International Journal of
Mathematical Modelling & Computations
Vol. 07, No. 01, Winter 2017, 79- 91

Approximate Solution of System of Nonlinear Volterra Integro-Differential Equations by Using Bernstein Collocation Method

S. Davaeifar[a,*] and J. Rashidinia[b,**]

[a,b] *Department of Mathematics, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*

Abstract. This paper presents a numerical matrix method based on Bernstein polynomials (BPs) for approximate the solution of a system of m-th order nonlinear Volterra integro-differential equations under initial conditions. The approach is based on operational matrices of BPs. Using the collocation points, this approach reduces the systems of Volterra integro-differential equations associated with the given conditions, to a system of nonlinear algebraic equations. By solving such arising nonlinear system, the Bernstein coefficients can be determined to obtain the finite Bernstein series approach. Numerical examples are tested and the resultes are incorporated to demonstrate the validity and applicability of the approach. Comparisons with a number of conventional methods are made in order to verify the nature of accuracy and the applicability of the proposed approach.

Received: 15 August 2016, Revised: 08 February 2016, Accepted: 11 May 2017.

Keywords: Systems of nonlinear Volterra integro-differential equations; The Bernstein polynomials and series; Operational matrices; Numerical matrix method; Collocation points.

AMS Subject Classification: 34A12, 34A34, 45D05, 45G15, 45J05, 65R20.

Index to information contained in this paper

1. Introduction

Many problems in physics and engineering give rise to integral and integro-differential equations, the solution of which is of crucial importance. When a phys-

*Email adress: sara.davaei@yahoo.com

**Corresponding author. Email: rashidinia@iust.ac.ir

http://ijm2c.iauctb.ac.ir

ical system is modeled under the differential sense; it prepares a differential equation, an integral equation or an integro-differential equation. Forthcoming of the first two equations mostly appear in the last one.

Several numerical and analytical methods have recently been applied to obtain the solution of systems of linear and nonlinear Volterra integro-differential equations. Babolian and Biazar [5, 8] used Adomian decomposition method (ADM) for solving the systems of nonlinear Volterra integral equations of first and second kind. Such method has been universally exerted for solving high order linear Volterra-Fredholm integro-differential equations in [13] and restarted ADM for system of Volterra integral equations [22]. ADM is an analytical technique that appraises the solution in the form of Adomian polynomials. In this way there is no need to simplify or discrete the main equation and can be applied to both linear and nonlinear problems.

The variational iteration method (VIM) is an analytical approach in a manner that a correction functional is constructed by a general Lagrange multiplier [21, 25, 26, 30]. The latter can optimally be identified through the variational theory.

Authors in [9, 30] have applied He's homotopy perturbation method (HPM) for systems of integro-differential equations. HPM depends on the classical perturbation method and the homotopy method in topology, which is also an analytical approach.

In [4] differential transform method (DTM) is applied to both systems of integro-differential and integral equations. In [11], the system of linear and nonlinear Volterra integral equations of the first and second kind has been solved by DTM. The latter is a semi analytical-numerical technique that depends on Taylor series.

The system of linear Volterra and Fredholm integral equations is solved by a practical direct method in [6]. This approach is based on vector forms of orthogonal triangular functions and its operational matrices. In addition, the system of integral equations reduces to a system of algebraic equations without any integration. The set of triangular orthogonal functions are also utilized as a basis functions in a direct method to approximate the solution of system of integro-differential equations [3].

The Taylor series expansion method for the solution of nonlinear Volterra integro-differential equations and system of nonlinear Volterra equations have been given in [19, 20].

The system of Volterra integro-differential equations of nonlinear type is solved in [24] by using Legendre wavelets operational method. In [17], the system of nonlinear Volterra integral equations is solved by using Simpsons 3/8 rule. The block by block method is introduced for solving system of nonlinear Voltera integral equations of the second kind in [2]. A system of Volterra-Fredholm integral equations has been solved by using Chebyshev collocation method in [14]. Bernstein collocation method have been used to solve system of linear and nonlinear Fredholm integral equations in [16, 23].

In this work we consider the systems of the m-th order nonlinear Volterra integro-differential equations with variable coefficients in the form

$$\begin{aligned}&\sum_{\theta=0}^{m}\sum_{j=1}^{k} g_{ij}^{\theta}(x)\left[u_j^{(\theta)}(x)\right]^{\gamma_{ij}} - \int_0^x \sum_{j=1}^{k} \kappa_{ij}(x,\, s)\, F_{ij}(s,\, U(s))\, ds \\ &= f_i(x); \quad i = 1,\, 2,\, \dots,\, k,\ 0 \le x,\, s \le 1,\end{aligned} \tag{1}$$

under initial conditions

$$u_j^{(t)}(0) = \delta_{j,\,t}, \quad t = 0,\, 1,\, \dots,\, m-1,\ j = 1, 2,\, \dots,\, k, \tag{2}$$

where $F_{ij}(s, U(s)) = F_{ij}(s, u_1(s), u_2(s), \ldots, u_k(s))$, $i, j = 1, 2, \ldots, k$, are given continuous functions which are nonlinear with respect to $u_j(s)$, $j = 1, 2, \ldots, k$, $\left\{g_{ij}^{\theta}(x)\right\}_{i,j=1,\theta=0}^{k,m}$, $\{f_i(x)\}_{i=1}^{k} \in L^2[0, 1]$ and $\{\kappa_{ij}(x, s)\}_{i,j=1}^{k} \in L^2([0, 1] \times [0, 1])$ are known functions and $u_j^{(\theta)}(x)$ is the θ- th order derivative of $u_j(x)$ for $j = 1, 2, \ldots, k$ and the real coefficients $\{\delta_{j,t}\}_{j=1,t=0}^{k,m-1}$ are appropriate constants. In this work we suppose $F_{ij}(s, U(s)) = u_1^{\lambda_1^{ij}}(s)\, u_2^{\lambda_2^{ij}}(s) \ldots u_k^{\lambda_k^{ij}}(s)$ where λ_l^{ij}, $i, j, l = 1, 2, \ldots, k$ are non-negative integers.

The aim is to use the BPs to solve systems of the m-th order nonlinear Volterra integro-differential equations of the form Eq. (1). BPs has numerous properties [12, 15].

2. BPs and their properties

For $n \geq 1$ the general form of the BPs of n-th degree over the interval [0, 1] as defined in [7] is given by:

$$B_{r,n}(x) = \binom{n}{r} x^r (1-x)^{n-r}, \quad 0 \leq x \leq 1, \; r = 0, 1, \ldots, n, \tag{3}$$

where

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}. \tag{4}$$

It is noteworthy that these polynomials have the following features:

$$\begin{array}{l} (i)\, B_{r,n}(x) = 0, \quad if \; r < 0 \; or \; r > n, \\ (ii)\, B_{r,n}(0) = B_{r,n}(1) = 0 \quad for \; 1 \leq r \leq n-1, \\ (iii)\, \sum_{r=0}^{n} B_{r,n}(x) = 1. \end{array} \tag{5}$$

$\{B_{r,n}(x),\, r = 0, 1, \ldots, n\}$ in Hilbert space $L^2[0, 1]$, is a complete nonorthogonal set [18]. A recursive definition can also be used to generate the BPs over [0, 1] so that the r-th, n-th degree BPs can be expressed:

$$B_{r,n}(x) = (1-x)\, B_{r,n-1}(x) - x\, B_{r-1,n-1}(x). \tag{6}$$

Any arbitrary polynomial of degree n can be expanded in terms of a linear combination of these basis functions.

2.1 *Function approximation*

A function $f(x)$ square integrable in $(0, 1)$, can be approximated by the BPs basis of degree n as:

$$f(x) \cong f_{n+1}(x) = \sum_{r=0}^{n} c_r\, B_{r,n}(x) = C^T \phi(x), \tag{7}$$

where C and $\phi(x)$ are $(n+1)\times 1$ vectors given by:

$$C = [c_0,\, c_1,\, \ldots,\, c_n]^T, \tag{8}$$

and

$$\phi\,(x) = [B_{0,\,n}(x),\, B_{1,\,n}(x),\, \ldots,\, B_{n,\,n}(x)]^T. \tag{9}$$

The function of two variables $\kappa(x,\, s) \in L^2\,([0,\,1]\times[0,\,1])$ can be approximated as follows:

$$\kappa(x,\, s) \cong \phi^T(x)\, K\, \phi\,(s), \tag{10}$$

where K is a $(n+1)\times(n+1)$ matrix with entries:

$$K_{i',\,j'} = \frac{(B_{i',\,n}(x),\, (\kappa(x,\, s),\, B_{j',\,n}(s)))}{(B_{i',\,n}(x),\, B_{i',\,n}(x))\; (B_{j',\,n}(s),\, B_{j',\,n}(s))}, \quad i',\, j' = 0,\, 1,\, \ldots,\, n, \tag{11}$$

so that (., .) shows the inner product.

2.2 *Operational matrix of integration*

The integration of the vector $\phi(x)$ can be approximated by

$$\int_0^x \phi\,(x')\, dx' \cong\, P\,\phi\,(x), \quad 0 \le x \le 1, \tag{12}$$

where P is an $(n+1)\times(n+1)$ operational matrix for integration that is given by [29]

$$P = A\,\Lambda\, E, \tag{13}$$

in which A, Λ and E are $(n+1)\times(n+1)$ matrices that have the following structures:

$$A = \begin{bmatrix} (-1)^0\binom{n}{0} & (-1)^1\binom{n}{0}\binom{n-0}{1} & (-1)^2\binom{n}{0}\binom{n-0}{2} & \ldots & (-1)^{n-0}\binom{n}{0}\binom{n-0}{n-0} \\ 0 & (-1)^0\binom{n}{1} & (-1)^1\binom{n}{1}\binom{n-1}{1} & \ldots & (-1)^{n-1}\binom{n}{1}\binom{n-1}{n-1} \\ 0 & 0 & (-1)^0\binom{n}{2} & \ldots & (-1)^{n-2}\binom{n}{2}\binom{n-2}{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & (-1)^0\binom{n}{n} \end{bmatrix},$$
$$\Lambda = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & \frac{1}{2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \frac{1}{n+1} \end{bmatrix}, \qquad E = \begin{bmatrix} A_2^{-1} \\ A_3^{-1} \\ \vdots \\ A_{n+1}^{-1} \\ c_{n+1}^T \end{bmatrix}, \tag{14}$$

that $A_{k'+1}^{-1}$ is the $k'+1$-th row of A^{-1} for $k' = 0,\, 1,\, \ldots,\, n$ and

$$c_{n+1} = \frac{Q^{-1}}{2n+2}\left[\, \frac{\binom{n}{0}}{\binom{2n+1}{n+1}}\ \frac{\binom{n}{1}}{\binom{2n+1}{n+2}}\ \cdots\ \frac{\binom{n}{n}}{\binom{2n+1}{2n+1}} \,\right]^T, \tag{15}$$

where Q is a $(n+1)\times(n+1)$ matrix, each element of which is defined as follow:

$$Q_{(i'+1),\,(j'+1)} = \frac{\binom{n}{i'}\binom{n}{j'}}{(2n+1)\binom{2n}{i'+j'}}, \quad i',\, j' = 0,\, 1,\, \ldots,\, n. \tag{16}$$

2.3 *Operational matrix of derivative*

The differentiation of the vector $\phi(x)$ can be approximated by

$$\phi'(x) = D\,\phi\,(x), \quad 0 \le x \le 1, \tag{17}$$

where D is an $(n+1)\times(n+1)$ operational matrix for derivative that is given by [29]

$$D = A\,\Lambda'\,E', \tag{18}$$

in which Λ' and E' are $(n+1)\times n$ and $n\times(n+1)$ matrices that have the following structures:

$$\Lambda' = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & n \end{bmatrix}, \qquad E' = \begin{bmatrix} A_1^{-1} \\ A_2^{-1} \\ A_3^{-1} \\ \vdots \\ A_n^{-1} \end{bmatrix}. \tag{19}$$

2.4 *Operational matrix of product*

Suppose that C is an arbitrary $(n+1)\times 1$ vector, then $\hat{C}$ is an $(n+1)\times(n+1)$ operational matrix of product whenever

$$C^T\phi\,(x)\,\phi^T(x) \cong \phi^T(x)\,\hat{C}, \tag{20}$$

in which

$$\hat{C} = \tilde{C}\,A^T, \tag{21}$$

where

$$\begin{aligned} &\tilde{C} = \left[\tilde{C}_1,\, \tilde{C}_2,\, \ldots,\, \tilde{C}_{n+1}\right], \\ &\tilde{C}_{k'+1} = \left[e_{k',0}\; e_{k',1}\, \ldots\, e_{k',n}\right]\, C,\;\; k' = 0,\, 1,\, \ldots,\, n. \end{aligned} \tag{22}$$

For further information see [29].

3. The method

Consider the system of m-th order nonlinear Volterra integro-differential equations (1) with the initial conditions (2).
First, the unknown functions $u_j(x),\;\; j = 1,\, 2,\, \ldots,\, k$ is approximated by BPs using

Eq. (7) as:

$$u_j(x) \cong u_{j,\,n+1}(x) = U_j^T \phi(x),\ j = 1,\, 2,\, \ldots,\, k,\ 0 \le x \le 1, \tag{23}$$

where U_j is the unknown $(n+1) \times 1$ vector similar to C defined in Eq. (8) and $\phi\,(x)$ is defined in Eq. (9). Likewise, $\{\kappa_{ij}(x,\, s)\}_{i,\,j=1}^{k}$ is also approximated by BPs as:

$$\kappa_{ij}(x,\, s) \cong \phi^{\,T}(x)\, K_{ij} \phi\,(s), \tag{24}$$

where $K_{ij},\ i, j = 1,\, 2,\, \ldots,\, k$ are $(n+1) \times (n+1)$ matrices similar to K defined in Eq. (10). Now, a general formula is presented to approximate $F_{ij}(s,\, U(s)),\ i,\, j = 1,\, 2,\, \ldots,\, k$ by BPs. By using Eqs. (23) and (20) for each element $u_l^{\lambda_l^{ij}}(s),\ i,\, j,\, l = 1,\, 2,\, \ldots,\, k$ of $F_{ij}(s,\, U(s)),\ i,\, j = 1,\, 2,\, \ldots,\, k$, we have

$$\begin{aligned} u_l^2(s) &\cong U_l^T \phi(s)\, \phi^T(s)\, U_l = U_l^T \hat{U}_l^T \phi\,(s), \\ u_l^3(s) &\cong U_l^T \hat{U}_l^T \phi\,(s)\, \phi^T(s)\, U_l = U_l^T (\hat{U}_l^T)^2\, \phi\,(s), \end{aligned} \tag{25}$$

and so by use of induction $u_l^{\lambda_l^{ij}}(s),\ i,\, j,\, l = 1,\, 2,\, \ldots,\, k$ can be approximated by the following:

$$u_l^{\lambda_l^{ij}}(s) \cong U_l^T (\hat{U}_l^T)^{\lambda_l^{ij}-1}\, \phi\,(s) = U_{l\,\lambda_l^{ij}}^T \phi\,(s),\ i,\, j,\, l = 1,\, 2,\, \ldots,\, k. \tag{26}$$

So,

$$\begin{aligned} F_{ij}(s,\, U(s)) &\cong U_{1\,\lambda_1^{ij}}^T \phi\,(s)\, \phi^T(s)\, U_{2\,\lambda_2^{ij}} \phi^T(s)\, U_{3\,\lambda_3^{ij}} \ldots\, \phi^T(s)\, U_{k\,\lambda_k^{ij}} \\ &= U_{1\,\lambda_1^{ij}}^T \hat{U}_{2\,\lambda_2^{ij}}^T \hat{U}_{3\,\lambda_3^{ij}}^T \ldots \hat{U}_{k\,\lambda_k^{ij}}^T \phi\,(s) \\ &= Z_{ij}^T \phi\,(s),\ i,\, j = 1,\, 2,\, \ldots,\, k. \end{aligned} \tag{27}$$

Applying Eqs. (24), (27), (20) and (12) in Volterra integral part of Eq. (1), we get

$$\begin{aligned} \int_0^x \kappa_{ij}(x,\, s)\, F_{ij}(s,\, U(s))\, ds &\cong \int_0^x \phi^T(x)\, K_{ij}\, \phi\,(s)\, \phi^T(s)\, Z_{ij}\, ds \\ &= \phi^T(x)\, K_{ij} \int_0^x \phi\,(s)\, \phi^T(s)\, Z_{ij}\, ds \\ &= \phi^T(x)\, K_{ij} \hat{Z}_{ij}^T \int_0^x \phi\,(s)\, ds \\ &= \phi^T(x)\, K_{ij} \hat{Z}_{ij}^T P\, \phi\,(x). \end{aligned} \tag{28}$$

Also to approximate the differential part of Eq. (1), making use of Eq. (17) we have

$$\begin{aligned} u_j^{(1)}(x) &\cong U_j^T \phi^{(1)}(x) = U_j^T D\, \phi\,(x) = W_{j1}^T\, \phi\,(x), \\ u_j^{(2)}(x) &\cong U_j^T \phi^{(2)}(x) = U_j^T D^2 \phi\,(x) = W_{j2}^T\, \phi\,(x), \\ &\vdots \\ u_j^{(\theta)}(x) &\cong U_j^T \phi^{(\theta)}(x) = U_j^T D^\theta \phi\,(x) = W_{j\theta}^T\, \phi\,(x), \end{aligned} \tag{29}$$

using Eqs. (29) and (26) we obtain

$$\left[u_j^{(\theta)}(x)\right]^{\gamma_{ij}} \cong W_{j\theta}^T \left(\hat{W}_{j\theta}^T\right)^{\gamma_{ij}-1} \phi\,(x). \tag{30}$$

and thus:

$$\sum_{\theta=0}^{m}\sum_{j=1}^{k} g_{ij}^{\theta}(x)\left[u_j^{(\theta)}(x)\right]^{\gamma_{ij}} \cong \sum_{\theta=0}^{m}\sum_{j=1}^{k} g_{ij}^{\theta}(x)\, W_{j\theta}^{T}\left(\hat{W}_{j\theta}^{T}\right)^{\gamma_{ij}-1}\phi\,(x). \tag{31}$$

Now by replacing Eqs. (31) and (28) in Eq. (1), we have

$$\begin{aligned}&\sum_{\theta=0}^{m}\sum_{j=1}^{k} g_{ij}^{\theta}(x)\, W_{j\theta}^{T}\left(\hat{W}_{j\theta}^{T}\right)^{\gamma_{ij}-1}\phi\,(x) - \sum_{j=1}^{k}\phi^{T}(x)\, K_{ij}\hat{Z}_{ij}^{T}P\,\phi\,(x) = f_i(x),\\ &i=1,\,2,\,\ldots,\,k,\quad 0\le x\le 1.\end{aligned} \tag{32}$$

Also, for the initial conditions, after substituting the approximate Eq. (29) into Eq. (2), we have

$$u_j^{(t)}(0) \cong U_j^{T}D^{t}\phi\,(0) = W_{j\,t}^{T}\,\phi\,(0) = \delta_{j,\,t},\;\; t=0,\,1,\,\ldots,\,m-1,\;\; j=1,\,2,\,\ldots,\,k. \tag{33}$$

Eq. (33) provides $k\,m$ linear equations. Since the number of unknowns for each vector U_j in Eq. (32) is $(n+1)$ and the proposed system has k equations, the total number of unknowns are $k\,(n+1)$. Then by using each equation of the system (32) by collocation points $\tau_\upsilon = 2\upsilon - 1/2(m+1),\;\; \upsilon = 1,\,2,\,\ldots,n-m+1$, the following can be derived:

$$\begin{aligned}&\sum_{\theta=0}^{m}\sum_{j=1}^{k} g_{ij}^{\theta}(\tau_\upsilon)\, W_{j\theta}^{T}\left(\hat{W}_{j\theta}^{T}\right)^{\gamma_{ij}-1}\phi\,(\tau_\upsilon) - \sum_{j=1}^{k}\phi^{T}(\tau_\upsilon)\, K_{ij}\hat{Z}_{ij}^{T}P\,\phi\,(\tau_\upsilon) = f_i(\tau_\upsilon),\\ &\upsilon=1,\,2,\,\ldots,\,n-m+1,\;\; i=1,\,2,\,\ldots,\,k,\end{aligned} \tag{34}$$

and hence the Eqs. (34) and (33) are arrived at:

$$\begin{cases}\sum_{\theta=0}^{m}\sum_{j=1}^{k} g_{ij}^{\theta}(\tau_\upsilon)\, W_{j\theta}^{T}\left(\hat{W}_{j\theta}^{T}\right)^{\gamma_{ij}-1}\phi\,(\tau_\upsilon) - \sum_{j=1}^{k}\phi^{T}(\tau_\upsilon)\, K_{ij}\hat{Z}_{ij}^{T}P\,\phi\,(\tau_\upsilon) = f_i(\tau_\upsilon),\\ \upsilon=1,\,2,\,\ldots,\,n-m+1,\;\; i=1,\,2,\,\ldots,\,k,\\ W_{j\,t}^{T}\,\phi\,(0) = \delta_{j,\,t},\;\; t=0,\,1,\,\ldots,\,m-1,\;\; j=1,\,2,\,\ldots,\,k,\end{cases} \tag{35}$$

which corresponds to a system of $k\,(n+1)$ nonlinear algebraic equations with the $k\,(n+1)$ unknown Bernstein coefficients $U_j,\; j=1,\,2,\,\ldots,\,k$. By solving the nonlinear system (35) using the Newton's or fixed point iteration method, the vectors $U_j,\; j=1,\,2,\,\ldots,\,k$ are uniquely specified. Thus, the systems of the m-th order nonlinear Volterra integro-differential equation with variable coefficients (1) under the initial conditions (2) can be solved uniquely by Eq. (23).

4. Accuracy of solution

The accuracy of the method can easily be checked. Since the truncated Bernstein series (23) is the approximate solution of Eq. (1), when the functions $u_{j,\,n+1}(x),\;\; j=1,\,2,\,\ldots,\,k$ and its derivatives, are substituted in Eq. (1), the resulting equation should therefore be approximately satisfied; that is, for $x = x_\vartheta \in [0,\,1];\;\; \vartheta = 0,\,1,\,2,\,...,$

Table 1. The comparison between absolute errors of Example 1 for $u_1(x)$.

x	B-spline wavelet method [24] ($m = 4$)	Legendre wavelet method [24] ($M = 8, k = 2$)	Present method ($n = 14$)
0.0	$1.2218E-04$	$4.6916E-11$	0
0.1	$3.9268E-05$	$3.6593E-13$	0
0.2	$1.4908E-05$	$2.4092E-13$	$2.2204E-16$
0.3	$4.0601E-05$	$2.3670E-13$	$3.1086E-15$
0.4	$3.6491E-05$	$3.8591E-13$	$1.1546E-14$
0.5	$3.4295E-07$	$5.8278E-11$	$4.0856E-14$
0.6	$4.0454E-05$	$5.1581E-13$	$1.8607E-13$
0.7	$4.8619E-05$	$2.0517E-13$	$9.2881E-13$
0.8	$1.8376E-05$	$1.4788E-13$	$4.1858E-12$
0.9	$5.8046E-05$	$7.1054E-13$	$1.5646E-11$
1.0	−	−	$4.6677E-11$

Table 2. The comparison between absolute errors of Example 1 for $u_2(x)$.

x	B-spline wavelet method [24] ($m = 4$)	Legendre wavelet method [24] ($M = 8, k = 2$)	Present method ($n = 14$)
0.0	$1.2218E-04$	$4.6916E-11$	0
0.1	$3.9267E-05$	$3.6515E-13$	$2.2204E-16$
0.2	$1.4920E-05$	$2.4336E-13$	$2.2204E-16$
0.3	$4.0655E-05$	$2.4403E-13$	$2.6645E-15$
0.4	$3.6655E-05$	$3.6804E-13$	$8.4377E-15$
0.5	$2.4367E-08$	$5.8232E-11$	$2.4203E-14$
0.6	$4.1125E-05$	$4.5519E-13$	$9.1704E-14$
0.7	$4.9651E-05$	$3.0687E-13$	$4.2388E-13$
0.8	$1.9757E-05$	$3.1641E-13$	$1.8863E-12$
0.9	$5.6400E-05$	$4.4642E-13$	$7.0188E-12$
1.0	−	−	$2.0020E-11$

$$E_i(x_\vartheta) = \left| \sum_{\theta=0}^{m} \sum_{j=1}^{k} g_{ij}^{\theta}(x_\vartheta) \left[u_j^{(\theta)}(x_\vartheta) \right]^{\gamma_{ij}} - \int_0^{x_\vartheta} \sum_{j=1}^{k} \kappa_{ij}(x_\vartheta,\, s)\, F_{ij}(s,\, U(s))\, ds - f_i(x_\vartheta) \right| \cong 0;$$
$$i = 1,\, 2,\, \ldots,\, k, \tag{36}$$

or

$$E_i(x_\vartheta) \leq 10^{-\theta_\vartheta} \;\; (\theta_\vartheta \; is\, any\, positive\, integer). \tag{37}$$

If max $10^{-\theta_\vartheta} = 10^{-\theta}$ (θ *is any positive* integer), is prescribed, then the truncation limit is increased until the difference $E_i(x_\vartheta)$ at each points x_ϑ becomes smaller than the prescribed $10^{-\theta}$. The error function can thus, be estimated by the following relation:

$$E_{i,\,n+1}(x) = \sum_{\theta=0}^{m} \sum_{j=1}^{k} g_{ij}^{\theta}(x) \left[u_j^{(\theta)}(x) \right]^{\gamma_{ij}} - \int_0^{x} \sum_{j=1}^{k} \kappa_{ij}(x,\, s)\, F_{ij}(s,\, U(s))\, ds - f_i(x) \quad i = 1,\, 2,\, \ldots,\, k. \tag{38}$$

Under the circumstances where $E_{i,\,n+1}(x) \to 0;\; i = 1,\, 2,\, \ldots,\, k$, and when n is sufficiently large enough, the resulting error decreases.

Table 3. The comparison between absolute errors of Example 2 for $u_1(x)$.

x	HPM [9] $(n=5)$	DTM [4] $(N=10)$	B-spline wavelet method [24] $(m=4)$	Legendre wavelet method [24] $(M=8, k=2)$	Present method $(n=14)$
0.0	0	0	$1.2218E-04$	$5.6987E-11$	0
0.1	$0.00E-09$	$2.00E-10$	$4.3169E-05$	$5.0759E-13$	0
0.2	$0.00E-09$	$1.00E-09$	$1.6592E-05$	$1.7697E-13$	$8.8818E-16$
0.3	$0.00E-09$	$0.00E-09$	$4.8574E-05$	$1.0747E-13$	$3.9968E-15$
0.4	$0.00E-09$	$0.00E-09$	$4.1975E-05$	$7.3830E-13$	$1.5099E-14$
0.5	$1.00E-09$	$0.00E-09$	$1.6843E-05$	$9.2065E-11$	$5.3291E-14$
0.6	$5.00E-09$	$0.00E-09$	$3.6679E-05$	$2.8115E-12$	$2.4070E-13$
0.7	$4.30E-08$	$0.00E-09$	$4.6916E-05$	$4.1602E-12$	$1.2030E-12$
0.8	$2.35E-07$	$2.00E-09$	$1.5826E-06$	$6.7928E-12$	$5.4068E-12$
0.9	$5.90E-08$	$9.92E-07$	$1.2841E-04$	$1.0750E-11$	$2.0197E-11$
1.0	$4.09E-06$	$2.70E-08$	$-$	$-$	$6.0509E-11$

Table 4. The comparison between absolute errors of Example 2 for $u_2(x)$.

x	HPM [9] $(n=5)$	DTM [4] $(N=10)$	B-spline wavelet method [24] $(m=4)$	Legendre wavelet method [24] $(M=8, k=2)$	Present method $(n=14)$
0.0	0	0	$1.2218E-04$	$5.7167E-11$	0
0.1	$0.00E-09$	$0.00E-09$	$4.2952E-05$	$4.4920E-13$	$4.4409E-16$
0.2	$0.00E-09$	$1.00E-09$	$1.7640E-05$	$3.0198E-13$	0
0.3	$0.00E-09$	$0.00E-09$	$5.1682E-05$	$3.0509E-13$	$3.7748E-15$
0.4	$0.00E-09$	$0.00E-09$	$4.9608E-05$	$4.6518E-13$	$1.0658E-14$
0.5	$1.00E-09$	$0.00E-09$	$4.0220E-08$	$9.4255E-11$	$3.4417E-14$
0.6	$1.00E-09$	$0.00E-09$	$6.4396E-05$	$8.0980E-13$	$1.4366E-13$
0.7	$1.00E-09$	$0.00E-09$	$8.1488E-05$	$3.5816E-13$	$6.9633E-13$
0.8	$2.00E-09$	$2.00E-09$	$3.4912E-05$	$2.7312E-13$	$3.1188E-12$
0.9	$1.30E-08$	$8.00E-09$	$9.4895E-05$	$1.1238E-12$	$1.1656E-11$
1.0	$6.00E-08$	$2.70E-08$	$-$	$-$	$3.4503E-11$

5. Illustrative examples

The method is applied to solve four different examples, for all the computations of which were carried out by the Matlab 7.6 on a PC computer. Tables show the values of the absolute error $|u_i(x) - u_{i,n+1}(x)|$, $i = 1, 2, \ldots, k$ at the selected points of the interval.

Example 1. Consider first system of the second order nonlinear Volterra integro-differential equations [24, 27]:

$$\begin{cases} u_1''(x) - \int_0^x \left((x-s)\,u_1^2(s) + (x-s)\,u_2^2(s)\right) ds = f_1(x), \\ u_2''(x) - \int_0^x \left((x-s)\,u_1^2(s) - (x-s)\,u_2^2(s)\right) ds = f_2(x), \end{cases} \tag{39}$$

where $f_1(x) = \cosh x - \frac{1}{2}\sinh^2 x - \frac{1}{6}x^4 - \frac{1}{2}x^2$ and $f_2(x) = -(1+4x)\cosh x + 8\sinh x - 4x$ with the initial conditions $u_1(0) = 1$, $u_1'(0) = 1$, $u_2(0) = -1$, $u_2'(0) = 1$. The exact solutions are $u_1(x) = x + \cosh x$ and $u_2(x) = x - \cosh x$. This is solved for $n = 14$ by the presented method. The values of absolute error are tabulated in Tables 1 and 2. These results are compared with the B-spline wavelet method and the Legendre wavelet method [24] for $m = 4$ and $M = 8$, $k = 2$, respectively. As can be observed, BPs is more accurate than the B-spline wavelet method and the Legendre wavelet for the less basis functions. Moreover, for this example, we calculate the values of error function $E_{i,n+1}(x)$, $i = 1, 2$ for $n = 14$. The relevant results are tabulated in Table 5. From this Table it follows that max $10^{-\theta_\vartheta}$ for example 1 is 10^{-9}.

Example 2. Consider the system of the second order nonlinear Volterra

Table 5. The error functions $E_{i,\,n+1}(x)$, $i = 1, 2$ of examples 1 and 2.

	Example 1 ($n = 14$)		Example 2 ($n = 14$)	
	$E_{1,\,n+1}(x)$	$E_{2,\,n+1}(x)$	$E_{1,\,n+1}(x)$	$E_{2,\,n+1}(x)$
0.0	0	0	0	0
0.1	$4.6934E-14$	$3.9295E-14$	$2.5091E-14$	$5.8731E-14$
0.2	$1.6601E-13$	$1.6728E-13$	$2.8341E-13$	$1.7279E-13$
0.3	$5.2997E-13$	$3.4694E-13$	$7.2051E-13$	$4.5282E-13$
0.4	$1.6431E-12$	$8.0869E-13$	$2.1608E-12$	$1.2640E-12$
0.5	$9.0894E-12$	$4.0445E-12$	$1.2036E-11$	$6.7370E-12$
0.6	$4.9549E-11$	$2.1879E-11$	$6.5964E-11$	$3.6528E-11$
0.7	$2.2078E-10$	$9.9024E-11$	$2.9572E-10$	$1.6331E-10$
0.8	$7.6455E-10$	$3.4528E-10$	$1.0383E-09$	$5.6675E-10$
0.9	$1.9428E-09$	$8.2146E-10$	$2.7245E-09$	$1.4184E-09$
1.0	$3.1337E-09$	$7.2811E-10$	$4.7448E-09$	$2.0490E-09$

Table 6. The comparison between absolute errors of Example 3 for $u_1(x)$.

x	B-spline wavelet method [24] ($m = 2$)	Legendre wavelet method [24] ($M = 4, k = 2$)	HPM [9] ($n = 6$)	operational Tau method [1] ($n = 10$)	Present method ($n = 7$)	Present method ($n = 14$)
0.0	$2.0842E-09$	$1.6979E-05$	0	0	$4.2861E-21$	$4.8602E-23$
0.1	$2.6564E-04$	$9.2899E-07$	$1.0000E-09$	$1.39E-17$	$2.3282E-09$	$2.9116E-14$
0.2	$2.8430E-04$	$1.5711E-06$	$1.3300E-07$	$5.27E-16$	$1.5954E-09$	$3.0476E-14$
0.3	$2.1122E-04$	$2.1762E-06$	$2.1750E-06$	$4.45E-14$	$1.6960E-09$	$2.9643E-14$
0.4	$1.9604E-04$	$1.2493E-06$	$1.5309E-05$	$1.05E-12$	$2.1325E-09$	$2.7756E-14$
0.5	$3.9319E-04$	$6.6341E-05$	$6.6846E-05$	$1.23E-11$	$2.2922E-09$	$2.2871E-14$
0.6	$2.3937E-05$	$5.1247E-06$	$2.1219E-04$	$9.11E-11$	$2.1033E-09$	$1.3434E-14$
0.7	$1.8552E-04$	$1.4895E-05$	$5.2779E-04$	$4.97E-10$	$2.3260E-09$	$2.0117E-13$
0.8	$1.2301E-04$	$1.6418E-05$	$1.0569E-03$	$2.16E-09$	$3.6133E-09$	$1.0212E-12$
0.9	$1.1484E-03$	$6.4467E-06$	$1.6609E-03$	$7.90E-09$	$1.7525E-09$	$3.8809E-12$
1.0	−	−	$1.7136E-03$	$2.52E-08$	$5.7928E-08$	$1.0573E-11$

integro-differential equations as following [4, 9, 24]:

$$\begin{cases} u_1''(x) + \frac{1}{2}\, u_2'^2(x) - \frac{1}{2}\int_0^x (u_1^2(s) + u_2^2(s))\, ds = f_1(x), \\ u_2''(x) + x\, u_1(x) - \frac{1}{4}\int_0^x (\, u_1^2(s) - u_2^2(s))\, ds = f_2(x), \end{cases} \tag{40}$$

where $f_1(x) = 1 - \frac{1}{3}x^3$ and $f_2(x) = -1 + x^2$ with the initial conditions $u_1(0) = 1,\, u_1'(0) = 2,\, u_2(0) = -1,\, u_2'(0) = 0$. The exact solutions are $u_1(x) = x + e^x$ and $u_2(x) = x - e^x$. This example is solved by using the method described in Section 3 with $n = 14$. The comparison among absolute error functions obtained by present method, HPM [9] for $n = 5$, DTM [4] for $N = 10$, B-spline wavelet method [24] for $m = 4$ and Legendre wavelet method [24] for $M = 8$, $k = 2$ are shown in Tables 3 and 4. As can be seen form Tables 3 and 4, the absolute error functions of HPM [9] are lower than B-spline wavelet method [24], the absolute error functions of DTM [4] are lower than HPM [9], the absolute error functions of Legendre wavelet method [24] are lower than DTM [4] and those obtained by the present method are superior to that by [4, 9, 24]. Moreover, for this example, we calculate the values of error function $E_{i,\,n+1}(x)$, $i = 1, 2$ for $n = 14$. The relevant results are tabulated in Table 5. From this Table it follows that max $10^{-\theta_\vartheta}$ for example 2 is 10^{-9}.

Example 3. Consider system of the nonlinear Volterra integro-differential equations as following [1, 9, 24]:

$$\begin{cases} u'_1(x) + \frac{1}{2}\, {u'_2}^2(x) - \int_0^x ((x-s)\, u_2(s) + u_2(s)u_1(s))\, ds = f_1(x), \\ u'_2(x) - \int_0^x ((x-s)\, u_1(s) - u_2^2(s) + u_1^2(s))\, ds = f_2(x), \end{cases} \tag{41}$$

where $f_1(x) = 1$ and $f_2(x) = 2x$ with the initial conditions $u_1(0) = 0$, $u_2(0) = 1$.

Table 7. The comparison between absolute errors of Example 3 for $u_2(x)$.

x	B-spline wavelet method [24] ($m = 2$)	Legendre wavelet method [24] ($M = 4, k = 2$)	HPM [9] ($n = 6$)	operational Tau method [1] ($n = 10$)	Present method ($n = 7$)	Present method ($n = 14$)
0.0	$1.9812E-03$	$6.8259E-05$	0	0	0	0
0.1	$6.3480E-04$	$1.0303E-05$	$1.3000E-08$	0	$1.2185E-09$	$1.0436E-14$
0.2	$2.5790E-04$	$1.4987E-06$	$7.9800E-07$	0	$1.0529E-09$	$9.9920E-15$
0.3	$7.0502E-04$	$1.3195E-06$	$9.0680E-06$	$1.11E-15$	$1.1130E-09$	$1.2212E-14$
0.4	$6.6797E-04$	$1.1008E-05$	$5.0626E-05$	$3.51E-14$	$8.4556E-10$	$1.7986E-14$
0.5	$5.8899E-05$	$8.3081E-05$	$1.9090E-04$	$5.10E-13$	$9.1503E-10$	$3.7748E-14$
0.6	$6.7631E-04$	$1.3848E-05$	$5.5873E-04$	$4.55E-12$	$9.9562E-10$	$1.3878E-13$
0.7	$7.8339E-04$	$8.5961E-07$	$1.3630E-03$	$2.90E-11$	$6.5407E-10$	$6.5814E-13$
0.8	$2.8238E-04$	$9.6797E-07$	$2.8783E-03$	$1.44E-10$	$1.1548E-09$	$2.9399E-12$
0.9	$9.8303E-04$	$1.4032E-05$	$5.3496E-03$	$5.92E-10$	$5.2941E-09$	$1.0934E-11$
1.0	−	−	$8.7171E-03$	$2.10E-09$	$8.4230E-08$	$3.1936E-11$

Table 8. The comparison between absolute errors of Example 4 for $u_1(x)$.

x	HPM [10] ($n = 5$)	Method based upon discretisation [28] ($n = 200$)	Simpson's 3/8 rule [17] ($h = 0.025$)	Present method ($n = 10$)
0.0	0	$3.1874E-22$	−	$3.3420E-09$
0.1	$1.4E-07$	$2.3788E-05$	$3.0E-10$	$5.5234E-10$
0.2	$3.5E-06$	$9.8303E-05$	$1.1E-09$	$2.7470E-10$
0.3	$5.5E-05$	$2.2569E-04$	$3.6E-09$	$2.5776E-10$
0.4	$3.8E-04$	$4.0980E-04$	$6.0E-09$	$1.0119E-10$
0.5	$1.6E-03$	$6.5639E-04$	$8.7E-09$	$1.1451E-10$
0.6	−	$9.7347E-04$	$1.4E-08$	$2.4562E-10$
0.7	−	$1.3717E-03$	$1.9E-08$	$5.7598E-10$
0.8	−	$1.8647E-03$	$2.4E-08$	$1.0330E-08$
0.9	−	$2.4695E-03$	$3.3E-08$	$1.0439E-07$
1.0	−	$3.2073E-03$	$4.0E-08$	$8.2416E-07$

Table 9. The comparison between absolute errors of Example 4 for $u_2(x)$.

x	HPM [10] ($n = 5$)	Method based upon discretisation [28] ($n = 200$)	Simpson's 3/8 rule [17] ($h = 0.025$)	Present method ($n = 10$)
0.0	$5.0E-08$	0.0	−	$2.7592E-09$
0.1	$3.2E-07$	$2.4986E-04$	$5.3E-10$	$6.4585E-11$
0.2	$1.1E-05$	$4.9945E-04$	$3.0E-10$	$1.2913E-10$
0.3	$1.2E-04$	$7.4893E-04$	$6.0E-10$	$6.4527E-12$
0.4	$6.3E-04$	$9.9871E-04$	$2.1E-09$	$6.5267E-11$
0.5	$2.2E-03$	$1.2496E-03$	$3.1E-09$	$9.0182E-11$
0.6	−	$1.5027E-03$	$5.3E-09$	$4.2038E-11$
0.7	−	$1.7597E-03$	$9.4E-09$	$1.0103E-10$
0.8	−	$2.0224E-03$	$1.4E-08$	$3.5706E-09$
0.9	−	$2.2928E-03$	$2.0E-08$	$3.5423E-08$
1.0	−	$2.5721E-03$	$2.9E-08$	$2.8554E-07$

The exact solutions are $u_1(x) = \sinh x$ and $u_2(x) = \cosh x$. This example is solved by using the method described in Section 3 with $n = 7, 14$. The comparison among absolute error functions obtained by present method, B-spline wavelet method [24] for $m = 2$, Legendre wavelet method [24] for $M = 4$, $k = 2$, HPM [9] for $n = 6$ and the operational Tau method [1] for $n = 10$ are exhibited in Tables 6 and 7. As can be seen form Tables 6 and 7, the absolute error functions of HPM [9] are lower than B-spline wavelet method [24], the absolute error functions of Legendre wavelet method [24] are lower than HPM [9], the absolute error functions of operational Tau method [1] are lower than Legendre wavelet method [24] and those obtained by the present method are more accurate than [1, 9, 24]. Moreover, for this example, we calculate the values of error function $E_{i,n+1}(x)$, $i = 1, 2$ for $n = 14$. The relevant results are tabulated in Table 10. From this Table it follows that max $10^{-\theta_\vartheta}$ for example 3 is 10^{-10}.

Table 10. The error functions $E_{i,\,n+1}(x)$, $i = 1, 2$ of examples 3 and 4.

	Example 3 ($n = 14$)		Example 4 ($n = 10$)	
	$E_{1,\,n+1}(x)$	$E_{2,\,n+1}(x)$	$E_{1,\,n+1}(x)$	$E_{2,\,n+1}(x)$
0.0	$3.2205E-12$	$1.1751E-12$	$3.3420E-09$	$2.7592E-09$
0.1	$4.2061E-15$	$3.0587E-15$	$4.9968E-10$	$1.3933E-11$
0.2	$1.8041E-14$	$4.8965E-15$	$3.1380E-10$	$7.9255E-11$
0.3	$2.5536E-14$	$4.0839E-14$	$2.3284E-10$	$5.8282E-11$
0.4	$7.1261E-14$	$1.1040E-13$	$1.3501E-10$	$2.6704E-11$
0.5	$3.3753E-13$	$4.1760E-13$	$1.1155E-10$	$2.9086E-11$
0.6	$2.0861E-12$	$2.1274E-12$	$2.3791E-10$	$9.4188E-11$
0.7	$1.1475E-11$	$1.0378E-11$	$5.3061E-10$	$1.2986E-10$
0.8	$5.1585E-11$	$4.2076E-11$	$9.6116E-09$	$3.2574E-09$
0.9	$1.8116E-10$	$1.3319E-10$	$9.5513E-08$	$3.1349E-08$
1.0	$4.0733E-10$	$3.0144E-10$	$7.4290E-07$	$2.5121E-07$

Example 4. Consider system of nonlinear Volterra integral equations [10, 17, 28]:

$$\begin{cases} u_1(x) - \int_0^x (u_1^2(s) + u_2^2(s))\, ds = f_1(x), \\ u_2(x) - \int_0^x u_1(s)\, u_2(s)\, ds = f_2(x), \end{cases} \tag{42}$$

where $f_1(x) = \sin x - x$ and $f_2(x) = \cos x - \frac{1}{2}\sin^2 x$. The exact solutions are $u_1(x) = \sin x$ and $u_2(x) = \cos x$. This example is solved by using the method described in Section 3 with $n = 10$. The comparison among absolute error functions obtained by present method, HPM [10] for $n = 5$, method based upon discretization [28] for $n = 200$ and the Simpsons 3/8 rule [17] for $h = 0.025$ are exhibited in Tables 8 and 9. As can be seen form Tables 8 and 9, the absolute error functions of HPM [10] are lower than the method based upon discretization [28], the absolute error functions of Simpsons 3/8 rule [17] are lower than the method based upon discretization [28] and those obtained by the present method are more accurate than [10, 17, 28]. Moreover, for this example, we calculate the values of error function $E_{i,n+1}(x)$, $i = 1, 2$ for $n = 10$. The relevant results are tabulated in Table 10. From this Table it follows that max $10^{-\theta_\vartheta}$ for example 2 is 10^{-7}.

6. Conclusion

A numerical method for solving the system of m-th order nonlinear Volterra integro-differential equations with variable coefficients is proposed which based on the BPs basis. One of the most important features of this method is the application of the computer program to find the BPs coefficients of the solution. The approach can further be extended to systems of partial differential equations.

Acknowledgments

The authors are grateful to the respected editor and referees for his (her) valuable comments and suggestions that improve the content of the manuscript.

References

[1] S. Abbasbandy, A. Taati, *Numerical solution of the system of nonlinear Volterra integro-differential equations with nonlinear differential part by the operational Tau method and error estimation*, Journal of Computational and Applied Mathematics, 231 (2009) 106–113.

[2] Ch.S. Ahmed, *Approximate solution for the system of nonlinear Volterra integral equations of the second kind by five block by block method*, Open Science Repository Mathematics, doi: 10.7392/openaccess. 70081992.

[3] H. Almasieh, *A direct method for the system of integro differential equations*, In: Proceedings of the annual international conference on computational mathematics, computational geometry and statistics, (CMCGS 2012), Singapore, (2012).

[4] A. Arikoglu, I. Ozkol, *Solutions of integral and integro-differential equation systems by using differential transform method*, Computers and Mathematics with Applications, 56 (2008) 2411–2417.

[5] E. Babolian, J. Biazar, *Solutions of a system of nonlinear Volterra equations of the second kind*, Far East Journal of Mathematical Sciences, 2 (2000) 935–945.

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

[7] M. I. Bhatti, P. Bracken, *Solutions of differential equations in a Bernstein polynomial basis*, Journal of Comput. and Appl. Math., 205 (2007) 272–280.

[8] J. Biazar, E. Babolian, R. Islam, *Solution of a system of Volterra integral equations of the first kind by Adomian method*, Appl. Math. Comput., 139 (2003) 249–258.

[9] J. Biazar, H. Ghazvini, M. Eslami, *He's homotopy perturbation method for systems of integro-differential equations*, Chaos Solitons. Fractals, 39 (2009) 1253–1258.

[10] J. Biazar, H. Ghazvini, *Hes homotopy perturbation method for solving systems of Volterra integral equations of the second kind*, Chaos, Solitons and Fractals, 39 (2009) 770–777.

[11] E. Celik, Kh. Tabatabaei, *Solving a class of Volterra integral equation systems by the differential transform method*, International Journal of Nonlinear Science, 16 (2013) 87–91.

[12] E.H. Doha, A.H. Bhrawy, M.A. Saker, *Integrals of Bernstein polynomials: an application for the solution of high even-order differential equations*, Applied Mathematics Letters, 24 (2011) 559–565.

[13] S.M. El-Sayed, D. Kaya, S. Zarea, *The decomposition method applied to solve high-order linear Volterra-Fredholm integro-differential equations*, International Journal of Nonlinear Sciences and Numerical Simulation, 5 (2004) 105–112.

[14] R. Ezzati, S. Najafalizadeh, *Application of Chebyshev polynomials for solving nonlinear Volterra-Fredholm integral equations system and convergence analysis*, Indian Journal of Science and Technology, 5 (2012) 2060–2064.

[15] R.T. Farouki, V.T. Rajan, *Algorithms for polynomials in Bernstein form*, Computer Aided Geometric Design, 5 (1988) 1–26.

[16] A. Jafarian, S. Nia, A. Golmankhaneh, D. Baleanu, *Numerical solution of linear integral equations system using the Bernstein collocation method*, Advances in Difference Equations, (2013).

[17] A. Kilicman, L. Kargaran Dehkordi, M. Tavassoli Kajani, *Numerical solution of nonlinear Volterra integral equations system using Simpson's 3/8 rule*, Mathematical Problems in Engineering, (2012).

[18] B.N. Mandal, S. Bhattacharya, *Numerical solution of some classes of integral equations using Bernstein polynomials*, Appl. Math. Comput., 190 (2007) 1707–1716.

[19] J. Rashidinia, A. Tahmasebi, S. Rahmany, *A reliable treatment for nonlinear Volterra integro-differential equations*, Journal of Information and Computing Science, 9 (2014) 003–010.

[20] J. Rashidinia, A. Tahmasebi, *Systems of nonlinear Volterra integro-differential equations*, Numerical Algorithms, 59 (2012) 197–212.

[21] J. Saberi-Nadjafi, M. Tamamgar, *The variational iteration method: A highly promising method for solving the system of integro-differential equations*, Computers and Mathematics with Applications, 56 (2008) 346–351.

[22] H. Sadeghi Goghary, Sh. Javadi, E. Babolian, *Restarted Adomian method for system of nonlinear Volterra integral equations*, Appl. Math. Comput., 161 (2005) 745–751.

[23] P.K. Sahu, S. Saha Ray, *A new numerical approach for the solution of nonlinear Fredholm integral equations system of second kind by using Bernstein collocation method*, Mathematical Methods in Applied Sciences, 38 (2015) 274–280.

[24] P.K. Sahu, S. Saha Ray, *Legendre wavelets operational method for the numerical solutions of nonlinear Volterra integro-differential equations system*, Applied Mathematics and Computation, 256 (2015) 715–723.

[25] N.H. Sweilam, *Fourth order integro-differential equations using variational iteration method*, Computers and Mathematics with Applications, doi:10.1016/j.camwa.2006.12.055.

[26] S.Q. Wang, J.H. He, *Variational iteration method for solving integro-differential equations*, Physics Letters, 367 (2007) 188–191.

[27] A.M. Wazwaz, *Linear and Nonlinear Integral Equations*, Methods and Applications, Springer-Verlag, Beijing, (2011).

[28] M.R. Yaghouti, *A numerical method for solving a system of Volterra integral equations*, World Applied Programming, 2 (2012) 18–33.

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

[30] E. Yusufoglu, *An efficient algorithm for solving integro-differential equations system*, Appl. Math. Comput., 192 (2007) 51–55.