Turkish Journal of Mathematics

Volume 46 | Number 8 Article 5

1-1-2022

A Galerkin-type approach to solve systems of linear Volterra-Fredholm integro-differential equations

MURAT KARAÇAYIR

ŞUAYİP YÜZBAŞI

Recommended Citation

KARAÇAYIR, MURAT and YÜZBAŞI, ŞUAYİP (2022) "A Galerkin-type approach to solve systems of linear Volterra-Fredholm integro-differential equations," *Turkish Journal of Mathematics*: Vol. 46: No. 8, Article 5.
https://doi.org/10.3906/tar-2206-78
Available at: https://journals.tubitak.gov.tr/math/vol46/iss8/5

Turkish Journal of Mathematics

http://journals.tubitak.gov.tr/math/

Research Article

Turk J Math
(2022) 46: 3121 – 3138
© TÜBİTAK
doi:10.55730/1300-0098.3323

A Galerkin-type approach to solve systems of linear Volterra-Fredholm integro-differential equations

Murat KARAÇAYIR, **Şuayip YÜZBAŞI***
Department of Mathematics, Faculty of Science, Akdeniz University, Antalya, Turkey

Received: 19.06.2022 • **Accepted/Published Online:** 15.08.2022 • **Final Version:** 09.11.2022

Abstract: The main interest of this paper is to propose a numerical scheme in order to solve linear systems of Volterra-Fredholm integro-differential equations given with mixed conditions. The proposed method is a weighted residual scheme which uses monomials up to a prescribed degree N as the basis functions. By taking inner product of the equation system with the elements of this basis set in a Galerkin-like fashion, the original problem is transformed into a linear algebraic equation system. After a suitable incorporation of the mixed conditions, a final algebraic system is obtained, from which the approximate solutions of the problem are computed. The proposed numerical scheme is illustrated with example problems taken from the literature.

Key words: Volterra-Fredholm integro-differential equations, systems of linear integro-differential equations, method of weighted residuals, Galerkin method, method of moments, numerical solutions

1. Introduction

Integro-differential equations arise in situations where the rate of change of a quantity depends on the history of that quantity throughout a continuous time interval as well as on its present state. Many applications in science and engineering possess this property, resulting in model problems that involve integro-differential equations. Such applications include circuit analysis [22, p. 263], computational neuroscience [16], heat flow [19], plate equation with memory [4] and image processing [5]. There are also several textbooks on the theory and applications of integro-differential equations. The reader who is interested in such treatises can see [18].

In some problems where integro-differential equations are relevant, there are more than one unknown functions involved, resulting in the problem being modeled as a system of integro-differential equations. Such applications include the relation between populations of immune cells and tumor [7], boundary value problems in electromagnetic theory [9], modeling of fluid waves with an application to oceanography [1], population dynamics [12] and statistics of polymer chains [17].

In this study, we consider the following system of linear integro-differential equations

$$\sum_{n=0}^{m}\sum_{j=1}^{k} P_{i,j}^{n}(x)y_j^{(n)}(x) = \int_a^b \sum_{n=0}^{m}\sum_{j=1}^{k} K_{i,j}^{n}(x,t)y_j^{(n)}(t)dt + \int_a^x \sum_{n=0}^{m}\sum_{j=1}^{k} L_{i,j}^{n}(x,t)y_j^{(n)}(t)dt + g_i(x),\ i=1,2,\dots,k \tag{1.1}$$

*Correspondence: syuzbasi@akdeniz.edu.tr
2010 *AMS Mathematics Subject Classification:* 45J05, 65L60

in the interval $a \le x \le b$ and under the mixed initial and boundary conditions

$$\sum_{j=0}^{m-1} \left(a_{i,j}^{n} y_n^{(j)}(a) + b_{i,j}^{n} y_n^{(j)}(b) \right) = \lambda_{n,i}, \; i = 1, 2, \ldots, m, \; n = 1, 2, \ldots, k. \tag{1.2}$$

In (1.1), the subscript i corresponds to the equations and the subscript j corresponds to the unknown functions; hence it is a system of k equations in k unknown functions. The degree of the highest derivative of the unknown functions in each of the k equations is equal to at most m, which means that each equation is of m-th or smaller order. $P_{i,j}^{n}$ are given functions of one variable defined in $[a, b]$ whereas the kernel functions $K_{i,j}^{n}$ and $L_{i,j}^{n}$ are known functions of two variables defined in the domain $[a, b] \times [a, b]$ for $i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, k$, $n = 0, 1, \ldots, m$. In addition, the nonhomogeneous terms g_i are known continuous functions of one variable over the interval $[a, b]$. The integrals having constant upper limit b are the Fredholm terms of the system, whereas those given with variable upper limit x constitutes the Volterra terms. Therefore, (1.1) is a system of linear Volterra-Fredholm integro-differential equations. As for the km conditions given by (1.2), they are of mixed type in the sense that some of them may contain terms evaluated at both boundary points a and b. Whenever suitable, we will refer to the system (1.1) given with the conditions (1.2) as problem (1.1)–(1.2).

As of today, there does not exist a known method that exactly solves the problem (1.1)–(1.2); therefore numerical techniques are required when one needs its solutions. Indeed, many researchers have been interested in obtaining numerical solutions of integro-differential equations using a variety of methods. To name a few of such studies, Biazar [8] used the well-known Adomian decomposition method to solve an initial value version of problem (1.1)–(1.2) where the left-hand side contains only the first derivative of the unknown functions. In [21], Saberi-Nadjafi and Tamamgar used the still more popular variational iteration method in order to solve problems of type (1.1)–(1.2). Another popular method, He's homotopy perturbation method, was employed for the same purpose by Yusufoğlu in [26]. In [2], Akyüz-Daşcıoğlu and Sezer obtained Chebyshev polynomial solutions of problem (1.1)–(1.2) by using Chebyshev nodes as collocation points. Hesameddini and Shahbazi solved the same problem using collocation approach in conjunction with hybrid Block-Pulse functions based on Bernstein polynomials [14]. Other schemes based on collocation points were employed by Sezer and Gülsu using Taylor polynomials [13], by Caliò et al. [11] who utilized a certain type of spline approximation functions and by Yüzbaşı using Bessel polynomials [27]. Lastly, operational matrix method in connection with Bernstein polynomials was applied to a version of problem (1.1)–(1.2) with slightly different mixed conditions by Maleknejad et al. in [20]. Apart from these numerical studies, Fredholm-Volterra integral or integro-differential equations and their various variants have extensively been studied from a qualitative point of view. Among recent studies in this category, we refer the reader to [24] and [23], where fundamental properties of certain types of nonlinear integro-delay differential equation systems such as stability, boundedness and integrability are investigated. In addition, stability and boundedness properties of solutions of a new mathematical model involving Volterra integro-differential equations with Caputo fractional derivative and constant delays have been studied in [10].

Our interest in this paper is to solve problem (1.1)–(1.2) by using a weighted residual scheme reminiscent of the Galerkin method. We will use standard polynomials up to a prescribed degree as the set of basis functions, meaning that the approximate solutions that will be obtained as a result of the method are polynomials. Similar numerical schemes were used in the case of single high-order Fredholm integro-differential equations [25] and Fredholm integro-differential equations with weakly singular kernel [28], Lotka-Volterra predator-prey population model [29] and linear Volterra integro-delay differential equations [30] in addition to systems of linear differential and integral equations [15]. Although there is a large number of numerical methods in the literature that can be used to solve integro-differential equations, the method presented in this paper has the virtue that it can easily be adapted to problems of different kind and it also improves on most of the other proposed schemes in terms of accuracy. This will be apparent in the succeeding sections.

The organization of the rest of the paper is as follows: Section 2 describes the proposed solution method in detail, whereas Section 3 explains a technique useful in estimating the error of an obtained approximate solution. Numerical examples are considered in Section 4. Finally, Section 5 contains some comments regarding the effectivity of the proposed scheme based on the obtained numerical results.

2. Method of solution

This section outlines the method that we will use to solve problems of type (1.1)–(1.2). In the sequel, we will often express the mathematical identities in their matrix forms since this makes it easier to program the numerical scheme in computer. In order to help the reader follow the method more easily, we will divide it into four steps. Step 1 consists of replacing the unknown functions in system (1.1) by their approximate forms (will be defined in the sequel) and hence obtain an approximate matrix version of the system. In step 2, we convert this matrix version to a system of linear algebraic equations by applying inner product with weight functions (will be defined in the sequel). In step 3, we feed the mixed conditions (1.2) to this system and thus obtain a modified system. In step 4, we obtain the approximate solutions of problem (1.1)–(1.2) using the solutions of this system.

Our only assumption on the solutions of problem (1.1)–(1.2) is that they are expressible in the form of power series in the interval $[a,b]$ as

$$y_i(x) = \sum_{n=0}^{\infty} a_{i,n} x^n = a_{i,0} + a_{i,1}x + a_{i,2}x^2 + \dots,\ i = 1,2,\dots,k.$$

Step 1: The first step of the proposed scheme starts by choosing a positive integer N and truncating the above solutions after their $(N+1)$-st terms, resulting in the approximate solutions

$$y_i \approx y_{i,N}(x) = \sum_{n=0}^{N} a_{i,n} x^n,\ i = 1,2,\dots,k.$$

As the notation suggests, we denote the i-th approximate solution by $y_{i,N}(x)$, which is a polynomial of degree N with the coefficient of x^n being denoted by $a_{i,n}$. The goal of the method is to determine these coefficients, which are also called the "weights". We call the monomials $1, x, x^2, \dots, x^N$ up to degree N the "basis functions" since each approximate solution is a linear combination of them. Our main task in this step is to impose on the approximate solutions $y_{i,N}$ the condition of satisfying system (1.1). To make this task easier, let us define some auxiliary matrices that will simplify the notation. Namely, we define

$$\mathbf{X}(x) = [\ 1 \quad x \quad x^2 \quad \dots \quad x^N\ ],\ \mathbf{A}_i = [\ a_{i,0} \quad a_{i,1} \quad \dots \quad a_{i,N}\ ]^T,\ i = 1,2,\dots,k,$$

where T denotes transpose. Under this setting, the approximate solutions can be expressed as

$$y_{i,N}(x) = \mathbf{X}(x)\mathbf{A}_i,\ i = 1,2,\dots,k.$$

Likewise, derivatives of the approximate solutions can also be expressed in terms of a matrix multiplication. This is achieved by defining a $(N+1)\times(N+1)$ square matrix, which we denote by $\mathbf{B}$, with entries $\mathbf{B}_{i,i+1} = i$ for $i = 1,2,\dots,N$ and $\mathbf{B}_{i,j} = 0$ for all other entries. More explicitly, the matrix

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

enables us to express the derivatives as $y_{i,N}^{(n)}(x) = \mathbf{X}(x)\mathbf{B}^n\mathbf{A}_i$ for $i = 1, 2, \ldots, k$ and $n = 1, 2, \ldots, m$. Note that this identity is also valid for $n = 0$; it reduces to $y_{i,N}(x) = \mathbf{X}(x)\mathbf{A}_i$ in this case. Next, we substitute the matrix expressions for the approximate solutions $y_{i,N}$ and their derivatives in the original system (1.1). The resulting equations are as follows:

$$\sum_{n=0}^{m}\sum_{j=1}^{k} P_{i,j}^{n}(x)\mathbf{X}(x)\mathbf{B}^n\mathbf{A}_j - \int_a^b \sum_{n=0}^{m}\sum_{j=1}^{k} K_{i,j}^{n}(x,t)\mathbf{X}(t)\mathbf{B}^n\mathbf{A}_j dt$$

$$- \int_a^x \sum_{n=0}^{m}\sum_{j=1}^{k} L_{i,j}^{n}(x,t)\mathbf{X}(t)\mathbf{B}^n\mathbf{A}_j dt = -g_i(x),\ i = 1, 2, \ldots, k.$$

Alternatively, grouping the unknown weights corresponding to the same unknown function, we can rewrite the above as

$$\sum_{j=1}^{k}\left(\sum_{n=0}^{m}\left(P_{i,j}^{n}(x)\mathbf{X}(x) - \int_a^b K_{i,j}^{n}(x,t)\mathbf{X}(t)dt - \int_a^x L_{i,j}^{n}(x,t)\mathbf{X}(t)dt\right)\mathbf{B}^n\right)\mathbf{A}_j$$
$$= \mathbf{F}_{i,1}(x)\mathbf{A}_1 + \mathbf{F}_{i,2}(x)\mathbf{A}_2 + \ldots + \mathbf{F}_{i,k}(x)\mathbf{A}_k = -g_i(x),\ i = 1, 2, \ldots, k, \tag{2.1}$$

where the notation $\mathbf{F}_{i,j}(x) := \sum_{n=0}^{m}\left(P_{i,j}^{n}(x)\mathbf{X}(x) - \int_a^b K_{i,j}^{n}(x,t)\mathbf{X}(t)dt - \int_a^x L_{i,j}^{n}(x,t)\mathbf{X}(t)dt\right)\mathbf{B}^n$ for $j = 1, 2, \ldots, k$ have been adopted for convenience. Here, each $\mathbf{F}_{i,j}(x)$ is a row vector of length $N+1$ whose entries are functions of x. It is possible to express the left-hand side of (2.1) as a single matrix multiplication by vertically concatenating the column vectors $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$. In addition, we construct $(N+1) \times k(N+1)$ matrices $\mathcal{I}_j$ from the $(N+1) \times (N+1)$ identity matrix $\mathbf{I}$ for $j = 1, 2, \ldots, k$. Namely, we define

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_k \end{bmatrix}, \quad \mathcal{I}_j = \Big[\underbrace{\mathbf{0} \ \cdots \ \mathbf{0}}_{j-1 \text{ times}}\ \mathbf{I}\ \underbrace{\mathbf{0} \ \cdots \ \mathbf{0}}_{k-j \text{ times}}\Big],\ j = 1, 2, \ldots, k.$$

In the definition of $\mathcal{I}_j$, $\mathbf{0}$ denotes the all-zero matrix of size $(N+1) \times (N+1)$. This new setting suggests that (2.1) can be written for a fixed i as

$$\left(\sum_{j=1}^{k} \mathbf{F}_{i,j}(x)\mathcal{I}_j\right)\mathbf{A} = \mathbf{F}_i(x)\mathbf{A} = g_i(x). \tag{2.2}$$

This is the i-th equation in the original system (1.1) with the unknown functions y_j replaced by their polynomial approximations $y_{j,N}$. Since there are a total of k equations in (1.1), the whole system can be expressed by vertically concatenating all k matrix equations of the from (2.2); namely by

$$\begin{bmatrix} \mathbf{F}_1(x) \\ \mathbf{F}_2(x) \\ \vdots \\ \mathbf{F}_k(x) \end{bmatrix}\mathbf{A} = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_k(x) \end{bmatrix} \text{ or } \mathbf{F}(x)\mathbf{A} = \mathbf{g}(x). \tag{2.3}$$

Here, $\mathbf{F}(x)$ is a $k \times k(N+1)$ matrix whose entries are functions of x. Thus, (2.3) can be viewed as $N+1$ functional equations in the $k(N+1)$ unknowns $a_{i,j}$. This completes step 1.

Step 2: Now, we have the approximate matrix version (2.3) of the original system (1.1). The goal of this step is to transform (2.3) to a linear algebraic system whose unknowns are the weights $a_{i,j}$. This is achieved by taking inner product of (2.3) with the weight functions $1, x, x^2, \ldots, x^N$ (which are the same as basis functions*) one at a time. This inner product is defined in the space $L^2[a,b]$ of square integrable functions on a real interval. More explicitly, if f and g are functions from $L^2[a,b]$, then their inner product is given by

$$< f, g >= \int_a^b f(t)g(t)\mathrm{d}t.$$

Recall that each line of (2.3) is an equation of the form $\mathbf{F}_i(x)\mathbf{A} = g_i(x)$ for some $i = 1, 2, \ldots, k$. Now, to each of these k equations we apply inner product with each of the weight functions $1, x, x^2, \ldots, x^N$. Due to the linearity of the inner product, this results in the following:

$$< \mathbf{F}_i(x), x^j > \mathbf{A} =< g_i(x), x^j >, \; i = 1, 2, \ldots, k, \; j = 0, 1, \ldots, N.$$

For a fixed i and a fixed j, this is a linear algebraic equation whose unknowns are the $k(N+1)$ unknown weights in $\mathbf{A}$. Since there are k equations and $N+1$ basis functions in total, this process results in a system of $k(N+1)$ algebraic equations in $k(N+1)$ unknowns. Using the linearity of inner product, we can shortly write this system as

$$\mathbf{WA} = \mathbf{G}, \tag{2.4}$$

where the entries of $\mathbf{W}$ and $\mathbf{G}$ are determined by

$$\mathbf{W}_{jk+i,l} =< \mathbf{F}_i(x)_l, x^j >, \; \mathbf{G}_{jk+i,1} =< g_i(x), x^j >,$$

or more explicitly by

$$\mathbf{W} = \begin{bmatrix} < \mathbf{F}_1(x)_1, 1 > & < \mathbf{F}_1(x)_2, 1 > & \cdots & < \mathbf{F}_1(x)_{k(N+1)}, 1 > \\ < \mathbf{F}_2(x)_1, 1 > & < \mathbf{F}_2(x)_2, 1 > & \cdots & < \mathbf{F}_2(x)_{k(N+1)}, 1 > \\ \vdots & \vdots & \cdots & \vdots \\ < \mathbf{F}_k(x)_1, 1 > & < \mathbf{F}_k(x)_2, 1 > & \cdots & < \mathbf{F}_k(x)_{k(N+1)}, 1 > \\ < \mathbf{F}_1(x)_1, x > & < \mathbf{F}_1(x)_2, x > & \cdots & < \mathbf{F}_1(x)_{k(N+1)}, x > \\ < \mathbf{F}_2(x)_1, x > & < \mathbf{F}_2(x)_2, x > & \cdots & < \mathbf{F}_2(x)_{k(N+1)}, x > \\ \vdots & \vdots & \cdots & \vdots \\ < \mathbf{F}_k(x)_1, x > & < \mathbf{F}_k(x)_2, x > & \cdots & < \mathbf{F}_k(x)_{k(N+1)}, x > \\ \vdots & \vdots & \cdots & \vdots \\ < \mathbf{F}_1(x)_1, x^N > & < \mathbf{F}_1(x)_2, x^N > & \cdots & < \mathbf{F}_1(x)_{k(N+1)}, x^N > \\ < \mathbf{F}_2(x)_1, x^N > & < \mathbf{F}_2(x)_2, x^N > & \cdots & < \mathbf{F}_2(x)_{k(N+1)}, x^N > \\ \vdots & \vdots & \cdots & \vdots \\ < \mathbf{F}_k(x)_1, x^N > & < \mathbf{F}_k(x)_2, x^N > & \cdots & < \mathbf{F}_k(x)_{k(N+1)}, x^N > \end{bmatrix}, \; \mathbf{G} = \begin{bmatrix} < g_1(x), 1 > \\ < g_2(x), 1 > \\ \vdots \\ < g_k(x), 1 > \\ < g_1(x), x > \\ < g_2(x), x > \\ \vdots \\ < g_k(x), x > \\ \vdots \\ < g_1(x), x^N > \\ < g_2(x), x^N > \\ \vdots \\ < g_k(x), x^N > \end{bmatrix}$$

for $i = 1, 2, \ldots, k, \; j = 0, 1, \ldots, N, \; l = 1, 2, \ldots, k(N+1)$ and $\mathbf{F}_i(x)_l$ denotes the l-th entry of the vector $\mathbf{F}_i(x)$. Since we have obtained the linear algebraic system (2.4), step 2 is complete.

Step 3: The solutions of the system (2.4) do not yield the approximate solutions $y_{i,N}$, since the mixed conditions (1.2) have not been taken into account yet. In this step we incorporate these conditions into system

*In general weight functions do not have to be same as the basis functions. In Galerkin method they are taken to be the same.

(2.4). There is a total of km mixed conditions, each of which imposes a linear equation in the unknown weights $a_{i,j}$. These equations can easily be expressed in terms of matrices, in a similar fashion to what we did in step 1. In the context of mixed conditions (1.2), the superscript n stands for the n-th unknown function y_n and the subscript i enumerates the m distinct mixed conditions corresponding to the same unknown function. The i-th mixed condition for the n-th approximate solution $y_{n,N}$ can be expressed as

$$\left[\sum_{j=0}^{m-1}\left(a_{i,j}^{n}\mathbf{X}(a)+b_{i,j}^{n}\mathbf{X}(b)\right)\mathbf{B}^{j}\right]\mathbf{A}_n=\left[\sum_{j=0}^{m-1}\left(a_{i,j}^{n}\mathbf{X}(a)+b_{i,j}^{n}\mathbf{X}(b)\right)\mathbf{B}^{j}\right]\mathcal{I}_n\mathbf{A}=\mathbf{C}_{n,i}\mathbf{A}=\lambda_{n,i},$$

where the matrix $\mathcal{I}_n$ is as before and we have defined $\mathbf{C}_{n,i}:=\left[\sum_{j=0}^{m-1}\left(a_{i,j}^{n}\mathbf{X}(a)+b_{i,j}^{n}\mathbf{X}(b)\right)\mathbf{B}^{j}\right]\mathcal{I}_n$. Clearly, the row vector $\mathbf{C}_{n,i}$, which represents the i-th condition corresponding to $y_{n,N}$, has length $k(N+1)$. Concatenating all vectors that represent the mixed conditions, we can express all of them by

$$\begin{bmatrix}\mathbf{C}_{1,1}\\ \vdots\\ \mathbf{C}_{1,m}\\ \vdots\\ \mathbf{C}_{k,1}\\ \vdots\\ \mathbf{C}_{k,m}\end{bmatrix}\mathbf{A}=\begin{bmatrix}\lambda_{1,1}\\ \vdots\\ \lambda_{1,m}\\ \vdots\\ \lambda_{k,1}\\ \vdots\\ \lambda_{k,m}\end{bmatrix}\quad\text{or } \mathbf{CA}=\mathbf{\Lambda}. \tag{2.5}$$

The systems (2.5) contains all the mixed conditions and has dimension $km\times k(N+1)$. In order to ensure that the approximate solutions $y_{i,N}$ satisfy the mixed conditions, we must combine system (2.5) with the previously obtained system (2.4). Since there should be exactly as many equations as unknowns, we incorporate all km mixed conditions by replacing this number of rows of system (2.4) by system (2.5). In order to be deterministic, let us replace the last km rows of $\mathbf{W}$ by $\mathbf{C}$ and those of $\mathbf{G}$ by $\mathbf{\Lambda}$. More explicitly, we form a new system

$$\tilde{\mathbf{W}}\mathbf{A}=\tilde{\mathbf{G}},$$

where

$$\tilde{\mathbf{W}}=\begin{bmatrix}\mathbf{W}_1\\ \vdots\\ \mathbf{W}_{k(N+1-m)}\\ \mathbf{C}\end{bmatrix},\ \tilde{\mathbf{G}}=\begin{bmatrix}\mathbf{G}_1\\ \vdots\\ \mathbf{G}_{k(N+1-m)}\\ \mathbf{\Lambda}\end{bmatrix}.$$

Here $\mathbf{W}_i$ and $\mathbf{G}_i$ denote the rows of $\mathbf{W}$ and $\mathbf{G}$.

Step 4: This step only consists of solving the final system $\tilde{\mathbf{W}}\mathbf{A}=\tilde{\mathbf{G}}$. Provided that the system matrix $\tilde{\mathbf{W}}$ has full rank, the matrix of unknown weights can be computed by $\mathbf{A}=\tilde{\mathbf{W}}^{-1}\tilde{\mathbf{G}}$. Finally, the approximate solutions are found by $y_{i,N}(x)=\sum_{n=0}^{N}a_{i,n}x^n$ for $i=1,2,\ldots,k$.

3. Error estimation for approximate solutions

In order to evaluate the accuracy of any numerical method, the straightforward way is to consider the absolute error of the approximate solution. In our context, if the exact solutions of problem (1.1)–(1.2) are denoted by $y_{i,\text{exact}}(x)$, the actual errors of the approximate solutions $y_{i,N}(x)$ are given by $e_{i,N}(x)=y_{i,\text{exact}}(x)-y_{i,N}(x)$ for

$i = 1, 2, \ldots, k$. However, this measure of accuracy is not applicable most of the time due to the unavailability of the exact solution. In such cases, the residual of the approximate solution, which is what remains when we substitute it in the original equation, can be used as a measure of accuracy. The residuals of the approximate solutions of problem (1.1)–(1.2) are given by

$$R_{i,N}(x) = \sum_{n=0}^{m}\sum_{j=1}^{k} P_{i,j}^{n}(x) y_{j,N}^{(n)}(x) - \int_a^b \sum_{n=0}^{m}\sum_{j=1}^{k} K_{i,j}^{n}(x,t) y_{j,N}^{(n)}(t)dt - \int_a^x \sum_{n=0}^{m}\sum_{j=1}^{k} L_{i,j}^{n}(x,t) y_{j,N}^{(n)}(t)dt \\ - g_i(x) \tag{3.1}$$

for $i = 1, 2, \ldots, k$. In addition, since $y_{i,\text{exact}}$ are the exact solutions, they satisfy system (1.1); therefore it is true that

$$\sum_{n=0}^{m}\sum_{j=1}^{k} P_{i,j}^{n}(x) y_{j}^{(n)}(x) - \int_a^b \sum_{n=0}^{m}\sum_{j=1}^{k} K_{i,j}^{n}(x,t) y_{j}^{(n)}(t)dt - \int_a^x \sum_{n=0}^{m}\sum_{j=1}^{k} L_{i,j}^{n}(x,t) y_{j}^{(n)}(t)dt - g_i(x) = 0 \tag{3.2}$$

for $i = 1, 2, \ldots, k$. Subtracting (3.1) from (3.2) yields

$$\sum_{n=0}^{m}\sum_{j=1}^{k} P_{i,j}^{n}(x) e_{j,N}^{(n)}(x) = \int_a^b \sum_{n=0}^{m}\sum_{j=1}^{k} K_{i,j}^{n}(x,t) e_{j,N}^{(n)}(t)dt + \int_a^x \sum_{n=0}^{m}\sum_{j=1}^{k} L_{i,j}^{n}(x,t) e_{j,N}^{(n)}(t)dt \\ - R_{i,N}(x),\ i = 1, 2, \ldots, k, \tag{3.3}$$

which is the same as system (1.1) with $g_i(x)$ replaced by negative $-R_{i,N}(x)$ of the residuals of approximate solutions $y_{i,N}$. The unknown functions of the above system are the actual errors $e_{i,N}$ of these approximate solutions. Since both $y_{i,\text{exact}}$ and $y_{i,N}$ satisfy the mixed conditions (1.2) for each $i = 1, 2, \ldots, k$, we have the following homogeneous mixed conditions for the actual errors $e_{n,N}$ where $n = 1, 2, \ldots, k$:

$$\sum_{j=0}^{m-1} \left(a_{n,j}^{i} e_{n,N}^{(j)}(a) + b_{n,j}^{i} e_{n,N}^{(j)}(b) \right) = \sum_{j=0}^{m-1} \left(a_{i,j}^{n} (y_{n,\text{exact}} - y_{n,N})^{(j)}(a) + b_{i,j}^{n} (y_{n,\text{exact}} - y_{n,N})^{(j)}(b) \right) \\ = \sum_{j=0}^{m-1} \left(a_{i,j}^{n} y_{n,\text{exact}}^{(j)}(a) + b_{i,j}^{n} y_{n,\text{exact}}^{(j)}(b) \right) - \sum_{j=0}^{m-1} \left(a_{i,j}^{n} y_{n,N}^{(j)}(a) + b_{i,j}^{n} y_{n,N}^{(j)}(b) \right) = 0,\ i = 1, \ldots, m. \tag{3.4}$$

The system (3.3) considered under the conditions (3.4) is called the *error problem* associated with the problem (1.1)–(1.2). Since the error problem (3.3)–(3.4) is of the same form as the original problem (1.1)–(1.2), we can use the method of Section 2 with the aim of obtaining its approximate solutions. To this end, we choose a parameter value, say M, and apply the numerical scheme to the error problem (3.3)–(3.4). The resulting approximate solution polynomials of degree M, which we will denote by $e_{i,N,M}$ for $i = 1, 2, \ldots, k$, can be regarded as *estimations* for the actual errors $e_{i,N}$. One can consider using these estimations as an assessment regarding the accuracy of the approximate solutions $y_{i,N}$. This will be made more clear during the discussion of the last example problem in Section 4.

Although the error estimates $e_{i,N,M}$ may provide a practical overview of the accuracy of the approximate solution $y_{i,N}$ as explained above, this estimation itself relies on the present numerical scheme; therefore it is plausible to resort to another measure of accuracy. Whenever there is no available exact solution to compare, such a measure is provided by the residual of the approximate solution as given by Equation (3.1). Namely, one can measure the absolute value $|R_{i,N}(x)|$ of the residual $R_{i,N}(x)$ corresponding to the approximate solution polynomial $y_{i,N}(x)$. If the accuracy required for a specific application has been determined as $\varepsilon > 0$, then N can be increased until $|R_{i,N}(x)| < \varepsilon$ for all $x \in [a,b]$.

4. Applications to example problems

This section applies the numerical scheme described in Section 2 to five example problems selected from several past works. In addition, error estimation method explained in Section 3 is applied to one of these problems. All the required calculations have been carried out in MATLAB.

Example 1: Firstly, let us examine the following linear system of Volterra integral equations from [14]:

$$\begin{aligned} y_1(x) &= 1 + x^2 - \frac{x^3+x^4}{6} + \int_0^x \frac{(x-t)^3}{2} y_1(t)\mathrm{d}t + \int_0^x \frac{(x-t)^2}{2} y_2(t)\mathrm{d}t, \\ y_2(x) &= 1 + x - x^3 - \frac{x^7}{840} - \frac{x^4+x^5}{8} + \int_0^x \frac{(x-t)^4}{2} y_1(t)\mathrm{d}t + \int_0^x \frac{(x-t)^3}{2} y_2(t)\mathrm{d}t\ . \end{aligned} \tag{4.1}$$

The exact solution is known to be $y_{1,\text{exact}}(x) = 1 + x^2$, $y_{2,\text{exact}}(x) = 1 + x - x^3$. Let us apply the proposed scheme with the choice $N = 3$. We have

$$y_{1,3}(x) = a_{1,0} + a_{1,1}x + a_{1,2}x^2 + a_{1,3}x^3,\ y_{2,3}(x) = a_{2,0} + a_{2,1}x + a_{2,2}x^2 + a_{2,3}x^3.$$

Replacing y_1 and y_2 by $y_{1,3}$ and $y_{2,3}$, respectively, in system (4.1) and rearranging gives

$$\begin{aligned} a_{1,0} + a_{1,1}x + a_{1,2}x^2 + a_{1,3}x^3 &- \int_0^x \frac{(x-t)^3}{2}(a_{1,0} + a_{1,1}t + a_{1,2}t^2 + a_{1,3}t^3)\mathrm{d}t \\ &- \int_0^x \frac{(x-t)^2}{2}(a_{2,0} + a_{2,1}t + a_{2,2}t^2 + a_{2,3}t^3)\mathrm{d}t = 1 + x^2 - \frac{x^3+x^4}{6}, \\ a_{2,0} + a_{2,1}x + a_{2,2}x^2 + a_{2,3}x^3 &- \int_0^x \frac{(x-t)^4}{2}(a_{1,0} + a_{1,1}t + a_{1,2}t^2 + a_{1,3}t^3)\mathrm{d}t \\ &- \int_0^x \frac{(x-t)^3}{2}(a_{2,0} + a_{2,1}t + a_{2,2}t^2 + a_{2,3}t^3)\mathrm{d}t = 1 + x - x^3 - \frac{x^4+x^5}{8} - \frac{x^7}{840}. \end{aligned}$$

Then, evaluating the integrals we have the following:

$$\begin{aligned} a_{1,0} + a_{1,1}x + a_{1,2}x^2 + a_{1,3}x^3 &- \frac{105a_{1,0}x^4 + 21a_{1,1}x^5 + 7a_{1,2}x^6 + 3a_{1,3}x^7}{840} \\ &- \frac{20a_{2,0}x^3 + 5a_{2,1}x^4 + 2a_{2,2}x^5 + a_{2,3}x^6}{120} = 1 + x^2 - \frac{x^3+x^4}{6}, \\ a_{2,0} + a_{2,1}x + a_{2,2}x^2 + a_{2,3}x^3 &- \frac{168a_{1,0}x^5 + 28a_{1,1}x^6 + 8a_{1,2}x^7 + 3a_{1,3}x^8}{1680} \\ &- \frac{105a_{2,0}x^4 + 21a_{2,1}x^5 + 7a_{2,2}x^6 + 3a_{2,3}x^7}{840} = 1 + x - x^3 - \frac{x^4+x^5}{8} - \frac{x^7}{840}. \end{aligned}$$

The above equations are the equations $\mathbf{F}_1(x)\mathbf{A} = g_1(x)$ and $\mathbf{F}_2(x)\mathbf{A} = g_2(x)$ described at the end of step 1 in Section 2, where

$$\mathbf{F}_1(x) = \left[1 - \frac{105x^4}{840}, \quad x - \frac{21x^5}{840}, \quad x^2 - \frac{7x^6}{840}, \quad x^3 - \frac{3x^7}{840}, \quad -\frac{20x^3}{120}, \quad -\frac{5x^4}{120}, \quad -\frac{2x^5}{120}, \quad -\frac{x^6}{120}\right],$$

$$\mathbf{F}_2(x) = \left[-\frac{168x^5}{1680}, \quad -\frac{28x^6}{1680}, \quad -\frac{8x^7}{1680}, \quad -\frac{3x^8}{1680}, \quad 1 - \frac{105x^4}{840}, \quad x - \frac{21x^5}{840}, \quad x^2 - \frac{7x^6}{840}, \quad x^3 - \frac{3x^7}{840}\right],$$

$$g_1(x) = 1 + x^2 - \frac{x^3 + x^4}{6}, \; g_2(x) = 1 + x - x^3 - \frac{x^4 + x^5}{8} - \frac{x^7}{840}.$$

This marks the end of step 1. Now we apply inner product with the weight functions $1, x, x^2, x^3$ to both of the above equations. In other words, we will multiply both sides of the two equations by the weight functions one at a time and integrate from 0 to 1. Each inner product will result in a linear equation in the unknown weights. Equations resulting from $\mathbf{F}_1(x)\mathbf{A} = g_1(x)$ are

$$\frac{39}{40}a_{1,0} + \frac{119}{240}a_{1,1} + \frac{93}{280}a_{1,2} + \frac{559}{2240}a_{1,3} - \frac{1}{24}a_{2,0} - \frac{1}{120}a_{2,1} - \frac{1}{360}a_{2,2} - \frac{1}{840}a_{2,3} = \frac{151}{120},$$

$$\frac{23}{48}a_{1,0} + \frac{277}{840}a_{1,1} + \frac{239}{960}a_{1,2} + \frac{503}{2520}a_{1,3} - \frac{1}{30}a_{2,0} - \frac{1}{144}a_{2,1} - \frac{1}{420}a_{2,2} - \frac{1}{960}a_{2,3} = \frac{31}{45},$$

$$\frac{53}{168}a_{1,0} + \frac{79}{320}a_{1,1} + \frac{43}{216}a_{1,2} + \frac{1397}{8400}a_{1,3} - \frac{1}{36}a_{2,0} - \frac{1}{168}a_{2,1} - \frac{1}{480}a_{2,2} - \frac{1}{1080}a_{2,3} = \frac{607}{1260},$$

$$\frac{15}{64}a_{1,0} + \frac{71}{360}a_{1,1} + \frac{199}{1200}a_{1,2} + \frac{439}{3080}a_{1,3} - \frac{1}{42}a_{2,0} - \frac{1}{192}a_{2,1} - \frac{1}{540}a_{2,2} - \frac{1}{1200}a_{2,3} = \frac{125}{336}.$$

Similarly, equations resulting from $\mathbf{F}_2(x)\mathbf{A} = g_2(x)$ are

$$-\frac{1}{60}a_{1,0} - \frac{1}{420}a_{1,1} - \frac{1}{1680}a_{1,2} - \frac{1}{5040}a_{1,3} + \frac{39}{40}a_{2,0} + \frac{119}{240}a_{2,1} + \frac{93}{280}a_{2,2} + \frac{559}{2240}a_{2,3} = \frac{2697}{2240},$$

$$-\frac{1}{70}a_{1,0} - \frac{1}{480}a_{1,1} - \frac{1}{1890}a_{1,2} - \frac{1}{5600}a_{1,3} + \frac{23}{48}a_{2,0} + \frac{277}{840}a_{2,1} + \frac{239}{960}a_{2,2} + \frac{503}{2520}a_{2,3} = \frac{8989}{15120},$$

$$-\frac{1}{80}a_{1,0} - \frac{1}{540}a_{1,1} - \frac{1}{2100}a_{1,2} - \frac{1}{6160}a_{1,3} + \frac{53}{168}a_{2,0} + \frac{79}{320}a_{2,1} + \frac{43}{216}a_{2,2} + \frac{1397}{8400}a_{2,3} = \frac{12871}{33600},$$

$$-\frac{1}{90}a_{1,0} - \frac{1}{600}a_{1,1} - \frac{1}{2310}a_{1,2} - \frac{1}{6720}a_{1,3} + \frac{15}{64}a_{2,0} + \frac{71}{360}a_{2,1} + \frac{199}{1200}a_{2,2} + \frac{439}{3080}a_{2,3} = \frac{61543}{221760}.$$

These eight equations together make up the system $\mathbf{WA} = \mathbf{G}$, where

$$\mathbf{W} = \begin{bmatrix} \frac{39}{40} & \frac{119}{240} & \frac{93}{280} & \frac{559}{2240} & -\frac{1}{24} & -\frac{1}{120} & -\frac{1}{360} & -\frac{1}{840} \\ -\frac{1}{60} & -\frac{1}{420} & -\frac{1}{1680} & -\frac{1}{5040} & \frac{39}{40} & \frac{119}{240} & \frac{93}{280} & \frac{559}{2240} \\ \frac{23}{48} & \frac{277}{840} & \frac{239}{960} & \frac{503}{2520} & -\frac{1}{30} & -\frac{1}{144} & -\frac{1}{420} & -\frac{1}{960} \\ -\frac{1}{70} & -\frac{1}{480} & -\frac{1}{1890} & -\frac{1}{5600} & \frac{23}{48} & \frac{277}{840} & \frac{239}{960} & \frac{503}{2520} \\ \frac{53}{168} & \frac{79}{320} & \frac{43}{216} & \frac{1397}{8400} & -\frac{1}{36} & -\frac{1}{168} & -\frac{1}{480} & -\frac{1}{1080} \\ -\frac{1}{80} & -\frac{1}{540} & -\frac{1}{2100} & -\frac{1}{6160} & \frac{53}{168} & \frac{79}{320} & \frac{43}{216} & \frac{1397}{8400} \\ \frac{15}{64} & \frac{71}{360} & \frac{199}{1200} & \frac{439}{3080} & -\frac{1}{42} & -\frac{1}{192} & -\frac{1}{540} & -\frac{1}{1200} \\ -\frac{1}{90} & -\frac{1}{600} & -\frac{1}{2310} & -\frac{1}{6720} & \frac{15}{64} & \frac{71}{360} & \frac{199}{1200} & \frac{439}{3080} \end{bmatrix}, \; \mathbf{G} = \begin{bmatrix} \frac{151}{120} \\ \frac{2697}{2240} \\ \frac{31}{45} \\ \frac{8989}{15120} \\ \frac{607}{1260} \\ \frac{12871}{33600} \\ \frac{125}{336} \\ \frac{61543}{221760} \end{bmatrix}$$

Since the linear algebraic system is obtained, step 2 is complete.

In this example problem step 3 is not required since there are no initial or boundary conditions in the problem. Therefore, the matrix of unknown weights is calculated by

$$\mathbf{A} = \mathbf{W}^{-1}\mathbf{G} = \begin{bmatrix} 1.0 & 0 & 1.0 & 0 & 1.0 & 1.0 & 0 & -1.0 \end{bmatrix}^T,$$

which means that the approximate solutions are

$$y_{1,3}(x) = 1 + x^2,\ y_{2,3}(x) = 1 + x - x^3,$$

which are the exact solutions. This happened because the supposition that the approximate solutions are polynomials of degree three is true and as a result, the linear algebraic system resulting from this supposition is exact. Consequently, the approximate solutions yielded by this system are exact. By the same reasoning, any $N > 3$ gives the exact solutions in this problem.

Example 2: Our second example is the following second order system of two variables which is taken from [27]:

$$\begin{cases} y_1''(x) - 3xy_2'(x) - 2y_1(x) & = 3(x-1)\sin(x) + 2\cos(x)(1 - \cos(1) - \sin(1) + \frac{x}{4}) + \frac{x^2}{2}(\cos^2(1) - 1) \\ & -\frac{1}{2}\sin(x)\cos^2(x) + \displaystyle\int_0^1 \Big(2\cos(x)ty_1'(t) - x^2\sin(t)y_2''(t)\Big)\mathrm{d}t \\ & + \displaystyle\int_0^x \Big(x\cos(t)y_1(t) - x\sin(t)y_1'(t) + \cos(x)\sin(t)y_2'(t)\Big)\mathrm{d}t, \\ y_2''(x) - 2xy_1'(x) + xy_2(x) & = -\frac{1}{2}\cos(x)(1 + \sin(2) + 2x) - \frac{1}{2}\sin(x)(x+2) + \frac{1}{2}\cos^3(x) + x \\ & + \frac{1}{4}x^2(1 - \cos(2)) + \displaystyle\int_0^1 \Big(x^2\cos(t)y_1''(t) + 2\cos(x)\sin(t)y_2'(t)\Big)\mathrm{d}t \\ & + \displaystyle\int_0^x \Big(\sin(x)\cos(t)y_1'(t) - \cos(x)ty_2'(t) + \sin(x)ty_2''(t)\Big)\mathrm{d}t, \end{cases} \tag{4.2}$$

$$y_1(0) = 0,\ y_1'(0) = 1,\ y_2(0) = 1,\ y_2'(0) = 0.$$

This problem has the exact solution $y_{1,\text{exact}}(x) = \sin(x)$, $y_{2,\text{exact}}(x) = \cos(x)$. We have solved this problem using several values for the parameter N. For instance, one can verify that for $N = 3$ first two steps of the present method outputs the linear algebraic system $\mathbf{WA} = \mathbf{G}$, where

$$\mathbf{W} = \begin{bmatrix} -\frac{787}{342} & -\frac{841}{460} & \frac{1259}{4458} & \frac{1313}{994} & 0 & -\frac{502}{311} & -\frac{362}{201} & -\frac{703}{404} \\ 0 & -\frac{2245}{1764} & -\frac{910}{437} & -\frac{1200}{499} & \frac{1}{2} & -\frac{586}{1827} & \frac{2820}{2501} & \frac{1613}{887} \\ -\frac{1463}{1196} & -\frac{754}{727} & \frac{719}{14282} & \frac{2485}{2262} & 0 & -\frac{531}{491} & -\frac{3400}{2513} & -\frac{6613}{4647} \\ 0 & -\frac{3209}{3704} & -\frac{820}{523} & -\frac{1367}{724} & \frac{1}{3} & -\frac{172}{11367} & \frac{1031}{1582} & \frac{1371}{950} \\ -\frac{1739}{2061} & -\frac{1134}{1555} & -\frac{41}{34991} & \frac{671}{771} & 0 & -\frac{595}{732} & -\frac{612}{565} & -\frac{943}{785} \\ 0 & -\frac{1661}{2528} & -\frac{1667}{1326} & -\frac{3385}{2176} & \frac{1}{4} & \frac{187}{4006} & \frac{1191}{2530} & \frac{5613}{4921} \\ -\frac{1583}{2448} & -\frac{447}{794} & -\frac{141}{8008} & \frac{507}{712} & 0 & -\frac{386}{593} & -\frac{791}{876} & -\frac{487}{469} \\ 0 & -\frac{281}{531} & -\frac{875}{834} & -\frac{999}{755} & \frac{1}{5} & \frac{165}{2624} & \frac{739}{1989} & \frac{3901}{4169} \end{bmatrix},\ \mathbf{G} = \begin{bmatrix} -\frac{1399}{1180} \\ -\frac{600}{701} \\ -\frac{789}{1372} \\ -\frac{907}{1917} \\ -\frac{1573}{4362} \\ -\frac{584}{1801} \\ -\frac{1229}{4812} \\ -\frac{625}{2552} \end{bmatrix}.$$

Let us now consider the four given mixed conditions, which are indeed initial conditions. The equation corresponding to $y_1(0) = 0$ is just $a_{1,0} = 0$. Likewise, equations corresponding to $y_1'(0) = 1, y_2(0) = 1$ and $y_2'(0) = 0$ are $a_{1,1} = 1, a_{2,0} = 1$ and $a_{2,1} = 0$, respectively. The matrix representations of these equations are

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\mathbf{A} = 0, \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\mathbf{A} = 1,$$
$$\begin{bmatrix}0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}\mathbf{A} = 1, \begin{bmatrix}0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix}\mathbf{A} = 0.$$

In order to obtain the final system $\tilde{\mathbf{W}}\mathbf{A} = \tilde{\mathbf{G}}$, last four rows of the system $\mathbf{WA} = \mathbf{G}$ must be replaced by the above matrix representations of the initial conditions. Thus, the final system becomes

$$\begin{bmatrix} -\frac{787}{342} & -\frac{841}{460} & \frac{1259}{4458} & \frac{1313}{994} & 0 & -\frac{502}{311} & -\frac{362}{201} & -\frac{703}{404} \\ 0 & -\frac{2245}{1764} & -\frac{910}{437} & -\frac{1200}{499} & \frac{1}{2} & -\frac{586}{1827} & \frac{2820}{2501} & \frac{1613}{887} \\ -\frac{1463}{1196} & -\frac{754}{727} & \frac{719}{14282} & \frac{2485}{2262} & 0 & -\frac{531}{491} & -\frac{3400}{2513} & -\frac{6613}{4647} \\ 0 & -\frac{3209}{3704} & -\frac{820}{523} & -\frac{1367}{724} & \frac{1}{3} & -\frac{172}{11367} & \frac{1031}{1582} & \frac{1371}{950} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{1,0} \\ a_{1,1} \\ a_{1,2} \\ a_{1,3} \\ a_{2,0} \\ a_{2,1} \\ a_{2,2} \\ a_{2,3} \end{bmatrix} = \begin{bmatrix} -\frac{1399}{1180} \\ -\frac{600}{701} \\ -\frac{789}{1372} \\ -\frac{907}{1917} \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

The solution of this system is

$$a_{1,0} = 0, a_{1,1} = 1, a_{1,2} = -0.0195274388, a_{1,3} = -0.1447398671$$

and

$$a_{2,0} = 1, a_{2,1} = 0, a_{2,2} = -0.5369955259, a_{2,3} = 0.0734172099.$$

As a result, the approximate solutions obtained using $N = 3$ are given by

$$y_{1,3}(x) = x - 0.0195274388x^2 - 0.1447398671x^3, \; y_{2,3}(x) = 1 - 0.5369955259x^2 + 0.0734172099x^3.$$

Since the actual solution is known for this problem, the best way to measure the quality of the approximate solutions is by means of considering their actual absolute error. Graphs of the absolute error functions $|e_{1,N}(x)|$ and $|e_{2,N}(x)|$ corresponding to $N = 3, 4, 6, 8$ are depicted in Figure 1. It is clearly understood from the graph that increasing the value of the parameter N greatly improves the accuracy.

In order to have an idea on the position of the present scheme with respect to other methods in the literature, let us compare our results with those obtained by Bessel collocation method in [27]. For this purpose, we consider the maximum absolute error defined according to the norm $\|f\| = \max_{a \le x \le b} |f(x)|$. This comparison is made in Table 1, where the maximum errors corresponding to $y_1(x)$ and $y_2(x)$ are listed separately. The table shows that the present scheme outperforms Bessel collocation method for listed N values except the values corresponding to $y_2(x)$ for $N = 9$.

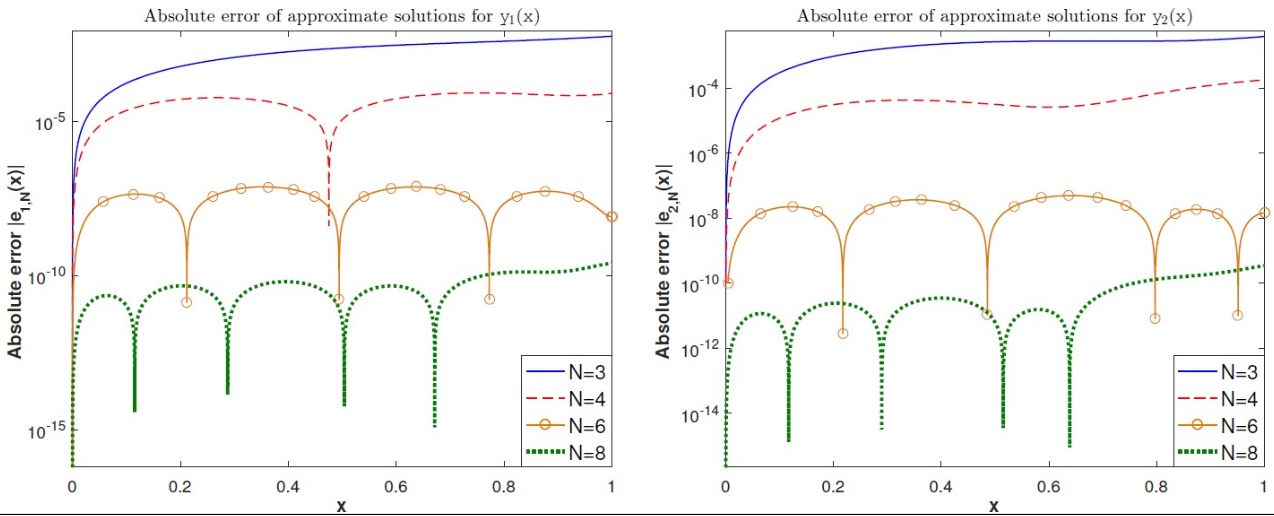

Figure 1. Graphics of the actual absolute errors of the approximate solutions of Problem (4.2) obtained using $N = 3, 4, 6, 8$ corresponding to $y_1(x)$ (left) and $y_2(x)$ (right).

Table 1. Comparison of the present method with Bessel collocation method with respect to the maximum absolute error of the approximate solutions obtained using $N = 3, 6, 9$ in Problem (4.2).

	Bessel collocation method [27]			Present method		
	$N=3$	$N=6$	$N=9$	$N=3$	$N=6$	$N=9$
$\|y_{1,N}\|_\infty$	5.828e − 2	5.309e − 5	8.356e − 9	5.738e − 3	7.649e − 8	3.092e − 9
$\|y_{2,N}\|_\infty$	7.096e − 2	2.411e − 5	3.054e − 9	3.880e − 3	5.012e − 8	3.599e − 9

Example 3: Next, let us examine the problem given by

$$\begin{cases} y_1^{(3)} + y_2'' + y_1' + e^{-x}y_2 = -1 - xe^x - 2e^{-x} + \int_{-1}^{1} \big(\sinh(t)y_1 + \cosh(t)y_2\big)\mathrm{d}t + \int_{-1}^{x} e^{x-t}y_2\,\mathrm{d}t, \\ y_2^{(3)} - e^x y_1^{(3)} - y_2' + xy_2 + y_1 = -6 - x - xe^{-x} + xe^{x+2} + \int_{-1}^{1} \big(3e^t y_1 - xe^{x+1}y_2\big)\mathrm{d}t \\ \qquad + \int_{-1}^{x} \big(-ty_1 + e^{-t}y_2\big)\mathrm{d}t, \end{cases} \tag{4.3}$$

$$y_1(-1) = e,\ y_1(0) = 1,\ y_1(1) = e^{-1},\ y_2(-1) = e,\ y_2(0) = 1,\ y_2(1) = e.$$

studied in [2]. Notice that this time the interval of interest is $-1 \le x \le 1$. Additionally, two of the mixed conditions are given at $x = 0$, which is not a boundary point. This does not pose any problem, however, since the conditions at $x = 0$ can be treated in the same manner as those given at $x = -1$ and $x = 1$.

The exact solution of this problem is known to be $y_{1,\text{exact}}(x) = e^{-x}$, $y_{2,\text{exact}}(x) = e^x$. Again, we have obtained the approximate solutions of this problem corresponding to several choices of the method parameter N. The absolute errors of some of these solutions are illustrated in Figure 2, which makes it clear that greater N values give rise to solutions with significantly smaller absolute error. In addition, we compare the absolute errors of three of our approximate solutions with those obtained by Chebyshev collocation method [2] in Tables 2 and 3. The tables suggest that the proposed scheme exhibits a close performance to the Chebyshev collocation method for this example problem.

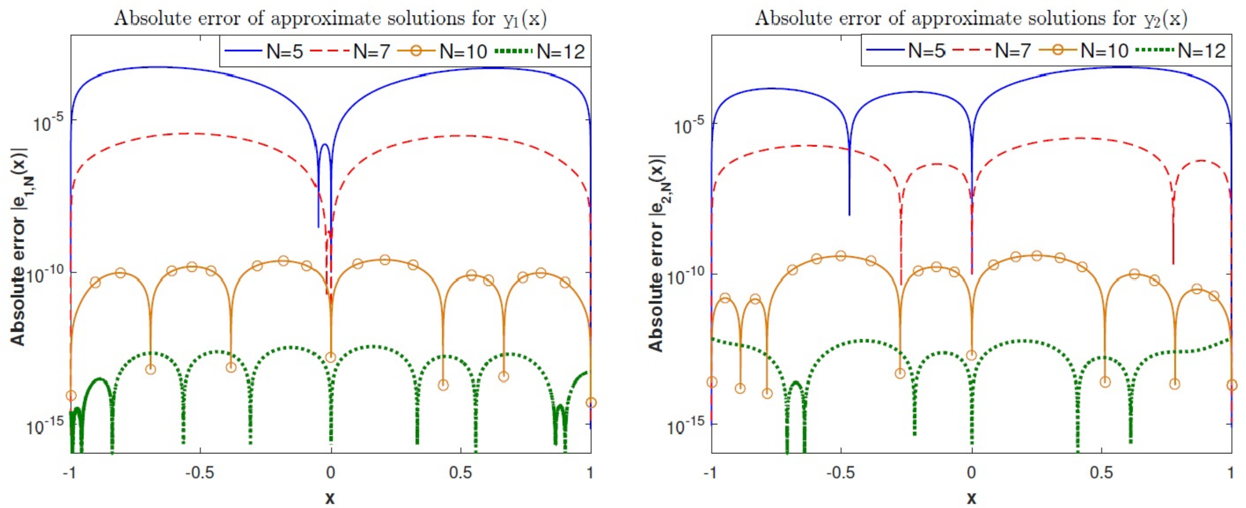

Figure 2. Graphics of the actual absolute errors of the approximate solutions of Problem (4.3) obtained using $N = 5, 7, 10, 12$ corresponding to $y_1(x)$ (left) and $y_2(x)$ (right).

Table 2. Comparison of the present method with Chebyshev collocation method with respect to the absolute error of the approximate solutions for $y_1(x)$ corresponding to $N = 5, 10, 13$ at several x values in Problem (4.3).

	Chebyshev collocation method [2]			Present method		
x	$\|e_{1,5}(x)\|$	$\|e_{1,10}(x)\|$	$\|e_{1,13}(x)\|$	$\|e_{1,5}(x)\|$	$\|e_{1,10}(x)\|$	$\|e_{1,13}(x)\|$
-0.75	6.88e − 4	3.72e − 10	2.66e − 14	5.24e − 4	6.62e − 11	3.10e − 15
-0.5	3.66e − 4	3.17e − 10	1.09e − 14	4.51e − 4	1.44e − 10	1.13e − 14
-0.25	8.99e − 5	1.93e − 11	7.11e − 15	1.33e − 4	2.03e − 10	2.42e − 14
0.25	3.27e − 5	1.21e − 10	2.89e − 15	1.82e − 4	2.41e − 10	1.77e − 14
0.5	8.35e − 5	1.18e − 10	3.89e − 15	4.59e − 4	6.61e − 11	8.65e − 15
0.75	9.63e − 5	7.95e − 11	3.89e − 15	4.51e − 4	7.57e − 11	2.93e − 14

Table 3. Comparison of the present method with Chebyshev collocation method with respect to the absolute error of the approximate solutions for $y_2(x)$ corresponding to $N = 5, 10, 13$ at several x values in Problem (4.3).

	Chebyshev collocation method [2]			Present method		
x	$\|e_{2,5}(x)\|$	$\|e_{2,10}(x)\|$	$\|e_{2,13}(x)\|$	$\|e_{2,5}(x)\|$	$\|e_{2,10}(x)\|$	$\|e_{2,13}(x)\|$
-0.75	7.25e − 4	2.61e − 10	2.80e − 14	1.54e − 4	3.63e − 11	1.56e − 13
-0.5	3.87e − 4	2.07e − 10	1.21e − 14	2.42e − 5	4.01e − 10	3.10e − 14
-0.25	8.61e − 5	2.16e − 11	8.22e − 15	1.16e − 4	4.96e − 11	1.57e − 14
0.25	3.24e − 5	1.29e − 10	4.00e − 15	3.82e − 4	4.25e − 10	1.70e − 14
0.5	1.15e − 4	1.01e − 10	5.11e − 15	7.26e − 4	2.06e − 11	1.99e − 14
0.75	1.47e − 4	9.18e − 11	5.77e − 15	6.09e − 4	2.57e − 11	1.05e − 13

Although the exact solution of Problem 4.3 is known, it may be interesting to see how the process of error estimation works for the approximate solutions obtained with the choice $N = 5$. To this end, we have formed the error problem corresponding to the approximate solutions $y_{1,5}(x)$ and $y_{2,5}(x)$ and applied the present scheme using the values $6, 7$ and 9 for the parameter M, as explained in Section 3. Thus we have obtained the error estimates $e_{1,5,M}(x)$ and $e_{2,5,M}(x)$ for $M = 6, 7, 9$. The values of these error estimates at selected values of x are given together with the actual errors $e_{1,5}(x)$ and $e_{2,5}(x)$ in Table 4. We see that error estimates corresponding to bigger M values are more accurate; they are even equal to the actual error to three decimal places for $M = 9$.

Lastly, as a demonstration of the comment closing Section 3, we consider the absolute residuals of the approximate solutions of Problem (4.3). While the absolute residuals are roughly equal to 10^{-2} throughout the interval $-1 \leq x \leq 1$ for $N = 5$, they are reduced to approximately 10^{-10} for $N = 12$. This situation can be observed in Figure 3.

Table 4. Comparison of the actual errors $e_{1,5}(x)$ and $e_{2,5}(x)$ corresponding to $N = 5$ with their three estimates obtained by setting $M = 6, 7, 9$ at several x values in Problem (4.3).

	Actual errors for $y_1(x)$				Actual errors for $y_2(x)$			
x	$e_{1,5}(x)$	$e_{1,5,6}(x)$	$e_{1,5,7}(x)$	$e_{1,5,9}(x)$	$e_{2,5}(x)$	$e_{2,5,6}(x)$	$e_{2,5,7}(x)$	$e_{2,5,9}(x)$
-0.75	5.240e $-$ 4	5.338e $-$ 4	5.264e $-$ 4	5.240e $-$ 4	1.548e $-$ 4	1.607e $-$ 4	1.564e $-$ 4	1.548e $-$ 4
-0.5	4.515e $-$ 4	4.941e $-$ 4	4.552e $-$ 4	4.515e $-$ 4	2.423e $-$ 5	$-$2.921e $-$ 6	2.582e $-$ 5	2.423e $-$ 5
-0.25	1.339e $-$ 4	1.772e $-$ 4	1.354e $-$ 4	1.338e $-$ 4	$-$1.162e $-$ 4	$-$1.556e $-$ 4	$-$1.163e $-$ 4	$-$1.162e $-$ 4
0.25	1.820e $-$ 4	1.361e $-$ 4	1.836e $-$ 4	1.820e $-$ 4	3.820e $-$ 4	4.396e $-$ 4	3.845e $-$ 4	3.820e $-$ 4
0.5	4.594e $-$ 4	4.073e $-$ 4	4.626e $-$ 4	4.594e $-$ 4	7.264e $-$ 4	8.035e $-$ 4	7.297e $-$ 4	7.264e $-$ 4
0.75	4.510e $-$ 4	4.272e $-$ 4	4.526e $-$ 4	4.510e $-$ 4	6.098e $-$ 4	6.535e $-$ 4	6.101e $-$ 4	6.098e $-$ 4

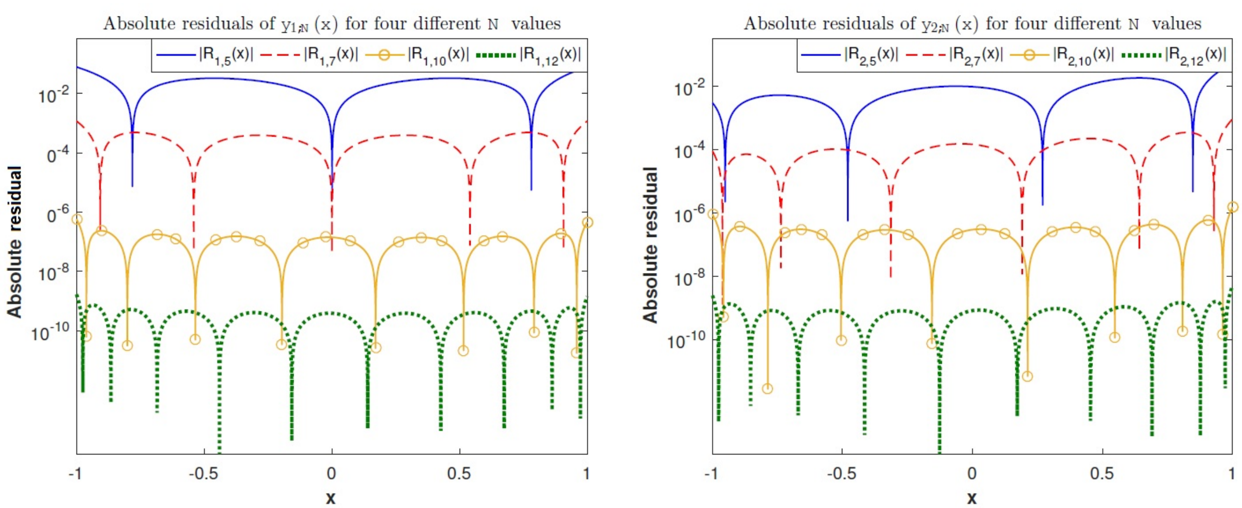

Figure 3. Absolute residuals of the approximate solutions of Problem (4.3) corresponding to $N = 5, 7, 10, 12$ for $y_1(x)$ (left) and $y_2(x)$ (right).

Example 4: Next, we consider the following system of second order linear Volterra integral equations studied in [3] and [20]:

$$\begin{aligned} y_1''(x) + 2xy_1'(x) - y_1(x) &= \int_0^x (y_1(t) - y_2(t))\, \mathrm{d}t + 2 + x - e^x + 2xe^x - \cos(x), \\ y_2''(x) + y_1'(x) - 2xy_2(x) &= \int_0^x (y_1(t) + y_2(t))\, \mathrm{d}t + 2\cos(x) - 3x - (1+2x)\sin(x) - e^x, \\ & y_1(0) = 1,\ y_1'(0) = 1,\ y_2(0) = 1,\ y_2'(0) = 1. \end{aligned} \tag{4.4}$$

The exact solutions are known to be $y_{1,\text{exact}}(x) = e^x$, $y_{2,\text{exact}}(x) = 1 + \sin(x)$ We have solved this problem using various N values. The absolute errors corresponding to $N = 3, 5, 8$ are depicted in Figure 4, which shows that accuracy improves greatly as we increase the value of N, in a similar fashion to the previous examples. We also compare the absolute errors of our results obtained by $N = 5$ to those of spectral method [3] and Bernstein operational matrix method [20] in Table 5. It is understood from the table that the present method is significantly better than the spectral method and slightly better than Bernstein operational matrix method for $N = 5$ in this problem. In addition, in order to give an idea on the speed of the present method, we included in Table 6 the CPU times (in seconds) needed to compute the approximate solutions in a computer with an Intel Core i5 2.80 GHz processor.

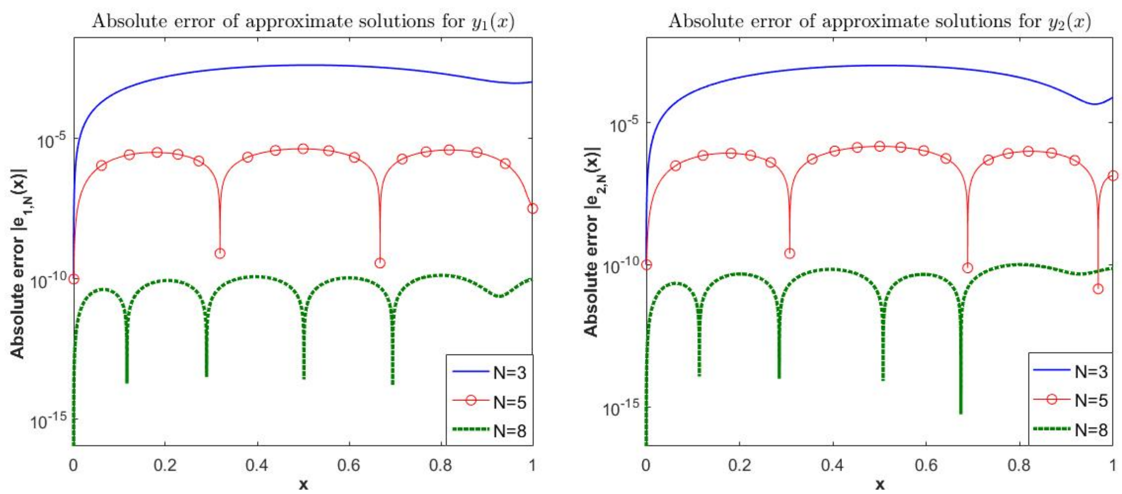

Figure 4. Graphics of the actual absolute errors of the approximate solutions of Problem (4.4) obtained using $N = 3, 5, 8$ corresponding to $y_1(x)$ (left) and $y_2(x)$ (right).

Table 5. Comparison of the present method with Bernstein operational matrix method (BOM) and the spectral method (SM) with respect to the absolute errors for $y_1(x)$ and $y_2(x)$ corresponding to $N = 5$ in Problem (4.4).

	Absolute errors for $y_1(x)$			Absolute errors for $y_2(x)$		
x	BOM [20]	SM [3]	Present	BOM	SM	Present
0	$8.881e-16$	0	0	$7.771e-16$	0	0
0.2	$1.364e-6$	$9.1e-8$	$3.137e-6$	$2.873e-7$	$2.0e-9$	$8.056e-7$
0.4	$2.420e-6$	$6.031e-6$	$2.834e-6$	$4.545e-7$	$3.25e-7$	$9.747e-7$
0.6	$3.315e-6$	$7.08e-5$	$2.545e-6$	$5.638e-7$	$5.527e-6$	$9.880e-7$
0.8	$1.213e-5$	$4.102e-4$	$3.899e-6$	$2.692e-6$	$4.124e-5$	$9.476e-7$
1	$1.295e-4$	$1.615e-3$	$3.251e-8$	$3.359e-5$	$1.956e-4$	$1.364e-7$

Table 6. CPU times in seconds corresponding to several N values in Problem (4.4).

	$N=3$	$N=4$	$N=5$	$N=6$	$N=7$	$N=8$
CPU time(s)	0.5938	0.6250	0.7188	0.7813	0.8438	1.8906

Example 5: Lastly, we consider the following system of linear Fredholm integral equations taken from

[6] and [11]:

$$
\begin{aligned}
22y_1(x) &= -\int_0^1 e^{x-t}y_1(t)\ \mathrm{d}t - \int_0^1 e^{(x+2)t}y_2(t)\ \mathrm{d}t + 23e^x + \frac{e^{1+x}-1}{1+x} \\
22y_2(x) &= -\int_0^1 e^{xt}y_1(t)\ \mathrm{d}t - \int_0^1 e^{x+t}y_2(t)\ \mathrm{d}t + 22e^{-x} + e^x + \frac{e^{1+x}-1}{1+x}.
\end{aligned} \tag{4.5}
$$

The exact solutions are known to be $y_{1,\text{exact}}(x) = e^x$, $y_{2,\text{exact}}(x) = e^{-x}$. We have solved this problem using $N = 3, 4, 5, 6$ by the present method. In Table 7, we compare the maximum absolute errors of our solutions with those of the fixed point algorithm in [11] with $i = 33$ nodes and $m = 6$ iterations, quasiinterpolatory spline collocation method [11] with order of precision $p = 4$ and hat basis functions method [6] with $n = 32$. It is seen that the present method outperforms the other methods (with the mentioned parameters) for $N = 6$ and is outperformed only by quasiinterpolatory spline collocation method for $N = 5$.

Table 7. Comparison of the present method (PM) with the fixed point algorithm (FPA), quasiinterpolatory spline collocation method (QISC) and hat basis functions method (HBF) with respect to the maximum absolute errors of the approximate solutions obtained using $N = 3, 4, 5, 6$ in Problem (4.5).

	FPA [11]	QISC [11]	HBF [6]	PM $N=3$	PM $N=4$	PM $N=5$	PM $N=6$
$\lVert y_{1,N}\rVert_\infty$	1.83e − 5	1.04e − 7	2.68e − 3	1.04e − 3	5.76e − 5	2.59e − 6	1.10e − 7
$\lVert y_{2,N}\rVert_\infty$	1.08e − 5	4.26e − 7	1.23e − 4	3.45e − 4	1.93e − 5	8.85e − 7	3.34e − 8

5. Conclusion

We have presented a weighted residual scheme to obtain approximate polynomial solutions of systems of linear Volterra-Fredholm integro-differential equations given with mixed initial and boundary conditions. This method basically consists of transforming the original problem into a linear algebraic system by taking inner product with the monomials up to a prescribed degree N in a Galerkin-like fashion. We have also described how residuals of the approximate solutions can be used to form an error problem from which error estimates for the obtained solutions are obtained. We have then applied the proposed scheme to five example problems. These simulations have revealed that increasing the parameter N makes the resulting approximate solutions significantly more accurate. In addition, it has been understood that, when applied to a problem whose exact solutions are polynomials of degree at most N, the method yields these exact solutions as long as the parameter is chosen to be at least N. We have also compared the accuracy of our solutions with those of several methods from the literature, where the results are either comparable or in favor of our method. The proposed scheme has the additional advantage that it can be applied to a large set of problems including Fredholm or Volterra integral equation systems, as exhibited by three of the studied example problems. Fairly low CPU times in obtaining the approximate solutions are another virtue of the present scheme. It can also be extended to nonlinear problems of similar type, which is a task that can be dealt with in a future work. On the whole, we can conclude that the numerical scheme presented in this paper can be relied on to solve linear Volterra-Fredholm integro-differential equation systems with high levels of accuracy and thus it is a reasonable contribution to the field.

References

[1] Abou-Dina MS, Helal MA. Reduction for the nonlinear problem of fluid waves to a system of integro-differential equations with an oceanographical application. Journal of Computational and Applied Mathematics 1998; 95 (1-2): 65-81. doi: 10.1016/S0377-0427(98)00072-7

[2] Akyüz-Daşcıoğlu A, Sezer M. Chebyshev polynomial solutions of systems of higher-order linear Fredholm-Volterra integro-differential equations. Journal of the Franklin Institute 2005; 342 (6): 688-701. doi: 10.1016/j.jfranklin.2005.04.001

[3] Al-Faour OMA, Saaed RK. Solution of a system of linear Volterra integral and integro-differential equations by spectral method. Al-Nahrain University Journal for Science 2006; 6 (2): 30-46.

[4] Andrade B, Viana A. Integrodifferential equations with applications to a plate equation with memory. Mathematische Nachrichten 2016; 289 (17-18): 2159-2172. doi: 10.1002/mana.201500205

[5] Athavale P, Tadmor E. Novel integro-differential equations in image processing and its applications. In: Proceedings of SPIE - The International Society for Optical Engineering 2010: 75330. doi: 10.1117/12.850779

[6] Babolian E, Mordad M. A numerical method for solving systems of linear and nonlinear integral equations of the second kind by hat basis functions. Computers & Mathematics with Applications 2011; 62 (1): 187-198.

[7] Bellomo N, Firmani B, Guerri L. Bifurcation analysis for a nonlinear system of integro-differential equations modelling tumor-immune cells competition. Applied Mathematics Letters 1999; 12 (2): 39-44. doi: 10.1016/S0893-9659(98)00146-3

[8] Biazar J. Solution of systems of integral-differential equations by Adomian decomposition method. Applied Mathematics and Computation 2005; 168 (2): 1232-1238. doi: 10.1016/j.amc.2004.10.015

[9] Bloom F. Asymptotic bounds for solutions to a system of damped integro-differential equations of electromagnetic theory. Journal of Mathematical Analysis and Applications 1980; 73 (2): 524-542. doi: 10.1016/0022-247X(80)90297-8

[10] Bohner M, Tunç O, Tunç C. Qualitative analysis of Caputo fractional integro-differential equations with constant delays. Computational and Applied Mathematics 2021; 40: 214. doi: 10.1007/s40314-021-01595-3

[11] Caliò F, Garralda-Guillem AI, Marchetti E, Ruiz Galán M. Numerical approaches for systems of Volterra-Fredholm integral equations. Applied Mathematics and Computation 2013; 225: 811-821. doi: 10.1016/j.amc.2013.10.006

[12] Gushing JM. Integrodifferential Equations and Delay Models in Population Dynamics. Berlin, Germany: Springer Verlag, 1977.

[13] Gülsu M, Sezer M. Taylor collocation method for solution of systems of high-order linear Fredholm Volterra integro-differential equations. International Journal of Computer Mathematics 2006; 83 (4): 429-448. doi: 10.1080/00207160600988342

[14] Hesameddini E, Shahbazi M. Solving system of Volterra-Fredholm integral equations with Bernstein polynomials and hybrid Bernstein Block-Pulse functions. Journal of Computational and Applied Mathematics 2017; 315: 182-194. doi: 10.1016/j.cam.2016.11.004

[15] Karaçayır M. Diferansiyel, fonksiyonel diferansiyel ve integro-diferansiyel denklemler ve sistemleri için Galerkin-tipi bir sayısal yöntem, PhD, Akdeniz University, Antalya, Turkey, 2018 (in Turkish).

[16] Kilpatrick ZP. Wilson-Cowan Model. Encyclopedia of Computational Neuroscience 2014: 3159-3163. doi: 10.1007/978-1-4614-7320-6_80-1

[17] Kyselka A. Properties of systems of integro-differential equations in the statistics of polymer chains. Polymer Science U.S.S.R. 1977; 19 (11): 2852-2858. doi: 10.1016/0032-3950(77)90303-3

[18] Lakshmikantham V. Theory of integro-differential equations. Amsterdam, The Netherlands: CRC Press, 1995.

[19] MacCamy RC. An integro-differential equation with application in heat flow. Quarterly of Applied Mathematics 1977; 35 (1): 1-19. doi: 10.1090/qam/452184

[20] Maleknejad K, Basirat B, Hashemizadeh E. A Bernstein operational matrix approach for solving a system of high order linear Volterra-Fredholm integro-differential equations. Mathematical and Computer Modelling 2012; 55 (3-4): 1363-1372. doi: 10.1016/j.mcm.2011.10.015

[21] Saberi-Nadjafi J, Tamamgar M. The variational iteration method: a highly promising method for solving the system of integro-differential equations. Computers & Mathematics with Applications 2008; 56 (2): 346-351. doi: 10.1016/j.camwa.2007.12.014

[22] Suresh Kumar KS. Electric Circuits & Networks. Delhi, India: Pearson, 2009.

[23] Tunç C, Tunç O. On the stability, integrability and boundedness analyses of systems of integro-differential equations with time-delay retardation. Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales, Serie A Matemáticas 2021; 115 (3): 115. doi: 10.1007/s13398-021-01058-8

[24] Tunç C, Wang Y, Tunç O, Jen-Chih Y. New and Improved Criteria on Fundamental Properties of Solutions of Integro-Delay Differential Equations with Constant Delay. Mathematics 2021; 9 (24): 3317. doi: 10.3390/math9243317

[25] Türkyılmazoğlu M. An effective approach for numerical solutions of high-order Fredholm integro-differential equations. Applied Mathematics and Computation 2014; 227: 384-398. doi: 10.1016/j.amc.2013.10.079

[26] Yusufoğlu (Agadjanov) E. An efficient algorithm for solving integro-differential equations system. Applied Mathematics and Computation 2007; 192 (1): 51-55. doi: 10.1016/j.amc.2007.02.134

[27] Yüzbaşı Ş. Numerical solutions of system of linear Fredholm-Volterra integro-differential equations by the Bessel collocation method and error estimation. Applied Mathematics and Computation 2015; 250: 320-338. doi: 10.1016/j.amc.2014.10.110

[28] Yüzbaşı Ş, Karaçayır M. A Galerkin-like approach to solve high-order integro-differential equations with weakly singular kernel. Kuwait Journal of Science 2016; 43 (2): 106-120.

[29] Yüzbaşı Ş, Karaçayır M. A Galerkin-like approach to solve continuous population models for single and interacting species. Kuwait Journal of Science 2017; 44 (2): 9-26.

[30] Yüzbaşı Ş, Karaçayır M. A Numerical Approach for Solving High-Order Linear Delay Volterra Integro-Differential Equations. International Journal of Computational Methods 2018; 15 (5): 1850042. doi: 10.1142/S0219876218500421