The International Journal of Engineering and Science (IJES)
|| Volume || 13 || Issue || 4 || Pages || PP 58-63 || 2024 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Numerical Methods for Solving First Order Ordinary Differential Equations

M.U. UWAEZUOKE
Department of Mathematics , Faculty of physical sciences Imo State University. P.m.b 2000, Owerri, Nigeria.

ABSTRACT

***Abstract**. In this paper, we use both Newton's Interpolation and Lagrange Polynomial to create cubic polynomials for solving the initial value problem. By this new method, it is simple to solve linear and nonlinear first order ordinary differential equations and to yield and implement actual precise results.Some numerical examples are provided to test the performance and illustrate the efficiency of the method.*

***Key Words and Phrases:** Numerical method, Initial value problems, Newton's interpolation, Lagrange polynomial.*

Date of Submission: 02-04-2024 Date of acceptance: 13-04-2024

I. Introduction

Many mathematical models in science and engineering fields *([3],[4],[5],[6],[7],[10],[13])* can be formulated in the form linear and nonlinear ordinary differential equations *([9],[12])* which need an analytical method *([2],[15].[17])* to solve the exact equations. However in some problems, we can obtain the exact solution by analytical method in *[7]* for example $y' = x^2 + y^2$. Therefore the numerical method is an important tool to solve this kind of problems such as Euler's method, Runge-Kutta method *([11],[14])* and Runge-Kutta-Fehlberg method *([8],[18])*. Many methods have been widely developed by a lot of researchers *([1],[11],[14] and [16])* to solve these problems. Some problems in form of partial differential equations can be converted to ordinary differential equations form *([10],[13])*. In this article, we consider only the first order ordinary differential equations with an initial condition (initial value problems) in form:

$$\frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0 \tag{1}$$

Where $f(x,y)$, is a known function and the value of initial conditions x_0, y_0 are known numbers. In 2018, to find solutions of (1), [14] used Newton's interpolation and three points to build a quadratic equation by Lagrange method. Also [11] and [16] used Newton's interpolation and constructed a quadratic equation by Aitken's method to solve the same problems. Moreover [20] gave an idea to solve these problems for combining Pi-card's method and Taylor's Series. Furthermore [21] proposed a new solving technique by improved Euler's method.

Ultimately [1] applied all techniques from [11], [14], [16], [19], [20] and [21] to approximate the solution of (1) and other systems of first order ordinary differential equations. The goal of this study is to estimate approximated solutions and relative errors by comparing the results of our new method with other methods such as Euler's method and methods in [11] and [14].

II. Our proposed method

To improve the accuracy of in [11] and [14] which are the same quadratic solutions, we combine Newton's interpolation and Lagrange's method for solving (1) to create higher degree polynomial. Newton's interpolation is applied to find four valuesy_i for i= 0,1,2,3 to form a cubic polynomial, $P_3(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$. This polynomial approximates solutions of (1) by using Lagrange's method as following.

2.1 Newton's interpolation method

This method uses the initial point (x_0, y_0) from (1) to estimate the values of a function from any intermediate values of the independent variables. The general form of the n^{th} degree polynomial that goes through n+1 points $((x_i, y_i)$ $for\ i = 0,1, \ldots n$ is written as

$f_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}),$
(2)

Where $a_0, a_1, a_2, a_3, \ldots$ are given by

$$a_0 = y_0, \tag{3}$$

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0}, \tag{4}$$

$$a_2 = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}, \tag{5}$$

$$a_3 = \frac{\frac{\frac{y_3 - y_2 - y_2 - y_1}{x_3 - x_2 - x_2 - x_1}}{x_3 - x_1} \qquad \frac{\frac{y_2 - y_1 - y_1 - y_0}{x_2 - x_1 - x_1 - x_0}}{x_2 - x_0}}{x_3 - x_0}, \ldots . \tag{6}$$

Since y_1, y_2 and y_3 in (4) - (6) are unknown values, we use differential values for approximation i.e

$$\frac{y_i - y_i - 1}{x_i - x_i - 1} \approx \frac{dy}{dx}|(x_i - 1, y_i - 1)$$

For our method, we require three more values y_1, y_2 and y_3 which are given by

$$y_1 = a_0 + a_1(x_1 - x_0), \tag{7}$$
$$y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1), \tag{8}$$
$$y_3 = a_0 + a_1(x_3 - x_0) + a_2(x_3 - x_0)(x_3 - x_1) + a_3(x_3 - x_0)(x_3 - x_3), \tag{9}$$

where $x_{i+1} = x_i + h$ and the step size h is very small constant.

2.2. Lagrange's method

Lagrange's method is a method to find a n^{th} degree polynomial that takes on certain values at an arbitrary point. We have only (x_0,y_0), (x_1,y_1),(x_2,y_2) and (x_3,y_3) to build the 3rd degree polynomial equation (cubic function), $P_3(x_i) = c_0 + c_1x + c_2x^2 + c_3x^3$

$$\begin{aligned} P_3(x) = {} & \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}y_1 \\ & + \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}y_3. \end{aligned} \tag{10}$$

Then we apply (10) to approximate $y_i = P_3(x_i)$ where $i = 4,5, \ldots 8$ which are solutions of (1)as shown in Figure 1

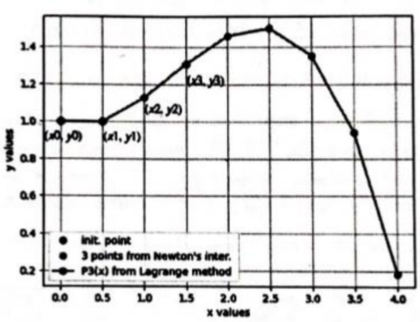

Figure 1: Showing an initial point, (x_0, y_0), the red points from the Newton's interpolation, $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3) and the solution points by (10).

2.3. Algorithm of our method

Consider IVP dy/dx = f(x,y) with initial conditions $y(x_0) = y_0$

1. Define function f(x,y)

2. Set values of initial condition (x_0, y_0), number of steps (n) and steps size (h)
3. Use Newton's interpolation method to find (x1, y1), (x2, y2) and (x3, y3)
4. Use Lagrange's method to find P3(x)
5. Set i = 4
6. Loop while i <= n
    yi = P3(xi)
    xi = xi + h
    i = i + 1
7. Display yi as results

III. Numerical results

We will use our method to find the numerical solutions and relative errors and compare results with Euler's method, methods of [11] and [14] in the following examples.

Example 1. *Consider the initial value problem*

$$\frac{dy}{dx} = 1 - y, \qquad y(0) = 0.$$

Then take the step size h = 0.1 and use Newton's interpolation

$$a_0 = 0.0,\ a_1 = 1.0,\ a_2 = -0.499999,\ a_3 = 0.166667$$

and

$$y_1 = 0.1,\ y_2 = 0.19,\ y_3 = 0.271.$$

Apply (0,0), (0.1, 0.1), (0.2, 0.19) and (0.3, 0.271) *to find cubic polynomial by* (10). *Then we obtain*

$$P_3(x) = 0.166667x^3 - 0.549997x^2 + 1.053333x \tag{11}$$

In order to approximate the solutions, we substitute x_i in $P_3(x)$ to get $y_i = P_3(x_i)$ for i = 4,5, …20 and compute relative error, $\left|\frac{y_{xi} - y_i}{y_{xi}}\right|$

where y_{xi}, is exact solutions at x_i, as shown in Table 1 and Figure 2.

The approximate solutions are close to exact solutions where $x \in [0,0.8]$ and relative errors when $x \geq 0.9$ highly increase as shown in Figure 2.

Table 1: Showing results of Example 1 with $h = 0.1$ on $x_i \in [0, 2]$

i	x_i	Approx. Sol. y_i			Exact	Relative errors		
		Euler	[11],[14]	Present	Solution	Euler	[11],[14]	Present
0	0.00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
1	0.10	0.100000	0.100000	0.100000	0.095163	0.050833	0.050833	0.050833
2	0.20	0.190000	0.190000	0.190000	0.181282	0.048090	0.048090	0.048090
3	0.30	0.271000	0.270000	0.271000	0.259215	0.045465	0.041608	0.045465
4	0.40	0.343900	0.340000	0.344000	0.329717	0.043017	0.031189	0.043320
5	0.50	0.409510	0.400000	0.410000	0.393490	0.040712	0.016543	0.041957
6	0.60	0.468559	0.450000	0.470000	0.451186	0.038506	0.002628	0.041699
7	0.70	0.521703	0.490000	0.525000	0.503399	0.036361	0.026617	0.042910
8	0.80	0.569533	0.520000	0.576000	0.550681	0.034234	0.055714	0.045978
9	0.90	0.612580	0.540000	0.624000	0.593488	0.032169	0.090124	0.051412
10	1.00	0.651322	0.550000	0.670000	0.632211	0.030228	0.130037	0.059773
11	1.10	0.686189	0.550000	0.715000	0.667221	0.028429	0.175686	0.071609
12	1.20	0.717570	0.540000	0.760000	0.698868	0.026761	0.227322	0.087473
13	1.30	0.745813	0.520000	0.806000	0.727479	0.025202	0.285203	0.107935
14	1.40	0.771232	0.490000	0.854000	0.753364	0.023718	0.349584	0.133583
15	1.50	0.794109	0.450000	0.905000	0.776808	0.022272	0.420706	0.165025
16	1.60	0.814698	0.400000	0.960000	0.798069	0.020837	0.498790	0.202904
17	1.70	0.833228	0.340000	1.020000	0.817336	0.019444	0.584015	0.247956
18	1.80	0.849905	0.270000	1.086000	0.834771	0.018130	0.676558	0.300956
19	1.90	0.864915	0.190000	1.159000	0.850528	0.016915	0.776609	0.362683
20	2.00	0.878423	0.100000	1.240000	0.864758	0.015803	0.884361	0.433928

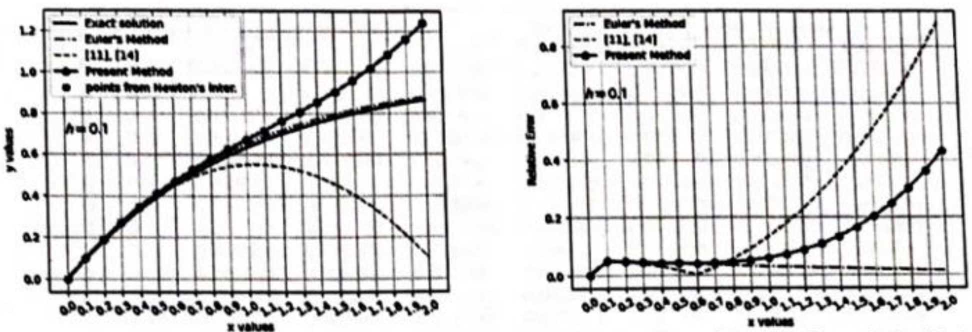

Figure 2: Comparing graph of approximate solutions (left) and relative errors (right) in Example 1 with $h = 0.1$ on $x_i \in [0, 2]$

Example 2. *Consider the initial value problem*

$$\frac{dy}{dx} = x^2 - y, \quad y(0) = 1.$$

Then take the step size h = 0.1 and use Newton's interpolation

$a_0 = 1.0$, $a_1 = -1.0$, $a_2 = 0.549999$, $a_3 = 0.149999$

and

$y_1 = 0.9$, $y_2 = 0.811$, $y_3 = 0.7339$.

Apply (0, 1.0), (0.1,0.9), (0.2, 0.811) and (0.3, 0.7339) to find cubic polynomial by (10). Then we obtain

$$P_3(x) = 0.149999x^3 + 0.505x^2 - 1.052x + 1 \tag{12}$$

In order to approximate the solutions, we substitute xi in $P_3(x)$ to get $y_i = P_3(x_i)$ for $i = 4,5,\ldots20$ in (12) and relative error as shown in Table 2 and Figure 3. The approximate solutions are close to exact solutions where $\in [0,0.6]$ and relative errors when x ≥ *0.7* highly increase as shown in Figure 3.

Table 2: Showing results of Example 2 with $h = 0.1$ on $x_i \in [0, 2]$

i	x_i	Approx. Sol. y_i			Exact	Relative errors		
		Euler	[11], [14]	Present	Solution	Euler	[11], [14]	Present
0	0.00	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000
1	0.10	0.900000	0.900000	0.900000	0.905163	0.005703	0.005703	0.005703
2	0.20	0.811000	0.811000	0.811000	0.821212	0.012435	0.012435	0.012435
3	0.30	0.733900	0.733000	0.733900	0.749005	0.020167	0.021368	0.020167
4	0.40	0.669510	0.666000	0.669600	0.689391	0.028839	0.033930	0.028708
5	0.50	0.618559	0.610000	0.619000	0.643129	0.038204	0.051513	0.037519
6	0.60	0.581703	0.565000	0.583000	0.610887	0.047773	0.075115	0.045650
7	0.70	0.559533	0.531000	0.562500	0.593241	0.056820	0.104916	0.051818
8	0.80	0.552580	0.508000	0.558400	0.590676	0.064496	0.139968	0.054642
9	0.90	0.561322	0.496000	0.571600	0.603586	0.070023	0.178245	0.052994
10	1.00	0.586189	0.495000	0.603000	0.632280	0.072896	0.217119	0.046308
11	1.10	0.627570	0.505000	0.653500	0.677123	0.073181	0.254198	0.034888
12	1.20	0.685813	0.526000	0.724000	0.738595	0.071462	0.287837	0.019760
13	1.30	0.761232	0.558000	0.815400	0.817114	0.068389	0.317109	0.002097
14	1.40	0.854109	0.601000	0.928600	0.913027	0.064531	0.341750	0.017056
15	1.50	0.964698	0.655000	1.064500	1.026610	0.060307	0.361978	0.036908
16	1.60	1.093228	0.720000	1.224000	1.158067	0.055989	0.378274	0.056934
17	1.70	1.239905	0.796000	1.408000	1.307528	0.051718	0.391218	0.076841
18	1.80	1.404915	0.883000	1.617400	1.475055	0.047551	0.401378	0.096502
19	1.90	1.588423	0.981000	1.853100	1.660676	0.043508	0.409277	0.115871
20	2.00	1.790581	1.090000	2.116000	1.864644	0.039720	0.415438	0.134801

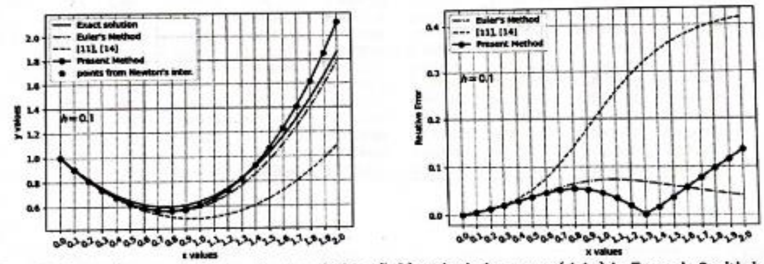

Figure 3: Comparing graph of approximate solutions (left) and relative errors (right) in Example 2 with $h = 0.1$ on $x_i \in [0, 2]$

Example 3. *Consider the initial value problem*

$$\frac{dy}{dx} = \frac{x-y}{e^{3+y}} \qquad y(0) = 1.$$

Then take the step h = 0.1 and use Newton's interpolation

$$a_0 = 1.0,\ a_1 = -0.367879,\ a_2 = 0.348868,\ a_3 = -0.129608$$

and

$$y_1 = 0.963212,\ y_2 = 0.933401,\ y_3 = 0.909791.$$

Apply (0,0.1), (0.1, 0.963212), (0.2, 0.933401) and (0.3, 0.909791) to find cubic polynomial by (10).Then we obtain

$$P_3(x) = -0.129608x^3 + 0.387751x^2 - 0.405358x + 1 \qquad (13)$$

In order to approximate the solutions, we substitute xi in $P_3(x)$ to get $y_i = P_3(x_i)$ for i = 4,5, …20 in(13) and relative error as shown in Table 3 and Figure 4. The approximate solutions are close to exact solutions where $x \in [0,0.8]$ and relative error when $x \geq 0.9$ highly increase as shown in Figure 4.

Table 3: Showing results of Example 2 with $h = 0.1$ on $x_i \in [0, 2]$

i	x_i	Approx. Sol. y_i Euler	Approx. Sol. y_i [11],[14]	Approx. Sol. y_i Present	Exact Solution	Relative errors Euler	Relative errors [11],[14]	Relative errors Present
0	0.00	1.000000	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000
1	0.10	0.963212	0.963212	0.963212	0.966759	0.003669	0.003669	0.003669
2	0.20	0.933401	0.933401	0.933401	0.940092	0.007117	0.007117	0.007117
3	0.30	0.909791	0.910568	0.909791	0.919197	0.010233	0.009387	0.010233
4	0.40	0.891603	0.894712	0.891602	0.903288	0.012936	0.009493	0.012937
5	0.50	0.878092	0.885834	0.878057	0.891643	0.015198	0.006515	0.015237
6	0.60	0.868562	0.883933	0.868380	0.883609	0.017029	0.000366	0.017235
7	0.70	0.862378	0.889009	0.861791	0.878595	0.018457	0.011854	0.019125
8	0.80	0.858974	0.901063	0.857514	0.876075	0.019520	0.028522	0.021186
9	0.90	0.857852	0.920094	0.854771	0.875591	0.020259	0.050826	0.023778
10	1.00	0.858579	0.946102	0.852784	0.876747	0.020723	0.079105	0.027332
11	1.10	0.860783	0.979088	0.850776	0.879215	0.020964	0.113594	0.032346
12	1.20	0.864150	1.019051	0.847968	0.882720	0.021037	0.154444	0.039369
13	1.30	0.868413	1.065991	0.843583	0.886895	0.020840	0.201936	0.048835
14	1.40	0.873349	1.119909	0.836845	0.891563	0.020430	0.256119	0.061373
15	1.50	0.878771	1.180804	0.826974	0.896627	0.019915	0.316940	0.077684
16	1.60	0.884528	1.248677	0.813193	0.902001	0.019371	0.384342	0.098456
17	1.70	0.890492	1.323527	0.794725	0.907601	0.018850	0.458270	0.124368
18	1.80	0.896562	1.405354	0.770792	0.913352	0.018383	0.538677	0.156085
19	1.90	0.902655	1.494159	0.740616	0.919185	0.017984	0.625525	0.194269
20	2.00	0.908704	1.589941	0.703420	0.925037	0.017657	0.718787	0.239576

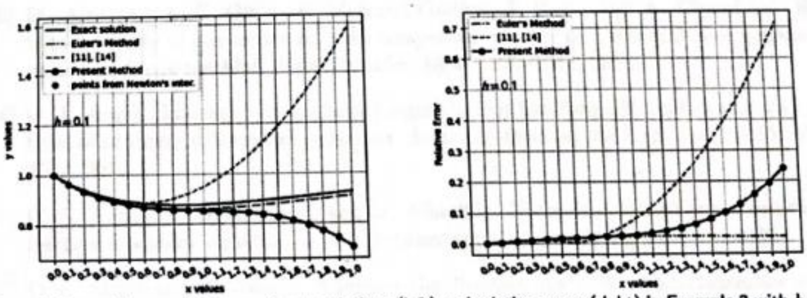

Figure 4: Comparing graph of approximate solutions (left) and relative errors (right) in Example 3 with $h = 0.1$ on $x_i \in [0, 2]$

IV. Conclusions

In this article, we studied many numerical techniques for solving initial value problems. Then we decided to combine the Newton's interpolation and Lagrange's method to construct cubic polynomials as the solutions of linear and non-linear of ordinary differential equations. We compared our numerical results and relative errors with the results of Euler's method, methods from [11] and [14] and exact solutions. From example 1-3, where we compare the result in other methods, our method gives numerical approximated solutions which is much closer to the exact solution as shown in Table 1-3. Also our relative errors are nearly close to zero and much smaller that errors from [11] and [14] methods are shown in Figure 2-4. Notice that to compute each yi in each step of Euler's method is so much time-consuming. Therefore, our method is much more accurate and simpler to find approximated solutions directly from $y_i = P_3(x_i)$ at any value of x_i. For the future work, we aim to apply our method to solve other problems of first differential equations or higher order differential equations.

References

[1]. J. D. Abraha. Comparison of numerical methods for system of first order ordinary differential equations. Pure Appl. Math. J., 9:32-36, 2020.

[2]. F. M. Al-Askar, C. Cesarano, and W. W. Mohammed. The analytical solutions of stochastic-fractional drinfel'd-sokolov-wilson equations via (g'/g)-expansion method. Symmetry, 14(10), 2022.

[3]. M. Alkasassbeh, Z. Omar, F. Mebarek-Oudina, J. Raza, and A. Chamkha. Heat transfer study of convective fin with temperature-dependent internal heat generation by hybrid block method. Heat Transfer, 48(4):1225-1244, 2019.

[4]. O. A. Arqub. The reproducing kernel algorithm for handling differential algebraic systems of ordinary differential equations. Math. Methods in the App. Sci., 39(15):4549-4562, 2016.

[5]. O. A. Arqub, H. O. Alsulami, and M. Alhodaly. Numerical hilbert space solution of fractional sobolev equation in (1+1)-dimensional space. Math. Sct., 16, 2022.

[6]. O. A. Arqub and B.'Maayah. Adaptive the dirichlet model of mobile/immobile advection/dispersion in a time-fractional sense with the reproducing kernel computational approach: Formulations and approximations. Int. J. Modern Physics B, 37, 2022.

[7]. H. Badawi, N. Shawagfeh, and O. A. Arqub. Fractional conformable stochastic inte-grodifferential equations: existence, uniqueness, and numerical simulations utilizing the shifted legendre spectral collocation algorithm. Math. Problems in Engineering, 2022,

[8]. R. Djebali, F. Mebarek-Oudina, and C. Rajashekhar. Similarity solution analysis of dynamic and thermal boundary layers: further formulation along a vertical flat plate. Physica Scripta, 96(8), 2021.

[9]. M. Farhan, Z. Omar, F. Mebarek-Oudina, Z. Shah J. Raza, R. V Choudhari, and O. D. Makinde. Implementation of the one-step one-hybrid block method on the equation of a circular sector oscillator. Comput. Math. Model, 31: 116-132, 2020.

[10]. M. Hassan, F. Mebarek-Oudina, A. Faisal, A. Ghafar, and A. I. Ismail. Thermal energy and mass transport of shear thinning fluid under effects of low to high shear rate viscosity. Int. J. Thermofluids, 15, 2022.

[11]. N. D. Ide. Using newton's interpolation and aitken's method for solving first order differential equation. World Appl. Sci. J., 38:191-194, 2022.

[12]. A. Ishkhanyan and C. Cesarano. Generalized-hypergeometric solutions of the general fuchsian linear ode having five regular singularities. Axioms, 8(3): 102, 2019.

[13]. U. Khan, F. Mebarek-Oudina, A. Zaib, A. Ishak, S. A. Bakar, E. M. Sherif, and D. Baleanu. An exact solution of a casson fluid flow induced by dust particles with hybrid nanofluid over a stretching sheet subject to lorentz forces. Waves in Random and Complex Media, 32, 2022.

[14]. F. C. Kosgei. Solution of first order differential equation using numerical newton's interpolation and Lagrange method. Int. J. Dev. Res., 8: 18973-18976, 2018.

[15]. A. Lupica, C. Cesarano, F. Crisanti, and A. Ishkhanyan. Analytical solution of the three-dimensional laplace equation in terms of linear combinations of hypergeometric functions. Mathematics, 9(24), 2021.

[16]. J. C. Mbagwu and N. D. Ide. Comparison of newton's interpolation and aitken's methods with some numerical methods for solving system of first and second order differential equation. World Sci. News, 164: 108-121, 2022.

[17]. W. W. Mohammed, F. M. Al-Askar, C. Cesarano, T. Botmart, and M. El-Morshedy. Wiener process effects on the solutions of the fractional (2 + 1)-dimensional heisenberg ferromagnetic spin chain equation. Mathematics, 10(12), 2022.

[18]. J. Raza, F. Mebarek-Oudina, and L. Ali Lund. The flow of magnetized convective casson liquid via a porous channel with shrinking and stationary walls. Pramana, 96, 2022.

[19]. A. Setia, Y.Liu, and A. S. Vatsala. Numerical solutions of fractional integro-differential equations with nonlocal boundary conditions. J. Frac. Cal. Appl., 8:155-165, 2014.

[20]. I. Singh and G. Singh. Numerical solutions of first order ordinary differential equations. Int. J. Sci. Res. Rev., 8:593-602, 2019.

[21]. S. Tapaswini and S. Chakraverty. A new Approach to fuzzy initial value problem by improved euler method. Fuzzy Information and Engineering, 4:293-312, 2012