New Trends in Mathematical Sciences
http://dx.doi.org/10.20852/ntmsci.2021.450

Euler polynomials method for solving linear integro differential equations

Deniz Elmacı[1] *and Nurcan Baykuş Savaşaneril*[2]

[1]Dokuz Eylul University, Bergama Vocational School, Izmir, Turkey.
[2]Dokuz Eylul University, Izmir Vocational School, Izmir, Turkey.

Received: 7 May 2021, Accepted: 28 June 2021
Published online: 12 September 2021.

Abstract: In this study, a matrix method called the Euler collocation method is presented for numerically solving linear integro-differential equations. Using the collocation points, this method transforms the integro-differential equation into a matrix equation, which corresponds to a system of linear algebraic equations with unknown Euler coefficients. To illustrate the method, it is applied to certain linear Fredholm, Volterra, and Fredholm-Volterra integro-differential equations and the results are compared other numerical methods. In addition, the convergence of the solutions to the problems are examined. To obtain the matrix equations and solutions for the selected problems, code has developed in MATLAB.

Keywords: Euler polynomials, Collocation points, Residual error analysis, the matrix method.

1 Introduction

Integro - differential equations (IDEs) are usually used in the modeling of physical phenomena and play an important role in the fields of science and engineering [1-6]. These equations are difficult to solve and so, numerical methods are required such as Taylor collocation, Chebyshev collocation, Chebyshev-Lobatto collocation, Lagrange collocation, Jacobi collocation, Laguerre collocation and Bessel polynomial methods etc [7-23]. In this study, we consider the m-th order linear Volterra-Fredholm integro-differential equation in the form

$$\sum_{k=0}^{m} P_k(t)y^{(k)}(t) = g(t) + \lambda_1 \int_a^b K_f(t,s)\,y(s)\,ds + \lambda_2 \int_a^t K_v(t,s)\,y(s)\,ds \tag{1}$$

with condition

$$\sum_{k=0}^{m-1} \left(a_{jk}y^{(k)}(a) + b_{jk}y^{(k)}(b)\right) = \mu_j, \quad j = 0,1,\ldots,m-1 \tag{2}$$

where $P_k(t)$, $g(t)$, $K_f(t,s)$ and $K_v(t,s)$ are functions defined on the interval $a \le t \le b$; a_{jk}, b_{jk} and μ_j are appropriate constants; $y(t)$ is an unknown solution function to be determined. If $\lambda_2 = 0$, Eq.(1) becomes the Fredholm integro differential equation. If $\lambda_1 = 0$, it becomes the Volterra integro differential equation.
Our aim is to find an approximate solution of Eq.(1) expressed in the truncated Euler series form

$$y(t) \cong y_N(t) = \sum_{n=0}^{N} a_n \mathrm{E}_n(t), \qquad a \leqslant t \leqslant b \tag{3}$$

* Corresponding author e-mail: nurcan.savasaneril@deu.edu.tr

where $E_n(t)$ indicates the Euler-Taylor polynomials which are described as

$$\frac{2e^{xt}}{e^t+1}=\sum_{n=0}^{\infty}E_n(x)\frac{t^n}{n!},\quad |t|<\pi \tag{4}$$

Euler polynomials are strictly connected with Bernoulli ones, and are used in the Taylor expansion in a neighborhood of the origin of trigonometric and hyperbolic secant functions. Recursive computation of Euler polynomials can be obtained by using the following formula [24];

$$E_n(t)+\sum_{k=0}^{n}\binom{n}{k}E_k(t)=2t^n,\quad n=1,2,\ldots \tag{5}$$

Also, Euler polynomials $E_n(t)$ can be defined as polynomials of degree $n\geq 0$ satisfying the conditions

$$E'_m(t)=mE_{m-1}(t),\quad m\geqslant 1. \tag{6}$$

By using Eq.(4), Eq.(5) or Eq.(6), the first Euler polynomials are described as

$$E_0(t)=1,\quad E_1(t)=t-\frac{1}{2},\quad E_2(t)=t^2-t,\quad E_3(t)=t^3-\frac{3}{2}t^2+\frac{1}{4},$$
$$E_4(t)=t^4-2t^3+t,\quad E_5(t)=t^5-\frac{5}{2}t^4+\frac{5}{2}t^2-\frac{1}{2},$$
$$E_6(t)=t^6-3t^5+5t^3-3t,\quad E_7(t)=t^7-\frac{7}{2}t^6+\frac{35}{4}t^4-\frac{21}{2}t^2+\frac{17}{8},$$

2 Materials and Methods

2.1 Matrix Relations for Euler Polynomials

The linear Volterra-Fredholm integro-differential equation in Eq.(1) is considered to create the matrices of each term. The desired solution y(t) defined by the truncated Euler series Eq.(3) of Eq.(1) is modified to extract the matrix form, for $n=0,1,2,\ldots,N$ as

$$y(t)\cong y_N(t)=\mathbf{E}(t)\mathbf{A} \tag{7}$$

where

$$\mathbf{E}(t)=[E_0(t)\ E_1(t)\ \cdots\ E_N(t)],\quad \mathbf{A}=[a_0\ a_1\ \cdots\ a_N]^T$$

On the other hand, using Euler polynomials and Taylor expansion, and by means of Eq.(5), the matrix relation between standard base matrix and Euler base matrix is constructed as

$$\mathbf{T}^T(t)=\left(\mathbf{S}^{-1}\right)^T\mathbf{E}^T(t)\Leftrightarrow\mathbf{T}(t)=\mathbf{E}(t)\left(\mathbf{S}^{-1}\right)\Rightarrow\mathbf{E}(t)=\mathbf{T}(t)\mathbf{S} \tag{8}$$

where

$$\mathbf{T}(t)=\left[1\ t\ \cdots\ t^N\right],$$

$$\left(\mathbf{S}^{-1}\right)^{T}=\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{1}{2}\binom{1}{0} & 1 & 0 & \cdots & 0 \\ \frac{1}{2}\binom{2}{0} & \frac{1}{2}\binom{2}{1} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}\binom{N}{0} & \frac{1}{2}\binom{N}{1} & \frac{1}{2}\binom{N}{2} & \cdots & 1 \end{bmatrix}_{(N+1)\times(N+1)}$$

The relation between the matrix $\mathbf{E}(t)$ and its derivatives is

$$\mathbf{E}'(t)=\mathbf{T}'(t)\mathbf{S}=\mathbf{T}(t)\mathbf{B}\mathbf{S}$$

where

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

$$y^{(k)}(t)=\mathbf{E}^{(k)}(t)\mathbf{A}=\mathbf{T}(t)\mathbf{B}^{k}\mathbf{S}\mathbf{A},\ k=0,1,2,\ldots. \tag{9}$$

Besides, the matrix form of the kernel function $K_f(t,s)$ and $K_v(t,s)$ in Eq.(1) is computed as follows

$$\begin{aligned} K_f(t,s)&=\mathbf{T}(t)\mathbf{K}_f\mathbf{T}(s)^{T} \\ K_v(t,s)&=\mathbf{T}(t)\mathbf{K}_v\mathbf{T}(s)^{T} \end{aligned} \tag{10}$$

where $\mathbf{K}_f=\mathbf{K}_v=\mathrm{K}=[k_{mn}], \quad m,n=0,1,\ldots,N$

$$k_{mn}=\frac{1}{m!n!}.\frac{\partial^{m+n}\mathrm{K}(0,0)}{\partial t^{m}\partial s^{n}}$$

$$\begin{aligned} \int_a^b \mathrm{K}_f(t,s)\,y(s)\,ds&=\mathbf{T}(t)\mathbf{K}_f\mathbf{Q}_f\mathbf{S}\mathbf{A} \\ \int_a^t \mathrm{K}_v(t,s)\,y(s)\,ds&=\mathbf{T}(t)\mathbf{K}_v\mathbf{Q}_v(t)\mathbf{S}\mathbf{A} \end{aligned} \tag{11}$$

where

$$\mathbf{Q}_f=[q^{f}{}_{mn}]=\int_a^b \mathbf{T}^{T}(s)\mathbf{T}(s)ds,$$

$$\mathbf{Q}_v(t)=[q^{v}{}_{mn}(t)]=\int_a^t \mathbf{T}^{T}(s)\mathbf{T}(s)ds,$$

$$\left.\begin{aligned} q^{f}{}_{mn} &= \frac{b^{m+n+1}-a^{m+n+1}}{m+n+1} \\ q^{v}{}_{mn}(t) &= \frac{t^{m+n+1}-a^{m+n+1}}{m+n+1} \end{aligned}\right\} \quad m,n=0,1,\ldots,N$$

By substituting the matrix relations Eq. (9) and Eq. (11) into Eq.(1) and then by using the collocation points

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

The system of matrix equations are obtained as follows:

$$\sum_{k=0}^{m} P_k(t_i)\,\mathbf{T}(t_i)\,\mathbf{B}^k\mathbf{SA} = g(t_i) + \lambda_1\mathbf{T}(t_i)\,\mathbf{K}_f\mathbf{Q}_f\mathbf{SA} + \lambda_2\mathbf{T}(t_i)\,\mathbf{K}_v\mathbf{Q}_v(t_i)\,\mathbf{SA}$$

$$\sum_{k=0}^{m} \mathbf{P}_k\mathbf{T}\mathbf{B}^k\mathbf{SA} = \mathbf{G} + \mathbf{T}\mathbf{K}_f\mathbf{Q}_f\mathbf{SA} + \overline{\mathbf{T}}\overline{\mathbf{K}}_v\overline{\mathbf{Q}}_v\mathbf{SA} \tag{12}$$

where

$$\mathbf{P}_k = \begin{bmatrix} P_k(t_0) & 0 & \cdots & 0 \\ 0 & P_k(t_1) & 0 & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & P_k(t_N) \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{T}(t_0) \\ \mathbf{T}(t_1) \\ \vdots \\ \mathbf{T}(t_N) \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} g(t_0) \\ g(t_1) \\ \vdots \\ g(t_N) \end{bmatrix}, \quad \overline{\mathbf{T}} = \begin{bmatrix} \mathbf{T}(t_0) & 0 & \cdots & 0 \\ 0 & \mathbf{T}(t_1) & 0 & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{T}(t_N) \end{bmatrix},$$

$$\overline{\mathbf{K}} = \begin{bmatrix} \mathbf{K} & 0 & \cdots & 0 \\ 0 & \mathbf{K} & 0 & \vdots \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{K} \end{bmatrix}, \quad \overline{\mathbf{Q}}(t) = \begin{bmatrix} \mathbf{Q}(t_0) \\ \mathbf{Q}(t_1) \\ \vdots \\ \mathbf{Q}(t_N) \end{bmatrix},$$

or briefly

$$\mathbf{WA} = \mathbf{G} \quad \Leftrightarrow \quad [\mathbf{W}:\mathbf{G}] \tag{13}$$

where

$$\mathbf{W} = \sum_{k=0}^{m} \mathbf{P}_k\mathbf{T}\mathbf{B}^k\mathbf{S} - (\mathbf{T}\mathbf{K}_f\mathbf{Q}_f + \overline{\mathbf{T}}\overline{\mathbf{K}}_v\overline{\mathbf{Q}}_v)\mathbf{S}$$

Besides, we can find for the condition Eq.(2), by using the relation Eq.(9),

$$\sum_{k=0}^{m-1}\left(a_{jk}\mathbf{T}(a)\,\mathbf{B}^k\mathbf{S} + b_{jk}\mathbf{T}(b)\,\mathbf{B}^k\mathbf{S}\right)\mathbf{A} = [\mu_j] \quad or \quad \left[\mathbf{U}:\mathbf{A}\right] \tag{14}$$

where

$$\mathbf{U}=\begin{bmatrix} u_{00} & u_{01} \cdots & u_{0N} \\ u_{10} & u_{11} \cdots & u_{1N} \\ \vdots & \cdots \ddots & \vdots \\ u_{(r-1)0} & \cdots \cdots & u_{(r-1)N} \end{bmatrix}$$

Consequently, any one row of Eq.(13) by the row matrix Eq.(14) is replaced, hence the desired augmented matrix or the resulted matrix equation comes out as

$$\widetilde{\mathbf{W}}\,\mathbf{A}=\widetilde{\mathbf{G}}\ \Rightarrow\ \left[\widetilde{\mathbf{W}}\,:\widetilde{\mathbf{G}}\right] \tag{15}$$

which suits to the system of linear algebraic equations with the Euler coefficients a_n. The solution of this system provides the matrix $\mathbf{A}$ and the solution of Eq.(1) - Eq.(2) is

$$y_N(t)=\mathbf{E}(t)\mathbf{A}=\mathbf{T}(t)\mathbf{SA}$$

2.2 Residual Error Analysis

We define the residual function using the linear part of Eq.(1) for the present method as [25-27]

$$R_N(t)=L[y_N(t)]-g(t) \tag{16}$$

where

$$L[y_N(t)]=\sum_{k=0}^{m}P_k(t)y^{(k)}(t)-\lambda_1\int_a^b K_f(t,s)\,y(s)\,ds-\lambda_2\int_a^t K_v(t,s)\,y(s)\,ds$$

3 Numerical Examples

In this section, some numerical examples of the problem Eq.(1) are given to illustrate the accuracy and effectiveness properties of the method.

Example 3.1. Let us first consider the first order Fredholm integro differential

$$y'(t)=y(t)+\frac{1-e^{t+1}}{t+1}+\int_0^1 e^{ts}y(s)\,ds,\ 0\leqslant t,s\leqslant 1$$

with the initial condition y(0) = 1 [28,29].

$$P_0(t)=-1,\ P_1(t)=1,\ g(t)=\frac{1-e^{t+1}}{t+1},\ \lambda_1=1,\ \lambda_2=0,\ K_f(t,s)=e^{ts}$$

Following the procedure in Section2, we find the solution of our problem for different values of N as follows:

$$y_2(t)=0.60906t^2+0.96948t+1$$

$$y_4(t) = 0.05947t^4 + 0.15469t^3 + 0.50165t^2 + 0.99900t + 1$$

$$y_6(t) = 0.00207t^6 + 0.00752t^5 + 0.04213t^4 + 0.16653t^3 + 0.5t^2 + 0.99999t + 1$$

$$y_8(t) = 0.00004t^8 + 0.00018t^7 + 0.00141t^6 + 0,.00832t^5 + 0.04167t^4 + 0.16667t^3 + 0.5t^2 + t + 1$$

The numerical results of this example are tabulated for $N = 2,4,6,8$ in Table 1 and in Table 2. And also, the results obtained by our method are compared with the results of Taylor polynomials given in [28], Hybrid Fourier (HF) and block-pulse functions and Fourier functions given in [29]. As can be seen from the tables, the result obtained by the present method is almost the same as the results of the exact solution. The present method is also effective and convenient. The absolute errors in the numerical solution of Example 3.1 are seen in Fig. 1. The error decreases when the integer N is increased.

Table 1: Comparisons of absolute errors for N= 2, 4, 6 in Example 3.1.

t	**Present method** $\lvert e_2(t)\rvert$	**Taylor polynomials** $\lvert e_2(t)\rvert$	**Present method** $\lvert e_4(t)\rvert$	**Taylor polynomials** $\lvert e_4(t)\rvert$	**Present method** $\lvert e_6(t)\rvert$	**Taylor polynomials** $\lvert e_6(t)\rvert$
0.05	0.00128	0.01261	4.72e-05	0.01261	5.74e-07	0.00012
0.15	0.00271	0.04148	0.00014	0.04148	1.88e-06	0.00041
0.25	0.00359	0.07615	0.00027	0.07615	3.48e-06	0.00073
0.35	0.00514	0.11782	0.00044	0.11782	5.36e-06	0.00111
0.45	0.00871	0.16784	0.00064	0.16784	7.54e-06	0.00155
0.55	0.01580	0.22772	0.00088	0.22772	1.01e-05	0.00206
0.65	0.02805	0.29912	0.00115	0.29912	1.33e-05	0.00265
0.75	0.04730	0.38384	0.00150	0.38384	1.73e-05	0.00331
0.85	0.07555	0.48391	0.00201	0.48391	2.28e-05	0.00406
0.95	0.11503	0.60155	0.00285	0.60155	3.15e-05	0.00490
CPU time	1.020 s		1.009 s		1.016 s	

Example 3.2. Let us consider the first order Volterra integro differential

$$y'(t) + y(t) = \int_0^t e^{s-t}\, y(s)\, ds, \quad 0 \leqslant t, s \leqslant 1$$

with the initial condition $y(0) = 1$ [30,31].

Following the procedure in Section2, we find the solution of our problem for different values of N as follows:

$$y_3(t) = -0.36788t^3 + 0.91590t^2 - t + 1$$

$$y_7(t) = -0.00555t^7 + 0.03599t^6 - 0.12774t^5 + 0.33124t^4 - 0.66625t^3 + 0.99996t^2 - t + 1$$

Table 2: Comparisons of absolute errors for N= 8 in Example 3.1.

t	**Exact solution** $y(t) = e^t$	**HF and block-pulse functions**	**Fourier functions**	**Taylor polynomials** $\lvert e_8(t)\rvert$	**Present method** $\lvert e_8(t)\rvert$
0.05	1.05127	8.90362e-06	0.17111	2.20047e-05	4.35243e-09
0.15	1.16183	5.75727e-06	0.07984	3.53323e-05	1.43054e-08
0.25	1.28403	2.45833e-05	0.06642	6.35704e-05	2.61028e-08
0.35	1.41907	1.24514e-05	0.06212	1.01215e-04	3.99067e-08
0.45	1.56831	2.18549e-06	0.06039	1.54963e-04	5.60546e-08
0.55	1.73325	6.98213e-06	0.05985	2.31844e-04	7.49678e-08
0.65	1.91554	9.17099e-06	0.06047	3.39093e-04	9.75358e-08
0.75	2.11700	2.99834e-05	0.06329	4.83948e-04	1.26045e-07
0.85	2.33965	1.31481e-05	0.07424	6.73161e-04	1.66217e-07
0.95	2.58571	1.03407e-05	0.16035	9.12919e-04	2.33085e-07
CPU time					1.075 s

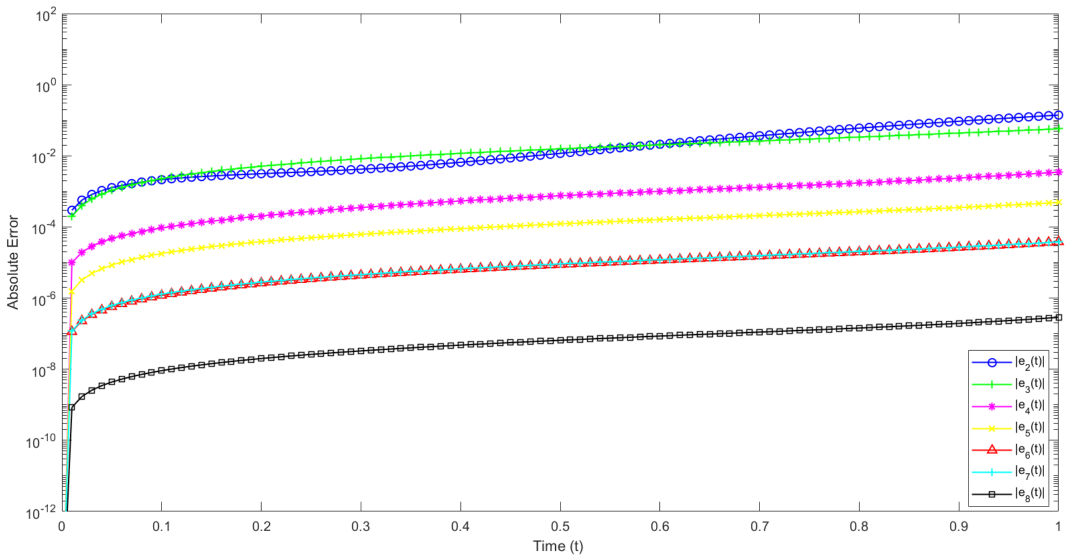

Fig. 1: The absolute errors of Example 3.1 for $2 \leq N \leq 8$.

The numerical results of this example are tabulated for $N = 3,7$ in Table 3. In addition, the results obtained by our method are compared with the results of Standard collocation (SCM), Chebyshev Gauss Lobatto collocation (CGLCM) methods given in [30], Finite difference method (FDM) and Homotopi perturbasyon method (HPM) given in [31]. As can be seen from the tables, the result obtained by the present method is almost the same as the results of the exact solution. The present method is also effective and convenient. The absolute errors in the numerical solution of Example 3.2 are seen in Fig. 2. The error decreases when the integer N is increased.

The first five values of the residual error function are shown in Fig. 3. It can be concluded that, the residual error decreases

Table 3: Comparisons of absolute errors for $N = 3,7$ in Example 3.2.

t	Exact solution $y(t) = e^{-t}\cosh t$	Present method $\lvert e_3(t)\rvert$	Present method $\lvert e_7(t)\rvert$	SCM $\lvert e_7(t)\rvert$	CGLCM $\lvert e_7(t)\rvert$	FDM $\lvert e_{12}(t)\rvert$	HPM $\lvert e_{10}(t)\rvert$
0.0000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.0833	0.923269	0.000426	8.414e-08	1.846e-05	1.793e-05	0.017692	2.822e-05
0.1667	0.858242	0.001194	9.981e-08	3.370e-05	9.948e-05	0.002145	2.388e-05
0.3333	0.756726	0.001901	6.849e-08	9.368e-06	2.378e-05	0.004541	1.711e-05
0.4167	0.717285	0.001567	7.543e-08	2.205e-05	6.132e-06	0.020633	1.448e-05
0.5000	0.683940	0.000950	6.699e-08	7.323e-06	8.835e-05	0.007136	3.143e-10
0.5833	0.655712	0.000397	6.591e-08	3.581e-06	0.000142	0.011048	1.038e-05
0.6667	0.631790	0.000400	1.124e-07	1.494e-05	0.000110	0.008200	8.786e-06
0.7500	0.611565	0.001570	2.042e-07	5.648e-06	3.471e-05	0.003413	1.326e-09
0.8333	0.594444	0.004621	3.793e-07	8.925e-07	8.169e-05	0.008170	6.295e-06
0.9167	0.579935	0.010359	1.205e-06	1.056e-05	0.000137	0.002889	5.334e-06
1.0000	0.567668	0.019646	5.014e-06	4.798e-06	0.000162	0.003272	9.482e-09
CPU time		0.984 s	1.021 s				

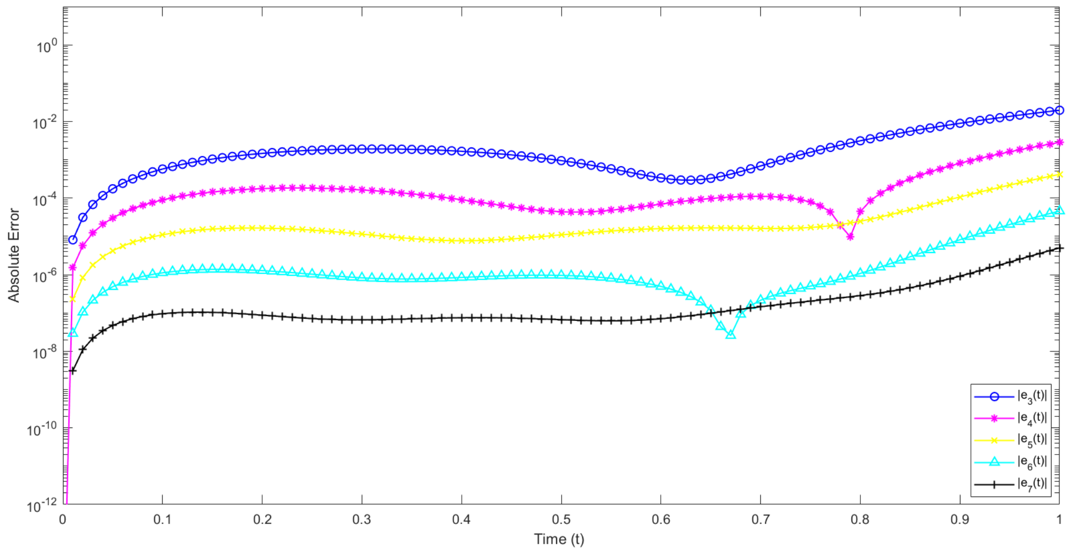

Fig. 2: Absolute errors of Example 3.2 for $3 \leq N \leq 7$.

as N values are increasing.

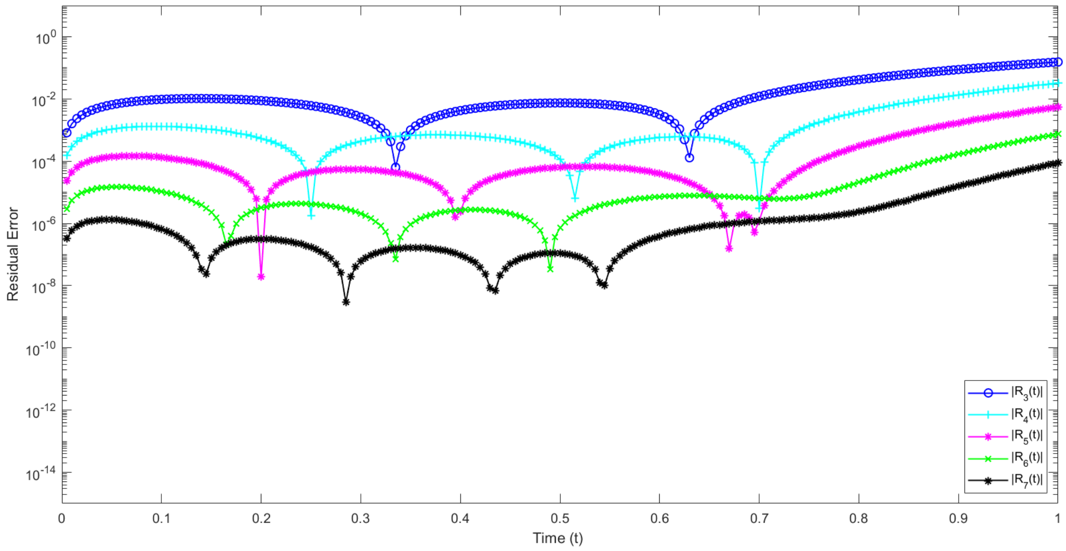

Fig. 3: The residual errors of Example 3.2 for $3 \leq N \leq 7$.

Example 3.3. Let us consider the fourth order Volterra integro differential

$$y^{(4)}(t) = t(1+e^t) + 3e^t + y(t) - \int_0^t y(s)\,ds \qquad 0 \leqslant t,s \leqslant 1$$

with the conditions $y(0)=1, \quad y(1)=1+e, \quad y''(0)=2, \quad y''(1)=3e$ [32].
Solutions of our problem for different values of N as follows:

$$y_5(t) = 0.04685t^5 + 0.16667t^4 + 0.53630t^3 + t^2 + 0.96847t + 1$$

$$y_8(t) = 0.00026t^8 + 0.00133t^7 + 0.00836t^6 + 0.04166t^5 + 0.16667t^4 + 0.50004t^3 + t^2 + 0.99996t + 1$$

$$y_{10}(t) = 3.843*10^{-6}t^{10} + 2.321*10^{-5}t^9 + 0.00020t^8 + 0.00139t^7 + 0.00833t^6 + 0.04167t^5 + 0.16667t^4 + 0.5t^3 + t^2 + t + 1$$

The numerical results of this example are tabulated for $N = 5, 8, 10$ in Table 4. In addition, the results obtained by our method are compared with the results of modified Homotopi perturbasyon method (MHPM) given in [32]. As can be seen from the tables, the result obtained by the present method is almost the same as the results of the exact solution. The present method is also very effective and convenient. The absolute errors in the numerical solution of Example 3.3 are seen in Fig. 4. The error decreases when the integer N is increased.

Table 4: Comparisons of absolute errors for $N = 5, 8, 10$ in Example 3.3.

t	Exact solution $y(t) = 1 + te^t$	Present method $\|e_5(t)\|$	MHPM $\|e_5(t)\|$	Present method $\|e_8(t)\|$	Present method $\|e_{10}(t)\|$
0	1.0000000000	0.000000	2.000e-09	0.000000	0.000000
0.1	1.1105170918	0.003117	0.016954	4.122e-06	1.741e-08
0.2	1.2442805516	0.006015	0.032409	7.983e-06	3.383e-08
0.3	1.4049576423	0.008474	0.045011	1.132e-05	4.797e-08
0.4	1.5967298791	0.010274	0.053608	1.388e-05	5.896e-08
0.5	1.8243606354	0.011210	0.057319	1.540e-05	6.555e-08
0.6	2.0932712802	0.011108	0.055595	1.563e-05	6.683e-08
0.7	2.4096268952	0.009860	0.048307	1.430e-05	6.153e-08
0.8	2.7804327428	0.007457	0.035833	1.122e-05	4.889e-08
0.9	3.2136428000	0.004047	0.019160	6.307e-06	2.794e-08
CPU time		1.049 s		1.052 s	1.062 s

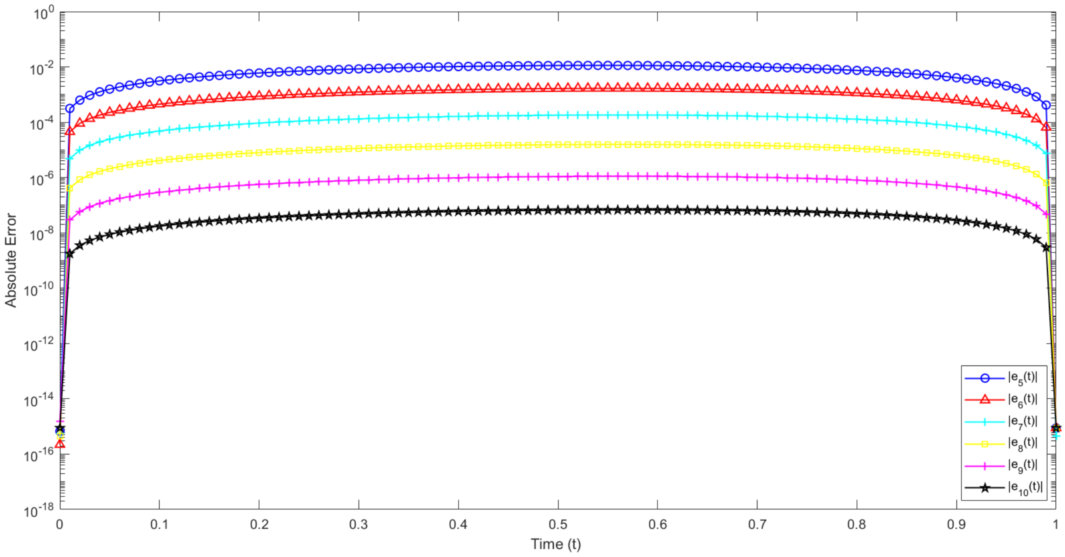

Fig. 4: Absolute errors of Example 3.3 for $5 \leq N \leq 10$.

Example 3.4. Our last example second order Volterra-Fredholm integro differential

$$ty''(t) - e^t y'(t) + y(t) = -t\cos t - \frac{1}{2}\sin^2 t - \frac{1}{2}\int_0^{\pi} s\cos t y(s)\,ds + \int_0^t (e^t + \sin s) y(s)\,ds$$

with the conditions $y(0)=1, \quad y'\left(\frac{\pi}{2}\right)=-1$ [33].
Solutions of our problem for different values of N as follows:

$$y_3(t)=0.17110t^3-0.80992t^2+0.27795t+1$$

$$y_6(t)=-0.00069t^6-0.00259t^5+0.04578t^4-0.00145t^3-0.50533t^2+0.00683t+1$$

$$y_{10}(t)=-2.194*10^{-7}t^{10}-5.567*10^{-8}t^9+2.385*10^{-5}t^8+4.344*10^{-6}t^7-0.00140t^6+9.567*10^{-6}t^5$$
$$+0.04166t^4+5.075*10^{-6}t^3-0.5t^2+0.00001t+1$$

The numerical results of this example are tabulated for $N=3,6,10$ in Table 5. And also, the results obtained by our method are compared with the results of Morgan Voyce collocation method given in [33]. As can be seen from the tables, the result obtained by the present method is almost the same as the results of the exact solution. The present method is also very effective and convenient. The residual errors in the numerical solution of Example 3.4 are seen in Fig. 5. The error decreases when the integer N is increased.

Table 5: Comparisons of absolute errors for N= 3, 6, 10 in Example 3.4.

t	**Exact Solution** $y(t)=\cos t$	**Present method** $\lvert e_3(t)\rvert$	**Morgan Voyce** $\lvert e_3(t)\rvert$	**Present method** $\lvert e_6(t)\rvert$	**Morgan Voyce** $\lvert e_6(t)\rvert$	**Present method** $\lvert e_{10}(t)\rvert$	**Morgan Voyce** $\lvert e_{10}(t)\rvert$
0	1.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000	0.0000000000
0.2	0.9800665778	0.0444950629	0.0444950629	0.0011476339	0.0011476339	0.0000024196	0.0000024197
0.4	0.9210609940	0.0714813445	0.0714813446	0.0018690395	0.0018690395	0.0000039357	0.0000039357
0.6	0.8253356149	0.0868190967	0.0868190967	0.0022308663	0.0022308663	0.0000046993	0.0000046993
0.8	0.6967067093	0.0949046687	0.0949046687	0.0023262859	0.0023262859	0.0000048725	0.0000048725
1	0.5403023059	0.0988226498	0.0988226498	0.0022485875	0.0022485875	0.0000046275	0.0000046275
1.2	0.3623577545	0.1005503085	0.1005503084	0.0020854119	0.0020854119	0.0000041525	0.0000041525
1.4	0.1699671429	0.1012061748	0.1012061747	0.0019254715	0.0019254716	0.0000036747	0.0000036747
1.6	-0.0291995223	0.1013328604	0.1013328603	0.0018678491	0.0018678492	0.0000034957	0.0000034957
1.8	-0.2272020947	0.1012028370	0.1012028367	0.0020225975	0.0020225976	0.0000040304	0.0000040304
2	-0.4161468365	0.1011349847	0.1011349843	0.0024904512	0.0024904513	0.0000058315	0.0000058315
2.2	-0.5885011173	0.1018092913	0.1018092908	0.0033090306	0.0033090307	0.0000095145	0.0000095145
2.4	-0.7373937155	0.1045671532	0.1045671526	0.0043529901	0.0043529903	0.0000153163	0.0000153164
2.6	-0.8568887534	0.1116853109	0.1116853101	0.0051761429	0.0051761432	0.0000216347	0.0000216347
2.8	-0.9422223407	0.1266124921	0.1266124911	0.0047846353	0.0047846356	0.0000211950	0.0000211950
3	-0.9899924966	0.1541593341	0.1541593329	0.0013317420	0.0013317425	0.0000076252	0.0000076251
π	-1.0000000000	0.1846715213	0.1846715219	0.0045871731	0.0045871576	0.0000723499	0.0000723486
CPU time		1.027 s		1.052 s		1.104 s	

It can be concluded that, the residual error decreases as N values are increasing.

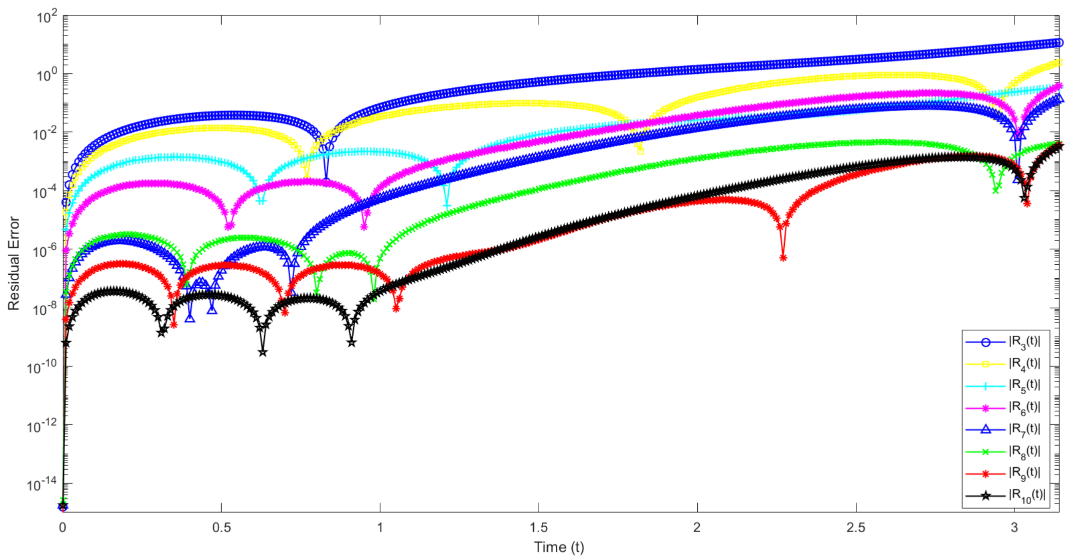

Fig. 5: The residual errors of Example 3.4 for $3 \leq N \leq 10$.

4 Conclusion

In this research, a collocation calculation model based on Euler polynomial for solutions of the linear integro-differential equations is presented. Furthermore, the control of the solutions is performed with the utilization of defined techniques. In addition, an error estimation is given with the residual error function. Comparison of the results obtained by the present method with that obtained by other methods reveals that the present method is very effective and convenient. Another advantage of the proposed technique is the utilization for testing reliability of the solutions of different problems. Tables and figures indicate that as N increases the errors decrease more rapidly; hence for better results, using large number N is recommended.

Acknowledgements

The authors would like to acknowledge to TUBITAK for their financial support with 2211/A National PhD Scholarship Program.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors have contributed to all parts of the article. All authors read and approved the final manuscript.

References

[1] Sezer M., *Taylor polynomial solutions of volterra integral equations*, Int. J. Math. Educ. Sci. Technol. 25, 625-633, doi:10.1080/0020739940250501, (1994).

[2] Volterra V., *Theory of Functionals and of Integral and Integro-Differential Equations*, Dover Publications, (2005).

[3] Brunner H., *Recent advances in the numerical analysis of volterra functional differential equations with variable delays*, Journal of Computational and Applied Mathematics, 228, 524-537, doi:10.1016/j.cam.2008.03.024, (2009).

[4] Yüzbaşı Ş., *Laguerre approach for solving pantograph type Volterra integro differential equations*, Applied Mathematics and Computation, 232, 1183?1199, doi:10.1016/j.amc.2014.01.075, (2014).

[5] Mirzaee, F., Bimes, L., Tohidi,E., *A numerical framework for solving high order pantograph delay Volterra integro-differential equations*, Kuwait J. Sci. 43(1), 69-83, (2016).

[6] Balcı M.A., Sezer M., *Hybrid Euler-Taylor matrix method for solving of generalized linear Fredholm integro-differential difference equations*, Applied Mathematics and Computation, vol. 273, 33?41, doi:10.1016/j.amc.2015.09.085, (2016).

[7] Yalçınbaş S., Sezer M., *A Taylor collocation method for the approximate solution of general linear Fredholm?Volterra integro-difference equations with mixed argument*, Applied Mathematics and Computation, vol. 175.1, 675?690, doi:10.1016/j.amc.2005.07.038, (2006).

[8] 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, vol. 44.4, 1065?1091, doi:10.3906/mat-2002-55, (2020).

[9] Yüzbaşı Ş., İsmailov N., *An operational matrix method for solving linear Fredholm?Volterra integro-differential equations*, Turkish Journal of Mathematics, vol. 42.1, 243?256, doi:10.3906/mat-1611-126, (2018).

[10] Yüzbaşı Ş., *Shifted Legendre method with residual error estimation for delay linear Fredholm integro-differential equations*, Journal of Taibah University for Science, vol. 11.2, 344?352, doi:10.1016/j.jtusci.2016.04.001, (2017).

[11] Yüzbaşı Ş., Karaçayır M., *A Galerkin-like approach to solve high-order integrodifferential equations with weakly singular kernel*, Kuwait Journal of Science, vol. 43.2, 106?120, (2016).

[12] Yüzbaşı Ş., *A collocation method based on Bernstein polynomials to solve nonlinear Fredholm?Volterra integro-differential equations*, Applied Mathematics and Computation, vol. 273, 142?154, doi:10.1016/j.amc.2015.09.091, (2017).

[13] Yüzbaşı Ş., Şahin N., Yıldırım A., *A collocation approach for solving high-order linear Fredholm?Volterra integro-differential equations*, Mathematical and Computer Modelling, vol. 55, 547?563, doi:10.1016/j.mcm.2011.08.032, (2012).

[14] Yüzbaşı Ş., *Improved Bessel collocation method for linear Volterra integro-differential equations with piecewise intervals and application of a Volterra population model*, Applied Mathematical Modelling, vol. 40, 5349-5363, doi:10.1016/j.apm.2015.12.029, (2016).

[15] Yüzbaşı Ş., Gök E., Sezer M., *Müntz-Legendre Matrix Method to solve Delay Fredholm Integro-Differential Equations with constant coefficients*, New Trends in Mathematical Sciences, vol. 3.2, 159-167, (2015).

[16] Yüzbaşı Ş., İsmailov N., *Solving systems of Volterra integral and integrodifferential equations with proportional delays by Differential Transformation Method*, Journal of Mathematics, Hindawi, doi:10.1155/2014/725648, (2014).

[17] Yalçınbaş S., Sezer M., *The approximate solution of high-order linear Volterra?Fredholm integro-differential equations in terms of Taylor polynomials*, Applied Mathematics and Computation, vol. 112.2-3, 291-308, doi:10.1016/S0096-3003(99)00059-4, (2000).

[18] Yüzbaşı Ş., *Numerical solutions of systems of linear Fredholm integro-differential equations with Bessel polynomial bases*, Computers and Mathematics with Applications, vol. 61, 3079-3096, doi:10.1016/j.camwa.2011.03.097, (2011).

[19] Gümgüm S., Baykuş Savaşaneril N., Kürkçü Ö.K., Sezer M., *A numerical technique based on Lucas polynomials together with standard and Chebyshev-Lobatto collocation points for solving functional integro-differential equations involving variable delays*, Sakarya University Journal of Science, vol. 22.6, 1659-1668 doi:10.16984/saufenbilder.384592, (2018).

[20] Karamete A., Sezer M., *A Taylor collocation method for the solution of linear integro-differential equations*, International Journal of Computer Mathematics, vol. 79.9, 987-1000, doi:10.1080/00207160212116, (2002).

[21] Oğuz C., Sezer M., *Chelyshkov collocation method for a class of mixed functional integro-differential equations*, Applied Mathematics and Computation, vol. 259, 943-954, doi:10.1016/j.amc.2015.03.024, (2015).

[22] Baykuş N., Sezer M., *Solution of high?order linear Fredholm integro?differential equations with piecewise intervals*, Numerical Methods for Partial Differential Equations, vol. 27.5, 1327-1339 doi:10.1002/num.20587 (2011).

[23] Baykuş Savaşaneril N., Sezer M., *Hybrid Taylor-Lucas Collocation Method for Numerical Solution of High-Order Pantograph Type Delay Differential Equations with Variables Delays*, Appl. Math. Inf. Sci. 11, No. 6, 1795-1801, doi:10.18576/amis/110627, (2017).

[24] Cheon G.S., *A note on the Bernoulli and Euler polynomials*, Applied Mathematics Letters, 16, 365-368, doi:10.1016/S0893-9659(03)80058-7, (2003).

[25] Kürkçü Ö.K., Aslan E., Sezer M., *A numerical method for solving some model problems arising in science and convergence analysis based on residual function*, Applied Numerical Mathematics, 121, 134?148, doi:10.1016/j.apnum.2017.06.015, (2017).

[26] Biçer K.E., Sezer M., *A computational method for solving differential equations with quadratic nonlinearity by using Bernoulli polynomials*, Thermal Science, 23-1, 275?283, doi:10.2298/TSCI181128041B, (2019).

[27] Tarakçı, M., Özel, M., Sezer, M., *Solution of nonlinear ordinary differential equations with quadratic and cubic terms by Morgan-Voyce matrix-collocation method*, Turkish Journal of Mathematics, 44 (3), 906-918, doi: 10.3906/mat-1908-102, (2020).

[28] Kurt, N., Sezer M., *Polynomial solution of high-order linear Fredholm integro-differential equations with constant coefficients*, Journal of the Franklin Institute 345, 839?850, doi.org/10.1016/j.jfranklin.2008.04.016, (2008).

[29] Asady, B., Kajani, M.T., *Direct method for solving integro differential equations using hybrid Fourier and block-pulse functions*, Int. J. Computer Math International Journal of Computer Mathematics. 82(7), doi.org/10.1080/00207160412331336044, 889?895, (2005).

[30] Agbolade, O. A., Anake, T. A., *Solutions of First-Order Volterra Type Linear Integrodifferential Equations by Collocation Method*, , Journal of Applied Mathematics, Hindawi, 1-5, doi.org/10.1155/2017/1510267, (2017).

[31] Raftari, B., *Numerical Solutions of the Linear Volterra Integro-differential Equations: Homotopy Perturbation Method and Finite Difference Method*, IWorld Applied Sciences Journal 9 (Special Issue of Applied Math), IDOSI Publications, (2010).

[32] Afrouzi, G.A., Ganji D. D., Hosseinzadeh H., Talarposhti R.A., *Fourth order Volterra integro-differential equations using modified homotopy-perturbation method*, The Journal of Mathematics and Computer Science, 3(2), 179-191, http://dx.doi.org/10.22436/jmcs. 03.02.10, (2011).

[33] Karaçayır, M., *Diferansiyel, fonksiyonel diferansiyel ve integro-diferansiyel denklemler ve sistemleri için Galerkin-tipi bir sayisal yöntem*, Phd thesis, Akdeniz University, Antalya, (2018).