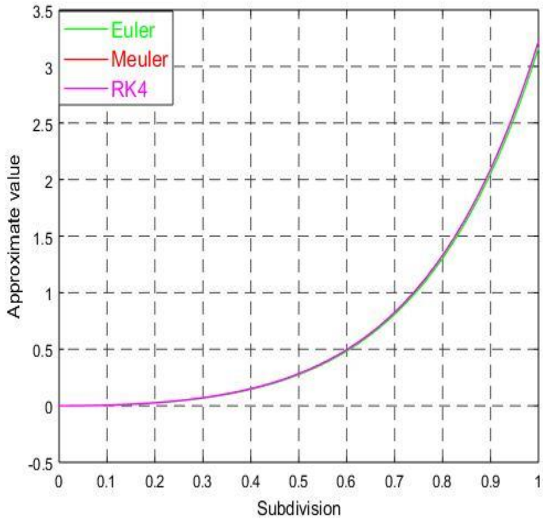

Figure 1(e). *Approximate solution curve for step size* 0.0125

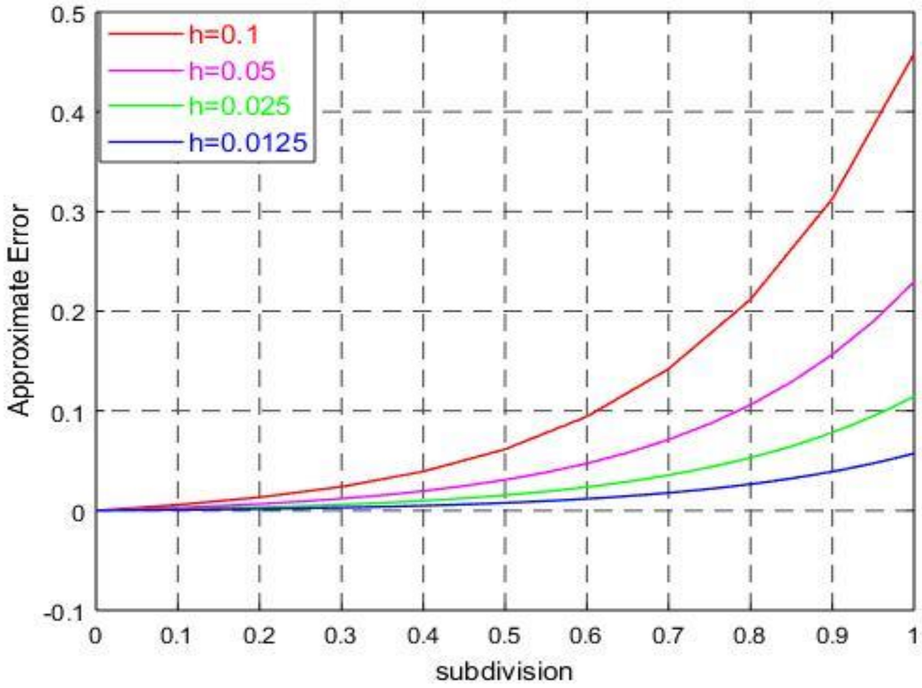

Figure 2. *Error estimation for different step sizes obtained by Euler's method using Matlab*

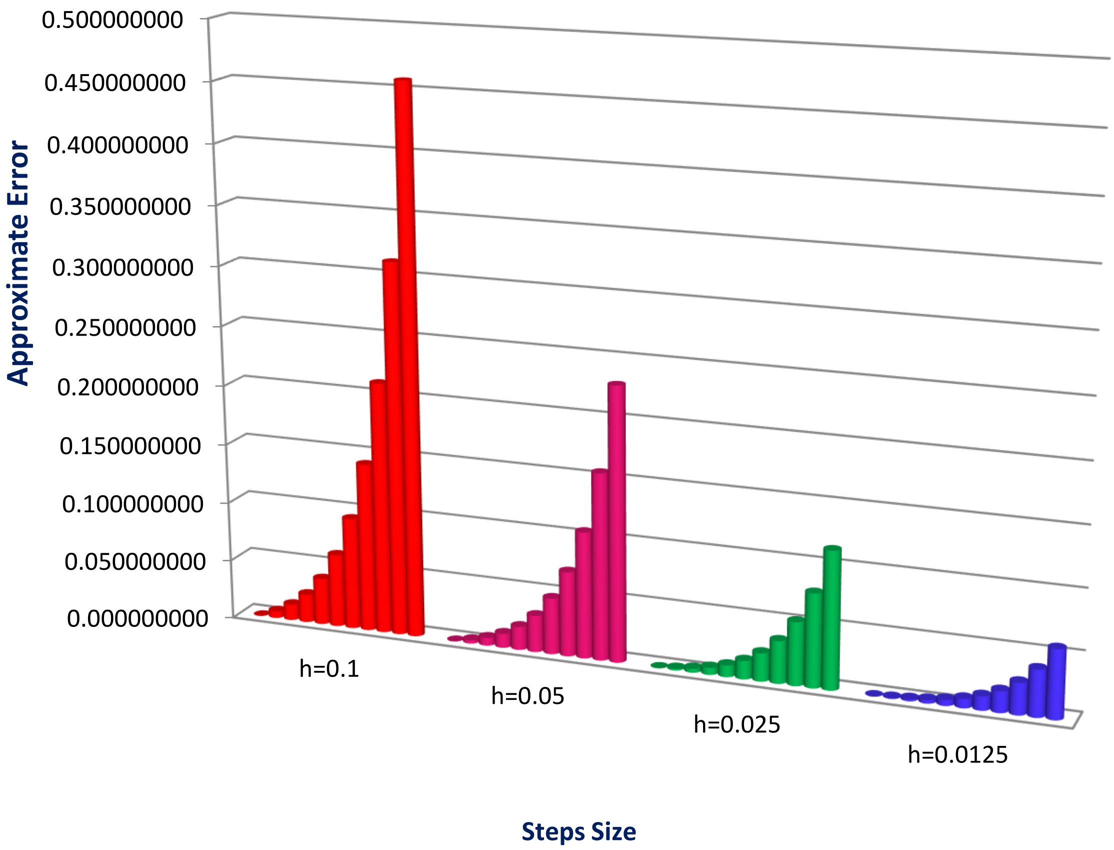

Figure 3. *Error estimation diagram obtained by Euler's method using MS Excel for different step sizes*

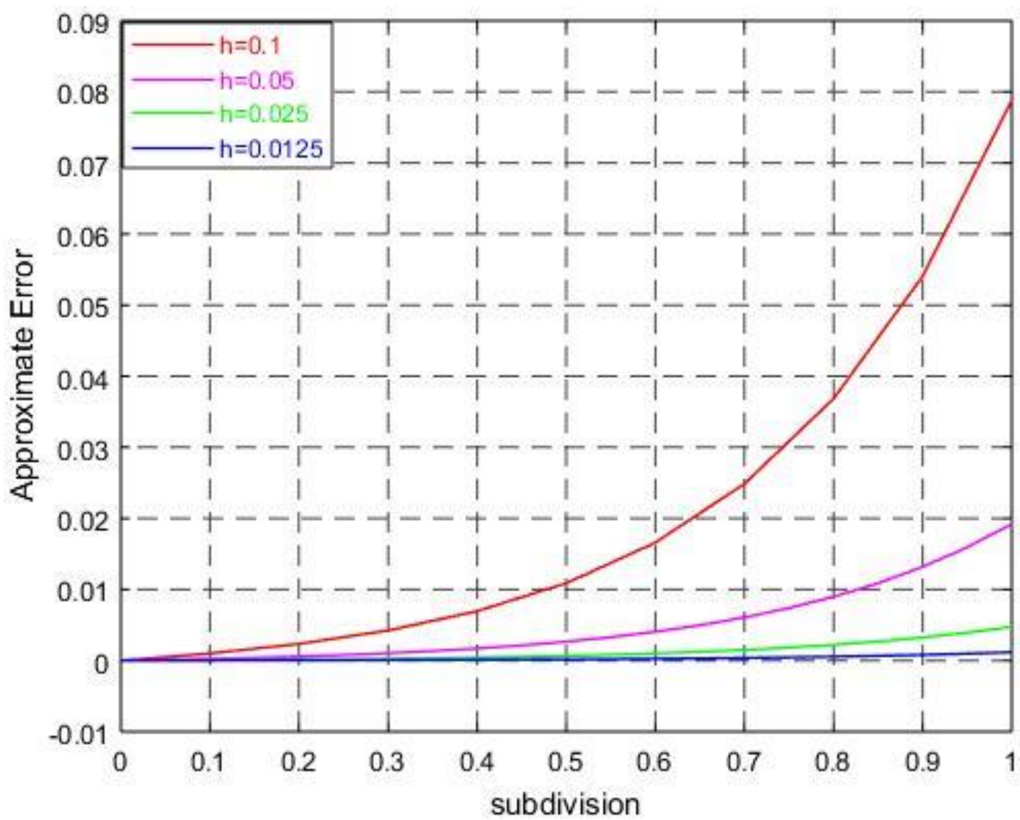

Figure 4. *Error estimation for different step sizes obtained by Modified Euler method using Matlab*

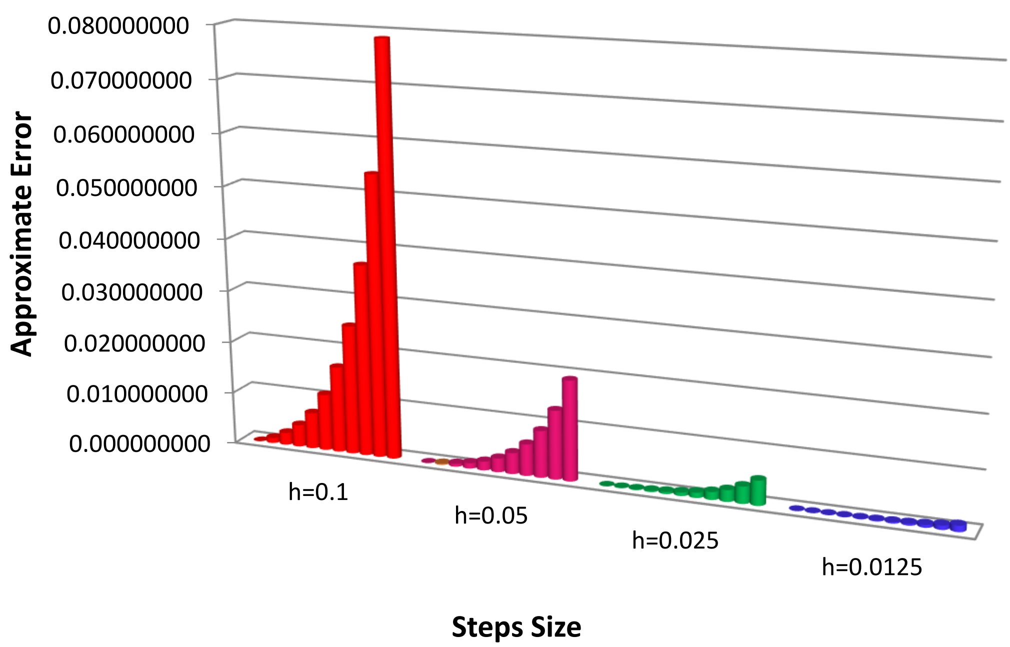

Figure 5. *Error estimation diagram obtained by Modified Euler method using MS Excel for different step size*

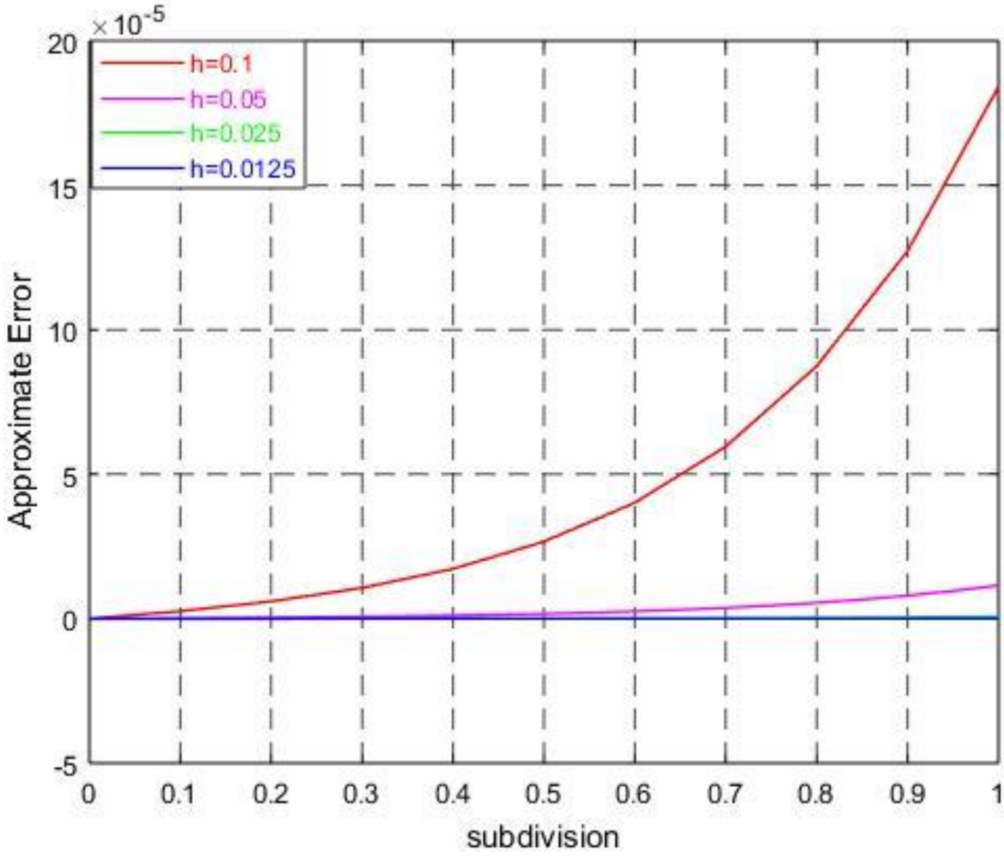

Figure 6. *Error estimation for different step size obtained by Runge-Kutta method using Matlab*

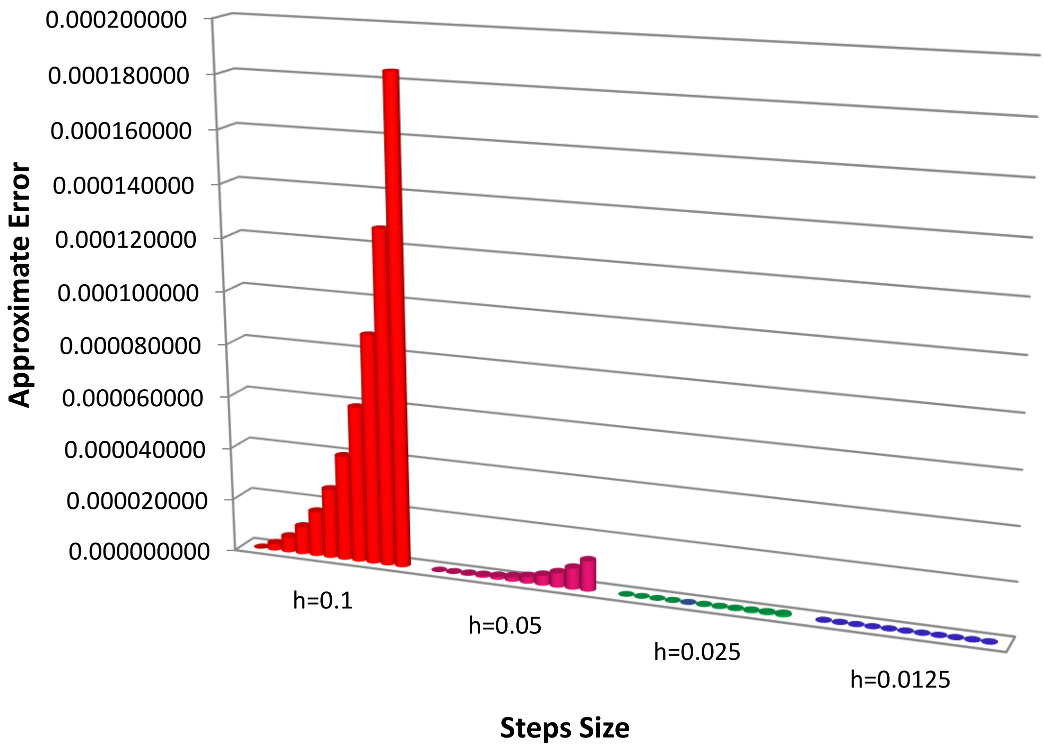

Figure 7. *Error estimation diagram obtained by Runge-Kutta method using MS Excel for different step size*

Example 2: Taking the IVP $y' = \cos 2x + \sin 3x$, $y(0) = 1$ with the interval $0 \le x \le 1$. Where exact solution of this problem is $y(x) = \frac{1}{2}sin2x - \frac{1}{3}cos3x + \frac{4}{3}$. [16]

Approximate solutions and errors are represented in Tables: 2(a)-(d) and the curves of the numerical solutions are shown in Figures: 8(a)-(e) and the error estimation are shown in Figures: 9-14. Table: 2(a), 2(b), 2(c), 2(d) represents numerical approximations and errors of step sizes 0.1, 0.05, 0.025, 0.0125 respectively.

Table 2(a): *Numerical approximation and Error analysis for step size 0.1*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000
0.1	1.100000000000	1.42E-02	1.113779339225	4.43E-04	1.114222599592	9.72E-08	1.114222502356
0.2	1.227558678450	2.54E-02	1.251843851820	1.09E-03	1.252930905339	2.72E-07	1.252930632851
0.3	1.376129025190	3.23E-02	1.406562151417	1.89E-03	1.408451759999	5.13E-07	1.408451247274
0.4	1.536995277644	3.42E-02	1.568432567409	2.79E-03	1.571226259522	7.99E-07	1.571225460624
0.5	1.699869857175	3.06E-02	1.726759721799	3.73E-03	1.730490866058	1.11E-06	1.730489758515
0.6	1.853649586422	2.14E-02	1.870459855690	4.63E-03	1.875088320681	1.41E-06	1.875086907881
0.7	1.987270124958	7.07E-03	1.988928950435	5.41E-03	1.994341921838	1.69E-06	1.994340233194
0.8	2.090587775913	1.17E-02	2.072900958825	6.02E-03	2.078919951205	1.91E-06	2.078918040035
0.9	2.155214141738	3.36E-02	2.115223031015	6.39E-03	2.121616590141	2.06E-06	2.121614529445
1.0	2.175231920292	5.73E-02	2.111480578868	6.50E-03	2.117981669124	2.12E-06	2.117979545613

Table 2(b): *Numerical approximation and Error analysis for step size 0.05*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000
0.1	1.107222114888	7.00E-03	1.114111784500	1.11E-04	1.114222508425	6.07E-09	1.114222502356
0.2	1.240516555275	1.24E-02	1.252659141960	2.71E-04	1.252930649855	1.70E-08	1.252930632851
0.3	1.392762794740	1.57E-02	1.407979357854	4.72E-04	1.408451279266	3.20E-08	1.408451247274
0.4	1.554809191611	1.64E-02	1.570527836494	6.98E-04	1.571225510469	4.98E-08	1.571225460624
0.5	1.716113147682	1.44E-02	1.729558079993	9.32E-04	1.730489827613	6.91E-08	1.730489758515
0.6	1.865526069799	9.56E-03	1.873931204433	1.16E-03	1.875086996022	8.81E-08	1.875086907881
0.7	1.992159266248	2.18E-03	1.992988678987	1.35E-03	1.994340338541	1.05E-07	1.994340233194
0.8	2.086258611653	7.34E-03	2.077415203110	1.50E-03	2.078918159262	1.19E-07	2.078918040035
0.9	2.140013755721	1.84E-02	2.120018200359	1.60E-03	2.121614657998	1.29E-07	2.121614529445
1.0	2.148232067272	3.03E-02	2.116356396560	1.62E-03	2.117979678083	1.32E-07	2.117979545613

Mohammad Asif Arefin *et al.*, *Int. Ann. Sci.;* Vol. 10, Issue 1, pp: 118-133, 2021

Table 2(c): *Numerical approximation and Error analysis for step size 0.025*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000
0.1	1.110749992637	3.47E-03	1.114194827443	2.77E-05	1.114222502735	3.79E-10	1.114222502356
0.2	1.246791479539	6.14E-03	1.252862772881	6.79E-05	1.252930633913	1.06E-09	1.252930632851
0.3	1.400725017356	7.73E-03	1.408333298913	1.18E-04	1.408451249273	2.00E-09	1.408451247274
0.4	1.563191769534	8.03E-03	1.571051091975	1.74E-04	1.571225463738	3.11E-09	1.571225460624
0.5	1.723534424552	6.96E-03	1.730256890708	2.33E-04	1.730489762831	4.32E-09	1.730489758515
0.6	1.870595480808	4.49E-03	1.874798048125	2.89E-04	1.875086913388	5.51E-09	1.875086907881
0.7	1.993587717283	7.53E-04	1.994002423653	3.38E-04	1.994340239775	6.58E-09	1.994340233194
0.8	2.082964124496	4.05E-03	2.078542420224	3.76E-04	2.078918047483	7.45E-09	2.078918040035
0.9	2.131213321269	9.60E-03	2.121215543589	3.99E-04	2.121614537476	8.03E-09	2.121614529445
1.0	2.133511693059	1.55E-02	2.117573857702	4.06E-04	2.117979553889	8.28E-09	2.117979545613

Table 2(d): *Numerical approximation and Error analysis for step size 0.0125*

x_n	Euler's method		Modified Euler method		Runge-Kutta method		Exact Solution(y_n)
	$y(x_n)$	Error	$y(x_n)$	Error	$y(x_n)$	Error	
0.0	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000	0.00E+00	1.000000000000
0.1	1.112493166509	1.73E-03	1.114215583912	6.92E-06	1.114222502379	2.40E-11	1.114222502356
0.2	1.249878021984	3.05E-03	1.252913668655	1.70E-05	1.252930632917	6.60E-11	1.252930632851
0.3	1.404617620904	3.83E-03	1.408421761683	2.95E-05	1.408451247399	1.25E-10	1.408451247274
0.4	1.567252209577	3.97E-03	1.571181870797	4.36E-05	1.571225460819	1.95E-10	1.571225460624
0.5	1.727070311723	3.42E-03	1.730431544801	5.82E-05	1.730489758784	2.70E-10	1.730489758515
0.6	1.872913413413	2.17E-03	1.875014697072	7.22E-05	1.875086908225	3.44E-10	1.875086907881
0.7	1.994048432560	2.92E-04	1.994255785745	8.44E-05	1.994340233605	4.11E-10	1.994340233194
0.8	2.081034992804	2.12E-03	2.078824140668	9.39E-05	2.078918040500	4.65E-10	2.078918040035
0.9	2.126513677844	4.90E-03	2.121514789004	9.97E-05	2.121614529947	5.02E-10	2.121614529445
1.0	2.125847047520	7.87E-03	2.117878129842	1.01E-04	2.117979546130	5.17E-10	2.117979545613

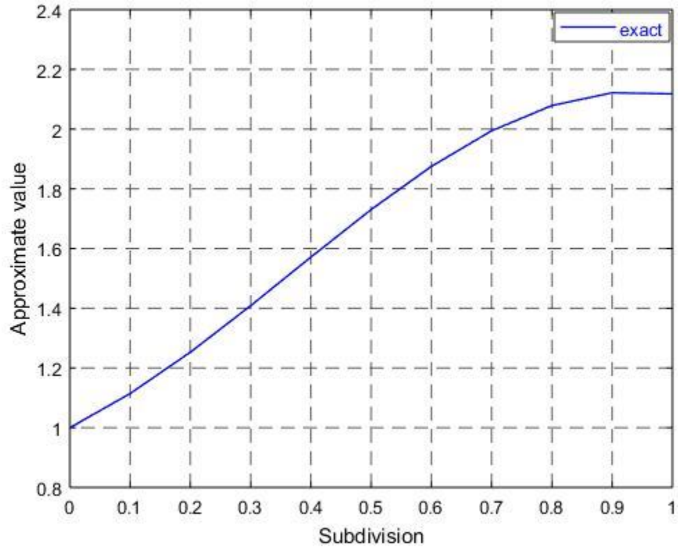

Figure 8(a). *Exact solutions curve*

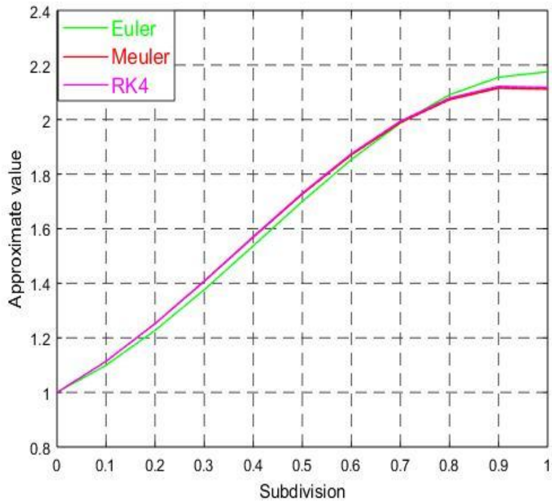

Figure 8(b). *Approximate solution curve for step size* 0.1

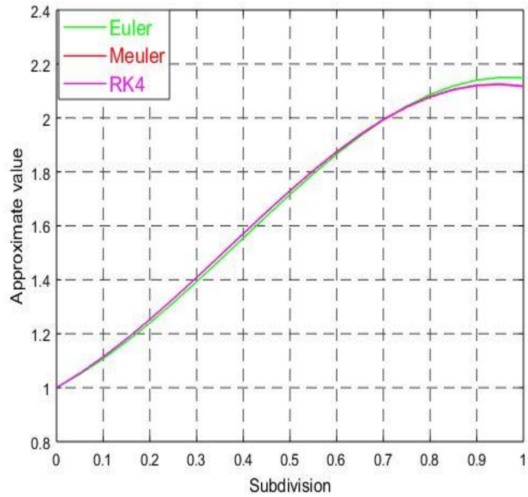

Figure 8(c). *Approximate solution curve for step size* 0.05

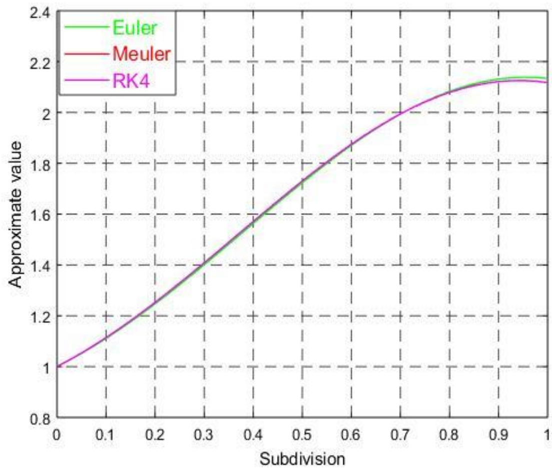

Figure 8(d). *Approximate solution curve for step size* 0.025

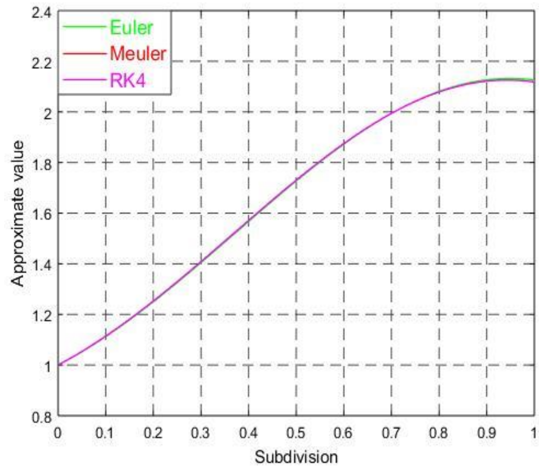

Figure 8(e). *Approximate solution curve for step size* 0.0125

Mohammad Asif Arefin *et al.*, *Int. Ann. Sci.;* Vol. 10, Issue 1, pp: 118-133, 2021

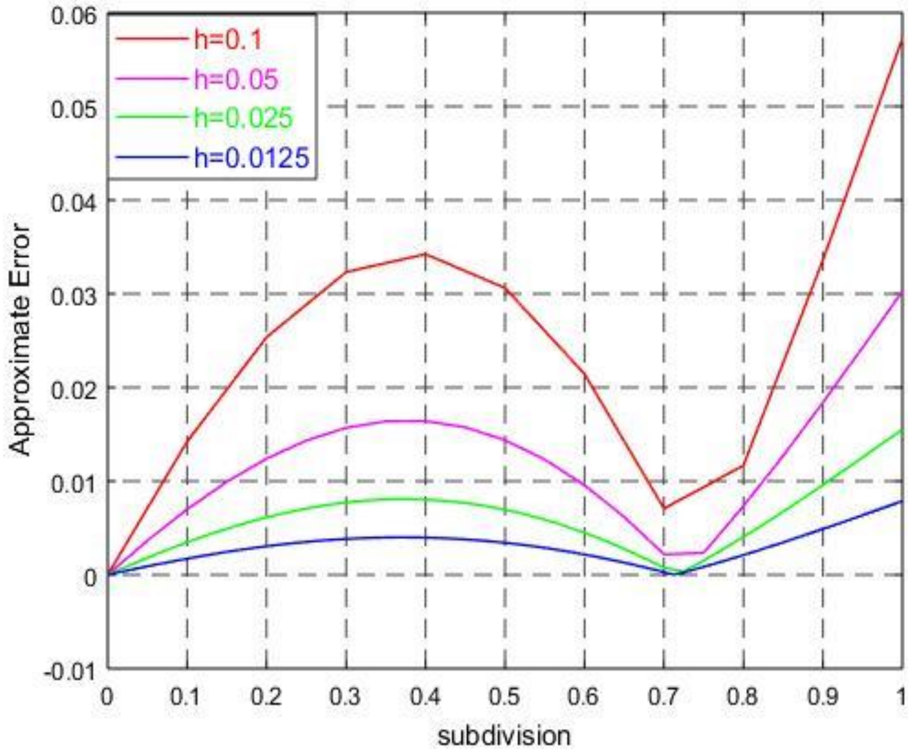

Figure 9. *Error estimation for different step sizes obtained by Euler's method using Matlab*

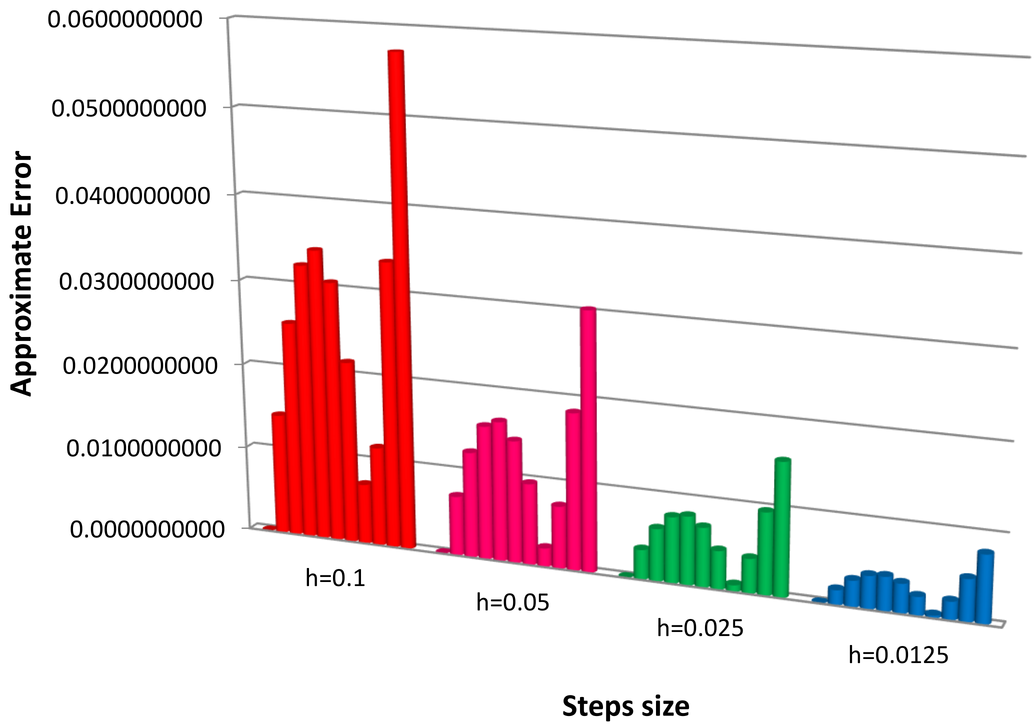

Figure 10. *Error estimation diagram obtained by Euler's method using MS Excel for different step sizes*

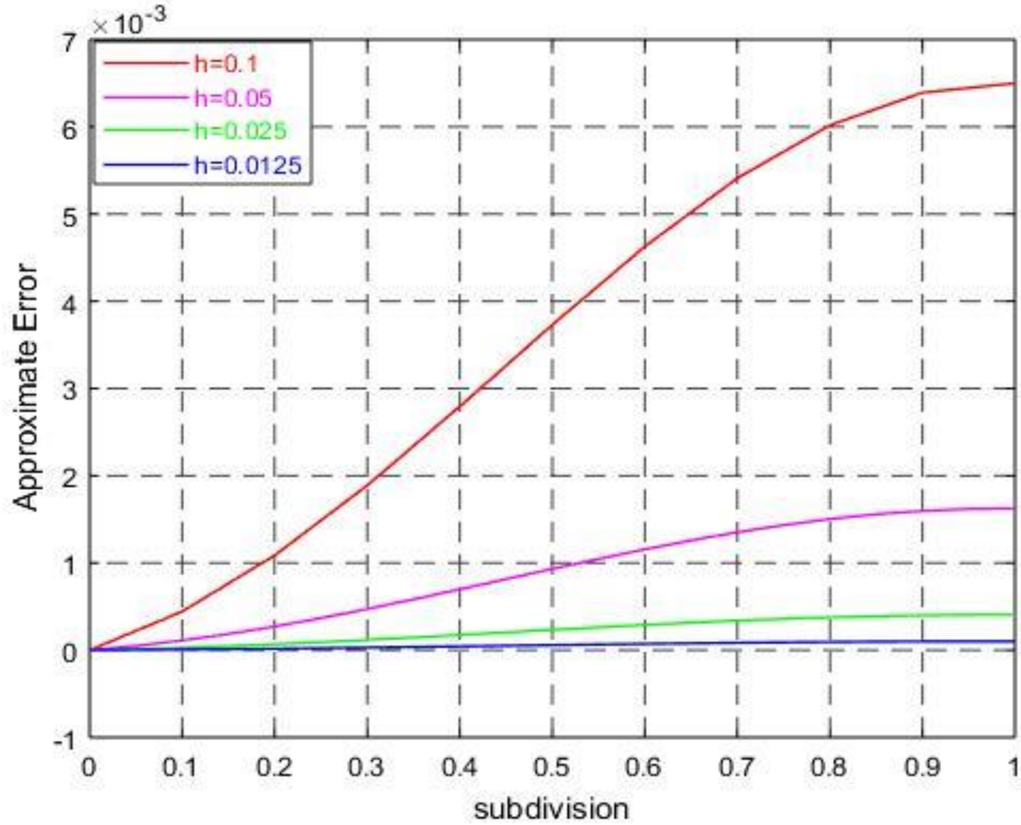

Figure 11. *Error estimation for different step sizes obtained by Modified Euler method using Matlab*

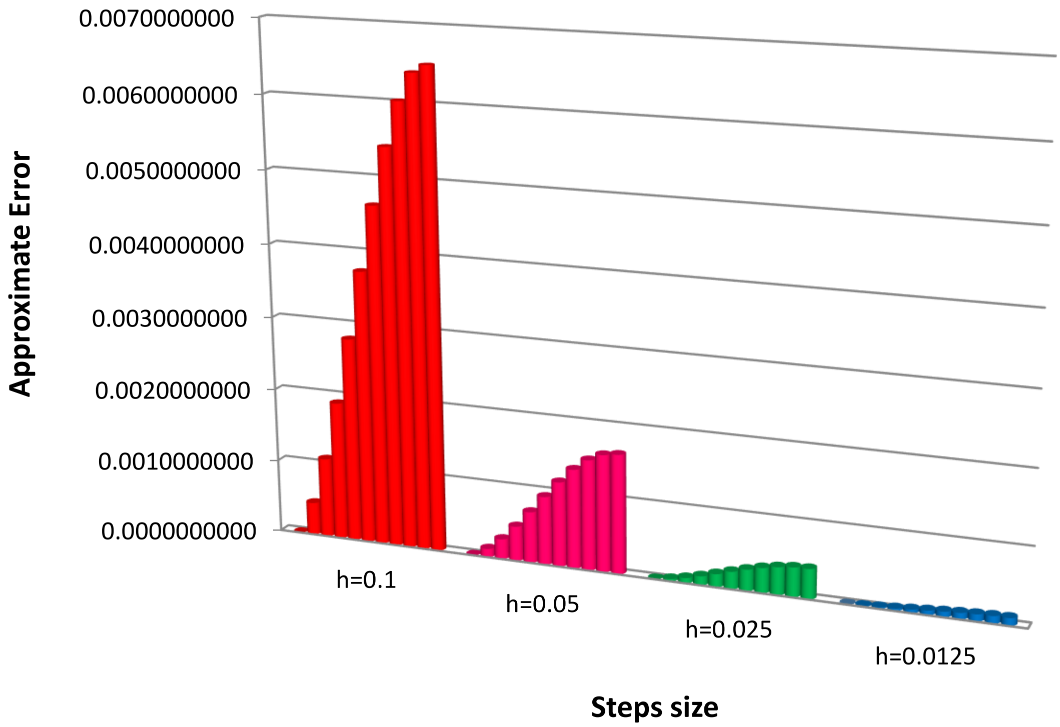

Figure 12. *Error estimation diagram obtained by Modified Euler method using MS Excel for different step sizes*

Mohammad Asif Arefin *et al.*, *Int. Ann. Sci.;* Vol. 10, Issue 1, pp: 118-133, 2021

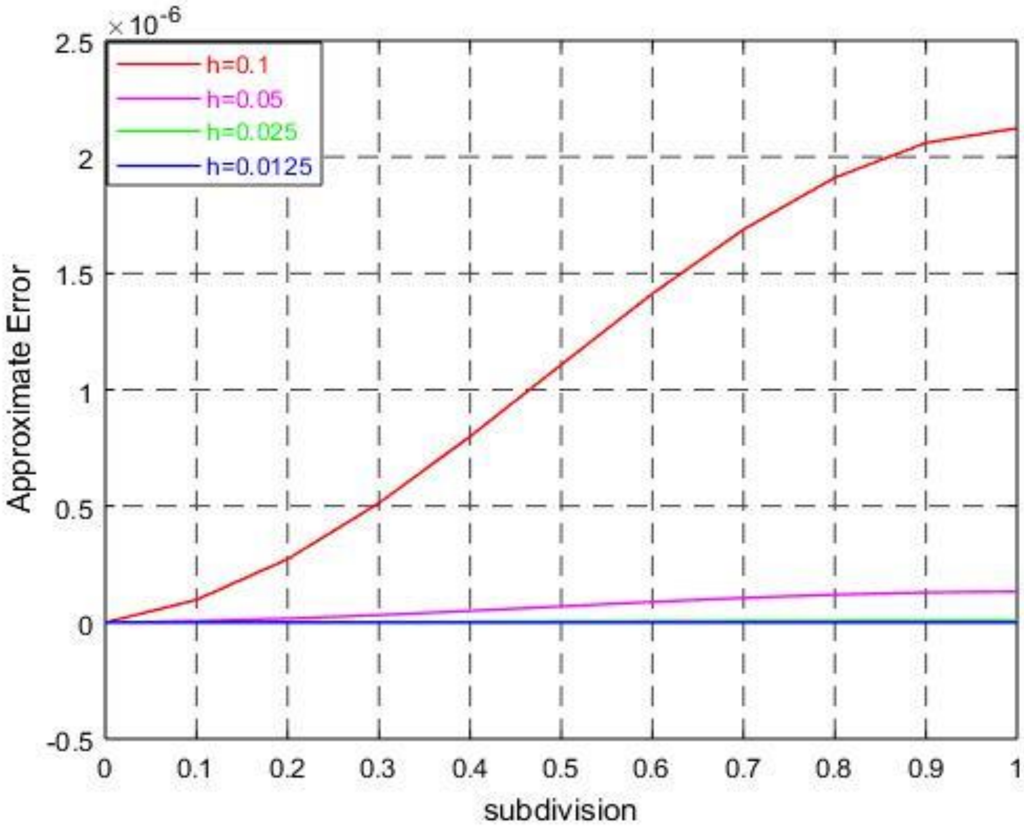

Figure 13. *Error estimation for different step sizes obtained by Runge-Kutta method using Matlab*

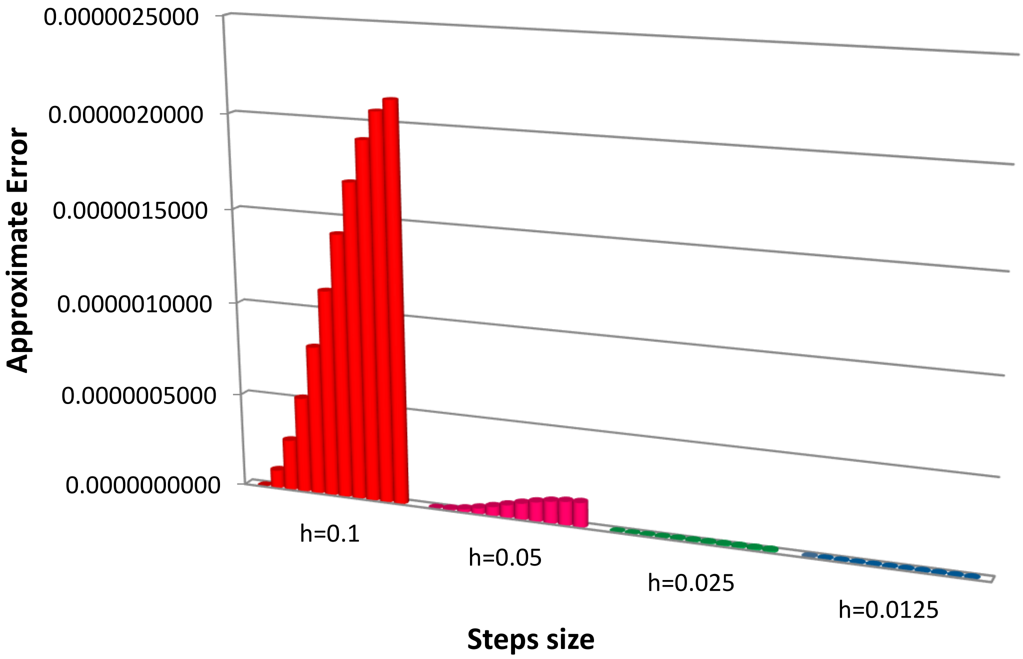

Figure 14. *Error estimation diagram obtained by Runge-Kutta method using MS Excel for different step sizes*

4 Results & Discussion

In this article, for a better understanding of the proposed three methods with different step sizes, various figures and result-table is used. Table: 1(a)-(e) and Table**:** 2(a)-(e**)** represents the approximate result analysis and also the error estimation for step sizes 0.1, 0.05, 0.025, and 0.0125 elaborately. Figures: 1(a)-(e) and Figures: 8(a)-(e) shown the exact solution curve along with the approximate solution curves for the different step sizes. Error estimation plays a vital role to determine the superiority of the proposed three methods. As a consequence, for a clearer appreciation, error analysis is shown in two different ways for every step size where Matlab programming language and MS Excel are used. Figures: 2-7 and Figures: 9-14 reveal the error estimation for the proposed three methods with different step sizes. It is observed that the accuracy level of all the proposed methods depends on the step sizes. The accuracy level of all the proposed methods is increased gradually when the step sizes become too much decreased. For all the proposed three methods it is revealed that a small step size provides a better approximation. Errors are analyzed for every step size, which established a comparison of accuracy and effectiveness among all the proposed three methods. The accuracy level and convergence rate are highest for Runge-Kutta fourth-order method compare to other methods.

5 Conclusion

In this article, three numerical methods namely Runge-Kutta fourth-order method, Modified Euler method, and Euler's method are discussed for solving the IVPs of the ODEs. The results of the proposed methods are best fitted with the exact solution for a relatively very much small step size h. From the result analysis table, it is observed that very much small step size should have taken for every method for achieving accurate results. The smaller step size is taken; the more accurate results are found. Among the three methods, the rate of convergence to the accurate solution was maximum for Runge-Kutta fourth-order method and minimum for the Euler method. It revealed that the Runge-Kutta fourth-order method is better compared to the other two methods in terms of accuracy and efficacy. To conclude, for the solution of IVPs of ODEs, in the field of science and engineering, the Runge-Kutta fourth-order method gives the best approximation result.

6 Declarations

6.1 Competing Interests

Authors guarantee that in this article, none of the authors have any contest of interests.

6.2 Publisher's Note

AIJR remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

How to Cite this Article:

M. A. Arefin, B. Gain, and R. Karim, "Accuracy Analysis on Solution of Initial Value Problems of Ordinary Differential Equations for Some Numerical Methods with Different Step Sizes", *Int. Ann. Sci.*, vol. 10, no. 1, pp. 118-133, Mar. 2021. https://doi.org/10.21467/ias.10.1.118-133

References

[1] M. A. Islam, "Accuracy Analysis of Numerical solutions of initial value problems (IVP) for ordinary differential equations (ODE)," *IOSR J. Math. Ver. III*, vol. 11, no. 3, pp. 2278–5728, 2015, doi: 10.9790/5728-11331823.

[2] M. A. Islam, "Accurate Solutions of Initial Value Problems for Ordinary Differential Equations with the Fourth Order Runge Kutta Method," *J. Math. Res.*, vol. 7, no. 3, pp. 41–45, 2015, doi: 10.5539/jmr.v7n3p41.

[3] A. Hasan, "Numerical Computation of Initial Value Problem by," vol. 1, no. 1, pp. 19–32, 2018.

[4] M. Babul Hossain, "A Comparative Study on Fourth Order and Butcher's Fifth Order Runge-Kutta Methods with Third Order Initial Value Problem (IVP)," *Appl. Comput. Math.*, vol. 6, no. 6, p. 243, 2017, doi: 10.11648/j.acm.20170606.12.

[5] S. Ivp, "Method for an Extension of the Runge-Iwita loo," vol. 42, no. 4, pp. 551–553, 1992.

[6] O. Abraham, "Improving the Modified Euler Method," *Leonardo J. Sci.*, vol. 6, no. 10, pp. 1–8, 2007.

[7] M. H. Afshar and M. Rohani, "Embedded modified Euler method: An efficient and accurate model," *Proc. Inst. Civ. Eng. Water Manag.*, vol. 162, no. 3, pp. 199–209, 2009, doi: 10.1680/wama.2009.162.3.199.

[8] G. D. Hahn, "A modified Euler method for dynamic analyses," *Int. J. Numer. Methods Eng.*, vol. 32, no. 5, pp. 943–955, 1991, doi: 10.1002/nme.1620320502.

[9] I. Journal, "Modified Euler Method for Finding Numerical Solution of Intuitionistic Fuzzy Differential Equation under Generalized Differentiability Concept," no. February, 2018.

[10] J. Zhang and B. Jiang, "Enhanced Euler's method to a free boundary porous media flow problem," *Numer. Methods Partial Differ. Equ.*, vol. 28, no. 5, pp. 1558–1573, 2012, doi: 10.1002/num.20691.

[11] A. B. M. Hame, I. Yuosif, I. A. Alrhama, and I. San, “Open Access The Accuracy of Euler and modified Euler Technique for First Order Ordinary Differential Equations with initial condition,” no. 9, pp. 334–338, 2017.

[12] J. Demsie Abraha, “Comparison of Numerical Methods for System of First Order Ordinary Differential Equations,” *Pure Appl. Math. J.*, vol. 9, no. 2, p. 32, 2020, doi: 10.11648/j.pamj.20200902.11.

[13] M. A. Islam, “A Comparative Study on Numerical Solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE) with Euler and Runge Kutta Methods,” *Am. J. Comput. Math.*, vol. 05, no. 03, pp. 393–404, 2015, doi: 10.4236/ajcm.2015.53034.

[14] Z. H. Maibed and M. S. Mechee, “A Compression Study of Multistep Iterative Methods for Solving Ordinary Differential Equations,” no. February, 2020.

[15] D. Houcque and R. R. Mccormick, “Applications of MATLAB: Ordinary Differential Equations (ODE),” pp. 101–323, 2008.

[16] Burden, R. L., and J. D. Faires. "Numerical Analysis (Thomson, Brooks/Cole, Belmont, CA, USA)." (2005).