Journal of
Applied & Computational Mathematics

Shahooth et al., J Appl Computat Math 2016, 5:2
DOI: 10.4172/2168-9679.1000298

Research Article Open Access

Approximation Solution to Solving Linear Volterra-Fredholm Integro-Differential Equations of the Second Kind by Using Bernstein Polynomials Method

Shahooth* MK, Ahmad RR, Din U-KS, Swidan W, Al-Husseini OK and Shahooth WK

Department of Mathematics, Faculty Science and Technology, National University of Malaysia, Malaysia

Abstract

The aim of this study is to present numerical method for solving the linear Volterra-Fredholm integro-differential equations of the second kind. This method is called the Bernestein polynomials method. This technique transforms the integro-differential equations to the system of algebraic equations. Some numerical results are presented to illustrate the efficiency and accuracy of this method.

Keywords: Bernstein polynomials; Volterra-Fredholm integro-differential equations; Numerical solution; Mathematical tools

Introduction

Mathematical modelling of real-life problems usually results in functional equations, such an ordinary or partial differential equations, integral and integro-differential equations and stochastic equations. Many mathematical formulations of physical phenomena contain integro-differential an equation, these equations arises in many fields like fluid dynamics, biological models and chemical kinetics. In fact, integro-differential equations are usually difficult to solve analytically so it is required to obtain an efficient approximate or numerical solution [1,2].

There are several solution methods including Galerkin, collocation, Block Pulse Functions and direct method, for integro-differential equations have been studied [3-8].

In this study Bernstein polynomial method (BPM) is used to solve the linear Volterra-Fredholm integro-differential equation of the second kind:

$$\sum_{i=0}^{m}\mu_i y^{(i)}(x)=f(x)+\lambda_1\int_a^x k_1(x,t)u(t)dt+\lambda_2\int_a^b k_2(x,t)u(t)dt \qquad (1)$$

with the initial condition $y^{(i)}(a)=y_i$, $i=1,\ldots,m$

where $a,b,\lambda_1,\lambda_2,y_i$ are constants values, $f(x),k_1(x,t),k_2(x,t)$ and $\mu_i,i=1,\ldots,m$ with $\mu_i(x)\neq 0$ are known functions that have derivatives on an interval $a \le x \le t \le b$ and $y(x)$ is the unknown function which must be determined.

Bernstein Polynomials Method

Polynomials are incredibly useful mathematical tools as they are simple to define, can be calculated quickly on computer systems and represent a tremendous variety of functions. The Bernstein polynomials of degree-n are defined by [9]:

$$B_i^n(t)=\binom{n}{i}t^i(1-t)^{n-i} \quad \text{for} \quad i=0,1,2,\cdots,n, \qquad (2)$$

where

$\binom{n}{i}=\frac{n!}{i!(n-i)!}$, n is the degree of polynomials, i is the index of polynomials and t is the variable.

The exponents on the t term increase by one as i increase, and the exponents on the $(1\text{-}t)$ term decrease by one as i increases.

The Bernstein polynomials of degree-n can be defined by blending together two Bernstein polynomials of degree-$(n\text{-}1)$ that is, the k^{th} n^{th}-degree Bernstein polynomial can be written as [9]:

$$B_k^n(t)=(1-t)B_k^{n-1}(t)+tB_{k-1}^{n-1}(t) \qquad (3)$$

Bernstein polynomials of degree-n can be written in terms of the power basis. This can be directly calculated using equation (2) and the Binomial theorem as follows [9]:

$$B_k^n(t)=\binom{n}{k}t^k(1-t)^{n-k}=\sum_{i=k}^{n}(-1)^{i-k}\binom{n}{i}\binom{i}{k}t^i,$$

where the Binomial theorem is used to expand $(1\text{-}t)^{n\text{-}k}$.

The derivatives of the n^{th}-degree Bernstein polynomials are polynomials of degree-$(n\text{-}1)$.

$$\frac{d}{dt}B_k^n(t)=\frac{d}{dt}\left(\binom{n}{k}t^k(1-t)^{n-k}\right)=n\left(B_k^{n-1}(t)-B_k^{n-1}(t)\right), \quad 0\le k\le n \qquad (4)$$

A Matrix Representation for Bernstein Polynomials

In many applications, a matrix formulation for the Bernstein polynomials is useful. These are straight forward to develop if only looking at a linear combination in terms of dot products. Given a polynomial written as a linear combination of the Bernstein basis functions [10]:

$$B(t)=c_0B_0^n(t)+c_1B_1^n(t)+c_2B_2^n(t)+\cdots+c_nB_n^n(t) \qquad (5)$$

The dot product of two vectors

***Corresponding author:** Shahooth MK, Department of Mathematics, Faculty Science and Technology, National University of Malaysia, Malaysia, Tel: +60389215555; E-mail: moha861122@yahoo.com

Received February 15, 2016; **Accepted** April 28, 2016; **Published** May 04, 2016

Citation: Shahooth MK, Ahmad R, Salma U, Swidan W, Al-Husseini OK, et al. (2016) Approximation Solution to Solving Linear Volterra-Fredholm Integro-Differential Equations of the Second Kind by Using Bernstein Polynomials Method. J Appl Computat Math 5: 298. doi:10.4172/2168-9679.1000298

$$B(t)=\begin{bmatrix} B_0^n(t) & B_1^n(t) & B_2^n(t) & \cdots & B_n^n(t)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix} \tag{6}$$

which can be converted to the following form:

$$B(t)=\begin{bmatrix}1 & t & t^2 & \cdots & t^n\end{bmatrix}\begin{bmatrix} b_{00} & 0 & 0 & \cdots & 0\\ b_{10} & b_{11} & 0 & \cdots & 0\\ b_{20} & b_{21} & b_{22} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nn}\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix} \tag{7}$$

where b_{nn} are the coefficients of the power basis that are used to determine the respective Bernstein polynomials, we note that the matrix in this case is lower triangular. The matrix of derivatives of Bernstein polynomials

$$B'(t)=\begin{bmatrix}0 & 1 & 2t & \cdots & nt^{n-1}\end{bmatrix}\begin{bmatrix} b_{00} & 0 & 0 & \cdots & 0\\ b_{10} & b_{11} & 0 & \cdots & 0\\ b_{20} & b_{21} & b_{22} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nn}\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix} \tag{7a}$$

Solution for Volterra-Fredholm Integro-Differential Equations of the Second Kind

In this section, Bernstein polynomials method is proposed to find the approximate solution for Volterra-Fredholm integro-differential equations of the second kind.

Consider the Volterra-Fredholm integro-differential equations of the second kind in equation (1):

$$\sum_{i=0}^{m}\mu_i y^{(i)}(x)=f(x)+\lambda_1\int_a^x k_1(x,t)y(t)dt+\lambda_2\int_a^b k_2(x,t)y(t)dt \quad ,x\in[a,b] \tag{8}$$

Let $y(x)=B(t)$ then,

$$y(x)=\begin{bmatrix} B_0^n(x) & B_1^n(x) & B_2^n(x) & \cdots & B_n^n(x)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix} \tag{9}$$

Substituting (9) into equation (8), we get:

$$\sum_{i=1}^{m}\mu_i\left[\begin{bmatrix} B_0^n(x) & B_1^n(x) & B_2^n(x) & \cdots & B_n^n(x)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}\right]^i=f(x)+$$

$$\lambda_1\int_a^x k_1(x,t)\begin{bmatrix} B_0^n(t) & B_1^n(t) & B_2^n(t) & \cdots & B_n^n(t)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt+\lambda_2\int_a^b k_2(x,t)\begin{bmatrix} B_0^n(t) & B_1^n(t) & B_2^n(t) & \cdots & B_n^n(t)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt \tag{10}$$

Applying equation (7) into equation (10), we have:

$$\sum_{i=1}^{m}\mu_i\left[\begin{bmatrix} B_0^n(x) & B_1^n(x) & B_2^n(x) & \cdots & B_n^n\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}\right]^i=f(x)$$

$$+\lambda_1\int_a^x k_1(x,t)\begin{bmatrix}1 & t & t^2 & \cdots & t^n\end{bmatrix}\begin{bmatrix} b_{00} & 0 & 0 & \cdots & 0\\ b_{10} & b_{11} & 0 & \cdots & 0\\ b_{20} & b_{21} & b_{22} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nn}\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt$$

$$+\lambda_2\int_a^b k_2(x,t)\begin{bmatrix}1 & t & t^2 & \cdots & t^n\end{bmatrix}\begin{bmatrix} b_{00} & 0 & 0 & \cdots & 0\\ b_{10} & b_{11} & 0 & \cdots & 0\\ b_{20} & b_{21} & b_{22} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nn}\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt \tag{11}$$

Now to find all integration in equation (11). Then in order to determine $C_0,C_1,\ldots,C_n$ we need n equations. Now chose $x_i,i=1,2,3,\ldots,n$ in the interval $[a,b]$, which gives n equations. Solve the n equations by Gauss elimination to find the values of $C_0,C_1,\ldots,C_n$. The following algorithm summarizes the steps for finding the approximate solution for the second kind of linear Volterra-Fredholm integro-differential equations.

Algorithm (BPM)

Input: $(f(x),k(x,t),y(x),a,b,x,\lambda_1,\lambda_2)$

Output: Polynomials of degree n

Step 1: Choice n the degree of Bernstein polynomials

$$B_i^n(t)=\binom{n}{i}t^i(1-t)^{n-i} \quad for\ i=0,1,2,\cdots,n$$

Step 2: Put the Bernstein polynomials in linear Volterra-Fredholm integro-differential equations of second kind.

$$\sum_{i=1}^{m}\mu_i y^{(i)}(x)=f(x)+\lambda_1\int_a^x k_1(x,t)\begin{bmatrix} B_0^n(t) & B_1^n(t) & B_2^n(t) & \cdots & B_n^n(t)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt+\lambda_2\int_a^b k_2(x,t)\begin{bmatrix} B_0^n(t) & B_1^n(t) & B_2^n(t) & \cdots & B_n^n(t)\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt$$

Step 3:

Compute Volterra integral $\lambda_1\int_a^x k_1(x,t)\begin{bmatrix}1 & t & t^2 & \cdots & t^n\end{bmatrix}\begin{bmatrix} b_{00} & 0 & 0 & \cdots & 0\\ b_{10} & b_{11} & 0 & \cdots & 0\\ b_{20} & b_{21} & b_{22} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nn}\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt$

Compute Fredholm integral $\lambda_2\int_a^b k_2(x,t)\begin{bmatrix}1 & t & t^2 & \cdots & t^n\end{bmatrix}\begin{bmatrix} b_{00} & 0 & 0 & \cdots & 0\\ b_{10} & b_{11} & 0 & \cdots & 0\\ b_{20} & b_{21} & b_{22} & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ b_{n0} & b_{n1} & b_{n2} & \cdots & b_{nn}\end{bmatrix}\begin{bmatrix} c_0\\ c_1\\ c_2\\ \vdots\\ c_n\end{bmatrix}dt$

Compute the term $\sum_{i=1}^{m}\mu_i y^{(i)}(x)$

Step4:

Compute $C_0,C_1,\ldots,C_n$, where $x_i,i=1,2,3,\ldots,n$, $x_i\in[a,b]$

End.

Numerical Examples

In this section, two numerical examples are exhibited to illustrate the Bernstein polynomials method. The computations associated with the examples were performed using Matlab ver.2013a.

Example 1: Consider the linear Volterra-Fredholm integro-differential equation of the second kind [11].

$$u'(x)=f(x)-\left(\int_0^x x^2yu(y)dy+\int_0^1 x(x-y)u(y)dy\right),$$

with the initial condition $u(0)=1$, $0 \le x \le 1$

Where $f(x)=\sin(x)-x^2\cos(x)-x^3\sin(x)+x^2-x^2\sin(1)+x\cos(1)+x\sin(1)-x$ and the exact solution is $u(x)=\cos(x)$.

Tables 1 and 2 show that numerical results and the error respectively with the exact solution for Example 1 for n=1,4 and 7 by using BPM (Figure 1).

Example 2: Consider the linear Volterra-Fredholm integro-differential equation of the second kind [12].

$$u'(x) = 2e^x - 2 + \int_0^x u(t)\,dt + \int_0^1 u(t)dt \ , u(0) = 0$$

and the exact solution is $u(x)=xe^x$ (Figure 2).

Comparison with other Methods

In this part, the BPM was compared its performance with Repeated Trapezoidal method and Repeated Simpson's 1/3 Method. A parameters here such as the degree of BPM and the error are considered as comparison. Throughout this manuscript, the convergence test with the proposed method is considered the last square error. We can notice that all the methods on the finite interval $[a,b]$. In BPM we proposed that we have a solution and we can develop it by increasing the degree of Bernstein polynomials method. Accordingly, the solution is convergence by increasing the number of the limits of Bernstein polynomials resulted from increasing the degree of Bernstein polynomials n, and the error decreases as results of that. As for Repeated Trapezoidal method and Repeated Simpson's 1/3 Method, the solution is a result of Trapezoidal and Simpson's 1/3 laws. Consequently, the error in this methods decrease in speed depending the h value which in its turn depends on the number of n points as mentioned earlier. Also, the accuracy of solution increases with the increase of n points number and the result will be a decrease at h value. Finally, the following table shows the error between these methods for example 1 (Table 3).

t	*yexact*	*yapp, n=1*	*yapp, n=4*	*yapp, n=7*
0	1.000000000000000	1.000000000000000	1.000000000000000	1.000000000000000
0.100000000000000	0.995004165278026	0.855760000000000	0.994277930000000	0.995005924500000
0.200000000000000	0.980066577841242	0.711520000000000	0.985024480000000	0.980071699200000
0.300000000000000	0.955336489125606	0.567280000000000	0.968301130000000	0.955343773900000
0.400000000000000	0.921060994002885	0.423040000000000	0.940052480000000	0.921068390400000
0.500000000000000	0.877582561890373	0.278800000000000	0.896106250000000	0.877589062500000
0.600000000000000	0.825335614909678	0.134560000000000	0.832173280000000	0.825341536000000
0.700000000000000	0.764842187284489	-0.009680000000000	0.743847530000000	0.764847740700000
0.800000000000000	0.696706709347165	-0.153920000000000	0.626606080000000	0.696710246400000
0.900000000000000	0.621609968270664	-0.298160000000000	0.475809130000000	0.621608734900000
1.000000000000000	0.540302305868140	-0.442400000000000	0.286700000000000	0.540300000000000

Table 1: Numerical results for Example 1 with exact solution by using BPM.

(*yexact-yapp, n*=1)^2	(*yexact-yapp, n*=4)^2	(*yexact-yapp, n*=7)^2
0	0	0
0.019388937563974	0.000000527417679	0.000000000003095
0.072117264470242	0.000024580793816	0.000000000026228
0.150587838752492	0.000168081913002	0.000000000053068
0.248024910467622	0.000360676540379	0.000000000054707
0.358540556423998	0.000343127021183	0.000000000042258
0.477170950153844	0.000046753663887	0.000000000035059
0.599884618595948	0.000440775634493	0.000000000030840
0.723565798654787	0.004914098234869	0.000000000012511
0.845976794532619	0.021257884440428	0.000000000001521
0.965703821958559	0.064314129541638	0.000000000005317

Table 2: The errors for Example 1 by using BPM.

Points	Repeated Trapezoidal *h*=0.01	Repeated Simpson's 1/3 h=0.01	Bernstein polynomials *yapp, n*=7
0.100000000000000	0.000492742	0.000492400	0.000000000003095
0.200000000000000	0.000969158	0.000967886	0.000000000026228
0.300000000000000	0.00142909	0.00142649	0.000000000053068
0.400000000000000	0.00187288	0.00186873	0.000000000054707
0.500000000000000	0.00230186	0.00229612	0.000000000042258
0.600000000000000	0.00271897	0.00271179	0.000000000035059
0.700000000000000	0.00312960	0.00312126	0.000000000030840
0.800000000000000	0.00354248	0.00353340	0.000000000012511
0.900000000000000	0.00397088	0.00396158	0.000000000001521
1.000000000000000	0.00443427	0.00440738	0.000000000005317

Table 3: Comparison the error between Trapezoidal and Simpson's 1/3 Methods with BPM for Example 1.

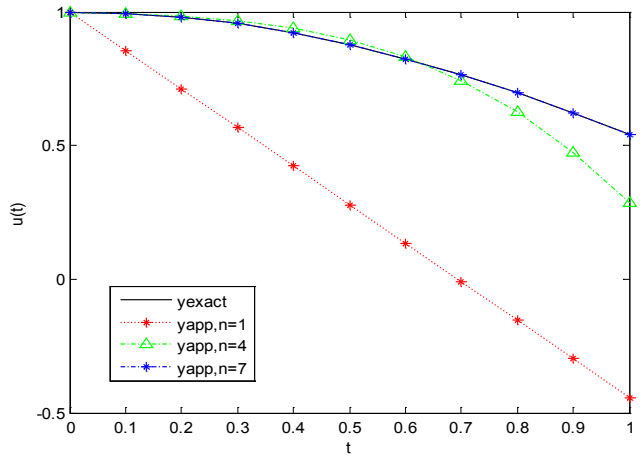

Figure 1: Approximation solutions and exact solution for Example 1 by using BPM.

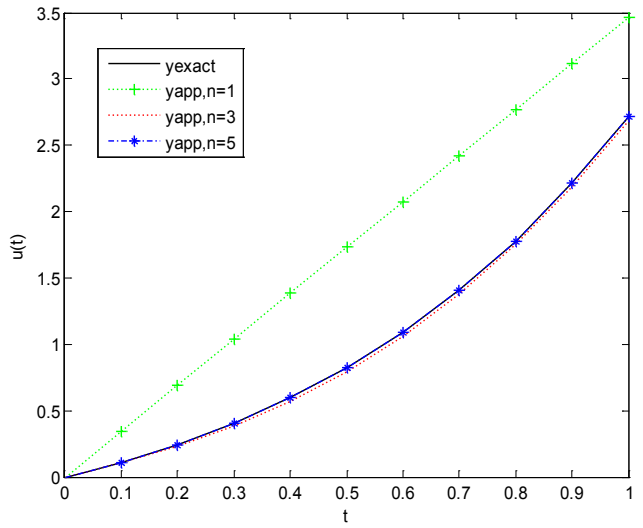

Figure 2: Approximation solutions and exact solution for Example 2 by using BPM.

Conclusion

In the present study we have successfully used the proposed method to find an approximation solution for solving a second kind Volterra-Fredholm integro-differential equation. We noted from our results the approximation solution is close to the exact solution when we only used the degree of BPM is n=4 in example 1 and the error is small but still impossible to get satisfactory results with using this degree. When n=7 the result becomes so accuracy, so efficiency and the curve of an approximation solution is exactly over the curve of the exact solution. The figuring comes about additionally demonstrate that this strategy is so productive and it can be successfully use in the numerical arrangement of such sort mathematical statements. The integro differential equations are usually difficult to solve analytically so they are required to obtain an efficient approximated method. For this reason, the presented method have been proposed for approximated solution to the linear Volterra-Fredholm integro-differential equations of the second kind. From numerical examples it can be seen that the proposed numerical method is efficient and accurate to estimate the solution of these equations. Also we noted that when the degree of Bernstein polynomials is increasing the errors decrease to smaller values.

References

1. Batiha B, Noorani MSM, Hashim I (2008) Numerical solutions of the nonlinear integro-differential equations. Int J Open Problems Compt Math 1: 34-42.
2. Altawallbeh Z, AL-Smadi MH, Abu-Gdairi R (2013) Approximate solution of second-order integro-differential equation of Volterra type in RKHS method. Int Journal of Math Analysis 7: 2145-2160.
3. Pedas A, Tamme E (2008) Discrete Galerkin method for Fredholm integro-differential equations with weakly singular kernels. Journal of Computational and Applied Mathematics 213: 111-126.
4. Abubakar A, Taiwo OA (2014) Integral Collocation Approximation Methods for the Numerical Solution of High-Orders Linear Fredholm-Volterra Integro-Differential Equations. American Journal of Computational and Applied Mathematics 4: 111-117.
5. Rahmani L, Rahimi B, Mordad M (2011) Numerical Solution of Volterra-Fredholm Integro-Differential Equation by Block Pulse Functions and Operational Matrices. Gen 4: 37-48.
6. Babolian E, Masouri Z, Hatamzadeh-Varmazyar S (2009) Numerical solution of nonlinear Volterra–Fredholm integro-differential equations via direct method using triangular functions. Computers & Mathematics with Applications 58: 239-247.
7. Sweilam NH, Khader MM (2010) A Chebyshev pseudo-spectral method for solving fractional-order integro-differential equations. The ANZIAM Journal 51: 464-475.
8. Rashidinia J, Tahmasebi A (2013) Approximate solution of linear integro-differential equations by using modified Taylor expansion method. World Journal of Modelling and Simulation 9: 289-301.
9. Joy KI (2000) Bernstein polynomials. University of California, Davis.
10. AL-Juburee AK (2010) Approximate Solution for linear Fredholm Integro-Differential Equation and Integral Equation by Using Bernstein Polynomials method. Al- Mustansiriyah University, Iraq.
11. Omran HH (2009) Numerical Methods For Solving the First Order Linear Fredholm-Volterra Integro-Differential Equations. Journal of Al-Nahrain University 12: 139-143.
12. Wazwaz A (2011) Linear And Nonlinear Integral Equations: Methods And Applications. Springer, USA.