Journal of Advances in Mathematics and Computer Science

36(6): 11-23, 2021; Article no.JAMCS.69889
ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Numerical Approximations of ODEs Initial Value Problem; A Case Study of Gluconic Acid Fermentation by *Pseudomonas ovalis*

N. A. Ihoeghian[1*] and B. O. John[1]

[1]Department of Chemical Engineering, University of Benin, Benin City, Edo State, Nigeria.

Authors' contributions

This work was carried out in collaboration between both authors. Author BOJ carried out this study as a research student under the supervision of author NAI as part of his requirement for the award of B. Eng degree. Author NAI designed the study, jointly developed the methodology of the study, reviewed this manuscript. Author BOJ jointly developed the study, the literature, coded the solvers, and wrote the first draft of the manuscript. Both authors read and approved the final manuscript.

Article Information

DOI: 10.9734/JAMCS/2021/v36i630369
Editor(s):
(1) Dr. Doina Bein, California State University, USA
Reviewers:
(1) Godlief Fredrik Neonufa, Artha Wacana Christian University, Indonesia.
(2) Agus Mirwan, Lambung Mangkurat University, Indonesia.
Complete Peer review History: https://www.sdiarticle4.com/review-history/69889

Original Research Article

Received 20 April 2021
Accepted 24 June 2021
Published 14 July 2021

Abstract

Across different sections of life, physical and chemical sciences, differential equations which could be ordinary differential equations (ODEs) or partial differential equations (PDEs) are used to model the various systems as observed. Some types of ODEs, and a few PDEs are solvable by analytical methods with much difficulties. However, the great majority of ODEs, especially the non-linear ones and those that involve large sets of simultaneous differential equations, do not have analytical solutions but require the application of numerical techniques. This work focused on exemplifying numerical approximations (Adams-Bashforth-Moulton, Bogacki-Shampine, Euler) of ODEs Initial value Problem in its simplest approach using a case study of gluconic acid frementation by *Psuedonomas Ovalis*. The performance of the methods was checked by comparing their accuracy. The accuracy was detrermined by the size of the discretization error estimated from the difference between analytical solution and numerical approximations. The results obtained are in good agreement with the exact solution. This work affirms that numerical methods give approximate solutions with less rigorous work and time as there is room for flexibility in terms of using different step sizes with the Euler solver as most accurate.

**Corresponding author: Email: newton.ihoeghian@uniben.edu, ihoeghiannewton@yahoo.com;*

Keywords: Adams-Bashforth-Moulton; Bogacki-Shampine; Euler; numerical simulation.

1 Introduction

All system undergone change can be described by differential equations, which can either be ordinary differential equations (ODEs) and partial differential equations (PDEs) (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 ODEs is classified into initial value and boundary value problems, depending on the conditions specified at the end points of the domain [1,2]. There are several excellent and exhaustive textbooks on this subject that may be consulted [3,4,5,6]. 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 [7]. Ihoeghian et al. [8] used Runge Kutta Lower order, Runge Kutta Higher Order and Adams-Bash Forth -Moulton to simulate non-isothermal plug flow reactor. From their findings, the solver accurately approximated the simulation. The non-isothermal viscoelastic flows at high Weissenberg numbers using a finite volume method on general unstructured meshes has been simulated numerically. They demonstrated the stability of the finite volume method in the experimentally relevant range of high Weissenberg numbers.their findings shows it was in good agreement with experimental data [9]. In this work numerical approximations to ODEs Initial value Problem using some selected methods [10] to solve a case study of gluconic acid frementation by *Psuedonomas Ovalis* has been presented to validate the adoption of numerical approximations for complex scientific systems studies.

2 Numerical Methods

Numerical methods become very important in any attempt to solve initial value problems in ODEs, particularly when the solution is not in closed form [1]. Here, three numerical methods are presented namely, Euler, Bogacki-Shampine and Adams-Bashforth-Moulton to generate numerical and approximate solutions to the initial value problem

$$y' = f(x, y), y(x_o) = y_o \tag{1}$$

2.1 Euler's Method

In mathematics and computational science, the Euler method is a first-order numerical procedure for solving ODEs with a given initial value. It is the most basic explicit method for numerical integration of ODEs, and it 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 ODEs of the form.

$$\frac{dy}{dx} = f(x, y), \qquad y(0), = y_0 \tag{2}$$

So only first order ODEs can be solved by using Euler's method [11]. Euler's method commonly called tangent line method, is the simplest numerical method for solving initial value problem in ODEs. It is particularly suitable for quick programming as originated by Leonhard Euler in 1768. This method is subdivided into three namely, Forward Euler's method, Improved Euler's method, Backward Euler's method [1].

2.1.1 Derivation of Euler's method

Derivation of Euler's method for generating, numerically, approximate solutions to the initial value problem in (1) is presented below, where x_0 and y_0 are initial values for x and y respectively. This is with an aim to

determine (approximately) the unknown function y(x) for $x \geq x_0$. Given the value of $y(x_0)$, as y_0, using equation (1), the instantaneous rate of change of y at point x_0 can be determine thus

$$y^{l}(x_0) = f\big((x_0), y(x_0)\big) = f(x_0, y_0) \tag{3}$$

Supposing the rate of change of y(x) remains f (x_0, y_0) for all point x, then y(x) would exactly equal $y_o + f(x_o, y_o)(x - x_o)$. The rate of change of y(x) does not remain $f(x_o, y_o)$ for all x, however, it can be expected to approximate f (x_0, y_0) for x close to x_0. When this is true, , then the value of y(x) will remain close to $y_o + f(x_o, y_o)(x - x_o)$ for x close to x_0, for small number h, we have;

$$x = x + h \tag{4}$$

$$y_1 = y_o + f(x_o, y_o)(x - x_o)$$

$$= y_o + hf(x_o, y_o) \tag{5}$$

Where

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

By the above argument,

$$y(x_1) \approx y_1 \tag{6}$$

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

$$x_2 = x_1 + h \tag{7}$$

$$y_2 = y_1 + f(x_1, y_1)(x_2 - x_1) = y_1 + hf(x_1, y_1) \tag{8}$$

We have $y(x_2) \approx y_2$ (9)

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

we have $x_n = x_0 + n_h$ (10)

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)) ≈ f (x_n, y(x_n)) ≈ f (x_n, y_n)

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

Thus,

$$y(x_{n+1}) \approx y_{n+1} = y_n + hf(x_n, y_n) \tag{11}$$

Hence,

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

Equation (12) is called Euler's method.

From (12), we have

$$\frac{y_{n+1} - y_n}{\mathrm{h}} = \mathrm{f(x, y)}, \mathrm{n} = \ 0,1,2,3, \ldots \tag{13}$$

2.1.2 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

$$f(x) = f(a) + f'(a)(x-a)\frac{f''(a)(x-a)^2}{2!} + \cdots + \frac{f'''(a)(x-a)^m}{m!} + \frac{f^{m+1}(\beta)(x-a)^{m+1}}{(m+1)!} \tag{14}$$

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

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

$$(x_{n+1}) = y(x_n) + hfy(x_n) + h^2 y''(\beta_n) \tag{15}$$

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

$$y'(x_n) = f(x_n, y(x_n)) \tag{16}$$

Hence,

$$y(x_{n+1}) = y(x_n) + hf\big(x_n, y(x_n)\big) + \frac{1}{2}h^2 y''(\beta_n) \tag{17}$$

So, when we consider (17) to Euler's approximation in (12), it becomes obvious that Euler's method is obtained by omitting the remainder term $1/2\,h^2 y''(\beta_n)$ in the Taylor expansion of $y(x_{n+1})$ at the point x_n. For each step, the truncation error in Euler's method is accounted for by the omitted term.

2.1.3 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 $0 \leq x \leq a$ 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

$$e_n = y_n - y(x_n) \tag{18}$$

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

$$y(x_n+1) = y(x_n) + hf\,(x_n, y_n) - t_n \tag{19}$$

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

$$e_{n+1} = e_n + h\left[f(x_n, y_n) - f\left(x_n, y(x_n)\right)\right] + t_n \tag{20}$$

Let us write

$$e_n M_n = f(x_n, y_n) - f(x_n, y(x_n)) \tag{21}$$

Substituting (19) into (20), then

$$e_{n+1} = e_n(1 + hM_n) \tag{22}$$

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 $T \geq t_n$ .Then we have

$$e_{n+1} \leq e_n(1 + hM) + T \tag{23}$$

To proceed further, we need the following lemma.

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

$$\left| e_n \right| = T\frac{(1+hM)^n - 1}{hM} + (1 + hM)^{''}\left| e_0 \right| \leq \frac{T}{hM}(e^{Lb} - 1) + e^{Lb}\left| e_0 \right| \tag{24}$$

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)

$$\left| e_{n+1} \right| = T\frac{(1 + hM)^n - 1}{hM} + (1 + hM)^{''} + T(1 + hM)^{n+1}\left| e_0 \right|$$

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

$$= T\frac{(1+hM)^{n+1} - (1+hM) + hM}{hM} + (1 + hM)^{n+1} \tag{25}$$

Hence (24) is true for n +1 and thus for all n .The second inequality in (24) follows from the fact that $nh \leq a$ and for $hM \geq 0, (1 + hM) \leq e^{Mh}$ so that $(1 + hM)^n \leq e^{Mnh} \leq e^{Ma}$, proving the lemma.

To continue the proof of the theorem, we need to investigate T, the bound on the local truncation error. From (19), 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 $0 \leq \theta \leq 1$,

$$\leq \mathrm{h}\left|\mathrm{f}\left(\mathrm{xn} + \theta\mathrm{h}, \mathrm{y(xn)}\right)\right| - f\left(xn, y(xn)\right) + h\left|\mathrm{f}\left(\mathrm{xn} + \theta\mathrm{h}, \mathrm{y(xn} + \theta\mathrm{h})\right) - \mathrm{f}\left(\mathrm{xn} + \theta\mathrm{h}, \mathrm{y(xn)}\right)\right|$$
$$\leq \mathrm{h}|\mathrm{f}\,(\mathrm{xn} + \theta\mathrm{h}, \mathrm{y(xn)}) - \mathrm{f}\,(\mathrm{xn}, \mathrm{y(xn)}) + \mathrm{h}\,\mathrm{y(xn} + \theta\mathrm{h}) - \mathrm{y(xn)}\,| \tag{25b}$$

The last term can be treated by the mean value theorem to get a bound $\mathrm{M}\theta\mathrm{h}^2\,|\mathrm{y}'(\mathrm{g})\,|\,\mathrm{h}^2\mathrm{MZ}$, 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 (25b) by $\mathrm{L}\theta\mathrm{h}^2$, where L is the Lipschitz constant for f (x).

Consequently, $|\mathrm{tn}| \leq \mathrm{h}^2(\mathrm{L} + \mathrm{MZ}) = \mathrm{T}$ and so from (23), we get

$$|e_n| \leq h\frac{L+MZ}{M}(e^{Ma}-1)+e^{Ma}\,|e_0| \tag{26}$$

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

2.2 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.

2.5.1 Derivation of Bogsacki-Shampine method

The Butcher tableau for the Bogacki-Shampine is:

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

Following the standard notation in (1), the differential equation to be solved is

$$y' = f(t,y)$$

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:

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

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

$$y_{n+1} = y_n + \tfrac{2}{9}h_n k_1 + \tfrac{1}{3}h_n k_2 + \tfrac{4}{9}h_n k_3 \tag{29}$$

$$y_{n+1} = y_n + \tfrac{2}{9}h_n k_1 + \tfrac{1}{3}h_n k_2 + \tfrac{4}{9}h_n k_3 \tag{30}$$

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

$$z_{n+1} = y_n + \tfrac{2}{24}h_n k_1 + \tfrac{1}{4}h_n k_2 + \tfrac{1}{3}h_n k_3 + \tfrac{1}{8}h_n k_4 \tag{32}$$

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 [12].

2.3 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 ODEs problem is particularly expensive to evaluate. It is designed for non-stiff problems [13].

2.3.1 Derivation of Adams-Bashforth Method

To simplify, let $f_k = f(x_k, y_k)$ (33)

Then the general form of Adams-Bashforth method is

$$y_{n+r} - y_{n+r+1} + h\sum_{k=1}^{r} \lambda_k f_{n+r-k} \tag{34}$$

Where $\sum_{k=1}^{r} \lambda_k$=1.

For the two-step Adams-Bashforth method, set

$$\lambda_1 = 1 - \lambda, \qquad \lambda_2 = \lambda.$$

Then (34) becomes

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

$$= y(t_{n+1}) + h(1-\lambda)y'(t_{n+1}) + \lambda y' t_n \tag{35}$$

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

$$y_{n+2} = y_{n+1} + \frac{3}{2}(t_{n+1}, y_{n+1}) - \frac{1}{2}hf(t_n, y_n) \tag{36}$$

Thus, the simplified form is

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

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

$$y_{n+2} = y(t_{n+1}) + hy'(t_{n+1}) + \frac{1}{2}(h^2)y''(t_{n+1}) + o(h^3) \tag{38}$$

Subtracting (37) from (38) and then requiring the h^2 term to cancel makes

$$\lambda = -\frac{1}{2}$$

The two-step Adams-Bashforth method is then

$$y_{n+2} = y_{n+1} + \frac{3}{2}(t_{n+1}, y_{n+1}) - \frac{1}{2}hf(t_n, y_n) \tag{39}$$

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 Atcovi [10,13].

3 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 (G1ucoseas Oxidase as Enzyme)

Gluconolactone hydrolysis:

Gluconolactone + H_2O → Gluconic acid

Peroxide decomposition:

H_2O_2 → H_20 + ½ O_2 (in the presence of manganese(IV) oxide)

Rate of Cell Growth

$$\frac{dy_1}{dt} = b_1 y_1 (1 - \frac{y_1}{b_2}) \tag{40}$$

Rate of Gluconolactone Formation

$$\frac{dy_2}{dt} = \frac{b_3 y_1 y_4}{b_4 + y_4} - 0.9082 b_5 y_2 \tag{41}$$

Rate of Gluconic acid formation

$$\frac{dy_3}{dt} = b_5 y_2 \tag{42}$$

Rate of Glucose Consumption

$$\frac{dy_4}{dt} = -1.011 \frac{b_3 y_1 y_4}{b_4 + y_4} \tag{43}$$

A mathematical model of the fermentation of the bacterium Pseudomonas ovalis, which produces Gluconic acid, has been developed by Rai and Constantinides (2000). 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

$b_1 = 0.949,$ (44)

b_2= 3.439, (45)

b_3 = 18.72, (46)

b_4= 37.51, (47)

b_5 = 1.169, (48)

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 (49)

Y_2(0) = 0.0 mg/l (50)

Y_3(0) = 0.0 mg/l (51)

Y_4(0) = 50.0mg/l (52)

(Adapted from: Constantinides and Mostoufi, 2000[2]).

4 Results

In order to confirm the applicability and suitability of the methods for solution of initial value problems in ODEs, 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.

Errors involved with the numerical methods for $y_1(t)$ in the example are presented in Fig. 1.

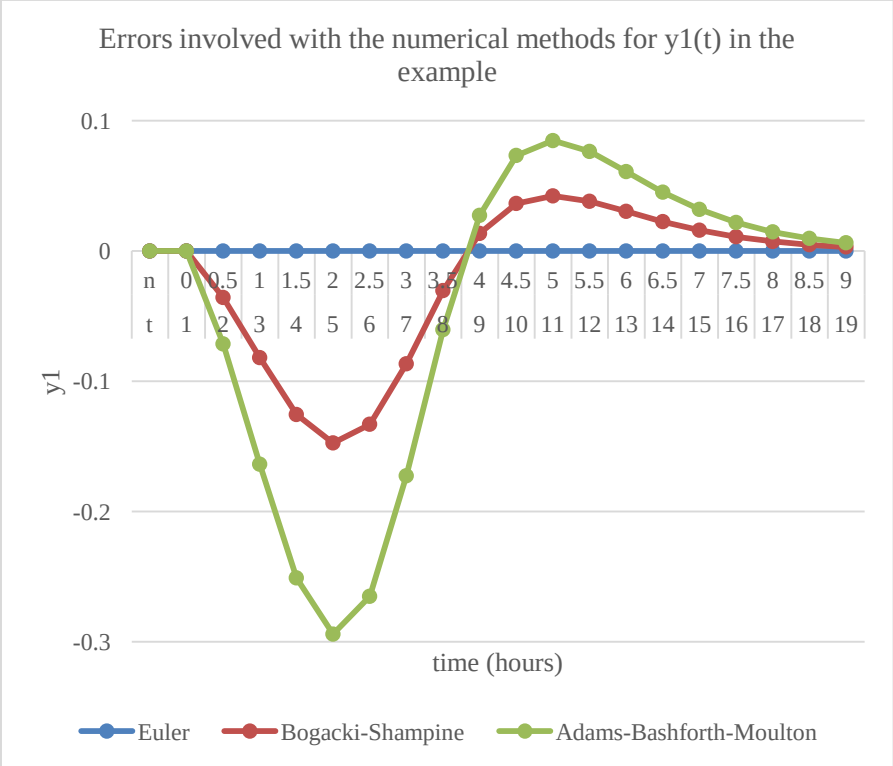

Fig. 1. Deviation of ODEs solvers from analytical method for y_1

Fig. 1 represents the deviation of ODEs 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.

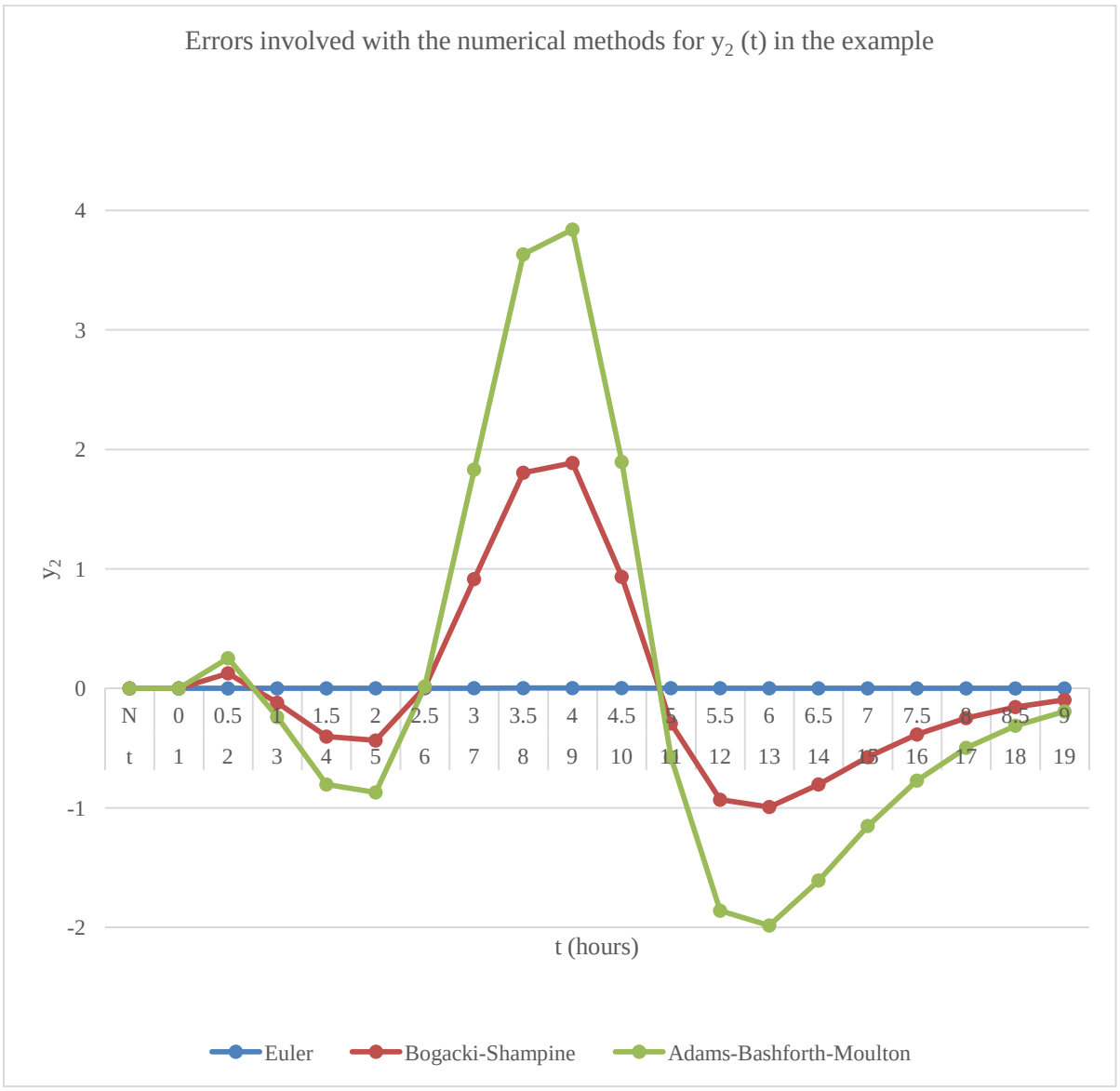

Fig. 2. Errors involved with the numerical methods for y_2 (t) in the example

Fig. 2 represents the deviation of ODEs 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.

Fig. 3 represents the deviation of ODEs 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. The average error results obtained show that ode23 and ode113 are the best solvers for evaluating the numerical approximations of the problem.

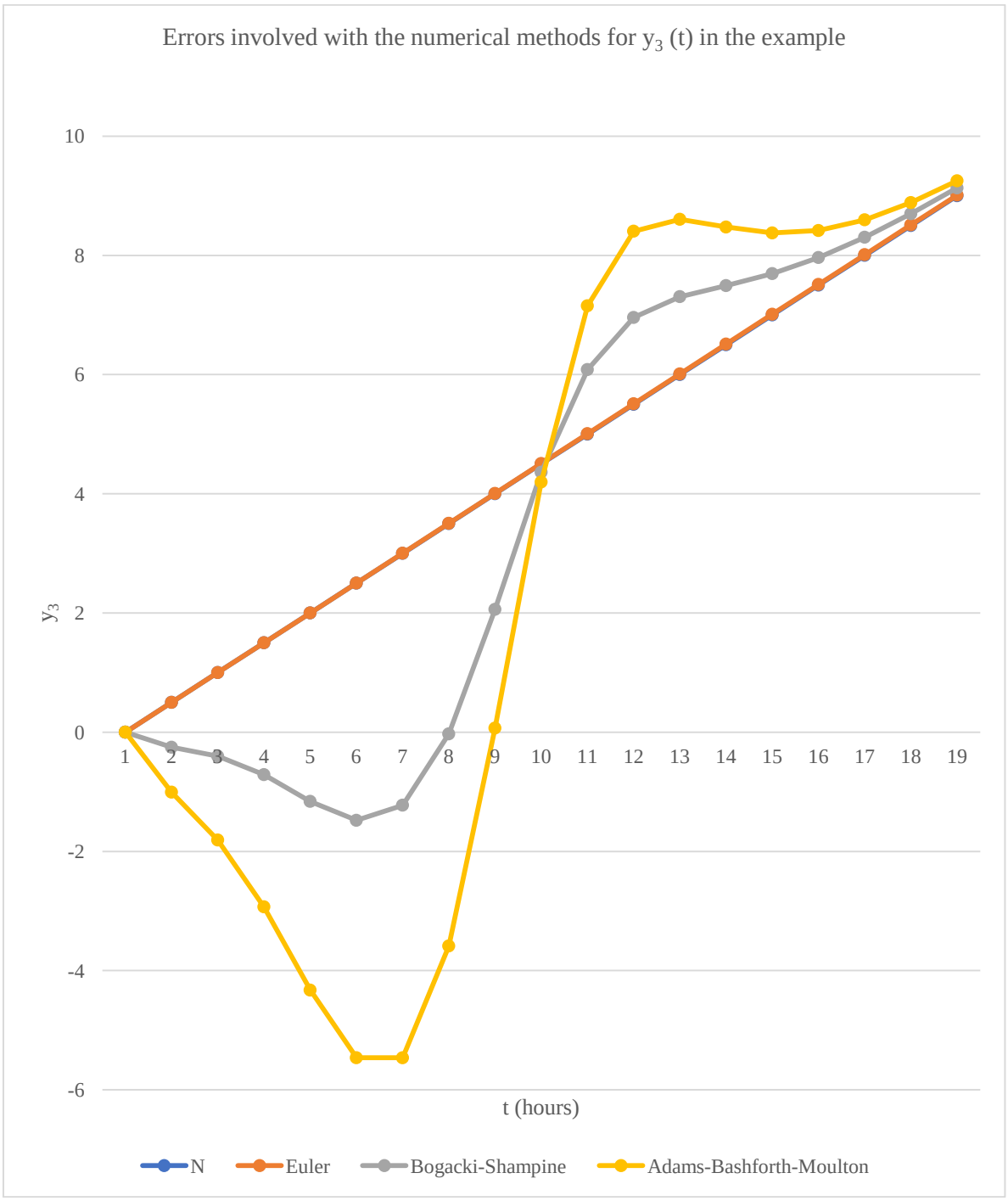

Fig. 3. Errors involved with the numerical methods for y_3 (t) in the example

Fig. 4 represents the deviation of ODEs 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.

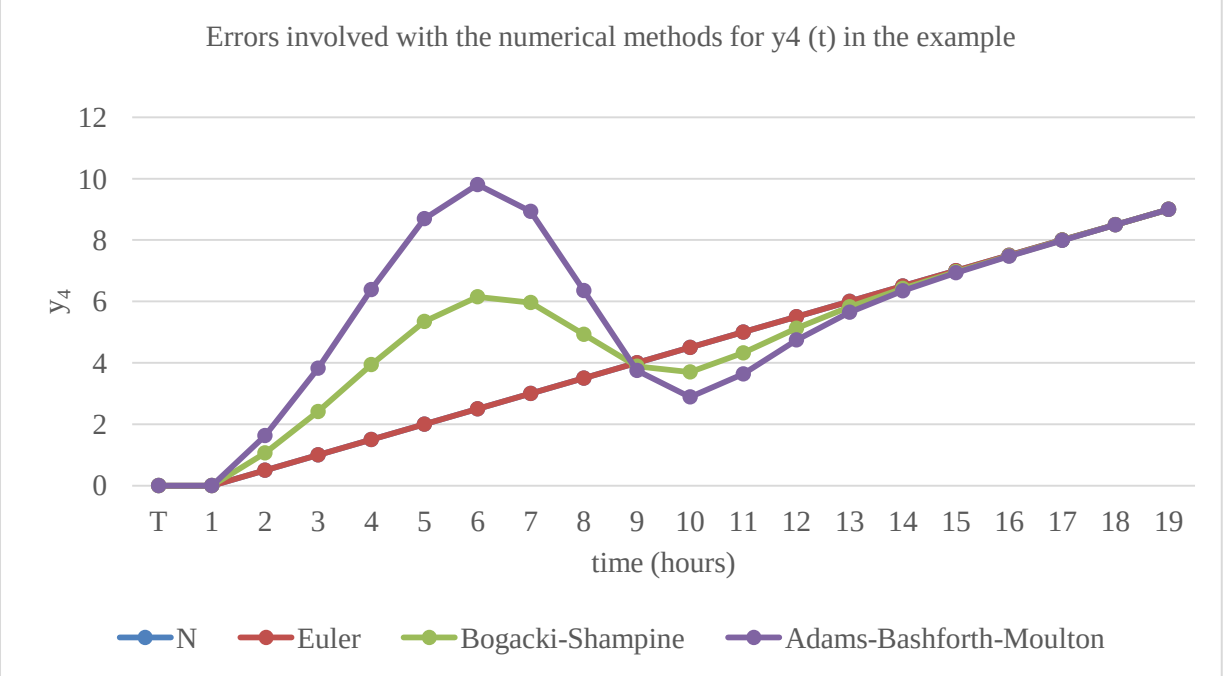

Fig. 4. Errors involved with the numerical methods for y_4 (t) in the example

5 Discussion of Result

Mathemathecally, the deviations observed with the numerical techniques were very small. This provides a basis for recommendation that numerical approximations with Euler, Bogachi-Shampine, and Adams-Bashforth-Moulton can within mathematical tolerance accurately approximate ODEs problems with Euler being most prefered. This position was not different from findings of Ihoeghian et al. in 2018 [8]. In their numerical simulation Runge Kutta Lower order was most accurate and significantly approximates their analytical solution [8]. Meburgerac et al. [9] demonstrated the stability of finite volume numerical method while studying thermo-rheological properties in Oldroyd-B type viscoelastic fluids. They found the results at different imposed wall temperatures, as well as Weissenberg numbers to be in good agreement with experimental data.

6 Conclusion

Numerical approximations (Adams-Bashforth-Moulton, Bogacki-Shampine, Euler) of ODEs Initial value Problem using a case study of gluconic acid frementation by *Psuedonomas Ovalis* has been studied. The performance of the methods was checked by comparing their accuracy. Less rigorous work and time was involved as it creats room for flexibility in terms of using different step sizes. The results obtained affirms that numerical methods give approximate solutions that are accurate and dependable for complex scientific system studies.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Ogunrinde RB, Fadigba SE, Okunola JT. Some numerical methods for solving initial value problems in ordinary differential equation. IOSR Journal of Mathematics (IOSRJM). 2012;1:25-31.

[2] Constatinides A, Mostoufi N. Numerical methods for chemical engineers with MatLab applications. Prentice Hall PTR, upper Saddle River, New Jersey 07458; 2005.

[3] Boyce WE, DiPrima RC. Elementary differential equation and boundary value problems, John Wiley and Son, Hoboken; 2001.

[4] Gilat A. Matlab: An Introduction with application. John Wiley and Sons; 2004.

[5] Kockler N. Numerical methods and scientific computing. Clarendon Press. Oxford London; 1994.

[6] Stephen MP. To compute numerically, concepts and strategy. Little Brown and Company, Canada; 1983.

[7] Abdel-Halim Hassan IH. Application to differential transformation method for solving systems of differential equations. Applied Mathematical Modelling. 2003;32:2552–2559.

[8] Ihoeghian N.A, Osagiede CA, Ikogwe KI. Numerical simulation of the non-isothermal plug flow reactor design for the dehydrogenation of ethylbenzene to styrene. Journal of the Nigeria Association of Mathematical Physics. 2018;44(1):311-318.

[9] Meburgerac S, Niethammer M, Bothe D, Schäfer M. Journal of Non-Newtonian Fluid Mechanics. 2021; 287(1):104451.
Available:https://doi.org/10.1016/j.jnnfm.2020.104451

[10] Atcovi. Adams-Bashforth and Adams-Moulton methods; 2017.
Available:https://en.wikiversity.org/w/index.php?title=Adams-Bashforh_and_Adams-Moulton_methods&oldid=1761775, (Accessed; 17 October 2018)

[11] Autar K, Garapati SH. Physical Problem for partial differential equations for general engineering. University of South Florida; 2011.

[12] Bogacki P, Shampine LF. A3(2) pair of Runge-Kutta formulas, Applied mathematics Letters. 1989;2(4):321-325.

[13] Lambert JD. Numerical method for ordinary systems of Initial value problems, John Wiley and Sons, New York; 1991.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
https://www.sdiarticle4.com/review-history/69889