A Comparative Study on Numerical Solutions of Initial Value Problems (IVP) for Ordinary Differential Equations (ODE) with Euler and Higher Order of Taylor's Methods Using Matlab

[1] C.Senthilnathan, [2] Dr.S.Karunanithi, [3] N.Gajalakshmi, [4] M.Malarvizhi
[1]Lecturer , [2]Ass.Professor, [3]Ass.Professor, Ass.Professor
[1]PG & Research Department of Mathematics,
[1]Government Thirumagal Mills College,Gudiyattam,Vellore Dist,TamilNadu,India

Abstract: This paper mainly presents Euler method and Higher order of Taylor's Method (TR4) for solving initial value problems (IVP) for ordinary differential equations (ODE). The two proposed methods are quite efficient and practically well suited for solving these problems. In order to verify the ac- curacy, we compare numerical solutions with the exact solutions. The numerical solutions are in good agreement with the exact solutions. Numerical comparisons between Euler method and Higher order of Taylor's Method have been presented. Also we 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 two proposed methods for different step sizes to examine superiority. Several numerical examples are given to demonstrate the reliability and efficiency.

Keywords: Initial Value Problem (IVP), Euler Method, Higher Order of Runge Kutta Method, Error Analysis, Matlab Software.

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.

A more robust and intricate numerical technique is the Runge Kutta method. This method is the most widely used one since it gives reliable starting values and is particularly suitable when the computation of higher de-rivatives is complicated. The numerical results are very encouraging. Finally, two examples of different kinds of ordinary differential equations are given to verify the proposed formulae. The results of each numerical example indicate that the convergence and error analysis which are discussed illustrate the efficiency of the methods. The use of Euler method to solve the differential equation numerically is less efficient since it requires *h* to be small for obtaining reasonable accuracy. It is one of the oldest numerical methods used for solving an ordinary initial value differential equation, where the solution will be obtained as a set of tabulated values of variables *x* and *y*. It is a simple and single step but a crude numerical method of solving first-order ODE, particularly suitable for quick programming because of their great simplicity, although their accuracy is not high. But in Runge Kutta method, the derivatives of higher order are not required and they are designed to give greater accuracy with the advantage of requiring only the functional values at some selected points on the sub-interval. Runge Kutta me-thod is a more general and improvised method as compared to that of the Euler method. We observe that in the Euler method excessively small step size converges to analytical solution. So, large number of computation is needed. In contrast, Runge Kutta method gives better results and it converges faster to analytical solution and has less iteration to get accuracy solution.

This paper is organized as follows: Section 2: problem formulations; Section 3:Euler Method section 4:Higher order of Taylor Method section 5: Error analysis; Section 6:Matlab Software section7: numerical examples; Section 7: discussion of results; and the last section: The conclusion of the paper.

2. Problem Formulation

In this section we consider two 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 $\{(x_n, y_n): n = 0,1,2, \dots, n\}$ and each point (x_n, y_n) is an approximation to the corresponding point $(x_n, y(x_n))$ on the solution curve.

3. Euler Method

Euler's method is the most elementary approximation technique for solving initial-value problems. Although it is seldom used in practice, the simplicity of its derivation can be used to illustrate the techniques involved in the construction of some of the more advanced techniques, without the cumbersome algebra that accompanies these constructions.

The object of Euler's method is to obtain approximations to the well-posed initial-value problem

$$\frac{dy}{dx} = f(x, y), \quad a \leq x \leq b, \quad y(a) = \alpha. \quad \text{.....(3.1)}$$

A continuous approximation to the solution $y(x)$ will not be obtained; instead, approximations to y will be generated at various values, called **mesh points**, in the interval $[a, b]$. Once the approximate solution is obtained at the points, the approximate solution at other points in the interval can be found by interpolation.

We first make the stipulation that the mesh points are equally distributed throughout the interval $[a, b]$. This condition is ensured by choosing a positive integer N and selecting the mesh points

$$x_i = a + ih, \quad \text{for each } i = 0,1,2,\dots,N \quad \text{.....(3.2)}$$

the common distance between the points $h = (b-a)/N = x_{i+1} - x_i$ is called the **step size**.

We will use Taylor's Theorem to derive Euler's method. Suppose that $y(x)$, the unique solution to (3.2), has two continuous derivatives on $[a, b]$, so that for each $i = 0,1,2,\dots,N-1$,

$$y(x_{i+1}) = y(x_i) + (x_{i+1} - x_i)y'(x_i) + \frac{(x_{i+1} - x_i)^2}{2} y''(\xi_i),$$

for some number ξ_i in (x_i, x_{i+1}). Because $h = x_{i+1} - x_i$, we have

$$y(x_{i+1}) = y(x_i) + hy'(x_i) + \frac{h^2}{2} y''(\xi_i),$$

and, because $y(x)$ satisfies the differential equation (3.1),

$$y(x_{i+1}) = y(x_i) + hf(x_i, y(x_i)) + \frac{h^2}{2} y''(\xi_i), \quad \text{.....(3.3)}$$

Euler's method constructs $w_i \approx y(x_i)$, for each $i = 0,1,2,\dots,N$, by deleting the remainder term.

Thus Euler's method is

$$w_0 = \alpha,$$

$$w_{i+1} = w_i + hf(x_i, w_i), \quad \text{for each } i = 0,1,2,.....,N-1. \qquad \text{……..(3.4)}$$

We handle the numerical problems through the MATLAB Programs with same step size.

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

4. Higher 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

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

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

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

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

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

$$y(x_{i+1}) = y(x_i) + hf(x_i, y(x_i)) + \frac{h^2}{2!} f'(x_i, y(x_i)) + \frac{h^3}{3!} f''(x_i, y(x_i)) + ...$$
$$+ \frac{h^n}{n!} f^{(n-1)}(x_i, y(x_i)) + \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)) \text{ ……..(4.2)}$$

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

Taylor method of order 'n'

$$w_0 = \alpha$$

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

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

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.

The tutorials are independent of the rest of the document. The primarily objective is to help you learn quickly the first steps. The emphasis here is "learning by doing". Therefore, the best way to learn is by trying it yourself. Working through the examples will give you a feel for the way that MATLAB operates. In this introduction we will describe how MATLAB handles simple numerical expressions and mathematical formulas.

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.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox.

There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering. In addition to the MATLAB documentation which is mostly available online.

MATLAB was first adopted by researchers and practitioners in control engineering, Little's specialty, but quickly spread to many other domains. It is now also used in education, in particular the teaching of linear algebra, numerical analysis, and is popular amongst scientists involved in image processing.

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.

First, we will review some basic concepts of numerical approximations and then introduce Euler's method, the simplest method. We will provide details on algorithm development using the Euler method as an example. Next we will discuss error approximation and discuss some better techniques. Finally we will use the algorithms that are built into the MATLAB programming environment.

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 $y' = y - x^2 + 1$, $0 \le x \le 3.2$, $y(0) = 0.5$ .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 $0 \le x \le 3.2$ and here the values for each x as like as Euler< (Exact ≈ Taylor4) < Taylor3 < Taylor2.

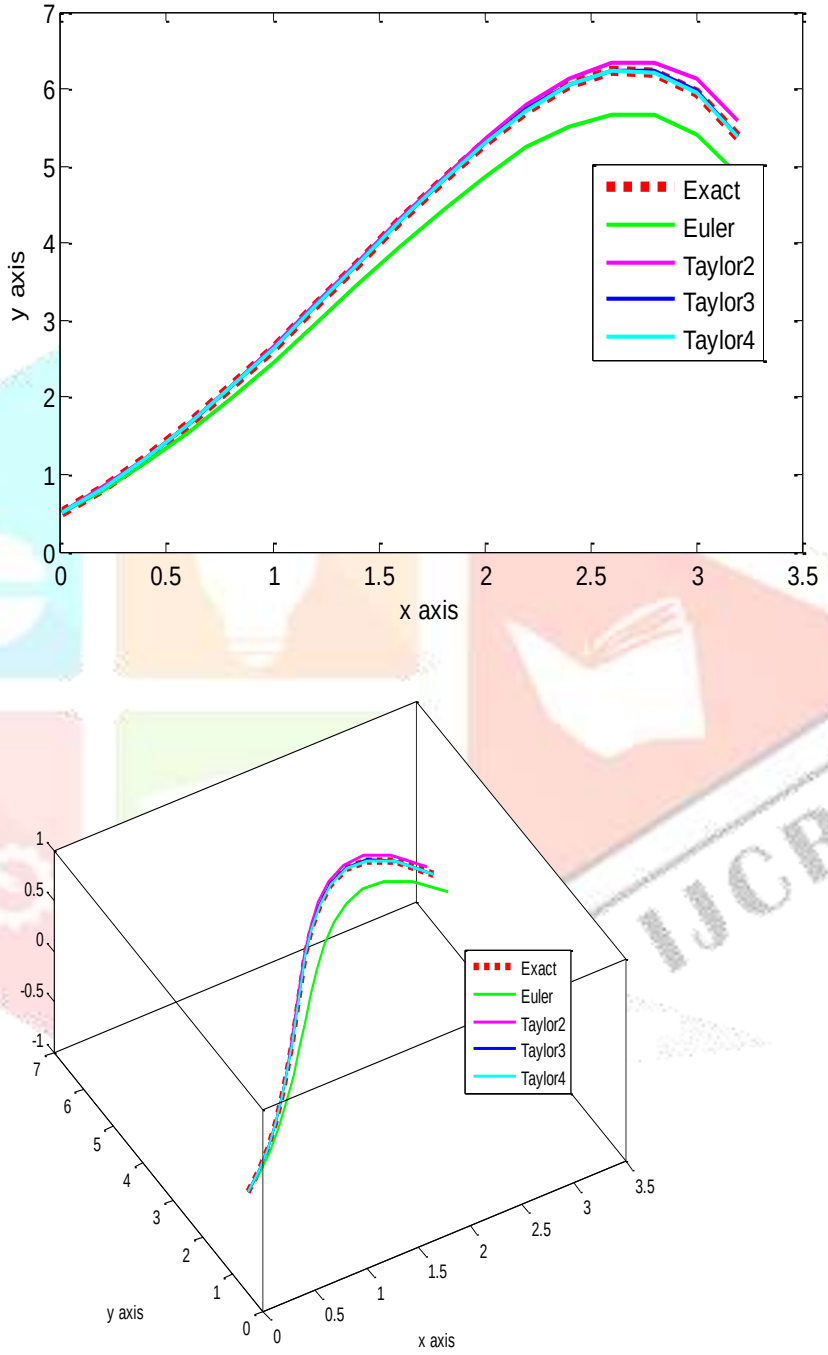
x-value	Exact	Euler	Taylor2	Taylor3	Taylor4
0	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000
0.2000000	0.8292986	0.8000000	0.8300000	0.8293333	0.8293000
0.4000000	1.2140876	1.1520000	1.2158000	1.2141725	1.2140911
0.6000000	1.6489406	1.5504000	1.6520760	1.6490959	1.6489468
0.8000000	2.1272295	1.9884800	2.1323328	2.1274824	2.1272397
1.0000000	2.6408591	2.4581759	2.6486459	2.6412454	2.6408744
1.2000000	3.1799417	2.9498112	3.1913481	3.1805077	3.1799641
1.4000000	3.7323999	3.4517734	3.7486446	3.7332065	3.7324321
1.6000000	4.2834840	3.9501281	4.3061466	4.2846098	4.2835283
1.8000000	4.8151765	4.4281540	4.8462987	4.8167233	4.8152375
2.0000000	5.3054719	4.8657846	5.3476844	5.3075714	5.3055553
2.2000000	5.7274933	5.2389412	5.7841749	5.7303138	5.7276053
2.4000001	6.0484118	5.5187297	6.1238933	6.0521698	6.0485611
2.5999999	6.2281308	5.6704755	6.3279500	6.2331038	6.2283287
2.8000000	6.2176766	5.6525707	6.3488989	6.2242174	6.2179365
3.0000000	5.9572315	5.4150848	6.1288567	5.9657907	5.9575715
3.2000000	5.3737350	4.8981018	5.5972052	5.3848858	5.3741779

Table-1(b) shows the errors of Euler's ,Taylor high order methods with exact method. These error values for each x are in the order Taylor4 < Taylor3 < Taylor2 < Euler.

x-value	Euler	Taylor2	Taylor3	Taylor4
0	0	0	0	0
0.2000000	0.0292986	0.0007014	0.0000347	0.0000014
0.4000000	0.0620877	0.0017123	0.0000848	0.0000034
0.6000000	0.0985406	0.0031354	0.0001553	0.0000062
0.8000000	0.1387495	0.0051032	0.0002530	0.0000101
1.0000000	0.1826831	0.0077868	0.0003862	0.0000153
1.2000000	0.2301303	0.0114065	0.0005661	0.0000225
1.4000000	0.2806266	0.0162446	0.0008066	0.0000321
1.6000000	0.3333557	0.0226626	0.0011259	0.0000447
1.8000000	0.3870225	0.0311223	0.0015470	0.0000615
2.0000000	0.4396875	0.0422123	0.0020994	0.0000834
2.2000000	0.4885519	0.0566816	0.0028206	0.0001121
2.4000001	0.5296821	0.0754815	0.0037582	0.0001494
2.5999999	0.5576553	0.0998188	0.0049726	0.0001976
2.8000000	0.5651059	0.1312222	0.0065406	0.0002599
3.0000000	0.5421466	0.1716250	0.0085591	0.0003402
3.2000000	0.4756330	0.2234701	0.0111507	0.0004432

Figure-1(a,b) shows the function $y=(1+x)^2-0.5e^x$ on the interval $[0,3.2]$ usingMATLAB where red, green, magenta, blue, cyan colour curves denote Exact and Taylor's of order2,3,4 curves respectively. The green curve Eulercurve is below to the other curves and the magenta colour Talyor2 curve is above to all curves but the blue colour Taylor3 is in-between Taylor2 andTaylor4. Here, The Taylor4th curve is more nearest to other curves.

(i).2Dand 3D View



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)**

x-value	Exact	Euler	Taylor2	Taylor3	Taylor4
0	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
0.2000000	2.4214027	2.4000001	2.4200001	2.4213333	2.4214001
0.4000000	2.8918247	2.8399999	2.8884001	2.8916552	2.8918180
0.6000000	3.4221189	3.3280001	3.4158480	3.4218080	3.4221065
0.8000000	4.0255408	3.8736000	4.0153346	4.0250349	4.0255208
1.0000000	4.7182817	4.4883199	4.7027082	4.7175093	4.7182512
1.2000000	5.5201168	5.1859841	5.4973040	5.5189848	5.5200720
1.4000000	6.4552002	5.9831810	6.4227109	6.4535866	6.4551358
1.6000000	7.5530324	6.8998170	7.5077071	7.5507808	7.5529428
1.8000000	8.8496475	7.9597802	8.7874031	8.8465538	8.8495245
2.0000000	10.3890562	9.1917362	10.3046312	10.3848572	10.3888893
2.2000000	12.2250137	10.6300840	12.1116505	12.2193727	12.2247896
2.4000001	14.4231768	12.3161001	14.2722130	14.4156599	14.4228773
2.5999999	17.0637379	14.2993202	16.8640995	17.0537930	17.0633430
2.8000000	20.2446461	16.6391850	19.9822025	20.2315655	20.2441273
3.0000000	24.0855370	19.4070225	23.7422867	24.0684185	24.0848560
3.2000000	28.7325306	22.6884251	28.2855892	28.7102280	28.7316437

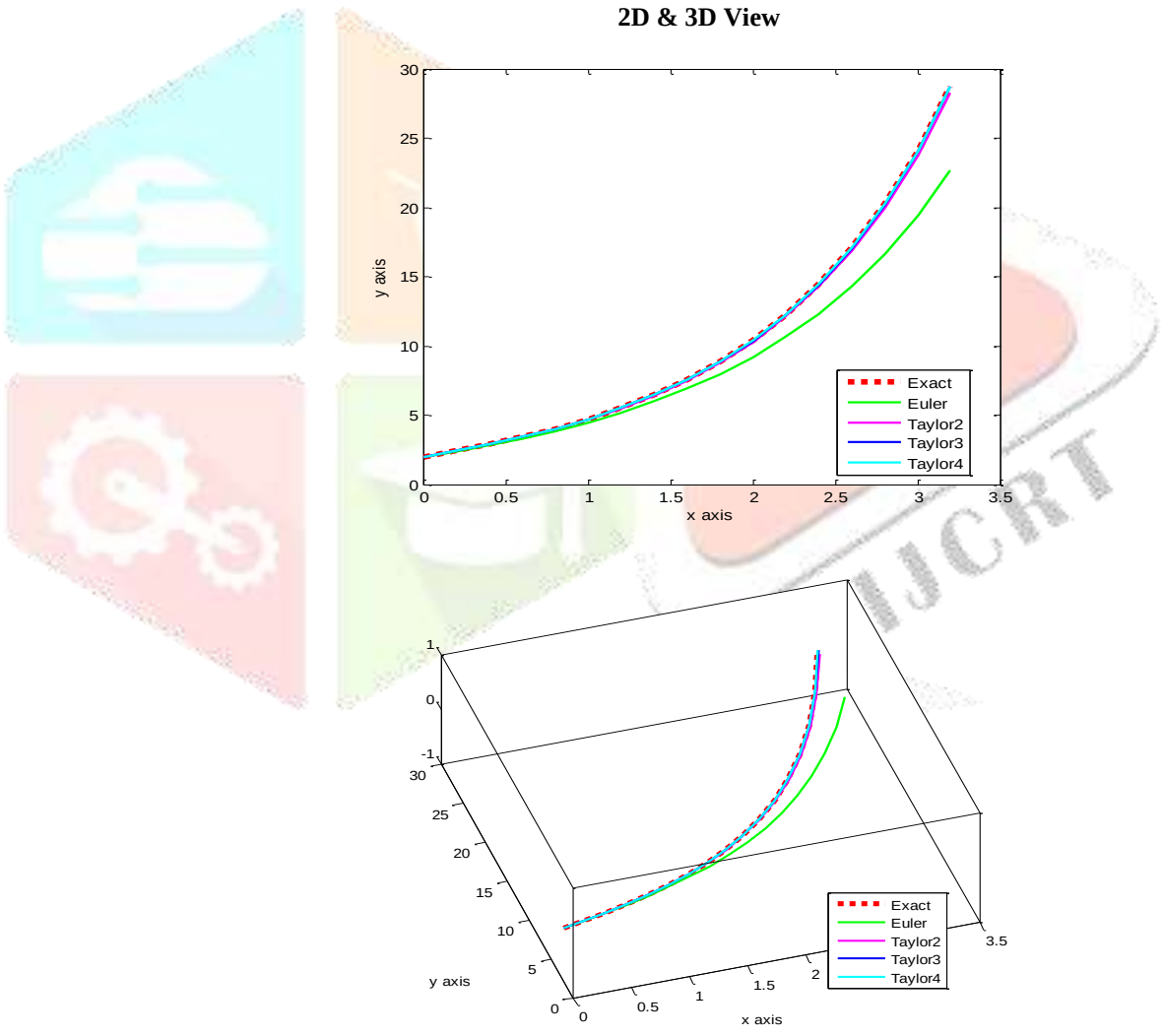
Table-2(a,b) lists the numerical and analytical values of $y = 1 + x + e^x$ for $0 \le x \le 3.2$ and here the values for each x as like as Euler< (Exact ≈ Taylor4) < Taylor3 < Taylor2.

The table-2(a,b) shows the errors of Euler's ,Taylor high order methods with exact method. These errors values for each x are in the orderTaylor4 < Taylor3 < Taylor2 < Euler.

x-value	Euler	Taylor2	Taylor3	Taylor4
0	0	0	0	0
0.2000000	0.0214028	0.0014028	0.0000694	0.0000028
0.4000000	0.0518247	0.0034247	0.0001696	0.0000067
0.6000000	0.0941188	0.0062708	0.0003107	0.0000123
0.8000000	0.1519409	0.0102064	0.0005060	0.0000201
1.0000000	0.2299618	0.0155737	0.0007725	0.0000307
1.2000000	0.3341329	0.0228130	0.0011321	0.0000450
1.4000000	0.4720192	0.0324891	0.0016132	0.0000641
1.6000000	0.6532155	0.0453252	0.0022518	0.0000895
1.8000000	0.8898671	0.0622447	0.0030941	0.0001230
2.0000000	1.1973196	0.0844247	0.0041989	0.0001669
2.2000000	1.5949298	0.1133632	0.0056412	0.0002242
2.4000001	2.1070759	0.1509630	0.0075164	0.0002987
2.5999999	2.7644174	0.1996377	0.0099453	0.0003952
2.8000000	3.6054621	0.2624443	0.0130812	0.0005199
3.0000000	4.6785154	0.3432500	0.0171182	0.0006803

3.2000000	6.0441041	0.4469401	0.0223015	0.0008864

Figure-2 (i),(ii) shows the function $y = 1 + x + e^x$ on the interval $[0, 3.2]$ using MATLAB.where red, green, magenta, blue, cyan colour curves denote Exact and Taylor's of order 2,3,4 curves respectively. The green curve Euler curve is below to the other curves and the magenta colour Taylor2 curve is above to all curves but the blue colour Taylor3 is in-between Taylor2 and Taylor4. Here, The Taylor4th 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 already compared Euler method, Taylor order2, Taylor order3 , Taylor order4 with Exact method in this paper.We compared among them through MATLAB program

9. Conclusion

The graphs that are plotted in previous ***chapters*** (iv),(v) the results obtained from different methods using *MATLAB* consequently, we can see ***Taylor's Method*** is ***more accurate*** than Euler's Method. In particular, the accuracy of solution of Taylor's higher 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.

ACKNOWLEDGMENT

First of all, I praise and glorify the ALMIGHTY GOD for his blessings throughout my studies.I am extremely grateful and deeply indebted to my guide **Dr. S.KARUNANITHI M.Sc.,B.Ed.,M.Phil.,Ph.D.,** Assistant professor & Head, Department of Mathematics , Government Thirumagal Mills College, Gudiyattam, for his valuable guidance and constant encouragement in each and every stage of this work. I am thankful to the Prof.N.Gajalakshmi & Prof.M.Malarvizhi encouragement in each and every stage of this Government Thirumagal Mills College, Gudiyattam, for allowing me to do this paper submit successfully. Thanks to all

References

1. Richard L.Burden, J.DouglasFaires, **Numerical Analysis,** 9th edition.
2. M.K.Jain, S.R.K.Iyengar, R.K.Jain, **Numerical Methods for Scientific and Engineering Computation,** 3rd edition, New Age International(p) Limited publishers(2002).
3. Amos Gilat ,**MATLAB An Introduction with Applicatins,** 5th edition, Published by John Wiley & Sons inc, U.K.(2015).
4. John.H.Mathews, Kurtis D.Fink, **Numerical Methods Using MATLAB,** 3rd edition, HogskolenivestfoidBiblioteket-Borre.(2000)
5. JaanKiusalass, **Numerical Methods in Engineering with Python,** Cambridge University Press, New York(2005).
6. AlfioQuarteroni, Riccardo Sacco, Fausto Saleri, **Numerical Mathematics,** Springer-Verlage, New York(2000).
7. D.Barton, I.M.Willers and R.V.M.Zahar,**The Automatic solution ofOrdinary Differential Equations,** by the method of Taylor series.(1971).
8. D.Barton, I.M.Willers and R.V.M.Zahar,**Taylor series Method forOrdinary Differential Equations, an Evaluation.**(1970).
9. P.R.Vittal, **Differential Equations, Fourier and Laplace Transforms, Probability,**Margham Publications,Chennai-17(2012).
10. P.K.Gupta, Sharda Gupta, **Real Analysis,** 1st edition, Sultan Chand & Sons, New Delhi(1993).
11. S.O.Fatunla ,**Numerical Methods for Initial Value Problems in Ordinary Differential Equations,** Academic press Inc. Harcourt Brace, Jovanovich publishers, New York (1988).
12. TOM.M.Apostol, **Mathematical Analysis,** 2nd Edition , Narosa Publishing House(1985).
13. E.Harier, S.P.Norsett, G.Wanner, **Solving Ordinary Differential Equations, Nonstiff problems,** Springer-Verlag, New York(1987).
14. E.Harier, S.P.Norsett, G.Wanner, **Solving Ordinary Differential Equations, Stiff and Different Algebraic problems,** Springer-Verlag, New York(1991).
15. K.E.Brenan, S.L.Campbell, L.R.Petzold, **Numerical Solution of Initial Value Problems in Differential Algebraic Equations,** SIAM Philadelphia(1996).
16. J.C.Butcher, **Numerical Methods for Ordinary Differential Equations,** 2nd Edition, John Wiley & Sons Ltd.
17. T.E.Hull, W.H.Enright, B.M.Fellen and A.E.Sedgwick, **Comparing Numerical Methods for Ordinary Differential Equations,** SIAM Journal on Numerical Analysis(1972).
18. G.G.Dahlquist, **Numerical Methods ,** Prentice –Hall Inc,(1974).
19. J.Nocedal and S.wrights, **Numerical Optimization,** Springer Verlag (1999).
20. A.Iserles, **A First Course in the Numerical Analysis of Differential Equations,** Cambridge university press(1996).
21. B.S.Grawal and J.S.Grawal , **Numerical Methods in Engineering and Science,** 6th Edition, Khanna Publishers, New Delhi (2004).
22. K.Atkinson, **Elementary Numerical Analysis,** Wiley, New York,(1985).
23. C.W.Gear, **Numerical Initial Value Problems in Ordinary Differential Equations,** Prentice-Hall, Englewood Cliffs, N.J.,(1971).
24. David F.Griffiths, **An Introduction to Matlab Version 3.1,** University of Dundee(1996).