The Comparative Study of the Accuracy of an Implicit Linear Multistep Method of Order Six and Classical Runge Kutta Method for the Solution of Initial Value Problems in Ordinary Differential Equations

[1]Fadugba S. Emmanuel, [2] Okunlola J. Temitayo

[1] Department of Mathematical Sciences, Ekiti State University, Ado Ekiti, Nigeria
[2] Department of Mathematical and Physical Sciences, Afe Babalola University, Ado Ekiti, Nigeria

1.0 Introduction

It has been discovered that mathematical models resulting into single or system of first order ordinary differential equations are largely applied in nearly all discipline most especially in Sciences, Engineering and Economics. Any system whose behavior can be modeled by first order ordinary differential equations can be solved numerically to any desired degree of accuracy. Numerical solution of ordinary differential equations remain an active field of investigation, though, the area of research vary significantly.

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, implicit linear multistep method of order six derived by [1] 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 [2], [5], [6], [7], and [8] just to mention few. In this work we present the practical use and the accuracy of an implicit linear multistep method and Runge Kutta method for the solution of initial value problems in ordinary differential equations.

2.0 The Methods

This section presents two numerical methods for the solution of initial value problems in ordinary differential equations.

2.1 Linear Multistep Method

Linear multistep method is a computational procedure whereby a numerical approximation x_{n+1} to the exact solution $x(t_{n+1})$ of the first order initial value problem of the form

$$x' = g\left(t, x\right), x(t_0) = x_0 \quad (1)$$

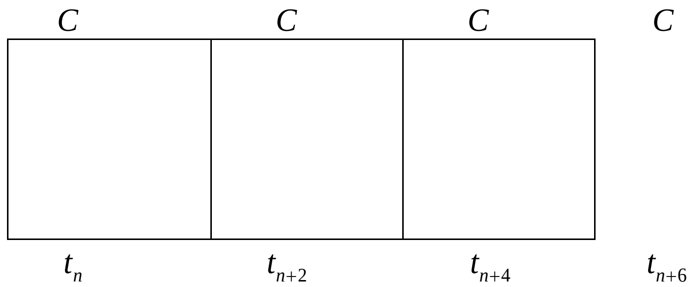
The general linear multistep method is given by [9]

$$\sum_{k=0}^{j} \alpha_k x_{n+k} = h \sum_{k=0}^{j} \beta_k g_{n+k} \quad (2)$$

Where α_k and β_k are constants, h is the step size. It is assumed that the function $g\left(t, x\right)$ is Lipschitz continuous throughout the interval $a \le t \le b$. Equation (2) includes Simpson method, Adam Bashforth and Adam Molton methods. All Adam's methods are regarded as constant coefficient method but in this paper, linear multistep method with constant coefficient of higher step number j is generated. The parameters of this method are determined by the collocation approach in which the approximate solution is determined from the condition that the equation must be stratified at certain given point. It involves the determination of an approximate solution in a suitable set of function called the basis function.

Now we shall derive an order six implicit linear multistep method for the solution of first order differential equation using collocation and interpolation methods.

Collocation points are used to collocate the different system. The interpolation points are used to interpolate the approximate solution with the diagram below. Both Collocation and interpolation are done at all even points $t = t_n, t_{n+2}$ and t_{n+4} while evaluation is done at $t = t_{n+6}$.



Next, we shall present the derivation of the scheme as follows.

2.1.1 Derivation of the Implicit Scheme [10]

The basis function is given by:

$$x(t) = \sum_{k=1}^{6} b_k t^k \qquad (3)$$

Equation (3) is needed in the derivation of the scheme for solving first order differential equation.

Expanding (3) we have,

$$x(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 + b_4 t^4 + b_5 t^5 + b_6 t^6 \qquad (4)$$

Differentiating (4) with respect to t

$$x'(t) = b_1 + 2b_2 t + 3b_3 t^2 + 4b_4 t^3 + 5b_5 t^4 + 6a_6 t^5 \qquad (5)$$

Collocating (5), we have that at $t = t_n, t_{n+2}, t_{n+4}$ and t_{n+6}

Therefore,

$$g_n = b_1 + 2b_2 t_n + 3b_3 t_n^2 + 4b_4 t_n^3 + 5b_5 t_n^4 + 6b_6 t_n^5 \qquad (6)$$

$$g_{n+2} = b_1 + 2b_2 t_{n+2} + 3b_3 t_{n+2}^2 + 4b_4 t_{n+2}^3 + 5b_5 t_{n+2}^4 + 6b_6 t_{n+2}^5 \qquad (7)$$

$$g_{n+4} = b_1 + 2b_2 t_{n+4} + 3b_3 t_{n+4}^2 + 4b_4 t_{n+4}^3 + 5b_5 t_{n+4}^4 + 6b_6 t_{n+4}^5 \qquad (8)$$

$$g_{n+6} = b_1 + 2b_2 t_{n+6} + 3b_3 t_{n+6}^2 + 4b_4 t_{n+6}^3 + 5b_5 t_{n+6}^4 + 6b_6 t_{n+6}^5 \qquad (9)$$

Interpolating at the points $t = t_n, t_{n+2}$ and t_{n+4}, then

$$x_n = b_0 + b_1 t_n + b_2 t_n^2 + b_3 t_n^3 + b_4 t_n^4 + b_5 t_n^5 + b_6 t_n^6 \qquad (10)$$

$$x_{n+2} = b_0 + b_1 t_{n+2} + b_2 t_{n+2}^2 + b_3 t_{n+2}^3 + b_4 t_{n+2}^4 + b_5 t_{n+2}^5 + b_6 t_{n+2}^6 \qquad (11)$$

$$x_{n+4} = b_0 + b_1 t_{n+4} + b_2 t_{n+4}^2 + b_3 t_{n+4}^3 + b_4 t_{n+4}^4 + b_5 t_{n+4}^5 + b_6 t_{n+4}^6 \qquad (12)$$

Using Gaussian Elimination method to determine the values of the coefficients $b_{0,} b_1, b_2, b_3, b_4, b_5$ and b_6, then we have that:

$$b_0 = x_n - b_1 t_n - b_2 t_n^2 - b_3 t_n^3 - b_4 t_n^4 - b_5 t_n^5 - b_6 t_n^6 \qquad (13)$$

$$b_1 = g_n - 2b_2 t_n - 3b_3 t_n^2 - 4b_4 t_n^3 - 5b_5 t_n^4 - 6b_6 t_n^5 \qquad (14)$$

$$b_2 = \frac{1}{4h^2}\left(x_{n+2} - x_n\right) - \frac{1}{2h} g_n - b_3\left(3t_n + 2h\right) - b_4\left(6t_n^2 + 8t_n h + 4h^2\right) - b_5\left(10t_n^3 + 20t_n^2 + 2ot_n h + 8h^3\right)$$
$$-b_6\left(15t_n^4 + 40t_n^3 h + 60t_n^2 h + 48t_n h^3 + 16h^4\right)$$
(15)

$$b_3 = \frac{1}{4h^2}\left(g_{n+2} + g_n\right) - \frac{1}{4h^3}\left(x_{n+2} - x_n\right) - b_4\left(4t_n + 4h\right) - b_5\left(10t_n^2 + 20t_n h + 12h^2\right)$$
(16)

$$b_4 = \frac{1}{64h^4}\left(x_{n+4} + 4g_{n+2} - 5x_n\right) - \frac{1}{16h^3}\left(2g_{n+2} + g_n\right) - b_5\left(5t_n + 8h\right) - b_6\left(15t_n^2 + 48t_n h + 44h^2\right)$$
(17)

$$b_5 = \frac{1}{64h^4}\left(x_{n+4} + 4g_{n+2} + g_n\right) - \frac{3}{128h^5}\left(x_{n+4} - x_n\right) - b_6\left(6t_n + 12h\right)$$
(18)

$$b_6 = \frac{1}{4224h^5}\left(g_{n+6} - 24g_{n+4} - 57g_{n+2} - 10g_n\right) + \frac{1}{2816h^5}\left(19g_{n+4} - 8g_{n+2} - 11g_n\right)$$
(19)

By inserting the coefficient b_0 into (1), evaluating at $t = t_{n+6}$, substituting equations (14), (15), (16), (17), (18) and (19) for b_1, b_2, b_3, b_4, b_5 and b_6 respectively and simplify we have the scheme:

$$x_{n+6} + \frac{27}{11}x_{n+4} - \frac{27}{11}x_{n+2} - x_n = \frac{6}{11}h\left(g_{n+6} + 9g_{n+4} + g_n\right)$$
(20)

Equation (20) is called an implicit linear multistep method of order six.

2.2 Classical 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 classical Runge Kutta method. We shall derive here the simplest of the Runge method. A formula of the following form is sought [3, 4]:

$$x_{n+1} = x_n + ak_1 + bk_2$$
(21)

Where $k_1 = hg(t_n, x_n)$, $k_2 = hg(t_n + \alpha h, x_n + \beta k_1)$ and a, b, α, β are constants to be determined so that (21) will agree with the Taylor algorithm. Expanding $x(t_{n+1})$ in a Taylor series of order h^3, we obtain

$$x(t_{n+1}) = x(t_n) + hx'(t_n) + \frac{h^2 x''(t_n)}{2} + \frac{h^3 x'''(t_n)}{6} + \ldots \quad =$$
$$x(t_n) + hg(t_n, x_n) + \frac{h^2(g_x + gg_y)_n}{2} + \frac{h^3(g_{tt} + 2gg_{tx} + g_{xx}g^2 + g_t g_x + g_x{}^2 g)_n}{6} + 0(h^4)$$

It should be noted that the expansions

$x' = g(t,x)$, $x'' = g_t + g_x g$ and $x''' = g_{tt} + 2g_{tx}g + g_{xx}g^2 + g_t g_x + g_x{}^2 g$. The subscript n means that all functions involved are to be evaluated at (t_n, x_n).

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

$$k_2 = g(t_n + \alpha h, x_n + \beta k_1) = g(t_n, x_n) + \alpha h g_t + \beta k_1 g_x + \frac{\alpha^2 h^2 g_{tt}}{2} + \alpha h \beta k_1 g_{tx} + \frac{\beta^2 k_1^2 g_{xx}}{2} + 0(h^3)$$

(22)

All the derivatives above are evaluated at (t_n, x_n). If we now substitute this expression for k_2 into (21) and note that $k_1 = hg(t_n, x_n)$, we find upon rearrangement in powers of h and by setting $a = b = \frac{1}{2}, \alpha = \beta = 1$ that

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

(23)

Where $k_1 = hg(t_n, x_n)$,

$k_2 = hg(t_n + \frac{h}{2}, x_n + \frac{1}{2}k_1)$, $k_3 = hg(t_n + \frac{h}{2}, x_n + \frac{1}{2}k_2)$ and $k_4 = hg(t_n + h, x_n + k_3)$

This method (23) is undoubtedly the most popular of all Runge Kutta methods. Indeed it is frequently referred to as "the fourth order Runge Kutta method or classical Runge Kutta Method". Many numerical analyst rely on (3), because it is quite stable, accurate and easy to program.

2.2.1 Error Estimate for Runge Kutta Method [3]

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 $t, t(0), x(0)$. 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.

3.0 Numerical Experiments

In order to confirm the suitability and applicability of the methods for the solution of initial value problem in ordinary differential equations, it was computerized in Q BASIC programming languages and implemented on a macro-computer adopting double precision arithmetic. The performance of the two methods under consideration was checked by comparing their accuracy and efficiency. Efficiency was determined from the number iterations counts and the number of the function s evaluations per step while the accuracy is determined by the size of the discretization errors estimated from the difference between the true solution and the numerical approximations.

The first order initial value problem considered in this paper is given by

$$x' = -x, x(0) = 1, t \in [0,1]$$

(24)

The true solution of equation (24) is given by

$$x(t) = \exp(-t)$$

(25)

The results obtained shown in Tables 1 and 2, the comparison of the two methods to the true solution and the error incurred respectively.

3.1 Table of Results

We present below the comparative result analysis and the error incurred from the two methods.

Table 1: Comparative Result Analysis of an Implicit Linear Multistep Method and Classical Runge Kutta Method

n	t_n	$x(t_n)$	x_{nI}	x_{nR}
0	**0.0**	**1.0000**	**1.0000**	**1.0000**
1	**0.1**	**0.9048**	**0.9047**	**0.9045**
2	**0.2**	**0.8187**	**0.8186**	**0.8185**
3	**0.3**	**0.7408**	**0.7407**	**0.7404**
4	**0.4**	**0.6703**	**0.6703**	**0.6701**
5	**0.5**	**0.6065**	**0.6065**	**0.6064**
6	**0.6**	**0.5488**	**0.5487**	**0.5485**
7	**0.7**	**0.4965**	**0.4965**	**0.4966**
8	**0.8**	**0.4493**	**0.4493**	**0.4491**
9	**0.9**	**0.4066**	**0.4067**	**0.4065**
10	**1.0**	**0.3678**	**0.3678**	**0.3679**

Table 2: Error incurred in an Implicit Linear Multistep Method and Classical Runge Kutta Method

n	t_n	$e_{nI} = \lvert x(t_n) - x_{nI} \rvert$	$e_{nR} = \lvert x(t_n) - x_{nR} \rvert$
0	**0.0**	**0.0000**	**0.0000**
1	**0.1**	**0.0001**	**0.0002**
2	**0.2**	**0.0001**	**0.0002**
3	**0.3**	**0.0001**	**0.0004**
4	**0.4**	**0.0000**	**0.0002**
5	**0.5**	**0.0000**	**0.0001**
6	**0.6**	**0.0001**	**0.0003**
7	**0.7**	**0.0000**	**0.0001**
8	**0.8**	**0.0000**	**0.0002**
9	**0.9**	**0.0001**	**0.0001**
10	**1.0**	**0.0000**	**0.0001**

3.2 Discussion of Results

As we can see from the above tables, using a step size of 0.1, the error incurred in an implicit linear multistep method is smaller than that of classical Runge Kutta method. Hence a six step implicit method is consistent and better in accuracy.

4.0 Conclusion

In this paper and two numerical methods for the solution of initial value problems in ordinary differential equations have been developed. From the two methods used, an order six implicit linear multistep method converges faster, provides the closest accurate value for the solution of any first order differential equations. Hence the method is more accurate than classical Runge Kutta method.

References

[1] D. O. Awoyemi, *On some continuous linear multistep methods for initial value problems*, Ph.D. Thesis (Unpublished), University of Ilorin, Nigeria, 1992.
[2] L. Collatz, *Numerical treatment of differential equations*, Springer Verlag Berlin, 1960.

[3] S. E Fadugba, R. B. Ogunrinde and J. T. Okunlola, On some numerical methods for solving initial value problems in ordinary differential equations, *International Organization Scientific Research*, *Journal of Mathematics, 1(3),* 2012, 25-31.

[4] S. E. Fadugba and J. T. Okunlola, On the error analysis of the new formulation of one step method into a linear multistep method for the solution of ordinary differential equations, *International Journal for Scientific and Technology Research, 1(9),* 2012, 35-37

[5] C. W. Gear, *Numerical initial value problems in ordinary differential equation*, Prentice Hall Englewood Cliffs, New Jersey, 1991.

[6] P. Henrici, *Discrete variable methods in ordinary differential equation*, John Wiley and Sons, New York, 1962.

[7] A. Gilat, Matlab: *An introduction with application*, John Wiley and Sons, 2004.

[8] N. Kockler, Numerical methods and scientific computing, Clarendon Press, Oxford London, 1994

[9] J. D. Lambert, *Computational methods in ordinary differential equation*, John Wiley and Sons, New York, 1976.

[10] J. T. Okunlola and S. E. Fadugba, On the convergence of an implicit linear multistep method of order six for the solution of ordinary differential equations, *International Journal of Advanced Research in Engineering and Applied Sciences, 1(3),* 2012, 10-15.