IOSR Journal of Mathematics (IOSRJM)
ISSN: 2278-5728 Volume 1, Issue 3 (July-Aug 2012), PP 25-31
www.iosrjournals.org

On Some Numerical Methods for Solving Initial Value Problems in Ordinary Differential Equations

Ogunrinde R. Bosede[1], Fadugba S. Emmanuel[2], Okunlola J. Temitayo[2]
[1]*Department of Mathematical Sciences, Ekiti State University, Ado Ekiti, Ekiti State, Nigeria.*
[2]*Department of Mathematical and Physical Sciences, Afe Babalola University, Ado Ekiti, Ekiti State, Nigeria.*

Abstract: *This work presents numerical methods for solving initial value problems in ordinary differential equations. Euler's method is presented from the point of view of Taylor's algorithm which considerably simplifies the rigorous analysis while Runge Kutta method attempts to obtain greater accuracy and at the same time avoid the need for higher derivatives by evaluating the given function at selected points on each subinterval. We discuss the stability and convergence of the two methods under consideration and result obtained is compared to the exact solution. The error incurred is undertaken to determine the accuracy and consistency of the two methods.*
Key words*: Differential Equation, Error, Euler's Method, Runge Kutta Method, Stability.*

I. Introduction

Differential equations can describe nearly all system undergone change. Many mathematicians have studied the nature of these equations and many complicated systems can be described quite precisely with compact mathematical expressions. However, many systems involving differential equations are so complex. It is in these complex systems where computer simulations and numerical approximations are useful. The techniques for solving differential equations based on numerical approximations were developed before programmable computers existed. The problem of solving ordinary differential equations is classified into initial value and boundary value problems, depending on the conditions specified at the end points of the domain. There are numerous methods that produce numerical approximations to solution of initial value problems in ordinary differential equations such as Euler's method which was the oldest and simplest method originated by Leonhard Euler in 1768, Improved Euler's method and Runge Kutta methods described by Carl Runge and Martin Kutta in 1895 and 1905 respectively. There are many excellent and exhaustive texts on this subject that may be consulted, such as [8], [4], [6], [5], and [1] just to mention few. In this work we present the practical use and the convergence of Euler method and Runge Kutta method for solving initial value problems in ordinary differential equations.

II. Numerical Method

Numerical method forms an important part of solving initial value problems in ordinary differential equations, most especially in cases where there is no closed form solution. Next we present two numerical methods namely Euler's Method and Runge Kutta method.
We present here the derivation of Euler's method for generating, numerically, approximate solutions to the initial value problem [2]

$$y' = f(x, y), y(x_0) = y_0 \tag{1}$$

2.1 Runge Kutta Method

Runge Kutta method is a technique for approximating the solution of ordinary differential equation. This technique was developed around 1900 by the mathematicians Carl Runge and Wilhelm Kutta. Runge Kutta method is popular because it is efficient and used in most computer programs for differential equation.
The following are the orders of Runge Kutta Method as listed below:

- Runge Kutta method of order one is called Euler's method.
- Runge Kutta method of order two is the same as modified Euler's or Heun's Method.
- The fourth order Runge Kutta method called classical Runge Kutta method.

In this paper, we shall only consider the fourth order Runge Kutta method.

2.2 Derivation of the Fourth Order Runge Kutta Method

We shall derive here the simplest of the Runge method. A formula of the following form is sought:

$$y_{n+1} = y_n + ak_1 + bk_2 \tag{2}$$

Where $k_1 = hf(x_n, y_n)$, $k_2 = hf(x_n + \alpha h, y_n + \beta k_1)$ and a, b, α, β are constants to be determined so that (2) will agree with the Taylor algorithm. Expanding $y(x_{n+1})$ in a Taylor series of order h^3, we obtain

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{h^2 y''(x_n)}{2} + \frac{h^3 y'''(x_n)}{6} + \ldots$$

$$= y(x_n) + hf(x_n, y_n) + \frac{h^2 (f_x + ff_y)_n}{2} + \frac{h^3 (f_{xx} + 2ff_{xy} + f_{yy} f^2 + f_x f_y + f_y{}^2 f)_n}{6} + 0(h^4)$$

It should be noted that the expansions

$y' = f(x, y)$, $y'' = f_x + f_y f$ and $y''' = f_{xx} + 2f_{xy} f + f_{yy} f^2 + f_x f_y + f_y{}^2 f$. The subscript n means that all functions involved are to be evaluated at (x_n, y_n).

On the other hand, using Taylor's expansion for functions of two variables, we find that

$$k_2 = f(x_n + \alpha h, y_n + \beta k_1 = f(x_n, y_n) + \alpha h f_x + \beta k_1 f_y + \frac{\alpha^2 h^2 f_{xx}}{2} + \alpha h \beta k_1 f_{xy} + \frac{\beta^2 k_1^2 f_{yy}}{2} + 0(h^3)$$

, all the derivatives above are evaluated at (x_n, y_n). If we now substitute this expression for k_2 into (2) and note that $k_1 = hf(x_n, y_n)$, we find upon rearrangement in powers of h and by setting

$a = b = \frac{1}{2}, \alpha = \beta = 1$ that

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

Where $k_1 = hf(x_n, y_n)$, $k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_1)$,

$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{1}{2}k_2)$ and $k_4 = hf(x_n + h, y_n + k_3)$.

This method (3) is undoubtedly the most popular of all Runge Kutta methods. Indeed it is frequently referred to as "the fourth order Runge Kutta method". Many numerical analyst rely on (3), because it is quite stable, accurate and easy to program.

2.3 Error Estimate for Runge Kutta Method

For all one step methods like Runge Kutta Method, the conceptually-simplest definition of local truncation error is that it is the error committed in the most recent integration step, on a single integration step. We denote the solution to the initial value problem (1) by $x, x(0), y(0)$. We have noted that the truncation error in p^{th} order Runge Kutta method is kp^{p+1}, where k is some constant. Bounds on k for $p = 2,3,4$ also exist. The derivation of these bounds is not a simple matter and moreover, their evaluation requires some quantities. One of the serious draw backs of Runge Kutta method is error estimation.

2.4 Euler's 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.

2.5 Derivation of Euler's method

We present below the derivation of Euler's method for generating, numerically, approximate solutions to the initial value problem (1), 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 explicitly the value of $y(x_0)$, namely y_0,

using the given differential equation (1), we can also determine the instantaneous rate of change of y at point x_0

$$y'(x_0) = f(x_0, y(x_0)) = f(x_0, y_0) \tag{4}$$

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. If this is the case, then the value of $y(x)$ will remain close to $y_0 + f(x_0, y_0)(x - x_0)$ for x close to x_0, for small number h, we have

$$x_1 = x_0 + h \quad (5)$$

$$\begin{aligned} y_1 &= y_0 + f(x_0, y_0)(x_1 - x_0) \\ &= y_0 + hf(x_0, y_0) \end{aligned} \tag{6}$$

Where $h = x_1 - x_0$ and is called the step size.

By the above argument,

$$y(x_1) \approx y_1 \tag{7}$$

Repeating the above process, we have at point x_1 as follows

$$x_2 = x_1 + h \tag{8}$$

$$\begin{aligned} y_2 &= y_1 + f(x_1, y_1)(x_2 - x_1) \\ &= y_1 + hf(x_1, y_1) \end{aligned} \tag{9}$$

We have

$$y(x_2) \approx y_2 \tag{10}$$

Then define for $n = 0,1,2,3,4,5,\ldots$, we have

$$x_n = x_0 + nh \tag{11}$$

Suppose that, for some value of n, we are already computed an approximate value y_n for $y(x_n)$. Then

The rate of change of $y(x)$ for x to x_n is $f(x, y(x)) \approx f(x_n, y(x_n)) \approx f(x_n, y_n)$

where $y(x_n) = y_n + f(x_n, y_n)(x - x_n)$.

Thus,

$$y(x_{n+1}) \approx y_{n+1} = y_n + hf(x_n, y_n) \tag{12}$$

Hence,

$$y_{n+1} = y_n + hf(x_n, y_n) \tag{13}$$

Equation (13) is called Euler's method.

From (13), we have

$$\frac{y_{n+1} - y_n}{h} = f(x_n, y_n), n = 0,1,2,3,\ldots \tag{14}$$

2.6 Truncation Errors for Euler's Method

Numerical stability and errors are discussed in depth in [3] and [7]. There are two types of errors arise in numerical methods namely truncation error which arises primarily from a discretization process and round off error which arises from the finiteness of number representations in the computer. Refining a mesh to reduce the truncation error often causes the round off error to increase. To estimate the truncation error for Euler's method, we first recall Taylor's theorem with remainder, which states that a function $f(x)$ can be expanded in a series about the point $x = a$

$$\begin{aligned} f(x) &= f(a) + f'(a)(x-a) \frac{f''(a)(x-a)^2}{2!} + \ldots \\ &+ \frac{f^m(a)(x-a)^m}{m!} + \frac{f^{m+1}(\beta)(x-a)^{m+1}}{(m+1)!} \end{aligned} \tag{15}$$

The last term of (15) is referred to as the remainder term. Where $x \le \beta \le a$.

In (15), let $x = x_{n+1}$ and $x = a$, in which

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + \frac{1}{2}h^2 y''(\beta_n) \tag{16}$$

Since y satisfies the ordinary differential equation in (1), which can be written as

$$y'(x_n) = f(x_n, y(x_n)) \tag{17}$$

Hence,

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + \frac{1}{2}h^2 y''(\beta_n) \tag{18}$$

By considering (18) to Euler's approximation in (13), it is clear that Euler's method is obtained by omitting the remainder term $\frac{1}{2}h^2 y(\beta_n)$ in the Taylor expansion of $y(x_{n+1})$ at the point x_n. The omitted term accounts for the truncation error in Euler's method at each step.

2.7 Convergence of Euler's Method

The necessary and sufficient conditions for a numerical method to be convergent are stability and consistency. Stability deals with growth or decay of error as numerical computation progresses. Now we state the following theorem to discuss the convergence of Euler's method.

Theorem: If $f(x, y)$ satisfies a Lipschitz condition in y and is continuous in x for $0 \le x \le a$ and defined a sequence y_n, where $n = 1,2,...,k$ and if $y_0 \to y(0)$, then $y_n \to y(x)$ as $n \to \infty$ uniformly in x where $y(x)$ is the solution of the initial value problem (1).

Proof: we shall start the proof of the above theorem by deriving a bound for the error

$$e_n = y_n - y(x_n) \tag{19}$$

Where y_n and $y(x_n)$ are called numerical and exact values respectively. We shall also show that this bound can be made arbitrarily small. If a bound for the error depends only on the knowledge of the problem but not on its solution $y(x)$, it is called an a priori bound. If, on the other hand, knowledge of the properties of the solution is required, its error bound is referred to as an a posteriori bound.

To get an a priori bound, let us write

$$y(x_{n+1}) = y(x_n) + hf(x_n, y_n) - t_n \tag{20}$$

Where t_n is called the local truncation error. It is the amount by which the solution fails to satisfy the difference method. Subtracting (20) from (13), we get

$$e_{n+1} = e_n + h[f(x_n, y_n) - f(x_n, y(x_n))] + t_n \tag{21}$$

Let us write

$$e_n M_n = f(x_n, y_n) - f(x_n, y(x_n)) \tag{22}$$

Substituting (21) into (22), then

$$e_{n+1} = e_n(1 + hM_n) \tag{23}$$

This is the difference equation for e_n. The error e_0 is known, so it can be solved if we know M_n and t_n. We have a bound of the Lipschitz constant M for $|M_n|$. Suppose we also have $T \ge |t_n|$. Then we have

$$|e_{n+1}| \le |e_n|(1 + hM) + T \tag{24}$$

To proceed further, we need the following lemma.

Lemma**:** If $|e_n|$ satisfies (24) and $0 \le nh \le a$, then

$$|e_n| = T\frac{(1+hM)^n - 1}{hM} + (1+hM)^n |e_0| \le \frac{T}{hM}(e^{Lb} - 1) + e^{Lb}|e_0| \tag{25}$$

Lemma: The first inequality of (25) follows by induction. It is trivially true for $n = 0$. Assuming that it is true for n, we have from (24)

$$|e_{n+1}| = T\frac{(1+hM)^n - 1}{hM}(1+hM)^n + T + (1+hM)^{n+1}|e_0|$$

$$= T\frac{(1+hM)^{n+1} - (1+hM) + hM}{hM} + (1+hM)^{n+1}|e_0|$$

$$= \mathrm{T}\frac{(1+hM)^{n+1} - 1}{hM} + (1+hM)^{n+1}|e_0| \tag{26}$$

Hence (25) is true for $n+1$ and thus for all n.

The second inequality in (25) follows from the fact that $nh \le a$ and for $hM \ge 0$, $(1+hM) \le e^{Mh}$ so that $(1+hM)^n \le e^{Mnh} \le e^{Ma}$, proving the lemma.

To continue the proof of the theorem, we need to investigate T, the bound on the local truncation error.
From (20), we have

$$-t_n = y(x_{n+1}) - y(x_n) - hf(x_n, y(x_n))$$

By the Mean value theorem, we get for $0 \le \theta \le 1$,

$$\le h|f(x_n + \theta h, y(x_n)) - f(x_n, y(x_n))| + h|f(x_n + \theta h, y(x_n + \theta h)) - f(x_n + \theta h, y(x_n))|$$

$$\le h|f(x_n + \theta h, y(x_n)) - f(x_n, y(x_n))| + h|y(x_n + \theta h) - y(x_n)| \tag{27}$$

The last term can be treated by the Mean value theorem to get a bound $M\theta h^2|y'(g)| \le h^2 MZ$, where $Z = \max|y'(x)|$, the inequality exists because of the continuity of y and f in a closed region. The treatment of the first term in (27) depends on our hypothesis. If we are prepared to assume that $f(x, y)$ also satisfies a Lipschitz condition in x, we can bound the first term in (27) by $L\theta h^2$, where L is the Lipschitz constant for $f(x)$. Consequently, $|t_n| \le h^2(L + MZ) = T$ and so from (25), we get

$$|e_n| \le h\frac{L + MZ}{M}(e^{Ma} - 1) + e^{Ma}|e_0| \tag{28}$$

Thus the numerical solution converges as $h \to 0$, if $|e_0| \to 0$.

III. Numerical Experiments

In order to confirm the applicability and suitability of the methods for solution of initial value problems in ordinary differential equations, it was computerized in Fortran Programing language and implemented on a macro-computer adopting double precision arithmetic. The performance of the methods was checked by comparing their accuracy and efficiency. The efficiency was determined from the number iterations counts and number of functions evaluations per step while the accuracy is determined by the size of the discretization error estimated from the difference between the exact solution and the numerical approximations.

Example 1:

The first problem considered in this illustration is the linear first order initial value problem $y' = 1 + y^2$, $y(0) = 2$ with step size $h = 0.1$ on the interval $0 \le x \le 1$ whose exact solution is given by $y(x) = \tan x$. The results obtained shown in Table 1 and Table 2, the comparison of the methods to the exact solution and the error incurred respectively.

Example 2:

We use Euler's method to approximate the solution of the initial value problem $y' = y - x, y(0) = 2$, with step size $h = 0.1$ on the interval $0 \le x \le 1$ whose exact solution is given by $y(x) = e^x + x + 1$. The results obtained shown in Table 3 and Table 4, the comparison of the methods to the exact solution and the error incurred respectively.

IV. Table of Results

Table 1: The Comparative Result Analysis of Runge Kutta Method and Euler's Method

n	x_n	$y(x_n)$	y_{nR}	y_{nE}
0	**0.0**	**0.0000**	**0.0000**	**0.0000**
1	**0.1**	**0.1003**	**0.1003**	**0.1000**
2	**0.2**	**0.2027**	**0.2027**	**0.2010**
3	**0.3**	**0.3093**	**0.3092**	**0.3050**
4	**0.4**	**0.4227**	**0.4228**	**0.4143**
5	**0.5**	**0.5463**	**0.5463**	**0.5315**
6	**0.6**	**0.6841**	**0.6841**	**0.6598**
7	**0.7**	**0.8422**	**0.8420**	**0.8033**
8	**0.8**	**1.0296**	**1.0296**	**0.9678**
9	**0.9**	**1.2601**	**1.2600**	**1.1615**
10	**1.0**	**1.5574**	**1.5574**	**1.3964**

Table 2: Error incurred in Runge Kutta Method and Euler's Method

n	x_n	$e_{nR} = y(x_n) - y_n$	$e_{nE} = y(x_n) - y$
0	**0.0**	**0.0000**	**0.0000**
1	**0.1**	**0.0000**	**0.0003**
2	**0.2**	**0.0000**	**0.0017**
3	**0.3**	**0.0001**	**0.0043**
4	**0.4**	**0.0000**	**0.0084**
5	**0.5**	**0.0000**	**0.0148**
6	**0.6**	**0.0000**	**0.0243**
7	**0.7**	**0.0002**	**0.0389**
8	**0.8**	**0.0000**	**0.0618**
9	**0.9**	**0.0001**	**0.0986**
10	**1.0**	**0.0000**	**0.1610**

Table 3: The Comparative Result Analysis of Runge Kutta Method and Euler's Method

n	x_n	$y(x_n)$	y_{nR}	y_{nE}
0	**0.0**	**2.0000**	**2.0000**	**2.0000**
1	**0.1**	**2.2052**	**2.2051**	**2.2000**
2	**0.2**	**2.4214**	**2.4213**	**2.4100**
3	**0.3**	**2.6498**	**2.6497**	**2.6310**
4	**0.4**	**2.8918**	**2.8918**	**2.8641**
5	**0.5**	**3.1487**	**3.1486**	**3.1105**
6	**0.6**	**3.4221**	**3.4221**	**3.3716**
7	**0.7**	**3.7137**	**3.7137**	**3.6487**
8	**0.8**	**4.0255**	**4.0255**	**3.9436**
9	**0.9**	**4.3596**	**4.3596**	**4.2579**
10	**1.0**	**4.7182**	**4.7182**	**4.5937**

Table 4: Error incurred in Runge Kutta Method and Euler's Method

n	x_n	$e_{nR} = y(x_n) - y_n$	$e_{nE} = y(x_n) - y_n$
0	**0.0**	**0.0000**	**0.0000**
1	**0.1**	**0.0001**	**0.0052**
2	**0.2**	**0.0001**	**0.0114**
3	**0.3**	**0.0001**	**0.0188**
4	**0.4**	**0.0000**	**0.0277**
5	**0.5**	**0.0001**	**0.0382**
6	**0.6**	**0.0000**	**0.0505**
7	**0.7**	**0.0000**	**0.0650**
8	**0.8**	**0.0000**	**0.0819**
9	**0.9**	**0.0000**	**0.1017**
10	**1.0**	**0.0000**	**0.1245**

V. Discussion of Results

We notice that in Tables 2 and 4, the error incurred in Euler's method is greater than that of Runge Kutta method and the same time get larger as n increases. Hence Runge Kutta method is more accurate than its counterpart Euler's method as we can see from Tables 1 and 3**.**

VI. Conclusion

We have in our disposal two numerical methods for solving initial value problems in ordinary differential equations. In general, numerical method has its own advantages and disadvantages of use: Euler's method is therefore best reserved for simple preferably, recursive derivatives that can be represented by few terms. It is simple to implement and simplifies rigorous analysis. The major disadvantages of Euler method are the tiresome, sometimes impossible calculation of higher derivatives and the slow convergence of the series for some functions which involves terms of opposite sign while Runge Kutta method is a self-starting because it does not use information from previously calculated points and generally stable but error estimation remains problematic. From the problems solved using FORTRAN programing language, it is observed that a lot of useful insights into numerical solution of initial value problems have been gained. We conclude that Runge Kutta method is consistent, convergent, quite stable and more accurate than Euler's method and it is widely used in solving initial value problems in ordinary differential equations.

References

[1] W. E. Boyce and R. C. DiPrima, *Elementary differential equation and boundary value problems*, (John Wiley and Son, 2001).

[2] L. Collatz, *Numerical treatment of differential equations*, (Springer Verlag Berlin, 1960).

[3] K. Erwin *Advanced Engineering Mathematics*, Eighth Edition, (Wiley Publisher, 2003).

[4] A. Gilat, *Matlab: An Introduction with application*, (John Wiley and Sons, 2004).

[5] N. Kockler, *Numerical methods and scientific computing*, (Clarendon Press, Oxford London, 1994).

[6] J. D. Lambert, *Numerical method for ordinary systems of Initial value problems*, (John Wiley and Sons, New York, 1991).

[7] D. C. Samuel, *Elementary numerical analysis (an algorithm approach)*, Third Edition, (Mc Graw International Book Company, 1981).

[8] M. P. Stephen, *To compute numerically, concepts and strategy*, (Little Brown and Company, Canada, 1983).