INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS

ISSN(print): 2643-9840, ISSN(online): 2643-9875

Volume 07 Issue 02 February 2024

DOI: 10.47191/ijmra/v7-i02-13, Impact Factor: 8.22

Page No. 522-528

Solving Ordinary Differential Equations with Boundary Conditions Numerically

M. A. Sandoval-Hernandez[1,2], G. J. Morales-Alarcon[3], U. A. Filobello-Nino[4], H. Vazquez-Leal[4], A. Alday-Garcia[5], C.E. Sampieri-Gonzalez[3], J. E. Pretelin-Canela[4], A. R. Escobar-Flores[4], A. E. Gasca-Herrera[4], O. Alvarez-Gasca[4], A. D.Contreras-Hernandez[4], J. E. Perez-Jacome-Friscione[4], S. E. Torreblanca-Bouchan[1]

[1] Centro de Bachillerato Tecnológico industrial y de servicios No. 190, Av. 15 Col. Venustiano Carranza 2da Sección, Boca del Río, 94297, Veracruz, México.

[2]Escuela de Ingeniería, Universidad de Xalapa, Carretera Xalapa-Veracruz Km 2 No. 341,91190 Xalapa, Veracruz, México

[3]Instituto de Psicología y Educación, Universidad Veracruzana, Agustín Melgar 2, col. 21 de Marzo, Xalapa, 91010 Veracruz, México.

[4]Facultad de Instrumentación Electrónica, Universidad Veracruzana, Circuito Gonzalo Aguirre Beltrán S/N, Xalapa, 91090, Veracruz, México.

[5]Facultad de Medicina, Calles Médicos y Odontólogos, Col. Unidad del Bosque, Xalapa 91010, Veracruz.

ABSTRACT: This article elucidates the method and implementation of finite differences, aiming to enhance clarity for students and provide detailed insights into its application, as many authors assume a pre-existing understanding of numerical analysis. It includes two case studies employing Dirichlet and Cauchy boundary conditions, demonstrating the precision and effectiveness of this numerical tool in obtaining results.

KEYWORDS: Numerical analysis, finite differences, ordinary differential equations, boundary conditions, Newton-Raphson.

I. INTRODUCTION

Numerical analysis, pivotal in mathematics and science, traces its origins to ancient times, gaining prominence with Neptune's discovery in 1846 [1], affirming Newton's law of gravitation. Originating with the Babylonians and evolving, it saw significant advancements with the 16th-century advent of algebra and logarithm tables. The 18th century saw Stirling and Taylor laying finite difference calculus foundations, a key aspect of modern numerical analysis. The 19th century marked a leap in numerical analysis with automatic calculators [2], further accelerated post-World War II by high-speed electronic computers, enabling scientific breakthroughs and emphasizing precision in numerical calculations.

Modern numerical analysis, initiated by John von Neumann and Herman Goldstine in 1947 [3], addressed rounding errors and introduced scientific computing fundamentals. It is now defined by the synergy of programmable computers, mathematical analysis, and complex problem-solving. Numerical methods, essential in engineering for solving nonlinear algebraic equations and differential equations, rely on basic arithmetic operations [4-9].

In science and technology, mathematical models describe real phenomena, with applied mathematics seeking appropriate tools for problem-solving. Unfortunately, classical analytical methods are not always applicable due to various limitations, leading to the advancement of numerical methods, significantly influenced by computing advancements. Numerical analysis consists of iterative algorithms providing solutions based on stopping criteria and error estimates. These algorithms serve multiple purposes, including calculating numerical derivatives, integrals, differential equations, linear algebra, interpolations, curve fitting, and polynomials [4-9]. Numerical methods enable understanding and applying numerical schemes for solving mathematical, engineering, and scientific problems on computers. They involve simplifying basic numerical schemes, programming, and solving problems on computers.

Numerical analysis aims to design methods for efficiently approximating mathematical problem solutions. Its primary goal is to find approximate solutions to complex problems using simple arithmetic operations, involving a sequence of algebraic and logical operations. Numerical analysis has become crucial in engineering process simulator software development. For instance, Pipe Flow, a piping system design and modeling software, calculates fluid flow in various network configurations. Pipe Flow Expert determines flow rates and pressure drops [10-12]. Matlab, including SIMULINK, models, simulates, and analyzes dynamic systems, widely used in signal processing and control engineering [13-16]. COMSOL Multiphysics® models physical phenomena, virtually any phenomenon describable by PDEs, including heat transfer, fluids, electromagnetism, and structural mechanics [17-19]. It offers multiphysics capabilities for modeling combinations of phenomena based on PDEs. COMSOL simplifies application development with its Model Library, making it accessible for users without extensive mathematical or numerical analysis knowledge. Its features make it applicable in various fields, including acoustics, electromagnetism, MEMS, microwave engineering, chemistry, fluid dynamics, structural mechanics, physics, geophysics, optics, photonics, quantum mechanics, control systems, and applied mathematics.

Cadence PSpice, a Spice-based simulator, specializes in simulating designs with analog and digital components. It offers a range of simulation models and integrates with Allegro® Design Entry HDL and OrCAD® Capture for schematic capturing [20, 21]. Opera Simulation Software, a comprehensive CAE toolset for electromagnetic system design, analysis, simulation, and optimization, includes multiphysics effects [22, 23]. Created in 1984, Opera facilitates the design and optimization of 2D and 3D electromechanical devices, solving various electromagnetic problems.

This article aims to guide students in solving ordinary differential equations using finite difference numerical solutions, considering Maple 17's exact numerical solution. It presents four case studies applying the methodology under different boundary conditions.

This paper is organized as follows: Section II introduces the finite difference method in the context of ordinary differential equations. Section III presents two case studies for analysis. Section IV is dedicated to the discussion of the findings, and Section V concludes the paper.

II. SOME BASICS OF FINITE DIFFERENCES METHOD

The finite difference method (FDM) is applied to find the solution of ordinary differential equations (ODEs) with boundary conditions, that is, when the values of the function at the ends of the interval [a, b] are known. This method involves replacing the numerical derivation formulas derived from the Newton-Gregory interpolating polynomial in the differential equation. Like the interpolating polynomial, the finite difference method adheres to specific rules, including constant spacing and pivoting. It consists of substituting the numerical derivation equations into the differential equation and then constructing a recurrence equation. This pivoting process, applied at n equidistant points within the interval [a, b], results in a set of equations that will be solved by a specific numerical method. The application of this method allows for effectively addressing and solving ODEs with defined boundary conditions.

The FDM is a numerical approach to solving differential equations, providing numerical rather than symbolic solutions. In this method, functions are represented as arrays of discrete points, with the accuracy of the solution improving as more points are used, albeit at the cost of increased computational intensity. This represents a fundamental trade-off in the method.
The procedure for FDM is as follows:

1 Identify the governing equation and boundary values.

For example, let the differential equation

$$\frac{d^2y(x)}{dx^2} - 2\frac{dy(x)}{dx} + 1 = 0, \tag{1}$$

with boundary values given by

$$y(0) = 0, \qquad y(1) = 0. \tag{2}$$

2 Approximate derivatives using finite differences, ensuring that each term in the finite-difference equation is evaluated at the same point.

The central finite difference equations for the second derivative, first derivative, and $y(x)$ are given by

$$\frac{d^2y(x)}{dx^2} = \frac{y(x+\Delta x) - 2y(x) + y(x-\Delta x)}{\Delta x}, \tag{3}$$

$$\frac{dy(x)}{dx} = \frac{y(x+\Delta x) - y(x-\Delta x)}{2\Delta x}, \tag{4}$$

$$y(x) = y(x). \tag{5}$$

Substituting (3) and (4) in (1) we obtain

$$\frac{y(x+\Delta x)-2y(x)+y(x-\Delta x)}{\Delta x}-2\frac{y(x+\Delta x)-y(x-\Delta x)}{2\Delta x}+1=0. \tag{6}$$

3 Express the finite-difference equation using array indices.

Rewriting (6) using indices we have

$$\frac{y_{i+1}-2y_i+y_{i-1}}{\Delta x}-2\frac{y_{i+1}-y_{i-1}}{2\Delta x}+1=0. \tag{7}$$

4 Rearrange the finite-difference equation for clarity.

Simplifying the process, the finite-difference equation (7) is rearranged so as to collect the y(x) terms.

$$(1+\Delta x)y_{i-1}-2y_i+(1-\Delta x)y_{i+1}+\Delta x^2=0. \tag{8}$$

5 Set up the computational grid.

For instance, let us solve this equation using 40 points on the grid.

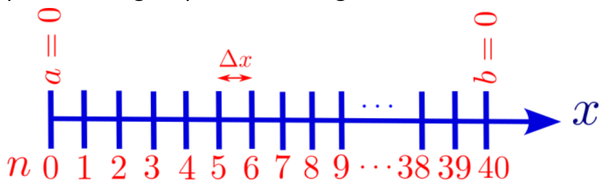

Figure 1. The grid spacing.

In this way, the grid spacing for $n=40$. Note that we have 41 points including zero, this way

$$\Delta x=\frac{x_b-x_0}{n}=\frac{1-0}{41-1}=0.025. \tag{9}$$

6 Revise the Finite-Difference Equation as needed.

In this way, substituting the value of Δx in (8) we obtain

$$1.025y_{i-1}-2y_i+0.975y_{i+1}+0.000625=0 \tag{10}$$

7 Apply the finite-difference equation at each grid point.

Write finite-difference equation at each point on grid. In this way we will count i from *1* to *39* because the zero point, $i-1=0, y_{1-1}=0$ corresponds to the boundary $x=a=0$ and the point 40, $i+1, y_{39+1}=0$, corresponds to the boundary $x=b=1$, i.e.

$$\begin{aligned}
1.025y_0-2y_1+0.975y_2+0.000625&=0,\\
1.025y_1-2y_2+0.975y_3+0.000625&=0,\\
1.025y_2-2y_3+0.975y_4+0.000625&=0,\\
&\vdots\\
1.025y_{36}-2y_{37}+0.975y_{38}+0.000625&=0,\\
1.025y_{37}-2y_{38}+0.975y_{39}+0.000625&=0,\\
1.025y_{38}-2y_{39}+0.975y_{40}+0.000625&=0,
\end{aligned} \tag{11}$$

This way, in this example we have $\mathbf{y_0=0, y_{40}=0}$ are the boundaries.

8 Formulate the set of equations as a single matrix equation.

Writing the system of equations in matrix form we obtain

$$\begin{bmatrix} \boxed{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.025 & -2 & 0.975 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.025 & -2 & 0.975 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.025 & -2 & 0.975 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.025 & -2 & 0.975 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \vdots & \vdots & \vdots & 0 & 0 \\ h_1 & 0 & 0 & 0 & 0 & 0 & 1.025 & -2 & 0.975 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.025 & -2 & 0.975 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ \vdots \\ y_{38} \\ y_{39} \\ y_{40} \end{bmatrix} = \begin{bmatrix} \boxed{0} \\ -0.000625 \\ -0.000625 \\ -0.000625 \\ -0.000625 \\ \vdots \\ \vdots \\ -0.000625 \\ -0.000625 \\ \boxed{0} \end{bmatrix} \tag{12}$$

See the red boxes that indicate the boundaries. Using the matrix form we can write as

$$\mathbb{A}\mathbb{y} = \mathbb{b} \tag{13}$$

9 Solve the matrix equation to find the solution.

Equation (13) can be solved by calculating the inverse of $\mathbb{A}$, using Newton-Raphson, Gauss-Jordan, among others. In this way solving for and we obtain

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ \vdots \\ y_{38} \\ y_{39} \\ y_{40} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.0084887 \\ 0.0167716 \\ 0.0248383 \\ 0.0326776 \\ \vdots \\ \vdots \\ 0.03003592 \\ 0.01570591 \\ 0 \end{bmatrix} \tag{14}$$

10 Plot the results, displaying x versus y(x), where y(x) is the solution vector obtained from the matrix equation.

Figure 2 shows the graph of the solution vector $\mathbb{y}$ against values of x.

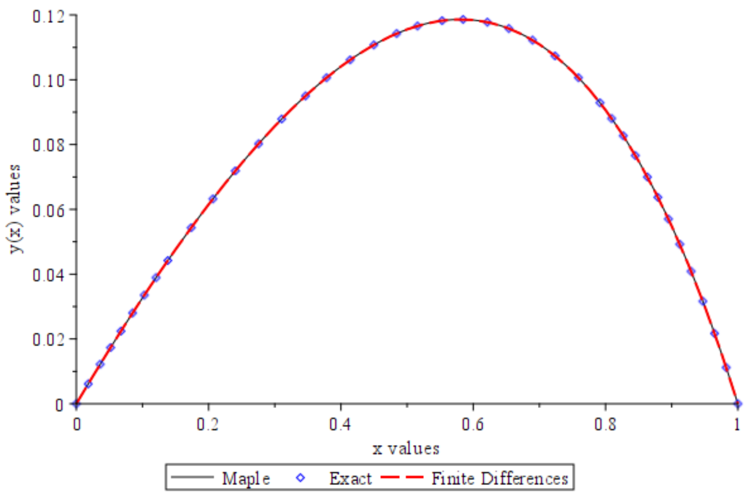

Figure 2. Solution vector $\mathbb{y}$ against values of x

III. STUDY CASES

A. *Case 1. Differential equation with Dirichlet condition*

In this case study, the second-order linear equation with Dirichlet conditions given by (1) will be solved.

Solution. The equation has an analytical solution and is given by

$$y(x) = -\frac{e^{2x}}{2(e^2-1)} + \frac{x}{2} + \frac{1}{2(e^2-1)}. \tag{15}$$

The solution process used finite FDM is the one presented in section II for $n = 40$ and $\Delta x = 0.025$. Figure 2 shows the solution using FDM in dash line, against the exact solution with diamonds and the numeric provided by Maple 17 with solid line. Analysis of significant digits is outside the scope of this article. For further details, see [24, 25].

B. Case 2. Differential equation with Cauchy condition

A Cauchy boundary condition defines the function value and its normal derivative at the domain's boundary. This is equivalent to applying both Dirichlet and Neumann boundary conditions simultaneously. The term is attributed to the renowned 19th-century French mathematician and analyst, Augustin Louis Cauchy. Now we will solve equation (1) using Cauchy conditions given by

$$y(0) = 0, \qquad y'(1) = 0. \tag{16}$$

Solution. Using the boundary conditions (16), the analytical solution of (1) is

$$y(x) = \frac{3e^{2x}}{4e^2} + \frac{x}{2} - \frac{3}{4e^2}. \tag{17}$$

Applying the FDM methodology of section II for $n = 100$ and $\Delta x = 0.01$, we obtain

$$1.01y_{i-1} - 2y_i + 0.99y_{i+1} + 0.0001 = 0. \tag{18}$$

We will now count *i* from 0 to 99 and we will have 99 equations with *100* unknowns. In $i = 0$ we we have the node of the first boundary condition. To achieve a total of 100 equations, we will incorporate the second boundary condition, utilizing its numerical derivative. As suggested in the literature on numerical algorithms, the implementation of a virtual node is a feasible approach for this substitution.

$$\begin{gathered} 1.01y_0 - 2y_1 + 0.99\,y_2 + 0.0001 = 0,\\ 1.01y_1 - 2y_2 + 0.99\,y_3 + 0.0001 = 0,\\ 1.01y_2 - 2y_3 + 0.99\,y_4 + 0.0001 = 0,\\ 1.01y_3 - 2y_4 + 0.99\,y_5 + 0.0001 = 0,\\ \vdots \qquad \vdots \qquad \vdots\\ 1.01y_{96} - 2y_{97} + 0.99\,y_{98} + 0.0001 = 0,\\ 1.01y_{97} - 2y_{98} + 0.99\,y_{99} + 0.0001 = 0,\\ 1.01y_{98} - 2y_{99} + 0.99\,y_{100} + 0.0001 = 0,\\ -y_{99} + y_{100} - 0.02 = 0. \end{gathered} \tag{19}$$

To resolve equation (19), we will first substitute the numerical value of the first boundary condition. Following this, we will employ numerical methods such as the Newton-Raphson and Gauss-Jordan algorithms, among others, to find the solution.

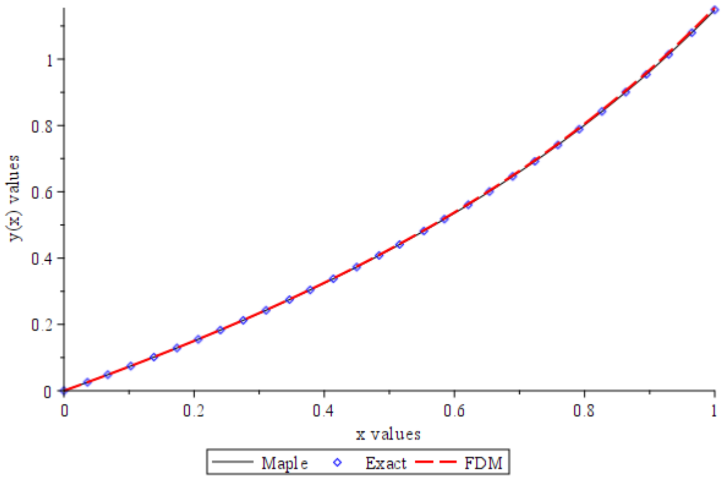

Figure 2. FDM solution vs analytical and numerical solution of Maple.

IV. DISCUSSION

The second-order linear differential equation (1), with a solution under analysis, was solved using the Finite Difference Method (FDM) and served as a model to illustrate the detailed steps involved in applying the FDM method. Numerically, equation (1) was solved with the Maple dsolve command for each of the two case studies, considering the boundary conditions (2) and (16). Figure 3 displays the absolute error between the numerical solutions derived from Maple 17 and the one obtained using FDM with $n = 40$.

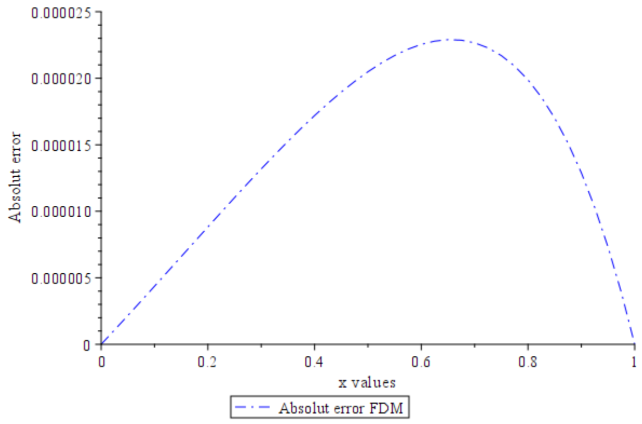

Figure 2. Absolut error for study case 3.FDM solution vs. analytical solution and Maple.

The maximum error encountered is $2.29001976x10^{-5}$ at $x = 0.65$ for i=*26.* It's important to highlight that this error is relatively small, considering that only 40 iterations were employed. To achieve higher accuracy, an increase in the number of nodes can be implemented, though it should be noted that this will demand more computational effort. Additionally, it is observed that the error is smaller at the boundaries, as expected, since these are the known points provided to the system of equations (19). The determination of the absolute error for case study 2 is left as an exercise for the reader.

V. CONCLUSIONS

This paper comprehensively presents the Finite Difference Method (FDM) for student comprehension, as numerical analysis textbooks often emphasize its application in partial differential equations while overlooking its use in ordinary differential equations. We advocate for an initial understanding and application of FDM in ordinary differential equations before implementing it in solving partial differential equations. In this study, two case analyses were conducted for a linear differential equation, considering Dirichlet and Cauchy conditions. The results demonstrated good accuracy when compared with solutions from numerical software like Maple.

DECLARATION OF INTERESTS STATEMENT

The authors declare that there are no conflicts of interest regarding the publication of this paper.

ACKNOWLEDGMENT

The authors would like to thank Roberto Ruiz Gomez for his contribution to this project.

REFERENCES

1) Lyttleton, R. A. 1960. The rediscovery of Neptune. *Vistas in Astronomy*, *3*, 25-46.
2) Swade, D. D. 2003. *Calculation and tabulation in the 19th century: Airy versus Babbage*. University of London, University College London (United Kingdom).
3) Von Neumann, J., & Goldstine, H. H. 1947. Numerical inverting of matrices of high order.
4) Burden, R., & Faires, J. 2011. Numerical Analysis 9th edn (Boston: Brooks/Cole).
5) Chapra, S. C., Canale, R. P., Ruiz, R. S. G., Mercado, V. H. I., Díaz, E. M., & Benites, G. E. (2011). *Métodos numéricos para ingenieros* Vol5. New York, NY, USA: McGraw-Hill.
6) Daniels, R. W. 1978. An introduction to numerical methods and optimization techniques. *Elsevier*.
7) Butenko, S., & Pardalos, P. M. 2014. *Numerical methods and optimization: An introduction*. CRC Press.
8) Corless, R. M., & Fillion, N. 2013. A graduate introduction to numerical methods. Springer.
9) Luthe, R., Olivera, A., & Schutz, F. 1978. Métodos Numéricos, Editorial Limusa.

10) Goswami, S., & Hemmati, A. 2020. Response of turbulent pipeflow to multiple square bar roughness elements at high Reynolds number. *Physics of Fluids*, *32*(7).
11) Messa, G. V., & Matoušek, V. 2020. Analysis and discussion of two fluid modelling of pipe flow of fully suspended slurry. *Powder Technology*, *360*, 747-768.
12) Avila, M., Barkley, D., & Hof, B. 2023. Transition to turbulence in pipe flow. *Annual Review of Fluid Mechanics*, *55*, 575-602.
13) Moore, H., Olguín, V. C., & Nuño, R. M. 2007. *MATLAB para ingenieros.* Pearson Educación.
14) Giron-Sierra, J. M. 2017. *Digital Signal Processing with Matlab Examples, Volume 1*. Singapore: Springer.
15) Wang, L. 2020. *PID control system design and automatic tuning using MATLAB/Simulink*. John Wiley & Sons.
16) Shanmugasundaram, N., Kumar, S. P., & Ganesh, E. N. 2022. Modelling and analysis of space vector pulse width modulated inverter drives system using MatLab/Simulink. *International Journal of Advanced Intelligence Paradigms*, *22*(1-2), 200-213.
17) Jeong, J., Kim, Y., Bae, S., & Youn, S. 2023. Simulation of classical axion electrodynamics using COMSOL multiphysics. *Journal of the Korean Physical Society*, 1-7.
18) Lliev, I. K., Gizzatullin, A. R., Filimonova, A. A., Chichirova, N. D., & Beloev, I. H. 2023. Numerical Simulation of Processes in an Electrochemical Cell Using COMSOL Multiphysics. *Energies*, *16*(21), 7265.
19) Hulkó, G., Belavý, C., Buček, P., & Zajíček, K. O. P. 2009. Control of Systems Modeled by COMSOL Multiphysics as Distributed Parameter Systems. In *COMSOL Conference 2009*.
20) López, D. B. 2009. *Análisis de circuitos con PSpice*. Alpha Editorial.
21) Sedra, A., Smith, K. C., Carusone, T. C., & Gaudet, V. 2020. Microelectronic circuits 8th edition Oxford.
22) Lalitha, M. P., Kumar, K. V. P., & Samala, V. 2014. Design and simulation of voltage and electric field distribution on disc insulators using finite element method in Opera software. In *2014 International Conference on Smart Electric Grid (ISEG)* (pp. 1-6). IEEE.
23) Jolissaint, L., Veran, J. P., & Marino, J. 2004. OPERA, an automatic PSF reconstruction software for Shack-Hartmann AO systems: application to Altair. In *Advancements in Adaptive Optics* (Vol. 5490, pp. 151-163). SPIE.
24) Sandoval-Hernández, M. A., Vazquez-Leal, H., Filobello-Nino, U. A., Morales-Alarcón, G. J., Velez-López, G. C., Castañeda-Sheissa, R., Campos-Dominguez S. Y., Ocaña-Pimentel, S., Domiguez-Chavez, J. A., Sampieri-Gonzalez, C.E., Pretelin-Canela, J. E., Bagatella-Flores, N., Escobar-Flores, A. R, Contreras-Hernandez, A. D., Alvarez-Gasca, O., Gasca-Herrera, A. E., Perez-Jacome-Friscione, J. E, Cuellar-Hernandez, L. & Pablo-López, B. 2023. Using Horner's Algorithm to Reduce Computing Times. International Journal of Engineering Research & Technology (IJERT), 12(6), 325-331, ISSN 2278-0181.
25) Sandoval-Hernandez, M., Vazquez-Leal, H., Filobello-Nino, U., De-Leo-Baquero, E., Bielma-Perez, A. C., Vichi-Mendoza, J. C., Alvarez-Gasca, O., Contreras-Hernandez, A. D., Bagatella-Flores, N., Palma-Grayeb, B. E., Sanchez-Orea, J., Cuellar-Hernandez, L. & Cuellar-Hernandez, L. 2021. The Quadratic Equation and its Numerical Roots. *International Journal of Engineering Research y Technology*, *10*(6), 301-305.