SAKARYA UNIVERSITY
JOURNAL OF SCIENCE

e-ISSN: 2147-835X
http://www.saujs.sakarya.edu.tr

Received
26-01-2018
Accepted
10-04-2018

Doi
10.16984/saufenbilder.384592

A numerical technique based on Lucas polynomials together with standard and Chebyshev-Lobatto collocation points for solving functional integro-differential equations involving variable delays

Sevin Gümgüm*[1], Nurcan Baykuş Savaşaneril[2], Ömür Kıvanç Kürkçü[1], Mehmet Sezer[3]

Abstract

In this paper, a new numerical matrix-collocation technique is considered to solve functional integro-differential equations involving variable delays under the initial conditions. This technique is based essentially on Lucas polynomials together with standard and Chebyshev-Lobatto collocation points. Some descriptive examples are performed to observe the practicability of the technique and the residual error analysis is employed to improve the obtained solutions. Also, the numerical results obtained by using these collocation points are compared in tables and figures.

Keywords: functional equations, matrix technique, collocation points, Lucas polynomials, Residual error analysis

1. INTRODUCTION

In this paper, we employ a new numerical technique based on Lucas polynomials to solve the following functional integro-differential equation with variable delays

$$\sum_{k=0}^{m} P_k(t)\, y^{(k)}(t) + \sum_{r=0}^{m_1} Q_r(t)\, y^{(r)}(\alpha_r t + \beta_r(t)) = g(t) + \sum_{j=0}^{m_2} \gamma_j \int_{u_j(t)}^{v_j(t)} K_j(t,s)\, y(\lambda_j s + \mu_j) ds \qquad (1)$$

under the inital conditions

$$\sum_{k=0}^{m-1} a_{ik} y^{(k)}(a) = \varphi_i \ , i = 0,1,2,\ldots,m-1 \qquad (2)$$

where $P_k(t)$, $Q_r(t)$, $\beta_r(t)$, $g(t)$, $K_j(t,s)$, $u_j(t)$ and $v_j(t)$ ($m_1 \le m$, $u_j(t) < v_j(t)$) are analytic functions defined on $a \le t,s \le b$; α_r, γ_j, λ_j, μ_j, a_{ik} and φ_i are suitable constants.

Functional differential and integro-differential equations with variable delays in the form (1) are usually used in modelling of physical phenomena and play an important role in mathematics, viscoelasticity, oscillating magnetic field, heat conduction, electromagnetics, biology and etc. [1-22,28-31]. It is generally hard to find the analytic solution of them. So, the numerical techniques are required to obtain their approximate solution. For example, some integro-differential equations and

* Corresponding Author
[1] Izmir University of Economics, Department of Mathematics, Izmir, sevin.gumgum@ieu.edu.tr; omur.kivanc@ieu.edu.tr
[2] Dokuz Eylul University, Izmir Vocational School, Izmir, nurcan.savasaneril@deu.edu.tr
[3] Manisa Celal Bayar University, Department of Mathematics, Manisa, mehmet.sezer@cbu.edu.tr

their other classes have been solved by using the numerical techniques such as homotopy perturbation [2], variational iteration [3]; Legendre [5], Taylor [6-9], Laguerre [10,11], Taylor-Lucas [12], Dickson [13-15], Chelyshkov [16], Lucas [17], Bessel [18], Bernoulli [19,20], Chebyshev [28,29] polynomial techniques and also, B-Splines [21], backward substitution [22], Sinc techniques [30].

In this paper, by considering the matrix technique based on collocation points, which have been used by Sezer and coworkers [5,6,8-19], we purpose a new numerical technique to find an approximate solution of the problem (1)-(2). The solution is of the form

$$y(t)\cong y_N(t)=\sum_{n=0}^{N}a_nL_n(t)\,, \tag{3}$$

where $L_n(t)$ is the Lucas polynomials and a_n, $n=0,1,2,...,N$ are unknown coefficients [12].

2. SOME BASICS OF LUCAS POLYNOMIALS

The Lucas polynomials are defined as follows [23-26]:

$$L_{n+1}(t)=t\,L_n(t)+L_{n-1}(t)\,,\ n\geq 1$$

with $L_0(t)=2$ and $L_1(t)=t$ or their explicit form is

$$L_n(t)=\sum_{k=0}^{\lfloor n/2\rfloor}\frac{n}{n-k}\binom{n-k}{k}t^{n-2k}, \tag{4}$$

where $n\geq 1$ and $\lfloor x\rfloor$ is the largest integer smaller than or equal to x.

The Lucas polynomials have the generating function [26]

$$\sum_{n=0}^{\infty}L_n(t)z^n=\frac{1+z^2}{1-z^2-zt}\,.$$

The derivative of $L_n(t)$ is of the form [26]

$$L_n'(t)=\frac{n}{t^2+4}\left(tL_n(t)+2L_{n-1}(t)\right).$$

The relation between Lucas and Fibonacci polynomials is [25]

$$\left(t^2+4\right)F_n(t)=L_{n+1}(t)+L_{n-1}(t)\,,$$

where $F_n(t)$ is the Fibonacci polynomials. For more properties of the Lucas polynomials, one can refer to [23-26].

3. FUNDAMENTAL MATRIX RELATIONS

In this section, we constitute the matrix forms of the unknown function $y(t)$ defined by the form (3), the derivative $y^{(k)}(t)$, the functional term $y^{(r)}(\alpha_r t+\beta_r(t))$, the kernel function $K_j(t,s)$ and the functional term $y(\lambda_j s+\mu_j)$ in Eq. (1). These matrix forms will enable us to solve the functional integro-differential equation (1) under the initial conditions (2). We can first write the truncated Lucas series (3) in the matrix form, for $n=0,1,2,...,N$,

$$y(t)\cong y_N(t)=\boldsymbol{L}(t)\boldsymbol{A}\,, \tag{5}$$

where

$$\mathbf{L}(t)=\begin{bmatrix}L_o(t) & L_1(t) & \cdots & L_N(t)\end{bmatrix},$$

$$\mathbf{A}=\begin{bmatrix}a_o & a_1 & \cdots & a_N\end{bmatrix}^T.$$

Then, by using the Lucas polynomials $L_n(t)$ given by (4), we write the matrix form $\boldsymbol{L}(t)$ as follows;

$$\boldsymbol{L}(t)=\boldsymbol{T}(t)\boldsymbol{M}^T \tag{6}$$

where

$$\boldsymbol{T}(t)=\begin{bmatrix}t & t^2 & \cdots & t^N\end{bmatrix},$$

if N is odd,

$$\mathbf{M}=\begin{bmatrix}
2 & 0 & 0 & \cdots & 0\\
0 & \frac{1}{1}\binom{1}{0} & 0 & \cdots & 0\\
\frac{2}{1}\binom{1}{1} & 0 & \frac{2}{2}\binom{2}{0} & \cdots & 0\\
0 & \frac{3}{2}\binom{2}{1} & 0 & \cdots & 0\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
\frac{(n-1)}{\binom{n-1}{2}}\binom{\frac{n-1}{2}}{\frac{n-1}{2}} & 0 & \frac{n-1}{\binom{n+1}{2}}\binom{\frac{n+1}{2}}{\frac{n-3}{2}} & \cdots & 0\\
0 & \frac{n}{\binom{n+1}{2}}\binom{\frac{n+1}{2}}{\frac{n-1}{2}} & 0 & \cdots & \frac{n}{n}\binom{n}{0}
\end{bmatrix}$$

if N is even,

$$\mathbf{M}=\begin{bmatrix} 2 & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{1}\binom{1}{0} & 0 & \cdots & 0 \\ \frac{2}{1}\binom{1}{1} & 0 & \frac{2}{2}\binom{2}{0} & \cdots & 0 \\ 0 & \frac{3}{2}\binom{2}{1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{n-1}{\binom{n}{2}}\binom{\frac{n}{2}}{\frac{n-2}{2}} & 0 & \cdots & 0 \\ \frac{n}{\binom{n}{2}}\binom{\frac{n}{2}}{\frac{n}{2}} & 0 & \frac{n}{\binom{\frac{n+2}{2}}{}}\binom{\frac{n+2}{2}}{\frac{n-2}{2}} & \cdots & \frac{n}{n}\binom{n}{0} \end{bmatrix}.$$

For example, we obtain M^T with N=4 as follows:

$$M^T=\begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 \\ 2 & 0 & 4 & 0 & 1 \end{bmatrix}.$$

By the matrix relations (5) and (6), it follows that

$$y_N(t)=\mathbf{T}(t)\mathbf{M}^T\mathbf{A}. \tag{7}$$

Besides, it is well known from [9] that the relation between $\boldsymbol{T}(t)$ and its derivative $\boldsymbol{T}^{(k)}(t)$ is of the form

$$\mathbf{T}^{(k)}(t)=\mathbf{T}(t)\mathbf{B}^k, \tag{8}$$

where

$$\mathbf{B}=\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & N \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$ and $\mathbf{B}^0$ is a unit matrix.

By using (7) and (8), we have the matrix relation

$$y_N^{(k)}(t)=\mathbf{T}(t)\mathbf{B}^k\mathbf{M}^T\mathbf{A},\, k=0,1,2,...,m. \tag{9}$$

Putting $t\to\alpha_r t+\beta_r(t)$ into (9), we obtain the matrix relation

$$y_N^{(r)}(\alpha_r t+\beta_r(t))=\mathbf{T}(t)\mathbf{S}^T(\alpha_r,\beta_r(t))\mathbf{B}^k\mathbf{M}^T\mathbf{A}, \tag{10}$$

where

$$\mathbf{S}(\alpha_r,\beta_r(t))=\begin{bmatrix} \binom{0}{0}(\alpha_r)^0(\beta_r(t))^0 & 0 & 0 & \cdots & 0 \\ \binom{1}{0}(\alpha_r)^0(\beta_r(t))^1 & \binom{1}{1}(\alpha_r)^1(\beta_r(t))^0 & 0 & \cdots & 0 \\ \binom{2}{0}(\alpha_r)^0(\beta_r(t))^2 & \binom{2}{1}(\alpha_r)^1(\beta_r(t))^1 & \binom{2}{2}(\alpha_r)^2(\beta_r(t))^0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \binom{N}{0}(\alpha_r)^0(\beta_r(t))^N & \binom{N}{1}(\alpha_r)^1(\beta_r(t))^{N-1} & \binom{N}{2}(\alpha_r)^2(\beta_r(t))^{N-2} & \cdots & \binom{N}{N}(\alpha_r)^N(\beta_r(t))^0 \end{bmatrix}$$

Note that the matrix $\mathbf{T}(\alpha_r t+\beta_r(t))$ can be constituted as $\mathbf{T}(\alpha_r t+\beta_r(t))=\mathbf{T}(t)\mathbf{S}^T(\alpha_r,\beta_r(t))$. On the other hand, the matrix forms of $y(\lambda_j s+\mu_j)$ and $K_j(t,s)$ can be written as

$$y(\lambda_j s+\mu_j)=\mathbf{T}(s)\mathbf{S}^T(\lambda_j,\mu_j)\mathbf{M}^T\mathbf{A} \tag{11}$$

and by using the Taylor series expansion of $K_j(t,s)$ [8], we have

$$K_j(t,s)=\mathbf{T}(t)\mathbf{K}_j\mathbf{T}^T(s),\ \mathbf{K}_j=\left[k_{pq}^j\right] \tag{12}$$

where

$$k_{pq}^j=\frac{1}{p!q!}\frac{\partial^{p+q}K_j(0,0)}{\partial t^p\partial s^q},\ p,q=0,1,...,N,$$
$$j=0,1,...,m_2.$$

By means of the relations (11) and (12), we obtain the matrix form of the integral part of Eq. (1) as follows;

$$\begin{aligned}\int_{u_j(t)}^{v_j(t)} K_j(t,s)\,y(\lambda_j s+\mu_j)ds&=\int_{u_j(t)}^{v_j(t)} \mathbf{T}(t)\mathbf{K}_j\mathbf{T}^T(s)\mathbf{T}(s)\mathbf{S}^T(\lambda_j,\mu_j)\mathbf{M}^T\mathbf{A}\,ds \\ &=\mathbf{T}(t)\mathbf{K}_j\mathbf{C}_j(t)\mathbf{S}^T(\lambda_j,\mu_j)\mathbf{M}^T\mathbf{A}\end{aligned} \tag{13}$$

where

$$\mathbf{C}_j(t)=\int_{u_j(t)}^{v_j(t)}\mathbf{T}^T(s)\mathbf{T}(s)ds=\left[c_{mn}^j(t)\right];$$

$$c_{mn}^j(t)=\frac{(v_j(t))^{m+n+1}-(u_j(t))^{m+n+1}}{m+n+1},\ m,n=0,1,...,N.$$

4. LUCAS MATRIX-COLLOCATION TECHNIQUE

In this section, we first constitute the following matrix equation corresponding to the functional integro-differential equation (1), by substituting the matrix relations (9), (10) and (13) into Eq. (1):

$$\left\{\begin{gathered}\sum_{k=0}^{m}\mathbf{P}_k(t)\mathbf{T}(t)\mathbf{B}^k+\sum_{r=0}^{m_1}\mathbf{Q}_r(t)\mathbf{T}(t)\mathbf{S}^T(\alpha_r,\beta_r(t))\mathbf{B}^r \\ -\sum_{j=0}^{m_2}\gamma_j\mathbf{T}(t)\mathbf{K}_j\mathbf{C}_j(t)\mathbf{S}^T(\lambda_j,\mu_j)\end{gathered}\right\}\mathbf{M}^T\mathbf{A}=g(t). \tag{14}$$

On the other hand, the standard (SCP) and Chebyshev-Lobatto (CLCP) collocation points we will use in the matrix equation (14) are defined respectively by

$$t_i = a + \frac{b-a}{N} i \text{ and } t_i = \frac{a+b}{2} + \frac{a-b}{2}\cos\left(\frac{\pi i}{N}\right),$$
$$i = 0,1,\ldots,N.$$

where

$$\mathbf{P_k} = diag\left[P_k(t_0) \quad P_k(t_1) \quad \cdots \quad P_k(t_N)\right],$$
$$\mathbf{Q_r} = diag\left[Q_r(t_0) \quad Q_r(t_1) \quad \cdots \quad Q_r(t_N)\right],$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}(t_0) \\ \mathbf{T}(t_1) \\ \vdots \\ \mathbf{T}(t_N) \end{bmatrix} = \begin{bmatrix} 1 & t_0 & \cdots & t_0^N \\ 1 & t_1 & \cdots & t_1^N \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & \cdots & t_N^N \end{bmatrix}, \bar{\mathbf{T}} = \begin{bmatrix} \mathbf{T}(t_0) & 0 & \cdots & 0 \\ 0 & \mathbf{T}(t_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{T}(t_N) \end{bmatrix},$$

$$\mathbf{S_r} = \begin{bmatrix} \mathbf{S}^T(\alpha_r, \beta_r(t_0)) \\ \mathbf{S}^T(\alpha_r, \beta_r(t_1)) \\ \vdots \\ \mathbf{S}^T(\alpha_r, \beta_r(t_N)) \end{bmatrix}, \quad \bar{\mathbf{C}_j} = \begin{bmatrix} C_j(t_0) & 0 & \cdots & 0 \\ 0 & C_j(t_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_j(t_N) \end{bmatrix},$$

$$\bar{\mathbf{S}}\left(\lambda_j, \mu_j\right) = \begin{bmatrix} \mathbf{S}^T\left(\lambda_j, \mu_j\right) \\ \mathbf{S}^T\left(\lambda_j, \mu_j\right) \\ \vdots \\ \mathbf{S}^T\left(\lambda_j, \mu_j\right) \end{bmatrix},$$

$$\overline{\mathbf{K}_j} = diag\left[\mathbf{K}_j \quad \mathbf{K}_j \quad \cdots \quad \mathbf{K}_j\right], \mathbf{G} = \left[g(t_0) \quad g(t_1) \quad \cdots \quad g(t_N)\right]^T.$$

Besides, the fundamental matrix equation (14) can be expressed in the form

$$\mathbf{WA} = \mathbf{G} \text{ or } \left[\mathbf{W};\ \mathbf{G}\right], \tag{15}$$

where

$$\mathbf{W} = \left[w_{mn}\right],\ m,n = 0,1,\ldots,N.$$

Now, by the relation (9), we can obtain the condition matrix form for the initial conditions (2) as

$$\mathbf{U}_i\mathbf{A} = \varphi_i \text{ or } \left[\mathbf{U}_i;\ \varphi_i\right],\ i = 0,1,\ldots,m-1 \tag{16}$$

such that

$$\mathbf{U}_i = \sum_{k=0}^{m-1} a_{ik}\mathbf{T}(a)\mathbf{B}^k\mathbf{M}^T = \left[u_{i0} \quad u_{i1} \quad \cdots \quad u_{iN}\right].$$

Eventually, in order to find the Lucas polynomial solution of the problem (1)-(2), by replacing *m* row matrices (16) into any *m* rows of the form (15). Thus, we have the augmented matrix

$$\left[\mathbf{W}^*;\ \mathbf{G}^*\right] \text{ or } \mathbf{W}^*\mathbf{A} = \mathbf{G}^*, \tag{17}$$

where

$$\left[\boldsymbol{W}^*;\boldsymbol{G}^*\right] = \begin{bmatrix} w_{00} & w_{01} & \cdots & w_{0N} & ; & g(t_0) \\ w_{10} & w_{11} & \cdots & w_{1N} & ; & g(t_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots; & \vdots \\ w_{N-m,0} & w_{N-m,1} & \cdots & w_{N-m,N} & ; & g(t_{N-m}) \\ u_{00} & u_{01} & \cdots & u_{0N} & ; & \varphi_0 \\ u_{10} & u_{11} & \cdots & u_{1N} & ; & \varphi_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots; & \vdots \\ u_{m-1,0} & u_{m-1,1} & \cdots & u_{m-1,N} & ; & \varphi_{m-1} \end{bmatrix}.$$

In Eq. (17), if rank $\mathbf{W}^*$=rank$\left[\mathbf{W}^*;\ \mathbf{G}^*\right] = N+1$, then the coefficient matrix $\mathbf{A}$ is uniquelly determined and so the solution of the problem (1)-(2) is obtained as

$$y_N(t) = \mathbf{L}(t)\mathbf{A} \text{ or } y_N(t) = \mathbf{T}(t)\mathbf{M}^T\mathbf{A}.$$

5. RESIDUAL ERROR ANALYSIS

In this section, an error analysis dependent on residual function is implemented to improve the Lucas polynomial solutions. By using Eq. (1), we can obtain the residual function on $t \in [a,b]$ as

$$\begin{aligned} R_N(t) = \sum_{k=0}^{m} P_k(t)\, y^{(k)}(t) + \sum_{r=0}^{m_1} Q_r(t)\, y^{(r)}(\alpha_r t + \beta_r(t)) \\ - \sum_{j=0}^{m_2} \gamma_j \int_{u_j(t)}^{v_j(t)} K_j(t,s)\, y(\lambda_j s + \mu_j)ds - g(t) \end{aligned} \tag{18}$$

In recent years, the residual error analysis has been applied by some authors [5,11,13,14,16,17,19,22]. Furthermore, the reader can refer to [15,27,28] for converge analysis based on residual function; residual correction and its theory. Let us now construct the residual error analysis for the Lucas polynomials. The error function $e_N(x)$ is defined by

$$e_N(t) = y(t) - y_N(t), \tag{19}$$

By Eqs. (18) and (19), the error equation is of the form

$$L\left[e_N(t)\right] = L\left[y(t)\right] - L\left[y_N(t)\right] = -R_N(t), \tag{20}$$

subject to the initial conditions

$$\sum_{k=0}^{m-1} a_{ik}e_N^{(k)}(a) = 0,\ i = 0,1,\ldots,m-1.$$

By Eqs. (19) and (20), we constitute the error problem. We solve this problem by following the procedure given in Sections 3 and 4. Thus, we obtain the estimated error function $e_{N,M}(t)$ (or called the solution of the error problem (19)-(20)), so

$$e_{N,M}(t)=\sum_{n=0}^{M} y_n^{*} L_n(t)\ ,\ (M>N).$$

Here, the corrected Lucas polynomial solution is obtained as $y_{N,M}(t)=y_N(t)+e_{N,M}(t)$ and the corrected error function is obtained as

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

6. NUMERICAL EXAMPLES

In this section, the practicability of the present technique are illustrated with the numerical results of some descriptive examples. Also, these results are discussed in tables and figures, by considering SCP and CLCP. A computer code written on Mathematica (on Pc with 2GB RAM and 2.80 GHz CPU) has been performed to obtain the precise results. In order to compare the numerical results, we also perform L_∞ error defined as follows [29]:

$$L_\infty=\max_{a\le i\le b}\left|y(t_i)-y_N(t_i)\right|,$$

where $y(t)$ is the exact solution.

6.1. Example 1:

Let us consider the second-order functional integro-differential equation with variable delays

$$y''(t)-y'(t)+2y'\left(\frac{t}{2}-\sin t\right)-2ty(t)=g(t)+\int_{t}^{e^{t+1}} 3t^2s^2y(s)ds$$

subject to the initial conditions $y(0)=0$ and $y'(0)=0$. Here, $0\le t\le 2$, $P_0(t)=-2t$, $P_1(t)=-1$, $P_2(t)=1$, $Q_1(t)=2$, $\alpha_1=1/2$, $\beta_1(t)=-\sin t$, $\gamma_0=1$, $K_0(t,s)=3t^2s^2$, $u_0(t)=t$, $v_0(t)=e^{t+1}$, $\lambda_0=1$, $\mu_0=0$,

$$g(t)=2-t^2\left(\frac{3e^{5t+5}}{5}+2t-\frac{3t^5}{5}\right)-4\sin(t).$$

We approximate the exact solution $y(t)$, by taking the Lucas polynomial solution form:

$$y(t)\cong y_2(t)=\sum_{n=0}^{2} a_n\,L_n(t)\,,\ 0\le t\le 2$$

and the standard collocation points for $a=0$, $b=2$ and $N=2$ are computed as $\{x_0=0,\ x_1=1,\ x_2=2\}$.

The fundamental matrix equation of this problem becomes

$$\left(\sum_{k=0}^{2}\mathbf{P}_k\mathbf{T}\mathbf{B}^k+\mathbf{Q}_1\left(\bar{\mathbf{T}}\right)\mathbf{S}^T(\alpha_1,\beta_1(t))\,\mathbf{B}^1-\gamma_0\bar{\mathbf{T}}\left(\overline{\mathbf{K}_0}\right)\left(\overline{\mathbf{C}_0}\right)\bar{\mathbf{S}}(1,0)\right)\mathbf{M}^T\mathbf{A}=\mathbf{G}$$

where

$$\mathbf{T}=\begin{pmatrix}1&0&0\\1&1&1\\1&2&4\end{pmatrix},\ \mathbf{B}=\begin{pmatrix}0&1&0\\0&0&2\\0&0&0\end{pmatrix},\ \mathbf{P}_0=\begin{pmatrix}0&0&0\\0&-2&0\\0&0&-4\end{pmatrix},$$

$$\mathbf{P}_1=\begin{pmatrix}-1&0&0\\0&-1&0\\0&0&-1\end{pmatrix},\ \mathbf{P}_2=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix},$$

$$\bar{\mathbf{T}}=\begin{bmatrix}1&0&0&0&0&0&0&0&0\\0&0&0&1&1&1&0&0&0\\0&0&0&0&0&0&1&2&4\end{bmatrix},$$

$$\mathbf{Q}_1=\begin{pmatrix}2&0&0\\0&2&0\\0&0&2\end{pmatrix},\ \mathbf{M}=\begin{bmatrix}2&0&0\\0&1&0\\2&0&1\end{bmatrix},\ \mathbf{G}=\begin{bmatrix}2\\-13218.6\\-7.85\times10^6\end{bmatrix},$$

$$\overline{\mathbf{C}_0}=\begin{bmatrix}2.71828&3.69453&6.69518&0&0&0&0&0&0\\3.69453&6.69518&13.6495&0&0&0&0&0&0\\6.69518&13.6495&29.6826&0&0&0&0&0&0\\0&0&0&6.389056&26.7991&134.143&0&0&0\\0&0&0&26.7991&134.143&744.989&0&0&0\\0&0&0&134.143&744.989&4405.09&0&0&0\\0&0&0&0&0&0&18.0855&199.714&2698.36\\0&0&0&0&0&0&199.714&2698.36&40684.7\\0&0&0&0&0&0&2698.36&40684.7&653797\end{bmatrix}$$

$$\mathbf{S}_1=\begin{bmatrix}\mathbf{S}^T(\alpha_1,\beta_1(0))\\\mathbf{S}^T(\alpha_1,\beta_1(1))\\\mathbf{S}^T(\alpha_1,\beta_1(2))\end{bmatrix}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\1&-\sin1&\sin^2(1)\\0&1&-\sin1\\0&0&1\\1&-\sin2&\sin^2(2)\\0&1&-\sin2\\0&0&1\end{bmatrix},\ \bar{\mathbf{S}}(1,0)=\begin{bmatrix}\mathbf{S}^T(1,0)\\\mathbf{S}^T(1,0)\\\mathbf{S}^T(1,0)\end{bmatrix}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

and

$$\overline{\mathbf{K}_0}=\begin{bmatrix}0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&3&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&3&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&3\end{bmatrix}.$$

Then, the augmented matrix is

$$[\mathbf{W}^*;\mathbf{G}^*]=\begin{bmatrix}0&1&2&;&2\\2&0&2&;&0\\0&1&0&;&0\end{bmatrix}.$$

Solving this system, $\mathbf{A}$ is obtained as $\mathbf{A}=\begin{bmatrix}-1&0&1\end{bmatrix}^T$. By the relation (7), $y(t)$ is obtained as

$$y(t)=\mathbf{T}(t)\mathbf{M}^T\mathbf{A}=\begin{bmatrix}1&t&t^2\end{bmatrix}\begin{bmatrix}2&0&2\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}-1\\0\\1\end{bmatrix},$$

thus, the solution of the problem becomes

$$y(t)=t^2,$$

which is the exact solution.

6.2. Example 2:

Consider the second-order integro-differential equation with variable delays

$$t^2y''(t)-2y'(t+1)-ty\left(\frac{t}{2}-\frac{t^2}{2}\right)=g(t)-\int_{t^2}^{t}tsy\left(\frac{s}{2}+1\right)ds$$

subject to $t\in[0,1]$, and the initial conditions $y(0)=1$ and $y'(0)=0$. The exact solution of this problem is $y(t)=\cos(t)$. Here, $g(t)$ can be easily found. By using SCP, we obtain the following solutions for N=4 and 7:

$$y_4(t)=1+2.77556\times10^{-16}t-0.520263t^2+0.0317765t^3+0.0259313t^4 \quad \text{and}$$

$$y_7(t)=1+1.36176\times10^{-16}t-0.499905t^2-0.000443857t^3+0.0422884t^4-0.000285t^5-0.00146t^6+0.000103t^7.$$

As seen from Figure 1, a fast approximation is provided, so the Lucas polynomial solutions coincide with the exact solution. Notice that the Lucas polynomial solutions obtained by using SCP are plotted in Figures 1-4. Also, in Table 1, the numerical results are obtained by using SCP and CLCP in our technique.

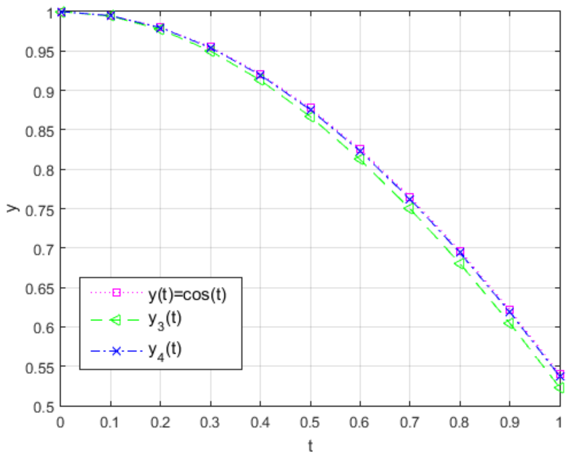

Figure 1. Comparison of the exact and Lucas polynomial solutions for Example 6.2.

Figure 2 and Table 1 show that the absolute errors decay, when N is increased and the residual error analysis is employed. In addition, when the interval of this problem is taken as the large interval [0,8], it is observed in Figure 3 that our solution $y_{20}(t)$ is in harmony with the exact solution. This consistency reflects on the phase plane of the solution $y_{20}(t)$ as seen in Figure 4.

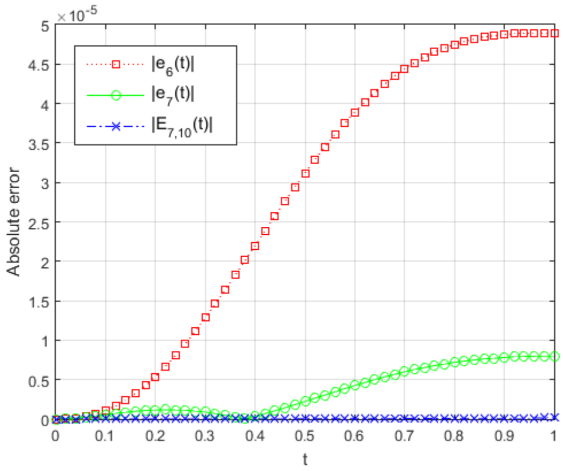

Figure 2. Behaviors of the absolute errors with respect to time t for Example 6.2.

Table 1. Comparison of the absolute errors and CPU time(s) (sec.) in terms of different collocation points for Example 6.2.

t_i	$\lvert e_4(t_i)\rvert$; SCP	$\lvert e_4(t_i)\rvert$; CLCP	$\lvert e_7(t_i)\rvert$; SCP	$\lvert e_7(t_i)\rvert$; CLCP
0.0	$2.22e-16$	$2.22e-16$	0	0
0.2	$5.81e-04$	$5.28e-04$	$1.16e-06$	$1.04e-06$
0.4	$1.61e-03$	$1.44e-03$	$3.03e-07$	$1.43e-05$
0.6	$2.41e-03$	$2.13e-03$	$4.12e-06$	$3.10e-05$
0.8	$2.78e-03$	$2.43e-03$	$6.95e-06$	$4.02e-05$
1.0	$2.86e-03$	$2.48e-03$	$7.63e-06$	$4.18e-05$
Time(s)	1.404 (N=4)	1.404 (N=4)	3.120 (N=7)	3.073 (N=7)

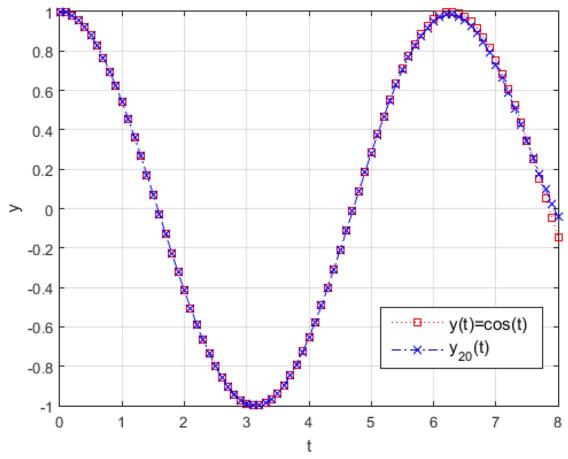

Figure 3. Oscillation of the exact and Lucas polynomial solution on large time interval [0,8] for Example 6.2.

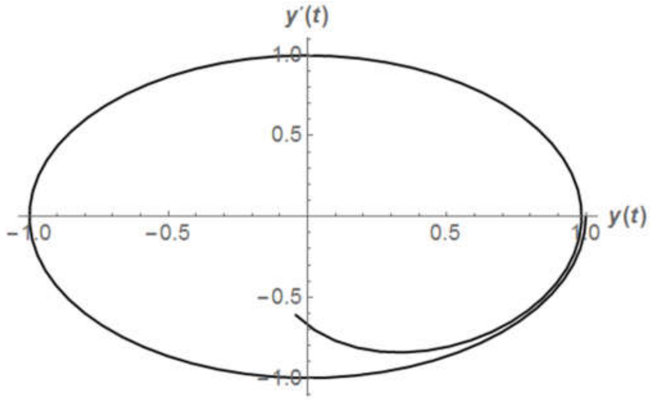

Figure 4. Consistency of the Lucas polynomial solution $y_{20}(t)$ in the phase plane for Example 6.2.

6.3. Example 3:

Consider the first-order pantograph Volterra delay-integro-differential equation [30]

$$y'(t)-y\left(\frac{t}{2}\right)=1-\frac{3t}{2}+\int_{0}^{t}y(s)\,ds+\int_{0}^{t/2}y(s)\,ds$$

$0\le t\le 1$,

subject to the initial condition $y(0)=0$. Here, the exact solution of the problem is $y(t)=1-e^{-t}$. After solving this problem by employing *N*=2-10, *M*=11, 12; SCP and CLCP, we obtain the Lucas polynomial solutions. It is seen from Figure 5 that we approach speedily to the exact solution, when *N*=2 and 3. It is also worth specifying in Figure 6 that our solutions obtained on [0,1] are consistent with the exact solution, even if these are on [0,5]. Notice that the Lucas polynomial solutions obtained by using CLCP are plotted in Figures 5 and 6.

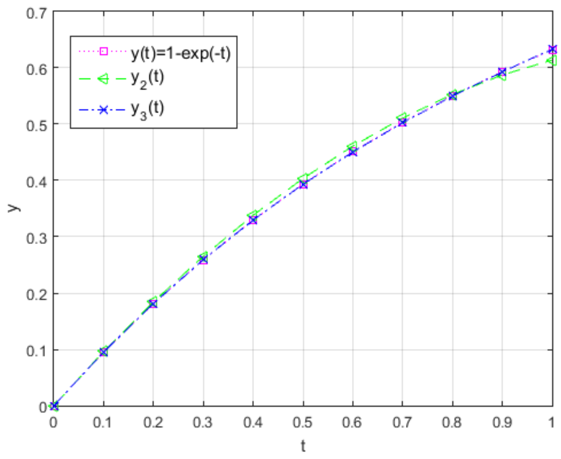

Figure 5. Comparison of the exact and Lucas polynomial solutions for Example 6.3.

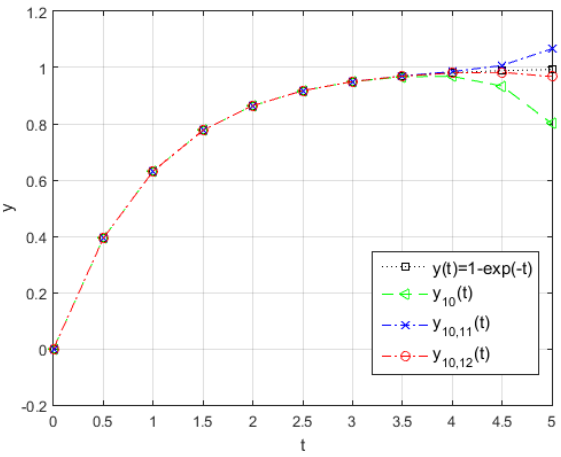

Figure 6. Comparison of the exact and Lucas polynomial solutions on large time interval [0,5] for Example 6.3.

By considering different collocation points, L_∞ errors are compared with the errors $\left\|E(h)\right\|_\infty$ of Sinc technique [30] in Table 2. As seen from Table 2, our error results are far better than those in the mentioned technique and CPU processes our computer program in a short time.

Table 2. Comparison of L_∞ errors and CPU time(s) (sec.) in terms of different *N* and collocation points for Example 6.3.

N,*M*	L_∞ error; SCP	L_∞ error; CLCP	$\left\|E(h)\right\|_\infty$ [30]	CPU time SCP	CPU time CLCP
2	$1.81e-02$	$1.81e-02$	–	0.3120	0.4056
3	$3.47e-03$	$1.10e-03$	–	0.4524	0.4836
4	$1.61e-04$	$6.27e-05$	–	0.5772	0.6084
5	$1.44e-05$	$1.61e-06$	$2.10e-03$	0.7644	0.8268
7	$3.48e-08$	$2.19e-09$	–	1.4040	1.2636
8	$1.18e-09$	$6.14e-11$	–	1.7472	1.5600
10	$1.55e-12$	$3.25e-14$	$1.26e-04$	2.9796	2.2152
10,11	$5.57e-14$	$6.61e-16$	–	n.a.	n.a.

6.4. Example 4:

Consider the first-order delay differential equation [31]

$$y'(t)-ty(t)-te^{3t^2/4}y\left(t-\left(\frac{t}{2}\right)\right)=0\,,\ 0\le t\le 2$$

subject to the initial condition $y(0)=1$. Here, the exact solution of the problem is $y(t)=e^{t^2}$.

Dix [31] investigated the asymptotic behavior of solutions of the first-order differential equation with variable delays, using Lyapunov functionals. We solve this problem, implementing our technique with SCP and CLCP for N=5-20 and 50. CPU time is obtained (sec.) as 0.0312 (N=5) and 0.796 (N=50) for CLCP. Also, we obtain L_∞ error as $8.53e-014$ for CLCP and N=50. The Lucas polynomial solutions are plotted along with the exact solution in Figure 7.

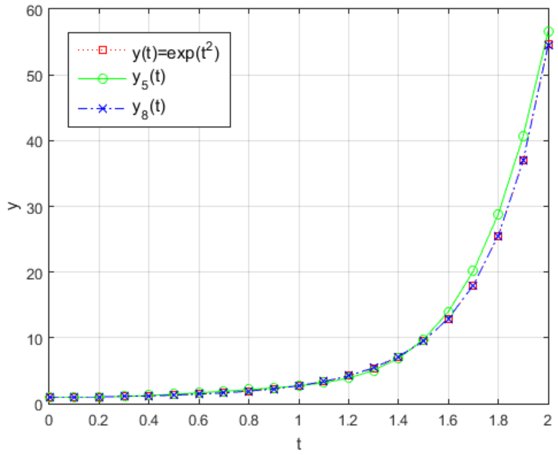

Figure 7. Comparison of the exact and Lucas polynomial solutions for Example 6.4.

Figure 8 shows that L_∞ errors decrease for different collocation points, as N is increased from 10 to 20. In addition, the numerical values of the exact, approximate solutions and L_∞ errors are compared in Table 3. It is clearly seen from Figure 8 and Table 3 that CLCP are more useful than SCP.

Table 3. Comparison of the exact, the approximate and the corrected approximate solutions and L_∞ errors in terms of different collocation points for Example 6.4.

t_i	Exact sol.	$y_{10}(t_i)$ SCP	$y_{10}(t_i)$ CLCP	$y_{10,13}(t_i)$ SCP	$y_{10,13}(t_i)$ CLCP
0.0	1	1	1	1	1
0.5	1.284	1.285	1.284	1.284	1.284
1.0	2.718	2.721	2.719	2.718	2.718
1.5	9.488	9.498	9.488	9.487	9.488
2.0	54.60	54.52	54.60	54.59	54.60
L_∞	-	$7.4e-02$	$9.1e-04$	$7.8e-03$	$4.3e-05$

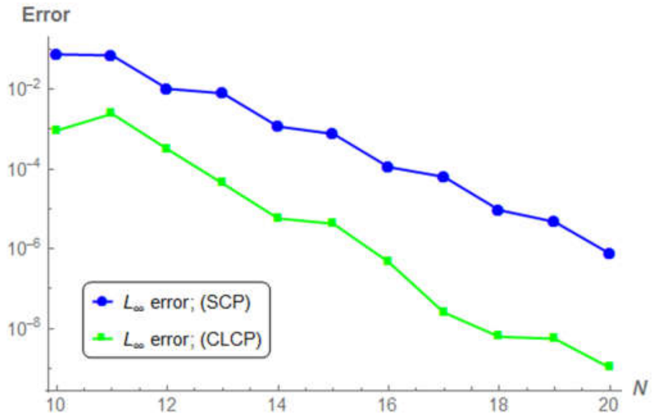

Figure 8. Logarithmic plot of L_∞ errors with SCP and CLCP with respect to N for Example 6.4.

7. CONSCLUSION

A practical Lucas matrix-collocation technique has been employed to solve functional integro-differential equations with variable delays. In Figures 1-8 and Tables 1-3, the comparisons of the present results shows that the proposed technique is very applicable, consistent and fast (according to CPU time(s)). The accuracy increases, as N is increased. Our solutions have been improved via the efficient residual error analysis as seen in Figures 2, 6 and Tables 2, 3.

On the other hand, by investigating the obtained results, we can deduce that the use of CLCP in the present technique is more advantageous than the use of SCP. It would be suitable to apply the present technique to other tough problems. In addition, it is readily seen that the present technique has a simple procedure and is very easy for computer programming.

REFERENCES

[1] V.V. Vlasov and R. Perez Ortiz, “Spectral Analysis of Integro-Differential Equations in Viscoelasticity and Thermal Physics”, *Math. Notes*, vol. 98, no. 4, pp. 689–693, 2015.

[2] M. Dehghan and F. Shakeri, “Solution of an integro-differential equation arising in oscillating magnetic field using He’s homotopy perturbation method”, *Prog. Electromagnet. Res. PIER*, vol. 78, pp. 361–376, 2008.

[3] Y. Chu, F. You, J. M. Wassick, and A. Agarwal, “Integrated planning and scheduling under production uncertainties: Bi-level model formulation and hybrid solution method,” *Computers and Chemical Engineering*, vol. 72, pp. 255–272, 2015.

[4] F. S. Chan, V. Kumar, and M. Tiwari, “Optimizing the Performance of an Integrated Process Planning and Scheduling Problem: An AIS-FLC based Approach,” *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–8, 2006.

[5] W. Tan and B. Khoshnevis, “Integration of process planning and scheduling— a review,” *Journal of Intelligent Manufacturing*, vol. 11, no. 1, pp. 51–63, 2000.

[6] N. Kurt and M. Sezer, “Polynomial solution of high-order linear Fredholm integro-differential equations with constant coefficients,” *J. Franklin Inst.*, vol. 345, pp. 839–850, 2008.

[7] N. Kurt and M. Çevik, “Polynomial solution of the single degree of freedom system by Taylor matrix method,” *Mech. Res. Commun.*, vol. 35, pp. 530–536, 2008.

[8] N. Baykuş and M. Sezer, “Solution of High-Order Linear Fredholm Integro-Differential Equations with Piecewise Intervals,” *Numer. Methods Partial Differ. Equ.*, vol. 27, pp. 1327–1339, 2011.

[9] M. Sezer and A. Akyüz-Daşcıoğlu, “A Taylor method for numerical solution of generalized pantograph equations with linear functional argument,” *J. Comput. Appl. Math.*, vol. 200, pp. 217–225, 2007.

[10] N. Baykuş Savaşaneril and M. Sezer, “Laguerre Polynomial Solution of High-Order Linear Fredholm Integro-Differential Equations”, *New Trends in Math. Sci.*, vol. 4, no. 2, 273–284, 2016.

[11] B. Gürbüz, M. Sezer and C. Güler, “Laguerre collocation method for solving Fredholm-integro-differential equations with functional arguments,” *J. Appl. Math.*, vol. 2014, Article ID: 682398, 12 pages, 2014.

[12] N. Baykuş Savaşaneril and M. Sezer, “Hybrid Taylor–Lucas Collocation Method for Numerical Solutional of High-Order Pantograph Type Delay Differential Equations with Variables Delays,” *Appl. Math. Inf. Sci.*, vol. 11, no. 6, pp. 1795–1801, 2017.

[13] Ö.K. Kürkçü, E. Aslan, and M. Sezer, “A numerical approach with error estimation to solve general integro-differential–difference equations using Dickson polynomials,” *Appl. Math. Comput.*, vol. 276, pp. 324–339, 2016.

[14] Ö.K. Kürkçü, E. Aslan, and M. Sezer, “A novel collocation method based on residual error analysis for solving integro-differential equations using hybrid Dickson and Taylor polynomials,” *Sains Malays.*, vol. 46, pp. 335–347, 2017.

[15] Ö.K. Kürkçü, E. Aslan, and M. Sezer, “A numerical method for solving some model problems arising in science and convergence analysis based on residual function,” *Appl. Numer. Math.*, vol. 121, pp. 134–148, 2017.

[16] C. Oğuz and M. Sezer, “Chelyshkov collocation method for a class of mixed functional integro-differential equations,” *Appl. Math. Comput.*, vol. 259, pp. 943–954, 2015.

[17] M. Çetin and M. Sezer, “C. Güler, Lucas polynomial approach for system of high-order linear differential equations and residual error estimation,” *Math. Prob. Eng.*, vol. 2015, Article ID: 625984, 14 pages, 2015.

[18] N. Şahin, Ş. Yüzbaşı, and M. Sezer, “A Bessel polynomial approach for solving general linear Fredholm integro-differential equations,” *Int. J. Comput. Math.*, vol. 88, no. 14, pp. 3093–3111, 2011.

[19] K. Erdem, S. Yalçınbaş, and M. Sezer, “A Bernoulli polynomial approach with residual correction for solving mixed linear Fredholm integro differential-difference equations,” *J. Difference Equ. Appl.*, vol. 19, no. 10, pp. 1619–1631, 2013.

[20] E. Tohidi, A.H. Bhrawy, and K. Erfani, "A collocation method based on Bernoulli operational matrix for numerical solution of generalized pantograph equation," *Appl. Math. Model.*, vol. 37, no. 6, pp. 4283–4294, 2013.

[21] H-E.D. Gherjalar and H. Mohammodikia, "Numerical solution of functional integral and integro-differential equations by using B-Splines," *Appl. Math.*, vol. 3, pp. 1940–1944, 2012.

[22] S. Yu. Reutskiy, "The backward substitution method for multipoint problems with linear Volterra-Fredholm integro-differential equations of the neutral type," *J. Comput. Appl. Math.*, vol. 296, pp. 724–738, 2016.

[23] T. Koshy, "Fibonacci and Lucas Numbers with Applications," *Wiley*, New York, USA, 2001.

[24] P. Filipponi and A.F. Horadam, "Second derivative sequences of Fibonacci and Lucas polynomials," *The Fibonacci Quarterly*, vol. 31, no. 3, pp. 194–204, 1993.

[25] A. Constandache, A. Das, and F. Toppan, "Lucas polynomials and a standard Lax representation for the polytropic gas dynamics," *Lett. Math. Phys.*, vol. 60, no. 3, pp. 197–209, 2002.

[26] E. W. Weisstein, "Lucas Polynomial, MathWorld: A Wolfram Web Resource," http://mathworld.wolfram.com/LucasPolynomial.html.

[27] F.A. Oliveira, "Collocation and residual correction," *Numer. Math.*, vol. 36, pp. 27–31, 1980.

[28] İ. Çelik, "Collocation method and residual correction using Chebyshev series," *Appl. Math. Comput.*, vol. 174, pp. 910–920, 2006.

[29] M.A. Ramadan, K.R. Raslan, T.S.E. Danaf, and M.A.A.E. Salam, "An exponential Chebyshev second kind approximation for solving high-order ordinary differential equations in unbounded domains, with application to Dawson's integral," *J. Egyptian Math. Soc.*, vol. 25, pp. 197–205, 2017.

[30] J. Zhao, Y. Cao, and Y. Xu, "Sinc numerical solution for pantograph Volterra delay-integro-differential equation," *Int. J. Comput. Math.*, vol. 94, no. 5, pp. 853–865, 2017.

[31] J.G. Dix, "Asymptotic behavior of solutions to a first-order differential equation with variable delays," *Comput. Math. Appl.*, vol. 50, pp. 1791–1800, 2005.