

1-1-2023

## Pell-Lucas collocation method for solving a class of second order nonlinear differential equations with variable delays

ŞUAYİP YÜZBAŞI

GAMZE YILDIRIM

Follow this and additional works at: <https://journals.tubitak.gov.tr/math>



Part of the [Mathematics Commons](#)

---

### Recommended Citation

YÜZBAŞI, ŞUAYİP and YILDIRIM, GAMZE (2023) "Pell-Lucas collocation method for solving a class of second order nonlinear differential equations with variable delays," *Turkish Journal of Mathematics*: Vol. 47: No. 1, Article 3. <https://doi.org/10.55730/1300-0098.3344>  
Available at: <https://journals.tubitak.gov.tr/math/vol47/iss1/3>

This Article is brought to you for free and open access by TÜBİTAK Academic Journals. It has been accepted for inclusion in Turkish Journal of Mathematics by an authorized editor of TÜBİTAK Academic Journals. For more information, please contact [academic.publications@tubitak.gov.tr](mailto:academic.publications@tubitak.gov.tr).

## Pell-Lucas collocation method for solving a class of second order nonlinear differential equations with variable delays

Şuayip YÜZBAŞI<sup>1,\*</sup>, Gamze YILDIRIM<sup>1,2</sup>

<sup>1</sup>Department of Mathematics, Faculty of Science, Akdeniz University, Antalya, Turkey

<sup>2</sup>Department of Mathematics, Faculty of Basic Science, Gebze Technical University, Kocaeli, Turkey

Received: 27.07.2022

Accepted/Published Online: 04.10.2022

Final Version: 13.01.2023

**Abstract:** In this study, the approximate solution of the nonlinear differential equation with variable delays is investigated by means of a collocation method based on the truncated Pell-Lucas series. In the first stage of the method, the assumed solution form (the truncated Pell-Lucas polynomial solution) is expressed in the matrix form of the standard bases. Next, the matrix forms of the necessary derivatives, the nonlinear terms, and the initial conditions are written. Then, with the help of the equally spaced collocation points and these matrix relations, the problem is reduced to a system of nonlinear algebraic equations. Finally, the obtained system is solved by using MATLAB. The solution of this system gives the coefficient matrix in the assumed solution form. Moreover, the error analysis is performed. Accordingly, two theorems about the upper limit of the errors and the error estimation are given and these theorems are proven. In addition, the residual improvement technique is presented. The presented methods are applied to three examples. The obtained results are displayed in tables and graphs. Also, the obtained results are compared with the results of other methods in the literature. All results in this study have been calculated by using MATLAB.

**Key words:** Collocation method, collocation points, delay differential equations, error analysis, nonlinear differential equations, Pell-Lucas polynomials

### 1. Introduction

The differential equations are important not only in the mathematics but also in many fields such as science and engineering. Analytical solutions of many differential equations either do not exist or they are quite difficult to obtain. Therefore, the approximation methods are needed to solve these problems. For this reason, the numerical methods are important for this type of the equations. Various numerical methods [4–6, 8, 9, 18, 21, 30, 32, 38–40] related to the nonlinear differential equations are available in the literature.

On the other hand, the time-delayed differential equations are also important in many fields such as science and engineering. For example, in biological systems, the delay is of a few hundred milliseconds, which is the human response process. In a significant transaction, the delays can be very significant, whether they are measured in microseconds or less. A theoretical time delay is a property of the same form as the efficiency gained by energizing a machine. The time-delay differential equations arise in population dynamics, bioscience problems, control problems, neural network modeling, and economic systems where the decisions and the effects are separated by some time interval. The examples of some important numerical methods developed to date

\*Correspondence: syuzbasi@akdeniz.edu.tr

2010 AMS Mathematics Subject Classification: 33C47, 34A34, 65L05, 65L60, 65L70, 65Q20

for this type of the equations are the variable multistep methods [25, 26], Chebyshev polynomials for the pantograph differential equation [33], a spectral Galerkin method for the nonlinear delayed convection-diffusion reaction equations [23], the power series method [2], the differential transformation method (DTM) [20], and various other methods [3, 10, 11, 13–15, 19, 22, 27–29, 33–35, 41–47, 50].

Şahin and Sezer studied the approximate solutions of high-order linear functional differential equations with hybrid delays by using the Pell-Lucas polynomials [36]. Yüzbaşı and Yıldırım used the Pell-Lucas polynomials to find approximate solutions of two population models [48] and high-order linear Fredholm-Volterra integro-differential equations [49]. Dönmez Demir et al. investigated the solutions of Fredholm-type delay integro-differential equations with variable delays with the help of the Pell-Lucas polynomials [7]. However, until now, there is no study in the literature to examine the approximate solutions of the nonlinear differential equations with variable delays by using the Pell-Lucas polynomials.

In this work, we investigate the Pell-Lucas polynomial solutions of the nonlinear differential equations with variable delays

$$\sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t)y^{(k)}(t - \tau_{kj}(t)) + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t)y^{(p)}(t)y^{(q)}(t) = g(t) \tag{1.1}$$

under the initial conditions

$$y(a) = \lambda \quad \text{and} \quad y'(a) = \mu \tag{1.2}$$

where  $y^{(0)}(t) = y(t)$  is the unknown function and  $M_{kj}(t)$ ,  $N_{pq}(t)$  and  $g(t)$  are the continuous functions on the interval  $0 \leq a \leq t \leq b$ . Also, the variable delays  $\tau_{kj}(t)$  are the continuous functions on the interval  $0 \leq a \leq t \leq b$  for  $\tau_{kj}(t) \geq 0$ . The aim of this study is to present the Pell-Lucas collocation method to find the approximate solution of the problem (1.1)-(1.2). Therefore, we seek approximate solution of the problem (1.1)-(1.2) in the truncated Pell-Lucas series of the form

$$y(t) \cong y_N(t) = \sum_{n=0}^N a_n Q_n(t), \quad a \leq t \leq b. \tag{1.3}$$

Here,  $a_n(n = 0, 1, \dots, N)$  are the Pell-Lucas coefficients.  $N$  is a cut-off limit, which is any positive integer. Also,  $Q_n(t)(n = 0, 1, \dots, N)$  are the Pell-Lucas polynomials and it is defined by

$$Q_n(t) = \sum_{k=0}^{\lfloor n/2 \rfloor} 2^{n-2k} \frac{n}{n-k} \binom{n-k}{k} t^{n-2k}.$$

The recurrence relation of the Pell-Lucas polynomials is represented as

$$Q_n(t) = 2tQ_{n-1}(t) + Q_{n-2}(t), \quad n \geq 2,$$

where  $Q_0(t) = 2$  and  $Q_1(t) = 2t$ . In addition, the derivatives of the Pell Lucas polynomials have the recurrence relation

$$Q'_n(t) = 2tQ'_{n-1}(t) + Q'_{n-2}(t) + 2Q_{n-1}(t), \quad n \geq 2 \tag{1.4}$$

where  $Q'_0(t) = 0$  and  $Q'_1(t) = 2$ . Please have a look at [16, 17] for more properties of the Pell-Lucas polynomials.

This study is summarized as follows:

- In Section 2, the matrix form of the assumed solution and the required other matrix relations are created. Based on these relations, the matrix form of the problem (1.1)-(1.2) is formed.
- In Section 3, the Pell-Lucas collocation method is created by using the theorems in Section 2 and the collocation points. Thus, the problem (1.1)-(1.2) is reduced to an algebraic matrix system.
- In Section 4, the two theorems about the error analysis are given and proven.
- In Section 5, the methods in Sections 3 and 4 are tested for three examples by using MATLAB. The obtained results are discussed in tables and graphs.
- In Section 6, the results of this study are highlighted.

## 2. Basic matrix relations

In this section, some basic matrix relations are created to be used in the method of solution.

**Lemma 2.1** *The vector  $\mathbf{Q}_N(t)$  is written as follows:*

$$\mathbf{Q}_N(t) = \mathbf{T}_N(t)\mathbf{D}_N \tag{2.1}$$

where  $\mathbf{T}_N(t) = [ 1 \ t \ t^2 \ \dots \ t^N ]$  and if  $N$  is odd

$$\mathbf{D}_N = \begin{bmatrix} 2 & 0 & 2^0 \frac{2}{1} \binom{1}{1} & 0 & 2^0 \frac{4}{2} \binom{2}{2} & 0 & \dots & 0 \\ 0 & 2^1 \frac{1}{1} \binom{1}{0} & 0 & 2^1 \frac{3}{2} \binom{1}{1} & 0 & 2^1 \frac{5}{3} \binom{3}{2} & \dots & 2^1 \frac{N}{2} \binom{\frac{N+1}{2}}{\frac{N-1}{2}} \\ 0 & 0 & 2^2 \frac{2}{2} \binom{2}{0} & 0 & 2^2 \frac{4}{3} \binom{3}{1} & 0 & \dots & 0 \\ 0 & 0 & 0 & 2^3 \frac{3}{3} \binom{3}{0} & 0 & 2^3 \frac{5}{4} \binom{4}{1} & \dots & 2^3 \frac{N}{2} \binom{\frac{N+3}{2}}{\frac{N-3}{2}} \\ 0 & 0 & 0 & 0 & 2^4 \frac{4}{4} \binom{4}{0} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 2^N \frac{N}{N} \binom{N}{0} \end{bmatrix},$$

and if  $N$  is even

$$\mathbf{D}_N = \begin{bmatrix} 2 & 0 & 2^0 \frac{2}{1} \binom{1}{1} & 0 & 2^0 \frac{4}{2} \binom{2}{2} & 0 & \dots & 2^0 \frac{N}{\frac{N}{2}} \binom{\frac{N}{2}}{\frac{N}{2}} \\ 0 & 2^1 \frac{1}{1} \binom{1}{0} & 0 & 2^1 \frac{3}{2} \binom{1}{1} & 0 & 2^1 \frac{5}{3} \binom{3}{2} & \dots & 0 \\ 0 & 0 & 2^2 \frac{2}{2} \binom{2}{0} & 0 & 2^2 \frac{4}{3} \binom{3}{1} & 0 & \dots & 2^2 \frac{N}{2} \binom{\frac{N+2}{2}}{\frac{N-2}{2}} \\ 0 & 0 & 0 & 2^3 \frac{3}{3} \binom{3}{0} & 0 & 2^3 \frac{5}{4} \binom{4}{1} & \dots & 0 \\ 0 & 0 & 0 & 0 & 2^4 \frac{4}{4} \binom{4}{0} & 0 & \dots & 2^4 \frac{N}{\frac{N+4}{2}} \binom{\frac{N+4}{2}}{\frac{N-4}{2}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 2^N \frac{N}{N} \binom{N}{0} \end{bmatrix}.$$

**Proof:** By multiplying the vector  $\mathbf{T}_N(t)$  by the matrix  $\mathbf{D}_N$  from the right side, the vector  $\mathbf{Q}_N(t) = \mathbf{T}_N(t)\mathbf{D}_N$  is gained.

**Lemma 2.2** *The approximate solution based on the Pell-Lucas polynomials in (1.3) is expressed in the matrix form*

$$y(t) \cong y_N(t) = \mathbf{T}_N(t)\mathbf{D}_N\mathbf{A}_N \tag{2.2}$$

where

$$\mathbf{A}_N = [ a_0 \quad a_1 \quad \cdots \quad a_N ]^T, i = 0, 1, \dots, N.$$

Here, the matrices  $\mathbf{T}_N(t)$  and  $\mathbf{D}_N$  are as in Lemma (2.1).

**Proof:** The vector  $\mathbf{Q}_N(t) = \mathbf{T}_N(t)\mathbf{D}_N$  in the Lemma (2.1) is multiplied by the vector  $\mathbf{A}_N$  from the right and so the relation (2.2) is obtained.

**Lemma 2.3** *The matrix representations for the derivatives of the assumed solution form (2.2) are given as*

$$y'(t) \cong y'_N(t) = \mathbf{T}_N(t)\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N \tag{2.3}$$

and

$$y''(t) \cong y''_N(t) = \mathbf{T}_N(t)\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N \tag{2.4}$$

where

$$\mathbf{B}_N = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & N \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Here, the matrices  $\mathbf{T}_N(t)$ ,  $\mathbf{D}_N$  and  $\mathbf{A}_N$  are as in Lemmas (2.1) and (2.2).

**Proof:** Firstly, by taking the first and second derivatives of Eq. (2.2), we have, respectively

$$y'(t) \cong y'_N(t) = \mathbf{T}'_N(t)\mathbf{D}_N\mathbf{A}_N \tag{2.5}$$

and

$$y''(t) \cong y''_N(t) = \mathbf{T}''_N(t)\mathbf{D}_N\mathbf{A}_N. \tag{2.6}$$

Secondly, if the first and second derivatives of the matrix  $\mathbf{T}_N(t)$  is taken, the following relations are found, respectively

$$\mathbf{T}'_N(t) = \mathbf{T}_N(t)\mathbf{B}_N \tag{2.7}$$

and

$$\mathbf{T}''_N(t) = \mathbf{T}_N(t)\mathbf{B}_N^2. \tag{2.8}$$

Finally, by substituting the relation (2.7) in (2.5) and by substituting the relation (2.8) in (2.6), we get the matrix representations for the first and second derivatives of the assumed solution form (2.2).

**Lemma 2.4** *The matrix representations of the terms with variable delay in Eq. (1.1) are, respectively*

$$y(t - \tau_{kj}(t)) \cong y_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t)\mathbf{S}_N(-\tau_{kj}(t))\mathbf{D}_N\mathbf{A}_N, \tag{2.9}$$

$$y'(t - \tau_{kj}(t)) \cong y'_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t)\mathbf{S}_N(-\tau_{kj}(t))\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N \tag{2.10}$$

and

$$y''(t - \tau_{kj}(t)) \cong y'_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t)\mathbf{S}_N(-\tau_{kj}(t))\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N \quad (2.11)$$

where

$$\mathbf{S}_N(-\tau_{kj}(t)) = \begin{bmatrix} \binom{0}{0}(-\tau_{kj}(t))^0 & \binom{1}{0}(-\tau_{kj}(t))^1 & \cdots & \binom{N}{0}(-\tau_{kj}(t))^N \\ 0 & \binom{1}{1}(-\tau_{kj}(t))^0 & \cdots & \binom{N}{1}(-\tau_{kj}(t))^{N-1} \\ 0 & 0 & \cdots & \binom{N}{2}(-\tau_{kj}(t))^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \binom{N}{N}(-\tau_{kj}(t))^0 \end{bmatrix}.$$

Here  $\mathbf{T}_N(t)$ ,  $\mathbf{B}_N$ ,  $\mathbf{D}_N$ , and  $\mathbf{A}_N$  are as in Lemmas (2.2) and (2.3).

**Proof:** If  $t - \tau_{kj}(t)$  is written instead of  $t$  in (2.2), (2.3), and (2.4), it becomes, respectively

$$y(t - \tau_{kj}(t)) \cong y_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t - \tau_{kj}(t))\mathbf{D}_N\mathbf{A}_N, \quad (2.12)$$

$$y'(t - \tau_{kj}(t)) \cong y'_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t - \tau_{kj}(t))\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N \quad (2.13)$$

and

$$y''(t - \tau_{kj}(t)) \cong y''_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t - \tau_{kj}(t))\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N. \quad (2.14)$$

On the other hand, if we multiply the matrix  $\mathbf{T}_N(t)$  by the matrix  $\mathbf{S}_N(-\tau_{kj}(t))$  from the right side, we get the relation

$$\mathbf{T}_N(t - \tau_{kj}(t)) = \mathbf{T}_N(t)\mathbf{S}_N(-\tau_{kj}(t)). \quad (2.15)$$

Hence, by substituting the relation (2.15) in (2.12), (2.13), and (2.14), the desired results are obtained.

**Lemma 2.5** *The matrix representations of the nonlinear terms in Eq. (1.1) are expressed as follows:*

$$(y^{(0)}(t))^2 = \mathbf{T}_N(t)\mathbf{D}_N\mathbf{A}_N\mathbf{T}_N(t)\mathbf{D}_N\mathbf{A}_N, \quad (2.16)$$

$$y^{(1)}(t)y^{(0)}(t) = \mathbf{T}_N(t)\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N\mathbf{T}_N(t)\mathbf{D}_N\mathbf{A}_N, \quad (2.17)$$

$$y^{(2)}(t)y^{(0)}(t) = \mathbf{T}_N(t)\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N\mathbf{T}_N(t)\mathbf{D}_N\mathbf{A}_N, \quad (2.18)$$

$$(y^{(1)}(t))^2 = \mathbf{T}_N(t)\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N\mathbf{T}_N(t)\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N, \quad (2.19)$$

$$y^{(2)}(t)y^{(1)}(t) = \mathbf{T}_N(t)\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N\mathbf{T}_N(t)\mathbf{B}_N\mathbf{D}_N\mathbf{A}_N \quad (2.20)$$

and

$$(y^{(2)}(t))^2 = \mathbf{T}_N(t)\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N\mathbf{T}_N(t)\mathbf{B}_N^2\mathbf{D}_N\mathbf{A}_N. \quad (2.21)$$

Here  $\mathbf{T}_N(t)$ ,  $\mathbf{B}_N$ ,  $\mathbf{D}_N$ , and  $\mathbf{A}_N$  are as in Lemmas (2.2) and (2.3).

**Proof:** If the relations in Lemmas (2.2) and (2.3) are substituted, respectively, for  $y^{(0)}(t)$ ,  $y^{(1)}(t)$  and  $y^{(2)}(t)$  in Eq. (1.1), the proof is completed.

**Theorem 2.6** *It is assumed that we seek the solution of Eq. (1.1) in the solution form (2.2). Then we have the matrix relation*

$$\sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t) \mathbf{T}_N(t) \mathbf{S}_N(-\tau_{kj}(t)) \mathbf{B}_N^k \mathbf{D}_N \mathbf{A}_N + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t) \mathbf{T}_N(t) (\mathbf{B}_N)^p \mathbf{D}_N \mathbf{A}_N \mathbf{T}_N(t) (\mathbf{B}_N)^q \mathbf{D}_N \mathbf{A}_N = g(t). \tag{2.22}$$

Here,  $\mathbf{T}_N(t)$ ,  $\mathbf{S}_N(-\tau_{kj}(t))$ ,  $\mathbf{B}_N$ ,  $\mathbf{D}_N$ , and  $\mathbf{A}_N$  are as in Lemmas (2.2), (2.3) and (2.4).

**Proof:** The matrix representations (2.9), (2.10), and (2.11) and the matrix representations (2.16)–(2.21) are written in Eq. (1.1), and so Eq. (2.22) is obtained. That is, the term  $y^{(k)}(t - \tau_{kj}(t))$  in Eq. (1.1) is replaced by Eqs. (2.9), (2.10), and (2.11) and the term  $y^{(p)}(t)y^{(q)}(t)$  in Eq. (1.1) is replaced by the equation (2.16)–(2.21). Thus, the proof of the theorem is completed.  $\square$

**Lemma 2.7** *The matrix representations based on the solution form (1.3) of the initial conditions (1.2) become as follows*

$$\begin{aligned} \mathbf{U}_1 \mathbf{A}_N &= \lambda, & \mathbf{U}_1 &= \mathbf{T}_N(a) \mathbf{D}_N \\ \mathbf{U}_2 \mathbf{A}_N &= \mu, & \mathbf{U}_2 &= \mathbf{T}_N(a) \mathbf{B}_N \mathbf{D}_N \end{aligned} \tag{2.23}$$

Here,  $\mathbf{T}_N(a)$ ,  $\mathbf{B}_N$ ,  $\mathbf{D}_N$ , and  $\mathbf{A}_N$  are as in Lemmas (2.2), (2.3) and (2.4).

**Proof:** If  $a$  is written instead of  $t$  in Eq. (2.2),  $y(a) \cong y_N(a) = \mathbf{T}_N(a) \mathbf{D}_N \mathbf{A}_N$  is found. This is the matrix representation of the first condition in Eq. (1.2). So,  $\mathbf{U}_1 \mathbf{A}_N = \lambda$ ,  $\mathbf{U}_1 = \mathbf{T}_N(a) \mathbf{D}_N$ . Similarly, if  $a$  is written instead of  $t$  in Eq. (2.3),  $y'(a) \cong y'_N(a) = \mathbf{T}_N(a) \mathbf{B}_N \mathbf{D}_N \mathbf{A}_N$  is found. This is the matrix representation of the second condition in Eq. (1.2). That is,  $\mathbf{U}_2 \mathbf{A}_N = \mu$ ,  $\mathbf{U}_2 = \mathbf{T}_N(a) \mathbf{B}_N \mathbf{D}_N$ .

### 3. The Pell-Lucas collocation method

The purpose of this section is to construct a method based on the evenly spaced collocation points by using the Pell-lucas polynomials. For this purpose, firstly, evenly spaced collocation points are defined. Next, the method of solution is created by using the matrix relations in the Section (2) and the collocation points. The evenly spaced collocation points are defined by

$$t_i = a + \frac{b-a}{N}i, \quad i = 0, 1, \dots, N. \tag{3.1}$$

It is assumed that the solution of the problem (1.1)–(1.2) is sought in the form of Eq. (2.2). Firstly, if the collocation points defined in Eq. (3.1) are used in Eq. (2.22), it becomes

$$\sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t_i) \mathbf{T}_N(t_i) \mathbf{S}_N(-\tau_{kj}(t_i)) \mathbf{B}_N^k \mathbf{D}_N \mathbf{A}_N + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t_i) \mathbf{T}_N(t_i) (\mathbf{B}_N)^p \mathbf{D}_N \mathbf{A}_N \mathbf{T}_N(t_i) (\mathbf{B}_N)^q \mathbf{D}_N \mathbf{A}_N = g(t_i) \tag{3.2}$$

or

$$\sum_{k=0}^2 \sum_{j=0}^1 \mathbf{M}_{kj} \mathbf{T}_{kj} \mathbf{A}_N + \sum_{p=0}^2 \sum_{q=0}^p \mathbf{N}_{pq} \bar{\mathbf{T}}_{pq} \mathbf{A}_N = \mathbf{G} \tag{3.3}$$

where

$$\mathbf{M}_{kj} = \text{diag}(M_{kj}(t_i))_{(N+1) \times (N+1)}, \quad \mathbf{N}_{pq} = \text{diag}(N_{pq}(t_i))_{(N+1) \times (N+1)}, \quad \mathbf{G} = [g(t_0) \quad g(t_1) \quad \cdots \quad g(t_N)]^T,$$

$$\mathbf{T}_{kj} = \begin{bmatrix} \mathbf{T}_N(t_0) \mathbf{S}_N(-\tau_{kj}(t_0)) \mathbf{B}_N^k \mathbf{D}_N \\ \mathbf{T}_N(t_1) \mathbf{S}_N(-\tau_{kj}(t_1)) \mathbf{B}_N^k \mathbf{D}_N \\ \vdots \\ \mathbf{T}_N(t_N) \mathbf{S}_N(-\tau_{kj}(t_N)) \mathbf{B}_N^k \mathbf{D}_N \end{bmatrix}, \quad \bar{\mathbf{T}}_{pq} = \begin{bmatrix} T_N(t_0) (\mathbf{B}_N)^p \mathbf{D}_N \mathbf{A}_N \mathbf{T}_N(t_0) (\mathbf{B}_N)^q \mathbf{D}_N \\ T_N(t_1) (\mathbf{B}_N)^p \mathbf{D}_N \mathbf{A}_N \mathbf{T}_N(t_1) (\mathbf{B}_N)^q \mathbf{D}_N \\ \vdots \\ T_N(t_N) (\mathbf{B}_N)^p \mathbf{D}_N \mathbf{A}_N \mathbf{T}_N(t_N) (\mathbf{B}_N)^q \mathbf{D}_N \end{bmatrix}.$$

Here,  $\mathbf{T}_N(t)$ ,  $\mathbf{S}_N(-\tau_{kj}(t))$ ,  $\mathbf{B}_N$ ,  $\mathbf{D}_N$ , and  $\mathbf{A}_N$  are as in Theorem (2.6). On the other hand, Eq. (3.3) is briefly written as

$$\mathbf{W}_1 \mathbf{A}_N + \mathbf{W}_2 \mathbf{A}_N = \mathbf{G} \tag{3.4}$$

where

$$\mathbf{W}_1 = \sum_{k=0}^2 \sum_{j=0}^1 \mathbf{M}_{kj} \mathbf{T}_{kj}, \quad \mathbf{W}_2 = \sum_{p=0}^2 \sum_{q=0}^p \mathbf{N}_{pq} \bar{\mathbf{T}}_{pq}.$$

Let us express the algebraic matrix system in Eq. (3.4) as  $[\mathbf{W}_1; \mathbf{W}_2; \mathbf{G}]$ . In addition, let us write the system in Eq. (2.23) as  $[\mathbf{U}_1; \mathbf{0}; \lambda]$  and  $[\mathbf{U}_2; \mathbf{0}; \mu]$ . Here, the matrix  $\mathbf{0}$  is the zeros matrix of size  $1 \times (N + 1)$ . Now, instead of any two rows of the matrix  $[\mathbf{W}_1; \mathbf{W}_2; \mathbf{G}]$ , the matrices  $[\mathbf{U}_1; \mathbf{0}; \lambda]$  and  $[\mathbf{U}_2; \mathbf{0}; \mu]$  are written. The obtained new matrix system is called  $[\widetilde{\mathbf{W}}_1; \widetilde{\mathbf{W}}_2; \widetilde{\mathbf{G}}]$ . Hence, we have the expanded matrix

$$[\widetilde{\mathbf{W}}_1; \widetilde{\mathbf{W}}_2; \widetilde{\mathbf{G}}]. \tag{3.5}$$

It should be noted that when obtaining the the expanded matrix  $[\widetilde{\mathbf{W}}_1; \widetilde{\mathbf{W}}_2; \widetilde{\mathbf{G}}]$ , any two rows of the matrix  $[\mathbf{W}_1; \mathbf{W}_2; \mathbf{G}]$  can be used instead of the last two rows. This obtained system is solved by using MATLAB and so the coefficient matrix  $\mathbf{A}_N$  is calculated. Finally, by substituting the calculated coefficient matrix  $\mathbf{A}_N$  in Eq. (2.2), the approximate solution based on the Pell-Lucas polynomials is obtained.

#### 4. Error analysis

In this section, two important theorems are given and proven. The first theorem is about the upper boundary of errors. In the second theorem, the error estimation is made by using the residual function. In addition, the residual improved technique is presented.

**Theorem 4.1** (*Upper Boundary of Errors*) *It is assumed that  $y(t)$ ,  $y_N(t)$ , and  $y_N^M(t)$  are the exact solution, the Pell-Lucas polynomial solution with  $N - th$  degree of the problem (1.1)-(1.2) and the expansion of the generalized Maclaurin series with  $N - th$  degree of  $y(t)$ , respectively. Then the absolute error of the Pell-Lucas polynomial solution for  $0 \leq a \leq t \leq b$  is bounded by the inequality*

$$\|y(t) - y_N(t)\|_\infty \leq k_N (\|\widetilde{\mathbf{A}}_N\|_\infty + \|\mathbf{D}_N^T\|_\infty \|\mathbf{A}_N\|_\infty) + \frac{b^{N+1}}{(N+1)!} \|y^{(N+1)}(c_t)\|_\infty \tag{4.1}$$

where  $\|\mathbf{T}_N(t)\|_\infty \leq \max\{b^N, 1\} := k_N$ ,  $\widetilde{\mathbf{A}}_N$  represents the coefficient matrix of the  $y_N^{Mac}(t)$  and  $\Delta \mathbf{A}_N = \|\mathbf{A}_{N+1}\|_\infty - \|\mathbf{A}_N\|_\infty$ .



**Proof:** Firstly, we add and subtract the Maclaurin expansion  $y_N^M(t)$  with  $N - th$  degree from  $y(t) - y_N(t)$ . Next, by using the triangle inequality, we can write

$$\|y(t) - y_N(t)\|_\infty = \|y(t) - y_N^M(t) + y_N^M(t) - y_N(t)\|_\infty \leq \|y(t) - y_N^M(t)\|_\infty + \|y_N^M(t) - y_N(t)\|_\infty. \quad (4.2)$$

We know that the Pell-Lucas polynomial solution  $y_N(t)$  is written in the matrix form as  $y_N(t) = \mathbf{T}_N(t)\mathbf{D}_N\mathbf{A}_N$  from Lemma (2.2) and the expansion of the Maclaurin series with  $N - th$  degree of  $y(t)$  is written as  $y_N^M(t) = \mathbf{T}_N(t)\tilde{\mathbf{A}}_N$ . Accordingly, we have the following expression:

$$\|y_N^M(t) - y_N(t)\|_\infty = \|\mathbf{T}_N(t)(\tilde{\mathbf{A}}_N - \mathbf{D}_N\mathbf{A}_N)\|_\infty \leq \|\mathbf{T}_N(t)\|_\infty \left( \|\tilde{\mathbf{A}}_N\|_\infty + \|\mathbf{D}_N\|_\infty \|\mathbf{A}_N\|_\infty \right). \quad (4.3)$$

Because of  $0 \leq t \leq b$ , we can write  $\|\mathbf{T}_N(t)\|_\infty$  in the inequality (4.3) as

$$\|\mathbf{T}_N(t)\|_\infty \leq \max \{b^N, 1\} := k_N \quad (4.4)$$

From here, the inequality (4.3) becomes

$$\|y_N^M(t) - y_N(t)\|_\infty \leq k_N \left( \|\tilde{\mathbf{A}}_N\|_\infty + \|\mathbf{D}_N\|_\infty \|\mathbf{A}_N\|_\infty \right). \quad (4.5)$$

As for the term  $\|y(t) - y_N^M(t)\|_\infty$  in Eq. (4.2), since the remainder term of the Maclaurin series  $y_N^M(t)$  with  $N - th$  degree is

$$\frac{t^{N+1}}{(N+1)!} y^{(N+1)}(c_t), \quad 0 \leq a \leq t \leq b, \quad (4.6)$$

the term  $\|y(t) - y_N^M(t)\|_\infty$  in Eq. (4.2) can be written as

$$\|y(t) - y_N^M(t)\|_\infty \leq \frac{b^{N+1}}{(N+1)!} \|y^{(N+1)}(c_t)\|_\infty. \quad (4.7)$$

As a result, by writing the inequalities (4.5) and (4.7) in (4.2), we get

$$\|y(t) - y_N(t)\|_\infty \leq k_N (\|\tilde{\mathbf{A}}_N\|_\infty + \|\mathbf{D}_N^T\|_\infty \|\mathbf{A}_N\|_\infty) + \frac{b^{N+1}}{(N+1)!} \|y^{(N+1)}(c_t)\|_\infty, \quad (4.8)$$

which completes the proof of theorem.  $\square$

**Theorem 4.2 (Error Estimation)** Let  $y(t)$  be the exact solution and  $y_N(x)$  be the Pell-Lucas polynomial solution with  $N - th$  degree of the problem (1.1)-(1.2). In this case, we have the error problem

$$\begin{cases} \sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t) e_N^{(k)}(t - \tau_{kj}(t)) + \sum_{k=0}^2 \alpha_k(t) e_N^{(k)} + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t) e_N^{(p)}(t) = -R_N(t) \\ e_N(a) = 0 \quad \text{and} \quad e_N'(a) = 0, \end{cases} \quad (4.9)$$

where

$$\begin{aligned} \alpha_0(t) &= 2N_{00}(t)y_N(t) + N_{10}(t)y_N'(t) + N_{20}(t)y_N''(t), \\ \alpha_1(t) &= N_{10}(t)y_N(t) + 2N_{11}(t)y_N'(t) + N_{21}(t)y_N''(t), \\ \alpha_2(t) &= N_{20}(t)y_N(t) + N_{21}(t)y_N'(t) + 2N_{22}(t)y_N''(t). \end{aligned}$$

Here,  $e_N(t) = y(t) - y_N(t)$  and  $R_N(t)$  is the residual function of the problem (1.1)-(1.2).

**Proof:** Inasmuch as the Pell-Lucas polynomial solution (1.3) provides Eq. (1.1), we can write

$$R_N(t) = \sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t)y_N^{(k)}(t - \tau_{kj}(t)) + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t)y_N^{(p)}(t)y_N^{(q)}(t) - g(t) \tag{4.10}$$

or

$$R_N(t) + g(t) = \sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t)y_N^{(k)}(t - \tau_{kj}(t)) + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t)y_N^{(p)}(t)y_N^{(q)}(t). \tag{4.11}$$

Similarly, since the Pell-Lucas polynomial solution (1.3) provides the initial conditions (1.2), we can write

$$y_N(a) = \lambda \quad \text{and} \quad y'_N(a) = \mu. \tag{4.12}$$

We know that  $e_N(t) = y(t) - y_N(t)$ . If we subtract the problem (4.11)-(4.12) from the problem (1.1)-(1.2) and if we write  $y(t) \rightarrow e_N(t) + y_N(t)$ , then we get the error problem

$$\begin{cases} \sum_{k=0}^2 \sum_{j=0}^1 M_{kj}(t)e_N^{(k)}(t - \tau_{kj}(t)) + \sum_{k=0}^2 \alpha_k(t)e_N^{(k)} + \sum_{p=0}^2 \sum_{q=0}^p N_{pq}(t)e_N^{(p)}(t) = -R_N(t) \\ e_N(a) = 0 \quad \text{and} \quad e'_N(a) = 0 \end{cases} \tag{4.13}$$

Hence, the desired result is achieved.  $\square$

**Corollary 4.1** *When we solve the error problem (4.9) according to the method in Section (3), we gain the estimated error function  $e_{N,M}(t)$ .*

**Corollary 4.2** *If we add the Pell-Lucas polynomial solution  $y_N(t)$  with the estimated error function  $e_{N,M}(t)$ , then we have the improved approximate solution  $y_{N,M}(t)$ .*

**Corollary 4.3** *The error function for the improved approximate solution is calculated by*

$$E_{N,M}(t) = y(t) - y_{N,M}(t). \tag{4.14}$$

### 5. Applications and discussions

In this section, the methods presented in the previous sections (3)-(4) are tested for the three examples. The obtained results are shown in tables and graphs. The comparisons are also made with other methods in the literature. All results are calculated by using MATLAB.

In this work,  $y(t)$  represents the exact solution,  $y_N(t)$  represents the Pell-Lucas polynomial solution,  $y_{N,M}(t)$  represents the improved approximate solution,  $|e_N(t)|$  represents the actual absolute error function,  $|e_{N,M}(t)|$  represents the estimated absolute error function and  $|E_{N,M}(t)|$  represents the improved absolute error function.

**Example 5.1** Firstly, we consider the second order nonlinear differential equation with variable delays  $t^3, t^2$  and  $-\frac{t}{2}$

$$y''(t - t^3) + y(t - t^2) - y\left(t + \frac{t}{2}\right) - t\left(y'(t)\right)^2 = g(t) \tag{5.1}$$

and the conditions

$$y(0) = 1, \quad y'(0) = 1 \tag{5.2}$$

where  $g(t) = \frac{(-t^2+t)^2}{2} - t(t+1)^2 - \frac{t}{2} - \frac{17t^2}{8} + 1$ .

The exact solution of Eq. (5.1) under the conditions (5.2) is  $\frac{t^2}{2} + t + 1$ . Now, let us investigate the solution of the problem (5.1)-(5.2) for  $N = 2$  in the form

$$y_2(t) = \sum_{n=0}^2 a_n Q_n(t). \tag{5.3}$$

For  $N = 2$ , the collocation points are  $t_0 = 0$ ,  $t_1 = 1/2$  and  $t_2 = 1$ . According to the method in Section (3), from Eq. (3.4) the fundamental matrix equation becomes

$$\mathbf{W}_1 \mathbf{A}_2 + \mathbf{W}_2 \mathbf{A}_2 = \mathbf{G} \tag{5.4}$$

where

$$\mathbf{W}_1 = \mathbf{M}_{20} \mathbf{T}_{20} + \mathbf{M}_{00} \mathbf{T}_{00} + \mathbf{M}_{01} \mathbf{T}_{01}, \quad \mathbf{W}_2 = \mathbf{N}_{11} \tilde{\mathbf{T}}_{11}, \quad \mathbf{M}_{20} = \text{eye}(3, 3), \quad \mathbf{M}_{00} = \text{eye}(3, 3), \quad \mathbf{M}_{01} = -\text{eye}(3, 3),$$

$$\mathbf{T}_{00} = \begin{bmatrix} \mathbf{T}_2(t_0) \mathbf{S}_2(t_0/2) \mathbf{B}_2^2 \mathbf{D}_2, \\ \mathbf{T}_2(t_1) \mathbf{S}_2(t_1/2) \mathbf{B}_2^2 \mathbf{D}_2, \\ \mathbf{T}_2(t_2) \mathbf{S}_2(t_2/2) \mathbf{B}_2^2 \mathbf{D}_2, \end{bmatrix}, \quad \mathbf{T}_{01} = \begin{bmatrix} \mathbf{T}_2(t_0) \mathbf{S}_2(-t_0^2) \mathbf{B}_2^2 \mathbf{D}_2, \\ \mathbf{T}_2(t_1) \mathbf{S}_2(-t_1^2) \mathbf{B}_2^2 \mathbf{D}_2, \\ \mathbf{T}_2(t_2) \mathbf{S}_2(-t_2^2) \mathbf{B}_2^2 \mathbf{D}_2, \end{bmatrix}, \quad \mathbf{T}_{20} = \begin{bmatrix} \mathbf{T}_2(t_0) \mathbf{S}_2(-t_0^3) \mathbf{B}_2^2 \mathbf{D}_2, \\ \mathbf{T}_2(t_1) \mathbf{S}_2(-t_1^3) \mathbf{B}_2^2 \mathbf{D}_2, \\ \mathbf{T}_2(t_2) \mathbf{S}_2(-t_2^3) \mathbf{B}_2^2 \mathbf{D}_2, \end{bmatrix},$$

$$\tilde{\mathbf{T}}_{11} = \begin{bmatrix} \mathbf{T}_2(t_0) \mathbf{B}_2 \mathbf{D}_2 \mathbf{A}_2 \mathbf{T}_2(t_0) \mathbf{B}_2 \mathbf{D}_2 \\ \mathbf{T}_2(t_1) \mathbf{B}_2 \mathbf{D}_2 \mathbf{A}_2 \mathbf{T}_2(t_1) \mathbf{B}_2 \mathbf{D}_2 \\ \mathbf{T}_2(t_2) \mathbf{B}_2 \mathbf{D}_2 \mathbf{A}_2 \mathbf{T}_2(t_2) \mathbf{B}_2 \mathbf{D}_2 \end{bmatrix}, \quad \mathbf{N}_{11} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1/2 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 1 \\ -7/8 \\ -45/8 \end{bmatrix},$$

$$\mathbf{D}_2 = \begin{bmatrix} 2 & 0 & 2 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{T}_2(t_i) = \begin{bmatrix} 1 \\ t_i \\ t_i^2 \end{bmatrix}^T.$$

Also, the matrix representations of the conditions (5.2) are as follows:

$$[\mathbf{U}_1; \mathbf{0}; 1] = [ 2 \ 0 \ 2 \ ; \ 0 \ 0 \ 0 \ ; \ 1 ],$$

$$[\mathbf{U}_2; \mathbf{0}; 1] = [ 0 \ 2 \ 0 \ ; \ 0 \ 0 \ 0 \ ; \ 1 ].$$

Instead of the last two lines of the system  $[\mathbf{W}_1; \mathbf{W}_2; \mathbf{G}]$ , the matrix forms of conditions  $[\mathbf{U}_1; \mathbf{0}; 1]$  and  $[\mathbf{U}_2; \mathbf{0}; 1]$  are written. Here,  $\mathbf{0} = [ 0 \ 0 \ 0 ]$ . The obtained system is solved by using MATLAB and so the Pell-Lucas coefficients matrix is calculated as

$$\mathbf{A}_2 = [ 3/8 \ 1/2 \ 1/8 ]^T.$$

It should be noted that any two lines can be changed instead of the last two lines. By substituting this coefficients matrix  $\mathbf{A}_2$  in Eq. (5.3), we get the approximate solution as  $\frac{t^2}{2} + t + 1$ , which is the exact solution.

**Example 5.2** For the second example, let us take the second order nonlinear differential equation with variable delay  $t - t^3/8$

$$y''(t) + y\left(\frac{t^3}{8}\right) + 2y(t) - y^2(t) = g(t), \quad t \in [0, 1] \tag{5.5}$$

and the initial conditions

$$y(0) = 0, \quad y'(0) = 1 \tag{5.6}$$

Here,  $g(t) = \sin(t^3/8) + \sin(t) - \sin^2(t)$ .

The exact solution of this problem is  $\sin(t)$ . Now, we find the Pell-Lucas polynomial solution of the problem (5.5)-(5.6) for  $N = 4$  in the truncated serial form

$$y_4(t) = \sum_{n=0}^4 a_n Q_n(t). \tag{5.7}$$

For  $N = 4$ , the collocation points are  $t_0 = 0, t_1 = 1/4, t_2 = 1/2, t_3 = 3/4$  and  $t_4 = 1$ . By using Eq. (3.4), we determine the basic matrix equation as

$$\mathbf{W}_1 \mathbf{A}_4 + \mathbf{W}_2 \mathbf{A}_4 = \mathbf{G} \tag{5.8}$$

where

$$\mathbf{W}_1 = \mathbf{M}_{20} \mathbf{T}_{20} + \mathbf{M}_{00} \mathbf{T}_{00} + \mathbf{M}_{01} \mathbf{T}_{01}, \quad \mathbf{W}_2 = \mathbf{N}_{11} \tilde{\mathbf{T}}_{11}, \quad \mathbf{M}_{20} = \text{eye}(5, 5), \quad \mathbf{M}_{00} = \text{eye}(5, 5), \quad \mathbf{M}_{01} = 2\text{eye}(5, 5),$$

$$\mathbf{N}_{11} = -\text{eye}(5, 5), \quad \mathbf{T}_{20} = \begin{bmatrix} \mathbf{T}_4(t_0) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_1) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_2) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_3) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_4) \mathbf{B}_4^2 \mathbf{D}_4, \end{bmatrix}, \quad \mathbf{T}_{00} = \begin{bmatrix} \mathbf{T}_4(t_0) \mathbf{S}_4(-t_0 - t_0^3/8) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_1) \mathbf{S}_4(-t_1 - t_1^3/8) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_2) \mathbf{S}_4(-t_2 - t_2^3/8) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_3) \mathbf{S}_4(-t_3 - t_3^3/8) \mathbf{B}_4^2 \mathbf{D}_4, \\ \mathbf{T}_4(t_4) \mathbf{S}_4(-t_4 - t_4^3/8) \mathbf{B}_4^2 \mathbf{D}_4, \end{bmatrix}, \quad \mathbf{A}_4 = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix},$$

$$\mathbf{T}_{01} = \begin{bmatrix} \mathbf{T}_4(t_0) \mathbf{D}_4, \\ \mathbf{T}_4(t_1) \mathbf{D}_4, \\ \mathbf{T}_4(t_2) \mathbf{D}_4, \\ \mathbf{T}_4(t_3) \mathbf{D}_4, \\ \mathbf{T}_4(t_4) \mathbf{D}_4, \end{bmatrix}, \quad \tilde{\mathbf{T}}_{11} = \begin{bmatrix} T_4(t_0) \mathbf{D}_4 \mathbf{A}_4 \mathbf{T}_4(t_0) \mathbf{D}_4 \\ T_4(t_1) \mathbf{D}_4 \mathbf{A}_4 \mathbf{T}_4(t_1) \mathbf{D}_4 \\ T_4(t_2) \mathbf{D}_4 \mathbf{A}_4 \mathbf{T}_4(t_2) \mathbf{D}_4 \\ T_4(t_3) \mathbf{D}_4 \mathbf{A}_4 \mathbf{T}_4(t_3) \mathbf{D}_4 \\ T_4(t_4) \mathbf{D}_4 \mathbf{A}_4 \mathbf{T}_4(t_4) \mathbf{D}_4 \end{bmatrix}, \quad \mathbf{D}_4 = \begin{bmatrix} 2 & 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 6 & 0 \\ 0 & 0 & 4 & 0 & 16 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 16 \end{bmatrix}, \quad \mathbf{T}_4(t_i) = \begin{bmatrix} 1 \\ t_i \\ t_i^2 \\ t_i^3 \\ t_i^4 \end{bmatrix}^T$$

$$\mathbf{G} = \begin{bmatrix} 0 \\ 1324/7037 \\ 567/2138 \\ 725/2688 \\ 1071/4150 \end{bmatrix}, \quad \mathbf{B}_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Also, the matrix representations of the conditions (5.6) are as follows:

$$[\mathbf{U}_1; \mathbf{0}; 0] = [ 2 \ 0 \ 2 \ 0 \ 2 \ ; \ 0 \ 0 \ 0 \ 0 \ 0 \ ; \ 0 ],$$

$$[\mathbf{U}_2; \mathbf{0}; 1] = [ 0 \ 2 \ 0 \ 6 \ 0 \ ; \ 0 \ 0 \ 0 \ 0 \ 0 \ ; \ 1 ].$$

From here, instead of the last two lines of the system  $[\mathbf{W}_1; \mathbf{W}_2; \mathbf{G}]$ , the matrix forms of conditions  $[\mathbf{U}_1; \mathbf{0}; 0]$  and  $[\mathbf{U}_2; \mathbf{0}; 1]$  are written. Here,  $\mathbf{0} = [0 \ 0 \ 0 \ 0 \ 0]$ . By solving the obtained system with the help of a code written in MATLAB, the Pell-Lucas coefficients matrix is calculated. Finally, this coefficients matrix  $\mathbf{A}_4$  is substituted in Eq. (5.7) and so we obtain the approximate solution as

$$y_4(t) = 1.0241e - 02t^4 - 1.7004e - 01t^3 + 2.4731e - 13t^2 + t.$$

In Table 1, the exact solution, the approximate solution and the improved approximate solution of the problem (5.5)-(5.6) for various values of  $N$  and  $M$  are given. According to Table 1, it can be observed that the results of the method are quite successful.

**Table 1.** Numerical results of the exact solution, the approximate solution for  $N = 4, 7, 10$ , and the improved approximate solution for  $(N, M) = (4, 5), (7, 8), (10, 11)$  of the problem (5.5)-(5.6).

	Exact solution	Approximate solution		
$t_i$	$y(t_i) = \sin(t_i)$	$y_4(t_i)$	$y_7(t_i)$	$y_{10}(t_i)$
0	0	0	6.3527471044073e-22	1.2914357313484e-22
0.2	0.19866933079506	0.19865610054763	0.19866933151654	0.19866933079517
0.4	0.38941834230865	0.38937988888193	0.38941834378405	0.38941834230888
0.6	0.56464247339504	0.56459953516704	0.56464247554152	0.56464247339537
0.8	0.71735609089952	0.71713646306981	0.71735609360186	0.71735609089992
1	0.84147098480790	0.84020534975983	0.84147087741528	0.84147098482352
	Exact solution	Improved approximate solution		
$t_i$	$y(t_i) = \sin(t_i)$	$y_{4,5}(t_i)$	$y_{7,8}(t_i)$	$y_{10,11}(t_i)$
0	0	0	1.8674429508997e-20	1.6933938102609e-22
0.2	0.19866933079506	0.19866912034635	0.19866933070487	0.19866933079506
0.4	0.38941834230865	0.38941761237277	0.38941834213530	0.38941834230866
0.6	0.56464247339504	0.56464000940943	0.56464247319947	0.56464247339505
0.8	0.71735609089952	0.71735218324262	0.71735609082121	0.71735609089956
1	0.84147098480790	0.84149662243629	0.84147097448162	0.84147098480771

Figure 1 compares the actual absolute error function for  $N = 5$  with the results of LWM [15] in the literature. In Figure 1, the estimated error function for  $(N, M) = (5, 6)$  is also compared. When the present method is compared with LWM [15], similar results are obtained. In addition, when Figure 1 is examined, it is observed that the error estimation method is also effective.

In Table 2, the actual absolute errors, the estimated absolute errors and the improved absolute errors are given. According to Table 2, we can infer three important conclusions. The first important result is that the errors decrease as the value of  $N$  increases. The second important consequence is that the results of the estimated absolute errors are quite close to the results of the actual absolute errors. From this result, it can be said that the error estimation method described in Section 4 is effective. The final important result is that the improved absolute errors yield better results than the actual absolute errors at most points in the given range. From this result, it can be concluded that the technique of improving approximate solutions based on the residual function is effective.

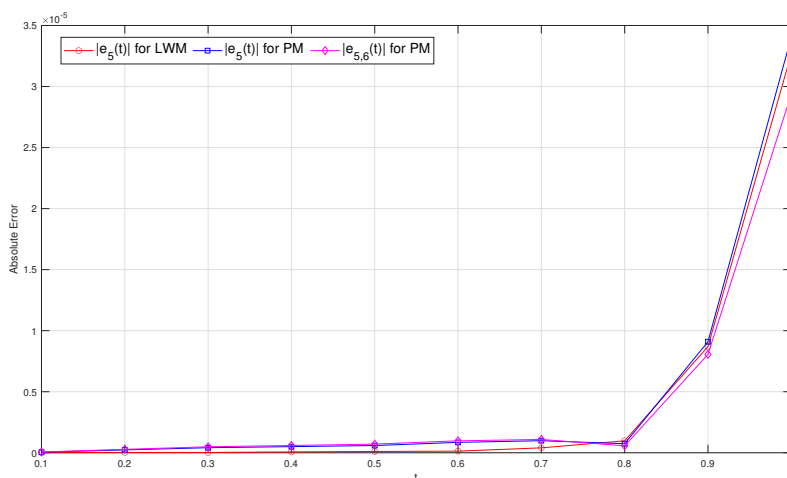


Figure 1. Comparison of the absolute error functions of the prob. (5.5)-(5.6) with LWM [15].

Table 2. Comparison of the absolute errors for  $(N, M) = (4, 5), (4, 6), (7, 8), (7, 9), (10, 11), (10, 12)$  of the problem (5.5)-(5.6).

	Absolute errors for the approximate solution	Estimated errors for the approximate solution		Absolute errors for the improved approximate solution	
$t_i$	$ e_4(t_i)  =  y(t_i) - y_4(t_i) $	$ e_{4,5}(t_i) $	$ e_{4,6}(t_i) $	$ E_{4,5}(t_i) $	$ E_{4,6}(t_i) $
0	0	0	1.2197e-19	0	1.2197e-19
0.2	1.3230e-05	1.3020e-05	1.3315e-05	2.1045e-07	8.5129e-08
0.4	3.8453e-05	3.7723e-05	3.8330e-05	7.2994e-07	1.2347e-07
0.6	4.2938e-05	4.0474e-05	4.1512e-05	2.4640e-06	1.4264e-06
0.8	2.1963e-04	2.1572e-04	2.1464e-04	3.9077e-06	4.9918e-06
1	1.2656e-03	1.2913e-03	1.2549e-03	2.5638e-05	1.0696e-05
$t_i$	$ e_7(t_i)  =  y(t_i) - y_7(t_i) $	$ e_{7,8}(t_i) $	$ e_{7,9}(t_i) $	$ E_{7,8}(t_i) $	$ E_{7,9}(t_i) $
0	6.3527e-22	1.931e-20	2.3164e-20	1.8674e-20	2.2529e-20
0.2	7.2149e-10	8.1166e-10	7.227e-10	9.0154e-11	1.1937e-12
0.4	1.4754e-9	1.6487e-9	1.4650e-9	1.7332e-10	1.0459e-11
0.6	2.1464e-9	2.3420e-9	2.0777e-9	1.9566e-10	6.8667e-11
0.8	2.7024e-9	2.7807e-9	2.4807e-9	7.8217e-11	2.2169e-10
1	1.07390e-7	9.7066e-8	1.0813e-7	1.0326e-8	7.4124e-10
$t_i$	$ e_{10}(t_i)  =  y(t_i) - y_{10}(t_i) $	$ e_{10,11}(t_i) $	$ e_{10,12}(t_i) $	$ E_{10,11}(t_i) $	$ E_{10,12}(t_i) $
0	1.2914e-22	2.9848e-22	5.9177e-22	1.6934e-22	7.2091e-22
0.2	1.4211e-13	1.0755e-13	1.0891e-13	2.8422e-14	2.8422e-14
0.4	2.8422e-13	2.2242e-13	2.2524e-13	5.6843e-14	5.6843e-14
0.6	2.2737e-13	3.1878e-13	3.2283e-13	1.1369e-13	1.1369e-13
0.8	4.5475e-13	3.6246e-13	3.6751e-13	1.1369e-13	1.1369e-13
1	1.5689e-11	1.5817e-11	1.5581e-11	1.1369e-13	1.1369e-13

**Example 5.3** The third example is the second order nonlinear differential equation with variable delays  $t^2$  and  $-t/2$

$$y''(t) + y'(t - t^2) - t^2y(t + \frac{t}{2}) - y'(t)y(t) + (y'(t))^2 = g(t), \quad 0 \leq t \leq 1 \tag{5.9}$$

and the initial conditions

$$y(0) = 1, \quad y'(0) = 1 \tag{5.10}$$

where  $g(t) = e^t + e^{t-t^2} - t^2e^{3t/2}$ .

The problem (5.9)-(5.10) has the exact solution  $e^t$ . We seek the approximate solution of the problem (5.9)-(5.10) for  $N = 4$  in the truncated Pell-Lucas serial form

$$y_4(t) = \sum_{n=0}^4 a_n Q_n(t). \tag{5.11}$$

The collocation points are  $t_0 = 0, t_1 = 1/4, t_2 = 1/2, t_3 = 3/4,$  and  $t_4 = 1$  for  $N = 4$ . When we use Eq. (3.4), we obtain the fundamental matrix equation

$$\mathbf{W}_1 \mathbf{A}_4 + \mathbf{W}_2 \mathbf{A}_4 = \mathbf{G} \tag{5.12}$$

where

$$\mathbf{W}_1 = \mathbf{M}_{20} \mathbf{T}_{20} + \mathbf{M}_{10} \mathbf{T}_{10} + \mathbf{M}_{00} \mathbf{T}_{00}, \quad \mathbf{W}_2 = \mathbf{N}_{10} \tilde{\mathbf{T}}_{10} + \mathbf{N}_{11} \tilde{\mathbf{T}}_{11}, \quad \mathbf{M}_{20} = eye(5, 5), \quad \mathbf{M}_{10} = eye(5, 5),$$

$$\mathbf{N}_{10} = -eye(5, 5), \quad \mathbf{N}_{11} = eye(5, 5), \quad \mathbf{A}_4 = [ a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 ]^T,$$

$$\mathbf{T}_{20} = \begin{bmatrix} \mathbf{T}_4(t_0) \mathbf{B}_4^2 \mathbf{D}_4 \\ \mathbf{T}_4(t_1) \mathbf{B}_4^2 \mathbf{D}_4 \\ \mathbf{T}_4(t_2) \mathbf{B}_4^2 \mathbf{D}_4 \\ \mathbf{T}_4(t_3) \mathbf{B}_4^2 \mathbf{D}_4 \\ \mathbf{T}_4(t_4) \mathbf{B}_4^2 \mathbf{D}_4 \end{bmatrix}, \quad \mathbf{T}_{10} = \begin{bmatrix} \mathbf{T}_4(t_0) \mathbf{S}_N(-t_0^2) \mathbf{B}_4 \mathbf{D}_4 \\ \mathbf{T}_4(t_1) \mathbf{S}_N(-t_1^2) \mathbf{B}_4 \mathbf{D}_4 \\ \mathbf{T}_4(t_2) \mathbf{S}_N(-t_2^2) \mathbf{B}_4 \mathbf{D}_4 \\ \mathbf{T}_4(t_3) \mathbf{S}_N(-t_3^2) \mathbf{B}_4 \mathbf{D}_4 \\ \mathbf{T}_4(t_4) \mathbf{S}_N(-t_4^2) \mathbf{B}_4 \mathbf{D}_4 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 2 \\ 703/293 \\ 3437/1430 \\ 2405/1512 \\ -242/317 \end{bmatrix}, \quad \mathbf{T}_4(t_i) = \begin{bmatrix} 1 \\ t_i \\ t_i^2 \\ t_i^3 \\ t_i^4 \end{bmatrix}^T,$$

$$\mathbf{M}_{00} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & -\frac{9}{16} & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{T}_{00} = \begin{bmatrix} \mathbf{T}_4(t_0) \mathbf{S}_N(-(-t_0/2)) \mathbf{D}_4 \\ \mathbf{T}_4(t_1) \mathbf{S}_N(-(-t_1/2)) \mathbf{D}_4 \\ \mathbf{T}_4(t_2) \mathbf{S}_N(-(-t_2/2)) \mathbf{D}_4 \\ \mathbf{T}_4(t_3) \mathbf{S}_N(-(-t_3/2)) \mathbf{D}_4 \\ \mathbf{T}_4(t_4) \mathbf{S}_N(-(-t_4/2)) \mathbf{D}_4 \end{bmatrix}, \quad \mathbf{B}_4 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{\mathbf{T}}_{10} = \begin{bmatrix} T_4(t_0) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_0) \mathbf{D}_4 \\ T_4(t_1) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_1) \mathbf{D}_4 \\ T_4(t_2) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_2) \mathbf{D}_4 \\ T_4(t_3) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_3) \mathbf{D}_4 \\ T_4(t_4) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_4) \mathbf{D}_4 \end{bmatrix}, \quad \tilde{\mathbf{T}}_{11} = \begin{bmatrix} T_4(t_0) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_0) \mathbf{B}_4 \mathbf{D}_N \\ T_4(t_1) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_1) \mathbf{B}_4 \mathbf{D}_N \\ T_4(t_2) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_2) \mathbf{B}_4 \mathbf{D}_N \\ T_4(t_3) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_3) \mathbf{B}_4 \mathbf{D}_N \\ T_4(t_4) \mathbf{B}_4 \mathbf{D}_4 \mathbf{A} T_4(t_4) \mathbf{B}_4 \mathbf{D}_N \end{bmatrix}, \quad \mathbf{D}_4 = \begin{bmatrix} 2 & 0 & 2 & 0 & 2 \\ 0 & 2 & 0 & 6 & 0 \\ 0 & 0 & 4 & 0 & 16 \\ 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 16 \end{bmatrix}.$$

The matrix representations of the conditions (5.10) become in the following matrix forms:

$$[\mathbf{U}_1; \mathbf{0}; 1] = [ 2 \quad 0 \quad 2 \quad 0 \quad 2 \quad ; \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad ; \quad 1 ],$$

$$[\mathbf{U}_2; \mathbf{0}; 1] = [ 0 \quad 2 \quad 0 \quad 6 \quad 0 \quad ; \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad ; \quad 1 ].$$

Accordingly, we write the matrix forms of conditions  $[\mathbf{U}_1; \mathbf{0}; 1]$  and  $[\mathbf{U}_2; \mathbf{0}; 1]$  instead of the last two lines of the system  $[\mathbf{W}_1; \mathbf{W}_2; \mathbf{G}]$ . Here,  $\mathbf{0} = [0 \ 0 \ 0 \ 0 \ 0]$ . Then, we solve the obtained system by using a code written in MATLAB and so we calculate the Pell-Lucas coefficients matrix. Consequently, by substituting this coefficients matrix  $\mathbf{A}_4$  in Eq. (5.11), we get the approximate solution as

$$y_4(t) = 5.3299e - 02t^4 + 1.6290e - 01t^3 + 5.0000e - 01t^2 + t + 1.$$

Table 3 compares the approximate solutions of the problem (5.9)-(5.10) for  $N = 5, N = 7$  with the exact solution. In addition, Table 3 compares the actual absolute errors for  $N = 5$  and  $N = 7$  and the estimated absolute errors for  $(N, M) = (5, 6)$  and  $(N, M) = (7, 8)$ . Figure 2 compares the actual absolute error functions for  $N = 5, N = 7$ , and  $N = 11$ . Accordingly, it can be said that a more accurate result is obtained if a larger number value of  $N$  is selected. Moreover, Table 3 compares the actual absolute errors for  $N = 5$  and  $N = 7$  with the results of LM [12] in the literature. When the present method is compared with LM [12], more successful results are achieved at more points by using the present method.

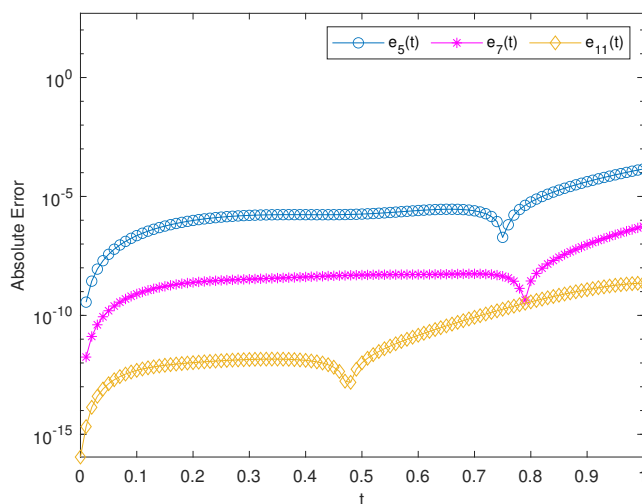
**Table 3.** Comparison of the solutions and the absolute errors of the problem (5.9)-(5.10) with LM [12].

$x_i$	Exact solution	Approximate solution for PM	Actual absolute error for PM	Estimated absolute error for PM	Actual absolute error for LM[12]
	$y(t)$	$y_5(t)$	$ e_5(t) $	$ e_{5,6}(t) $	$ e_5(t) $
0.2	1.2214027581602	1.2214037379204	9.7976e-7	1.1918e-6	2.05e-06
0.4	1.4918246976413	1.4918264106894	1.7130e-6	2.3238e-6	6.22e-06
0.6	1.8221188003905	1.8221213633236	2.5629e-6	3.5678e-6	1.37e-05
0.8	2.2255409284925	2.2255350651018	5.8634e-6	4.6792e-6	1.85e-05
1	2.718281828459	2.7181382540537	1.4357e-04	1.3939e-04	9.09e-05
	$y(t)$	$y_7(t)$	$ e_7(t) $	$ e_{7,8}(t) $	$ e_7(t) $
0.2	1.2214027581602	1.2214027605892	2.4290e-9	2.8969e-9	2.79e-09
0.4	1.4918246976413	1.4918247017708	4.1296e-9	5.4339e-9	5.53e-09
0.6	1.8221188003905	1.8221188056232	5.2326e-9	7.4407e-9	8.75e-09
0.8	2.2255409284925	2.2255409256982	2.7944e-9	4.4515e-10	4.84e-09
1	2.718281828459	2.7182812712682	5.5719e-7	5.5707e-7	5.35e-07

### 6. Conclusions

This article presents a collocation method for the nonlinear differential equations with the variable delays. In the method, the approximate solution is investigated depending on the  $N - th$  order truncated Pell-Lucas polynomials. In addition to the Pell-Lucas collocation method, error analysis is performed. The method is tested for three examples. After applying the method to the first example, the exact solution is obtained. Because the exact solution of the first example is of polynomial type, this result is an important advantage of the method. After applying the method to the second example, three important results are obtained. The first important result is that the errors decrease as the value of  $N$  increases, and this result is observed from Table 2. The second important result is that the results of the estimated absolute errors are quite close to the results of the actual absolute errors. From here, it can be said that the error estimation method is effective and this





**Figure 2.** Comparison of the actual absolute error functions of the problem (5.9)-(5.10).

result can be observed from Table 2. The final important result is that the results of the improved approximate solution give better results than the Pell-Lucas polynomial solution at most points. From this, it is concluded that the results of the improved absolute errors give more accurate results than the results of the actual absolute errors, and this result can be observed from Table 2. As a result of the last example, the comments made for the second example can be said. These results are observed from Table 3 and Figure 2. In the last example, the present method is compared with the result obtained from another method in the literature for  $N = 5$  and  $N = 7$ . Accordingly, it can be said that the present method is more successful, albeit with a small difference. All calculations and graphics in this study have been obtained in a short time by using MATLAB. According to all these results, it can be said that the method is quite successful, effective, and reliable. Moreover, the method can also be applied for the fractional-order derivative differential equations after the necessary matrix relations are made and MATLAB codes are created.

## References

- [1] Akyüz-Daşcıoğlu A, Yaslan HÇ. An approximation method for solution of nonlinear integral equations. *Applied Mathematics and Computation* 2006; 174: 619-629. <https://doi.org/10.1016/j.amc.2005.04.108>
- [2] Benhammouda B, Vazquez-Leal H, Hernandez-Martinez L. Procedure for exact solutions of nonlinear pantograph delay differential equations. *British Journal of Mathematics & Computer Science* 2014; 4: 2738-2751.
- [3] Dehghan M, Abbaszadeh M. A Legendre spectral element method (SEM) based on the modified bases for solving neutral delay distributed-order fractional damped diffusion-wave equation. *Mathematical Methods in the Applied Sciences* 2018; 41: 3476-3494. <https://doi.org/10.1002/mma.4839>
- [4] Dehghan M, Abbaszadeh M, Mohebbi A. The numerical solution of nonlinear high dimensional generalized Benjamin-Bona-Mahony-Burgers equation via the meshless method of radial basis functions. *Computers & Mathematics with Applications* 2014; 68: 212-237. <https://doi.org/10.1016/j.camwa.2014.05.019>
- [5] Dehghan M, Manafian J, Saadatmandi A. Solving nonlinear fractional partial differential equations using the homotopy analysis method. *Numerical Methods for Partial Differential Equations: An International Journal* 2010; 26: 448-479. <https://doi.org/10.1002/num.20460>

- [6] Dehghan M, Saadatmandi A. The numerical solution of a nonlinear system of second-order boundary value problems using the sinc-collocation method. *Mathematical and Computer Modelling* 2007; 46: 1434-1441. <https://doi.org/10.1016/j.mcm.2007.02.002>
- [7] Dönmez Demir D, Lukonde AP, Kürkçü ÖK, Sezer M. Pell-Lucas series approach for a class of Fredholm-type delay integro-differential equations with variable delays. *Mathematical Sciences* 2021; 15: 55-64. <https://doi.org/10.1007/s40096-020-00370-5>
- [8] Eftekhari A, Saadatmandi A. DE sinc-collocation method for solving a class of second-order nonlinear BVPs. *Mathematics Interdisciplinary Research* 2021; 6: 11-22. <https://doi.org/10.22052/MIR.2020.220050.1195>
- [9] Eslahchi MR, Dehghan M, Ahmadi Asl S. The general Jacobi matrix method for solving some nonlinear ordinary differential equations. *Applied Mathematical Modelling* 2012; 36: 3387-3398. <https://doi.org/10.1016/j.apm.2011.09.082>
- [10] Ghorbani A. Approximate solution of delay differential equations via variational iteration method. *Nonlinear Science Letters A: Mathematics, Physics and Mechanics* 2017; 8: 236-239.
- [11] Gümgüm S, Baykuş Savaşaneril N, Kürkçü ÖK, Sezer M. Lucas polynomial solution for neutral differential equations with proportional delays. *TWMS Journal of Applied and Engineering Mathematics* 2020; 10: 259-269.
- [12] Gümgüm S, Baykuş Savaşaneril N, Kürkçü ÖK, Sezer M. Lucas polynomial solution of nonlinear differential equations with variable delays. *Hacettepe Journal of Mathematics and Statistics* 2020; 49: 553-564. <https://doi.org/10.15672/hujms.460975>
- [13] Gümgüm S, Özdek DE, Özaltun G, Bildik N. Legendre wavelet solution of neutral differential equations with proportional delays. *Journal Of Applied Mathematics and Computing* 2019; 61: 389-404. <https://doi.org/10.1007/s12190-019-01256-z>
- [14] Gümgüm S. Chebyshev wavelet solution of nonlinear ordinary differential equations with delays. *Journal of Abstract and Computational Mathematics* 2019; 4: 1-14.
- [15] Gümgüm S, Özdek DE, Özaltun G. Legendre wavelet solution of high order nonlinear ordinary delay differential equations. *Turkish Journal of Mathematics* 2019; 43: 1339-1352. <https://doi.org/10.3906/mat-1901-109>
- [16] Horadam AF, Mahon Bro JM. Pell and Pell-Lucas polynomials. *The Fibonacci Quarterly* 1985; 23: 7-20.
- [17] Horadam AF, Swita B, Filipponi P. Integration and derivative sequences for Pell and Pell-Lucas polynomials. *The Fibonacci Quarterly* 1994; 32: 130-35.
- [18] Imani A, Aminataei A, Imani A. Collocation method via Jacobi polynomials for solving nonlinear ordinary differential equations. *International Journal of Mathematics and Mathematical Sciences* 2011; 2011: Article ID 673085. <https://doi.org/10.1155/2011/673085>
- [19] Jafari H, Mahmoudi M, Noori Skandari MH. A new numerical method to solve pantograph delay differential equations with convergence analysis. *Advances in Difference Equations* 2021; 2021. <https://doi.org/10.1186/s13662-021-03293-0>
- [20] Karako F, Bereketoglu H. Solution of delay differential equations by the differential transform. *International Journal of Computational Mathematics* 2009; 86: 914-923. <https://doi.org/10.1080/00207160701750575>
- [21] Kharrat BN, Toma G. Differential transform method for solving initial value problems represented by strongly nonlinear ordinary differential equations. *Middle-East Journal of Scientific Research* 2019; 27: 576-579. <https://doi.org/10.5829/idosi.mejsr.2019.576.579>
- [22] Li M, Wang J. Finite time stability of fractional delay differential equations. *Applied Mathematics Letters* 2017; 64: 170-176. <https://doi.org/10.1016/j.aml.2016.09.004>
- [23] Liu B, Zhang C. A spectral galerkin method for nonlinear delay convection-diffusion-reaction equations. *Computers & Mathematics with Applications* 2015; 69: 709-724. <https://doi.org/10.1016/j.camwa.2015.02.027>

- [24] Maleknejad K, Mahmoudi Y. Taylor polynomial solutions of high-order nonlinear Volterra-Fredholm integro-differential equation. *Applied Mathematics and Computation* 2003; 145: 641-653. [https://doi.org/10.1016/S0096-3003\(03\)00152-8](https://doi.org/10.1016/S0096-3003(03)00152-8)
- [25] Martin J, Garcia O. Variable multistep methods for delay differential equations. *Mathematical and Computer Modelling* 2002; 35: 241-257. [https://doi.org/10.1016/S0895-7177\(01\)00162-5](https://doi.org/10.1016/S0895-7177(01)00162-5)
- [26] Martin J, Garcia O. Variable multistep methods for higher-order delay differential equations. *Mathematical and Computer Modelling* 2002; 36: 805-820. [https://doi.org/10.1016/S0895-7177\(02\)00228-5](https://doi.org/10.1016/S0895-7177(02)00228-5)
- [27] Mohammed OH, Wadi Q. A modified method for solving delay differential equations of fractional order. *IOSR journal of Mathematics (IOSR-JM)* 2016; 12: 1-7.
- [28] Rabiebi K, Ordokhani Y. Solving fractional pantograph delay differential equations via fractional-order Boubaker polynomials. *Engineering with Computers* 2019; 35: 1431-1441. <https://doi.org/10.1007/s00366-018-0673-8>
- [29] Rahimkhani P, Ordokhani Y, Babolian E. A new operational matrix based on Bernoulli wavelets for solving fractional delay differential equations. *Numerical Algorithms* 2017; 74: 223-245. <https://doi.org/10.1007/s11075-016-0146-3>
- [30] Rawashdeh MS, Maitama S. Solving nonlinear ordinary differential equations using the NDM. *Journal of Applied Analysis and Computation* 2015; 5: 77-88. <https://doi.org/10.11948/2015007>
- [31] Razzaghi M, Yousefi S. Legendre wavelets method for the nonlinear Volterra-Fredholm integral equations. *Mathematics and Computers in Simulation* 2005; 70: 1-8. <https://doi.org/10.1016/j.matcom.2005.02.035>
- [32] Sadollah A, Eskandar H, GuenYoo D, Hoon Kim J. Approximate solving of nonlinear ordinary differential equations using least square weight function and metaheuristic algorithms. *Engineering Applications of Artificial Intelligence* 2015; 40: 117-132. <https://doi.org/10.1016/j.engappai.2015.01.014>
- [33] Sedaghat S, Ordokhani Y, Dehghan M. Numerical solution of the delay differential equations of pantograph type via Chebyshev polynomials. *Communications in Nonlinear Science and Numerical Simulation* 2012; 17: 4815-4830. <https://doi.org/10.1016/j.cnsns.2012.05.009>
- [34] Shakeri F, Dehghan M. Solution of delay differential equations via a homotopy perturbation method. *Mathematical and Computer Modelling* 2008; 48: 486-498. <https://doi.org/10.1016/j.mcm.2007.09.016>
- [35] Shiralashetti SC, Hoogar BS, Kumbinarasaiah S. Hermite wavelet based method for the numerical solution of linear and nonlinear delay differential equations. *International Journal of Engineering, Science and Mathematics* 2017; 6: 71-79.
- [36] Şahin M, Sezer M. Pell-Lucas collocation method for solving high-order functional differential equations with hybrid delays. *Celal Bayar University Journal of Science* 2018; 14: 141-149. <https://doi.org/10.18466/cbayarfbe.307282>
- [37] Wazwaz AM. The combined Laplace transform-Adomian decomposition method for handling nonlinear Volterra integro-differential equations. *Applied Mathematics and Computation* 2010; 216: 1304-1309. <https://doi.org/10.1016/j.amc.2010.02.023>
- [38] Yang Z, Liao S. A HAM-based wavelet approach for nonlinear ordinary differential equations. *Communications in Nonlinear Science and Numerical Simulation* 2017; 48: 439-453. <https://doi.org/10.1016/j.cnsns.2017.01.005>
- [39] Yüzbaşı Ş. A collocation method based on Bernstein polynomials to solve nonlinear Fredholm-Volterra integro-differential equations. *Applied Mathematics and Computation* 2016; 273: 142-154. <https://doi.org/10.1016/j.amc.2015.09.091>
- [40] Yüzbaşı Ş. A numerical scheme for solutions of a class of nonlinear differential equations. *Journal of Taibah University for Science* 2017; 11: 1165-1181. <https://doi.org/10.1016/j.jtusci.2017.03.001>
- [41] Yüzbaşı Ş, Baykuş Savaşaneril N. Hermite polynomial approach for solving singular perturbed delay differential equations. *Journal of Science and Arts* 2020; 4: 845-854. <http://doi.org/10.46939/J.Sci.Arts-20.4-a06>

- [42] Yüzbaşı Ş, Gök E, Sezer M. Müntz-Legendre polynomial solutions of linear delay Fredholm integro-differential equations and residual correction. *Mathematical and Computational Applications* 2013; 18: 476-485. <https://doi.org/10.3390/mca18030476>
- [43] Yüzbaşı Ş, Karaçayır M. A Galerkin-like approach to solve multi-pantograph type delay differential equations. *Filomat* 2018; 32: 409-422. <https://doi.org/10.2298/fil1802409y>
- [44] Yüzbaşı Ş, Karaçayır M. An exponential approximation for solutions of generalized pantograph-delay differential equations. *Applied Mathematical Modelling* 2013; 37: 9160-9173. <https://doi.org/10.1016/j.apm.2013.04.028>
- [45] Yüzbaşı Ş, Karaçayır M. A numerical method for solutions of Lotka-Volterra predator-prey model with time-delay. *International Journal of Biomathematics* 2018; 11: 2018 <https://doi.org/10.1142/S1793524518500286>
- [46] Yüzbaşı Ş, Karaçayır M. Application of homotopy perturbation method to solve two models of delay differential equation systems. *International Journal of Biomathematics* 2017; 10. <https://doi.org/10.1142/S1793524517500802>
- [47] Yüzbaşı Ş. Shifted Legendre method with residual error estimation for delay linear Fredholm integro-differential equations. *Journal of Taibah University for Science* 2017; 11: 344-352. <https://doi.org/10.1016/j.jtusci.2016.04.001>
- [48] Yüzbaşı Ş, Yıldırım G. Pell-Lucas collocation method for numerical solutions of two population models and residual correction. *Journal of Taibah University for Science* 2020; 14: 1262-1278. <https://doi.org/10.1080/16583655.2020.1816027>
- [49] Yüzbaşı Ş, Yıldırım G. Pell-Lucas collocation method to solve high-order linear Fredholm-Volterra integro-differential equations and residual correction. *Turkish Journal of Mathematics* 2020; 44: 1065-1091. <https://doi.org/10.3906/mat-2002-55>
- [50] Zhang GL. High order Runge-Kutta methods for impulsive delay differential equations. *Applied Mathematics and Computation* 2017; 313: 12-23. <https://doi.org/10.1016/j.amc.2017.05.054>