Sains Malaysiana 51(12)(2022): 4125-4144
http://doi.org/10.17576/jsm-2022-5112-20

Numerical Approach for Delay Volterra Integro-Differential Equation

(Pendekatan Berangka Bagi Penyelesaian Persamaan Pembezaan Lengah-Kamilan Volterra)

NUR AUNI BAHARUM[1], ZANARIAH ABDUL MAJID[1,2,*], NORAZAK SENU[1,2] & HALIZA ROSALI[1,2]

[1]*Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia*
[2]*Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia*

Received: 12 May 2022/Accepted: 29 August 2022

ABSTRACT

The delay integro-differential equation for the Volterra type has been solved by using the two-point multistep block (2PBM) method with constant step-size. The proposed block method of order three is formulated using Taylor expansion and will simultaneously approximate the numerical solution at two points. The 2PBM method is developed by combining the predictor and corrector formulae in the PECE mode. The predictor formulae are explicit, while the corrector formulae are implicit. The algorithm for the approximate solutions were constructed and analyzed using the 2PBM method with Newton-Cotes quadrature rules. This paper focused on constant and pantograph delay types, and the previous values are used to interpolate the delay solutions. Moreover, the studies also carried out on the stability analysis of the proposed method. Some numerical results are tested to validate the competency of the multistep block method with quadrature rule approach.

Keywords: Multistep block; Newton-Cotes rule; Volterra delay integro-differential equation

ABSTRAK

Persamaan pembezaan lengah kamilan bagi jenis Volterra telah diselesaikan menggunakan kaedah blok berbilang langkah dua titik (2PBM) untuk langkah yang malar. Kaedah blok peringkat tiga yang dicadangkan telah dirumus menggunakan pengembangan Taylor dan akan menganggar penyelesaian berangka secara serentak pada dua titik. Kaedah 2PBM dibangunkan dengan menggabungkan formula peramal dan pembetul dalam mod PECE. Kaedah peramal adalah tak tersirat manakala kaedah pembetul adalah tersirat. Algoritma penyelesaian anggaran dibina dan dianalisis menggunakan kaedah 2PBM dengan peraturan kuadratur Newton-Cotes. Kertas ini memberi tumpuan kepada jenis kelengahan malar dan pantograf serta nilai sebelumnya digunakan untuk menginterpolasi penyelesaian kelengahan. Selain itu, kajian juga dijalankan ke atas analisis kestabilan bagi kaedah yang dicadangkan. Beberapa keputusan berangka diuji untuk mengesahkan kecekapan kaedah blok berbilang langkah dengan pendekatan peraturan kuadratur.

Kata kunci: Blok berbilang langkah; peraturan Newton-Cotes; persamaan pembezaaan lengah-kamilan Volterra

INTRODUCTION

The model of the delay integro-differential equation is considered as follows in this study,

$$y'(x) = f\big(x, y(x), y(x-\tau)\big) + \int_0^x K\big(x, u, y(u), y(u-\tau)\big)\, du \ , x \in [a, b] \tag{1}$$

with subject to the arbitrary initial function, $y(x) = \phi(x)$, for $x \in [a\text{-}\tau, a]$.

The $\tau = \tau(x)$ constant is a delay term, whereas the delay argument is (x-τ). The $y(x\text{-}\tau)$ expression relates to the delay solution. The approximate numerical solution can be obtained when equation (1) reduces to a standard initial value problem (IVP).

$$y'(x) = F\big(x, y(x), y(x-\tau), z(x)\big) \tag{2}$$

where

$$z(x) = \int_0^x K\big(x, u, y(u), y(u-\tau)\big)\, du. \tag{3}$$

Block methods have been developed which provide a polynomial approximation to the solution of the standard initial value problem on a mesh point with $x_0 = a$, $x_k = X$, $x_0 < x_1 < \cdots < x_k$. These methods are often analyzed with the hypothesis that the step size, h is constant.

Delay elements in the integro-differential equation are often encountered in real-life phenomena throughout scientific and engineering problems. Material science (Baker 2000) and control problems (Kolmanovskii & Myshkis 2012) range from the area covered by delayed integro-differential equations. The delay integro-differential equation is too complicated for analytical solutions to be solved. In order to obtain the solution for delay integro-differential equation, reliable numerical schemes are necessary.

Several methods devised to treat the equation (1) numerically have been observed in the last few decades. Yüzbaşı and Karaçayır (2018) used the Galerkin-like method of Taylor polynomial to achieve an approximate delay integro-differential solution. Ali (2009) used the expansion method to estimate the numerical solution (1) using B-spline polynomials. Moreover, Salih et al. (2010) implemented a B-spline function with the Galerkin method to test the delay integro-differential equation's convolution type. Zaidan (2012) adapted the Bernstein polynomial as a basic function of the Galerkin method to approximate solution delay integro-differential equation which contains three kinds of equation (retarded, neutral and mixed). Mustafa and Mohammed (2018) introduced Galerkin to solve the linear delay integro-differential equation using the polynomial of Chebyshev.

Ayad (2001) obtained the convergence of the spline method for nonlinear delay integro-differential equation and solved the equation numerically. Qin et al. (2018) studied the stability of additive Runge-Kutta methods for delay integro-differential equation.

The multistep block method has been introduced to solve the Volterra integro-differential equation without delay and delay differential equation. Majid and Mohamed (2019) developed fifth order fully implicit multistep block method for evaluating the numerical results of Volterra integro-differential equation without delay. The hybrid block method with aid of quadrature rule has been proposed by Janodi et al. (2020). Baharum, Majid and Senu (2022) applied the diagonally implicit multistep block method of order five to approximate the solution of integro-differential equation of Volterra type without delay terms. Meanwhile, Ismail, Majid and Senu (2020) proposed hybrid multistep block method to solve the delay differential equation.

This paper aims to investigate the delay integro-differential equation and obtains some new numerical results based on the two points multistep block method with the quadrature rule. A few examples demonstrate that the numerical results of the proposed method are efficient.

FORMULATION OF THE METHOD

The two approximate solutions will be represented by the first point, y_{n+1} at x_{n+1} and the second point, y_{n+2} at x_{n+2}. These two solutions will be computed simultaneously using the 2PBM method. Furthermore, the 2PBM method is based on the predictor-corrector method and applied to find the solution for the standard initial value problem of the delay integro-differential equation. The corrector formulae of 2PBM are known as an implicit method.

To begin the derivation of 2PBM method, the linear difference operator, L associated with

$$L[y(x); h] = \sum_{i=0}^{k} \alpha_i\, y(x+ih) - h \sum_{i=0}^{k} \beta_i\, y'(x+ih), \tag{4}$$

where $y(x)$ is an arbitrary function and continuously differentiable on $[a,b]$. The Taylor expansion substitutes the dependent variable, $y(x)$ and its derivatives, $y'(x)$ and constitutes the linear difference operator,

$$L[y(x); h] = \sum_{i=0}^{k} \alpha_i \left(y(x) + ihy'(x) + \frac{i^2}{2!} h^2 y''(x) + \cdots \right) - h \sum_{i=0}^{k} \beta_i \left(y'(x) + ihy''(x) + \frac{i^2}{2!} h^2 y'''(x) + \cdots \right), \tag{5}$$

The development of 2PBM method is to evaluate $y(x_{n+1})$ and $y(x_{n+2})$, respectively, with their delay argument and delay solution. As shown below, a linear multistep method (LMM) (4) expanded to develop the first-point corrector formula.

$$\sum_{i=2}^{k} \alpha_i\, y(x+ih) = h \sum_{i=0}^{k-3} \beta_i\, y'(x+ih) + h \sum_{i=2}^{k} \beta_i\, y'(x+ih).$$

The first point corrector formula of 2PBM generated for the step number, k=3:

$$\sum_{i=2}^{3} \alpha_i\, y(x+ih) = h\sum_{i=0}^{0} \beta_i\, y'(x+ih) + h\sum_{i=2}^{3} \beta_i\, y'(x+ih), \quad (6)$$

$$\alpha_2 y(x+2h) + \alpha_3 y(x+3h) = h\,\beta_0 y'(x) + h\beta_2 y'(x+2h) + h\beta_3 y'(x+3h). \quad (7)$$

By letting $\alpha_3 = 1$, $\alpha_2 = -1$, yields,

$$y(x+3h) - y(x+2h) = h\beta_0 y'(x) + h\beta_2 y'(x+2h) + h\beta_3 y'(x+3h), \quad (8)$$

$$y(x+3h) = y(x+2h) + h\beta_0 y'(x) + h\beta_2 y'(x+2h) + h\beta_3 y'(x+3h). \quad (9)$$

The expression of $y(x+3h)$, $y(x+2h)$, $y'(x)$, $y'(x+2h)$, $y'(x+3h)$ will be generated by Taylor expansion,

$$y(x+3h) = y(x) + 3hy'(x) + \frac{9}{2}h^2y''(x) + \frac{27}{6}h^3y'''(x),$$

$$y(x+2h) = y(x) + 2hy'(x) + \frac{4}{2}h^2y''(x) + \frac{8}{6}h^3y'''(x),$$

$$y'(x+2h) = y'(x) + 2h\,y''(x) + \frac{4}{2}h^2y'''(x),$$

$$y'(x+3h) = y'(x) + 3h\,y''(x) + \frac{9}{2}h^2y'''(x).$$

The Taylor series expansion terms are truncated at the third derivative as the method order three. Further, substituting the Taylor expansion into the equation (9) and yields,

$$\begin{aligned} &y(x) + 3h\,y'(x) + \frac{9}{2}h^2y''(x) + \frac{27}{6}h^3y'''(x) \\ &\cong y(x) + 2hy'(x) + \frac{4}{2}h^2y''(x) + \frac{8}{6}h^3y'''(x) + h\beta_0\big(y'(x)\big) \\ &+ h\beta_2\left(y'(x) + 2hy''(x) + \frac{4}{2}h^2y'''(x)\right) \\ &+ h\beta_3\left(y'(x) + 3h\,y''(x) + \frac{9}{2}h^2y'''(x)\right), \end{aligned} \quad (10)$$

$$\begin{aligned} &y(x) + 3h\,y'(x) + \frac{9}{2}h^2y''(x) + \frac{27}{6}h^3y'''(x) \\ &\cong y(x) + h\,y'(x)(2 + \beta_0 + \beta_2 + \beta_3) + h^2y''(x)\left(\frac{}{2} + 2\beta_2 + 3\beta_3\right) \\ &+ h^3y'''(x)\left(\frac{8}{6} + \frac{4}{2}\beta_2 + \frac{9}{2}\beta_3\right). \end{aligned} \quad (11)$$

Associating the left-hand side of equation (9) and the right-hand side of the equation (11) would give

$$2 + \beta_0 + \beta_2 + \beta_3 = 3,$$

$$\frac{4}{2} + 2\beta_2 + 3\beta_3 = \frac{9}{2},$$

$$\frac{8}{6} + \frac{4}{2}\beta_2 + \frac{9}{2}\beta_3 = \frac{27}{6}.$$

Therefore, the coefficient of β_i obtained as follows,

$$\beta_0 = -\frac{1}{36}, \qquad \beta_2 = \frac{7}{12}, \qquad \beta_3 = \frac{4}{9}.$$

As the result, the formula could be written as

$$y_{n+3} = y_{n+2} + h\left(-\frac{1}{36}F_n + \frac{7}{12}F_{n+2} + \frac{4}{9}F_{n+3}\right).$$

By letting $n \rightarrow n$-2, the derivation developed the first point corrector formula of 2PBM;

$$y^c_{n+1} = y_n + h\left(-\frac{1}{36}F_{n-2} + \frac{7}{12}F_n + \frac{4}{9}F_{n+1}\right). \quad (12)$$

The second point of the corrector method is derived as follows on the basis of LMM,

$$\sum_{i=2}^{k-2} \alpha_i\, y(x+ih) + \sum_{i=k}^{k} \alpha_i\, y(x+ih) = h\sum_{i=0}^{k-4} \beta_i\, y'(x+ih) + h\sum_{i=3}^{k} \beta_i\, y'(x+ih).$$

Taking the step number, k = 4, will be;

$$\sum_{i=2}^{2} \alpha_i\, y(x+ih) + \sum_{i=4}^{4} \alpha_i\, y(x+ih) = h\sum_{i=0}^{0} \beta_i\, y'(x+ih) + h\sum_{i=3}^{4} \beta_i\, y'(x+ih), \quad (13)$$

$$\alpha_2 y(x+2h) + \alpha_4 y(x+4h) = h\,\beta_0 y'(x) + h\beta_3 y'(x+3h) + h\beta_4 y'(x+4h). \quad (14)$$

Hence, letting $\alpha_4 = 1$, $\alpha_2 = -1$,

$$y(x+4h) - y(x+2h) = h\beta_0 y'(x) + h\beta_3 y'(x+3h) + h\beta_4 y'(x+4h), \quad (15)$$

$$y(x+4h) = y(x+2h) + h\beta_0 y'(x) + h\beta_3 y'(x+3h) + h\beta_4 y'(x+4h). \quad (16)$$

Taylor expansion is chosen to be expanding the terms of $y(x)$ and $y'(x)$, and the terms are truncated at third derivatives since the proposed method of 2PBM will be derived as the third-order method.

$$y(x+4h) = y(x) + 4hy'(x) + \frac{16}{2}h^2y''(x) + \frac{64}{6}h^3y'''(x),$$

$$y(x+2h) = y(x) + 2hy'(x) + \frac{4}{2}h^2y''(x) + \frac{8}{6}h^3y'''(x),$$

$$y'(x+3h) = y'(x) + 3h\,y''(x) + \frac{9}{2}h^2y'''(x),$$

$$y'(x+4h) = y'(x) + 4h\,y''(x) + \frac{16}{2}h^2y'''(x).$$

Substituting the Taylor expansion toward equation (16) and yields,

$$\begin{aligned} &y(x) + 4hy'(x) + \frac{16}{2}h^2y''(x) + \frac{64}{6}h^3y'''(x) \\ &\cong y(x) + 2hy'(x) + \frac{4}{2}h^2y''(x) + \frac{8}{6}h^3y'''(x) + h\beta_0y'(x) \\ &+ h\beta_3\left(y'(x) + 3h\,y''(x) + \frac{9}{2}h^2y'''(x)\right) \\ &+ h\beta_4\left(y'(x) + 4h\,y''(x) + \frac{16}{2}h^2y'''(x)\right), \end{aligned}$$

$$\begin{aligned} &y(x) + 4hy'(x) + \frac{16}{2}h^2y''(x) + \frac{64}{6}h^3y'''(x) \\ &\cong y(x) + hy'(x)(2 + \beta_0 + \beta_3 + \beta_4) + \\ &h^2y''(x)\left(\frac{4}{2} + 3\beta_3 + 4\beta_4\right) + h^3y'''(x)\left(\frac{8}{6} + \frac{9}{2}\beta_3 + \frac{16}{2}\beta_4\right). \end{aligned} \quad (17)$$

Equating equation (16) with both the left and the right side of equation (17) would give

$$2 + \beta_0 + \beta_3 + \beta_4 = 4,$$

$$\frac{4}{2} + 3\beta_3 + 4\beta_4 = \frac{16}{2},$$

$$\frac{8}{6} + \frac{9}{2}\beta_3 + \frac{16}{2}\beta_4 = \frac{64}{6},$$

thus,

$$\beta_0 = \frac{1}{18}, \qquad \beta_3 = \frac{16}{9}, \qquad \beta_4 = \frac{1}{6}.$$

The formula can be written

$$y_{n+4} = y_{n+2} + h\left(\frac{1}{18}F_n + \frac{16}{9}F_{n+3} + \frac{1}{6}F_{n+4}\right)$$

The second point corrector formula can be established after letting n→n-2,

$$y^c_{n+2} = y_n + h\left(\frac{1}{18}F_{n-2} + \frac{16}{9}F_{n+1} + \frac{1}{6}F_{n+2}\right). \quad (18)$$

Therefore, the first point and second point predictor formulae of 2PBM will be derived using similar procedure as the corrector formulae. However, the way to predict y^c_{n+1} and y^c_{n+2} is to use an explicit method. In order to satisfy the explicit formula, the derivation of the predictor formulae will be one order less and need fewer point than the corrector formulae.

Thus, the predictor formulae can be generated based on the LMM as follows for the first point predictor formula,

$$\sum_{i=2}^{3}\alpha_i y(x+ih) = h\sum_{i=0}^{1}\beta_i y'(x+ih),$$

and the second point of the predictor formula as below,

$$\sum_{i=2}^{2}\alpha_i y(x+ih) + \sum_{i=4}^{4}\alpha_i y(x+ih) = h\sum_{i=0}^{1}\beta_i y'(x+ih).$$

Despite the predictor formula being of order two, the Taylor expansion terms will be truncated at the second derivative. Therefore, the predictor formulae would be determined as follows,

$$y^p_{n+1} = y_n + h\left(-\frac{3}{2}F_{n-2} + \frac{5}{2}F_{n-1}\right), \quad (19)$$

$$y^p_{n+2} = y_n + h(-4F_{n-2} + 6F_{n-1}). \quad (20)$$

Formulae for the 2PBM

Predictor,

$$y^p_{n+1} = y_n + h\left(-\frac{3}{2}F_{n-2} + \frac{5}{2}F_{n-1}\right),$$

$$y^p_{n+2} = y_n + h(-4F_{n-2} + 6F_{n-1}).$$

Corrector,

$$y^c_{n+1} = y_n + h\left(-\frac{1}{36}F_{n-2} + \frac{7}{12}F_n + \frac{4}{9}F_{n+1}\right),$$

$$y^c_{n+2} = y_n + h\left(\frac{1}{18}F_{n-2} + \frac{16}{9}F_{n+1} + \frac{1}{6}F_{n+2}\right).$$

ORDER OF THE METHOD

By using the matrix difference equation, as shown below, the derived method could be construed.

$$\alpha Y_N = \beta h F_N, \quad (22)$$

where,

$$Y_N = \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \\ y_{n+1} \\ y_{n+2} \end{bmatrix}, \qquad F_N = \begin{bmatrix} F_{n-2} \\ F_{n-1} \\ F_n \\ F_{n+1} \\ F_{n+2} \end{bmatrix}.$$

The 2PBM method can be indicated in the form of the LMM (4). The general version of (5) can be represent as

$$L[y(x);h] = C_0 y(x) + C_1 h y'(x) + C_2 h^2 y''(x) + \cdots + C_p h^p y^p(x) + \cdots, \tag{23}$$

whereas

$$C_p = \sum_{d=0}^{k} \frac{d^p \alpha_d}{p!} - \sum_{d=0}^{k} \frac{d^{(p-1)} \beta_d}{(p-1)!}, \qquad p = 0,1,2,\ldots \tag{24}$$

where α_d, β_d are, respectively, the vector columns of the 2PBM method in the matric form.

Definition 1

The LMM is order p where $C_0 = C_1 = C_2 = \cdots = C_{p-1} = C_p = 0$ and $C_{p+1} \neq 0$, respectively, (Lambert 1973).

Consequently, in the matrix difference form, the derived method is drafted as (21)

$$\begin{bmatrix} 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_n \\ y_{n+1} \\ y_{n+2} \end{bmatrix} = h \begin{bmatrix} -\frac{1}{36} & 0 & \frac{7}{12} & \frac{4}{9} & 0 \\ \frac{1}{18} & 0 & 0 & \frac{16}{9} & \frac{1}{6} \end{bmatrix} \begin{bmatrix} F_{n-2} \\ F_{n-1} \\ F_n \\ F_{n+1} \\ F_{n+2} \end{bmatrix}. \tag{25}$$

The associated set of coefficients from equation (24) implemented to achieve

$$C_0 = \sum_{d=0}^{4} \frac{d^0}{0!} \alpha_d = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$C_1 = \sum_{d=0}^{4} \frac{d^1}{1!} \alpha_d - \sum_{d=0}^{4} \frac{d^0}{0!} \beta_d = 0\alpha_0 + 1\alpha_1 + 2\alpha_2 + 3\alpha_3 + 4\alpha_4 - (\beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4),$$

$$= 1\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2\begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4\begin{bmatrix} 0 \\ 1 \end{bmatrix} - \left(\begin{bmatrix} -\frac{1}{36} \\ \frac{1}{18} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{7}{12} \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{4}{9} \\ \frac{16}{9} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{6} \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$C_2 = \sum_{d=0}^{4} \frac{d^2}{2!} \alpha_d - \sum_{d=0}^{4} \frac{d^1}{1!} \beta_d,$$

$$= \frac{1}{2}(1^2\alpha_1 + 2^2\alpha_2 + 3^2\alpha_3 + 4^2\alpha_4) - (1\beta_1 + 2\beta_2 + 3\beta_3 + 4\beta_4),$$

$$= \frac{1}{2}\left(1^2\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2^2\begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3^2\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4^2\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) - \left(1\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2\begin{bmatrix} \frac{7}{12} \\ 0 \end{bmatrix} + 3\begin{bmatrix} \frac{4}{9} \\ \frac{16}{9} \end{bmatrix} + 4\begin{bmatrix} 0 \\ \frac{1}{6} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$C_3 = \sum_{d=0}^{4} \frac{d^3}{3!} \alpha_d - \sum_{d=0}^{4} \frac{d^2}{2!} \beta_d,$$

$$= \frac{1}{6}(1^3\alpha_1 + 2^3\alpha_2 + 3^3\alpha_3 + 4^3\alpha_4) - \frac{1}{2}(1^2\beta_1 + 2^2\beta_2 + 3^2\beta_3 + 4^2\beta_4),$$

$$= \frac{1}{6}\left(1^3\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2^3\begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3^3\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4^3\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) - \frac{1}{2}\left(1^2\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2^2\begin{bmatrix} \frac{7}{12} \\ 0 \end{bmatrix} + 3^2\begin{bmatrix} \frac{4}{9} \\ \frac{16}{9} \end{bmatrix} + 4^2\begin{bmatrix} 0 \\ \frac{1}{6} \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$C_4 = \sum_{d=0}^{4} \frac{d^4}{4!} \alpha_d - \sum_{d=0}^{4} \frac{d^3}{3!} \beta_d,$$

$$= \frac{1}{24}(1^4\alpha_1 + 2^4\alpha_2 + 3^4\alpha_3 + 4^4\alpha_4) - \frac{1}{6}(1^3\beta_1 + 2^3\beta_2 + 3^3\beta_3 + 4^3\beta_4),$$

$$= \frac{1}{24}\left(1^4\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2^4\begin{bmatrix} -1 \\ -1 \end{bmatrix} + 3^4\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 4^4\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) - \frac{1}{6}\left(1^3\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2^3\begin{bmatrix} \frac{7}{12} \\ 0 \end{bmatrix} + 3^3\begin{bmatrix} \frac{4}{9} \\ \frac{16}{9} \end{bmatrix} + 4^3\begin{bmatrix} 0 \\ \frac{1}{6} \end{bmatrix}\right)$$

$$= \begin{bmatrix} -\frac{5}{72} \\ \frac{2}{9} \end{bmatrix},$$

while $C_4 \neq 0$.

$$C_{p+1} = C_4 = \begin{bmatrix} -\frac{5}{72} & \frac{2}{9} \end{bmatrix}^T. \tag{26}$$

Regarding Definition 1, the 2PBM method can be concluded that it satisfies an order three and C_4 is the vector of the error constant.

Definition 2

The LMM is consistent if the method holds an order of at least one (Lambert 1973).

Recognizing the order of method is three, which is greater than one, thus this proposed method is consistent.

Definition 3

The LMM is zero-stable if the first characteristics polynomial $\rho(r)$ specified as $\rho(r) = \det\left|\sum_{j=0}^{k} A^{(j)} r^{(k-j)}\right| = 0$ having roots such that $|r_j| \leq 1$ and if $|r_j| = 1$, the multiplicity must not exceed two. The roots define as r_j where $j = 1, 2, \ldots, k$.

Zero stability examines method stability at step size limit $h \rightarrow 0$ while the stability theory investigates method stability if h is a fixed non-zero value.

The 2PBM method can be drafted in the following matrix form,

$$A^0 Y_M - A^1 Y_{M-1} = 0,$$
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix} = 0. \tag{27}$$

The first characteristics polynomial examined to explain the zero stability as follow,

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

Since $|r_j| \leq 1$, the method achieved zero stability.

Definition 4

The LMM is convergent when it satisfies with zero-stable and consistent of the method.

The zero-stable and consistency of the method have been proved previously, hence, the 2PBM method is convergent.

IMPLEMENTATION

There are three parts of implementing the delay integro-differential equation in this study. The implementation of the proposed method uses C code to approximate the numerical solution of the problem.

The procedure consists of locating the delay arguments to determine the delay solution. The solution of delay depends on the location of $(x - \tau)$. The location may recall the previously calculated delay solution $y(x - \tau)$ stored since the constant step size implementation. Furthermore, the initial function, $\phi(x)$ implemented to compute $y(x - \tau)$ when $(x - \tau) \leq a$. If $(x - \tau) \geq a$, Lagrange interpolation polynomial will be adapted in finding the solution for the delay solution $y(x - \tau)$.

Therefore, the 2PBM method will compute the approximate solution for the delay integro-differential equation. The implementation of the 2PBM method is achieved by analysing the numerical scheme used to solve the ODE part of the delay integro-differential equation.

Some numerical integration methods are adapted to deal with an integral part of the delay integro-differential equation since it is impractical to solve this part explicitly. Consequently, the formula of the numerical integration method uses is composite Simpson rule, and it can be written as follow;

$$z_{n+1} = \frac{h}{3} \sum_{i=0}^{n+1} \omega_i^s \, K\left(x_{n+1}, x_i, y(x_i), y_{delay}(x_i)\right),$$

$$\begin{aligned} z_{n+2} = \frac{h}{3} \sum_{i=0}^{n+1} \omega_i^s \, K\left(x_{n+1}, x_i, y(x_i), y_{delay}(x_i)\right) \\ + \frac{h}{6}\Bigg(K\left(x_{n+2}, x_{n+1}, y(x_{n+1}), y_{delay}(x_{n+1})\right) \\ + 4K\left(x_{n+2}, x_{n+\frac{3}{2}}, y(x_{n+\frac{3}{2}}), y_{delay}(x_{n+\frac{3}{2}})\right) \\ + K\left(x_{n+2}, x_{n+2}, y(x_{n+2}), y_{delay}(x_{n+2})\right)\Bigg), \end{aligned}$$

where the ω_i^s are Simpson's rule weight, 1, 4,2,4,…,2,4,1. The unknown value of $y_{n+\frac{3}{2}}$ evaluate by quadratic interpolation,

$$y_{n+\frac{3}{2}} = \frac{1}{16} y_{n-1} - \frac{5}{16} y_n + \frac{15}{6} y_{n+1} + \frac{5}{16} y_{n+2}.$$

The procedure is repeated, and the details of the method as described in the algorithm.

STABILITY REGION

The following test equation evaluated the stability properties of the proposed method,

$$y'(x) = \xi y(x - \tau) + \nu \int_0^x y(u) \, du. \tag{28}$$

Assume $\tau = mh$ where the internal staged are not required to estimate $y(x - \tau)$ and substitute $y(x - \tau) = Y_{N-m}$. Following the multistep method in the matrix form;

$$\sum_{k=0}^{2} A_k \, Y_{N+k} = h \sum_{k=0}^{2} B_k \, F_{N+k}, \tag{29}$$

$$\begin{aligned} A_0 Y_N + A_1 Y_{N+1} + A_2 Y_{N+2} = hB_0 F_N + \\ hB_1 F_{N+1} + hB_2 F_{N+2}, \end{aligned} \tag{30}$$

where

$$Y_N = \begin{bmatrix} y_{n-3} \\ y_{n-2} \end{bmatrix}, \quad Y_{N+1} = \begin{bmatrix} y_{n-1} \\ y_n \end{bmatrix}, \quad Y_{N+2} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \end{bmatrix},$$

$$F_N = \begin{bmatrix} F_{n-3} \\ F_{n-2} \end{bmatrix}, \quad F_{N+1} = \begin{bmatrix} F_{n-1} \\ F_n \end{bmatrix}, \quad F_{N+2} = \begin{bmatrix} F_{n+1} \\ F_{n+2} \end{bmatrix},$$

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

$$B_0 = \begin{bmatrix} 0 & -\frac{1}{36} \\ 0 & \frac{1}{18} \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & \frac{7}{12} \\ 0 & 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} \frac{4}{9} & 0 \\ \frac{16}{9} & \frac{1}{6} \end{bmatrix}.$$

From the test equation (28), we obtain

$$F_N = y'(x) = \xi y(x-\tau) + \nu \int_0^x y(u)\, du, \tag{31}$$

$$F_N = \xi Y_{N-m} + \nu \int_0^x y(u)\, du\,. \tag{32}$$

The numerical integration method will be adapted into an integral part of the delay integro-differential equation, and Simpson's rule will be applied;

$$\int_0^x y(u)\, du = h\left(\frac{1}{3}Y_{N-2} + \frac{4}{3}Y_{N-1} + \frac{1}{3}Y_N\right). \tag{33}$$

By implementing the test equation into the 2PBM method and yield;

$$\begin{aligned} &A_0Y_N + A_1Y_{N+1} + A_2Y_{N+2} \\ &\quad = hB_0\left(\xi Y_{N-m} + \nu h\left(\frac{1}{3}Y_{N-2} + \frac{4}{3}Y_{N-1} + \frac{1}{3}Y_N\right)\right) \\ &\quad + hB_1\left(\xi Y_{N+1-m} + \nu h\left(\frac{1}{3}Y_{N-1} + \frac{4}{3}Y_N + \frac{1}{3}Y_{N+1}\right)\right) \\ &\quad + hB_2\left(\xi Y_{N+2-m} + \nu h\left(\frac{1}{3}Y_N + \frac{4}{3}Y_{N+1} + \frac{1}{3}Y_{N+2}\right)\right), \end{aligned} \tag{34}$$

Rearranging the equation yields,

$$\begin{aligned} &\left(A_2 - \frac{1}{3}H_2B_2\right)Y_{N+2} + \left(A_1 - \frac{1}{3}H_2B_1 - \frac{4}{3}H_2B_2\right)Y_{N+1} \\ &\quad + \left(A_0 - \frac{1}{3}H_2B_0 - \frac{4}{3}H_2B_1 - \frac{1}{3}H_2B_2\right)Y_N + \\ &\quad \left(-\frac{4}{3}\nu h^2B_0 - \frac{1}{3}\nu h^2B_1\right)Y_{N-1} \\ &\quad + \left(-\frac{1}{3}\nu h^2B_0\right)Y_{N-2} - \xi hB_2Y_{N+2-m} - \\ &\quad \xi hB_1Y_{N+1-m} - \xi hB_0Y_{N-m} = 0. \end{aligned} \tag{35}$$

Substituting $H_1 = \xi h$, and $H_2 = \nu h^2$,

$$\begin{aligned} &\left(A_2 - \frac{1}{3}H_2B_2\right)Y_{N+2} + \left(A_1 - \frac{1}{3}H_2B_1 - \frac{4}{3}H_2B_2\right)Y_{N+1} \\ &\quad + \left(A_0 - \frac{1}{3}H_2B_0 - \frac{4}{3}H_2B_1 - \frac{1}{3}H_2B_2\right)Y_N \\ &\quad + \left(-\frac{4}{3}H_2B_0 - \frac{1}{3}H_2B_1\right)Y_{N-1} \\ &\quad + \left(-\frac{1}{3}H_2B_0\right)Y_{N-2} - H_1B_2Y_{N+2-m} - \\ &\quad - H_1B_1Y_{N+1-m} - H_1B_0Y_{N-m} = 0. \end{aligned} \tag{36}$$

Therefore, the stability polynomial determines as follows;

$$\begin{aligned} \pi(H_1, H_2; r) = \det\Bigg(&\left(A_2 - \frac{1}{3}H_2B_2\right)r^{m+2} + \\ &\left(A_1 - \frac{1}{3}H_2B_1 - \frac{4}{3}H_2B_2\right)r^{m+1} \\ &+ \left(A_0 - \frac{1}{3}H_2B_0 - \frac{4}{3}H_2B_1 - \frac{1}{3}H_2B_2\right)r^m + \\ &\left(-\frac{4}{3}H_2B_0 - \frac{1}{3}H_2B_1\right)r^{m-1} \\ &+ \left(-\frac{1}{3}H_2B_0\right)r^{m-2} - H_1B_2r^2 - H_1B_1r^1 - H_1B_0r^0\Bigg). \end{aligned} \tag{37}$$

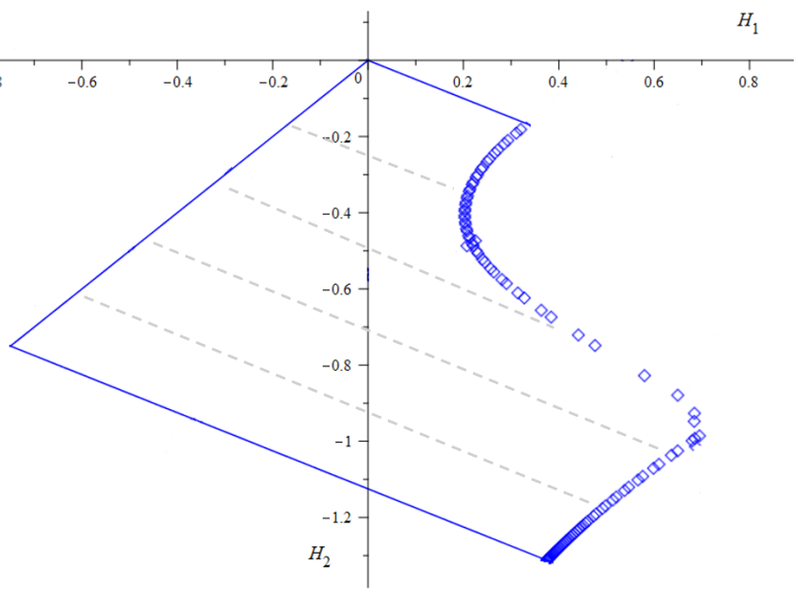

FIGURE 1. Stability region in the H_1- H_2plane

Consider the numerical stability region for fixed m=10.

$$\pi(H_1, H_2; r) =$$
$$\frac{2}{27}r^4H_1^2 - \frac{28}{27}r^3H_1^2 + \frac{2}{27}r^2H_1^2 + \frac{2}{243}r^{24}H_2^2 - \frac{4}{81}r^{23}H_2^2 - \frac{62}{81}r^{22}H_2^2$$
$$- \frac{472}{243}r^{21}H_2^2 - \frac{62}{81}r^{20}H_2^2 - \frac{4}{81}r^{19}H_2^2 + \frac{2}{243}r^{18}H_2^2 + \frac{4}{81}r^{14}H_1H_2$$
$$- \frac{8}{3}r^{12}H_1H_2 - \frac{40}{81}r^{13}H_1H_2 - \frac{40}{81}r^{11}H_1H_2 + \frac{4}{81}r^{10}H_1H_2 - \frac{11}{18}r^{14}H_1$$
$$- \frac{4}{3}r^{13}H_1 - \frac{1}{18}r^{12}H_1 - \frac{11}{54}r^{24}H_2 - \frac{34}{27}r^{23}H_2 - 2r^{22}H_2 - \frac{14}{27}r^{21}H_2$$
$$- \frac{1}{54}r^{20}H_2 + r^{24} - r^{23} = 0. \tag{38}$$

The boundary locus technique of the absolute stability region in H_1- H_2 plane illustrated by replacing r with -1, 0, 1 and $r = \cos\theta + i \sin\theta$ for $0 \le \theta \le 2\pi$ in stability polynomial. The region's points obtained by separating the real and the imaginary part of $r = \cos\theta + i \sin\theta$ was then solved simultaneously. The stability region is absolutely stable as the set of all roots in the stability polynomial satisfies $|r| \le 1$ and lies within the region's boundary in Figure 1.

CONVERGENCE OF THE METHOD

This section will discuss the convergence of the 2PBM method. The efficiency of implementing the 2PBM approach to any differential problem can be explained by studying the convergence of the method. The method is said to have converged when the approximate values produced are closer to the exact values given as:

$$\lim_{h\to 0} Y_{n+1} = y(x_{n+1}),$$
$$\lim_{h\to 0} Y_{n+2} = y(x_{n+2}), \tag{39}$$
$$\lim_{h\to 0} Z_{n+2} = z(x_{n+2}),$$

where Y_{n+1}, Y_{n+2} and Z_{n+2} are the approximate solutions. Hence, the approximate solution for Z_{n+2} will be referred to Simpson's rule and the formulae of the 2PBM and Simpson's rule are given as follows by letting $Y(x_{n+1}) = Y_{n+1}$;

$$Y_{n+1} = Y_n + h\left(-\frac{1}{36}F_{n-2} + \frac{7}{12}F_n + \frac{4}{9}F_{n+1}\right),$$
$$Y_{n+2} = Y_n + h\left(\frac{1}{18}F_{n-2} + \frac{16}{9}F_{n+1} + \frac{1}{6}F_{n+2}\right).$$
$$Z_{n+2} = Z_n + h\left(\frac{1}{3}Y_n + \frac{4}{3}Y_{n+1} + \frac{1}{3}Y_{n+2}\right).$$

While, the exact solution will be acknowledged as follows,

$$y(x_{n+1}) = y(x_n) + h\left(-\frac{1}{36}f_{n-2} + \frac{7}{2}f_n + \frac{4}{9}f_{n+1}\right) - \frac{5}{72}h^4Y^{(4)}(t_n),$$
$$y(x_{n+2}) = y(x_n) + h\left(\frac{1}{18}f_{n-2} + \frac{16}{9}f_{n+1} + \frac{1}{6}f_{n+2}\right) + \frac{2}{9}h^4Y^{(4)}(t_n),$$
$$z(x_{n+2}) = z(x_n) + h\left(\frac{1}{3}y(x_n) + \frac{4}{3}y(x_{n+1}) + \frac{1}{3}y(x_{n+2})\right) -$$
$$\frac{1}{90}h^5Y^{(5)}(t_n).$$

As the result of Lipschitz condition,

$$|y'(x_n) - Y'_n| = \left|f\left(x, y'(x_n), y'(x_{n+m}), z(x_n)\right) -$$
$$F(x, Y'_n, Y'_{n+m}, Z_n)\right| \le L|d_n|,$$
$$|z'(x_n) - Z'_n| = \left|f\left(x, y(x_n), y(x_{n+m}), z'(x_n)\right) -$$
$$F(x, Y_n, Y_{n+m}, Z'_n)\right| \le L|a_n|.$$

where d_n will refer to the function y and a_n will represent the function z. As shown below, the approximate solution subtracted from the exact solution.

$$y(x_{n+1}) - Y_{n+1}$$
$$= y(x_n) - Y_n + \left(-\frac{1}{36}hf_{n-2} + \frac{7}{2}hf_n + \frac{4}{9}hf_{n+1}\right)$$
$$- \left(-\frac{1}{36}hF_{n-2} + \frac{7}{12}hF_n + \frac{4}{9}hF_{n+1}\right) - \frac{5}{72}h^4Y^{(4)}(t_n),$$
$$y(x_{n+2}) - Y_{n+2}$$
$$= y(x_n) - Y_n + \left(\frac{1}{18}hf_{n-2} + \frac{16}{9}hf_{n+1} + \frac{1}{6}hf_{n+2}\right)$$
$$- \left(\frac{1}{18}hF_{n-2} + \frac{16}{9}hF_{n+1} + \frac{1}{6}hF_{n+2}\right) + \frac{2}{9}h^4Y^{(4)}(t_n),$$
$$z(x_{n+2}) - Z_{n+2}$$
$$= z(x_n) - Z_n + \left(\frac{1}{3}hy(x_n) + \frac{4}{3}hy(x_{n+1}) + \frac{1}{3}hy(x_{n+2})\right)$$
$$- \left(\frac{1}{3}hY_n + \frac{4}{3}hY_{n+1} + \frac{1}{3}hY_{n+2}\right) - \frac{1}{90}h^5Y^{(5)}(t_n).$$

Denoting $y(x_{n+1})$ - $Y_{n+1} = d_{n+1}$, $y(x_{n+2})$ - $Y_{n+2} = d_{n+2}$, $y(x_n)$ -$Y_n = d_n$,…, hence $z(x_{n+2})$ - $Z_{n+2} = a_{n+2}$ and attain,

$$|d_{n+1}| \le |d_n| - \frac{5}{72}h^4\left|Y^{(4)}(t_n)\right|,$$
$$|d_{n+2}| \le |d_n| + \frac{2}{9}h^4\left|Y^{(4)}(t_n)\right|,$$
$$|a_{n+2}| \le |a_n| - \frac{1}{90}h^5\left|Y^{(5)}(t_n)\right|.$$

and as $h \longrightarrow 0$ would give,

$$|d_{n+1}| \le |d_n| \Rightarrow Y_{n+1} - y_{n+1} \le Y_n - y_n \Rightarrow Y_{n+1} - Y_n \le y_{n+1} - y_n,$$
$$|d_{n+2}| \le |d_n| \Rightarrow Y_{n+2} - y_{n+2} \le Y_n - y_n \Rightarrow Y_{n+2} - Y_n \le y_{n+2} - y_n,$$
$$|a_{n+2}| \le |a_n| \Rightarrow Z_{n+2} - z_{n+2} \le Z_n - z_n \Rightarrow Z_{n+2} - Z_n \le z_{n+2} - z_n$$

which demonstrates that the approximate and exact solutions are equivalent. Hence, $|d_{n+1}| \le |d_n|$, $|d_{n+2}| \le |d_n|$ and $|a_{n+2}| \le |a_n|$ satisfy the convergence condition in equation (39). Therefore, the 2PBM is considered to be converged.

ALGORITHM OF THE 2PBM METHOD

Step 1: Set N, $x_0 = a$, $x_n = b$, $h = \frac{b-a}{N}$, $y_0 = y(x_0)$, $z_0 = z(x_0, y(x_0))$.

Step 2 : For $n = 0$, compute delay argument, $(x - \tau)$, delay solution, $y(x - \tau)$ and function F_0.

Step 3 : For $n = 1$ to 2, enumerate $x_n = x_0 + nh$, the starting value using Runge-Kutta method as the ODE part and Simpson's 1/3 rule for the integral part. Compute the delay solution $y(x_n - \tau)$ and the function F_n.

Step 4 : Set $n = 3$,
For i = 0 to 2, enumerate $x_{i+3} = x_n + ih$.
Step 5 : Compute the predictor approximation for y^p_{n+1} and y^p_{n+2} as follows,

$$y^p_{n+1} = y_n + h\left(-\frac{3}{2}F_{n-2} + \frac{5}{2}F_{n-1}\right),$$
$$y^p_{n+2} = y_n + h(-4F_{n-2} + 6F_{n-1}).$$

- Locate the position of the delay argument $y(x - \tau)$ via the initial function.
- Use the composite Simpson's rule to approach the integral part and estimate the F^p_{n+1} and F^p_{n+2}.

Step 6 : Compute the corrector approximate solution for y^c_{n+1} and y^c_{n+2}

$$y^c_{n+1} = y_n + h\left(-\frac{1}{36}F_{n-2} + \frac{7}{12}F_n + \frac{4}{9}F_{n+1}\right),$$
$$y^c_{n+2} = y_n + h\left(\frac{1}{18}F_{n-2} + \frac{16}{9}F_{n+1} + \frac{1}{6}F_{n+2}\right).$$

- Locate the delay argument position and compute the delay function, $y(x - \tau)$ use the the initial function.
- The composite Simpson's rule is implemented for the solution of the integral part and estimate the F^c_{n+1} and F^c_{n+2}.

Step 7 : Repeat Step 4-6.

Step 8 : Complete.

ALGORITHM OF THE RKS METHOD

Step 1: Set N, $x_0 = a$, $x_n = b$, $h = \frac{b-a}{N}$, $y_0 = y(x_0)$, $z_0 = z(x_0, y(x_0), y_{delay}(x_0))$.

Step 2 : For $n=0$, compute delay argument, $(x-\tau)$, delay solution, $y(x - \tau)$ and function F_0.

Step 3 : For $n = 0, \ldots, N$,
Compute the numerical solution for the Runge-Kutta 3rd order.

Step 4 : $k_1 = hF\left(x_n, \tilde{y}(x_n), \tilde{y}_{delay}(x_n), \tilde{z}(x_n)\right)$,

Step 5 : $k_2 = hF\left(x_{n+\frac{1}{2}}, \tilde{y}^b_{n+\frac{1}{2}}, \widetilde{y_{delay}}^b, \tilde{z}^b_{n+\frac{1}{2}}\right)$,

where,

$$x_{n+\frac{1}{2}} = x_n + \frac{h}{2}, \qquad \tilde{y}^b_{n+\frac{1}{2}} = \tilde{y}_n + \frac{h}{2}k_1,$$

- Locate the delay argument, $\left(x_{n+\frac{1}{2}} - \tau\right)$ and calculate the delay solution, $\widetilde{y_{delay}}^b\left(x_{n+\frac{1}{2}} - \tau\right)$

$$\tilde{z}^b_{n+\frac{1}{2}} = \frac{h}{4}\left(\tilde{y}_n + \tilde{y}^b_{n+\frac{1}{2}}\right),$$

Step 6 :

$$k_3 = hF\left(x_{n+1}, \tilde{y}^p_{n+1}, \widetilde{y_{delay}}^p, \tilde{z}^p_{n+1}\right),$$

where,

$$x_{n+1} = x_n + h, \qquad \tilde{y}^p_{n+1} = \tilde{y}_n - hk_1 + 2hk_2,$$

- Locate the delay argument, $(x_{n+1} - \tau)$ and calculate the delay solution, $\widetilde{y_{delay}}^p(x_{n+1} - \tau)$

$$\tilde{z}^p_{n+1} = \frac{h}{6}\left(\tilde{y}_n + 4\tilde{y}^b_{n+\frac{1}{2}} + \tilde{y}^p_{n+1}\right),$$

Step 7 : Thus, the third order Runge-Kutta

$$y_{n+1} = y_n + \frac{h}{6}[k_1 + 4k_2 + k_3].$$

Step 8 : Locate the new delay argument when x_{n+1} and find the delay solution, $y_{delay}(x_{n+1} - \tau)$.

Step 9 : For z_1 where $n = 0$, the trapezoidal rule will be applied since there is not enough point to calculate,

$$z_1 = \frac{h}{2}(y_0 + y_1).$$

For $n = 1$, Simpson's rule will be applied to approximate the integral component

$$z_{n+1} = \frac{h}{3}(y_n + 4y_{n+1} + y_{n+2}).$$

Step 10 : Hence, estimate the $F_{n+1.}$
Step 11 : Repeat Step 3-10.
Step 12 : Complete.

ALGORITHM OF THE ABMS METHOD

Step 1 : Set N, $x_0 = a$, $x_n = b$, $h = \frac{b-a}{N}$, $y_0 = y(x_0)$, $z_0 = z(x_0, y(x_0))$.

Step 2 : For $n = 0$, compute delay argument, $(x - \tau)$, delay solution, $y(x - \tau)$ and function F_0.

Step 3 : For $n = 1$,
$x_1 = x_0 + h$,
$y_{n+1} = y_n + hF_0$,
Approximate $y_{delay}(x_{n+1} - \tau)$,
$z_{n+1} = \frac{h}{2}(y_0 + y_1)$
Calculate $F(x_1, y_1, y_{delay}(x_{n+1} - \tau), z_1)$

Step 4 : For $n = 2, \ldots, N$.
$x_{n+1} = x_n + nh$,

Applied Adam Bashforth two-step explicit method,

$$y_{n+1}^p = y_n + \frac{h}{2}\Big(3F\big(x_n, y_n, y_{delay}(x_n - \tau), z_n\big) - F\big(x_{n-1}, y_{n-1}, y_{delay}(x_{n-1} - \tau), z_{n-1}\big)\Big),$$

Locate the delay argument and approximate $y^p_{delay}(x_{n+1} - \tau)$,

$$z_{n+1}^p = \frac{h}{3}\big(y_{n-1} + 4y_n + y_{n+1}^p\big),$$

Calculate $F^p\big(x_{n+1}, y_{n+1}^p, y_{delay}^p(x_{n+1} - \tau), z_{n+1}^p\big)$.

Step 5 : Applied Adam Moulton two-step implicit method,

$$y_{n+1}^c = y_n + \frac{h}{12}\Big(5F^p\big(x_{n+1}, y_{n+1}^p, y_{delay}^p(x_{n+1} - \tau), z_{n+1}^p\big) + 8F\big(x_n, y_n, y_{delay}(x_n - \tau), z_n\big) - F\big(x_{n-1}, y_{n-1}, y_{delay}(x_{n-1} - \tau), z_{n-1}\big)\Big),$$

Locate the delay argument and approximate $y_{delay}^c(x_{n+1} - \tau)$,

$$z_{n+1}^c = \frac{h}{3}(y_{n-1} + 4y_n + y_{n+1}^c),$$

Calculate $F^c\big(x_{n+1}, y_{n+1}^c, y_{delay}^c(x_{n+1} - \tau), z_{n+1}^c\big)$.

Step 6 : Repeat Step 4-5.

Step 7 : Complete.

TABLE 1. Comparison of the numerical results for solving Problem 1

x	Absolute error at $h = 0.1$		
	2PBM	Galerkin	RKS
0.0	0.0000(+00)	1.3900(-02)	0.0000(+00)
0.1	4.0277(-06)	7.0000(-05)	6.9444(-08)
0.2	2.2222(-06)	1.3760(-02)	2.7777(-07)
0.3	2.2222(-06)	2.1420(-02)	1.2708(-05)
0.4	2.2222(-06)	4.1420(-02)	6.2222(-05)
0.5	2.2222(-06)	5.5250(-02)	2.2159(-04)
0.6	2.2222(-06)	6.9080(-02)	6.1416(-04)
0.7	2.2222(-06)	8.2910(-02)	1.4532(-03)
0.8	2.2222(-06)	9.6740(-02)	3.0422(-03)
0.9	2.2222(-06)	1.1057(-01)	5.8143(-03)
1.0	2.2222(-06)	1.2440(-01)	1.0333(-02)
TS	6	-	10
TFC	19	-	40

TABLE 2. The numerical results of 2PBM at different step size, h for Problem 1

x		Absolute error at $h = 0.1$		
	Exact	0.1	0.01	0.001
0.0	1.0000	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.1	1.1000	4.0277(-06)	4.0277(-11)	2.2204(-16)
0.2	1.2000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.3	1.3000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.4	1.4000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.5	1.5000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.6	1.6000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.7	1.7000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.8	1.8000	2.2222(-06)	2.2222(-11)	2.2204(-16)
0.9	1.9000	2.2222(-06)	2.2222(-11)	2.2204(-16)
1.0	2.0000	2.2222(-06)	2.2222(-11)	4.4408(-16)

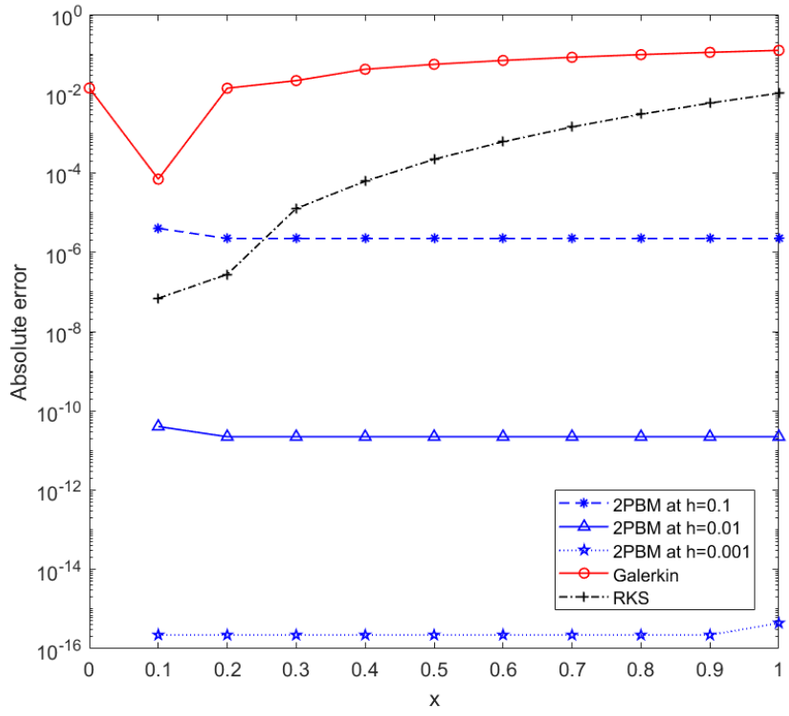

FIGURE 2. Performance graph of numerical results for Problem 1

NUMERICAL RESULTS AND DISCUSSION

Numerical results of 2PBM for solving Problem 1 - 6 are summarized in Tables 1 - 10. The algorithm has been implemented using C language with constant step-size.

Problem 1

Consider (Zaidan 2012) for the constant delay type,

$$y'(x) = 1 - \frac{x^4}{3} + \int_0^x x\, u\, y(u\text{-}1)\, du,$$

$$\phi(x) = x + 1, \qquad -1 \le x \le 0.$$

where $\phi(x)$ is an arbitrary initial function for $x \in [-1,0]$, that is continuously differentiated. The exact solution is $y(x) = x + 1$ for $0 \le x \le 1$.

In Table 1, it is observed that the absolute error of 2PBM is smaller compared to Galerkin method and RKS method using $h = 0.1$. Moreover, in contrast with other methods, 2PBM obtained less function calls to complete the computation. Table 2 shows that as the step-size becomes smaller, the 2PBM achieves better accuracy. Figure 2 illustrates the absolute errors for the 2PBM at different step sizes compared to the existing results when $h = 0.1$.

Problem 2

Consider (Salih et al. 2010) for the constant delay type,

$$y'(x) = 1 + x + x^2 - xy\left(x - \frac{1}{2}\right) + \int_0^x e^{(x-u)}\, y\left(u - \frac{1}{2}\right) du,$$

$$\phi(x) = x + \frac{1}{2}, \qquad -\frac{1}{2} \le x \le 0.$$

where $\phi(x)$ is an arbitrary initial function for $x \in \left[-\frac{1}{2}, 0\right]$, that is continuously differentiated. The exact solution is $y(x) = e^x - \frac{1}{2}$ for $\left[0, \frac{1}{2}\right]$.

TABLE 3. Comparison of the numerical results for solving Problem 2

x	Absolute error at $h = 0.05$		
	2PBM	Galerkin	RKS
0.00	0.0000(+00)	1.3900(-02)	0.0000(+00)
0.05	7.9195(-06)	5.0000(-04)	8.7243(-08)
0.10	5.6958(-05)	6.0000(-04)	2.8427(-06)
0.15	5.6482(-05)	3.0000(-04)	9.4990(-06)
0.20	5.8511(-05)	1.0000(-04)	3.2361(-05)
0.25	5.7985(-05)	5.0000(-04)	8.6838(-05)
0.30	6.0228(-05)	1.0000(-04)	2.0013(-04)
0.35	5.9648(-05)	4.0000(-04)	4.0332(-04)
0.40	6.2128(-05)	1.0000(-04)	7.4008(-04)
0.45	6.1487(-05)	1.0000(-04)	1.2587(-03)
0.50	6.4228(-05)	2.8000(-03)	2.0123(-03)
TS	6	-	10
TFC	14	-	30

TABLE 4. The numerical results of 2PBM at the different step size, h for Problem 2

x	Absolute error at different h			
	Exact	0.05	0.005	0.0005
0.00	0.5000	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.05	0.5513	7.9195(-06)	5.6264(-08)	5.6267(-11)
0.10	0.6052	5.6958(-05)	5.7010(-08)	5.7015(-11)
0.15	0.6618	5.6482(-05)	5.7794(-08)	5.7802(-11)
0.20	0.7214	5.8511(-05)	5.8618(-08)	5.8629(-11)
0.25	0.7840	5.7985(-05)	5.9485(-08)	5.9498(-11)
0.30	0.8499	6.0228(-05)	6.0395(-08)	6.0412(-11)
0.35	0.9191	5.9648(-05)	6.1353(-08)	6.1373(-11)
0.40	0.9918	6.2128(-05)	6.2360(-08)	6.2383(-11)
0.45	1.0683	6.1487(-05)	6.3418(-08)	6.3444(-11)
0.50	1.1487	6.4228(-05)	6.4531(-08)	6.4561(-11)

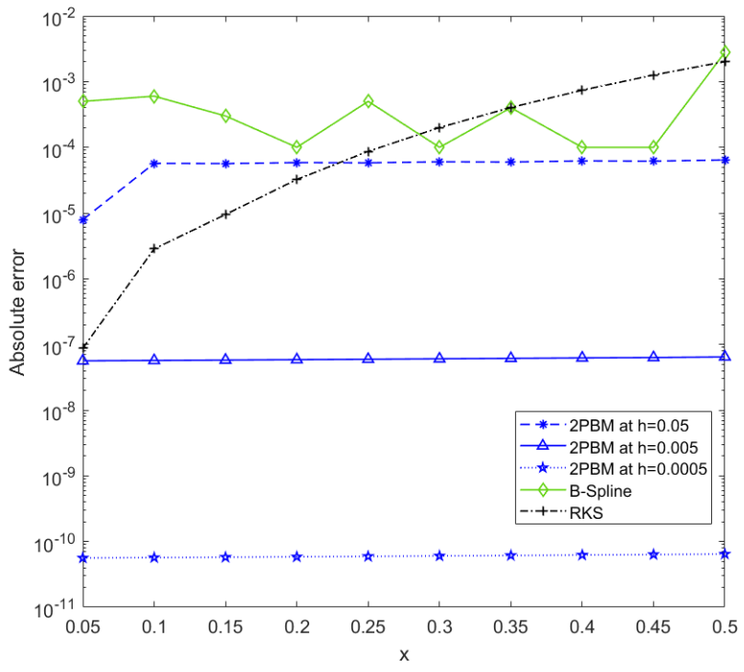

FIGURE 3. Performance graph of numerical results for Problem 2

Tables 3 - 4 display the numerical results for solving Problem 2. In Table 3, the 2PBM outperforms B-spline and RKS methods in terms of accuracy when $h = 0.05$. In Table 3, the total function calls and total steps are lesser for 2PBM compared to RKS. As the step-size getting smaller, the 2PBM achieves better accuracy, as seen in Table 4 and Figure 3.

Problem 3

Consider (Salih, Hassan & Atheer 2014) for the constant delay type,

$$y'(x) = \frac{1}{2}(1 - x + e^x) - xy\left(x - \frac{1}{2}\right) + \int_0^x e^{(x-u)}\, y\left(u - \frac{1}{2}\right) du,$$

$$\phi(x) = e^x - \frac{1}{2}, \qquad -\frac{1}{2} \le x \le 0,$$

where $\phi(x)$ is an arbitrary initial function for $x \in \left[-\frac{1}{2}, 0\right]$, that is continuously differentiated. The exact solution is $y(x) = x + \frac{1}{2}$ for $\left[0, \frac{1}{2}\right]$.

TABLE 5. Comparison of the numerical results for solving Problem 3

x	Absolute error at $h = 0.05$		
	2PBM	ABMS	RKS
0.00	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.05	5.0633(-06)	9.8970(-07)	2.1975(-08)
0.10	3.4984(-05)	5.6884(-11)	8.2500(-06)
0.15	3.4984(-05)	8.7975(-04)	2.6635(-05)
0.20	3.4984(-05)	5.5898(-04)	9.1273(-05)
0.25	3.4985(-05)	4.7720(-04)	2.1488(-04)
0.30	3.4985(-05)	3.9257(-04)	4.4532(-04)
0.35	3.4985(-05)	2.1518(-04)	8.0816(-04)
0.40	3.4985(-05)	1.2670(-04)	1.3652(-03)
0.45	3.4986(-05)	1.1482(-04)	2.1574(-03)
0.50	3.4989(-05)	2.1016(-03)	3.2631(-03)
TS	6	10	10
TFC	19	26	30

TABLE 6. The numerical results of 2PBM at the different step size, h for Problem 3

x	Absolute error at different h			
	Exact	0.05	0.005	0.0005
0.00	0.5000	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.05	0.5513	5.0633(-06)	3.3822(-08)	3.3708(-11)
0.10	0.6052	3.4984(-05)	3.3822(-08)	3.3708(-11)
0.15	0.6618	3.4984(-05)	3.3822(-08)	3.3708(-11)
0.20	0.7214	3.4984(-05)	3.3822(-08)	3.3708(-11)
0.25	0.7840	3.4985(-05)	3.3822(-08)	3.3708(-11)
0.30	0.8499	3.4985(-05)	3.3822(-08)	3.3708(-11)
0.35	0.9191	3.4985(-05)	3.3823(-08)	3.3708(-11)
0.40	0.9918	3.4985(-05)	3.3823(-08)	3.3708(-11)
0.45	1.0683	3.4986(-05)	3.3823(-08)	3.3708(-11)
0.50	1.1487	3.4989(-05)	3.3823(-08)	3.3708(-11)

In Table 5, the numerical results show that the 2PBM achieved better accuracy, less total function calls and total steps compared to ABMS and RKS. Table 6 shows that the 2PBM achieves smaller absolute error as the step size decreases, as presented in Figure 4.

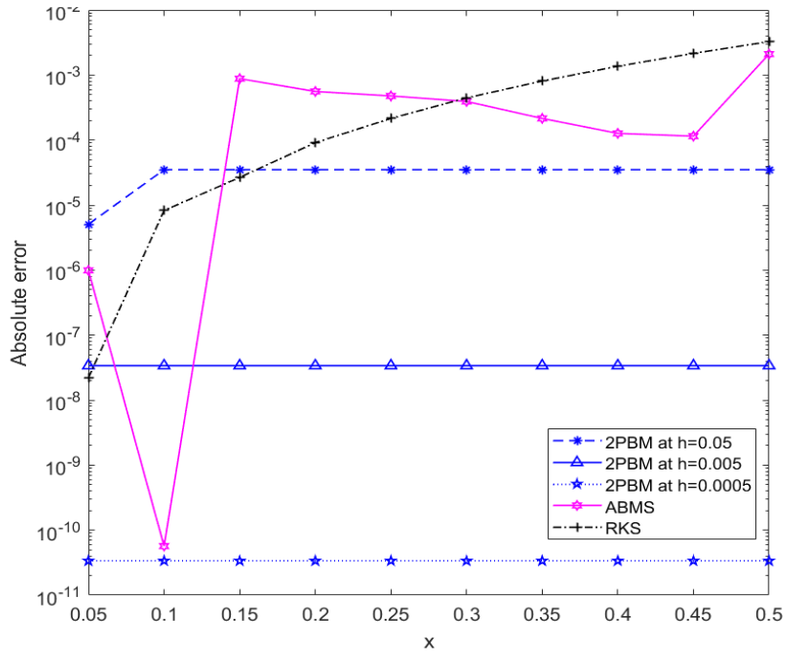

FIGURE 4. Performance graph of numerical results for Problem 3

Problem 4

Consider (Ayad 2001) for the pantograph delay type,

$$y'(x) = y\left(\frac{x}{2}\right) - e^{\left(-\frac{x}{2}\right)} + 1 + \int_0^x y(u)\,du,$$

$$\phi(x) = e^x, \quad x \le 0,$$

where $\phi(x)$ is an arbitrary initial function for $x \le 0$, that is continuously differentiated. The exact solution is $y(x) = e^x$ for [0,1].

TABLE 7. Comparison of the numerical results for solving Problem 4

x	Absolute error at $h = 0.1$		
	2PBM	ABMS	RKS
0.00	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.1	1.4244(-02)	1.1000(-01)	5.6666(-02)
0.2	1.1776(-02)	1.2000(-01)	1.4886(-02)
0.3	1.4356(-02)	1.4000(-01)	1.0439(-02)
0.4	5.6342(-02)	1.5000(-01)	1.9222(-02)
0.5	1.0513(-02)	1.8000(-01)	4.3551(-02)
TS	4	-	5
TFC	5	-	11

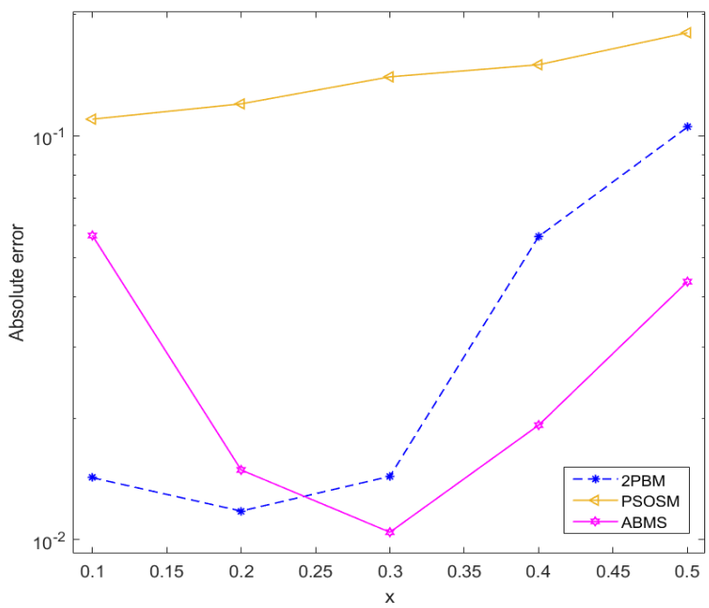

FIGURE 5. Performance graph of numerical results for Problem 4

Problem 5
Consider (Ayad 2001) for the pantograph delay type,

$$y'(x) = \left[y\left(\frac{x}{2}\right)\right]^2 - e^x + 1 + \int_0^x \left[y\left(\frac{u}{2}\right)\right]^2 du,$$

$$\phi(x) = e^x, \quad x \leq 0,$$

where ϕ(x) is an arbitrary initial function for $x \leq 0$, that is continuously differentiated. The exact solution is $y(x) = e^x$ for [0,1].

TABLE 8. Comparison of the numerical results for solving Problem 5

x	Absolute error at $h = 0.1$		
	2PBM	ABMS	RKS
0.00	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.1	2.3952(-02)	1.1000(-01)	8.5635(-02)
0.2	3.8781(-02)	1.2000(-01)	6.4571(-02)
0.3	1.0110(-01)	1.4000(-01)	9.6081(-02)
0.4	2.4317(-02)	1.6000(-01)	1.0025(-01)
0.5	4.0750(-02)	2.0000(-01)	1.1921(-01)
TS	4	-	5
TFC	5	-	11

Tables 7 - 8 show the numerical results for solving Problem 4 - 5, respectively, using 2PBM, PSOSM and ABMS methods. In the previous work (Ayad 2001), the author computed the approximate solutions for solving Problems 4 - 5 using PSOSM for $x \in [0,0.5]$, and the comparison is made based on the interval of x in the previous work. We could observe in Tables 7 - 8, the accuracy of 2PBM is better compared to PSOSM and ABMS methods. The plot of absolute errors in Tables 7 - 8 can be referred in Figures 5 - 6, respectively. Table 9 display the results of 2PBM is able to achieve smaller absolute error as the step size decreases, as presented in Figure 6 compared to PSOSM and ABMS.

The order of convergence (OC) was identified by using the general formula of

$$OC := \frac{\log\left(\frac{MAXE\ for\ h_1}{MAXE\ for\ h_2}\right)}{\log\left(\frac{h_1}{h_2}\right)}$$

The order of convergence has been calculated for four problems i.e., Problem 1 - 3 and Problem 5 when solving using 2PBM method. Problems 1, 2, and 3 have produced third order accuracy while Problem 5 could not achieve the third order accuracy due to the pantograph delay type problem. The pantograph delay solutions are being estimated with lower order Lagrange interpolating polynomial from the first iteration (at the beginning of the interval), thus the accuracy of the 2PBM is affected.

TABLE 9. The numerical results of 2PBM at the different step size, h for Problem 5

		Absolute error at different h		
x	Exact	0.1	0.01	0.001
0.00	1.0000	0.0000(+00)	0.0000(+00)	0.0000(+00)
0.10	1.1051	2.3952(-02)	4.5491(-03)	4.5916(-04)
0.20	1.2214	3.8781(-02)	5.5015(-03)	4.7452(-04)
0.30	1.3499	1.0110(-01)	5.8210(-03)	4.8962(-04)
0.40	1.4918	2.4317(-02)	6.2209(-03)	5.0380(-04)
0.50	1.6487	4.0750(-02)	6.6493(-03)	5.1625(-04)
0.60	1.8221	5.5280(-02)	7.1018(-03)	5.2602(-04)
0.70	2.0138	7.7750(-02)	7.5974(-03)	5.3192(-04)
0.80	2.2255	4.2487(-02)	8.1185(-03)	5.3280(-04)
0.90	2.4596	5.7062(-02)	8.6800(-03)	5.2685(-04)
1.00	2.7183	3.1292(-02)	9.2596(-03)	5.1227(-04)

Problem 6

Consider the nonlinear case of Volterra's population systems of 'predator-prey' with constant delay (Shakourifar & Dehghan 2008), in the form

$$N_1'(x) = N_1(x)\left(\epsilon_1 - \gamma_1 N_2(x) - \int_{x-\tau}^{x} F_1(x-u)\, N_2(u)\, du\right),$$

$$N_2'(x) = N_2(x)\left(-\epsilon_2 + \gamma_2 N_1(x) + \int_{x-\tau}^{x} F_2(x-u)\, N_1(u)\, du\right),$$

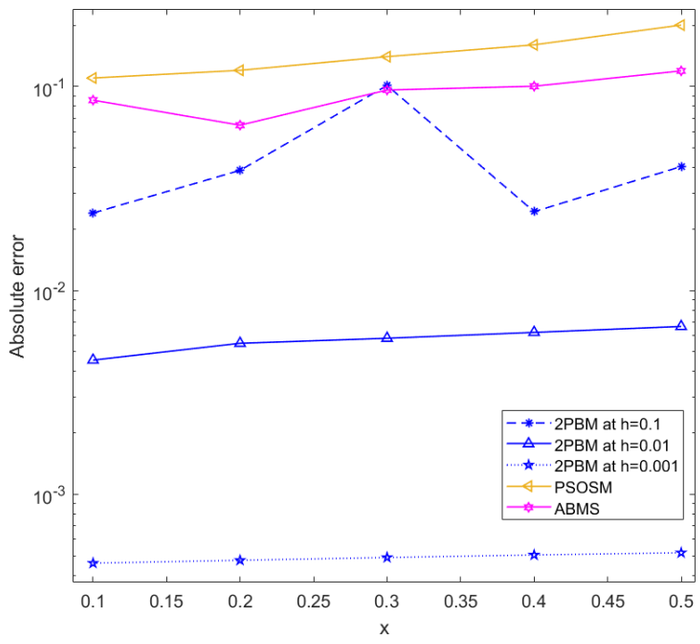

FIGURE 6. Performance graph of numerical results for Problem 5

with the following set of parameters,

$\epsilon_1 = 0.02,\ \epsilon_2 = 1.0,\ \gamma_1 = 1.0,\ \gamma_2 = 1.0,\ \tau = 0.2,\ X = 2.0.$

The initial functions are $\phi_1(x) = \phi_2(x) = 3.0$, $x \in [-\tau, 0]$ and $F_1(x) = F_2(x) = \frac{1}{2!}x^3e^{-3x}$.

This problem does not have an exact solution. $N_1(x)$ and $N_2(x)$ represent the size of two populations (prey and predator) at time $x \geq 0$. These equations can be naturally extended to describe the dynamics of ecological systems with multiple species. Various generalizations of Volterra's integro-differential system serve as the foundation for mathematical studies of ecological and chemostat delay models. The delay in chemostat models indicates that a specie's growth depends on the concentration of nutrients in the past.

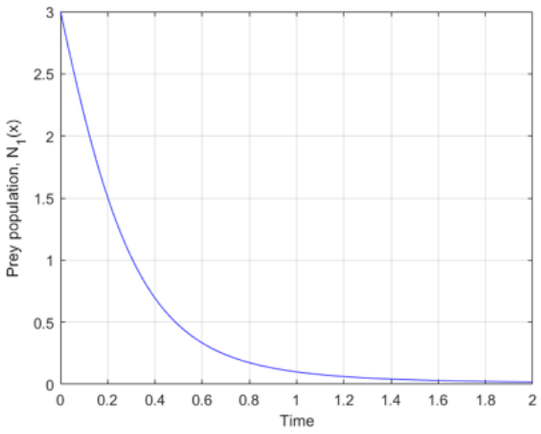

FIGURE 7. Prey populations

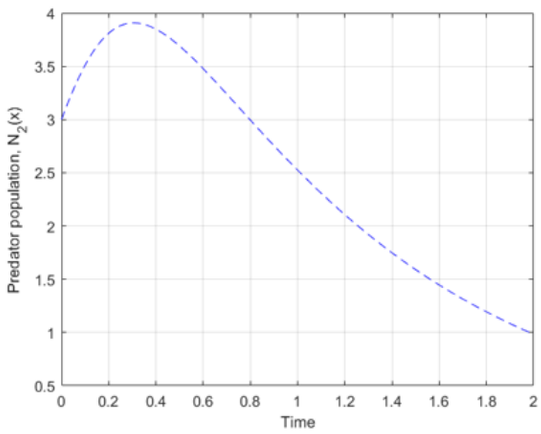

FIGURE 8. Predator populations

TABLE 10. Approximate solutions using 2PBM method

Remarks	Mesh size	$N_1(2.0)$	$N_2(2.0)$
Sample meshes	20	0.018848615861546	0.980298733086037
	40	0.018548158536900	0.981326160958517
	80	0.018484849739939	0.982598239403244
	160	0.018455005503500	0.983321844586489

Figures 7 and 8 illustrate the population of predator and prey changes over time. In Table 10, a few samples mesh sizes have been used to approximate the solutions using the 2PBM method. Table 10 shows that the approximate solutions convergence as the mesh size increased.

CONCLUSIONS

The proposed 2PBM method with a constant step-size was adequate and competitive to ensure better accuracy compared to the existing methods for solving the Volterra-type delay integro-differential equations. The results also

validate the convergence analysis where the numerical solution of the presented method indeed converges and approach the exact solution as the step-size decreases. The accuracy of the numerical results for 2PBM method are more accurate and achieved smaller absolute error as the step sizes decreased.

ACKNOWLEDGEMENTS

This research was funded by the Ministry of Higher Education, Malaysia under Fundamental Research Grant (Project code: FRGS/1/2020/STG06/UPM/01/1).

REFERENCES

Ali, H.A. 2009. Expansion method for solving linear delay integro-differential equation using B-spline functions. *Engineering and Technology Journal* 27(10): 1651-1661.

Ayad, A. 2001. The numerical solution of first order delay integro-differential equations by spline functions. *International Journal of Computer Mathematics* 77(1): 125-134.

Baharum, N.A., Majid, Z.A. & Senu, N. 2022. Boole's strategy in multistep block method for Volterra integro-differential equation. *Malaysian Journal of Mathematical Sciences* 16(2): 237-256.

Baker, C.T.H. 2000. A perspective on the numerical treatment of Volterra equations. *Journal of Computational and Applied Mathematics* 125(1-2): 217-249.

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

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

Kolmanovskii, V. & Myshkis, A. 2012. *Applied Theory of Functional Differential Equations*. New York: Kluwer Academic Publisher.

Lambert, J.D. 1973. *Computational Methods in Ordinary Differential Equations*. New York: Wiley.

Majid, Z.A. & Mohamed, N.A. 2019. Fifth order multistep block method for solving Volterra integro-differential equations of second kind. *Sains Malaysiana* 48(3): 677-684.

Mustafa, Q.K. & Mohammed, A.M. 2018. Numerical method for solving delay integro-differential equations. Research *Journal of Applied Sciences* 13: 103-105.

Qin, H., Zhiyong, W., Fumin, Z. & Jinming, W. 2018. Stability analysis of additive Runge-Kutta methods for delay-integro-differential equations. *International Journal of Differential Equations*. 2018: Article ID. 8241784. https://doi.org/10.1155/2018/8241784

Salih, R.K., Hassan, I.H. & Atheer, J.K. 2014. An approximated solutions for nth order linear delay integro-differential equations of convolution type using B-spline functions and Weddle method. *Baghdad Science Journal* 11(1): 168-177.

Salih, R.K., Hassan, I.H., Atheer, J.K. & Fuad, A.H. 2010. B-spline functions for solving nth order linear delay integro-differential equations of convolution type. *Engineering and Technology Journal* 28(23): 6801-6813.

Shakourifar, M. & Dehghan, M. 2008. On the numerical solution of nonlinear systems of Volterra integro-differential equations with delay arguments. *Computing* 82(4): 241-260.

Yüzbaşı, Ş. & Karaçayır, M. 2018. A numerical approach for solving high-order linear delay Volterra integro-differential equations. *International Journal of Computational Methods* 15(5): 1850042.

Zaidan, L.I. 2012. Solving linear delay Volterra integro-differential equations by using Galerkin's method with Bernstien polynomial. *J. Bablyon Appl. Sci.* 20: 1305-1313.

*Corresponding author; email: am_zana@upm.edu.my