Numerical method in solving neutral and retarded Volterra delay integro-differential equations

Nur Inshirah Naqiah Ismail¹, Zanariah Abdul Majid², Norazak Senu¹ and Nadihah Wahi²
¹Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
²Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
E-mail: am_zana@upm.edu.my

Abstract. The aim of this research is to produce accurate numerical results in solving neutral Volterra delay integro-differential equations (NVDIDE) and retarded Volterra delay integro-differential equations (RVDIDE) of constant type. A third-order explicit multistep block method is derived by applying the Taylor series. The consistency, zero stability, and convergence of the method are determined. The problems are solved by approximating two points simultaneously with constant step size. The delay arguments are approximated using previously calculated values while the integration part is approximated using the quadrature rule. The numerical results obtained have shown that the proposed explicit method is comparable with the other methods and is assumed to be reliable in solving NVDIDE and RVDIDE of constant type.

1. Introduction
In modelling phenomena in science and technology, such as mathematics, physics, population dynamics, complex systems, and biology, integro-differential equation (IDE) is of major importance. The problems listed are generally hard for researchers to solve analytically, thus a numerical method is required. One-step and multistep approximation methods are well-known among global methods for researchers in applied mathematics. Some expressions may be applied in this area of approximation to simplify algebraic computations. For example, to simplify the related algebraic equation, the operations of integration and differentiation can be used in the computations. The implementation of these operations also makes it easy to develop the applications. This manuscript aims to develop a multistep numerical method in solving an integro-differential equation with neutral and retarded Volterra delay type. The equation is rarely solved by multistep method, while Volterra integro-differential equation (VIDE) and Neutral delay differential equation (NDDE) are commonly solved by numerical methods.

2. Development of method
In 1981, [1] has investigated the Adams-Moulton method for the numerical solution of NVDIDE. A general convergence theorem is presented and the order of convergence of these methods is also examined. Other than that, [2] have also represented an algorithm based on divided difference of fully implicit one-step method. The NVDIDE are solved using predictor-corrector approach for...
non-stiff equation and using modified Newton method for stiff equation. Their proposed method is concluded to be quite accurate and reliable. Later, an explicit continuous Runge-Kutta (RK) method has been applied by [3] to solve NVDIDE. The advantage of the proposed method is that it is effective in delivering a continuous approximate solution with an associated global error. In 1999, a new polynomial collocation solution has been constructed by [4] where the parameters chosen are shifted Gauss points, shifted Radau II points, shifted Lobatto points and shifted $m-1$ Gauss points with $c_m = 1$. The existence of mild solutions of a nonlinear NVDIDE in a Banach space is proven by [5]. The results are obtained by using the Schaefer fixed point theorem.

After that, [6] have focused on the error behaviour of RK methods for nonlinear NVDIDE with constant delay. The convergence properties of RK methods with two classes of quadrature technique, compound quadrature rule and Pouzet type quadrature technique are investigated. Both methods were proved to be convergent under some conditions. The numerical stability of implicit RK methods is considered by [7]. The nonlinear stability conditions for the proposed methods are derived. Then, [8] has presented the approximate solution of linear NVDIDE of special type by using Galerkin’s method with Bernsien polynomial as a basic function. The numerical results obtained are very efficient. In 2013, [9] have provided the solution of NVDIDE based on spectral approach. The convergence properties of the spectral method are analyzed to approximate smooth solutions. A collocation method based on Laguerre polynomial is presented by [10] to solve pantograph equation of lower and higher order NVDIDE. The stability of symmetric boundary value methods (BVMs) for the linear NVDIDE is studied by [11]. Four families of symmetric BVMs are considered such as Extended Trapezoidal Rules of first (ETRs) and second kind (ETRs₂), Top Order Method (TOMs) and B-spline linear multistep method (BS). The methods are proved to be stable. The convergence of RK methods to solve the initial value problem of NVDIDE are studied by [12]. In the same year, an Euler polynomial of degree $m$ is presented by [13] to solve NVDIDE of pantograph delay type. The method gives more accurate solutions using fewer number of basis function. The effort required to implement the method is very low while the accuracy obtained is high.

The error analysis of one-leg methods for a class of nonlinear NVDIDE is given by [14]. It is proven that an A-stable one-leg method with an appropriate quadrature rule applied is convergent. A Galerkin-like approach has been presented by [15] to solve higher order NVDIDE and the results turned out to be accurate in addition to its simplicity. Then, [16] have considered the sinc collocation method for the numerical solution of pantograph equation. The method reduced RVDIDE into an explicit system of algebraic equation. The general linear multistep methods for general VIDE have been proposed by [17] for the classical stability, consistency and convergence theories of the methods. The methods and theories presented are also applicable to RVIDE. Then, a class of block boundary value method is constructed by [18] for the solution of linear NVDIDE with weakly singular kernels. The method is converged under suitable conditions on the data obtained. An effective numerical technique has also been introduced by [19] for finding the solutions of the first order NVDIDE with variable delays. The method is expressed by fundamental matrices, Laguerre polynomials with their matrix forms. The solution has been obtained by using the collocation points with regard to the reduced system of algebraic equation and Laguerre series. From the literature review, we can conclude that VDIDE has never been solved by applying multistep explicit multistep block method. Thus, we are taking this opportunity to derive a third order explicit multistep block method with constant step size technique as an approach to solve VDIDE with constant delay.

3. Methodology
In this section, the third order explicit multistep block numerical method (2PEB3) will be analysed in terms of formulation, order and convergence to solve NVDIDE with constant type.
The general form of NVDIDE with constant type is given as shown below:

\[ y'(x) = f(x, y(x)) + \int_{x-\tau}^{x} K(x, y(x), y(x-\tau)), \quad x \in [a, b], \]
\[ y(x) = \phi(x), \quad x \in [-\tau, 0], \]

(1)

while the general form for RVDIDE is given by:

\[ y'(x) = f(x, y(x)) + \int_{x-\tau}^{x} K(x, y(x), y(x-\tau)), \quad x \in [a, b], \]
\[ y(x) = \phi(x), \quad x \in [-\tau, 0], \]

(2)

where \( \tau \) is the delay term, \( y(x) = \phi(x) \) is the initial function given, \( K \) is the kernel while \( y(x-\tau) \) and \( y'(x-\tau) \) is the delay argument and its derivative. In this research, the second kind of Volterra integro-differential equation is being considered with the value of kernel is equal to 1.

The development of 2PEB3 is adapted based on predictor Adam-Bashforth mode. The proposed method will approximate the solution for NVDIDE and RVDIDE at two point simultaneously as shown in Figure 1 below:

![Figure 1. Two-point explicit multistep block method](image-url)

In Figure 1, it is clearly shown that each block contains two points where those points will be evaluated concurrently using the proposed 2PEB3 method. Referring to [20], the solutions for the first block will be applied as the initial values for the second block and the procedure will keep repeating for the next iterations in another blocks. Based on [21], the purpose of approximating two solutions at a time is to reduce the time taken as the calculation is lessened.

3.1. Formulation of method

A linear multistep method as stated by [22] is formulated as:

\[ \sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}, \]

(3)

where the terms \( y_{n+j} \) and \( f_{n+j} = y'_{n+j} \) will be expanded using Taylor series given by:

\[ L[y(x) : h] = C_0 y(x) + C_1 h y^{(1)}(x) + \ldots + C_p h^p y^{(p)}(x). \]

(4)

Then, the general form for the proposed 2PEB3 after letting \( k = 1 \) will be:

\[ y_{n+k} + \alpha_0 y_{n+(k-1)} = h \sum_{i=k-1}^{k+1} \beta_i y' \left[ x + (i - (k + 1))h \right], \]
\[ y_{n+(k+1)} + \alpha_0 y_{n+(k-1)} = h \sum_{i=k}^{k+2} \beta_i y' \left[ x + (i - (k + 1))h \right]. \]

(5)
By letting $\alpha_0 = -1$, expanding equation (5) using Taylor series mentioned in equation (4) and substituting them back in equation (5) will give:

$$\begin{align*}
[y(x) + 3hy'(x) + \frac{9}{2}h^2y''(x) + \frac{9}{2}h^3y'''(x)] = & \\
[y(x) + 2hy'(x) + 2h^2y''(x) + \frac{4}{3}h^3y'''(x)] + h\beta_0 & \\
y'(x) + h\beta_1 [y'(x) + hy''(x) + \frac{1}{2}h^2y'''(x)] + h\beta_2 & \\
y'(x) + 2hy''(x) + 2h^2y'''(x) &
\end{align*}$$

and,

$$\begin{align*}
[y(x_{n+1}) + 3hy'(x_{n+1}) + \frac{9}{2}h^2y''(x_{n+1}) + \frac{9}{2}h^3y'''(x_{n+1})] = & \\
[y(x_{n+1}) + hy'(x_{n+1}) + h^2y''(x_{n+1}) + \frac{1}{2}h^3y'''(x_{n+1})] + h\beta_1 & \\
y'(x_{n+1}) + h\beta_2 [y'(x_{n+1}) + hy''(x_{n+1}) + \frac{1}{2}h^2y'''(x_{n+1})] + h\beta_3 & \\
y'(x_{n+1}) + 2hy''(x_{n+1}) + 2h^2y'''(x_{n+1}) &
\end{align*}$$

collecting all terms from equations (6) and (7) will produce a two point fourth-order explicit multistep block method (2PEB3):

$$\begin{align*}
y_{n+1} = y_n + \frac{h}{12} \left[ 23f_n - 16f_{n-1} + 5f_{n-2} \right], & \\
y_{n+2} = y_n + \frac{h}{3} \left[ 19f_n - 20f_{n-1} - 7f_{n-2} \right].
\end{align*}$$

The formula produced above will be applied to solve NVDIDE and RVDIDE with constant delay type.

### 3.2. Order and error constant

In this section, the order of the proposed 2PEB3 will be studied. The 2PEB3 is being constructed in a matrix form:

$$\begin{align*}
\begin{bmatrix}
0 & 0 & -1 & 1 & 0 \\
0 & 0 & -1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
y_{n-2} \\
y_{n-1} \\
y_n \\
y_{n+1} \\
y_{n+2}
\end{bmatrix} = h
\begin{bmatrix}
\frac{5}{12} & -\frac{16}{3} & \frac{23}{3} & 0 & 0 \\
-\frac{12}{3} & \frac{23}{3} & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
f_{n-2} \\
f_{n-1} \\
f_n \\
f_{n+1} \\
f_{n+2}
\end{bmatrix},
\end{align*}$$

where,

$$\begin{align*}
\alpha_0 = \begin{bmatrix}
0 \\
0
\end{bmatrix}, & \quad \alpha_1 = \begin{bmatrix}
0 \\
0
\end{bmatrix}, & \quad \alpha_2 = \begin{bmatrix}
-1 \\
-1
\end{bmatrix}, & \quad \alpha_3 = \begin{bmatrix}
1 \\
0
\end{bmatrix}, & \quad \alpha_4 = \begin{bmatrix}
0 \\
1
\end{bmatrix}, \\
\beta_0 = \begin{bmatrix}
\frac{5}{12} \\
\frac{12}{3}
\end{bmatrix}, & \quad \beta_1 = \begin{bmatrix}
-\frac{16}{3} \\
-\frac{12}{3}
\end{bmatrix}, & \quad \beta_2 = \begin{bmatrix}
\frac{23}{3} \\
\frac{23}{3}
\end{bmatrix}, & \quad \beta_3 = \begin{bmatrix}
0 \\
0
\end{bmatrix}, & \quad \beta_4 = \begin{bmatrix}
0 \\
0
\end{bmatrix}.
\end{align*}$$
According to [22], the order of any linear multistep method can be determined by applying:

\[
C_0 = \sum_{j=0}^{k} \alpha_j
\]

\[
C_1 = \sum_{j=0}^{k} \left(j\alpha_j - \beta_j\right)
\]

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

The method is said to be of order \(p\) if \(C_0 = C_1 = \cdots = C_p = 0\) and \(C_{p+1}\) is called as an error constant where \(p = 2, 3, 4, \ldots\). Thus, 2PE4 is concluded to be of order four when:

\[
C_0 = \sum_{j=0}^{k} \alpha_j = \sum_{j=0}^{4} \alpha_j = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\]

\[
C_3 = \sum_{j=0}^{k} \left(\frac{j^3\alpha_j}{3!} - \frac{j^2\beta_j}{2!}\right) = \sum_{j=0}^{3} \left(\frac{j^3\alpha_j}{3!} - \frac{j^2\beta_j}{2!}\right)
= \left(\frac{0^2\alpha_0}{6} + \frac{1^3\alpha_1}{6} + \frac{2^3\alpha_2}{6} + \frac{3^3\alpha_3}{6} + \frac{4^3\alpha_4}{6}\right) - \left(\frac{0^2\beta_0}{2} + \frac{1^2\beta_1}{2} + \frac{2^2\beta_2}{2} + \frac{3^2\beta_3}{2} + \frac{4^2\beta_4}{2}\right)
= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\]

while the error constant is:

\[
C_4 = \sum_{j=0}^{k} \left(\frac{j^4\alpha_j}{4!} - \frac{j^3\beta_j}{3!}\right) = \sum_{j=0}^{4} \left(\frac{j^4\alpha_j}{4!} - \frac{j^3\beta_j}{3!}\right)
= \left(\frac{0^4\alpha_0}{24} + \frac{1^4\alpha_1}{24} + \frac{2^4\alpha_2}{24} + \frac{3^4\alpha_3}{24} + \frac{4^4\alpha_4}{24}\right) - \left(\frac{0^3\beta_0}{6} + \frac{1^3\beta_1}{6} + \frac{2^3\beta_2}{6} + \frac{3^3\beta_3}{6} + \frac{4^3\beta_4}{6}\right)
= \begin{bmatrix} 3 \\ \frac{3}{8} \end{bmatrix}.
\]

### 3.3. Convergence of method

In this section, the convergence of proposed 2PEB3 is explored. The effectiveness of applying the proposed method to any differential problems will finally be explained by investigating the convergence of any intended method. The method is said to be converged when the approximate values produced is closer to the exact values given as:

\[
\lim_{h \to 0} y_{n+1} = Y_{n+1}
\]

\[
\lim_{h \to 0} y_{n+2} = Y_{n+2}.
\]
where the approximate solution, \( y_{n+1} \) and \( y_{n+2} \) are denoted in equation (8) while the exact solution is given as follows:

\[
Y_{n+1} = y_n + \frac{h}{12} \left[ 23f_n - 16f_{n-1} + 5f_{n-2} \right] + \frac{3}{8} h^4 Y^{(4)}(\xi_n)
\]

\[
Y_{n+2} = y_n + \frac{h}{3} \left[ 19f_n - 20f_{n-1} - 7f_{n-2} \right] + \frac{8}{3} h^4 Y^{(4)}(\xi_n).
\]

By following Lipschitz condition below:

\[
|Y_n' - y_n'| = |f(x_n, Y_n, \int_{x-\tau}^x K(x_n, Y_n, Y_{m-n}, Y_{m-n}')) - f(x_n, y_n, \int_{x-\tau}^x K(x_n, y_n, y_{m-n}, y_{m-n}'))| \leq L|d_n|,
\]

the approximate value will be subtracted by the exact value as shown below:

\[
Y_{n+1} - y_{n+1} = \left[ Y_n + \frac{23}{12} h f_n - \frac{16}{12} h f_{n-1} + \frac{5}{12} h f_{n-2} \right] - \left[ y_n + \frac{23}{12} h f_n - \frac{16}{12} h f_{n-1} + \frac{5}{12} h f_{n-2} \right] + \frac{3}{8} h^4 Y^{(4)}(\xi_n)
\]

\[
Y_{n+2} - y_{n+2} = \left[ Y_n + \frac{19}{3} h f_n - \frac{20}{3} h f_{n-1} - \frac{7}{3} h f_{n-2} \right] - \left[ y_n + \frac{19}{3} h f_n - \frac{20}{3} h f_{n-1} - \frac{7}{3} h f_{n-2} \right] + \frac{8}{3} h^4 Y^{(4)}(\xi_n).
\]

Letting \( Y_{n+1} - y_{n+1} = d_{n+1}, Y_{n+2} - y_{n+2} = d_{n+2}, Y_n - y_n = d_n, \ldots \) will yield to:

\[
|d_{n+1}| \leq |d_n| + \frac{23}{12} h L|d_n| - \frac{16}{12} h L|d_{n-1}| + \frac{5}{12} h L|d_{n-2}| + \frac{3}{8} h^4 y^{(4)}(\xi_n)
\]

\[
|d_{n+2}| \leq |d_n| + \frac{19}{3} h L|d_n| - \frac{20}{3} h L|d_{n-1}| - \frac{7}{3} h L|d_{n-2}| + \frac{8}{3} h^4 Y^{(4)}(\xi_n).
\]

As \( h \) tends to zero, then:

\[
|d_{n+1}| \leq |d_n| \implies Y_{n+1} - y_{n+1} \leq Y_n - y_n \implies Y_{n+1} - Y_n \leq y_{n+1} - y_n
\]

\[
|d_{n+2}| \leq |d_n| \implies Y_{n+2} - y_{n+2} \leq Y_n - y_n \implies Y_{n+2} - Y_n \leq y_{n+2} - y_n,
\]

which shows that the approximate solution is equal to the exact solution. Thus, the convergence conditions in equation (11) are satisfied as \( |d_{n+1}| \leq |d_n| \) and \( |d_{n+2}| \leq |d_n| \) are satisfied. Hence, 2PEB3 is said to be converged.

4. Implementation of method

The constant type of NVDIDE and RVDIDE will be solved by applying the proposed 2PEB3 where two solutions will be produced concurrently at a single step in a block. A constant step size technique will be applied in order to find the solutions. In this research, Euler method has been chosen as an initial method to estimate the first three initial solutions for the NVDIDE and RVDIDE:

\[
y_{n+1} = y_n + hf\left(x_n, y_n, y_{n-m}, y'_{n-m}\right)
\]

There are few parts of implementation in solving the DDE and integral part of NVDIDE. The first part is, the delay argument \((x - \tau)\) is located first before solving the delay term \(y(x - \tau)\). The delay term, \(x - \tau\) will be evaluated using the initial function given in equation (1) and (2) if \( x \leq 0 \). If \( x > 0 \), then the Lagrange interpolating polynomial will be applied:

\[
P(x) = L_{n,0}(x)f(x_0) + \ldots + L_{n,n}(x)f(x_n) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x),
\]
where
\[ L_{n,k}(x) = \prod_{j=0}^{n} \frac{(x - x_i)}{(x_k - x_i)}, \quad k = 0, 1, \ldots, n. \]

Then, the second part is when the derivative of the delay term \( y'(x - \tau) \) is solved by applying backward and forward divided difference formulae as shown below:
\[
y'(x - \sigma_i) = \frac{y(x - \tau_i) - y(x - \tau_i - h)}{h},
\]
\[
y'(x - \sigma_i) = \frac{(y(x - \tau_i) + h) - y(x - \tau_i)}{h}, \tag{18}
\]

The forward difference formula may be applied first to compute the solution for the derivative of the delay argument if the starting formula given is a negative function. However, if the function given is positive, the backward difference formula is applied. The third part is where the integral will be solved by applying composite Simpson’s rule which is one of the Newton Cotes quadrature rule as shown below:
\[
\int_{a}^{b} f(x)dx = \frac{h}{3} \left[ f(a) + 2 \sum_{j=1}^{n-1} f(x_{2j}) + 4 \sum_{j=1}^{n-1} f(x_{2j-1}) + f(b) \right] - \frac{b - a}{12} h^4 f^{(4)}(\xi), \tag{19}
\]

The numerical results will be computed in C programme with constant step size technique. The results obtained will prove the efficiency of the proposed method.

5. Numerical results and discussions
Some numerical results for 2PEB3 have been presented to prove the applicability of the multistep block method in solving NVDIDE and RVDIDE. Two first order NVDIDE and one RVDIDE problems of constant type have been tested. The following notations are applied in the table:

| h     | Step size | 2PEB3 Two-point Explicit Multistep Block Numerical |
|-------|-----------|--------------------------------------------------|
| MTD   | Method    | AB3 Adam Bashforth Method (Order 3)               |
| FCN   | Total function calls | RK4 Runge-Kutta Method (Order 4) |
| TS    | Total Step | 5e-7 5 \times 10^{-7}                           |
| MAXE  | Maximum Error | Average Error                                   |

Example 1 (NVDIDE) [8]
\[
y'(x - 1) = 1 - \frac{x^3}{6} + \int_{0}^{x} (x - u)y(u)du, \quad x \in [0, 1]
\]

Exact solution:
\[ y(x) = x, \quad x \in [0, 1] \]
Table 1. Numerical results for Example 1

| h     | MTD | FCN | TS | MAXE       | AVERE       |
|-------|-----|-----|----|------------|-------------|
| 0.1   | 2PEB3 | 7   | 6  | 1.7713e-03 | 5.1449e-04 |
|       | AB3  | 11  | 10 | 1.7958e-03 | 4.1007e-04 |
|       | RK4  | 41  | 10 | 4.3693e-03 | 1.3981e-03 |
| 0.01  | 2PEB3 | 52  | 51 | 2.7088e-05 | 8.8373e-06 |
|       | AB3  | 101 | 100| 6.3428e-04 | 1.5615e-04 |
|       | RK4  | 401 | 100| 2.9689e-04 | 7.6773e-05 |
| 0.001 | 2PEB3 | 502 | 501| 2.8126e-07 | 9.2858e-08 |
|       | AB3  | 1001| 1000| 6.9350e-05 | 1.7256e-05 |
|       | RK4  | 4001| 1000| 2.8179e-05 | 7.0475e-06 |

Example 2 (RVDIDE) [23]

\[
y'(x) = y(x-1) + \int_{x-1}^{x} y(u)du, \quad x \in [0, 1]
\]

Exact solution:

\[
y(x) = e^x, \quad x \in [0, 1]
\]

Table 2. Numerical results for Example 2

| h     | MTD | FCN | TS | MAXE       | AVERE       |
|-------|-----|-----|----|------------|-------------|
| 0.1   | 2PEB3 | 7   | 6  | 1.7458e-02 | 1.2621e-02 |
|       | AB3  | 11  | 10 | 2.7959e-02 | 1.6173e-02 |
|       | RK4  | 41  | 10 | 2.4654e-04 | 8.1180e-05 |
| 0.01  | 2PEB3 | 52  | 51 | 1.5693e-04 | 1.1872e-04 |
|       | AB3  | 101 | 100| 1.7027e-03 | 7.2353e-04 |
|       | RK4  | 401 | 100| 1.4210e-02 | 8.0824e-03 |
| 0.001 | 2PEB3 | 502 | 501| 1.5458e-06 | 1.1764e-06 |
|       | AB3  | 1001| 1000| 1.6025e-04 | 6.3995e-05 |
|       | RK4  | 4001| 1000| 1.5543e-02 | 8.7845e-03 |

Example 3 (NVDIDE) [4]

\[
y'(x) = 2e^{1-x} - 3y(x) - 3 \int_{x-1}^{x} y(u)du - \int_{x-1}^{x} y'(u)du, \quad x \in [0, 1]
\]

Exact solution:

\[
y(x) = e^{-x}, \quad x \in [0, 1]
\]
Table 3. Numerical results for Example 3

| h    | MTD  | FCN | TS  | MAXE   | AVERE  |
|------|------|-----|-----|--------|--------|
| 0.1  | 2PEB3| 7   | 6   | 2.0730e-02 | 1.2952e-02 |
|      | AB3  | 11  | 10  | 1.8304e-02 | 1.0975e-02 |
|      | RK4  | 41  | 10  | 2.8936e-02 | 1.9479e-02 |
| 0.01 | 2PEB3| 52  | 51  | 1.2500e-03 | 9.1899e-04 |
|      | AB3  | 101 | 100 | 1.2362e-03 | 9.0937e-04 |
|      | RK4  | 401 | 100 | 5.8060e-03 | 4.7481e-03 |
| 0.001| 2PEB3| 502 | 501 | 1.2314e-04 | 8.9923e-05 |
|      | AB3  | 1001| 1000| 1.2299e-04 | 8.9822e-05 |
|      | RK4  | 4001| 1000| 8.9637e-03 | 7.2559e-03 |

From the numerical results in Table 1, the proposed 2PEB3 has produced comparable maximum and average error with the AB3. As the step size is decreasing, the numerical results for 2PEB3 become more accurate. The function call produced is also lesser than AB3 and RK4 since 2PEB3 is a block method and has proven to reduce the calculation needed to approximate the solutions of the problems.

The same conclusion applied for numerical results in Table 2 where 2PEB3 has produced lesser function calls with better results than the other comparison methods. The total steps are also lesser than ABM4 at \( h = 0.1, 0.01 \) and 0.001. Even though RK4 has produced better results at \( h = 0.1 \), the method is still concluded to be expensive since its function call is higher than 2PEB3 and AB3.

In Example 3, the maximum and average error are comparable at \( h = 0.1, 0.01 \) and 0.001 for both 2PEB3 and AB3. However, AB3 has produced more function call and total step than the propose 2PEB3. Thus, 2PEB3 is concluded to be more efficient. The proposed method is proven to be reliable in solving NVDIDE and RVDIDE with lesser function calls and total step size.

6. Conclusion

The proposed method is concluded to be efficient in solving NVDIDE and RVDIDE since the numerical results obtained have shown that the proposed explicit method is comparable with the other methods and is reliable in solving NVDIDE and RVDIDE of constant type.

Acknowledgement

The authors are mostly grateful for the financial support of Fundamental Research Grant (Project code: FRGS/1/2020/STG06/UPM/01/1) from Ministry of Higher Education Malaysia.

References

[1] Jackiewicz Z 1981 The numerical solution of volterra functional differential equations of neutral type Journal on Numerical Analysis 18(4) 615-626
[2] Jackiewicz Z and Lo E 1995 The numerical integration of neutral functional differential equations by fully implicit onestep methods Journal of Applied Mathematics and Mechanics 75(3) 207-221
[3] Enright W H and Hu M 1997 Continuous Runge-Kutta methods for neutral volterra integro-differential equations with delay Applied Numerical Mathematics 24(2-3) 175-190
[4] Horvat V 1999 On polynomial spline collocation methods for neutral volterra integro-differential equations with delay arguments Conference on Applied Mathematics and Computation 13 pp 113-128
[5] Dauer J P and Balachandran K 2000. Existence of solutions of nonlinear neutral integro differential equations in Banach spaces Journal of Mathematical Analysis and Applications 251(1) 93-105
[6] Wang W and Li S 2009 Convergence of runge-kutta methods for neutral volterra delay-integro-differential equations Frontiers of Mathematics in China 4(1) 195-216
[7] Wang W and Li D 2011 Stability analysis of runge-kutta methods for nonlinear neutral volterra delay-integro-differential equations Numerical Mathematics: Theory, Methods & Applications 4(4) pp 538-562
[8] Zaidan L I 2012 Solving linear delay volterra integro-differential equations by using galerkin’s method with bernstein polynomial *Journal of University of Babylon* **20**(5) 1405-1413

[9] Sedaghat S, Ordokhani Y and Dehghan M 2014 On spectral method for volterra functional integro-differential equations of neutral type *Numerical Functional Analysis and Optimization* **35**(2) 223-239

[10] Yuzbasi S 2014 Laguerre approach for solving pantograph-type volterra integro-differential equations *Applied Mathematics and Computation* **232** 1183-1199

[11] Zhao J, Fan Y and Xu Y 2015 An analysis of delay-dependent stability of symmetric boundary value methods for the linear neutral delay integro-differential equations with four parameters *Applied Mathematical Modelling* **39**(9)

[12] Wen L and Yu Y 2016 Convergence of rungekutta methods for neutral delay integro-differential equations *Applied Mathematics and Computation* **282** 84-96

[13] Mirzaee F, Bimesl S and Tohidi E 2016 A numerical framework for solving high-order pantograph-delay volterra integro-differential equations *Kuwait Journal of Science* **43**(1) pp 69-83

[14] Wen L and Zhou Y 2017. Convergence of one-leg methods for neutral delay integro-differential equations *Journal of Computational and Applied Mathematics* **317** 432-446

[15] Yuzbasi S and Karacayir M 2018 A numerical approach for solving high-order linear delay volterra integro-differential equations *International Journal of Computational Methods* **15**(05) 1850042

[16] Zhao J, Cao Y and Xu Y 2017 Sinc numerical solution for pantograph Volterra delay integro differential equation. *International Journal of Computer Mathematics* **94**(5) 853-865

[17] Bohner M and Tunc O 2020. Qualitative analysis of integro-differential equations with variable retardation *Discrete & Continuous Dynamical Systems B* **34**D05, K20, **45**J05

[18] Zhou Y and Stynes M 2021 Solving linear neutral Volterra integro differential equations with weakly singular kernels by block boundary value methods *Applied and Computational Mathematics Division*

[19] Gurbuz B 2021 A numerical scheme for the solution of neutral integro-differential equations including variable delay *Mathematical Sciences* **13** 1-9

[20] Ismail N I N, Majid Z A and Senu N 2020 solving neutral delay differential equation of pantograph type *Malaysian Journal of Mathematical Sciences* (ICoAIMS2019) **14**(S) 107-121

[21] Ismail N I N, Majid Z A and Senu N 2020 Hybrid multistep block method for solving neutral delay differential equations *Sains Malaysiana* **49**(4) 929-940

[22] Lambert J D 1973 *Computational methods in ordinary differential equations* (London: Wiley)

[23] Rihan F A, Doha E H, Hassan M I and Kamel N M 2009 Numerical treatments for volterra delay integro-differential equations *Computational Methods in Applied Mathematics* **9**(3) 292-318