NUMERICAL SIMULATION STUDY ON INITIAL VALUE PROBLEM; A CASE STUDY OF GLUCONIC ACID FERMENTATION BY PSEUDOMONAS OVALIS

Abstract

In this paper, numerical solutions for a system of differential equation by using the Euler, Bogacki-Shampine and Adams-Bashforth-Moulton methods are presented. The convergences of these methods have been discussed with an example which was presented to show the ability of the methods for linear systems of differential equations. The results obtained are in good agreement with the exact solution. These results show that the techniques used here are accurate and easy to apply. The convergence of the methods under consideration and results obtained were compared to the exact solution. The error incurred is undertaken to determine the accuracy of the methods.

1.0 Introduction

All system undergone change can be described by differential equations, which can either be ordinary differential equations and partial differential equations (Constantinides and Mostoufi, 2000). 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 (Ogunrinde, et al, 2012, Constantinides and Mostoufi, 2000).

There are many excellent and exhaustive texts on this subject that may be consulted (Boyce, et al,2001, Gilat, 2004, Kockler,1994 and Stephen,1983).

The purpose of this paper is to employ the numerical methods to systems of differential equations which are often encountered in chemical engineering. A variety of methods, exact, approximate, and purely numerical are available for the solution of systems of differential

equations. Most of these methods are computationally intensive because they are trial-and-error in nature, or need complicated symbolic computations (Hassan, 2003).

In this paper, the practical use and the convergence of Euler, Bogacki-Shampine and Adams-Bashforth-Moulton methods for solving initial value problems in ordinary differential equations are presented.

2.0 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, three numerical methods are presented namely, Euler, Bogacki-Shampine and Adams-Bashforth-Moulton.

2.1 Euler's Method

In mathematics and computational science, the Euler method is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations and is the simplest Runge–Kutta method. The Euler method is named after Leonhard Euler, who treated it in his book Institutionum calculi integralis (published 1768–70). Euler's method is a numerical technique to solve ordinary differential equations of the form.

$$\frac{dy}{dx} = f(x, y), \qquad y(0), = y_0 \tag{1}$$

So only first order ordinary differential equations can be solved by using Euler's method (Autar, 2010). 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 (Ogunrinde et al, 2012).

2.2 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

$y'(x_0) = f(x_0, y(x_0)) = f(x_0, y_0)$ (2)

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

$x = x + h$ (3)

$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0)$

$= y_0 + hf(x_0, y_0)$ (4)

Where

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

By the above argument,

$y(x_1) \approx y_1$ (5)

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

$x_2 = x_1 + h$ (6)

$y_2 = y_1 + f(x_1, y_1)(x_2 - x_1)$

$= y_1 + hf(x_1, y_1)$ (7)

We have

$y(x_2) \approx y_2$ (8)

Then define for n = 0,1,2,3,4,5,..., we have

$x_n = x_0 + n_h$ (9)

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,

$y(x_{n+1}) \approx y_{n+1} = y_n + hf(x_n, y_n)$ (10)

Hence,

$y_{n+1} = y_n + hf(x_n, y_n)$ (11)

Equation (11) is called Euler's method.

From (11), we have

$$\frac{y_{n+1} - y_n}{h} = f(x, y), n = 0,1,2,3,... \quad (12)$$

2.3 Truncation Errors for Euler's

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

$$f(x) = f(a) + f'(a)(x-a)\frac{f''(a)(x-a)^2}{2!} +$$

$$\frac{f'''(a)(x-a)^m}{m!} + \frac{f^{m+1}(\beta)(x-a)^{m+1}}{(m+1)!} \quad (13)$$

The last term of (13) is referred to as the remainder term. Where $x \le \beta \le a$

In (13), let $x = x_{n+1}$ and $x = a$, in which

$y(x_{n+1}) = y(x_n) + hy'(x_n) + h^2 y''(\beta_n)$ (14)

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

$y'(x_n) = f(x_n, y(x_n))$ (15)

Hence,

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

By considering (18) to Euler's approximation in (11), 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.

2.4 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

$$e_n = y_n - y(x_n) \qquad (17)$$

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

$$y(x_{n+1}) = y(x_n) + hf(x_n, y_n) - t_n \qquad (18)$$

Where t_n is called the local truncation error. It is the amount by which the solution fails to satisfy the difference method. Subtracting (18) from (11), we get

$$e_{n+1} = e_n + h[f(x_n, y_n) - f(x_n, y(x_n))] + t_n \qquad (19)$$

Let us write

$$e_n M_n = f(x_n, y_n) - f(x_n, y(x_n)) \qquad (20)$$

Substituting (19) into (20), then

$$e_{n+1} = e_n (1 + hM_n) \qquad (21)$$

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

$$e_{n+1} \le e_n (1 + hM) + T \qquad (22)$$

To proceed further, we need the following lemma.

Lemma**:** If e_n satisfies (22) and $0 \le nh \le a$, then

$$\left|e_n\right| = T\frac{(1+hM)^n - 1}{hM} + (1+hM)^n\left|e_0\right| \le \frac{T}{hM}(e^{Lb} - 1) + e^{Lb}\left|e_o\right| \quad (23)$$

Lemma: The first inequality of (23) follows by induction. It is trivially true for n = 0. Assuming that it is true for n = 0, we have from (21)

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

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

$$= T\frac{(1+hM)^{n+1} - 1}{hM} + (1+hM)^{n+1}\left|e_0\right| \quad (24)$$

Hence (23) is true for n +1and thus for all n .

The second inequality in (23) 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 (18), 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 $0 \le \theta \le 1$,

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

$$\le h\left|f(x_n + \theta h, y(x_n)) - f(x_n, y(x_n)) + h\, y(x_n + \theta h) - y(x_n)\right| \quad (25)$$

The last term can be treated by the Mean value theorem to get a bound $M\theta h^2\left|y'(g)\right| \le h^2 MZ$, where

$Z = \max\left|y'(x)\right|$, the inequality exists because of the continuity of y and f in a closed region. The treatment of the first term in (25) 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 (25) by $L\theta h^2$, where L is the Lipschitz constant for f (x) .

Consequently, $\left|t_n\right| \le h^2(L + MZ) = T$ and so from (23), we get

$$\left|e_n\right| \le h\frac{L + MZ}{M}(e^{Ma} - 1) + e^{Md}\left|e_o\right| \quad (28)$$

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

2.5 Bogacki-Shampine method

This solver is designed for solving non-stiff problems. It is a method that is based on second and third order Runge-Kutta pair called the Bogacki-Shampine method. Bogacki-Shampine method is less expensive than ode45 in that it requires less computation steps than ode45. But it is of a lower order, although it may be more efficient at crude tolerance and in the presence of mild stiffness. Bogacki-Shampine method is one-step solver. The Bogacki-Shampine method is a Runge-Kutta method of order 3 with four stages proposed by Pzemyslaw Bogsacki and Lawrence F. shampine in I989. It uses three function evaluations per step. It has embedded second order method which is used to implement adaptive step size for the method.

2.6 Derivation of Bogsacki-Shampine method

The Butcher tableau for the Bogacki-Shampine is:

$$\begin{array}{cccccc} \frac{1}{2} & \frac{1}{2} & & & & \\ \frac{3}{4} & 0 & \frac{3}{4} & & & \\ 1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & & \\ & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 & \\ & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{8} & \end{array}$$

Following the standard notation, the differential equation to be solved is

$$y' = f(t, y) \tag{26}$$

Furthermore, y_n denotes the numerical solution at time t_n and h_n is the step size, defined by

$$h_n = t_{n+1} - t_n \tag{27}$$

Then, one step of the Bogacki-shampine method is given by:

$$k_2 = f\left(t_n + \frac{1}{2}h_n, y_n + \frac{1}{2}h_n k_1\right) \tag{28}$$

$$k_3 = f(t_n + \frac{3}{4}h_n, y_n + \frac{3}{4}h_n k_2) \quad (29)$$

$$y_{n+1} = y_n + \frac{2}{9}h_n k_1 + \frac{1}{3}h_n k_2 + \frac{4}{9}h_n k_3 \quad (30)$$

$$k_4 = f(t_n + h_n, y_{n+1}) \quad (31)$$

$$z_{n+1} = y_n + \frac{7}{24}h_n k_1 + \frac{1}{4}h_n k_2 + \frac{1}{3}h_n k_3 + \frac{1}{8}h_n k_4 \quad (32)$$

Here, z_{n+1} is a second-order approximation to the exact solution. The method for calculating y_{n+1} is due to Ralston (1965). On the other hand, y_{n+1} is a third-order approximation, so the difference between y_{n+1} and z_{n+1} can be used to adapt the step size. The FSAL-first same as last-property is that the stage value k_4 in one step equals k_1 in the next step; thus only function evaluations are needed per step (Bogacki.et al,1989).

2.7 Adams-Bashforth-Moulton

Adams-Bashforth-Moulton is a multi-step variable order method which uses Adams-Bashforth-Moulton predictor and corrector of order 1 to 13. It may be more efficient than Runge-kutta at stringed tolerance and when the ODE problem is particularly expensive to evaluate. It is designed for non-stiff problems.

2.8 Derivation of Adams-Bashforth Method

To simplify, let $f_k = f(x_k, y_k)$. (32)

Then the general form of Adams-Bashforth method is

$$y_{n+r} = y_{n+r+1} + h\sum_{k=1}^{r} \lambda_k f_{n+r-k} \quad (33)$$

Where $\sum_{k=1}^{r} \lambda_k = 1$. For the two-step

Adams-Bashforth method, set $\lambda_1 = 1 - \lambda, \lambda_2 = \lambda$. The (33) becomes

$$y_{n+2} = y_{n+1} + h((1-\lambda) f_{n+1} + \lambda f_n)$$

$$= y(t_{n+1}) + h((1-\lambda) y'(t_{n+1}) + \lambda y'(t_n) \quad (34)$$

By using Taylor's theorem, expand $y'(t_n)$ at t_{n+1} to get

$$y_{n+2} = y_{n+1)} + \frac{3}{2}(t_{n+1,} y_{n+1)} - \frac{1}{2} hf\,(t_{n,} y_n) \quad (35)$$

Thus, the simplified form is

$$y_{n+2} = y(t_{n+1)} + hy'(t_{n+1)} - \lambda(h^2) y''(t_{n+1}) + o(h^3) \quad (36)$$

Expanding $y(t_{n+2})$ at $y(t_{n+1})$ yields

$$y_{n+2} = y(t_{n+1)} + hy'(t_{n+1)} + \tfrac{1}{2}(h^2) y''(t_{n+1}) + o(h^3) \quad (37)$$

Subtracting (36) from (37) and then requiring the h^2 term to cancel makes $\lambda = -\frac{1}{2}$

The two-step Adams-Bashforth method is then

$$y_{n+2} = y_{n+1)} + \frac{3}{2}(t_{n+1,} y_{n+1)} - \frac{1}{2} hf\,(t_{n,} y_n) \quad (38)$$

Since

$y(t_{n+2)} - y_{n+2)} = o(h^3)$. The local truncation error is of order $o(h^3)$ and thus the method is second order (wikiversity, 2018).

3.0 Numerical example

In this section, the numerical methods are applied to solve systems (linear) of differential equation. To illustrate the methods proposed in this paper, the example below was considered:

Process Description

The overall mechanism of the fermentation process that performs this transformation can be described as follows:

Cell growth:

Glucose + Cells → Cells

Glucose oxidation:

Glucose + 0_2 → Gluconolactone + H_20_2 (G1ucoseasOxidase as Enzyme)

Gluconolactone hydrolysis:

Gluconolactone + H_2O → Gluconic acid

Peroxide decomposition:

H_2O_2 → H_20 + ½ O_2 (in the presence of a Catalyst)

Rate of Cell Growth

$$\frac{dy_1}{dt} = b_1 y_1 (1 - \frac{y_1}{b_2}) \quad (39)$$

Rate of Gluconolactone Formation

$$\frac{dy_2}{dt} = \frac{b_3 y_1 y_4}{b_4 + y_4} - 0.9082 b_5 y_2 \quad (40)$$

Rate of Gluconic acid formation

$$\frac{dy_3}{dt} = b_5 y_2 \quad (41)$$

Rate of Glucose Consumption

$$\frac{dy_4}{dt} = -1.011 \frac{b_3 y_1 y_4}{b_4 + y_4} \quad (42)$$

A mathematical model of the fermentation of the bacterium Pseudomonas ovalis, which produces Gluconic acid, has been developed by Rai and Constantinides. This model, which describes the dynamics of the logarithmic growth phases, can be summarized as follows:

Where y_1 =concentration of cell

y_2= concentration of Gluconolactone

y_3= concentration of Gluconic acid

y_4= concentration of glucose

$b_1 – b_5$ =parameters of the system which are functions of temperature and pH. At the operating conditions of 30°C and pH 6.6,the values of the five parameters were determined from experimental data to be

$b_1 = 0.949,$ (43)

$b_2 = 3.439,$ (44)

$b_3 = 18.72,$ (45)

$b_4 = 37.51,$ (46)

$b_5 = 1.169,$ (47)

At these conditions, develop the time profiles of all variables, y_1 to $y_4$for the period $0 \geq \mathbf{t} \geq 9$ h. The initial conditions at the start of this period are

$Y_1 (0) = 0.5$ U.O.D/ml (48)

$Y_2(0) = 0.0$ mg/l (49)

$Y_3(0) = 0.0$ mg/l (50)

$Y_4(0) = 50.0$mg/l (51)

(Constantinides and Mostoufi, 2000).

4.0 Results

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 MatLab Programing language and implemented on a computer. The performance of the methods was checked by comparing their accuracy. The accuracy is determined by the size of the discretization error estimated from the difference between the exact solution and the numerical approximations.

Table 1
Errors involved with the numerical methods for $y_1(t)$ in the Example

		Euler	Bogacki-Shampine	Adams-Bashforth-Moulton
t	n	y_1	y_1	y_1
1	0	0	0	0
2	0.50000000	0	-0.03566	-0.03566
3	1.00000000	0	-0.08183	-0.08183
4	1.50000000	0	-0.12552	-0.12541
5	2.00000000	0	-0.14733	-0.14679
6	2.50000000	0	-0.13299	-0.13209
7	3.00000000	0	-0.0866	-0.08595
8	3.50000000	0	-0.03039	-0.02998
9	4.00000000	0	0.013509	0.013861
10	4.50000000	0	0.036522	0.036849
11	5.00000000	0	0.042305	0.042445
12	5.50000000	0	0.038183	0.038271
13	6.00000000	0	0.030454	0.030504
14	6.50000000	0	0.022574	0.022607
15	7.00000000	0	0.015972	0.015993
16	7.50000000	0	0.010951	0.010965
17	8.00000000	0	0.007345	0.007354
18	8.50000000	0	0.004848	0.004854
19	9.00000000	0	0.003162	0.003165

Table 1 represents the deviation of ODE solvers employed from the analytical method of the calculus with, Euler, Bogacki-Shampine and Adams-Bashforth-Moulton having average errors of 0, -0.021815212, -0.021624024 respectively. From the average error results obtained it show that Euler is the best for evaluating the numerical approximations of the problem.

Table 2
Errors involved with the numerical methods for y_2 (t) in the Example

		Euler	Bogacki-Shampine	Adams-Bashforth-Moulton
t	n	y_2	y_2	y_2
1	0	0	0	0
2	0.50000000	0	0.126387	0.126592
3	1.00000000	0.000313	-0.12075	-0.11901
4	1.50000000	0.000722	-0.40386	-0.40091
5	2.00000000	0.001178	-0.43692	-0.43449
6	2.50000000	0.001668	0.007742	0.006162
7	3.00000000	0.002159	0.912617	0.916113
8	3.50000000	0.002574	1.802717	1.827354
9	4.00000000	0.002791	1.884324	1.953168
10	4.50000000	0.002687	0.93115	0.961873
11	5.00000000	0.00225	-0.2959	-0.28314
12	5.50000000	0.00164	-0.9335	-0.92843
13	6.00000000	0.001071	-0.99325	-0.99184

14	6.50000000	0.000649	-0.80417	-0.80414
15	7.00000000	0.000374	-0.57572	-0.57602
16	7.50000000	0.000208	-0.38569	-0.38614
17	8.00000000	0.000113	-0.24831	-0.24877
18	8.50000000	6.02E-05	-0.15583	-0.1562
19	9.00000000	3.16E-05	-0.09612	-0.09639

Table 2 represents the deviation of ODE solvers employed from the analytical method of the calculus with, Euler, Bogacki-Shampine and Adams-Bashforth-Moulton having average errors of 0, 00.373235515, and 0.019250918 respectively. From the average error results obtained it show that Euler is the best solvers for evaluating the Numerical approximations of the problem.

Table 3
Errors involved with the numerical methods for y_3 (t) in the Example

		Euler	Bogacki-Shampine	Adams-Bashforth-Moulton
t	n	y_3	y_3	y_3
1	0	0	0	0
2	0.50000000	0	-0.75325	-0.75362
3	1.00000000	0	-1.40337	-1.40645
4	1.50000000	0.000183	-2.21297	-2.21796
5	2.00000000	0.000605	-3.16186	-3.16674
6	2.50000000	0.001293	-3.98042	-3.98346
7	3.00000000	0.002268	-4.22742	-4.23812

8	3.50000000	0.00353	-3.5326	-3.55845
9	4.00000000	0.005035	-1.94519	-1.99185
10	4.50000000	0.006666	-0.14577	-0.164
11	5.00000000	0.008236	1.074989	1.069742
12	5.50000000	0.009551	1.448092	1.44729
13	6.00000000	0.01051	1.296463	1.297421
14	6.50000000	0.011136	0.981003	0.982303
15	7.00000000	0.011515	0.681589	0.682695
16	7.50000000	0.011734	0.45176	0.452668
17	8.00000000	0.011855	0.291758	0.292463
18	8.50000000	0.011921	0.186245	0.18675
19	9.00000000	0.011956	0.118947	0.119291

Table 3 represents the deviation of ODE solvers employed from the analytical method of the calculus with, Euler, Bogacki-Shampine and Adams-Bashforth-Moulton having average errors of 0.006210211, -0.780631789, -0.786843526 respectively. From the average error results obtained it show that ode23 and ode113 are the best solvers for evaluating the numerical approximations of the problem.

Table 4
Errors involved with the numerical methods for $y_4(t)$ in the Example

		Euler	Bogacki-Shampine	Adams-Bashforth-Moulton
t	N	Y_4	Y_4	Y_4
1	0	0	0	0
2	0.50000000	0	0.563847	0.563979
3	1.00000000	0	1.410953	1.412029
4	1.50000000	0	2.441129	2.442733
5	2.00000000	0	3.346666	3.348682
6	2.50000000	0	3.649831	3.654217
7	3.00000000	0	2.96319	2.969472
8	3.50000000	0	1.426896	1.425723
9	4.00000000	0	-0.11155	-0.13831
10	4.50000000	0	-0.79871	-0.81303
11	5.00000000	0	-0.67805	-0.68613
12	5.50000000	0	-0.37542	-0.37982
13	6.00000000	0	-0.17549	-0.17779
14	6.50000000	0	-0.07685	-0.07808
15	7.00000000	0	-0.03283	-0.03354
16	7.50000000	0	-0.01388	-0.01427
17	8.00000000	0	-0.00585	-0.00603

18	8.50000000	0	-0.00246	-0.00254
19	9.00000000	0	-0.00103	-0.00107

Table 4 represents the deviation of ODE solvers employed from the analytical method of the calculus with, Euler, Bogacki-Shampine and Adams-Bashforth-Moulton having average errors of 0, 0.712125895, and 0.709801316 respectively. From the average error results obtained it show that Euler is the best Solvers for evaluating the Numerical approximations of the problem.

5. Conclusion

The following can be concluded:

The results show that the numerical techniques can be used to obtain the numerical approximations for the problem and they have little deviations from the analytical solution. Euler proved to be the best numerical method with 0, 0, 0.006210211,0, average errors for the various concentrations of y_1, y_2, y_3 and y_4 respectively. Based on the results and analyses carried out, Euler solver was concluded to be the best numerical solver for the mathematical model involved in the chemical system this research work considered due to the lesser deviation it showed from the analytical solution when compared with other solvers such as Bogacki-Shampine and Adam-Bashforth-Moulton methods. This report then concludes that numerical simulation of complicated mathematical models found in chemical engineering provides an improved alternative to analytical or exact solutions of calculus as less time and drudgery are required to obtain numerical answers. Numerical approximation is an efficient way of solving complex Chemical Engineering problems and MatLab is very useful computational software that can be used for numerical approximations.

References

Abdel-Halim Hassan, I.H. (2003). Application to differential transformation method for solving systems of differential equations, Applied Mathematical Modelling 32, pp. 2552–2559.

Autar, K., Garapati S.H. (2011). Physical Problem for partial Differential Equations for General engineering, University of South Florida.

Bogacki, P., Shampine, L.F. (1989). A3(2) pair of Runge-Kutta formulas, Applied mathematics Letters, 2 (4), pp 321-325.

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

Constatinides, A., and Mostoufi, N. (2005). Numerical methods for Chemical Engineers with MatLab Applications, Prentice Hall PTR, upper Saddle River, New Jersey 07458.

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

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

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

Ogunrinde, R.B., Fadigba, S.E., Okunola, J. T. (2012).Some Numerical Methods for Solving Initial Value Problems in Ordinary Differential Equation, IOSR Journal of Mathematics (IOSRJM) Volume 1, Issues. pp. 25-31.

Retrieved-https://en.wikiversity.org/w/index.php?title=Adams-Bashforh_and_Adams-Moulton_methods&oldid=1761775

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