Journal of Quality Measurement and Analysis *JQMA* 22(1) 2026, 1-21
e-ISSN: 2600-8602 *http://www.ukm.my/jqma*
https://doi.org/10.17576/jqma.2201.2026.01

TWO-POINT HYBRID MULTISTEP BLOCK METHOD FOR SOLVING FIRST-ORDER INITIAL VALUE PROBLEMS OF VOLTERRA INTEGRO-DIFFERENTIAL EQUATION WITH NEUTRAL DELAY

(*Kaedah Blok Dua Titik Berbilang Langkah Hibrid untuk Menyelesaikan Persamaan Kamiran-Pembezaan Masalah Nilai Awal Volterra dengan Kelengahan Neutral*)

NUR INSHIRAH NAQIAH ISMAIL* & ZANARIAH ABDUL MAJID

ABSTRACT

A generalized two-point hybrid multistep block method is proposed for solving Volterra integro-differential equations with neutral delay (NVDIDEs). Initial value problems (IVPs) for NVDIDEs are estimated synchronously at two points, along with an off-grid point forming a hybrid block scheme. Detailed strategies for solving the integral and differential parts are addressed. Convergence and stability analyses of the developed method are analyzed with the stability region derived from the stability polynomial. Numerical results are provided to evaluate the performance of the proposed method in comparison to existing non-block methods. The results are concluded to be comparable with the existing methods and reliable to solve NVDIDE.

Keywords: hybrid multistep block method; off-step method; NVDIDE

ABSTRAK

Satu kaedah blok dua titik berbilang langkah yang menyeluruh dicadangkan untuk menyelesaikan persamaan kamiran-pembezaan Volterra dengan kelengahan neutral (PKPVN). Masalah nilai awal (MNA) bagi PKPVN dianggarkan secara serentak pada dua titik, bersama dengan titik luar langkah yang membentuk skema blok hibrid. Strategi terperinci untuk menyelesaikan bahagian kamiran dan pembezaan dibincangkan. Penumpuan dan kestabilan kaedah yang dibangunkan dianalisis dengan kawasan kestabilan yang diperoleh daripada polinomial kestabilan. Keputusan numerik dibentangkan untuk menilai keberkesanan kaedah yang dicadangkan dengan kaedah bukan blok yang sedia ada. Keputusan menunjukkan bahawa kaedah yang dicadangkan adalah setanding dengan kaedah sedia ada dan boleh dipercayai untuk menyelesaikan NVDIDE.

Kata kunci: kaedah blok berbilang langkah hibrid; kaedah titik luar langkah; PKPVN

1. Introduction

Delay differential equations of the neutral type and integro-differential equations are of significant interest in various fields, particularly in science and engineering. These equations have gained growing importance as models for time delay problems in the engineering sector. Time delays often arise in feedback loops involving sensors and actuators. Such transmission systems are commonly found in communication technologies, including remote control of cars, operation of large construction cranes, trains, trams, and TVs, among others, Kuang (1993). Neutral-type equations govern some well-known biological processes such as growth, death, and birth. Different names have been given by many authors for delay-related problems for example, time-delay systems, equations with deviating arguments, ordinary differential equations (ODEs) with time lags, and retarded ODEs. In this field of approximation, operational integration and differentiation are applied in computations to simplify the associated equation, Mirzaee *et al.* (2016). The original form of the delay Volterra integro-differential equation (DVIDE) is as follows,

$$y'(x) = f\big(x, y(x)\big) + \int_x^{\alpha} K\big(x, y(x), y(\alpha)\big), \quad x \in [a, b], \tag{1}$$
$$y(x) = \phi(x),$$

where (1) is also known as retarded delay Volterra integro-differential equation (RVDIDE). The term α refers to its delay and $y(\alpha)$ as its delay solutions. α can be any type of delay for example, a constant, proportional or mixed delay types. This paper aims to analyze the DVIDE class using the neutral type. The general form of this class is shown below,

$$y'(x) = f\big(x, y(x)\big) + \int_x^{\alpha} K\big(x, y(x), y(\alpha), y'(\alpha)\big), \quad x \in [a, b], \tag{2}$$
$$y(x) = \phi(x),$$

The equations are cited from Ismail and Majid (2024). The initial function given is required to determine the unique solution, $y_n(x)$. The difference between RVDIDE to NVDIDE is that NVDIDE has a delay derivative. Within the framework of this research, NVDIDE will be presented in two cases: case A (constant delay) and case B (proportional delay). A convolution integral over the time delay function is used in the integral term of these equations, which are made up of both differential equations and integral equations. These equations are extremely non- linear and difficult to solve since they contain both differential and integral elements. Solving NVDIDE analytically is often impossible, and numerical methods are used instead. However, numerical methods can also be challenging due to the highly non- linear nature of these equations. Consequently, there is ongoing research to develop more efficient and accurate numerical methods to solve NVDIDEs.

2. Development of Hybrid Multistep Block Method

In 1997, Enright and Hu presented a method to resolve NVDIDE with constant delay using continuous Runge-Kutta methods. The authors first discretized the delay term using a collocation method, then used continuous Runge-Kutta methods to discretize the remaining terms. It is demonstrated that the resulting method is converged and provided precise numerical solutions. The authors also presented numerical examples to demonstrate the effectiveness of their method. They concluded that their approach provides a promising alternative to existing numerical methods for solving neutral Volterra integro-differential equations with delay. Later, to resolve NVDIDE with case A, Horvat (1999) suggested a polynomial spline collocation technique. The method approximated the solution using polynomial splines, then collocated the resultant equations at a set of points to obtain a system of algebraic equations. The author demonstrated that the method is convergent and achieves higher accuracy than other existing methods for solving such equations. In 2009, Rihan *et al.* examined a variety of numerical methods to solve RVDIDE with constant delay. The authors provided insight into several existing numerical methods, including collocation, iterative and spectral methods, and discussed their benefits and limitations. They also suggested a new numerical method that implies the combination of a collocation method with a finite difference scheme. The proposed approach was consistent and delivered accurate numerical solutions. The paper also included numerical examples to demonstrate the efficacy of the proposed methodology. The authors concluded that there is no one-size-fits-all approach to solving DVIDE and that the choice of method depends on the particular problem to be solved.

They suggested that researchers continue to develop and refine numerical methods for these equations to improve their precision and effectiveness.

A few years later, Yüzbaşı (2014) presented a new numerical method to solve NVDIDEs with case B (also known as the pantograph equation) by using Laguerre polynomials. The author initially derived a general formula for the pantograph equation, then showed that this formula can be expressed as a series of Laguerre polynomials. The coefficients of this series can be obtained using a recursive algorithm, and the resulting series provided an approximate solution to the original equation. The author showed that the method is convergent and delivered precise numerical solutions, and compared the method with other existing numerical methods. The paper also included numerical examples to demonstrate the efficacy of the proposed methodology. Then, Reutskiy (2016) proposed a numerical method for solving NVDIDE with proportional delay. The methodology involves the application of back substitution to a system of linear algebraic equations derived from the original integro-differential equation. The author shows that the methodology is consistent and provides precise numerical solutions. The article also includes numerical examples to demonstrate the efficacy of the proposed approach.

Recently, Ismail *et al.* (2021) generated accurate numerical results in the resolution of NDIVDE and RDVIDDE of constant delay. An explicit third-order multistep block method is derived using the Taylor series. The method's consistency, zero stability, and are established. Problems are solved by simultaneously considering two points using a constant step size. The delay arguments are approximated with previously computed values, while the integration part is approximated using the quadrature rule. The numerical results demonstrate that the proposed explicit method is comparable to other methods and is considered reliable for solving constant type NVDIDE and RVDIDE. In this study, an implicit hybrid multistep block approach is derived, which offers a distinct contribution compared to the explicit multistep block method introduced by Ismail *et al.* (2021). This is the primary distinction between the current research and their earlier work. The choice of using an implicit method in the hybrid multistep block approach, as opposed to the explicit method used by Ismail *et al.* offers certain advantages such as improved stability and better handling of stiff equations, which is often a challenge in solving delay differential equations. From Ismail and Majid (2024), a hybrid multistep block method named 2OBM4 for solving NVDIDE with constant delay is proposed. The scheme uses divided differences and composite Simpson's rule for discretization. It was shown to be consistent, zero-stable, and convergent, with numerical examples confirming its superior performance compared to traditional block methods. A recent work from Yu (2025) adapted general linear schemes for nonlinear NDVIDEs. The paper derives stability and asymptotic stability conditions for algebraically stable implementations and provides numerical tests demonstrating high accuracy and robustness

Moreover, a thorough review of existing literature reveals that, to date, no other researchers have employed a hybrid multistep block method to solve Nonlinear Delay Differential Equations (NVDIDEs). The gap in the existing approaches highlights the novelty of the current work. By developing this new method, the study addresses a significant challenge in solving NVDIDEs, which are known for their complexity due to the time delays in their formulations. Therefore, the approach taken in this research is motivated by the need to explore more efficient and robust numerical methods for solving these types of equations. This new method aims to fill a gap in the current body of research by providing an alternative solution strategy that has the potential to improve both the accuracy and computational efficiency when solving NVDIDEs.

3. Derivation of Method

Consider the first-order NVDIDE, denoted in (2) where the interval $[a, b]$ is subdivided into a series of blocks. Each block contains four points including one off-step point for the implicit hybrid method with constant step size,

$$h = \frac{(b-a)}{N}$$

where

$$a = x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}, \dots, x_N = b$$

as shown in Figure 1 below,

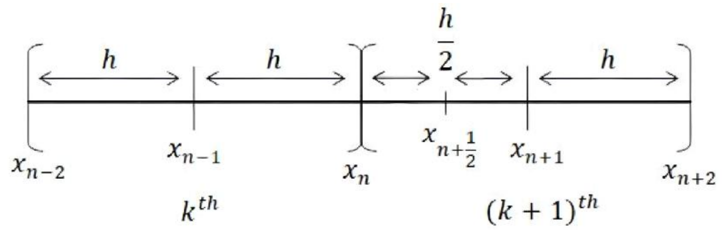

Figure 1: The two-point blocks with one off-step point.

Figure 1 above is taken from Ismail *et al.* (2020). According to Lambert (1973), multistep method with an off-step point is also known as a hybrid multistep method given by,

$$L[y(x); h] = \sum_{j=0}^{k} [\alpha_j y(x + jh) - h\beta_j y'(x + jh)] - h\beta_v y'(x + vh). \tag{3}$$

where α_0 and β_0 are both non-zero, $j\{0,1, \dots, k\}, v\ \{0,1, \dots, k\}$ as mentioned in Lambert (1973). Formulating a hybrid multistep block method involves Taylor series polynomial,

$$L[y(x); h] = C_0 y(x) + C_1 h y^{(1)}(x) + \dots + C_q h^q y^{(q)}(x), \tag{4}$$

to extend $\mathrm{y}(x + jh)$ and $y'(x + jh)$ by carefully selecting the coefficients of each method and determining the appropriate order and number of steps to be used. Hence, the approximation of y_{n+1}^c and y_{n+2}^c in the corrector form are obtained as follows,

$$\begin{aligned} y_{n+4} + \alpha_0 y_{n+3} &= h\left[\sum_{i=1}^{4} \beta_i\, y'(x + ih) + \beta_{\frac{7}{2}} y'(x + vh)\right], \\ y_{n+5} + \alpha_0 y_{n+3} &= h\left[\sum_{i=2}^{5} \beta_i\, y'(x + ih) + \beta_{\frac{7}{2}} y'(x + vh)\right]. \end{aligned} \tag{5}$$

The subscript $\frac{7}{2}$ (i.e., 3.5) indicates an off-step point, meaning the method includes evaluations between regular mesh points $(x + 3.5h)$. these hybrid-type multistep methods aim to improve accuracy or reduce local truncation error by using fractional-step evaluations. The term β_i is for integer-step evaluations $y'(x+ih)$, where $i = 1,2,3,4,5$. Letting $\alpha_0 = 1$, where α_0 is a coefficient associated with the previous value y_{n+3}. By expanding individual terms in (5) using Taylor series expansion in (4). The position of points usually depend on the order of the method derived. In multistep methods, such as Adams-Bashforth or backward differentiation formulas (BDF), the goal is to use multiple previous points to approximate solution at a new point. By introducing off-step points, which may not directly correspond to a previously computed value (such as taking a value at a midpoint or different intermediate time), the accuracy of the method can be improved by increasing the order of the approximation. When selecting an off-step point, a new way to make use of past information is explored in a more computationally efficient manner. By picking points that are strategically offset or placed differently, redundant evaluations can be avoided. Additionally, using off-step points can reduce the number of steps needed to reach an accurate solution, minimizing the computational workload while maintaining the desired precision. The difference between the derived method and existing multistep methods could lie in how the off-step points are selected and utilized. While traditional methods tend to use adjacent points, the derived method may choose more strategically located points. After substituting the Taylor terms into the above equations, a series of equations below should be equated to obtained β_i coefficients, for y_{n+1}^c,

$$\begin{aligned}
1 &= \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_{\frac{7}{2}},\\
\frac{7}{2} &= \beta_1 + 2\beta_2 + 3\beta_3 + 4\beta_4 + \frac{7}{2}\beta_{\frac{7}{2}},\\
\frac{37}{6} &= \frac{1}{2}\beta_1 + 2\beta_2 + \frac{9}{2}\beta_3 + 8\beta_4 \\
&\qquad + \frac{49}{8}\beta_{\frac{7}{2}},\\
\frac{175}{24} &= \frac{1}{6}\beta_1 + \frac{4}{3}\beta_2 + \frac{9}{2}\beta_3 + \frac{32}{3}\beta_4 + \frac{343}{48}\beta_{\frac{7}{2}},\\
\frac{781}{120} &= \frac{1}{24}\beta_1 + \frac{2}{3}\beta_2 + \frac{27}{8}\beta_3 + \frac{32}{3}\beta_4 + \frac{2401}{384}\beta_{\frac{7}{2}},
\end{aligned} \tag{6}$$

and y_{n+2}^c,

$$\begin{aligned}
2 &= \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_{\frac{5}{2}},\\
6 &= \beta_2 + 2\beta_3 + 3\beta_4 + 4\beta_5 + \frac{5}{2}\beta_{\frac{5}{2}},\\
\frac{28}{3} &= \frac{1}{2}\beta_2 + 2\beta_3 + \frac{9}{2}\beta_4 + 8\beta_5 + \frac{25}{8}\beta_{\frac{5}{2}},\\
10 &= \frac{1}{6}\beta_2 + \frac{4}{3}\beta_3 + \frac{9}{2}\beta_4 + \frac{32}{3}\beta_5 + \frac{125}{48}\beta_{\frac{5}{2}},\\
\frac{124}{15} &= \frac{1}{24}\beta_2 + \frac{2}{3}\beta_3 + \frac{27}{8}\beta_4 + \frac{32}{3}\beta_5 + \frac{625}{384}\beta_{\frac{5}{2}},
\end{aligned} \tag{7}$$

subsequently. Then, by letting $n = n - 3$, the corrector formulae for the hybrid block method are as follows,

$$y_{n+1}^{c} = y_n + \frac{h}{1800}\left(295f_{n+1} + 1216f_{n+\frac{1}{2}} + 285f_n + 5f_{n-1} - 6f_{n-2}\right), \tag{8}$$
$$y_{n+2}^{c} = y_n + \frac{h}{135}\left(41f_{n+2} + 216f_{n+1} - 64f_{n+\frac{1}{2}} + 81f_n - 4f_{n-1}\right).$$

The same techniques are used to derive the predictor formulae where the general forms for both formulae are given below,

$$\begin{aligned} y_{n+\frac{7}{2}} + \alpha_0 y_{n+3} &= h\sum_{i=0}^{4} \beta_i\, y'(x+ih), \\ y_{n+5} + \alpha_0 y_{n+4} &= h\sum_{i=0}^{4} \beta_i\, y'(x+ih), \\ y_{n+7} + \alpha_0 y_{n+5} &= h\sum_{i=1}^{5} \beta_i\, y'(x+ih), \end{aligned} \tag{9}$$

for $y_{n+\frac{1}{2}}^{p}, y_{n+1}^{p}$ and y_{n+2}^{p} respectively. The predictor formulae are derived in the same manner as the corrector formulae that included the set of points $\{x_n, x_{n-1}, x_{n-2}, x_{n-3}, x_{n-4}\}$. The generated predictor-corrector hybrid block method yields as follows,

$$\begin{aligned} y_{n+1}^{p} &= y_n + \frac{h}{720}(1901f_n - 2774f_{n-1} + 2616f_{n-2} - 1274f_{n-3} + 251f_{n-4}), \\ y_{n+\frac{1}{2}}^{p} &= y_n + \frac{h}{5760}(314f_n + 3199f_{n-1} - 921f_{n-2} + 349f_{n-3} - 61f_{n-4}), \\ y_{n+2}^{p} &= y_n + \frac{h}{90}(1079f_n - 2396f_{n-1} + 2544f_{n-2} - 1316f_{n-3} + 269f_{n-4}), \\ y_{n+1}^{c} &= y_n + \frac{h}{1800}\left(295f_{n+1} + 1216f_{n+\frac{1}{2}} + 285f_n + 5f_{n-1} - f_{n-2}\right), \\ y_{n+2}^{c} &= y_n + \frac{h}{135}\left(41f_{n+2} + 216f_{n+1} - 64f_{n+\frac{1}{2}} + 81f_n - 4f_{n-1}\right), \end{aligned} \tag{10}$$

where the method is named a multistep block method with an off-step point, 1OBM5 since it is of order five which will be shown in the next section.

4. Order of Method

The order of the method can be obtained by quoting Jator (2010),

Definition 4.1 (Jator 2010) *The proposed hybrid multistep block method is said to be of order q if, $C_0 = C_1 = \cdots = C_q = 0$ and $C_{q+1} \neq 0$ is called an error constant where $q = 2,3, \ldots$*

$$C_0 = \sum_{j=0}^{k} \alpha_j$$

$$C_1 = \sum_{j=0}^{k} j\alpha_j - \sum_{j=0}^{k} \beta_j - \sum_{j=1}^{2} \beta_{v_j} \tag{11}$$
$$\vdots$$
$$C_q = \frac{1}{q!}\left[\sum_{j=1}^{k} j^q \alpha_j - q\left[\sum_{j=1}^{k} j^{(q-1)}\beta_j + \sum_{j=1}^{2} v_j^{(q-1)}\beta_{v_j}\right]\right].$$

where C_{q+1} is the error constant.

By letting $k = 4$, the corrector formulae for 1OBM5 in Eq. (10) are rewritten in matrix form as displayed below,

$$\begin{bmatrix} 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} y_{i-2} \\ y_{i-1} \\ y_i \\ y_{i+1} \\ y_{i+2} \end{bmatrix} =$$
$$h\begin{bmatrix} -\frac{1}{1800} & \frac{5}{1800} & \frac{285}{1800} & \frac{295}{1800} & 0 \\ 0 & -\frac{4}{135} & \frac{81}{135} & \frac{216}{135} & \frac{41}{135} \end{bmatrix}\begin{bmatrix} f_{i-2} \\ f_{i-1} \\ f_i \\ f_{i+1} \\ f_{i+2} \end{bmatrix} + h\begin{bmatrix} \frac{1216}{1800} \\ -\frac{64}{135} \end{bmatrix} f_{i+\frac{1}{2}} \tag{12}$$

Then, the related linear difference operator for the corrector formulae in matrices generates the vector columns as demonstrate below,

$$\alpha_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \alpha_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \alpha_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \alpha_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \alpha_4 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$
$$\beta_0 = \begin{bmatrix} -\frac{1}{1800} \\ 0 \end{bmatrix}, \beta_1 = \begin{bmatrix} \frac{5}{1800} \\ -\frac{4}{135} \end{bmatrix}, \beta_2 = \begin{bmatrix} \frac{285}{1800} \\ \frac{81}{135} \end{bmatrix}, \beta_3 = \begin{bmatrix} \frac{295}{1800} \\ \frac{216}{135} \end{bmatrix}, \beta_4 = \begin{bmatrix} 0 \\ \frac{41}{135} \end{bmatrix},$$
$$\beta_{v_1} = \begin{bmatrix} \frac{1216}{1800} \\ -\frac{64}{135} \end{bmatrix}$$

In this work, the summation index α ranges from 0 to 4, resulting in five terms in total. By referring to Definition 4.1, the respective coefficients are applied in calculating the order, yielding the following results,

$$\begin{aligned} C_0 &= \sum_{j=0}^{4} \alpha_j = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \end{aligned}$$

$$
\begin{aligned}
C_1 &= \sum_{j=0}^{4} j\alpha_j - \sum_{j=0}^{4} \beta_j - \beta_{v_1} \\
&= (0\alpha_0 + 1\alpha_1 + 2\alpha_2 + 3\alpha_3 + 4\alpha_4) - (\beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4) - \beta_{v_1} \\
&= \left(0 \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 1 \cdot \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2 \cdot \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) \\
&\quad - \left(\begin{bmatrix} -\frac{1}{1800} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{5}{1800} \\ -\frac{4}{135} \end{bmatrix} + \begin{bmatrix} \frac{285}{1800} \\ \frac{81}{135} \end{bmatrix} + \begin{bmatrix} \frac{295}{1800} \\ \frac{216}{135} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{41}{135} \end{bmatrix}\right) - \begin{bmatrix} \frac{1216}{1800} \\ -\frac{64}{135} \end{bmatrix} \\
&= \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} \frac{73}{225} \\ \frac{334}{135} \end{bmatrix} - \begin{bmatrix} \frac{1216}{1800} \\ -\frac{64}{135} \end{bmatrix} \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
C_2 &= \frac{1}{2!}\left[\sum_{j=1}^{4} j^2\alpha_j - 2\left(\sum_{j=1}^{4} j\beta_j - v_1\beta_{v_1}\right)\right] \\
&= \frac{1}{2}\left[(1^2\alpha_1 + 2^2\alpha_2 + 3^2\alpha_3 + 4^2\alpha_4) - 2\left((1\beta_1 + 2\beta_2 + 3\beta_3 + 4\beta_4) + \frac{5}{2}\beta_{v_1}\right)\right] \\
&= \frac{1}{2}\left[\left(1^2 \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2^2 \begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3^2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4^2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)\right. \\
&\quad \left. -2\left(\left(1\begin{bmatrix} \frac{5}{1800} \\ -\frac{4}{135} \end{bmatrix} + 2\begin{bmatrix} \frac{285}{1800} \\ \frac{81}{135} \end{bmatrix} + 3\begin{bmatrix} \frac{295}{1800} \\ \frac{216}{135} \end{bmatrix} + 4\begin{bmatrix} 0 \\ \frac{41}{135} \end{bmatrix}\right) + \frac{5}{2}\begin{bmatrix} \frac{1216}{1800} \\ -\frac{64}{135} \end{bmatrix}\right)\right] \\
&= \frac{1}{2}\left[\begin{bmatrix} 5 \\ 12 \end{bmatrix} - 2\left(\begin{bmatrix} \frac{73}{90} \\ \frac{194}{27} \end{bmatrix} + \begin{bmatrix} \frac{76}{45} \\ -\frac{32}{27} \end{bmatrix}\right)\right] \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\end{aligned}
$$

$$
\begin{aligned}
C_3 &= \frac{1}{3!}\left[\sum_{j=1}^{4} j^3\alpha_j - 3\left(\sum_{j=1}^{4} j^2\beta_j + \mathrm{v}_1^2\beta_{v_1}\right)\right] \\
&= \frac{1}{6}\left[(1^3\alpha_1 + 2^3\alpha_2 + 3^3\alpha_3 + 4^3\alpha_4)\right. \\
&\qquad\qquad \left. - 3\left((1^2\beta_1 + 2^2\beta_2 + 3^2\beta_3 + 4^2\beta_4) + \left(\frac{5}{2}\right)^2 \beta_{v_1}\right)\right] \\
&\vdots \\
&= \begin{bmatrix} 0 \\ 0 \end{bmatrix},
\end{aligned}
$$

$$C_4 = \frac{1}{4!}\left[\sum_{j=1}^{4} j^4\alpha_j - 4\left(\sum_{j=1}^{4} j^3\beta_j + \mathrm{v}_1^3\beta_{v_1}\right)\right]$$
$$= \frac{1}{24}\left[(1^4\alpha_1 + 2^4\alpha_2 + 3^4\alpha_3 + 4^4\alpha_4)\right.$$
$$\left. - 4\left((1^3\beta_1 + 2^3\beta_2 + 3^3\beta_3 + 4^3\beta_4) + \left(\frac{5}{2}\right)^3\beta_{v_1}\right)\right]$$
$$\vdots$$
$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$C_5 = \frac{1}{5!}\left[\sum_{j=1}^{4} j^5\alpha_j - 5\left(\sum_{j=1}^{4} j^4\beta_j + \mathrm{v}_1^4\beta_{v_1}\right)\right]$$
$$= \frac{1}{120}\left[(1^5\alpha_1 + 2^5\alpha_2 + 3^5\alpha_3 + 4^5\alpha_4)\right.$$
$$\left. -5\left((1^4\beta_1 + 2^4\beta_2 + 3^4\beta_3 + 4^4\beta_4) + \left(\frac{5}{2}\right)^4\beta_{v_1}\right)\right]$$
$$\vdots$$
$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$C_6 = \frac{1}{6!}\left[\sum_{j=1}^{4} j^6\alpha_j - 6\left(\sum_{j=1}^{4} j^5\beta_j + \mathrm{v}_1^5\beta_{v_1}\right)\right]$$
$$= \frac{1}{720}\left[(1^6\alpha_1 + 2^6\alpha_2 + 3^6\alpha_3 + 4^6\alpha_4)\right.$$
$$\left. -6\left((1^5\beta_1 + 2^5\beta_2 + 3^5\beta_3 + 4^5\beta_4) + \left(\tfrac{5}{2}\right)^5\beta_{v_1}\right)\right]$$
$$\vdots$$
$$= \begin{bmatrix} -\dfrac{1}{3600} \\ -\dfrac{1}{180} \end{bmatrix}.$$

These calculations have concluded that 1OBM5 is of order five where, $C_0 = C_1 = C_2 = C_3 = C_4 = C_5 = [0,0]^T$ with error constant, $C_{q+1} = C_6 = \left[-\frac{1}{3600}, -\frac{1}{180}\right]^T \neq [0,0]^T$. By applying the same steps to compute the order of 1OBM5's predictor formulae, the error constant obtained is, $C_{q+1} = C_6 = \left[\frac{23153}{288}, \frac{26897}{90}, -\frac{65}{9216}\right]^T \neq [0,0,0]^T$ where the predictors are of order five.

5. Consistency and Zero-Stability of Method

Based on Jator (2018), the proposed method, 1OBM5 is consistent since the order analysis procedure revealed that all of the proposed methods have an order larger than one,

Definition 5.1 (Jator 2018) *The block linear multistep method (LMM) is said to be consistently provided that the order of the method is at least one.*

Referring to Lambert (1973), the first and second conditions for consistency denoted in Eq. (13), are also satisfied given as follows,

Definition 5.2 (Lambert 1973) *The LMM is said to be consistent if it has order $p \geq 1$ and the method is also consistent if and only if,*

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

ii. $\sum_{j=0}^{k} j\alpha_j = \sum_{j=0}^{k} \beta_j.$ (13)

Zero stability can be demonstrated by generating the roots of the first characteristic polynomial idealized by Fatunla (1995),

Definition 5.3 (Fatunla 1995) *The LMM is said to be zero-stable if no root of the first characteristic polynomial in (14) has modulus greater than one,*

$$\rho(\xi) = det\left[\sum_{j=0}^{k} \alpha_j \, \xi^{k-j}\right] = 0, \tag{14}$$

for the corrector formulae below,

$$det\left[\sum_{j=0}^{k} \alpha_j \, \xi^{j}\right] = 0,$$

where,

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

According to Definition 5.3, the first characteristic polynomial is analyzed as follows,

$$\begin{aligned}
\rho(\xi) &= det\left[\sum_{j=0}^{k} \alpha_j \, \xi^{j}\right] = det\left[\sum_{j=0}^{4} \alpha_j \, \xi^{j}\right],\\
&= det[\alpha_0\xi^0 + \alpha_1\xi^1 + \alpha_2\xi^2 + \alpha_3\xi^3 + \alpha_4\xi^4],\\
&= det\left[\begin{bmatrix}0\\0\end{bmatrix}\xi^0 + \begin{bmatrix}0\\0\end{bmatrix}\xi^1 + \begin{bmatrix}-1\\-1\end{bmatrix}\xi^2 + \begin{bmatrix}1\\0\end{bmatrix}\xi^3 + \begin{bmatrix}0\\1\end{bmatrix}\xi^4\right],\\
&= det\begin{bmatrix}-\xi^2 + \xi^3\\ -\xi^2 + \xi^4\end{bmatrix} = 0,
\end{aligned}$$

where the values of ξ_j are less than or equal to one,

$$\xi^2(\xi - 1) = 0,$$
$$\xi = 0,0,1,$$

$$\xi^2(\xi^2 - 1) = 0,$$
$$\xi = 0,0,1,1.$$

Referring to Theorem 5.4 by Ackleh (2009),

Theorem 5.4 *The LMM is considered convergent if it satisfies both the conditions of consistency and zero-stability.*

The proposed method is converged since both consistency and zero-stability requirements are met.

6. Stability of Method

This section examines the stability properties in solving NVDIDE. The analysis was first introduced by Bellen *et al.* (1988), and the test equation for the linear first-order NVDIDE with constant delay is given by,

$$y'(x) = \xi y(x - \tau) + \nu \int_0^{x-\tau} y(u)du + \eta y'(x - \tau). \tag{15}$$

The test equation is applied to build stability regions by applying the corrector formulae of the proposed method. The stability for predictor formulae is not a concern in this research since the computed numerical results obtained are dependent on the plausible results produced by the corrector formula with the provided time step. The underlying idea for tracing the region of constrained stability in the $H1 - H2$ plane is to replace t in the stability polynomial with the values $1, -1$ and $e^{i\theta}$ for $0 \le \theta \le 2\pi$. This replacement will result in a complex equation when $t = e^{i\theta}$ is replaced. The real and imaginary sections will be solved concurrently. The stability areas of the developed method can then be displayed from the points that satisfy $|t| \le 1$. The $H1 - H2$ plane refers to a two-dimensional parameter space used to visualize and analyze the stability properties of numerical methods, especially for linear systems. These parameters are often associated with the step size and complex eigenvalues of the analyzed method. Assume for the purpose of convenience that $x - \tau = mh(m \in I)$ and $y(x - \tau) = Y_{N-m}$, Rihan *et al.* (2009). The corrector formula for 1OBM5 is rearranged below,

$$\sum_{j=0}^{2} A_j Y_{N+j} = h \sum_{j=0}^{2} B_j F_{N+j}, \tag{16}$$

where,

$$A_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, A_1 = \begin{bmatrix} -1 & 0 \\ -1 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$B_0 = \begin{bmatrix} -\dfrac{1}{1800} & \dfrac{5}{1800} \\ 0 & -\dfrac{4}{135} \end{bmatrix}, B_1 = \begin{bmatrix} \dfrac{285}{1800} & \dfrac{1216}{1800} \\ \dfrac{81}{135} & -\dfrac{64}{135} \end{bmatrix}, B_2 = \begin{bmatrix} \dfrac{295}{1800} & 0 \\ \dfrac{216}{135} & \dfrac{41}{135} \end{bmatrix}.$$

After applying the abovementioned conditions and altering $H_1 = \eta h$ and $H_2 = \nu h^2$ (Baharum *et al.* 2022), the stability polynomial is obtained using Maple software after evaluating the determinant below,

$$\begin{aligned} \pi(H_1, H_2; t) = det \Big[& \left(A_2 - \frac{1}{3}H_2B_2\right)t^{m+2} + \left(A_1 - \frac{1}{3}H_2B_1 - \frac{4}{3}H_2B_2\right)t^{m+1} \\ & + \left(A_0 - \frac{1}{3}H_2B_0 - \frac{4}{3}H_2B_1 - \frac{1}{3}H_2B_2\right)t^{m} \\ & + \left(-\frac{4}{3}H_2B_0 - \frac{1}{3}H_2B_1\right)t^{m-1} + \left(-\frac{1}{3}H_2B_0\right)t^{m-2} \\ & + (-H_1B_2 - \eta B_2)t^2 + (-H_1B_1 - \eta B_1)t^1 + (-H_1B_0 - \eta B_0)t^0 \Big] \\ = \ & 0. \end{aligned} \tag{17}$$

The area of stability for the 1OBM5 is represented in Figure 2 below by setting $\eta = 1$,

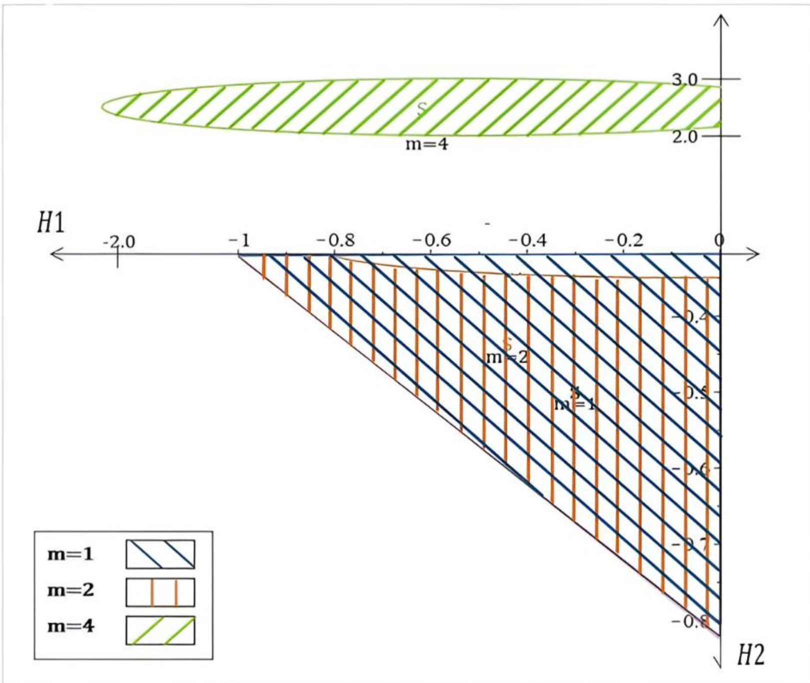

Figure 2: The regions of numerical stability for 1OBM5 with different values of $m = 1; 2; 4$.

According to Lambert (1973),

Definition 5 (Lambert 1973) *The hybrid multistep method is considered absolutely stable if all roots of the stability polynomial satisfy* $|t_s| < 1, s = 1,2, \dots, k$ *and is deemed absolutely unstable if this condition is not met.*

Since all the roots in the stability polynomial satisfy $|t_s| < 1$, for $s = 1,2,\ldots,k$ (as obtained from Maple software) the resulting stability regions are absolutely stable, meaning the numerical solution tends to zero. According to Torelli (1991),

Definition 6 (Torelli 1991**)** *Consider a stable method in relation to the test equation, where the interpolant y has a constant $m = 1$. In this case, the resulting VDIDE method is GPN-stable.*

Based on the above definitions, any stability test that focuses on the value of m is referred to as GPN-stable, while the region where the roots of the stability polynomial satisfies $|t_s| < 1, s = 1,2,\ldots,k$, is called the absolute stability region. The need for GPN-stability arises from the fact that many delay differential equations exhibit behaviours that are not adequately captured by traditional stability definitions (A-stability, L-stability). In particular, the interplay between nonlinear terms and delays can lead to phenomena such as oscillations or other complex behaviours that are not well understood or captured by basic stability concepts like A-stability or L-stability. GPN-stability provides a more comprehensive understanding of how the method behaves under these conditions, ensuring that the numerical solution remains accurate and stable even for more challenging nonlinear delay problems.

The test equation for the linear first-order NVDIDE with proportional delay is expressed as,

$$y'(x) = \xi y(qx) + \nu \int_0^{qx} y(u)du + \eta y'(qx). \tag{18}$$

For simplicity, assume that $qx = mh(m \in I)$ where $0 < q < 1$ and $y(qx) = Y_{Nm}$. The assumption of delay m has been applied by many authors previously (Rihan *et al.* (2009), Ishak *et al.* (2013), Aziz (2015) and Baharum *et al.* (2022)). The corrector formulae are applied to construct the stability regions. In constructing the region of limited stability in the $H1 - H2$ plane, t is substituted with the values $1, -1$ and $e^{i\theta}$ for $0 \leq \theta \leq 2\pi$ in the stability polynomial. The stability areas are then represented by using the locations that satisfy $|t| \leq 1$. The corrector multistep formula for 1OBM5 is rearranged using the same processes previously, by replacing $H_1 = \eta h$ and $H_2 = \eta h^2$,

$$\begin{aligned}\pi(H_1, H_2; t) = det\Big[&\left(A_2 - \frac{1}{3}H_2B_2\right)t^{m+2} + \left(A_1 - \frac{1}{3}H_2B_1 - \frac{4}{3}H_2B_2\right)t^{m+1} \\ &+ \left(A_0 - \frac{1}{3}H_2B_0 - \frac{4}{3}H_2B_1 - \frac{1}{3}H_2B_2\right)t^{m} \\ &+ \left(-\frac{4}{3}H_2B_0 - \frac{1}{3}H_2B_1\right)t^{m-1} + \left(-\frac{1}{3}H_2B_0\right)t^{m-2} \\ &+(-H_1B_2 - \eta B_2)t^{3m} + (-H_1B_1 - \eta B_1)t^{2m} + (-H_1B_0 - \eta B_0)t^{0}\Big] \\ =\ & 0.\end{aligned} \tag{19}$$

The stability polynomial is obtained using the Maple software after evaluating the determinant above. The stability region for 1OBM5 is then obtained and displayed in Figure 3 below,

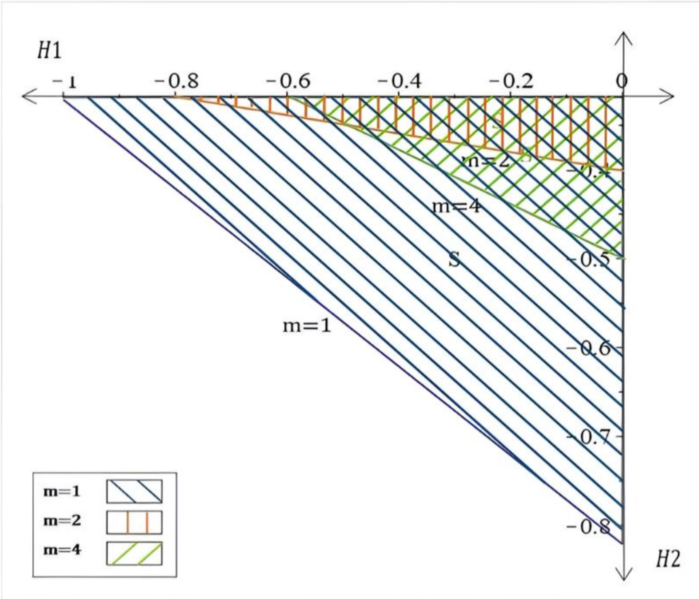

Figure 3: The regions of numerical stability for 1OBM5 with different values of $m = 1; 2; 4$

From definitions stated by Lambert (1973) and Torelli (1991), the stability region is named the GPN-stable absolute stability region. The stability of the numerical method depends on how these points are chosen, and how the method handles errors when advancing from one step to the next. The delay term, m, is chosen wisely to produce bigger stability region. Based on the figures above, a smaller value of the delay parameter m leads to a larger range of step sizes h that can be applied. As the range of h increases, the proposed method becomes suitable for solving a wider range of differential equations, including stiff problems.

7. Implementation of Method

The first order NVDIDE of constant type,

$$y'(x) = f\big(x, y(x)\big) + \int_{x-\tau}^{x} K\big(t, y(t), y(x-\tau), y'(x-\tau)\big)dt\,, \quad x \in [a, b], \qquad (20)$$
$$y(x) = \phi(x), \quad x \leq a,$$

are solved in this section by considering the developed method where two approximations are computed within a single block using a constant step size, h. To solve NVDIDE, both integral and differential parts need to be considered. The integral part is solved by applying a suitable quadrature rule whereas the differential part involved two delay functions, one that involved the delay derivative of an unknown function, $y'(x-\tau)$, and another that does not entail the consideration of its derivative, $y(x-\tau)$. However, before calculating the proposed approach, the position of delays must be established beforehand. Other authors have solved the delay derivative in various ways, such as using the first derivative technique. In this study, $y'(x-\tau)$, will be evaluated using the backward and forward divided difference formulae. The backward divided difference formula (BDF) and forward divided difference formula (FDF) of order five are denoted as follows,

$$y_n' = \frac{3y_n - 16y_{n-1} + 36y_{n-2} - 48y_{n-3} + 25y_{n-4}}{12h} (BDF5), \tag{21}$$
$$y_n' = \frac{-25y_n + 48y_{n+1} - 36y_{n+2} + 16y_{n+3} - 3y_{n+4}}{12h} (FDF5).$$

The value of delay, τ, is estimated by applying the initial function specified in the Eq. (20), if $x - \tau_i \leq a$. Otherwise, if $x - \tau_i > a$, a Lagrange interpolating polynomial is used. The derivatives of the delay solutions, $y'(x-\tau_i)$ and $y'(qx)$, are approximated using backward and forward divided difference formulas, based on the availability of calculated points. If the previous points are insufficient to solve the current iteration, then FDF5 is applied instead of BDF5. By using both methods, the algorithm becomes more flexible and can be applied to a broader range of delay differential equations. This approach increases the algorithm's applicability to various types of DDEs that may involve mixed types of delays, as is common in many real-world problems, such as biological models, population dynamics, and control systems. Runge-Kutta 5th order (RK5) is employed to estimate the initial values of constant delay while the implicit Euler method is employed for solving NVDIDEs with pantograph delay. Some constraints necessitate the use of a lower-order method for determining the initial values for the pantograph delay. The first constraint arises from the application of the polynomial, as the Lagrange interpolation order is low at the start of the iteration, and there are insufficient points to calculate higher-order Lagrange polynomials. Therefore, using higher-order methods is limited because the Lagrange interpolating polynomial requires many points to yield accurate results. The second constraint is that the Lagrange polynomial is evaluated repeatedly within a single iteration, which reduces the accuracy of the results. A constant delay situation, on the other hand, normally does not require the evaluation of Lagrange since the delay terms will fall precisely on the prior approximate values. The integration problem is handled using the Composite Simpson quadrature rule. Simpson's approach is employed for each subinterval, and the results are summed to produce an approximation for the integral across the full interval. This method is known as the composite Simpson's rule. Assume the interval $[a, b]$ is partitioned into n subintervals, the composite Simpson's rule is thus represented as shown below,

$$\begin{aligned}\int_a^b f(x)\, dx &\approx \frac{h}{3}\sum_{j=1}^{n/2}\left[f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j})\right] \\ &= \frac{h}{3}\left[f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\right],\end{aligned} \tag{22}$$

where $x_j = a + jh$ for $j = 0,1, \dots, n-1, n$, with $h = \frac{b-a}{n}$ in particular, $x_0 = a$ and $x_n = b$. Since the developed method is a hybrid multistep block approach that does not stand alone, the starting solutions must be addressed before adopting the proposed method. As the predictor formulae are of order five, hence five initial solutions are required. Before applying the main approach, a Runge-Kutta of order five (RK5) has been chosen to evaluate the first solutions for the suggested methods. The formulae for RK5 are shown below,

Runge-Kutta of order 5 (RK5), taken from Butcher (2008)

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{h}{90}(7k_1 + 32k_2 + 12k_4 + 32k_5 + 7k_6) \\
k_1 &= f\left(x_n, y_n + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right) \\
k_2 &= f\left(x_n + \frac{h}{4}, \left(y_n + \frac{h}{4}k_1\right) + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right) \\
k_3 &= f\left(x_n + \frac{h}{4}, \left(y_n + \frac{h}{8}k_1 + \frac{h}{8}k_2\right)\right. \\
&\qquad \left. + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right) \\
k_4 &= f\left(x_n + \frac{h}{2}, \left(y_n - \frac{h}{2}k_2 + hk_3\right) + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right) \\
k_5 &= f\left(x_n + \frac{3h}{4}, \left(y_n + \frac{3h}{16}k_1 + \frac{9h}{16}k_4\right) + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right) \\
k_6 &= f\left(x_n + h, \left(y_n - \frac{3h}{7}k_1 + \frac{2h}{7}k_2 + \frac{12h}{7}k_3 - \frac{12h}{7}k_4 + \frac{8h}{7}k_5\right)\right. \\
&\qquad \left. + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right).
\end{aligned}
\tag{23}
$$

The proposed method, using a predictor-corrector approach, will generate approximations concurrently. First, the off point will be applied, after which the predictor-corrector mode will be approximated. Once the half point is calculated, the predictor will estimate the solution for each iteration, and the result will be inserted into the corrector. The corrected values will be iterated until convergence is achieved. The numerical results are computed using a C program with a fixed step size.

The detailed approaches to the developed methods for solving NVDIDE with constant delay via constant step length are displayed in this section. The algorithms are presented in detail on how to handle an integral part, the delay term, and its derivative. The following notations are used in the algorithms,

a	Initial value
b	End value
h	Step size
N	Number of iterations
y_0	Initial solution
$y(x-\tau)$	Delay term
$y'(x-\tau)$	Delay derivative

Algorithms of 1OBM5 For Solving First Order NVDIDE

Step 1: All values given in equation, $x_0 = a, x_n = b, h, N, y_0, y'(x-\tau) \leq a$ are set

Step 2: The Volterra integro-differential equation of neutral delay is defined:

$$y'(x) = f\left(x_n, y_n + \int_{n-m}^{n} K(x_n, y_{n-m}, y'_{n-m})\right).$$

Step 3: The original function given in (20) is used if $x - \tau \leq a$.

Step 4: The Lagrange interpolating polynomial is used if $x - \tau > a$.

Step 5: BDF5 and FDF5 in (21) are tested to find $y'(x-\tau)$.
Step 6: Composite Simpson in (22) is applied to approximate the integral part.
Step 7: For n=0,1,2,3,4,
The initial solution is computed by applying the RK5 (23)
Step 8: For n=5,7,9…
Approximate NVDIDE by using the proposed method, 1OBM5 denoted in (10)
Step 9: Average and maximum error, total steps, and function calls are calculated computationally.
Step 10: Stop

8. Numerical Results

h	Step size
MTD	Method
FCN	Total function calls
TS	Total Step
AVERE	Average Error
MAXE	Maximum Error
1OBM5	Two-point one off-step point hybrid multistep block method (Order 5)
2PBM5,Aziz (2015)	Two-point Multistep Block Method (Order 5)
ABM5	Adam-Bashforth-Moulton Method (Order 5)
RK5	Runge Kutta Method (Order 5)
5e-10	5×10^{-10}

Example 1: Zaidan (2012) Case A

$$y'(x-1) = 1 - \frac{x^3}{6} + \int_0^x (x-t)y(t)dt\,,$$
$$y(0) = 0,$$

Exact solution:

$$y(x) = x, \qquad x \in [0,1]$$

h	MTD	FCN	TS	MAXE	AVERE
0.01	1OBM5	72	52	7.1444e-06	2.3640e-06
	2PBM5	115	52	8.0150e-06	2.6899e-06
	ABM5	115	100	5.4562e-06	1.8125e-06
	RK5	600	100	1.3519e-04	3.4709e-05
0.001	1OBM5	522	502	7.1533e-08	2.3648e-08
	2PBM5	1015	502	8.0246e-08	2.6569e-08
	ABM5	1015	1000	5.4806e-08	1.8125e-08
	RK5	6000	1000	1.3116e-05	3.2777e-06
0.0001	1OBM5	5022	5002	7.1545e-10	2.3652e-10
	2PBM5	10015	5002	8.0254e-10	2.6535e-10
	ABM5	10015	10000	5.4834e-10	1.8128e-10
	RK5	60000	10000	1.3076e-06	3.2583e-07

Example 2: Horvat (1999) Case A

$$y'(x) = 2e^{1-x} - 3y(x) - 3\int_{x-1}^{x} y(t)dt - \int_{x-1}^{x} y'^{(t)}dt,$$
$$\mathrm{y}(0) = 1,$$

Exact solution:

$$y(x) = e^{-x}, \qquad x \in [0,1]$$

h	MTD	FCN	TS	MAXE	AVERE
0.01	1OBM5	72	52	1.6334e-04	5.9313e-05
	2PBM5	115	52	6.1889e-04	2.1668e-05
	ABM5	115	100	1.1636e-03	8.3482e-04
	RK5	600	100	1.6334e-04	1.0377e-04
0.001	1OBM5	522	502	1.1939e-06	6.1937e-07
	2PBM5	1015	502	7.9898e-05	2.4278e-07
	ABM5	1015	1000	1.2266e-04	8.9458e-05
	RK5	6000	1000	1.1939e-06	7.5507e-07
0.0001	1OBM5	5022	5002	2.9877e-07	6.3531e-09
	2PBM5	10015	5002	8.1740e-06	1.0023e-08
	ABM5	10015	10000	1.2334e-05	9.0140e-06
	RK5	60000	10000	2.9877e-07	1.8887e-07

Example 3: Case B

$$y'(x) = cos\left(\frac{x}{2}\right)y\left(\frac{x}{2}\right) + \int_{0}^{\frac{x}{2}}[sin(t)y(t) - cos(t)y'(t)]dt + cos(x),$$
$$y(0) = 0,$$

Exact solution:

$$y(x) = sin(x), \qquad x \in [0,1]$$

h	MTD	FCN	TS	MAXE	AVERE
0.01	1OBM5	52	52	8.4891e-04	3.5504e-04
	2PBM5	100	52	7.1393e-04	3.1132e-04
	ABM5	100	100	7.6228e-04	3.1865e-04
	RK5	600	100	1.8716e-03	8.8577e-04
0.001	1OBM5	502	502	1.0180e-04	4.8399e-05
	2PBM5	1000	502	7.9047e-05	3.9722e-05
	ABM5	1000	1000	8.3957e-05	4.0963e-05
	RK5	6000	1000	1.8745e-04	8.7411e-05
0.0001	1OBM5	5002	5002	1.0349e-05	4.9883e-06
	2PBM5	10000	5002	7.9819e-06	4.0696e-06
	ABM5	10000	10000	8.4732e-06	4.2012e-06
	RK5	60000	10000	1.8748e-05	8.7293e-06

1OBM5 is the method proposed in this manuscript. It is a fifth-order hybrid block method that combines the benefits of both multistep and off-step techniques. It uses two main mesh points and one additional off-step point to improve the accuracy of derivative and integral approximations. This off-step point allows the method to better capture the behavior of delay or memory terms in integro-differential equations. Introduced by Aziz (2015), the 2PBM5 method is a fifth-order block method that uses two main mesh points for computation, but does not employ off-step evaluations. It is efficient and well-suited for ordinary differential equations, but may lack the flexibility and precision needed when dealing with delay or integral components. It serves as a baseline comparison to highlight the added advantage of using an off-step point in hybrid methods like 1OBM5. The Adams-Bashforth-Moulton method is a predictor-corrector scheme that combines the explicit Adams-Bashforth method (predictor) with the implicit Adams-Moulton method (corrector). The Runge-Kutta fifth-order method is a classical single-step scheme that computes several intermediate stages to achieve high accuracy. While it is general-purpose and does not rely on prior steps (unlike multistep methods), its major drawback is the high number of function evaluations per time step, leading to increased computational cost.

The numerical performance of the methods was tested on three standard examples with known exact solutions. The results were measured in terms of maximum error (MAXE), average error (AVERE), number of function calls (FCN), and total steps (TS). Across all three examples, the proposed 1OBM5 method consistently produces the lowest or near-lowest maximum and average errors, particularly as the step size decreases. In Example 1, where the solution is linear ($y(x)$=x), all methods perform well. However, 1OBM5 maintains a slight edge in accuracy at every tested step size, especially with h=0.0001, achieving a maximum error of just 7.15×10^{-10}, which is marginally better than ABM5 and RK5. The ABM5 achieves slightly smaller errors across step sizes because the problem is essentially polynomial in form and contains no strong integro-differential effects. ABM5, being a classical predictor–corrector scheme, is well suited for such smooth problems. By contrast, 1OBM5 incorporates off-step approximations designed to treat integro-differential delays. In purely linear cases, this additional machinery does not provide extra benefit and may introduce a small additional truncation error. However, 1OBM5 was designed to efficiently approximate terms involving integral kernels and delayed derivatives, which are not directly handled by ABM5. While both methods are convergent of order five, 1OBM5 retains absolute stability and GPN-stability even in the presence of neutral and proportional delays, where ABM5's performance can deteriorate. In Example 2, with exact solution $y(x) = e^{-x}$, 1OBM5 clearly outperforms the other methods. RK5 matches the error of 1OBM5 at h=0.01 but requires significantly more function evaluations. ABM5 and 2PBM5 show notably higher errors, especially for larger step sizes, indicating reduced stability or inability to handle the combined delay-integro-differential nature effectively. In Example 3, a more complex nonlinear equation with proportional delay, 2PBM5 slightly outperforms 1OBM5 at the coarsest level (h=0.01), but 1OBM5 overtakes it as the step size decreases. RK5 continues to yield the highest errors, likely due to sensitivity in trigonometric and proportional delay. The ABM5 again shows smaller errors due to its well-known stability in the regimes. Our 1OBM5 method remains stable, but its off-step point interpolation introduces slight error amplification when the proportional delay component dominates. However, 1OBM5 achieves two approximations per block with fewer function evaluations, making it more efficient than ABM5 and other multistep methods when applied to complex NVDIDEs. In Test 2, where strong delay–integral is present, 1OBM5 clearly outperforms ABM5 across all step sizes, demonstrating its effectiveness in the target class of problems. It is important to highlight that Examples 1 and 2 used in this study are identical to those tested in Ismail *et al.* (2021), where the authors proposed a third-order Two-Point Explicit Block Method (2PEB3) for solving neutral and retarded Volterra delay integro-differential equations (NVDIDE and RVDIDE). In contrast, the current study introduces a fifth-order hybrid multistep block method (1OBM5), which incorporates an off-step point to enhance the approximation of derivative and integral components. The numerical results clearly indicate that 1OBM5 achieves significantly lower maximum and average errors than 2PEB3 across all tested step sizes, while maintaining fewer function calls than traditional methods such as RK5 and ABM5. As in comparison with 2PEB3, these results demonstrate that 1OBM5 not only improves upon the computational efficiency seen in 2PEB3, but also delivers superior accuracy due to its fifth-order structure and hybrid off-step design. As such, the comparative performance with 2PEB3 is recommend to be highlighted in the discussion to emphasize the advancement achieved in this research.

9. Conclusion

The proposed method has shown its efficiency in solving NVDIDE with case A and B. The hybrid multistep block method has produced less computational work since the calculation involves two approximations in a single step. The AVERE and MAXE produced are comparable to the existing methods and it should be compatible to solve NVDIDE with constant and proportional delays. Overall, NVDIDEs provide a powerful tool for modeling complex systems with memory and delayed feedback. In this capacity, they continue to be an active field of research and a valuable tool for understanding and predicting the behavior of real-world systems. The resulting hybrid method should be both precise and effective, with a high level of stability and low calculation cost. In addition, it is important to fully test the new method for a variety of problems to ensure its reliability and efficiency. Overall, the creation of a multi-step hybrid method requires a comprehensive understanding of numerical methods and a careful balance of their strengths and weaknesses.

Acknowlegments

The authors sincerely appreciate the financial support provided by the Fundamental Research Grant Scheme (project code: FRGS/1/2020/STG06/UPM/01/1) from the Ministry of Higher Education Malaysia. We also extend our gratitude to the referees for their valuable feedback and suggestions.

References

Ackleh A.S., Allen E.J., Kearfott R.B. & Seshaiyer P. 2009. *Classical and Modern Numerical Analysis: Theory, Methods and Practice.* Boca Raton, FL: CRC Press.

Aziz N.H.B.A. 2015. Block multistep methods for solving first order retarded and neutral delay differential equations. PhD Thesis. Universiti Putra Malaysia.

Baharum N.A., Majid Z.A., Senu N. & Rosali H. 2022. Numerical approach for delay Volterra integro-differential equation. *Sains Malaysiana* **51**(12): 4125-4144.

Bellen A., Jackiewicz Z. & Zennaro M. 1988. Stability analysis of one-step methods for neutral delay-differential equations. *Numerische Mathematik* **52**(6): 605-619.

Butcher J.C. 2008. *Numerical Methods for Ordinary Differential Equation.* New York, NY: John Wiley & Sons.

Fatunla S.O. 1995. A class of block methods for second order IVPs. *International Journal of Computer Mathematics* **55**(1-2): 119-133.

Horvat V. 1999. On polynomial spline collocation methods for neutral Volterra integro-differential equations with delay arguments. *Proceedings of the 1st Conference on Applied Mathematics and Computation,* pp. 113–128.

Ishak F., Suleiman M.B. & Majid Z.A. 2013. Block method for solving pantograph type functional differential equations. *Proceedings of the World Congress on Engineering*, pp. 948–952.

Ismail N.I. N. & Majid Z. A. 2024. Numerical solution on neutral delay Volterra integro-differential equation. *Bulletin of the Malaysian Mathematical Sciences Society* 47(3): 85.

Ismail N.I.N., Majid Z.A., Senu N. & Wahi N. 2021. Numerical method in solving neutral and retarded Volterra delay integro-differential equation. *Journal of Physics: Conference Series* **1988**(1): 012033.
Ismail N.I.N., Majid Z.A. & Senu N. 2020. Hybrid multistep block method for solvingneutral delay differential equations. *Sains Malaysiana* **49**(4): 929-940.
Jator S.N. 2010. On the hybrid method with three off-step points for initial value problems. *International Journal of Mathematical Education in Science and Technology* **41**(1): 110-118.
Jator S., Okunlola T., Biala T. & Adeniyi R. 2018. Direct integrators for the general third-order ordinary differential equations with an application to the Korteweg-de Vries equation. *International Journal of Applied and Computational Mathematics* **4**(5): 110.
Kuang Y. 1993. *Delay Differential Equations: With Applications in Population Dynamics*. Volume 191. Boston, MA: Academic Press.
Lambert J.D. 1973. *Computational Methods in Ordinary Differential Equations.* New York, NY: John Wiley & Sons.
Mirzaee F., Bimesl S. & Tohidi E. 2016. A numerical framework for solving high-order pantograph-delay Volterra integro-differential equations. *Kuwait Journal of Science* **43**(1): 69–83.
Reutskiy S.Y. 2016. The backward substitution method for multipoint problems with linear Volterra–Fredholm integro-differential equations of the neutral type. *Journal of Computational and Applied Mathematics*, **296**: 724–738.
Rihan F.A., Doha E.H., Hassan M.I. & Kamel N. 2009. Numerical treatments for Volterra delay integro-differential equations. *Computational Methods in Applied Mathematics* **9**(3): 292-318.
Torelli L. 1991. A suffcient condition for GPN-stability for delay differential equations. *Numerische Mathematik* **59**(1): 311-320.
Yu Y. 2025. Solving nonlinear neutral delay integro-differential equations via general linear methods. *Journal of Computational and Applied Mathematics* **458**: 116342.
Yüzbaşı Ş. 2014. Laguerre approach for solving pantograph-type Volterra integro-differential equations. *Applied Mathematics and Computation* **232**: 1183-1199.
Zaidan L.I. 2012. Solving linear delay Volterra integro-differential equations by using Galerkin's method with Bernstien polynomial. *Journal of Babylon University/Pure and Applied Sciences* **20**(5): 1305-1313.

School of Computing and Data Sciences
Faculty of Robotics and Data Sciences
Nilai University
71800 Nilai,
Negeri Sembilan, MALAYSIA
*E-mail: inshirahismail8996@gmail.com**

Department of Mathematics and Statistics
Faculty of Science
Universiti Putra Malaysia
43400 UPM Serdang
Selangor, MALAYSIA
E-mail: am_zana@upm.edu.my

Received: 15 August 2024
Accepted: 3 October 2025

*Corresponding author