Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 10, No. 1, 2018 Article ID IJIM-001060, 9 pages
Research Article

A Fast and Accurate Expansion-Iterative Method For Solving Second Kind Volterra Integral Equations

S. Hatamzadeh-Varmazyar *†, Z. Masouri ‡

Received Date: 2017-04-27 Revised Date: 2017-09-07 Accepted Date: 2017-10-22

Abstract

This article proposes a fast and accurate expansion-iterative method for solving second kind linear Volterra integral equations. The method is based on a special representation of vector forms of triangular functions and their operational matrix of integration. By using this approach, solving the integral equation reduces to solve a recurrence relation. The approximate solution of integral equation is iteratively produced via the recurrence relation. This approach does not use any projection method such as collocation, Galerkin, etc., for setting up the recurrence relation. Some test problems are provided to illustrate the accuracy and computational efficiency of the method and to show that the resulting algorithm runs very fast.

Keywords : Numerical solution; Second kind Volterra integral equation; Triangular functions; Vector forms; Recurrence relation; Operational matrix; Iterative method.

1 Introduction

Due to the importance of functional equations for modeling of many problems in physical science and engineering as well as inaccessibility to an analytical solution for most of them, the numerical methods have been finding an important role for analysis of such models and obtaining an approximate solution for them.

Generally, the above mentioned models appear in form of integral equations, integro-differential equations, or differential equations. Each of them may be linear or nonlinear, homogeneous or inhomogeneous, one-dimensional or higher, and etc. Moreover, they may appear as a system of functional equations [5, 13, 11, 9, 14, 16, 8, 10, 15, 1, 2, 3, 4].

This article proposes an expansion-iterative method for numerical solution of Volterra integral equation of the second kind. This approach uses a special representation of the vector forms of triangular functions (TFs) [6]. Using the method, solving the integral equation reduces to solve a recurrence relation. The approximate solution is iteratively produced via the recurrence relation. A similar work has been done in [12] by using block-pulse functions (BPFs). However, the algorithm of the BPFs method is time-consuming with a moderate accuracy. The main goal in the current article is to overcome these drawbacks. The advantages of the presented method are as follows:

[•]The algorithm is simple and clear to use and can be implemented easily. This approach does not use any projection method

*Corresponding author. s.hatamzadeh@yahoo.com.
†Department of Electrical Engineering, Islamshahr Branch, Islamic Azad University, Tehran, Iran.
‡Department of Mathematics, Islamshahr Branch, Islamic Azad University, Tehran, Iran.

such as collocation, Galerkin, etc., for setting up the recurrence relation. The method is fast and the running times of its algorithm, even for high degrees of approximation, are very short. This is due to the fact that this method uses sampling of functions instead of integration. Computing the numerical solution does not need to directly solve any linear system of algebraic equations and to use any matrix inversion. The proposed method is considerably more accurate than the BPFs method.

We organize this article as follows. A brief review on TFs and their vector forms is provided in section 2. A special representation of TFs, introduced in [5], is surveyed in section 3. Section 4 presents the expansion-iterative method for numerically solving Volterra integral equations of the second kind. Section 5 includes some test problems to confirm the accuracy and computational efficiency of the proposed approach. The results obtained by the proposed method will also be compared with those of the BPFs method. Moreover, for further evaluation of the computational efficiency of the method, the running times of its algorithm as well as those of the BPFs method will be given there to confirm that the algorithm of the proposed method runs very fast. Finally, conclusions will be given in section 6.

2 Review of triangular functions [6]

2.1 Definition

Two m-sets of TFs are defined over the interval $[0, T)$ as [6, 5]

$$T1_i(t) = \begin{cases} 1 - \frac{t-ih}{h}, & ih \le t < (i+1)h, \\ 0, & \text{otherwise}, \end{cases}$$
$$T2_i(t) = \begin{cases} \frac{t-ih}{h}, & ih \le t < (i+1)h, \\ 0, & \text{otherwise}, \end{cases} \tag{2.1}$$

where $i = 0, 1, \dots, m-1$, with a positive integer value for m. Also, consider $h = T/m$, and $T1_i$ as the ith left-handed TF and $T2_i$ as the ith right-handed TF.

In this paper, it is assumed that $T = 1$, so TFs are defined over $[0, 1)$, and $h = 1/m$.

From the definition of TFs, it is clear that they are disjoint, orthogonal, and complete [6]. Also, we can write

$$\varphi_i(t) = T1_i(t) + T2_i(t), \quad i = 0, 1, \dots, m-1, \tag{2.2}$$

where $\varphi_i(t)$ is the ith BPF defined as

$$\varphi_i(t) = \begin{cases} 1, & ih \leqslant t < (i+1)h, \\ 0, & \text{otherwise}, \end{cases} \tag{2.3}$$

where $i = 0, 1, ..., m-1$.

2.2 Vector forms

Consider the first m terms of left-handed TFs and the first m terms of right-handed TFs and write them concisely as m-vectors:

$$\mathbf{T1}(t) = [T1_0(t), T1_1(t), ..., T1_{m-1}(t)]^T,$$
$$\mathbf{T2}(t) = [T2_0(t), T2_1(t), ..., T2_{m-1}(t)]^T, \tag{2.4}$$

where $\mathbf{T1}(t)$ and $\mathbf{T2}(t)$ are called left-handed triangular functions (LHTF) vector and right-handed triangular functions (RHTF) vector, respectively.

2.3 TFs expansion

The expansion of a function $f(t)$ over $[0, 1)$ with respect to TFs, may be compactly written as

$$f(t) \simeq \sum_{i=0}^{m-1} c_i T1_i(t) + \sum_{i=0}^{m-1} d_i T2_i(t) = \mathbf{c}^T \mathbf{T1}(t) + \mathbf{d}^T \mathbf{T2}(t), \tag{2.5}$$

where we may put $c_i = f(ih)$ and $d_i = f((i+1)h)$ for $i = 0, 1, \dots, m-1$. So, approximating a known function by TFs needs no integration to evaluate the coefficients.

2.4 Operational matrix of integration

Expressing $\int_0^s \mathbf{T1}(\tau)d\tau$ and $\int_0^s \mathbf{T2}(\tau)d\tau$ in terms of TFs follows [6]:

$$\int_0^s \mathbf{T1}(\tau)d\tau \simeq P1\mathbf{T1}(s) + P2\mathbf{T2}(s),$$
$$\int_0^s \mathbf{T2}(\tau)d\tau \simeq P1\mathbf{T1}(s) + P2\mathbf{T2}(s), \tag{2.6}$$

where $P1_{m\times m}$ and $P2_{m\times m}$ are called operational matrices of integration in TFs domain and represented as follows:

$$P1 = \frac{h}{2}\begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 1 & \dots & 1 \\ 0 & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$
$$P2 = \frac{h}{2}\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}. \tag{2.7}$$

So, the integral of any function $f(t)$ can be approximated as

$$\int_0^s f(\tau)d\tau \simeq \int_0^s \left[\mathbf{c}^T\mathbf{T1}(\tau) + \mathbf{d}^T\mathbf{T2}(\tau)\right] d\tau$$
$$\simeq (\mathbf{c}+\mathbf{d})^T P1\mathbf{T1}(s)$$
$$+ (\mathbf{c}+\mathbf{d})^T P2\mathbf{T2}(s). \tag{2.8}$$

3 A special representation of TFs vector forms and other properties [5]

In this section, we survey a special representation of TFs vector forms that has originally been introduced in [5]. Then, some characteristics of TFs are presented based on this definition.

3.1 Definition and expansion

Let $\mathbf{T}(t)$ be a $2m$-vector defined as [5]

$$\mathbf{T}(t) = \begin{pmatrix} \mathbf{T1}(t) \\ \mathbf{T2}(t) \end{pmatrix}, \quad 0 \leqslant t < 1, \tag{3.9}$$

where $\mathbf{T1}(t)$ and $\mathbf{T2}(t)$ have been defined in (2.4). Now, the expansion of $f(t)$ with respect to TFs can be written as

$$\begin{aligned} f(t) &\simeq F1^T\mathbf{T1}(t) + F2^T\mathbf{T2}(t) \\ &= F^T\mathbf{T}(t) \\ &= \mathbf{T}^T(t)F, \end{aligned} \tag{3.10}$$

where $F1$ and $F2$ are TFs coefficients with $F1_i = f(ih)$ and $F2_i = f((i+1)h)$, for $i = 0, 1, \dots, m-1$. Also, $2m$-vector F is defined as

$$F = \begin{pmatrix} F1 \\ F2 \end{pmatrix}. \tag{3.11}$$

Now, assume that $k(s,t)$ is a function of two variables. It can be expanded with respect to TFs as follows:

$$k(s,t) \simeq \mathbf{T}^T(s)\ K\ \mathbf{T}(t), \tag{3.12}$$

where $\mathbf{T}(s)$ and $\mathbf{T}(t)$ are $2m_1$- and $2m_2$-dimensional TFs respectively, and K is a $2m_1 \times 2m_2$ TFs coefficient matrix. For convenience, we put $m_1 = m_2 = m$. So, matrix K can be written as

$$K = \begin{pmatrix} (K11)_{m\times m} & (K12)_{m\times m} \\ (K21)_{m\times m} & (K22)_{m\times m} \end{pmatrix}, \tag{3.13}$$

where $K11$, $K12$, $K21$, and $K22$ can be computed by sampling of function $k(s,t)$ at points s_i and t_i such that $s_i = t_i = ih$, for $i = 0, 1, \dots, m$. Therefore,

$$\begin{aligned} (K11)_{i,j} &= k(s_i,t_j), & i &= 0,1,\dots,m-1, \\ & & j &= 0,1,\dots,m-1, \\ (K12)_{i,j} &= k(s_i,t_j), & i &= 0,1,\dots,m-1, \\ & & j &= 1,2,\dots,m, \\ (K21)_{i,j} &= k(s_i,t_j), & i &= 1,2,\dots,m, \\ & & j &= 0,1,\dots,m-1, \\ (K22)_{i,j} &= k(s_i,t_j), & i &= 1,2,\dots,m, \\ & & j &= 1,2,\dots,m. \end{aligned} \tag{3.14}$$

3.2 Product properties

Let X be a $2m$-vector which can be written as $X^T = (X1^T \quad X2^T)$ such that $X1$ and $X2$ are m-vectors. Now, it can be concluded that [5]

$$\mathbf{T}(t)\mathbf{T}^T(t)X \simeq \tilde{X}\mathbf{T}(t), \tag{3.15}$$

where $\tilde{X} = diag(X)$ is a $2m\times 2m$ diagonal matrix.

Now, let B be a $2m \times 2m$ matrix. We have [5]

$$\mathbf{T}^T(t)B\mathbf{T}(t) \simeq \hat{B}^T\mathbf{T}(t), \tag{3.16}$$

in which $\hat{B}$ is a $2m$-vector with elements equal to the diagonal entries of matrix B. Moreover, it is concluded that [5]

$$\int_0^1 \mathbf{T}(t)\mathbf{T}^T(t)\ t \simeq D, \tag{3.17}$$

where D is a $2m \times 2m$ matrix defined as

$$D = \begin{pmatrix} \frac{h}{3}I_{m\times m} & \frac{h}{6}I_{m\times m} \\ \frac{h}{6}I_{m\times m} & \frac{h}{3}I_{m\times m} \end{pmatrix}. \tag{3.18}$$

3.3 Operational matrix

Expressing $\int_0^s \mathbf{T}(\tau)\tau$ in terms of $\mathbf{T}(s)$, and from Eqs. (2.6), we can write [5]

$$\int_0^s \mathbf{T}(\tau)\tau \simeq P\mathbf{T}(s), \tag{3.19}$$

where $P_{2m\times 2m}$, operational matrix of $\mathbf{T}(s)$, is

$$P = \begin{pmatrix} P1 & P2 \\ P1 & P2 \end{pmatrix}, \tag{3.20}$$

in which $P1$ and $P2$ are given by (2.7).

Now, the integral of any function $f(t)$ can be approximated as

$$\int_0^s f(\tau)\tau \simeq \int_0^s F^T\mathbf{T}(\tau)\tau \simeq F^T P\mathbf{T}(s). \tag{3.21}$$

4 Formulation of the proposed method

Here, by using the mentioned TFs vector forms and properties, we propose a fast and accurate expansion-iterative method for numerically solving Volterra integral equation of the second kind.

Let us consider Volterra integral equation of the second kind of the form

$$x(s) + \int_0^s k(s,t)x(t)t = f(s), \qquad 0 \leqslant s < 1, \tag{4.22}$$

where the functions k and f are known but x is the unknown function to be determined. Moreover, $k \in L^2([0,1) \times [0,1))$ and $f \in L^2([0,1))$. Also, without loss of generality, it is supposed that the interval of integration in Eq. (4.22) is $[0,s)$ and $0 \leqslant s < 1$, since any finite interval can be transformed to this interval by linear maps [7].

From Eq. (4.22), the following iterative process can be proposed [7]:

$$x^{(n)}(s) + \int_0^s k(s,t)x^{(n-1)}(t)t = f(s), \tag{4.23}$$

with the initial value (initial guess) $x^{(0)}(s)$.

The recurrence relation (4.23) shows that one has to carry out analytically the integrals of the form $\int_0^s k(s,t)x^{(n-1)}(t)t$.

To overcome this, we use TFs and their operational matrix, and produce a recurrence relation based on algebraic operations, multiplication, and addition of matrices.

Approximating the functions k, f, and x with respect to TFs, using Eqs. (3.10) and (3.12), gives

$$\begin{aligned} &k(s,t) \simeq \mathbf{T}^T(s)K\mathbf{T}(t), \\ &f(s) \simeq F^T\mathbf{T}(s) = \mathbf{T}^T(s)F, \\ &x(s) \simeq X^T\mathbf{T}(s) = \mathbf{T}^T(s)X, \end{aligned} \tag{4.24}$$

where the $2m$–vectors F, X, and $2m \times 2m$ matrix K are TFs coefficients of f, x, and k, respectively. Note that X in Eq. (4.24) is the unknown vector and should be obtained.

Substituting (4.24) into (4.23) gives

$$X^{(n)^T}\mathbf{T}(s) + \int_0^s \mathbf{T}^T(s)K\mathbf{T}(t)\mathbf{T}^T(t)X^{(n-1)}t \simeq F^T\mathbf{T}(s). \tag{4.25}$$

Using Eq. (3.15) follows

$$X^{(n)^T}\mathbf{T}(s) + \int_0^s \mathbf{T}^T(s)K\tilde{X}^{(n-1)}\mathbf{T}(t)t \simeq F^T\mathbf{T}(s), \tag{4.26}$$

or

$$X^{(n)^T}\mathbf{T}(s) + \mathbf{T}^T(s)K\tilde{X}^{(n-1)}\int_0^s \mathbf{T}(t)t \simeq F^T\mathbf{T}(s). \tag{4.27}$$

Using operational matrix P in Eq. (3.19) gives

$$X^{(n)^T}\mathbf{T}(s) + \mathbf{T}^T(s)K\tilde{X}^{(n-1)}P\mathbf{T}(s) \simeq F^T\mathbf{T}(s), \qquad (4.28)$$

where $K\tilde{X}^{(n-1)}P$ is a $2m \times 2m$ matrix. So, from (3.16) we have

$$\mathbf{T}^T(s)K\tilde{X}^{(n-1)}P\mathbf{T}(s) \simeq \hat{U}^T\mathbf{T}(s), \qquad (4.29)$$

in which $\hat{U}$ is a $2m$-vector with components equal to the diagonal entries of matrix $K\tilde{X}^{(n-1)}P$. Combining (4.28) and (4.29) gives

$$X^{(n)^T}\mathbf{T}(s) + \hat{U}^T\mathbf{T}(s) \simeq F^T\mathbf{T}(s), \qquad (4.30)$$

or

$$X^{(n)^T} + \hat{U}^T \simeq F^T, \qquad (4.31)$$

and finally

$$X^{(n)} + \hat{U} \simeq F. \qquad (4.32)$$

Computing $\hat{U}$ follows

$$\hat{U} = \frac{h}{2}\begin{pmatrix} 0 \\ k_{1,0}x_0^{(n-1)} + k_{1,m}x_m^{(n-1)} \\ k_{2,0}x_0^{(n-1)} + k_{2,1}x_1^{(n-1)} + k_{2,m}x_m^{(n-1)} + k_{2,m+1}x_{m+1}^{(n-1)} \\ \vdots \\ k_{m-1,0}x_0^{(n-1)} + k_{m-1,1}x_1^{(n-1)} + \cdots + k_{m-1,m-2}x_{m-2}^{(n-1)} + k_{m-1,m}x_m^{(n-1)} + k_{m-1,m+1}x_{m+1}^{(n-1)} + \cdots + k_{m-1,2m-2}x_{2m-2}^{(n-1)} \\ k_{m,0}x_0^{(n-1)} + k_{m,m}x_m^{(n-1)} \\ k_{m+1,0}x_0^{(n-1)} + k_{m+1,1}x_1^{(n-1)} + k_{m+1,m}x_m^{(n-1)} + k_{m+1,m+1}x_{m+1}^{(n-1)} \\ \vdots \\ k_{2m-1,0}x_0^{(n-1)} + k_{2m-1,1}x_1^{(n-1)} + \cdots + k_{2m-1,2m-1}x_{2m-1}^{(n-1)} \end{pmatrix}, \qquad (4.33)$$

and hence

$$\hat{U} = \frac{h}{2} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ k_{1,0} & 0 & 0 & \cdots & 0 & 0 & k_{1,m} & 0 & 0 & \cdots & 0 & 0 \\ k_{2,0} & k_{2,1} & 0 & \cdots & 0 & 0 & k_{2,m} & k_{2,m+1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ k_{m-2,0} & k_{m-2,1} & k_{m-2,2} & \cdots & 0 & 0 & k_{m-2,m} & k_{m-2,m+1} & k_{m-2,m+2} & \cdots & 0 & 0 \\ k_{m-1,0} & k_{m-1,1} & k_{m-1,2} & \cdots & k_{m-1,m-2} & 0 & k_{m-1,m} & k_{m-1,m+1} & k_{m-1,m+2} & \cdots & k_{m-1,2m-2} & 0 \\ k_{m,0} & 0 & 0 & \cdots & 0 & 0 & k_{m,m} & 0 & 0 & \cdots & 0 & 0 \\ k_{m+1,0} & k_{m+1,1} & 0 & \cdots & 0 & 0 & k_{m+1,m} & k_{m+1,m+1} & 0 & \cdots & 0 & 0 \\ k_{m+2,0} & k_{m+2,1} & k_{m+2,2} & \cdots & 0 & 0 & k_{m+2,m} & k_{m+2,m+1} & k_{m+2,m+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ k_{2m-2,0} & k_{2m-2,1} & k_{2m-2,2} & \cdots & k_{2m-2,m-2} & 0 & k_{2m-2,m} & k_{2m-2,m+1} & k_{2m-2,m+2} & \cdots & k_{2m-2,2m-2} & 0 \\ k_{2m-1,0} & k_{2m-1,1} & k_{2m-1,2} & \cdots & k_{2m-1,m-2} & k_{2m-1,m-1} & k_{2m-1,m} & k_{2m-1,m+1} & k_{2m-1,m+2} & \cdots & k_{2m-1,2m-2} & k_{2m-1,2m-1} \end{pmatrix} \begin{pmatrix} x_0^{(n-1)} \\ x_1^{(n-1)} \\ x_2^{(n-1)} \\ \vdots \\ x_{2m-1}^{(n-1)} \end{pmatrix}. \tag{4.34}$$

Replacing (4.34) into (4.32), and substituting "$\simeq$" sign with "$=$" sign, we obtain the following recurrence relation:

$$X^{(n)} = RX^{(n-1)} + Q, \qquad \text{for } n = 1, 2, 3, \ldots, \tag{4.35}$$

in which, the $2m \times 2m$ matrix R and $2m$-vector Q are as follows:

$$R = -\frac{h}{2} \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ k_{1,0} & 0 & 0 & \cdots & 0 & 0 & k_{1,m} & 0 & 0 & \cdots & 0 & 0 \\ k_{2,0} & k_{2,1} & 0 & \cdots & 0 & 0 & k_{2,m} & k_{2,m+1} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ k_{m-2,0} & k_{m-2,1} & k_{m-2,2} & \cdots & 0 & 0 & k_{m-2,m} & k_{m-2,m+1} & k_{m-2,m+2} & \cdots & 0 & 0 \\ k_{m-1,0} & k_{m-1,1} & k_{m-1,2} & \cdots & k_{m-1,m-2} & 0 & k_{m-1,m} & k_{m-1,m+1} & k_{m-1,m+2} & \cdots & k_{m-1,2m-2} & 0 \\ k_{m,0} & 0 & 0 & \cdots & 0 & 0 & k_{m,m} & 0 & 0 & \cdots & 0 & 0 \\ k_{m+1,0} & k_{m+1,1} & 0 & \cdots & 0 & 0 & k_{m+1,m} & k_{m+1,m+1} & 0 & \cdots & 0 & 0 \\ k_{m+2,0} & k_{m+2,1} & k_{m+2,2} & \cdots & 0 & 0 & k_{m+2,m} & k_{m+2,m+1} & k_{m+2,m+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ k_{2m-2,0} & k_{2m-2,1} & k_{2m-2,2} & \cdots & k_{2m-2,m-2} & 0 & k_{2m-2,m} & k_{2m-2,m+1} & k_{2m-2,m+2} & \cdots & k_{2m-2,2m-2} & 0 \\ k_{2m-1,0} & k_{2m-1,1} & k_{2m-1,2} & \cdots & k_{2m-1,m-2} & k_{2m-1,m-1} & k_{2m-1,m} & k_{2m-1,m+1} & k_{2m-1,m+2} & \cdots & k_{2m-1,2m-2} & k_{2m-1,2m-1} \end{pmatrix},$$

$$Q = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{2m-1} \end{pmatrix}. \tag{4.36}$$

Now, considering the initial value (initial guess) $X^{(0)} = \mathbf{0}$, where $\mathbf{0}$ is the zero $2m$-vector, or $X^{(0)} = Q$, and using recurrence relation (4.35), one may steadily increase the degree of approximation until convergence is reached to a sufficient

Table 1: Numerical results for problem 5.1.

s	Exact solution	Proposed method	BPFs method
0	0	0	0.062500
0.1	0.100000	0.099990	0.062500
0.2	0.200000	0.199934	0.187501
0.3	0.300000	0.299803	0.312502
0.4	0.400000	0.399569	0.437503
0.5	0.500000	0.499207	0.562504
0.6	0.600000	0.598644	0.562504
0.7	0.700000	0.697940	0.687504
0.8	0.800000	0.797135	0.812504
0.9	0.900000	0.896286	0.937504

Table 2: Numerical results for problem 5.2.

s	Exact solution	Proposed method	BPFs method
0	1.000000	1.000000	0.998699
0.1	0.990050	0.989140	0.990933
0.2	0.960789	0.960254	0.952998
0.3	0.913931	0.913476	0.923699
0.4	0.852144	0.851663	0.847692
0.5	0.778801	0.778898	0.754026
0.6	0.697676	0.697636	0.702870
0.7	0.612626	0.612840	0.596596
0.8	0.527292	0.527638	0.543244
0.9	0.444858	0.445434	0.439992

Table 3: Running times of the algorithms in seconds.

m	Problem 5.1		Problem 5.2	
	Proposed method	BPFs method	Proposed method	BPFs method
8	0.0156	0.1872	0.0156	0.1872
16	0.0156	0.6084	0.0156	0.6084
32	0.0468	2.4024	0.0468	2.3556
64	0.0780	9.1885	0.0780	9.0481
128	0.1404	36.598	0.1404	36.052
256	0.2808	146.77	0.2808	143.38
512	1.1856	591.26	1.1544	581.59
1024	6.4896	2376.4	6.3336	2341.5

accuracy. To do this,

$$\|X^{(n)} - X^{(n-1)}\| < \varepsilon$$

or

$$\frac{\|X^{(n)} - X^{(n-1)}\|}{\|X^{(n)}\|} < \varepsilon$$

, for arbitrary small ε, may be considered as stopping condition, where $\|.\|$ is an arbitrary vector norm. Then, an approximate solution $x(s) \simeq X^T\mathbf{T}(s)$ can be computed for integral equation (4.22).

5 Numerical results

In this section, by using the proposed numerical method, we solve some test problems. The approximate results obtained by the method are also compared with the exact solutions and the results of BPFs method presented in [12]. Moreover, the running times associated with both methods are given. It should be mentioned that all the computations associated with the methods are performed using Matlab software on a laptop having the Intel 2.10 GHz processor.

5.1 Test problems

Two problems are solved here. The exact and approximate results for these problems are calculated at ten points $s = 0, 0.1, 0.2, \ldots, 0.9$.

1. **Corollary 5.1** *[12]Let us consider the following Volterra integral equation of the second kind:*

$$x(s) + \int_0^s (st^2 + s^2t)x(t)t = s + \frac{7}{12}s^5, \quad (5.37)$$

with the exact solution $x(s) = s$. Table 1 gives the numerical results for $m = 8$.

Corollary 5.2 *[12] For the following Volterra integral equation of the second kind:*

$$x(s) + \int_0^s stx(t)t = e^{-s^2} + \frac{s(1-e^{-s^2})}{2}, \quad (5.38)$$

with the exact solution $x(s) = e^{-s^2}$, Table 2 shows the results for $m = 16$.

Referring to the numerical results given in Tables 1 and 2, it is obvious that the proposed method in this article is more accurate than the BPFs method.

5.2 Running times

For further evaluation of the computational efficiency of the proposed method, the running times of its algorithm as well as those of the BPFs method are given here. Table 3 gives the running times of both algorithms in seconds which clearly show that the algorithm of the proposed method runs very fast in comparison with that of the BPFs method.

6 Conclusion

A numerical method was proposed for solving second kind Volterra integral equations based on vector forms and a special representation of triangular functions. We saw that this approach without applying any projection method transforms the integral equation to a recurrence relation. The accuracy and computational efficiency of the method was checked on some test problems. The results showed that the method runs very fast and has a good accuracy.

References

[1] E. Babolian, Z. Masouri, S. Hatamzadeh-Varmazyar, A direct method for numerically solving integral equations system using orthogonal triangular functions, *International Journal of Industrial Mathematics* 1 (2009) 135-145.

[2] E. Babolian, Z. Masouri, S. Hatamzadeh-Varmazyar, A set of multi-dimensional orthogonal basis functions and its application to solve integral equations,*International Journal of Applied Mathematics and Computation* 2 (2010) 032-049.

[3] E. Babolian, Z. Masouri, S. Hatamzadeh-Varmazyar, Introducing a direct method to solve nonlinear Volterra and Fredholm integral equations using orthogonal triangular functions, *Mathematics Scientific Journal* 5 (2009) 11-26.

[4] E. Babolian, Z. Masouri, S. Hatamzadeh-Varmazyar, New direct method to solve nonlinear Volterra-Fredholm integral and integro-differential equations using operational matrix with block-pulse functions, *Progress In Electromagnetics Research B* 8 (2008) 59-76.

[5] E. Babolian, Z. Masouri, S. Hatamzadeh-Varmazyar, Numerical solution of nonlinear Volterra-Fredholm integro-differential equations via direct method using triangular functions, *Computers & Mathematics with Applications* 58 (2009) 239-247.

[6] A. Deb, G. Sarkar, A. Sengupta, Triangular Orthogonal Functions for the Analysis of Continuous Time Systems, *Anthem Press, London*, 2011.

[7] L. M. Delves, J. L. Mohamed, Computational Methods for Integral Equations, *Cambridge University Press, Cambridge*, 1985.

[8] S. Hatamzadeh-Varmazyar, Z. Masouri, A fast numerical method for analysis of one- and two-dimensional electromagnetic scattering using a set of cardinal functions, *Engineering Analysis with Boundary Elements* 36 (2012) 1631-1639.

[9] S. Hatamzadeh-Varmazyar, Z. Masouri, A numerical approach for calculating the radar cross-section of two-dimensional perfect electrically conducting structures, *Journal of Electromagnetic Waves and Applications* 28 (2014) 1360-1375.

[10] S. Hatamzadeh-Varmazyar, Z. Masouri, Calculation of electric charge density based on a numerical approximation method using triangular functions, *Advanced Computational Techniques in Electromagnetics* (2013) 1-11.

[11] S. Hatamzadeh-Varmazyar, Z. Masouri, Numerical expansion-iterative method for analysis of integral equation models arising in one-and two-dimensional electromagnetic scattering, *Engineering Analysis with Boundary Elements* 36 (2012) 416-422.

[12] Z. Masouri, Numerical expansion-iterative method for solving second kind Volterra and Fredholm integral equations using block-pulse functions, *Advanced Computational Techniques in Electromagnetics* (2012) 1-7.

[13] Z. Masouri, E. Babolian, S. Hatamzadeh-Varmazyar, An expansion-iterative method for numerically solving Volterra integral equation of the first kind, *Computers & Mathematics with Applications* 59 (2010) 1491-1499.

[14] Z. Masouri, S. Hatamzadeh-Varmazyar, An analysis of electromagnetic scattering from finite-width strips, *International Journal of Industrial Mathematics* 5 (2013) 199-204.

[15] Z. Masouri, S. Hatamzadeh-Varmazyar, Applying integral equation modeling technique in determination of charge distribution on conducting structures, *Advanced Computational Techniques in Electromagnetics* (2013) 1-8.

[16] Z. Masouri, S. Hatamzadeh-Varmazyar, Evaluation of current distribution induced on perfect electrically conducting scatterers, *International Journal of Industrial Mathematics* 5 (2013) 167-173.

Saeed Hatamzadeh-Varmazyar, Ph.D. in Electrical Engineering, is an Assistant Professor at Islamshar Branch, Islamic Azad University, Tehran, Iran. His research interests include computational Electromagnetics, numerical methods for solving integral equations, electromagnetic theory, singular integral equations, electromagnetic radiation and antenna, and propagation of electromagnetic waves, including scattering, diffraction, and etc. Dr. Hatamzadeh-Varmazyar is the author of many research articles published in scientific journals. More details may be found on his official website available online at http://www.hatamzadeh.ir/ and http://www.hatamzadeh.org/.

Zahra Masouri received her B.Sc., M.Sc., and Ph.D. degrees in Applied Mathematics, respectively, in 1998, 2000, and 2010, as the first rank graduate at all the mentioned educational levels. She is currently an Assistant Professor at Islamshar Branch, Islamic Azad University, Tehran, Iran. Her research interest is in numerical analysis; numerical solution of linear and nonlinear integral, integro-differential, and differential equations; numerical linear algebra; and computational Electromagnetics. Dr. Masouri was the research manager and the research vice-chancellor at Khorramabad Branch of Islamic Azad University for some years. She also is the author of many research articles published in scientific journals or presented at various conferences and member of the editorial board of some international journals.