

Research Article

Solving Volterra Integrodifferential Equations via Diagonally Implicit Multistep Block Method

Nur Auni Baharum,¹ Zanariah Abdul Majid ^{1,2} and Norazak Senu ^{1,2}

¹Institute for Mathematical Research, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

²Mathematics Department, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Correspondence should be addressed to Zanariah Abdul Majid; zana_majid99@yahoo.com

Received 15 June 2017; Revised 13 October 2017; Accepted 23 November 2017; Published 1 January 2018

Academic Editor: Theodore E. Simos

Copyright © 2018 Nur Auni Baharum et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The performance of the numerical computation based on the diagonally implicit multistep block method for solving Volterra integrodifferential equations (VIDE) of the second kind has been analyzed. The numerical solutions of VIDE will be computed at two points concurrently using the proposed numerical method and executed in the predictor-corrector (PECE) mode. The strategy to obtain the numerical solution of an integral part is discussed and the stability analysis of the diagonally implicit multistep block method was investigated. Numerical results showed the competence of diagonally implicit multistep block method when solving Volterra integrodifferential equations compared to the existing methods.

1. Introduction

Consider the Volterra integrodifferential equation of the second kind

$$y'(x) = F(x, y(x), z(x)), \quad y(x_0) = y_0, \quad (1)$$

where

$$z(x) = \int_0^x K(x, s) y(s) ds, \quad 0 \leq s \leq x. \quad (2)$$

The numerical methods are generated to solve (1) which is a standard algorithm for ordinary differential equations and Newton-Cote integration formulae are required for solving the integral part since it cannot be solved explicitly. These equations usually appeared in physics, biology, and engineering applications such as biological models, neutron diffusion, wind ripple in the desert, heat transfer, and many more.

For many years, several methods had been applied to solve first-order problem of VIDE. Day [1] proposed Newton-Cotes integration formula of the trapezoidal rule for the solutions of outer and inner integral to obtain approximate

solutions of integrodifferential equations. A comparison between the variational iteration method and trapezoidal rule revealing that the variational iteration method is more efficient and convenient to solve linear VIDE was discussed by Saadati et al. [2].

In [3], finite difference method is used for solving linear VIDE by Raftari. He transforms the Volterra integrodifferential equation in a matrix form and solved it by using finite difference method based on Simpson's rule and trapezoidal rule. A fourth-order robust numerical method was presented by Filiz [4] with a combination of the trapezoidal rule and Simpson's 1/3 rule to evaluate the solution of VIDE for kernel equal to one. Then, he extended his work with a Runge-Kutta-Verner method in [5] and used higher rules of numerical integration method for solving the integral part.

The extended trapezoidal method [6] was proposed for the numerical solution of VIDE of the second kind and implemented the method in PECE scheme. Mohamed and Majid [7] had solved the second kind of VIDE using one-step block method and the Newton-Cotes quadrature formula was employed for finding the solution of the integral part. The multistep block method in [8] had implemented two approaches for solving VIDE for $K(x, s) = 1$ and $K(x, s) \neq 1$.

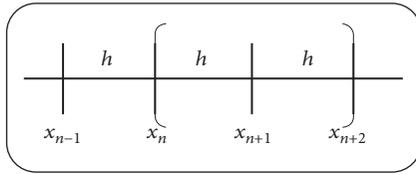


FIGURE 1: Two-point multistep block method.

2. Numerical Method

The proposed numerical method is in the form of block method and it will generate two or more solutions at the same time. The proposed method is a two-point block method; hence it will generate two solutions in one block.

In Figure 1, the two approximate values of y_{n+1} and y_{n+2} will be computed simultaneously in a block. The approximate values of y_{n+1} can be developed by integrating (1) over the interval $[x_n, x_{n+1}]$ while the interval for values y_{n+2} is $[x_n, x_{n+2}]$. Hence, the formulae of y_{n+1} and y_{n+2} can be obtained as

$$\int_{x_n}^{x_{n+r}} y'(x) dx = \int_{x_n}^{x_{n+r}} F(x, y, z) dx. \tag{3}$$

Therefore,

$$y_{n+r} - y_n = \int_{x_n}^{x_{n+r}} F(x, y, z) dx, \tag{4}$$

where $r = 1, 2$. Then, function $F(x, y, z)$ in (4) will be approximated using Lagrange interpolating polynomial, and the interpolation points involved in obtaining the corrector formula of y_{n+1} are $\{x_{n-1}, x_n, x_{n+1}\}$. Taking $x = x_{n+1} + sh$, $dx = hds$, and replacing into (4), the limit of integration in (4) will be -1 to 0 .

The formulation of y_{n+2} can be obtained when three points are involved in the interpolation polynomial, that is, $\{x_{n-1}, x_{n+1}, x_{n+2}\}$. Considering $x = x_{n+2} + sh$, $dx = hds$ in (4) and the limit of integration will be changed from -2 to 0 . The corrector formulae of y_{n+1} and y_{n+2} will be obtained using MAPLE software. The predictor formulae are one order less than the corrector formulae and the same process of derivation is applied.

Diagonally Implicit Multistep Block Method

Predictor

$$\begin{aligned} y_{n+1}^p &= y_n + h \left[\frac{3}{2} F_n - \frac{1}{2} F_{n-1} \right], \\ y_{n+2}^p &= y_n + h [4F_n - 2F_{n-1}]. \end{aligned} \tag{5}$$

Corrector

$$\begin{aligned} y_{n+1}^c &= y_n + h \left[\frac{5}{12} F_{n+1} + \frac{2}{3} F_n - \frac{1}{12} F_{n-1} \right], \\ y_{n+2}^c &= y_n + h \left[\frac{2}{9} F_{n+2} + \frac{5}{3} F_{n+1} + \frac{1}{9} F_{n-1} \right]. \end{aligned} \tag{6}$$

The matrix form of the corrector formulae is

$$\begin{aligned} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} &= \begin{bmatrix} 0 & -1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} \\ &+ h \begin{bmatrix} -\frac{1}{12} & \frac{2}{3} \\ \frac{1}{9} & \frac{5}{3} \end{bmatrix} \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix} \\ &+ h \begin{bmatrix} \frac{5}{12} & 0 \\ \frac{5}{3} & \frac{2}{9} \end{bmatrix} \begin{bmatrix} F_{n+1} \\ F_{n+2} \end{bmatrix} \end{aligned} \tag{7}$$

which is equivalent to the difference equations

$$A_0 Y_m = A_1 Y_{m-1} + h(B_0 F_{m-1} + B_1 F_m), \tag{8}$$

where A_0, A_1, B_0 , and B_1 are the coefficients with m -vectors Y_m, Y_{m-1}, F_{m-1} , and F_m defined as

$$\begin{aligned} Y_m &= \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix}, \\ Y_{m-1} &= \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix}, \\ F_{m-1} &= \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix}, \\ F_m &= \begin{bmatrix} F_{n+1} \\ F_{n+2} \end{bmatrix}. \end{aligned} \tag{9}$$

3. Analysis of Diagonally Implicit Multistep Block Method

3.1. Order and Convergence of the Method. The order of the method can be obtained by referring [9]

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

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

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

Definition 1. The numerical method is said to be in order p if the linear operator of numerical method is

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

where C_{p+1} is called as an error constant of the method.

The order of diagonally implicit multistep block method in (7) can be determined by applying the formula in (11); hence the values of α and β are obtained as follows:

$$\begin{aligned} \alpha_0 &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ \alpha_1 &= \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \\ \alpha_2 &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \\ \alpha_3 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ \beta_0 &= \begin{bmatrix} -\frac{1}{12} \\ \frac{1}{9} \end{bmatrix}, \\ \beta_1 &= \begin{bmatrix} \frac{2}{3} \\ 0 \end{bmatrix}, \\ \beta_2 &= \begin{bmatrix} \frac{5}{12} \\ \frac{5}{3} \end{bmatrix}, \\ \beta_3 &= \begin{bmatrix} 0 \\ \frac{2}{9} \end{bmatrix}. \end{aligned} \tag{13}$$

Substitute the values of α and β into (11) and obtain

$$\begin{aligned} C_0 &= \sum_{j=0}^3 \frac{1}{0!} j^0 \alpha_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ C_1 &= \sum_{j=0}^3 \frac{1}{1!} j^1 \alpha_j - \sum_{j=0}^3 \frac{1}{0!} j^0 \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ C_2 &= \sum_{j=0}^3 \frac{1}{2!} j^2 \alpha_j - \sum_{j=0}^3 \frac{1}{1!} j^1 \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ C_3 &= \sum_{j=0}^3 \frac{1}{3!} j^3 \alpha_j - \sum_{j=0}^3 \frac{1}{2!} j^2 \beta_j = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ C_4 &= \sum_{j=0}^3 \frac{1}{4!} j^4 \alpha_j - \sum_{j=0}^3 \frac{1}{3!} j^3 \beta_j = \begin{bmatrix} -\frac{1}{24} \\ \frac{1}{9} \end{bmatrix}. \end{aligned} \tag{14}$$

Therefore, the diagonally implicit multistep block method is third-order where the coefficient of error constant is

$$C_{p+1} = C_4 = \begin{bmatrix} -\frac{1}{24} & \frac{1}{9} \end{bmatrix}^T \neq [0 \ 0]^T. \tag{15}$$

Definition 2. The local truncation error at x_{n+k} of the method is defined to be expression $L[y(x_n); h]$, when $y(x)$ is the theoretical solution of the initial value problem

$$L[y(x_n); h] = \sum_{j=0}^k [\alpha_j y(x + jh) - h\beta_j y'(x + jh)]. \tag{16}$$

For the formula y_{n+1}^c ,

$$y_{n+1}^c = y_n + h \left[\frac{5}{12} F_{n+1} + \frac{2}{3} F_n - \frac{1}{12} F_{n-1} \right]. \tag{17}$$

Since $F_{n+1} = y'(x_{n+1})$, Taylor expansion will be applied to the derivatives $y'(x_{n+1})$ and $y'(x_{n-1})$ where $F_{n-1} = y'(x_{n-1})$,

$$\begin{aligned} y'(x_{n+1}) &= y'(x_n) + hy''(x_n) + \frac{h^2}{2!} y'''(x_n) \\ &\quad + O(h^3), \end{aligned} \tag{18}$$

$$\begin{aligned} y'(x_{n-1}) &= y'(x_n) - hy''(x_n) + \frac{h^2}{2!} y'''(x_n) \\ &\quad + O(h^3). \end{aligned}$$

Then, since $F_n = y'(x_n)$ we have

$$y_{n+1} = y_n + hy'_n + \frac{h^2}{2} y''_n + \frac{h^3}{6} y'''_n + O(h^4). \tag{19}$$

So, the local truncation error for y_{n+1} is $O(h^4)$. For the formula y_{n+2}^c ,

$$y_{n+2}^c = y_n + h \left[\frac{2}{9} F_{n+2} + \frac{5}{3} F_{n+1} + \frac{1}{9} F_{n-1} \right]. \tag{20}$$

The Taylor expansion for $y'(x_{n+2}) = F_{n+2}$ is given as

$$\begin{aligned} y'(x_{n+2}) &= y'(x_n) + 2hy''(x_n) + \frac{(2h)^2}{2!} y'''(x_n) \\ &\quad + O(h^3). \end{aligned} \tag{21}$$

Then, we will have

$$y_{n+2} = y_n + 2hy'_n + 2h^2 y''_n + \frac{4}{3} h^3 y'''_n + O(h^4). \tag{22}$$

This shows that the local truncation error for y_{n+2} is $O(h^4)$.

Definition 3. The numerical method is said to be consistent if the order of method is $p \geq 1$ and the method is consistent if and only if

$$\sum_{j=0}^k \alpha_j = 0, \tag{23}$$

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

Proof. (i) $\sum_{j=0}^k \alpha_j = 0$,

$$\begin{aligned} \sum_{j=0}^3 \alpha_j &= \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{24}$$

(ii) $\sum_{j=0}^k j \cdot \alpha_j = \sum_{j=0}^k \beta_j$,

$$\begin{aligned} \sum_{j=0}^3 j \cdot \alpha_j &= 0 \cdot \alpha_0 + 1 \cdot \alpha_1 + 2 \cdot \alpha_2 + 3 \cdot \alpha_3 \\ &= 0 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \\ \sum_{j=0}^3 \beta_j &= \beta_0 + \beta_1 + \beta_2 + \beta_3 \end{aligned} \tag{25}$$

$$= \begin{bmatrix} -\frac{1}{12} \\ \frac{1}{9} \end{bmatrix} + \begin{bmatrix} \frac{2}{3} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{5}{12} \\ \frac{5}{3} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2}{9} \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Therefore,

$$\sum_{j=0}^3 j \cdot \alpha_j = \sum_{j=0}^3 \beta_j = \begin{bmatrix} 1 \\ 2 \end{bmatrix}. \tag{26}$$

By Definitions 1 and 3, the diagonally implicit multistep block method is consistent. \square

Definition 4. 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

$$\rho(R) = \det \left[\sum_{j=0}^k A^{(j)} R^{(k-j)} \right] = 0 \tag{27}$$

satisfies $|R_j| \leq 1$ and those roots with $|R_j| = 1$.

Proof. The values of A can be obtained in (7):

$$\begin{aligned} \rho(r) &= \det [RA_0 - A_1] = \det \left[R \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & -1 \\ 0 & -1 \end{bmatrix} \right] \\ &= \det \begin{bmatrix} R & 1 \\ 0 & R+1 \end{bmatrix} = R(R+1). \end{aligned} \tag{28}$$

The diagonally implicit multistep block method is zero stable since $|R| \leq 1$. \square

Theorem 5. *The method is said to be convergent if and only if the method is consistent and zero-stable.*

Proof. By Definitions 1, 3, and 4, the diagonally implicit multistep block method is convergent. \square

3.2. Stability Region of the Method. In this section, the stability region of the diagonally implicit multistep block method of order three is discussed for the numerical solution of VIDE. The test equation for first-order VIDE of the second kind [10] is

$$y'(x) = \xi y(x) + \eta \int_0^x y(t) dt, \tag{29}$$

where ξ and η are real constants, $\xi = \lambda + \mu$ and $\eta = -\lambda\mu$. Therefore,

$$y'(x) = (\lambda + \mu) y(x) - \lambda\mu \int_0^x y(t) dt. \tag{30}$$

Definition 6. The method is said to be A -stable if and only if the region of absolute stability contains at the quarter plane $h\xi < 0, h^2\eta < 0$.

From the proposed method for the numerical solution, the characteristics polynomials $\rho(r)$, $\sigma(r)$, $\bar{\rho}(r)$, and $\bar{\sigma}(r)$ can be developed as follows.

Corrector formula for y_{n+1} is

$$\begin{aligned} \rho(r) &= r^2 - r, \\ \sigma(r) &= \frac{5}{12}r^2 + \frac{2}{3}r - \frac{1}{12}. \end{aligned} \tag{31}$$

Corrector formula for y_{n+2} is

$$\begin{aligned} \rho(r) &= r^3 - r, \\ \sigma(r) &= \frac{2}{9}r^3 + \frac{5}{3}r^2 + \frac{1}{9}. \end{aligned} \tag{32}$$

Simpson's rule is

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

The stability polynomial of the diagonally implicit multistep block method can be determined by substituting (31), (32), and (33) into this particular formula,

$$\begin{aligned} \pi(r, h\xi, h^2\eta) &= \bar{\rho}(r) [\rho(r) - H_1\sigma(r)] \\ &\quad - H_2\bar{\sigma}(r)\sigma(r), \end{aligned} \tag{34}$$

where $H_1 = h\xi$ and $H_2 = h^2\mu$. Thus, the stability polynomial of the proposed method is obtained:

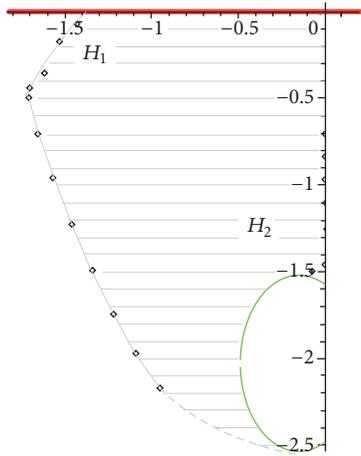


FIGURE 2: Stability region in H_1, H_2 plane.

$$\begin{aligned}
 & -\frac{7}{36}H_1 + \frac{7}{108}H_2 - \frac{2}{27}H_1^2 - \frac{2}{243}H_2^2 + \frac{5}{54}H_1^2r^4 \\
 & - \frac{71}{54}H_1^2r^3 + \frac{41}{18}H_1^2r^2 - \frac{53}{54}H_1^2r + \frac{3}{2}H_1r^2 \\
 & - \frac{7}{9}H_1r - \frac{23}{36}H_1r^4 + \frac{1}{9}H_1r^3 + \frac{4}{81}H_1H_2 \\
 & + \frac{5}{486}H_2^2r^4 - \frac{131}{486}H_2^2r^3 + \frac{95}{54}H_2^2r^2 - \frac{5}{486}H_2^2r \quad (35) \\
 & - \frac{23}{108}H_2r^4 - \frac{205}{54}H_2r^3 + \frac{116}{27}H_2r^2 - \frac{19}{54}H_2r \\
 & - \frac{61}{81}H_1H_2r^3 + \frac{61}{81}H_1H_2r - \frac{1}{9}H_1H_2r^2 \\
 & + \frac{5}{81}H_1H_2r^4 + r^4 - 3r^3 + 3r^2 - r = 0.
 \end{aligned}$$

The region of stability polynomial can be illustrated in Figure 2.

Regarding Definition 6, the third-order of diagonally implicit multistep block method in Figure 2 is A -stable within the shaded region.

4. Implementation

The one-step methods are required for finding the first starting point at x_{n+1} since Volterra integrodifferential equations of the second kind have two types of kernels. For $K(x, s) = 1$, Runge-Kutta method is involved in solving differential part of VIDE, while, midpoint method is needed to solve VIDE when $K(x, s) \neq 1$. Hence, the predictor and the corrector formula can be implemented until the end of the interval

$$y_{n+1}^p = y_n + h \left[\frac{3}{2}F_n - \frac{1}{2}F_{n-1} \right],$$

$$y_{n+2}^p = y_n + h [4F_n - 2F_{n-1}],$$

$$y_{n+1}^c = y_n + h \left[\frac{5}{12}F_{n+1} + \frac{2}{3}F_n - \frac{1}{12}F_{n-1} \right],$$

$$y_{n+2}^c = y_n + h \left[\frac{2}{9}F_{n+2} + \frac{5}{3}F_{n+1} + \frac{1}{9}F_{n-1} \right].$$

(36)

Since $z(x)$ in $F(x, y(x), z(x))$ is the integral term in VIDE and cannot be solved explicitly, therefore, Simpson's rule is adapted for solving the integral part.

(i) For $K(x, s) = 1$, Simpson's 1/3 rule is applied to solve the integral term in VIDE:

$$z_{n+2} = z_n + h \left[\frac{1}{3}y_n + \frac{4}{3}y_{n+1} + \frac{1}{3}y_{n+2} \right]. \quad (37)$$

(ii) For $K(x, s) \neq 1$, composite Simpson's rule is employed for solving the integral part:

$$\begin{aligned}
 z_{n+2} &= \frac{h}{3} \sum_{i=0}^{n+2} \omega_i^s K(x_{n+2}, x_i, y_i), \\
 z_{n+3} &= \frac{h}{3} \sum_{i=0}^{n+2} \omega_i^s K(x_{n+3}, x_i, y_i) \quad (38)
 \end{aligned}$$

$$\begin{aligned}
 & + \frac{h}{6} [K(x_{n+3}, x_{n+2}, y_{n+2}) \\
 & + 4K(x_{n+2}, x_{n+5/2}, y_{n+5/2}) + K(x_{n+3}, x_{n+3}, y_{n+3})],
 \end{aligned}$$

where ω_i^s are Simpson's rule weights 1, 4, 2, 4, ..., 2, 4, 1. The unknown value of $y_{n+5/2}$ in (38) can be estimated by using Lagrange interpolation at the point $\{x_n, x_{n+1}, x_{n+2}, x_{n+3}\}$,

$$y_{n+5/2} = \frac{1}{16}y_n - \frac{5}{16}y_{n+1} + \frac{15}{16}y_{n+2} + \frac{5}{16}y_{n+3}. \quad (39)$$

4.1. Algorithm of the Method. The input of the programming is the endpoints of a and b and the integer, N . The developed algorithm for the method is given as follows.

Step 1. Set

$$\begin{aligned}
 & x_0 = a; \\
 & y_0 = \alpha; \\
 & z_0 = 0; \\
 & h = (b - a)/N; \\
 & \text{OUTPUT: } (x_0, y_0, z_0).
 \end{aligned}$$

Step 2. For $i = 1$,

When $K(x, s) = 1$, using RK3 to evaluate the value of y .
When $K(x, s) \neq 1$, applying Midpoint Method.

Step 3. For $i = 2, \dots, (N/2)$, do Steps 4-6.

Step 4. Set $x = a + ih$.

TABLE 1: Numerical results for Example 1.

h	0.025	0.0125	0.00625	0.003125
<i>MAXE</i>				
RK3	8.3229(-08)	1.0910(-08)	1.3953(-09)	1.7637(-10)
ABM3	1.9468(-06)	2.4548(-07)	3.0813(-08)	3.8593(-09)
2PMBM	4.2079(-07)	4.9165(-08)	5.9272(-09)	7.2715(-10)
DIMBM	4.1354(-07)	4.7815(-08)	5.8390(-09)	7.2152(-10)
<i>TFC</i>				
RK3	120	240	480	960
ABM3	82	162	322	642
2PMBM	43	83	163	323
DIMBM	42	82	162	322
<i>TS</i>				
RK3	40	80	160	320
ABM3	40	80	160	320
2PMBM	21	41	81	161
DIMBM	21	41	81	161
<i>Time</i>				
RK3	0.1035	0.1840	0.2970	0.4527
ABM3	0.0860	0.1410	0.2264	0.3454
2PMBM	0.0670	0.1200	0.2024	0.3204
DIMBM	0.0450	0.1120	0.1600	0.2660

Step 5. Calculate for y_{n+1}^p and z_{n+1}^p, y_{n+2}^p and z_{n+2}^p .

Step 6. Compute the solution for y_{n+1}^c and z_{n+1}^c, y_{n+2}^c and z_{n+2}^c .

Step 7. Calculate the error.

Step 8. OUTPUT: (x, y, z) and the absolute error.

Step 9. STOP.

Exact solution is $y(x) = \exp(-x)$.
Source: [11].

Example 3 ($(K(x, s) \neq 1)$ nonlinear VIDE).

$$y'(x) = x \exp(1 - y(x)) - \frac{1}{(1+x)^2} - x - \int_0^x \frac{x}{(1+s)^2} \exp(1 - y(s)) ds \tag{42}$$

$$y(0) = 1, \quad 0 \leq x \leq 4.$$

Exact solution is $y(x) = 1/(1+x)$.
Source: [12].

5. Numerical Results

Four tested problems of first-order Volterra integrodifferential equations were considered in order to study the performance of the diagonally implicit multistep block method.

Example 1 ($(K(x, s) = 1)$ linear VIDE).

$$y'(x) = 1 - \int_0^x y(s) ds, \quad y(0) = 0, \quad 0 \leq x \leq 1. \tag{40}$$

Exact solution is $y(x) = \sin(x)$.
Source: [4].

Example 2 ($(K(x, s) \neq 1)$ linear VIDE).

$$y'(x) = -\sin(x) - \cos(x) + \int_0^x 2 \cos(x-s) y(s) ds, \tag{41}$$

$$y(0) = 1, \quad 0 \leq x \leq 5.$$

Example 4 ($(K(x, s) \neq 1)$ nonlinear VIDE).

$$y'(x) = 2x - \frac{1}{2} \sin(x^4) + \int_0^x x^2 s \cos(x^2 y(s)) ds, \tag{43}$$

$$y(0) = 0, \quad 0 \leq x \leq 2.$$

Exact solution is $y(x) = x^2$.
Source: [13].

Notations used in Tables 1–4 are as follows:

h : step size

MAXE: maximum error

TABLE 2: Numerical results for Example 2.

<i>h</i>	0.25	0.125	0.0625	0.03125
	<i>MAXE</i>			
BVMs	2.1607(-01)	2.8411(-02)	3.6378(-03)	4.6011(-04)
ABM3	1.5401(-02)	3.1321(-03)	5.3778(-04)	7.9127(-05)
2PMBM	5.7923(-02)	1.2209(-03)	3.7500(-04)	8.4172(-05)
DIMBM	1.8211(-02)	3.4489(-03)	4.6803(-04)	6.0807(-05)
	<i>TFC</i>			
BVMs	-	-	-	-
ABM3	84	164	324	644
2PMBM	50	90	170	330
DIMBM	22	42	82	162
	<i>TS</i>			
BVMs	-	-	-	-
ABM3	20	40	80	160
2PMBM	11	21	41	81
DIMBM	11	21	41	81
	<i>Time</i>			
BVMs	-	-	-	-
ABM3	0.0715	0.1546	0.2433	0.3546
2PMBM	0.0460	0.0780	0.1710	0.2738
DIMBM	0.0140	0.0410	0.1170	0.1480

TABLE 3: Numerical results for Example 3.

<i>h</i>	0.025	0.0125	0.00625	0.003125
	<i>MAXE</i>			
ABM3	2.3797(-06)	3.2252(-07)	4.2061(-08)	5.3727(-09)
2PMBM	8.0020(-06)	9.7319(-07)	1.2000(-07)	1.4862(-08)
DIMBM	3.6545(-06)	4.6274(-07)	5.8216(-08)	7.2999(-09)
	<i>TFC</i>			
ABM3	644	1284	2564	5124
2PMBM	330	650	1290	2570
DIMBM	322	642	1282	2562
	<i>Time</i>			
ABM3	0.3543	0.5677	1.1650	1.9855
2PMBM	0.2810	0.4210	0.7496	1.3570
DIMBM	0.2610	0.3010	0.6740	1.1050

TS: total steps

TFC: total functions call

Time: the execution time taken

RK3: Runge-Kutta method of order 3 with Simpson's 1/3 rule by Filiz [4]

ABM3: Adam Bashforth Moulton Order 3 with Simpson's rule.

BVMs: Combination of BVMs and third-order Generalized Adams Method by Chen and Zhang [11]

2PMBM: Two points Multistep Block Method of Order 3 with Simpson's rule by Mohamed and Majid [8]

DIMBM: Diagonally implicit multistep block method with Simpson's rule proposed in this paper

Tables 1-4 display the numerical results for the four tested problems when solved using the proposed block method and the code was written in C language.

The numerical results for Examples 1-4 displayed in Tables 1-4 are solved numerically using the proposed numerical method with Simpson's rule. In Table 1, the numerical

TABLE 4: Numerical results for Example 4.

h	2/9	2/17	2/33	2/65
<i>MAXE</i>				
ABM3	5.0218(-02)	1.8761(-03)	1.1046(-04)	5.8996(-06)
2PMBM	2.7425(-02)	6.5258(-04)	1.7781(-04)	6.6888(-06)
DIMBM	8.8008(-03)	8.6068(-04)	1.7703(-04)	6.6994(-06)
<i>TFC</i>				
ABM3	40	72	136	246
2PMBM	22	38	70	134
DIMBM	10	18	34	66
<i>Time</i>				
ABM3	0.0670	0.0723	0.1291	0.2030
2PMBM	0.0140	0.0352	0.0662	0.1336
DIMBM	0.0050	0.0280	0.0350	0.0420

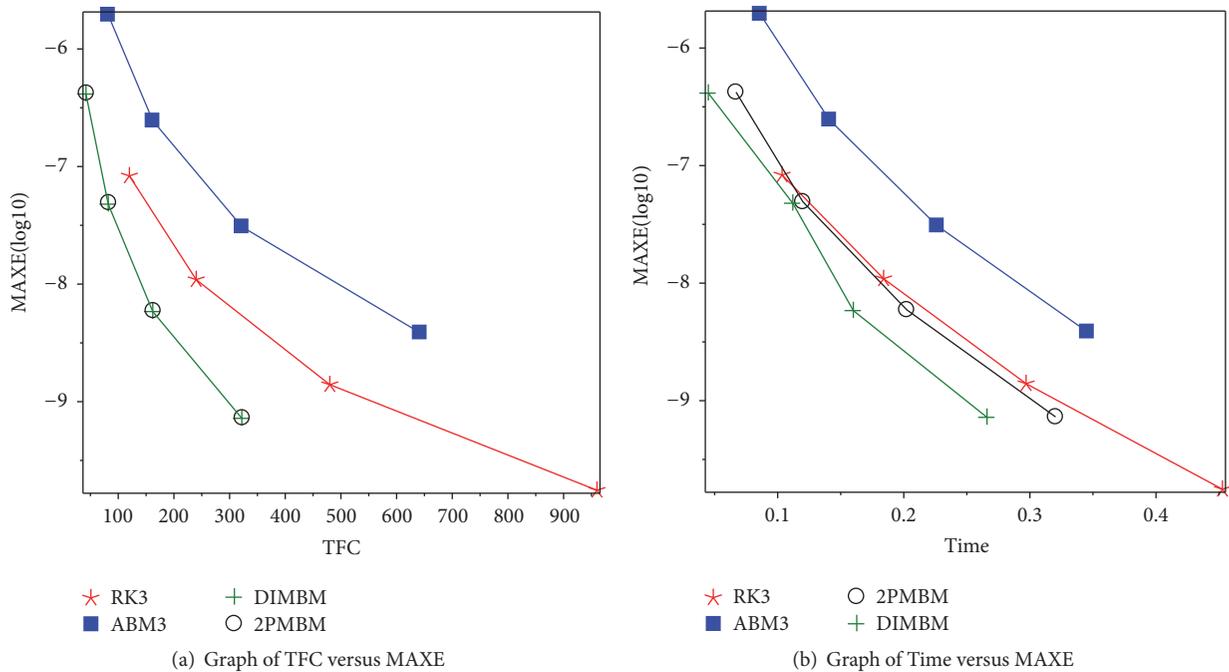


FIGURE 3: Graph of the numerical results when solving Example 1.

results will be obtained when the step size $h = 0.025, 0.0125, 0.00625, \text{ and } 0.003125$ for the case when $K(x, s) = 1$. The maximum error of DIMBM is comparable compared to 2PMBM at all tested h but the order of accuracy is the same or one order less compared to RK3 and ABM3. The performance of DIMBM is better in terms of total functions call and total number of steps compared to RK3 and ABM3.

In Tables 2, 3, and 4, the numerical results are solved using the proposed numerical method via composite Simpson's for the integral part when $K(x, s) \neq 1$. In Table 2, the maximum error of DIMBM is comparable compared to 2PMBM and

ABM3. The DIMBM manage to obtain less total functions call compared to ABM3 and 2PMBM. For the nonlinear Examples 3 and 4, we could observe that ABM3 and 2PMBM are expensive in terms of total functions call, respectively. Figures 3–6 display the numerical results of maximum error versus total functions call when solving the tested problems. This has shown the advantage of DIMBM in the form of a standard multistep method because the cost per step is cheaper and the numerical results are more accurate when the step size is reduced. In terms of timing, DIMBM gave faster results compared to ABM3 and 2PMBM.

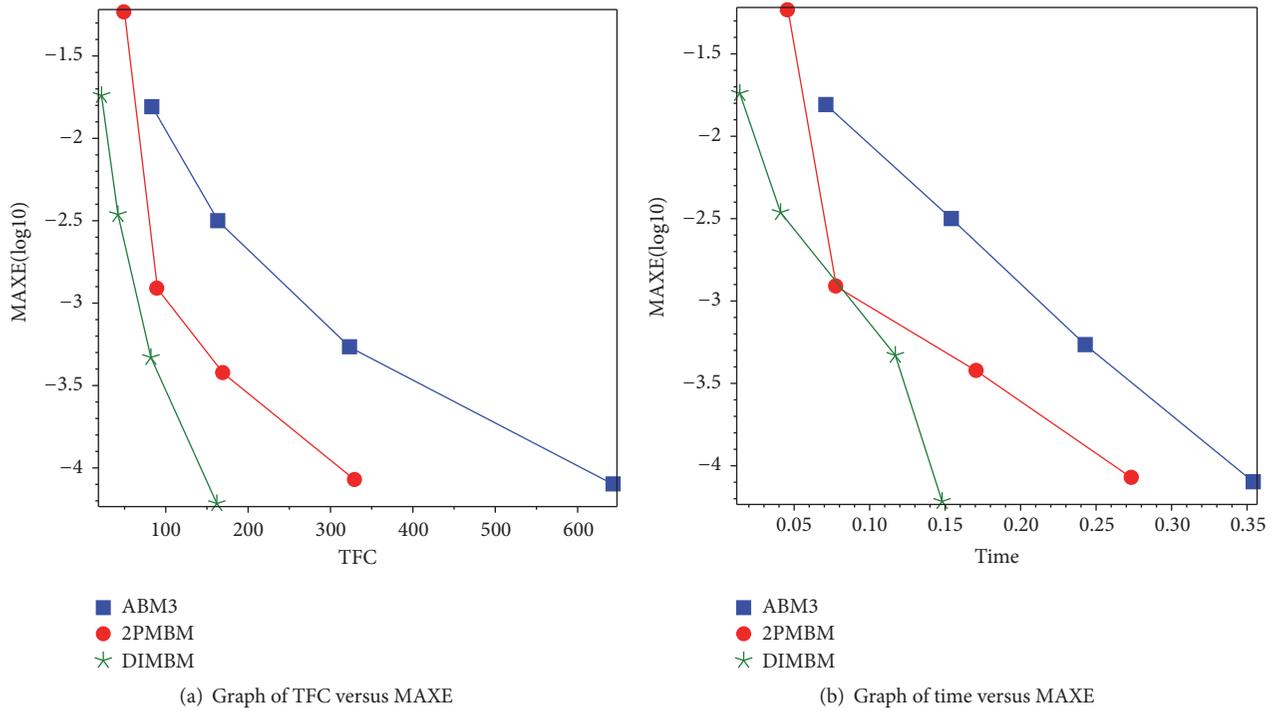


FIGURE 4: Graph of the numerical results when solving Example 2.

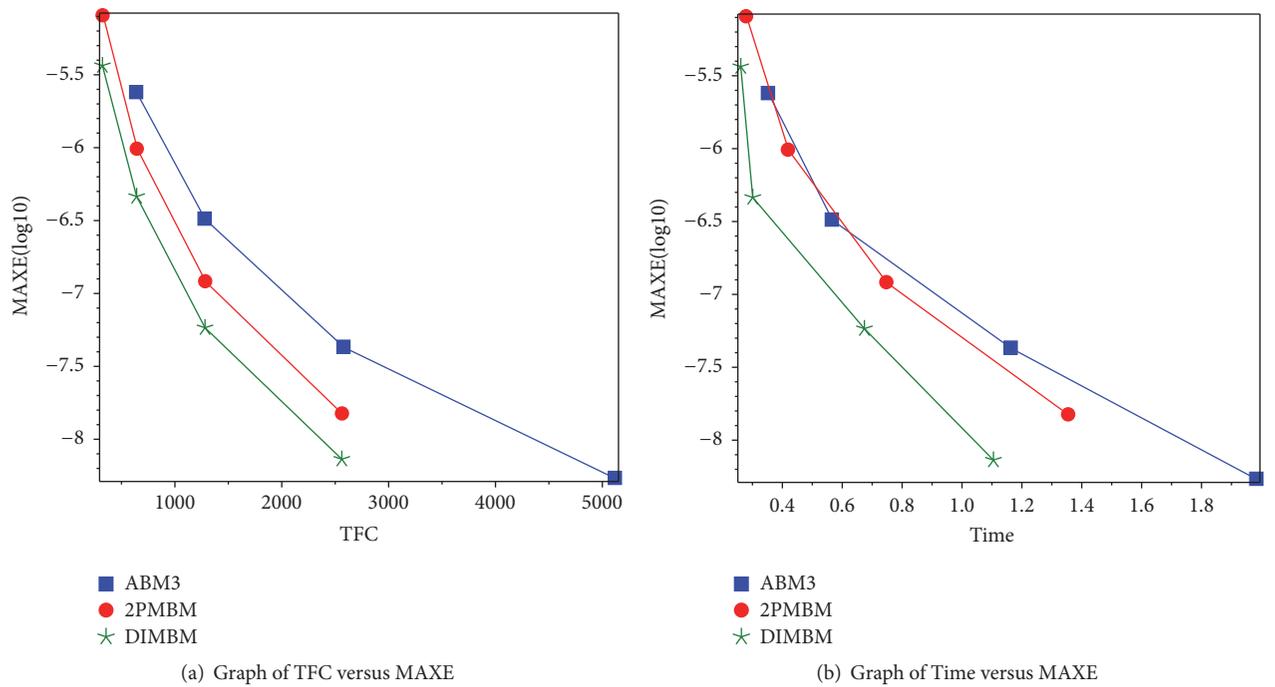


FIGURE 5: Graph of the numerical results when solving Example 3.

6. Conclusion

In this research, we proposed the diagonally implicit multi-step block method for solving linear and nonlinear Volterra

integrodifferential equations and comparisons were made with the existing method. Comparisons with existing method reveal that the diagonally implicit multistep block method is more efficient and cheaper.

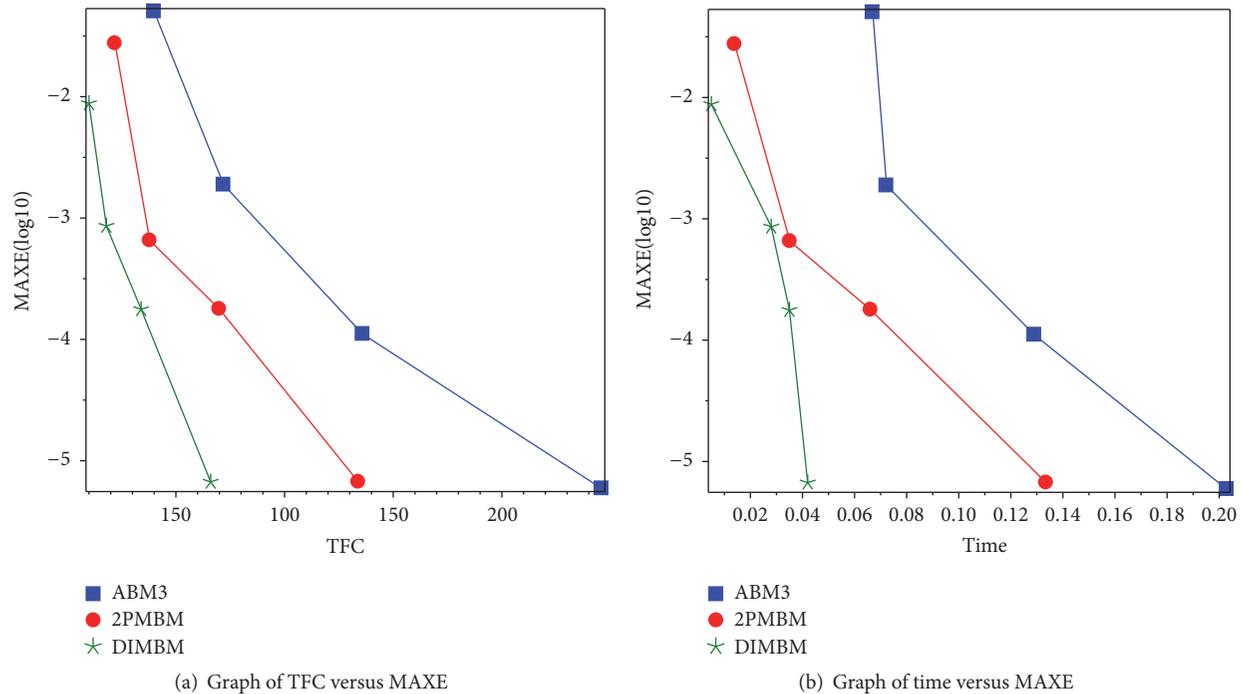


FIGURE 6: Graph of the numerical results when solving Example 4.

Conflicts of Interest

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

Acknowledgments

The authors gratefully acknowledged the financial support of Fundamental Research Grant Scheme (FRGS:5524973) and Graduate Research Fund (GRF) from Universiti Putra Malaysia.

References

- [1] J. T. Day, "Note on the Numerical Solution of Integro-Differential Equations," *The Computer Journal*, vol. 9, no. 4, pp. 394-395, 1967.
- [2] R. Saadati, B. Raftari, H. Adibi, S. M. Vaezpour, and S. Shakeri, "A comparison between the Variational Iteration method and Trapezoidal rule for solving linear integro-differential equations," *World Applied Sciences Journal*, vol. 4, no. 3, pp. 321-325, 2008.
- [3] B. Raftari, "Numerical solutions of the linear volterra integro-differential equations: homotopy perturbation method and finite Difference method," *World Applied Sciences Journal*, vol. 9, pp. 7-12, 2010.
- [4] A. Filiz, "Fourth-Order Robust Numerical Method for Integro-differential Equations," *Asian Journal of Fuzzy and Applied Mathematics*, vol. 1, no. 1, pp. 28-33, 2013.
- [5] A. Filiz, "Numerical Solution of a Non-linear Volterra Integro-differential Equation via Runge-Kutta-Verner Method," *International Journal of Scientific and Research Publications*, vol. 3, article 9, 2013.
- [6] F. Ishak and S. N. Ahmad, "Development of Extended Trapezoidal Method for Numerical Solution of Volterra Integro-Differential Equations," *International Journal of Mathematics, Computational, Physical, Electrical and Computer Engineering*, vol. 10, no. 11, article 52856, 2016.
- [7] N. A. B. Mohamed and Z. A. Majid, "One-step block method for solving Volterra integro-differential equations," *AIP Conference Proceedings*, vol. 1682, no. 1, Article ID 020018, 2015.
- [8] N. A. Mohamed and Z. A. Majid, "Multistep block method for solving volterra integro-differential equations," *Malaysian Journal of Mathematical Sciences*, vol. 10, pp. 33-48, 2016.
- [9] J. D. Lambert, *Computational Methods in Ordinary Differential Equations*, John Wiley & Sons, New York, NY, USA, 1973.
- [10] H. Brunner and J. D. Lambert, "Stability of numerical methods for volterra integro-differential equations," *Computing*, vol. 12, no. 1, pp. 75-89, 1974.
- [11] H. Chen and C. Zhang, "Boundary value methods for Volterra integral and integro-differential equations," *Applied Mathematics and Computation*, vol. 218, no. 6, pp. 2619-2630, 2011.
- [12] R. E. Shaw, "A parallel algorithm for nonlinear volterra integro-differential equations," in *Proceedings of the 2000 ACM Symposium on Applied Computing, SAC 2000*, pp. 86-88, Como, Italy, March 2000.
- [13] M. Dehghan and R. Salehi, "The numerical solution of the non-linear integro-differential equations based on the meshless method," *Journal of Computational and Applied Mathematics*, vol. 236, no. 9, pp. 2367-2377, 2012.



Hindawi

Submit your manuscripts at
www.hindawi.com

