Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://asr.nsps.org.ng/index.php/asr

African Scientific Reports 5 (2026) 476

African Scientific Reports

Computational method for Volterra integro-differential equations of the second kind

Abdullahi Muhammed Ayinde[a,*], Adam Ajimoti Ishaq[b], Lukman Olalekan Ahmed[c], Emmanuel Jacob[d]

[a]*Department of Mathematics, University of Abuja, FCT, 900211, Nigeria*
[b]*Department of Physical Sciences, Al-Hikmah University, Ilorin, Nigeria*
[c]*Department of Mathematics, Baze University, Abuja, Nigeria*
[d]*Department of Mathematical Sciences, Prince Abubakar Audu University, Anyigba, Nigeria*

Abstract

This study presents a computational method (CM) for solving second-kind Volterra integro-differential equations (VIDEs), which are widely used in engineering, physics, biology, and control systems. The method combines interpolation and collocation techniques to produce a continuous one-step computational scheme that accurately approximates the solution while maintaining stability and convergence properties. Consistency, zero-stability, order of accuracy, convergence, and the region of absolute stability are examined to confirm the suitability of the method for both linear and nonlinear problems. Numerical experiments with selected VIDEs compare the CM with the fifth–order Adams–Bashforth–Moulton predictor–corrector method, the two-point three-step block method, the trigonometrically fitted block method, the Haar wavelet method, and trapezoidal schemes. Across the tested examples, the CM provides accurate, stable, and efficient approximations, making it a useful tool for solving complex integro-differential systems in practical applications.

DOI:10.46481/asr.2026.5.2.476

Keywords: Volterra integro-differential equations, Interpolation technique, Stability analysis, Collocation technique

Article History:
Received: 12 March 2026
Received in revised form: 12 May 2026
Accepted for publication: 16 May 2026
Available online: 15 July 2026

1. Introduction

Volterra integro-differential equations (VIDEs) of the second kind are fundamental mathematical tools used to model various physical, biological, and engineering systems. These equations assess the current state of a system by combining the present rate of change with its prior development. The general form of a Volterra integro-differential equation is given by

$$Y(g)y^{(d)}(g) = \Theta(g) + T\int_{\omega}^{g} K(g,x)y(x)\,dx, \tag{1}$$

*Corresponding author Tel. No.: +234-806-222-8679
Email address: `ayinde.abdullahi@uniabuja.edu.ng` (Abdullahi Muhammed Ayinde)

where T is a constant parameter, $K(g,x)$ is the kernel of the integral equation, $\Theta(g)$ is a function, and g and ω are limits of integration that may be constants, variables, or a combination of both. When $Y(g) = 1$, Eq. (1) becomes

$$y^{(d)}(g) = \Theta(g) + T \int_0^g K(g,x)y(x)\,dx. \tag{2}$$

The integral Eq. (2) is called the Volterra integro-differential equation of the second kind. The research developed a numerical method to solve Volterra integro-differential equations, which are expressed in Eq. (2).

Second-kind VIDEs demonstrate applications in four distinct fields: viscoelasticity, population dynamics, control systems, and heat conduction problems. The development of numerical methods for solving these equations has seen significant progress through recent advances in computational mathematics. Researchers have developed block methods[1, 2], hybrid schemes [3], and iterative methods such as the Variational Iteration Method and Adomian Decomposition Method [4, 5] that effectively solve both linear and nonlinear problems. Brunner [6] evaluated these methods using three main criteria: convergence analysis, error estimation, and computational cost evaluation. The field of numerical analysis continues to actively investigate robust computational methods for solving second-kind VIDEs.

The second kind of Volterra integro-differential equations presents major difficulties for both analytical work and computational methods because it involves both differential and integral components. The integral component functions as a memory term that establishes nonlocal connections, making both theoretical research and numerical approximation more challenging [7, 8]. Real-world problem solvers face difficulties because analytical solutions are often only available for linear equations with simple kernels. Existing numerical methods may suffer from issues such as low accuracy, accumulation of truncation errors, high computational cost, instability for stiff problems, and difficulty handling nonlinear kernels [9, 10]. These limitations make these methods less effective and dependable when scientists and engineers use them to develop real-world scientific and engineering models [11].

Many traditional computational methods require stepwise integration, which demands more computation time and results in lower performance across all problem types, especially when dealing with large-scale or rapidly changing systems. Some methods also lack rigorous convergence analysis or produce results that are sensitive to step size selection [12]. Therefore, there is a clear need to develop a more efficient, stable, and accurate computational method that can directly solve second-kind Volterra integro-differential equations without creating excessive computational demands.

Previous studies have shown substantial advancements in solving VIDEs and related models through both analytical and computational methods. Maturi [13] used the Variational Iteration Method (VIM) on second-kind VIDEs through Maple 18, demonstrating accuracy through numerical examples that showed exact solutions in both tabulated results and graphical comparisons. Uwaheren *et al.* [14] used Akbari-Ganji's Method (AGM) with Legendre polynomial bases to solve Volterra integro-differential difference equations. These studies demonstrate how semi-analytical iterative methods produce accurate results with less complex implementation.

The theoretical foundations of various contributions receive equal importance with their computational techniques. Alharbi and Althubiti [15] transformed second-order integro-differential equations into equivalent Volterra–Fredholm integral equations, establishing existence and uniqueness results within a Hilbert space framework using contraction mapping principles. Ghosh and Mohapatra [16] developed a finite-difference method to solve time-fractional Volterra integro-differential systems, which use Caputo derivatives by combining L1 discretization with the composite trapezoidal rule. The research offers precise demonstrations about existence and uniqueness together with stability results and convergence proofs while providing error analysis which covers both smooth solutions and weakly singular solutions, thus enhancing the dependability of the suggested method.

Recent studies show that hybrid and spectral-based methods have become essential research tools. Otaide and Oluwayemi [17] developed a computational framework for higher-order linear VIDEs by combining the variational iteration algorithm with a collocation method using fourth-kind shifted Chebyshev polynomials. They used Banach's fixed-point theorem to prove convergence and demonstrated that higher polynomial degrees produced more accurate results. Ouaret and Malendas [18] developed an iterative collocation method for nonlinear delay Volterra integral equations using Lagrange polynomials with complete theoretical foundations covering Lipschitz conditions and discrete Gronwall inequalities. The studies show how researchers increasingly combine spectral approximation methods with iterative correction techniques.

The present study extends the development of solution techniques for various Volterra-type equations exhibiting different forms of delay and spatial interaction effects. Cimen and Yatar [10] proposed a fitted finite difference method for solving linear first-order Volterra delay integro-differential equations, incorporating a priori bounds, truncation error analysis, and a proof of first-order convergence. Their work was further expanded to nonlocal models by introducing a peridynamic differential operator into the Volterra integro-differential framework, enabling the simulation of discontinuities in viscoelastic materials [19]. To obtain numerical solutions, the authors employed Galerkin finite element methods for spatial discretization alongside the backward Euler scheme for time integration, demonstrating both stability and first-order convergence of the method.

2. Formation of the computational method

Consider the power-series polynomial of the form

$$y(g) = \sum_{j=0}^{r+s-1} a_j g^j. \tag{3}$$

We consider the approximation of Eq. (3) as a basis function, where r and s are the number of distinct interpolation and collocation points, respectively. By differentiating Eq. (3) three times, we have

$$y'''(g) = \sum_{j=0}^{r+s-1} j(j-1)(j-2)a_j g^{j-3}. \tag{4}$$

A one-step grid is considered with a constant step size h given by $h = g_{n+1} - g_n$, $j = 0, 1$, and four off-step points at $g_{n+\frac{1}{8}}, g_{n+\frac{1}{4}}, g_{n+\frac{3}{8}}$ and $g_{n+\frac{1}{2}}$

Interpolating (3) at point $g_{n+r}, r = \frac{1}{8}(\frac{1}{4})\frac{3}{8}$ and collocating (4) at points $g_{n+s}, s = 0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, 1$, gives a system of equations of the form

$$AG = U. \tag{5}$$

where

$$A = [a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8]^T,$$
$$U = [y_{n+\frac{1}{8}} \quad y_{n+\frac{1}{4}} \quad y_{n+\frac{3}{8}} \quad f_n \quad f_{n+\frac{1}{8}} \quad f_{n+\frac{1}{4}} \quad f_{n+\frac{3}{8}} \quad f_{n+\frac{1}{2}} \quad f_{n+1}]^T \quad \text{and}$$

$$G = \begin{bmatrix} 1 & g_{n+\frac{1}{8}} & g^2_{n+\frac{1}{8}} & g^3_{n+\frac{1}{8}} & g^4_{n+\frac{1}{8}} & g^5_{n+\frac{1}{8}} & g^6_{n+\frac{1}{8}} & g^7_{n+\frac{1}{8}} & g^8_{n+\frac{1}{8}} \\ 1 & g_{n+\frac{1}{4}} & g^2_{n+\frac{1}{4}} & g^3_{n+\frac{1}{4}} & g^4_{n+\frac{1}{4}} & g^5_{n+\frac{1}{4}} & g^6_{n+\frac{1}{4}} & g^7_{n+\frac{1}{4}} & g^8_{n+\frac{1}{4}} \\ 1 & g_{n+\frac{3}{8}} & g^2_{n+\frac{3}{8}} & g^3_{n+\frac{3}{8}} & g^4_{n+\frac{3}{8}} & g^5_{n+\frac{3}{8}} & g^6_{n+\frac{3}{8}} & g^7_{n+\frac{3}{8}} & g^8_{n+\frac{3}{8}} \\ 0 & 0 & 0 & 6 & 24g_n & 60g_n^2 & 120g_n^3 & 210g_n^4 & 336g_n^5 \\ 0 & 0 & 0 & 6 & 24g_{n+\frac{1}{8}} & 60g^2_{n+\frac{1}{8}} & 120g^3_{n+\frac{1}{8}} & 210g^4_{n+\frac{1}{8}} & 336g^5_{n+\frac{1}{8}} \\ 0 & 0 & 0 & 6 & 24g_{n+\frac{1}{4}} & 60g^2_{n+\frac{1}{4}} & 120g^3_{n+\frac{1}{4}} & 210g^4_{n+\frac{1}{4}} & 336g^5_{n+\frac{1}{4}} \\ 0 & 0 & 0 & 6 & 24g_{n+\frac{3}{8}} & 60g^2_{n+\frac{3}{8}} & 120g^3_{n+\frac{3}{8}} & 210g^4_{n+\frac{3}{8}} & 336g^5_{n+\frac{3}{8}} \\ 0 & 0 & 0 & 6 & 24g_{n+\frac{1}{2}} & 60g^2_{n+\frac{1}{2}} & 120g^3_{n+\frac{1}{2}} & 210g^4_{n+\frac{1}{2}} & 336g^5_{n+\frac{1}{2}} \\ 0 & 0 & 0 & 6 & 24g_{n+1} & 60g^2_{n+1} & 120g^3_{n+1} & 210g^4_{n+1} & 336g^5_{n+1} \end{bmatrix}.$$

Solving Eq. (5) for $a_j, j = 1(0)8$ which are constants to be obtained and substituting into Eq. (3) gives a one-step continuous computational method of the form,

$$y(g) = \alpha_{\frac{1}{8}}(g)y_{n+\frac{1}{8}} + \alpha_{\frac{1}{4}}(g)y_{n+\frac{1}{4}} + \alpha_{\frac{3}{8}}(g)y_{n+\frac{3}{8}} + h^3\left[\sum_{j=0}^{1}\beta_j(g)f_{n+j} + \beta_s(g)f_{n+s}\right], s = \frac{1}{8}\left(\frac{1}{8}\right)\frac{1}{2}, 1, \tag{6}$$

where $\alpha_r(g), \beta_j(g)$, and $\beta_s(g)$ are presented as functions of x, with

$$t = \frac{g - g_n}{h}, \tag{7}$$

where

$$\left.\begin{aligned}
\alpha_{\frac{1}{8}} &= 32t^2 - 20t + 3\\
\alpha_{\frac{1}{4}} &= -64t^2 + 32t - 3\\
\alpha_{\frac{3}{8}} &= 32t^2 - 12t + 1\\
\beta_0 &= -\frac{32}{63}t^8 + \frac{64}{35}t^7 - \frac{23}{9}t^6 + \frac{11}{6}t^5 - \frac{53}{72}t^4 + \frac{1}{6}t^3 - \frac{51529}{2580480}t^2 + \frac{1787}{1720320}t - \frac{7}{655360}\\
\beta_{\frac{1}{8}} &= \frac{1024}{441}t^8 - \frac{17408}{2205}t^7 + \frac{448}{45}t^6 - \frac{1856}{315}t^5 + \frac{32}{21}t^4 - \frac{16027}{188160}t^2 + \frac{3679}{225792}t - \frac{401}{430080}\\
\beta_{\frac{1}{4}} &= -\frac{256}{63}t^8 + \frac{4096}{315}t^7 - \frac{664}{45}t^6 + \frac{328}{45}t^5 - \frac{4}{3}t^4 - \frac{449}{35840}t^2 + \frac{7199}{645120}t - \frac{257}{245760}\\
\beta_{\frac{3}{8}} &= \frac{1024}{315}t^8 - \frac{1024}{105}t^7 + \frac{448}{45}t^6 - \frac{65}{15}t^5 + \frac{32}{45}t^4 - \frac{757}{80640}t^2 + \frac{3}{17920}t + \frac{1}{20480}\\
\beta_{\frac{1}{2}} &= -\frac{64}{63}t^8 + \frac{128}{45}t^7 - \frac{118}{45}t^6 + \frac{47}{45}t^5 - \frac{1}{6}t^4 - \frac{893}{430080}t^2 + \frac{1}{73728}t - \frac{13}{983040}\\
\beta_1 &= \frac{32}{2205}t^8 - \frac{64}{2205}t^7 + \frac{1}{45}t^6 - \frac{1}{126}t^5 + \frac{1}{840}t^4 - \frac{89}{6021120}t^2 + \frac{11}{36126720}t + \frac{1}{13762560}
\end{aligned}\right\}. \tag{8}$$

Evaluating Eq. (6) at a non-interpolating point gives the continuous form

$$\begin{bmatrix} y_n \\ y_{n+\frac{1}{2}} \\ y_{n+1} \end{bmatrix} - \begin{bmatrix} 3 & -3 & 1 \\ 1 & -3 & 3 \\ 15 & -35 & 21 \end{bmatrix} \begin{bmatrix} y_{n+\frac{1}{8}} \\ y_{n+\frac{1}{4}} \\ y_{n+\frac{3}{8}} \end{bmatrix} = h^3 \begin{bmatrix} -\frac{7}{655360} & -\frac{401}{430080} & -\frac{257}{245760} & \frac{1}{20480} & -\frac{13}{983040} & \frac{1}{13762560} \\ \frac{11}{1966080} & -\frac{3}{143360} & \frac{247}{245760} & \frac{59}{61440} & \frac{1}{327680} & \frac{1}{13762560} \\ \frac{3943}{393216} & -\frac{4757}{86016} & \frac{2249}{16384} & -\frac{1361}{12288} & \frac{16943}{196608} & \frac{871}{917504} \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\frac{1}{8}} \\ f_{n+\frac{1}{4}} \\ f_{n+\frac{3}{8}} \\ f_{n+\frac{1}{2}} \\ f_{n+1} \end{bmatrix}. \tag{9}$$

Differentiating Eq. (6) once, gives

$$y'(g) = \alpha'_{\frac{1}{8}}(g)y_{n+\frac{1}{8}} + \alpha'_{\frac{1}{4}}(g)y_{n+\frac{1}{4}} + \alpha'_{\frac{3}{8}}(g)y_{n+\frac{3}{8}} + h^3\left[\sum_{j=0}^{1}\beta'_j(g)f_{n+j} + \beta'_s(g)f_{n+s}\right], s = \frac{1}{8}\left(\frac{1}{8}\right)\frac{1}{2}, 1. \tag{10}$$

Evaluating Eq. (10) at g_n, $g_{n+\frac{1}{8}}$, $g_{n+\frac{1}{4}}$, $g_{n+\frac{3}{8}}$, $g_{n+\frac{1}{2}}$ and g_{n+1}, we have

$$\begin{bmatrix} hy'_n \\ hy'_{n+\frac{1}{8}} \\ hy'_{n+\frac{1}{4}} \\ hy'_{n+\frac{3}{8}} \\ hy'_{n+\frac{1}{2}} \\ hy'_{n+1} \end{bmatrix} - \begin{bmatrix} -20 & 32 & -12 \\ -12 & 16 & -4 \\ -4 & 0 & 4 \\ 4 & -16 & 12 \\ 12 & -32 & 20 \\ 44 & -96 & 52 \end{bmatrix} \begin{bmatrix} y_{n+\frac{1}{4}} \\ y_{n+\frac{1}{8}} \\ y_{n+\frac{3}{8}} \end{bmatrix} = h^3 \begin{bmatrix} \frac{1787}{1720320} & \frac{3678}{225792} & \frac{7199}{645120} & \frac{3}{17920} & -\frac{1}{73728} & \frac{11}{36126720} \\ -\frac{773}{10321920} & \frac{1003}{752640} & \frac{5339}{1290240} & -\frac{83}{322560} & \frac{121}{1720320} & -\frac{31}{72253440} \\ \frac{1}{129024} & -\frac{13}{80640} & -\frac{247}{107520} & -\frac{13}{80640} & \frac{1}{129024} & 0 \\ \frac{29}{1146880} & -\frac{209}{2257920} & \frac{5029}{1290240} & \frac{23}{15360} & -\frac{619}{5160960} & \frac{31}{72253440} \\ \frac{19}{1032192} & \frac{19}{376320} & \frac{7309}{645120} & \frac{2609}{161280} & \frac{307}{286720} & -\frac{11}{36126720} \\ \frac{446413}{5160960} & -\frac{534377}{1128960} & \frac{77283}{71680} & -\frac{166841}{161280} & \frac{1575353}{2580480} & \frac{136181}{12042240} \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\frac{1}{8}} \\ f_{n+\frac{1}{4}} \\ f_{n+\frac{3}{8}} \\ f_{n+\frac{1}{2}} \\ f_{n+1} \end{bmatrix}. \tag{11}$$

Differentiating Eq. (6) twice, yields

$$y''(g) = \alpha''_{\frac{1}{8}}(g)y_{n+\frac{1}{8}} + \alpha''_{\frac{1}{4}}(g)y_{n+\frac{1}{4}} + \alpha''_{\frac{3}{8}}(g)y_{n+\frac{3}{8}} + h^3[\sum_{j=0}^{1}\beta''_j(g)f_{n+j} + \beta''_s(g)f_{n+s}], s = \frac{1}{8}(\frac{1}{8})\frac{1}{2}, 1. \tag{12}$$

Evaluating Eq. (12) at g_n, $g_{n+\frac{1}{8}}$, $g_{n+\frac{1}{4}}$, $g_{n+\frac{3}{8}}$, $g_{n+\frac{1}{2}}$ and g_{n+1}, gives

$$\begin{bmatrix} h^2y''_n \\ h^2y''_{n+\frac{1}{8}} \\ h^2y''_{n+\frac{1}{4}} \\ h^2y''_{n+\frac{3}{8}} \\ h^2y''_{n+\frac{1}{2}} \\ h^2y''_{n+1} \end{bmatrix} - \begin{bmatrix} 64 & -128 & 64 \\ 64 & -128 & 64 \\ 64 & -128 & 64 \\ 64 & -128 & 64 \\ 64 & -128 & 64 \\ 64 & -128 & 64 \end{bmatrix} \begin{bmatrix} y_{n+\frac{1}{4}} \\ y_{n+\frac{1}{8}} \\ y_{n+\frac{3}{8}} \end{bmatrix} = h^3 \begin{bmatrix} -\frac{51529}{1290240} & -\frac{16027}{94080} & -\frac{449}{17920} & -\frac{757}{40320} & \frac{893}{215040} & -\frac{89}{3010560} \\ -\frac{187}{86016} & -\frac{14537}{282240} & \frac{4441}{53760} & \frac{121}{13440} & -\frac{1339}{645120} & \frac{37}{3010560} \\ -\frac{227}{430080} & \frac{183}{31360} & -\frac{43}{32256} & -\frac{103}{13440} & \frac{221}{215040} & -\frac{42}{9031680} \\ \frac{1013}{1290240} & -\frac{403}{94080} & \frac{1357}{17920} & \frac{2267}{40320} & -\frac{149}{43008} & \frac{37}{3010560} \\ -\frac{451}{430080} & \frac{419}{56448} & \frac{2237}{53760} & \frac{2137}{13440} & \frac{27767}{645120} & -\frac{89}{3010560} \\ \frac{207421}{430080} & -\frac{81801}{31360} & \frac{913463}{16280} & -\frac{76711}{13440} & \frac{120601}{43008} & \frac{1046261}{9031680} \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\frac{1}{8}} \\ f_{n+\frac{1}{4}} \\ f_{n+\frac{3}{8}} \\ f_{n+\frac{1}{2}} \\ f_{n+1} \end{bmatrix}. \qquad (13)$$

Combining and solving Eq. (9),Eq. (11) and Eq. (13) simultaneously, to give the explicit computational method as

$$\begin{bmatrix} y_{n+\frac{1}{8}} \\ y_{n+\frac{1}{4}} \\ y_{n+\frac{3}{8}} \\ y_{n+\frac{1}{2}} \\ y_{n+1} \end{bmatrix} - \begin{bmatrix} 1 & \frac{1}{8} & \frac{1}{128} \\ 1 & \frac{1}{4} & \frac{1}{32} \\ 1 & \frac{3}{8} & \frac{9}{128} \\ 1 & \frac{1}{2} & \frac{1}{8} \\ 1 & 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} y_n \\ hy'_n \\ h^2y''_n \end{bmatrix} = h^3 \begin{bmatrix} -\frac{51529}{1290240} & -\frac{16027}{94080} & -\frac{449}{17920} & -\frac{757}{40320} & \frac{893}{215040} & -\frac{89}{3010560} \\ -\frac{187}{86016} & -\frac{14537}{282240} & \frac{4441}{53760} & \frac{121}{13440} & -\frac{1339}{645120} & \frac{37}{3010560} \\ -\frac{227}{430080} & \frac{183}{31360} & -\frac{43}{32256} & -\frac{103}{13440} & \frac{221}{215040} & -\frac{42}{9031680} \\ \frac{1013}{1290240} & -\frac{403}{94080} & \frac{1357}{17920} & \frac{2267}{40320} & -\frac{149}{43008} & \frac{37}{3010560} \\ -\frac{451}{430080} & \frac{419}{56448} & \frac{2237}{53760} & \frac{2137}{13440} & \frac{27767}{645120} & -\frac{89}{3010560} \\ \frac{207421}{430080} & -\frac{81801}{31360} & \frac{913463}{16280} & -\frac{76711}{13440} & \frac{120601}{43008} & \frac{1046261}{9031680} \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\frac{1}{8}} \\ f_{n+\frac{1}{4}} \\ f_{n+\frac{3}{8}} \\ f_{n+\frac{1}{2}} \\ f_{n+1} \end{bmatrix}, \qquad (14)$$

$$\begin{bmatrix} y'_{n+\frac{1}{8}} \\ y'_{n+\frac{1}{4}} \\ y'_{n+\frac{3}{8}} \\ y'_{n+\frac{1}{2}} \\ y'_{n+1} \end{bmatrix} - \begin{bmatrix} 1 & \frac{1}{8} \\ 1 & \frac{1}{4} \\ 1 & \frac{3}{8} \\ 1 & \frac{1}{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} y'_n \\ hy''_n \end{bmatrix} = h^2 \begin{bmatrix} \frac{20017}{5160960} & \frac{715}{112896} & -\frac{2509}{645120} & \frac{31}{16128} & -\frac{1123}{2580480} & \frac{107}{36126720} \\ \frac{361}{40320} & \frac{461}{17640} & -\frac{29}{4032} & \frac{11}{2520} & -\frac{41}{40320} & \frac{1}{141120} \\ \frac{8007}{573440} & \frac{29790}{62720} & \frac{153}{71680} & \frac{15}{1792} & -\frac{477}{286720} & \frac{9}{802816} \\ \frac{191}{10080} & \frac{152}{2205} & \frac{4}{315} & \frac{8}{315} & -\frac{1}{1008} & \frac{1}{70560} \\ \frac{79}{630} & -\frac{704}{2205} & \frac{344}{315} & -\frac{64}{63} & \frac{191}{315} & \frac{5}{441} \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\frac{1}{8}} \\ f_{n+\frac{1}{4}} \\ f_{n+\frac{3}{8}} \\ f_{n+\frac{1}{2}} \\ f_{n+1} \end{bmatrix}, \qquad (15)$$

$$\begin{bmatrix} y''_{n+\frac{1}{8}} \\ y''_{n+\frac{1}{4}} \\ y''_{n+\frac{3}{8}} \\ y''_{n+\frac{1}{2}} \\ y''_{n+1} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} y''_n \end{bmatrix} = h \begin{bmatrix} \frac{3881}{92160} & \frac{599}{5040} & -\frac{221}{3840} & \frac{1}{36} & -\frac{287}{46080} & \frac{3}{71680} \\ \frac{227}{5760} & \frac{37}{210} & \frac{19}{720} & \frac{1}{90} & -\frac{1}{320} & \frac{1}{40320} \\ \frac{417}{10240} & \frac{93}{560} & \frac{129}{1280} & \frac{3}{40} & -\frac{39}{5120} & \frac{3}{71680} \\ \frac{7}{180} & \frac{8}{45} & \frac{1}{15} & \frac{8}{45} & \frac{7}{180} & 0 \\ \frac{47}{90} & -\frac{256}{105} & \frac{256}{45} & -\frac{256}{45} & \frac{14}{5} & \frac{73}{630} \end{bmatrix} \begin{bmatrix} f_n \\ f_{n+\frac{1}{8}} \\ f_{n+\frac{1}{4}} \\ f_{n+\frac{3}{8}} \\ f_{n+\frac{1}{2}} \\ f_{n+1} \end{bmatrix}. \qquad (16)$$

3. Basic properties of the computational method

The basic properties of the computational method are established in this section.

3.1. Order and error constant

Consider the linear operator L defined by

$$L[y(x):h] = A^{(0)}Y_m^{(i)} - \sum_{i=0}^{k}\frac{jh^{(i)}}{i!}y_n^{(i)} - h^{3-1}[d_i f(y_n) + b_i F(Y_m)]. \tag{17}$$

Applying the Taylor series expansion of Y_m and $F(Y_m)$ and comparing the coefficients of h gives

$$L[y(x):h] = C_0 y(x) + C_1 y'(x) + \cdots + C_p h^p y^{(p)}(x) + C_{p+1}h^{p+1}y^{p+1}(x) + C_{p+2}h^{p+2}y^{p+2}(x) + \ldots . \tag{18}$$

The linear operator L and the associated computational method are said to be of order p if $C_0 = C_1 = \cdots = C_p = C_{p+1} = C_{p+2} = 0$, $C_{p+3} \neq 0$. C_{p+3} is called the error constant and implies that the truncation error is given by $t_{n+k} = C_{p+3}h^{p+3}y^{(p+3)}(x) + 0(h^{p+4})$.

Expanding the computational method in Taylor series gives

$$\left[\begin{array}{c}
\sum_{j=0}^{\infty}\frac{(\frac{1}{8})^j}{j!} - y_n - \frac{1}{8}hy'_n - \frac{1}{128}h^2y''_n - \frac{31849}{165150720}h^3y'''_n - \sum_{j=0}^{\infty}\frac{h^{j+3}}{j!}y_n^{(j+3)}\begin{bmatrix} -\frac{1637}{7225344}(\frac{1}{8}) + \frac{3167}{20643840}(\frac{1}{4}) \\ -\frac{397}{5160960}(\frac{3}{8}) + \frac{1447}{82575360}(\frac{1}{2}) \\ -\frac{139}{1156055040}(1) \end{bmatrix} \\
\sum_{j=0}^{\infty}\frac{(\frac{1}{4})^j}{j!} - y_n - \frac{1}{4}hy'_n - \frac{1}{32}h^2y''_n - \frac{1289}{1290240}h^3y'''_n - \sum_{j=0}^{\infty}\frac{h^{j+3}}{j!}y_n^{(j+3)}\begin{bmatrix} -\frac{11}{5040}(\frac{1}{8}) + \frac{31}{32256}(\frac{1}{4}) \\ -\frac{1}{2016}(\frac{3}{8}) + \frac{73}{645120}(\frac{1}{2}) \\ -\frac{1}{1290240}(1) \end{bmatrix} \\
\sum_{j=0}^{\infty}\frac{(\frac{3}{8})^j}{j!} - y_n - \frac{3}{8}hy'_n - \frac{9}{128}h^2y''_n - \frac{44577}{18350080}h^3y'''_n - \sum_{j=0}^{\infty}\frac{h^{j+3}}{j!}y_n^{(j+3)}\begin{bmatrix} -\frac{27297}{4014080}(\frac{1}{8}) + \frac{3159}{2293760}(\frac{1}{4}) \\ -\frac{99}{81920}(\frac{3}{8}) + \frac{2511}{9175040}(\frac{1}{2}) \\ -\frac{243}{128450560}(1) \end{bmatrix} \\
\sum_{j=0}^{\infty}\frac{(\frac{1}{2})^j}{j!} - y_n - \frac{1}{2}hy'_n - \frac{1}{8}h^2y''_n - \frac{181}{40320}h^3y'''_n - \sum_{j=0}^{\infty}\frac{h^{j+3}}{j!}y_n^{(j+3)}\begin{bmatrix} -\frac{31}{2205}(\frac{1}{8}) + \frac{1}{2520}(\frac{1}{4}) \\ -\frac{1}{315}(\frac{3}{8}) + \frac{1}{2016}(\frac{1}{2}) \\ -\frac{1}{282240}(1) \end{bmatrix} \\
\sum_{j=0}^{\infty}\frac{(1)^j}{j!} - y_n - hy'_n - \frac{1}{2}h^2y''_n - \frac{73}{2520}h^3y'''_n - \sum_{j=0}^{\infty}\frac{h^{j+3}}{j!}y_n^{(j+3)}\begin{bmatrix} -\frac{32}{2205}(\frac{1}{8}) - \frac{44}{315}(\frac{1}{4}) \\ +\frac{32}{315}(\frac{3}{8}) - \frac{53}{630}(\frac{1}{2}) \\ -\frac{17}{17640}(1) \end{bmatrix}
\end{array}\right]. \tag{19}$$

Comparing the coefficients of h, the order p of our method is $p = 5$ and the error constants are given by $C_{p+3} = [-1.1274 \times 10^{-8} \quad -7.3165 \times 10^{-9} \quad -1.0644 \times 10^{-8} \quad -4.0367 \times 10^{-9} \quad -3.2294 \times 10^{-6}]$.

3.2. Consistency

A computational method is said to be consistent if it has order greater than one. Our investigation demonstrates that our method has order $p = 5 > 1$; therefore, the method is consistent [20].

3.3. Zero stability of the computational method

A computational method is said to be zero-stable as $h \to 0$ if the roots of the first characteristic polynomial $\rho(r) = 0$ satisfy $|\sum A^0 R^{k-1}| \leq 1$ and those roots with $R = 1$ must be simple. Now,

$$\rho(r) = \left| r\begin{bmatrix} 1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1 \end{bmatrix} - \begin{bmatrix} 0&0&0&0&1\\0&0&0&0&1\\0&0&0&0&1\\0&0&0&0&1\\0&0&0&0&1 \end{bmatrix} = \begin{bmatrix} r&0&0&0&-1\\0&r&0&0&-1\\0&0&r&0&-1\\0&0&0&r&-1\\0&0&0&0&r-1 \end{bmatrix} = r^4(r-1)\right|.$$

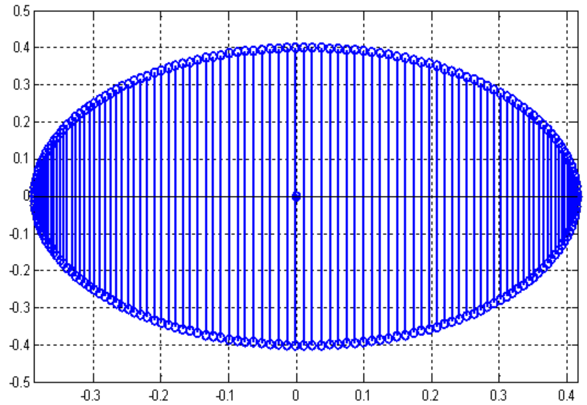

Figure 1: Region of absolute stability of our method.

Then, solving for z in

$$r^4(r-1), \tag{20}$$

gives $r = 0, 0, 0, 0, 1$. Hence, the computational method is zero-stable [20].

3.4. Convergence of the computational method

Theorem: The necessary and sufficient conditions for a linear multistep method to be convergent are consistency and zero-stability. The method developed in this study shows consistency according to the findings of [21].

3.5. Region of absolute stability of the computational method

Applying the boundary locus method, we obtain the stability polynomial as

$$\left.\begin{aligned}\bar{h}(W) = {} & h^{15}\left(-\frac{523}{2597962206\;\;20800}W^4 - \frac{1}{6061057848\;\;1152000}W^5\right) \\ & + h^{12}\left(-\frac{10705061}{2272896693\;\;0432000}W^4 - \frac{8429}{3409345039\;\;5648000}W^5\right) \\ & + h^{9}\left(-\frac{463317839}{2841120866\;\;30400}W^4 - \frac{2311}{52613334937\;\;600}W^5\right) + h^{6}\left(-\frac{7441243}{5284823040}W^4 - \frac{11}{880803840}W^5\right) \\ & + h^{6}\left(-\frac{184343}{860160}W^4 - \frac{27}{28672}W^5\right) + W^5 - \frac{5}{2}W^4\end{aligned}\right\}. \tag{21}$$

Applying the stability polynomial, we obtain the region of absolute stability shown in Figure 1.

4. Numerical implementation of the computational method

The Computational Method (CM) was numerically implemented to estimate its efficiency and validate its accuracy on selected second-kind Volterra integro-differential equations.

4.1. Example 1

Consider the Volterra integro-differential equation of the form

$$y'(g) = -\sin(g) - \cos(g) + \int_0^{\omega} 2\cos(\omega - g)y(g)dg, \quad y(0) = 1, 0 \leq g \leq 5, \tag{22}$$

with the exact solution

$$y(g) = \exp(-g). \tag{23}$$

The newly developed Computational Method (CM) was applied to solve the equation (22). The error associated with the Computational Method, denoted by the Error in Computational Method (ECM), was computed by comparing the numerical solution obtained using CM with the exact solution. The ECM values were then compared with the errors reported for the Fifth-Order Adams–Bashforth–Moulton Predictor–Corrector Method (ABM5) and the Two-Point Three-Step Block Method (P3BM) presented by Majid and Mohamed [22], as well as the Fifth-Order Third Derivative Trigonometrically Fitted Block Method (BTFM) developed by Olowe *et al.* [12]. The comparative results are presented in Table 1.

Table 1: Numerical results for Example 1.

g	Exact Solution	Computed Solution	ECM	BTFM	ABM5	P3BM
0.25	0.77880078307140486825	0.77880078307139900694	5.8613(-15)	1.6847(-06)	8.1337(-03)	6.1138(-03)
0.125	0.88249690258459540286	0.88249690258459539701	5.8500(-18)	5.0991(-08)	4.7616(-04)	3.9009(-04)
0.0625	0.93941306281347578612	0.93941306281347578612	0.00000	1.5744(-09)	2.1034(-05)	1.6881(-05)
0.03125	0.96923323447634408185	0.96923323447634408185	0.00000	4.8948(-11)	7.8509(-07)	7.8509(-07)
0.015625	0.98449643700540840599	0.98449643700540840599	0.00000	1.5261(-12)	2.6828(-08)	2.0516(-08)
0.0078125	0.99221793826024351211	0.99221793826024351211	0.00000	4.7637(-14)	8.7684(-10)	6.6334(-10)

Table 2: Numerical results for Example 2.

g	Exact Solution	Computed Solution	ECM	TM	ETM	NETM
0.025	0.02499739591471233066	0.02499739591471233066	0.0000	5.1916(-06)	5.1818(-06)	6.3815(-09)
0.050	0.04997916927067832880	0.04997916927067832879	1.0000(-20)	1.0247(-05)	1.0208(-05)	5.8198(-08)
0.075	0.07492970727274234209	0.07492970727274234176	3.3000(-19)	1.5012(-05)	1.4926(-05)	2.0634(-07)
0.100	0.09983341664682815231	0.09983341664682814802	4.2900(-18)	1.9339(-05)	1.9187(-05)	4.9983(-07)
0.125	0.12467473338522768996	0.12467473338522765811	3.1850(-17)	2.3086(-05)	2.2852(-05)	9.8478(-07)
0.150	0.14943813247359922150	0.14943813247359905770	1.6380(-16)	2.6120(-05)	2.5790(-05)	1.7034(-06)
0.175	0.17410813759359595189	0.17410813759359529851	6.5338(-16)	2.8323(-05)	2.7883(-05)	2.6932(-06)
0.200	0.19866933079506121546	0.19866933079505905187	2.1636(-15)	2.9588(-05)	2.9027(-05)	3.9860(-06)
0.225	0.22310636213174544782	0.22310636213173923390	6.2139(-15)	2.9825(-05)	2.9136(-05)	5.6073(-06)
0.250	0.24740395925452292960	0.24740395925450698043	1.5949(-14)	2.8961(-05)	2.8139(-05)	7.5758(-06)

Table 3: Numerical results for Example 3.

g	Exact Solution	Computed Solution	ECM	TM	ETM	NETM
0.100	0.09983341664682815231	0.09983341664682814802	4.2900(-18)	8.4317(-05)	8.2792(-05)	1.2502(-07)
0.200	0.19866933079506121546	0.19866933079505905187	2.1636(-15)	1.6558(-04)	1.6310(-04)	7.3015(-07)
0.300	0.29552020666133957511	0.29552020666125835328	8.1222(-14)	2.4402(-04)	2.3848(-04)	2.5669(-06)
0.400	0.38941834230865049167	0.38941834230760478302	1.0457(-12)	3.1661(-04)	3.0657(-04)	6.4301(-06)
0.500	0.47942553860420300027	0.47942553859675189429	7.4511(-11)	3.8012(-04)	3.6514(-04)	1.3074(-05)
0.600	0.56464247339503535720	0.56464247335869400161	3.6341(-11)	4.3328(-04)	4.1210(-04)	2.3195(-05)
0.700	0.64421768723769105367	0.64421768710190933378	1.3578(-10)	4.7376(-04)	4.4558(-04)	3.7422(-05)
0.800	0.71735609089952276163	0.71735609048430228912	4.1522(-10)	4.9978(-04)	4.6392(-04)	5.6296(-05)
0.900	0.78332690962748338846	0.78332690854442458017	1.0831(-09)	5.0976(-04)	4.6571(-04)	8.0264(-05)
1.000	0.84147098480789650665	0.84147098233409403428	2.4738(-09)	5.0243(-04)	4.4985(-04)	1.0966(-04)

4.2. Example 2

Consider the Volterra integro-differential equation of the form

$$\int_0^{\omega} \cos(\omega - g)(y''(g))dg = 2\sin(g). \tag{24}$$

The initial conditions are $y(0) = y'(0) = 0$ with $g \in [0, 1]$ and exact solution given by

$$y(g) = g^2. \tag{25}$$

The results are displayed in Table 2.

4.3. Example 3

Consider the Volterra integro-differential equation of the form

$$y'(g) = 1 - \int_0^{g} y(g)dg,\ y(0) = 0, \quad 0 \le g \le 5, \tag{26}$$

with the exact solution

$$y(g) = \sin(g). \tag{27}$$

The results for $h = 0.025$ and $h = 0.1$ are displayed in Table 3.

5. Discussion of results

In Example 1, the CM was used to solve a test problem with a known exact solution. The results were compared with those from ABM5, P3BM [22], and BTFM [12]. The CM demonstrates excellent agreement with the exact solution throughout all tested step sizes, achieving results that match or exceed the performance of other methods. Notably, the CM errors are on the order of 10^{-15} to 10^{-18}, while the comparison methods show errors ranging from 10^{-6} to 10^{-14}. The CM consistently produces more accurate results. In Example 2, the CM was evaluated against the Haar Wavelet Method by Singh and Kumar [23] and BTFM [12]. The CM successfully reproduces the exact solution for all grid points considered, matching the performance of HWM and exceeding BTFM in some cases. The results indicate that the CM can handle problems with rapidly increasing solution values effectively. Example 3 compared CM against TM, ETM as described by Ishak and Ahmad [24], and NETM developed by Fuziyah and Muhammad [25]. The numerical results show that the CM provides highly accurate approximations at various step sizes, performing significantly better than TM and ETM while remaining competitive with NETM. The results show that the CM maintains its accuracy through different step sizes, demonstrating both convergence and stability properties. The Computational Method (CM) demonstrates its effectiveness, reliability, and high accuracy through numerical tests in three examples of second-kind Volterra integro-differential equations. The method performs consistently across various problem types, and its performance exceeds that of many existing numerical methods, establishing it as an effective tool for real-world computational tasks.

6. Conclusion

This study presents a newly developed Computational Method (CM) for solving second-kind Volterra integro-differential equations. The CM was developed through interpolation and collocation techniques and was tested to evaluate its performance through several properties, including order, consistency, zero-stability, convergence, and absolute stability. The Computational Method (CM) functions as a dependable numerical solution tool that delivers accurate results for second-kind Volterra integro-differential equations. The solution method shows improved performance compared with many existing numerical methods while maintaining both stability and convergence when solving problems with rapid solution changes or large solution values. The study establishes CM as a viable option for practical applications in engineering, physics, and applied mathematics where precise solutions of integro-differential equations are required. Future research will apply this method to develop solutions for nonlinear fractional and multi-dimensional Volterra-type problems by testing adaptive step-size methods and large-scale computational optimization.

Data availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declaration of competing interest

The authors declare that they have no known competing interests or personal relationships that could have appeared to influence the work reported in this manuscript.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgment

The authors thank the Department of Mathematics, University of Abuja, for providing the necessary support and resources for this research. The authors also acknowledge the valuable contributions of colleagues who reviewed this manuscript.

References

[1] A. Adu, N. C. Ine & C. A. John, "Three-step three off-grid points for solving Volterra integral equation of the second kind", Asian Journal of Pure and Applied Mathematics **5** (2023) 512. https://jofmath.com/index.php/AJPAM/article/view/4.

[2] J. C. Ahmadu, F. Ogunfiditimi & I. Sale, "Derivation of continuous linear multistep hybrid block method for the solution of Volterra integral equation of second kind", Asian Basic and Applied Research Journal **6** (2024) 59. https://jofresearch.com/index.php/ABAARJ/article/view/140.

[3] O. O. Aduroja, L. Adiku & O. Agbolade, "Collocation approximation method for the solution of Volterra integro-differential equations", FUOYE Journal of Management, Innovation and Entrepreneurship **2** (2023) 240. https://fjmie.fuoye.edu.ng/index.php/public_html/article/view/53.

[4] G. Agarwala, M. Mohana, A. M. Menon, A. Sharma, T. C. Dakal & S. D. Purohit, "Analysis of the Adomian decomposition method to estimate the COVID-19 pandemic", in Methods of Mathematical Modeling, H. Singh, H. M. Srivastava, and D. Baleanu (Eds.), Academic Press, 2022, pp. 173–187. https://doi.org/10.1016/B978-0-323-99888-8.00011-5.

[5] B. C. Agbata, B. N. Ani, M. M. Shior, I. G. Ezugorie, R. V. Paul & P. K. Meseda, "Analysis of Adomian decomposition method and its application for solving linear and nonlinear differential equations", Asian Research Journal of Mathematics **18** (2022) 56. https://www.researchgate.net/publication/359326653_Analysis_of_Adomian_Decomposition_Method_and_Its_Application_for_Solving_Linear_and_Nonlinear_Differential_Equations.
[6] H. Brunner, Volterra Integral Equations: An Introduction to Theory and Applications, Cambridge University Press, Cambridge, United Kingdom, 2017. https://doi.org/10.1017/9781107341661.
[7] A. Ajileye, A. O. Olamipos, I. A. Grace & A. M. Ajiley, "Numerical method for the solution of linear Volterra integral equations", International Journal of Development Mathematics **1** (2024) 37. https://doi.org/10.62054/ijdm/0101.04.
[8] G. Ajileye & S. A. Amoo, "Numerical solution to Volterra integro-differential equations using collocation approximation", Mathematics and Computational Sciences **4** (2023) 1. https://www.researchgate.net/publication/369295547_Numerical_solution_to_Volterra_integro-differential_equations_using_collocation_approximation.
[9] N. M. Kamoh, T. Aboiyar & E. S. Onah, "On investigating some quadrature rules for the solution of second-order Volterra integro-differential equations", IOSR Journal of Mathematics **13** (2017) 45. https://iosrjournals.org/iosr-jm/papers/Vol13-issue5/Version-3/H1305034550.pdf.
[10] E. Cimen & S. Yatar, "Numerical solution of Volterra integro-differential equation with delay", Journal of Mathematics and Computer Science **20** (2020) 255. https://doi.org/10.22436/jmcs.020.03.08.
[11] D. U. Islam & A. Srivastava, "Comparative analysis of Adomian decomposition method and variational iteration method for solving nonlinear integral equations", JuniKhyat, UGC Care Group I Listed Journal **14** (2024) 129. https://www.researchgate.net/publication/386336425_COMPARATIVE_ANALYSIS_OF_ADOMIAN_DECOMPOSITION_METHOD_AND_VARIATIONAL_ITERATION_METHOD_FOR_SOLVING_NONLINEAR_INTEGRAL_EQUATIONS.
[12] R. A. Olowe, O. A. Akinfenwa, R. I. Abdulganiy & A. Okunuga, "A trigonometrically fitted method for solving Volterra integro-differential equations", Journal of the Nigerian Mathematical Society **43** (2024) 203. https://www.researchgate.net/publication/381742554_A_TRIGONOMETRICALLY_FITTED_METHOD_FOR_SOLVING_VOLTERRA_INTEGRO-DIFFERENTIAL_EQUATIONS.
[13] D. A. Maturi, "The variational iteration method for solving Volterra integro-differential equation using Maple", Applied Mathematical Sciences **13** (2019) 897. https://doi.org/10.12988/ams.2019.98120.
[14] O. A. Uwaheren, A. F. Adebisi, C. Y. Ishola, M. T. Raji, A. O. Yekeen & O. J. Peter, "Numerical solution of Volterra integro-differential equations by Akbari-Ganji's method", BAREKENG: Journal of Mathematics and Its Applications **16** (2022) 1123. https://www.researchgate.net/publication/364151036_NUMERICAL_SOLUTION_OF_VOLTERRA_INTEGRO-DIFFERENTIAL_EQUATIONS_BY_AKBARI-GANJI'S_METHOD.
[15] F. M. Alharbi & S. S. Althubiti, "Numerical simulation of initial value problem of integro-differential equation models", European Journal of Pure and Applied Mathematics **17** (2024) 286. https://doi.org/10.29020/nybg.ejpam.v17i1.5024.
[16] B. Ghosh & J. Mohapatra, "Numerical solution of time-fractional Volterra integro-differential equations", Miskolc Mathematical Notes **26** (2025) 395. https://doi.org/10.18514/MMN.2025.4723.
[17] I. J. Otaide & M. O. Oluwayemi, "Numerical treatment of linear Volterra integro-differential equations using variational iteration algorithm with collocation", Partial Differential Equations in Applied Mathematics **10** (2024) 100693. https://doi.org/10.1016/j.padiff.2024.100693.
[18] C. Ouaret & Z. Malendas, Numerical solution of some Volterra nonlinear integral equations by using spline iterative collocation method, Master's thesis, Department of Mathematics, Centre Universitaire Abdelhafid Boussouf Mila, Mila, Algeria, 2023. https://theses-algerie.com/2452229097228842/memoire-de-master/centre-universitaire-abdel-hafid-boussouf---mila/www.centre-univ-mila.dz/?lang=en.
[19] Y. Zhu, T. Dong, X. Zheng, Y. Li, S. Jia & X. Guo, "Numerical approximation and simulation of a Volterra integro-differential equation with a peridynamic differential operator", Zeitschrift für Angewandte Mathematik und Mechanik **105** (2025) e202400619. https://doi.org/10.1002/zamm.202400619.
[20] J. Sabo, A. M. Ayinde, T. Oyedepo & A. A. Ishaq, "Simulation of two-step block approach for solving oscillatory differential equations", Cumhuriyet Science Journal **45** (2024) 366. https://doi.org/10.17776/csj.1345303.
[21] P. Tumba, Y. Skwame & D. Raymond, "Half-step implicit linear hybrid block approach of order four for solving third-order ordinary differential equations", Dutse Journal of Pure and Applied Sciences **7** (2021) 124. https://globaljournals.org/GJSFR_Volume21/5-Half-Step-Implicit-Linear.pdf.
[22] Z. A. Majid & N. A. Mohamed, "Fifth-order multistep block method for solving Volterra integro-differential equations of second kind", Sains Malaysiana **48** (2019) 677. https://doi.org/10.17576/jsm-2019-4803-22.
[23] I. Singh & S. Kumar, "Haar wavelet method for some nonlinear Volterra integral equations of the first kind", Journal of Computational and Applied Mathematics **292** (2016) 541. https://doi.org/10.1016/j.cam.2015.07.022.
[24] F. Ishak & S. N. Ahmad, "Development of extended trapezoidal method for numerical solution of Volterra integro-differential equations", International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering **10** (2016) 579. https://doi.org/10.5281/zenodo.1127402.
[25] I. Fuziyah & N. F. S. Muhammad, "New development of extended trapezoidal method for solving first-order linear Volterra integro-differential equations", ASM Science Journal **13** (2020) 1. https://doi.org/10.32802/asmscj.2020.sm26(5.24).