Journal of Statistics Applications & Probability
An International Journal

http://dx.doi.org/10.18576/jsap/120126

Statistical Analysis of a Linear Multi-Step Numerical Treatment

A. T. Zaidi[1] *and O. L. Al-Harbi* [2,*]

[1]Department of Mathematics, College of Science, Qassim University P.O.Box 6644, Buraydah 51452, Saudi Arabia
[2]Department of Mathematics, College of Science and Arts in Uglat Asugour, Qassim University Buraydah, Saudi Arabia

Received: 2 May 2022, Revised: 12 Jun. 2022, Accepted: 30 Jul. 2022.
Published online: 1 Jan. 2023.

Abstract: The aim of this paper is to compare the relative accuracies between predictor-corrector methods, Adams-Bashforth method and Adams-Moulton method for solving initial value Differential Equations numerically to observe which methods tend to function well in which step-size brackets as well as which ones provide the minimum amount of error when compared to the true value. The statistical analysis shows that there is always a small amount of error present using Heun's method; however, the error is rarely large enough unless the function is rapidly rising.

Keywords: Adams-Bashforth, Adams-Moulton, Linear Multi-step, Numerical methods, Predictor-Corrector.
2010 Mathematics Subject Classification. Primary 26A33, 35A20; Secondary 33B15.

1 Introduction

There are several techniques to solving differential equations of these techniques, analytical methods are the most common which include different types of integration methods and other formulae derived from said methods [1,2,3]. However, not all differential equations are easily solvable using analytical techniques [4,5]. Sometimes, the complexity of the differential equations can be too high, or previously known techniques cannot be applied to certain differential equations. For example, some techniques include the separation of variables, while others involve using Bernoulli's equation, variation of parameters, or other techniques. However, when none of the techniques can be applied, we can resort to other methods of solution known as numerical methods [6,7].

Numerical methods are known as numerical integration methods which can be used to approximate the solutions to different types of ordinary differential equations by repetitive iterations. For many practical applications, it's often not required to know the analytical solution of a differential equation. It is adequate to find the numeric approximation which leads us to apply the different algorithms to solve differential equations numerically. Alternate methods include using series expansions of the solutions using calculus to achieve an approximate solution [6].

There are also other techniques that convert partial differential equations into ordinary differential equations which then allow these equations to be numerically solved. There are many different types of numerical methods that can be used to solve initial value problems (hereafter abbreviated at IVP). The generalized name of these numerical method solutions is general linear methods [8].

General linear methods are further categorized into two different variations. The first variation is known as linear multistep methods. The second variation is known as Runge-Kutta methods. Each of these two methods is further divided into implicit and explicit methods. The primary example of implicit linear multistep methods is the Adams-Moulton method and the backwards differentiation method. The primary example of implicit Runge-Kutta methods is known as the Diagonally Implicit Runge-Kutta method. The explicit form of linear multistep methods is known as the Adams-Bashforth method. There are other types of explicit Runge-Kutta methods that are designated using a low diagonal Butcher tableau. Differential

[*]Corresponding author e-mail: 4438@qu.edu.sa

equations are also classified into different types, mainly stiff and non-stiff, which also helps to decide which techniques can be applied to achieve a numerical solution. Generally, stiff differential equations are easily solvable using implicit methods while non-stiff problems can be tackled using explicit methods [9,10,11].

Many of the initial methods that have been mentioned were initially discovered in the 18th and 19^{th} centuries, with Euler's method all the way back in 18^{th} century. In 1883, Bashforth and Adams published their method and shortly, two years later, Runge followed suit [10]. The Adams-Bashforth method, hence, was released which was integrated into many computer softwares for interpolation techniques. It was also the starting point for the Adams-Moulton method and the first integration of Taylor series methods to practically aid in numerical integration. The paper by Runge is often credited to be the first one-step method for numerical integration. These two, combined with Euler's method, were the first techniques on which other numerical methods are based [12,13].

Many of these papers tend to follow a similar technique for the development of their methods. They begin with an initial value problem in which some function has been given (usually defined as the $f(t,y)$) where the initial value has been given at the initial value xo. The objective, then, is to solve the differential equation $y'(t) = f(t, y(t))$ given the initial value $y(t_0) = y_0$. The approach to the problem begins with the objective of finding an approximation for the value of $y(t)$.

The first method of solving the differential equations numerically, provided by Bashforth and Adams, consists of an arbitrary second-order differential equation, with y and t as the variables. The derivation of the method consists of using the initial equation and differentiating it repeatedly to get formulae for higher derivatives of the same function. By substituting the initial value, we can evaluate the initial value and expand the solution using Taylor series. Eventually, the technique leads us to what is famously known as Adams-Bashforth method.

The second method provided for solving differential equations numerically was developed by Runge. While the Adams-Bashforth method is a multistep method, Runge developed a single step algorithm for the calculation of the approximate solution. While Adams-Bashforth used a series of different known approximations to estimate the next approximation, the Runge method focused on the most recent computed point to find the next. To reach the required level of accuracy, the approximations are found between the two computed points.

The next two significant contributors to these methods were Heun and Kutta. The method provided by Kutta is named the Runge-Kutta method and was the first time that differential equations being solved numerically became viable for equations other than quadrature formulae. The Runge-Kutta methods deal with the different inaccuracies present in the Runge methods which are used to satisfy the conditions of the three-order behavior of a differential equation. Kutta found classifications of the solutions to all the conditions required in the 4^{th} order Runge method which have widely become a reliable method to achieve solutions to numerical methods. Heun also later published his method, widely, also known as the improved Euler's method, which is a classification of a predictor-corrector method, as opposed to the Bashforth method which is a predictor method [3].

The last method that we will discuss is the Adams-Moulton method. The implicit part of the Adams method was not discussed in their own paper initially, until Moulton published his technique. This was called the Adams-Moulton method which had some significant improvements over the previous methods. Firstly, the number of previous values required was reduced significantly. Secondly, the error constants became very small. However, since their implicit nature comes with a certain level of difficulty, the method has limited practical usage. The Moulton method is known as a corrector method. By combining the Adams-Bashforth and Adams-Moulton methods, a predictor-corrector mode can be accomplished, similar to the Heun's method. This is the most common use of the Moulton method, in which the initial values predicted y_n are corrected using Heun's method which reduces the over-all error and leads to a more accurate result, at the cost of twice the processing power [1, 14].

Our paper will focus on a simulation of three methods, the Adams-Bashforth method, the Adams-Moulton method and the Heun's predictor-corrector method which uses Euler's method as a predictor and the trapezoidal method as the corrector. We will use the MATLAB software to program the methods into different iterations in which we will take two arbitrary differential equations as samples which will be solved using numerical methods mentioned above. The errors for all the iterations will be calculated, and each simulation will be run twice with different step sizes to observe the effect of step size on the accuracy of the system. A table will be generated in each method, with values of t the numerically calculated value of y, the true value of y achieved through analytical solution of the differential equation, and the percentage error between the two.

Eventually, we will also attempt to measure the performance of each algorithm in terms of the time taken to see which one reaches minimal levels of error fastest, if at all.

2 Methodology

For the purposes of our experiment, we will use the MATLAB software. We will program three different methods, namely, the Adams-Bashforth method, the Adams-Moulton method and the Heun's Predictor-Corrector method and solve two different equations. Each differential equation will be solved twice, once with a step size of 0.1 and then with a step size of 1 to compare the accuracy of the method with different step sizes. In some cases, other step sizes may be used to gain further insight, to be able to view the results clearly on a graph. In each iteration, a graph will be generated and compared with the previous graph to see whether the graph converges, diverges, or is unstable.

2.1 Differential Equations

For the purposes of this experiment, we will be taking two random differential equations with their initial conditions provided. These will be ordinary differential equations, treated as initial value problems.

1) First equation and its initial conditions. The first differential equation will be:

$$\frac{dy}{dt} + y = t \tag{1}$$

Where the initial conditions are given to be:

$$y_0 = y(t_0) = y(0) = -2 \,,$$

And after analytically solving the differential equation, the solution to the above differential equation comes out to be:

$$y = t - 1 - \frac{1}{e^t} \tag{2}$$

2) Second equation and its initial conditions. The second differential equation will be:

$$\frac{dy}{dt} + 3 = 5y \tag{3}$$

where the initial conditions are given to be:

$$y_0 = y(t_0) = y(0) = 2$$

And the analytical solution gives us the equation:

$$y = 1.4e^{5t} + 0.6 \tag{4}$$

Both of these differential equations will be solved using three separate techniques outlined below.

2.2 Heun's Method (Predictor-Corrector)

Heun's method is often called the improved Euler's method, which can be used to solve ordinary differential equations with provided initial values. The idea behind solving equations numerically using this method consists of using Euler's method as a predictor while using the trapezoidal rule as a corrector to increase the accuracy of the method. To implement the Heun's method, the following equations have to be used:

$$y'_{i+1} = y_i + [hf(t_i\,, y_i)] \tag{5}$$

$$y_{i+1} = y_i + \frac{h}{2}\left[f(t_i\,, y_i) + f\left(t_{i+1}\,, y'_{i+1}\right)\right] \tag{6}$$

Where $h = stepsize$ and $x_{i+1} = x_i + h$.

This means that for each iteration, the first step, equation (5) will be used to predict the next value for y, and this predictedvalue will be used in the next step, equation (6), to correct for errors and achieve the corrected value of y_{i+1}.

This is a relatively accurate and easy method to execute, and the accuracy of this method is improved quadratically.

2.3. 4th Order Adams-Bashforth Method

Adams-Bashforth is one of the very first numerical methods proposed to solve differential equations. This is an explicit method, which is also known as a predictor method. The core concept remains the same as the other multi-step methods, in which previous known values are used to predict other future values. Thus, the following equation is used to predict thefourth value, given that the first three values are known.

$$y_{n+4} = y_{n+3} + \frac{h}{24}\left(55\, f(t_{n+3}\, , y_{n+3}) - 59 f(t_{n+2}\, , y_{n+2}) + 37\, f(t_{n+1}\, , y_{n+1}) - 9\, f(t_n\, , y_n)\right) \qquad (7)$$

Obviously, first looking at this method, we realize that just the initial conditions are inadequate to apply the 4th Order Adams-Bashforth Method. However, there are other methods that can be applied to get the subsequent 3 values using initial conditions. One approach is to use the Euler's method to find the first value, and then use the first order Adams- Bashforth method to find the second value, use the consequent value in the second order Adams-Bashforth method and so on, until we have enough initial values to substitute into the equation for the fourth order Adams-Bashforth method. However, for the sake of simplicity, we used the $4th$ order Runge-Kutta method which can be repeated three times to getthe initial values required. The process to numerically calculate the required values will be outlined in the last part of thissection.

2.4. 4th Order Adams-Moulton Method

Adams-Moulton method was proposed by Adams, similar to the Adams-Bashforth method, and Moulton was credited with coming up with the realization that the two methods could be combined into a predictor-corrector pair. Adam used Newton's methods to solve the implicit equations which are what makes this an implicit method, as compared to the previous explicit Adams-Bashforth method. Since on its own, the Adams-Moulton method is a corrector method, it requires one more initial value than the Adams-Bashforth method. The equation to predict the fifth value, given the firstfive values, is:

$$y_{n+4} = y_{n+3} + \frac{h}{720}\left(251 f\,(t_{n+4}\, , y_{n+4}) + 646 f\,(t_{n+3}\, , y_{n+3}) - 264 f\,(t_{n+2}\, , y_{n+2}) + 106 f\,(t_{n+1}\, , y_{n+1}) - 19 f(t_n\, , y_n)\right) \qquad (8)$$

Once again, to calculate the first four values using a given initial condition, we will be using the order Runge Kutta method. We could use the Moulton-methods of lower orders as well, but for consistency, we used the Runge Kutta methods to calculate the first few values for both the Adams-Moulton and Adams-Bashforth methods.

2.5. 4th Order Runge Kutta

Of the single step methods, the 4th order Runge-Kutta methods were proposed by Runge and Kutta in the beginning of the 20th century. The Runge method uses specific increments between the present and next number to predict the next number provided the present number is known. This means, the 4th order Runge-Kutta methods require just a single initial condition to reliably predict the next value. The 4^{th}Order Runge Kutta method is:

$$k_1 = hf(t_n\, , y_n) \qquad (9)$$

$$k_2 = hf\left(t_n + \frac{h}{2}\, , y_n + \frac{k_1}{2}\right) \qquad (10)$$

$$k_3 = hf\left(t_n + \frac{h}{2}\, , y_n + \frac{k_2}{2}\right) \qquad (11)$$

$$k_4 = hf(t_n + h, y_n + k_3) \qquad (12)$$

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \qquad (13)$$

Despite there being four separate calculations, since there is no need to know any value of y other than the previous one, this is a Runge-Kutta method instead of a linear multi-step method. For the purposes of our experiment, we will use the 4th order Runge-Kutta method to find unknown values in multi-step methods to allow the multi-step methods to kick-off.

3 Analysis

During our analysis, we conducted simulations of our three methods of choice, which is the Heun's method, Adams-Bashforth method, and Adams-Moulton method. Each method was simulated to solve two differential equations whose solutions were known and compare how accurate each method was. Each differential equation was solved twice, from the values $t_0 = 0$ to $t = 10$, while varying the step size from $h = 0.1$ and $h = 1$. Then, the true value of the function was calculated, and the error between the true value and the predicted value was generated in a table.

3.1 Heun's Method (Predictor-Corrector)

(i) First differential equation.

(i) Step size $h = 1$

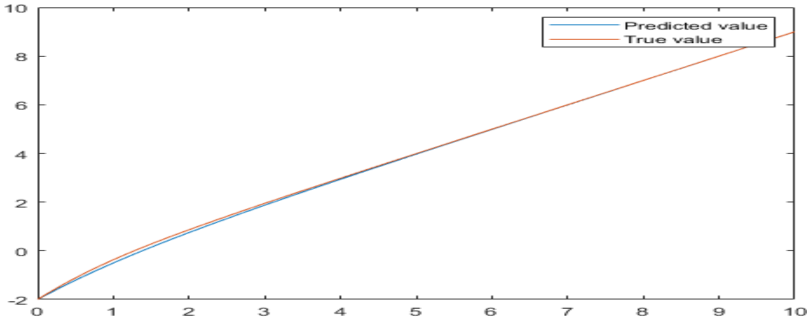

Fig. 1: Graphical comparison of solution of function (1) with step size h = 1: Heun's Method

From the graph generated when running the simulation of Heun's method, we can observe that there is an initial errorwhen using the step size 1. However, this error is quickly diminished to zero as the number of iterations increase.

Table *1:* comparison of solution of function (1) with step size h = 1: Heun's Method

t	y	ytrue	error
0	-2	-2	0
1	-0.5	-0.36788	-35.914
2	0.75	0.86466	13.261
3	1.875	1.9502	3.8567
4	2.9375	2.9817	1.4819
5	3.9688	3.9933	0.61384
6	4.9844	4.9975	0.26306
7	5.9922	5.9991	0.11503
8	6.9961	6.9997	0.051014
9	7.998	7.9999	0.022872
10	8.999	9	0.010346

From the table generated using the simulation, we can observe the accuracy of the system, with an initial errorof 35% and a final error of 0.01%. Thus, in just five iterations, the error was reduced to less than 1%.

(ii) Step size $h = 0.1$

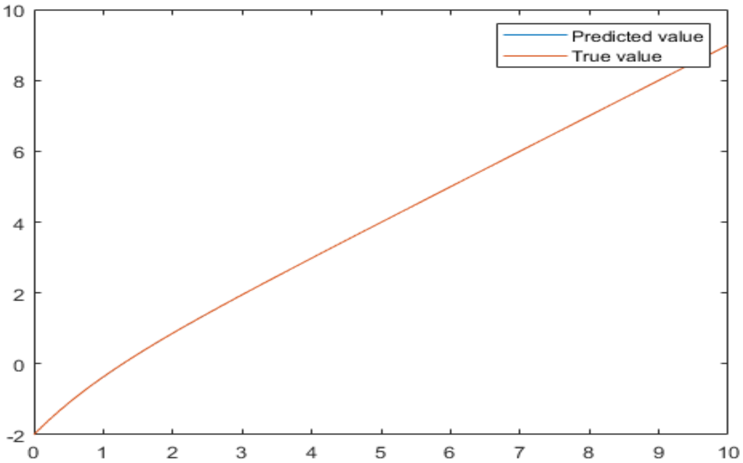

Fig. 2: Graphical comparison of solution of function (1) with step size $h = 0.1$: Heun's Method

When reducing the step size to0.1, we can observe an immediate increase in the accuracy of the system. The initial error that was previously present is now completely diminished. This leads to the idea that, reducing the step size will sometimes cause a method to converge while increasing the step size can cause a method to diverge.

Table 2: comparison of solution of function (1) with step size $h = 0.1$: Heun's Method

t	y	ytrue	error
0	-2	-2	0
0.1	-1.805	-1.8048	-0.0090081
0.2	-1.619	-1.6187	-0.018178
0.3	-1.4412	-1.4408	-0.027721
0.4	-1.2708	-1.2703	-0.037936
0.5	-1.1071	-1.1065	-0.049263
0.6	-0.9494	-0.94881	-0.062387
0.7	-0.79721	-0.79659	-0.07845
0.8	-0.64998	-0.64933	-0.099532
0.9	-0.50723	-0.50657	-0.12988
1	-0.36854	-0.36788	-0.17983

The table generated for the first ten values shows the error doesn't exceed 0.2% for the first ten values. Comparing previously to when the step size was 1, the error at the value of $t = 1$ is now 0.009% whereas previously this error was 35%.

1) Second differential equation.

(i) Step size $h = 1$

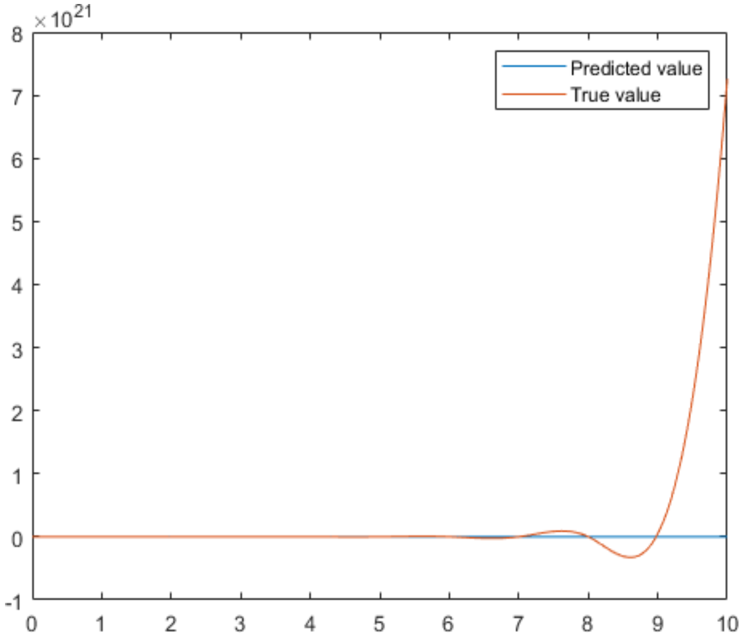

Fig. 3: Graphical comparison of solution of function (2) with step size h = 1: Heun's Method

When using Heun's method to calculate the value of the second differential equation, using step size 1, we can observe that the algorithm has difficulty in keeping up with the sudden increase in the value of the function. This is partially due to the fact that the step size is too high; however, another contributor to this factor is the nature of the function itself, which is an exponential function and as we can see, in just ten values of t, the value of the function is in the 21^{st} power of ten.

Table 3: comparison of solution of function (2) with step size h = 1: Heun's Method

t	Y	ytrue	error
0	2	2	0
1	26.5	208.38	87.283
2	479.75	30838	98.444
3	8864.9	4.5766 exp (+06)	99.806
4	1.6399 exp (+05)	6.7923 exp (+08)	99.976
5	3.0338 exp (+06)	1.0081 exp (+11)	99.997
6	5.6125 exp (+07)	1.4961 exp (+13)	100
7	1.0383 exp (+09)	2.2204 exp (+15)	100
8	1.9209 exp (+10)	3.2954 exp (+17)	100
9	3.5536 exp (+11)	4.8908 exp (+19)	100
10	6.5742 exp (+12)	7.2586 exp (+21)	100

From the table of this graph, we can see that the method is actually chasing after the true value; it's not a case of complete diversion. However, the true value increases at an extremely high rate, which is why the Heun's method hastrouble keeping up.

(ii) Step size $h = 0.1$

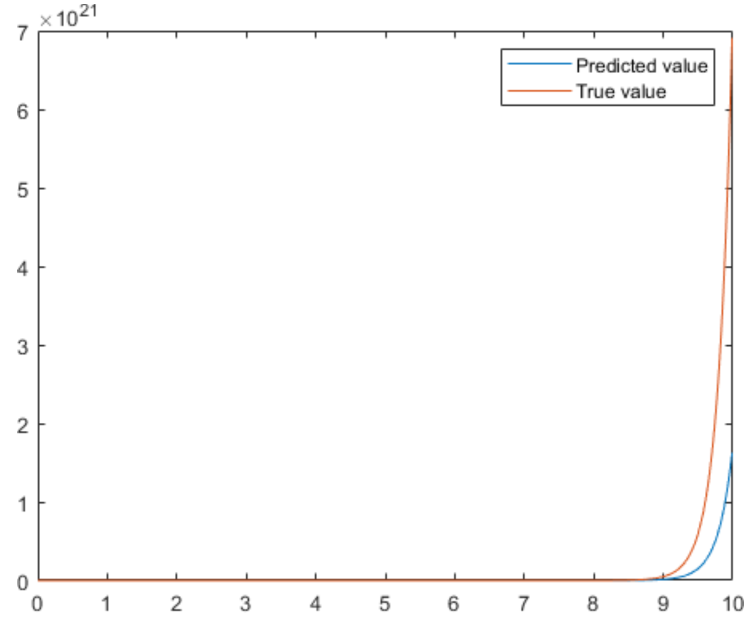

Fig. 4: Graphical comparison of solution of function (2) with step size $h = 0.1$: Heun’s Method

When we decrease the step size, we see a much closer graph. Where previously it only took two iterations for the errorto reach 100%, now, we can see a relatively accurate graph with deviations only becoming significant when the value of t is significantly large.

Table 4: comparison of solution of function (2) with step size $h = 0.1$: Heun’s Method

t	y	ytrue	error
0	2	2	0
0.1	2.875	2.9082	1.1419
0.2	4.2969	4.4056	2.4678
0.3	6.6074	6.8744	3.8832
0.4	10.362	10.945	5.3233
0.5	16.463	17.655	6.7523
0.6	26.378	28.72	8.154
0.7	42.489	46.962	9.5237
0.8	68.67	77.037	10.862
0.9	111.21	126.62	12.17
1	180.35	208.38	13.452

From the table of the first ten values of the function, we can observe that the initial error is 2% and this error is increasing with the number of iterations. The reduction in step size led to a significant increase in accuracy; however, thealgorithm still has trouble keeping up with the rapidly increasing true value of the function. Consequently, for exponential functions that rise quickly, the step size needs to further decrease to maintain accuracy.

3.2 Adams--*Bashforth Method*

1) First differential equation.

(i) Step size *h* = 1

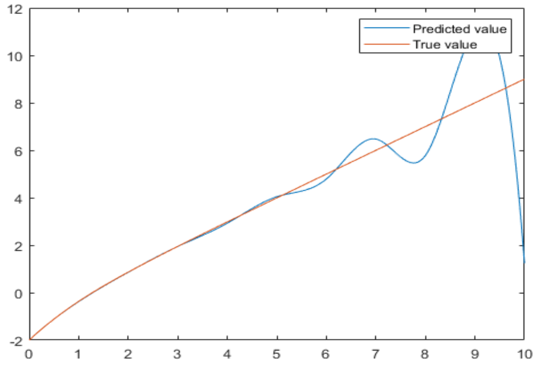

Fig. 5: Graphical comparison of solution of function (1) with step size *h* = 1: Adams-Bashforth

From the first iteration of the Adams-Bashforth simulation, we can observe a diverging pattern of the predicted value as compared with the true value. Therefore, for this function, the Adams-Bashforth algorithm is not approaching an error of zero.

Table 5: comparison of solution of function (1) with step size *h* = 1: Adams-Bashforth

t	y	ytrue	error
0	-2	-2	0
1	-0.375	-0.36788	-1.9356
2	0.85938	0.86466	0.61177
3	1.9473	1.9502	0.15113
4	2.9255	2.9817	1.8831
5	4.0427	3.9933	-1.2384
6	4.7903	4.9975	4.1457
7	6.4708	5.9991	-7.8638
8	5.7826	6.9997	17.387
9	11.069	7.9999	-38.367
10	1.2385	9	86.239

From the first ten values of the table, we can see this in effect, that the code is failing to achieve an accurate value after the first six values. For some reason, the predicted value has a final error of 86% which proves that the method is diverging for this function with this step size.

(ii) Step size *h* = 0.1

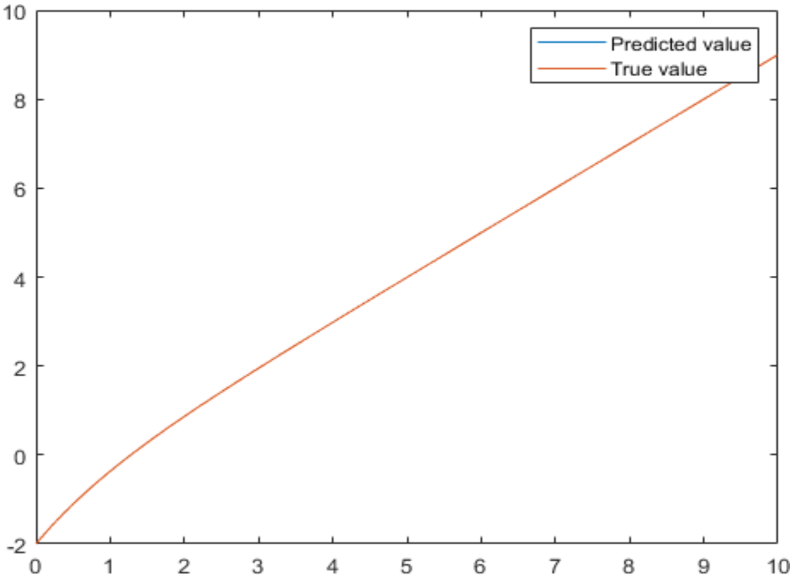

Fig. 6: Graphical comparison of solution of function (1) with step size h = 0.1: Adams-Bashforth

Now, with a step size of 0.1, the function suddenly is exactly and correctly predicted by the algorithm and the error has been diminished to zero from the start. This tells us that there are some conditions in which a function can be both diverging and converging when predicting the solution of a differential equation, depending on the step size that has been selected.

Table 6: comparison of solution of function (1) with step size h = 0.1: Adams-Bashforth

T	y	ytrue	error
0	-2	-2	0
0.1	-1.8048	-1.8048	-4.5414exp (-06)
0.2	-1.6187	-1.6187	-9.1632exp (-06)
0.3	-1.4408	-1.4408	-1.3973exp (-05)
0.4	-1.2703	-1.2703	-0.00024033
0.5	-1.1065	-1.1065	-0.00045037
0.6	-0.94882	-0.94881	-0.00072927
0.7	-0.79659	-0.79659	-0.0010325
0.8	-0.64934	-0.64933	-0.0014343
0.9	-0.50658	-0.50657	-0.0019868
1	-0.36789	-0.36788	-0.0028858

From the graph of the first ten values, we see a certain pattern in the error. First of all, it's always negative. Secondly, it seems to be increasing; however, since it's difficult to display all 100 values, we have to observe the graph to see that the error is actually not increasing. From the complete table, however, we observed that the maximum error reached 0.01% at a value of t = 0 and then began decreasing again.

2) Second differential equation.

(i) Step size h=1

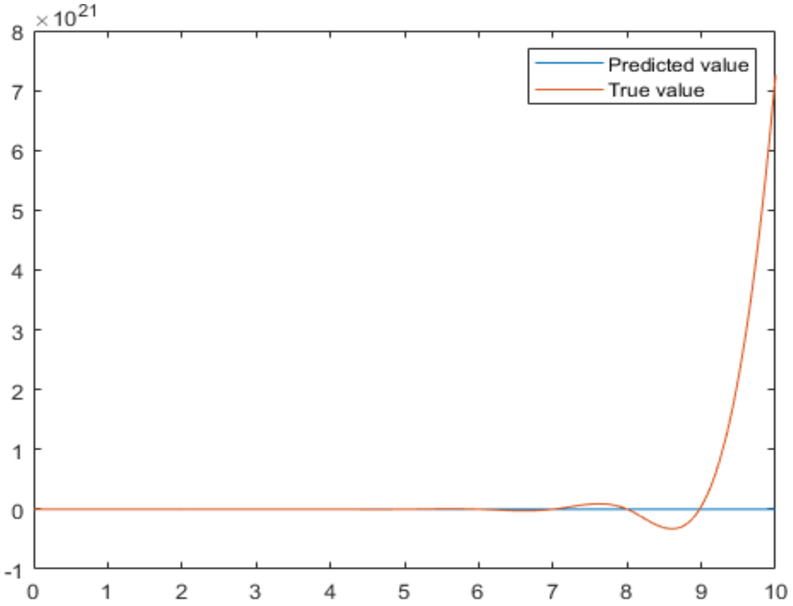

Fig. 7: Graphical comparison of solution of function (2) with step size *h* = 1: Adams-Bashforth

For the Adams-Bashforth method applied to the second equation we can observe, once again, a diverging result where the true value rises quite sharply as the algorithm struggles to rise with the true value. This is, once again, partially due to the large step size and partially due to the nature of the solution of the function which is an exponential curve.

Table 7: comparison of solution of function (2) with step size *h* = 1: Adams-Bashforth

t	Y	ytrue	error
0	2	2	0
1	92.125	208.38	55.79
2	5984	30838	80.595
3	3.9117exp (+05)	4.5766exp (+06)	91.453
4	4.8005exp (+06)	6.7923exp (+08)	99.293
5	5.5044exp (+07)	1.0081exp (+11)	99.945
6	6.2975exp (+08)	1.4961exp (+13)	99.996
7	7.2053exp (+09)	2.2204exp (+15)	100
8	8.2441exp (+10)	3.2954exp (+17)	100
9	9.4326exp (+11)	4.8908exp (+19)	100
10	1.0792exp (+13)	7.2586exp (+21)	100

The table provides further evidence of the diverging nature of the solution. The error quickly reaches 80% in two iterations and becomes 100% in the first seven iterations. This also reinforces the idea that, the nature of the solution for any method relies on the step size.

(ii) Step size *h* = 0.1

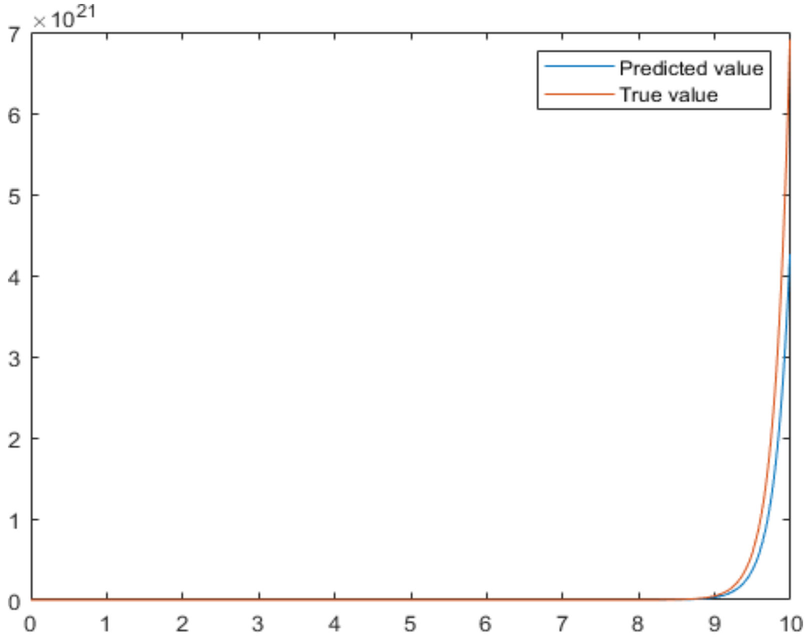

Fig. 8: Graphical comparison of solution of function (2) with step size $h = 0.1$: Adams-Bashforth

As we proceed to run the simulation with a step size of 0.1, we immediately notice a change in the accuracy of the system as it gets closer and closer to the actual solution. Compared to the previous run at $h = 1$, the performance is much better; however, it's still not quite convergent as the higher we proceed in values of t, the more the solution diverges.

Table 8: comparison of solution of function (2) with step size $h = 0.1$: Adams-Bashforth

t	y	Ytrue	error
0	2	2	0
0.1	2.9078	2.9082	0.013661
0.2	4.4043	4.4056	0.029732
0.3	6.8711	6.8744	0.04712
0.4	10.897	10.945	0.43445
0.5	17.487	17.655	0.95196
0.6	28.301	28.72	1.4567
0.7	46.047	46.962	1.9484
0.8	75.158	77.037	2.4396
0.9	122.91	126.62	2.9297
1	201.26	208.38	3.4163

The first ten readings from the table also reinforce the idea that, as the function is rising, the error between the predicted and the true value is increasing. However, where previously at $t = 1$ the error was 55%, this time at $t = 1$ the error is only 3.4%. In fact, from the full table of data, we observed that the maximum error in the simulation only reached 38% and never reached 100%. This can be largely attributed to the exponential function of the solution.

4 Adams-Moulton Method

4.1 First differential equation.

(i) Step size $h = 1$

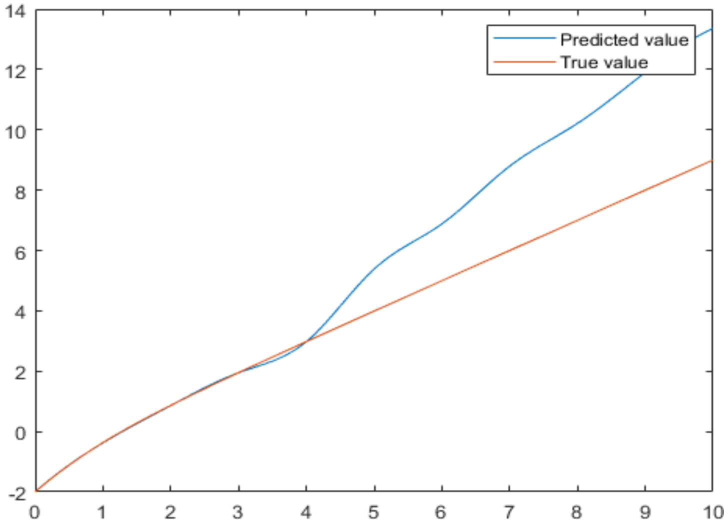

Fig. 9: Graphical comparison of solution of function (1) with step size $h = 1$: Adams-Moulton

When we ran the simulation for $h = 1$ using the Adams-Moulton method, the initial result showed the worst performance. After a certain point, the predicted value deviates from the true value and never re-converges. This is by far the worst result achieved so far.

Table 9: comparison of solution of function (1) with step size $h = 1$: Adams-Moulton

t	y	ytrue	error
0	-2	-2	0
1	-0.375	-0.36788	-1.9356
2	0.85938	0.86466	0.61177
3	1.9473	1.9502	0.15113
4	2.9787	2.9817	0.099052
5	5.3837	3.9933	-34.82
6	6.8815	4.9933	-37.699
7	8.7942	5.9991	-46.591
8	10.213	6.9997	-45.908
9	11.903	7.9999	-48.792
10	13.355	9	-48.39

The table displays results that show how the initial error is just about 2% or less; after values of $t = 5$ the graph suddenly deviates, and the error seems to increase.

(ii) Step size $h = 0.1$

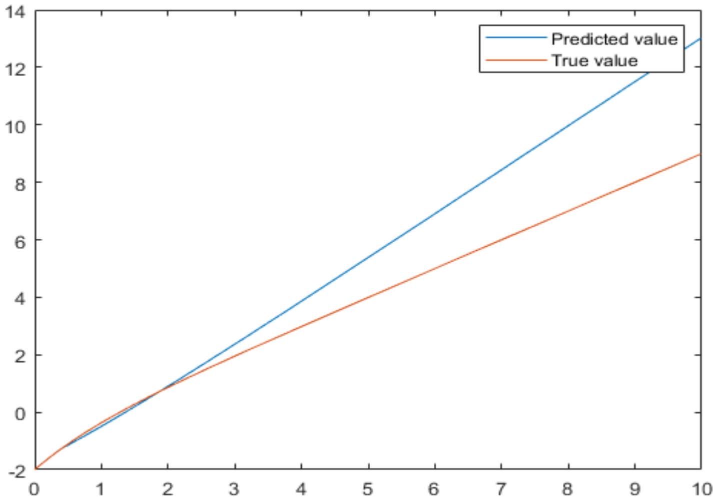

Fig. 10: Graphical comparison of solution of function (1) with step size $h = 0.1$: Adams-Moulton

When we decrease the step size, the result is counter-intuitive. Previously, there was a significant improvement in the performance of the algorithm when the step size was reduced. However, in this case, the algorithm remains divergent and seems to never converge with the true value. This is, thus, the first instance of our experiment where reducing step size did not have a significant effect on the error.

Table 10: comparison of solution of function (1) with step size $h = 0.1$: Adams-Moulton

t	y	ytrue	error
0	-2	-2	0
0.1	-1.8048	-1.8048	-4.5414exp(-06)
0.2	-1.6187	-1.6187	-9.1632exp(-06)
0.3	-1.4408	-1.4408	-1.3973exp(-05)
0.4	-1.2703	-1.2703	-1.5272exp(-05)
0.5	-1.1451	-1.1065	-3.4861
0.6	-1.017	-0.94881	-7.187
0.7	-0.88784	-0.79659	-11.456
0.8	-0.75697	-0.64933	-16.577
0.9	-0.62465	-0.50657	-23.31
1	-0.49098	-0.36788	-33.461

The table provides further proof, with performance actually decreasing. Previously, with a large step size of $h = 1$, the function at $t = 1$ showed an error of 1.9% whereas in this case, the error at x=1 is 33%. This means the performance of the method has significantly deteriorated.

4.2 Second differential equation.

(i) Step size *h* = 1

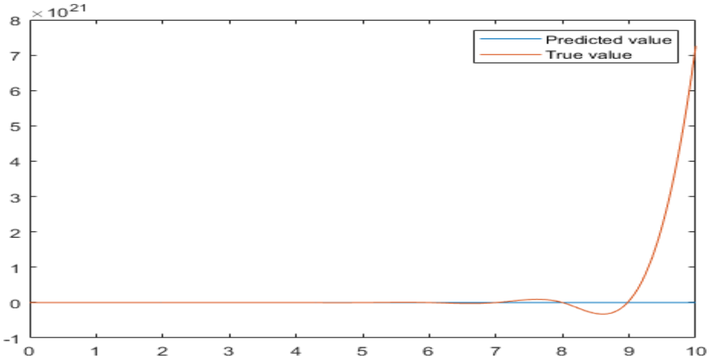

Fig. 11: Graphical comparison of solution of function (2) with step size *h* = 1: Adams-Moulton

From the figure above, running the algorithm for the second differential equation with a step size of 1, we can see not much difference in the performance between all the three methods. Each method quickly gets to a large error; however, this is the only method where the error didn't reach 100% even at the last value of *t* = 10 which means this algorithm has had the most effective solution to the differential equation.

Table 11: comparison of solution of function (2) with step size *h* = 1: Adams-Moulton

t	y	ytrue	Error
0	2	2	0
1	92.125	208.38	55.79
2	5984	30838	80.595
3	3.9117exp (+05)	4.5766exp (+06)	91.453
4	4.671exp (+07)	6.7923exp (+08)	93.123
5	3.1696exp (+09)	1.0081exp (+11)	96.856
6	2.0781exp (+11)	1.4961exp (+13)	98.611
7	1.3589exp (+13)	2.2204exp (+15)	99.388
8	8.8837exp (+14)	3.2954exp (+17)	99.73
9	5.8077exp (+16)	4.8908exp (+19)	99.881
10	3.17exp (+17)	7.2586exp (+21)	99.996

As the table provides further evidence of the fact that the algorithm is quickly adjusting, and the error never reaches 100% as it was previously assessed in the other two algorithms. This is by far the most effective solution.

(ii) Step size *h* = 0.1

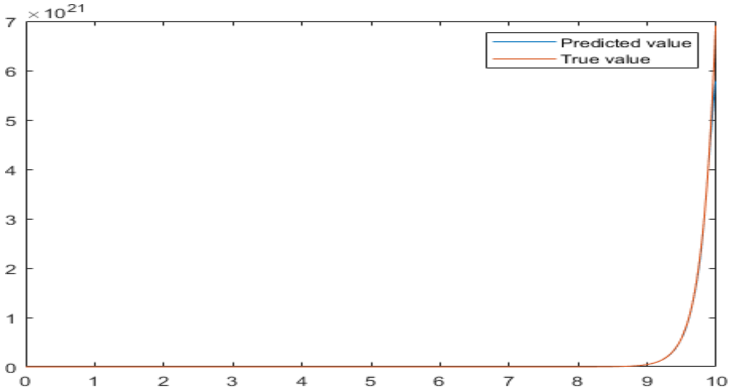

Fig. 12: Graphical comparison of solution of function (2) with step size *h* = 0.1: Adams-Moulton

Finally, as we decrease the step size for this algorithm, we can see a near overlap of the true and predicted values. We can see that the previous two methods were unable to accurately keep up with the rapidly rising nature of the exponential curve; however, with a step size of 0.1, the solution calculated by this algorithm is nearly overlapping with the true solution.

Table 12: comparison of solution of function (2) with step size *h* = 0.1: Adams-Moulton

t	y	ytrue	error
0	2	2	0
0.1	2.9078	2.9082	0.013661
0.2	4.4043	4.4056	0.029732
0.3	6.8711	6.8744	0.04712
0.4	10.94	10.945	0.04026
0.5	17.649	17.655	0.035501
0.6	28.71	28.72	0.034732
0.7	46.944	46.962	0.042072
0.8	77.005	77.037	0.042072
0.9	126.56	126.62	0.049264
1	208.26	208.38	0.058271

The first ten values of the table also provide concrete evidence of the accuracy of this method. The error never goes beyond 1% and is able to easily sustain itself with the rapidly rising values of the exponential curve.

5 Conclusion

From our experiments, we can conclude that there are different applications for the different types of methods. The Heun's method has shown the most robustness of the three. As it is a predictor corrector method, there always tends to be small amounts of error present using Heun's method; however, the error is rarely large enough unless the function is rapidly rising. The first differential equation was most accurately solved by Heun's method, even with a large step size. This lets us know that Heun's method is quite robust and works for larger step sizes too. The second differential equation was most accurately solved by the Adams-Moulton method with a small step size, which tells us that this method is most viable using low step sizes (thus requiring higher computing power) and is most effective at correcting itself, which means that the larger the change between two consequent solutions of the function the more viable this process gets. The worst two performances of the algorithms were on solving the first differential equation, which becomes a linear equation for certain parts of the curve which the Adams-Moulton and Adams-Bashforth methods are ineffective at predicting, since one of them is a predictor algorithm and tends to approach the previous value as accurately as possible, thus leading towards overshoots in the prediction of the solution. The Adams-Moulton method is a corrector method and tends to be more stable; we can observe that the error is increasing; however, there are no overshoots and the graph produced is always stable. Although the computation time of each approach is important but not mentioned in this paper, thanks to parallel computing there is most of time an adaptive solution to have a reasonable execution time for a more complicated problem which not the case.

Acknowledgements

The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publishing of this project.

References

[1] John C Butcher. ”Numerical methods for ordinary differential equations in the 20th century”. In: Journal of Computational and Applied Mathematics 125.1-2 (2000), pp. 1-29.

[2] John Charles Butcher. The numerical analysis of ordinary differential equations: Runge- Kutta and general linear methods. Wiley- Interscience, 1987

[3] Karl Heun et al. ”Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabha¨ngigen Vera¨nderlichen”. In: Z. Math. Phys 45 (1900), pp. 23-38.

[4] Sümeyra Kara, Ömer Ünsal, Analytical solutions to new forms of two nonlinear partial differential equations via two variable expansion method, Partial Differential Equations in Applied Mathematics, Volume 5, (2022), https://doi.org/10.1016/j.padiff.2021.100210

[5] Zayed EME, Hoda Ibrahim SA. Exact solutions of nonlinear evolution equations in mathematical physics using the modified simple equation method. Chinese Phys Lett. (2012);29(6):060201.

[6] Carmen Chicone. Ordinary differential equations with applications. Vol. 34. Springer Science & Business Media, 2006.

[7] David Francis Griffiths and Desmond J Higham. Numerical methods for ordinary differential equations: initial value problems. Vol. 5. Springer, 2010.

[8] John Charles Butcher. The numerical analysis of ordinary differential equations: Runge- Kutta and general linear methods. Wiley- Interscience, 1987

[9] Roger Alexander. “Diagonally implicit RungeKutta methods for stiff ODE’s”. In: SIAM Journal on Numerical Analysis 14.6 (1977), pp. 1006-1021.

[10] JR Cash. ”Diagonally implicit Runge-Kutta formulae with error estimates”. In: IMA Journal of Applied Mathematics 24.3 (1979), pp. 293-301.

[11] David Francis Griffiths and Desmond J Higham. Numerical methods for ordinary differential equations: initial value problems. Vol. 5. Springer, 2010.

[12] Francis Bashforth and John Couch Adams. An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid. University Press, 1883.

[13] Carl Runge. ”U¨ ber die numerische Auflo¨sung von Differentialgleichungen”. In: Mathematische Annalen 46.2 (1895), pp. 167178.

[14] FR Moulton. ”Discussions: New Methods in Exterior Ballistics”. In: The American Mathematical Monthly 35.5 (1928), pp. 246- 250.