Pure and Applied Mathematics Journal
2020; 9(2): 32-36
http://www.sciencepublishinggroup.com/j/pamj
doi: 10.11648/j.pamj.20200902.11
ISSN: 2326-9790 (Print); ISSN: 2326-9812 (Online)

Comparison of Numerical Methods for System of First Order Ordinary Differential Equations

Jemal Demsie Abraha

Depatiment of Mthematics, Wolaita Sodo University, Wolaita Sodo, Ethiopia

Email address:
jemaldemsie@gmail.com

To cite this article:
Jemal Demsie Abraha. Comparison of Numerical Methods for System of First Order Ordinary Differential Equations. *Pure and Applied Mathematics Journal*. Vol. 9, No. 2, 2020, pp. 32-36. doi: 10.11648/j.pamj.20200902.11

Received: January 17, 2020; **Accepted:** February 26, 2020; **Published:** April 14, 2020

Abstract: In this paper three numerical methods are discussed to find the approximate solutions of a systems of first order ordinary differential equations. Those are Classical Runge-Kutta method, Modified Euler method and Euler method. For each methods formulas are developed for n systems of ordinary differential equations. The formulas explained by these methods are demonstrated by examples to identify the most accurate numerical methods. By comparing the analytical solution of the dependent variables with the approximate solution, absolute errors are calculated. The resulting value indicates that classical fourth order Runge-Kutta method offers most closet values with the computed analytical values. Finally from the results the classical fourth order is more efficient method to find the approximate solutions of the systems of ordinary differential equations.

Keywords: Euler Method, Modified Euler Method, Runge-Kutta Method, System of Ordinary Differential Equations

1. Introduction

Many physical phenomenon in sciences and engineering are modeled by using a systems of n first order ordinary differential equations defined by [1, 2, 3] as

$$\begin{aligned}\frac{dy_1}{dx} &= f_1(x, y_1, y_2, ..., y_n)\\ \frac{dy_2}{dx} &= f_2(x, y_1, y_2, ..., y_n)\\ &.\\ &.\\ &.\\ \frac{dy_n}{dx} &= f_n(x, y_1, y_2, ..., y_n)\end{aligned} \tag{1}$$

Where each equation represents the first derivative of each unknown functions as a mapping depending on the independent variable x, and n unknown functions $f_1, f_2, ..., f_n$ and the initial conditions $f_1(x_0), f_2(x_0), ..., f_n(x_0)$ are prescribed.

The comparison between a domain decomposition method and Runge Kutta methods for system of ordinary differential equation was analysed by [4]. An n^{th} order initial value problems could be also reduced to a systems of n first order ordinary differential equations discussed by [5, 6]. A second order initial value problem is reduced to two first order systems and solved by fourth order and Butcher's fifth order Runge Kutta Methods [7].

The main purpose of this paper is to compare the numerical methods by obtaining the approximate solutions of a systems of first order ordinary differential equations. Section 2, deals on the detail discussion and derivation of numerical methods. Section 3, emphasizes on the computational aspects. And finally a conclusion is given in the last section.

2. Methods

2.1. Euler's Method

An Euler method for a single equation is elaborated and explained in [2, 3, 5, 6, 8, 9, 10, 11] as

$$y_{i+1} = y_i + hf(x_i, y_i), i = 0, 1, ..., n \tag{2}$$

From (2) it could be deduced the iterative formula for systems of n differential equations and has the following form

$$\begin{aligned}
y_{1,i+1} &= y_{1,i} + hf_1(x_i, y_{1,i}, y_{2,i}....y_{n,i}) \\
y_{2,i+1} &= y_{2,i} + hf_2(x_i, y_{1,i}, y_{2,i}....y_{n,i}) \\
&. \\
&. \\
&. \\
y_{n,i+1} &= y_{n,i} + hf_n(x_i, y_{1,i}, y_{2,i}....y_{n,i})
\end{aligned} \tag{3}$$

That is if we know the entire vector y of the unknown functions at the point $x = x_i$, then we can find the entire vector of unknown functions at the next point $x_{i+1} = x_i + h$ by means of (3).

That is

$$y_i(x_{i+1}) = y_i(x_i) + hf(x_i, y_1(x_i), y_2(x_i), ..., y_i(x_i))$$

for

$$i = 1, 2, ..., n$$

From (3) for each iteration it need the computation of n equations.

For demonstration purpose consider a systems of two ordinary differential equations, then Euler methods results the following

$$\begin{aligned}
y_{1,i+1} &= y_{1,i} + hf_1(x_i, y_{1,i}, y_{2,i}) \\
y_{2,i+1} &= y_{2,i} + hf_2(x_i, y_{1,i}, y_{2,i})
\end{aligned}$$

at each step we compute the vector of approximate values of the two unknown functions from the corresponding vector at the immediately preceding step.

2.2. Modified Euler's Method

A modified Euler method for a single ordinary differential equation is discussed and implemented by the authors [2, 3, 5, 6, 10, 12] and given by

$$y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2)h \tag{4}$$

where

$$\begin{aligned}
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + h, y_i + k_1 h)
\end{aligned}$$

Extending (4) for a systems of n differential equations we obtained

$$\begin{aligned}
y_{1,i+1} &= y_{1,i} + \frac{1}{2}(k_{1,1} + k_{1,2})h \\
y_{2,i+1} &= y_{2,i} + \frac{1}{2}(k_{2,1} + k_{2,2})h \\
&. \\
&. \\
&. \\
y_{n,i+1} &= y_{n,i} + \frac{1}{2}(k_{n,1} + k_{n,2})h
\end{aligned} \tag{5}$$

where

$$\begin{aligned}
k_{1,1} &= f_1(x_i, y_{1,i}, ..., y_{n,i}) \\
k_{2,1} &= f_2(x_i, y_{1,i}, ..., y_{n,i}) \\
&. \\
&. \\
&. \\
k_{n,1} &= f_n(x_i, y_{1,i}, ..., y_{n,i})
\end{aligned} \tag{6}$$

and

$$\begin{aligned}
k_{1,2} &= f_1(x_i + h, y_{1,i} + k_{1,1}h, ..., y_{n,i} + k_{n,1}h) \\
k_{2,2} &= f_2(x_i + h, y_{1,i} + k_{1,1}h, ..., y_{n,i} + k_{n,1}h) \\
&. \\
&. \\
&. \\
k_{n,2} &= f_n(x_i + h, y_{1,i} + k_{1,1}h, ..., y_{n,i} + k_{n,1}h)
\end{aligned} \tag{7}$$

As seen from (5), (6) and (7) we observed that the computation for every iteration needs in solving 3n equations.

As similar to Euler's method,when we performed for two systems of ordinary differential equations and we have got

$$\begin{aligned}
y_{1,i+1} &= y_{1,i} + \frac{1}{2}(k_{1,1} + k_{1,2}) \\
y_{2,i+1} &= y_{2,i} + \frac{1}{2}(k_{2,1} + k_{2,2})
\end{aligned}$$

where

$$\begin{aligned}
k_{1,1} &= f_1(x_i, y_{1,i}, y_{2,i}) \\
k_{2,1} &= f_2(x_i, y_{1,i}, y_{2,i})
\end{aligned}$$

and

$$\begin{aligned}
k_{1,2} &= f_1(x_i + h, y_{1,i} + k_{1,1}h, y_{2,i} + k_{2,1}h) \\
k_{2,2} &= f_2(x_i + h, y_{1,i} + k_{1,1}h, y_{2,i} + k_{2,1}h)
\end{aligned}$$

$k_{1,1}$ and $k_{1,2}$ are needed to compute $y_{1,i+1}$ and similarly $k_{2,1}$ and $k_{2,2}$ are needed to evaluate $y_{2,i+1}$.

2.3. Classical Fourth-Order Runge-Kutta Method

The classical fourth-order Runge-Kutta method for solving a single differential equation is presented by the authors [2, 3, 4, 5, 6, 7, 10, 11, 12, 13] as a form of

$$y_{i+1} = y_i + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4)h$$

where

$$\begin{aligned}
m_1 &= f(x_i, y_i) \\
m_2 &= f(x_i + \frac{h}{2}, y_i + \frac{m_1}{2}h) \\
m_3 &= f(x_i + \frac{h}{2}, y_i + \frac{m_2}{2}h) \\
m_4 &= f(x_i + h, y_i + m_3 h)
\end{aligned} \tag{8}$$

By applying a slight modification on (8) the iterative formulae are derived for a systems of n differential equations as

$$y_{1,i+1} = y_{1,i} + \frac{1}{6}(m_{1,1} + 2m_{1,2} + 2m_{1,3} + m_{1,4})h$$
$$y_{2,i+1} = y_{2,i} + \frac{1}{6}(m_{2,1} + 2m_{2,2} + 2m_{2,3} + m_{2,4})h$$
$$.$$
$$.$$
$$.$$
$$y_{n,i+1} = y_{n,i} + \frac{1}{6}(m_{n,1} + 2m_{n,2} + 2m_{n,3} + m_{n,4})h \quad (9)$$

Where

$$m_{1,1} = f_1(x_i, y_{1,i}, ..., y_{n,i})$$
$$m_{2,1} = f_2(x_i, y_{1,i}, ..., y_{n,i})$$
$$.$$
$$.$$
$$.$$
$$m_{n,1} = f_n(x_i, y_{1,i}, ..., y_{n,i}) \quad (10)$$

and

$$m_{1,2} = f_1(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,1}}{2}h, ..., y_{n,i} + \frac{m_{n,1}}{2}h)$$
$$m_{2,2} = f_2(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,1}}{2}h, ..., y_{n,i} + \frac{m_{n,1}}{2}h)$$
$$.$$
$$.$$
$$.$$
$$m_{n,2} = f_n(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,1}}{2}h, ..., y_{n,i} + \frac{m_{n,1}}{2}h) \quad (11)$$

and

$$m_{1,3} = f_1(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,2}}{2}h, ..., y_{n,i} + \frac{m_{n,2}}{2}h)$$
$$m_{2,3} = f_2(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,2}}{2}h, ..., y_{n,i} + \frac{m_{n,2}}{2}h)$$
$$.$$
$$.$$
$$.$$
$$m_{n,3} = f_n(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,2}}{2}h, ..., y_{n,i} + \frac{m_{n,2}}{2}h) \quad (12)$$

Finally

$$m_{1,4} = f_1(x_i + h, y_{1,i} + m_{1,3}h, ..., y_{n,i} + m_{n,3}h)$$
$$m_{2,4} = f_2(x_i + h, y_{1,i} + m_{1,3}h, ..., y_{n,i} + m_{n,3}h)$$
$$.$$
$$.$$
$$.$$
$$m_{n,4} = f_n(x_i + h, y_{1,i} + m_{1,3}h, ..., y_{n,i} + m_{n,3}h) \quad (13)$$

From (9), (10), (11), (12) and (13) Classical Runge-Kutta requires solving 5n equations for a single iteration.

To make clear the above formulae for n systems are presented for a systems of two ordinary differential equations

$$y_{1,i+1} = y_{1,i} + \frac{1}{6}(m_{1,1} + 2m_{1,2} + 2m_{1,3} + m_{1,4})h$$
$$y_{2,i+1} = y_{2,i} + \frac{1}{6}(m_{2,1} + 2m_{2,2} + 2m_{2,3} + m_{2,4})h$$

where

$$m_{1,1} = f_1(x_i, y_{1,i}, y_{2,i})$$
$$m_{2,1} = f_2(x_i, y_{1,i}, y_{2,i})$$

and

$$m_{1,2} = f_1(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,1}}{2}h, y_{2,i} + \frac{m_{2,1}}{2}h)$$
$$m_{2,2} = f_2(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,1}}{2}h, y_{2,i} + \frac{m_{2,1}}{2}h)$$

and

$$m_{1,3} = f_1(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,2}}{2}h, y_{2,i} + \frac{m_{2,2}}{2}h)$$
$$m_{2,3} = f_2(x_i + \frac{h}{2}, y_{1,i} + \frac{m_{1,2}}{2}h, y_{2,i} + \frac{m_{2,2}}{2}h)$$

and finally

$$m_{1,4} = f_1(x_i + h, y_{1,i} + m_{1,3}h, y_{2,i} + m_{2,3}h)$$
$$m_{2,4} = f_2(x_i + h, y_{1,i} + m_{1,3}h, y_{2,i} + m_{2,3}h)$$

3. Computational Aspects

A test problem is chosen to numerically validate the comparison of the methods to solve systems of ordinary differential equations. For this purpose I used a systems of two first order ordinary differential equation of the form:

$$y_1' = y_1 + 3y_2$$
$$y_2' = 2y_1 + 2y_2$$

with initial conditions $y_1(0) = 5$ and $y_2(0) = 0$. And the analytical solution of this systems is $y_1(x) = 3e^{-x} + 2e^{4x}$ and $y_2(x) = -2e^{-x} + 2e^{4x}$.

Solving this systems numerically by the above methods for some specific step size h = 0.05 for the independent variable x and applying Matlab codes described by the authors [3, 14, 15] leads the following tables of results for each dependent variable y_1 and y_2.

Table 1. *Numerical test results for the described techniques to evaluate* y_1.

Xi for h=0.05	y_1 by Euler	y_1 by Modified Euler	y_1 by RK4	y_1 Exact	Absolute error by Euler	Absolute error by modified Euler	Absolute error by RK4
0.0000	5.0000	5.0000	5.0000	5.0000	0.0000	0.0000	0.00000
0.0500	5.2500	5.2938	5.2965	5.2965	0.0465	0.0027	0.0000
0.1000	5.5875	5.6914	5.6981	5.6982	0.1107	0.0068	0.0001
0.1500	6.0281	6.2140	6.2263	6.2264	0.1983	0.0124	0.0001
0.2000	6.5907	6.8871	6.9072	6.9073	0.3166	0.0202	0.0001
0.2500	7.2980	7.7421	7.7729	7.7730	0.4750	0.0309	0.0001
0.3000	8.1772	8.8174	8.8626	8.8627	0.6855	0.0453	0.0001
0.3500	9.2614	10.1598	10.2243	10.2245	0.9631	0.0647	0.0002
0.4000	10.5899	11.8267	11.9168	11.9170	1.3271	0.0903	0.0002
0.4500	12.2103	13.8881	14.0119	14.0122	1.8019	0.1241	0.0003
0.5000	14.1797	16.4292;	16.5974	16.5977	2.4180	0.1685	0.0003
0.5500	16.5666	19.5546	19.7804	19.7809	3.2143	0.2263	0.0005
0.6000	19.4533	23.3913	23.6922	23.6928	4.2395	0.3015	0.0006
0.6500	22.9387	28.0948	28.4928	28.4936	5.5549	0.3988	0.0008
0.7000	27.1414	33.2546	34.3780	34.3790	7.2376	0.5244	0.001
0.7500	32.2039	40.9021	41.5868	41.5882	9.3843	1.0686	0.0014
0.8000	38.2972	49,5196	50.4113	50.4130	12.1158	0.8934	0.0017
0.8500	45.6266	60.0516	61.2081	61.2104	15.5838	1.1588	0.0023
0.9000	54.4383	72.9182	74.4132	74.4162	19.9779	1.4980	0.0030
0.9500	65.0281	88.6322	90.5588	90.5626	25.5345	1.9304	0.0038
1.0000	77.7507	107.8194	110.2950	110.2999	32.5492	2.4805	0.0049

Table 2. *Numerical test results for the described techniques to evaluate* y_2.

Xi for h=0.05	y_2 by Euler	y_2 by Modified Euler	y_2 by RK4	y_2 Exact	Absolute error by Euler	Absolute error by modified Euler	Absolute error by RK4
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0500	0.5000	0.5375	0.5403	0.5403	0.0403	0.0028	0.0000
0.1000	1.0750	1.1670	1.1740	1.1740	0.0990	0.007	0.0000
0.1500	1.7413	1.9102	1.9228	1.9228	0.1815	0.0126	0.0000
0.2000	2.5182	2.7931	2.8136	2.8136	0.2954	0.0205	0.0000
0.2500	3.4291	3.8476	3.8789	3.8790	0.4499	0.0314	0.0001
0.3000	4.5018	5.1128	5.1585	5.1586	0.6568	0.0458	0.0001
0.3500	5.7697	6.6358	6.7009	6.7010	0.9313	0.0652	0.0001
0.4000	7.2728	8.4745	8.5652	8.5654	1.2926	0.0909	0.0002
0.4500	9.0591	10.6993	10.8238	10.8240	1.7649	0.1247	0.0002
0.5000	11.1860	13.3959	13.5647	13.5651	2.3791	0.1692	0.0004
0.5500	13.7226	16.6691	16.8957	16.8961	3.1736	0.2270	0.0004
0.6000	16.7515	20.6465	20.9481	20.9487	4,1972	0.3022	0.0006
0.6500	20.3720	25.4838	25.8826	25.8834	5.5114	0.3996	0.0008
0.7000	24.7030	31.3709	31.8951	31.8961	7.1931	0.5252	0.001
0.7500	29.8875	38.5395	39.2250	39.2263	6.3388	0.6868	0.0013
0.8000	36.0966	47.2722	48.1646	48.1664	12.0698	0.8942	0.0018
0.8500	43.5360	57.9137	59.0711	59.0734	15.5374	1.1597	0.0023
0.9000	52.4522	70.8845	72.3804	72.3833	19.9311	1.4988	0.0029
0.9500	63.1413	86.6977	88.6251	88.6289	25.4876	1.9312	0.0038
1.0000	75.9582	105.9792	108.4556	108.4605	32.5023	2.4813	0.0049

The solution for the dependent variable y_1 and y_2 are separately solved using Euler, Modified Euler and classical fourth order Runge Kutta method. By comparing the approximate result with the analytical solution and by taking small step size the absolute error is calculated. The results obtained from the two tables, reflects that the fourth order Runge- Kutta method is the best mechanism of solving system of first order ordinary differential equations using numerical method similar to that of a single ordinary differential equations.

4. Conclusion

In this paper ,three numerical methods were applied to the system of first order ordinary differential equations. Upon solving the systems by Euler method, Modified Euler method and classical fourth order Runge-Kutta methods, the error arise from each method has significant difference. Comparing among the results the accuracy of the classical fourth order Runge-Kutta method is very high and almost the same with the analytical solution. The accuracy sequence in decreasing order becomes classical fourth order,Modified Euler and Euler respectively even if the computational cost was different.

References

[1] Dennis G. Zill. A First Course in Differential Equations with Modeling Applications.Richard Stratton, USA, Tenth Edition, 2013.

[2] Richard L., J.Burden, F.Douglas. Numerical Analysis. Richard Stratton, Canada, ninth edition, 2011.

[3] Steven C. Chapra. Applied Numerical Methods with MATLAB for Engineers and Scientists. HcGraw-Hill, New York, NY10020, third edition, 2012.

[4] N. Shawagfer, D. Kaya (2004) Comparing Numerical Methods for the Solutions of Systems of Ordinary Differential Equations.Elsevier, Applied Mathematics Letters 17, pp. 323-328.

[5] S. R. K. Iyengar and R. K. Jain. Numerical Methods. New Age International (P) Limited. New Delhi-110002, 2009.

[6] Joe D. Hoffman. Numerical Methods for Engineers and Scientists. Mc Graw-Hill,Inc(New York, 1992) second edition.

[7] Md. Jahangir Hossain, Md. Shah Alam, Md. Babul Hossain (2017). A stdy on the Numerical Solutions of Second Order Initial Value Problems(IVP) for Ordinary Differential Equations with Fourth Order and Butcher's Fifth Order Runge-Kutta Mthods. American Journal of Computational and Applied Mathematics, 7 (5), pp. 129-137.

[8] A. Iserles. A First Course in the Numerical Analysis of Differential Equations, Cambridge University Press, Cambridge, United Kingdom, 1996.

[9] S. E. Fadugba and T. E. Olaosebikan (2018). Comparative Study of Class of One-Step Methods for Numerical Solution of Some Initial Value Problems in Ordinary Differential Equations. Research Journal of Mathematics and Computer Science, 2: 9, pp. 1-11.

[10] Gadamsetty Revathi (2017). Numerical Solution of Ordinary Differential equations and Applications, International Journal of Management and Applied Science, ISSN: 2394-7926 Vol 3, pp. 1-5.

[11] C. Senthilnathan(2018). Anumerical 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) ISSN (online): 2319-6734, ISSN (print): 2319-6726 Volume 7, pp. 25-31.

[12] Murad Hossen, Zain Ahmed, Rashadul Kabir, Zakir Hossan (2019). Acomparative Investigation on Numerical Solution of Initial Value Problem by Using Modiffied Euler Method and Runge Kutta Method. ISOR Journal of Mathematics (IOSR-JM)e-ISSN: 2278-5728, P-ISSN: 2319-765X. Volume 15, pp. 40-45.

[13] J. C. Butcher. The Numerical Analysis of Ordinary Differential Equations: Runge Kutta and General Linear Methods. John Wiley, New York, NY, 1987.

[14] John H. Mathews, Kurtis D. Fink. Numerical Methods Using Matlab, prentice Hall, Upper Saddle River, NJ 07458, third edition, 1999.

[15] Stephen J. Chapman. MATLAB Programming for Engineers, Thomson Learning, 2004.