

New numerical algorithm for solving linear and nonlinear Volterra Integro-Differential equations of the Second Kind

Akanbi Kareem Bello¹, Taiye Oyedepo^{2*}, Tayo Musiliu Raji³

¹ Department of Mathematics, University of Ilorin, Nigeria .

² Department of Health Information Management, Federal University of Allied Health Sciences, Nigeria

³ Department of Mathematics, Federal University of Agriculture, Nigeria

* Corresponding author's e-mail: oyedepotaiye@yahoo.com

ABSTRACT

This study presents the development, analysis and numerical performance of a new numerical algorithm designed for solving Volterra integro-differential equations of the second kind. The method is derived using a power series approach with interpolation and collocation techniques, leading to a continuous and discrete block formulation suitable for solving linear and nonlinear Volterra integro-differential equations of the second kind. The fundamental properties of the algorithm such as order, error constant, consistency, zero-stability, convergence and absolute stability are rigorously examined and the method is shown to satisfy all necessary conditions for reliability and convergence. To assess its accuracy and efficiency, three numerical examples involving linear and nonlinear Volterra integro-differential equations with known exact solutions are solved. The results demonstrate that the developed algorithm consistently produces extremely small absolute errors, significantly outperforming several existing numerical methods. The algorithm proves to be highly accurate, stable and suitable for a wide range of volterra integro-differential problems.

Keywords: Volterra integro-differential equation; block algorithm; numerical method; convergence; stability analysis; error estimation; power series method

Introduction

Integral equations are essential tools in science, engineering and technology because they model systems involving cumulative effects, interactions and constraints, and they are widely applied in physics such as in electrodynamics, fluid mechanics and quantum mechanics where equations like Maxwell's can be expressed in integral form to analyze wave equation behavior and field distributions (Aduroja et al., 2025); they are also used in mechanics to determine stress and strain through boundary integral methods and in thermal sciences to study heat conduction in various materials (Otaide & Oluwayemi, 2024). In engineering, integral equations support applications in signal processing, control systems, antenna design and circuit analysis (Oyedepo et al., 2023), and they underpin boundary element methods for structural analysis, aerodynamics and acoustics. Furthermore, in computer science and technology, they are crucial for image reconstruction, inverse problems, artificial intelligence and numerical simulations (Oyedepo, Ajileye, et al., 2024), making them indispensable for advancing modern engineering and technological innovations (Oyedepo, Ayinde, et al., 2024).

This study considered the second kind Volterra integro-differential equation of the form

$$y(x) = f(x) + \tau \int_0^x K(x, t) y(t) dt \quad (1)$$

where $K(x, t)$ is the kernel of Volterra integral equation, $f(x)$ is the function and τ is the a parameter. The equation (1) is called linear because the unknown function $y(x)$ under the integral sign occur linearly, i.e. the power of $y(x)$ is one.

The Adomian Decomposition Method (ADM), developed by George Adomian between the 1970s and 1990s, is a semi-analytical technique designed to solve linear, nonlinear, and stochastic differential equations without requiring linearization or perturbation. By decomposing complex problems into rapidly convergent series and using Adomian polynomials to approximate nonlinear terms, ADM provides accurate closed-form approximations suitable for a wide range of mathematical models. Its effectiveness demonstrated in studies such as Maturi and Malaikah (2021), including applications to the Fisher–Kolmogorov equation extends further to stochastic systems through Ito integral formulations (Chang, 2019; Agarwala et al., 2022), making ADM widely applicable in physics, engineering, and applied mathematics.

Ongoing research has strengthened ADM's computational performance and convergence properties. Modified decomposition approaches proposed by Linjun (2023) have reduced computational burdens associated with Volterra-type integral equations, while Agbata et al. (2022) demonstrated ADM's capacity to solve complex differential equations such as the Emden–Fowler type. Despite advantages such as stability and analytical flexibility, ADM faces limitations including sensitivity to initial conditions and the computational complexity of higher-order polynomials. These challenges were highlighted in the work of Islam and Srivastava (2024a, 2024b). Nevertheless, practical applications such as Agarwala et al. (2022) modeling of COVID-19 transmission in India reinforce ADM's relevance as a robust and adaptable technique.

The Variational Iteration Method (VIM) offers another powerful analytical tool for solving ordinary, partial, fractional, and integro-differential equations through correction functionals constructed using Lagrange multipliers. Unlike ADM, VIM does not require perturbation, small parameters, or the computation of Adomian polynomials. Studies by Pan et al. (2016), Cimen and Yatar (2020) show that VIM often outperforms traditional techniques for nonlinear equations such as the Burger's, Schrödinger, and Korteweg–de Vries equations. Improvements like the VIM-Chebyshev wavelet hybrid Memon et al. (2023) and the Modified VIM combined with the Boundary Shape Function Method Liu et al. (2025) have enhanced its accuracy, convergence speed, and ability to handle complex boundary and convection-diffusion-reaction problems, confirming VIM's versatility in mathematical modeling and physical systems.

Derivation of the Method

A block algorithm shall be derived for the computation of Volterra Integro-Differential Equations of the Second Kind of the form (1). We shall employ the power series approximate solution of the form,

$$y(t) = \sum_{j=0}^{r+s-1} a_j t^j \quad (2)$$

for the derivative of the block algorithm given by

$$A^{(0)}Y_m^{(i)} = \sum_{j=0}^1 \frac{(jh)^{(i)}}{i!} e_i y_n^{(i)} + h^{(3-i)} [d_i f(y_n) + b_i F(Y_m)] \quad (3)$$

Where r and s in (2) are the number of interpolation and collocation points respectively. Equation (2) is differentiated three times and substituted into (1), that is,

$$f(t, y, y', y'') = \sum_{j=0}^{r+s-1} j(j-1)(j-2)a_j t^{j-3} \quad (4)$$

A grid of one step length is considered in this paper with a constant step size h given by $h = t_{n+i} - t_n, i = 0, 1$ and off-step points at $t_{n+\frac{1}{6}}, t_{n+\frac{1}{3}}, t_{n+\frac{1}{2}}$.

Evaluate the Interpolating of equation (2) at a point $t_{n+s}, s = \frac{1}{6}\left(\frac{1}{3}\right)\frac{1}{2}$ and collocating of equation (4)

at $t_{n+r}, r = 0\left(\frac{1}{6}\right)1$, give a system of nonlinear equation of the form

$$TA = U \quad (5)$$

where

$$A = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8 \quad a_9]$$

$$U = \begin{bmatrix} y_{n+\frac{1}{6}} & y_{n+\frac{1}{3}} & y_{n+\frac{1}{2}} & f_n & f_{n+\frac{1}{6}} & f_{n+\frac{1}{3}} & f_{n+\frac{1}{2}} & f_{n+\frac{2}{3}} & f_{n+\frac{5}{6}} & f_{n+1} \end{bmatrix}$$

$$T = \begin{bmatrix} 1 & t_{n+\frac{1}{6}} & t_{n+\frac{1}{6}}^2 & t_{n+\frac{1}{6}}^3 & t_{n+\frac{1}{6}}^4 & t_{n+\frac{1}{6}}^5 & t_{n+\frac{1}{6}}^6 & t_{n+\frac{1}{6}}^7 & t_{n+\frac{1}{6}}^8 & t_{n+\frac{1}{6}}^9 \\ 1 & t_{n+\frac{1}{3}} & t_{n+\frac{1}{3}}^2 & t_{n+\frac{1}{3}}^3 & t_{n+\frac{1}{3}}^4 & t_{n+\frac{1}{3}}^5 & t_{n+\frac{1}{3}}^6 & t_{n+\frac{1}{3}}^7 & t_{n+\frac{1}{3}}^8 & t_{n+\frac{1}{3}}^9 \\ 1 & t_{n+\frac{1}{2}} & t_{n+\frac{1}{2}}^2 & t_{n+\frac{1}{2}}^3 & t_{n+\frac{1}{2}}^4 & t_{n+\frac{1}{2}}^5 & t_{n+\frac{1}{2}}^6 & t_{n+\frac{1}{2}}^7 & t_{n+\frac{1}{2}}^8 & t_{n+\frac{1}{2}}^9 \\ 0 & 0 & 0 & 6 & 24t_n & 60t_n^2 & 120t_n^3 & 210t_n^4 & 336t_n^5 & 504t_n^6 \\ 0 & 0 & 0 & 6 & 24t_{n+\frac{1}{6}} & 60t_{n+\frac{1}{6}}^2 & 120t_{n+\frac{1}{6}}^3 & 210t_{n+\frac{1}{6}}^4 & 336t_{n+\frac{1}{6}}^5 & 504t_{n+\frac{1}{6}}^6 \\ 0 & 0 & 0 & 6 & 24t_{n+\frac{1}{3}} & 60t_{n+\frac{1}{3}}^2 & 120t_{n+\frac{1}{3}}^3 & 210t_{n+\frac{1}{3}}^4 & 336t_{n+\frac{1}{3}}^5 & 504t_{n+\frac{1}{3}}^6 \\ 0 & 0 & 0 & 6 & 24t_{n+\frac{1}{2}} & 60t_{n+\frac{1}{2}}^2 & 120t_{n+\frac{1}{2}}^3 & 210t_{n+\frac{1}{2}}^4 & 336t_{n+\frac{1}{2}}^5 & 504t_{n+\frac{1}{2}}^6 \\ 0 & 0 & 0 & 6 & 24t_{n+\frac{2}{3}} & 60t_{n+\frac{2}{3}}^2 & 120t_{n+\frac{2}{3}}^3 & 210t_{n+\frac{2}{3}}^4 & 336t_{n+\frac{2}{3}}^5 & 504t_{n+\frac{2}{3}}^6 \\ 0 & 0 & 0 & 6 & 24t_{n+\frac{5}{6}} & 60t_{n+\frac{5}{6}}^2 & 120t_{n+\frac{5}{6}}^3 & 210t_{n+\frac{5}{6}}^4 & 336t_{n+\frac{5}{6}}^5 & 504t_{n+\frac{5}{6}}^6 \\ 0 & 0 & 0 & 6 & 24t_{n+1} & 60t_{n+1}^2 & 120t_{n+1}^3 & 210t_{n+1}^4 & 336t_{n+1}^5 & 504t_{n+1}^6 \end{bmatrix}$$

Solving equation (5) for a_j , where $j = 0, 1, 2, \dots, 9$ are constants to be determined and putting back into (2) gives a continuous form of block algorithm as

$$y(t) = a_1(t)y_{n+\frac{1}{6}} + a_1(t)y_{n+\frac{1}{3}} + a_1(t)y_{n+\frac{1}{2}} + h^3 \left[\sum_{j=0}^1 \beta_j(t)f_{n+j} + \beta_s(t)f_{n+s} \right], s = \frac{1}{6}\left(\frac{1}{6}\right)\frac{5}{6} \quad (6)$$

where $a_r(t)$, $\beta_j(t)$ and $\beta_s(t)$ are express as a function of x with

$$x = \frac{t - t_n}{h} \quad (7)$$

to obtain the continuous form as

$$\left. \begin{aligned} a_{\frac{1}{6}}(t) &= 3 - 15x + 18x^2, a_{\frac{1}{3}}(t) = -324x - 36x^2, a_{\frac{1}{2}}(t) = 1 - 9x + 18x^2 \\ \beta_0(t) &= -\frac{221}{6531840} + \frac{116857}{65318400}x - \frac{36733}{145120}x^2 + \frac{1}{6}x^3 - \frac{49}{80}x^4 + \frac{203}{150}x^5 - \frac{147}{80}x^6 + \frac{3}{2}x^7 - \frac{27}{40}x^8 + \frac{9}{70}x^9 \\ \beta_{\frac{1}{6}}(t) &= -\frac{4703}{2177280} + \frac{319073}{10886400}x - \frac{367}{3024}x^2 + \frac{3}{2}x^4 - \frac{261}{50}x^5 + \frac{87}{10}x^6 - \frac{279}{35}x^7 + \frac{27}{7}x^8 - \frac{27}{35}x^9 \\ \beta_{\frac{1}{3}}(t) &= -\frac{2831}{1088640} + \frac{2363}{124416}x + \frac{1153}{483840}x^2 - \frac{15}{8}x^4 + \frac{351}{40}x^5 - \frac{1383}{80}x^6 + \frac{1233}{70}x^7 - \frac{513}{56}x^8 + \frac{27}{14}x^9 \\ \beta_{\frac{1}{2}}(t) &= \frac{887}{3265920} + \frac{4589}{3265920}x - \frac{193}{5184}x^2 + \frac{5}{3}x^4 - \frac{127}{15}x^5 + \frac{93}{5}x^6 - \frac{726}{35}x^7 + \frac{81}{7}x^8 - \frac{18}{7}x^9 \\ \beta_{\frac{2}{3}}(t) &= -\frac{307}{2177280} + \frac{3623}{4354560}x + \frac{1153}{483840}x^2 - \frac{15}{16}x^4 + \frac{99}{20}x^5 - \frac{921}{80}x^6 + \frac{963}{70}x^7 - \frac{459}{56}x^8 + \frac{27}{14}x^9 \\ \beta_{\frac{5}{6}}(t) &= \frac{17}{435456} + \frac{479}{1555200}x - \frac{79}{12096}x^2 + \frac{3}{10}x^4 - \frac{81}{10}x^5 + \frac{39}{10}x^6 - \frac{171}{35}x^7 + \frac{108}{35}x^8 - \frac{27}{35}x^9 \\ \beta_1(t) &= -\frac{1}{204120} - \frac{3023}{65318400}x - \frac{1313}{1451520}x^2 + \frac{1}{24}x^4 + \frac{137}{600}x^5 - \frac{9}{16}x^6 + \frac{51}{70}x^7 - \frac{27}{56}x^8 + \frac{9}{70}x^9 \end{aligned} \right\} \quad (8)$$

Solving (6) for the independent solution gives a continuous algorithm of the form

$$y(t) = \sum_{i=0}^1 \frac{(jh)^i}{i!} y_n^{(i)} + h^3 \left[\sum_{j=0}^1 \sigma_j(t) f_{n+j} + \sigma_s(t) f_{n+s} \right], s = \frac{1}{6} \left(\frac{1}{6} \right) \frac{5}{6} \quad (9)$$

Evaluating (9) at $t = \frac{1}{6} \left(\frac{1}{6} \right) 1$, gives a discrete algorithm of the form (1) where,

$$Y_m^{(i)} = \begin{bmatrix} y_{n+\frac{1}{6}}^{(i)} & y_{n+\frac{1}{3}}^{(i)} & y_{n+\frac{1}{2}}^{(i)} & y_{n+\frac{2}{3}}^{(i)} & y_{n+\frac{5}{6}}^{(i)} & y_{n+1}^{(i)} \end{bmatrix}, F(Y_m) = \begin{bmatrix} f_{n+\frac{1}{6}} & f_{n+\frac{1}{3}} & f_{n+\frac{1}{2}} & f_{n+\frac{2}{3}} & f_{n+\frac{5}{6}} & f_{n+1} \end{bmatrix},$$

$$y_m^{(i)} = \begin{bmatrix} y_{n-\frac{1}{6}}^{(i)} & y_{n-\frac{1}{3}}^{(i)} & y_{n-\frac{1}{2}}^{(i)} & y_{n-\frac{2}{3}}^{(i)} & y_{n-\frac{5}{6}}^{(i)} & y_n^{(i)} \end{bmatrix}$$

And $A^{(0)}$ is a 6×6 identity matrix. When $i = 0$

$$e_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, e_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{5}{6} \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, e_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{72} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{18} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 & 0 & \frac{4}{18} \\ 0 & 0 & 0 & 0 & 0 & \frac{25}{72} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

$$d_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{343801}{783820800} \\ 0 & 0 & 0 & 0 & 0 & \frac{6887}{3061800} \\ 0 & 0 & 0 & 0 & 0 & \frac{1959}{358400} \\ 0 & 0 & 0 & 0 & 0 & \frac{3863}{382725} \\ 0 & 0 & 0 & 0 & 0 & \frac{505625}{31352832} \\ 0 & 0 & 0 & 0 & 0 & \frac{33}{1400} \end{bmatrix}, b_0 = \begin{bmatrix} \frac{6031}{9331200} & -\frac{32981}{52254720} & \frac{5177}{9797760} & -\frac{15107}{52254720} & \frac{5947}{65318400} & -\frac{9809}{783820800} \\ \frac{1499}{255150} & -\frac{233}{58320} & \frac{52}{15309} & -\frac{379}{204120} & \frac{149}{255150} & -\frac{491}{6123600} \\ \frac{1599}{1599} & -\frac{537}{537} & \frac{1}{1} & -\frac{327}{327} & \frac{129}{129} & -\frac{71}{71} \\ \frac{89600}{4664} & -\frac{71680}{226} & \frac{120}{272} & -\frac{71680}{31} & \frac{89600}{344} & -\frac{358400}{142} \\ \frac{127575}{162125} & -\frac{25515}{85625} & \frac{15309}{66875} & -\frac{3645}{119375} & \frac{127575}{1625} & -\frac{382725}{18625} \\ \frac{2612736}{33} & -\frac{10450944}{3} & \frac{1959552}{2} & -\frac{10450944}{3} & \frac{373248}{3} & -\frac{31352832}{1} \\ \frac{350}{350} & -\frac{560}{560} & \frac{35}{35} & -\frac{280}{280} & \frac{350}{350} & -\frac{1200}{1200} \end{bmatrix}$$

When $i = 1$

$$e_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, e_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{6} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{2} \\ 0 & 0 & 0 & 0 & 0 & \frac{3}{5} \\ 0 & 0 & 0 & 0 & 0 & \frac{6}{1} \end{bmatrix}, d_1 = \begin{bmatrix} \frac{28549}{4354560} \\ \frac{1027}{68040} \\ \frac{253}{10752} \\ \frac{272}{8505} \\ \frac{35225}{870912} \\ \frac{41}{840} \end{bmatrix}$$

$$b_1 = \begin{bmatrix} \frac{275}{20736} & -\frac{5717}{483840} & \frac{10621}{1088640} & -\frac{7703}{1451520} & \frac{403}{241920} & -\frac{199}{870912} \\ \frac{97}{1890} & -\frac{2}{81} & \frac{197}{8505} & -\frac{97}{7560} & \frac{23}{5670} & -\frac{19}{34020} \\ \frac{163}{1792} & -\frac{267}{17920} & \frac{5}{128} & -\frac{363}{17920} & \frac{57}{8960} & -\frac{47}{53760} \\ \frac{376}{2835} & -\frac{2}{945} & \frac{656}{8505} & -\frac{2}{81} & \frac{8}{945} & -\frac{2}{1701} \\ \frac{8375}{8375} & -\frac{3125}{3125} & \frac{25625}{25625} & -\frac{625}{625} & \frac{275}{275} & -\frac{1375}{1375} \\ \frac{48384}{3} & -\frac{290304}{3} & \frac{217728}{17} & -\frac{96768}{3} & \frac{20736}{3} & -\frac{870912}{0} \\ \frac{14}{14} & -\frac{140}{140} & \frac{105}{105} & -\frac{280}{280} & \frac{70}{70} & 0 \end{bmatrix}$$

When $i = 2$

$$e_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, d_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{19087}{362880} \\ 0 & 0 & 0 & 0 & 0 & \frac{1139}{22680} \\ 0 & 0 & 0 & 0 & 0 & \frac{137}{2688} \\ 0 & 0 & 0 & 0 & 0 & \frac{143}{2835} \\ 0 & 0 & 0 & 0 & 0 & \frac{3715}{72576} \\ 0 & 0 & 0 & 0 & 0 & \frac{41}{840} \end{bmatrix}, b_1 = \begin{bmatrix} \frac{2713}{15120} & -\frac{15487}{120960} & \frac{293}{2835} & -\frac{6737}{120960} & \frac{263}{15120} & -\frac{863}{362880} \\ \frac{47}{189} & -\frac{11}{7560} & \frac{166}{2835} & -\frac{269}{7560} & \frac{11}{945} & -\frac{37}{22680} \\ \frac{27}{387} & -\frac{387}{387} & \frac{17}{17} & -\frac{243}{243} & \frac{9}{9} & -\frac{29}{29} \\ \frac{112}{232} & -\frac{4480}{64} & \frac{105}{752} & -\frac{4480}{29} & \frac{560}{8} & -\frac{13440}{4} \\ \frac{945}{725} & -\frac{945}{2125} & \frac{2835}{125} & -\frac{945}{3875} & \frac{945}{235} & -\frac{2835}{275} \\ \frac{3024}{9} & -\frac{24192}{9} & \frac{567}{34} & -\frac{24192}{9} & \frac{3024}{9} & -\frac{72576}{41} \\ \frac{35}{35} & -\frac{280}{280} & \frac{105}{105} & -\frac{280}{280} & \frac{35}{35} & -\frac{840}{840} \end{bmatrix}$$

Analysis of Basic Properties of the Block Algorithm

In this section, we examine the fundamental characteristics of a general block algorithm designed for solving third-order oscillatory differential equations of the form (1).

The analysis focuses on key properties such as the order of the method, error constant, consistency, zero-stability, and the region of absolute stability.

Order and Error Constant

Definition 1: Order of an Algorithm

The linear operator associated with the discrete algorithm (9) is defined as;

$$\ell\{y(t):h\} = A^{(0)}Y_m^{(i)} - \sum_{i=0}^1 \frac{(jh)^{(i)}}{i!} e_i y_n^{(i)} + h^{(3-i)} [d_i f(y_n) + b_i F(Y_m)] \quad (10)$$

Assuming that $y(t)$ is sufficiently differentiable, we write the terms in (10) as a Taylor series expansion about the point t to obtain the expression;

$$\ell\{y(t):h\} = C_0 y(t) + C_1 y'(t) + \dots + C_p h^p y^p(t) + C_{p+1} h^{p+1} y^{p+1}(t) + C_{p+2} h^{p+2} y^{p+2}(t) + \dots \quad (11)$$

where the constant coefficients $c_p, p = 0, 1, 2, \dots$ are given by;

$$\left. \begin{aligned} c_0 &= \sum_{j=0}^k \alpha_j \\ c_1 &= \sum_{j=0}^k (j\alpha_j - \beta_j) \\ &\vdots \\ &\vdots \\ c_p &= \sum_{j=0}^k \left[\frac{1}{q!} j^q \alpha_j - \frac{1}{(q-1)!} j^{q-1} \beta_j \right], \quad q = 2, 3, \dots \end{aligned} \right\} \quad (12)$$

The block algorithm (3) and the associated linear difference operators are said to have order p if $C_0 = C_1 = \dots = C_p = C_{p+1} = C_{p+2} = 0, C_{p+3} \neq 0$. The order is also defined as the largest positive real number p that quantifies the rate of convergence of a numerical approximation of a differential equation to that of the exact solution.

Definition 2: Error Constant

The term C_{p+3} is called the error constant and implies that the local truncation error for (9) is given by,

$$T_{n+k} = C_{p+3} h^{p+3} y^{p+3}(t) + O(h^{p+4}) \quad (13)$$

The error constant is the accumulated error when the order of an algorithm has been computed. Therefore, according to definition 1 and 2 above, the order of the block algorithm is $p = [8 \ 8 \ 8 \ 8 \ 8 \ 8]$ and its error constant of block algorithm is

$$C_{p+3} = \left[\frac{4001}{109709829734400} \quad \frac{199}{857108044800} \quad \frac{29}{50164531200} \quad \frac{29}{26784626400} \quad \frac{7625}{4388393189376} \quad \frac{1}{391910400} \right]$$

Consistency

Definition 3: Consistency

The continuous block algorithm (9) is said to be consistent if it satisfies the following conditions:

- (i) the order $p \geq 1$,
- (ii) $\sum_{j=0}^k \alpha_j = 0$, and
- (iii) $\rho'(1) = \sigma(1)$

Remark

Condition (i) is a sufficient condition for the associated block algorithm (9) to be consistent, i.e. $p \geq 1$. Hence, the block algorithm (9) is consistent

Zero-Stability

Definition 4: Zero-Stability

A block algorithm (9) is said to be zero-stable if the roots z_s , $s = 1, 2, \dots, n$ of the first characteristic polynomial $\bar{\rho}(z)$, defined by

$$\bar{\rho}(z) = \det[zA^{(0)} - E] \quad (14)$$

satisfies $|z_s| \leq 1$ and every root with $|z_s| = 1$ has multiplicity not exceeding the order of the differential equation as $h \rightarrow 0$. Moreover, as $h \rightarrow 0$, $\rho(z) = z^{r-\mu}(z-1)^\mu$, where μ is the order of the differential equation, r is the order of the matrices $A^{(0)}$ and E . The main consequence of zero-stability is to control the propagation of the error as the integration proceeds. Applying definition 4 on (9), the first characteristic polynomial is given by,

$$\rho(z) = \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] - z \left[\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{cccccc} z & 0 & 0 & 0 & 0 & -1 \\ 0 & z & 0 & 0 & 0 & -1 \\ 0 & 0 & z & 0 & 0 & -1 \\ 0 & 0 & 0 & z & 0 & -1 \\ 0 & 0 & 0 & 0 & z & -1 \\ 0 & 0 & 0 & 0 & 0 & z-1 \end{array} \right] = z^5(z-1)$$

Thus, solving for z in

$$z^5(z-1) = 0 \quad (15)$$

gives $z_1 = z_2 = z_3 = z_4 = z_5 = 0$, $z_6 = 1$. Hence, the block algorithm (9) is zero-stable.

Convergence

Definition 5: Convergence

The block algorithm (9) is said to be convergent if, for all problems of the form (1) subject to the hypothesis of the Lipchitz condition, we have that

$$\lim_{\substack{h \rightarrow 0 \\ jh = x-a}} y_j = y(x_j) \quad (16)$$

holds for $x \in [a, b]$ and for all solutions $\{y_j\}$.

The convergence of the continuous linear multistep method (2) is considered in the light of the basic properties discussed earlier in conjunction with the fundamental theorem of Dahlquist for LMMs. In what follows, we state without proof, the Dahlquist theorem.

Theorem 3.1:

The necessary and sufficient conditions for a linear multistep method to be convergent are that it be consistent and zero-stable. Thus, the block algorithm (9) is convergent. The proof can be seen in Otaide and Oluwayemi (2024).

Region of Absolute Stability of our Method

Definition 6: Region of absolute stability is a region in the complex z plane, where $z = \lambda h$. It is defined as those values of z such that the numerical solutions of $y''' = -\lambda^3 y$ satisfy $y_j \rightarrow 0$ as $j \rightarrow \infty$ for any initial condition (Oyedepo, Ayinde, et al., 2024).

To determine the regions of absolute stability of block algorithm (9), a method that requires neither the computation of roots of a polynomial nor solving of simultaneous inequalities was adopted. This method according to Lambert is called the Boundary Locus Method (BLM). The stability region obtained in Figure 1 is A-stable, since it contains the whole of the left-half plane of the figure.

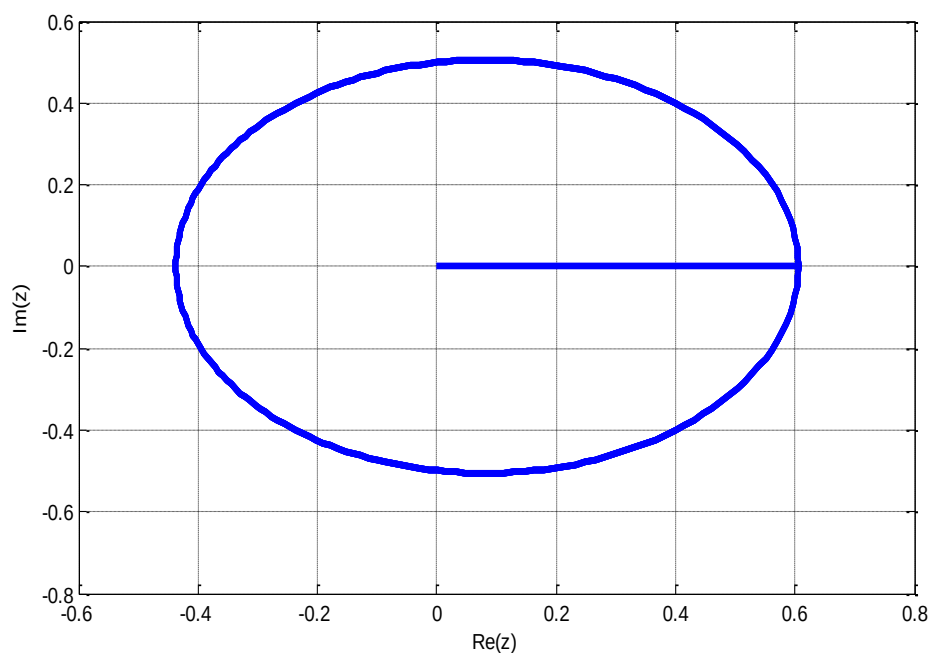


Figure 1. Region of Absolute Stability of our method.

Numerical Examples

This section deals with the numerical simulation of the performance of the New Algorithm (NA) presented in equation (9) for numerical integrations of the Volterra integro-differential equation of the second kind (1). The absolute error or maximum error in the NA is computed and compared with the existing methods of Mohamed and Zanariah (2015), Majid and Mohamed (2019), Olowe et al. (2023), Kamoh et al. (2017a), Abdulganiy et al. (2018) and Kamoh et al. (2017b). The results are tabulated and graphical shown.

Example 1

Consider the Linear Volterra integro-differential equation of the form

$$y'(x) = 1 - \int_0^x y(t)dt, y(0) = 0, 0 \leq x \leq 1 \quad (17)$$

with exact solution given by

$$y(x) = \sin(x) \quad (18)$$

Example 1 was solved using the New Algorithm ENA, table 1 and figure 2 shows the comparison of new algorithm (NA) in equation (9) with the existing methods of EABM5 and E2P3SBM in Mohamed and Zanariah (2015) and E2P3B in Majid and Mohamed (2019) in terms of errors.

Example 2

Consider the Volterra integro-differential equation of the form

$$y'(x) = -\sin(x) - \cos(x) + \int_0^x 2\cos(x-t)y(x)dt, y(0) = 1 \quad (19)$$

with exact solution given by

$$y(x) = \exp(-x) \quad (20)$$

Table 2 together with figure 3 present the results of example 2. The numerical results in comparison of ENA with the EABM5 and Two Point three-step block method E2P3B both as given in Majid and Mohamed (2019). The results are also compared with the EBTfM of Olowe et al. (2023).

Example 3

Consider the nonlinear Volterra integro-differential equation of the form

$$y''(x) + \int_0^x (y(t))^2 dt + \left(\frac{x}{2} - \sinh(x) - \frac{1}{4} \sinh(2x) \right) = 0, y(0) = 0, y'(0) = 1 \quad (21)$$

with exact solution given by

$$y(x) = \sinh(x) \quad (22)$$

Table 3 together with figure 4 shows the comparison example 3, which was solved using the NA and the results are compared with those given by ECMM developed by Kamoh et al. (2017a). The results are also compared with EFDTFBBDF developed Abdulganiy et al. (2018) and methods E4SBM and E5SBM developed by Kamoh et al. (2017b).

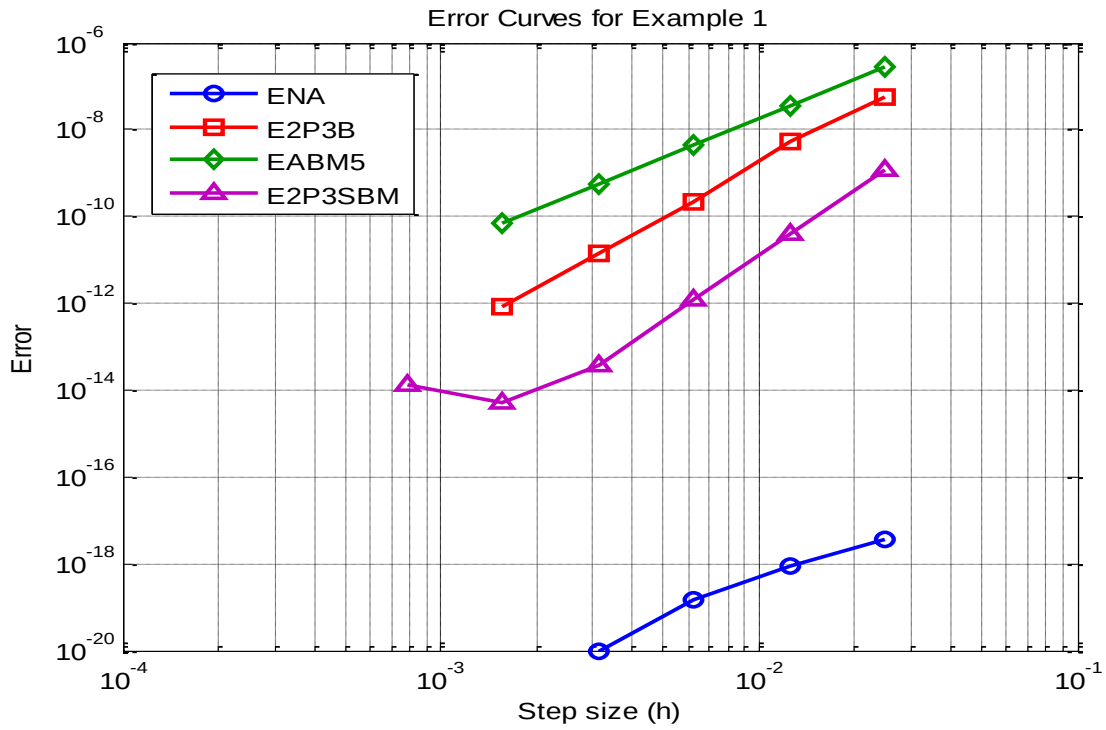


Figure 2. Textual Curve of Table 1

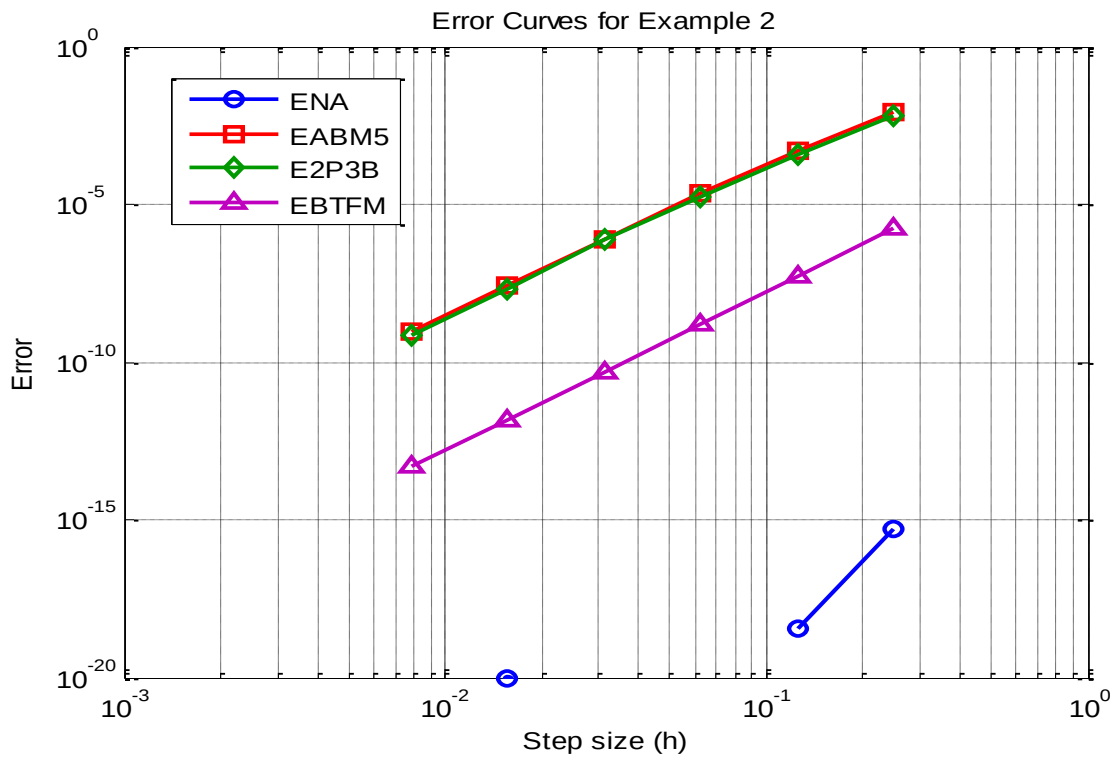


Figure 3. Textual Curve of Table 2

Table 1. the numerical Results for Example 1

x	Exact Solution	Computed Solution	ENA	E2P3B	EABM5	E2P3SBM
0.025	0.02499739591471233066	0.02499739591471232708	3.5800(-18)	5.7323(-08)	2.8951(-07)	1.2349(-09)
0.0125	0.01249967448170978872	0.01249967448170978962	9.0000(-19)	5.5893(-09)	3.6127(-08)	3.8642(-11)
0.00625	0.00624995930997530612	0.00624995930997530597	1.5000(-19)	2.2443(-10)	4.3953(-09)	1.2080(-12)
0.003125	0.00312499491373946269	0.00312499491373946268	1.0000(-20)	1.3908(-11)	5.4213(-10)	3.7751(-14)
0.0015625	0.00156249936421720001	0.00156249936421720001	0.0000(00)	8.6930(-13)	6.7325(-11)	5.3291(-15)
0.00078125	0.00078124992052714272	0.00078124992052714272	0.0000(00)	-	-	1.3545(-14)

Table 2. numerical results for example 2

x	Exact Solution	Computed Solution	ENA	EABM5	E2P3B	EBTFM
0.250	0.77880078307140486825	0.77880078307140540806	5.3981(-16)	8.1337(-03)	6.1138(-03)	1.6867(-06)
0.1250	0.88249690258459540286	0.88249690258459540322	3.6000(-19)	4.7616(-04)	3.9009(-04)	5.0991(-08)
0.06250	0.93941306281347578612	0.93941306281347578612	0.0000(00)	2.1034(-05)	1.6881(-05)	1.5744(-09)
0.031250	0.96923323447634408185	0.96923323447634408185	0.0000(00)	7.8509(-07)	7.8509(-07)	4.8948(-11)
0.0156250	0.98449643700540840599	0.98449643700540840598	1.0000(-20)	2.6828(-08)	2.0516(-08)	1.5261(-12)
0.00781250	0.99221793826024351211	0.99221793826024351211	0.0000(00)	8.7684(-10)	6.6334(-10)	4.7637(-14)

Table 3. numerical results for example 3

x	Exact Solution	Computed Solution	ENA	ECMM	EFDTFBBDf	E4SBM	E5SBM
0.16	0.16068354101279944828	0.16068354101279129786	8.1504(-15)	4.5700(-08)	7.8800(-05)	4.5800(-08)	4.5700(-08)
0.32	0.32548936363113307984	0.32548936363113985654	6.7767(-15)	3.7140(-07)	2.9800(-04)	3.7160(-07)	3.7150(-07)
0.48	0.49864550519337626463	0.49864550519337347295	2.7917(-15)	1.2858e(06)	6.5900(-04)	1.2861(-06)	1.2859(-06)
0.64	0.68459422763095139805	0.68459422763099268709	4.1289(-14)	3.1554(-06)	1.1900(-03)	3.1559(-06)	3.1555(-06)
0.80	0.88810598218762300658	0.88810598218761237564	1.0631(-14)	6.4379(-06)	1.8900(-03)	6.4386(-06)	6.4381(-06)
0.96	1.11440179372400284780	1.11440179374552187898	2.1519(-11)	1.1717(-05)	2.7900(-03)	1.1718(-05)	1.1718(-05)

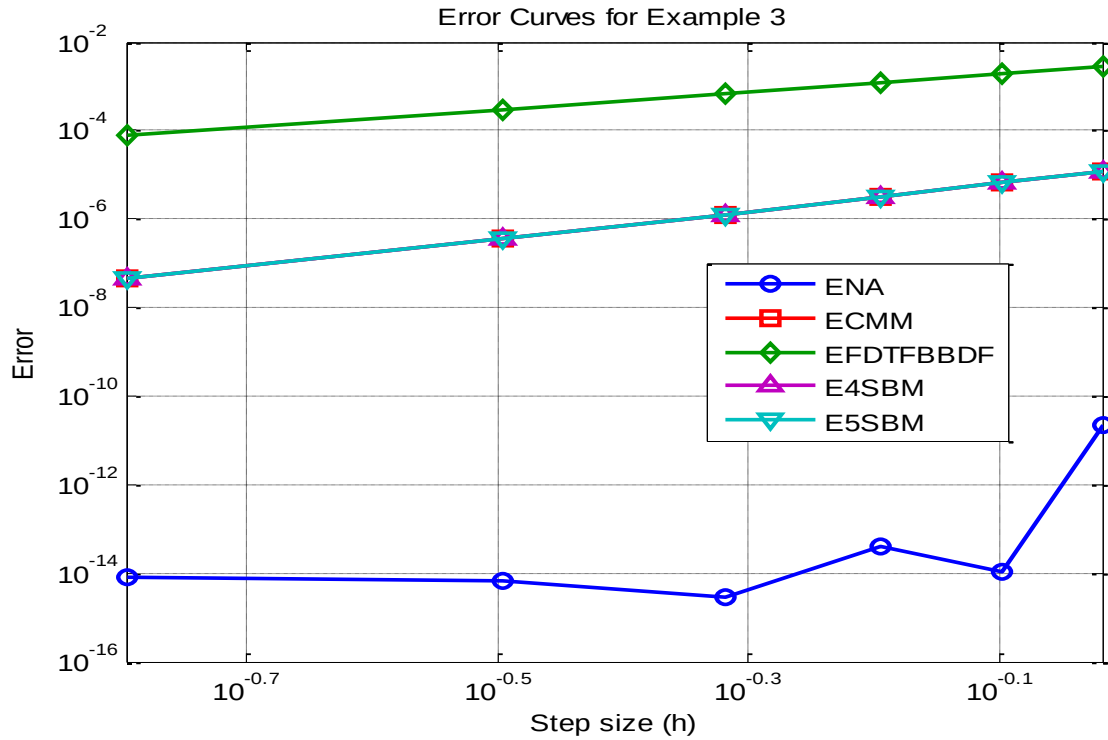


Figure 4. Textual Curve of Table 3

Discussion

The numerical performance of a newly developed algorithm (NA) designed for solving Volterra integro-differential equations of the second kind. The study evaluates the accuracy of the developed ENA method by solving three examples involving both linear and nonlinear equations with known exact solutions. For each example, the absolute errors of ENA are calculated and compared with several established numerical schemes in existing literature, such as EABM5, E2P3B, E2P3SBM, EBTfM, ECMM, EFDTFBBD, E4SBM, and E5SBM. This comparison provides a detailed assessment of the algorithm's efficiency, precision and stability.

In Example 1, the numerical results demonstrate that the ENA method yields extremely small error values, often as low as 10^{-18} to 10^{-20} , which are considerably lower than those produced by EABM5, E2P3B, and E2P3SBM. This confirms the high level of accuracy achieved by the new algorithm when applied to linear Volterra equations. In Example 2, which involves an exponential-type equation, the ENA method once again produces near-zero errors, outperforming both older and more recent methods, including EABM5, E2P3B, and EBTfM. Across all tested step sizes, ENA remains the most accurate method, indicating its strong computational effectiveness.

Example 3 assesses the performance of ENA when applied to a nonlinear Volterra integro-differential equation. The results show that while other competing methods such as ECMM, EFDTFBBD, E4SBM, and E5SBM generate larger errors, sometimes up to the order of 10^{-3} , ENA maintains very small error values throughout. This highlights the method's robustness, consistency and reliability even under nonlinear and more complex problem settings. The accompanying figures visually confirm ENA's closeness to the exact solution. The results demonstrate that the new algorithm offers significant improvements in accuracy and convergence, making it a highly promising tool for solving various types of Volterra integro-differential equations.

The numerical examples demonstrate that the new algorithm (NA) delivers highly accurate and stable results across a variety of Volterra integro-differential equations. In all three examples, the computed solutions closely match the exact solutions and consistently outperform existing methods such as EABM5, E2P3B, E2P3SBM, ECMM, and EFDTFBDDF. Example 1 shows that the NA maintains precise results for linear equations, while Examples 2 and 3 illustrate its effectiveness for both linear and nonlinear problems. The results remain reliable across different step sizes, indicating strong numerical stability. Furthermore, the algorithm's performance across multiple problem types highlights its versatility and suitability for a wide range of integro-differential equations in scientific and engineering applications.

References

- Abdulganiy, R. I., Akinfenwa, O. A., & Okunuga, S. A. A. (2018). Construction of L-stable second derivative trigonometrically fitted block backward differentiation formula for the solution of oscillatory initial value problems. *African Journal of Science, Technology, Innovation and Development*, 10(4), 411–419. <https://doi.org/10.1080/20421338.2018.1467859>
- Aduroja, O. O., Ajileye, A. M., Ajileye, G., & Oyedepo, T. (2025). Numerical solution of Volterra integro-differential equations of fractional order with initial conditions using collation approach. *Journal of Fractional Calculus and Application*, 16(1), 1–10.
- Agarwala, G., Mohana, M., Menon, A. M., Sharma, A., Dakal, T. C., & Purohit, S. D. (2022). Analysis of the Adomian decomposition method to estimate the COVID-19 pandemic. In *Methods of mathematical modeling* (pp. 173–187). Elsevier. <https://doi.org/10.1016/B978-0-323-99888-8.00011-5>
- Agbata, B. C., Ani, B. N., Shior, M. M., Ezugorie, I. G., Paul, R. V., & Meseda, P. K. (2022). Analysis of Adomian decomposition method and its application for solving linear and nonlinear differential equations. *Asian Research Journal of Mathematics*, 18(2), 56–70. <https://doi.org/10.9734/ARJM/2022/v18i230368>
- Chang, S. H. (2019). A variational iteration method involving Adomian polynomials for a strongly nonlinear boundary value problem. *East Asian J. Appl. Math.*, 9, 153–164.
- Cimen, E., & Yata, E. (2020). Numerical solution of Volterra integro-differential equation with dealy. *Journal of Mathematics and Computer Science*, 20, 255–263. <https://doi.org/10.22436/jmcs.02003.08>
- Islam, D. U., & Srivastava, A. (2024a). Comparative analysis of Adomian decomposition method and variational iteration method for solving nonlinear integral equations. *UGC Care Group I Listed Journal*, 14(8, No. 01), 129–133.
- Islam, D. U., & Srivastava, A. (2024b). Solving Fredholm integral equations by numerical methods: Adomian decomposition method and variational iteration method. *International Journal of Engineering Research & Technology*, 13(6), 1–7. <https://doi.org/10.386336418>
- Kamoh, N. M., Aboiyar, T., & Kimbir, A. R. (2017a). Continuous multistep methods for Volterra integro-differential equation of the second-order. *ScienceWorld Journal*, 12(3), 11–14.
- Kamoh, N. M., Aboiyar, T., & Kimbir, A. R. (2017b). On investigating some quadrature rules for the solution of second-order Volterra integro-differential equation. *Journal of Mathematics*, 13(5), 45–50.
- Linjun, X. (2023). A new modification of Adomian decomposition method for Volterra integral equations of the second kind. *Journal of Applied Mathematics*, 200. <https://doi.org/10.1155/2013/795015>
- Liu, C. S., Li, B., & Kuo, C. L. (2025). Variational iteration and linearized Liapunov methods for seeking the analytic solutions of nonlinear boundary value problems. *Mathematics*, 13(354), 1–38. <https://doi.org/10.3390/math13030354>
- Majid, Z. A., & Mohamed, N. A. (2019). Fifth order multistep block method for solving Volterra integro-differential equations of second kind. *Sains Malaysiana*, 48(3), 677–684. <https://doi.org/10.17576/jms-2019-4803-22>

- Maturi, D. A., & Malaikah, H. M. (2021). The Adomian decomposition method for solving nonlinear partial differential equation using Maple. *Advances in Pure Mathematics*, 11, 595–603. <https://doi.org/10.2436/apm.2019.116038>
- Memon, M., Amur, K. B., & W. A. (2023). Combined variational iteration method with Chebyshev wavelet for the solution of convection-diffusion-reaction problem. *Mehran University Research Journal of Engineering and Technology*, 42(2), 93–107. <https://doi.org/10.22581/muet1982.2302.11>
- Mohamed, N. A., & Zanariah, A. M. (2015). One-step block method for solving Volterra integro-differential equations. *AIP Conference Proceedings*, 1682, 020018. <https://doi.org/10.1063/1.4932427>
- Olowe, R. A., Akinfewa, O. A., Abdulganiy, R. I., & Okunuga, S. A. (2023). Trigonometrically-fitted Simpson's method for solving Volterra integro-differential equations. *International Journal of Mathematical Sciences and Optimization: Theory and Applications*, 8(2), 68–78. <https://doi.org/10.6084/m9.figshare.22015772>
- Otaide, I. J., & Oluwayemi, M. O. (2024). Numerical treatment of linear Volterra integro differential equations using variational iteration algorithm with collocation. *Partial Differential Equations in Applied Mathematics*, 10, 1–7. <https://doi.org/10.1016/j.padiff.2024.100693>
- Oyedepo, T., Ajileye, G., Ayinde, A. M., & Otaide, I. J. (2023). Bernstein collocation technique for Volterra-Fredholm fractional equations. *Journal of Fractional Calculus and Application*, 14(1), 228–235.
- Oyedepo, T., Ajileye, G., & Ayoade, A. A. (2024). Bernstein computational algorithm for integro-differential equations. *Partial Differential Equations in Applied Mathematics*, 11, 1–7. <https://doi.org/10.1016/j.padiff.2024.100896>
- Oyedepo, T., Ayinde, A. M., & Didigwu, E. N. (2024). Vieta-Lucas polynomial computational technique for Volterra integro-differential equations. *Electronic Journal of Mathematical Analysis and Application, of Fractional Calculus and Application*, 12(1), 1–8.
- Pan, C., Zhang, R., Luo, H., & Shen, H. (2016). Simplified variational iteration method for solving ordinary differential equations and eigenvalue problems. *Advances in Mechanical Engineering*, 8, 1–13. <https://doi.org/10.1177/1687814016681469>
- Sunday, J. (2018). On the oscillation criteria and computation of third order oscillatory differential equations. *Communication in Mathematics and Applications*, 6(4), 615–625.