A Study on Numerical Solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE) with 4th-Order of Taylor's Method and 4th-Order of RK Methods Using Matlab

[1]C.Senthilnathan, [2]Dr.S.Karunanithi, [3]N.Gajalakshmi,[4]M.Malarvizhi

[1]Lecturer,[2,3,4]Asst.Professor,

[1]PG & Research Department of Mathematics,

[1]Government Thirumagal Mills College,Gudiyattam,VelloreDist,TamilNadu,India

Abstract :So far , This paper mainly presents fourth-order of Taylor's Method and fourth-order of Runge-kutta's method for solving initial value problems (IVP) for ordinary differential equations (ODE). The three methods are quite efficient and practically well suited for solving these problems. In order to verify the accuracy, we compare numerical solutions with the exact solutions in Matlab. The numerical solutions are in good agreement with the exact solutions. Numerical comparisons between Fourth-order Taylor's Method and fourth-order of Runge-kutta's method have been compare. We also compare the performance and the computational effort of such methods. In order to achieve higher accuracy in the solution, the step size needs to be very small. Finally we investigate and compute the errors of the three proposed methods for different step sizes to examine superiority.

Keywords: Initial Value Problem (IVP), Fourth-Order of Taylor's method , Fourth-Order of Runge Kutta Method, Error Analysis, MATLAB

I. INTRODUCTION

Numerical analysis is the study of algorithms that use numerical approximation (as opposed to general symbolic manipulations) for the problems of mathematical analysis. One of the earliest mathematical writings is a Babylonian tablet from the Yale Babylonian Collection (YBC 7289), which gives a sexagesimal numerical approximation of p^2, the length of the diagonal in a unit square. Being able to compute the sides of a triangle (and hence, being able to compute square roots) is extremely important, for instance, in astronomy, carpentry and construction. Numerical analysis continues this long tradition of practical mathematical calculations. Much like the Babylonian P approximation of 2, modern numerical analysis does not seek exact answers, because exact answers are often impossible to obtain in practice. Instead, Much of numerical analysis is concerned with obtaining approximate solutions while maintaining reasonable bounds on errors.

Numerical analysis naturally finds applications in all fields of engineering and the physical science, but in 21st century also the life sciences and even the arts have adopted elements of scientific computations.

Many great mathematicians of were preoccupied by numerical analysis, as it is obvious form the names of important algorithms Euler's method, Taylor's method. We are familiar to the differential equations is the one of the area of the numerical analysis.

values at some selected points on the sub-interval. Runge Kutta method is a more general and improvised method as compared to that of the Euler method. But We compare to fourth-order Taylor's method , fourth-order of RungaKutta method observe that in the Euler method excessively small step size converges to analytical solution. So, large number of computation is needed. In contrast, Taylor's method gives better results and it converges faster to analytical solution and has less iteration to get accuracy solution.

This paper as follows: Section 2: problem formulations; Section 3:Fourth –Order of Taylor Method section 4: Fourth-order of Runga-Kutta Method section 5: Error analysis; Section 6:Matlab Software, section 7: numerical examples; Section 8: discussion of results; section 9: The conclusion of the paper and the last is Reference

2. Problem Formulation

In this section we consider Three numerical methods for finding the approximate solutions of the initial value problem (IVP) of the first-order ordinary differential equation has the form

$$y' = f(x, y(x)), x \in (x_0, x_n) \quad (1)$$
$$y(x_0) = y_0$$

Where x_0 and y_0 are initial values for x and y respectively.

Our aim is to determine (approximately) the unknown function $y(x)$ for $x \geq x_0$. We are told explicity the value of $y(x_0)$,namely y_0,using the given differential equation (1),we can also determine exactly the instantaneous rate of change of y at point x_0

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

If the rate of change of $y(x)$ were to remain $f(x_0, y_0)$ for all point x, then $y(x)$ would exactly $y_0 + f(x_0, y_0)(x - x_0)$. The rate of change of y(x) does not remain $f(x_0, y_0)$ for all x, but it is reasonable to expect that it remains close to $f(x_0, y_0)$ for x close to x_0 , for small number h ,and is called the step size.The numerical solutions of (1) is given by a set of points $\{(x_n, y_n): n = 0,1,2, \ldots, n\}$ and each point (x_n, y_n) is an approximation to the corresponding point $(x_n, y(x_n))$ on the solution curve.

3. Fourth-Order of Taylor Method

The object of a numerical techniques is to determine accurate approximations with minimal effort, we need a means for comparing the efficiency of various approximation methods.

Consider the initial value problem

$$y' = f(x, y), \quad a \leq x \leq b, \quad y(a) = \alpha.$$

has $(n+1)$ continuous derivatives. If we expand the solution, $y(x)$, in terms of its nth Taylor polynomial about x_i and evaluate x_{i+1}, we obtain

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2!}y''(x_i) + \frac{h^3}{3!}y'''(x_i) + \ldots + \frac{h^n}{n!}y^{(n)}(x_i) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_i) \quad \ldots\ldots\ldots\ldots\ldots(4.1)$$

for some ξ_i in (x_i, x_{i+1}), where $h = \frac{b-a}{N}$, $i = 0,1,2,\ldots..N-1$.

Successive differentiation of the solution $y(x)$ gives

$y'(x) = f(x, y(x))$, $y''(x) = f'(x, y(x))$,and generally, $y^{(k)}(x) = f^{(k-1)}(x, y(x))$.

Substituting these results into(4.1) gives

$$y(x_{i+1}) = y(x_i) + hf(x_i, y(x_i)) + \frac{h^2}{2!}f'(x_i, y(x_i)) + \frac{h^3}{3!}f''(x_i, y(x_i)) + \ldots$$

$$+ \frac{h^n}{n!}f^{(n-1)}(x_i, y(x_i)) + \frac{h^{n+1}}{(n+1)!}f^{(n)}(\xi_i, y(\xi_i)) \quad \ldots\ldots\ldots\ldots\ldots(4.2)$$

The difference equation method corresponding to (4.2) is obtained by deleting the remainder term involving ξ_i .

Taylor method of order 'n'

$$w_0 = \alpha$$

$$w_{i+1} = w_i + hT^{(n)}(x_i, w_i) \quad \text{for each } i = 0,1,2,\ldots..N-1, \quad \ldots\ldots\ldots\ldots\ldots(4.3)$$

where $T^{(n)}(x_i, w_i) = f(x_i, w_i) + \frac{h}{2}f'(x_i, w_i) + \ldots + \frac{h^{n-1}}{n!}f^{(n-1)}(x_i, w_i)$ .Euler's method is Taylor's method of order one.

4.Fourth-Order of Runge-Kutta Method

The Runge Kutta method is most popular because it is quite accurate, stable and easy to program. This method is distinguished by their order in the sense that they agree with Taylor's series solution up to terms of $h^r$where r is the order of the method. It do not demand prior computational of higher derivatives of $y(x)$ asin Taylor's series method. The fourth order Runge Kutta method (RK4) is widely used for solving initial value problems (IVP) for ordinary differential equation (ODE).

Runge Kutta 's Method of Order 4

$$y(x_{i+1}) = y(x_i) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), i = 0,1,2,3, \ldots$$

Where ,

$$k_1 = hf(x_i, y_i), k_2 = hf\left(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_1\right), k_3 = hf\left(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}k_2\right), k_4 = hf(x_i + h, y_i + k_3),$$

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

$$y(x + h) = y(x) + \Delta y.$$

5.Error Analysis

There are two types of errors in numerical solution of ordinary differential equations. Round-off errors and Truncation errors occur when ordinary differential equations are solved numerically. Rounding errors originate from the fact that computers can only represent numbers using a fixed and limited number of significant figures. Thus, such numbers or cannot be represented exactly in computer memory. The discrepancy introduced by this limitation is call Round-off error. Truncation errors in numerical analysis arise when approximations are used to estimate some quantity. The accuracy of the solution will depend on how small we make the step size, *h*.

A numerical method is said to be convergent if

$$\lim_{\substack{h\to 0 \\ 1\le n\le N}} |y(x_n) - y_n| = 0.$$

where y(x) denotes the approximate solution and y_n denote the exact solution. In this thesis we consider two initial value problems to verify accuracy of the proposed methods. The approximated solution is evaluated by using MATLAB software for two proposed numerical method at different step.

The maximum error by

$$e_r = \max_{1\le n\le steps} (|y(x_n) - y_n|).$$

6.Matlab Software

MATLAB is widely used in all areas of applied mathematics, in education and research at universities, and in the industry. MATLAB stands for MATrixLABoratory and the software is built up around vectors and matrices.
This makes the software particularly useful for linear algebra but MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration.

MATLAB has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language, and is one of the easiest programming languages for writing mathematical programs. MATLAB also has some tool boxes useful for signal processing, image processing, optimization, etc.

MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment. It has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research.

The techniques for solving differential equations based on numerical approximations were developed before programmable computers existed. During World War II, it was common to find rooms of people (usually women) working on mechanical calculators to numerically solve systems of differential equations for military calculations. Before programmable computers, it was also common to exploit analogies to electrical systems to design analog computers to study mechanical, thermal, or chemical systems.

As programmable computers have increased in speed and decreased in cost increasingly complex systems of differential equations can be solved with simple programs written to run on a common PC. Currently, the computer on your desk can tackle problems that were inaccessible to the fastest supercomputers just 5 or 10 years ago.

Numerical methods for solving ordinary differential equations are discussed in many textbooks. Here we will discuss how to use some of them in MATLAB. In particular, we will examine how a reduction in the “step size” used by a particular algorithm reduces the error of the numerical solution, but only at a cost of increased computation time. In line with the philosophy that we are not emphasizing programming in this manual, MATLAB routines for these numerical methods are made available.

7. Numerical Examples

In this section we consider two numerical examples to prove which numerical methods converge faster to analytical solution. Numerical results and errors are computed and the outcomes are represented by graphically.

Example-1 : We consider the initial value problem $y' = y - x^2 + 1$, $0 \le x \le 2.4$, $y(0) = 0.5$ .The exact solution of the given problem is given by $y = (x+1)^2 - 0.5e^x$ The approximate results and maximum errors are obtained and shown in **Tables 1(a,b)** and the graphs of the numerical solutions are displayedin**Figures 1(a,b)**.

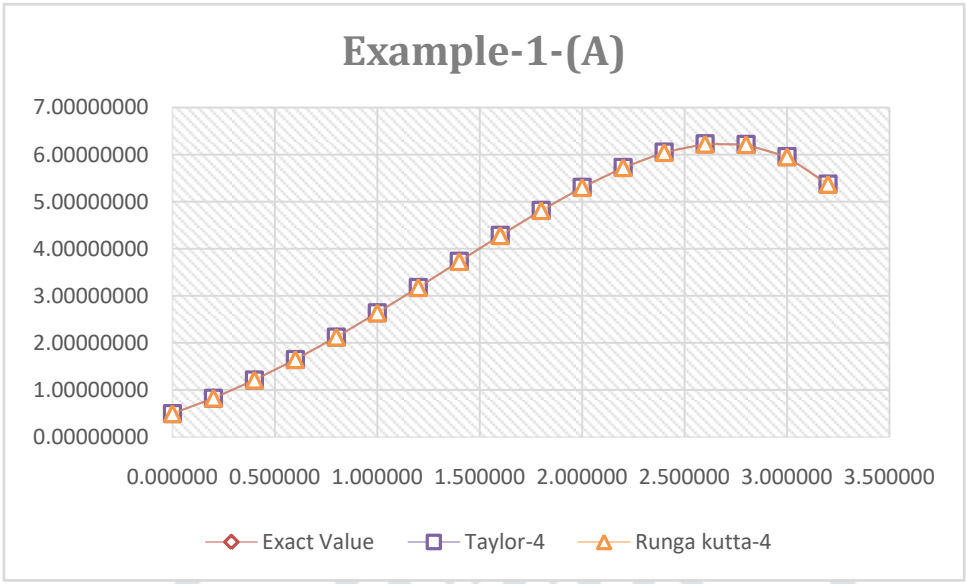
Table: 1(a) Lists the numerical and analytical values of $y=(1+x)^2-0.5e^x$ for $x \le x \le 2.4$

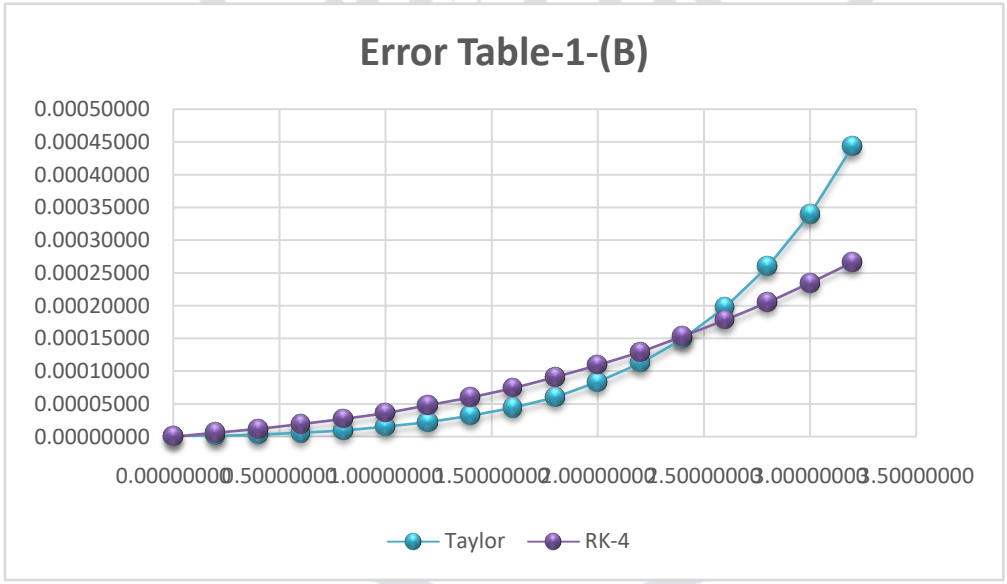
X-Value	Exact Value	Taylor-4	Runga kutta-4
0.000000	0.50000000	0.50000000	0.50000000
0.200000	0.82929900	0.82930000	0.82929300
0.400000	1.21408800	1.21409110	1.21407600
0.600000	1.64894100	1.64894680	1.64892200
0.800000	2.12723000	2.12723970	2.12720300
1.000000	2.64085900	2.64087440	2.64082300
1.200000	3.17994200	3.17996410	3.17989400
1.400000	3.73240000	3.73243210	3.73234000
1.600000	4.28348400	4.28352830	4.28341000
1.800000	4.81517700	4.81523750	4.81508600
2.000000	5.30547200	5.30555553	5.30536300
2.200000	5.72749300	5.72760530	5.72736400
2.400000	6.04841200	6.04856110	6.04825900
2.600000	6.22813100	6.22832870	6.22795300
2.800000	6.21767700	6.21793650	6.21747200
3.000000	5.95723200	5.95757150	5.95699700
3.200000	5.37373500	5.37417790	5.37346900

Table-1(b) shows the errors of Taylor and Runga-Kutta methods with exact method. These error values for each x are in the order Taylor4 > Runga-kutta4

X-Value	Exact Value	Taylor	RK-4
0.00000000	0.50000000	0.00000000	0.00000000
0.20000000	0.82929860	0.00000100	0.00000600
0.40000000	1.21408760	0.00000310	0.00001200
0.60000000	1.64894060	0.00000580	0.00001900
0.80000000	2.12722950	0.00000970	0.00002700
1.00000000	2.64085910	0.00001540	0.00003600
1.20000000	3.17994170	0.00002210	0.00004800
1.40000000	3.73239990	0.00003210	0.00006000
1.60000000	4.28348400	0.00004430	0.00007400
1.80000000	4.81517650	0.00006050	0.00009100
2.00000000	5.30547190	0.00008353	0.00010900
2.20000000	5.72749330	0.00011230	0.00012900
2.40000010	6.04841180	0.00014910	0.00015300
2.59999990	6.22813080	0.00019770	0.00017800
2.80000000	6.21767660	0.00025950	0.00020500
3.00000000	5.95723150	0.00033950	0.00023500
3.20000000	5.37373500	0.00044290	0.00026600

Figure-1(a,b) shows the function $y=(1+x)^2-0.5e^x$ on the interval $[0,3.2]$ usingMATLABHere, The Taylor4th and Rungakutta 4th curve is more nearest to other curves.





Example-2:
We consider the initial value problem $y'=y-x,\ 0\le x\le 3.2,\ y(0)=2.$The exact solution of the given problem is given by $y=1+x+e^x$. The approximate results and maximum errors are obtained and shown in **Tables 2(a,b)**and the graphs of the numerical solutions are displayed in **Figures :2 (a,b)**

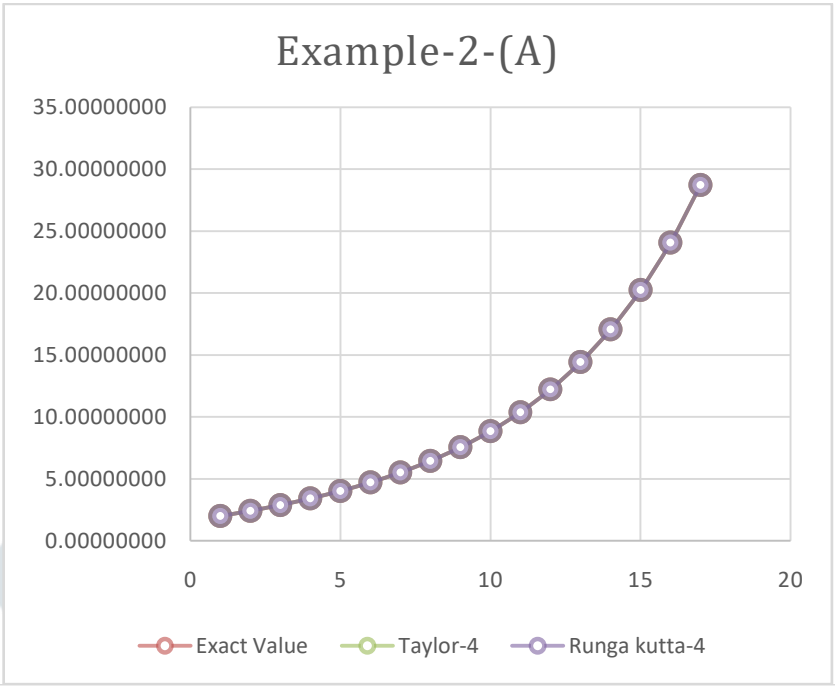
Table-2(a,b) lists the numerical and analytical values of $y = 1 + x + e^x$ for $0 \le x \le 3.2$ and here the values for each x as like as (Exact > Taylor4) ≈ (Runga kutta4)

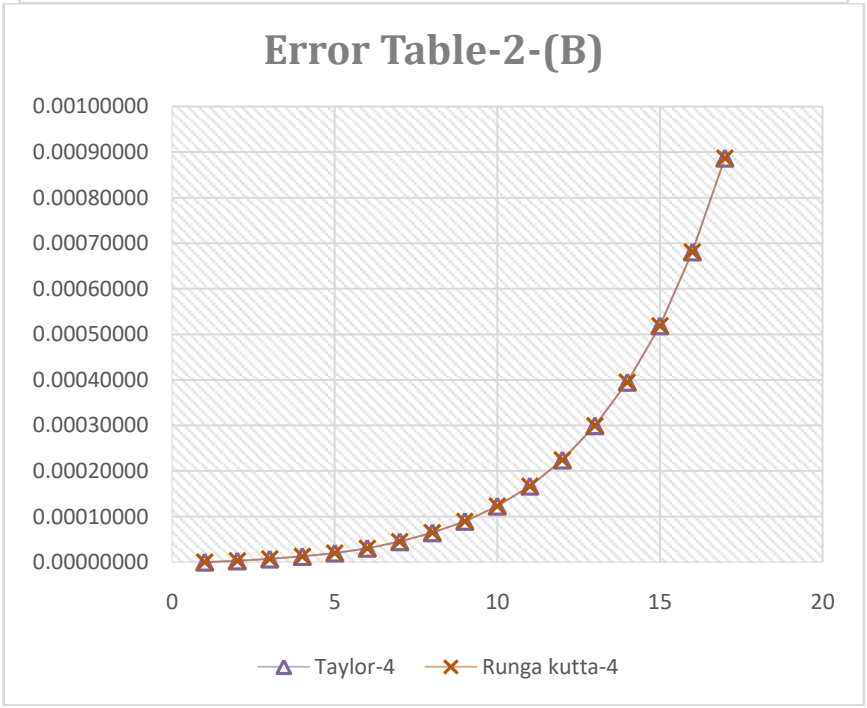
X-Value	Exact Value	Taylor-4	Runga kutta-4
0.00000000	2.00000000	2.00000000	2.00000000
0.20000000	2.42140270	2.42140010	2.42140010
0.40000000	2.89182470	2.89181800	2.89181800
0.60000000	3.42211890	3.42210650	3.42210650
0.80000000	4.02554080	4.02552080	4.02552080
1.00000000	4.71828170	4.71825120	4.71825120
1.20000000	5.52011680	5.52007200	5.52007200
1.40000000	6.45520020	6.45513580	6.45513580
1.60000000	7.55303240	7.55294280	7.55294280
1.80000000	8.84964750	8.84952450	8.84952450
2.00000000	10.38905620	10.38888930	10.38888930
2.20000000	12.22501370	12.22478960	12.22478960
2.40000000	14.42317680	14.42287730	14.42287730
2.60000000	17.06373790	17.06334300	17.06334300
2.80000000	20.24464610	20.24412730	20.24412730
3.00000000	24.08553700	24.08485600	24.08485600
3.20000000	28.73253060	28.73164370	28.73164370

The table-2(a,b) shows the errors of Euler's ,Taylor and Runga-kuttahigh order methods with exact method. These errors values for each x are in the order Taylor4 ~ Runga-kutta4<exact

X-Value	Exact Value	Taylor-4	Runga kutta-4
0.00000000	2.00000000	0.00000000	0.00000000
0.20000000	2.42140270	0.00000260	0.00000260
0.40000000	2.89182470	0.00000670	0.00000670
0.60000000	3.42211890	0.00001240	0.00001240
0.80000000	4.02554080	0.00002000	0.00002000
1.00000000	4.71828170	0.00003050	0.00003050
1.20000000	5.52011680	0.00004480	0.00004480
1.40000000	6.45520020	0.00006440	0.00006440
1.60000000	7.55303240	0.00008960	0.00008960
1.80000000	8.84964750	0.00012300	0.00012300
2.00000000	10.38905620	0.00016690	0.00016690
2.20000000	12.22501370	0.00022410	0.00022410
2.40000000	14.42317680	0.00029950	0.00029950
2.60000000	17.06373790	0.00039490	0.00039490
2.80000000	20.24464610	0.00051880	0.00051880
3.00000000	24.08553700	0.00068100	0.00068100
3.20000000	28.73253060	0.00088690	0.00088690

Figure-2 (i),(ii) shows the function $y = 1 + x + e^x$ on the interval $[0, 3.2]$ usingMATLAB. Here, The Taylor4th and Runga-kutta 4thorder curve is more nearest to other curves





8. Discussion of Results

The Taylor series method is of general applicability and it is the standard to which we compare the accuracy of the various other numerical methods for solving a ***Linear Ordinary Differential Equation with Initial Values.*** We compared Runga-kuttaorder4 and Taylor's method with Exact method in this paper.We compared among them through MATLAB program

9. Conclusion

The graphs that are table (1(a,b) and Table 2(a,b) the results obtained from different methods using *MATLAB* consequently, we can see ***Runge-kutta-4*** is ***more accurate*** than Taylor's Method . And very nearest to **Exact Value**. In particular, the accuracy of solution of Runge-Kutta 4^{th} order method is***nearer*** to that of the exact solution and ***error*** is also ***very less*** for higher order when compared to lower orders.

Thus, if we want better accuracy for the solution of IVP, we should use higher order approximations. Thus , one can easily adapt the MATLAB coded as needed for a different type of problems.Thus, if we want better accuracy for the solution of IVP, we should use higher order approximations. Thus one can easily adapt the MATLAB coded as needed for a different type of problems.

References

[1] Islam, Md.A. (2015) Accuracy Analysis of Numerical solutions of Initial Value Problems (IVP) for Ordinary DifferentialEquations (ODE). *IOSR Journal of Mathematics*, **11**, 18-23.

[2] Islam, Md.A. (2015) Accurate Solutions of Initial Value Problems for Ordinary Differential Equations with FourthOrder Runge KuttaMethod. *Journal of Mathematics Research*, **7**, 41-45. http://dx.doi.org/10.5539/jmr.v7n3p41

[3] Ogunrinde, R.B., Fadugba, S.E. and Okunlola, J.T. (2012) On Some Numerical Methods for Solving Initial Value Problems in Ordinary Differential Equations. *IOSR Journal of Mathematics*, **1**, 25-31.http://dx.doi.org/10.9790/5728-0132531

[4] Shampine, L.F. and Watts, H.A. (1971) Comparing Error Estimators for Runge-Kutta Methods. *Mathematics of Computation*, **25**, 445-455. http://dx.doi.org/10.1090/S0025-5718-1971-0297138-9

[5] Eaqub Ali, S.M. (2006) A Text Book of Numerical Methods with Computer Programming. Beauty Publication, Khulna.

[6] Akanbi, M.A. (2010) Propagation of Errors in Euler Method, Scholars Research Library. *Archives of Applied ScienceResearch*, **2**, 457-469.

[7] Kockler, N. (1994) Numerical Method for Ordinary Systems of Initial value Problems. John Wiley and Sons, NewYork.

[8] Lambert, J.D. (1973) Computational Methods in Ordinary Differential Equations. Wiley, New York.

[9] Gear, C.W. (1971) Numerical Initial Value Problems in Ordinary Differential Equations. Prentice-Hall, Upper SaddleRiver.

[10] Hall, G. and Watt, J.M. (1976) Modern Numerical Methods for Ordinary Differential Equations. Oxford UniversityPress, Oxford.

[11] Hossain, Md.S., Bhattacharjee, P.K. and Hossain, Md.E. (2013) Numerical Analysis. Titas Publications, Dhaka.

[12] Balagurusamy, E. (2006) Numerical Methods. Tata McGraw-Hill, New Delhi.

[13] Sastry, S.S. (2000) Introductory Methods of Numerical Analysis. Prentice-Hall, India.

[14] Burden, R.L. and Faires, J.D. (2002) Numerical Analysis. Bangalore, India.

[15] Gerald, C.F. and Wheatley, P.O. (2002) Applied Numerical Analysis. Pearson Education, India.

[16] Mathews, J.H. (2005) Numerical Methods for Mathematics, Science and Engineering. Prentice-Hall, India.

[17] Richard L.Burden, J.DouglasFaires, **Numerical Analysis,** 9^{th} edition.

[18] M.K.Jain, S.R.K.Iyengar, R.K.Jain, **Numerical Methods for Scientific and Engineering Computation,** 3^{rd} edition, New Age International(p) Limited publishers(2002).

[19] Amos Gilat ,**MATLAB An Introduction with Applicatins,** 5^{th} edition, Published by John Wiley & Sons inc, U.K.(2015).

[20] John.H.Mathews, Kurtis D.Fink, **Numerical Methods Using MATLAB,** 3^{rd} edition, HogskolenivestfoidBiblioteket-Borre.(2000)

[21] JaanKiusalass, **Numerical Methods in Engineering with Python,** Cambridge University Press, New York(2005).

[22] AlfioQuarteroni, Riccardo Sacco, Fausto Saleri, **Numerical Mathematics,** Springer-Verlage, New York(2000).

[23] D.Barton, I.M.Willers and R.V.M.Zahar,**The Automatic solution ofOrdinary Differential Equations,** by the method of Taylor series.(1971).

[24] D.Barton, I.M.Willers and R.V.M.Zahar,**Taylor series Method forOrdinary Differential Equations, an Evaluation.**(1970).

[25] P.R.Vittal, **Differential Equations, Fourier and Laplace Transforms, Probability,**MarghamPublications,Chennai-17(2012).