Math Sci (2017) 11:27–37
DOI 10.1007/s40096-016-0201-1

ORIGINAL RESEARCH

# Bernstein Multiscaling polynomials and application by solving Volterra integral equations

**M. Mohamadi**[1] · **E. Babolian**[1] · **S. A. Yousefi**[2]

Received: 21 August 2016 / Accepted: 7 November 2016 / Published online: 10 December 2016

**Abstract** In this paper, we present a direct computational method to solve Volterra integral equations. The proposed method is a direct method based on approximate functions with the Bernstein Multiscaling polynomials. In this method, using operational matrices, the integral equation turns into a system of equations. Our approach can solve nonlinear integral equations of the first kind and the second kind with piecewise solution. The computed operational matrices in this article are exact and new. The comparison of obtained solutions with the exact solutions shows that this method is acceptable. We also compared our approach with two direct and expansion–iterative methods based on the block-pulse functions. Our method produces a system, which is more economical, and the solutions are more accurate. Moreover, the stability of the proposed method is studied and analyzed by examining the noise effect on the data function. The appropriateness of noisy solutions with the amount of noise approves that the method is stable.

**Keywords** Volterra integral equation · Bernstein polynomials (BPs) · Bernstein Multiscaling polynomials (BMSPs) · Transformation matrices · Operational matrices

✉ E. Babolian
babolian@khu.ac.ir

M. Mohamadi
m.mohamadi@amoliau.ac.ir; m.moahamdi2000@gmail.com

S. A. Yousefi
s-yousefi@sbu.ac.ir

[1] Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Department of Mathematics, Shahid Beheshti University, G. C. Tehran, Iran

## Introduction

Most of integral equations of the first kind are ill-posed problems. Many authors paid attention to solve these equations and presented some methods [1–8]. These problems have application in mathematics, physics, and engineering. Recently, using polynomials have been common to solve these equations, see [9–27].

BMSPs are more general forms of BPs. One of the advantages of BMSPs compared to BPs is that they can approximate piecewise functions. In addition, using BMSP basis, we will have two degrees of freedom which increase accuracy of the method. One of these parameters is $m$, the degree of polynomials, and the other one $k$, which corresponds to the number of partitions in the interval $[a, b)$.

In this paper, we review BP properties and preliminary theorems in Subsect. 1 of Sect. “Review of Bernstein polynomials”. In Subsects. “Tranformation matrices” and “Operational matrices”, transformation matrices and operational matrices for BPs are computed. In Sect. “Bernstein Multiscaling polynomials”, BMSPs are defined. Transformational matrices and operational matrices for BMSPs are obtained in Sects. “Transformation matrices” and “BMSPs operational matrices”, respectively. In Sect. “Solution of Volterra integral equation”, by applying obtained matrices and functions approximation, the integral equations are turned into a system of equations. We present some numerical examples to illustrate the accuracy and ability of this method in Sect. “Numerical examples”. In next section, we compare our approach with two direct and expansion–iterative methods based on block-pulse functions. In Sect. “Stability of method”, stability is shown and we end this paper with a short conclusion in Sect. “Conclusion”.

## Review of Bernstein polynomials

### Preliminaries

**Definition 1**. Suppose $m$ is a positive integer number, BPs of degree $m$ on the interval $[a, b]$ are defined as

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

In addition, $B_{i,m}(x) = 0$ if $i < 0$ or $i > m$.

For convenience, we consider $[a, b] = [0, 1]$, namely

$$B_{i,m}(x) = \binom{m}{i} x^i (1-x)^{m-i}, \quad i = 0, \ldots, m.$$

We denote $\Phi_m$, an $m+1-$column vector, as follows: $\Phi_m(x) = [\phi_0(x), \ldots, \phi_m(x)]^T$, where $\phi_i(x) = B_{i,m}(x)$, $i = 0, \ldots, m$.

The BPs have many interesting properties [24–30]. However, here, some of them that are useful in our work are stated:

$$(\boldsymbol{p1}) \quad B_{i,m}(x) B_{j,m}(x) = \frac{\binom{m}{i}\binom{m}{j}}{\binom{2m}{i+j}} B_{i+j,2m}(x), \quad i, j = 0, \ldots, m.$$

$$(\boldsymbol{p2}) \quad B_{i,m}(x) = \sum_{j=0}^{k} \frac{\binom{m}{i}\binom{k}{j}}{\binom{m+k}{i+j}} B_{i+j,m+k}(x), \quad j, k = 0, \ldots, m.$$

$$(\boldsymbol{p3}) \quad \frac{B_{i,m}(x)}{\binom{m}{i}} = \frac{B_{i,m+1}(x)}{\binom{m+1}{i}} + \frac{B_{i+1,m+1}(x)}{\binom{m+1}{i+1}}, \quad i = 0, \ldots, m.$$

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

**Theorem A** [31] *Suppose* $H = l^2([a, b])$ *is a Hilbert space with the inner product defined by* $\langle f, g \rangle = \int_a^b f(t) g(t) \mathrm{d}t$, *and in addition,* $Y = \mathrm{Span}\{B_{0,m}(x), B_{1,m}(x), \ldots, B_{m,m}(x)\}$ *is the span space by Bernstein's 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. Therefore, f has the unique best approximation out of Y, such that* $y_0$

$$\exists y_0 \in Y; \forall y \in Y : \|f - y_0\|_2 \leq \|f - y\|_2.$$

*Therefore, there are the unique coefficients* $\alpha_j, j = 0, 1, \ldots, m$, *such that*

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

*where* $\alpha = [\alpha_0 \ \alpha_0 \ldots \alpha_m]^T$ *can be obtained by*

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

*such that* $\langle f(t), \Phi_m(t) \rangle = \int^b f(t) \Phi_m(t) \, \mathrm{d}t$.

*In above theorem, we*[a]*denote* $Q = \langle \Phi_m(t), \Phi_m(t) \rangle$ *as dual matrix. Furthermore, it is easy to see*

$$Q_{i,j} = \frac{\binom{m}{i-1}\binom{m}{j-1}}{(2m+1)\binom{2m}{i+j-2}}, \quad i, j = 1, \ldots, m+1.$$

*Next theorem indicates that dual matrix is symmetric and invertible.*

**Theorem B** [31] *Elements* $y_1, y_2, \ldots, y_n$ *of a Hilbert space H constitute a linearly independent set in H if and only if* $G(y_1, y_2, \ldots, y_n) \neq 0$.

*where* $G(y_1, y_2, \ldots, y_n)$ *is the Gram determinant of* $y_1, y_2, \ldots, y_n$ *defined by*

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

*For a two-dimensional function* $k(x, t) \in l^2([0, 1] \times [0, 1])$, *it can be similarly expanded with respect to BPs, such as*

$$k(x, t) \simeq \Phi_m^T(x) K \Phi_m(t),$$

*and K is the* $(m+1) \times (m+1)$ *BP coefficient matrix with*

$$K_{i,j} \simeq Q^{-1} \left( \int_0^1 (Q^{-1} \int_0^1 k(x, t) \phi_i(t) \, \mathrm{d}t) \phi_j(x) \, \mathrm{d}x \right), \quad i, j = 0, \ldots, m.$$

## Transformation matrices

Transformation matrix is used to change the dimension of the problem. In other words, this matrix can convert $\Phi_m$ to $\Phi_n$ 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 $\Phi_m$ to $\Phi_n$. In other words, $\Phi_m = T_m^n . \Phi_n$.

The increasing transformation matrix can be computed as follows:

$$[T^n_m]_{i,j} = \begin{cases} 0, \text{if } \ i<j \, \text{or} \, j>i+k, \\ \dfrac{\binom{m}{i-1}\binom{k}{j-1}}{\binom{m+k}{i+j-2}}, \text{otherwise.} \end{cases}$$

It is sufficient to use $p3$, $k$ times, where $k=n-m$.

In addition, decreasing transformation matrix is an $(n+1)\times(m+1)$ matrix, which is shown by $T^m_n$, and converts $\Phi_n$ to $\Phi_m$, where $n$ is greater than $m$. In other words, $\Phi_n = T^m_n\Phi_m$.

The $i$th row of decreasing transformation matrix can be calculated as follows:

$$[T^m_n]_{i+1} = \frac{1}{m+n+1} \left[ \frac{1}{\binom{m+n}{i}} \quad \frac{1}{\binom{m+n}{i+1}} \quad \cdots \quad \frac{1}{\binom{m+n}{i+m}} \right]^T Q^{-1}, \quad i=0,\ldots,n.$$

## Operational matrices

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

*Operational matrix of integration*

**Lemma 3** *Suppose* $\Phi_m(x) = [\phi_0(x),\ldots,\phi_m(x)]^T$,, *then*

$$\int_0^x \Phi_m(x)\mathrm{d}x = M\Phi_m(x). \tag{1}$$

*where M is called operational matrix of integration.*

*Proof* With a simple calculation can be seen

$$B_{i,m}(x) = \int_0^x m\big(B_{i-1,m-1}(t) - B_{i,m-1}(t)\big)\mathrm{d}t.$$

Assume $0\le k\le m$,

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

Therefore,

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

where

$$M_k = \frac{1}{m+1}\left[\overbrace{0\cdots0}^{k+1} \quad \overbrace{0\cdots1}^{m+1-k}\right]^T, \quad k=0,\ldots,m.$$

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, $M = im\,T^m_{m+1}$.

*Operational matrix of product*

**Lemma 4** *Let C be an* $(m+1)\times m+1$ *matrix, then*

$$\Phi_m^T(x)C\Phi_m(x) = \hat{C}^T\Phi_{2m}(x), \tag{2}$$

*where*

$$\hat{C}_k = \sum_{j=0}^{k}\frac{\binom{m}{k-j}\binom{m}{j}}{\binom{2m}{k}}C_{k-j,j}, \quad k=0,1,\ldots,2m.$$

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

$$\Phi_m^T(x)C\Phi_m(x) = \sum_{i=0}^{m}\sum_{j=0}^{m} c_{ij}\phi_i\phi_j.$$

Using $p1$ gives:

$$\Phi_m^T(x)C\Phi_m(x) = \sum_{i=0}^{m}\sum_{j=0}^{m} c_{ij}\frac{\binom{m}{i}\binom{m}{j}}{\binom{2m}{i+j}}\phi^*_{i+j}$$
$$= \left[\frac{\binom{m}{0}\binom{m}{0}}{\binom{2m}{0}}c_{00} \quad \sum_{j=0}^{1}\frac{\binom{m}{1-j}\cdot\binom{m}{j}}{\binom{2m}{1}}.c_{1-j,j} \quad \cdots \quad \sum_{j=0}^{k}\frac{\binom{m}{k-j}\cdot\binom{m}{j}}{\binom{2m}{k}}.C_{k-j,j}\cdots \quad \frac{\binom{m}{m}\binom{m}{m}}{\binom{2m}{2m}}c_{mm}\right]\Phi_{2m} = \hat{C}^T\Phi_{2m}(x).$$

**Lemma 5**. *Let u be an arbitrary* $(m+1)$ *vector, then*

$$\Phi_m(x)\Phi_m^T(x)u = \tilde{u}\Phi_{2m}(x). \tag{3}$$

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

$$\tilde{u}_{i+1,j+1} = \begin{cases} 0, & \mathit{if}\ j\langle i\ \mathit{or}\ j\rangle m+i, \\ \dfrac{\binom{m}{i}\cdot\binom{m}{j}}{\binom{2m}{i+j}}u_j, & \mathit{otherwise}, \end{cases}$$

$i = 0,\ldots,m,\ j = 0,\ldots,2m.$

*Proof*. Property $p1$ implies

$$\Phi_m(x)\Phi_m^T(x)u = \left[\sum_{j=0}^{m}\frac{\binom{m}{0}\binom{m}{j}}{\binom{2m}{j}}u_j\phi_j^* \quad \sum_{j=0}^{m}\frac{\binom{m}{1}\binom{m}{j}}{\binom{2m}{j+1}}u_j\phi_{j+1}^* \quad \cdots \quad \sum_{j=0}^{m}\frac{\binom{m}{m}\binom{m}{j}}{\binom{2m}{j+m}}u_j\phi_{j+m}^*\right]^T.$$

Now, the $i$th entry of the above matrix can be rewritten as follows:

$$\sum_{j=0}^{m}\frac{\binom{m}{i-1}\binom{m}{j}}{\binom{2m}{j+i-1}}u_j\phi_{j+i-1}^* = \left[\underbrace{0\ldots0}_{i-1} \quad \frac{\binom{m}{i-1}\binom{m}{0}}{\binom{2m}{i-1}}u_0 \quad \cdots \quad \frac{\binom{m}{i-1}\binom{m}{m}}{\binom{2m}{m+i-1}}u_m \quad \underbrace{0\ldots0}_{m-i+1}\right]\Phi_{2m}.$$

## Bernstein Multiscaling polynomials

**Definition 3.** Suppose $B_{i,m}(x)$ is the $i$th BPs of degree $m$ on interval $[0,1]$, Bernstein Multiscaling polynomials on $[0,1)$ define as follows:

$$\psi_{i,j}(t) = \begin{cases} B_{i,m}(kt-j), & \dfrac{j}{k}\le t < \dfrac{j+1}{k}, \\ 0, & \text{other wise,} \end{cases} \tag{4}$$

where $k\ge1$ is the number of partitions on $[0,1]$ and $i = 0,\ldots,m$, and in addition, $j = 0,\ldots,k-1$.

Now, every function $f\in\mathcal{L}^2([0,1))$ has the unique best approximation with respect to span space by BMSPs as follows:

$$f(x) = \sum_{j=0}^{k-1}\sum_{i=0}^{m}c_{i,j}\psi_{i,j} = C^T.\Psi,$$

where $C^T = [c_{0,0},c_{1,0},\ldots,c_{m-1,0},c_{m,0},\ldots,c_{0,k-1},c_{1,k-1},\ldots,c_{m-1,k-1},c_{m,k-1}]$.

We denote

$\Psi = [\Psi_0\cdots\Psi_{k-2}\quad \Psi_{k-1}]^T$ where $\Psi_i = [\psi_{0i}\ \psi_{1i}\cdots\psi_{mi}]^T,\quad i = 0,\ldots,k-1.$

Equation (4) implies that $\psi_i^{'}$s are disjoint. In other words when

$$i\ne j : \psi_i^T\psi_j = 0. \tag{5}$$

In addition, Eq. (4) and $p3$ imply

$$\psi_{i,j(x)}\psi_{p,q(x)} = \begin{cases} 0, & j\ne q, \\ \dfrac{\binom{m}{i}\cdot\binom{m}{j}}{\binom{2m}{i+j}}\psi^*_{i+p,j(x),} & j=q, \end{cases}\text{, where } \psi^*_{i+p,j}(x) \text{ is a elemet of } \Psi_{2m}. \tag{6}$$

**Lemma 6.** *Suppose $\bar{Q}$ is the dual operational matrix of BMSPs, then*

$$\overline{Q} = \frac{1}{k}\begin{bmatrix} Q & \bar{0} & \cdots & \bar{0} \\ \bar{0} & Q & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{0} \\ \bar{0} & \cdots & \bar{0} & Q \end{bmatrix},$$

*where $\bar{0}$ is an $(m+1)\times(m+1)$ zero matrix, and $Q$ is the dual operational matrix of BPs.*

*Proof.* With respect to (5) and (6), it is obvious.For any arbitrary function $f(x) \in \mathcal{L}^2([0,1))$, there is an expansion with respect to BMSPs, such that

$$f(x) \simeq \sum_{j=0}^{k-1}\sum_{i=0}^{m} f_{i,j}\psi_{i,j} = F^T\Psi = \Psi^T F, \tag{7}$$

where $F$ is a $k(m+1)-$ vector

Theorem A indicates

$$F \simeq \overline{Q}^{-1}\int_0^1 f(x)\Psi(x)\mathrm{d}x.$$

Let $k(x,t) \in \mathcal{L}^2([0,1)\times[0,1))$ be a two-dimensional function. With respect to BMSPs, $k$ has the following expansion:

$$k(x,t) \simeq \Psi^T(t)K\Psi(x), \tag{8}$$

where

$$K_{i+1,j+1} = \overline{Q}^{-1}\left(\int_0^1\left(\overline{Q}^{-1}\int_0^1 k(x,t)\phi_i(t)\mathrm{d}t\right)\phi_j(x)\mathrm{d}x\right), i,j = 0,\ldots,k(m+1).$$

## Transformation matrices

Let $\Psi_m$ and $\Psi_n$ be two different BMSPs and $m \le n$. There are two matrices, $\tau_m^n$ and $\tau_n^m$, such that $\Psi_m = \tau_m^n.\Psi_n$ and $\Psi_n = \tau_n^m.\Psi_m$. These transformation matrices have dimensions $k(m+1)\times k(n+1)$ and $k(n+1)\times k(m+1)$, respectively:

$$\tau_m^n = \begin{bmatrix} T_m^n & \bar{0} & \cdots & \bar{0} \\ \bar{0} & T_m^n & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{0} \\ \bar{0} & \ldots & \bar{0} & T_m^n \end{bmatrix} \text{and } \tau_n^m = \begin{bmatrix} T_n^m & \bar{0} & \cdots & \bar{0} \\ \bar{0} & T_n^m & \ddots & \vdots \\ \vdots & \ddots & \ddots & \bar{0} \\ \bar{0} & \ldots & \bar{0} & T_n^m \end{bmatrix},$$

where $T_m^n$ and $T_n^m$ are increasing transformation matrix and decreasing transformation matrix, respectively.

## BMSP operational matrices

Operational matrices for BMSPs are obtained by BPs operational matrices and results are similar.

### Operational matrices of product

**Lemma 7**. *Assume $A$ is a $k(m+1)\times k(m+1)$ matrix, then*

$$\Psi^T A\Psi = \hat{A}^T\Psi^*, \tag{9}$$

*where $\hat{A}$ is a $k(2m+1)-$ vector.*

*Proof.* Let $C$ is a $(m+1)\times(m+1)$ matrix, Eq. (5) implies

$$\Psi_i^T C\Psi_j = 0, \tag{10}$$

where $i \neq j$.

Furthermore, Lemma 4 gives:

$$\Psi_i^T C\Psi_i = \hat{C}^T\Psi_i^*, \tag{11}$$

where $\hat{C}$ is a $(2m+1)-$ vector.

Now, consider $A$ as follows:

$$A = \begin{bmatrix} A_{0,0} & \cdots & A_{0,k-1} \\ \vdots & \ddots & \vdots \\ A_{k-1,0} & \cdots & A_{k-1,k-1} \end{bmatrix},$$

where $A_{i,j}(i,j = 0,\ldots,k-1)$ is an $(m+1)\times(m+1)$ matrix.

Equation (10) implies:

$$\Psi^T A\Psi = \sum_{i=0}^{k-1}\Psi_i^T A_{ii}\Psi_i.$$

Equation (11) gives

$$\Psi^T A\Psi = \sum_{i=0}^{k-1}\hat{A}_{ii}^T.\Psi_i^*.$$

Therefore,

$$\Psi^T A\Psi = [\hat{A}_{0,0}\ldots\hat{A}_{k-1,k-1}]\Psi^* = \hat{A}^T\Psi^*,$$

where $\hat{A}_{i,i}$ is a $2m+1$ vector.

$$\sum_{i=0}^{k-1}\hat{A}_{ii}^T\Psi_i^* = [\hat{A}_{0,0}\ldots\hat{A}_{k-1,k-1}]\Psi^* = \hat{A}^T\Psi^*.$$

**Lemma 8.** *Suppose $C$ is an arbitrary $k(m+1)$ vector, then,*

$$\Psi\Psi^T C = \tilde{C}\Psi^*, \tag{12}$$

*where $\tilde{C}$ is a $k(m+1)\times k(2m+1)$ matrix.*

*Proof.* Suppose

$$C^T = [\, C_0 \quad C_1 \quad \cdots \quad C_{k-1}\,],$$

where $C_i(i=0,\ldots,k-1)$ is an $m+1-$ vector.

Equation (3) implies:

$$\Psi_i\Psi_i^T C_i = \tilde{C}_i\Psi_i^*, i=0,\ldots,k-1, \tag{13}$$

where $\tilde{C}_i$ is an $(m+1)\times(2m+1)$ matrix.

Now

$$\Psi.\Psi^T.C = \begin{bmatrix} \Psi_0 \\ \vdots \\ \Psi_{k-1} \end{bmatrix} . \left[\Psi_0^T \ \cdots \ \Psi_{k-1}^T\right] . \begin{bmatrix} C_0 \\ \vdots \\ C_{k-1} \end{bmatrix}$$

$$= \begin{bmatrix} \Psi_0.\Psi_0^T.C_0 & 0 & 0\cdots\ 0 \\ 0 & \Psi_1.\Psi_1^T.C_1 & \ddots\ \ \vdots \\ \vdots\ \ddots & \ddots & 0 \\ 0\ \cdots & 0 & \Psi_{k-1}.\Psi_{k-1}^T.C_i \end{bmatrix}.$$

Equation (12) gives:

$$\Psi\Psi^T C = \begin{bmatrix} \tilde{C}_0.\Psi_0^* & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{C}_{k-1}.\Psi_{k-1}^* \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{C}_0 & 0 & \cdots & 0 \\ 0 & \tilde{C}_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{C}_{k-1} \end{bmatrix}\Psi^* = \tilde{C}\Psi^*.$$

### Operational matrix of integration

**Lemma 7**. *Assume* $\mathcal{M}$ be the operational matrix of integration, then

$$\int_0^x \Psi(t)\mathrm{d}t = \mathcal{M}\Psi(x).$$

*Proof* Lemma 1 implies:

$$\int_0^x \Psi_0(t)\mathrm{d}t = \begin{cases} \dfrac{1}{k}M\Psi_0(t), & 0\le x<\dfrac{1}{k}, \\ \dfrac{1}{k(m+1)}, & x\ge\dfrac{1}{k}. \end{cases}$$

For $i=1,\ldots,k-2$, Eq. (1) gives:

$$\int_0^x \Psi_i(t)\mathrm{d}t = \begin{cases} 0, & 0\le x<\dfrac{i}{k}, \\ \dfrac{1}{k}M\Psi_i(t), & \dfrac{i}{k}\le x<\dfrac{i+1}{k}, \\ \dfrac{1}{k(m+1)}, & x\ge\dfrac{i+1}{k}, \end{cases}.$$

and

$$\int_0^x \Psi_{k-1}(t)\mathrm{d}t = \begin{cases} 0, & 0\le x<\dfrac{k-1}{k}, \\ \dfrac{1}{k}M\Psi_{k-1}(t), & \dfrac{k-1}{k}\le x<1. \end{cases}$$

Consequently,

$$\mathcal{M} = \begin{bmatrix} \frac{1}{k}M & \frac{1}{k(m+1)}\bar{1} & \cdots & \frac{1}{k(m+1)}\bar{1} \\ 0 & \frac{1}{k}M & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{1}{k(m+1)}\bar{1} \\ 0 & \cdots & 0 & \frac{1}{k}M \end{bmatrix},.$$

where $\bar{1}$ is an $(m+1)\times(m+1)$ matrix that all entries are one.

## Solution of Volterra integral equation

In this section, we are going to convert an integral equation to a system.

### Linear Volterra integral equation of the first kind

Consider the following Volterra integral equation of the first kind:

$$f(x) = \int_0^x k(x,t)u(t)\mathrm{d}t. \tag{14}$$

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 BMSPs gives:

$$f(x) = F^T\Psi(x) = \Psi^T(x)F,$$

$$u(t) = U^T\Psi(t) = \Psi^T(t)U,$$

$$k(x,t) \simeq \Psi^T(x)K\Psi(t), \tag{15}$$

where the vectors $F, U$, and matrix $K$ are BMSP coefficients of $f(x), u(t)$, and $k(x,t)$, respectively.

Now, replacing (15) into (14) gives:

$$F^T.\Psi_m(x) = \int_0^x \Psi_m^T(x)K\Psi_m(t)\Psi_m^T(t)U\mathrm{d}t$$

$$= \Psi_m^T(x)K\int_0^x \Psi_m(t)\Psi_m^T(t)U\mathrm{d}t.$$

Using (12) follows:

$$F^T\Psi_m(x) = \Psi_m^T(x)K\int_0^x \tilde{U}\Psi_{2m}(t)\mathrm{d}t$$

$$= \Psi_m^T(x)K\tilde{U}\int_0^x \Psi_{2m}(t)\mathrm{d}t. \tag{16}$$

Using operational matrix of integration $\mathcal{M}$, in Eq. (16) gives

$$F^T\Psi_m(x) = \Psi_m^T(x)K\tilde{U}M\Psi_{2m}(x). \tag{17}$$

Let $U^* = K\tilde{U}M\tau_{2m}^m$, where $U^*$ is a $k(m+1) \times k(m+1)$.

Equation (17) changes to:

$$F^T\Psi_m(x) = \Psi_m^T(x)U^*\Psi_m(x) \tag{18}$$

Using Eq. (11) in (18) gives:

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

Using decreasing transformation matrix $\tau_{2m}^m$, gives the final system:

$$\overline{U} = F, \tag{19}$$

where $\overline{U}^T = \hat{U^*}^T\tau_{2m}^m$.

## Nonlinear Volterra integral equation of the first kind

Consider the following nonlinear Volterra integral equation:

$$f(x) = \int_0^x k(x,t)g(u(t))\mathrm{d}t, \tag{20}$$

Put $w(x) = g(u(x))$ and subsequently $w(x) = W^T\Psi_m(x)$.

$$\tag{21}$$

where $W$ is an unknown $k(m+1)$ vector. Following the same procedure, the final system is as follows:

$$\overline{W} = F,$$

Finally, $u(x) = g^{-1}(w(x))$ is the desire solution.

One advantage of this method is solving linear or nonlinear Volterra integral equation of the second kind with piecewise functions. In these equations, solution, kernel, or data function can be piecewise. It is essential $k$, the number of partitions, be chosen, such that discontinuity points lie on boundary point of partitions.

## Linear Volterra integral equation of the second kind

Consider linear Volterra integral equations as the following form:

$$u(x) = f(x) + \int_0^x k(x,t)u(t)\mathrm{d}t. \tag{22}$$

Substituting (15) into (22) and a process similar to the previous state final linear system is

$$U - \overline{U} = F.$$

## Nonlinear Volterra integral equation of the second kind

Consider the following nonlinear Volterra equation of the second kind:

$$u(x) = f(x) + \int_0^x k(x,t)g(u(t))\mathrm{d}t \tag{23}$$

**Table 1** Results of Example 1

| $x$ | $m = 8$, $k = 4$ | Absolute error | Exact solution |
|---|---|---|---|
| 0 | 1.000000000 | 0.00000000 | 1.00000000 |
| 0.1 | 0.9048374873 | $6.93 \times 10^{-8}$ | 0.904837418 |
| 0.2 | 0.8187326019 | $1.81 \times 10^{-6}$ | 0.8187307531 |
| 0.3 | 0.740829897 | $1.16 \times 10^{-5}$ | 0.7408182207 |
| 0.4 | 0.670361092 | $4.1 \times 10^{-5}$ | 0.670320046 |
| 0.5 | 0.606644000 | $1.13 \times 10^{-4}$ | 0.6065306597 |
| 0.6 | 0.549040000 | $2.28 \times 10^{-4}$ | 0.5488116361 |
| 0.7 | 0.497190000 | $6.04 \times 10^{-4}$ | 0.4965853038 |
| 0.8 | 0.449100000 | $3.28 \times 10^{-4}$ | 0.4493289641 |
| 0.9 | 0.405500000 | $3.06 \times 10^{-4}$ | 0.4065696597 |

**Table 2** Results of Example 2

| $x$ | $m = 4$, $k = 4$ | Absolute error | $m = 8$, $k = 4$ | Absolute error | Exact solution |
|---|---|---|---|---|---|
| 0 | 0.0006126852892 | $5.91 \times 10^{-4}$ | 0.0001240979 | $1.24 \times 10^{-4}$ | 0.0000000000 |
| 0.1 | 0.1998624698 | $1.95 \times 10^{-4}$ | 0.1998263752 | $1.59 \times 10^{-4}$ | 0.1996668333 |
| 0.2 | 0.3978000958 | $4.61 \times 10^{-4}$ | 0.3971608136 | $1.77 \times 10^{-4}$ | 0.3973386616 |
| 0.3 | 0.5906752400 | $3.65 \times 10^{-4}$ | 0.5911505000 | $1.10 \times 10^{-4}$ | 0.5910404134 |
| 0.4 | 0.7785010600 | $3.35 \times 10^{-4}$ | 0.7788225000 | $1.41 \times 10^{-5}$ | 0.7788366846 |
| 0.5 | 0.9606750920 | $1.82 \times 10^{-3}$ | 0.9590100000 | $1.58 \times 10^{-4}$ | 0.9588510772 |
| 0.6 | 1.130310725 | $1.02 \times 10^{-3}$ | 1.1293000000 | $1.50 \times 10^{-5}$ | 1.129284947 |
| 0.7 | 1.286968954 | $1.46 \times 10^{-3}$ | 1.2890000000 | $5.64 \times 10^{-4}$ | 1.288435374 |
| 0.8 | 1.443023053 | $8.31 \times 10^{-3}$ | 1.4345600000 | $1.52 \times 10^{-4}$ | 1.434712182 |
| 0.9 | 1.56155138 | $5.10 \times 10^{-3}$ | 1.566430000 | $2.23 \times 10^{-4}$ | 1.566653819 |

**Table 3** Results of Example 3

| $x$ | $m = 5$, $k = 4$ | Absolute error | Exact solution |
|---|---|---|---|
| 0.00 | 0.0000207940 | $2.0 \times 10^{-5}$ | 0.00000000 |
| 0.15 | 0.1500076065 | $7.6 \times 10^{-6}$ | 0.15000000 |
| 0.30 | 0.3000093281 | $9.3 \times 10^{-6}$ | 0.30000000 |
| 0.45 | 0.4499909631 | $9.0 \times 10^{-6}$ | 0.45000000 |
| 0.60 | 0.5999915857 | $8.4 \times 10^{-6}$ | 0.60000000 |
| 0.75 | 0.7499869395 | $1.3 \times 10^{-5}$ | 0.75000000 |
| 0.90 | 0.9000073666 | $7.3 \times 10^{-6}$ | 0.90000000 |

where $g(x)$ is one to one on $[0, 1]$.

Let $w(t) = g(u(t))$, substituting (23) gives

$$w(x) = g\left(f(x) + \int_0^x k(x,t)u(t)\mathrm{d}t\right). \tag{24}$$

Now, substituting (15) and (20) into (24) implies:

$$W^T\Psi = g\left(F^T\Psi + \Psi_m^T(x)K\int_0^x \Psi(t)\Psi^T(t)W\mathrm{d}t\right).$$

Equations (16)–(19) give:

$$W = g(F + \bar{W}).$$

## Numerical examples

Now, we test our method on some numerical examples; in every example, we use a table to show approximations, exact solution, and absolute errors in some points.

*Example 1* Suppose $u(x) = e^{-x}$ be the exact solution of the following Volterra integral equation of the first kind:

$$xe^x = \int_0^x e^{x+t}u(t)\mathrm{d}t.$$

Table 1 shows results of Example 1.

*Example 2* Consider the following integral equation:

$$x\sin x = \int_0^x \cos(x-t)u(t)\mathrm{d}t$$

with the exact solution $u(x) = 2\sin x$.

Table 2 shows approximated solutions, absolute errors, and exact solution in some points.

*Example 3* $x = \int_0^x (x - t + 1)e^{-u(t)}\mathrm{d}t$

Table 3 shows results of Example 3.

*Example 4* Consider the following linear integral equation $u(x) = f(x) + \int_0^x (t - x)u(t)\mathrm{d}t$, with exact solution:

$$u(x) = \begin{cases} x, & 0 < x < 0.5, \\ x^2, & 0.5 < x < 1, \end{cases}$$

and the nonsmooth data function:

$$f(x) = \begin{cases} x + \dfrac{1}{6}x^3, & 0 < x < 0.5, \\ \dfrac{-5}{192} + \dfrac{1}{12}x + x^2 + \dfrac{1}{12}x^4, & 0.5 < x < 1. \end{cases}$$

Results of Example 4 are presented in Table 4.

*Example 5* Assume $u(x) = \begin{cases} 1 - x, & 0 < x < 0.5, \\ \sqrt{x}, & 0.5 < x < 1, \end{cases}$ be the exact solution of the following nonlinear Volterra integral equation of the second kind:

$$u(x) = f(x) + \int_0^x xtu^2(t)\mathrm{d}t,$$

where

$$f(x) = \begin{cases} 0, & x<0, \\ 1 - x - \dfrac{x^3}{2} - \dfrac{2}{3}x^4 - \dfrac{x^5}{4}, & 0<x\leq 0.5, \\ \sqrt{x} - 0.01x - \dfrac{1}{3}x^4, & 0.5\leq x<1, \\ -0.35x, & x>1. \end{cases}$$

Table 5 shows results of Example 5.

**Table 4** Results of Example 4

| $x$ | $m = 4$, $k = 4$ | Absolute error | Exact solution |
|---|---|---|---|
| 0 | 0.0000000000 | $1.1 \times 10^{-5}$ | 0.0000000000 |
| 0.1 | 0.0999999999 | $3.5 \times 10^{-6}$ | 0.0998334166 |
| 0.2 | 0.2000000000 | $4.7 \times 10^{-6}$ | 0.1986693308 |
| 0.3 | 0.3000000000 | $0.9 \times 10^{-7}$ | 0.2955202067 |
| 0.4 | 0.4000000001 | $2.1 \times 10^{-6}$ | 0.3894183423 |
| 0.5 | 0.2500000000 | $3.2 \times 10^{-9}$ | 0.2500000000 |
| 0.6 | 0.3600000002 | $3.9 \times 10^{-9}$ | 0.3600000000 |
| 0.7 | 0.4900000460 | $4.6 \times 10^{-9}$ | 0.490000000 |
| 0.8 | 0.6399999998 | $5.4 \times 10^{-9}$ | 0.640000000 |
| 0.9 | 0.8100000002 | $6.3 \times 10^{-9}$ | 0.810000000 |

**Table 5** Results of Example 5

| $x$ | $m = 3$, $k = 4$ | Absolute error | Exact solution |
|---|---|---|---|
| 0 | 1.0000000000 | 0.0000000 | 1.0000000000 |
| 0.1 | 0.8999990300 | $907 \times 10^{-7}$ | 0.9000000000 |
| 0.2 | 0.7999912000 | $8.8 \times 10^{-6}$ | 0.8000000000 |
| 0.3 | 0.6999800000 | $2.0 \times 10^{-5}$ | 0.7000000000 |
| 0.4 | 0.5999939000 | $6.1 \times 10^{-5}$ | 0.6000000000 |
| 0.5 | 0.7069967812 | $1.1 \times 10^{-4}$ | 0.7071067812 |
| 0.6 | 0.7744366692 | $1.6 \times 10^{-4}$ | 0.7745966692 |
| 0.7 | 0.8364200265 | $2.4 \times 10^{-4}$ | 0.8366600265 |
| 0.8 | 0.894097191 | $3.3 \times 10^{-4}$ | 0.8944271910 |
| 0.9 | 0.9482032981 | $4.8 \times 10^{-4}$ | 0.9486832981 |

## Comparison

Block-pulse functions are a special case of BMSPs. However, our method is different from the methods, as presented in [32] and [33]. Consider Example 1, in Table 6, the expansion–iterative method and direct method are compared with our method. In Table 6, we presented mean-absolute errors for the expansion–iterative method and direct method and absolute error of BMSPs for two different values of $m$ and $k$. With respect to dimensions of the final system, our method is more accurate than the expansion–iterative method and direct method.

Consider Example 2, Table 7 shows a comparison between BMSPs method and the expansion–iterative method and direct method with block-pulse functions. Mean-absolute errors for methods are presented in Table 7.

## Stability of method

To demonstrate the stability of the method, we review effect of noise on data function. In other words, we replace $f(x)$ by $(1+\varepsilon p)f(x)$ into integral equation. Where $p$ is a real random number between $-1$ and 1, and $\varepsilon$ is percent of noise. Now, we want to show that our method is stable and noise is proportional to the variations of solutions.

**Table 6** Results of comparison in Example 1

| Method | Mid-points, $k = 32$ | Mid-points, $k = 64$ | Ten points, $k = 32$ | Ten points, $k = 64$ |
|---|---|---|---|---|
| Expansion–iterative method | $6.6 \times 10^{-4}$ | $1.9 \times 10^{-4}$ | $5.2 \times 10^{-3}$ | $2.6 \times 10^{-3}$ |
| Direct method | $3.3 \times 10^{-3}$ | $1.6 \times 10^{-3}$ | $5.9 \times 10^{-3}$ | $2.9 \times 10^{-3}$ |
| | $m = 5$, $k = 4$ | $m = 3$, $k = 1$ | | |
| BMSPs method | $1.26 \times 10^{-4}$ | $2.01 \times 10^{-3}$ | | |

**Table 7** Results of comparison in Example 2

| Method | Mid-points, $k = 64$ | Mid-points, $k = 128$ | Ten points, $k = 64$ | Ten points, $k = 128$ |
|---|---|---|---|---|
| Expansion–iterative method | $4.9 \times 10^{-4}$ | $1.4 \times 10^{-4}$ | $6.5 \times 10^{-3}$ | $3.3 \times 10^{-3}$ |
| Direct method | $5.2 \times 10^{-3}$ | $2.6 \times 10^{-3}$ | $8.2 \times 10^{-3}$ | $4.1 \times 10^{-3}$ |
| | $m = 4$, $k = 4$ | $m = 8$, $k = 4$ | | |
| BMSPs method | $1.96 \times 10^{-4}$ | $1.70 \times 10^{-5}$ | | |

**Table 8** Results of Example 6

| $x$ | $m = 5, k = 4$ | $m = 5, k = 4, \varepsilon = 0.01$ | $m = 5, k = 4, \varepsilon = 0.02$ | $m = 5, k = 4\varepsilon = 0.03$ | Exact solution |
|---|---|---|---|---|---|
| 0.0 | 0.00000 | 0.01558624677 | 0.01538079594 | 0.01523810893 | 0.00000 |
| 0.1 | 0.01000 | 0.00740436021 | 0.0747198483 | 0.007231113603 | 0.01000 |
| 0.2 | 0.04000 | 0.040255392 | 0.039719848 | 0.0393334275 | 0.04000 |
| 0.3 | 0.09000 | 0.089864898 | 0.088666389 | 0.0878033869 | 0.09000 |
| 0.4 | 0.16000 | 0.159423786 | 0.157302906 | 0.1557740489 | 0.16000 |
| 0.5 | 0.25000 | 0.255541400 | 0.252166300 | 0.2496420000 | 0.25000 |
| 0.6 | 0.36000 | 0.360208100 | 0.3553999000 | 0.3519660000 | 0.36000 |
| 0.7 | 0.49000 | 0.490870000 | 0.4833260000 | 0.4786120000 | 0.49000 |
| 0.8 | 0.64000 | 0.633360000 | 0.6249109639 | 0.6189400000 | 0.64000 |
| 0.9 | 0.81000 | 0.804420000 | 0.7936019242 | 0.7858300000 | 0.81000 |

*Example 6* Consider the following Volterra integral equation of first kind:

$$\frac{7}{12}x^4 = \int_0^x (x+t)u(t)\mathrm{d}t$$

with exact solution $u(x) = x^2$.

Table 8 shows approximated solution, noisy solutions, and exact solution.

## Conclusion

BMSPs that we use to solve Volterra integral equations have acceptable accuracy. Operational matrices, which we have computed, are exact. These exact matrices lead to fewer errors in our computations. In addition, BMSPs can solve piecewise Volterra integral equations of the second kind. Effect of noise on data function shows that our method is reliable and ill-posedness does not occur. This method with respect to complexity of computations and desirable accuracy is recommended. Furthermore, this method can be used to solve optimal control equations, differential equations, and systems of integral or differential equations.

## References

1. Delves, L.M., Walsh, J.: Numerical solution of integral equations. Clarendon Press, Oxford (1974)
2. Delves, L.M., Mohamed, J.L.: Computational methods for integral equations. Cambridge University Press, Cambridge (1985)
3. Golberg, M.A.: Numerical solution of integral equations. Plenum Press, Berlin (1990)
4. Atkinson, K.E.: The numerical solution of integral equations of the second kind. Cambridge University Press, Cambridge (1997)
5. Collins, P.J.: Differential and integral equations. Oxford University Press, Oxford (2006)
6. Rahman, M.: Integral equations and their applications. WIT Press, Southampton (2007)
7. Wazwaz, A.: Linear and nonlinear integral equations methods and applications. Higher Education Press, Beijing (2011)
8. Chakrabarti, A., Martha, S.C.: Methods of solution of singular integral equations. Math. Sci. **6**, 15 (2012)
9. Rashidinia, J., Najafi, E., Arzhang, A.: An iterative scheme for numerical solution of Volterra integral equations using collocation method and Chebyshev polynomials. Math. Sci. **6**, 60 (2012)
10. Maleknejad, K., Hashemizadeh, E., Ezzati, R.: A new approach to the numerical solution of Volterra integral equations by using Bernstein's approximation, Commun. Nonlinear Sci. Numer. Simul. **16**, 647–655 (2011)
11. Mandal, B.N., Bhattacharya, S.: Numerical solutions of some classes of integral equations using Bernstein polynomials. Appl. Math. Comput. **190**, 1707–1716 (2007)
12. Shahsavaran, A.: 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**, 685–696 (2011)
13. Bellour, A., Rawashdeh, A.E.: Numerical solution of first kind integral equations by using Taylor polynomials. J. Inequal. Spec. Func. **1**, 23–29 (2011)
14. Wang, W.: A mechanical algorithm for solving the Volterra integral equation. Appl. Math. Comput. **172**, 1323–1341 (2006)
15. Shirin, A., Islam, M.S.: Numerical solutions of Fredholm integral equations using Bernstein polynomials. J. Sci. Res. **2**(2), 264–272 (2010)
16. Sannuti, P.: Analysis and synthesis of dynamic systems via block pulse functions. Proc. Inst. Electr. Eng. **124**, 569–571 (1977)
17. Hwang, C., Shih, Y.P.: Solution of integral equations via Laguerre polynomials. J. Optim. Theory Appl. **45**, 101–112 (1985)
18. Paraskevopoulos, P.N., Sparis, P.D., Mouroutsos, S.G.: The Fourier series operational matrix of integration. Int. J. Syst. Sci. **16**, 171–176 (1985)
19. Tsay, S.C., Lee, T.T.: Solution of the integral equations via Taylor series. Int. J. Control **44**, 701–709 (1986)
20. Paraskevopoulos, P. N.: The operational matrices of integration and differentiation for Fourier sine-cosine and exponential series. IEEE Trans. Autom. Control. **32**, 648–651 (1987)

21. Paraskevopoulos, P.N., Sklavounos, P., Georgiou, G.C.: The operation matrix of integration for Bessel functions. J. Franklin Inst. **327**, 329–341 (1990)
22. Babolian, E., Mokhtari, R., Salmani, M.: Using direct method for solving variational problems via triangular orthogonal functions. Appl. Math. Comput. **191**, 206–217 (2007)
23. Babolian, E., Marzban, H.R., Salmani, M.: Using triangular orthogonal functions for solving Fredholm integral equations of the second kind. Appl. Math. Comput. **201**, 452–464 (2008)
24. Yousefi, S.A., Behroozifar, M.: Operational matrices of Bernstein polynomials and their applications. Int. J. Syst. Sci. **41**(6), 709–716 (2010)
25. Yousefi, S.A., Behroozifar, M., Dehghan, M.: The operational matrices of Bernstein polynomials for solving the parabolic equation subject to specification of the mass. J. Comput. Appl. Math. **235**, 5272–5283 (2011)
26. Maleknejad, K., Basirat, B., Hashemizadeh, E.: A Bernstein operational matrix approach for solving a system of high order linear Volterra-Fredholm integro-differential equations. Math. Comput. Model. **55**, 1363–1372 (2012)
27. Yousefi, S.A., Behroozifar, M., Dehghan, M.: Numerical solution of the nonlinear age-structured population models by using the operational matrices of Bernstein polynomials. Appl. Math. Model. **36**, 945–963 (2012)
28. Joy, K. I.: Bernstein polynomials. On-line Geom. Model. Notes. (2000). http://www.idav.ucdavis.edu/education/CAGDNotes/Bernstein-Polynomials/Bernstein-Polynomials.html. Accessed 28 Nov 2000
29. Phillips, G.M.: Interpolations and approximation by polynomials. Springer, New York (2003)
30. Parand, K., Kaviani, S.A.: Application of the exact operational matrices based on the Bernstein polynomials. J. Math. Comput. Sci. **6**, 36–59 (2013)
31. Kreyszig, E.: Introduction functional analysis with applications. Wiley, New Jersey (1978)
32. Babolian, E., Masouri, Z.: Direct method to solve Volterra integral equation of the first kind using operational matrix with block-pulse functions. J. Comput. Appl. Math. **220**, 51–57 (2008)
33. Masouri, Z., Babolian, E., Varmazyar, S.H.: An expansion iterative method for numerically solving Volterra integral equation of the first kind. Comput. Math Appl. **59**, 1491–1499 (2010)