QJPS
Al-Qadisiyah Journal of Pure Science

Al-Qadisiyah Journal of Pure Science

ISSN(Printed): 1997-2490 ISSN(Online): 2411-3514

DOI: /10.29350/jops.

http://qu.edu.iq/journalsc/i

An Efficient Numerical Scheme for Solving Time Delay Volterra Integral Equations by Using Basis Functions

Authors Names

Jameel Kamel Saeed

Article History

Received on: X/2/2024
Revised on: X/2/2024
Accepted on: X/2/2024

Keywords:

Operational matrix; Delay integral equation; orthogonal functions; Delay operational matrix.

DOI:https://doi.org/10.29350/ jops. XXXXX

ABSTRACT

This article proposes a simple efficient method for solving some volterra integral equations with time delay that arises in different applied issues. By using basis orthogonal functions and their operational matrix of integration, integral equations can be reduced to a sparse linear lower triangular system which can be solved by forward substitution. Numerical examples show that the proposed scheme has a suitable degree of accuracy.

1.Introduction

Many numerical methods have been introduced to solve IE numerically, but each method has its own shortcomings. Most of the numerical methods convert the model equations containing IE into discrete model equations, which appear in the form of a set of linear or nonlinear algebraic equations. In the case of direct solvers for solving this set of algebraic equations, the computational cost of a large system worsens due to the large computational time and memory requirements required for mathematical operations. Therefore, it is still a challenging task to introduce a simple and efficient technique for IEs. In this method, we focus on improving the efficiency of direct solvers by using

Department of Mathematics, South Tehran Branch, Islamic Azad University, Tehran, Iran. Email

orthogonal base functions method. Direct solvers are useful for solving problems that iterative solvers struggle with. Most of time delay Integral equations are frequently encountered in engineering and biological modeling processes. In [1,2], delay integral equations (DIEs) and delay integro-differential equations (DIDEs) are solved by different methods. In [3], the approximate solutions of optimal control of time delay systems are derived by Block pulse functions. Apiecewise fuzzy polynomial interpolation method is proposed to approximate the solutions of fuzzy delay integral equations with weakly singular kernels has been reviewed [4]. In [5], a numerical method to solve a fuzzy differential equation via differential inclusions with their membership distribution functions is obtained. Bloor et al. [6] used collocation technique to solve delayed IEs using Taylor polynomials. Mosleh and Otadi [7] used the least squares technique to solve Volterra delay IEs. Bica and Popescu [8] developed an iterative technique to solve nonlinear Volterra fuzzy IEs with fixed delay.

Nowadays, basis functions such as Haar wavelets were used to derive solutions of integral and differential equations that can be seen in [9-11]. It also presents a new numerical method for solving integral Volterra functional equations with variable bounds and mixed delays [12]. The problem of the existence of solution to functional integral equations has been investigated in different references [13] and applications to this type of equations have also been found in models of swelling porous media [14], [15].

In this paper we use Block pulse functions for numerical solving Volterra integral equations with constant time delay $\tau > 0$ as,

$$g(t) = f(t) + \int_0^t k(t,s)g(s-\tau)ds \quad t \in [0,T], \qquad \tau \in (0,T)$$

This article is organized as follows. In section 2, we explain block pulse functions and integration operational matrix and functions containing time delay $f(t-\tau)$. Section 3 is devoted to solving Volterra integral equations with time delay. Section 4 is devoted to error estimation in Block Pulse functions approximation and in Section 5 we achieve numerical examples to show the accuracy of the method and the culmination of paper in section 6 is the conclusion.

2. Preliminaries

The aim of this section is to interpret notations and definition of the block pulse functions that have expressed entirely in [9].

2.1. Definition

We define the m-set of BPFs as,

$$\phi_i^{(m)}(t) = \begin{cases} 1 & (i-1)h \le t < ih \\ 0 & \text{otherwise.} \end{cases}$$

with $t \in [0,T)$, $i = 1,2,...,m$ and $h = \dfrac{T}{m}$.

The primary properties of BPFs are disjointness and orthogonality that can be expressed as follows

$$\phi_i^{(m)}(t)\phi_j^{(m)}(t) = \delta_{ij}\phi_i^{(m)}(t),$$

$$\int_0^T \phi_i^{(m)}(t)\phi_j^{(m)}(t)dt = h\delta_{ij}, \qquad i, j = 1, 2, \dots, m$$

2.2. Functions approximation

The orthogonality property of BPFs is the base of expanding functions into their block pulse series. A real bounded function $f(t) \in L^2[0,T)$, can be expanded into a block pulse series as

$$f(t) \square \hat{f}_m(t) = \sum_{i=1}^{m} f_i \phi_i^{(m)}(t),$$

where $f_i = \frac{1}{h}\int_0^T f(t)\phi_i^{(m)}(t)\,dt$ is the block pulse coefficient with respect to the ith BPF $\phi_i^{(m)}(t)$. In the vector form we have,

$$f(t) \square \hat{f}_m(t) = F^T\Phi(t) = \Phi^T(t)F,$$

where $F = (f_1, f_2, \dots, f_m)^T$. Let $k(s,t) \in L^2([0,T_1)\times[0,T_2))$. It can be expanded as $k(s,t) = \Psi^T(s)K\Phi(t) = \Phi^T(t)K^T\Psi(s),$ where $\Psi(s),\ \Phi(t)$ are m_1, m_2 dimensional BPFs vectors respectively, and $K = (k_{ij}),\ i = 1, 2, \dots, m_1,\ j = 1, 2, \dots, m_2$ is the $m_1 \times m_2$ block pulse coefficient matrix with

$$k_{ij} = \frac{1}{h_1 h_2}\int_0^{T_1}\int_0^{T_2} k(s,t)\Psi_i^{(m_1)}(s)\Phi_j^{(m_2)}(t)dtds,$$

where $h_1 = \frac{T_1}{m_1}, h_2 = \frac{T_2}{m_2}$. For convenience, we put $m_1 = m_2 = m$.

2.3. Integration operational matrix

Computing $\int_0^t \phi_i^{(m)}(s)ds$ one obtains

$$\int_0^t \phi_i^{(m)}(s)ds = \begin{cases} 0 & 0 \le t < (i-1)h, \\ t-(i-1)h & (i-1)h \le t < ih, \\ h & ih \le t < T \end{cases}$$

From [5], we will have:

$$\int_0^t \Phi(s)ds \square P\Phi(t),$$

where the operational matrix of integration is given by

$$P = \frac{h}{2}\begin{pmatrix} 1 & 2 & 2 & \cdots & 2 \\ 0 & 1 & 2 & \cdots & 2 \\ 0 & 0 & 1 & \cdots & 2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{m\times m}$$

So, the integral of every function f(t) can be approximated as follows

$$\int_0^t f(x)dx \cong \int_0^t F^T\Phi(x)dx \cong F^T P\Phi(t)$$

2.4. Functions containing time delay

In order to approximate a function containing time delay, we consider a block pulse function containing time delay $\tau = (q+\lambda)h$ with a nonnegative integer q and $0 \le \lambda < 1$ that can be expressed as

$$\phi_i^{(m)}(t-\tau)=\begin{cases}\phi_{i+q}^{(m)}(t)+\phi_\lambda^{(m)}(t-(i+q)h)-\phi_\lambda^{(m)}(t-(i+q-1)h) & \text{for } i<m-q\\ \phi_{i+q}^{(m)}(t)-\phi_\lambda^{(m)}(t-(i+q-1)h) & \text{for } i=m-q.\\ 0 & \text{for } i>m-q\end{cases}$$

Where $\phi_\lambda^{(m)}(t)=\begin{cases}1 & 0\le t<\lambda h,\\ 0 & \text{otherwise.}\end{cases}$

In a vector form, $\phi_i^{(m)}(t-\tau)=\Delta_i^T H^q \Phi(t)-\Delta_i^T H^q \Phi_\lambda(t)+\Delta_i^T H^{q+1}\Phi_\lambda(t),$

where $\Delta_i=(0,\ldots,0,1,0,\ldots,0)^T$ with 1 in i-th position and

$$H=\begin{pmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\ \vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\\0&0&0&\cdots&0\end{pmatrix}_{m\times m}$$

To avoid the expression $\Phi_\lambda(t)$ in the above equation, we expand the function $\phi_i^{(m)}(t-\tau)$ into its block pulse series :

$$\phi_i^{(m)}(t-\tau)=(c_{i1},c_{i2},\ \ldots,\ c_{im})\Phi(t),$$

where the block pulse coefficients $c_{ij}\ (i,j=1,2,\cdots,m)$ are:

$$\begin{aligned}c_{ij}&=\frac{1}{h}\int_0^T \phi_i(t-\tau)\phi_j(t)dt=\frac{1}{h}\int_{(j-1)h}^{jh}\phi_i(t-\tau)dt,\\ &=\frac{1}{h}\Delta_i^T H^q\ (\int_{(j-1)h}^{jh}\Phi(t)dt-\int_{(j-1)h}^{jh}\Phi_\lambda(t)dt+H\int_{(j-1)h}^{jh}\Phi_\lambda(t)dt\ )\\ &=\Delta_i^T\ ((1-\lambda)H^q+\lambda H^{q+1})\Delta_j.\end{aligned}$$

Noticing that the expression $\Delta_i^T((1-\lambda)H^q+\lambda H^{q+1})\Delta_j$ is just the single entry positioned in the ith row and jth column of the matrix $(1-\lambda)H^q+\lambda H^{q+1}$, we can expand the whole block pulse function vector containing time delay $\tau=(q+\lambda)h$ into its block pulse series in a vector form :

$$\Phi(t-\tau)=((1-\lambda)H^q+\lambda H^{q+1})\Phi(t).$$

In the above equation, the matrix $(1-\lambda)H^q+\lambda H^{q+1}$ is usually called the block pulse operational matrix for time delay, or simply the delay operational matrix. Expressing concretely, it is :

$$(1-\lambda)H^q+\lambda H^{q+1}=\begin{array}{cccccccc} & & & \text{(q+1)-th} & & & & \\ & & & \downarrow & & & & \end{array}$$

$$\left(\begin{array}{cccccccc}0&\cdots&0&1-\lambda&\lambda&0&\cdots&0\\0&\cdots&0&0&1-\lambda&\lambda&\cdots&0\\ \vdots&\cdots&\vdots&\vdots&\vdots&\vdots&\ddots&\vdots\\0&\cdots&0&0&0&0&\cdots&\lambda\\0&\cdots&0&0&0&0&\cdots&1-\lambda\\0&\cdots&0&0&0&0&\cdots&0\\ \vdots&\cdots&\vdots&\vdots&\vdots&\vdots&\cdots&\vdots\\0&\cdots&0&0&0&0&\cdots&0\end{array}\right).$$

Therefore, the block pulse series of a function containing time delay can easily be obtained as :

$$f(t-\tau) \simeq F^T \Phi(t-\tau) = F^T((1-\lambda)H^q + \lambda H^{q+1})\Phi(t).$$

3. Solving Volterra integral equations with time delay

We consider the following Volterra integral equation with constant time delay $\tau > 0$

$$g(t) = f(t) + \int_0^t k(t,s) g(s-\tau) ds, \quad t \in [0,T], \quad \tau \in (0,T),$$

where the function $g \in L^2[0,T]$ is the unknown function, while the functions $f \in L^2[0,T]$ and $k(t,s) \in L^2([0,T]\times[0,T])$ are the known functions.

We approximate $g(t)$, $f(t)$, $k(t,s)$ and $g(s-\tau)$ by relations as follows

$$g(t) \simeq G^T\Phi(t) = \Phi^T(t)G, \quad f(t) \simeq F^T\Phi(t) = \Phi^T(t)F,$$

$$k(t,s) \simeq \Phi^T(t)K\Psi(s) = \Psi^T(s)K^T\Phi(t),$$

$$g(s-\tau) \simeq G^T\Psi(s-\tau) \simeq G^T((1-\lambda)H^q + \lambda H^{q+1})\Psi(s),$$

If we put $A = (1-\lambda)H^q + \lambda H^{q+1}$, then we have,

$$g(s-\tau) \simeq G^T A \Psi(s).$$

With substituting above approximation in equation, we have

$$G^T\Phi(t) \simeq F^T\Phi(t) + \int_0^t G^T A\Psi(s)\Psi^T(s)K^T\Phi(t)ds,$$

$$\simeq F^T\Phi(t) + G^T A(\int_0^t \Psi(s)\Psi^T(s)ds)K^T\Phi(t).$$

Let K_i be the ith row of the constant matrix K^T , R_i be the ith row of the integration operational matrix P and D_{K_i} be a diagonal matrix with K_i as its diagonal entries. By the previous relations and assuming

$m_1 = m_2$, we will have,

$$(\int_0^t \Psi(s)\Psi^T(s)ds)K^T\Phi(t) = (\int_0^t \Phi(s)\Phi^T(s)ds)K^T\Phi(t)$$

$$= \begin{pmatrix} R_1\Phi(t) & 0 & \cdots & 0 \\ 0 & R_2\Phi(t) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_m\Phi(t) \end{pmatrix} \begin{pmatrix} K_1 \\ K_2 \\ \vdots \\ K_m \end{pmatrix} \Phi(t)$$

$$= \begin{pmatrix} R_1\Phi(t)K_1\Phi(t) \\ R_2\Phi(t)K_2\Phi(t) \\ \vdots \\ R_m\Phi(t)K_m\Phi(t) \end{pmatrix} = \begin{pmatrix} R_1\Phi(t)\Phi^T(t)K_1^T \\ R_2\Phi(t)\Phi^T(t)K_2^T \\ \vdots \\ R_m\Phi(t)\Phi^T(t)K_m^T \end{pmatrix}$$

$$= \begin{pmatrix} R_1 D_{K_1} \\ R_2 D_{K_2} \\ \vdots \\ R_m D_{K_m} \end{pmatrix} \Phi(t) = B\Phi(t),$$

where

$$B = \frac{h}{2}\begin{pmatrix} k_{11} & 2k_{21} & \cdots & 2k_{m1} \\ 0 & k_{22} & \cdots & 2k_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & k_{mm} \end{pmatrix}_{m\times m} .$$

we will have

$$G^T \Phi(t) \square F^T \Phi(t) + G^T AB\Phi(t),$$

or

$$G^T (I - AB) \square F^T ,$$

So, by setting $M = I - AB$ and replacing $\square$ by =, we will have,

$$M^T G = F ,$$

which is a linear system of equations with lower triangular coefficients matrix that gives the approximate block pulse coefficient of the unknown function $g(t)$.

4. Error estimation in block pulse functions approximation

In this section, we will show that the rate of block pulse functions approximation is O(h) and because of it we can obtain a good degree of accuracy.

Theorem 1. Suppose that f(t) is an arbitrary real bounded function, which is square integrable in the interval [0,1), and $e(t) = f(t) - \hat{f}_m(t)$, $t \in I = [0,1)$, which $\hat{f}_m(t) = \sum_{i=1}^{m} f_i \phi_i(t)$ is the block pulse series of $f(t)$. Then, $\|e(t)\| \le \frac{h}{2\sqrt{3}} \sup_{t\in I} |f \ (t)|$

Proof. Let,

$$e_i(t) = \begin{cases} f(t) - f_i & t \in D_i \\ 0 & t \in I - D_i \end{cases}$$

where $D_i = \left\{ t \quad (i - \ )h \le t < ih \quad h = \frac{1}{m} \right\}$ and $i = 1, 2, ..., m.$ We have,

$$e_i(t) = f(t) - \frac{1}{h}\int_{(i-1)h}^{ih} f(s)ds = \frac{1}{h}\int_{(i-1)h}^{ih} \left(f(t) - f(s)\right)ds$$

now by mean value theorem, we get,

$$e_i(t) = \frac{f\ '(\eta_i)}{h}\int_{(i-1)h}^{ih} (t - s)ds = f\ '(\eta_i)(t + \left(-i + \frac{1}{2}\right)h) \qquad t \ \ \eta_i \in D_i \qquad i = \qquad m$$

then

$$\|e_i(t)\|^2 = \int_{(i-1)h}^{ih} |e_i(t)|^2 dt = (f \ \ (\eta_i) \ ^2 \int_{(i-1)h}^{ih} \ t + \left(-i + \frac{1}{2}\right)h \ ^2 dt$$

$$= \frac{h^3}{12}(f\ '(\eta_i))^2, \quad \eta_i \in D_i \quad i = \qquad ..., m$$

Consequently

$$\left\|e(t)\right\|^{2}=\int_0^1\left|e(t)\right|^2 dt=\int_0^1\left(\sum_{i=1}^{m}e_i(t)\right)^2 dt$$

$$=\int_0^1\left(\sum_{i=1}^{m}e_i^{\,2}(t)+2\sum_{i<j}e_i(t)e_j(t)\right)dt=\sum_{i=1}^{m}\int_0^1 e_i^2(t)dt=\sum_{i=1}^{m}\left\|e_i(t)\right\|^2$$

$$=\frac{h^3}{12}\sum_{i=1}^{m}\left(f'(\eta_i)\right)^2\le\frac{h^2}{12}\sup_{t\in I}\left|f'(t)\right|^2,$$

or, $\left\|e(t)\right\|\le\dfrac{h}{2\sqrt{3}}\sup_{t\in I}\left|f\ \ (t)\right|$

hence, $\left\|e(t)\right\|=O\ h$

Theorem 2. Suppose that $f\ (s,t)\in L^2\left([0,1)\times[0,1)\right)$ and $e(s,t)=f\ (s,t)-\hat{f}_m(s,t)\quad(s\ \ t)\in D=[\quad)\times[\quad)$ where $\hat{f}_m(s,t)=\sum_{i=1}^{m}\sum_{j=1}^{m}f_{ij}\,\psi_i^{(m)}(s)\phi_j^{(m)}(t)$ is the block pulse series of $f\ (s,t)$. Then,

$$\left\|e(s\ \ t)\right\|\le\frac{h}{2\sqrt{3}}\left(\ _{(x,y)\in D}\left|f_s'\ x\ \ y\ \ \right|^2+\ _{(x,y)\in D}\left|f_t'\ x\ \ y\ \ \right|^2\right)^{\frac{1}{2}}$$

Proof. Let,

$$e_{ij}(s\ \ t)=\begin{cases}f\ (s,t)-f_{ij} & (s,t)\in D_{ij}\\ 0 & (s,t)\in D-D_{ij}\ .\end{cases}$$

Where $D_{ij}=\left\{(s,t):(i-1)h\le s<ih\quad(j-\ \ )h\le t<jh\ \ h=\dfrac{1}{m}\right\}$ and $i,j=1,2,...,m$. We have,

$e_{ij}(s,t)=f\ (s,t)-\dfrac{1}{h^2}\int_{(i-1)h}^{ih}\int_{(j-1)h}^{jh}f\ (x,y)dydx=\dfrac{1}{h^2}\int_{(i-1)h}^{ih}\int_{(j-1)h}^{jh}\left(f\ (s,t)-f\ (x,y)\right)dydx$, now by mean value theorem, we get,

$$e_{ij}(s,t)=\frac{1}{h^2}\int_{(i-1)h}^{ih}\int_{(j-1)h}^{jh}\left((s-x)f_s'(\eta_i,\eta_j)+(t-y)f_t'(\eta_i,\eta_j)\right)dydx$$

$$=f_s'(\eta_i,\eta_j)\left(s+\left(-i+\frac{1}{2}\right)h\right)+f_t'(\eta_i,\eta_j)\left(t+\left(-j+\frac{1}{2}\right)h\right),\ \ (s,t),(\eta_i,\eta_j)\in D_{ij}\,.$$

then

$$\left\|e_{ij}(s\ \ t)\right\|^2=\int_{(i-1)h}^{ih}\int_{(j-1)h}^{jh}\left|e_{ij}(s\ \ t)\right|^2 dtds$$

$$=\frac{h^4}{12}\left(f\,{}_s'^2(\eta_i,\eta_j)+f\,{}_t'^2(\eta_i,\eta_j)\right),\ \ \left(\eta_i,\eta_j\right)\in D_{ij}\quad i\ \ j=\qquad\quad m$$

Consequently

$$\|e(s\ \ t)\|^2 = \int_0^1\int_0^1 |e(s\ \ t)|^2\, dtds = \int_0^1\int_0^1 \left(\sum_{i=1}^{m}\sum_{j=1}^{m} e_{ij}(s,t)\right)^2 dtds$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{m}\int_0^1\int_0^1 \left(e_{ij}(s,t)\right)^2 dtds = \sum_{i=1}^{m}\sum_{j=1}^{m}\|e_{ij}(s,t)\|^2$$

$$= \frac{h^4}{12}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(f_s'^2(\eta_i,\eta_j) + f_t'^2(\eta_i,\eta_j)\right) \le \frac{h^2}{12}\left(\sup_{(x,y)\in D}|f_s'(x,y)|^2 + \sup_{(x,y)\in D}|f_t'(x,y)|^2\right),$$

$$\|e(s\ \ t)\| \le \frac{h}{2\sqrt{3}}\left(\ _{(x,y)\in D}|f_s'\ x\ \ y\ \ |^2 + \ _{(x,y)\in D}|f_t'\ x\ \ y\ \ |^2\right)^{\frac{1}{2}}$$

or,

Hence, $\|e(s\ \ t)\| = O\ \ h$

5. Numerical examples

To illustrate the theoretical results stated in Section 3 we consider the examples below. Let G_i denote the Block pulse coefficient of exact solution of the given examples, and let g_i be the Block pulse coefficient of computed solutions by the presented method. The error is defined as, $E\|_{\infty} = max_{1\le i\le m} \mid G_i - g_i \mid$.

Example 1. Consider the following Volterra integral equation with (constant) time delay $\tau > 0$,

$$g(t) = -\frac{t^4}{12} + \tau\frac{t^3}{3} + (1 - \frac{\tau^2}{2})t^2 + \int_0^t (t-s)g(s-\tau)ds \quad s,t \in [0,T], \tau \in (0,T)$$

With the exact solution $g(t) = t^2$, for $0 \le t \le T$ .The numerical results are shown in Table 1.

Example 2. Consider the following Fredholm integral equation time delay $\tau > 0$,

$$g(t) = t(Tcos(T-\tau) - sin(T-\tau) - sin(\tau)) + sin(t)$$
$$+\int_0^T (ts)g(s-\tau)ds \quad s,t \in [0,T], \tau \in (0,T)$$

With the exact solution $g(t) = sin\,t$, for $0 \le t \le T$ .The numerical results are shown in Table 2.

Conclusion

Using Block pulse functions as basic functions to solve the Volterra integral equations with constant time delay is very simple and effective in comparison with other methods. Its applicability and accuracy is checked on some examples. In these examples the norm infinity of error is given only for 10 specific values of τ . The benefits of this method are low cost of setting up the equations without applying any projection method such as Galerkin, collocation, etc.

Table 1. Results for example 1 with m=32.

$T = 0.1$		$T = 0.5$		$T = 1$	
τ	$\|E\|_\infty$	τ	$\|E\|_\infty$	τ	$\|E\|_\infty$

0.001	2.252537E – 10	0.005	1.432365E – 8	0.01	2.417813E – 7
0.004	7.890189E – 10	0.020	5.053922E – 8	0.04	8.895481E – 6
0.007	1.051373E – 9	0.035	6.775494E – 8	0.07	1.189044E – 6
0.010	3.235521E – 9	0.050	2.075733E – 6	0.10	3.594127E – 5
0.013	7.100636E – 8	0.065	4.540855e – 5	0.13	7.787479E – 5
0.016	1.311426E – 8	0.080	8.363133E – 5	0.16	1.422281E – 4
0.019	2.172782E – 7	0.095	1.382092E – 5	0.19	2.332652E – 4
0.022	3.337662E – 7	0.110	2.118146E – 5	0.22	3.550228E – 4
0.025	4.847992E – 7	0.125	3.070148E – 4	0.25	5.113481E – 4

Table 2. Results for example 2 with m=32.

$T = 0.1$		$T = 0.5$		$T = 1$	
τ	$\|E\|_\infty$	τ	$\|E\|_\infty$	τ	$\|E\|_\infty$
0.001	2.585682E – 8	0.005	1.790986E – 6	0.01	4.756345E – 5
0.004	2.066313E – 8	0.020	1.435647E – 6	0.04	3.782767E – 5
0.007	3.904944E – 9	0.035	3.296881E – 6	0.07	1.189205E – 4
0.010	3.318922E – 8	0.050	2.090175E – 5	0.10	4.142219E – 4
0.013	9.939014E – 8	0.065	6.382695E – 5	0.13	1.323771E – 4
0.016	2.034678E – 7	0.080	1.310208E – 4	0.16	2.704209E – 3
0.019	3.541915E – 7	0.095	2.279810E – 4	0.19	4.642849E – 3
0.022	5.603294E – 7	0.110	3.601617E – 4	0.22	7.220738E – 3
0.025	8.306449E – 7	0.125	5.329749E – 4	0.25	1.051349E – 3

7.

References

[1]. *L. Wen, W. Wang, Y. Yu*, Dissipativity and asymptotic stability of nonlinear neutral delay integro-differential equations, Nonlinear Analysis 72 (2010) 1746–1754.

[2]. *B. Xu, R. Yuan,* On the positive almost periodic type solutions for some nonlinear delay integral equations, J. Math. Anal. Appl. 304 (2005) 249–268.

[3]. *H. R. Marzban, M. Razzaghi*, Optimal control of linear delay systems via hybrid of block-pulse and Legendre polynomials, Journal of the Franklin Institute, 341 (2004) 279–293.

[4]. Ting Deng, Jin Huang, Hongyan Liu, Hu Li, Approximate solutions of fuzzy delay integral equations with weakly singular kernels by piecewise fuzzy polynomial interpolation, Fuzzy Sets and Systems, Volume 470, 30 October 2023, 108652.

[5]. Xiao-Ming Liu a b, Jun Jiang a, Ling Hong, A numerical method to solve a fuzzy differential equation via differential inclusions, Fuzzy Sets and Systems, Volume 404, 1 February 2021, Pages 38-61.

[6]. A. Bellour, M. Bousselsal, A Taylor collocation method for solving delay integral equations, Numer. Algorithms 65(4) (2014) 843–857.

[7]. M. Mosleh, M. Otadi, Least squares approximation method for the solution of Hammerstein Volterra delay integral equations, Appl. Math. Comput. 258 (2015) 105–110.

[8]. A.M. Bica, C. Popescu, Numerical solutions of the nonlinear fuzzy Hammerstein Volterra delay integral equations, Inf. Sci. 223 (2013) 236–255.

[9]. R. Amin, S. Nazir, I.G. Magarino, Efficient sustainable algorithm for numerical solution of nonlinear delay Fredholm-Volterra integral equations via Haar wavelet for dense sensor networks in emerging telecommunications, Trans. Emerg. Telecommun. Technol. 30 (11) (2020) 1–12.

[10]. M. M. Khashan, R. Amin, M.I. Syam, A new algorithm for fractional Riccati type differential equations by using Haar wavelet, Mathematics 7 (2019) 545–557.

[11]. M. H. Heydari, M. R. Hooshmandasl, C. Cattani, F. M. Maalek Ghaini, An efficient computational method for solving nonlinear stochastic Itô integral equations: Application for stochastic problems in physics, Journal of Computational Physics 283 (2015) 148–168.

[12]. Dajana Conte, Eslam Farsimadan, Leila Moradi, Francesco Palmieri, Beatric Paternoster, Numerical solution of delay Volterra functional integral equations with variable bounds, Applied Numerical Mathematics, Available online 11 May 2023.

[13]. Sohrab Bazm, Pedro Lima, Somayeh Nemati, Analysis of the Euler and trapezoidal discretization methods for the numerical solution of nonlinear functional Volterra integral equations of Urysohn type, Journal of Computational and Applied Mathematics Volume 398, 15 December 2021, 113628.

[14]. RochaAdson M, Numerical analysis of a collocation method for functional integral equations Appl. Numer. Math. (2018).

[15]. Adson M. Rocha, Juarez S. Azevedo, Saulo P. Oliveira, Maicon R. Correa, Numerical analysis of a collocation method for functional integral equations, Applied Numerical Mathematics Volume 134, December 2018, Pages 31-45.