

Research Article

Approximation Technique for Solving Linear Volterra Integro-Differential Equations with Boundary Conditions

Mohamed E. A. Alnair^{1,2} and Ahmed A. Khidir ^{1,3}

¹Department of Mathematics, Faculty of Sciences, University of Tabuk, P.O. Box 741, Tabuk, Saudi Arabia

²Department of Mathematics and Physics, Gezira University, P. O. Box 20, Sudan

³Faculty of Technology of Mathematical Sciences and Statistics, Alneelain University, Khartoum, Sudan

Correspondence should be addressed to Ahmed A. Khidir; ahmed.khidir@yahoo.com

Received 1 March 2022; Revised 20 March 2022; Accepted 29 March 2022; Published 12 April 2022

Academic Editor: Yufeng Xu

Copyright © 2022 Mohamed E. A. Alnair and Ahmed A. Khidir. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new technique for solving linear Volterra integro-differential equations with boundary conditions. The method is based on the blending of the Chebyshev spectral methods. The application of the proposed method leads the Volterra integro-differential equation to a system of algebraic equations that are easy to solve. Some examples are introduced and the obtained results are compared with exact solution as well as the methods that reported in the literature to illustrate the effectiveness and accuracy of the method. The results demonstrate that there is congruence between the numerical and the exact results to a high order of accuracy. Tables were generated to verify the accuracy convergence of the method and error. Figures are presented to show the excellent agreement between the results of this study and the results from literature.

1. Introduction

Many Mathematical models of physical phenomena usually results in linear or nonlinear Volterra integral and integro-differential equations. These equations play a crucial role in many fields of sciences such as economics, chemistry, finance, biology, and engineering. In engineering, physics, and mathematics areas, solving a problem means solving a set of differential equations and integral or integro-differential equations. In general, finding analytic solutions of Volterra integro-differential equations is usually difficult so it is required to obtain an approximate solution. Therefore, they have been of great interest by several researchers and scientists of Volterra integro-differential equations. Several studies have been carried out and developed in the literature to find analytical or numerical solutions for Volterra integro-differential equations. Loh and Phang [1] used Genocchi polynomials to solve numerically a system of Volterra integro-differential equations which is done by approximating functions using Genocchi polynomials and

derivatives using Genocchi polynomials operational matrix of integer-order derivative. Loh et. al [2] obtained the approximate solution for a new class of time-fractional partial integro-differential equation of the Caputo-Volterra type. Mirzaee et. al [3] used the moving least squares and spectral collocation method to estimate the solution of nonlinear stochastic Volterra integro-differential equations. A class of block boundary value methods has been constructed by Zhou and Stynes [4] for the solution of linear neutral Volterra integro-differential equations with weakly singular kernels. Adewumi et al. [5] presented new efficient numerical methods for solving Volterra integro-differential equations and a system of nonlinear delay integro-differential equations which arises in biology. In [6], the author discussed variational iteration method for the solutions of different linear and nonlinear Volterra integral equations. For some more recent and interesting results, we refer to [7–12] and the references cited therein.

Recently, Lotfi and Alipanah [13] presented a spectral element method to solve linear Volterra integro-differential

equations; Alrabaiah et al. [14] used Laplace Adomian decomposition method to find the semianalytic solutions of nonlinear Volterra fractional integro-differential equations; a numerical method was obtained to find a solution for a singularly perturbed Volterra integro-differential equation by Nana et al. [15].

The main contribution of this work is to introduce an efficient numerical technique for solving the linear Volterra integro-differential equations together with boundary conditions. The technique is based on the Chebyshev spectral collocation method. The Chebyshev spectral collocation methods have been applied successfully in different fields of sciences and engineering because of their ability to give high accuracy of boundary value problems (see [16–20]. The application of the current method leads the Volterra integro-differential equation to a system of algebraic equations that are easy to solve when compared to a system of Volterra integro-differential equation. The main benefit of this method is that we used our proposed technique without any assumptions on large or small parameters. The investigation is mainly targeted to devise this consistent method to integro-differential equations with boundary conditions. The main advantages of this approach over the standard same approaches are (i) the technique suggests a standard way of choosing the linear operator of the integro-differential equation whereas the other methods are choosing a linear operator to be simple in order to ensure that the integro-differential equation can be easily solved, and (ii) this algorithm transforms the integro-differential equation into a system of linear algebraic equations that is easier and faster to solve when compared to a system of integral equations.

This paper is organized as follows: in Section 2, we introduce a description of the proposed method. In Section 3, the method is implemented using some examples with known analytical solution. The results are discussed and investigated in Section 4. Finally, the conclusions are given in Section 5.

2. Description of the New Technique

Consider the following linear second order Volterra integro-differential equation of the second kind given by

$$P_2(x) \frac{d^2 u}{dx^2} + P_1(x) \frac{du}{dx} + P_0(x)u + \int_0^x K(x,t)u(t)dt = f(x), 0 \leq x \leq t \leq b, \tag{1}$$

under the boundary conditions

$$u(0) = \alpha, u(b) = \beta, \tag{2}$$

where $u(x)$ is an unknown function to be determined; $P_0(x), P_1(x), P_2(x), K(x, t)$, and $f(x)$ are known real valued functions defined on $0 \leq x \leq t \leq b$.

To illustrate the idea of the new algorithm, we assume that the kernel $K(x, t)$ of Equation (1) can be separated into

a product of two functions, namely, $w(x)$ and $v(t)$; consequently, Equation (1) can be obtained as follows:

$$P_2(x)u''(x) + P_1(x)u'(x) + P_0(x)u + w(x) \int_0^x v(t)u(t)dt = f(x). \tag{3}$$

Now, the integral term of Volterra integro-differential equation above is expressed as

$$\int_0^x v(t)u(t)dt = \Phi(x). \tag{4}$$

One approach is to note that the integral equation (4) is an initial value problem expressed as

$$\Phi'(x) = v(x)u(x), \text{ with } \Phi(0) = 0. \tag{5}$$

This differential equation is easy to solve using any method; here, we used the Chebyshev spectral collocation method; the functions $\Phi(x), v(x)$ and $u(x)$ are approximated as a truncated series of Chebyshev polynomials given by the form [21–23].

$$\begin{aligned} \Phi(x) &\approx \Phi(x_j) = \sum_k^N \tilde{\Phi}_k T_k(x_j), \\ u(x) &\approx u(x_j) = \sum_k^N \tilde{u}_k T_k(x_j), \\ v(x) &\approx v(x_j) = \sum_k^N \tilde{v}_k T_k(x_j), \end{aligned} \tag{6}$$

where T_k is the k th Chebyshev polynomial and $\tilde{\Phi}, \tilde{u}$, and \tilde{v} are the Chebyshev coefficients, and $x_j, j = 0, 1, 2, \dots, N$ are the Gauss-Lobatto collocation points (see [21]) defined by

$$x_j = \frac{1}{2}x_N \left(1 - \cos \frac{j\pi}{N} \right), j = 0, 1, 2, \dots, N, \tag{7}$$

where $N + 1$ is the number of collocation points. The derivatives at the collocation points are represented as

$$\frac{d^r(\cdot)}{dx^r} \approx \sum_{k=0}^N \mathcal{D}_{jk}^r(\cdot) = \mathcal{D}^r(\cdot), \tag{8}$$

where r is the order of differentiation and \mathcal{D} is the Chebyshev spectral differentiation matrix whose entries (see [22]) are given by

$$\left. \begin{aligned} \mathcal{D}_{00} &= \frac{2N^2 + 1}{6}, \\ \mathcal{D}_{jk} &= \frac{cj(-1)^{j+k}}{ckx_j - x_k}, \quad j \neq k, j, k = 0, 1, 2, \dots, N, \\ \mathcal{D}_{kk} &= \frac{x_k}{1 - x_k^2}, \quad k = 1, 2, \dots, N - 1, \\ \mathcal{D}_{NN} &= -\frac{2N^2 + 1}{6} \end{aligned} \right\} \quad (9)$$

Here, $c_0 = c_N = 2$ and $c_j = 1$ with $1 \leq j \leq N-1$.

Substituting the above assumptions in Equation (5) yields a system of algebraic equations expressed as the following matrix equation

$$\begin{bmatrix} \mathcal{D}_{0,0} & \cdots & \mathcal{D}_{0,N} \\ \mathcal{D}_{1,0} & \cdots & \mathcal{D}_{1,N} \\ \vdots & & \vdots \\ \mathcal{D}_{N,0} & \cdots & \mathcal{D}_{N,N} \end{bmatrix} \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \vdots \\ \Phi_N \end{bmatrix} = \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_N \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix}, \quad (10)$$

where $\Phi_i = \Phi(x_i)$, $u_i = u(x_i)$, and $v_i = v(x_i)$. The solution of this system for $\Phi(x_i)$ is obtained by

$$\begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \vdots \\ \Phi_N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \mathcal{D}_{1,0} & & & & \mathcal{D}_{1,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{D}_{N,0} & & & & \mathcal{D}_{N,N} \end{bmatrix}^{-1} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_N \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_N \end{bmatrix} - \begin{bmatrix} u_0 v_0 \\ u_0 v_0 \\ \vdots \\ u_0 v_0 \end{bmatrix}, \quad (11)$$

where we observe that the first row of Chebyshev differential matrix \mathcal{D} in equation above is replaced by the row $[1, 0, 0, \dots]$, and we add the vector $-[u_0 v_0, u_0 v_0, \dots, u_0 v_0]^T$; this is caused by imposing the condition $\phi(0) = 0$ into Equation (10). According to the integral Equation (4) and Equation (11), we introduce the following integral operator $L_{[v]}(\cdot)$ defined as

$$L_{[v]}u \approx u_0 v_0 + \int_0^x v(t)u(t)dt, \quad (12)$$

where $u = [u_0 u_1 \cdots u_N]^T$, $[v] = [v_0 v_1 \cdots v_N]^T$ and

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \mathcal{D}_{1,0} & \cdots & & \mathcal{D}_{1,N} \\ \vdots & & & \vdots \\ \mathcal{D}_{N,0} & \cdots & & \mathcal{D}_{N,N} \end{bmatrix}^{-1}. \quad (13)$$

The operator L is square matrix of size $(N + 1) \times (N + 1)$.

2.1. The Linearity of the Operator L . It is clear that the above operator is a linear operator because

$$\begin{aligned} & \int_0^x [v_1(t)\psi_1(x)u_1(t) + v_2(t)\psi_2(x)u_2(t) + \cdots + v_n(t)\psi_n(x)u_n(t)]dt \\ &= \int_0^x v_1(t)\psi_1(x)u_1(t)dt + \int_0^x v_2(t)\psi_2(x)u_2(t)dt + \cdots \\ &+ \int_0^x v_n(t)\psi_n(x)u_n(t)dt = \psi_1(x) \int_0^x v_1(t)u_1(t)dt \\ &+ \psi_2(x) \int_0^x v_2(t)u_2(t)dt + \cdots + \psi_n(x) \int_0^x v_n(t)u_n(t)dt \\ &= [\psi_1(x)]L_{[v_1]}u_1 + [\psi_2(x)]L_{[v_2]}u_2 + \cdots + [\psi_n(x)]L_{[v_n]}u_n \\ &- \psi_1(x)v_1(0)u_1(0) - \psi_2(x)v_2(0)u_2(0) - \cdots - \psi_n(x)v_n(0)u_n(0). \end{aligned} \quad (14)$$

Therefore, the integral operator L is a linear operator.

2.2. Existence and Uniqueness of the Operator L . In this subsection, we show that the operator L exists and is unique.

Theorem 1.

- (i) Any square matrix A is invertible (nonsingular) if and only if $\det(A) \neq 0$
- (ii) If A is an invertible matrix, then its inverse is exist and unique

Proof. See Larson and David [24]. □

According to the theorem above, the existence and uniqueness of the linear operator L depends on the existence and uniqueness of inverse of matrix

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ \mathcal{D}_{1,0} & \cdots & & \mathcal{D}_{1,N} \\ \vdots & & & \vdots \\ \mathcal{D}_{N,0} & \cdots & & \mathcal{D}_{N,N} \end{bmatrix}. \quad (15)$$

The determinants of matrix above have been computed on the domain $[0, b]$ for $N = 10, 11, \dots, 100$. In Figure 1, we plotted the determinants for different values of b varied N . We observe that from the figure, all the computed determinants are not equal to zero, and it is noticed that all the determinants are greater than or equal to 1, i.e., the inverse of the operator L exists and is unique.

Now, replace the derivative operator d^i/dx^i of the integro-differential Equation (1) by the Chebyshev spectral differentiation matrix \mathcal{D}^i defined in (9) and also replace the integral term of the integro-differential Equation (1) by the operator L , which gives

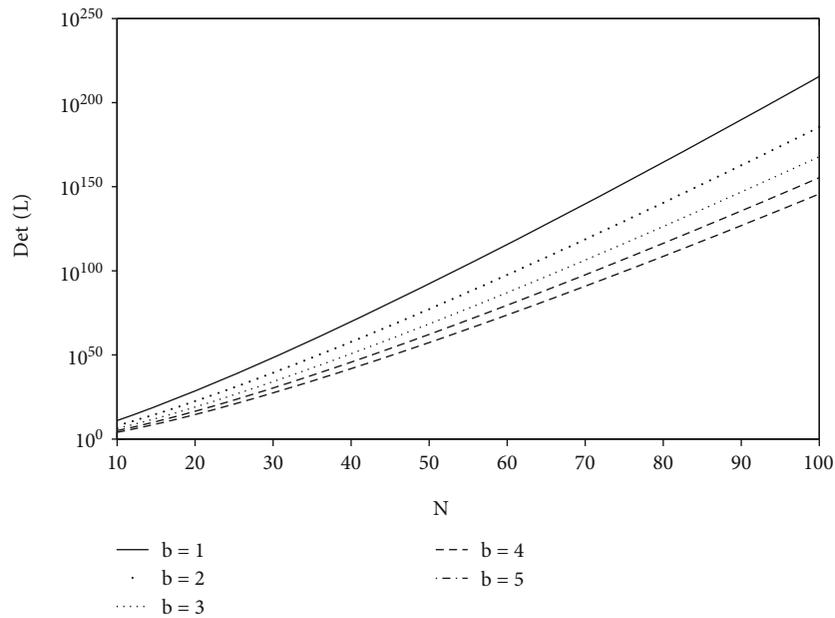


FIGURE 1: The determinants (det) of the matrices L .

$$\begin{aligned} & \text{diag} [P_2(x)]\mathcal{D}^2[u(x)]^T + \text{diag} [P_1(x)]\mathcal{D}[u(x)]^T \\ & + \text{diag} [P_0(x)][u(x)]^T + \text{diag} [w(x)]L_{[v(x)]}[u(x)]^T = [f(x)]^T. \end{aligned} \tag{16}$$

Here, the superscript T denotes matrix transpose, $[\dots]$ is the diagonal matrix of size $(N + 1) \times (N + 1)$ that puts the vector $[\dots]$ of size $(N + 1) \times 1$ on the main diagonal, and $[P_2(x)]$, $[P_1(x)]$, $[P_0(x)]$, $[f(x)]$, $[v(x)]$, and $[u(x)]$ are vectors of sizes $(N + 1) \times 1$. The functions $w(x)$ and $v(x)$ are obtained from the kernel function $K(x, t)$ which can be written as $K(x, t) = w(x)v(x)$ as we mentioned before.

Thus, Equation (16) can be written as

$$Au = F, \tag{17}$$

where A is a $(N + 1) \times (N + 1)$ square matrix and F is column vector of sizes $(N + 1) \times 1$ defined by

$$\begin{aligned} A &= \text{diag} [P_2(x)]\mathcal{D}^2 + \text{diag} [P_1(x)]\mathcal{D} + \text{diag} [P_0(x)] \\ & + \text{diag} [w(x)]L_{[v(x)]}, \\ F &= [f(x_0) - \alpha w(x_0)v(x_0), f(x_1) - \alpha w(x_1)v(x_1), \dots, f(x_N) \\ & - \alpha w(x_N)v(x_N)]^T. \end{aligned} \tag{18}$$

The boundary conditions $u(0) = u(x_0) = \alpha$ and $u(b) = u(x_N) = \beta$ are implemented into the system (17) by modifying the first and last rows of matrices A and F in such a way that the system of linear Equation (17) takes the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ A_{1,1} & A_{1,2} & A_{1,N-1} & A_{1,N} \\ \vdots & \vdots & \vdots & \vdots \\ A_{N,1} & A_{N,2} & A_{N,N-1} & A_{N,N} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_{N-1}) \\ u(x_N) \end{bmatrix} = \begin{bmatrix} \alpha \\ F(x_1) \\ \vdots \\ F(x_{N-1}) \\ \beta \end{bmatrix}. \tag{19}$$

After modifying the matrix system (17) to incorporate boundary conditions, the final approximate solution of the linear second order Volterra integro-differential Equation (1) is obtained as

$$u = \tilde{A}^{-1} \tilde{F}, \tag{20}$$

where \tilde{A} and \tilde{F} are the the modified matrices of A and F , respectively.

The general idea underpinning the use of the proposed method is to convert the linear Volterra integro-differential equation by a system of linear algebraic equations that replace the derivative and integral parts in the integral equation by the differential matrix and the introduced integral operator, respectively. The obtained linear algebraic equations can easily be solved with the help of symbolic computation software such as Maple, Mathematica, MATLAB, and other symbolic computer packages.

3. Numerical Examples

In this section, some numerical examples are carried out to illustrate the effectiveness and accuracy of the proposed method. In addition, the numerical results are compared with exact solution, and all of them were performed on the

TABLE 1: Maximum absolute errors of Example 1 for different values of x and N .

x	$N = 5$	$N = 15$	$N = 40$	$N = 60$
0.1	$6.823e - 007$	$1.055e - 014$	$1.776e - 014$	$-1.144e - 014$
0.2	$1.660e - 006$	$1.232e - 014$	$5.884e - 015$	$-5.573e - 014$
0.3	$1.033e - 006$	$6.328e - 015$	$-2.143e - 014$	$-4.929e - 014$
0.4	$3.076e - 007$	$1.565e - 014$	$6.384e - 014$	$-2.319e - 013$
0.5	$9.545e - 007$	$5.551e - 015$	$6.706e - 014$	$-5.529e - 014$
0.6	$3.111e - 007$	$4.441e - 015$	$-2.620e - 014$	$-8.005e - 014$
0.7	$1.001e - 006$	$3.997e - 015$	$4.885e - 015$	$-1.331e - 013$
0.8	$1.610e - 006$	$4.774e - 015$	$2.098e - 014$	$-7.117e - 014$
0.9	$6.875e - 007$	$1.443e - 015$	$-1.665e - 015$	$-6.284e - 014$
CPU				
Time (sec)	0.0495	0.0685	0.0867	0.1242

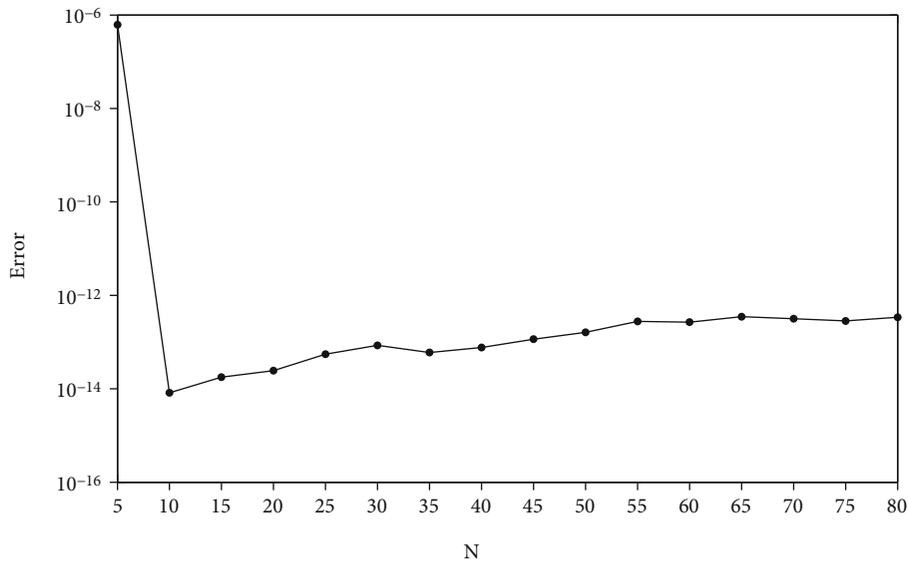


FIGURE 2: Absolute errors of $u(x)$ varied N for Example 1.

computer using a program written in MATLAB v7.5.0 (R2007b). We also showed in figures and tables the approximate solutions, exact solutions, and the absolute error function at the selected points of the given interval.

Example 1. Consider the following linear second-order Volterra integro-differential Equation [25]

$$u''(x) + u(x) + \int_0^x x \tan(t)u(t)dt = x(1 - \cos(x)), \quad (21)$$

together with the boundary conditions $u(0) = 1$ and $u(1) = \cos(1)$. The exact solution is $u(x) = \cos(x)$. The technique of the solution starts by rewriting the integral term in the above equation as

$$\int_0^x x \tan(t)u(t)dt = x \int_0^x \tan(x)u(x)dx. \quad (22)$$

Applying the proposed algorithm on Equation (21) and according to the assumptions (8) and (12), one can write the integro-differential Equation (21) in a system of linear equations:

$$Au = F, \quad (23)$$

where $A = \mathcal{D}^2 + I + \text{diag}[x]L_{[\tan(x)]}$, $F = [x(1 - \cos(x))]^T$, and I is an identity matrix.

To implement the boundary conditions to the systems (23), we replace all the entries of the first and last rows of the matrix A by zeros and then set $A_{00} = A_{NN} = 1$, $F_0 = 1$, and $F_N = \cos(1)$. After modifying the matrix system (23) to incorporate the boundary conditions, the solution is obtained as

$$u = \tilde{A}^{-1} \tilde{F}, \quad (24)$$

TABLE 2: Maximum absolute errors of Example 2 for different values of x and N .

x	$N = 5$	$N = 15$	$N = 40$	$N = 60$
0	0	0	0	0
$\frac{\pi}{10}$	$1.577e - 005$	$1.044e - 014$	$1.454e - 014$	$4.752e - 014$
$\frac{\pi}{9}$	$1.699e - 005$	$1.188e - 014$	$1.282e - 013$	$4.170e - 013$
$\frac{\pi}{8}$	$1.827e - 005$	$1.099e - 014$	$5.718e - 014$	$4.008e - 014$
$\frac{\pi}{7}$	$1.959e - 005$	$7.383e - 015$	$3.453e - 014$	$4.780e - 014$
$\frac{\pi}{6}$	$2.100e - 005$	$2.442e - 015$	$5.262e - 014$	$4.113e - 014$
$\frac{\pi}{5}$	$2.267e - 005$	$1.443e - 015$	$4.841e - 014$	$6.620e - 014$
$\frac{\pi}{4}$	$2.456e - 005$	$1.943e - 015$	$5.501e - 014$	$1.778e - 014$
$\frac{\pi}{3}$	$2.239e - 005$	$5.440e - 015$	$3.786e - 014$	$1.398e - 013$
$\frac{\pi}{2}$	0	0	0	0
CPU				
Time (sec)	0.0511	0.0726	0.0953	0.1394

TABLE 3: Comparison between relative errors of HAM [28] and present results varied x of Example 2.

x	HAM [28]	Present
0.1	$5.556e - 010$	$2.165e - 014$
0.2	$3.105e - 011$	$5.107e - 015$
0.3	$4.853e - 012$	$8.993e - 015$
0.4	$2.342e - 011$	$1.044e - 014$
0.5	$2.588e - 010$	$3.719e - 015$
0.6	$2.110e - 009$	$1.277e - 015$
0.7	$1.873e - 008$	$1.582e - 015$
0.8	$3.717e - 007$	$2.859e - 015$
0.9	$1.400e - 007$	$5.079e - 015$
1.0	$1.987e - 007$	$1.610e - 015$

where \tilde{A} and \tilde{F} are the modified matrices of A and F , respectively.

Example 2. Consider the following Volterra integro-differential equation [26]:

$$\begin{aligned}
 u''(x) + u(x) - \int_0^x e^{-t} \sin(x) u'(t) dt \\
 = \frac{1}{2} e^{-x} \sin(2x) - \sin(x), 0 \leq x \leq \frac{\pi}{2},
 \end{aligned} \tag{25}$$

subject to the boundary conditions $u(0) = -1, u(\pi/2) = 1$. The exact solution is $u(x) = \sin(x) - \cos(x)$.

The integral term of integro-differential Equation (25) can be expressed as

$$\int_0^x e^{-t} \sin(x) u'(t) dt = \sin(x) \int_0^x e^{-x} u'(x) dx. \tag{26}$$

According to the assumptions (8) and (12), we can apply the proposed algorithm on the integro-differential Equation (25); the equation is transformed into the following system of linear equations

$$Au = F, \tag{27}$$

where $A = \mathcal{D}^2 + I - \text{diag}[\sin(x)]L_{[e^{-x}]} \mathcal{D}$ and $F = [1/2e^{-x} \sin(2x) - 2 \sin(x)]^T$. After modifying the matrix system (27) to incorporate the boundary conditions of the integro-differential Equation (25), the solution is obtained as

$$u = \tilde{A}^{-1} \tilde{F}, \tag{28}$$

where \tilde{A} and \tilde{F} are the modified matrices of A and F , respectively.

Example 3. Consider the linear Volterra integro-differential equation given by [26, 27]

$$u'(x) + u(x) - \int_0^x x(1 + 2x)e^{t(x-t)} u(t) dt = 1 + 2x, \tag{29}$$

with the initial conditions $u(0) = 1$ and the exact solution $u(x) = e^{x^2}$.

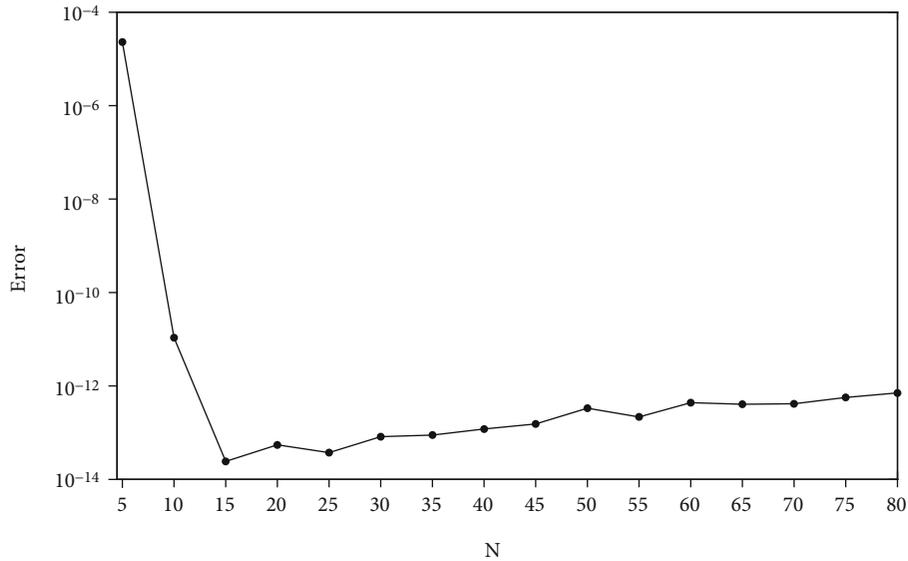


FIGURE 3: Absolute errors of $u(x)$ varied N for Example 2.

TABLE 4: Comparison between relative errors of the Tau method [26], practical matrix method [27], and present results varied x of Example 3.

x	Tau method	Practical matrix method	Present method
0	0	0	0
0.2	$5.722e-012$	$3.785e-012$	$1.998e-015$
0.4	$2.385e-008$	$3.123e-012$	$2.220e-016$
0.6	$3.186e-006$	$7.364e-011$	0
0.8	$1.049e-004$	$1.835e-008$	$2.887e-015$
1.0	$1.615e-003$	$1.214e-006$	0

According to the assumptions (8) and (12), one can express the integral term of integro-differential Equation (29) as

$$\int_0^x x(1+2x)e^{t(x-t)}u(t)dt = x(1+2x) \sum_{i=0}^m \left(\frac{x^i}{i!} \int_0^x x^i e^{-x^2} u(x) dx \right). \tag{30}$$

Applying the proposed algorithm on the integro-differential Equation (29), the equation is transformed into the following system of linear equations

$$Au = F, \tag{31}$$

where $A = \mathcal{D} + I - \text{diag} [x(1+2x)] \sum_{i=0}^m (\text{diag} [x^i/i!] L_{[x^i e^{-x^2}]})$ and $F = [1+x-2x^2]^T$.

Now, incorporate the initial condition $u(0) = 1$ into the system (31) to obtain the final solution of integro-differential Equation (25) as

$$u = \tilde{A}^{-1} \tilde{F}, \tag{32}$$

where \tilde{A} and \tilde{F} are the modified matrices of A and F , respectively, after imposing the initial condition.

Example 4. Consider the following system of linear Volterra integral equations

$$u''(x) - v'(x) + \sin(x)u(x) + v(x) + \int_0^x 4(\cos(x) + \cos(t)u(t) + v(t))dt = f_1(x), \tag{33}$$

$$2v''(x) - 2\sin(x)u(x) + \cos(x)v(x) + \int_0^x (4\sin(t)u(t) - 2\cos(x)v(t))dt = f_2(x), \tag{34}$$

subject to the boundary conditions

$$u(0) = 0, v(0) = 1, \text{ and } u(\pi) = 0, v(\pi) = -1, x \in [0, \pi], \tag{35}$$

where

$$f_1(x) = \frac{1}{2} \sin(x)(7 - \sin(x)) + 3 \cos(x)(\cos(x) - 1). \tag{36}$$

$f_2(x) = 1 - x + 2 \cos(x)(\cos(x) - 2) - 2 \sin(x)$. The exact solutions are $u(x) = -1/2 \sin(x)$ and $v(x) = \sin(x) + \cos(x)$.

Applying the proposed algorithm, one can express the integral parts of integro-differential Equations (33) and (34) as

$$\left. \begin{aligned} \int_0^x 4((\cos(x) + \cos(t))u(t) + v(t))dt &= 4 \cos(x) \int_0^x u(x)dx + 4 \int_0^x \cos(x)u(x)dx + \int_0^x v(x)dx \\ \int_0^x (4 \sin(t)u(t) - 2 \cos(x)v(t))dt &= 4 \int_0^x \sin(x)u(x)dx - 2 \cos(x) \int_0^x v(x)dx \end{aligned} \right\} \tag{37}$$

Applying the proposed algorithm on the system of integro-differential Equations (33) and (34), the equation is transformed into the following system of linear equations

$$AU = F, \tag{38}$$

where A is a $(2N + 2) \times (2N + 2)$ square matrix while F and U are $(2N + 2) \times 1$ column vectors defined by

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, U = \begin{bmatrix} u \\ v \end{bmatrix}, F = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \tag{39}$$

where

$$\begin{aligned} u &= [u(x_0), u(x_1), \dots, u(x_{N-1}), u(x_N)]^T, \\ v &= [v(x_0), v(x_1), \dots, v(x_{N-1}), v(x_N)]^T, \\ f_1 &= [f_1(x_0), f_1(x_1), \dots, f_1(x_{N-1}), f_1(x_N)]^T, \\ f_2 &= [f_2(x_0), f_2(x_1), \dots, f_2(x_{N-1}), f_2(x_N)]^T, \\ A_{11} &= \mathcal{D}^2 + \text{diag} [\sin(x)] + 4L_{[1]} + 4L_{[\cos(x)]}, \\ A_{12} &= -\mathcal{D} + \text{diag} [1] + L_{[1]}, \\ A_{21} &= -2 \text{diag} [\sin(x)] + 4L_{[\sin(x)]}, \\ A_{22} &= 2\mathcal{D}^2 + \text{diag} [\cos(x)] - 2 \text{diag} [\cos(x)]L_{[1]}. \end{aligned} \tag{40}$$

In the above definitions, $[1]$ is a column vector whose entries are 1's. The boundary conditions (35) are imposed on Equation (38) by modifying the first and last rows of A_{mn} , ($m, n = 1, 2$) and f_1, f_2 in such a way that the modified matrices A and F take the form

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & A_{11} & & & & & A_{12} & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ & A_{21} & & & & & A_{22} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ f_1 \\ 0 \\ 1 \\ f_2 \\ -1 \end{bmatrix}, \tag{41}$$

After modifying the matrix system (38) to incorporate boundary conditions, the solution is obtained as

$$U = \tilde{A}^{-1} \tilde{F}, \tag{42}$$

where \tilde{A} and \tilde{F} are the the modified matrices of A and F , respectively.

4. Results and Discussion

In this section, we give the results obtained by applying the proposed algorithm on various linear Volterra integro-differential equations. Implementation of the numerical schemes was performed using personal computer of 2.5 GHz CPU speed including Matlab2017 package to perform the simulation results. The accuracy of the method is demonstrated by presenting infinity error between exact and approximate solutions computed as

$$\text{Error} = |u_E(x) - u_N(x)|, \tag{43}$$

where $u_E(x)$ and $u_N(x)$ are the exact and numerical solutions, respectively. To check the validity and accuracy of the current algorithm, we made comparisons between the obtained results with the exact solutions and also with those other methods reported in the literature. The results of our solutions for all introduced examples in this report are showed and discussed in tables and graphs.

Table 1 shows absolute errors of Example 1 for $N = 5, 15, 40$, and 60 at selected values of x between 0 and 1 . Also, the time taken for the computation has been presented at the bottom of the table. The absolute errors for various values of number of grid points N are shown in Figure 2. From Table 1 and Figure 2, clearly that the maximum absolute errors are generally very small and results obtained by the current algorithm are almost the same as the results of the exact solution, it is also noted that when increasing the number of grid points N does not result in a significant improvement in the accuracy of the current approximation. It is clear that from Table 1, the present algorithm is computationally fast as accurate results are generated in a fraction of a second as it is shown in bottom of the table.

In Table 2, the exact errors of Example 2 have been calculated for various x in $[0, \pi/2]$ to measure the extent of agreement between the exact and approximate solution. We also presented the time taken for the computation at the end of the table. From the table, it can be seen that the current algorithm provides us with the accurate approximate solution of Example 2. We remark that increasing the

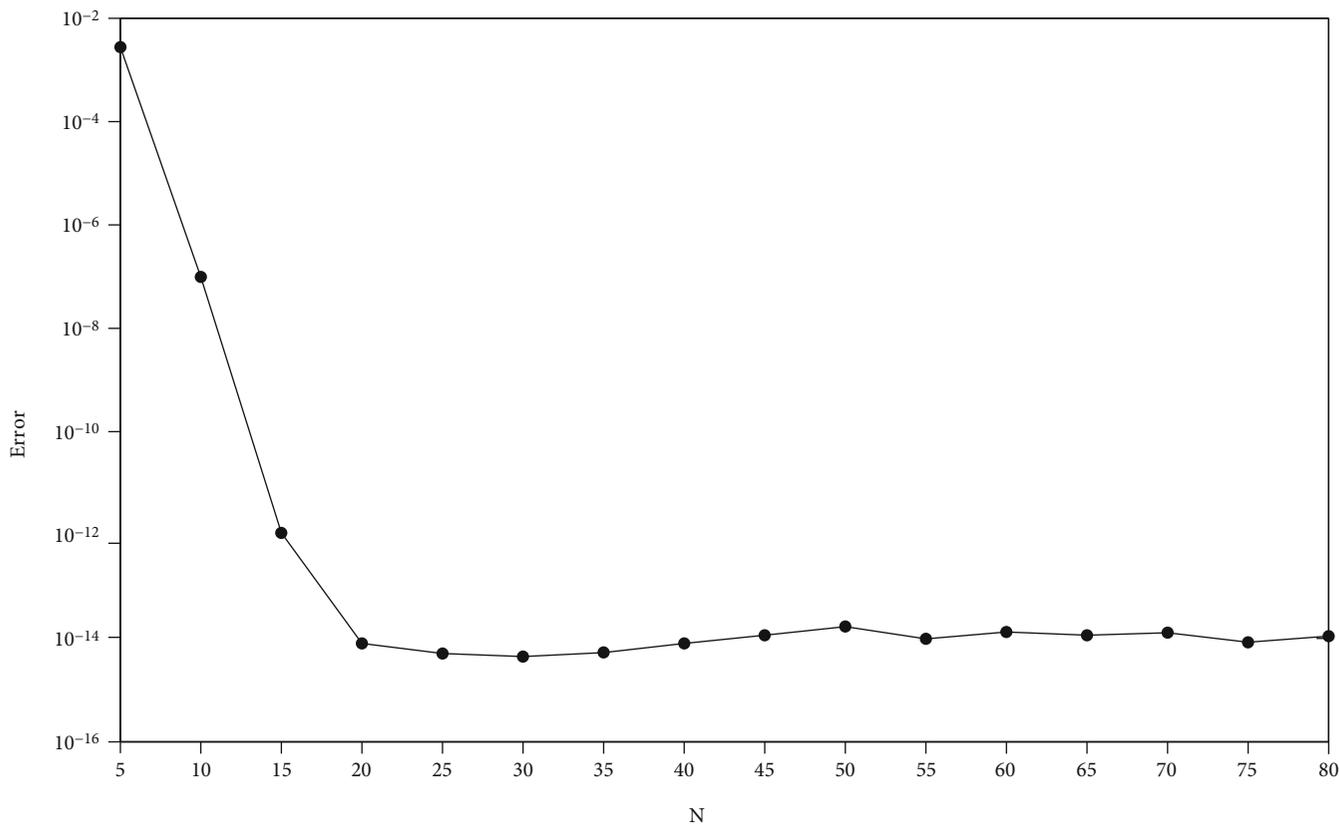


FIGURE 4: Absolute errors of $u(x)$ varied N for Example 3.

TABLE 5: Maximum absolute errors of Example 3 for different values of x and N .

x	$N = 5$	$N = 10$	$N = 15$	$N = 20$
0.1	$2.734e - 003$	$1.492e - 009$	$1.014e - 012$	$1.998e - 015$
0.2	$1.252e - 003$	$8.575e - 008$	$3.511e - 013$	$1.998e - 015$
0.3	$5.545e - 005$	$1.460e - 008$	$3.710e - 013$	$4.663e - 015$
0.4	$3.250e - 005$	$1.924e - 008$	$4.376e - 013$	$2.665e - 015$
0.5	$9.964e - 004$	$6.703e - 008$	$3.826e - 013$	$2.887e - 015$
0.6	$1.799e - 003$	$2.264e - 008$	$3.264e - 013$	$3.553e - 015$
0.7	$1.802e - 003$	$2.075e - 008$	$4.197e - 013$	$4.885e - 015$
0.8	$1.239e - 003$	$5.380e - 008$	$4.334e - 013$	$5.107e - 015$
0.9	$1.019e - 003$	$3.501e - 008$	$3.877e - 013$	$6.217e - 015$
1.0	$1.507e - 003$	$5.156e - 008$	$5.485e - 013$	$0.001e - 015$
CPU				
Time (sec)	0.1100	0.1651	0.2346	0.3054

number of nodes (i.e., increasing N) does not result in a significant improvement in the accuracy of the approximation.

We also compare the relative error estimates with those obtained using the homotopy analysis method (HAM) reported by El-Ajou et al. [28] for x in $[0,1]$ in Table 3. Full convergence upto 7 decimal places accurate results reported by HAM is achieved for $x \geq 0.8$, and an increase in the values

of x results in an increase in the maximum error form, while high convergence is achieved to 15 decimal places accurate results for the present method for all values of x ; also, increasing in nodes (N) leads to decrease in the accuracy of HAM for Example 2 while in increasing N does not result in a significant improvement in the accuracy using the current approximation. One can note that the present method gives

TABLE 6: Maximum absolute errors of $u(x)$ and $v(x)$ for Example 4 at various values of x and N .

x	$N = 10$	$N = 15$	$N = 18$	$N = 20$
$u(x)$				
0.1	$8.510e - 007$	$1.599e - 012$	$2.479e - 014$	$2.718e - 013$
0.2	$1.642e - 006$	$4.224e - 012$	$4.630e - 014$	$3.248e - 013$
0.3	$2.383e - 006$	$1.176e - 011$	$1.429e - 013$	$2.869e - 013$
0.4	$3.053e - 006$	$2.435e - 011$	$2.323e - 013$	$7.092e - 014$
0.5	$3.628e - 006$	$3.864e - 011$	$3.179e - 013$	$3.609e - 013$
0.6	$4.092e - 006$	$5.321e - 011$	$3.849e - 013$	$4.597e - 013$
0.7	$4.437e - 006$	$6.899e - 011$	$4.263e - 013$	$5.213e - 013$
0.8	$4.661e - 006$	$8.703e - 011$	$4.757e - 013$	$5.907e - 013$
0.9	$4.764e - 006$	$1.068e - 010$	$5.748e - 013$	$6.329e - 013$
$v(x)$				
0.1	$7.275e - 007$	$1.339e - 011$	$7.572e - 014$	$1.830e - 013$
0.2	$1.359e - 006$	$2.661e - 011$	$2.573e - 013$	$4.627e - 013$
0.3	$2.005e - 006$	$4.003e - 011$	$1.514e - 013$	$4.847e - 013$
0.4	$2.631e - 006$	$5.329e - 011$	$3.753e - 014$	$6.117e - 013$
0.5	$3.174e - 006$	$6.627e - 011$	$5.973e - 014$	$4.596e - 014$
0.6	$3.590e - 006$	$7.917e - 011$	$1.321e - 013$	$9.970e - 014$
0.7	$3.871e - 006$	$9.223e - 011$	$1.910e - 013$	$5.933e - 013$
0.8	$4.034e - 006$	$1.056e - 010$	$2.416e - 013$	$8.356e - 013$
0.9	$4.112e - 006$	$1.194e - 010$	$3.022e - 013$	$7.929e - 013$
CPU				
Time (sec)	0.1262	0.1574	0.1769	0.1967

much better numerical results compared to HAM. Figure 3 shows the errors between the exact and numerical solutions for Example 2 varied N . Very small errors are achieved ($10^{-14} < \text{Error} < 10^{-12}$) for the values $N \geq 15$, i.e., that accuracy does not improved when for large values of N .

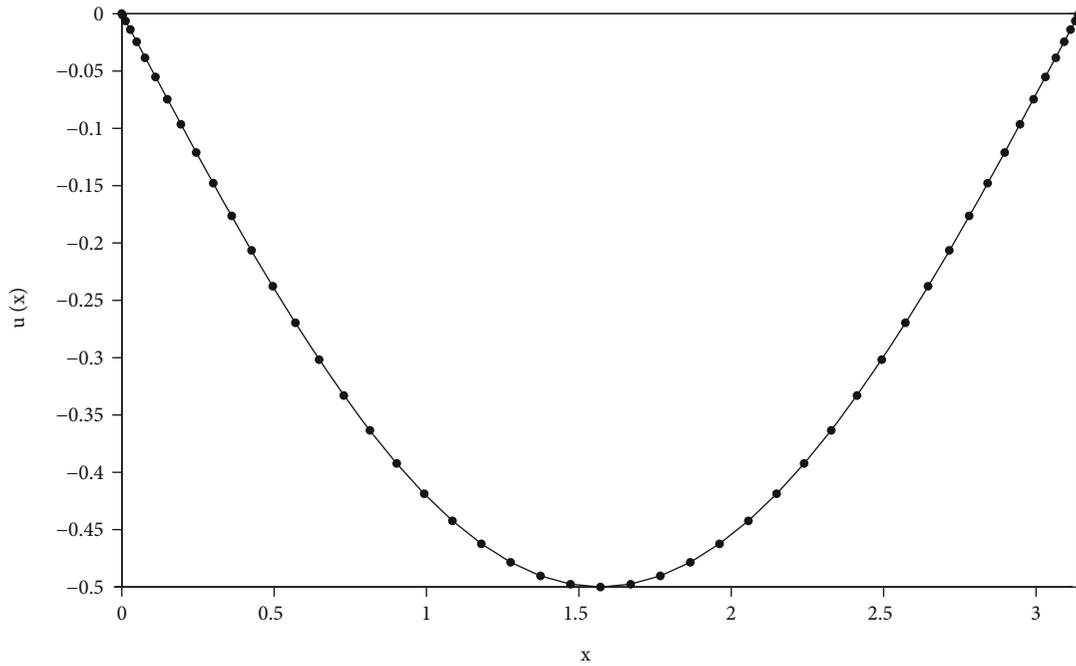
In Table 4, the absolute error is obtained for Example 3 using current method, Tau method investigated by Hosseini and Shahmorad [26], and practical matrix method obtained by Yüzbaşı et al. [27]. It is seen from the table that the results obtained by the present method is better than those obtained by Tau and practical matrix methods; also, it is noticed from Table 4 that when x increased, the error is increased for Tau and practical matrix methods while in the present method the error is approximately fixed for all values of x in the domain. Figure 4 showed the error of the numerical results against the nodes N of Example 3. It can be seen from this graph that when the number of nodes (N) increase, the error becomes smaller, and then, no effect when $N \geq 15$. In Table 5, we give the maximum errors between the exact and present results of Example 3; the results were computed in the domain $0.1 \leq x \leq 1$. To give a sense of the computational efficiency of the method, the computational time to generate the results is also given. The accuracy is seen to improve with an increase in the number of collocation points N . It is observed that accurate results with errors of order up to 10^{-15} are obtained using very few collocation

points. We remark, also, that this algorithm is computationally fast as accurate results are generated in a fraction of a second.

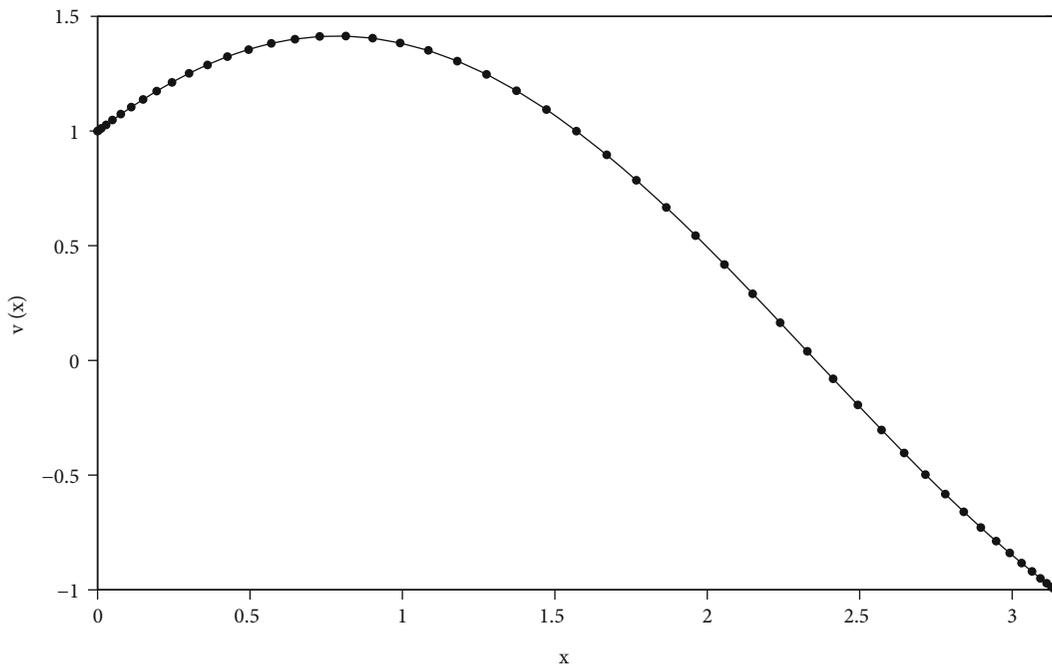
Table 6 and Figure 5 are showed the maximum absolute errors of the approximate solutions for Example 4 at selected values of x and N . The maximum absolute errors are very small for $u(x)$ and $v(x)$ and further decrease with an increase in the values of N upto $N = 18$, and then, increasing N does not result in a significant improvement in the accuracy of the approximate solution. We also observed that the current algorithm is computationally fast as accurate results are generated in a fraction of a second even for large values on N . Figure 5 shows a comparison between the exact and approximate solutions for Example 4 of $u(x)$ and $v(x)$. We observe that there is good agreement between the two sets of results, proving the efficiency of the proposed approach.

In Figure 6, the maximum absolute errors for Example 4 is presented. The figures display a variation of the errors at a various values of N . It can be seen that, in the problem considered, the algorithm needs small values of N to achieve good currency and to converge fully between the exact and approximate results. Also, we notice that there is no significant improvement in the accuracy when the number of nodes (N) is increasing.

A comparison between numerical results with the exact solution for Example 4 is shown in Figure 5. We can see that



(a)



(b)

FIGURE 5: Comparison of the exact solutions (solid line) with the numerical solutions (filled circles) solutions of (a) $u(x)$ and (b) $v(x)$ for Example 4.

the two graphs overlap indicating the very good accuracy of the method. In Figure 6, convergence analysis graphs for Example 4 are presented. The figures present a varia-

tion of the error at various values of collocation points N . It can be seen that, in almost all, the iteration scheme starts at $N = 16$ to converge fully. It is remarkable to note

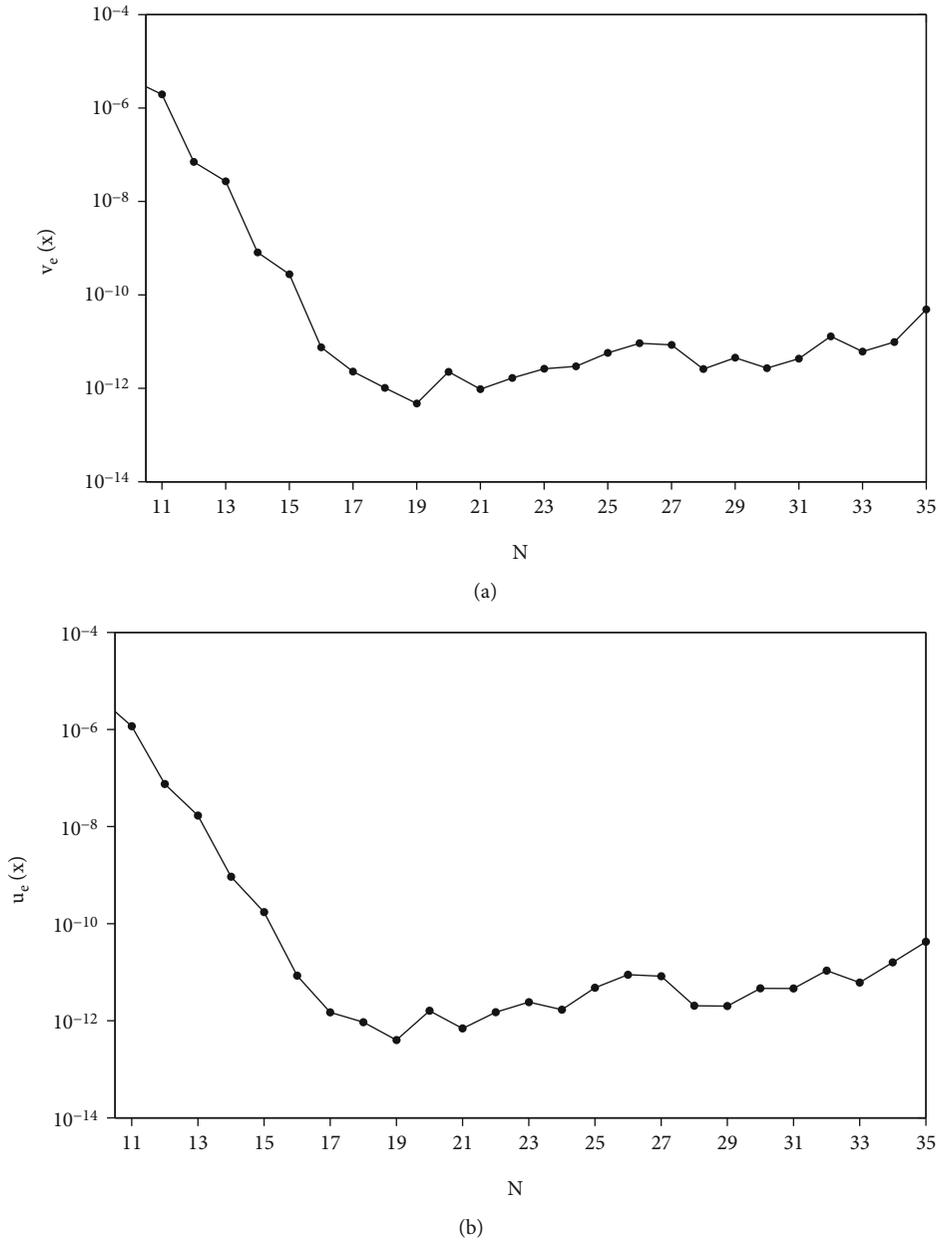


FIGURE 6: Maximum absolute errors of (a) $u(x)$ and (b) $v(x)$ at different values of N for Example 4.

accurate results with errors of order up to 10^{-10} . It is worth remarking that the accuracy of the method is not dependent on the number of collocation points.

All the above observations are clear indication that this algorithm is a powerful method that is appropriate in solving linear Volterra integro-differential equations.

5. Conclusion

In this study, we have proposed a simple and easy algorithm for solving linear Volterra integro-differential equations with boundary conditions. This technique has been constructed by introducing a new integral operator together with Chebyshev pseudospectral method. The method allows us to

choose the integral operators in the integro-differential equations in terms of the Chebyshev spectral differentiation matrix. The numerical solutions have been shown in tables and graphs and compared with the exact solutions and other methods in the literature. The accuracy of the obtained result is a point of interest; moreover, if we increase the number of nodes approximation, the numerical solution get closer to the exact solutions. The fast convergence and accuracy of the proposed algorithm are valid reasons for researcher to use this method for different problems of Volterra integro-differential equations arising in various areas of science.

The main conclusions emerging from this study may be summarized as follows:

- (1) The proposed method is highly accurate and efficient, converges rapidly, very easy to understand, and is sufficient to give good agreement with the exact solution
- (2) The suggested technique does not depend on the number of collocation points, and it requires a few numbers of collocation points to achieve high accuracy of results
- (3) No any iteration is required to achieve the currency while other methods required more iterations
- (4) The method is more flexible than other numerical methods in the literature such as HAM, Tau method, and practical matrix method

Finally, for the future study, we suggest that how is the nonseparable kernel dealt with and also how to apply this technique on nonlinear Volterra and Fredholm integral equations.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] J. R. Loh and C. Phang, "A new numerical scheme for solving system of Volterra integro-differential equation," *Alexandria Engineering Journal*, vol. 57, no. 2, pp. 1117–1124, 2018.
- [2] J. R. Loh, C. Phang, and K. G. Tay, "New method for solving fractional partial integro-differential equations by combination of Laplace transform and resolvent kernel method," *Chinese Journal of Physics*, vol. 67, pp. 666–680, 2020.
- [3] F. Mirzaee, E. Solhi, and S. Naserifar, "Approximate solution of stochastic Volterra integro-differential equations by using moving least squares scheme and spectral collocation method," *Applied Mathematics and Computation*, vol. 410, article 126447, 2021.
- [4] Y. Zhou and M. Stynes, "Block boundary value methods for solving linear neutral Volterra integro-differential equations with weakly singular kernels," *Journal of Computational and Applied Mathematics*, vol. 401, article 113747, 2022.
- [5] A. O. Adewumi, S. O. Akindeinde, and R. S. Lebelo, "Sumudu Lagrange-spectral methods for solving system of linear and nonlinear Volterra integro-differential equations," *Applied Numerical Mathematics*, vol. 169, pp. 146–163, 2021.
- [6] A. M. Wazwaz, "The variational iteration method for solving linear and nonlinear Volterra integral and integro-differential equations," *International Journal of Computer Mathematics*, vol. 87, no. 5, pp. 1131–1141, 2010.
- [7] H. Rezazadeh, M. S. Osman, M. Eslami et al., "Hyperbolic rational solutions to a variety of conformable fractional Boussinesq-like equations," *Nonlinear Engineering*, vol. 8, no. 1, pp. 224–230, 2019.
- [8] M. S. Osman, H. Rezazadeh, and M. Eslami, "Traveling wave solutions for (3+1) dimensional conformable fractional Zakharov-Kuznetsov equation with power law nonlinearity," *Nonlinear Engineering*, vol. 8, no. 1, pp. 559–567, 2019.
- [9] M. S. Osman, "New analytical study of water waves described by coupled fractional variant Boussinesq equation in fluid dynamics," *Pramana*, vol. 93, no. 2, p. 26, 2019.
- [10] M. S. Osman, D. Lu, and M. M. A. Khater, "A study of optical wave propagation in the nonautonomous Schrodinger-Hirota equation with power-law nonlinearity," *Results in Physics*, vol. 13, article 102157, 2019.
- [11] M. S. Osman and A. M. Wazwaz, "A general bilinear form to generate different wave structures of solitons for a (3+1)-dimensional Boiti-Leon-Manna-Pempinelli equation," *Mathematical Methods in the Applied Sciences*, vol. 42, no. 18, pp. 6277–6283, 2019.
- [12] M. Higazy, S. Aggarwal, and T. A. Nofal, "Sawi decomposition method for Volterra integral equation with application," *Journal of Mathematics*, vol. 2020, Article ID 6687134, 13 pages, 2020.
- [13] M. Lotfi and A. Alipanah, "Legendre spectral element method for solving Volterra-integro differential equations," *Results in Applied Mathematics*, vol. 7, pp. 100116–100116, 2020.
- [14] H. Alrabaiah, M. Jamil, K. Shah, and R. A. Khan, "Existence theory and semi-analytical study of non-linear Volterra fractional integro-differential equations," *Alexandria Engineering Journal*, vol. 59, no. 6, pp. 4677–4686, 2020.
- [15] N. Nana, S. C. O. Noutchie, and R. Y. Massoukou, "A second order finite difference scheme for singularly perturbed Volterra integro-differential equation," *Alexandria Engineering Journal*, vol. 59, no. 4, pp. 2441–2447, 2020.
- [16] A. A. Khidir and S. L. Alsharari, "Application of successive linearisation method on the boundary layer flow problem of heat and mass transfer with radiation effect," *International Journal of Analysis and Applications*, vol. 19, no. 5, pp. 725–742, 2021.
- [17] A. A. Khidir, "A new numerical technique for solving Volterra integral equations using Chebyshev spectral method," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9230714, 11 pages, 2021.
- [18] Y. Daoud, M. Abdalbagi, and A. A. Khidir, "On the solution of magneto-hydrodynamics three-dimensional flow due to a stretching sheet in a porous medium using the successive linearization method," *Chinese Journal of Physics*, vol. 73, pp. 232–238, 2021.
- [19] A. A. Khidir and A. F. Aljohani, "On successive linearization method for differential equations with nonlinear conditions," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 23, 2021.
- [20] A. A. Khidir and W. I. Alfaifi, "Application of successive linearisation method on mixed convection from an exponentially stretching surface with magnetic field effect," *International Journal of Scientific Research in Mathematical and Statistical Sciences*, vol. 8, no. 6, pp. 13–19, 2021.
- [21] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, Berlin, 1988.
- [22] W. S. Don and A. Solomonoff, "Accuracy and speed in computing the Chebyshev collocation derivative," *SIAM Journal on Scientific Computing*, vol. 16, no. 6, pp. 1253–1268, 1995.
- [23] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.
- [24] R. Larson and D. C. Falve, *Elementary Linear Algebra*, Houghton mifflin harcourt publishing company Boston, New York, NY, USA, 6th edition, 2008.

- [25] S. Cravero, G. Pittaluga, and L. Sacripante, "An algorithm for solving linear Volterra integro-differential equations," *Numerical Algorithms*, vol. 60, no. 1, pp. 101–114, 2012.
- [26] S. M. Hosseini and S. Shahmorad, "Numerical solution of a class of integro-differential equations by the Tau method with an error estimation," *Applied Mathematics and Computation*, vol. 136, no. 2-3, pp. 559–570, 2003.
- [27] Ş. Yüzbaşı, N. Şahin, and M. Sezer, "Bessel polynomial solutions of high-order linear Volterra integro-differential equations," *Computers and Mathematics with Applications*, vol. 62, no. 4, pp. 1940–1956, 2011.
- [28] A. El-Ajou, O. Abu Arqub, and S. Momani, "Homotopy analysis method for second-order boundary value problems of integrodifferential equations," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 365792, 18 pages, 2012.