Benghazi University … Faculty of Educatio
Journal of Faculty Education …The eleven number… April ... 2022

Numerical Solution of First Order Initial Value Problems Using Euler and Taylor methods, and comparing the solutions by using the Java language

Hanan Salem Abd Alhafied, Department of Mathematics, Faculty of Education, University of Benghazi, Libya (00218), hanan_e_m74@yahoo.com.

Bothaina Bukhatwa, Department of Mathematics, Faculty of Education, University of Benghazi, Benghazi, Libya (00218), *Bothaina.Bukhatwa@uob.edu.ly.*

Nada Ahmed Abdalqadr Alzwy, Department of Mathematics, Faculty of Education, University of Benghazi, Benghazi, Libya (00218), dodalzwy182@gmail.com.

Abstract

This research aims to find numerical solution of first order initial value problems using Euler and Taylor methods, and comparing the solutions by using the Java language. There is an example in the literature where solutions obtained manually are compared with results obtained using a method based on programming in Java. Initial results indicated that the use of the Java language helped to obtain more accurate results for solving this type of equation with the least error rate and the fastest possible time. This study recommends the use of various mathematical programs to help both the student and the teacher solve most mathematical problems in a faster and less errors way.

Key words

Numerical solutions, linear differential equations, Taylor series, Euler series, Java language.

الملخص

يهدف هذا البحث إلى إيجاد الحلول العددية لمسائل القيمة الابتدائية للمعادلات التفاضلية الخطية باستخدام طرق أويلر وتايلورومقارنة الحلول باستخدام لغة جافا. يوجد مثال على الأدبيات التي تم إجراؤها وتتم مقارنة الحلول التي تم الحصول عليها يدوياً بالنتائج التي تم الحصول عليها باستخدام طريقة تعتمد على البرمجة في Java. أشارت النتائج الأولية إلى أنّ استخدام لغة Java ساعد في الحصول على نتائج أكثر دقة لحل هذا النوع من المعادلات بأقل معدل خطأ، وأسرع وقت ممكن.

توصي هذه الدراسة باستخدام البرامج الرياضية المختلفة لمساعدة كل من الطالب والمعلم على حل معظم المشكلات الرياضية بطريقة أسرع وأقل أخطاء.

1 Introduction

Ordinary differential equations (ODEs) can describe phenomena that undergoing change. It used in dealing with many problems in various aspects such as physical, engineering and biological sciences, in addition to their contribution to the study of mathematical analysis and economic sciences, etc [1], [2]. To understand these problems, it was necessary to solve these differential equations, but it is not easy to solve all differential equations by analytical methods or even approximate methods by form of the solution on an infinite series, when failure to obtain its accurate or approximate solution [3]. There are many advanced solution methods, including the analytical method and the numerical method [3].

However, in many applications a solution is determined in a more complicated way to obtain the analytical solution for of (ODEs) [6]. Therefore, the simplest numerical method for solving these complex problems is called Euler’s method [1], [7]. Many studies work on numerical solutions to approximate the solution of ordinary differential equations with initial value problems (IVP) in order to obtain high accuracy solution. These studies using numerical technique, such as Taylor’s method, Euler’s method [8], [9], [10], [7], [11], [12], [13], [14].

In most cases of the (ODEs) problems is complex to find an exact solutions or even approximate solution by hand: an efficient reliable computer simulation is required [2]. With the technological development, most of the solution methods can be simulated using different software to appropriate for high accuracies. Different programming languages have been used to implement Euler's and Taylor's methods for numerous purposes. For example, the study of [15] has estimated and developed Euler pole parameters using MATLAB. As a result, this developed the Euler pole calculator software by using mathematical algorithms. Another study by [16] has improve the Euler number computing algorithm using MATLAB and it shows that the applied method outperforms significantly conventional Euler number computing algorithms. The research by [17] has studied a solution of Euler's an inviscid compressible fluid mechanics problem in two different ways, the solution produced using FORTRAN as well as MATLAB. After that, both solutions have been compared when both programs were initialized in the same technique, both solutions are it qualitatively were founded. A method using MATLAB presented in study of [18] to generate, present and manipulate Taylor series expansion by using matrices is. This method gives algebraic manipulation as well as differentiation in a very intuitive way to experiment with different numerical schemes with manual errors free. With many programs will save the time and effort required to calculate it [22] [6]

Maple is another program language that is useful for solving Taylor series method to solve ODEs numerically as announced in [19]. This study results some numerical tests for systems of equations through employing Maple. Mathematica program is also has been used to solve Taylor series. In particular, the study of [20] has found solutions to class of ODEs upon Mathematica codes and compare them with other existing techniques.

This paper will describe the solution of linear ordinary differential equation using numerical analysis (numerical methods) by Taylor's method (Taylor series), the Euler method to find an approximate solution to such equations [3], [21]. Java language, was used in the numerical solution of differential equation to facilitate the solution. Thus, this research will compare the result that introduced by using java language to the measured results and find the error value.

In practice, this research will contribute to giving an initial idea to undergraduate students and those interested in studying numerical solutions of liner differential equations on how to use the Java language to find numerical solutions to linear differential equations using Euler's method and Taylor method with the least possible errors. It also highlights the importance of different software applications of research mathematics to solve many of the various life problems in various fields.

2 Research problem

The differential equations are important in general and their close relationship with other sciences and engineering as well as economics, social science, biology, business, health care, etc [1]. Many mathematicians have studied the ordinary differential equations (ODEs) and developed several solution techniques. Often, some of the purely analytical solution to these equations is not controllable and it takes long time and great effort [22], [26], [27], [28]. However, the computer simulations and numerical methods for these equations are useful. Researchers faced many errors and a long time in solving the Euler method arising from the Taylor series. The techniques used in this paper based on numerical approximations developed using Java programming to avoid these errors and solve them in the least time possible.

3 Preliminaries

In this section we present a theoretical background on ODEs. First, we will review some basic concepts such as ordinary differential equations, linear differential equations, analytical solution, numerical solutions, the initial value problem, Taylor series, Euler's method, the total error and Java language. There is an example that illustrate the solution of ordinary differential equations using error approximation along with algorithms and practical implementation programs that are built into the Java programming environment. The use of Java program allows the student to compare the solutions and focus more on the concepts and less on the programming as well.

3.1 Ordinary Differential Equations (ODEs)

An Ordinary Differential Equations (ODEs) can be defined as a relation among an independent variable x, unknown function $y(x)$ of that variable, and certain of its derivatives. The most general form which an ODEs may assume is therefore $F\left(x, y, y', \ldots y^{(n)}\right) = 0$. The objects that satisfy the equations are functions called solutions or integrals of the differential equations

$y = g(x, c_1, c_2, \ldots c_n)$ where c_n is constant, [4], [5]. In this paper we will discuss the first order differential equation that has general form $\frac{dy}{dt} = f(\mathrm{y}, \mathrm{t})$, where $\frac{dy}{dt}$ means the change in y with respect to time, $f(\mathrm{y}, \mathrm{t})$ is any function of y and t.

3.2 Linear Differential Equations

A linear differential equation is any differential equation that can be written in the following form.

$$a_n\,(t)y^{(n)}(t) + a_{n-1}\,(t)y^{(n-1)}(t) + \cdots + a_1\,(t)y'(t) + a_0\,(t)y(t) = g(t)$$

Where the coefficients $a_n\,(t), a_{n-1}\,(t), a_1\,(t), a_0\,(t), g(t)$ can be zero or non-zero functions, constant or non-constant functions, linear or non-linear functions. However, there are no products of the function, $y(t)$, and its derivatives and neither the function or its derivatives occur to any power other than the first power. Also, neither the function or its derivatives are "inside" another function [1], [5].

3.3 Analytical solution

There are diverse solution techniques for the (ODEs) for instance, it can use methods that contain analytical integration for problems of separable variables. Some equations are complex to solve and require more unwieldly techniques, however, for some linear ODEs there are no general methods and they have no analytical solution [31].

3.4 Numerical solution of initial value problems IVP

Some of ODEs have an analytical solution that is complex or do not have it, for these two cases it is necessary to implement numerical methods that approximate the response with a small margin of error. To find an approximate solution to the ODEs [32]. numerical methods for ordinary differential equations are methods that are utilized to find numerical approximations to the solutions of (ODEs), through this research, Euler's method will be applied to find out the solutions [24].

3.5 The initial value problem (IVP)

The initial value problem (IVP) consists of an ordinary differential equation with an initial condition

$$\frac{dy}{dt} = f(\mathrm{y},\mathrm{t}), \quad t_\circ \le t \le t_n \;, \quad y(t_\circ) = y_\circ$$

3.6 Taylor series to solve (ODEs)

The Taylor series allows to obtain local approximations for a function around a point, in terms of the function and its derivatives of higher order. The IVP provides the first derivative of the function, solution of the differential equation and the coordinate of a point that satisfies the solution. Taylor series theorem states that if a function $y = f(t)$ with all its derivatives about a point a are continues where $t \in [a, b]$, then $f(t)$ can be expressed as in [25]:

$$f(t) = f(a) + (t-a)f'(a) + \frac{1}{2!}(t-a)^2 f''(a) + \frac{1}{3!}(t-a)^3 f'''(a) + \cdots + \frac{1}{n!}(t-a)^n f^n(a) + R_n(t,a),$$

where the $R_n(t,a)$ is given by $R_n(t,a) = \frac{1}{(n+1)!}(t-c)^{n+1}f^{n+1}(a) \quad c \in [a,t]$

3.7 Euler's method

Euler's method is the simple numerical method for solving the first-order initial value problem. It is resulting from dropping terms of second and higher order of the Taylor series to obtain approximate solution value [32].

We divide the interval [a, b] into N equal subintervals, with step size $h = \frac{b-a}{N}$

$t_i = a + ih \quad , \qquad i = 0,1,\dots,N$

Given the initial data t_0 and w_0, and the step size h. For $i = 0,1,\dots,N-1$

$$w_{i+1} = w_i + h\,f(t_i, w_i)$$

Where a numerical function $w(t)$ corresponds to the analytical solution function $y(t)$.

3.8 Taylor's method of order n

Assume that $w_0 = \alpha$, where α is a constant, the formula of Taylor method is given by

$w_{i+1} = w_i + h\,T^{(n)}(t_i\,, w_i)$, where $i = 0,1,\dots,N-1\,,\quad t_i = a + ih \quad, h = \frac{b-a}{N}$,

$a = t_0\,, t_1, t_2, \dots, t_n = b,\quad T^{(n)}(t_i\,, w_i)$ is Continuous and differentiable function

$$T^{(n)}(t_i\,, w_i) = f(t_i\,, w_i) + \frac{h}{2!}f'(t_i\,, w_i) + \cdots + \frac{h^{n-1}}{n!}f^{(n-1)}(t_i\,, w_i),$$

The analytical solution function $y(t)$ corresponds to a numerical function $w(t)$.

A continuous approximation to the solution y(t) will not be obtained; instead, approximations to y will be generated at various values, called grid points, on the interval [a, b]. Therefore, the smaller the value of h, the smaller the error ratio between the analytical and numerical solution.

3.9 The total error

The total error is the difference between the computed value w_i and the true value y_i [23].

3.10 Solve equations using Java

The Java language is one of the very modern languages in the world of programming, and Java is distinguished by having many tools that help in writing programs. Java language may be introduced as a programming language that is similar to C^{++} program and it is considered to be an improved edition of it, the program written in Java can be moved and run on another computer that has an operating system different from the first computer without problems [29]. As shown in the study of [30], java program is one of the software that can be used for solving ODEs. In addition, this study has compared the results of java to the results of Matlab as well. During this study, java program will be used to compare the results generated from Euler's method of solving ODEs.

4 Test problem

Here is an illustrative example for solving linear differential equations based on Taylor's method of the first orders (Euler's method), second and fourth order Modified Euler's method numerical approximations within the context of undergraduate study are investigated. (Atkinson, Han, & Stewart, 2011; Douglas and Burden,1993). We will use the algorithms that are built into the Java programming environment and will solve the same example to clarify that the Java program was faster and help in reducing numerical errors resulting from repetition of values when solving linear differential equations using Euler and Modified Euler's method. The initial value problem equation

$$y' = y - t^2 + 1 \ \ , N = 10,$$

$$y(0) = 0.5, \ 0 \le t \le 2,$$

We will apply Taylor's method of the first orders, second and fourth order and Modified Euler's method to obtain approximate value of y and compare the numerical solution obtained with the exact solution.

- **Solving by using the first order Taylor method (Euler method)**

In this exemple, we have $f(t, y) = y - t^2 + 1,$ with $h = 0.2$ **,** $i = 0,1, \ldots ,9$, $t_i = 0.2i$

$t_0 = 0\ ,\ \ y_0 = 0.5$

$t_1 = t_0 + h = 0.5\ \ , y_1 = y_0 + hf(t_0\,, y_0) = 0.5 + 0.5[0.5 + (0)^2 + 1] = 1.25$

$t_2 = t_1 + h = 1.0\ \ ,y_2 = y_1 + hf(t_1\,, y_1) = 1.25 + 0.5[1.25 + (0.5)^2 + 1] = 2.25$

$t_3 = t_2 + h = 1.5\ \ , y_3 = y_2 + hf(t_2\,, y_2) = 2.25 + 0.5[2.25 + (1)^2 + 1] = 3.375$

$t_4 = t_3 + h = 2.0, y_4 = y_3 + hf(t_3\,, y_3) = 3.375\ + 0.5[3.375 + (0)^2 + 1] = 1.25$

$f(t_i\,, w_i) = w_i - t_i^2 + 1$

$w_0 = 0.5 \quad , \quad w_{i+1} = 1.2w_i - 0.008i^2 + 0.2$

The analytical solution to the equation is $y(t) = (t + 1)^2 - 0.5\, e^t$

All the approximations and their errors are done by Java program and shown in table 1.

When we plot of the numerical solution and exact solution, we note that the error grows as t increases. The approximations become more accurate as the step size h =0.2 gets smaller (see the figure1 and figure2 below). However, the approximations become more accurate as the step size h gets smaller such as an example of h = 1 /10, h = 1 /20, h = 1 /50 .

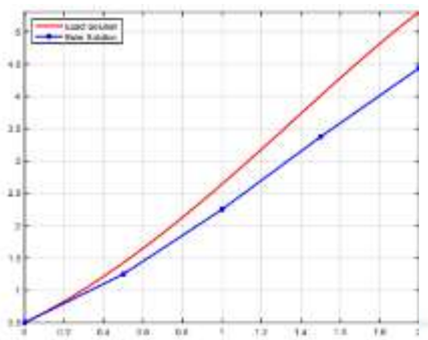

*Figure1: step size h =*0.2

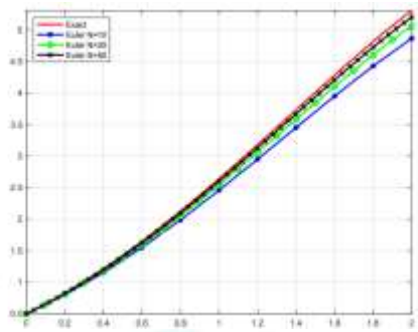

Figure 2: step size h = 1 /10 , h = 1 /20 , h = 1 /50

- **Solving by using the Taylor's method of second order**

$\because f(t, y(t)) = y(t) - t^2 + 1$

$f'\big(t\,, y(t)\big) = y'(t) - 2t = y(t) - t^2 - 2t + 1$

Therefore, the second order Taylor's method is

$T^{(2)}(t_i\,, w_i) = \left(1 + \frac{h}{2}\right)(w_i - t_i^2 + 1) - ht_i$

$h = 0.2$ **,** $t_i = 0.2i$, $i = 0,1,2, \dots, 9$

$w_0 = 0.5$

$$w_{i+1} = w_i + 0.2\left[\left(1 + \frac{0.2}{2}\right)(w_i - 0.04i^2 + 1) - (0.04i)\right]$$

$w_{i+1} = 1.22w_i - 0.0088i^2 - 0.008i + 0.22$

The first two steps give the approximations

$y(0.2) \approx w_1 = 1.22(0.5) - 0.0088(0) - 0.008(0) + 0.22 = 0.83$

$y(0.4) \approx w_2 = 1.22(0.83) - 0.0088(0.2)^2 - 0.008(0.2) + 0.22 = 1.2158$

All the approximations and their errors are done by Java program and shown in table 1

- **Solving by using the Taylor's method of fourth order**

$\because f(t, y(t)) = y(t) - t^2 + 1$

$$f'(t\,, y(t)) = y(t) - t^2 - 2t + 1$$

$$f''(t\,, y(t)) = y(t) - t^2 - 2t - 1$$

$$f'''(t\,, y(t)) = y(t) - t^2 - 2t - 1$$

$$T^{(4)}(t_i\,, w_i) = w_i - t_i^2 + 1 + \frac{h}{2}(w_i - t_i^2 - 2t_i + 1) + \frac{h^2}{6}(w_i - t_i^2 - 2t_i - 1) + \frac{h^3}{24}(w_i - t_i^2 - 2t_i - 1)$$

This is the numerical solution of the fourth order

$$w_{i+1} = 1.2214w_i - 0.008856i^2 - 0.00856i + 0.2186$$

The first two steps give the approximations as following

$$y(0.2) \approx w_1 = 1.2214(0.5) - 0.008856(0)^2 - 0.00856(0) + 0.2186 = 0.8293$$

$$y(0.4) \approx w_2 = 1.2214(0.8293) - 0.008856(0.2)^2 - 0.00856(0.2) + 0.2186 = 1.214091$$

All the approximations and their errors are done by Java program and shown in table 1. The analytical solution to the equation is $y(t) = (t+1)^2 - 0.5\,e^t$

When we plot of the exact solution, Euler, the second order and the 4th-order of Taylor method, (see the figure 2 below) we note that the 4th-order Taylor method generates very accurate approximation at the $x = 0.2, 0.4, 0.6\,, \dots, 1.8, 2.0$

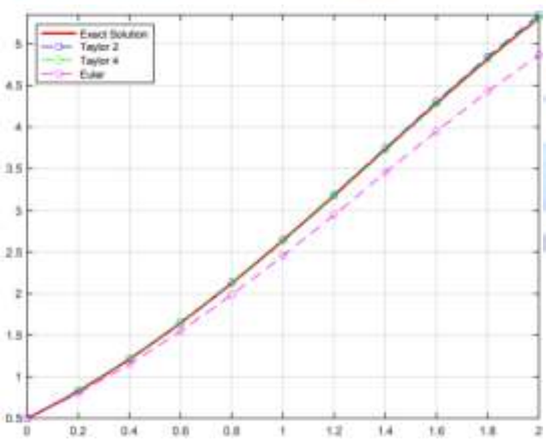

Figure 3 the exact solution, Euler, the second order and the 4th-order of Taylor method

- Solving by using Java language

Euler's method is easily completed by hand but it takes longer time; however, the complication derives from the amount of values and calculations one has to make to get possible close to the real values. Consequently, approach of a computer's programs such as Java has been taken for making the calculations to solve problems faster. We will be solved the problem numerically using the Java language. To create the program using the Java language, the researchers divided the program into different functions, including creating arrays, the numerical and analytical equation for the differential equation using Euler algorithm, and the equation of the absolute error ratio between the two solutions.

This program has the advantage that to solve a different linear differential equation you do not need to modify the code on it, only the equations that we want to solve and the number of periods will be changed according to the given equation. Figure 1 shows using Euler algorithm to solve it by Java program.

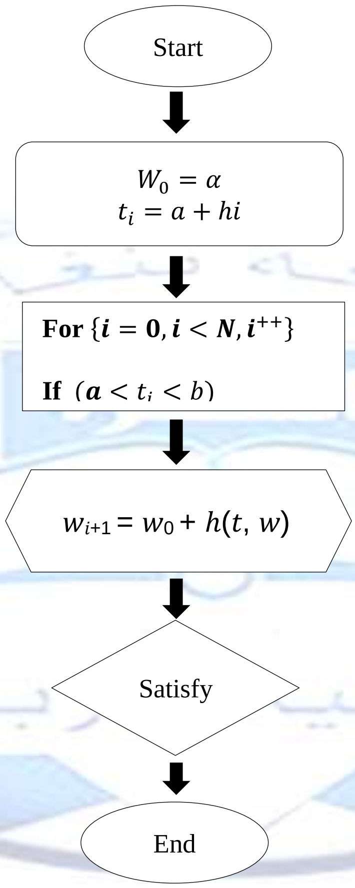

Figure 4 Euler algorithm to solve it by Java

Euler's method is easy and faster to implement on Java which generate a table of numerical solutions to the initial value problem allowing us to write a code for the Taylor series and we will change the inputs and the equations every time that. The inputs should contain, function f, endpoints of interval a and b, initial point $x_\circ, y_\circ$, the number of partition N or the step size h. It will calculate the approximate solution values $y_1, y_2, \ldots, y_n$.

As a result, from Java program, table (1) indicates that the smaller the h value, the smaller the error percentage. Note that the error grows a little as the value of t increases. This controlled error growth is a consequence of the stability of Euler's method, which implies that the error is expected to grow in no worse than a linear manner. The values of the derivative, decrease as we continue further on x.

If we compare the results of Taylor's method of second order and fourth order, we will see that the fourth-order results are vastly superior. The results from Table 1 indicate the Taylor's method of order 4 results are quite accurate at the nodes 0.2, 0.4. Furthermore, the approximate values are similar to the exact values up to three decimal places. As you can see from the table Euler's method gives us the solution to a differential equation, but it gets heavy on the calculation side, and since is not completely accurate, we leave this technique as a tool at hand to problems we cannot obtain an

Table 1 a result, from Java program, for the approximations and their errors

t_i	Euler wi	Exact $y_i = y(t_i)$	$\lvert y_i - w_i \rvert$	Taylor Order 2 wi	Error $\lvert y(t_i) - wi \rvert$	Taylor Order 4 wi	Error $\lvert y(t_i) - wi \rvert$
0.0	0.5000000	0.5000000	0.0000000	0.5000000	0	0.5000000	0
0.2	0.8000000	0.8292986	0.0292986	0.8300000	0.0007014	0.8293000	0.0000014
0.4	1.1520000	1.2140877	0.0620877	1.1258000	0.0017123	1.2140910	0.0000034
0.6	1.5504000	1.6489406	0.0985406	1.6520760	0.0031354	1.6489468	0.0000062
0.8	1.9884800	2.1272295	0.1387495	2.1323327	0.0051032	2.1272396	0.0000101
1.0	2.4581760	2.6408591	0.1826831	2.6486459	0.0077868	2.6408744	0.0000153
1.2	2.9498112	3.1799415	0.2301303	3.1913480	0.0114065	3.1799640	0.0000225
1.4	3.4517734	3.7324000	0.2806266	3.7486446	0.0162446	3.7324321	0.0000321
1.6	3.9501281	4.2834838	0.3333557	4.3061464	0.0226626	4.2835285	0.0000447
1.8	4.4281538	4.8151763	0.3870225	4.8462986	0.0311223	4.8152377	0.0000615
2.0	4.8657845	5.3054720	0.4396874	5.3476843	0.0422123	5.3055554	0.0000834

5 Conclusion

In this study we have considered the Taylor methods to approximate the solutions to initial-value problems for ordinary differential equations. The Taylor methods were considered as generalizations of Euler's method. They were found to be accurate but unwieldly because of the need to determine extensive partial derivatives of the defining function of the differential equation. It requires a lot of time and effort with the possibility of making mistakes, given the length of the solution steps. Therefore, the use of different programming languages helps in avoiding this type of error while saving time in the solution. The researchers concluded the following: The percentage of error between the analytical solution and the numerical solution decreases with our use of Euler and then in Taylor methods. Also, the use of Java programming reduced the proportion of numerical errors resulting from repetition of values when solving normal differential equations using Euler's method was in an easier way and in the least possible time.

6 Discussion

The Java language was used primarily due to the deficiency of references for solving differential equations in this language, in Libya. Given the importance of modern technology, the Department of Mathematics at the Faculty of Education in Benghazi was interested in integrating the theoretical study of mathematics courses, especially in the differential equations course with practical application using different programming languages. This enables the student studying in this department to make a comparison between the analytical solution and the numerical solution for the same problems and compare them in terms of accuracy and speed, which makes it easier for the student to understand and comprehend. With this approach, graduates of the College of Education, Department of Mathematics, can use modern technical means in learning and teaching mathematics in schools to facilitate and understand the scientific material. Also, this link between mathematics and the various means of technology will help in changing the traditional view of mathematics from abstract theoretical studies to practical applications; and it will facilitate for student students to grasp mathematical concepts in a sensory way.

7 Recommendations

This study recommends paying attention to research and studies in mathematics using various programs such as Java and other programming languages that would help in the speed of solving several mathematical problems. It also recommends providing computer laboratories for the specialization of mathematics to develop the study of mathematics using modern technological methods to keep with modern scientific development. In addition, this study may encourage students of the Department of Mathematics and other scientific departments to use programming languages in general to solve various mathematical equations related to their field. In addition, it may inspire students applying different programming languages such as Java and ready-made mathematical programs such as Matlab, Maple and Mathematica in some mathematical courses for students of the Mathematics Department of the Faculty of Education in Benghazi.

8 Acknowledgment

The authors express their gratitude to the Department of Mathematics, Faculty of Education, University of Benghazi, Libya. We would also like to thank all those who made it possible for us to complete this paper.

9 Disclosure statement

The authors certify that the manuscript represents original work and has not been submitted for publication, nor has it been published in whole or in part elsewhere. On behalf of all co-authors, the corresponding author shall bear full responsibility for submission. The authors declare no conflict of interest.

References

[1] Atkinson, K., Han, W., & Stewart, D. E., Numerical solution of ordinary differential equations, *John Wiley & Sons*, 108, 2011.

[2] Ascher, U.M. and Petzold, L.R., Computer Method for Ordinary Differential Equations and Differential-Algebraic Equations. *Society for Industrial and Applied Mathematics (SIAM)*, 61, 1998.

[3] NEUMAIER, R., Introduction to Numerical Analysis, *CAMBRIDGE UNIVERSITY PRESS*, 2001.

[4] Piccinini, L., Stampacchia, G., & Vidossich, G., Ordinary differential equations in Rn: problems and methods. *Springer Science & Business Media*, 39, 2012.

[5] Murphy, G. M., Ordinary differential equations and their solutions. *Courier Corporation*, 2011.

[6] Shampine, L., Kierzenka, J., & Reichelt, M., Solving boundary value problems for ordinary differential equations in MATLAB with bvp4c. *Tutorial notes*, 1-27, 2000.

[7] S. Nooraida, M.M.Y. Nurhafizah, M.S. Anis, A.M.M. Fahmi, A.W.F. Syarul & A.M. Iliana, Cube Polygon: A New Modified Euler Method to Improve Accuracy of Ordinary Differential Equation (ODE), *Journal of Physics: Conference Series*, 1532, 2020.

[8] Amirul, I., A comparative study on numerical solutions of initial value problems (IVP) for ordinary differential equations (ODE) with Euler and RungeKutta methods. *American Journal of Computational Mathematics*. 8:393-404, 2015.

[9] Gadisa, G., Garoma, H., Comparison of higher order Taylor's method and RungeKutta methods for solving first order ordinary differential equations. *Journal of Computer and Mathematical Sciences*,8(1):12-23, 2017.

[10] Kamruzzaman, M. and Nath M., Acomparative study on numerical solution of initial value problem by using Euler's method, modified Euler's method and RungeKutta method. *Journal of Computer and Mathematical Sciences*,9(5):493-500, 2018.

[11] Ochoche, A., Improving the modified Euler method. *Leonardo Journal of Sciences*,10:1-8, 2007.

[12] Ochoche, A., Improving the improved modified Euler method for better performance on autonomous initial value problems. *Leonardo Journal of Sciences*,12:57-66, 2008.

[13] Trofimets, E. & Trofimets, V., Research of numerical methods for solving ordinary differential equations in MS Excel. In *Journal of Physics: Conference Series*, 1691, 2020.

[14] Martins, F. P., Fleury, A. T., & Trigo, F. C., Explaining the importance of Euler's method in rigid-body dynamics. *International Journal of Mechanical Engineering Education*, 2020, 0306419020934123

[15] Goudarzi, M., Cocard, M. & Santerre, R., EPC: Matlab software to estimate Euler pole parameters, GPS Solut, 18:153–162, 2014.

[16] Yao, B., Wu, H., Yang, Y., Chao, Y. He, L., An improvement on Euler number computing algorithm used in MATLAB, *IEEE international conference of IEEE region 10 (TENCON), 2013.*

[17] Nordli, A. & Khawaja, H., Comparison of Explicit Method of Solution for CFD Euler Problems using MATLAB (R) and FORTRAN 77, *INTERNATIONAL JOURNAL OF MULTIPHYSICS. 13, 2019.*

[18] Pantaleon, C. & Ghosh, A., Taylor series expansion using matrices: an implementation in MATLAB (R), Computer & Fluids, 112, 79-82, 2015.

[19] Miletics, E. & Molnárka, G., Taylor series method with numerical derivatives for initial value problems. *Journal of Computational Methods in Sciences and Engineering*. 4. 105-114. 10.3233/JCM-2004-41-213, 2004.

[20] Verma, L., Pandit, B. & Verma, A., Taylor Series Solution of Some Real-Life Problems ODEs & PDEs, *Preprint*, 2020. 10.13140/RG.2.2.26947.55846.

[21] Faires , J. and Burden, R., Numerical methods. *20 Park Plaza Bosten. Massachusetts* 02116. PWS, 1993.

[22] Leite, E., MATLAB – modelling, programming and simulations. SCIYO, 2010.

[23] Everitt, B. S.; Skrondal, A., The Cambridge Dictionary of Statistics, Cambridge University Press, 2010.

[24] Arfken, G. & Weber, H., Mathematical Methods for Physicists, 6th Edition, 352-354, Academic press 200, 2005.

[25] BRADIE, B. A, friendly introduction to numerical analysis. Upper Saddle River, NJ, Pearson Prentice Hall, 2006.

[26] Ochoche A., Accurate solution of initial value problems for ordinary differential equations with fourth order RungeKutta method. Journal of Mathematical Research.;7(3):41-45, 2015.

[27] Ogunrinde RB, Fadugba SE, Okunlola JT, On some numerical methods for solving initial value problems in ordinary differential equations. IOSR Journal of Mathematics.;1:25-31, 2012.

[28] Gabriel, N Ordinary Differential Equations Mathematics Department, Michigan State University, East Lansing, 2021.

[29] Howard A., in Data Acquisition Techniques Using PCs (Second Edition), 2003.

[30] Mohammad Rafiq Muqri, Hasan Muqri and Shih Ek Chng, Leveraging the power of Java and Matlab to solve ODE's, American *Society for Engineering Education, 121st ASEE annual conference & exposition*, 2014.

[31] J. D. Faires, R. L. Burden and P. J. P. Escolano, "Métodos numéricos", Thomson-Paraninfo, España, 2007.

[32] Narvaez, D. M. D., Mesa, F., & Vélez, G. C. (2021). Comparative Analysis of Numerical Solutions of ODEs with Initial Value Problems using Improved Euler Methods. Scientia et Technica, 26(03), 391-397.