



A Galerkin-Like Approach to Solve Multi-Pantograph Type Delay Differential Equations

Şuayip Yüzbaşı^a, Murat Karaçayır^a

^aDepartment of Mathematics, Faculty of Science, Akdeniz University, TR 07058 Antalya, Turkey

Abstract. In this paper, a Galerkin-like approach is presented to numerically solve multi-pantograph type delay differential equations. The method includes taking inner product of a set of monomials with a vector obtained from the equation under consideration. The resulting linear system is then solved, yielding a polynomial as the approximate solution. We also provide an error analysis and discuss the technique of residual correction, which aims to increase the accuracy of the approximate solution. Lastly, the method, error analysis and the residual correction technique are illustrated with several examples. The results are also compared with numerous existing methods from the literature.

1. Introduction

Basically, delay differential equations (DDEs) arise whenever a phenomenon has a delayed effect in a differential equation. Since this happens quite frequently, DDEs are ubiquitous in nature. Their most common applications include models on population dynamics [19, 20], control theory [31], dynamics of neural networks [5], models on the spread of diseases [6] and applications on feedback problems in electrical and electronics engineering [18, 23]. Among other applications are modelling problems in material science [15, 24], applications in quantum dot lasers [25], and topics in climate science [1]. Interested reader can refer to [12] for an extensive survey on the uses of DDEs in mathematical biology and a monograph on the numerical solutions of DDEs can be found in [32].

The first order linear DDE with constant coefficients and proportional delay

$$u'(x) = au(x) + bu(qx), \quad x \in I = [0, T], \quad 0 < q < 1$$

is called the pantograph equation and first arose in the mathematical modelling of the wave motion in the current line between an electric locomotive and an overhead catenary wire [22, 33]. For any $u_0 \in \mathbb{R}$, it has a unique (continuously differentiable) solution $u(x)$ over I satisfying $u(0) = u_0$. More generally, if the coefficients a and b are not constants but functions from $C^m(I)$, then there is a unique solution $u(x)$ from $C^{m+1}(I)$ satisfying the same initial condition [3]. Additional remarks on the analytical and approximate solutions of DDEs can be found in [8, 10].

2010 Mathematics Subject Classification. 34K06, 65L60

Keywords. Delay differential equations, pantograph equation, Galerkin method, residual error correction.

Received: 05 September 2016; Revised: 10 February 2017; Accepted: 08 August 2017

Communicated by Marko Petković

Research supported by Scientific Research Project Administration of Akdeniz University.

Email addresses: syuzbasi@akdeniz.edu.tr (Şuayip Yüzbaşı), mkaracayir@akdeniz.edu.tr (Murat Karaçayır)

The pantograph equation is an important kind of DDE which plays a role in a wide range of applications. Over the last two decades, there has been an increasing interest in the numerical solutions of DDEs of pantograph type and researchers have made use of numerous methods. To name a few, Dehghan et al. used Adomian decomposition method for this purpose and they also established the convergence of their approach in [7]. Ishiwata and Muroya proposed two kinds of collocation methods employing special mesh distributions [16]. Other collocation methods were presented by the authors of [17] using Bernstein series in conjunction with polynomial interpolation and by the authors of [38] making use of Bessel series. In [28], shifted Chebyshev polynomials were utilized to solve pantograph type DDEs and an applicable error analysis was provided. For a demonstration of how a set of exponential functions are used as a basis rather than polynomials, the reader is referred to [36]. Popular methods such as the variational iteration method [14] have been used to solve a generalized pantograph equation [27] and the homotopy perturbation method [13] was used by Feng [11], where sufficient conditions for convergence were also given.

In this paper, we consider the first-order multi-pantograph equation with (possibly) a nonhomogeneous term, given by

$$u'(x) = f(x) + \alpha(x)u(x) + \sum_{i=1}^r \beta_i(x)u(q_i x), \quad a \leq x \leq b, \quad 0 < q_i < 1 \text{ for } i = 1, 2, \dots, r \tag{1}$$

with the initial condition $u(a) = u_a$, where all the given functions are smooth in their domains. By using a numerical scheme reminiscent of the discrete Galerkin method, our aim is to obtain approximate solutions of Eq. (1) that outperform those obtained by other methods which were also interested in solving it.

The organization of the paper is as follows: The solution method is explained in Section 2. Section 3 describes the technique of residual correction as a means of deriving better approximations from a known one. Section 4 contains numerical examples, where comparisons with other methods are also made. Finally, a summary of the paper is given in Section 5.

2. Solution Method

In this section, we outline the Galerkin-like procedure we will use in order to solve Eq. (1). The method, which relies on taking inner product of an expression with monomials, was employed in [34] to obtain approximate solutions of high-order Fredholm integro-differential equations. In addition, a similar scheme was applied to high-order integro-differential equations with singular kernel functions in [35].

To begin with, we assume that the unique solution $u(x)$ of Eq. (1) can be expressed as a power series of the form

$$u(x) = \sum_{k=0}^{\infty} a_k x^k.$$

We then truncate this power series after the $(N + 1)$ st term so that

$$u_N(x) = \sum_{k=0}^N a_k x^k = \mathbf{X}(x) \cdot \mathbf{A}$$

where

$$\mathbf{X}(x) = [1 \quad x \quad x^2 \quad \dots \quad x^N], \quad \mathbf{A} = [a_0 \quad a_1 \quad a_2 \quad \dots \quad a_N]^T.$$

Here, the coefficients a_i will be obtained as the output of the procedure. The derivative $u'_N(x)$ of the approximate solution can also be expressed as a product of matrices. Namely, if \mathbf{B} is the $(N + 1) \times (N + 1)$

matrix defined by $\mathbf{B}_{i,i+1} = i$ for $i = 1, 2, \dots, N$ and $\mathbf{B}_{i,j} = 0$ elsewhere, explicitly given by

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

then the equality $u'_N(x) = \mathbf{X}(x)\mathbf{B}\mathbf{A}$ holds. The delayed terms in Eq. (1) can be expressed in a similar manner. If we denote by $\mathbf{\Lambda}_N(x)$ the $(N + 1) \times (N + 1)$ diagonal matrix with $\mathbf{\Lambda}_N(x)_{i,i} = x^{i-1}$, or explicitly by

$$\mathbf{\Lambda}_N(q) = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & q & 0 & \dots & 0 \\ 0 & 0 & q^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & q^N \end{bmatrix}$$

then the corresponding matricial expression is $u_N(qx) = \mathbf{X}(x)\mathbf{\Lambda}_N(q)\mathbf{A}$. Now, substituting the matricial expressions for $u_N(x)$, $u'_N(x)$ and $u_N(qx)$, which have already been established, into Eq. (1) and rearranging yields

$$\left\{ \mathbf{X}(x)\mathbf{B} - \alpha(x)\mathbf{X}(x) - \sum_{i=1}^r \beta_i(x)\mathbf{X}(x)\mathbf{\Lambda}_N(q_i) \right\} \mathbf{A} = f(x). \tag{2}$$

Now, since we are basically trying to calculate the approximate solution $u_N(x)$ in terms of the elements of the basis set $\mathbf{\Phi} = \{1, x, x^2, \dots, x^N\}$ of all polynomials having degree at most N , the coefficients a_i can be determined by taking inner product of the elements of $\mathbf{\Phi}$ with the vector of length $N + 1$ on the left-hand side of Eq. (2). More precisely, if we define $\mathbf{G}(x)$ by

$$\mathbf{G}(x) := \mathbf{X}(x)\mathbf{B} - \alpha(x)\mathbf{X}(x) - \sum_{i=1}^r \beta_i(x)\mathbf{X}(x)\mathbf{\Lambda}_N(q_i),$$

then taking the inner product of each element of $\mathbf{\Phi}$ with $\mathbf{G}(x)$ results in $N + 1$ linear equations in the unknowns a_0, a_1, \dots, a_N . Explicitly, the coefficient matrix \mathbf{W} and the column vector \mathbf{F} on the right-hand side of this system are given by

$$\mathbf{W}_{i,j} = \langle x^{i-1}, \mathbf{G}(x)_{1,j} \rangle, \mathbf{F}_{i,1} = \langle x^{i-1}, f(x) \rangle,$$

where the inner product $\langle \cdot, \cdot \rangle$ is defined by

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx.$$

Here, f and g are functions from the Hilbert space $L^2[a, b]$. Since the initial condition $u(a) = u_a$ should also be satisfied, one of the rows of \mathbf{W} and the corresponding entry of \mathbf{F} should be modified accordingly. For the sake of being deterministic, let us fix the modified entries to be the first row of \mathbf{W} and the first entry of \mathbf{F} . If

$$\mathbf{X}(a) = [1 \quad a \quad a^2 \quad \dots \quad a^N]$$

denotes the vector resulting from replacing x by a in $\mathbf{X}(x)$, then the initial condition corresponds to the matricial equation given by

$$\mathbf{X}(a)\mathbf{A} = [1 \quad a \quad a^2 \quad \dots \quad a^N] \cdot [a_0 \quad a_1 \quad a_2 \quad \dots \quad a_N]^T = u_a.$$

Therefore, in order to feed the initial condition to our system, we replace the first row of \mathbf{W} by $\mathbf{X}(a)$ and the first entry of \mathbf{F} by u_a . After performing this step, we are left with the modified system $\tilde{\mathbf{W}}\mathbf{A} = \tilde{\mathbf{F}}$ explicitly given by

$$\tilde{\mathbf{W}} = \begin{bmatrix} 1 & a & \dots & a^N \\ \langle x, \mathbf{G}(x)_{1,1} \rangle & \langle x, \mathbf{G}(x)_{1,2} \rangle & \dots & \langle x, \mathbf{G}(x)_{1,N+1} \rangle \\ \langle x^2, \mathbf{G}(x)_{1,1} \rangle & \langle x^2, \mathbf{G}(x)_{1,2} \rangle & \dots & \langle x^2, \mathbf{G}(x)_{1,N+1} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle x^N, \mathbf{G}(x)_{1,1} \rangle & \langle x^N, \mathbf{G}(x)_{1,2} \rangle & \dots & \langle x^N, \mathbf{G}(x)_{1,N+1} \rangle \end{bmatrix},$$

$$\tilde{\mathbf{F}} = \begin{bmatrix} u_a & \langle x, f(x) \rangle & \langle x^2, f(x) \rangle & \dots & \langle x^N, f(x) \rangle \end{bmatrix}^T,$$

from which we obtain the matrix $\mathbf{A} = \tilde{\mathbf{W}}^{-1}\tilde{\mathbf{F}}$ and the approximate solution

$$u_N(x) = a_0 + a_1x + \dots + a_Nx^N.$$

3. Error Analysis and Residual Correction

In this section, the error estimation of our method is carried out based on the residual function of the multi-pantograph equation given by Eq. (1). Then, we show how this estimation is used to obtain a better approximate solution, called the *corrected* solution for the equation. Lastly, we use Taylor’s Theorem to obtain an upper bound for the error of this corrected solution.

3.1. Error Estimation and Improving Solutions Using Residual Function

Let us consider the residual function

$$R(x) = u'(x) - \alpha(x)u(x) - \sum_{i=1}^r \beta_i(x)u(q_i x) - f(x) = 0 \tag{3}$$

of Eq. (1). We now replace $u(x)$ by the approximate solution $u_N(x)$ and obtain

$$R_N(x) = u'_N(x) - \alpha(x)u_N(x) - \sum_{i=1}^r \beta_i(x)u_N(q_i x) - f(x). \tag{4}$$

Subtracting Eq. (3) from Eq. (4) and rearranging yields

$$e'_N(x) = -R_N(x) + \alpha(x)e_N(x) + \sum_{i=1}^r \beta_i(x)e_N(q_i x), \tag{5}$$

which is just Eq. (1) with nonhomogeneous term $-R_N(x)$ and $u(x)$ replaced by $e_N(x) = u(x) - u_N(x)$. Since the approximate solution satisfies the initial condition, we have

$$e_N(a) = u(a) - u_N(a) = u_a - u_a = 0$$

as the initial condition of Eq. (5). The next step is to employ the method of Section 2 to obtain an approximate solution $e_{N,M}(x)$ to Eq. (5) for some choice of M . Lastly, we use this approximation to obtain a corrected solution

$$u_{N,M}(x) = u_N(x) + e_{N,M}(x)$$

of Eq. (1). In what follows, $E_{N,M}(x)$ will denote the actual error of $u_{N,M}(x)$ given by $E_{N,M}(x) = u(x) - u_{N,M}(x)$.

3.2. An Upper Bound on the Error

For any natural number M , the well-known Taylor truncation error can be utilized in a straightforward manner to give an upper bound for the absolute error $|E_{N,M}(x)|$ of the corrected solution, which depends on the first $M + 1$ derivatives of the exact solution $u(x)$ and the first M derivatives of $E_{N,M}(x)$. In the following result, which appears also in [37], $u_{T,M}(x)$ denotes the M -th degree truncated Taylor polynomial of $u(x)$ around a point $c \in [a, b]$ and $R_{T,M}(x)$ denotes its truncation error given by $\frac{(x - c)^{M+1}}{(M + 1)!}u^{(M+1)}(\xi)$ for some $\xi \in (c, x)$.

Theorem 3.1. *If the exact solution $u(x)$ of Eq. (1) is of class $C^{M+1}[a, b]$, then we have the following upper bound for the absolute error of the corrected solution $u_{N,M}(x)$:*

$$|E_{N,M}(x)| = |u(x) - u_{N,M}(x)| \leq |R_{T,M}(x)| + |u_{T,M}(x) - u_{N,M}(x)|. \tag{6}$$

Proof. Taylor’s Theorem with Lagrange-type remainder term states that

$$\begin{aligned} u(x) &= u(c) + u'(c)(x - c) + \dots + \frac{u^{(M)}(c)}{M!}(x - c)^M + \frac{(x - c)^{M+1}}{(M + 1)!}u^{(M+1)}(\xi) \\ &= u_{T,M}(x) + R_{T,M}(x) \end{aligned}$$

for some $\xi \in (a, b)$. Thus we have $u(x) - u_{T,M}(x) = R_{T,M}(x)$. Using this equality and the triangle inequality for $|E_{N,M}(x)|$ we get

$$\begin{aligned} |E_{N,M}(x)| = |u(x) - u_{N,M}(x)| &= |u(x) - u_{T,M}(x) + u_{T,M}(x) - u_{N,M}(x)| \\ &\leq |R_{T,M}(x)| + |u_{T,M}(x) - u_{N,M}(x)|, \end{aligned}$$

which is the desired result. \square

The above result can be interpreted as follows: Since $u_{N,M}$ is a known polynomial of degree M for a fixed M , we can write it as

$$u_{N,M}(x) = a_{N,0} + a_{N,1}x + \dots + a_{N,M}x^M,$$

where $a_{N,i}$ are constants; or alternatively as

$$u_{N,M}(x) = \tilde{a}_{N,0} + \tilde{a}_{N,1}(x - c) + \dots + \tilde{a}_{N,M}(x - c)^M.$$

After this adjustment, Eq. (6) can be expressed as

$$|u(x) - u_{N,M}(x)| \leq \left| \frac{(x - c)^{M+1}}{(M + 1)!}u^{(M+1)}(\xi) \right| + \left| \sum_{k=0}^M \left(\frac{u^{(k)}(c)}{k!} - \tilde{a}_{N,k} \right) (x - c)^k \right|. \tag{7}$$

Then, in order to express this bound in a more compact form, we introduce a notation to denote the maximum absolute values of the derivatives of $u(x)$. Namely, we let $S_k = \sup_{a \leq x \leq b} |u^{(k)}(x)|$. Since $c \in [a, b]$,

assuming $x \in [a, b]$ we also have $(x - c)^k \leq (b - a)^k$. Combining these, Eq. (7) takes the following form:

Corollary 3.2. *The absolute error $|E_{N,M}(x)|$ of $u_{N,M}(x)$ satisfies*

$$|u(x) - u_{N,M}(x)| \leq \frac{S_{M+1}}{(M + 1)!}(b - a)^{M+1} + \sum_{k=0}^M \left(\frac{S_k}{k!} + |\tilde{a}_{N,k}| \right) (b - a)^k.$$

Thus, the error of our method is closely related to the magnitude of the first M derivatives of the exact solution $u(x)$. Note, however, having the knowledge of S_i for $i = 0, 1, \dots, M + 1$, this form of the upper bound is not really appropriate for computation since it will yield unnecessarily big values. This comes from the fact that the coefficient of $(x - c)^k$ in $u_{T,M}(x)$, which is equal to $\frac{u^{(k)}(c)}{k!}$, will almost always have the same sign with $\tilde{a}_{N,k}$ in practice, causing $\frac{u^{(k)}(c)}{k!} - \tilde{a}_{N,k}$ to be much smaller than $\frac{S_k}{k!} + |\tilde{a}_{N,k}|$. Therefore, Eq. (7) is the form of the upper bound to be used in computations; the aim of Corollary 3.2 is just to emphasize the dependence of the error on the first $M + 1$ derivatives of the exact solution $u(x)$.

Table 1: Absolute errors at some points for Example 1.

x	Muroya [21]	Taylor method [29]		Present method		
		$N = 10$	$N = 12$	$N = 5$	$N = 8$	$N = 10$
2^{-1}	2.19E – 5	2.00E – 10	1.24E – 10	6.40E – 7	3.77E – 11	1.96E – 11
2^{-2}	1.08E – 6	1.00E – 10	9.74E – 11	5.34E – 7	6.95E – 11	3.03E – 11
2^{-3}	3.81E – 8	1.00E – 10	7.00E – 11	9.09E – 8	4.68E – 11	2.08E – 11
2^{-4}	1.26E – 9	1.00E – 10	9.14E – 11	3.67E – 8	9.04E – 11	3.83E – 11
2^{-5}	4.09E – 11	1.00E – 11	5.28E – 11	3.71E – 7	9.28E – 11	4.11E – 11
2^{-6}	1.20E – 12	0	1.95E – 11	2.46E – 7	6.51E – 11	2.96E – 11
CPU time(s)				0.0625	0.3750	0.4219

4. Illustrative Examples

In this section, we apply the method of Section 2 to several pantograph equations and compare the resulting approximate solutions with some other methods present in the literature. To each equation, we apply residual correction to obtain better approximate solutions using the existing ones. Having the exact solution at hand, we also use Theorem 3.1 to compute upper bounds for the error of the corrected solutions. Although not usable as an a priori error estimate, in this way we aim to make an evaluation of the precision of the bound in question.

Example 1. Let us consider the following pantograph equation with a nonhomogeneous term first studied in [21] and later solved in [28, 29]:

$$u'(x) = -u(x) + (q/2)u(qx) - (q/2)e^{-qx}, \quad u(0) = 1, \quad 0 \leq x \leq 1.$$

The exact solution of this problem is $u(x) = e^{-x}$. For $q = 0.2$, Table 1 and Figure 1 show the absolute error values for different methods. It is seen that the absolute errors of the present method with $N = 5$ are greater than the errors obtained with Taylor method. When we increase N to 8, the errors become close to the errors of Taylor method with $N = 12$. The solutions obtained with $N = 10$ are even better. Therefore, the present scheme outperforms Muroya’s method [21] and Taylor method [29] for this example problem. We also included in Table 1 the CPU times (in seconds) needed to compute the approximate solutions in a computer with a 2.80 GHz processor.

We now illustrate the technique of residual correction for this example:

$$u'(x) = -u(x) + 0.1u(0.2x) - 0.1e^{-0.2x}, \quad u(0) = 1, \quad 0 \leq x \leq 1. \tag{8}$$

Taking $N = 4$ and employing our method results in the approximate solution

$$u_4(x) = 0.0242846452x^4 - 0.1510973316x^3 + 0.4937451715x^2 - 0.9990362373x + 1.$$

Substituting this into Eq. (8) yields the residual function

$$R_4(x) = 0.1e^{-0.2x} + 0.0242807x^4 - 0.0538378x^3 + 0.0384781x^2 - 0.0084348x - 0.0990362.$$

Now, subtracting from this the residual expression corresponding to Eq. (8) gives the following multi-pantograph equation:

$$e_4'(x) = -e_4(x) + 0.1e_4(0.2x) - R_4(x), \quad e_4(0) = 0, \quad 0 \leq x \leq 1.$$

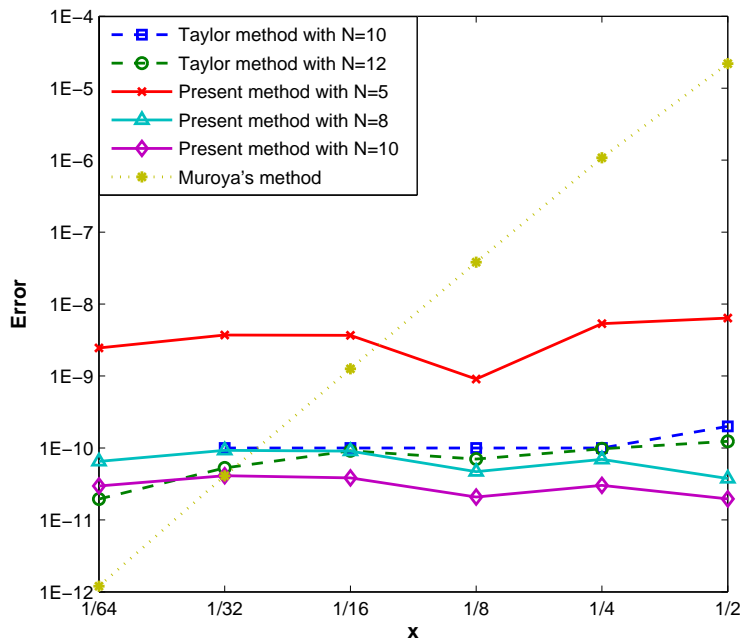


Figure 1: Graphics of the absolute error functions of Muroya method, Taylor method and the present method for different values of N in Example 1.

Applying our method with $M = 5$, we obtain the approximate solution

$$e_{4,5}(x) = -0.0048905037x^5 + 0.0135052502x^4 - 0.0133984972x^3 + 0.0056679576x^2 - 0.0009002485x.$$

The last step is to correct the approximate solution $u_4(x)$ by adding $e_{4,5}(x)$ to it, yielding the corrected approximate solution

$$u_{4,5}(x) = -0.0048905037x^5 + 0.0377898954x^4 - 0.1644958289x^3 + 0.4994131291x^2 - 0.9999364858x + 1.$$

The corrected solutions for any other (N, M) pair can be calculated similarly. Table 2 compares the absolute error values for the first approximate solution $u_N(x)$, the estimated absolute errors $e_{N,M}(x)$ and the absolute errors of the corrected approximate solution $u_{N,M}(x)$ for the parameter values $N = 4, M = 5$ and $N = 7, M = 8$. Figure 2 graphically illustrates the significant improvement in the absolute error function as a result of the procedure. The CPU times included in Table 2 reveal that residual correction does not bring about too much computational burden.

As an illustration of the error bound given in Theorem 3.1, let us consider the Maclaurin polynomial of the exact solution $u(x) = e^{-x}$ truncated after the fifth term:

$$e^{-x} = 1 - x + \frac{x^2}{2} - \frac{x^3}{6} + \frac{x^4}{24} - \frac{x^5}{120} + \frac{e^{-\xi}}{720}x^6.$$

Subtracting $u_{4,5}(x)$ from this then yields the following:

$$|u(x) - u_{4,5}(x)| \leq |-0.0000635142x + 0.0005868709x^2 - 0.0021708377x^3 + 0.0038767712x^4 - 0.0034428296x^5| + \frac{e^{-\xi}}{720}x^6.$$

Table 2: Actual, estimated and corrected errors at some points for Example 1 with parameter values $N = 4, M = 5$ and $N = 7, M = 8$.

x	Actual error for $u_4(x)$ $N = 4$	Estimated error $e_{4,5}(x)$ $N = 4, M = 5$	Actual error for $u_{4,5}(x)$ $N = 4, M = 5$
0.2	$4.18E - 5$	$4.04E - 5$	$1.40E - 6$
0.4	$1.61E - 5$	$1.50E - 5$	$1.06E - 6$
0.6	$2.51E - 5$	$2.37E - 5$	$1.38E - 6$
0.8	$2.41E - 5$	$2.35E - 5$	$6.06E - 7$
1	$1.68E - 5$	$1.60E - 5$	$7.64E - 7$
CPU time(s)	0.0469	0.1250	
x	Actual error for $u_7(x)$ $N = 7$	Estimated error $e_{7,8}(x)$ $N = 7, M = 8$	Actual error for $u_{7,8}(x)$ $N = 7, M = 8$
0.2	$1.30E - 9$	$1.25E - 9$	$5.14E - 11$
0.4	$1.97E - 9$	$1.93E - 9$	$3.99E - 11$
0.6	$1.09E - 9$	$1.04E - 9$	$4.99E - 11$
0.8	$1.25E - 9$	$1.23E - 9$	$2.47E - 11$
1	$9.60E - 10$	$9.31E - 10$	$2.81E - 11$
CPU time(s)	0.3438	0.5157	

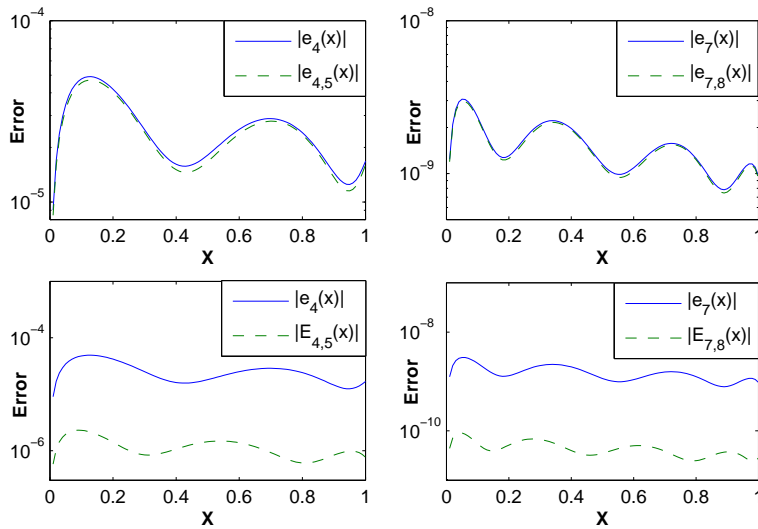


Figure 2: Comparison of the actual absolute error function $|e_N(x)|$, the estimated absolute error function $|e_{N,M}(x)|$ and the actual absolute error function $|E_{N,M}(x)|$ of the corrected result $u_{N,M}(x)$ for $N = 4, M = 5$ and $N = 7, M = 8$ in Example 1.

Table 3: Absolute errors at some points for Example 2.

x	Spline [9]	Taylor method [29]		Present method	
		N = 7	N = 9	N = 7	N = 9
0.2	1.980E – 8	2.560E – 11	7.050E – 15	6.261E – 9	1.082E – 11
0.4	4.730E – 8	3.330E – 9	1.060E – 11	1.089E – 8	1.615E – 11
0.6	8.470E – 8	5.770E – 8	2.940E – 10	1.081E – 8	2.074E – 11
0.8	1.350E – 7	4.380E – 7	3.860E – 9	1.497E – 8	2.230E – 11
1	2.010E – 7	2.120E – 6	2.900E – 8	1.788E – 8	2.931E – 11
CPU time(s)				0.1406	0.1719

Taking supremum of the absolute value of the above 5-th degree polynomial over the interval [0, 1] yields the value 0.0000023114 attained at $x \approx 0.0914$. The remainder term is clearly bounded from above by $\frac{1}{720}$. Adding these two values together, we get $|E_{4,5}(x)| \leq 0.0013912003$, which is consistent with the values given in Table 2.

The same analysis can be carried out for the corrected solution $u_{7,8}(x)$ corresponding to $N = 7, M = 8$. Since

$$u_7(x) = -0.0001174389x^7 + 0.0012605197x^6 - 0.0082208034x^5 + 0.0416099371x^4 - 0.1666505469x^3 + 0.4999976284x^2 - 0.9999998539x + 1$$

and

$$e_{7,8}(x) = 0.0000147213x^8 - 0.0000626809x^7 + 0.0001094299x^6 - 0.0001007312x^5 + 0.0000522907x^4 - 0.0000151501x^3 + 0.0000022598x^2 - 0.0000001405x,$$

summing $u_7(x)$ and $e_{7,8}(x)$ we obtain

$$u_{7,8}(x) = 0.0000147213x^8 - 0.0001801198x^7 + 0.0013699496x^6 - 0.0083215346x^5 + 0.0416622278x^4 - 0.1666656970x^3 + 0.4999998882x^2 - 0.9999999944x + 1.$$

Then, expanding the exact solution $u(x) = e^{-x}$ as before, this time to nine terms, we find that $\sup_{0 \leq x \leq 1} |u_{7,8}(x) -$

$$u_{7,8}(x)| = 2.14 \times 10^{-9} \text{ and } |R_{7,8}(x)| = \frac{e^{-\xi}}{362880} \leq \frac{1}{362880}. \text{ Therefore, in view of Theorem 3.1, we have}$$

$$|E_{7,8}(x)| = |u(x) - u_{7,8}(x)| \leq 2.14 \times 10^{-9} + \frac{1}{362880} \approx 2.857 \times 10^{-6}.$$

The upper bounds we have found for the errors of both $u_{4,5}(x)$ and $u_{7,8}(x)$ reveal that the bound provided by Theorem 3.1 is a loose one. This comes from the fact that the residual term $R_{T,M}(x)$ dominates the calculations, partly due to taking supremum on ξ over the entire interval $[a, b]$ even though the real value of $|f^{(M+1)}(\xi)|$ might be much smaller in reality.

Example 2. Our second example is the following homogeneous equation first studied in [30]:

$$u'(x) = \frac{u(x)}{2} + \frac{1}{2}e^{x/2}u(x/2), \quad u(0) = 1, \quad 0 \leq x \leq 1.$$

The exact solution is $u(x) = e^x$. Table 3 and Figure 3 compare our method with two older methods for different values of N . It is revealed that our method slightly outperforms Taylor method with the same N values, especially for greater values of x . In addition, the error in our method does not change significantly throughout the interval $[0, 1]$.

We applied residual correction to the approximate solution $u_7(x)$ for various values of M . The actual error values after the application of residual correction are shown in Table 4. The data in the table are

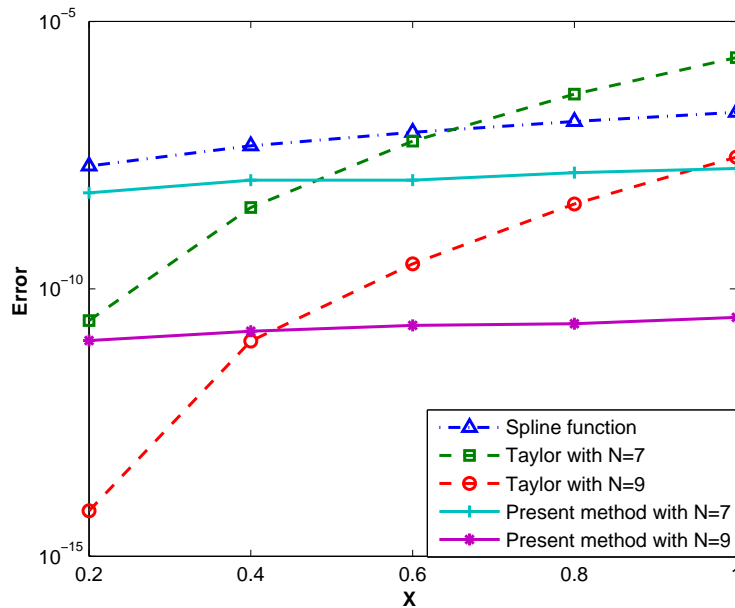


Figure 3: Graphics of the absolute error functions of Spline function method, Taylor method and the present method for different values of N in Example 2.

illustrated graphically in Figure 4. It is obvious from the table and the graph that increasing M brings about a significant improvement in the approximate solution. As for the upper bounds given by Theorem 3.1, we have $|E_{7,8}(x)| \leq 1.055 \times 10^{-5}$, $|E_{7,9}(x)| \leq 1.052 \times 10^{-6}$ and $|E_{7,10}(x)| \leq 3.708 \times 10^{-7}$.

Example 3. Let us next consider the following nonhomogeneous equation investigated by the authors of [4]:

$$u'(x) = au(x) + bu(qx) + \cos(x) - a \sin(x) - b \sin(qx), \quad u(0) = 0, \quad 0 \leq x \leq 1.$$

The exact solution is $u(x) = \sin(x)$. Table 5 compares the maximum absolute errors of the present method and the methods of [4] on the interval $[0, 1]$, for the parameter values $a = -1, b = 0.5, q = 0.5$. It can be commented from Table 5 that our method with $N = 5$ gives a better result than the methods of [4]. Employing residual error correction with $N = 6$ and $N = 8$ greatly improves the result, as also depicted in Figure 5. As for the error bounds of Theorem 3.1, we have $|E_{5,6}(x)| \leq 0.000394$ and $|E_{5,8}(x)| \leq 5.485 \times 10^{-6}$.

Example 4. Our last example is the following problem:

$$u'(x) = u(x) + (x^2 + 1)u(0.2x) - \frac{3 + 25x(2 + 3x + 25x^3)}{(1 + 25x^2)^2}, \quad u(0) = 1, \quad 0 \leq x \leq 1. \tag{9}$$

The exact solution of this problem is $u(x) = \frac{1}{1+25x^2}$. The function $\frac{1}{1+25x^2}$ was originally given by Runge [26] as an example to functions for which polynomial interpolation with equidistant nodes does not converge. This situation, known as “Runge phenomenon”, has been addressed in many studies, see [2] for an example. It can be interesting to apply the present scheme to Eq. (9) and observe the change in the error as we increase N . With this in mind, we obtained its approximate solutions corresponding to several N values and collected the results in Table 6. Looking at the values in the table, we see that the approximate solutions exhibit a general improvement as we increase the parameter N , although the decrease in the error is not as fast as in the case of previous problems. This situation is also apparent from Figure 6. Still, the steady decrease in errors suggests that the present method yields reasonable results when applied to a class of more challenging problems like this one.

Table 4: Comparison of the actual absolute errors for $u_7(x)$ and its three improvements for $M = 8, 9, 10$ in Example 2.

x	Actual error for $u_7(x)$ $N = 7$	Actual error for $u_{7,8}(x)$ $N = 7, M = 8$	Actual error for $u_{7,9}(x)$ $N = 7, M = 9$	Actual error for $u_{7,10}(x)$ $N = 7, M = 10$
0.2	6.26E – 9	2.17E – 10	6.83E – 12	2.08E – 13
0.4	1.08E – 8	2.63E – 10	6.45E – 12	2.15E – 13
0.6	1.08E – 8	3.75E – 10	9.50E – 12	2.49E – 13
0.8	1.49E – 8	4.02E – 10	1.15E – 12	3.24E – 13
1	1.78E – 8	5.16E – 10	1.35E – 12	3.96E – 13
CPU time(s)	2.0000	2.3750	2.7656	

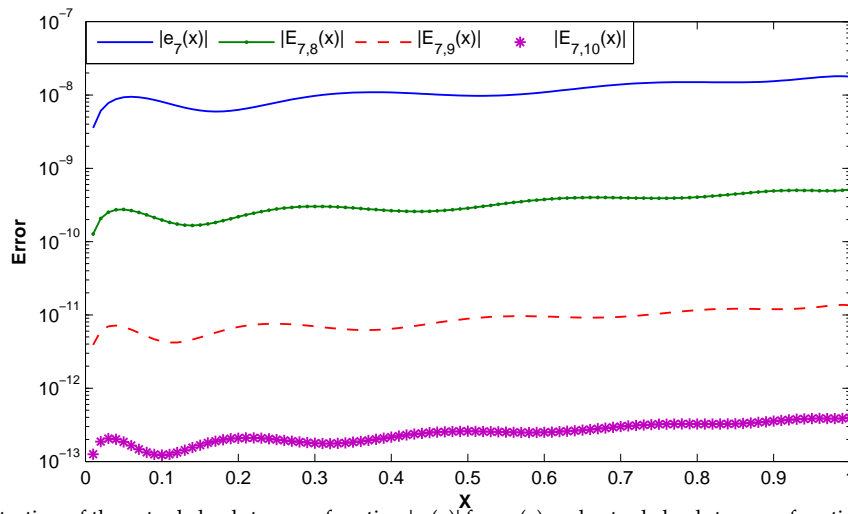


Figure 4: Illustration of the actual absolute error function $|e_7(x)|$ for $u_7(x)$ and actual absolute error functions $|E_{7,8}(x)|, |E_{7,9}(x)|, |E_{7,10}(x)|$ of its three improvements in Example 2.

Table 5: Maximum errors for solutions of Example 3 with $a = -1, b = q = 0.5$.

	Piecewise linear DG with $m = 64$ [4]	Piecewise quadratic DG with $m = 16$ [4]	Present method with $N = 5$	Present method with $N = 5, M = 6$	Present method with $N = 5, M = 8$
—Error—	2.142×10^{-5}	1.464×10^{-6}	1.398×10^{-6}	1.018×10^{-7}	8.891×10^{-11}
CPU time(s)			0.0781	0.1719	0.2344

Table 6: Absolute errors at some points corresponding to several N values in Example 4.

x	$N = 4$	$N = 5$	$N = 6$	$N = 9$	$N = 12$	$N = 14$
0.2	3.71E – 2	3.12E – 2	1.04E – 2	1.60E – 3	1.52E – 4	4.68E – 5
0.4	1.01E – 2	3.35E – 2	1.48E – 2	4.07E – 4	9.12E – 5	4.28E – 5
0.6	7.15E – 2	1.73E – 2	7.89E – 3	1.34E – 3	8.20E – 5	2.44E – 5
0.8	4.62E – 2	2.14E – 3	8.27E – 3	8.49E – 4	1.43E – 4	2.35E – 6
1	4.71E – 2	3.27E – 3	4.75E – 3	8.97E – 4	3.56E – 5	1.41E – 5
CPU time(s)	0.0625	0.0781	0.0938	0.1406	0.2031	0.2813

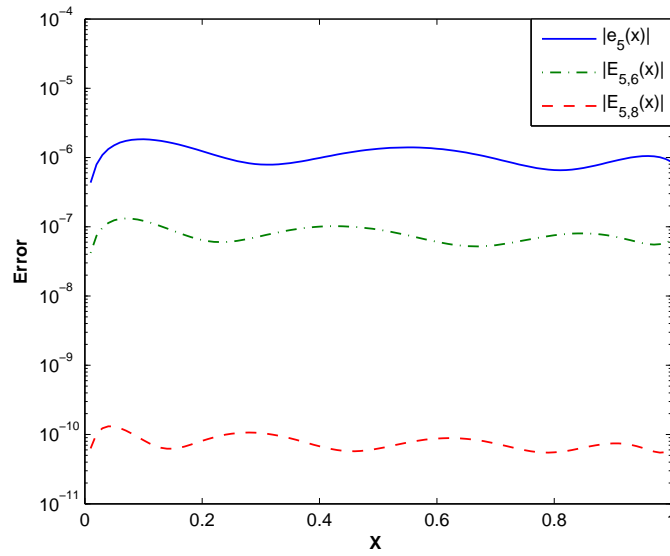


Figure 5: Illustration of the actual absolute error function $|e_5(x)|$ for $u_5(x)$ and actual absolute error functions $|E_{5,6}(x)|, |E_{5,8}(x)|$ of its improvements in Example 3.

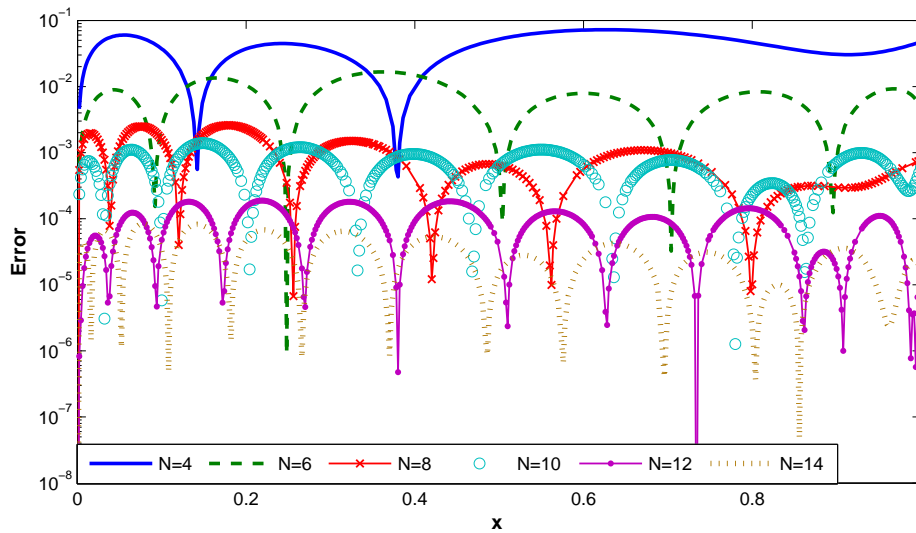


Figure 6: Illustration of the actual absolute error functions corresponding to solutions obtained by $N = 4, 6, 8, 10, 12, 14$ in Example 4.

5. Conclusions

In this paper, we have presented a Galerkin-like approach for the approximate solution of multi-pantograph equations. In brief, the method comprises taking inner product of an expression obtained from the equation with a set of $N + 1$ monomials. The result, then, is a polynomial of degree N , which is taken as the approximate solution. The method has been tested on some problems taken from other works and the results have been compared with several popular methods present in the literature. The related tables and figures reveal that our approach gives better results compared to the methods in question. In addition, results of the last problem suggest that the present numerical scheme can be expected to yield acceptable results when applied to more problematic examples. Residual error correction technique, which uses the residual function of an approximate solution to estimate its error and thus to reduce it, has also been discussed. Simulation results show that substantial improvements in the approximate solutions can be achieved as a result of residual correction with negligible computational cost. On the whole, the numerical scheme presented in this paper is a fast and easy-to-implement method that can be relied on when one needs to solve multi-pantograph equations with a high degree of accuracy.

Acknowledgement

The authors are supported by the Scientific Research Project Administration of Akdeniz University (FDK-2017-2399).

References

- [1] D. S. Battisti, On the role of off-equatorial oceanic Rossby waves during ENSO, *J. Phys. Oceanogr.*, **19**(1989), 551-559.
- [2] J. P. Boyd, J. R. Ong, Exponentially-convergent strategies for defeating the Runge phenomenon for the approximation of non-periodic functions, Part I: Single-interval schemes, *Commun. Comput. Phys.*, **5**(2009), 484-497.
- [3] H. Brunner, *Collocation Methods for Volterra Integral and Related Functional Equations*, Cambridge University Press, Cambridge, 2004.
- [4] H. Brunner, Q. Huang, H. Xies, Discontinuous Galerkin methods for delay differential equations of pantograph type, *SIAM J. Numer. Anal.*, **48**(2010), 1944-1967.
- [5] S. A. Campbell, *Time Delays in Neural Systems*, available at <http://www.math.uwaterloo.ca/~sacampbe/preprints/brain.pdf>.
- [6] R.V. Culshaw, S. Ruan, A delay-differential model of HIV infection of $CD4^+$ T-cells, *Math. Biosci.*, **165**(2000), 27-39.
- [7] M. Dehghan, M. Shakourifar, A. Hamidi, The solution of linear and non-linear systems of Volterra functional equations using Adomian-Padé technique, *Chaos Solitons Fract.*, **39**(2009), 2509-2521.
- [8] G. A. Derfel, F. Vogl, On the asymptotics of solutions of a class of linear functional-differential equations, *Eur. J. Appl. Math.*, **7**(1996), 511-518.
- [9] A. El-Safty, S. M. Abo Hasha, On the application of spline functions to initial value problems with retarded arguments, *Int. J. Comput. Math.*, **32**(1990), 173-179.
- [10] A. Feldstein, On neutral functional-differential equations with variable time delays, *Math. Proc. Camb. Phil. Soc.*, **124**(1998), 371-398.
- [11] X. Feng, An analytic study on the multi-pantograph delay equations with variable coefficients, *Bull. Math. Soc. Sci. Math. Roumanie Tome*, **56**(2013), 205-215.
- [12] J. E. Forde, *Delay Differential Equation Models in Mathematical Biology*, PhD Dissertation, University of Michigan, 2005.
- [13] J. H. He, Homotopy Perturbation Technique, *Comput. Methods Appl. Mech. Engrg.*, **178**(1999), 257-262.
- [14] J. H. He, Variational iteration method—a kind of non-linear analytical technique: some examples, *Int. J. Nonlinear Mech.*, **34**(1999), 699-708.
- [15] P. Höfer, A. Lion, Modelling of frequency-and amplitude-dependent material properties of filler-reinforced rubber, *J. Mech. Phys. Solids*, **57**(2009), 500-520.
- [16] E. Ishiwata, Y. Muroya, On collocation methods for delay differential and Volterra integral equations with proportional delay, *Front Math. China*, **4**(2009), 89-111.
- [17] R. Işık, Z. Güney, M. Sezer, Bernstein series solutions of pantograph equations using polynomial interpolation, *J. Differ. Equ. Appl.*, **18**(2012), 357-374.
- [18] W. Just, T. Bernard, M. Ostheimer, E. Reibold, H. Benner, Mechanism of time-delayed feedback control, *Phys. Rev. Lett.*, **78**(1997), 203-206.
- [19] Y. Kuang, *Delay Differential Equations with Applications in Population Dynamics*, Academic Press, New York, 1993.
- [20] N. MacDonald, *Biological Delay Systems: Linear Stability Theory*, Cambridge University Press, Cambridge, 1989.
- [21] Y. Muroya, E. Ishiwata, H. Brunner, On the attainable order of collocation methods for pantograph integro-differential equations, *J. Comput. Appl. Math.*, **152**(2003), 347-366.

- [22] J. R. Ockendon, A. B. Tayler, The dynamics of a current collection system for an electric locomotive, *Proc. Roy. Soc. London, Ser. A*, **322**(1971), 447-468.
- [23] K. Pyragas, Control of chaos via extended delay feedback, *Phys. Lett. A*, **206**(1995), 323-330.
- [24] M. Rendek, A. Lion, Amplitude dependence of filler-reinforced rubber: Experiments, constitutive modelling and FEM-Implementation, *Int. J. Solids Struct.*, **47**(2010), 2918-2936.
- [25] M. Rossetti, P. Bardella, I. Montrosset, Modeling Passive Mode-Locking in Quantum Dot Lasers: A Comparison Between a Finite-Difference Traveling-Wave Model and a Delayed Differential Equation Approach, *IEEE J. Quant. Electron.*, **47**(2011), 569-576.
- [26] C. Runge, Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, *Zeitschrift für Mathematik und Physik*, **246**(1901), 224-243.
- [27] A. Saadatmandi, M. Dehghan, Variational iteration method for solving a generalized pantograph equation, *Comput. Math Appl.*, **58**(2009), 2190-2196.
- [28] S. Sedaghat, Y. Ordokhani, M. Dehghan, Numerical solution of the delay differential equations of pantograph type via Chebyshev polynomials, *Commun. Nonlinear Sci. Numer. Simulat.*, **17**(2012), 4815-4830.
- [29] M. Sezer, S. Yalçınbaş, M. Gülsu, A Taylor polynomial approach for solving generalized pantograph equations with nonhomogeneous term, *Int. J. Comput. Math.*, **85**(2011), 1055-1063.
- [30] M. Shadia, Numerical solution of delay differential and neutral differential equations using spline methods, PhD Dissertation, Assuit University, 1992.
- [31] L. F. Shampine, P. Gahinet, Delay-differential-algebraic equations in control theory, *Appl. Numer. Math.*, **56**(2006), 574-578.
- [32] L. F. Shampine, L. Thompson, Numerical Solution of Delay Differential Equations, available at <http://www.radford.edu/~thompson/RP/solvingddes.pdf>.
- [33] A. B. Tayler, *Mathematical Models in Applied Mechanics*, Clarendon Press, Oxford, 1986.
- [34] M. Türkyılmazoğlu, An effective approach for numerical solutions of high-order Fredholm integro-differential equations, *Appl. Math. Comput.*, **227**(2014), 384-398.
- [35] Ş. Yüzbaşı, M. Karaçayır, A Galerkin-like approach to solve high-order integro-differential equations with weakly singular kernel, *Kuwait J. Sci.*, **43**(2016), 106-120.
- [36] Ş. Yüzbaşı, M. Sezer, An exponential approximation for solutions of generalized pantograph-delay differential equations, *Appl. Math. Model.*, **37**(2013), 9160-9173.
- [37] Ş. Yüzbaşı, M. Sezer, Shifted Legendre approximation with the residual correction to solve pantograph-type differential equations, *Appl. Math. Model.*, **39**(2015), 6529-6542.
- [38] Ş. Yüzbaşı, N. Şahin, M. Sezer, A Bessel collocation method for numerical solution of generalized pantograph equations, *Numer. Meth. Part. D. E.*, **28**(2011), 1105-1123.