Galerkin Method for Nonlinear Volterra-Fredholm Integro-Differential Equations Based on Chebyshev Polynomials

W. Abbas[1,*], Mohamed Fathy[1], M. Mostafa[2], Hesham A. M. A[2]

[1] Basic and applied science department, college of engineering and technology, Arab academy for science, technology and maritime transport, Cairo, Egypt.
[2] Physics and engineering mathematics department, Helwan University, Cairo, Egypt.

Abstract

We aim in this paper to develop a new algorithm for approximating the analytic solution for the integro-differential equations using the Galerkin method. The bases of the solution obtained by the proposed algorithm are Chebyshev polynomials. Meanwhile, some theorems are deducted to simplify the nonlinear algebraic set resulted from applying the Galerkin method, while Newton's method is used to solve the resulting nonlinear algebraic system. Examples are introduced to prove the effectiveness of this algorithm in comparison with some other methods.

Keywords Integro-differential equations, Chebyshev polynomials, Newton's method, Gauss quadrature, Volterra, Fredholm.

1. Introduction

The integro-differential equations stem from the mathematical modeling of many complex real-life problems. Many scientific phenomena have been formulated using integro-differential equations. The integro-differential equations can be encountered in various fields of science such as physics, chemistry, biology, and engineering. These kinds of equations can also be found in numerous applications, such as the theory of elasticity, biomechanics, electromagnetic, electrodynamics, fluid dynamics, heat and mass transfer, oscillating magnetic field…, etc. [1].

Many researchers introduced methods to solve integro-differential equations for example Rahimkhani et al. [2] proposed the numerical solution of linear and nonlinear fractional integro-differential equations using a new set of functions called fractional-order Bernoulli functions. Aydogan et al. [3] used the fractional Caputo–Fabrizio derivative to solve the higher order fractional integro-differential equations. Erfanian et al. [4] used a new sequential approach for solving the integro-differential equation via Haar wavelet bases. Rong et al. [5] applied a new operational matrix via Genocchi polynomials to solve fractional integro-differential equations. Numerous works have been focusing on the development of more advanced and efficient methods for nonlinear Volterra-Fredholm integral and integro-differential equations. For example, hybrid Legendre polynomials and Block-Pulse functions [6], triangular functions [7], Taylor polynomials [8] [9], least squares method and Chebyshev polynomials [10], collocation method and radial basis functions [11], Taylor collocation method [12], least squares approximation method [13], shifted Legendre polynomials approximation [14], fixed point technique and Schauder bases [15], Haar wavelets [16], operational matrix with block-pulse functions [17], homotopy analysis method [18].

In this paper, we try to find an appropriate algorithm to approximate the analytic solution of integro-differential equations, with a good accuracy and high rate of convergence to the exact solution. Our proposed algorithm is based on the Chebyshev polynomials as bases in applying the Galerkin

*Corresponding Author : wael_abass@aast.edu

method to find the approximate solution of the aimed integro-differential equation to be solved that is in the form

$$\sum_{l=0}^{N} \mu_l(x) u^{(l)}(x) + \zeta(x) u(x) u'(x) = f(x) + \int_a^x k_1(x,t) u^{\alpha}(t) dt + \int_a^b k_2(x,t) u^{\beta}(t) dt, \quad a \le x \le b, \quad (1.1)$$

with conditions

$$u^{(p_s)}(x_s) = q_s, \quad s = 1,\ 2,\ 3,\ \dots,\ N, \quad p_s, \alpha, \beta \in \square,$$

where $\mu_l(x)$, $\zeta(x)$, $k_1(x,t)$ and $k_2(x,t)$ are continuous functions in L^2 space, and $p_s < N$. Paper is organized as following: in Section 2, we summarize the Chebyshev polynomials and their properties, in addition to developed theorems to support our approach. Section 3 will show the analysis made to the integro-differential equation and the sequence followed by the algorithm to find the approximate solution of the equation. Section 4 is presenting the numerical examples used to prove the usefulness of the algorithm with results compared to other methods. Section 5 will give a brief conclusion.

2. Chebyshev Functions Preliminaries

Orthogonal polynomials are used in many applications such as solving partial differential equations, integral equations…etc. One of these polynomial sets is the first kind of Chebyshev polynomials set $T_n(x)$ which is the solution set of Chebyshev differential equation defined as

$$\left(1 - x^2\right) y'' - x y' + n^2 y = 0, \quad -1 \le x \le 1, \qquad (2.1)$$

and its solution defined in the form

$$T_n(x) = \cos\left(n \cos^{-1}(x)\right), \quad -1 \le x \le 1, \qquad (2.2)$$

or

$$T_n(x) = \sum_{k=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \binom{n}{2k} \left(x^2 - 1\right)^k x^{n-2k}, \quad -1 \le x \le 1,, \qquad (2.3)$$

with inverse

$$x^n = 2^{1-n} \sum_{\substack{j=0 \\ n-j \text{ even}}}^{n}{}' \binom{n}{\frac{n-j}{2}} T_j(x), \quad -1 \le x \le 1, \qquad (2.4)$$

and the extreme values of these polynomials at endpoints of interval of definition are

$$T_n(\pm 1) = (\pm 1)^n \qquad (2.5)$$

The Chebyshev polynomials and their derivatives are defined by the following recurrence relations

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x), \qquad (2.6)$$

$$T_n'(x)=\begin{cases}2n\left[T_{n-1}(x)+T_{n-3}(x)+T_{n-5}(x)+\cdots+T_1(x)\right], & n \text{ even},\\ 2n\left[T_{n-1}(x)+T_{n-3}(x)+T_{n-5}(x)+\cdots+T_2(x)\right]+nT_0(x), & n \text{ odd},\end{cases} \tag{2.7}$$

and the derivatives of the Chebyshev polynomials are given as

$$\frac{d^p}{dx^p}T_n(x)=\frac{2^{2-p}n}{\Gamma(p)}\sum_{\substack{k\geq 0\\ n-p-k \text{ even}}}^{n-p}{}' \frac{\prod_{i=1}^{\frac{n+p-k-2}{2}}\left[(n+p-2i)^2-k^2\right]}{\prod_{i=0}^{\frac{n-p-k-2}{2}}\left[(n-p-2i)^2-k^2\right]}T_k(x), \quad p\geq 1. \tag{2.8}$$

The product of two Chebyshev polynomials is given by

$$T_m(x)\,T_n(x)=\frac{1}{2}\left(T_{n+m}(x)+T_{|n-m|}(x)\right). \tag{2.9}$$

These polynomials are orthogonal on the interval $[-1,1]$ with the weight function $\frac{1}{\sqrt{1-x^2}}$ where

$$\int_{-1}^{1}T_m(x)T_n(x)\frac{dx}{\sqrt{1-x^2}}=\begin{cases}0, & n\neq m,\\ \pi, & n=m=0,\\ \pi/2, & n=m\neq 0,\end{cases} \tag{2.10}$$

also, the integration $\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}dx$ could be computed using Gaussian-Chebyshev quadrature method as following

$$\int_{-1}^{1}\frac{f(x)}{\sqrt{1-x^2}}dx\approx\frac{\pi}{n}\sum_{i=1}^{n}f(x_i), \tag{2.11}$$

where x_i are the nodes of Chebyshev polynomials given by

$$x_i=\cos\left(\frac{2i-1}{2n}\pi\right), \quad i=1,\ldots,n. \tag{2.12}$$

The integration of the first kind of Chebyshev polynomials is given by

$$\int T_n(x)\,dx=\begin{cases}\frac{1}{2}\left(\frac{T_{n+1}(x)}{n+1}-\frac{T_{n-1}(x)}{n-1}\right)+c, & n\neq 1\\ \frac{T_2(x)}{4}+c, & n=1\end{cases} \tag{2.13}$$

Theorem 2.1 *For n, m, p and q are positive integers such that, n and $m \le N$*

$$\int_{-1}^{1} x^q T_n^{(p)}(x) T_m(x) \frac{dx}{\sqrt{1-x^2}} = \sum_{\substack{j=0 \\ q-j \\ \text{even}}}^{q}{}' \sum_{\substack{k\ge 0 \\ n-p-k \\ \text{even}}}^{n-p}{}' S_{n,p,k,q,j}\left[\delta_1(k,j,m)+\delta_2(k,j,m)\right], \quad p\ge 1, \tag{2.14}$$

where

$$\delta_1(k,j,m)=\begin{cases} 0, & k+j\ne m, \\ \pi, & k=j=m=0, \\ \pi/2, & k+j=m\ne 0, \end{cases} \qquad \delta_2(k,j,m)=\begin{cases} 0, & |k-j|\ne m, \\ \pi, & |k-j|=m=0, \\ \pi/2, & |k-j|=m\ne 0, \end{cases}$$

and

$$S_{n,p,k,q,j} = \frac{2^{2-p-q} n}{\Gamma(p)} \binom{q}{\frac{q-j}{2}} \frac{\prod_{i=1}^{\frac{n+p-k-2}{2}} \left[(n+p-2i)^2 - k^2\right]}{\prod_{i=0}^{\frac{n-p-k-2}{2}} \left[(n-p-2i)^2 - k^2\right]}, \quad p\ge 1.$$

Proof: By recalling (2.4) and (2.8), $x^q T_n^{(p)}(x)$ could be written as

$$x^q T_n^{(p)}(x) = \frac{2^{3-p-q} n}{\Gamma(p)} \sum_{\substack{j=0 \\ q-j \text{ even}}}^{q}{}' \sum_{\substack{k\ge 0 \\ n-p-k \text{ even}}}^{n-p}{}' \binom{q}{\frac{q-j}{2}} \frac{\prod_{i=1}^{\frac{n+p-k-2}{2}} \left[(n+p-2i)^2 - k^2\right]}{\prod_{i=0}^{\frac{n-p-k-2}{2}} \left[(n-p-2i)^2 - k^2\right]} T_k(x) T_j(x), \quad p\ge 1. \tag{2.15}$$

The left hand side of (2.14) could be introduced as

$$\int_{-1}^{1} \frac{x^q T_n^{(p)}(x) T_m(x)\, dx}{\sqrt{1-x^2}} = \frac{2^{3-p-q} n}{\Gamma(p)} \sum_{\substack{j=0 \\ q-j \text{ even}}}^{q}{}' \sum_{\substack{k\ge 0 \\ n-p-k \text{ even}}}^{n-p}{}' \binom{q}{\frac{q-j}{2}} \frac{\prod_{i=1}^{\frac{n+p-k}{2}-1} \left[(n+p-2i)^2 - k^2\right]}{\prod_{i=0}^{\frac{n-p-k}{2}-1} \left[(n-p-2i)^2 - k^2\right]} \int_{-1}^{1} \frac{T_k(x) T_j(x) T_m(x)\, dx}{\sqrt{1-x^2}}.$$

By simplifying the previous equation using (2.9) and (2.10), Theorem 2.1 is proved.

Theorem 2.2 *For n, c_i, p, i, r, h, m and q are positive integers such that, n and $c_i \le N$*

$$\int_{-1}^{1} \frac{x^q T_n^{(p)}(x)}{\sqrt{1-x^2}} \prod_{i=1,\, r\ge 2}^{r} T_{c_i}(x)\, dx = 2^{1-r} \sum_{\substack{j=0 \\ q-j \text{ even}}}^{q}{}' \sum_{\substack{k\ge 0 \\ n-p-k \text{ even}}}^{n-p}{}' \sum_{\substack{h=1 \\ r\ge 2}}^{2^{r-1}} S_{n,p,k,q,j}\left[\delta_1\left(k,j,L_{r,h,c_i}\right)+\delta_2\left(k,j,L_{r,h,c_i}\right)\right], \tag{2.16}$$

where

$$\delta_1\left(k, j, L_{r,h,c_i}\right)=\begin{cases} 0, & k+j \neq L_{r,h,c_i}, \\ \pi, & k=j=L_{r,h,c_i}=0, \\ \pi/2, & k+j=L_{r,h,c_i} \neq 0, \end{cases} \qquad \delta_2\left(k, j, L_{r,h,c_i}\right)=\begin{cases} 0, & |k-j| \neq L_{r,h,c_i}, \\ \pi, & |k-j|=L_{r,h,c_i}=0, \\ \pi/2, & |k-j|=L_{r,h,c_i} \neq 0, \end{cases}$$

$$L_{r,h,c_i}=\left|c_1+\sum_{i=2}^{r}(-1)^{\Delta(h,i) \bmod 2} c_i\right|, \qquad \Delta(h,i)=\begin{cases} h-1, & i=2, \\ \left\lfloor \dfrac{\Delta(h,i-1)}{2} \right\rfloor, & i \geq 3, \end{cases}$$

Proof: By recalling (2.9) and using mathematical induction, the product $\prod_{i=1,\ r\geq 2}^{r} T_{c_i}(x)$ could be written as

$$\prod_{i=1,\ r\geq 2}^{r} T_{c_i}(x)=2^{1-r} \sum_{h=1}^{2^{r-1}} T_{\left|c_1+\sum_{i=2}^{r}(-1)^{\Delta(h,i) \bmod 2} c_i\right|}(x) \tag{2.17}$$

where

$$\Delta(h,i)=\begin{cases} h-1, & i=2, \\ \left\lfloor \dfrac{\Delta(h,i-1)}{2} \right\rfloor, & i \geq 3. \end{cases}$$

By replacing $T_m(x)$ in (2.14) by $\prod_{i=1,\ r\geq 2}^{r} T_{c_i}(x)$ in (2.17) and simplifying the result, proof is complete.

3. The Chebyshev–Galerkin Method

Let the approximate finite Chebyshev series expansion of any continuous function $f(x)$ in the interval $[-1,1]$ be in the form

$$f(x) \approx \sum_{j=0}^{n} a_j T_j(x), \tag{3.1}$$

by multiplying both sides of the equation by $\dfrac{T_i(x)}{\sqrt{1-x^2}}$ and integrating over the interval $[-1,1]$ with respect to x yields

$$\int_{-1}^{1} \frac{f(x) T_i(x)}{\sqrt{1-x^2}} dx \approx \sum_{j=0}^{n} a_j \int_{-1}^{1} \frac{T_j(x) T_i(x)}{\sqrt{1-x^2}} dx,$$

where the constant a_j could be calculated by the form

$$a_j \approx \begin{cases} \dfrac{1}{\pi}\left\langle f(x), T_j(x)\right\rangle, & j=0, \\ \dfrac{2}{\pi}\left\langle f(x), T_j(x)\right\rangle, & j\neq 0, \end{cases} \tag{3.2}$$

where the inner product $\left\langle f(x), T_j(x)\right\rangle$ is defined as

$$\left\langle f(x), T_j(x)\right\rangle = \int_{-1}^{1} \frac{f(x)T_j(x)}{\sqrt{1-x^2}}dx,$$

and could be calculated using (2.11) after replacing $f(x)$ with $f(x)T_j(x)$

$$\left\langle f(x), T_j(x)\right\rangle \approx \frac{\pi}{m}\sum_{i=1}^{m} f(x_i)T_j(x_i). \tag{3.3}$$

where x_i are the nodes of Chebyshev polynomials given by (2.12), and m is the number of nodes.

Let the approximate solution of the integro-differential equation be in the form

$$u(x) \approx u_n(x) = \sum_{j=0}^{n} c_j T_j(x). \tag{3.4}$$

By plugging the approximate solution (3.4) in the integro-differential equation (1.1), and taking $\alpha = 2$ and $\beta = 3$, the equation could be written as

$$\sum_{l=0}^{N}\sum_{j=0}^{n} c_j \mu_l(x) T_j^{(l)}(x) + \zeta(x)\left(\sum_{j=0}^{n} c_j T_j(x)\right)\left(\sum_{j=0}^{n} c_j T_j'(x)\right)$$
$$= f(x) + \int_a^x k_1(x,t)\left(\sum_{j=0}^{n} c_j T_j(t)\right)^2 dt + \int_a^b k_2(x,t)\left(\sum_{j=0}^{n} c_j T_j(t)\right)^3 dt, \tag{3.5}$$

which could be expanded as

$$\sum_{l=0}^{N}\sum_{j=0}^{n} c_j \mu_l(x) T_j^{(l)}(x) + \zeta(x)\sum_{i=0}^{n}\sum_{j=0}^{n} c_i c_j T_i(x) T_j'(x)$$
$$= f(x) + \sum_{j=0}^{n}\sum_{i=0}^{n} c_j c_i \int_a^x k_1(x,t) T_j(t) T_i(t) dt + \sum_{j=0}^{n}\sum_{i=0}^{n}\sum_{h=0}^{n} c_j c_i c_h \int_a^b k_2(x,t) T_j(t) T_i(t) T_h(t) dt.$$

By replacing the functions $\mu_l(x)$, $\zeta(x)$ and $f(x)$ by their approximate finite Chebyshev series expansion using (3.1), the integro-differential equation could be introduces as

$$\sum_{l=0}^{N}\sum_{a=0}^{n_1}\sum_{j=0}^{n} d_a c_j T_a(x) T_j^{(l)}(x) + \sum_{b=0}^{n_2}\sum_{i=0}^{n}\sum_{j=0}^{n} d_b c_i c_j T_b(x) T_i(x) T_j'(x) = \sum_{j=0}^{n_3} d_j T_j(x)$$
$$+\sum_{j=0}^{n}\sum_{i=0}^{n} c_j c_i \int_a^x k_1(x,t) T_j(t) T_i(t)\,dt + \sum_{j=0}^{n}\sum_{i=0}^{n}\sum_{h=0}^{n} c_j c_i c_h \int_a^b k_2(x,t) T_j(t) T_i(t) T_h(t)\,dt, \qquad (3.6)$$

where d_a, d_b and d_j are known constants resulting from Chebyshev series expansion (3.2), and n_1, n_2 and n_3 are chosen numbers of terms of the approximation series for each function.

Integrals in the right hand side of the equation could be approximated using Gauss-Legendre quadrature method by the form

$$\int_a^b k(x,t)\,dt \approx \frac{b-a}{2}\sum_{i=1}^{n} \frac{2}{\left(1-t_i^{\,2}\right)\left[P_n'(t_i)\right]^2} k\left(x, \frac{b-a}{2} t_i + \frac{b+a}{2}\right), \qquad (3.7)$$

where t_i is the i-th root of the Legendre polynomials $P_n(t)$ which are obtained by

$$P_n(t) = \frac{1}{2^n n!}\frac{d^n}{dt^n}\left(t^2-1\right)^n, \qquad (3.8)$$

and if the integro-differential equation contains the Volterra term, it could be transformed as

$$\int_a^x k(x,t)\,dt = \frac{x-a}{b-a}\int_a^b k(x,\eta)\,d\eta, \qquad (3.9)$$

where $t = \frac{x-a}{b-a}(\eta - a) + a$. Then the integro-differential equation could be written as

$$\sum_{l=0}^{N}\sum_{a=0}^{n_1}\sum_{j=0}^{n} d_a c_j T_a(x) T_j^{(l)}(x) + \sum_{b=0}^{n_2}\sum_{i=0}^{n}\sum_{j=0}^{n} d_b c_i c_j T_b(x) T_i(x) T_j'(x)$$
$$= \sum_{j=0}^{n_3} d_j T_j(x) + \sum_{j=0}^{n}\sum_{i=0}^{n} c_j c_i g_{i,j}(x) + \sum_{j=0}^{n}\sum_{i=0}^{n}\sum_{h=0}^{n} c_j c_i c_h y_{i,j,h}(x), \qquad (3.10)$$

where $g_{i,j}(x)$ and $y_{i,j,h}(x)$ are the resulting functions from (3.7) and (3.9).

By applying the Galerkin method using Chebyshev bases, the weak formula of the equation (3.10) is introduced as

$$\sum_{l=0}^{N}\sum_{a=0}^{n_1}\sum_{j=0}^{n} d_a c_j \left\langle T_a(x) T_j^{(l)}(x), T_r(x)\right\rangle + \sum_{b=0}^{n_2}\sum_{i=0}^{n}\sum_{j=0}^{n} d_b c_i c_j \left\langle T_b(x) T_i(x) T_j'(x), T_r(x)\right\rangle$$
$$= \sum_{j=0}^{n_3} d_j \left\langle T_j(x), T_r(x)\right\rangle + \sum_{j=0}^{n}\sum_{i=0}^{n} c_j c_i \left\langle g_{i,j}(x), T_r(x)\right\rangle + \sum_{j=0}^{n}\sum_{i=0}^{n}\sum_{h=0}^{n} c_j c_i c_h \left\langle y_{i,j,h}(x), T_r(x)\right\rangle, \qquad (3.11)$$

where the inner products in the left hand side of the equation are calculated using (2.16), and the inner products in the right hand side of the equation are calculated using (3.3).

This would result in the algebraic nonlinear system of equations that could be written as

$$\mathbf{A}\mathbf{c}+\mathbf{B}\tilde{\mathbf{c}}=\mathbf{E}+\mathbf{D}\tilde{\tilde{\mathbf{c}}} \tag{3.12}$$

where

$$\mathbf{A}=\begin{pmatrix} \sum_{l=0}^{N}\sum_{a=0}^{n_1} d_a\left\langle T_a(x)T_0^{(l)}(x),T_0(x)\right\rangle & \cdots & \sum_{l=0}^{N}\sum_{a=0}^{n_1} d_a\left\langle T_a(x)T_n^{(l)}(x),T_0(x)\right\rangle \\ \vdots & \ddots & \vdots \\ \sum_{l=0}^{N}\sum_{a=0}^{n_1} d_a\left\langle T_a(x)T_0^{(l)}(x),T_{n-N}(x)\right\rangle & \cdots & \sum_{l=0}^{N}\sum_{a=0}^{n_1} d_a\left\langle T_a(x)T_n^{(l)}(x),T_{n-N}(x)\right\rangle \\ T_0^{(p_1)}(x_1) & \cdots & T_n^{(p_1)}(x_1) \\ \vdots & \ddots & \vdots \\ T_0^{(p_N)}(x_N) & \cdots & T_n^{(p_N)}(x_N) \end{pmatrix},$$

$$\mathbf{B}=\begin{pmatrix} \left\langle \sum_{b=0}^{n_2} d_b T_b(x)T_0(x)T_0'(x)-g_{0,0}(x),T_0(x)\right\rangle & \cdots & \left\langle \sum_{b=0}^{n_2} d_b T_b(x)T_n(x)T_n'(x)-g_{n,n}(x),T_0(x)\right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle \sum_{b=0}^{n_2} d_b T_b(x)T_0(x)T_0'(x)-g_{0,0}(x),T_{n-N}(x)\right\rangle & \cdots & \left\langle \sum_{b=0}^{n_2} d_b T_b(x)T_n(x)T_n'(x)-g_{n,n}(x),T_{n-N}(x)\right\rangle \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix},$$

$$\mathbf{D}=\begin{pmatrix} \left\langle y_{0,0,0}(x),T_0(x)\right\rangle & \cdots & \left\langle y_{n,n,n}(x),T_0(x)\right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle y_{0,0,0}(x),T_{n-N}(x)\right\rangle & \cdots & \left\langle y_{n,n,n}(x),T_{n-N}(x)\right\rangle \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix},$$

$$\mathbf{E}=\left[\sum_{j=0}^{n_3} d_j\left\langle T_j(x),T_0(x)\right\rangle \quad \sum_{j=0}^{n_3} d_j\left\langle T_j(x),T_1(x)\right\rangle \quad \cdots \quad \sum_{j=0}^{n_3} d_j\left\langle T_j(x),T_{n-N}(x)\right\rangle \quad q_1 \quad \cdots \quad q_{N-1} \quad q_N\right]^T,$$

$$\mathbf{c}=\left[c_0 \quad c_1 \quad \cdots \quad c_{n-1} \quad c_n\right]^T,$$

$$\tilde{\mathbf{c}}=\left[c_0c_0 \quad c_1c_0 \quad \cdots \quad c_{n-1}c_n \quad c_nc_n\right]^T,$$

and

$$\tilde{\tilde{\mathbf{c}}} = \begin{bmatrix} c_0 c_0 c_0 & c_1 c_0 c_0 & \cdots & c_{n-1} c_n c_n & c_n c_n c_n \end{bmatrix}^T.$$

By obtaining the coefficients values, the approximate solution (3.1) of the integro-differential equation (1.1) is found, and our algorithm is complete.

Newton's Method

The system (3.10) could be written in the form

$$\mathbf{F}(\mathbf{c}) = \begin{bmatrix} g_0(\mathbf{c}) & g_1(\mathbf{c}) & \cdots & g_n(\mathbf{c}) \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^T, \tag{3.13}$$

where $\mathbf{c}$ is the coefficients column matrix. The Jacobian of the given system as

$$\mathbf{J}(\mathbf{c}) = \begin{pmatrix} \frac{\partial g_0}{\partial c_0}(\mathbf{c}) & \frac{\partial g_0}{\partial c_1}(\mathbf{c}) & \cdots & \frac{\partial g_0}{\partial c_n}(\mathbf{c}) \\ \frac{\partial g_1}{\partial c_0}(\mathbf{c}) & \frac{\partial g_1}{\partial c_1}(\mathbf{c}) & \cdots & \frac{\partial g_1}{\partial c_n}(\mathbf{c}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial c_0}(\mathbf{c}) & \frac{\partial g_n}{\partial c_1}(\mathbf{c}) & \cdots & \frac{\partial g_n}{\partial c_n}(\mathbf{c}) \end{pmatrix}, \tag{3.14}$$

and by using Taylor's theorem for approximation of multivariable functions neglecting higher-order terms, one can easily derive the recursive relation

$$\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} - \mathbf{J}^{-1}\left(\mathbf{c}^{(i)}\right)\mathbf{F}\left(\mathbf{c}^{(i)}\right). \tag{3.15}$$

Iterations would be stopped when the allowable tolerance is greater than the absolute maximum difference between two successive iterations as following

$$\text{Error} \geq \max\left\|\mathbf{c}^{(i+1)} - \mathbf{c}^{(i)}\right\|. \tag{3.16}$$

4. Numerical Examples

This section demonstrates the reliability and efficiency of our proposed algorithm, four numerical examples of nonlinear integro-differential equations with boundary conditions of higher-order are performed. The solutions of the Chebyshev-Galerkin method are compared with the exact solution or discussed with other previous methods. Calculations are done by using *Mathematica 10.0* on a personal computer. The maximum absolute error is defined as

$$\| E(x) \| = \max \| u(x) - u_n(x) \|, \quad a \leq x \leq b,$$

where $u_n(x)$ is the approximate solution and $u(x)$ is the exact solution.

Example 4.1 Considering the 1st order nonlinear integro-differential equation

$$u'(x)+u(x)=\frac{1}{10}x^6+x^2+2x-\frac{1}{32}-\frac{1}{2}\int_0^x xu^2(t)\,dt-\frac{1}{4}\int_0^1 tu^3(t)\,dt,\quad 0\le x\le 1,$$

subjected to

$$u(0)=0,$$

with the exact solution

$$u(x)=x^2.$$

Table 1 shows the absolute errors of the comparison between the introduced method and with the Haar wavelet method, the Triangular factorization method as well as the Hybrid Legendre polynomials.

Table 1 Maximum absolute error $\|E(x)\|$ corresponding to different methods for Example 4.1

Proposed method	Haar wavelets method [19]	Triangular functions method [20]	Hybrid Legendre polynomials method [21]
2.168E-19	2.200E-04	2.800E-04	9.700E-04

Example 4.2 [22] Consider the nonlinear 2nd order integro-differential equation

$$u''(x)+u(x)u'(x)=f(x)-\frac{1}{3}\int_0^x x^2tu^2(t)\,dt+\frac{1}{2}\int_0^1 xtu^3(t)\,dt,\quad 0\le x\le 1,$$

with the boundary conditions $u(0)=u(1)=1$, and the exact solution is $u(x)=x^2-x+1$ and

$$f(x)=2-\frac{83x}{560}-(-1+2x)(1-x+x^2)+\frac{1}{3}\left(\frac{x^2}{4}-\frac{2x^5}{3}+\frac{3x^6}{4}-\frac{2x^7}{5}+\frac{x^8}{6}\right).$$

Yüzbaşı introduced the solution using a collocation method that based on Bernstein polynomials for this problem. He got the exact closed form using his method. The solution obtained by our method in polynomial form is

$$u_7(x)=1-x+x^2-3.18\times10^{-17}x^3+2.17\times10^{-17}x^4+1.6904\times10^{-17}x^5+3.28\times10^{-18}x^7-1.4\times10^{-17}x^7,$$

with maximum absolute error is 3.8E-18 for $n=7$

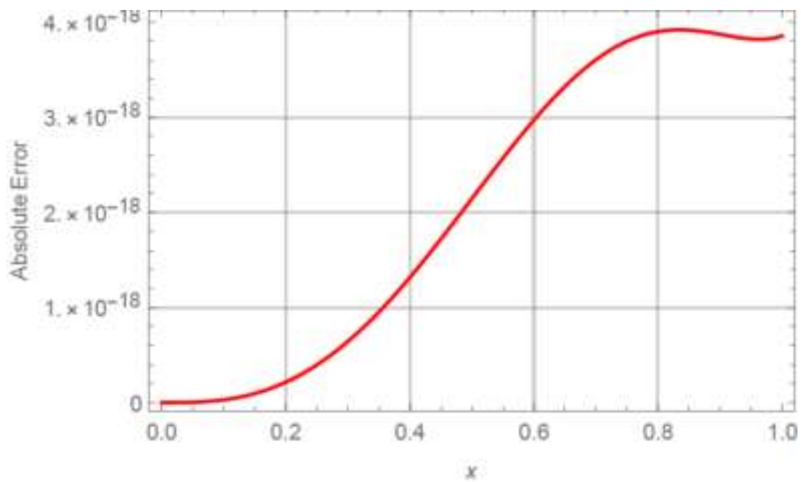

Figure 1: Absolute error corresponding to $n=7$

Example 4.3 [23] Consider the 1st order nonlinear integro-differential equation

$$u'(x)=-\frac{1}{3}\cos(x)-\frac{2}{3}\cos(2x)-2\sin(x)+\int_0^x \cos(x-t)u^2(t)\,dt, \quad 0\le x\le 1,$$

subjected to

$$u(0)=1,$$

whose exact solution is

$$u(x)=\cos(x)-\sin(x).$$

The solution obtained by our method in polynomial form is

$$\begin{aligned}u_{16}(x)\approx{}&1+x+0.5x^2+0.1666x^3+0.04166x^4+0.00833x^5+0.0013888x^6+0.000198x^7\\&+0.0000248x^8+2.75573\times10^{-6}x^9+2.75573\times10^{-7}x^{10}+2.5052\times10^{-8}x^{11}\\&+2.087729\times10^{-9}x^{12}+1.6054\times10^{-10}x^{13}+1.151256\times10^{-11}x^{14}+7.46220\times10^{-13}x^{15}\\&+5.9380\times10^{-14}x^{16},\end{aligned}$$

Table 2 shows the maximum absolute error in different values of n. The comparison between our method and modified Laplace Adomian decomposition method listed in table 4.

Table 2 Maximum absolute error corresponding to different values of n for Example 4.3

n	$\lVert E(x)\rVert$

6	7.875E-6
10	5.617E-11
16	3.330E-16

Example 4.4 [23] Consider the nonlinear integral differential equation

$$u(x)=\frac{1}{4}+\frac{x}{2}+e^{x}-\frac{e^{2x}}{4}+\int_{0}^{x}(x-t)u^{2}(t)\,dt, \quad 0\le x\le 1,$$

whose exact solution is

$$u(x)=e^{x}.$$

The solution obtained by our method in polynomial form is

$$\begin{aligned} u_{16}(x) \approx\ & 1-x-0.5x^{2}+0.1666x^{3}+0.04166x^{4}-0.00833x^{5}-0.0013888x^{7}+0.000198x^{7} \\ & +0.0000248x^{8}-0.000002755\times10^{-6}x^{9}-2.75573\times10^{-7}x^{10}+2.5052\times10^{-8}x^{11} \\ & +2.087729\times10^{-9}x^{12}-1.6054\times10^{-10}x^{13}-1.151256\times10^{-11}x^{14}+7.46220\times10^{-13}x^{15} \\ & +5.9380\times10^{-14}x^{16}, \end{aligned}$$

Table 3 represents the maximum absolute error in different values of n. The comparison between our method and modified Laplace Adomian decomposition method listed in table 4.

Table 3 Maximum absolute error corresponding to different values of n for Example 4.4

n	$\lVert E(x)\rVert$
6	4.691E-6
10	3.320E-11
15	8.881E-16

Table 4 Maximum absolute error $\lVert E(x)\rVert$ corresponding to method in [23] for examples 4.3 and 4.4

	Example 4.3			Example 4.4		
x	Exact	Proposed method	Method in [23]	Exact	Proposed method	Method in [23]
0.0	1	1	1	1	1	1
0.1	0.895171	0.895171	0.8964	1.10517	1.10517	1.1044

0.2	0.781397	0.781397	0.7858	1.22140	1.22140	1.2188
0.3	0.659816	0.659816	0.6687	1.34986	1.34986	1.3441
0.4	0.531643	0.531643	0.5455	1.49182	1.49182	1.4819
0.5	398157	0.398157	0.4166	1.64872	1.64872	1.6338
0.6	0.260693	0.260693	0.2828	1.82212	1.82212	1.8018
0.7	0.120625	0.120625	0.1451	2.01375	2.01375	1.9887
0.8	-0.0206494	-0.0206494	0.0049	2.22554	2.22554	2.1978
0.9	-0.161717	-0.161717	-0.1361	2.4596	2.4596	2.4335
1.0	-0.301169	-0.301169	-0.2758	2.71828	2.71828	2.7014

5. Conclusion

In this article, we have studied nonlinear Volterra-Fredholm integro-differential equations. The Chebyshev-Galerkin method is utilized to get the approximate series solution of the given problem. In order to apply this method, we proved some theorems to use it in our technique. We have solved the given numerical examples to explain the proposed method and its implementation in our work. To check our solution, we get the absolute error graphs corresponding to some numerical examples on the solution domain.

6. References

[1] S. Momani, O. A. Arqub, T. Hayat, and H. Al-Sulami, “A computational method for solving periodic boundary value problems for integro-differential equations of Fredholm–Volterra type,” *Appl. Math. Comput.*, vol. 240, pp. 229–239, 2014.

[2] P. Rahimkhani, Y. Ordokhani, and E. Babolian, “Fractional-order Bernoulli functions and their applications in solving fractional Fredholem–Volterra integro-differential equations,” *Appl. Numer. Math.*, vol. 122, pp. 66–81, Dec. 2017.

[3] M. S. Aydogan, D. Baleanu, A. Mousalou, and S. Rezapour, “On high order fractional integro-differential equations including the Caputo–Fabrizio derivative,” *Bound. Value Probl.*, vol. 2018, no. 1, p. 90, Jun. 2018.

[4] M. Erfanian, M. Gachpazan, and H. Beiglo, “A new sequential approach for solving the integro-differential equation via Haar wavelet bases,” *Comput. Math. Math. Phys.*, vol. 57, no. 2, pp. 297–305, Feb. 2017.

[5] J. R. Loh, C. Phang, and A. Isah, “New Operational Matrix via Genocchi Polynomials for Solving Fredholm-Volterra Fractional Integro-Differential Equations,” *Advances in Mathematical Physics*, Jan. 16, 2017. https://www.hindawi.com/journals/amp/2017/3821870/ (accessed Feb. 20, 2021).

[6] K. Maleknejad, B. Basirat, and E. Hashemizadeh, “Hybrid Legendre polynomials and Block-Pulse functions approach for nonlinear Volterra–Fredholm integro-differential equations,” *Comput. Math. Appl.*, vol. 61, no. 9, pp. 2821–2828, May 2011.

[7] E. Babolian, Z. Masouri, and S. Hatamzadeh-Varmazyar, “Numerical solution of nonlinear Volterra–Fredholm integro-differential equations via direct method using triangular functions,” *Comput. Math. Appl.*, vol. 58, no. 2, pp. 239–247, Jul. 2009.

[8] P. Darania and K. Ivaz, “Numerical solution of nonlinear Volterra–Fredholm integro-differential equations,” *Comput. Math. Appl.*, vol. 56, no. 9, pp. 2197–2209, Nov. 2008.

[9] K. Maleknejad and Y. Mahmoudi, “Taylor polynomial solution of high-order nonlinear Volterra–Fredholm integro-differential equations,” *Appl. Math. Comput.*, vol. 145, no. 2, pp. 641–653, Dec. 2003.

[10] H. Laeli Dastjerdi and F. M. Maalek Ghaini, “Numerical solution of Volterra–Fredholm integral equations by moving least square method and Chebyshev polynomials,” *Appl. Math. Model.*, vol. 36, no. 7, pp. 3283–3288, Jul. 2012.
[11] K. Parand and J. A. Rad, “Numerical solution of nonlinear Volterra–Fredholm–Hammerstein integral equations via collocation method based on radial basis functions,” *Appl. Math. Comput.*, vol. 218, no. 9, pp. 5292–5309, Jan. 2012.
[12] K. Wang and Q. Wang, “Taylor collocation method and convergence analysis for the Volterra–Fredholm integral equations,” *J. Comput. Appl. Math.*, vol. 260, pp. 294–300, Apr. 2014.
[13] Q. Wang, K. Wang, and S. Chen, “Least squares approximation method for the solution of Volterra–Fredholm integral equations,” *J. Comput. Appl. Math.*, vol. 272, pp. 141–147, Dec. 2014,.
[14] S. Nemati, “Numerical solution of Volterra–Fredholm integral equations using Legendre collocation method,” *J. Comput. Appl. Math.*, vol. 278, pp. 29–36, Apr. 2015.
[15] M. I. Berenguer, D. Gámez, and A. J. López Linares, “Fixed point techniques and Schauder bases to approximate the solution of the first order nonlinear mixed Fredholm–Volterra integro-differential equation,” *J. Comput. Appl. Math.*, vol. 252, pp. 52–61, Nov. 2013.
[16] Siraj-ul-Islam, I. Aziz, and A. S. Al-Fhaid, “An improved method based on Haar wavelets for numerical solution of nonlinear integral and integro-differential equations of first and higher orders,” *J. Comput. Appl. Math.*, vol. 260, pp. 449–469, Apr. 2014.
[17] E. Babolian, Z. Masouri, and and S. Hatamzadeh, “New Direct Method to Solve Nonlinear Volterra-Fredholm Integral and Integro-Differential Equations Using Operational Matrix with Block-Pulse Functions,” *Prog. Electromagn. Res. B*, vol. 8, pp. 59–76, 2008.
[18] M. A. Fariborzi Araghi and S. S. Behzadi, “Numerical solution of nonlinear Volterra-Fredholm integro-differential equations using Homotopy Analysis Method,” *J. Appl. Math. Comput.*, vol. 37, no. 1, pp. 1–12, Oct. 2011.
[19] Siraj-ul-Islam, I. Aziz, and A. S. Al-Fhaid, “An improved method based on Haar wavelets for numerical solution of nonlinear integral and integro-differential equations of first and higher orders,” *J. Comput. Appl. Math.*, vol. 260, pp. 449–469, Apr. 2014.
[20] E. Babolian, Z. Masouri, and S. Hatamzadeh-Varmazyar, “Numerical solution of nonlinear Volterra–Fredholm integro-differential equations via direct method using triangular functions,” *Comput. Math. Appl.*, vol. 58, no. 2, Art. no. 2, Jul. 2009.
[21] K. Maleknejad, B. Basirat, and E. Hashemizadeh, “Hybrid Legendre polynomials and Block-Pulse functions approach for nonlinear Volterra–Fredholm integro-differential equations,” *Comput. Math. Appl.*, vol. 61, no. 9, Art. no. 9, May 2011.
[22] Ş. Yüzbaşı, “A collocation method based on Bernstein polynomials to solve nonlinear Fredholm–Volterra integro-differential equations,” *Appl. Math. Comput.*, vol. 273, pp. 142–154, Jan. 2016.
[23] D. Rani and V. Mishra, “Solutions of Volterra integral and integro-differential equations using modified Laplace Adomian decomposition method.”, *Journal of Applied Mathematics, Statistics and Informatics*, vol. 15, 2019.