Special conference edition, November, 2019 http://dx.doi.org/10.4314/bajopas.v12i1.70S

Bayero Journal of Pure and Applied Sciences, 12(1): 462 - 469
ISSN 2006 - 6996

Numerical Solution of System of Linear Volterra Integral Equations Via Hybrid of Taylor and Block-pulse Functions

[1]Sirajo Lawan Bichi and [2]Shaaibu Umar Faruk*
[1,2]Department of Mathematical Sciences,
Bayero University Kano, Kano, Nigeria,
Correspondence author: faroukuumar32@gmail.com, 08062717058

ABSTRACT

In this paper, we use a combination of Taylor and Block-Pulse functions on the interval[0, 1/2], which is a hybrid function, to develop a numerical method for approximating the solution of a system of linear Volterra integral equation of the second kind. To achieve the proposed method, the problem of system of linear Volterra integral equation of the second kind is reduced to a system of linear equations using Operational matrices. By using numerical examples we show our estimation have a good degree of accuracy.

Keywords: *Block-pulse functions, Volterra, Operational matrices, Taylor*

Introduction

Integral equations are a class of important models in applied sciences. Since it is difficult to obtain the analytic solutions of these equations, numerical methods to obtain approximate solutions are of interest. One of such methods is the Piecewise constant basis functions (Babolian et al.,2011). which were introduced by Alfred Haar in 1910. Volterra integral equations naturally appear in history dependent problems such as population dynamics, renewal equations, nuclear reactor dynamics, viscoelasticity, study of epidemics, superfluidity, damped vibrations, heat conduction, and diffusion (Babolian and Masouri, 2008). Block Pulse functions have been used by many researchers for various problems such as solving differential equations , integral equations , population balance equations (Balakumar and Murugesan, 2013).
Recently, Maleknajad and Mahmoudi developed a numerical solution of linear fredholm integral equation by using hybrid Taylor and block pulse funtions [Maleknajad and Mahmoudi(2004)]. Maleknejad et al. developed a numerical solution of integral equations system of the second kind by Block pulse functions [Maleknajad et al.(2005)]. Babolian and Masouri proposed a direct method to solve volterra integral equation of the first kind by using Operational matrix with Block pulse functions[Babolian and Masouri (2008)].
Leyla et al. developed a numerical solution of volterra fredholm integro differential equation by Block pulse functions and Operational matrices [leyla Rahmani et al.(2011)].

Babolian et al proposed a numerical method for solving fredholm volterra integral equations in two dimensional spaces by using Block pulse functions and an Operational matrix(Babolian et a al., 2011). Balakumar and Murugesan proposed a numerical solution of systems of linear volterra integral equations by using Block pulse functions (Balakumar and Murugesan, 2013).
Farshid and Ali established a numerical solution of nonlinear Volterra-Fredholm integral equations by using hybrid of Block-pulse functions and Taylor series (Farshid Mirzae and Ali Akbar hoseini, 2013).
Balakumar and Murugesan developed a numerical solution of Volterra integral algebraic equations by using Block pulse functions(Balakumar and Murugesan K, 2015) In this paper, we present a hybrid of Taylor and Block Pulse Functions (HTBPFs) method for the computation of numerical solution of system of linear Volterra integral equations (SLVIEs) of the second kind of the following form(Balakumar and Murugesan, 2013).

$$\mathbf{F(t)} = \mathbf{G(t)} + \int_{0}^{t} \mathbf{K(t,s)F(s)}ds, 0 \le t \le 1 \quad (1)$$

where

$\mathbf{F(t)} = [f_1(t) f_2(t) \cdots f_n(t)]^T$

$\mathbf{G(t)} = [g_1(t) g_2(t) \cdots g_n(t)]^T, \mathbf{K(t,s)} = [k_{pq}(t,s)]$

$p, q = 1, 2, \ldots, n$

This Paper has been organized as follows; In section 2, the definition of some basic terms and some mathematical preliminaries which are relevant to the derivation of the hybrid Taylor and Block pulse functionHTBPFs are given . In section 3, we derive the numerical scheme of the proposed method and, in section 4, the efficiency of the proposed method is showned using some numerical examples and finally in section 5, the conclusion is presented.

Hybrid Functions

In this section, the concept of the hybrid method is presented as follows

Block-pulse functions(BPFs)

A set of Block pulse-function $b_i(t)$,$i = 1, 2, ..., N$ on the interval [0,1) is defined as follows (Jung and Schanfelberger, 1992).

$$b_i(t) = \begin{cases} 1, & \frac{i-1}{N} \le t < \frac{i}{N} \\ 0, & otherwise \end{cases} \quad (2)$$

The preceding definition shows that the interval $[0, 1)$ is divided into N equidistant subintervals and the ith block pulse function $b_i(t)$, $i = 1, 2, .., N$, has only one rectangular pulse of unit height in the ith subinterval $\frac{i-1}{N} \le t < \frac{i}{N}$. Then, i is called the order of block-pulse functions.

The following are the properties of BPFs

1.**Disjointness.** The BPFs are disjoint with each other; i.e.,

$$b_i(t)b_j(t) = \begin{cases} b_i(t), & i = j \\ 0, & i \ne j \end{cases} \quad (3)$$

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

2.**Orthogonality.** The BPFs are orthogonal with each other; i.e.,

$$\int_0^1 b_i(t)b_j(t)dt = \begin{cases} h, & i = j \\ 0, & otherwise, \end{cases} \quad (4)$$

in the region of $t\epsilon$ [0,1) where $i, j = 0,1,...,m$.

$$\int_0^1 f^2(t)dt = \sum_{i=0}^{\infty} f_i^2 \|b_i(t)\|^2 \quad (5)$$

$$where f_i = \frac{1}{h}\int_0^1 f(t)b_i(t)dt$$

Hybrid Taylor-Block-pulse functions(HTBPFs)

Consider the Taylor polynomials $T_m(t)=t^m$ on the interval [0 1).For m=0 1 2 ...,M-1 and n=0,1,2,. ,N , the Hybrid Taylor Block pulse function is defined as (Marzban and Razzaghi, 2005).

3.**Completeness.** For every $f\epsilon L^2([0,1))$ when m goes to infinity. Parseval identity holds:

$$b_{nm}(t) = \begin{cases} T_m(Nt - (n - 1)), & \frac{n-1}{N} \le t < \frac{n}{N} \\ 0, & otherwise \end{cases} \quad (6)$$

where n and m are the order of block-pulse functions and Taylor polynomials respectively.

Function Approximation

A function $f\epsilon\ L^2[0, 1)$ can be approximated as

$$f(t) \simeq \sum_{n=1}^{N}\sum_{m=0}^{M-1} f(n, m)b(n, m, t) = F^T B(t), \quad (7)$$

where
B(t)=$[b_{10}(t), ..., b_{1(M-1)}(t), b_{20}(t), ..., b_{2(M-1)}(t), ..., b_{N0}(t), ..., b_{N(M-1)}(t)]^T$, and $f(n,m)$ is defined as follows(Jung and Schanfelberger, 1992).

$$f(n, m) = \frac{1}{N^M m!}\left(\frac{d^m f(t)}{dt^m}\right)\Big|_{t=\frac{n-1}{N}} \quad (8)$$

for $n = 1, 2, ...\ N$ and $m = 0, 1, ..., M - 1$

We can also approximate the function $k(t, s)\epsilon L^2([0, 1] \times [0, 1])$ as follows :

$$k(t, s) \simeq B^T(t)kB(s), \quad (9)$$

where $B(t)$ and $B(s)$ are $NM\times NM$ dimensional vectors respectively and k is an $NM \times NM$ matrix defined as follows:

$$k_{nm} = \frac{1}{N^{u+v}u!v!}\left(\frac{d^{n+m}k(t,s)}{dt^n ds^m}\right)\Big|_{(t,s)=(\frac{n}{N},\frac{m}{N})} \quad (10)$$

for $n, m = 0, 1, ..., MN - 1, u = n - [\frac{n}{N}]N, v = m - [\frac{m}{N}]N$

Operational Matrix

If $B(t) = [b_{10}(t), .., b_{1(M-1)}(t), b_{20}(t), ..., b_{2(M-1)}(t), ..., b_{N0}(t), ..., b_{N(M-1)}(t)]^T$, be the vector function of Hybrid Taylor and Block pulse functions on $[0, 1)$, the integration of this vector $B(t)$ follows :

$$\int_0^t B(t')dt \simeq PB(t), \quad (11)$$

where P is an $NM \times NM$ matrix, that is called the operation matrix of Hybrid Taylor and Block-pulse functions(Jung and Schanfelberger, 1992).

Suppose that $E_i, i = 1, 2, ..., M$ be the operation matrix of Taylor polynomials on i^{th} sub interval$[\frac{i-1}{N}, \frac{i}{N}]$, then the operation matrix P has the following form (Razzaghi and Arabshahi, 1989).

$$P = \begin{bmatrix} E_1 & H_{12} & \dots & H_{1N} \\ 0 & E_2 & \dots & H_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & E_N \end{bmatrix}_{(NMXNM)}$$

where H_{1j} is an $M \times M$ matrix and is defined as follows(Razzaghi and Arabshahi, 1989).

$$H_{1j} = \frac{1}{N}\begin{bmatrix} 1 & 0 & \dots & 0 \\ \frac{1}{2} & 0 & & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} & o & \dots & 0 \end{bmatrix}_{(MXM)}$$

also E_i on the i^{th} interval is defined as follows :

$$E_i = \frac{1}{N}\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{2} & \dots & 0 \\ & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \frac{1}{M-1} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}_{(MXM)}$$

The Product Operation Matrix

The following property of the product of two Hybrid Taylor and Block pulse vector functions will also be used :
$B(t)B^T(t)C \simeq \widetilde{C}^T B(t)$,
where C is a given NM dimensional column vector and $\widetilde{C}$ is an $NM \times NM$ diagonal matrix with its diagonal entries equal to the NM dimensional column vector C such that:

$$\widetilde{C} = \begin{bmatrix} \widetilde{C}_1 & 0 & \dots & 0 \\ 0 & \widetilde{C}_2 & \dots & 0 \\ \vdots & \cdot & & \cdot \\ 0 & 0 & & C_N \end{bmatrix}$$

and
$\widetilde{C}_1 = \widetilde{C}_2 = \dots = \widetilde{C}_N$(Razzaghi and Arabshahi,1989).

Derivation of (HTBPFs) Method

In this section, we derive the proposed method for solving system of linear volterra integral equations The method is based on approximation.
We note that : For $B(t) = [b_1(t), b_2(t), \dots b_N(t)]^T$, then

$$B(t)B^T(t) = diag(B(t)) \qquad (12)$$

where diag($B(t)$) is a diagonal matrix with its diagonal entries equal to $B(t)$.
Let V be an m vector and $V = diag(V)$. Then;

$$B(t)B^T(t)V = \widetilde{V}B(t) \qquad (13)$$

for every mxm matrix A,

$$B^T(t)AB(t) = \hat{A}^T B(t) \qquad (14)$$

where $\hat{A}$ is an m vector with elements equal to the diagonal enteries of matrix A

HTBPFs method for system of linear volterra integral equations

Consider the following SLVIEs, of the second kind as below;

$$F(t) = G(t) + \int_0^t K(t,s)F(s)ds, 0 \le t \le 1 \qquad (15)$$

where $F(t) = [f_1(t)\ f_2(t) \quad f_{N_1}(t)]^T$,
$G(t) = [g_1(t)\ g_2(t)\ \cdot \quad g_{N_1}(t)]^T$,
$K(t,s) = [k_{pq}(t,s)], p = 1, 2, \dots, N_1, q = 1\ 2\ \dots, M_1$
The system can also be written as:

$$f_p(t) = g_p(t) + \sum_{q=1}^{M_1}\int_0^t k_{pq}(t,s)F_q ds \quad 0 \le t \le 1 \qquad (16)$$

where $p = 1, 2, \dots N_1$

HTBPFs expansion for the functions f_p, f_q, g_p, k_{pq} can be written as:
$f_p(t) \simeq F_p^T B(t) = B^T(t)F_p$
$f_q(s) \simeq F_q^T B(s) = B^T(s)F_q$
$g_p(t) \simeq G_p^T B(t) = B^T(t)G_p$
$k_{pq} \simeq B^T(t)k_{pq}B(s)$

where $F_p = [f_{p10}, \dots, f_{p1(M-1)}, f_{p20}, \dots, f_{p2(M-1)}, \dots f_{pN0}, \dots, f_{pN(M-1)}]^T$,

$F_q = [f_{q10} \dots, f_{q1(M-1)}, f_{q20}, \dots, f_{q2(M-1)} \dots, f_{qN0}, \dots f_{qN(M-1)}]^T$

$G_p = [g_{p10}, \dots g_{p1(M-1)}, g_{p20} \dots g_{p2(M-1)}, \dots, g_{pN0}, \dots, g_{pN(M-1)}]^T$

Now the P^{th} equation in 16 above can be expanded in NM terms HTBPFs expansion as follows

$$\begin{aligned} F_p^T B(t) &\simeq G_p^T B(t) + \sum_{q=1}^{M_1}\int_0^t B^T(t)k_{pq}B(s)B^T(s)F_q ds \\ &= G_p^T B(t) + B^T(t)\sum_{q=1}^{M_1} K_{pq}\int_0^t B(s)B^T(s)F_q ds \end{aligned} \qquad (17)$$

where $p = 1, 2, \dots, N_1$
Using 13 gives

$$\begin{aligned} F_p^T B(t) &\simeq G_p^T B(t) + B^T(t)\sum_{q=1}^{M_1} k_{pq}\int_0^t \widetilde{F}_q B(s)ds \\ &= G_p^T B(t) + B^T(t)\sum_{q=1}^{M1} K_{pq}\widetilde{F}_q\int_0^t B(s)ds \end{aligned} \qquad (18)$$

where $\widetilde{F}_q = diag(F_q)$ Using 11 gives

$$F_p^T B(t) \simeq G_p^T B(t) + B^T(t) \sum_{q=1}^{M_1} k_{pq}\widetilde{F}_q P B(t) \quad (19)$$

$$= G_p^T B(t) + B^T(t)[Kp_1F_1P + Kp_2F_2P + \ldots + Kp_nF_nP]B(t) \quad (20)$$

Let $A_p = [Kp_1F_1P + Kp_2F_2P + .. + Kp_nF_nP]$

Then,

$$F_p^T B(t) \simeq G_p^T B(t) + B^T(t)A_pB(t) \quad (21)$$

Using 14) gives,

$$F_p^T B(t) \simeq G_p^T B(t) + \hat{A}_p^T B(t) \quad (22)$$

$$F_p \simeq G_p + \hat{A}_p \quad (23)$$

where A_p is an NM-vector with components equal to the diagonal entries of the $NM \times NM$ matrix $\hat{A}_p = \sum_{q=1}^{M1} k_{pq}\widetilde{F}_q P$, and we have that $\widetilde{F}_q = diag(f_q)$

i.e, since $f_q = [f_{q10}, \ldots, f_{q1(M-1)}, f_{q20}, .., f_{q2(M-1)}, \ldots, f_{qN0}, \ldots, f_{qN(M-1)}]^T$, then

$$\widetilde{F}_q = \begin{bmatrix} f_{q10} & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & f_{q11} & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f_{q1(M-1)} & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & f_{q20} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & f_{q21} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & f_{qN0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & \cdots & f_{qN(M-1)} \end{bmatrix}_{(NM\,X\,NM)}$$

Now $\hat{A}_p$ can be computed as follows:

$$\begin{aligned} \sum_{q=1}^{M_1} k_{pq}F_qP &= [Kp_1\widetilde{F}_1P + Kp_2\widetilde{F}_2P + \ldots + Kp_{M_1}\widetilde{F}_{M_1}P] \\ &= [Kp_1\widetilde{F}_1 + Kp_2\widetilde{F}_2 + \ldots + Kp_{M_1}\widetilde{F}_{M_1}]P \end{aligned} \quad (24)$$

So,

$$A_p = \begin{bmatrix} \sum_{q=1}^{M_1} K_{pq}f_{q10} & .. & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & .. & \sum_{q=1}^{M_1} K_{pq}f_{q1(M-1)} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & .. & 0 & \cdots & \sum_{q=1}^{M_1} K_{pq}f_{qN(M-1)} \end{bmatrix} \begin{bmatrix} E_1 & H_{12} & \ldots & H_{1N} \\ 0 & E_2 & \ldots & H_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & E_N \end{bmatrix} \quad (25)$$

Multiplying the above matrices and then taking the diagonal entries gives us $\hat{A}_p$,which is an NM vector:

So

$$\hat{A}_p = \begin{bmatrix} \frac{1}{N}\sum_{q=1}^{M_1} K_{pq} f_{q10} \\ \frac{1}{2N}\sum_{q=1}^{M_1} K_{pq} f_{q11} \\ \vdots \\ \frac{1}{N(M-1)}\sum_{q=1}^{M_1} K_{pq} f_{q1(M-1)} \\ \frac{1}{N}\sum_{q=1}^{M_1} K_{pq} f_{q20} \\ \frac{1}{2N}\sum_{q=1}^{M_1} K_{pq} f_{q21} \\ \vdots \\ \frac{1}{N(M-1)}\sum_{q=1}^{M_1} K_{pq} f_{q2(M-1)} \\ \vdots \\ \frac{1}{N}\sum_{q=1}^{M_1} K_{pq} f_{qN0} \\ \vdots \\ \frac{1}{N(M-1)}\sum_{q=1}^{M_1} K_{pq} f_{qN(M-1)} \end{bmatrix} \tag{26}$$

using 26 in 16 and replacing $\simeq$ with $=$ yields,

$$\begin{bmatrix} f_{p10} \\ f_{p11} \\ \vdots \\ f_{p1(M-1)} \\ f_{p20} \\ f_{p21} \\ \vdots \\ f_{p2(M-1)} \\ \vdots \\ f_{pN0} \\ \vdots \\ f_{pN(M-1)} \end{bmatrix} = \begin{bmatrix} g_{p10} \\ g_{p11} \\ \vdots \\ g_{p1(M-1)} \\ g_{p20} \\ g_{p21} \\ \vdots \\ g_{p2(M-1)} \\ \vdots \\ g_{pN0} \\ \vdots \\ g_{pN(M-1)} \end{bmatrix} + \begin{bmatrix} \frac{1}{N}\sum_{q=1}^{M_1} K_{pq} f_{q10} \\ \frac{1}{2N}\sum_{q=1}^{M_1} K_{pq} f_{q11} \\ \vdots \\ \frac{1}{N(M-1)}\sum_{q=1}^{M_1} K_{pq} f_{q1(M-1)} \\ \frac{1}{N}\sum_{q=1}^{M_1} K_{pq} f_{q20} \\ \frac{1}{2N}\sum_{q=1}^{M_1} K_{pq} f_{q21} \\ \vdots \\ \frac{1}{N(M-1)}\sum_{q=1}^{M_1} K_{pq} f_{q2(M-1)} \\ \vdots \\ \frac{1}{N}\sum_{q=1}^{M_1} K_{pq} f_{qN0} \\ \vdots \\ \frac{1}{N(M-1)}\sum_{q=1}^{M_1} K_{pq} f_{qN(M-1)} \end{bmatrix}$$

The system above gives the HTBPFs coefficients recursively. Using this coefficients with $f(t) = [f^{(1)}\ f^{(2)}\ \cdots f^{(m)}]B(t)$ numerical solutions can be easily computed

Numerical Results

In this section, we have compared the performance of our proposed method which is the hybrid of Taylor and Block pulse function method with the method of Block pulse function(BPF).

A suitable value of N plays a significant role in transforming the original problem into an NM-system of equations. So, the appropriate value of N allows one to easily determine where the location of the exact solution might be. After choosing the appropriate value of N, small values of M is required to get some good accuracy. The Tables below displays the results of the proposed method. We denote the computed numerical values of the method by f_i^*, the exact solutions by f_i, and maximum absolute errors by $e_i(t) = \max\{|f_i^* - f_i|, 0 \le t \le 0.5\}, i = 1, 2, ..., n,$

Example 1: *consider the system of volterra equations:* (Balakumar and Murugesan, 2013).

$$\begin{cases} f_1(t) = t + \frac{t^3}{2} + \frac{t^4}{12} - \frac{t^5}{5} + \int_0^t (s^2 - t)(f_1(s) + f_2(s))ds \\ f_2(t) = t^2 \quad \frac{t^3}{3} + \frac{t^4}{4} + \int_0^t s(f_1(s) + f_2(s))ds, \end{cases} \tag{27}$$

with exact solutions $f_1(t) = t$ and $f_2(t) = t^2$

Table 1: **The Numerical Result for f_1^* at N=11 ,M=1 and f_2^* at N=3 ,M=3 in Example 1(HTBPFs)**

t	f_1^*	f_1	$\|f_1^* - f_1\|$	f_2^*	f_2	$\|f_2^* - f_2\|$
0.0	0.0032	0.0000	3.2947 x 10^{-3}	0.0000	0.0000	0.0 00
0.1	0.0912	0.1000	8.738 x 10^{-3}	0.0099	0.0100	1.0000 x 10^{-10}
0.2	0.1848	0.2000	1.5125 x 10^{-2}	0.0400	0.0400	0.0000
0.3	0.2830	0.3000	1.6971 x 10^{-2}	0.0899	0.0900	1.0000 x 10^{-9}
0.4	0.3878	0.4000	1.2136 x 10^{-2}	0.1324	0.1600	2.7531 x 10^{-2}
0.5	0.5011	0.5000	1.1792 x 10^{-3}	0.1959	0.2500	5.4012 x 10^{-2}

The maximum absolute errors are: $e_1(t)$ = **1.6971 x 10^{-2} and** $e_2(t)$ = **5.4012 x 10^{-2}**

Table 2: **Numerical Result for f_1^{**} and f_2^{**} in Example 1(BPFs)**

	t	f_1^{**}	f_1	$\|f_1^{**} - f_1\|$	f_2^{**}	f_2	$f_2^{**} - f_2\|$
	0.0	0.0156	0.0000	1.5600 x 10^{-2}	0.0003	0.0000	3.0000 x 10^{-4}
	0.1	0.1094	0.1000	9.4000 x 10^{-3}	0.0121	0.0100	2.1000 x 10^{-3}
N=32	0.2	0.2031	0.2000	3.1000 x 10^{-3}	0.0414	0.0400	1.4000 x 10^{-3}
	0.3	0.2969	0.3000	3.1000 x 10^{-3}	0.0883	0.0900	1.7000 x 10^{-3}
	0.4	0.3906	0.4000	9.4000 x 10^{-3}	0.1527	0.1600	7.3000 x 10^{-3}
	0.5	0.5057	0.5000	5.7000 x 10^{-3}	0.2660	0.2500	1.6000 x 10^{-2}

$e_1(t)$ = **1.5600 x 10^{-2} and** $e_2(t)$ = **1.6000 x 10^{-2}**

Example 2: *consider the system of volterra equations:* (Balakumar and Murugesan, 2013).

$$\begin{cases} f_1(t) = 1 - \frac{t^2}{2} + \int_0^t (f_1(s) + se^s f_2(s))ds \\ f_2(t) = 1 + \frac{t^2}{2} + \int_0^t (-se^{-s} f_1(s) - f_2(s))ds, \end{cases} \tag{28}$$

with exact solutions $f_1(t) = e^t$ and $f_2(t) = e^{-t}$

Table 3 **The Numerical Result for f_1^* at N=8 ,M=1 and f_2^* at N=3 ,M=1 in Example 2(HTBPFs)**

t	f_1^*	f_1	$\|f_1^* - f_1\|$	f_2^*	f_2	$\|f_2^* - f_2\|$
0 0	1 2769	1 0000	2 7695 x 10^{-1}	0 7500	1 0000	2 5000 x 10^{-1}
0 1	1 2769	1 1051	1 718 x 10^{-1}	0 7500	0 9048	1 5484 x 10^{-1}
0.2	0.9922	1 2214	2.2922 x 10^{-1}	0.7500	0.8187	6.8731 x 10^{-2}
0.3	0.9688	1 3499	3.8111 x 10^{-1}	0.7500	0.7408	9.1818 x 10^{-3}
0.4	0.9297	1 4918	5.6214 x 10^{-1}	1.0556	0.6703	3.8524 x 10^{-1}
0.5	0.8750	1 6487	7.7372 x 10^{-1}	1.0556	0.6065	4.4902 x 10^{-1}

The maximum absolute errors are: $e_1(t)$ = **7.7372 x 10^{-1} and** $e_2(t)$ = **4.4902 x 10^{-1}**

Table 4: **Numerical Result for f_1^{**} and f_2^{**}in Example 2(BPFs)**

	t	f_1^{**}	f_1	$\mid f_1^{**} - f_1 \mid$	f_2^{**}	f_2	$\mid f_2^{**} - f_2\mid$
N=32	0.0	1.0160	1.0000	1.6000 x 10^{-2}	0.9848	1.0000	1.5200 x 10^{-2}
	0.1	1.1158	1.1051	1.0600 x 10^{-2}	0.8964	0.9048	8.4000 x 10^{-3}
	0.2	1.2255	1.2214	4.1000 x 10^{-3}	0.8162	0.8187	2.5000 x 10^{-3}
	0.3	1.3460	1.3499	3.9000 x 10^{-3}	0.7432	0.7408	2.4000 x 10^{-3}
	0.4	1.4783	1.4918	1.3500 x 10^{-2}	0.6767	0.6703	6.4000 x 10^{-3}
	0.5	1.6752	1.6487	2.6500 x 10^{-2}	0.5971	0.6065	9.4000 x 10^{-3}

$e_1(t)$ = **2.6500 x 10^{-2} and** $e_2(t)$ = **1.5200 x 10^{-2}**

Example 3: *consider the system of volterra equations:* (Balakumar and Murugesan, 2013).

$$\begin{cases} f_1(t) = 1 + t^2 - \frac{t^3}{3} - \frac{t^4}{3} + \int_0^t ((t-s)^3 f_1(s) + (t-s)^2 f_2(s))ds \\ f_2(t) = 1 - t - t^3 - \frac{t^4}{4} - \frac{t^5}{4} - \frac{t^7}{420} + \int_0^t ((t-s)^4 f_1(s) + (t-s)^3 f_2(s))ds, \end{cases} \quad (29)$$

with exact solutions $f_1(t) = 1 + t^2$ and $f_2(t) = 1 + t - t^3$

Table 5: **The Numerical Result for f_1^* at N=5 ,M=5 andf_2^* at N=2 ,M=1 in Example 3(HTBPF)**

t	f_1^*	f_1	$\mid f_1^* - f_1 \mid$	f_2^*	f_2	$\mid f_2^* - f_2 \mid$
0.0	1.0001	1.0000	1.0000 x 10^{-4}	1.1321	1.0000	1.3211 x 10^{-1}
0.1	1.0164	1.0100	6.4065 x 10^{-3}	1.1321	1.0990	3.3109 x 10^{-2}
0.2	1.0368	1.0400	3.2000 x 10^{-3}	1.1321	1.1920	5.9891 x 10^{-2}
0.3	1.0768	1.0900	1.3200 x 10^{-2}	1.1321	1.2730	1.4089 x 10^{-1}
0.4	1.1301	1.1600	2.9867 x 10^{-2}	1.1321	1.3360	2.0389 x 10^{-1}
0.5	1.1875	1.2500	6.2500 x 10^{-2}	0.5284	1.3750	8.4656 x 10^{-1}

The maximum absolute errors are: $e_1(t)$ = **6.2500 x 10^{-2} and** $e_2(t)$ = **8.4656 x 10^{-1}**

Table 6: **Numerical Result for f_1^{**} and f_2^{**}in Example 3(BPFs)**

N=32

t	f_1^*	f_1	$\mid f_1^* - f_1 \mid$	f_2^*	f_2	$\mid f_2^* - f_2 \mid$
0.0	1.0003	1.0000	3.0000 x 10^{-4}	1.0156	1.0000	1.5600 x 10^{-2}
0.1	1.0120	1.0100	2.0000 x 10^{-3}	1.1080	1.0990	9.0000 x 10^{-3}
0.2	1.0413	1.0400	1.3000 x 10^{-3}	1.1947	1.1920	2.7000 x 10^{-3}
0.3	1.0882	1.0900	1.8000 x 10^{-3}	1.2706	1.2730	2.4000 x 10^{-3}
0.4	1.1527	1.1600	7.3000 x 10^{-3}	1.3309	1.3360	5.1000 x 10^{-3}
0.5	1.2660	1.2500	1.6000 x 10^{-2}	1.3784	1.3750	3.4000 x 10^{-3}

$e_1(t)$ =**1.6000 x 10^{-2} and** $e_2(t)$ = **1.5600 x 10^{-2}**

Conclusion

In this paper, we presented the method for solving system of linear volterra integral equations of the second kind using the interval $\{0 \leq t \leq 0.5\}$ for a suitable N and M . The results obtained show that the proposed method have a good degree of accuracy for solving system of linear volterra integral equations.

References

Babolian, E. and Masouri, Z. (2008). Direct method to solve Volterra integral equation of the first kind using operational matrix with block pulse function. *journal of computational and Applied Mathematics,* ***220:***51-57.

Babolian, E., Maleknajad, K., Mordad, M. and Rahimi B. (2011). A numerical method for solving Fredholm Volterra integral equations in two dimensional spaces using block pulse function and an operational matrix. *journal of computational and Applied Mathematics,* ***235:***3965-3971.

Balakumar, V. and Murugesan, K. (2013). Numerical solution of system of linear Volterra intregral equations using block pulse functions. *Malaya Journal of Matematik,* ***5****(1):77-84.*

Balakumar, V. and Murugesan, K. (2015). Numerical solution of Volterra integral algebraic equations using block pulse functions. *Applied Mathematics and Computation,* ***263:*** *165-170.*

Beauchamp, K.G. (1984). Application of walsh and related functions with an introduction to sequence theory. *Accademic Press London*.

Farshid Mirzae, Ali Akbar Hoseini. (2013). Numerical solution of nonlinear Volterra-Fredholm integral equations using hybrid of block pulse functions and Taylor series. *Alexandria Engineering Journal,**152***:551-555.

Jung, Z.H. and Schanfelberger, W. (1992). Block pulse functions and their applications in control systems. *springer Verlag Berlin Heidelberg.*

Leyla Rahmani, Bijan Rahimi, Mohammad Mordad. (2011). Numerical solution of Volterra-Fredholm integro Differential equation by Block pulse functions and Operational Matrices. *General Mathematics Notes.* ***4***(2):37-48.

Linz, P. (1985). Analytical and Numerical methods for Volterra equations. *Society for industrial and Applied Mathematics.* ***7.***

Maleknajad, K. and Mahmoudi, Y. (2004). Numerical solution of Linear Fredholm integral equation by using hybrid Taylor and Block pulse functions. *Applied Mathematics and computation* ***149:***799-806.

Maleknajad, K., Shahrazee, M. and Khatami, H. (2005). Numerical solution of integral equations system of the second kind by Block pulse functions. *Journal of computational and Applied Mathematics.* ***166:*** 15-24.

Marzban, H.R. and Razzaghi, M. (2005). Analysis of time delay systems via hybrid of block pulse functions and Taylor series. *j. Vib control.* ***17:***1455-1468.

Razzaghi, M. and Arabshahi, A. (1989). Optimal control of linear distributed parameter system via polynomial series *International Journal of systems science.* ***20:***1141-1148.