Sigma J Eng & Nat Sci 38 (2), 2020, 995-1005

Sigma Journal of Engineering and Natural Sciences
Sigma Mühendislik ve Fen Bilimleri Dergisi

**Research Article**

# A GALERKIN-TYPE METHOD FOR SOLUTIONS OF PANTOGRAPH-TYPE VOLTERRA-FREDHOLM INTEGRO-DIFFERENTIAL EQUATIONS WITH FUNCTIONAL UPPER LIMIT

**Şuayip YÜZBAŞI*[1], Murat KARAÇAYIR[2]**

[1]*Akdeniz University, Department of Mathematics, ANTALYA;* ORCID: 0000-0002-5838-7063
[2]*Akdeniz University, Department of Mathematics, ANTALYA;* ORCID: 0000-0001-6230-3638

**Received: 27.01.2020 Revised: 20.03.2020 Accepted: 02.04.2020**

**ABSTRACT**

In this study, we present a Galerkin-type method for obtaining approximate solutions of linear Volterra-Fredholm delay integro-differential equations with a functional upper limit under mixed conditions. The method gives an approximate solution of the problem in power series form truncated after a certain term. Using an integer value N as the truncation point and making use of the matrix representations of a polynomial and its derivatives, we obtain the matrix form of the problem expressed in terms of the approximate solution polynomial. By applying inner product to these relations with monomials up to degree N and incorporating the mixed conditions, the problem is reduced to a system of linear algebraic equations. The approximate solution of the problem is then determined from this linear system. In addition, we discuss a way of improving an obtained approximate solution by means of its estimated error function. The presented scheme has the advantages of (1) being applicable to a wide range of problems including pantograph-type equations with or without Fredholm and Volterra integral terms, and (2) giving accurate results as demonstrated by applications to example problems taken from existing studies.
**Keywords:** Volterra-Fredholm integro-differential equations, pantograph-type integro-differential equations, Galerkin type method, numerical solutions, error estimation, residual correction.

## 1. INTRODUCTION

When one desires to describe a phenomenon that encompasses both the rate of change of an unknown function and cummulative effects on that function expressed by an integral, integro-differential equations come into picture [1, p. 196]. A general classification applies to these equations according to the upper limits of the integral term; namely, Fredholm integro-differential equations constitute the class with a constant upper limit in the integral term, while those with variable upper limit are known as Volterra integro-differential equations.

Each type of integro-differential equations has its own resources and applications. For instance, neural networks [2] and signal processing [3] are among the well-known applications of Fredholm integro-differential equations, whereas Volterra integro-differential equations arise in the study of population dynamics [4] and heat transfer [5] among many other things. There are also model problems that require the incorporation of both types of integral terms, thus resulting

* Corresponding Author: e-mail: syuzbasi@akdeniz.edu.tr, tel: (242) 310 23 83

in Volterra-Fredholm integro-differential equations. The topics where these equations play a role include spread of an epidemic in both space and time [6] and contact mechanics [7].

In this paper, we will be interested in obtaining approximate solutions of the linear Volterra-Fredholm delay integro-differential equation given by

$$\sum_{k=0}^{m}\sum_{j=0}^{J} P_{j,k}(x)y^{(k)}(\lambda_j x+\mu_j) = g(x) + \int_a^b \sum_{r=0}^{m} K_r(x,t)y^{(r)}(t)dt + \int_a^x \sum_{r=0}^{m} L_{1,r}(x,t)y^{(r)}(t)dt + \int_a^{h(x)} \sum_{r=0}^{m} L_{2,r}(x,t)y^{(r)}(t)dt \quad (1)$$

under the following mixed conditions:

$$\sum_{j=0}^{m-1}\left(a_{i,j}y^{(j)}(a) + b_{i,j}y^{(j)}(b)\right) = l_i, i = 1,2,\dots,m. \quad (2)$$

In the left-hand side of equation (1), the unknown function $y$ and its derivatives up to *m*-th order appear with linear delayed terms of $x$, where the equation is reduced to the standard Volterra-Fredholm integro-differential equation if $\lambda_j = 1$ and $\mu_j = 0$ for all $j = 0,1,\dots,J$. The given $g$ and $P_{j,k}$ for all $j = 0,1,\dots,J$ and $k = 0,1,\dots,m$ are known continuous functions of one variable in the interval $a \le x \le b$. In addition, the known functions of two variables $K_r, L_{1,r}$ and $L_{2,r}$ for $r = 0,1,\dots,m$ are continuous in the domain $[a,b]\times[a,b]$. Equation (1) also contains an integral term having a known function $h(x) \neq x$ as the upper limit. By "functional upper limit", we are referring to the existence of such a term in the considered problem. Since the order of the equation is equal to *m*, there are a total of *m* initial and boundary conditions given by (2), which we refer to as mixed-type conditions in view of the fact that some of them may contain terms evaluated at both boundary points $a$ and $b$. Often we will refer to equation (1) and mixed conditions (2) together as problem (1)-(2).

Integro-differential equations are difficult to solve in the general case, resulting in many researchers having applied a variety of numerical techniques to problems involving them. For instance, linear Volterra-Fredholm integro-differential equations have been solved by various collocation methods employing Taylor [8] and Bessel [9] polynomials in addition to an operational matrix method using the standard polynomial basis [10]. In addition, popular methods such as He's variational iteration method and homotopy perturbation method have been applied to Volterra-Fredholm integro-differential equations containing nonlinearity to second degree in [11]. Pantograph-type Volterra integro-differential equations, which are special cases of equation (1) without a Fredholm term, were considered in [12], where Laguerre polynomial solutions were found by a suitable utilization of collocation points. A type of linear Volterra-Fredholm integral equation with a functional upper limit where there is also a functional term in the left-hand side was numerically solved by Taylor polynomials method in [13]. Next, collocation methods in conjunction with Chelyshkov and Dickson polynomials were applied to problems similar to (1)-(2) in [14] and [15], respectively. Lastly, Saeedi et al. [16] have presented a numerical method by combining difference method with successive approximation method, natural cubic spline interpolation method and trapezoidal quadrature rule for solving integro-differential equations.

The main interest of this study is to introduce a numerical scheme based on Galerkin method in order to obtain approximate polynomial solutions to problem (1)-(2). The details of this method will be explained in Section 2. In Section 3, we outline the residual correction technique, which exploits the linearity of equation (1) to estimate the error of any of its approximate solutions and constructs a hopefully more accurate solution using this estimation. Numerical simulations are presented in Section 4. Finally, the conclusions of the paper are summarized in Section 5.

## 2. METHOD OF SOLUTION

We present the Galerkin-like method that will be used to solve problem (1)-(2) in this section. We emphasize that a Galerkin-like method has been used to solve Fredholm integro-differential equations [17], Fredholm integro-differential equations with weakly singular kernel functions [18] and delayed HIV Infection model of CD4+ T-cells [19]. In addition, Türkyılmazoğlu solved both nonlinear Lane-Emden-Fowler type differential equations [20] and nonlinear heat transfer problems in fins [21] using a generalized version of the method presented in this section.

To begin with, we assume that the solution of problem (1)-(2) can be expressed as

$$y(x) = a_0 + a_1 x + a_2 x^2 + \cdots = \sum_{n=0}^{\infty} a_n\, x^n.$$

The proposed scheme is initiated with choosing an integer parameter $N$. Then, we truncate the above power series after the $(N+1)$-st term, resulting in the following polynomial of degree $N$:

$$y_N(x) := a_0 + a_1 x + \cdots + a_N x^N = \textstyle\sum_{n=0}^{N} a_n\, x^n.$$

Thus, the approximate solution of problem (1)-(2) will be an $N$-th degree polynomial $y_N$ whose coefficients are denoted by $a_0, a_1, \ldots, a_N$. The aim of the scheme is to compute these coefficients. Before proceeding with the details of this computation, let us define a few matrices that will make the succeeding discussion significantly easier. First, we note that $y_N(x) = \mathbf{X}(x)\mathbf{A}$, where

$$\mathbf{X}(x) = [1 \quad x \quad x^2 \quad \ldots \quad x^N], \mathbf{A} = [a_0 \quad a_1 \quad \ldots \quad a_N]^T.$$

Here, $\mathbf{X}(x)$ is a row vector of length $N+1$ that contains the powers of the indeterminate $x$ up to order $N$, whereas the column vector $\mathbf{A}$ of the same length consists of all the unknown coefficients in the approximate solution polynomial $y_N$. The derivatives of $y_N$ can also be written in terms of a matrix product with the help of an auxiliary $(N+1)\times(N+1)$ matrix defined by

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & N \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}.$$

More formally, the entries of $\mathbf{B}$ are given by $\mathbf{B}_{i,i+1} = i$ for $i = 0,1,\ldots,N$ and $\mathbf{B}_{i,j} = 0$ for all other entries. Then it can easily be verified that the identity $y_N^{(n)}(x) = X(x)B^n A$ holds for the $n$-th derivative of $y_N$. Next step is to substitute the approximate solution $y_N(x)$ into equation (1). Doing this and rearranging yields

$$\sum_{k=0}^{m}\sum_{j=0}^{J} P_{j,k}(x) y_N^{(k)}(\lambda_j x + \mu_j) - \int_a^b \textstyle\sum_{r=0}^{m} K_r(x,t) y_N^{(r)}(t)dt - \int_a^x \textstyle\sum_{r=0}^{m} L_{1,r}(x,t) y_N^{(r)}(t)dt - \int_a^{h(x)} \textstyle\sum_{r=0}^{m} L_{2,r}(x,t) y_N^{(r)}(t)dt = g(x), \tag{3}$$

or in matrix form

$$\sum_{k=0}^{m}\sum_{j=0}^{J} P_{j,k}(x)\mathbf{X}(\lambda_j x + \mu_j)\mathbf{B}^k\mathbf{A} - \int_a^b \textstyle\sum_{r=0}^{m} K_r(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt - \int_a^x \textstyle\sum_{r=0}^{m} L_{1,r}(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt - \int_a^{h(x)} \textstyle\sum_{r=0}^{m} L_{2,r}(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt = g(x). \tag{4}$$

The terms $\mathbf{X}(\lambda_j x + \mu_j)$ containing delay can also be expressed by means of a matrix product as follows: Let $\mathbf{B}(\lambda,\mu)$ be the $(N+1)\times(N+1)$ square matrix whose entries are defined by the

rules $\mathbf{B}_{i,j}(\lambda,\mu)=\binom{j-1}{i-1}\lambda^{i-1}\mu^{j-i}$ for $i\le j$ and $\mathbf{B}_{i,j}(\lambda,\mu)=0$ for $i>j$. More explicitly, we have

$$\mathbf{B}(\lambda,\mu)=\begin{bmatrix}1 & \mu & \mu^2 & \dots & \mu^N\\ 0 & \lambda & 2\lambda\mu & \dots & N\lambda\mu^{N-1}\\ 0 & 0 & \lambda^2 & \dots & \frac{N(N-1)}{2}\lambda^2\mu^{N-2}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \dots & \lambda^N\end{bmatrix}.$$

Then, it is straightforward to check that $\mathbf{X}(\lambda x+\mu)=\mathbf{X}(x)\mathbf{B}(\lambda,\mu)$. Thus, we can rewrite (4) as

$$\sum_{k=0}^{m}\sum_{j=0}^{J}P_{j,k}(x)\mathbf{X}(x)\mathbf{B}(\lambda_j,\mu_j)\mathbf{B}^kA-\int_a^b\sum_{r=0}^{m}K_r(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt \\ -\int_a^x\sum_{r=0}^{m}L_{1,r}(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt-\int_a^{h(x)}\sum_{r=0}^{m}L_{2,r}(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt=g(x). \tag{5}$$

At this point, it will be advantageous to introduce some extra notation for the terms on the left-hand side. Let us denote the first term, which is a sum of at most $(m+1)(J+1)$ separate terms containing delay, by $\mathbf{D}(x)$. More explicitly, let us define

$$\mathbf{D}(x):=\sum_{k=0}^{m}\sum_{j=0}^{J}P_{j,k}(x)\mathbf{X}(x)\mathbf{B}(\lambda_j,\mu_j)\mathbf{B}^k\mathbf{A}.$$

In a similar manner, for the integral terms we define

$$\mathbf{F}(x):=\int_a^b\sum_{r=0}^{m}K_r(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt,\ \mathbf{V}_1(x):=\int_a^x\sum_{r=0}^{m}L_{1,r}(x,t)\mathbf{X}(t)\mathbf{B}^r\mathbf{A}dt,$$
$$\mathbf{V}_2(x):=\int_a^{h(x)}\sum_{r=0}^{m}L_{2,r}(x,t)\mathbf{X}(t)\mathbf{B}^rAdt.$$

Using these newly defined notation, considering that the factor $\mathbf{A}$ is common in all the terms on the left-hand side of (5) and making use of the linearity of the integral operator, we can write

$$(\mathbf{D}(x)-\mathbf{F}(x)-\mathbf{V}_1(x)-\mathbf{V}_2(x))\mathbf{A}=g(x). \tag{6}$$

Here, each term in the parantheses is a row vector of length $N+1$ having expressions containing $x$ as its entries. If we have $\mathbf{V}_1(x)=\mathbf{V}_2(x)=0$, in other words, if the original equation (1) contains only the Fredholm integral, then the problem is a Fredholm delay integro-differential equation. Similarly, if $\mathbf{F}(x)=\mathbf{V}_2(x)=0$, then it is a Volterra delay integro-differential equation. Lastly, if only $\mathbf{V}_2(x)=0$, the problem is a Volterra-Fredholm delay integro-differential equation. Thus, the present numerical scheme covers all these types of equations. To make the notation more concise, let us denote the sum inside the parantheses in (6) by $\mathbf{W}(x)$, thereby writing

$$\mathbf{W}(x)\mathbf{A}=g(x). \tag{7}$$

Now, we apply the main idea of our numerical scheme. Namely, we apply inner product to equation (7) with the elements of the base set $\{1,x,x^2,\dots,x^N\}$. By "inner product" we mean the inner product induced by the standard $L^2$-norm in the space $L^2(a,b)$ of the square integrable functions on $a\le x\le b$. More explicitly, if $f,g\in L^2(a,b)$, their inner product is $<f,g>=\int_a^b f(t)g(t)dt$. Thanks to the linearity of inner product, for an arbitrary $k=0,1,\dots,N$ we have

$$<\mathbf{W}(x),x^k>\mathbf{A}=<g(x),x^k>.$$

Here, $<\mathbf{W}(x),x^k>$ is simply a row vector of length $N+1$ resulting from taking inner product of $x^k$ with all entries (which are functions of $x$) in $\mathbf{W}(x)$ one by one. Therefore, for each $k$ the above is a linear equation in the unknown coefficients $a_0,a_1,\dots,a_N$. More explicitly, we have

$$< \mathbf{W}(x), x^k >= [< \mathbf{W}_1(x), x^k > \quad < \mathbf{W}_2(x), x^k > \quad \cdots \quad < \mathbf{W}_{N+1}(x), x^k >],$$

where $\mathbf{W}_j(x)$ denotes the $j$-th entry in $\mathbf{W}(x)$. Since there are $N+1$ possible values for $k$, hence there are $N+1$ such inner products, we have a total of $N+1$ linear equations in the same number of unknowns $a_0, a_1, \dots, a_N$ expressed by the system

$$\mathbf{WA} = \mathbf{G}, \tag{8}$$

where

$$\mathbf{W} = \begin{bmatrix} < \mathbf{W}(x),1 > \\ < \mathbf{W}(x), x^2 > \\ \vdots \\ < \mathbf{W}(x), x^N > \end{bmatrix}, \mathbf{G} = \begin{bmatrix} < g(x),1 > \\ < g(x), x^2 > \\ \vdots \\ < g(x), x^N > \end{bmatrix}.$$

The polynomial formed by the solutions of the system (8) is not taken as the approximate solution of problem (1)-(2) since the mixed conditions (2) have not been taken into account yet. In order to incorporate these conditions, we transform them into matrix equations by means of the auxiliary matrices $\mathbf{X}(x)$, $\mathbf{A}$ and $\mathbf{B}$ defined before. Thus, the $i$-th mixed condition can be rewritten as

$$\sum_{j=0}^{m-1} a_{i,j}\, \mathbf{X}(a)\mathbf{B}^j\mathbf{A} + b_{i,j}\mathbf{X}(b)\mathbf{B}^j\mathbf{A} = (\sum_{j=0}^{m-1} a_{i,j}\, \mathbf{X}(a)\mathbf{B}^j + b_{i,j}\mathbf{X}(b)\mathbf{B}^j)\mathbf{A} = \mathbf{C}_i\mathbf{A} = \lambda_i,$$

for $i = 1,2,\dots,m$, where we used the notation $\mathbf{C}_i = \sum_{j=0}^{m-1} a_{i,j}\mathbf{X}(a)\mathbf{B}^j + b_{i,j}\mathbf{X}(b)\mathbf{B}^j$. In order to impose the mixed conditions (2) on the approximate solution $y_N$, we must incorporate them into the system (8). Since there should be as many equations as unknowns in a linear system for it to have a unique solution, we do this by replacing the last $m$ rows of (8) by the rows corresponding to the $m$ mixed conditions, thus obtaining the new matrices

$$\widetilde{\mathbf{W}} = [< \mathbf{W}(x), 1 > ; < \mathbf{W}(x), x > ; \dots ; < \mathbf{W}(x), x^{N-m} > ; \mathbf{C}_1 ; \mathbf{C}_2; \dots ; \mathbf{C}_m],$$
$$\widetilde{\mathbf{G}} = [< g(x), 1 > ; < g(x), x > ; \dots ; < g(x), x^{N-m} > ; \lambda_1 ; \lambda_2 ; \dots ; \lambda_m].$$

Here, we used the convention that a semicolon starts a new row. If the resulting modified system $\widetilde{\mathbf{W}}\mathbf{A} = \widetilde{\mathbf{G}}$ is of full rank, it has a unique solution computed by $\mathbf{A} = \widetilde{\mathbf{W}}^{-1}\widetilde{\mathbf{G}}$. This solution gives rise to the polynomial $y_N(x) = \sum_{n=0}^{N} a_n\, x^n$, which is the approximate solution of problem (1)-(2) which strictly satisfies the mixed conditions (2).

In obtaining the unknown coefficients $a_0, a_1, \dots, a_N$, it is important to consider the stability of the numerical method used to solve the linear system $\widetilde{\mathbf{W}}\mathbf{A} = \widetilde{\mathbf{G}}$. The standard matrix inversion algorithm in most computer algebra systems (e.g. MATLAB) is based on LU factiorization, which is known to be unstable in some cases [22]. This may be an issue in our case as the numerical scheme described in this section is prone to yield coefficient matrices with high condition numbers, making the system $\widetilde{\mathbf{W}}\mathbf{A} = \widetilde{\mathbf{G}}$ ill-conditioned. For this reason, we employed a stable iterative method to find the inverse of the coefficient matrix $\widetilde{\mathbf{W}}$. This method was considered in [23] and also in [24], where its convergence was proved. In order to find the inverse of an $N \times N$ matrix $\mathbf{M}$, starting from an initial guess $\mathbf{A}_0$, the method computes

$$\mathbf{A}_{n+1} = \mathbf{A}_n\big(3I - \mathbf{M}\mathbf{A}_n(3I - \mathbf{M}\mathbf{A}_n)\big), n = 0,1,2, \dots$$

where $I$ is the $N \times N$ identity matrix. As the initial guess, we used $\frac{\mathbf{M}^T}{|\mathbf{M}|_1|\mathbf{M}|_\infty}$, which was suggested in [24]. The solutions of the final linear algebraic system in each of the example problems in Section 4 have been carried out using this iterative algorithm.

Another issue that might interest the reader is the applicability of the presented method to nonlinear problems. If the equation to be solved is nonlinear, then in the matrix form

$$\mathbf{W}(x)\mathbf{A} = g(x)$$

of the equation, the vector $\mathbf{W}(x)$ involves the unknown coefficients $a_0, a_1, \dots, a_N$ in its entries, which means that the system

$$\mathbf{WA} = \mathbf{G}$$

of algebraic equations would not be linear. In this case, the solution of the problem requires solving a nonlinear system of algebraic equations. We refer the interested reader to [19] in order to help understand the application of the discussed scheme to nonlinear problems.

## 3. ERROR ESTIMATION AND RESIDUAL CORRECTION

Let us denote the exact solution of problem (1)-(2) by $y_{\text{exact}}$. If this solution is known, we can measure the accuracy of an approximate solution $y_N$ by considering its absolute error given by $|e_N(x)| = |y_{\text{exact}}(x) - y_N(x)|$. However, the exact solution is not at hand in most cases, depriving us of this direct measure. In such cases, the residual of the approximate solution can be used as an alternative measure . In Section 2, we already considered this residual in (3), given by

$$\sum_{k=0}^{m} \sum_{j=0}^{J} P_{j,k}(x) y_N^{(k)}(\lambda_j x + \mu_j) - \int_a^b \sum_{r=0}^{m} K_r(x,t) y_N^{(r)}(t)dt - \int_a^x \sum_{r=0}^{m} L_{1,r}(x,t) y_N^{(r)}(t)dt - \int_a^{h(x)} \sum_{r=0}^{m} L_{2,r}(x,t) y_N^{(r)}(t)dt - g(x). \tag{9}$$

On the other hand, since $y_{\text{exact}}$ is the exact solution, its residual is zero as given by

$$R_N(x) := \sum_{k=0}^{m} \sum_{j=0}^{J} P_{j,k}(x) y_{\text{exact}}^{(k)}(\lambda_j x + \mu_j) - \int_a^b \sum_{r=0}^{m} K_r(x,t) y_{\text{exact}}^{(r)}(t)dt - \int_a^x \sum_{r=0}^{m} L_{1,r}(x,t) y_{\text{exact}}^{(r)}(t)dt - \int_a^{h(x)} \sum_{r=0}^{m} L_{2,r}(x,t) y_{\text{exact}}^{(r)}(t)dt - g(x) = 0. \tag{10}$$

In view of the identity $e_N^{(r)}(x) = y_{\text{exact}}^{(r)}(x) - y_N^{(r)}(x)$, subtracting (9) from (10) yields

$$\sum_{k=0}^{m} \sum_{j=0}^{J} P_{j,k}(x) e_N^{(k)}(\lambda_j x + \mu_j) = -R_N(x) + \int_a^b \sum_{r=0}^{m} K_r(x,t) e_N^{(r)}(t)dt + \int_a^x \sum_{r=0}^{m} L_{1,r}(x,t) e_N^{(r)}(t)dt + \int_a^{h(x)} \sum_{r=0}^{m} L_{2,r}(x,t) e_N^{(r)}(t)dt. \tag{11}$$

The unknown of this equation is the actual error $e_N$ corresponding to the approximate solution $y_N$. In addition, since both $y_{\text{exact}}$ and $y_N$ satisfy the mixed conditions (2), we have

$$\sum_{j=0}^{m-1} \left( a_{i,j} e_N^{(j)}(a) + b_{i,j} e_N^{(j)}(b) \right) = \sum_{j=0}^{m-1} \left( a_{i,j} (y_{\text{exact}} - y_N)^{(j)}(a) + b_{i,j} (y_{\text{exact}} - y_N)^{(j)}(b) \right)$$
$$= \sum_{j=0}^{m-1} \left( a_{i,j} y_{\text{exact}}^{(j)}(a) + b_{i,j} y_{\text{exact}}^{(j)}(b) \right) - \sum_{j=0}^{m-1} \left( a_{i,j} y_N^{(j)}(a) + b_{i,j} y_N^{(j)}(b) \right) = \lambda_i - \lambda_i = 0. \tag{12}$$

The solution of equation (11) subject to the homogeneous conditions (12) is the actual error $e_N$. The problem (11)-(12) is called the "error problem" associated with the approximate solution $y_N$. Since it is exactly of the same form as the original problem (1)-(2), we can apply the present scheme using a new parameter value $M$, and obtain an approximation to the actual error $e_N$. We denote the obtained solution by $e_{N,M}$. Since $e_{N,M}$ is not the actual solution but an approximation to it, we call this an *estimation* for the actual error. Since the actual error satisfies $y_{\text{exact}}(x) = y_N(x) + e_N(x)$, this estimation is used to obtain a new approximate solution $y_{N,M}$ as follows:

$$y_{N,M}(x) = y_N(x) + e_{N,M}(x).$$

For the error $E_{N,M}$ of this new approximate solution we can write

$$E_{N,M}(x) = y_{\text{exact}}(x) - y_{N,M}(x)$$
$$= (y_N(x) + e_N(x)) - (y_N(x) + e_{N,M}(x)) = e_N(x) - e_{N,M}(x).$$

Thus, the accuracy of the new approximation $y_{N,M}$ is equal to the accuracy of the estimation $e_{N,M}$ for the actual error $e_N$. Referring to the expectation that $y_{N,M}$ is more accurate than $y_N$, it is called a corrected approximate solution of problem (1)-(2), and the process of obtaining it is called residual correction. The effectivity of this technique will be made clear in one of the example problems in the next section.

## 4. APPLICATIONS TO EXAMPLE PROBLEMS

In this section, we apply the numerical scheme described in Section 2 to three example problems taken from present works. We also demonstrate the application of residual correction on one of these problems. All calculations have been carried out in MATLAB.

**Example 1:** First example problem is the following second order equation taken from [15]:

$$(x^2+1)y''(0.2x) + xy'(x-1) - (x-1)y(x+2) + y(x) = 6 - \frac{14x}{3} - x^2 - \frac{5x^3}{2} - \frac{x^4}{3} + 2\int_0^1 xy'(t)dt + \int_0^{0.5x+1}(xt^2+tx^2)\,y''(t)dt,\ \ y(0) = -1, y'(1) = 3. \tag{13}$$

In this problem, we have $\mathbf{V}_1(x) = 0$ in the terminology we introduced in Section 2. As for the other terms which are present in (6), we have when we apply the numerical scheme with $N = 2$:

$$\mathbf{D}(x) = [2 - x,\quad 2 + x - x^2,\quad 6 - 2x + 2x^2 - x^3],\ \mathbf{F}(x) = [0,\quad 2x,\quad 2x],$$
$$\mathbf{V}_2(x) = \left[0,\quad 0,\quad \frac{2x}{3} + 2x^2 + \frac{3x^3}{2} + \frac{x^4}{3}\right].$$

As a result, the left-hand side of equation (7) is computed as

$$\mathbf{W}(x) = \left[2 - x,\quad 2 - x - x^2,\quad 6 - \frac{14x}{3} - \frac{5x^3}{2} - \frac{x^4}{3}\right].$$

After applying inner product with the monomials $1, x, x^2$, we obtain the system $\mathbf{WA} = \mathbf{G}$, where

$$\mathbf{W} = \begin{bmatrix} 3/2 & 7/6 & 119/40 \\ 2/3 & 5/12 & 8/9 \\ 5/12 & 13/60 & 31/84 \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} 317/120 \\ 23/36 \\ 71/420 \end{bmatrix}.$$

The matrices for the mixed conditions are $[1\quad 0\quad 0]\mathbf{A} = -1$ and $[0\quad 1\quad 2]\mathbf{A} = 3$. Replacing the last two rows of $\mathbf{W}$ and $\mathbf{G}$ with these gives the modified system $\widetilde{\mathbf{W}}\mathbf{A} = \widetilde{\mathbf{G}}$, where

$$\widetilde{\mathbf{W}} = \begin{bmatrix} 3/2 & 7/6 & 119/40 \\ 1 & 0 & 0 \\ 0 & 1 & 2 \end{bmatrix},\ \widetilde{\mathbf{G}} = \begin{bmatrix} 317/120 \\ -1 \\ 3 \end{bmatrix}.$$

The solution of this system is $\mathbf{A} = [-1\ \ 1\ \ 1]^T$, yielding the solution $y_2(x) = -1 + x + x^2$. This is equal to the exact solution of problem (13). Using $N$ values greater than 2 does not change this: We still obtain this exact solution as a result of applying the numerical scheme.

**Example 2:** Next example is a second order generalized pantograph-type Volterra integro-differential equation with an integral term having a functional upper limit taken from [12, 15]:

$$y''(x) - xy'(0.5x-1) - y(x-1) = g(x) + \int_0^x(\cos(x)ty(t) + \sin(x)ty'(t))dt + \int_0^{x+1}\sin(x)ty'(t)dt,\quad \begin{cases} y(0)=1, \\ y'(0)=1. \end{cases} \tag{14}$$

Here we have $g(x) = 0.5\sin(1)\,x + x\sin(0.5x-1) - \cos(x+1) - \cos(2x) - 0.5\cos(2x+1) - 0.5\sin(2x+1) - x\sin(2x) - 0.5x\sin(2x+1) + 0.5\cos(1) + 0.5\sin(1)$. Exact solution is known to be $y_{\text{exact}}(x) = \cos(x)$. We have solved this problem using various

choices for the parameter $N$. As an illustration, the solutions corresponding to $N = 3$ and $N = 6$ are given by

$$y_3(x) = 1 - 0.4720061108x^2 + 0.0495159638x^3,$$
$$y_6(x) = 1 - 0.5004728649x^2 - 0.0007124006x^3 + 0.0417241808x^4 + 0.0000382967x^5 - 0.0012101609x^6.$$

The absolute errors of the approximate solutions are depicted in Figure 1. It is seen that increasing $N$ makes the solutions significantly more accurate. In addition, we compare the absolute errors obtained using $N = 4, 8$ with those of Laguerre collocation method [12] and Dickson collocation method [15] in Table 1. It is understood that the present method performs closely to the other two methods for $N = 4$ while it is slightly inferior to them for $N = 8$.

**Table 1.** Comparison of the actual absolute errors of the present method with Laguerre collocation method and Dickson collocation method corresponding to $\boldsymbol{N = 4, 8}$ in Problem (14).

| | Laguerre col. method | | Dickson col. method | | Present method | |
|---|---|---|---|---|---|---|
| $x$ | $\lvert e_4(x)\rvert$ | $\lvert e_8(x)\rvert$ | $\lvert e_4(x)\rvert$ | $\lvert e_8(x)\rvert$ | $\lvert e_4(x)\rvert$ | $\lvert e_8(x)\rvert$ |
| 0.2 | 1.408E − 5 | 2.370E − 7 | 7.037E − 5 | 1.273E − 6 | 4.644E − 5 | 1.626E − 6 |
| 0.4 | 1.270E − 4 | 1.104E − 6 | 2.918E − 4 | 6.516E − 6 | 1.804E − 4 | 8.162E − 6 |
| 0.6 | 1.726E − 4 | 2.936E − 6 | 8.687E − 4 | 1.612E − 5 | 5.967E − 4 | 2.051E − 8 |
| 0.8 | 2.950E − 4 | 6.067E − 6 | 2.174E − 3 | 2.793E − 5 | 1.677E − 3 | 3.703E − 5 |
| 1 | 1.849E − 3 | 1.054E − 5 | 4.598E − 3 | 3.715E − 5 | 3.837E − 3 | 5.308E − 5 |

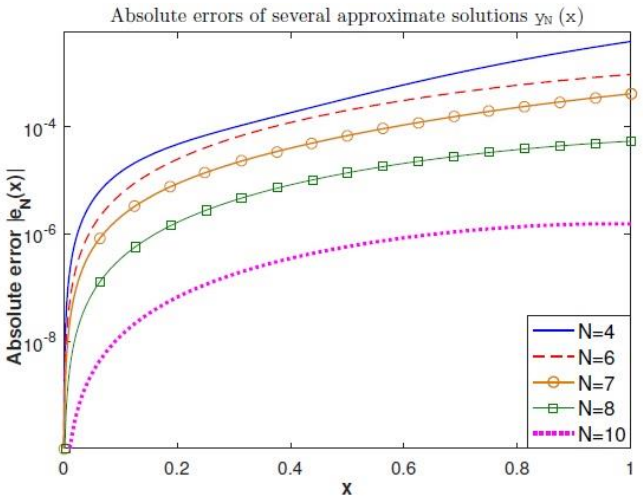


**Figure 1.** Errors of the approximate solutions of Problem (14) obtained by N = 4, 6, 7, 8, 10.

**Example 3:** Lastly, we solve the Volterra delay integro-differential equation from [14, 15]:

$$y'(x) + 2y'(x - 0.5) - y(x) + (x^2 - x)y(0.5x - 1) = g(x) + \int_0^x xe^{-t}y(t)dt + \int_0^{0.5x}((x^2 - 2t - 2)y'(t))dt,\ \text{y(0)=1}. \quad (15)$$

Here, $g(x) = (x^2 - x)e^{0.5x-1} + 2e^{x-0.5} - x^2e^{0.5x} + xe^{0.5x}$ and the exact solution is $y_{\text{exact}}(x) = e^x$. We have solved this problem using $N = 4$ and $N = 7$. We also compared the absolute errors with Chelyshkov collocation method [14] and Dickson collocation method [15] in Table 2.

**Table 2.** Comparison of the actual absolute errors of the present method with Chelyshkov collocation method and Dickson collocation method corresponding to $\boldsymbol{N = 4, 7}$ in Problem (15).

| | Chelyshkov col. method | | Dickson col. method | | Present method | |
|---|---|---|---|---|---|---|
| $x$ | $\lvert e_4(x)\rvert$ | $\lvert e_7(x)\rvert$ | $\lvert e_4(x)\rvert$ | $\lvert e_7(x)\rvert$ | $\lvert e_4(x)\rvert$ | $\lvert e_7(x)\rvert$ |
| 0.2 | 2.926E − 4 | 7.378E − 6 | 1.738E − 4 | 7.414E − 7 | 2.619E − 4 | 5.775E − 7 |
| 0.4 | 2.704E − 4 | 1.773E − 6 | 5.303E − 5 | 1.355E − 6 | 4.195E − 4 | 1.746E − 6 |
| 0.6 | 4.497E − 4 | 1.851E − 5 | 4.756E − 4 | 4.143E − 7 | 9.470E − 5 | 1.685E − 6 |
| 0.8 | 1.740E − 3 | 2.969E − 5 | 8.732E − 4 | 1.903E − 6 | 6.680E − 4 | 7.401E − 7 |
| 1 | 2.432E − 3 | 2.117E − 6 | 5.833E − 4 | 3.347E − 6 | 8.687E − 4 | 3.357E − 6 |

Let us now improve the approximate solutions corresponding to $N = 4$ and $N = 8$ by using residual correction. Solving the corresponding error problems using $M = 6$ and $M = 8$ yields

$$e_{4,6}(x) = 0.0008877708x + 0.0036931793x^2 - 0.0055931160x^3 - 0.0101119108x^4 + 0.0085305959x^5 + 0.0017169189x^6,$$

$$e_{7,8}(x) = -0.0000005664x - 0.0000139856x^2 - 0.0000059217x^3 + 0.0000398116x^4 + 0.0000331595x^5 - 0.0000382998x^6 - 0.0000413434x^7 + 0.0000304212x^8.$$

We then use these error estimates to obtain the corrected approximate solutions $y_{4,6}$ and $y_{7,8}$ as

$$y_{4,6}(x) = y_4(x) + e_{4,6}(x),\ y_{7,8}(x) = y_7(x) + e_{7,8}(x),$$

The accuracy of these corrected solutions depends directly on the accuracy of the error estimates $e_{4,6}$ and $e_{7,8}$. For this reason, it is a good idea to sketch the graphs of these estimates versus the original actual errors $e_4$ and $e_7$. This is done in Figure 2, where it is seen that both estimates are reasonably close to the original actual errors. In addition, the actual errors are given together with their estimates in Table 3. We can infer that the accuracy of the error estimates has given rise to reduced error values exhibited by the improved solutions, shown in fourth and seventh columns. The reader may find it helpful to compare this situation to the note closing Section 3.

**Table 3.** Actual errors of the original aproximate solutions $\boldsymbol{y_4}$ and $\boldsymbol{y_7}$ compared with their estimations and improved versions obtained using $\boldsymbol{M = 6}$ and $\boldsymbol{M = 8}$ in Problem (15).

| | Errors corresponding to $(N, M) = (4,6)$ | | | Errors corresponding to $(N, M) = (7,8)$ | | |
|---|---|---|---|---|---|---|
| $x$ | $e_4(x)$ | $e_{4,6}(x)$ | $E_{4,6(x)}$ | $e_7(x)$ | $e_{7,8}(x)$ | $E_{7,8}(x)$ |
| 0.2 | 2.619E − 4 | 2.672E − 4 | −5.296E − 6 | −5.775E − 7 | −6.486E − 7 | 7.118E − 8 |
| 0.4 | 4.195E − 4 | 4.235E − 4 | −3.988E − 6 | −1.746E − 6 | −1.689E − 6 | −5.679E − 8 |
| 0.6 | 9.470E − 5 | 8.703E − 5 | 7.666E − 6 | −1.685E − 6 | −1.349E − 6 | −3.368E − 7 |
| 0.8 | −6.680E − 4 | −6.862E − 4 | 1.823E − 5 | 7.401E − 7 | 1.130E − 6 | −3.899E − 7 |
| 1 | −8.687E − 4 | −8.765E − 4 | 7.773E − 6 | 3.357E − 6 | 3.275E − 6 | 8.191E − 8 |

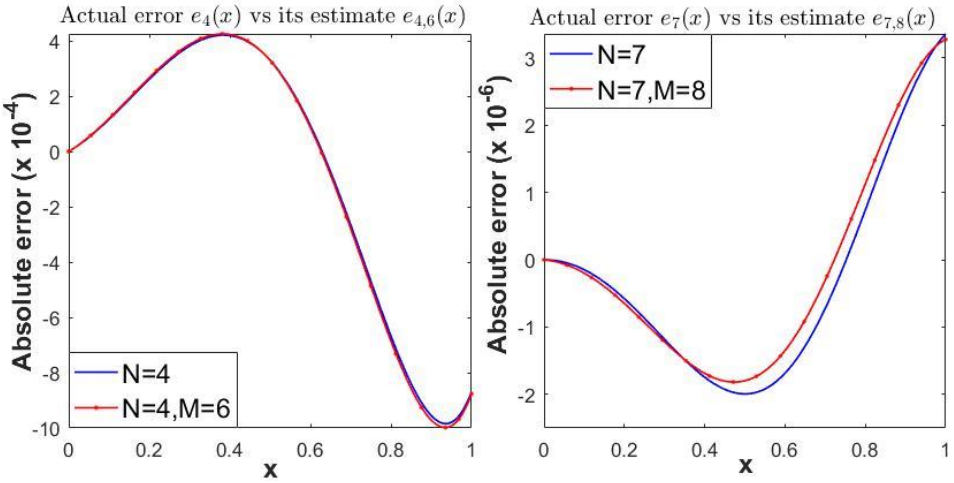


**Figure 2.** Comparison of the actual errors $\boldsymbol{e_4}$ and $\boldsymbol{e_7}$ with their estimations in Problem (15).

## 5. CONCLUSION

In this paper, we have presented a Galerkin-like numerical scheme in order to find approximate solutions of linear Volterra-Fredholm delay integro-differential equations with an integral term having functional upper limit. Simulation results reveal that increasing the parameter $N$ of the method yields significantly more accurate solutions. In addition, the present scheme has the virtue of yielding the exact solution in case that solution is a polynomial. Residual error correcting technique, which makes it possible to improve the accuracy of an obtained approximate solution, has been tested on an example problem with satisfactory results. Comparisons with several other popular methods exhibit results which are rather close or in favour of the present scheme. A drawback of the present scheme is the possible impossibility of the exact evaluation of the relevant integrals for certain types of kernel functions; but this issue can be overcome by the use of a suitable numerical integration technique. Considering all these factors, one can rely upon the presented scheme when the need arises to obtain accurate solutions of linear integro-differential equations of similar type.

## REFERENCES

[1] Kapur J.N. (2005) Mathematical Modelling, *New Age International Ltd.*, New Delhi, India.

[2] Jackiewicz Z., Rahman M. and Welfert B.D. (2006) Numerical solution of a Fredholm integro-differential equation modelling neural networks, *Appl. Numer. Math.* 56, 423-432.

[3] Bates A.P., Khalid Z. and Kennedy R.A. (2017) Slepian Spatial-Spectral Concentration Problem on the Sphere: Analytical Formulation for Limited Colatitude-Longitude Spatial Region, *IEEE T. Signal Proces.* 65 (6), 1527-1537.

[4] Scudo F.M. (1971) Vito Volterra and theoretical ecology, *Theor. Popul. Biol.* 2, 1-23.

[5] MacCamy R.C. (1977) An integro-differential equation with application in heat flow, *Q. Appl. Math.* 35 (1), 1-19.

[6] Diekman O. (1978) Thresholds and traveling waves for the geographical spread of infection, *J. Math. Biol.* 6, 109-130.

[7] Abdou M.A. (2003) On asymptotic methods for Fredholm-Volterra integral equation of the second kind in contact problems, *J. Comput. Appl. Math.* 154, 431-446.

[8] Gülsu M. and Sezer M. (2016) Taylor collocation method for solution of systems of high-order linear Fredholm-Volterra integro-differential equations, *Int. J. Comput. Math.* 83 (4), 429-448.

[9] Yüzbaşı Ş., Şahin N. and Yıldırım A. (2012) A collocation approach for solving high-order linear Fredholm-Volterra integro-differential equations, *Math. Comput. Model.* 55, 547-563.

[10] Yüzbaşı Ş. and Ismailov N. (2018) An operational matrix method for solving linear Fredholm-Volterra integro-differential equations, *Turk. J. Math.* 42, 243-256.

[11] Taiwo O.A. and Adio A.K. (2014) Variational Iteration and Homotopy Perturbation Methods for Solving Fredholm-Volterra Integro-Differential Equations, *IJMSI* 2 (1), 49-55.

[12] Yüzbaşı Ş. (2014) Laguerre approach for solving pantograph-type Volterra integro-differential equations, *Appl. Math. Comput.* 232, 1183-1199.

[13] Wang K. and Wang Q. (2014) Taylor polynomial method and error estimation for a kind of mixed Volterra-Fredholm integral equations, *Appl. Math. Comput.* 229, 53-59.

[14] Oğuz C. And Sezer M. (2015) Chelyshkov collocation method for a class of mixed functional integro-differential equations, *Appl. Math. Comput.* 259, 943-954.

[15] Kürkçü Ö.K., Aslan E. And Sezer M. (2016) A numerical approach with error estimation to solve general integro-differential-difference equations using Dickson polynomials, *Appl. Math. Comput.* 259, 324-339.

[16] Saaedi L., Tari A. and Babolian E. (2019) Numerical Solution of Functional Volterra-Hammerstein Integro-differential Equations of Order Two, *Iranian Journal of Science and Technology Transactions A – Science* 43 (3), 2981-2992.

[17] Türkyılmazoğlu M. (2014) An effective approach for numerical solutions of high-order Fredholm integro-differential equations, *Appl. Math. Comput.* 227, 384-398.

[18] Yüzbaşı Ş. and Karaçayır M. (2016) A Galerkin-like approach to solve high-order integro-differential equations with weakly singular kernel, *Kuwait J. Sci.* 43 (2), 106-120.

[19] Yüzbaşı Ş. and Karaçayır M. (2018) A Galerkin-Type Method for Solving a Delayed Model on HIV Infection of $CD4^+$ T-cells, *Iranian Journal of Science and Technology Transactions A – Science* 42 (2), 1087-1095.

[20] Türkyılmazoğlu M. (2013) Effective computation of exact and analytic approximate solutions to singular nonlinear equations of Lane-Emden-Fowler type, *Appl. Math. Model.* 37, 7539-7548.

[21] Türkyılmazoğlu M. (2014) Effective Computation of Solutions for Nonlinear Heat Transfer Problems in Fins, *J. Heat Transfer* 136 (9), 091901.

[22] Du Croz J.J. and Higham N.J. (1992) Stability of Methods for Matrix Inversion, *IMA Journal of Numerical Analysis* 12 (1), 1-19.

[23] Amat S., Busquier S. and Gutiérrez J.M. (2003) Geometric construction of iterative functions to solve nonlinear equations, *J. Comput. Appl. Math.* 153, 197-205.

[24] Li H.B., Huang T.Z., Zhang Y., Liu X.P. and Gu T.X. (2011) Chebyshev-type methods and preconditioning techniques, *Appl. Math. Comput.* 218 (2), 260-270.