Malaysian Journal of Mathematical Sciences 16(2): 237–256 (2022)
https://doi.org/10.47836/mjms.16.2.05

Malaysian Journal of Mathematical Sciences

Journal homepage: https://einspem.upm.edu.my/journal

Boole's Strategy in Multistep Block Method for Volterra Integro-Differential Equation

Baharum, N. A.[1], Majid, Z. A.*[1,2], and Senu, N.[1,2]

[1]*Institute for Mathematical Research, Universiti Putra Malaysia, Malaysia*
[2]*Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, Malaysia*

E-mail: aunibaharum@gmail.com
**am_zana@upm.edu.my*
**Corresponding author*

Received: 30 October 2020
Accepted: 16 February 2022

Abstract

This article presents a numerical approach for solving the second kind of Volterra integro- differential equation (VIDE). The multistep block-Boole's rule method will estimate the solutions for the linear and nonlinear problems of VIDE. The method computes two solutions for VIDE along the interval. The proposed method is developed by derivation of the Lagrange interpolating polynomial. The convergence and stability analysis of the derived method are discussed. From the perspective of total function calls and time-saving, the computation results explained that the derived method performs better than other existing methods

Keywords: diagonally implicit; block method; Boole's rule; Volterra integro-differential equation.

1 Introduction

The general form of the Volterra integro-differential equation will be discussed

$$y'(x) = F(x, y(x), z(x)), \tag{1}$$

where

$$z(x) = \int_0^x K(x, s, y(s))\, ds, \quad y(x_0) = y_0. \tag{2}$$

Volterra integro-differential equation occurs in several fields of science and engineering, including chemical kinetics and fluid dynamics. Several decades ago, many researchers implemented a variety of numerical methods to address VIDE issues.

Several authors have proposed the numerical solution for VIDE with numerical integration approach. Day [5] proposed the trapezoidal rule and composite trapezoidal to find the solution of the integro-differential equation. Meanwhile, Linz [12] employed Simpson's rule to solve the equation (1). Ishak and Ahmad [9] applied the extended trapezoidal method for solving the first order VIDE and implemented it in the PECE scheme. Instead, Brunner and Lambert [2] established a weak stability theory for VIDE. They introduced the general linear test equation of VIDE. Chang and Day [3] studied the nonlinear VIDE properties and established the theorem of existence and uniqueness. An implicit Runge-Kutta approach is suggested by Yuan and Tang [19] for solving a nonlinear integro-differential equation. Thus, Tang [16] introduced the spline collocation methods to solve VIDE in weakly singular kernels.

Filiz [8] proposed Runge-Kutta method for solving VIDE and applied the Newton-Cotes formulae to solve the integrals term of VIDE. The author implemented the trapezoidal rule, Simpson's $1/3$ rule, Simpson's $3/8$ and Boole's rule for the integral term. In 2015, some researchers were attracted by the block method to apply this method in VIDE. Mohamed and Majid [14] proposed the two points one-step block method for solving VIDE. While, Tunç and Tunç [17] introduced the sufficient conditions for stability, boundedness, uniformly asymptotic stability, integrability and square integrability of solutions of a few scalar nonlinear Volterra integrodifferential equations and a Volterra integro-differential system. Their approach is based on Lyapunov's second method. Janodi et al. [10] developed a one-step hybrid block method with quadrature rules for solving linear and nonlinear problems in VIDE. The authors discussed the stability analysis of the hybrid one step method, including the method's order, consistency, zero stability, and stability region of the method.

The fifth order numerical method and an appropriate numerical integration method for solving equation (1) and (2) are discussed in this paper. We demonstrate how to adapt this strategy to the solution of both linear and nonlinear problems of VIDE.

2 Derivation of Diagonally Multistep Block

Figure 1 describes the approximate solutions for y_{n+1} and y_{n+2} in a block of three back values, respectively, at x_{n+1} and x_{n+2}.

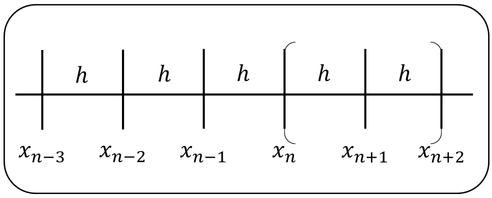

Figure 1: Diagonally multistep block method.

The formulae of the multistep block method was derived from the Lagrange interpolating polynomial. For the first point corrector formula, y_{n+1} can be produced by integrating once over $[x_n, x_{n+1}]$ and yield to

$$\begin{aligned}\int_{x_n}^{x_{n+1}} y'(x)\,dx &= \int_{x_n}^{x_{n+1}} F(x, y(x), z(x))\,dx,\\ y(x_{n+1}) - y(x_n) &= \int_{x_n}^{x_{n+1}} F(x, y(x), z(x))\,dx. \end{aligned} \tag{3}$$

Thus, the degree four Lagrange interpolating polynomial will be associated with $F(x, y(x), z(x))$. The set of interpolation points used to acquire the first point of the corrector formula, y_{n+1}, is $\{x_{n-3}, x_{n-2}, x_{n-1}, x_n, x_{n+1}\}$. It takes $x = x_{n+1} + sh$ and gets the $dx = hds$ substitute. The integration limit from -1 to 0 will be fixed. The first correcter formulation point can be written by simplifying the integration using **MAPLE** software as follows,
First point of corrector formula:

$$y_{n+1} = y_n + h\left[-\frac{19}{720}F_{n-3} + \frac{53}{360}F_{n-2} - \frac{11}{30}F_{n-1} + \frac{323}{360}F_n + \frac{251}{720}F_{n+1}\right]. \tag{4}$$

The same procedure will be applied to develop the second point of the corrector formula. Equation (1) will be integrated over $[x_n, x_{n+2}]$ once gives,

$$\begin{aligned}\int_{x_n}^{x_{n+2}} y'(x)\,dx &= \int_{x_n}^{x_{n+2}} F(x, y(x), z(x))\,dx,\\ y(x_{n+2}) - y(x_n) &= \int_{x_n}^{x_{n+2}} F(x, y(x), z(x))\,dx. \end{aligned} \tag{5}$$

For the second point of the corrector formula, five points will be considered in the Lagrange interpolating polynomial i.e. $\{x_{n-2}, x_{n-1}, x_n, x_{n+1}, x_{n+2}\}$ and then substitute $F(x, y(x), z(x))$ with the polynomial. Therefore, take $dx = hds$ and fix the range of the integration limit from -2 to 0.

The **MAPLE** software is applied to evaluate the integrals for the implementation of the corrector formula for the second point of the multistep block method as follows;
Second point of corrector formula:

$$y_{n+2} = y_n + h\left[-\frac{1}{90}F_{n-2} + \frac{2}{45}F_{n-1} + \frac{4}{15}F_n + \frac{62}{45}F_{n+1} + \frac{29}{90}F_{n+2}\right]. \tag{6}$$

The proposed method is a combination of predictor and corrector formulae. When four points are involved in the Lagrange interpolating polynomial, the predictor formulae will be established. The formulation of the predictor formulae of the first point and second point can be obtained from the same process as described above and gives the predictor formulae as follows;

$$y_{n+1} = y_n + h\left[-\frac{3}{8}F_{n-3} + \frac{37}{24}F_{n-2} - \frac{59}{24}F_{n-1} + \frac{55}{24}F_n\right], \tag{7}$$

$$y_{n+2} = y_n + h\left[-\frac{8}{3}F_{n-3} + \frac{31}{3}F_{n-2} - \frac{44}{3}F_{n-1} + 9F_n\right]. \tag{8}$$

The following corrector formulae can be expressed into a matrix form, which is equivalent to the equation (4) and (6)

$$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} y_{n-3} \\ y_{n-2} \\ y_{n-1} \\ y_n \\ y_{n+1} \\ y_{n+2} \end{bmatrix} = h\begin{bmatrix} -\frac{19}{720} & \frac{53}{360} & -\frac{11}{30} & \frac{323}{360} & \frac{251}{720} & 0 \\ 0 & -\frac{1}{90} & \frac{2}{45} & \frac{4}{15} & \frac{62}{45} & \frac{29}{90} \end{bmatrix}\begin{bmatrix} F_{n-3} \\ F_{n-2} \\ F_{n-1} \\ F_n \\ F_{n+1} \\ F_{n+2} \end{bmatrix}. \tag{9}$$

Regarding [11], the order of the method can be discovered

$$\sum_{j=0}^{k}[\alpha_j y(x+jh) - h\beta_j y'(x+jh)] = C_p y^p + O(h^{p+1}), \tag{10}$$

where p is the order of the linear multistep method, $O(h^{p+1})$ is the local truncation error and C_p is developed as follows;

$$C_p = \sum_{j=0}^{k}\left(\frac{j^p\alpha_j}{p!} - \frac{j^{(p-1)}\beta_j}{(p-1)!}\right). \tag{11}$$

Definition 2.1. *The numerical method is said to be in order p if,*

$$C_0 = C_1 = C_2 = \cdots = C_p = 0, \qquad C_{p+1} \neq 0, \tag{12}$$

where the error constant for the method is called as C_{p+1}, [11].

The coefficients of α_j and β_j can be obtained from (9),

$$\alpha_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \alpha_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \alpha_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \alpha_3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \alpha_4 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \alpha_5 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$\beta_0 = \begin{bmatrix} -\frac{19}{720} \\ 0 \end{bmatrix}, \quad \beta_1 = \begin{bmatrix} \frac{53}{360} \\ -\frac{1}{90} \end{bmatrix}, \quad \beta_2 = \begin{bmatrix} -\frac{11}{30} \\ \frac{2}{45} \end{bmatrix}, \quad \beta_3 = \begin{bmatrix} \frac{323}{360} \\ \frac{4}{15} \end{bmatrix}, \quad \beta_4 = \begin{bmatrix} \frac{251}{720} \\ \frac{62}{45} \end{bmatrix}, \quad \beta_5 = \begin{bmatrix} 0 \\ \frac{29}{90} \end{bmatrix}.$$

Therefore, by equation (11), the order and error constant of the derived method will be generated as follows,

$$
\begin{aligned}
C_0 &= \sum_{j=0}^{5} \frac{1}{0!} \cdot j^0 \cdot \alpha_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_1 &= \sum_{j=0}^{5} \frac{1}{1!} \cdot j^1 \cdot \alpha_j - \sum_{j=0}^{5} \frac{1}{0!} \cdot j^0 \cdot \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_2 &= \sum_{j=0}^{5} \frac{1}{2!} \cdot j^2 \cdot \alpha_j - \sum_{j=0}^{5} \frac{1}{0!} \cdot j^1 \cdot \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_3 &= \sum_{j=0}^{5} \frac{1}{3!} \cdot j^3 \cdot \alpha_j - \sum_{j=0}^{5} \frac{1}{2!} \cdot j^2 \cdot \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_4 &= \sum_{j=0}^{5} \frac{1}{4!} \cdot j^4 \cdot \alpha_j - \sum_{j=0}^{5} \frac{1}{3!} \cdot j^3 \cdot \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_5 &= \sum_{j=0}^{5} \frac{1}{5!} \cdot j^5 \cdot \alpha_j - \sum_{j=0}^{5} \frac{1}{4!} \cdot j^4 \cdot \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\
C_6 &= \sum_{j=0}^{5} \frac{1}{6!} \cdot j^6 \cdot \alpha_j - \sum_{j=0}^{5} \frac{1}{5!} \cdot j^5 \cdot \beta_j = \begin{bmatrix} -\frac{3}{160} \\ -\frac{1}{90} \end{bmatrix}.
\end{aligned}
$$

The derived method is of order five concerning Definition 2.1 and the error constant is assessed.

$$
C_{p+1} = \quad C_6 \quad = \begin{bmatrix} -\dfrac{3}{160} & -\dfrac{1}{90} \end{bmatrix} \neq \; [0 \quad 0]^T. \tag{13}
$$

Definition 2.2. *If the order of method is $p \geq 1$, then the numerical method is consistent. The numerical method will be consistent if and only if,* [11]*,*

$$
\sum_{j=0}^{k} \alpha_j = 0, \qquad \sum_{j=0}^{k} j\alpha_j = \sum_{j=0}^{k} \beta_j. \tag{14}
$$

By referring to Definition 2.2, the consistency of the method can be proved. The proposed method (9) is consistent if and only if it satisfies with two conditions as follows,

1. $\sum\limits_{j=0}^{k} \alpha_j = 0.$

   Proof:

$$
\begin{aligned}
\sum_{j=0}^{5} \alpha_j &= \sum_{j=0}^{5} \frac{1}{0!} \cdot j^0 \cdot \alpha_j = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5, \\
&= \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.
\end{aligned} \tag{15}
$$

2. $\sum_{j=0}^{k} j \cdot \alpha_j = \sum_{j=0}^{k} \beta_j$.

Proof:

$$\begin{aligned}
\sum_{j=0}^{5} j \cdot \alpha_j &= \sum_{j=0}^{5} \frac{1}{1!} \cdot j^1 \cdot \alpha_j = 3^1 \cdot \alpha_3 + 4^1 \cdot \alpha_4 + 5^1 \cdot \alpha_5, \\
&= 3 \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 5 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}. \\
\sum_{j=0}^{5} \beta_j &= \sum_{j=0}^{5} \frac{1}{0!} \cdot j^0 \cdot \beta_j = \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5, \\
&= \begin{bmatrix} -\frac{19}{720} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{53}{360} \\ -\frac{1}{90} \end{bmatrix} + \begin{bmatrix} -\frac{11}{30} \\ \frac{2}{45} \end{bmatrix} + \begin{bmatrix} \frac{323}{360} \\ \frac{4}{15} \end{bmatrix} + \begin{bmatrix} \frac{251}{720} \\ \frac{62}{45} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{29}{90} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.
\end{aligned}$$

Hence,

$$\sum_{j=0}^{5} j \cdot \alpha_j = \sum_{j=0}^{5} \beta_j = \begin{bmatrix} 1 \\ 2 \end{bmatrix}. \tag{16}$$

The equation (9) is also equivalent to

$$A_0 Y_m = A_1 Y_{m-1} + h(B_0 F_m + B_1 F_{m-1} + B_2 F_{m-2}),$$

where

$$\begin{aligned}
\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} &= \begin{bmatrix} 0 & -1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} + h \begin{bmatrix} -\frac{19}{720} & \frac{53}{360} \\ 0 & -\frac{1}{90} \end{bmatrix} \begin{bmatrix} F_{n-3} \\ F_{n-2} \end{bmatrix} \\
&\quad + h \begin{bmatrix} -\frac{11}{30} & \frac{323}{360} \\ \frac{2}{45} & \frac{4}{15} \end{bmatrix} \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix} + h \begin{bmatrix} \frac{251}{720} & 0 \\ \frac{62}{45} & \frac{29}{90} \end{bmatrix} \begin{bmatrix} F_{n+1} \\ F_{n+2} \end{bmatrix}.
\end{aligned} \tag{17}$$

Definition 2.3. *A block method is said to be zero-stable if and only if providing the roots of R_j, $j = 1(1)k$ of the first characteristic polynomial, $\rho(R)$ specified as*

$$\rho(R) = \det \left[\sum_{j=0}^{k} A^{(i)} R^{(k-i)} \right] = 0, \tag{18}$$

satisfies with $|R_j| \leq 1$ and those roots with $|R_j| = 1$, [11].

Since the roots of the method are $| R | \leq 1$, thus the associated method is zero stable concerning Definition 2.3. The verification of the zero stability of the derived method (9) is validated as below,

$$\begin{aligned}
\rho(r) &= \det [RA_0 - A_1] = 0, \\
&= \det \left[R \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 0 & -1 \end{bmatrix} \right], \\
&= \det \begin{bmatrix} R & 1 \\ 0 & R+1 \end{bmatrix}, \\
&= R(R+1).
\end{aligned} \tag{19}$$

Theorem 2.1. *The method is said to be convergent if it is consistent and zero-stable, [11].*

Corollary 2.1. *The method of order five is consistent and zero stable, and then it converges on the theorem 2.1.*

3 Implementation

The derived method requires three initial values before being used. The three initial values were computed using the Adam Bashforth Moulton method. Then, the method derived can be implemented until the end of the interval;

$$\begin{aligned}
y^p_{n+1} &= y_n + h\left[-\frac{3}{8}F_{n-3} + \frac{37}{24}F_{n-2} - \frac{59}{24}F_{n-1} + \frac{55}{24}F_n\right],\\
y^p_{n+2} &= y_n + h\left[-\frac{8}{3}F_{n-3} + \frac{31}{3}F_{n-2} - \frac{44}{3}F_{n-1} + 9F_n\right],\\
y^c_{n+1} &= y_n + h\left[-\frac{19}{720}F_{n-3} + \frac{53}{360}F_{n-2} - \frac{11}{30}F_{n-1} + \frac{323}{360}F_n + \frac{251}{720}F_{n+1}\right],\\
y^c_{n+2} &= y_n + h\left[-\frac{1}{90}F_{n-2} + \frac{2}{45}F_{n-1} + \frac{4}{15}F_n + \frac{62}{45}F_{n+1} + \frac{29}{90}F_{n+2}\right]. \qquad (20)
\end{aligned}$$

The appropriate numerical integration method for finding solutions for the integral part of VIDE is adapted to the derived method. For the solution of the integral part, the composite Boole's rule (refer [18]) is implemented . This approach will be applied to the proposed method when $n = 0, 4, 8, 16, \ldots$,

$$z_{n+4} = \frac{2h}{45}\sum_{i=0}^{n+4}\omega^s_i K(x_{n+4}, x_i, y_i), \qquad (21)$$

$$\begin{aligned}
z_{n+5} &= \frac{2h}{45}\sum_{i=0}^{n+4}\omega^s_i K(x_{n+5}, x_i, y_i)\\
&\quad + \frac{h}{90}\Big[7K(x_{n+5}, x_{n+4}, y_{n+4}) + 32K(x_{n+5}, x_{n+\frac{17}{4}}, y_{n+\frac{17}{4}})\\
&\quad + 12K(x_{n+5}, x_{n+\frac{9}{2}}, y_{n+\frac{9}{2}}) + 32K(x_{n+5}, x_{n+\frac{19}{4}}, y_{n+\frac{19}{4}})\\
&\quad + 7K(x_{n+5}, x_{n+5}, y_{n+5})\Big]. \qquad (22)
\end{aligned}$$

where ω^s_i are Boole's rule weights $7, 32, 12, 32, 14, 32, 12, 32, 14, \ldots, 32, 12, 32, 7$. The equations of $y_{n+\frac{17}{4}}$, $y_{n+\frac{9}{2}}$ and $y_{n+\frac{19}{4}}$ can be obtained by Lagrange interpolating polynomial at points $\{x_n, x_{n+1}, x_{n+2}, x_{n+3}, x_{n+4}, x_{n+5}\}$,

$$\begin{aligned}
y_{n+\frac{17}{4}} &= \frac{117}{8192}y_n - \frac{765}{8192}y_{n+1} + \frac{1105}{4096}y_{n+2} - \frac{1989}{4096}y_{n+3} + \frac{9945}{8192}y_{n+4} + \frac{663}{8192}y_{n+5},\\
y_{n+\frac{9}{2}} &= \frac{7}{256}y_n - \frac{45}{256}y_{n+1} + \frac{63}{128}y_{n+2} - \frac{105}{128}y_{n+3} + \frac{315}{256}y_{n+4} + \frac{63}{256}y_{n+5},\\
y_{n+\frac{19}{4}} &= \frac{231}{8192}y_n - \frac{1463}{8192}y_{n+1} + \frac{1995}{4096}y_{n+2} - \frac{3135}{4096}y_{n+3} + \frac{7315}{8192}y_{n+4} + \frac{4389}{8192}y_{n+5}. \qquad (23)
\end{aligned}$$

Since $n = 0, 4, 8, 16, \ldots$, another two values of z are required to satisfy the requirement of n. As below are derived, two other formulae of z.

$$\begin{aligned} z_{n+6} &= \frac{2h}{45}\sum_{i=0}^{n+4}\omega_i^s K(x_{n+6}, x_i, y_i) \\ &+\frac{h}{90}\Big[7K(x_{n+6}, x_{n+4}, y_{n+4}) + 32K(x_{n+6}, x_{n+\frac{17}{4}}, y_{n+\frac{17}{4}}) \\ &\quad +12K(x_{n+6}, x_{n+\frac{9}{2}}, y_{n+\frac{9}{2}}) + 32K(x_{n+6}, x_{n+\frac{19}{4}}, y_{n+\frac{19}{4}}) \\ &\quad +7K(x_{n+6}, x_{n+5}, y_{n+5})\Big] \\ &+\frac{h}{90}\Big[7K(x_{n+6}, x_{n+5}, y_{n+5}) + 32K(x_{n+6}, x_{n+\frac{21}{4}}, y_{n+\frac{21}{4}}) \\ &\quad +12K(x_{n+6}, x_{n+\frac{11}{2}}, y_{n+\frac{11}{2}}) + 32K(x_{n+6}, x_{n+\frac{23}{4}}, y_{n+\frac{23}{4}}) \\ &\quad +7K(x_{n+6}, x_{n+6}, y_{n+6})\Big], \end{aligned} \tag{24}$$

$$\begin{aligned} z_{n+7} &= \frac{2h}{45}\sum_{i=0}^{n+4}\omega_i^s K(x_{n+7}, x_i, y_i) \\ &+\frac{h}{90}\Big[7K(x_{n+7}, x_{n+4}, y_{n+4}) + 32K(x_{n+7}, x_{n+\frac{17}{4}}, y_{n+\frac{17}{4}}) \\ &\quad +12K(x_{n+7}, x_{n+\frac{9}{2}}, y_{n+\frac{9}{2}}) + 32K(x_{n+7}, x_{n+\frac{19}{4}}, y_{n+\frac{19}{4}}) \\ &\quad +7K(x_{n+7}, x_{n+5}, y_{n+5})\Big] \\ &+\frac{h}{90}\Big[7K(x_{n+7}, x_{n+5}, y_{n+5}) + 32K(x_{n+7}, x_{n+\frac{21}{4}}, y_{n+\frac{21}{4}}) \\ &\quad +12K(x_{n+7}, x_{n+\frac{11}{2}}, y_{n+\frac{11}{2}}) + 32K(x_{n+7}, x_{n+\frac{23}{4}}, y_{n+\frac{23}{4}}) \\ &\quad +7K(x_{n+7}, x_{n+6}, y_{n+6})\Big] \\ &+\frac{h}{90}\Big[7K(x_{n+7}, x_{n+6}, y_{n+6}) + 32K(x_{n+7}, x_{n+\frac{25}{4}}, y_{n+\frac{25}{4}}) \\ &\quad +12K(x_{n+7}, x_{n+\frac{13}{2}}, y_{n+\frac{13}{2}}) + 32K(x_{n+7}, x_{n+\frac{27}{4}}, y_{n+\frac{27}{4}}) \\ &\quad +7K(x_{n+7}, x_{n+7}, y_{n+7})\Big]. \end{aligned} \tag{25}$$

The unknown values of $y_{n+\frac{17}{4}}, y_{n+\frac{9}{2}}$ and $y_{n+\frac{19}{4}}$ can be solved by equation (23). Lagrange interpolating polynomial at a set of points $\{x_{n+1}, x_{n+2}, x_{n+3}, x_{n+4}, x_{n+5}, x_{n+6}\}$ are used to develop new equations for $y_{n+\frac{21}{4}}, y_{n+\frac{11}{2}}$ and $y_{n+\frac{23}{4}}$;

$$\begin{aligned} y_{n+\frac{21}{4}} &= \frac{117}{8192}y_{n+1} - \frac{765}{8192}y_{n+2} + \frac{1105}{4096}y_{n+3} - \frac{1989}{4096}y_{n+4} + \frac{9945}{8192}y_{n+5} + \frac{663}{8192}y_{n+6}, \\ y_{n+\frac{11}{2}} &= \frac{7}{256}y_{n+1} - \frac{45}{256}y_{n+2} + \frac{63}{128}y_{n+3} - \frac{105}{128}y_{n+4} + \frac{315}{256}y_{n+5} + \frac{63}{256}y_{n+6}, \\ y_{n+\frac{23}{4}} &= \frac{231}{8192}y_{n+1} - \frac{1463}{8192}y_{n+2} + \frac{1995}{4096}y_{n+3} - \frac{3135}{4096}y_{n+4} + \frac{7315}{8192}y_{n+5} + \frac{4389}{8192}y_{n+6}. \end{aligned} \tag{26}$$

The values of $y_{n+\frac{25}{4}}$, $y_{n+\frac{13}{2}}$ and $y_{n+\frac{27}{4}}$ will be obtained by Lagrange interpolating polynomial. The interpolating points involved are $\{x_{n+2}, x_{n+3}, x_{n+4}, x_{n+5}, x_{n+6}, x_{n+7}\}$ and the developed equations can be expressed as,

$$\begin{aligned}
y_{n+\frac{25}{4}} &= \frac{117}{8192}y_{n+2} - \frac{765}{8192}y_{n+3} + \frac{1105}{4096}y_{n+4} - \frac{1989}{4096}y_{n+5} + \frac{9945}{8192}y_{n+6} + \frac{663}{8192}y_{n+7}, \\
y_{n+\frac{13}{2}} &= \frac{7}{256}y_{n+2} - \frac{45}{256}y_{n+3} + \frac{63}{128}y_{n+4} - \frac{105}{128}y_{n+5} + \frac{315}{256}y_{n+6} + \frac{63}{256}y_{n+7}, \\
y_{n+\frac{27}{4}} &= \frac{231}{8192}y_{n+2} - \frac{1463}{8192}y_{n+3} + \frac{1995}{4096}y_{n+4} - \frac{3135}{4096}y_{n+5} + \frac{7315}{8192}y_{n+6} + \frac{4389}{8192}y_{n+7}.
\end{aligned} \tag{27}$$

4 Stability Analysis

The stability of the derived method on Volterra integro-differential equation will be discussed. The linear test equation of VIDE is considered as,

$$y' = \xi y(x) + \eta \int_0^x y(s)\, ds, \tag{28}$$

where ξ, η are real constants. The solutions of (28) tend to zero as $x \to \infty$ if and only if $\xi < 0$ and $\eta < 0$. Then, the region of absolute stability is the set of points $(h\xi, h^2\eta)$ for which all zeros of the stability polynomial lie in the interior of the unit disk.

The stability polynomial is developed using the derived method associated, including the quadrature rule. The stability polynomial of the generated methods is determined after substituting the first characteristics polynomial into the general form of VIDE stability polynomial. The general form of stability polynomial for VIDE, [2] is considered as,

$$\pi(r, h\xi, h^2\eta) = \widetilde{\rho}(r)\,[\rho(r) - h\xi\,\sigma(r)] - h^2\eta\,\widetilde{\sigma}(r)\,\sigma(r), \tag{29}$$

where the parameter $h\xi, h^2\eta \in R$. By letting $H_1 = h\xi$ and $H_2 = h^2\eta$, hence the stability polynomial of VIDE of the second kind will be,

$$\pi(r, H_1, H_2) = \widetilde{\rho}(r)\,[\rho(r) - H_1\,\sigma(r)] - H_2\,\widetilde{\sigma}(r)\,\sigma(r), \tag{30}$$

where $\rho(r)$ is the first characteristics polynomial and $\sigma(r)$ is the second characteristic polynomial of the linear multistep method and defined as

$$\rho(r) = \sum_{j=0}^{k} \alpha_j r^j, \qquad \sigma(r) = \sum_{j=0}^{k} \beta_j r^j.$$

Hence, the characteristics polynomial of the associated method and Boole's rule are given as,

i) Corrector formula for the first point, y_{n+1},

$$\begin{aligned} \rho_1(r) &= r^4 - r^3, \\ \sigma_1(r) &= \frac{251}{720}r^4 + \frac{323}{360}r^3 - \frac{11}{30}r^2 + \frac{53}{360}r - \frac{19}{720}. \end{aligned} \tag{31}$$

ii) Corrector formula for the second point, y_{n+2},

$$\begin{aligned} \rho_2(r) &= r^5 - r^3, \\ \sigma_2(r) &= \frac{29}{90}r^5 + \frac{62}{45}r^4 + \frac{4}{15}r^3 + \frac{2}{45}r^2 - \frac{1}{90}r. \end{aligned} \tag{32}$$

iii) Boole's rule

$$\begin{aligned} \widetilde{\rho}(r) &= r^4 - 1, \\ \widetilde{\sigma}(r) &= \frac{14}{45}r^4 + \frac{64}{45}r^3 + \frac{24}{45}r^2 + \frac{64}{45}r + \frac{14}{45}. \end{aligned} \tag{33}$$

The stability polynomial for the corrector formulae is evaluated by substituting the characteristics polynomial of the derived method and Boole's rule into (30). Thus, the stability polynomial can be obtain as follows;

$$\begin{aligned} \pi(r, H_1, H_2) = & \left(\frac{19}{64800} + \frac{7279}{64800}r^8 - \frac{10217}{8100}r^7 - \frac{2339}{4050}r^6 + \frac{20357}{8100}r^5 + \frac{5303}{6480}r^4 - \frac{10063}{8100}r^3\right. \\ & \left. - \frac{2863}{8100}r^2 - \frac{77}{8100}r\right) H_1^2 + \left(\frac{931}{32805000} + \frac{356671}{32805000}r^8 - \frac{1279486}{4100625}r^7 + \frac{13924927}{8201250}r^6\right. \\ & \left. + \frac{20115938}{4100625}r^5 + \frac{55652293}{16402500}r^4 + \frac{3789509}{8201250}r^3 - \frac{35056}{8201250}r^2 - \frac{11143}{8201250}r\right) H_2^2 \\ & + \left(+\frac{50953}{729000}r^8 - \frac{121201}{182250}r^7 - \frac{63383}{40500}r^6 - \frac{91513}{182250}r^5 + \frac{446771}{364500}r^4 + \frac{422917}{364500}r^3\right. \\ & \left. + \frac{49133}{182250}r^2 + \frac{31}{4500}r - \frac{133}{729000}\right) H_1 H_2 + \left(-\frac{161}{240}r^8 - \frac{223}{240}r^7 + \frac{697}{720}r^6 + \frac{1319}{720}r^5\right. \\ & \left. + \frac{11}{144}r^4 - \frac{631}{720}r^3 - \frac{269}{720}r^2 - \frac{19}{720}r\right) H_1 + \left(-\frac{1127}{5400}r^8 H_2 - \frac{70319}{16200}r^7 H_2 - \frac{59639}{16200}r^6 H_2\right. \\ & \left. + \frac{5069}{1080}r^5 H_2 + \frac{797}{200}r^4 H_2 - \frac{5849}{16200}r^3 H_2 - \frac{1537}{16200}r^2 H_2 + \frac{133}{16200}r H_2\right) H_2 + r^8 \\ & - r^7 - 2r^6 + 2r^5 - r^3 + r^4 = 0. \end{aligned} \tag{34}$$

Substituting $-1, 0, 1$ and $\cos(\theta) + i\sin(\theta)$ for r in the $H_1 - H_2$ plane yields the absolute stability boundary. The stability region is illustrated using Maple software and presented in Figure 2. From Figure 2, it explained that the method is stable within the shaded region. The stability region demonstrates that the proposed method could generate the appropriate results with the given values of the time steps.

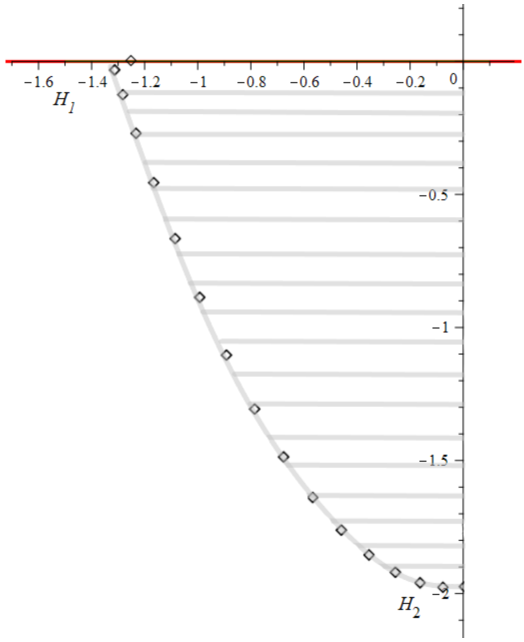

Figure 2: Stability region of the proposed method in H_1-H_2 plane.

5 Numerical Results and Discussion

The mathematical results are tabulated in Tables 1 - 4. The notations implemented for the following tables and figures are as below;

h	:	Step size.
MAXE	:	Maximum errors.
TS	:	Total of steps.
TFC	:	Total function calls.
Time	:	The execution time taken in second.
ABM5	:	Adam-Bashforth-Moulton of fifth order method with Boole's rule.
DIMBM	:	Diagonally implicit multistep block method of third order with Simpson's rule by [1].
2P3BVIDE	:	Two-point three-step block method of fifth order with Boole's rule in [13].
MBBM5	:	Multistep block-Boole's rule method of fifth order in this research.

The maximum error can be determined as

$$MAXE = \max_{0 \le n \le N} |y(x_n) - y_n|.$$

Contemplate the following problems, the efficiency of the proposed method can be contrasted with existing methods.

Problem 1 (Linear VIDE)

$$\begin{aligned} y'(x) &= 1-\int_0^x y(s)\,ds, \\ y(0) &= 0, \qquad 0\le x\le 1. \end{aligned}$$

Exact solution: $y(x)=sin(x)$.
Source: [7]

Problem 2 (Linear VIDE)

$$\begin{aligned} y'(x) &= -\sin(x)-\cos(x)+\int_0^x 2\cos(x-s)y(s)\,ds, \\ y(0) &= 1, \qquad 0\le x\le 5. \end{aligned}$$

The exact solution is $y(x)=\exp(-x)$.
Source: [4]

Problem 3 (Non-linear VIDE)

$$\begin{aligned} y'(x) &= 2x-\frac{1}{2}\sin(x^4)+\int_0^x x^2 s\cos(x^2y(s))\,ds, \\ y(0) &= 0, \qquad 0\le x\le 2. \end{aligned}$$

with the exact solution, $y(x)=x^2$.
Source:[6]

Problem 4 (Non-linear VIDE)

$$\begin{aligned} y'(x) &= x\exp(1-y(x))-\frac{1}{(1+x)^2}-x-\int_0^x \frac{x}{(1+s)^2}\exp(1-y(s))\,ds, \\ y(0) &= 1, \qquad 0\le x\le 4. \end{aligned}$$

The exact solution for this problem is $y(x)=\dfrac{1}{1+x}$.
Source: [15]

All the computation results computed in C language on the Code::Blocks platform, where the performance of DIMBM, ABM5, 2P3BVIDE and MBBM5 can be compared. The ABM5, 2P3BVIDE and MBBM5 satisfied the method of order five, while DIMBM is method of order three. Moreover, MBBM5 and DIMBM are diagonally implicit block methods, whereas 2P3BVIDE is a fully implicit block method. Furthermore, all the methods are in the form of the multistep method and these methods used the constant step size in their formulation. Tables 1 - 4 provide all of the numerical results.

Table 1: Computational result for Problem 1.

h	Method	MAXE	TFC	TS	Time
$\frac{1}{40}$	DIMBM	4.1354(-07)	42	21	0.0450
	ABM5	4.4529(-09)	88	40	0.1020
	2P3BVIDE	1.2349(-09)	50	22	0.0700
	MBBM5	2.8892(-09)	42	22	0.0310
$\frac{1}{80}$	DIMBM	4.7815(-08)	82	41	0.1120
	ABM5	2.3862(-10)	168	80	0.1579
	2P3BVIDE	3.8642(-11)	90	42	0.1166
	MBBM5	1.4990(-10)	82	42	0.0450
$\frac{1}{160}$	DIMBM	5.8390(-09)	162	81	0.1600
	ABM5	1.4271(-11)	328	160	0.2622
	2P3BVIDE	1.2080(-12)	170	82	0.2034
	MBBM5	8.5145(-12)	162	82	0.1390
$\frac{1}{320}$	DIMBM	7.2152(-10)	322	161	0.2660
	ABM5	8.6009(-13)	648	320	0.4222
	2P3BVIDE	3.7751(-14)	330	162	0.3124
	MBBM5	5.0404(-13)	322	162	0.2000

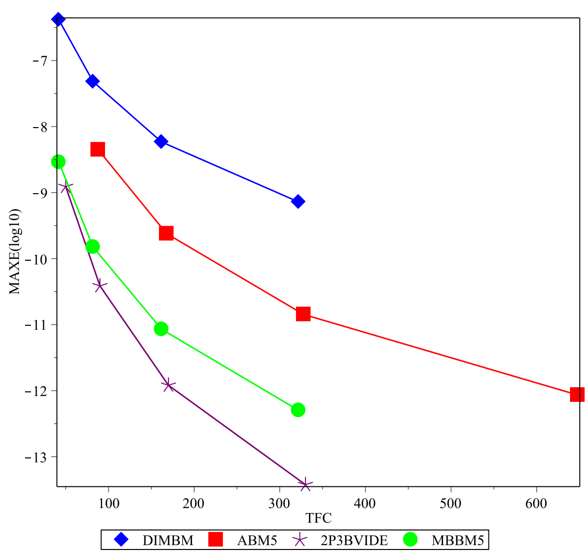

Figure 3: Graph of total function calls against MAXE for Problem 1.

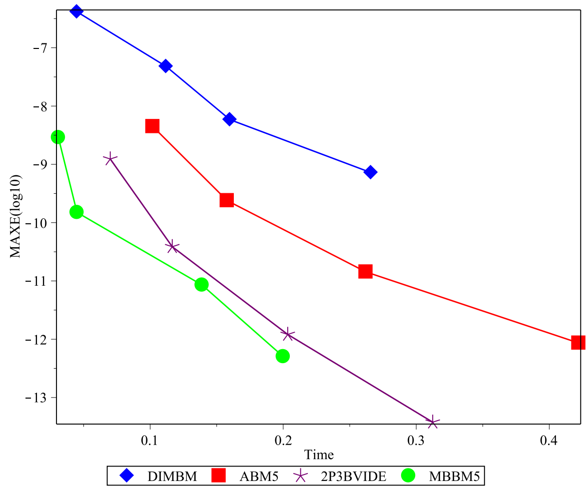

Figure 4: Graph of the execution time take against MAXE for Problem 1.

Problem 1 was solved by setting the step size to $h = 1/40, 1/80, 1/160$ and $1/320$, respectively. Table 1, the tabulated data shows the 2P3BVIDE achieved better accuracy than DIMBM, ABM5 and MBBM5. The 2P3BVIDE is the fully implicit block method compared to the MBBM5, which is the diagonally implicit block method. Therefore, the 2P3BVIDE can have extra total function calls even though the order of method is the same. With the least number of total function calls and execution time taken, it gives advantages to the MBBM5 compared to the ABM5 and 2P3BVIDE, visualised clearly in Figures 3 and 4.

Table 2: Computational result for Problem 2.

h	Method	MAXE	TFC	TS	Time
$\frac{1}{4}$	DIMBM	1.8211(-02)	22	11	0.0140
	ABM5	8.1337(-03)	85	20	0.0715
	2P3BVIDE	6.1138(-03)	59	11	0.0462
	MBBM5	6.0777(-04)	40	12	0.0030
$\frac{1}{8}$	DIMBM	3.4489(-03)	42	21	0.0410
	ABM5	4.7616(-04)	165	40	0.1623
	2P3BVIDE	3.9009(-04)	99	21	0.0900
	MBBM5	9.9481(-05)	80	22	0.0060
$\frac{1}{16}$	DIMBM	4.6803(-04)	170	41	0.1170
	ABM5	2.1034(-05)	325	80	0.2139
	2P3BVIDE	1.6881(-05)	179	41	0.1930
	MBBM5	5.1571(-06)	160	42	0.0370
$\frac{1}{32}$	DIMBM	6.0807(-05)	162	81	0.1480
	ABM5	7.8509(-07)	645	160	0.3278
	2P3BVIDE	6.1208(-07)	339	81	0.2494
	MBBM5	2.0353(-07)	320	82	0.0990

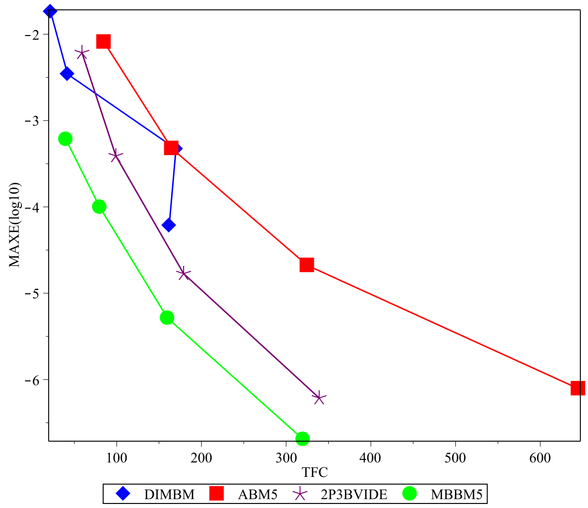

Figure 5: Graph of total function calls against MAXE for Problem 2.

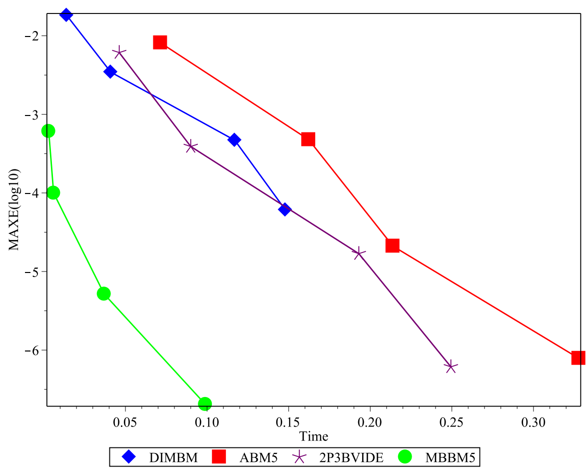

Figure 6: Graph of the execution time take against MAXE for Problem 2.

Table 2 evaluate the performance of MBBM5 at the different step sizes to DIMBM, ABM5 and 2P3BVIDE in solving Problem 2. The MBBM5 demonstrated greater precision than other existing methods. Figure 5 shows the MBBM5 obtained fewer function calls to complete the interval than the other fifth order methods.

Table 3: Computational result for Problem 3.

h	Method	MAXE	TFC	TS	Time
$\frac{2}{9}$	DIMBM	8.8008(-03)	10	-	0.0050
	ABM5	7.2747(-02)	41	7	0.0520
	2P3BVIDE	6.8284(-02)	35	5	0.0470
	MBBM5	6.2668(-02)	16	6	0.0010
$\frac{2}{17}$	DIMBM	8.6068(-04)	18	-	0.0280
	ABM5	7.8868(-03)	73	15	0.0730
	2P3BVIDE	8.4729(-03)	51	9	0.0352
	MBBM5	8.5662(-03)	32	10	0.0040
$\frac{2}{33}$	DIMBM	1.7703(-04)	34	-	0.0350
	ABM5	8.9015(-05)	137	31	0.1355
	2P3BVIDE	9.3109(-05)	83	17	0.0780
	MBBM5	9.2589(-05)	64	18	0.0060
$\frac{2}{65}$	DIMBM	6.6994(-06)	66	-	0.0420
	ABM5	2.9296(-07)	265	63	0.2133
	2P3BVIDE	3.0567(-07)	147	33	0.1560
	MBBM5	3.3921(-07)	128	34	0.0530

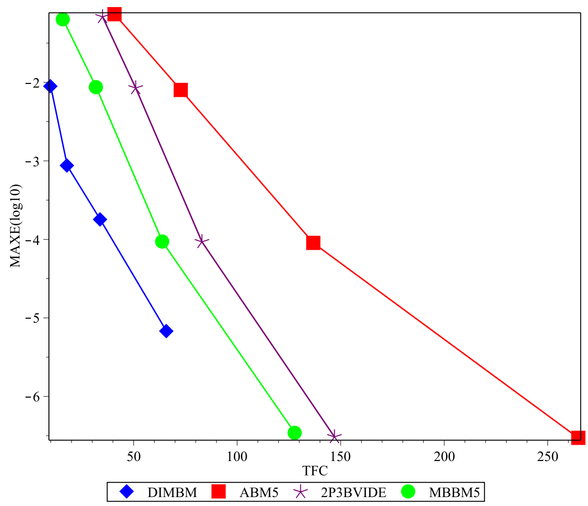

Figure 7: Graph of total function calls against MAXE for Problem 3.

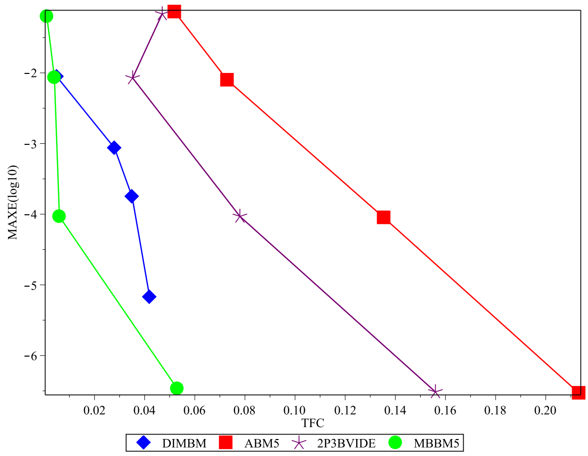

Figure 8: Graph of the execution time take against MAXE for Problem 3.

The nonlinear problem of VIDE was solved numerically and the numerical results were tabulated in Tables 3 and 4. In table 3, the accuracy achieved by DIMBM is significantly better than MBBM5 at the step sizes, $h = 2/9$ and $2/17$ due to its starting method, which is a one-step method. Meanwhile, the MBBM5 began with the Adam Bashforth Moulton method, which is also known as the multistep method. The total function calls of DIMBM are lesser than MBBM5, as illustrated in Figure 7. This is because the third order DIMBM does only one iteration, whereas the MBBM5 performed two iterations to obtain more accurate results. However, as the step size gets smaller, the MBBM5 dominates other methods in terms of timing, as depicted in Figure 8.

Table 4: Computational result for Problem 4.

h	Method	MAXE	TFC	TS	Time
$\frac{1}{40}$	DIMBM	3.6545(-06)	322	-	0.2610
	ABM5	1.7212(-08)	645	160	0.3430
	2P3BVIDE	8.3237(-08)	339	81	0.2850
	MBBM5	1.1817(-08)	320	82	0.1990
$\frac{1}{80}$	DIMBM	4.6274(-07)	642	-	0.3010
	ABM5	3.0551(-09)	1285	320	0.5544
	2P3BVIDE	3.8384(-09)	659	161	0.3879
	MBBM5	4.2375(-10)	640	162	0.3250
$\frac{1}{160}$	DIMBM	5.8216(-08)	1282	-	0.6740
	ABM5	1.9089(-10)	2565	640	1.1720
	2P3BVIDE	2.0775(-10)	1299	321	0.6778
	MBBM5	1.4194(-11)	1280	322	0.6310
$\frac{1}{320}$	DIMBM	7.2999(-09)	2562	-	1.1050
	ABM5	1.1926(-11)	5125	1280	1.8678
	2P3BVIDE	1.2654(-11)	2579	641	1.4850
	MBBM5	4.6324(-13)	2560	642	1.0990

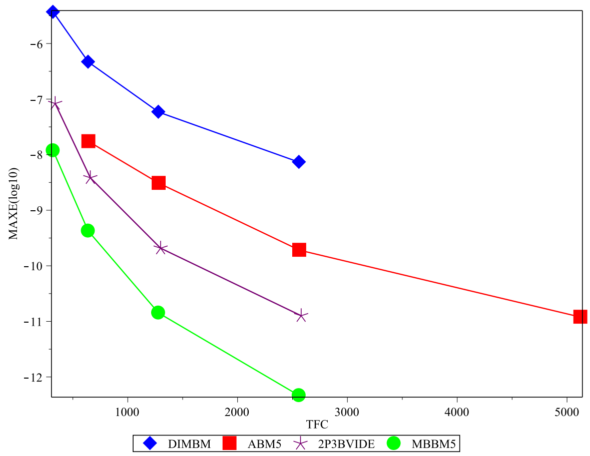

Figure 9: Graph of total function calls against MAXE for Problem 4.

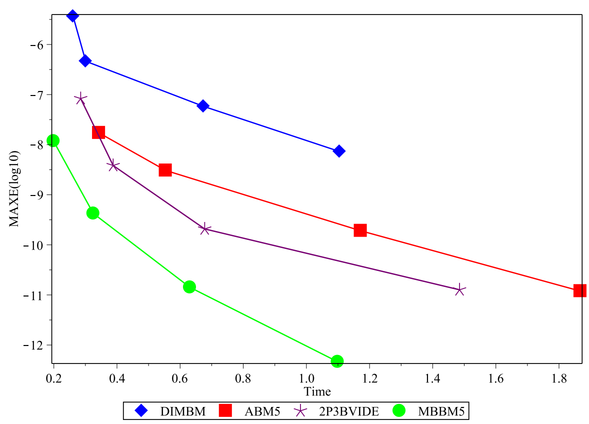

Figure 10: Graph of the execution time take against MAXE for Problem 4.

The maximum errors for the MBBM5 with the existing methods are slightly better in Table 4. The numerical results show that the total function calls by the MBBM5 are lesser than the existing method. Moreover, the MBBM5 saves substantial time and much quicker than the existing method due to the cost per step is much cheaper and less function evaluation required per step.

The order of convergence was calculated for all tested problems and yields,

Order of convergence for Problem 1:

$$= \log\left(\frac{\left(\frac{2.8892(-09)}{1.4990(-10)}\right)}{\left(\frac{\frac{1}{40}}{\frac{1}{80}}\right)}\right) = 4.2685.$$

Order of convergence for Problem 2:

$$= \log\left(\frac{\left(\frac{9.9481(-05)}{5.1571(-06)}\right)}{\left(\frac{\frac{1}{8}}{\frac{1}{16}}\right)}\right) = 4.2697.$$

Order of convergence for Problem 3:

$$= \log\left(\frac{\left(\frac{8.5662(-03)}{9.2589(-05)}\right)}{\left(\frac{\frac{2}{17}}{\frac{2}{33}}\right)}\right) = 6.8256.$$

Order of convergence for Problem 4:

$$= \log\left(\frac{\left(\frac{4.2357(-10)}{1.4194(-11)}\right)}{\left(\frac{\frac{1}{80}}{\frac{1}{160}}\right)}\right) = 4.8992.$$

Problem 1 and 2 have achieved fourth order accuracy while Problem 3 and 4 has produced sixth order accuracy and fifth order accuracy. In Problem 1 and 2, the implementation of Adam Bashforth method to estimate the first and second initial values may have affected the accuracy. The order of the method used is less than four due to lack of available points at the beginning. Furthermore, the accuracy of the proposed method also could be influenced by the lower order Lagrange interpolating polynomial in the composite Boole's rule.

6 Conclusions

The numerical results demonstrated that the maximum error of MBBM5 produced better accuracy as the step size decreases. The MBBM5 outperforms the other methods in terms of execution times and total function calls. Therefore, the MBBM5 is reliable in determining the approximate solutions for the second kind of VIDE. This proposed method is both efficient and economical.

Acknowledgement The author gratefully acknowledged the Fundamental Research Grant Scheme (Project Code:FRGS/1/2016/STG06/UPM/01/3) from the Minister of Education Malaysia and Graduate Research Fund (GRF) from Universiti Putra Malaysia.

Conflicts of Interest The authors declare no conflict of interest.

References

[1] N. A. Baharum, Z. A. Majid & N. Senu (2018). Solving Volterra integrodifferential equations via diagonally implicit multistep block method. *International Journal of Mathematics and Mathematical Sciences*, *2018*, Article ID: 7392452. https://doi.org/10.1155/2018/7392452.

[2] H. Brunner & J. D. Lambert (1974). Stability of numerical methods for Volterra integro-differential equations. *Computing*, *12*(1), 75–89. https://doi.org/10.1007/BF02239501.

[3] S. H. Chang & J. T. Day (1978). On the numerical solution of a certain nonlinear integro-differential equation. *Journal of Computational Physics*, *26*(2), 162–168. https://doi.org/10.1016/0021-9991(78)90088-8.

[4] H. Chen & C. Zhang (2011). Boundary value methods for Volterra integral and integro-differential equations. *Applied Mathematics and Computation*, *218*(6), 2619–2630. https://doi.org/10.1016/j.amc.2011.08.001.

[5] J. T. Day (1970). Note on the numerical solution of integro-differential equations. *BIT Numerical Mathematics*, *10*, 511–514. https://doi.org/10.1007/BF01935570.

[6] M. Dehghan & R. Salehi (2012). The numerical solution of the non-linear integro-differential equations based on the meshless method. *Journal of Computational and Applied Mathematics*, *236*(9), 2367–2377. https://doi.org/10.1016/j.cam.2011.11.022.

[7] A. Filiz (2013). A fourth-order robust numerical method for integro-differential equations. *Asian Journal of Fuzzy and Applied Mathematics*, *1*(1), 28–33.

[8] A. Filiz (2014). Numerical method for a linear Volterra integro-differential equation with Cash-Karp method. *Asian Journal of Fuzzy and Applied Mathematics*, *2*(1), 1–11.

[9] F. Ishak & S. N. Ahmad (2016). Development of extended trapezoidal method for numerical solution of Volterra integro-differential equations. *International Journal of Mathematics, Computational, Physical, Electrical and Computer Engineering*, *10*(11), 52856. https://doi.org/10.5281/zenodo.1127402.

[10] M. R. Janodi, Z. A. Majid, F. Ismail & N. Senu (2020). Numerical solution of Volterra integro-differential equations by hybrid block with quadrature rules method. *Malaysian Journal of Mathematical Sciences*, *14*(2), 191–208.

[11] J. D. Lambert (1973). *Computational methods in ordinary differential equations*. John Wiley & Sons, Hoboken, United States.

[12] P. Linz (1969). Linear multistep methods for Volterra integro-differential equations. *Journal of the ACM (JACM)*, *16*(2), 295–301. https://doi.org/10.1145/321510.321521.

[13] Z. A. Majid & N. A. Mohamed (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/JSM-2019-4803-22.

[14] N. A. Mohamed & Z. A. Majid (2015). One-step block method for solving Volterra integro-differential equations. In *AIP Conference Proceedings*,. AIP Publishing, Seattle, United States.

[15] R. E. Shaw (2000). A parallel algorithm for nonlinear Volterra integro-differential equations. In *Proceedings of the 2000 ACM Symposium on Applied Computing*, pp. 86–88. ACM, Como, Italy.

[16] T. Tang (1993). A note on collocation methods for Volterra integro-differential equations with weakly singular kernels. *IMA Journal of Numerical Analysis*, *13*(1), 93–99. https://doi.org/10.1093/imanum/13.1.93.

[17] C. Tunç & O. Tunç (2018). New qualitative criteria for solutions of volterra integro-differential equations. *Arab Journal of Basic and Applied Sciences*, *25*(3), 158–165. https://doi.org/10.1080/25765299.2018.1509554.

[18] P. V. Ubale (2012). Numerical solution of Boole's rule in numerical integration by using general quadrature formula. *Bulletin of Society for Mathematical Services & Standards (B SO MA SS)*, *1*(2), 1–5. https://doi.org/10.18052/www.scipress.com/BSMaSS.2.1.

[19] W. Yuan & T. Tang (1990). The numerical analysis of implicit Runge-Kutta methods for a certain nonlinear integro-differential equation. *Mathematics of Computation*, *54*(189), 155–168. https://doi.org/10.2307/2008687.