Available online at http://ijim.srbiau.ac.ir/
Int. J. Industrial Mathematics (ISSN 2008-5621)
Vol. 10, No. 1, 2018 Article ID IJIM-001038, 9 pages
Research Article

A Solution For Volterra Integral Equations of the First Kind Based on Bernstein Polynomials

M. Mohamadi *, E. Babolian †‡, S. A. Yousefi §

Received Date: 2017-03-03 Revised Date: 2017-07-08 Accepted Date: 2017-10-22

Abstract

In this paper, we present a new computational method to solve Volterra integral equations of the first kind based on Bernstein polynomials. In this method, using operational matrices turn the integral equation into a system of equations. The computed operational matrices are exact and new. The comparisons show this method is acceptable. Moreover, the stability of the proposed method is studied.

Keywords : Volterra integral equation; Bernstein polynomials; Operational matrices; Transformation matrices.

1 Introduction

Integral equations are an important topic in mathematics, physics and engineering sciences. Many researchers spent their time to find a solution for these equations. Their efforts led to many numerical and analytical methods like Neumann series, Nystrom method, expansion method, collocation methods, residual methods, Galerkin methods, homotopy methods, perturbation method, the variational iteration method, the Laplace transform method, the Adomian decomposition method, the series solution method, and the direct computation methods [9, 10, 11, 1, 8, 27, 33]. Recently polynomials play a fundamental role in some valid numerical methods. In some of these approaches integral equations convert to a linear or nonlinear system and by solving the system the approximate solution of the integral equation will be found. Malek nejad et al. [17] and Mandal and Bhattacharya used Bernstein polynomials in approximation techniques [19], Shahsavaran employed Block Pulse functions and Taylor Expansion method [29]. Taylor polynomials were also used by Bellour and Rawashdeh [7] and Wang [32] with computer algebra. These polynomials have been also used for solving Fredholm integral equations of the second kind by Shirin and Islam [30]. Babolian and Delves have described an augmented Galerkin technique for the numerical solution of the first kind Fredholm integral equations [2]. In [12], a numerical solution of Fredholm integral equations of the first kind via piecewise interpolation is proposed. Lewis studied a computational method to solve first kind integral equations[16], also, for more researches see [28, 13, 24, 31, 25, 3, 4, 5, 34, 35, 18, 36]

*Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran.
†Corresponding author. babolian@khu.ac.ir, Tel: +98(21)88329220
‡Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran.
§Department of Mathematics, Shahid Beheshti University, G. C. Tehran, Iran.

In the next section, we review Bernstein polynomials and some basic theorems and concepts. In section 3, transformation matrices are defined. In subsections of section 4, operational matrices are computed. These matrices are new and exact. In section 5, the integral equations are changed to a linear or nonlinear system. Some illustrative examples, in section 6, show accuracy and exactness of method. Then, a comparison between our method with a direct method and an expansion-iterative method is presented in section 7. Finally, in section 8, effect of a random noise on data function is investigated.

2 Preliminaries

Definition 2.1 *Suppose m is a positive integer number, BPs of degree m on interval $[a,b]$ are defined as follows:*

$$B_{i,m}(x)=\binom{m}{i}\frac{(x-a)^i(b-x)^{m-i}}{(b-a)^m},\quad 0\leq i\leq m.$$

Also, $B_{i,m}(x)=0$ if $i<0$ or $i>m$. For convenience we consider $[a,b]=[0,1]$, namely $B_{i,m}(x)=\binom{m}{i}x^i(1-x)^{m-i}$, $\quad 0\leq i\leq m$.

We denote Φ_m , an m-column vector as follows: $\Phi_m(x)=\begin{bmatrix}\phi_0(x) & \phi_1(x) & \cdots & \phi_m(x)\end{bmatrix}^T$, where $\phi_i(x)=B_{i,m}(x),\quad 0\leq i\leq m$.

The BPs have many interesting properties [14, 26, 20, 22]. But, here some of them that are useful in our work are stated.

$P1) B_{i,m}(x)B_{j,m}(x)=\frac{\binom{m}{i}\binom{m}{j}}{\binom{2m}{i+j}}B_{i+j,2m}(x), 0\leq i,j\leq m.$

$$P2)\frac{B_{i,m}}{\binom{m}{i}}=\frac{B_{i,m+1}}{\binom{m+1}{i}}+\frac{B_{i+1,m+1}}{\binom{m+1}{i+1}}, i=0,...,m.$$

The following theorems are a fundamental tool that justifies the use of polynomials.

Theorem 2.1 *[15].Suppose $H=l^2([a,b])$ is a Hilbert space with the inner product defined by $\langle f,g\rangle=\int_a^b f(t)g(t)dt$ and also, $Y=Span\{B_{0,m}(x),B_{1,m}(x),...,B_{m,m}(x)\}$ be the span space by Bernsteins polynomials of degree m . Let f be an arbitrary element in H . Since Y is a finite dimensional and closed subspace, it is a complete subset of H . So, f has the unique best approximation out of Y such that y_o*

$$\exists y_0\in Y;\forall y\in Y:\parallel f-y_0\parallel_2\leqslant\parallel f-y\parallel_2 .$$

Therefore, there are the unique coefficients $\alpha_j, 0\leq j\leq m.$ such that

$$f(t)\approx y_0(t)=\sum_{j=0}^{m}\alpha_jB_{j,m}(t)=\alpha^T.\Phi_m$$

where,$\alpha=\begin{bmatrix}\alpha_0 & \alpha_1 & \cdots & \alpha_m\end{bmatrix}^T$, can be obtained by

$$\alpha=\frac{\langle f(t),\Phi_m(t)\rangle}{\langle\Phi_m(t),\Phi_m(t)\rangle}$$

such that $\langle f,\Phi_m(t)\rangle=\int_a^b f(t)\Phi_m(t)dt$. In the above theorem we denote $Q=\langle\Phi_m(t),\Phi_m(t)\rangle$ as dual matrix. Furthermore, it is easy to see $Q_{i,j}=\frac{\binom{m}{i-1}\binom{m}{j-1}}{(2m+1)\binom{2m}{i+j-2}}, i,j=1,...,m+1.$

Next theorem indicates dual matrix is symmetric and invertible.

Theorem 2.2 *[15]. Elements $y_0,y_1,...,y_n$ of a Hilbert space H constitute a linearly independent set in H if and only if $G(y_0,y_1,...,y_n)\neq 0$.*
Where $G(y_0,y_1,...,y_n)$ is thr Gram determinant of $y_0,y_1,...,y_n$ defined by

$$G(y_0,y_1,...,y_n)=\begin{vmatrix}\langle y_1,y_1\rangle & \langle y_1,y_2\rangle & \cdots & \langle y_1,y_n\rangle\\ \langle y_2,y_1\rangle & \langle y_2,y_2\rangle & \cdots & \langle y_2,y_n\rangle\\ \vdots & \vdots & \ddots & \vdots\\ \langle y_n,y_1\rangle & \langle y_n,y_2\rangle & \cdots & \langle y_n,y_n\rangle\end{vmatrix}.$$

For a 2-dimensional function $k(x,t)\in l^2([0,1]\times[0,1])$, it can be similarly expanded with respect to BPs such as $k(x,t)\simeq\Phi^T(t)K\Phi(x)$, and K is the BP coefficient matrix with $K_{i,j}=Q^{-1}(\int_0^1(Q^{-1}\int_0^1 k(x,t)\phi_i(t)dt)\phi_j(x)dx), 0\leqslant i,j\leqslant m.$

3 Transformation matrices

Transformation matrix is used to change the dimension of the problem. In other words, these matrix can convert Φ_m to Φ_n and vice versa. Suppose m is less than n ,T_m^n is an $(m+1)\times(n+1)$ matrix , called increasing transformation matrix, that converts Φ_m to Φ_n . In other words, $\Phi_m=T_m^n\Phi_n$. The increasing transformation matrix can be computed as follows:

$$[T_m^n]_{i,j}=\begin{cases}0 & \text{if } i<j \text{ or } j>i+k\\ \frac{\binom{m}{i-1}\binom{k}{j-1}}{\binom{m+k}{i+j-2}} & otherwise\end{cases}$$

It is sufficient to use $P2$, k times where $k = n-m$. Also, decreasing transformation matrix is an $(n+1)\times(m+1)$matrix which is shown by T_n^m and converts Φ_n to Φ_m where n is greater than m. In other weords, $\Phi_n = T_n^m\Phi_m$. The i th row of decreasing transformation matrix can be calculated as follows:

$$\frac{1}{m+n+1}\left[\frac{1}{\binom{m+n}{i}} \quad \frac{1}{\binom{m+n}{i+1}} \quad \cdots \quad \frac{1}{\binom{m+n}{i+m}}\right] Q^{-1}$$

$, i = 0, ..., n.$

4 Operational matrices

Operational matrix is a matrix that works on basis like an operator, in other words, if Λ is an operator an operational matrix is a matrix like P such that $\Lambda(\Phi) \simeq P\Phi$.

4.1 Operational matrix of integration

Lemma 4.1 *Let M be operational matrix of integration and $\Phi_m(x) = \left[\phi_0(x) \quad \phi_1(x) \quad \cdots \quad \phi_m(x)\right]^T$, then*

$$\int_0^x \Phi_m(x)dx = M\Phi_m(x). \tag{4.1}$$

Proof. With a simple calculation can be seen

$$B_{i,m}(x) = \int_0^x m(B_{i-1,m-1}(t) - B_{i,m-1}(t))dt$$

Assume $0 \le k \le m$,

$$\sum_{i=k}^m B_{i,m}(x) = \sum_{i=k}^m \int_0^x m(B_{i-1,m-1}(t) - B_{i,m-1}(t))dt = m\int_0^x B_{k-1,m-1}(t)dt.$$

Therefore,

$$\int_0^x B_{k,m}(t)dt = \frac{1}{m+1}\sum_{i=k+1}^{m+1} B_{i,m+1}(x) = M_k^T.\Phi_{m+1}$$

where

$$M_k = \frac{1}{m+1}[\overbrace{0,\cdots,0}^{k+1},\overbrace{1,\cdots,1}^{m+k-1}]^T,$$

it is obvious ,$im = \begin{bmatrix} M_0^T \\ M_1^T \\ \vdots \\ M_m^T \end{bmatrix}$ is an $(m+2)\times(m+1)$ matrix. Accordingly $M = imT_{m+1}^m$.

4.2 Operational matrix of of product

Lemma 4.2 *Let C be an $(m+1)\times(m+1)$matrix then,*

$$\Phi_m^T C\Phi_m = \hat{C}^T\Phi_{2m} \tag{4.2}$$

where

$$\hat{C}_k = \sum_{j=0}^k \frac{\binom{m}{k-j}.\binom{m}{j}}{\binom{2m}{k}}.C_{k-j,j} \quad k = 0, ..., 2m.$$

Proof. Let $\phi_i^*(x) = B_{i,2m}(x)$,for $i = 0, ..., 2m$,

$$\Phi_m^T C\Phi_m = \sum_{i=0}^m\sum_{j=0}^m c_{i,j}\phi_i\phi_j$$

using **P1** gives

$$\Phi_m^T C\Phi_m = \sum_{i=0}^m\sum_{j=0}^m \frac{\binom{m}{i}\binom{m}{j}}{\binom{2m}{i+j}}\phi_{i+j}^*(x)$$

$$= \left[\sum_{j=0}^1 \frac{\binom{m}{1-j}\binom{m}{j}}{\binom{2m}{1}}C_{1-j,j}\right.$$

$\cdots$

$$\sum_{j=0}^k \frac{\binom{m}{k-j}\binom{m}{j}}{\binom{2m}{k}}C_{k-j,j}\cdots\frac{\binom{m}{m}\binom{m}{m}}{\binom{2m}{2m}}C_{m,m}$$

Φ^*
$= \hat{}\mathrm{T}\Phi_{2m}(x).$

$$\sum_{j=0}^k \frac{\binom{m}{k-j}\binom{m}{j}}{\binom{2m}{k}}C_{k-j,j}\cdots\frac{\binom{m}{m}\binom{m}{m}}{\binom{2m}{2m}}C_{m,m}$$

Φ^*
$= \hat{C}^T\Phi_{2m}(x).$

Lemma 4.3 *Let u be an arbitrary $(m+1)-$vector then,*

$$\Phi_m\Phi_m^T u = \tilde{u}\Phi_{2m}, \tag{4.3}$$

where $\tilde{u}$ is an $(m+1)\times(2m+1)$matrix with elements

Table 1: Results of example 6.1 in some special points.

x	$m=8$	$e_8(x)$	$m=10$	e˙10(x)	exact solution
0	0.99274777	7.25×10^{-3}	1.0049204	4.92×10^{-3}	1.00
0.1	0.90237509	2.46×10^{-3}	0.90793960	3.10×10^{-3}	0.904837418
0.2	0.82088162	2.15×10^{-3}	0.81635890	2.37×10^{-3}	0.818730753
0.3	0.73878286	2.03×10^{-3}	0.74336086	2.54×10^{-3}	0.740818220
0.4	0.67145627	1.13×10^{-3}	0.66769187	2.63×10^{-3}	0.670320046
0.5	0.60698528	4.54×10^{-4}	0.60840722	1.87×10^{-3}	0.606530659
0.6	0.54718461	1.62×10^{-3}	0.54837489	4.37×10^{-4}	0.548811636
0.7	0.49821291	1.62×10^{-3}	0.49561978	9.65×10^{-4}	0.496585303
0.8	0.448407286	9.21×10^{-4}	0.45118619	1.85×10^{-3}	0.449328964
0.9	0.40743778	8.68×10^{-4}	0.40466694	1.90×10^{-3}	0.406569659

Table 2: Results of example 6.2 in some special points.

x	$m=5$	$e_5(x)$	$m=10$	e˙10(x)	exact solution
0	0.99945089	5.91×10^{-4}	1.0000	0.000	1.00
0.1	0.90500947	1.72×10^{-4}	0.90483741	$1.\times10^{-10}$	0.904837418
0.2	0.81849734	2.33×10^{-4}	0.81873075	$1.\times10^{-10}$	0.818730753
0.3	0.74075709	6.11×10^{-5}	0.74081822	1.5×10^{-11}	0.740818220
0.4	0.67060120	2.81×10^{-4}	0.67060120	3.6×10^{-11}	0.670320046
0.5	0.60669652	1.65×10^{-4}	0.60653065	$2.\times10^{-10}$	0.606530659
0.6	0.54849402	3.17×10^{-4}	0.54881163	$1.\times10^{-10}$	0.548811636
0.7	0.49620362	3.81×10^{-4}	0.49658530	$1.\times10^{-10}$	0.496585303
0.8	0.4498142362	4.85×10^{-4}	0.44932896	1.9×10^{-11}	0.449328964
0.9	0.4071590358	5.89×10^{-4}	0.40656965	$1.\times10^{-10}$	0.406569657

Table 3: Results of example 6.3 in some special points.

x	$m=8$	$e_8(x)$	$m=12$	$e_{12}(x)$	Exact solution
0	-0.032078831	3.20×10^{-2}	0.00008978829	8.97×10^{-5}	0.0000000000
0.1	0.191776504	7.89×10^{-3}	0.1996870087	2.01×10^{-5}	0.1996668333
0.2	0.405304229	7.96×10^{-3}	0.3973191521	1.95×10^{-5}	0.3973386616
0.3	0.581192769	9.84×10^{-3}	0.5910549712	1.45×10^{-5}	0.5910404134
0.4	0.786819169	7.98×10^{-3}	0.7788219798	1.47×10^{-5}	0.7788366846
0.5	0.959541169	6.90×10^{-4}	0.9588688383	1.77×10^{-5}	0.9588510772
0.6	1.115121169	1.41×10^{-2}	1.129263004	2.19×10^{-5}	1.1292849470
0.7	1.316221169	2.77×10^{-2}	1.288461250	2.58×10^{-5}	1.2884353740
0.8	1.391721169	4.29×10^{-2}	1.434688846	2.33×10^{-5}	1.4347121820
0.9	1.637921169	7.12×10^{-2}	1.566617164	3.66×10^{-5}	1.5666538190

Table 4: Results of example 6.4 in some special points.

x	$m=4$	$e_4(x)$	$m=8$	$e_8(x)$	Exact solution
0.00	0.0084887020	8.4×10^{-3}	0.000000046	4.6×10^{-9}	0.000
0.15	0.1478336203	2.1×10^{-3}	0.1500000083	8.3×10^{-9}	0.150
0.30	0.3058088085	5.8×10^{-3}	0.3000000032	3.2×10^{-9}	0.300
0.45	0.4463277275	3.6×10^{-3}	0.4499999969	3.1×10^{-9}	0.450
0.60	0.5904142554	9.5×10^{-3}	0.5999999962	3.8×10^{-9}	0.600
0.75	0.7674230210	1.6×10^{-3}	0.7500000045	4.5×10^{-9}	0.750
0.90	0.8934282466	6.5×10^{-3}	0.9000001447	1.4×10^{-7}	0.900

Table 5: Comparison between BPs method and block-pulse methods in example 6.1.

method	Mid-points, $k = 32$	Mid-points, $k = 64$	Ten points , $k = 32$	Ten points ,$k = 64$
Direct method	3.3×10^{-3}	1.6×10^{-3}	5.9×10^{-3}	2.9×10^{-3}
expansion-iterative rule	6.6×10^{-4}	1.9×10^{-4}	5.2×10^{-3}	2.6×10^{-3}
	$m = 5$	$m = 8$	$m = 10$	
BPs method	7.70×10^{-4}	1.78×10^{-3}	2.62×10^{-3}	

Table 6: Comparison between BPs method and block-pulse methods in example 6.3.

method	Mid-points, $k = 64$	Mid-points, $k = 128$	Ten points , $k = 64$	Ten points ,$k = 128$
Direct method	5.2×10^{-3}	2.6×10^{-3}	8.2×10^{-3}	4.1×10^{-3}
expansion-iterative rule	4.9×10^{-4}	1.4×10^{-4}	6.5×10^{-3}	3.3×10^{-3}
	$m = 5$	$m = 8$	$m = 12$	
BPs method	1.3×10^{-3}	2.5×10^{-4}	3.96×10^{-5}	

Table 7: Effect of noise on example 7.1.

x	$m = 4$	$m = 4, \varepsilon = 0.01$	$m = 4, \varepsilon = 0.02$	$m = 4, \varepsilon = 0.03$	Exact solution
0.0	0.00000	-0.00312608987	-0.00311339313	-0.003136934445	0.0000000
0.1	0.010000	0.01100083307	0.01097076095	0.01103789223	0.0100000
0.2	0.040000	0.03865110727	0.03854449585	0.03877979162	0.0400000
0.3	0.090000	0.08851961494	0.08827616242	0.08881386514	0.0900000
0.4	0.160000	0.15944664250	0.15900825960	0.1599768635	0.1600000
0.5	0.250000	0.24858913690	0.24790549840	0.2494161014	0.2500000
0.6	0.360000	0.35572905650	0.35475073140	0.3569125103	0.3600000
0.7	0.490000	0.48371881310	0.48238874880	0.4853278320	0.4900000
0.8	0.640000	0.63506381300	0.63331794130	0.6371759494	0.6400000
0.9	0.810000	0.80464208650	0.80242983060	0.8073183571	0.8100000
1.0	1.000000	0.96856101440	0.96589645720	0.9717837716	1.0000000

Table 8: Effect of noise on example 7.2.

x	$m = 8$	$m = 8, \varepsilon = 0.01$	$m = 8, \varepsilon = 0.02$	$m = 8, \varepsilon = 0.03$	Exact solution
0.0	4.60×10^{-8}	0.005199858864	0.003101836424	0.0795857226	0.0000
0.1	0.100000018	0.1063241402	0.1042278036	0.1027680108	0.1000
0.2	0.199999982	0.2062665325	0.2041697813	0.2071972880	0.2000
0.3	0.300000003	0.3063655647	0.3042692155	0.3021127705	0.3000
0.4	0.400000025	0.4061393977	0.4040429008	0.4112898448	0.4000
0.5	0.499999963	0.5063383726	0.5042415973	0.5047514740	0.5000
0.6	0.599999995	0.6064074962	0.6043110989	0.6021586937	0.6000
0.7	0.700000068	0.7059853291	0.7038889421	0.7138653497	0.7000
0.8	0.799999897	0.8065711616	0.8044741803	0.7994762500	0.8000
0.9	0.900000144	0.9058954184	0.9037997147	0.9145590776	0.9000

$$\tilde{u}_{i,j} = \begin{cases} 0 & \text{if } j < i \text{ and } j > i + m \\ \frac{\binom{m}{i}\binom{m}{j}}{\binom{2m}{i+j}} u_j & \text{otherwise} \end{cases}$$

for $i, j = 0, ..., m$.

Proof. Property P1 implies

$$\begin{aligned} &\Phi_m(x)\Phi_m^T(x)u \\ &= [\sum_{j=0}^{m} \frac{\binom{m}{0}\binom{m}{j}}{\binom{2m}{j}} u_j \phi_j^* \quad \sum_{j=0}^{m} \frac{\binom{m}{1}\binom{m}{j}}{\binom{2m}{j+1}} u_j \phi_{j+1}^* \\ &\qquad \ldots \quad \sum_{j=0}^{m} \frac{\binom{m}{m}\binom{m}{j}}{\binom{2m}{m+j}} u_j \phi_{m+j}^*]^T. \end{aligned}$$

Now, ith entry of the above matrix can be rewritten as follows:

$$\sum_{j=0}^{m} \frac{\binom{m}{i-1}\binom{m}{j}}{\binom{2m}{j+i-1}} u_j \phi^*_{j+i-1} =$$
$$\left[\overbrace{0\cdots 0}^{i-1} \quad \frac{\binom{m}{i-1}\binom{m}{0}}{\binom{2m}{i-1}} u_0 \quad \cdots \quad \frac{\binom{m}{i-1}\binom{m}{m}}{\binom{2m}{m+i-1}} u_m \quad \overbrace{0\cdots 0}^{m-i+1}\right] \Phi_{2m}$$

5 Solution of integral equation of the first kind

In this section, with respect to operational matrices and function approximation, integral equation converts to a system of equations.

5.1 Linear Volterra integral equation of the first kind

Consider the following Volterra integral equation of the first kind

$$f(x) = \int_0^x k(x,t)u(t)dt \tag{5.4}$$

where f and k are known but u is not. Moreover, $k(x,t) \in l^2([0,1]\times[0,1])$ and $f(t) \in l^2([0,1])$. Approximating functions f, u and $k$with respect to BPs gives

$$\begin{aligned} f(x) &= F^T\Phi_m(x) = \Phi_m^T(x)F \\ u(t) &= U^T\Phi_m(t) = \Phi_m^T(t)U \\ k(x,t) &\simeq \Phi_m^T(x)K\Phi_m(t) \end{aligned} \tag{5.5}$$

where the vectors F, U and matrix K are BPs coefficients of $f(x), u(t)$ and $k(x,t)$respectively. Now, replacing (5.5) into the (5.4) gives:

$$\begin{aligned} F^T\Phi_m(x) &= \int_0^x \Phi_m^T(x)K\Phi_m(t)\Phi_m^T(t)Udt \\ &= \Phi_m^T(x)K\int_0^x \Phi_m(t)\Phi_m^T(t)Udt. \end{aligned}$$

Using (4.3) follows:

$$\begin{aligned} F^T\Phi_m(x) &= \Phi_m^T(x)K\int_0^x \tilde{U}\Phi_{2m}(t)dt \\ &= \Phi_m^T(x)K\tilde{U}\int_0^x \Phi_{2m}(t)dt. \end{aligned} \tag{5.6}$$

Using operational matrix of integration M, in Eq. (5.6) gives:

$$F^T\Phi_m(x) = \Phi_m^T(x)K\tilde{U}M\Phi_{2m}(t)dt. \tag{5.7}$$

Let $U^* = K\tilde{U}MT_{2m}^m$, where U^* is an $(m+1)\times(m+1)$ matrix. Eq. (5.7) changes to:

$$F^T\Phi_m(x) = \Phi_m^T(x)U^*\Phi_m(x). \tag{5.8}$$

Using Eq. (4.2) in (5.8) gives:

$$F^T\Phi_m(x) = \Phi_m^T(x)U^*\Phi_m(x) = \hat{U^*}^T\Phi_{2m}(x).$$

Using decreasing transformation matrixT_{2m}^m , gives the final system:

$$\bar{U} = F,$$

where $\bar{U}^T = \hat{U^*}^T T_{2m}^m$.

5.2 Nonlinear Volterra integral equation of the first kind

Consider the following nonlinear Volterra integral equation

$$f(x) = \int_0^x k(x,t)g(u(t))dt \tag{5.9}$$

Put $w(t) = g(u(t))$ and Subsequently $w(t) = W^T\Phi_m(t)$. Where W is an unknown $(m+1)$- vector. Following the same procedure, final system is as follows: $\bar{W} = F$.Finally, $u(x) = g^{-1}(w(x))$ is the desired solution.

6 Numerical examples

To show the efficiency of the proposed numerical method, we implement it on some Volterra integral equations. For every example we use a table that shows exact solution, our approximation and absolute errors in some points. In the following examples, the absolute error is used to check the accuracy. The amount is far more than other computational errors like mean absolute error.

Example 6.1 *$u(x) = e^{-x}$is the exact solution of the following Volterra integral equation of the first kind $xe^x = \int_0^x e^{x+t}u(t)dt$. Numerical solution of this equation and its errors are shown in table 1.*

Example 6.2 *Consider $x = \int/_0^x (x+t-1)u(y)dt$ with the exact solution $u(x) = e^{-x}$. The table 2 shows approximation solutions, error and exact solution in some points.*

Example 6.3 *$u(x) = 2sinx$ is the exact solution of the following Volterra integral equation of the first kind $xsinx = \int_0^x cos(x-t)u(t)dt$. Result of example 6.3 are shown in table 3.*

Example 6.4 *$x = \int_0^x (x-t+1)e^{-u(t)}dt$is a nonlinear Volterra integral equation of the first kind with exact solution $u(x) = x$. Table 4 shows the exact solution, approximation solutions and absolute errors at some points.*

Now, we compare our method with a direct method to solve Volterra integral equation of the rst kind using operational matrix with block-pulse functions [6] and an expansion-iterative method based on the block-pulse functions [21]. Consider example 6.1, Table 5 shows the mean-absolute errors for direct method and expansion-iterative method for two values of k, where k is the number of partitions of $[0, 1)$ also we can see absolute errors for some values of m, for the same example.
Table 6 shows the same errors for some different values of k and m for example 6.3. Tables 5 and 6 show our method is more accurate with respect to dimensions of the system. The final system in our method has smaller size than block-pulse methods also, as another advantage if $k(x, x) = 0$ then the block-pulse methods do not work and their final systems are incompatible but our method works correctly.

7 Stability

To demonstrate the stability of the method, we review effect of noise on data function. In other word, we replace $f(x)$ by$(1 + \varepsilon p)f(x)$ in (5.4) or (5.9).where$p$ is a real random number between -1 and 1, and ε is percent of noise.

Example 7.1 *Consider the following Volterra integral equation of the first kind $\frac{7}{12}x^4 = \int_)^x (x + t)u(t)dt$ with the exact solution $u(x) = x^2$.
Suppose p is a random real number in $(0, 1)$and $\varepsilon = 0.01, 0.02, 0.03$. In table 7, we present exact solution, approximate and noisy solutions at some points.*

Example 7.2 *$u(x) = x$ is the exact solution of the following Volterra integral equation of the first kind $x = \int_0^x (x + t)u(t)dt$.
Table 8 shows exact solution, approximate solution and noisy solutions.*

As a result of the tables, errors are proportional to the amount of noise.

8 Conclusion

In this article, we applied Bernsteins approximation to approximate the solution of linear and nonlinear Volterra integral equations of the first kind. In this method, we obtained some new operational matrices based on Bernstein polynomials. Our achieve results in this paper, show that our approach for solving Volterra integral equations of the first kind is very effective, simple and stable. The answers are trusty and their accuracy are high and we this method can be can executed in a computer easily. The numerical examples support this claim. The method can be applied for integro-differential equations, integral equations of the second and control problems.

References

[1] K. E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind,*Cambridge University Press, Cambridge* (1997).

[2] E.Babolian, L. Delves, An augmented Galerkin method for first kind Fredholm equations, *IMA Journal of Applied Mathematics* 24 (1979) 157-174.

[3] E. Babolian, R. Mokhtari, M. Salmani, Using direct method for solving variational problems via triangular orthogonal functions, *Applied Mathematics and Computation* 191 (2007) 206-217.

[4] E. Babolian, H. Marzban, M. Salmani, Using triangular orthogonal functions for solving Fredholm integral equations of the second kind, *Applied Mathematics and Computation* 201 (2008) 452-464.

[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] E. Babolian, Z. Masouri, Direct method to solve Volterra integral equation of the first kind using operational matrix with block-pulse functions, *Journal of Computational and Applied Mathematics* 220 (2008) 51-57.

[7] A. Bellour, E. Rawashdeh, Numerical solution of first kind integral equations by using Taylor polynomials, *J. Inequal. Speci. Func* 1 (2010) 23-29.

[8] P. J. Collins , Differential and integral equations, *Oxford University Press, Oxford* (2006).

[9] L. M. Delves, J. Walsh, Numerical solution of integral equations, *Clarendon Press, Oxford, UK* (1974).

[10] L. M. Delves, J.L. Mohamed, Computational methods for integral equations, *Cambridge: Cambridge University Press* (1985).

[11] M. A. Golberg, Numerical solution of integral equations, *Plenum Press, New York* (1990).

[12] G. Hanna, J. Roumeliotis, A. Kucera, Collocation and Fredholm integral equations of the first kind, *Journal of Inequalities in Pure and Applied Mathematics* 6 (2005) 1-8.

[13] C. Hwang, Y. Shih, Optimal control of delay systems via block pulse functions, *Journal of optimization theory and applications*,45 (1985) 101-112.

[14] K. Joy, Bernstein polynomials, *On-Line Geometric Modeling Notes* (2000).

[15] E. Kreyszig, Introductory functional analysis with applications, *Vol. 81, wiley, New York* (1989).

[16] B. A. Lewis, On the numerical solution of Fredholm integral equations of the first kind, *IMA Journal of Applied Mathematics* 16 (1975) 207-220.

[17] K. Maleknejad, E. Hashemizadeh, R. Ezzati, A new approach to the numerical solution of Volterra integral equations by using Bernsteins approximation, *Communications in Nonlinear Science and Numerical Simulation* 16 (2011) 647-655.

[18] K. Maleknejad, B. Basirat, E. Hashemizadeh, A Bernstein operational matrix approach for solving a system of high order linear Volterra Fredholm integro-differential equations, *Mathematical and Computer Modelling* 55 (2012) 1363-1372.

[19] B. Mandal, S. Bhattacharya, Numerical solution of some classes of integral equations using Bernstein polynomials, *Applied Mathematics and computation* 190 (2007) 1707-1716.

[20] F. L. Martinez, Some properties of two-dimensional Bernstein polynomials, *Journal of approximation theory* 59 (1989) 300-306.

[21] 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.

[22] K. Parand, S. A. Kaviani, Application of the exact operational matrices based on the Bernstein polynomials, *Journal of Mathematics and Computer Science* 6 (2013) 36-59.

[23] P. Paraskevopoulos, P. Sparis, S. Mouroutsos, The Fourier series operational matrix of integration, *International journal of systems science* 16 (1985) 171-176.

[24] P. Paraskevopoulos, The operational matrices of integration and differentiation for the Fourier sine-cosine and exponential series, *Automatic Control, IEEE Transactions on* 32 (1987) 648-651.

[25] P. Paraskevopoulos, P. Sklavounos, G. C. Georgiou, The operational matrix of integration for Bessel functions, *Journal of the Franklin Institute* 327 (1990) 329-341.

[26] G. M. Phillips, Interpolation and approximation by polynomials, *Vol. 14, Springer Science Business Media* (2003).

[27] M. Rahman, Integral Equations and Their Applications, *WIT Press, Southampton* (2007).

[28] P. Sannuti, Analysis and synthesis of dynamic systems via block-pulse functions, *Proceedings of the Institution of Electrical Engineers* 124 (1977) 569-571.

[29] A. Shahsavaran, Numerical approach to solve second kind Volterra integral equations of Abel type using Block-Pulse functions

and Taylor expansion by collocation method, *Appl. Math. Sci.* 5 (2011) 685-696.

[30] A. Shirin, M. Islam, Numerical solutions of Fredholm integral equations using Bernstein polynomials, *arXiv preprint arXiv:*1309.6311 (2013).

[31] S. C. Tsay, T. T. Lee, Solutions of integral equations via Taylor series, *International Journal of Control*, 44 (1986) 701-709.

[32] W. Wang, A mechanical algorithm for solving the Volterra integral equation, *Applied mathematics and computation* 172 (2006) 1323-1341.

[33] A. Wazwaz, Linear and nonlinear integral equations methods and applications, *Higher Education Press, Beijing Heidelberg* (2011).

[34] S. A. Yousefi, M. Behroozifar, Operational matrices of Bernstein polynomials and their applications, *International Journal of Systems Science* 41 (2010) 709-716.

[35] S. A. Yousefi, M. Behroozifar, M. Dehghan, The operational matrices of Bernstein polynomials for solving the parabolic equation subject to specification of the mass, *Journal of computational and applied mathematics* 235 (2011) 5272-5283.

[36] S. A. Yousefi, M. Behroozifar, M. Dehghan, Numerical solution of the nonlinear age-structured population models by using the operational matrices of Bernstein polynomials, *Applied Mathematical Modelling* 36 (2012) 945-963.

Mohsen Mohamadi is Ph.D. Of applied mathematics and Faculty of Basic Science, Mathematics Department, Islamic Azad University, Ayatolah Amoli Branch, Amol,Iran. Interested in Numerical Analysis, Integral Equations, and Differential equations.

Esmail Babolian - is Professor of applied mathematics and Faculty of Mathematical sciences and computer, Kharazmy University, Tehran, Iran. Interested in numerical solution of functional Equations, integral equations, differential equations, numerical linear algebra and mathematical education.

Sohrab Ali Yousefi- is Professor of applied mathematics and Faculty of Mathematical sciences and computer, Shahid Beheshti University, Tehran, Iran. Interested in Numerical Analysis, Inverse Problems, Integral Equations, Wavelets, Fractional Equations.