Explicit Solution of First-Order Differential Equation Using Aitken's and Newton's Interpolation Methods

Salisu Ibrahim[1]

[1]Mathematics Education Departments, Faculty of Education, Tishk International University-Erbil, Kurdistan Region, Iraq
Correspondence: Salisu Ibrahim, Mathematics Education Departments, Faculty of Education, Tishk International University-Erbil, Kurdistan Region, Iraq
Email: ibrahimsalisu46@yahoomail.com
Doi: 10.23918/eajse.v9i1p46

Abstract: The struggle to find the analytic solution of several differential equations leads to several issues like difficulty in finding solutions, singularities, convergent issues, and stability. Because of these problems, most of the researchers come up with explicit approaches such as the Runge Kutta method, Euler's method, and Taylor's polynomial method for finding numerical solutions to the ordinary differential equation. In this work, we combine both the Aitken methods and Newton's interpolation method (NIM) to solve first-order differential equations. The numerical results obtained provide minimal error. The result is supported by solving an example.

Keywords: Ordinary Differential Equation, Aitken's Method, Newton Interpolation Polynomial, Numerical Approximation

1. Introduction

A differential equation is a type of mathematical concept that shows the derivatives of unknown functions. It can be used to describe the relationship between one or more independent variables and one or more unknown functions (Zill, 2012). A differential equation is the change of the function with respect to time, so this gives us an equation that says that the derivative of the function is proportional to the function itself (Atkinson et al, 2009). "Ordinary derivatives are written using either the Leibniz notation $d^n y/dx^n$, or the prime notation $y', y'', y''', \dots y^{(n)}$, is that the former is used to denote the first three derivatives prime notation is used to denote only the first three derivatives; the fourth and up to *n* derivative is written $y^{(n)}$, the general form of first-order DE given by $\frac{dy}{dx} = f(x,y)$ and second-order is given by $\frac{d^2y}{dx^2} - \frac{dy}{dx}p(t) + q(t) = g(t)$, (Chasnov, 2014).

But the problem of finding a solution to some difficult differential equations has been a major problem for scientists to deal with, tackling these kinds of problems necessitates the authors in (Ibrahim, 2020; Ibrahim, & Isah, 2021; Isah, & Ibrahim, 2021; Ibrahim, & Isah, 2022; Salisu, 2022b) to introduce a numerical method for solving ODEs, partial differential equations (PDEs), and fractional differential equations (FDEs). Therefore, commutativity is very important from a practical point of view.

Newton's interpolation concept is a widely used technique in image processing and numerical analysis. It is unique in that it considers the various points in a function and finds an approximate value for those values (Zou, 2020).

Received: May 24, 2022
Accepted: December 15, 2022
Ibrahim, S. (2023). Explicit Solution of First-Order Differential Equation Using Aitken's and Newton's Interpolation Methods. *Eurasian Journal of Science and Engineering*, *9*(1),46-56.

In this process, we want to find the exact values of the function for the new points in the function (Kreyszig, 2011). “In the mathematical field of numerical analysis, a Newton polynomial, named after its inventor Isaac Newton, is an interpolation polynomial for a given set of data points. The Newton polynomial is sometimes called Newton's divided differences interpolation polynomial because the coefficients of the polynomial are calculated using Newton's divided differences method” (Obradovic, Mishra, & Mishra). Many arithmetic calculations are involved in Lagrange's interpolation. However, this approach works for both equal and unequally spaced points. Newton's divided difference method is theoretically highly influential because it may be used to construct numerous interpolation equations.

The authors (Ibrahim, & Koksal, 2021a) studied the commutativity with non-zero initial conditions (ICs), and their effects on the sensitivity were studied (Salisu, 2022a; Salisu, 2022c) while the realization and decomposition of a fourth-order LTVSs with nonzero ICs by cascaded two Second-Order commutative pairs was introduced by (Ibrahim, & Koksal, 2021b; Salisu, & Rababah, 2022). The authors (Rababah, & Ibrahim, 2016a; Rababah, & Ibrahim, 2016b; Rababah, & Ibrahim, 2018) come up with a numerical approximative process for degree reduction of curves and surfaces which approaches can be used to solve complex ODES, PDEs, and FDEs.

The Aitken method is a polynomial interpolation technique developed by mathematician Alexander Aitken. It's comparable to Neville's algorithm. As A. C. Aitken has lately created a method of practical interpolation that is especially well suited for computing machines; no differences or tables of interpolation coefficients are utilized, and the essential operations are most simply accomplished on current computing machines. Furthermore, the degree of the interpolating polynomial reduces by two at each stage, reducing the amount of labor required.

(Faith, 2018) investigated “Solution of First Order Differential Equation Using Numerical Newton’s Interpolation and Lagrange Method”. “Numerical study for Solving Bernoulli Differential Equations by using Newton's Interpolation and Aitken's Method” was studied by (Al Din, 2020a). “Solving Bernoulli Differential Equations by using Newton's Interpolation and Lagrange Methods” was studied by (Al Din, 2020b). “Comparison of Newton's Interpolation and Aitken's Methods with Some Numerical Methods for Solving System of First and Second Order Differential Equation” was investigated by (Mbagwu, 2021)

The above study investigates Aitken's method and Newton’s interpolation approach to solve a first-order differential equation. The purpose of this work is to find the numerical solution to the first-order differential equations using the proposed methods.

2. Preliminaries

In this section we introduce Euler’s method for solving the first and second-order differential equation, we also show the formula of Taylor polynomial and Runge-Kutta method for solving ODEs, PDEs, and FDEs, all these are numerical techniques that prove to be the good numerical method for finding the solution ODEs, but in this work, we are going to consider Aitken's method and Newton interpolation method to solve first-order ODEs, which will explain the next section.

Considering the first and second-order systems described as

$$\boldsymbol{y}'(x) + Q(x)y(x) = \mathrm{f}(x), \quad [1]$$

$$y''_B(x) + \mathrm{P}(x)\boldsymbol{y}'(x) + \mathrm{Q}(x)y(x) = \mathrm{f}(x), \quad [2]$$

Where $\mathrm{P}(x)$, $\mathrm{Q}(x)$ *and* $\mathrm{f}(x)$ are function of x.

2.1 Euler Method

Euler's method is a developed numerical solution to an initial value problem of the type

$$\boldsymbol{y}'(x) = f(x, y), \quad [3]$$

$$y(x_0) = y_0. \quad [4]$$

Beginning with the initial condition y_0, we create the rest of the solution using repeated formulas. To develop a numerical solution to an initial value issue of the form:

$$x_{n+1} = x_n + h, \quad [5]$$

$$y_{n+1} = y_n + hf(x_n, y_n), \quad [6]$$

Where h is the step function, x_{n+1} is the independent value that can be divided into h sub intervals and y_{n+1} is the solution numerical solution.

2.2 Taylor Series

The Taylor series can be defined as

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 +, \ldots \quad [7]$$

To determine a condition that must be true for a Taylor series to exist for a function, we first construct the nth degree Taylor mathematical model of that function.

$$f(x) = \sum_{i=0}^{n} \frac{f^{(i)}(a)}{i!}(x-a)^i. \quad [8]$$

This polynomial has a maximum degree of n and it is called the Taylor polynomial.

2.3 Runge-Kutta Method

We do have deferential expression as first, second, third, and fourth Runge-kutta methods as yield in Eq. (9)

$$k_1 = hf(x_n, y_n),$$

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

$$k_3 = hf\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right), \quad [9]$$

$$k_4 = hf(x_n + h, y_n + k_3),$$

$$k = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

3. Aitken's Method and Newton Interpolation Polynomial

In this section. We are going to consider the Aitken's method and the Newton interpolation method.

3.1 Aitken's Methods

Aitken's procedure is a systematically and successively numerical method that yields better interpolation polynomials. One of its major advantages is that the accuracy by comparing successive steps. The Aitken's formula can be express as:

$$P_{0,k}(x) = \frac{1}{x_k - x_0}\begin{vmatrix} y_0 & x_0 - x \\ y_k & x_k - x \end{vmatrix}, \qquad [10]$$

Where, for each k = 0, 1, ..., n.

And

$$P_{0,1,2} = = \frac{1}{x_2 - x_1}\begin{vmatrix} P_{0,1}(x) & x_1 - x \\ P_{0,2}(x) & x_2 - x \end{vmatrix}, \qquad [11]$$

Where, for each k = 0, 1, ..., n,

$$y_n = P_{0,1,2,\dots,n}(x) = P_{0,1,2,\dots,n}(x) = \frac{1}{x_n - x_{n-1}}\begin{vmatrix} P_{0,1,\dots,(n-1)}(x) & x_{n-1} - x \\ P_{0,1,\dots,(n-2),n}(x) & x_n - x \end{vmatrix}. \qquad [12]$$

The Aitken iteration process is described below: At first step, a set of linear polynomials between the points (x_0, y_0) and (x_j, y_j), $j = 1,2,\dots,n$ is determined. Regarding the first approximation. Let the linear polynomial for the points (x_0, y_0) and (x_1, y_1) be denoted by $p_{0,1}(x)$ and is given by"

$$P_{0,1}(x) = \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1. \qquad [13]$$

$$P_{0,1}(x) = \frac{1}{x_1 - x_0}[(x_1 - x)y_0 - (x_0 - x)y_1]. \qquad [14]$$

$$P_{0,1}(x) = \frac{1}{x_1 - x_0}\begin{vmatrix} y_0 & x_0 - x \\ y_1 & x_1 - x \end{vmatrix}. \qquad [15]$$

In second step, a set of quadratic polynomials is generated for the points (x_0, y_0), (x_1, y_1) and (x_j, y_j), $j = 2,3,\dots,n$.

$$P_{0,j}(x) = \frac{1}{x_j - x_0}\begin{vmatrix} y_0 & x_0 - x \\ y_j & x_j - x \end{vmatrix}, j = 2,\dots,n. \qquad [16]$$

The second approximation is performed with the help of first approximation to reduce the computational effort, this can be seen in Eq (11). This process is repeated for n times.

The example below shows the Aitken application:

x	1	2	3
$y(x)$	7	5	2

By considering Eq. (10) for $\mathrm{k} = 1, 2$ now we obtain

$$P_{0,1} = \begin{vmatrix} 7 & 1-x \\ 5 & 2-x \end{vmatrix} = 9 - 2x \quad [17]$$

$$P_{0,2} = \frac{1}{2}\begin{vmatrix} 7 & 1-x \\ 2 & 3-x \end{vmatrix} = \frac{1}{2}(19 - 2x) \quad [18]$$

The second approximation is given by considering Eq. (11) for k = 2

$$P_{0,1,2} = \begin{vmatrix} 9-2x & 2-x \\ 0.5(19-5x) & 3-x \end{vmatrix} = 8 - 0.5x - 0.5x^2 \quad [19]$$

3.2 Newton's Interpolation Method

Newton interpolation is a quadratic interpolation methodology used in numerical methods and outcomes. The interpolation formula in most classic procedures is particular to the data. This paper discusses Newton type polynomial interpolation approaches.

The forward difference formula and the backward difference formula are used in Newton polynomial interpolation.

$$y_0(x) = a_0, \quad [20]$$

$$y_1(x) = a_0 + a_1(x - x_0), \quad [21]$$

$$y_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), \quad [22]$$

$$y_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0) + \cdots \ldots . + a_n(x - x_0)(x - x_1) \ldots (x - x_{n-1}), \quad [23]$$

Where

$$a_0 = y_0, \quad [24]$$

$$a_1 = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)}, \quad [25]$$

$$a_2 = \frac{\frac{f(x_2) - f(x_1)}{(x_2 - x_1)} - \frac{f(x_1) - f(x_0)}{(x_1 - x_0)}}{(x_2 - x_0)}, \quad [26]$$

$$a_3 = \frac{\frac{\frac{f(x_3) - f(x_2)}{x_3 - x_2} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}}{x_3 - x_1} - \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}}{x_3 - x_0}, \quad [27]$$

$$a_n = f[x_k, x_{k-1}, \ldots, x_1, x_0] = \frac{f[x_k, x_{k-1}, \ldots, x_2, x_1] - f[x_{k-1}, x_{k-2}, \ldots, x_1, x_0]}{x_k - x_0}. \quad [28]$$

3.3 Forward Difference

The value inside the boxes of the following difference in Table 1 is used in deriving the newton forward difference interpolation formula by setting $x = x_0 + ph$ and $a_0, a_{1, \ldots a_n}$, the newton forward difference equation is given as

$$p_n(x) = y_0 + p\Delta y_0 + \frac{p(p-1)}{2i}\Delta^2 y_0 + \frac{p(p-1)(p-2)}{3i}\Delta^3 y_0 + \cdots + \frac{p(p-1)(p-2)\cdots(p-n+1)}{ni}\Delta^n y_0. \quad [29]$$

Table 1: Forward difference interpolation formula

Value of x	Value of $y = f(x)$	First Difference $\Delta f(x)$	Second Difference $\Delta^2 f(x)$	Third Difference $\Delta^3 f(x)$	Fourth Difference $\Delta^4 f(x)$
x_0	y_0				
		Δy_0			
$x_0 + h$	y_1		$\Delta^2 y_0$		
		Δy_1		$\Delta^3 y_0$	
$x_0 + 2h$	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$
		Δy_2		$\Delta^3 y_1$	
$x_0 + 3h$	y_3		$\Delta^2 y_2$		
		Δy_3			
$x_0 + 4h$	y_4				

3.4 Backward Difference

The value inside the boxes of the following difference Table 2 is used in deriving the newton backward difference interpolation formula which is given as

$$p_n(x) = y_n + p\nabla y_0 + \frac{p(p+1)}{2i}\nabla^2 y_n + \frac{p(p+1)(p-2)}{3i}\nabla^3 y_0 + \cdots + \frac{p(p+1)(p-2)\cdots(p+n-1)}{ni}\nabla^n y_n. \quad [30]$$

Table 2: Backward difference interpolation formula

Value of x	Value of $y = f(x)$	First Difference $\nabla f(x)$	Second Difference $\nabla^2 f(x)$	Third Difference $\nabla^3 f(x)$	Fourth Difference $\nabla^4 f(x)$
x_0	y_0				
		∇y_1			
$x_0 + h$	y_1		$\nabla^2 y_2$		
		∇y_2		$\nabla^3 y_3$	
$x_0 + 2h$	y_2		$\nabla^2 y_3$		$\nabla^4 y_4$
		∇y_3		$\Delta^3 y_4$	
$x_0 + 3h$	y_3		∇y_4		
		∇y_4			
$x_0 + 4h$	y_4				

4. Application

In this section, we make use of the formula and conditions obtained from the previous section and illustrate the numerical solution of first-order differential equations.

Example 1. Let us first consider the following first-order differential equations

$$f(x) = \frac{\sqrt{3}}{2}x + \frac{11}{4}y \qquad y(0) = 0 \qquad h = 0.01. \quad [31]$$

By applying the Newton interpolation of Eqs. (20), (21), (22), (24), (25), and (26), we obtain the following

$$a_o = 0 = y_o, \quad [32]$$

$$a_1 = \frac{f(x_1) - f(x_o)}{x_1 - x_o} = \left[\frac{dy}{dx}\right]_{0,0} = \frac{\sqrt{3}}{2}(0) + \frac{11}{4}(0) = 0 \quad [33]$$

$$y_1 = a_o + a_1(x - x_o) = 0 + 1(0.01 - 0) = 0 \quad [34]$$

$$a_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_o)}{x_1 - x_o}}{x_2 - x_o} = \frac{\left[\frac{dy}{dx}\right]_{0.01,\ 0} - \left[\frac{dy}{dx}\right]_{0,0}}{0.02 - 0}$$

$$a_2 = \frac{\frac{\sqrt{3}}{2}(0.01) + \frac{11}{4}(0) - [\frac{\sqrt{3}}{2}(0) + \frac{11}{4}(0)]}{0.02 - 0} = \frac{0.008660254 - 1}{0.02} = 0.4330127 \quad [35]$$

$$y_2 = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$
$$= 0 + 0(0.02 - 0) + 0.4330127(0.02 - 0)(0.02 - 0.01)$$
$$= 0.00008660254 \quad [36]$$

Where

$$x_o = 0, \ \ x_1 = 0.01, \ \ x_2 = 0.02, \quad [37]$$

Considering the values x_o, x_1, x_2 of Eq. (37) and y_o, y_1, y_2 of Eq. (32), Eq. (34), and Eq. (36) respectively, and inserting them in Eq. (10) for $k = 0{,}1$ and Eq. (11), we obtain

$$P_{0,1}(x) = \frac{1}{0.01-0}\begin{vmatrix} 0 & 0-x \\ 0 & 0.01-x \end{vmatrix} = 0. \quad [38]$$

$$P_{0,2}(x) = \frac{1}{0.02-0}\begin{vmatrix} 0 & 0-x \\ 0.00008660254 & 0.02-x \end{vmatrix} = 0.004330127x. \quad [39]$$

$$P_{0,1,2}(x) = \frac{1}{0.02-0.01}\begin{vmatrix} x & 0.01-x \\ 0.004330127x & 0.02-x \end{vmatrix} = 0.4330127x^2 - 0.004330127x \quad [40]$$

The exact solution is given by

$$y_{exact} = \frac{2}{121}\sqrt{3}\left(-4 + 4e^{11x/4} - 11x\right). \quad [41]$$

The figures below depict the graph of approximate solution with exact solutions and error between.

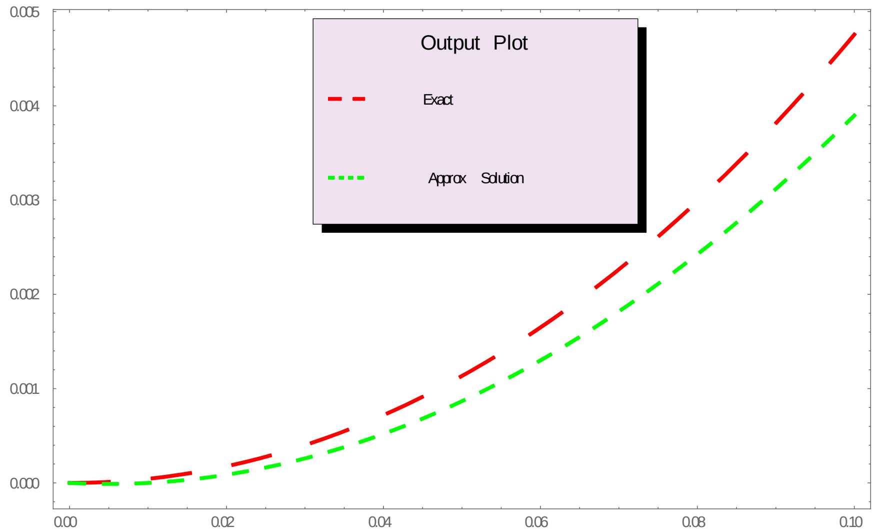

Figure 4.1: Solution of Example 1 with Aitkens and Newton Method.

The error is defined as

$$error = y_{exact} - y_2 \quad [42]$$

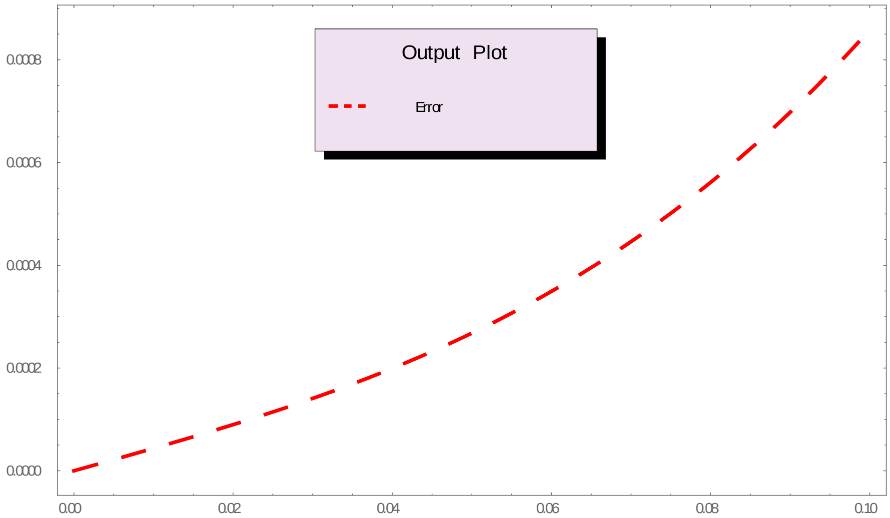

Figure 4.2: Error Example 1 with Aitkens and Newton Method

Table 4.2: The Table showing the result of Example 1

x	Exact Values	Newtons & Aitken Methods.	Errors
0	0	0	0
0.01	0.0000437	0	0.000043
0.02	0.000176	0.000087	0.0000898
0.03	0.000401	0.0002598	0.0001408
0.04	0.000719	0.0399969	0.0001993
0.05	0.00113	0.000866	0.0002679
0.06	0.001648	0.001299	0.0003492
0.07	0.002265	0.0018187	0.0004461
0.08	0.002986	0.0024249	0.0005613
0.09	0.003815	0.0031177	0.0006979
0.1	0.004755	0.00389711	0.0008588

5. Conclusion

This paper studies the solution of first-order differential equations using the Aitken method and Newton interpolation approached. The result obtained shows that the proposed method and approached gives a minimal approximation errors and outperforms the existing methods. The numerical results are verified by an example that is computed using Mathematica and MATLAB.

References

Al Din, I. N. (2020). Solving Bernoulli Differential Equations by using Newton's Interpolation and Aitken's Methods.

Al Din, I. N. (2020). Using Newton's Interpolation and Aitken's Method for Solving First Order Differential Equation. *World Applied Sciences Journal*, 38(3), 191-194.

Atkinson, K., Han, W., & Stewart, D. E. (2009). Numerical solution of ordinary differential equations. *John Wiley & Sons*, 10.

Chasnov, J. R. (2014). Differential Equations with YouTube Examples-eBooks and textbooks from bookboon. com.

Faith, C. (2018). Solution of first order differential equation using numerical newton's interpolation and lagrange method. *Int. J. Dev. Res, 8*, 18973-18976.

Kreyszig, E. et al (2011). Advanced Engineering Mathematics 10th Edition. John Wiley & Sons.

Ibrahim, S. Numerical Approximation Method for Solving *Differential Equations. Eurasian Journal of Science & Engineering*, 6(2), 157-168, 2020.

Ibrahim, S., & Isah, A. (2021). Solving System of Fractional Order Differential Equations Using Legendre Operational Matrix of Derivatives. *Eurasian Journal of Science & Engineering,* 7(1), 25-37.

Ibrahim, S., & Isah, A. (2022) Solving Solution for Second-Order Differential Equation Using Least Square Method. *Eurasian Journal of Science & Engineering*, 8(1), 119-125.

Ibrahim, S., & Koksal, M. E. (2021a). Commutativity of Sixth-Order Time-Varying Linear Systems. *Circuits Syst Signal Process*. https://doi.org/10.1007/s00034-021-01709-6

Ibrahim, S., & Koksal, M. E. (2021b). Realization of a Fourth-Order Linear Time-Varying Differential System with Nonzero Initial Conditions by Cascaded Two Second-Order Commutative Pairs. *Circuits Syst Signal Process*. https://doi.org/10.1007/s00034-020-01617-1.

Isah, A., & Ibrahim, S. (2021). Shifted Genocchi Polynomial Operational Matrix for Solving Fractional Order System. *Eurasian Journal of Science & Engineering,* 7(1) 25-37.

Mbagwu, J. P., & Ide, N. A. D. (2021). Comparison of Newton's Interpolation and Aitken's Methods with Some Numerical Methods for Solving System of First and Second Order Differential Equation. *World Scientific News*, 164, 108-121.

Rababah, A., & Ibrahim, S. (2016a). Weighted G^1-Multi-Degree Reduction of Bézier Curves. *International Journal of Advanced Computer Science and Applications*, 7(2), 540-545. https://thesai.org/Publications/ViewPaper?Volume=7&Issue=2&Code=ijacsa&SerialNo=70

Rababah, A., & Ibrahim, S. (2016b). Weighted Degree Reduction of Bézier Curves with G^2-continuity. *International Journal of Advanced and Applied Science*, 3(3), 13-18.

Rababah, A., & Ibrahim, S. (2018). Geometric Degree Reduction of Bézier curves, *Springer Proceeding in Mathematics and Statistics*, Book Chapter 8. 2018. https://www.springer.com/us/book/9789811320941

Salisu, I. (2022a,). Commutativity of high-order linear time-varying systems. *Advances in Differential Đ Equations and Control Processes*, 27(1) 73-83. Đ http://dx.doi.org/10.17654/0974324322013

Salisu, I. (2022b). Discrete least square method for solving differential equations, Advances and Applications in Discrete Mathematics 30, 87-102. Đ http://dx.doi.org/10.17654/0974165822021

Salisu, I. (2022c). Commutativity Associated with Euler Second-Order Differential Equation. *Advances in Differential Equations and Control Processes,* 28(2022), 29-36. Đ http://dx.doi.org/10.17654/0974324322022

Salisu, I., & Abedallah R. (2022). Decomposition of Fourth-Order Euler-Type Linear Time-Varying Differential System into Cascaded Two Second-Order Euler Commutative Pairs, *Complexity*, 2022, https://doi.org/10.1155/2022/3690019.

Zill, D. G. (2012). *A first course in differential equations with modeling applications*: Cengage Learning.

Zou, L., Song, L., Wang, X., Weise, T., Chen, Y., & Zhang, C. (2020). A new approach to Newton-type polynomial interpolation with parameters. *Mathematical Problems in Engineering, 2020*.