Journal of Mahani Mathematical Research
Shahid Bahonar University of Kerman
Print ISSN: 2251-7952 Online ISSN: 2645-4505

# AN OPERATIONAL COLLOCATION BASED ON THE BELL POLYNOMIALS FOR SOLVING HIGH ORDER VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS

N. Kasaei, E. Hesameddini ✉, and M. Nabati

Article type: Research Article
(Received: 05 July 2023, Received in revised form 31 October 2023)
(Accepted: 21 November 2023, Published Online: 21 November 2023)

Abstract. In this paper, an operational matrix method based on the Bell polynomials has been presented to find approximate solutions of high-order Volterra integro-differential equations. This method uses a simple computational manner to obtain a quite acceptable approximate solution. The main characteristic behind this method lies in the fact that on the one hand, the problem will be reduced to a system of algebraic equations and on the other hand, the efficiency and accuracy of the Bell polynomials for solving these equations are acceptable. The convergence analysis of this method will be shown by preparing some theorems. Moreover, we will obtain an estimation of the error bound for this algorithm. Finally, some examples are presented to illustrate the applicability, efficiency and accuracy of this scheme in comparison with some other well-known methods such as Legendre, Bernoulli, Taylor and Bessel polynomial algorithms.

*Keywords*: Volterra Integro-Differential Equations, Bell Polynomials, Operational Matrix, Error Estimation.
*2020 MSC*: Primary 45G10, 65R20, 11B73.

## 1. **Introduction**

Integro-differential equations appeared in many physical applications such as glassforming process, nanohydrodinamics, heat transfer, diffusion process in general, neutron diffusion and biological species coexisting together with increasing and decreasing rates of generating, and wind ripple in the desert [21]. Many numerical methods have been used for solving these equations such as the Bernestein polynomials method [9], Bernoulli polynomials scheme [1, 3, 13], Legendre method [17, 20], Bernoulli matrix method [2], collocation methods [4, 33], meshless methods [5], spectral methods [6], Taylor polynomial method [11, 23], He's homotopy perturbation [19] and so on.

Also, in [29], the authors utilized Muntz-Legendre polynomials to solve linear delay Fredholm integro-differential equations. In [27], the Laguerre approach was employed to solve pantograph-type Volterra integro-differential equation.

✉ hesameddini@sutech.ac.ir, ORCID: 0000-0002-7012-1161

https://doi.org/10.22103/jmmr.2023.21830.1469
Publisher: Shahid Bahonar University of Kerman
How to cite: N. Kasaei, E. Hesameddini, M. Nabati, *An operational collocation based on the Bell polynomials for solving high order Volterra integro-differential equations*, J. Mahani Math. Res. 2024; 13(2): 121 - 139.

In [31], the Galerkin-type method was used to solve high-order linear delay Volterra integro-differential equations. In [30], the Taylor operational matrix method was used to solve linear Fredholm-Volterra integro-differential equations. In [34], the authors applied the Pell-Lucass collocation method to solve high order linear Fredholm-Volterra integro-differential equations. In [18], Saadatmandi and Dehghan used Legendre polynomials in order to solve the linear Fredholm integro-differential-difference equation of high order. In [15], the Taylor method was employed to solve nonlinear Fredholm integro-differential equations with time delay. In [32], the authors proposed an algorithm by using the Bessel polynomial for the high order linear Volterra integro-differential equations. In [10], Maleknejad et al. used a Bernstein operational matrix approach for solving a system of high order linear Volterra-Fredholm integro differential equations. In [28], the Legendre method was employed to solve delay linear Fredholm integro-differential equations. Moreover, in [8], Hesameddini and Shahbazi used the Bernstein polynomials for solving Fredholm integro-differential-difference equations.

In recent years, the Bell polynomials have been extensively used for solving problems formulated by mixed Fredholm-Volterra integral equations [14], Fredholm integro-differential equations [24], fractional integro-differential equation [26], differential equations [16] and fractional differential equation [25]. The Bell polynomials were given by Eric Temple Bell in 1934 [22]. These polynomials naturally occur from differentiating a composite function several times. The Bell polynomials have many applications in number theory, classical analysis, combinatorial analysis and statistics [14].

This article concerns the following high-order Volterra integro-differential equation in the form of

$$\sum_{j=0}^{m} c_j y^{(j)}(x) = f(x) + \int_0^x k(x,s)y(s)ds, \quad 0 \le x \le 1, \tag{1}$$

with initial conditions

$$y^{(j)}(0) = \alpha_j,\ j = 0, 1, ..., m-1, \tag{2}$$

where $y(x)$ and$f(x)$ are continuous differentiable functions of desired order, $k(x,s)$ is a separable kernel, $c_j's$ are constant coefficients and $y^{(j)}(x) = \frac{d^j}{dx^j}y(x)$. In this paper, we suggest an efficient method based on the Bell polynomials for solving high-order Volterra integro-differential equations. The basic idea is to approximate the solution of Eq. (1) via the Bell polynomials. The properties of the Bell polynomials are used to convert the equation into a system of algebraic equations. This makes the problem easy for programming software. Furthermore, the coefficients matrix of the Bell polynomials are always nonsingular and this allows us to search for the solution in polynomial form. Moreover, the operational matrix of integration is sparse and this is another advantage of the Bell polynomials for solving these equations.

This article is classified as follows: in Section 2, we have presented a matrix representation of the Bell polynomials. Section 3 is related to the approximation of a function by the Bell polynomials. In Section 4, the operational matrix of integration is presented. The analysis of this method is explained in Section 5 and Section 6 is devoted to the study of the convergence and error estimation of this scheme. In Section 7, some numerical examples to illustrate the efficiency of this algorithm are given. Also, the comparison of the method with some other well-known methods is shown. A brief conclusion is given in Section 8.

## 2. Bell polynomials

In mathematics, the Bell polynomials are used in the study of set partitions. Also, they occur in many applications such as the Blissard problem, the representation of Lucas polynomials of the first and second kinds, the recurrence relations for a class of Freud-type polynomials, the representation of symmetric functions of a countable set of numbers and in generalization of the classical algebraic Newton-Girard formulas [14].
These polynomials can be expressed in some ways. It can be written as a series expansion of generating exponential function and may be given by the second kind of Stirling numbers.

**Definition 2.1.** The Bell polynomials can be computed as

$$B_n(x) = \sum_{k=0}^{n} s(n,k)x^k,$$

where $s(n,k)$ for $k = 0, 1, ..., n$ are the Stirling numbers of the second kind which are

$$s(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n. \tag{3}$$

Therefore, the Bell polynomials can be approximated as

$$B(x) = SX(x), \tag{4}$$

where

$$B(x) = [B_0(x), B_1(x), ..., B_N(x)]^T,$$

and

$$X(x) = [1, x, x^2, ..., x^N]^T,$$

also,

$$S = \begin{pmatrix} s(0,0) & 0 & \cdots & 0 \\ s(1,0) & s(1,1) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ s(N,0) & s(N,1) & \cdots & s(N,N) \end{pmatrix}. \tag{5}$$

One can obtain the entries of this matrix by using equation (3). For example, in the case of $N = 3$, we get

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 3 & 1 \end{pmatrix}.$$

So, the first few Bell polynomials are

$$B_0(x) = 1,\ B_1(x) = x,\ B_2(x) = x + x^2,\ B_3(x) = x + 3x^2 + x^3.$$

Since matrix S is a lower triangular matrix with nonzero diagonal elements, so this matrix is nonsingular and hence $S^{-1}$ exists. Now, using (4), we have

$$X(x) = S^{-1}B(x). \tag{6}$$

## 3. **Function approximation**

Suppose $\mathcal{H} = L^2[0,1]$, $\{B_0(x), B_1(x), ..., B_N(x)\} \subset \mathcal{H}$ be the set of Bell polynomials and

$$\mathcal{S} = span\{B_0(x), B_1(x), ..., B_N(x)\}.$$

If $f$ is any member in $\mathcal{H}$, since $\mathcal{S}$ is a finite dimensional vector space, so $f$ has the best approximation out of $\mathcal{S}$ such as $\hat{f} \in \mathcal{S}$. This means that [33]

$$\forall g \in \mathcal{S}\ \|f - \hat{f}\| \leq \|f - g\|.$$

Since $\hat{f} \in \mathcal{S}$, there exist unique coefficients $f_0, f_1, ..., f_N$ such that

$$f(x) \simeq \hat{f}(x) = \sum_{i=0}^{N} f_i B_i(x) = F^T B(x), \tag{7}$$

in which

$$B^T(x) = [B_0(x), B_1(x), ..., B_N(x)],$$

and

$$F^T = [f_0, f_1, ..., f_N]. \tag{8}$$

For computing the coefficients $f_i$, we let

$$h_j = \int_0^1 f(x)B_j(x)dx,\ j = 0, 1, \ldots, N.$$

So, we have

$$h_j = \int_0^1 \sum_{i=0}^{N} f_i B_i(x) B_j(x) dx$$

$$= \sum_{i=0}^{N} f_i \int_0^1 B_i(x) B_j(x) dx = \sum_{i=0}^{N} f_i q_{ij}, \; j = 0, 1, \ldots, N, \tag{9}$$

where

$$q_{ij} = \int_0^1 B_i(x) B_j(x) dx. \tag{10}$$

Also, let

$$Q = [q_{ij}]_{(N+1)\times(N+1)}, \; \textit{and} \; H = [h_0, h_1, \ldots, h_N]^T.$$

From relation (9), we have

$$H^T = F^T Q \Rightarrow F^T = H^T Q^{-1}. \tag{11}$$

Also, the kernel $k(x, s)$ is separable bivariable function. Therefore, it can be written as

$$k(x, s) = f^T(x) g(s).$$

Doing the same procedure as (7) up to (11) for the one variable function $f(x)$ and $g(s)$, one obtains

$$k(x, s) \simeq k_N(x, s) = B^T(x) K B(s), \tag{12}$$

where $K = [k_{ij}]$ is an $(N+1)\times(N+1)$ matrix in which $k_{ij}$ can be obtained as

$$K = Q^{-1} \left[ \int_0^1 \int_0^1 k(x, s) B(x) B(s) dx ds \right] Q^{-1}. \tag{13}$$

## 4. **Operational matrix**

Using the definition of standard basis, we have

$$\int_0^x X(y)dy = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{N} \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{N-1} \\ x^N \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \frac{x^{N+1}}{N+1}$$

$$= MX(x) + \frac{x^{N+1}}{N+1} I_{N+1}. \tag{14}$$

Omitting the second term of (14), one can approximate the integration of the vector $X(x)$ as follows

$$\int_0^x X(y)dy \simeq MX(x). \tag{15}$$

By using (4), (6) and (15), the operational matrix of integration based on the Bell polynomials will be obtained as

$$\int_0^x B(y)dy = S\int_0^x X(y)dy = SMX(x) = SMS^{-1}B(x) = PB(x). \tag{16}$$

The matrix $P$ in (16), is called $(N+1)\times(N+1)$ operational matrix of integration.
The operational matrix of integration, $P$, is a sparse matrix, which is one of the advantages of using the Bell polynomials for solving equations under study. As an example, for $N=3$, $P$ is

$$P = SMS^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 3 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \dfrac{1}{3} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 2 & -3 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{-1}{2} & \dfrac{1}{2} & 0 \\ 0 & \dfrac{1}{6} & \dfrac{-1}{2} & \dfrac{1}{3} \\ 0 & \dfrac{3}{2} & \dfrac{-5}{2} & 1 \end{pmatrix}.$$

Also, by computing the operational matrix for higher degrees, it can be seen that as $N$ increases, the matrix becomes more sparse.

## 5. **Method of solution**

For solving the Volterra integro-differential equation (1), we can approximate the unknown function $y^{(m)}(x)$ by the Bell matrix as

$$y^{(m)}(x) = A_m^T B(x), \tag{17}$$

where

$$A_m^T = [a_0, a_1, ..., a_N],$$

is an unknown vector. Integrating (17), one obtains

$$\int_0^x y^m(t)dt = \int_0^x A_m^T B(t)dt,$$
$$y^{m-1}(x) - y^{m-1}(0) = A_m^T \int_0^x B(t)dt,$$
$$y^{m-1}(x) = \alpha_{m-1} + A_m^T P B(x),$$
$$y^{m-2}(x) = \alpha_{m-2} + \alpha_{m-1}x + A_m^T P^2 B(x),$$
$$y^{m-3}(x) = \alpha_{m-3} + \alpha_{m-2}x + \frac{1}{2}\alpha_{m-1}x^2 + A_m^T P^3 B(x),$$
$$\vdots$$
$$y^{m-k}(x) = \alpha_{m-k} + \alpha_{m-k+1}x + \frac{1}{2!}\alpha_{m-k+2}x^2 + \frac{1}{3!}\alpha_{m-k+3}x^3 + ...+$$
$$\frac{1}{(k-1)!}\alpha_{m-k+(k-1)}x^{k-1} + A_m^T P^k B(x), \tag{18}$$

Integrating $m$ times of (17), results in

$$y(x) = \alpha_0 + \alpha_1 x + \frac{1}{2!}\alpha_2 x^2 + \frac{1}{3!}\alpha_3 x^3 + ... + \frac{1}{(m-1)!}\alpha_{m-1}x^{m-1}$$
$$+ A_m^T P^m B(x). \tag{19}$$

As an example, for $m = 3$ we have

$$y^{(3)}(x) = A_3^T B(x),$$

and by integrating 3 times, one obtains

$$y(x) = \alpha_0 + \alpha_1 x + \frac{1}{2}\alpha_2 x^2 + A_3^T P^3 B(x).$$

Suppose that

$$A_i^T = [\alpha_i, 0, 0, 0], \ i = 0, 1, 2,$$
$$P = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{-1}{2} & \frac{1}{3} \\ 0 & \frac{3}{2} & \frac{-5}{2} & 1 \end{pmatrix}, \ P^2 = \begin{pmatrix} 0 & \frac{-1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & \frac{-1}{2} & \frac{1}{6} \\ 0 & \frac{1}{3} & \frac{-1}{2} & \frac{1}{6} \\ 0 & \frac{1}{3} & \frac{-1}{2} & \frac{1}{6} \end{pmatrix},$$
$$B^T(x) = [1, \ x, \ x + x^2, \ x + 3x^2 + x^3].$$

Then,

$$A_0^T B(x) = \alpha_0, \quad A_1^T P B(x) = \alpha_1 x, \quad A_2^T P^2 B(x) = \frac{1}{2}\alpha_2 x^2.$$

So, (18) implies that

$$y^{(i)}(x) = \sum_{j=i}^{m} A_j^T P^{j-i} B(x), \tag{20}$$

in which

$$A_j^T = [\alpha_j, 0, 0, ..., 0], \ j = 0, 1, ..., m-1. \tag{21}$$

Using relations (7), (12) and (20), we approximate $f(x)$, $K(x,s)$ and $y^{(i)}(x)$ by the Bell polynomials. Replacing the results and relations (17) and (20) in equation (1), we get

$$\begin{aligned}\sum_{i=0}^{m} c_i \sum_{j=i}^{m} A_j^T P^{j-i} B(x) &= F^T B(x) \\ &+ B(x) K \left( \int_0^x B^T(s) B(s) ds \right) \sum_{j=0}^{m} (P^j)^T A_j. \end{aligned}\tag{22}$$

Now, we collocate equation (22) in $N+1$ following nodes

$$x_i = \frac{2i-1}{2(N+1)}, \ i = 1, 2, ..., N+1. \tag{23}$$

Therefore, we obtain a system of linear algebraic equations of $(N+1) \times (N+1)$ degree, with an unknown vector $A_m$. By solving this linear system, one can approximate the solution of equation (1) as follows

$$y(x) = \sum_{j=0}^{m} A_j^T P^j B(x). \tag{24}$$

## 6. **Convergence and error estimation**

In this section, we prove the convergence of the proposed method and then an approximation for the error bound of our numerical method is presented. To do this, at first we recall the following theorem.

**Theorem 6.1.** *Suppose that $y(x)$ is sufficiently smooth function on $[0,1]$ and $P_N(x)$ is the interpolating polynomials of $y(x)$ at points $x_i$, $i = 0, 1, \dots, N$, in which for $i = 0, 1, \dots, N$, the points $x_i$ are the roots of the shifted Chebyshev polynomial of order $N+1$ on interval $[0,1]$. Then we have [7]*

$$y(x) - P_N(x) = \frac{\partial^{N+1} y(\vartheta)}{\partial x^{N+1} (N+1)!} \prod_{i=0}^{N} (x - x_i),$$

*where $\vartheta \in [0,1]$.*

According to this theorem, we have

$$|y(x) - P_N(x)| \leq \max_{x \in [0,1]} \left| \frac{\partial^{N+1} y(x)}{\partial x^{N+1}} \right| \frac{\prod_{i=0}^{N} |x - x_i|}{(N+1)!}. \tag{25}$$

Suppose that there is the following upper bound error

$$\max_{x\in[0,1]}\left|\frac{\partial^{N+1}y(x)}{\partial x^{N+1}}\right| \le \mu, \tag{26}$$

By replacing (26) into (25) and taking into account the estimates of Chebyshev interpolation nodes [12], one can conclude that

$$|y(x) - P_N(x)| \le \mu\frac{(\frac{1}{2})^{N+1}}{(N+1)!2^N}. \tag{27}$$

**Theorem 6.2.** *Let $y_N(x)$ be the best approximation of real sufficiently smooth function $y(x)$ by using Bell polynomials. Then, there is a real constant $\mu$ such that*

$$\|y(x) - y_N(x)\|_2 \le \mu\frac{(\frac{1}{2})^{N+1}}{(N+1)!2^N}. \tag{28}$$

*Proof.* Suppose that $\Pi_N$ is the space of Bell polynomials with order $N$. According to the definition of $y_N(x)$ which is the best approximation of $y(x)$, we have

$$\forall g(x) \in \Pi_N;\ \|y(x) - y_N(x)\|_2 \le \|y(x) - g(x)\|_2.$$

In particular, by considering $g(x) = P_N(x)$ and using (27), results in

$$\begin{aligned}\|y(x) - y_N(x)\|_2^2 \le \|y(x) - P_N(x)\|_2^2 &= \int_0^1 |y(x) - P_N(x)|^2 dx\\ &\le \int_0^1 \left[\mu\frac{(\frac{1}{2})^{N+1}}{(N+1)!2^N}\right]^2 dx = \left[\mu\frac{(\frac{1}{2})^{N+1}}{(N+1)!2^N}\right]^2,\end{aligned}$$

and the proof is completed.

□

*Remark* 6.3. From (28), one can obtain

$$\|y(x) - y_N(x)\|_2 = \mathcal{O}\left(\frac{1}{(N+1)!2^{2N+1}}\right).$$

So, if $N \to \infty$ then $\frac{1}{(N+1)!2^{2N+1}} \to 0$, which means that the approximate solution $y_N(x)$ will be converged to the exact solution $y(x)$.

**Theorem 6.4.** *Consider $m = 0$ and $c_0 = 1$ in equation (1). Let $y(x)$ be the exact solution and $y_N(x)$ be the approximated solution of (1) with the given assumption. Also, suppose that*

$$1 - \gamma - \theta > 0. \tag{29}$$

*Then, the upper bound error for the presented algorithm will be obtained as*

$$\|y(x) - y_N(x)\| \leq \frac{\varphi + \theta\beta}{1 - (\gamma + \theta)},$$

*with the following assumption:*

$$\begin{aligned}
&\max |y(x)| = \beta, \ \forall x \in I = [0,1], \\
&\max |k(x,s)| = \gamma, \ \forall (x,s) \in I \times I, \\
&\max |f(x) - f_N(x)| = \varphi, \\
&\max |k(x,s) - k_N(x,s)| = \theta, \ \forall (x,s) \in I \times I,
\end{aligned}$$

*Proof.* Our purpose is to determine an upper bound for the associated error between the exact solution $y(x)$ and the approximated solution $y_N(x)$ for the equation (1) through the presented scheme. According to the equation (1), we have

$$\begin{aligned}
\|y(x) - y_N(x)\| &= \left\| f(x) + \int_0^x k(x,s)y(s)ds - f_N(x) - \int_0^x k_N(x,s)y_N(s)ds \right\| \\
&\leq \|f(x) - f_N(x)\| + \|k(x,s)y(s) - k_N(x,s)y_N(s)\|, \qquad (30)
\end{aligned}$$

on the other hand

$$\begin{aligned}
&\|k(x,s)y(s) - k_N(x,s)y_N(s)\| \\
&\qquad = \|k(x,s)y(s) - k(x,s)y_N(s) + k(x,s)y_N(s) - k_N(x,s)y_N(s)\| \\
&\qquad \leq \|k(x,s)\|\|y(s) - y_N(s)\| + \|k(x,s) - k_N(x,s)\|\|y_N(s)\| \\
&\qquad \leq \|k(x,s)\|\|y(s) - y_N(s)\| + \|k(x,s) - k_N(x,s)\|(\|y(s)\| + \|y(s) - y_N(s)\| \\
&\qquad \leq (\gamma + \theta)\|y(s) - y_N(s)\| + \theta\beta. \qquad (31)
\end{aligned}$$

Using relations (30) and (31), we get

$$\|y(x) - y_N(x)\| \leq \varphi + (\gamma + \theta)\|y(x) - y_N(x)\| + \theta\beta. \qquad (32)$$

Therefore, by this relation and (29), one obtains

$$\|y(x) - y_N(x)\| \leq \frac{\varphi + \theta\beta}{1 - (\gamma + \theta)}, \qquad (33)$$

and this completes the proof. □

## 7. Numerical examples

In this section, we will apply the Bell polynomials method to some examples of Volterra integro-differential equations with the initial conditions and compare the quality of the computed solutions with those obtained by some other efficient methods such as Legendre, Bernoulli, Taylor and Bessel schemes. To do this, the error function $e(x)$ is defined as follows

$$e(x) = |y(x) - y_N(x)|,$$

and the root mean square (RMS) error will be obtained as

$$E = \sqrt{\frac{1}{k+1}\sum_{i=0}^{k}(y(x_i) - y_N(x_i))^2},$$

where $y(x)$ is the exact solution and $y_N(x)$ is the approximate solution of the equation by our scheme. Numerical examples are implemented by the Matlab R2015$a$ software.

**Example 7.1.** *Consider the following Volterra integro-differential equation of the fourth order*

$$y^{(4)}(x) - y(x) = x(1+e^x) + 3e^x - \int_0^x y(s)ds,\ 0 \leq x \leq 1, \tag{34}$$

*with the initial conditions*

$$y(0) = 1,\ y^{'}(0) = 1,\ y^{''}(0) = 2,\ y^{'''}(0) = 3.$$

*The exact solution of this equation is* $y(x) = 1 + xe^x$ *[20]. We approximate* $y^{(4)}(x)$ *by the Bell polynomials of order* $N$ *as*

$$y^{(4)}(x) = A_4^T B(x). \tag{35}$$

*Also, by using the initial conditions and the operational matrix of integration (16), one obtains*

$$y(x) = A_4^T P^4 B(x) + A_3^T P^3 B(x) + A_2^T P^2 B(x) + A_1^T P B(x) + A_0^T B(x), \tag{36}$$

*where*

Table 1. Numerical results of Example 1.

| $x_i$ | *Exact solution* | *Our method,* $N=4$ | *Our method,* *N=8* |
|---|---|---|---|
| 0.0 | 1.0 | 1.0 | 1.0 |
| 0.1 | 1.1105170 | 1.1105166 | 1.1105170 |
| 0.2 | 1.2442805 | 1.2442666 | 1.2442805 |
| 0.3 | 1.4049576 | 1.4048501 | 1.4049576 |
| 0.4 | 1.5967298 | 1.5962671 | 1.5967298 |
| 0.5 | 1.8243606 | 1.8229178 | 1.8243607 |
| 0.6 | 2.0932712 | 2.0896034 | 2.0932718 |
| 0.7 | 2.4096268 | 2.4015211 | 2.4096290 |
| 0.8 | 2.7804327 | 2.7642742 | 2.7804395 |
| 0.9 | 3.2136428 | 3.1838623 | 3.2136620 |
| 1.0 | 3.7182818 | 3.6666852 | 3.7183302 |

Table 2. Comparison of the absolute errors and RMS error of the presented method with Legendre and Bernoulli collocation methods for Example 1.

| $x_i$ | *Our Method,* $N=4$ | *Legendre method,* $N=4$ | *Our Method,* $N=8$ | *Bernoulli method,* $N=8$ |
|---|---|---|---|---|
| 0.0 | 0 | 0 | 0 | 0 |
| 0.2 | $1.3855e-005$ | $2.8055e-004$ | $1.8035e-009$ | $7.7721e-005$ |
| 0.4 | $4.6273e-004$ | $4.7299e-004$ | $1.4577e-008$ | $3.0926e-005$ |
| 0.6 | $3.6688e-003$ | $2.5270e-002$ | $5.3616e-007$ | $6.0544e-004$ |
| 0.8 | $1.6158e-002$ | $8.4330e-002$ | $6.8545e-006$ | $4.1000e-003$ |
| 1.0 | $5.1596e-002$ | $2.1828e-001$ | $4.8384e-005$ | $1.7700e-003$ |
| *RMS* | $2.2100e-002$ | $2.1828e-001$ | $1.9951e-005$ | $1.7700e-003$ |

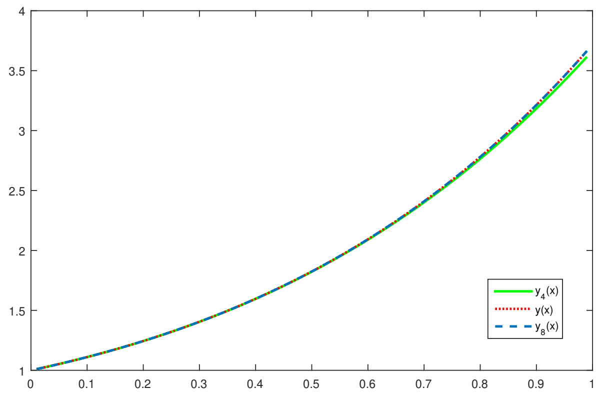


Figure 1. The graph of the exact solution and approximate solutions for $N=4$ and 8 in Example 1.

$$A_3^T=[3,0,0,0,0],\ A_2^T=[2,0,0,0,0],\ A_1^T=[1,0,0,0,0],\ A_0^T=[1,0,0,0,0].$$

*Now, we approximate $f(x)$ and $k(x,s)$ by the Bell polynomials as stated in section* 3. *Also, we substitute these approximate functions and relations (35) and (36) in equation (34). Using the collocation points (23), one obtains a system of linear algebraic equations to get an $A_4$ vector.*

*We implement the suggested method with different values of $N$ to approximate the solution of (34). Tables 1, 2 and Figure 1 show the numerical results for this example. Table 1 shows approximate solutions for $N=4$ and 8. Table 2 compares the absolute errors and RMS of the Bell polynomials method with the Legendre method [20] and the Bernoulli method [13]. The outcomes reveal that the results of our method are very promising and superior to the Legendre* *method and the Bernoulli method. It is seen that as N is increased, the error is decreased and the accuracy increases as well. Figure 1 depicts the approximate solutions for $N = 4$ and 8.*

**Example 7.2.** *Consider the following Volterra integro-differential equation of the first order*

$$y^{'}(x) = 1 - \int_0^x y(s)ds, \ 0 \leq x \leq 1, \tag{37}$$

*with initial condition*

$$y(0) = 0.$$

*The exact solution of this equation is $y(x) = \sin(x)$ [23]. We approximate $y^{'}(x)$ by the Bell polynomials of order $N$ as*

$$y^{'}(x) = A_1^T B(x).$$

*Using the given initial condition and the operational matrix of integration (16), results in*

$$y(x) = A_1^T PB(x).$$

*Then, equation (37) can be written as*

Table 3. Numerical results of Example2.

| $x_i$ | *Exact solution* | *Our method,* $N = 4$ | *Our method,* $N = 5$ | *Our method,* $N = 6$ |
|---|---|---|---|---|
| 0.0 | 0 | 0 | 0 | 0 |
| 0.1 | 0.0998334 | 0.0998333 | 0.0998334 | 0.0998334 |
| 0.2 | 0.1986693 | 0.1986666 | 0.1986693 | 0.1986693 |
| 0.3 | 0.2955202 | 0.2954999 | 0.2955202 | 0.2955202 |
| 0.4 | 0.3894183 | 0.3893333 | 0.3894186 | 0.3894186 |
| 0.5 | 0.4794255 | 0.4791666 | 0.4794270 | 0.4794270 |
| 0.6 | 0.5646424 | 0.5639999 | 0.5646479 | 0.5646479 |
| 0.7 | 0.6442176 | 0.6428333 | 0.6442338 | 0.6442338 |
| 0.8 | 0.7173560 | 0.7146666 | 0.7173971 | 0.7173972 |
| 0.9 | 0.7833269 | 0.7784999 | 0.7834205 | 0.7834206 |
| 1.0 | 0.8414798 | 0.8333333 | 0.8416663 | 0.8416664 |

$$A_1^T B(x) = F^T B(x) - B^T(x)K \left( \int_0^x B(s)B^T(s)ds \right) P^T A_1.$$

*Using the collocation points (23), one obtains a system of linear algebraic equations to get an $A_1$ vector.*
*The numerical results for this example are displayed in Tables 3, 4 and Figure 2. Table 3 exhibits the approximate solutions by the Bell polynomial method. The absolute errors and the RMS errors of this method, Taylor method [23] and Bessel method [33] are shown in Table 4. From these results, it is evident that*

Table 4. Comparison of the absolute errors and RMS error of the presented method with Taylor and Bessel collocation methods for Example 2.

| $x_i$ | *Our method,* $N=5$ | *Taylor method,* $N=5$ | *Our method,* $N=7$ | *Bessel method* |
|---|---|---|---|---|
| 0.0 | 0 | 0 | 0 | 0 |
| 0.2 | $3.7760e-009$ | $2.500e-009$ | $1.9534e-010$ | $4.0240e-007$ |
| 0.4 | $2.9382e-007$ | $3.2440e-007$ | $3.6070e-011$ | $2.0574e-007$ |
| 0.6 | $5.4466e-006$ | $5.5266e-005$ | $2.3569e-008$ | $3.7576e-007$ |
| 0.8 | $4.1079e-005$ | $4.1242e-005$ | $3.5299e-007$ | $1.8172e-007$ |
| 1.0 | $1.9539e-003$ | $1.9568e-004$ | $2.6911e-006$ | $9.6665e-006$ |
| *RMS* | $8.1543e-005$ | $8.4701e-005$ | $9.4350e-007$ | $3.9543e-006$ |

*the presented method provided a good approximate solution. Figure 2 depicts the approximate solutions for* $N=4$ *and* $6$*. It is seen that as* $N$ *is increased, the error is decreased and the accuracy increases as well.*

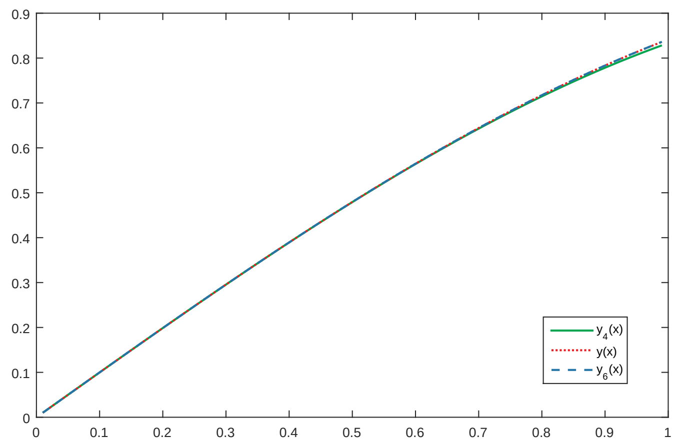


Figure 2. The graph of the exact solution and approximate solutions for $N=4$ and 6 in Example 2.

**Example 7.3.** *Consider the following Volterra integro-differential equation*

$$y'(x) + y(x) = \int_0^x e^{s-x} y(s)ds, \ 0 \le x \le 1, \tag{38}$$

*with initial condition*

$$y(0) = 1.$$

*The exact solution of this equation is* $y(x) = e^{-x}coshx$ *[13]. We approximate* $y^{'}(x)$ *by the Bell polynomials of order* $N$ *as*

$$y^{'}(x) = A_1^T B(x),$$

TABLE 5. Numerical results of Example3.

| $x_i$ | *Exact solution* | *Our method,* $N = 4$ | *Our method,* $N = 6$ |
|---|---|---|---|
| 0.0 | 1.0 | 1.0 | 1.0 |
| 0.1 | 0.9093653 | 0.9093934 | 0.9093651 |
| 0.2 | 0.8351600 | 0.8351749 | 0.8351491 |
| 0.3 | 0.7744058 | 0.7745891 | 0.7744073 |
| 0.4 | 0.7246644 | 0.7254755 | 0.7246687 |
| 0.5 | 0.6839397 | 0.6860852 | 0.6839958 |
| 0.6 | 0.6505971 | 0.6566208 | 0.6508095 |
| 0.7 | 0.6232984 | 0.6344677 | 0.6238789 |
| 0.8 | 0.6009482 | 0.6257380 | 0.6025596 |
| 0.9 | 0.5826494 | 0.6209277 | 0.5859553 |
| 1.0 | 0.5676676 | 0.6224829 | 0.5752348 |

TABLE 6. Comparison of the absolute errors and RMS error of the presented method with Bernoulli collocation method for Example 3.

| $x_i$ | *Our method,* $N = 4$ | *Bernoulli method,* $N = 4$ | *Our method,* $N = 6$ | *Bernoulli method,* $N = 6$ |
|---|---|---|---|---|
| 0.0 | 0 | 0 | 0 | 0 |
| 0.1 | $2.8122e-005$ | $7.8212e-003$ | $2.4676e-007$ | $3.2108e-004$ |
| 0.2 | $1.7918e-004$ | $1.1000e-003$ | $1.0876e-005$ | $2.4000-004$ |
| 0.3 | $1.8335e-004$ | $8.4639e-004$ | $1.1495e-006$ | $7.9000e-004$ |
| 0.4 | $7.3223e-004$ | $8.5470e-003$ | $4.2351e-006$ | $1.8000e-004$ |
| 0.5 | $2.1455e-003$ | $3.4000e-003$ | $5.6093e-005$ | $3.3800e-003$ |
| *RMS* | $9.3147e-004$ | $4.9000e-003$ | $2.2395e-005$ | $1.4000e-003$ |

*where,* $A_1$ *is the unknown vector. Also, by using the initial condition and the operational matrix of integration (16), we have*

$$y(x) = A_1^T PB(x) + A_0^T B(x),$$

*where* $A_0^T = [1, 0, 0, 0, 0]$*. Then, equation (38) can be written as*

$$A_1^T B(x) + A_1^T PB(x) + A_0^T B(x) = B^T(x)K \left( \int_0^x B(s)B^T(s)ds \right) (A_0 + P^T A_1).$$

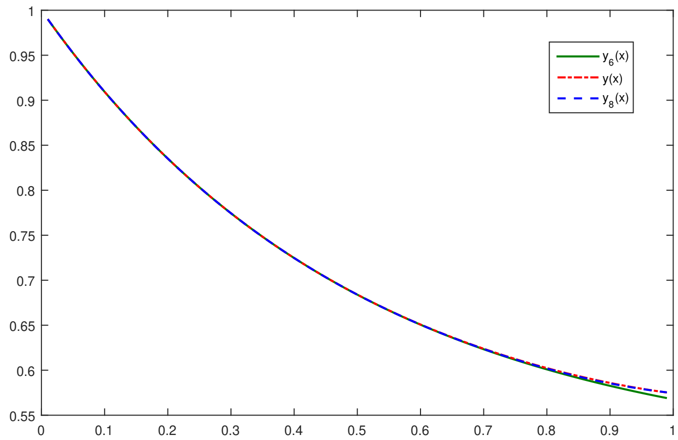


FIGURE 3. The graph of the exact solution and approximate solutions for $N = 6$ and 8 in Example 3.

*Now, by using collocating points* (23) *and solving the resulting linear algebraic system, the* $A_1$ *vector will be determined.*
*The numerical results of this example are displayed in Tables 5, 6 and Figures 3. Table 5 exhibits the exact solution and the approximate solutions with* $N = 4$ *and* 6*. Table 6 exhibits the values of the absolute errors and RMS error for the proposed method and the Bernoulli method with different values of* $N$*. This table reveals that our method can provide more accurate results in comparison with the Bernoulli method. From Table 6, we conclude that as* $N$ *is increased, the error is decreased. Figure 3 displays the exact solution and the approximate solutions with* $N = 6$ *and* 8*.*

## 8. **Conclusion**

In this paper, the Bell polynomials method was applied to obtain a numerical solution of high order Volterra integro-differential equations with the given initial conditions. The properties of the Bell polynomials were used to convert the equation into a system of algebraic equations which could be solved very easily. Some theorems were performed to show the convergence and error estimation of this method. The obtained results showed that the Bell polynomials method for solving Volterra integro-differential equations of high order with initial conditions was very effective and capable with a high accuracy compared with some other well-known methods such as Legendre, Bernoulli, Taylor and Bessel polynomials algorithms.

## References


[1] Bazm, S. (2015). Bernoulli polynomials for the numerical solution of some classes of linear and nonlinear integral equations. J. Comput. Appl. Math., 275, 44-60. https://doi.org/10.1016/j.cam.2014.07.018

[2] Bhrawy, AH, Tohidi, E., & Soleymani, F. (2012). A new Bernoulli matrix method for solving high-order linear and nonlinear Fredholm integro-differential equations with piecewise intervals. Appl. Math. Comput., 219(2), 482-497. https://doi.org/10.1016/j.amc.2012.06.020

[3] Bicer, GG, Öztürk, Y., & Gülsu, M. (2018). Numerical approach for solving linear Fredholm integro-differential equation with piecewise intervals by Bernoulli polynomials. Int. J. Comput. Math., 95(10), 2100-2111. https://doi.org/10.1080/00207160.2017.1366458

[4] Brunner, H. (1990). On the numerical solution of nonlinear Volterra-Fredholm integral equations by collocation methods. SIAM J. Numer. Anal., 27(4), 987-1000. https://doi.org/10.1137/0727057

[5] Dehghan, M., & Salehi, R. (2012). The numerical solution of the nonlinear integro-differential equations based on the meshless method. J. Comput. Appl. Math., 236(9), 2367-2377. https://doi.org/10.1016/j.cam.2011.11.022

[6] Fakhar-Izadi, F., & Dehghan, M. (2011). The spectral methods for parabolic Volterra integro-differential equations. J. Comput. Appl. Math., 235(14), 4032-4046. https://doi.org/10.1016/j.cam.2011.02.030

[7] Gasca, M., & Sauer, T. (2001). On the history of multivariate polynomial interpolation. Numerical Analysis: Historical Developments in the 20th Century. 135-147. https://doi.org/10.1016/B978-0-444-50617-7.50007-0

[8] Hesameddini, E., & Shahbazi, M. (2022). Application of Bernstein polynomials for solving Fredholm integro-differential-difference equations. Appl. Math. J. Chin. Univ., 37(4), 475-493. https://doi.org/10.1007/s11766-022-3620-9

[9] Hesameddini, E., & Shahbazi, M. (2017). Solving system of Volterra-Fredholm integral equations with Bernstein polynomials and hybrid Bernstein Block-Pulse functions. J. Comput. Appl. Math., 315, 182-194. https://doi.org/10.1016/j.cam.2016.11.004

[10] Maleknejad, K., Basirat, B., & Hashemizadeh, E. (2012). A Bernstein operational matrix approach for solving a system of high order linear Volterra-Fredholm integro-differential equations. Math. Comput. Model., 55(3-4), 1363-1372. https://doi.org/10.1016/j.mcm.2011.10.015

[11] Maleknejad, K., & Mahmoudi, Y. (2003). Taylor polynomial solution of high-order nonlinear Volterra-Fredholm integro-differential equations. Appl. Math. Comput., 145(2-3), 641-653. https://doi.org/10.1016/S0096-3003(03)00152-8

[12] Mason, JC, & Handscomb, DC (2002). Chebyshev Polynomials. CRC Press LLC. https://doi.org/10.1201/9781420036114

[13] Matinfar, M., Abdollahi Lashaki, H., & Akbari, M. (2017). Numerical approximation based on the Bernoulli polynomials for solving Volterra integro-differential equations of a high order. Math. Rese., 2(3), 19-32. https://doi.org/10.29252/mmr.2.3.19

[14] Mirzaee, F. (2017). Numerical solution of nonlinear Fredholm-Volterra integral equations via Bell polynomials. Comput. Methods Differ. Equ., 5(2), 88-102. https://dorl.net/dor/20.1001.1.23453982.2017.5.2.1.1

[15] Ordokhani, Y., & Mohtashami, MJ (2010). Approximate solution of nonlinear Fredholm integro-differential equations with time delay by using Taylor method. J. Sci. TMU., 9(1), 73-84.

[16] Rebenda, J. (2019). An application of Bell polynomials in numerical solving of nonlinear differential equations. arXiv preprint arXiv:1901.10418, 1-10. https://doi.org/10.48550/arXiv.1901.10418

[17] Rohaninasab, N., Maleknejad, K., & Ezzati, R. (2018). Numerical solution of high-order Volterra-Fredholm integro-differential equations by using Legendre collocation method. Appl. Math. Comput., 328, 171-188. https://doi.org/10.1016/j.amc.2018.01.032

[18] Saadatmandi, A., & Dehghan, M. (2010). Numerical solution of the higher-order linear Fredholm integro-differential-difference equation with variable coefficients. Comput. Math. with Appl., 59(8), 2996-3004. https://doi.org/10.1016/j.camwa.2010.02.018

[19] Saberi-Nadjafi, J., & Ghorbani, A. (2009). He's homotopy perturbation method: an effective tool for solving nonlinear integral and integro-differential equations. Comput. Math. with Appl., 58(11-12), 2379-2390. https://doi.org/10.1016/j.camwa.2009.03.032

[20] Venkatesh, SG, Ayyaswamy, SK, & Raja Balachander, S. (2012). Legendre approximation solution for a class of high-order Volterra integro-differential equations. Ain Shams Eng. J., 3(4), 417-422. https://doi.org/10.1016/j.asej.2012.04.007

[21] Wazwaz, AM (2011). Linear and Nonlinear Integral Equations. Berlin: Springer. https://doi.org/10.1007/978-3-642-21449-3-14

[22] Wheeler, FS (1987). Bell polynomials. Acm Sigsam Bulletin, 21(3), 44-53. https://doi.org/10.1145/29309.29317

[23] Yalcınbas, S., & Sezer, M. (2000). The approximate solution of high-order linear Volterra-Fredholm integro-differential equations in terms of Taylor polynomials. Appl. Math. Comput., 112(2-3), 291-308. https://doi.org/10.1016/s0096-3003(99)00059-4

[24] Yıldız, G., Tınaztepe, G., & Sezer, M. (2020). Bell polynomial approach for the solutions of Fredholm integro-differential equations with variable coefficients. Comp. Model. Eng. Sci., 123(3), 973-993. https://doi.org/10.32604/cmes.2020.09329

[25] Yüzbası, S. (2022). A new Bell function approach to solve linear fractional differential equations. Appl. Numer. Math., 174, 221-235. https://doi.org/10.1016/j.apnum.2022.01.014

[26] Yüzbası, S. (2022). Fractional Bell collocation method for solving linear fractional integro-differential equations. Math. Sci., 1-12. https://doi.org/10.1007/s40096-022-00482-0

[27] Yüzbası, S. (2014). Laguerre approach for solving pantograph-type Volterra integro-differential equations. Appl. Math. Comput., 232, 1183-1199. https://doi.org/10.1016/j.amc.2014.01.075

[28] Yüzbası, S. (2017). Shifted Legendre method with residual error estimation for delay linear Fredholm integro-differential equations. J. Taibah Univ. Sci., 11(2), 344-352. https://doi.org/10.1016/j.jtusci.2016.04.001

[29] Yüzbası, S., Gok, E., & Sezer, M. (2013), Muntz-Legendre polynomial solutions of linear delay Fredholm integro-differential equations and residual correction. Math. Comput. Appl., 18(3), 476-485. https://doi.org/10.3390/mca18030476

[30] Yüzbası, S., & Ismailov, N. (2018). An operational matrix method for solving linear Fredholm-Volterra integro-differential equations. Turkish J. Math., 42(1), 243-256. https://doi.org/10.3906/mat-1611-126

[31] Yüzbası, S., & Karaçayir, M. (2018). A numerical approach for solving high-order linear delay Volterra integro-differential equations. Int J Comput Methods, 15(5), 1850042. https://doi.org/10.1142/S0219876218500421

[32] Yüzbası, S., Sahın, N., & Sezer, M. (2011). Bessel polynomial solutions of high-order linear Volterra integro-differential equations. Comput. Math. with Appl., 62(4), 1940-1956. https://doi.org/10.1016/j.camwa.2011.06.038

[33] Yüzbası, S., Sahın, N., & Yıldirim, A. (2012). A collocation approach for solving high-order linear Fredholm-Volterra integro-differential equations. Math Comput Model, 55(3-4), 547-563. https://doi.org/10.1016/j.mcm.2011.08.032

[34] Yüzbası, S., & Yıldirim, G. (2020). Pell-Lucas collocation method to solve high-order linear Fredholm-Volterra integro-differential equations and residual correction. Turkish J. Math., 44(4), 1065-1091. https://doi.org/10.3906/mat-2002-55

Najmeh Kasaei
Orcid number: 0009-0007-1537-6990
Department of Mathematics
Shiraz University of Technology
Shiraz, Iran
*Email address*: N.Kasaiei@sutech.ac.ir

Esmail Hesameddini
Orcid number: 0000-0002-7012-1161
Department of Mathematics
Shiraz University of Technology
Shiraz, Iran
*Email address*: hesameddini@sutech.ac.ir

Mohammad Nabati
Orcid number: 0000-0003-2864-7415
Department of Basic Sciences
Petroleum University of Technology
Abadan, Iran
*Email address*: Nabati@put.ac.ir