On the application of Euler’s method to linear integro differential equations and comparison with existing methods

Deniz ELMACI¹; Nurcan BAYKUŞ SAVAŞANERİL², Fadime DAL³, Mehmet SEZER⁴

¹Bergama Vocational School, Dokuz Eylul University, Izmir, Turkey, https://orcid.org/0000-0002-9234-6361
²İzmir Vocational School, Dokuz Eylul University, Izmir, Turkey, https://orcid.org/0000-0002-3098-2936
³Department of Mathematics, Faculty of Science, Ege University, Izmir, Turkey, https://orcid.org/0000-0002-9058-223X
⁴Department of Mathematics, Faculty of Science, Celal Bayar University, Manisa, Turkey, https://orcid.org/0000-0002-7744-2574

Abstract: In this study, a collocation method using Euler method for solving systems of linear integro-differential equations is presented. The solution process is illustrated and various physically relevant results are obtained. Comparison of the obtained results with exact solutions and solutions obtained by other methods show that the proposed method is an effective and highly promising for linear integro-differential equation systems. All of numerical calculations have been made on a computer using a program written in Matlab.

Key words: Euler polynomials, Collocation points, Residual error analysis, Systems of integro differential equations.

1. Introduction

Since, the some problems in applied science cannot be expressed with a single equations; differential, integral, or with a set of integro differential equations consisted of their linear combination can be used for expressing of these problems [37]. These types of systems are encountered in many branches of physics and engineering. Differential equation systems have been attracted many interests in the field of Elasticity theory [10], Dynamics [18], Fluid mechanics [1], Circuit problems [43], Oscillation problems [14, 32], Quantum dynamics [12] etc. Although, the integral and integro-differential equations systems are used in many fields of science such as; Electromagnetic theory [7], Thermoelasticity [20], Biology [15], Mechanics [38], Wave refraction [8], there are no general method for solving these systems. Therefore, finding new solution systems is an important task for modeling and developing new applications in the field of physics and engineering. Since, it is usually difficult to find analytical solutions of IDE, a numerical approximation is required. In recent years, many numerical methods were utilized for the solution of IDES such as; Adomian decomposition method [5], the homotopy perturbation [39, 41] and the modified homotopy perturbation method [17], Bessel collocation method [42], the variational iteration method [34], the Tau method [33], Bernstein operational matrix approach [22], the Spectral method [3], the Sinc collocation method [16], the Legendre matrix method [35], the Lagrange method

*Correspondence: deniz.elmaci@deu.edu.tr

2010 AMS Mathematics Subject Classification: 11B68, 65L60, 65G50, 45J05

This work is licensed under a Creative Commons Attribution 4.0 International License.
[40], the differential transformation method [4], the Taylor collocation method [13], the Galerkin method [19], the Chebyshev polynomial method [2], the Fibonacci polynomial method [24]. Among them, Euler collocation method is a promising system that has many advantages such as it is simple to construct the main matrix equations and to increase the efficiency and easiness in the programming. In addition, this method has interesting features. One of them is finding the analytical solutions if the system has exact solutions that are polynomial functions. If the solution of the system is not polynomial function, the upper bound for error can be established by using this method. Another feature of the method is a shorter computation time and improvement of overall accuracy. According to the unique properties of the system many recent studies have used these systems for solving such kind of problems. Recently, Mirzaee et al. have studied the Euler matrix method for numerical solutions of the linear and a class of nonlinear FIDEs [25], the telegraph equations [26], the systems of linear Volterra integral equations [27], high-order pantograph delay Volterra IDEs [28], nonlinear Volterra type Fractional IDEs [29], systems of high-order linear differential–difference equations [30] and second-order linear hyperbolic partial differential equations [31].

In this study a new method based on the Euler polynomials has been suggested for obtaining the closest numerical solutions to exact solution with the aid of Euler collocation method for IDEs. Then, the developed method was utilized for the solution of some selected systems of IDEs which are particularly important in the field of wind ripple in the desert, nano-hydrodynamics, population growth model, glass-forming process, and oceanography. Numerical examples have been illustrated to demonstrate the efficiency and applicability of the method developed. The results of present method are compared with some existing solutions such as; Bessel collocation, Bernstein collocation, Sinc collocation, Adomian decomposition, Homotopy perturbation, Chebyshev wavelet, etc. and present method is found superior by means of accuracy.

In this study, system of linear Volterra - Fredholm integro-differential equation is shown as

\[
\sum_{k=0}^{m_1} \sum_{j=1}^{J} P_{ij}^k (t) y_j^{(k)} (t) = g_i (t) + \sum_{r=0}^{m_2} \sum_{j=1}^{J} \int_a^b K_{fij}^r (t, s) y_j^{(r)} (s) \, ds \quad + \sum_{l=0}^{m_3} \sum_{j=1}^{J} \int_a^t K_{vij}^l (t, s) y_j^{(l)} (s) \, ds \quad (1.1)
\]

\[
i = 1, 2, \ldots, J, \quad m = \max \{m_1, m_2, m_3\}, \quad a \leq t, s \leq b \text{ with the condition of}
\]

\[
\sum_{k=0}^{m-1} \left( a_{ik} y_j^{(k)} (a) + b_{ik} y_j^{(k)} (b) \right) = \mu_{ji}, \quad j = 1, \ldots, J, \quad i = 0, 1, \ldots, m - 1 \quad (1.2)
\]

where, \( P_{ij}^k (t) \) and \( g_i (t) \) are functions defined on the interval \( a \leq t \leq b \); \( a_{ik}, b_{ik} \) and \( \mu_{ji} \) are appropriate constants; \( y_j (t) \) is an unknown solution function to be determined. For this purpose, the Euler polynomials solution of the problem Equation (1.1) and Equation (1.2) in the finite series form is assumed

\[
y_j (t) \cong y_{jN} (t) = \sum_{n=0}^{N} a_{jn} E_n (t), \quad j = 1, 2, \ldots, J \quad (1.3)
\]

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

\[
\frac{2e^{xt}}{e^t + 1} = \sum_{n=0}^{\infty} E_n (x) \frac{t^n}{n!} \quad |t| < \pi \quad (1.4)
\]
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 [9];
\[ E_n(t) + \sum_{k=0}^{N} \binom{n}{k} E_k(t) = 2t^n, n = 1, 2, \ldots \] (1.5)

Also, Euler polynomials \( E_n(t) \) can be defined as polynomials of degree \( n \geq 0 \) satisfying the conditions
\[ E'_m(t) = mE_{m-1}(t), m \geq 1 \] (1.6)

By means of Equation (1.4), Equation (1.5) and Equation (1.6), the first Euler polynomials are described as
\[
\begin{align*}
E_0(t) &= 1, \\
E_1(t) &= t - \frac{1}{2}, \\
E_2(t) &= t^2 - t, \\
E_3(t) &= t^3 - \frac{3}{2}t^2 + \frac{1}{4}, \\
E_4(t) &= t^4 - 2t^3 + t, \\
E_5(t) &= t^5 - \frac{5}{2}t^4 + \frac{5}{4}t^2 - \frac{1}{4}, \\
E_6(t) &= t^6 - 3t^5 + 5t^3 - 3t, \\
E_7(t) &= t^7 - \frac{7}{2}t^6 + \frac{21}{4}t^4 - \frac{21}{8}t^2 + \frac{17}{16}.
\end{align*}
\]

2. Matrix Relations for Euler Polynomials

The system of Volterra-Fredholm integro-differential Equation (1.1) is considered to create the matrices of each
term. The desired solution \( y_j(t) \) defined by the truncated Euler series Equation (1.3) of Equation (1.1) is
modified to extract the matrix form, for \( n = 0, 1, 2, \ldots, N \) as
\[ y_j(t) = E(t)A_j, \quad j = 1, 2, \ldots, J \] (2.1)
where
\[
E(t) = \begin{bmatrix} E_0(t) & E_1(t) & \ldots & E_N(t) \end{bmatrix}
\]
\[
A_1 = \begin{bmatrix} a_{10} \\ a_{11} \\ \vdots \\ a_{1N} \end{bmatrix}, \quad A_2 = \begin{bmatrix} a_{20} \\ a_{21} \\ \vdots \\ a_{2N} \end{bmatrix}, \quad \ldots, \quad A_J = \begin{bmatrix} a_{J0} \\ a_{J1} \\ \vdots \\ a_{JN} \end{bmatrix}
\]
\[
T(t) = \begin{bmatrix} 1 & t & \ldots & t^N \end{bmatrix}
\]
\[
(S^{-1})^T = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ \frac{1}{2} \left( \begin{array}{c} 1 \\ 0 \end{array} \right) & 1 & 0 & \ldots & 0 \\ \frac{1}{2} \left( \begin{array}{c} 2 \\ 0 \end{array} \right) & \frac{1}{2} \left( \begin{array}{c} 2 \\ 1 \end{array} \right) & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2} \left( \begin{array}{c} N \\ 0 \end{array} \right) & \frac{1}{2} \left( \begin{array}{c} N \\ 1 \end{array} \right) & \frac{1}{2} \left( \begin{array}{c} N \\ 2 \end{array} \right) & \ldots & 1 \end{bmatrix}
\]
\[ T(t) = E(t)(S)^{-1} \Rightarrow E(t) = T(t)S \]  \hfill (2.2)

can be expressed as

\[ y_j(t) = T(t)SA_j \]

The relation between the matrix \( E(t) \) and its derivatives are

\[ E'(t) = T'(t)S = T(t)BS \]

where

\[
\begin{bmatrix}
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 2 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & N \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}_{(N+1) \times (N+1)}
\]

\[
\begin{bmatrix}
1 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}_{(N+1) \times (N+1)}
\]

\[ y_j^{(k)}(t) = E^k(t)A = T(t)B^kSA_j, \quad k = 0, 1, 2, \ldots \]  \hfill (2.3)

Besides, the matrix form of the kernel function \( K_{rij}^f(t, s) \) and \( K_{rij}^l(t, s) \) in Equation (1.1) is computed as follows

\[ K_{rij}^f(t, s) = \left[ k_{mn}^f \right], \quad m, n = 0, 1, \ldots, N \]

\[ k_{mn}^f = \frac{1}{m!n!} \frac{\partial^{m+n}K_{rij}^f(0,0)}{\partial t^m \partial s^n} \]

\[ K_{rij}^r(t, s) = T(t)K_{rij}^r T(s)^T \]  \hfill (2.4)

\[ \int_a^b K_{rij}^r(t, s) y_j^{(r)}(s) ds = T(t)K_{rij}^r Q_f B^r SA_j = F_{ij}^r(t)A_j \]  \hfill (2.5)

\[ F_{ij}^r(t) = T(t)K_{rij}^r Q_f B^r S \]

\[ Q_f = [q_{mn}] = \int_a^b T^T(s)T(s) ds \]

\[ m, n = 0, 1, \ldots, N \]

\[ q_{mn} = \frac{b^{m+n+1} - a^{m+n+1}}{m + n + 1} \]
ELMACI, D., BAYKUS SAVASANERIL, N., DAL, F. and SEZER, M./Turk J Math

\[ K^l_{viij}(t, s) = [k^m_{mn}], \quad m, n = 0, 1, \ldots, N \]

\[ k^m_{mn} = \frac{1}{m!n!} \frac{\partial^{m+n} K^l_{viij}(0, 0)}{\partial t^m \partial s^n} \]

\[ K^l_{viij}(t, s) = T(t) K^l_{viij} T(s)^T \] (2.6)

\[ \int_a^t K^l_{viij}(t, s) y^{(l)}(s) ds = T(t) K^l_{viij} Q_v(t) B^l S A_j = V^l_{ij}(t) A_j \] (2.7)

\[ V^l_{ij}(t) = T(t) K^l_{viij} Q_v(t) B^l S \]

\[ Q_v(t) = [q_{mn}(t)] = \int_a^t T^T(s) T(s) ds \]

\[ m, n = 0, 1, \ldots, N \quad q_{mn}(t) = \frac{t^{m+n+1} - q^{m+n+1}}{m + n + 1} \]

By substituting the matrix relations Equation (2.3), Equation (2.5) and Equation (2.7) into Equation (1.1):

\[ \sum_{k=0}^{m_1} \sum_{j=1}^{J} P^k_{ij}(t) T(t) B^k S A_j = g_i(t) + \sum_{r=0}^{m_2} \sum_{j=1}^{J} F^r_{ij}(t) A_j + \sum_{l=0}^{m_3} \sum_{j=1}^{J} V^l_{ij}(t) A_j \] (2.8)

\[ \sum_{k=0}^{m_1} P^k(t) T(t) B^k S A = G(t) + \sum_{r=0}^{m_2} F^r(t) A + \sum_{l=0}^{m_3} V^l(t) A \]

where

\[
P_k = \begin{bmatrix} P^k_{11} & P^k_{12} & \cdots & P^k_{1J} \\ P^k_{21} & P^k_{22} & \cdots & P^k_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ P^k_{J1} & P^k_{J2} & \cdots & P^k_{JJ} \end{bmatrix}, \quad T(t) = \begin{bmatrix} T(t) & 0 & \cdots & 0 \\ 0 & T(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & T(t) \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_J \end{bmatrix}
\]

\[
B^k = \begin{bmatrix} B^k & 0 & \cdots & 0 \\ 0 & B^k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & B^k \end{bmatrix}, \quad S = \begin{bmatrix} S & 0 & \cdots & 0 \\ 0 & S & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S \end{bmatrix}, \quad G(t) = \begin{bmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_J(t) \end{bmatrix}
\]
\[ F_r(t) = \begin{bmatrix} F_{11}^r(t) & F_{12}^r(t) & \cdots & F_{1J}^r(t) \\ F_{21}^r(t) & F_{22}^r(t) & \cdots & F_{2J}^r(t) \\ \vdots & \vdots & \ddots & \vdots \\ F_{J1}^r(t) & F_{J2}^r(t) & \cdots & F_{JJ}^r(t) \end{bmatrix}, \quad V_l(t) = \begin{bmatrix} V_{11}^l(t) & V_{12}^l(t) & \cdots & V_{1J}^l(t) \\ V_{21}^l(t) & V_{22}^l(t) & \cdots & V_{2J}^l(t) \\ \vdots & \vdots & \ddots & \vdots \\ V_{J1}^l(t) & V_{J2}^l(t) & \cdots & V_{JJ}^l(t) \end{bmatrix} \]

and then by using the collocation points

\[ t_u = a + \frac{b-a}{N} u, \quad u = 0, 1, \ldots, N \]

we obtain the system of matrix equations

\[ \sum_{k=0}^{m_1} P_k(t_u) T(t_u) B^T S A = G(t_u) + \sum_{r=0}^{m_2} F_r(t_u) A + \sum_{l=0}^{m_3} V_l(t_u) A \quad (2.9) \]

and their the fundamental matrix form

\[ \sum_{k=0}^{m_1} P_k T B^T S A = G + \sum_{r=0}^{m_2} F_r A + \sum_{l=0}^{m_3} V_l A \quad (2.10) \]

where

\[ [P_k] \in \mathbb{J}(N+1) \times J(N+1), \quad T \in \mathbb{J}(N+1) \times J(N+1), \quad B^T \in \mathbb{J}(N+1) \times J(N+1), \quad S \in \mathbb{J}(N+1) \times J(N+1), \quad A \in \mathbb{J}(N+1) \times 1, \]

\[ G \in \mathbb{J}(N+1) \times 1, \quad [F_r] \in \mathbb{J}(N+1) \times J(N+1), \quad [V_l] \in \mathbb{J}(N+1) \times J(N+1). \]

or briefly

\[ W A = G \quad \iff \quad [W : G] \quad (2.11) \]

where

\[ W = \sum_{k=0}^{m_1} P_k T B^T S - \sum_{r=0}^{m_2} F_r A - \sum_{l=0}^{m_3} V_l A \]

Besides, we can find for the condition Equation (1.2), by using the relation Equation (2.3),

\[ \sum_{k=0}^{m_1} \left( a_i^j T(a) B^k S + b_i^j T(b) B^k S \right) A = [\mu_{jk}] \]

or

\[ [U_k : \mu_k], \quad k = 0, 1, \ldots, m-1. \quad (2.12) \]

Consequently, any rows of Equation (2.11) by the rows matrix Equation (2.12) is replaced, hence the desired augmented matrix or the resulted matrix equation comes out as
If \( \text{rank } \tilde{W} = \text{rank } \begin{bmatrix} \tilde{W} & \tilde{G} \end{bmatrix} = J(N+1) \) then we can write \( A = \tilde{W}^{-1} \tilde{G} \).

By solving the system, the unknown Euler coefficients matrix is determined and the Euler polynomial solution is obtained.

\[
y_j(t) \equiv y_{jN}(t) = \sum_{n=0}^{N} a_{jn} E_n(t), \quad a \leq t \leq b, \quad i = 1, 2, \ldots, n
\]

However, when \( \text{det}(\tilde{W}) = 0 \), if \( \text{rank } \tilde{W} = \text{rank } \begin{bmatrix} \tilde{W} & \tilde{G} \end{bmatrix} < J(N+1) \), then we may find a particular solution. Otherwise if \( \text{rank } \tilde{W} \neq \text{rank } \begin{bmatrix} \tilde{W} & \tilde{G} \end{bmatrix} < J(N+1) \), then there is no solution.

3. Accuracy of Solution

The accuracy of the present method is tested. We define the residual function using the linear parts of Equation (1.1) for the present method as [6, 21, 36].

\[
R_{j,N}(t) = L[y_{j,N}(t)] - g_j(t)
\] (3.1)

where the linear part is

\[
L[y_{j,N}(t)] = \sum_{k=0}^{m_1} \sum_{j=1}^{J} P_{ij}^k(t) y_{j,N}^{(k)}(t) - \sum_{r=0}^{m_2} \sum_{j=1}^{J} \int_{a}^{b} K_{rij}^r(t,s) y_{j,N}^{(r)}(s) \, ds - \sum_{l=0}^{m_3} \sum_{j=1}^{J} \int_{a}^{b} K_{lij}^l(t,s) y_{j,N}^{(l)}(s) \, ds.
\]

By means of the residual function defined by \( R_{j,N}(t) \) and the mean value of the function \( |R_{j,N}(t)| \) on the interval \([a,b] \), the accuracy of the solution can be controlled and the error can be estimated [6, 36]. Thus, we can estimate the upper bound of the mean error \( \bar{R}_{j,N} \) as follows:

\[
\left| \int_{a}^{b} R_{j,N} (t) \, dt \right| \leq \int_{a}^{b} |R_{j,N} (t) | \, dt
\]

\[
\int_{a}^{b} |R_{j,N} (t) | \, dt = (b - a) |R_{j,N} (c)|, \quad a \leq c \leq b
\]

\[
\Rightarrow \int_{a}^{b} R_{j,N} (t) \, dt = (b - a) |R_{j,N} (c)|
\]

\[
\Rightarrow (b - a) |R_{j,N} (c)| \leq \int_{a}^{b} |R_{j,N} (t) | \, dt
\]

\[
|R_{j,N} (c)| \leq \frac{\int_{a}^{b} |R_{j,N} (t) | \, dt}{b - a} = \frac{\bar{R}_{j,N}}{b - a}
\]
Kürkçü and coworkers developed the convergence of Dickson polynomial solution of the nonlinear model problem using the residual function in Banach space [21]. We will use the convergence criterions for solutions system of Volterra-Fredholm integro differential equations with Euler polynomials.

4. Applications

In this section, four examples are solved with the present method, then the obtained results are compared with the exact solutions and solutions obtained by other methods. To do this, a special module on Matlab run on PC with 8 GB RAM and i7-8550M processor with 1.80 Ghz.

On the other hand, it is worth specifying that CPU running time corresponding to different N is separately evaluated via Timing module on Matlab, after quitting the local kernel of our module. Also, by making use of a computational order of convergence formula deployed in [11], the experimental order of timing complexity ($EOT_N$) versus N can be established in accordance with the main structure of the present method as

$$EOT_N = \log \left| \frac{\tau_{N-1}/\tau_N}{(N-1)/N} \right|,$$

where $\tau_N$ is an exact time operated by Timing module corresponding to N. Thereby, timing complexity of the present method is discussed in the numerical applications.

Example 1. [19] Consider the following system of linear Volterra-Fredholm integro differential equations

\[ y''_1(t) + ty'_2(t) - y'_1(t) + 2ty_2(t) = g_1(t) + \int_0^t ((ts^2)y'_1(s) + tsy''_1(s) - (ts^3)y'_2(s)) \, ds \]

\[ + \int_0^t ((t^2s^3)y''_1(s) - tsy'''_2(s) + (2t^3s)y'''_2(s)) \, ds \]

\[ y'''_2(t) + ty'_1(t) - ty_2(t) - y_1(t) = g_2(t) + \int_0^t ((ts^2)y'_1(s) + tsy''_1(s) - (2ts)y'''_2(s)) \, ds \]

\[ + \int_0^t ((3t^3s)y_1(s) - t^3s^2y''_2(s)) \, ds \]

\[ y_1(0) = -1, \quad y_1(2) = 1, \quad y'_1(0) = 1 \]

\[ y_2(0) = 2, \quad y_2(2) = 8, \quad y'_2(0) = -3 \]

where

\[ g_1(t) = 5 + \frac{218}{15}t + 3t^2 - 3t^3 + 8t^4 - 12t^5 - \frac{3}{2}t^6 \]

\[ g_2(t) = 13 + \frac{602}{15}t + 2t^2 - 7t^3 + \frac{3}{2}t^5 - \frac{5}{3}t^6 + \frac{9}{2}t^7 - \frac{3}{5}t^8 \]
with exact solutions $y_1(t) = t^3 - 2t^2 + t - 1$, $y_2(t) = 2t^3 - t^2 - 3t + 2$. To use the Euler matrix formulation illustrated above, we define the following for $N = 3$ in $0 \leq t, s \leq 2$:

$$P_0(t) = \begin{bmatrix} 0 & 2t \\ -1 & 0 \end{bmatrix}, \quad P_1(t) = \begin{bmatrix} -1 & 0 \\ 0 & -t \end{bmatrix}, \quad P_2(t) = \begin{bmatrix} 0 & t \\ t & 0 \end{bmatrix}, \quad P_3(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$P_0 = \begin{bmatrix} P_0(0) & 0 & 0 & 0 \\ 0 & P_0(2/3) & 0 & 0 \\ 0 & 0 & P_0(4/3) & 0 \\ 0 & 0 & 0 & P_0(2) \end{bmatrix}, \quad P_1 = \begin{bmatrix} P_1(0) & 0 & 0 & 0 \\ 0 & P_1(2/3) & 0 & 0 \\ 0 & 0 & P_1(4/3) & 0 \\ 0 & 0 & 0 & P_1(2) \end{bmatrix}$$

$$P_2 = \begin{bmatrix} P_2(0) & 0 & 0 & 0 \\ 0 & P_2(2/3) & 0 & 0 \\ 0 & 0 & P_2(4/3) & 0 \\ 0 & 0 & 0 & P_2(2) \end{bmatrix}, \quad P_3 = \begin{bmatrix} P_3(0) & 0 & 0 & 0 \\ 0 & P_3(2/3) & 0 & 0 \\ 0 & 0 & P_3(4/3) & 0 \\ 0 & 0 & 0 & P_3(2) \end{bmatrix}$$

$$K_{f0}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad K_{f1}(t) = \begin{bmatrix} ts^2 & -ts^2 \\ ts^2 & 0 \end{bmatrix}, \quad K_{f2}(t) = \begin{bmatrix} ts & 0 \\ t^2s & 0 \end{bmatrix}, \quad K_{f3}(t) = \begin{bmatrix} 0 & 0 \\ 0 & -2ts \end{bmatrix}$$

$$K_{e0}(t) = \begin{bmatrix} 0 & 0 \\ 3t^3s & 0 \end{bmatrix}, \quad K_{e1}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad K_{e2}(t) = \begin{bmatrix} 0 & -ts \\ 0 & -t^3s^2 \end{bmatrix}, \quad K_{e3}(t) = \begin{bmatrix} t^2s^3 & 2t^3s \\ 0 & 0 \end{bmatrix}$$
\[ B_1 = \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}, \quad B_2 = \begin{bmatrix} B^2 & 0 \\ 0 & B^2 \end{bmatrix}, \quad B_3 = \begin{bmatrix} B^3 & 0 \\ 0 & B^3 \end{bmatrix}, \quad Q_f = \begin{bmatrix} Q_f & 0 \\ 0 & Q_f \end{bmatrix} \]

\[ T = \begin{bmatrix} T(0) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & T(0) & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & T(\frac{2}{3}) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & T(\frac{2}{3}) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & T(\frac{4}{3}) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & T(2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & T(2) \end{bmatrix} \]

\[ G = \begin{bmatrix} g(0) \\ g(\frac{2}{3}) \\ g(\frac{4}{3}) \\ g(2) \end{bmatrix}, \quad g(0) = \begin{bmatrix} 5 \\ 13 \end{bmatrix}, \quad g(\frac{2}{3}) = \begin{bmatrix} \frac{9106}{607} \\ \frac{2526}{65} \end{bmatrix}, \quad g(\frac{4}{3}) = \begin{bmatrix} \frac{1489}{154} \\ \frac{12035}{154} \end{bmatrix}, \quad g(2) = \begin{bmatrix} \frac{4949}{15} \\ 409 \end{bmatrix} \]

The augmented matrix for this fundamental matrix equation is calculated as

\[ [W : G] = \begin{bmatrix} 0 & -1 & 1 & 6 & 0 & 0 & 0 & 0 & 0 & : & 5 \\ -1 & \frac{1}{2} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 6 & : & 13 \\ -16 & \frac{917}{45} & -\frac{943}{45} & -\frac{4013}{340} & \frac{4}{3} & 2 & \frac{140}{27} & -\frac{451}{405} & : & \frac{9106}{607} \\ -\frac{1073}{81} & \frac{10043}{263} & -\frac{3426}{407} & \frac{5222}{3} & 0 & -\frac{2}{3} & \frac{2187}{9} & \frac{538}{65} & : & \frac{2526}{65} \\ -32 & -\frac{1789}{45} & -\frac{1931}{45} & -\frac{1883}{29} & \frac{8}{3} & 52 & \frac{424}{27} & -\frac{14809}{82} & : & -\frac{1489}{134} \\ -\frac{7633}{81} & -\frac{5679}{80} & -\frac{252}{436} & -\frac{13559}{3} & 0 & -\frac{4}{3} & \frac{8018}{3} & \frac{2841}{35} & : & \frac{12035}{154} \\ -\frac{48}{15} & -\frac{887}{15} & -\frac{973}{35} & -\frac{15324}{3} & 4 & 34 & \frac{116}{3} & -\frac{7273}{5} & : & -\frac{4949}{15} \\ -\frac{529}{30} & -\frac{15133}{15} & -\frac{6658}{140} & -\frac{43443}{140} & 0 & -\frac{2}{3} & \frac{1006}{3} & \frac{1066}{15} & : & 409 \end{bmatrix} \]
\[ [U : \mu] = \begin{bmatrix}
1 & -\frac{1}{2} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 : -1 \\
0 & 0 & 0 & 0 & 1 & -\frac{1}{2} & 0 & \frac{1}{4} : 2 \\
1 & \frac{3}{2} & 2 & \frac{9}{4} & 0 & 0 & 0 & 0 : 1 \\
0 & 0 & 0 & 0 & 1 & \frac{3}{2} & 2 & \frac{9}{4} : 8 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 : 1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 : -3
\end{bmatrix} \]

Hence, the new augmented matrix based on condition can be obtained as follows:

\[ [\tilde{W} : \tilde{G}] = \begin{bmatrix}
0 & -1 & 1 & 6 & 0 & 0 & 0 & 0 : 5 \\
-1 & \frac{1}{2} & 0 & -\frac{1}{4} & 0 & 0 & 0 & 6 : 13 \\
1 & -\frac{1}{2} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 : -1 \\
0 & 0 & 0 & 0 & 1 & -\frac{1}{2} & 0 & \frac{1}{4} : 2 \\
1 & \frac{3}{2} & 2 & \frac{9}{4} & 0 & 0 & 0 & 0 : 1 \\
0 & 0 & 0 & 0 & 1 & \frac{3}{2} & 2 & \frac{9}{4} : 8 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 : 1 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 : -3
\end{bmatrix} \]

By solving this system, substituting the resulting unknown Euler coefficients matrix:

\[ A = \begin{bmatrix}
-1 & \frac{1}{2} & -\frac{1}{2} & 1 & 1 & -1 & 2 & 2
\end{bmatrix}^T \]

into Equation (2.1) we obtain the exact solution for \( N = 3 \) as

\[ y_{1,3}(t) = t^3 - 2t^2 + t - 1, \quad y_{2,3}(t) = 2t^3 - t^2 - 3t + 2 \]

**Example 2.** Consider the following system of linear Volterra integro differential equations

\[ y'_1(t) + y_2(t) = 1 + t + t^2 + \int_0^t (-y_1(s) - y_2(s)) \, ds \]

\[ y'_2(t) - y_1(t) = -1 - t + \int_0^t (-y_1(s) + y_2(s)) \, ds \]
We selected our second example from [5, 23, 39, 42], which solved this system of VIDEs by the Adomian Decomposition (ADM), Homotopy Perturbation (HPM), Bessel collocation and Fibonacci polynomials method. The proposed method is applied and the approximate solutions are obtained as

\[ y_1(t) = 0.23699t^3 + 0.47415t^2 + 2t + 1 \]

\[ y_2(t) = -0.23252t^3 - 0.47802t^2 - 1 \]

\[ y_1(0) = 1, y_2(0) = -1 \]

The comparison of the approximate solutions obtained by the Euler collocation method with the absolute errors obtained by Adomian decomposition (ADM), Homotopy perturbation (HPM) and Bessel collocation methods are given in Table 2 and 3. The results obtained with the proposed method and the results obtained with Fibonacci polynomials are very close to each other for \( N = 10 \).

It can be seen that the approximate solutions obtained by Euler collocation method give better results than the solutions obtained by Adomian decomposition, Homotopy perturbation and Bessel collocation method in Table 1.

<table>
<thead>
<tr>
<th>( N )</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
<td>1.0630</td>
<td>1.0880</td>
<td>1.0900</td>
<td>1.1020</td>
<td>1.1040</td>
<td>1.1170</td>
<td>1.1880</td>
</tr>
<tr>
<td>( EOT_N )</td>
<td>0.0928</td>
<td>0.1042</td>
<td>0.0701</td>
<td>0.0710</td>
<td>0.0736</td>
<td>0.0794</td>
<td>0.5849</td>
</tr>
</tbody>
</table>

It can be inferred from Table 1 that the CPU running time results with the experimental order of timing complexity (\( EOT_N \)) are highly remarkable in the meaning of the numerical computations.
Table 2. Absolute errors of $y_1(t)$ for example 2.

| t   | ADM $|e_{1,7}(t)|$ | HPM $|e_{1,7}(t)|$ | Bessel $|e_{1,7}(t)|$ | Euler $|e_{1,7}(t)|$ | Bessel $|e_{1,10}(t)|$ | Euler $|e_{1,10}(t)|$ |
|-----|-----------|-------------|----------------|----------------|----------------|----------------|
| 0   | 0         | 0           | 0              | 0              | 0              | 0              |
| 0.2 | 3.0e-09   | 3.0e-09     | 3.0530e-09     | 3.0530e-09     | 1.1569e-13     | 1.0303e-13     |
| 0.4 | 3.20e-07  | 3.20e-07    | 3.2758e-09     | 3.2759e-09     | 1.6609e-13     | 1.168e-13      |
| 0.6 | 5.364e-06 | 5.364e-06   | 3.1170e-09     | 3.1173e-09     | 2.3115e-13     | 1.2346e-13     |
| 0.8 | 3.909e-05 | 3.909e-05   | 2.4377e-09     | 2.4382e-09     | 2.9798e-13     | 1.1324e-13     |
| 1   | 1.79861e-04 | 1.79861e-04 | 8.4851e-08     | 8.4852e-08     | 4.0115e-12     | 4.289e-12      |

Table 3. Absolute errors of $y_2(t)$ for example 2.

| t   | ADM $|e_{2,7}(t)|$ | HPM $|e_{2,7}(t)|$ | Bessel $|e_{2,7}(t)|$ | Euler $|e_{2,7}(t)|$ | Bessel $|e_{2,10}(t)|$ | Euler $|e_{2,10}(t)|$ |
|-----|-----------|-------------|----------------|----------------|----------------|----------------|
| 0   | 0         | 0           | 0              | 0              | 0              | 0              |
| 0.2 | 2.0e-09   | 2.0e-09     | 2.0681e-09     | 2.0681e-09     | 5.3069e-14     | 6.6835e-14     |
| 0.4 | 3.20e-07  | 3.20e-07    | 1.6889e-09     | 1.6890e-09     | 2.6645e-15     | 5.6399e-14     |
| 0.6 | 5.359e-06 | 5.359e-06   | 1.1459e-09     | 1.1461e-09     | 6.7946e-14     | 4.8406e-14     |
| 0.8 | 3.9028e-05 | 3.9028e-05 | 2.0071e-10     | 2.0118e-10     | 1.6120e-13     | 3.4417e-14     |
| 1   | 1.79279e-04 | 1.79279e-04 | 8.8769e-08     | 8.8770e-08     | 4.8770e-12     | 4.5959e-12     |

Figure 1. Comparison of the actual absolute error functions for $y_1(t)$ and $y_2(t)$ of system for example 2. And also, absolute error functions for $y_1(t)$ and $y_2(t)$ of system for Example 2 illustrated in Figure 1.

1 It can be clearly seen that these approach to zero, as N is increased.
2 The upper bounds of the residual error functions of the solutions in $0 < t < 1$: 
The first eight elements of previously by using Tau, Bernstein collocation and spectral methods.

The following system of linear Fredholm integro differential equations was solved

It can be easily seen that \( \mu_{1,N} = \{ |R_{1,4}|, |R_{1,5}|, |R_{1,6}|, \ldots \} = \{0.068764, 0.071350, 0.050404 \ldots \} \)

\[
\frac{|R_{1,N+1}|}{|R_{1,N}|} < 1, \quad \frac{|R_{2,N+1}|}{|R_{2,N}|} < 1
\]

It can be easily seen that \( \{ |R_{1,N}| \}_{N=3}^{\infty} \) satisfies the inequality \( |R_{1,N+1}| \leq \mu_{1,N}|R_{1,N}| \) with respect to \( \mu_{1,N} < 1 \)

and \( \{ |R_{2,N}| \}_{N=3}^{\infty} \) satisfies the inequality \( |R_{2,N+1}| \leq \mu_{2,N}|R_{2,N}| \) with respect to \( \mu_{2,N} < 1 \). Hence \( \{ |R_{1,N}| \}_{N=3}^{\infty} \)

and \( \{ |R_{2,N}| \}_{N=3}^{\infty} \) are convergent.

The first eight elements of \( \{ |R_{1,N}| \}_{N=3}^{\infty} \) and \( \{ |R_{2,N}| \}_{N=3}^{\infty} \) are illustrated in Figure(2) and one can see that these approach to zero, as \( N \) is increased.

**Example 3.** [3, 22, 33] The following system of linear Fredholm integro differential equations was solved previously by using Tau, Bernstein collocation and spectral methods.

\[
y''_1(t) + y''_2(t) = 8 + 3/10t + 3t^2 - \int_0^1 ((2ts)y_1(s) - 6tsy_2(s)) \, ds
\]
The CPU running time results with the experimental order of timing complexity (EOT$_N$) are given in Table 4. The comparison of the approximate solutions obtained by the Euler collocation method with the absolute errors obtained by Tau, Spectral and Bernstein operational matrix methods are given in Table 5 and 6. It can be seen that the approximate solutions obtained by Euler collocation method give better results than the solutions obtained by Tau, Spectral and Bernstein operational matrix method in Table 5 and 6. And also,
Table 5. Absolute errors of $y_1(t)$ for example 3.

| t  | $|e_{1,3}(t)|$ | Bernstein $|e_{1,3}(t)|$ | Euler $|e_{1,3}(t)|$ | Spectral $|e_{1,5}(t)|$ | Bernstein $|e_{1,5}(t)|$ | Euler $|e_{1,5}(t)|$ |
|----|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0  | 3.1e-14        | 3.55271e-15     | 0.0e-09         | 8.88178e-16     | 9.99201e-16     | 1.33227e-15     |
| 0.2| 3.1e-14        | 3.10862e-15     | 9.1e-08         | 1.36470e-06     | 1.33227e-15     | 1.33227e-15     |
| 0.4| 3.1e-14        | 2.66454e-15     | 6.031e-06       | 2.42077e-06     | 1.33227e-15     | 1.33227e-15     |
| 0.6| 3.1e-14        | 2.22045e-15     | 7.08e-05        | 3.31536e-06     | 1.33227e-15     | 1.33227e-15     |
| 0.8| 3.1e-14        | 1.33227e-15     | 4.10261e-04     | 1.21331e-05     | 1.77636e-15     | 1.77636e-15     |
| 1  | 3.1e-14        | 1.33227e-15     | 1.61516e-03     | 1.29523e-04     | 1.77636e-15     | 1.77636e-15     |

Table 6. Absolute errors of $y_2(t)$ for example 3.

| t  | $|e_{1,3}(t)|$ | Bernstein $|e_{1,3}(t)|$ | Euler $|e_{1,3}(t)|$ | Spectral $|e_{1,5}(t)|$ | Bernstein $|e_{1,5}(t)|$ | Euler $|e_{1,5}(t)|$ |
|----|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 0  | 3.1e-14        | 1.77636e-15     | 0.0e-09         | 7.77156e-16     | 9.99201e-16     | 1.33227e-15     |
| 0.2| 2.7e-14        | 1.55431e-15     | 0.0e-09         | 2.8736e-07      | 9.99201e-16     | 1.33227e-15     |
| 0.4| 2.4e-14        | 1.22125e-15     | 3.25e-07        | 4.54527e-07     | 7.77156e-16     | 1.33227e-15     |
| 0.6| 2.3e-14        | 1.05471e-15     | 5.527e-06       | 5.63863e-07     | 5.55112e-16     | 1.33227e-15     |
| 0.8| 2.0e-14        | 8.88178e-16     | 4.1242e-05      | 2.69271e-06     | 0               | 1.33227e-15     |
| 1  | 2.0e-14        | 6.66134e-16     | 1.95682e-04     | 3.35922e-05     | 8.88178e-16     | 1.33227e-15     |

Figure 3. Comparison of the actual absolute error functions for $y_1(t)$ and $y_2(t)$ of system for Example 3 illustrated in Figure 3. It can be clearly seen that these approach to zero, as $N$ is increased.

The upper bounds of the residual error functions of the solutions in $0 < t < 1$:  

1
Example 4. [16] As a last example, we consider the following system:

\[
\begin{align*}
|\mathcal{R}_{1,3}| &= 0.1433e - 14, \\
|\mathcal{R}_{1,4}| &= 0.1812e - 14, \\
|\mathcal{R}_{1,5}| &= 0.7568e - 15, \\
|\mathcal{R}_{1,6}| &= 0.1305e - 13, \\
|\mathcal{R}_{1,7}| &= 0.2640e - 13, \\
|\mathcal{R}_{1,8}| &= 0.3924e - 13, \\
|\mathcal{R}_{1,9}| &= 0.1892e - 13, \\
|\mathcal{R}_{1,10}| &= 0.1852e - 13, \\
|\mathcal{R}_{2,3}| &= 0.1008e - 14, \\
|\mathcal{R}_{2,4}| &= 0.3885e - 14, \\
|\mathcal{R}_{2,5}| &= 0.1116e - 15, \\
|\mathcal{R}_{2,6}| &= 0.4518e - 13, \\
|\mathcal{R}_{2,7}| &= 0.9779e - 13, \\
|\mathcal{R}_{2,8}| &= 0.1319e - 12, \\
|\mathcal{R}_{2,9}| &= 0.3777e - 13, \\
|\mathcal{R}_{2,10}| &= 0.3864e - 14,
\end{align*}
\]

Figure 4. The residual functions of example 3

The first eight elements of \(\{|\mathcal{R}_{1,N}|\}_{N=3}^{\infty}\) and \(\{|\mathcal{R}_{2,N}|\}_{N=3}^{\infty}\) are illustrated in Figure 4. It’s clearly seen that in Figure 3 and in Figure 4, Euler collocation method give better results than the others.

Example 4. [16] As a last example, we consider the following system:

\[
y_1'(t) = -2 + t^2 - t^4 + \frac{3t^5}{20} + 2t^6 + \frac{t^7}{5} - \frac{t^8}{8} + \int_0^t (s^3 - t^2) y_1(s) + (12s^2 - t)y_2(s) \, ds
\]

\[
y_2'(t) = 4 - 8t - \frac{t^3}{3} + 2t^4 - \frac{8t^5}{5} + \frac{t^6}{30} - 4e^t + \int_0^t ((s - t)y_1(s) + 8(1 - s)y_2(s) + 2y_3(s)) \, ds
\]

\[
y_3'(t) = 3 - \frac{7t^2}{2} + \frac{4t^3}{3} + \frac{6t^5}{5} - \frac{7t^6}{30} + \int_0^t ((2t - s)y_1(s) + 6s y_2(s) + 2y_3(s)) \, ds
\]

\[
y_1(0) = 0, \quad y_2(0) = 1, \quad y_3(0) = 2
\]

The approximate solutions are obtained for \(N = 3\) and \(N = 8\) as

\[
y_{1,3}(t) = 1.33512t^3 - 0.44503t^2 - 2t
\]
\[ y_{2,3}(t) = -0.99908t^3 - 0.00096t^2 + 1 \]
\[ y_{3,3}(t) = 0.46547t^3 + 0.95226t^2 + 3t + 2 \]
\[
y_{1,8}(t) = -1.43534 \times 10^{-11}t^8 + 7.10806 \times 10^{-11}t^7 - 3.71604 \times 10^{-12}t^6 \\
+ 2.13106 \times 10^{-11}t^5 + t^4 + 1.95871 \times 10^{-12}t^3 - 1.45595 \times 10^{-13}t^2 \\
- 2t + 2.35502 \times 10^{-15} 
\]
\[
y_{2,8}(t) = 1.72905 \times 10^{-10}t^8 + 6.44961 \times 10^{-10}t^7 - 2.96804 \times 10^{-9}t^6 \\
+ 3.97823 \times 10^{-9}t^5 - 2.50223 \times 10^{-9}t^4 - t^3 + 1.9207 \times 10^{-11}t^2 \\
- 3.07188 \times 10^{-16}t + 1 
\]
\[
y_{3,8}(t) = 7.71779 \times 10^{-5}t^8 + 0.00035t^7 + 0.00282t^6 + 0.01664t^5 \\
+ 0.08334t^4 + 0.33333t^3 + t^2 + 3t + 2 
\]

for \( N = 3, 8 \) respectively.

When the solutions are examined, it will be seen that the exact solution of Volterra IDE is the Taylor expansion of \( y_1(t) = 3t^2 + 1 \), \( y_2(t) = t^3 + 2t - 1 \).

### Table 7. CPU time and \( EOT_N \) for example 3.

<table>
<thead>
<tr>
<th>( N )</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU time</td>
<td>1.1450</td>
<td>1.1530</td>
<td>1.1850</td>
<td>1.2080</td>
<td>1.2370</td>
</tr>
<tr>
<td>( EOT_N )</td>
<td>0.1520</td>
<td>0.0312</td>
<td>0.1501</td>
<td>0.1247</td>
<td>0.1777</td>
</tr>
</tbody>
</table>

The CPU running time results with the experimental order of timing complexity (\( EOT_N \)) are given in Table 7. The comparison of the approximate solutions obtained by the Euler collocation method with the absolute errors obtained by Sinc collocation and Chebyshev wavelet methods are given in Table 8, 9 and 10.

### Table 8. Absolute errors of \( y_1(t) \) for example 4.

| \( t \) | Sinc collocation \( |e_{1,8}(t)| \) | Chebyshev wavelet \( |e_{1,8}(t)| \) | Euler \( |e_{1,8}(t)| \) |
|-------|------------------|------------------|------------------|
| 0     | 0                | 2.6908366696184e-13 | 2.355018829071e-15 |
| 0.1   | 1.358996248868038e-9 | 2.293998324631729e-13 | 2.026157019941e-15 |
| 0.2   | 4.6938202390865e-9  | 1.59372515184912e-13 | 2.664535259100e-15 |
| 0.3   | 7.953928582438152e-8 | 4.38316501220118e-13 | 1.976196983833e-14 |
| 0.4   | 2.280309290281224e-7 | 9.81881242978488e-13 | 1.424416109943e-13 |
| 0.5   | 3.158464549151807e-7 | 2.2981660974074e-13 | 6.540323838067e-13 |
| 0.6   | 1.91852210115262e-7 | 4.035888978706207e-12 | 2.228883744327e-12 |
| 0.7   | 1.1059680320557e-7  | 1.16204823514658e-11 | 6.22344142234e-12 |
| 0.8   | 3.56242512599862e-7 | 2.28779177924075e-11 | 1.56063907942e-11 |
| 0.9   | 3.85786830733385e-7 | 4.14988043928588e-11 | 3.277045301786e-11 |
It can be seen that the approximate solutions obtained by Euler collocation method give better results than the solutions obtained by Tau, Spectral and Bernstein operational matrix method in Table 8, 9 and 10. And also, absolute error functions for $y_1(t)$, $y_2(t)$ and $y_3(t)$ of system for Example 4 illustrated in Figure 5. It can be clearly seen that these approach to zero, as N is increased.

The upper bounds of the residual error functions of the solutions in $0 < t < 1$:

$$|R_{1,3}| = 0.1121, \quad |R_{1,4}| = 0.3236e - 05, \quad |R_{1,5}| = 0.2348e - 07, \quad |R_{1,6}| = 0.2497e - 09,$$

$$|R_{1,7}| = 0.6027e - 11, \quad |R_{1,8}| = 0.1199e - 12,$$

$$|R_{2,3}| = 0.000241, \quad |R_{2,4}| = 0.7797e - 05, \quad |R_{2,5}| = 0.2716e - 06, \quad |R_{2,6}| = 0.1179e - 07,$$

$$|R_{2,7}| = 0.4102e - 09, \quad |R_{2,8}| = 0.1402e - 10,$$

$$|R_{3,3}| = 1.586408, \quad |R_{3,4}| = 1.573905, \quad |R_{3,5}| = 1.572941, \quad |R_{3,6}| = 1.572877,$$

$$|R_{3,7}| = 1.572873, \quad |R_{3,8}| = 1.572873,$$

$$\mu_{1,N} = \left\{ \frac{|R_{1,4}|}{|R_{1,3}|}, \frac{|R_{1,5}|}{|R_{1,4}|}, \frac{|R_{1,6}|}{|R_{1,5}|}, \ldots \right\} = \{0.000029, 0.000725, 0.010632 \ldots\}$$
The first six elements of solutions approach to the exact solution, as \( N \) is increased. This situation reflects on the residual functions of examples. In Examples 1 - 4, we have computed absolute errors and residual errors. The Euler polynomial satisfies the inequality \( |e_{3,N+1}|_{R_{3,N}} \leq \mu_{3,N} |e_{3,N+1}|_{R_{3,N}} \) with respect to \( \mu_{3,N} < 1 \). Hence \( \{ |e_{1,N}|_{R_{1,N}} \}_{N=3}^{\infty} \), \( \{ |e_{2,N}|_{R_{2,N}} \}_{N=3}^{\infty} \) and \( \{ |e_{3,N}|_{R_{3,N}} \}_{N=3}^{\infty} \) are convergent.

The first six elements of \( \{ |e_{1,N}|_{R_{1,N}} \}_{N=3}^{\infty} \), \( \{ |e_{2,N}|_{R_{2,N}} \}_{N=3}^{\infty} \) and \( \{ |e_{3,N}|_{R_{3,N}} \}_{N=3}^{\infty} \) are illustrated in Figure 6 and one can see that these approach to zero, as \( N \) is increased.

### 5. Conclusion

We have applied the Euler polynomial matrix collocation method to some model problems. We have also solved rapidly and efficiently these problems without requiring detailed procedure as can be followed CPU running time with \( EOT_{N} \). As seen from Tables 1 - 10, the Euler polynomial solutions coincide with the exact solutions of model problems. To show efficiency of the method, we have applied the presented scheme for four numerical examples. In Examples 1 - 4, we have computed absolute errors and residual errors. The Euler polynomial solutions approach to the exact solution, as \( N \) is increased. This situation reflects on the residual functions of...
problems. Hence, it is easily observed that the present method is so convenient to solve the systems of linear integro differential model problems.

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

References


