International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 4 Ver. III || April 2018 || PP 25-31

A Numerical Solutions Of Initial Value Problems (Ivp) For Ordinary Differential Equations (Ode) With Euler And Higher Order Of Runge Kutta Methods Using Matlab

C.Senthilnathan[1]
[1](PG & Research Department Of Mathematics,G.T.M.College,Gudiyattam,Vellore Dist,Tamilnadu,India)

Abstract : *This Paper Mainly Presents Euler Method And 4thorder Runge Kutta Method (RK4) 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 Accuracy, 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 Runge Kutta Method Have Been Presented Using Matlab. 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.*

Keywords *-Initial Value Problem (IVP), Euler Method, Higher Order Of Runge Kutta Method, Error Analysis, Matlab Software.*

Date of Submission: 31-03-2018 Date of acceptance: 16-04-2018

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 Method 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: Error Analysis; Section 4: Numerical Examples; Section 5: Discussion Of Results; And The Last Section: The Conclusion Of The Paper.

II. 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 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.

2.1. Euler Method

Euler's Method Is Also Called Tangent Line Method And Is The Simplest Numerical Method For Solving Initial Value Problem In Ordinary Differential Equation, Particularly Suitable For Quick Programming Which Was Originated By Leonhard Euler In 1768. This Method Subdivided Into Three Namely:

- Forward Euler's Method.
- Improved Euler's Method.
- Backward Euler's Method.

In This Work We Shall Only Consider Forward Euler's 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.

Euler's Method Is The Simplest One-Step Method. It Is Basic Explicit Method For Numerical Integration Of Ordinary Differential Equations. Euler Proposed His Method For Initial Value Problems (IVP) In 1768. It Is First Numerical Method For Solving IVP And Serves To Illustrate The Concepts Involved In The Advanced Methods. It Is Important To Study Because The Error Analysis Is Easier To Understand. The General Formula For Euler Approximation Is

$$\boldsymbol{y_{n+1}(x) = y_n(x) + hf(x_n, y_n), n = 0, 1, 2, 3, \dots .}$$

2.2. Higher 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 2

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2)$$

Where ,$k_1 = hf(x, y); k_2 = hf\left(x + \frac{h}{2}, y + \frac{k_1}{2}\right) and\ \Delta y = k_2$

Runge Kutta 'S Method Of Order 3

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3)$$

Where,

$$k_1 = hf(x, y); k_2 = hf\left(x + \frac{h}{2}, y + \frac{k_1}{2}\right); k_3 = hf(x + h), y + 2k_2 - k_1); \Delta y = \frac{1}{6}[k_1 + 4k_2 + k_3]$$

Runge Kutta 'S Method Of Order 4

$$y_{n+1}(x) = y_n(x) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4), n = 0,1,2,3, \dots$$

Where ,

$$k_1 = hf(x_n, y_n), k_2 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right), k_3 = hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right), k_4 = hf(x_n + h, y_n + k_3),$$

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

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

III. 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 Paper 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

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

IV. 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 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.

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.

V. 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 Exact 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 ≈ Runge Kutta-4) < Runge Kutta-2 < Runge Kutta-3.

X-Value	Exact	Euler	Runge-2	Runge-3	Runge-4
0	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000
0.2000000	0.8292986	0.8000000	0.8260000	0.8342000	0.8292933
0.4000000	1.2140876	1.1520000	1.2069200	1.2259636	1.2140762
0.6000000	1.6489406	1.5504000	1.6372424	1.6702501	1.6489220
0.8000000	2.1272295	1.9884800	2.1102357	2.1608863	2.1272027
1.0000000	2.6408591	2.4581759	2.6176877	2.6903121	2.6408226
1.2000000	3.1799417	2.9498112	3.1495788	3.2492688	3.1798942
1.4000000	3.7323999	3.4517734	3.6936862	3.8264179	3.7323401
1.6000000	4.2834840	3.9501281	4.2350974	4.4078732	4.2834096
1.8000000	4.8151765	4.4281540	4.7556186	4.9766288	4.8150859
2.0000000	5.3054719	4.8657846	5.2330546	5.5118580	5.3053632
2.2000000	5.7274933	5.2389412	5.6403265	5.9880552	5.7273636
2.4000001	6.0484118	5.5187297	5.9443984	6.3739848	6.0482593
2.5999999	6.2281308	5.6704755	6.1049662	6.6313934	6.2279534
2.8000000	6.2176766	5.6525707	6.0728588	6.7134333	6.2174716
3.0000000	5.9572315	5.4150848	5.7880878	6.5627313	5.9569969
3.2000000	5.3737350	4.8981018	5.1774669	6.1090250	5.3734694

The Table 1 (B)Shows The Errors Of Euler's ,Runge-Kutta High Ordermethods With Exact Method. These Error Values For Each X Are In The Orderrunge Kutta-4 < Runge Kutta-2 < Euler < Runge Kutta-3.

X-Value	Euler	Runge-2	Runge-3	Runge-4
0	0	0	0	0
0.2000000	0.0292986	0.0032986	0.0049014	0.0000053
0.4000000	0.0620877	0.0071677	0.0118759	0.0000114
0.6000000	0.0985406	0.0116982	0.0213095	0.0000186
0.8000000	0.1387495	0.0169938	0.0336567	0.0000269
1.0000000	0.1826831	0.0231715	0.0494530	0.0000364
1.2000000	0.2301303	0.0303627	0.0693273	0.0000474
1.4000000	0.2806266	0.0387138	0.0940179	0.0000599
1.6000000	0.3333557	0.0483866	0.1243893	0.0000743
1.8000000	0.3870225	0.0595577	0.1614524	0.0000906
2.0000000	0.4396875	0.0724173	0.2063859	0.0001089
2.2000000	0.4885519	0.0871666	0.2605620	0.0001295
2.4000001	0.5296821	0.1040133	0.3255732	0.0001525
2.5999999	0.5576553	0.1231648	0.4032626	0.0001777
2.8000000	0.5651059	0.1448179	0.4957567	0.0002051
3.0000000	0.5421466	0.1691439	0.6054999	0.0002345
3.2000000	0.4756330	0.1962679	0.7352902	0.0002654

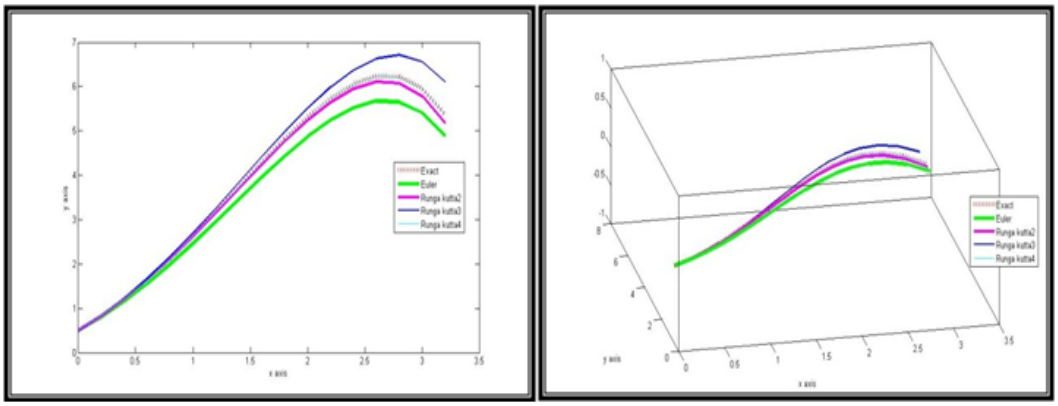


Figure-1(A),(B) Shows The Function $y=(1+x)^2-0.5e^x$ On The Interval $[0,3.2]$ Using MATLAB. Where Red, Green, Magenta, Blue, Cyan Colour Curves Denote Exact And Runge Kutta's Of Order 2,3,4 Curves Respectively. The Green Curve Euler Curve Is Below To The Other Curves And The Magenta Colour Runge Kutta-2 Curve Is Above To All Curves But The Blue Colour Runge Kutta-3 Is In Between Runge Kutta-2 And Runge Kutta-4. Here, The Runge Kutta 4th Curve Is More Nearest To Exact Value.

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)**

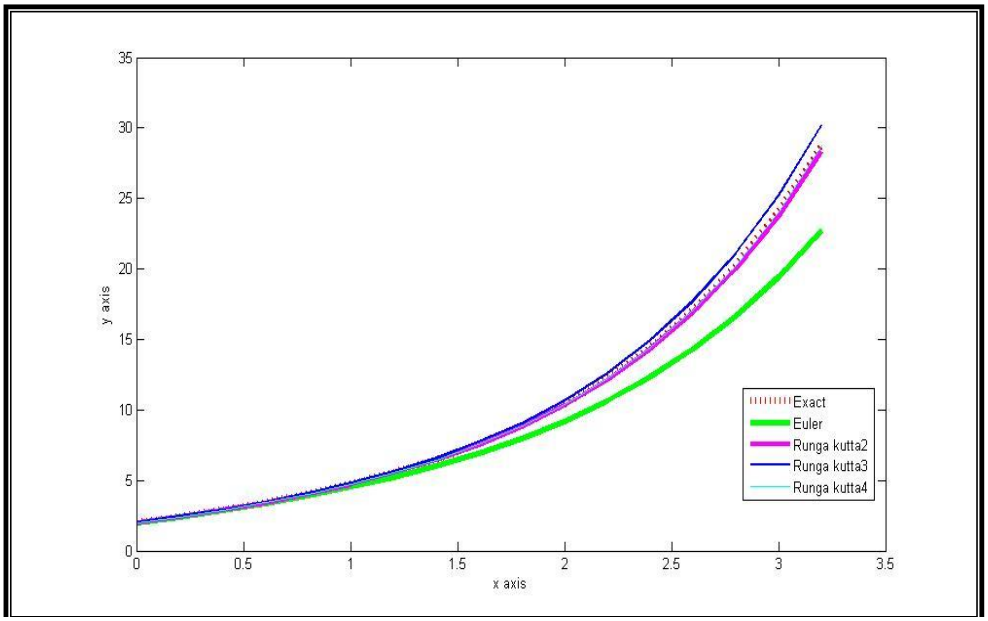
Table :2(A) Lists The Numerical And Exact Values Of $y=1+x+e^x$ For $0\le x\le 3.2$ And Here The Values For Each X As Like As Euler < (Exact ≈ Runge Kutta-4) < Runge Kutta-2 < Runge Kutta-3.

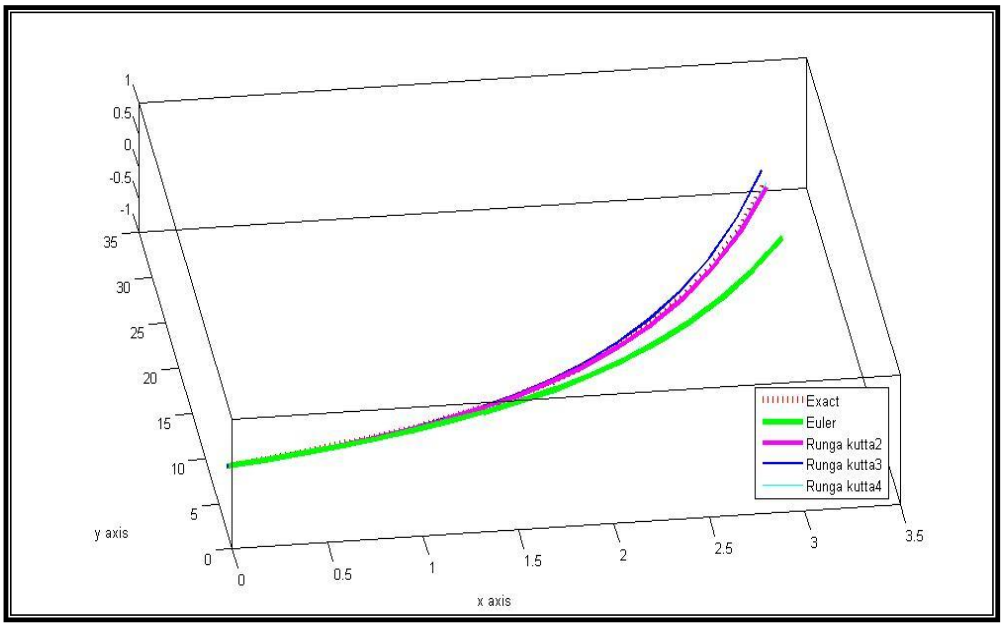
X-Value	Exact	Euler	Runge-2	Runge-3	Runge-4
0	2.0000000	2.0000000	2.0000000	2.0000000	2.0000000
0.2000000	2.4214027	2.4000001	2.4200001	2.4280000	2.4214001
0.4000000	2.8918247	2.8399999	2.8884001	2.9072239	2.8918180
0.6000000	3.4221189	3.3280001	3.4158480	3.4491804	3.4221065
0.8000000	4.0255408	3.8736000	4.0153346	4.0679626	4.0255208
1.0000000	4.7182817	4.4883199	4.7027082	4.7808318	4.7182512
1.2000000	5.5201168	5.1859841	5.4973040	5.6089253	5.5200720
1.4000000	6.4552002	5.9831810	6.4227109	6.5781307	6.4551358
1.6000000	7.5530324	6.8998170	7.5077071	7.7201505	7.5529428
1.8000000	8.8496475	7.9597802	8.7874031	9.0738115	8.8495245
2.0000000	10.389056 2	9.1917362	10.3046312	10.6866608	10.3888893
2.2000000	12.225013 7	10.630840	12.1116505	12.6169300	12.2247896
2.4000001	14.423176 8	12.3161001	14.2722130	14.9359341	14.4228773
2.5999999	17.063737 9	14.2993202	16.8640995	17.7310066	17.0633430
2.8000000	20.244646 1	16.6391850	19.9822025	21.1091080	20.2441273
3.0000000	24.085537 0	19.4070225	23.7422867	25.2012196	24.0848560
3.2000000	28.732530 6	22.6884251	28.2855892	30.1677608	28.7316437

Table :2(B) Shows The Errors Of Euler's ,Runge Kutta High Order Methods With Exact Method. These Errors Values For Each X Are In The Order Runge Kutta-4 < Runge Kutta-2 < Runge Kutta-3 < Euler.

X-Value	Euler	Runge-2	Runge-3	Runge-4
0	0	0	0	0
0.2000000	0.0214028	0.0014028	0.0065972	0.0000028
0.4000000	0.0518247	0.0034247	0.0153993	0.0000067
0.6000000	0.0941188	0.0062708	0.0270615	0.0000123
0.8000000	0.1519409	0.0102064	0.0424219	0.0000201
1.0000000	0.2299618	0.0155737	0.0625500	0.0000307
1.2000000	0.3341329	0.0228130	0.0888084	0.0000450
1.4000000	0.4720192	0.0324891	0.1229306	0.0000641
1.6000000	0.6532155	0.0453252	0.1671182	0.0000895
1.8000000	0.8898671	0.0622447	0.2241637	0.0001230
2.0000000	1.1973196	0.0844247	0.2976046	0.0001669
2.2000000	1.5949298	0.1133632	0.3919170	0.0002242
2.4000001	2.1070759	0.1509630	0.5127578	0.0002987
2.5999999	2.7644174	0.1996377	0.6672693	0.0003952
2.8000000	3.6054621	0.2624443	0.8644602	0.0005199
3.0000000	4.6785154	0.3432500	1.1156828	0.0006803
3.2000000	6.0441041	0.4469401	1.4352303	0.0008864

Figure-2(A,B) 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 Runge Kutta'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 Runge Kutta-3 Is In Between Runge Kutta-2 And Runge Kutta-4. Here, The Runge Kutta-4th Curve Is More Nearest To Exact Value.





VI. Discussion Of Results

The Runge Kutta 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, Runge Kutta Order2, Runge Kutta Order3 , Runge Kutta Order4 With Exact Method . We Compared Among Them Through MATLAB Program.

VII. Conclusion

The Graphs That Are Plotted The Results Obtained From Different Methods Using MATLAB Consequently, We Can See **Runge Kutta's 4th Order Method** Is **More Accurate** Than Euler's Method. In Particular, The Accuracy Of Solution Of Runge Kutta's 4th Order Method Is**nearer** To That Of The Exact Solution And **Error** Is Also **Very Less** For 4th Order When Compared To Lower Orders And Euler Method.

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 Kutta Method. *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 ValueProblems 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.

C.Senthilnathan"A Numerical Solutions Of Initial Value Problems (Ivp) For Ordinary Differential Equations (Ode) With Euler And Higher Order Of Runge Kutta Methods Using Matlab"International Journal of Engineering Science Invention (IJESI), vol. 07, no. 04, 2018, pp 25-31