Bernstein Polynomials Solving One Dimensional Delay Volterra Integro Differential Equations

Shymaa Hussain Salih*

Received on: 8/6/2010
Accepted on:7/10/2010

Abstract

The main purpose of this paper lies briefly in submitting least square method for solving linear delay Volterra integro differential equation of the second kind containing three types (Retarded,Neutral and mixed)with the aid Bernstein polynomials as basis functions to compute the approximated solutions of delay volterra intgro differential equations .Three examples are given for determining the results of this method.

Keywords: Delay Volterra integro differential equation of the second kind, least square method,Bernstien polynomials, least square error.

استخدمت متعددات الحدود برنشتاين لحل معادلات فولتيرا التكاملية التفاضلية التباطؤية ذات البعد الواحد

الخلاصة:

الهدف الاساسي في هذا العمل هو تقديم طريقة التقريبات الصغرى لحل معادلات فولتيرا التكاملية التباطؤية الخطية من النوع الثاني والمتضمنة انواعها الثلاثة {التراجعية ، المتعادلة والمختلطة} وباستخدام متعددات الحدود برنشتاين كدوال اساسيه لحساب الحل التقريبي لهذه المعادلات واعطيت ثلاث امثلة لتوضيح هذه الطريقة.

Introduction

Delay integro differential equation is an equation involving one or more unkown function U(x) togther with both differential and integral operations on U(x).this means that it is an equation containing derivative of the unknown function U(x) which appears outside the integral sign.

Recall form of linear Volterra Integro-differential equation of the second kind [1]:

$$\frac{dU(x)}{dx} = f(x) + l\int_0^x K(x,y)U(y)\,dy \quad ..(1)$$

where $K(x,y)$ is the kernel function and $f(x)$ is any continuous function and l is a scalar parameter.

* Applied Science Department, University of Technology /Baghdad

Three *types* of linear delay Volterra integro differential equationl are defined

1. Retarded integro-differential equation when delay comes in the unknown function u(x) involved in the integrand sign.

$$\frac{du(x)}{dx} = f(x) + \int_0^x K(x,t)u(y-t)dy, 0 \le x$$

2. Neutral integro-differential equation when delay comes in the derivative of u(x) outside the integral.

$$\frac{du(x-t)}{dx} = f(x) + \int_0^x K(x,t)u(y)dy, 0 \le x$$

3. Mixed integro-differential equation

$$\frac{du(x-t_1)}{dx} = f(x) + \int_0^x K(x,t)u(y-t_2)dy, 0 \le x$$

Where t, t_1, t_2 are positive integer called delay or time lage.

Numerical Solution: The discrete form for the exact solution *U(x)* for equation (1) can be written in the form: $U(x) \approx U_N(x)$,
where N is a positive integer This paper pivoted to implement Bernstein polynomials (B-spline) as a discrete function (polynomial) of $U_N(x)$ i.e.

$$U_N(x) = \sum_{i=0}^{N} a_i B_{i,N}(x) \qquad \ldots(2)$$

where $B_{i,N}(x) = \binom{N}{i} x^i (1-x)^{N-i}$

for $i=0,1,2,\ldots,n$, where the combination

$$\binom{N}{i} = \frac{N!}{i!(N-i)!}$$

There are N+1 Nth degree Bernstein polynomials.
The Bernstein polynomials of degree 1 are:

$$B_{0,1}(x) = 1 - x$$
$$B_{1,1}(x) = x$$

For the graph of these functions when $0 \le x \le 1$ see [2].
The Bernstein polynomials of deg.2 are

$$B_{0,2}(x) = (1-x)^2$$
$$B_{1,2}(x) = 2x(1-x)$$
$$B_{2,2}(x) = x^2$$

The graph of which can be found in [3]. Finally, the Bernstein polynomials of degree 3 may be calculated as:

$$B_{0,3}(x) = (1-x)^3$$
$$B_{1,3}(x) = 3x(1-x)^2$$
$$B_{2,3}(x) = 3x^2(1-x)$$
$$B_{3,3}(x) = x^3$$

and their graph has shown in [3, 4] as well. Some necessary characters of Bernstein polynomials: (for the proof see [5,6]) Therefore we could write the function

$$U(x) \approx U_N(x) = a_0 B_{0,N}(x) + a_1 B_{1,N}(x) + \ldots + a_N B_{N,N}(x)$$

$$= a_0 \sum_{k=0}^{N} (-1)^k \binom{N}{k}\binom{k}{0} x^k + \ldots +$$

$$+ a_N \sum_{k=N}^{N} (-1)^{k-N} \binom{N}{k}\binom{k}{N} x^k$$

$$= a_0 + a_1\left[\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{1}\binom{1}{1}\right]x^1 + \ldots +$$

$$+ a_N\left[\sum_{k=0}^{N}(-1)^{k-N}\binom{N}{N}\binom{N}{N}\right]x^N$$

This system can be written in a matrix form as follows

$$U_N(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^N \end{bmatrix}\begin{bmatrix} a_{0,0} & 0 & \cdots & 0 \\ a_{1,0} & a_{1,1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,0} & a_{N,1} & \cdots & a_{N,N} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_N \end{bmatrix} \quad \ldots (3)$$

i) A recursive definition of the Bernstein polynomials of degree N canbe written as:

$$B_{i.N}(x) = (1-x)B_{i,N-1}(x) + xB_{i-1,N-1}(x)$$

$$i = 1,2,\ldots,N$$

ii) The Bernstein Polynomials are all non-negative for $0 \le x \le 1$ (see[5]).

iii) The Bernstein polynomials form a partition of unity.

iv) Converting from Bernstein basis to power basis (proof see 5) as:

$$B_{i,N}(x) = \sum_{k=i}^{N}(-1)^{k-i}\binom{N}{k}\binom{k}{i}x^k$$

$i=1,2,\ldots,N$

$$U(x) \approx U_N(x) = a_0B_{0,N}(x) + a_1B_{1,N}(x) + \ldots + a_NB_{N,N}(x)$$

$$= a_0\sum_{k=0}^{N}(-1)^{k}\binom{N}{k}\binom{k}{0}x^k + \ldots +$$

$$+ a_N\sum_{k=N}^{N}(-1)^{k-N}\binom{N}{k}\binom{k}{N}x^k$$

$$= a_0 + a_1\left[\sum_{k=1}^{N}(-1)^{k-1}\binom{N}{1}\binom{1}{1}\right]x^1 + \ldots +$$

$$+ a_N\left[\sum_{k=0}^{N}(-1)^{k-N}\binom{N}{N}\binom{N}{N}\right]x^N$$

Using operator forms, equation (3) can be written as $L[U]=f(x)$ …… (4)

Using operator forms this equation (1) can be written as L [U] =f(x) … (5)

Where the operator L is defined for each type of delay intergo–differential equation as:

1.Retarded integro-differential equation

$$L\,[u(x)] = \frac{du(x)}{dx} - \int_0^x k(x,t)u(t-t)dt$$

2. .Neutral integro-differential equation

$$L\,[u(x)] = \frac{du(x-t)}{dx} - \int_0^x k(x,t)u(t)dt$$

3. .Mixed integro-differential equation

$$L\,[u(x)] = \frac{du(x-t_1)}{dx} - \int_0^x k(x,t)u(t-t_2)dt$$

The unknown function U(x) is approximated by the form

$$U_N(x) = \sum_{i=0}^{N} a_i B_i(x) \quad \ldots. \quad (6)$$

Substituting equation (6) in equation (5)

$$L[U_N] = f(x) + E_N(x)$$

Where

$$L[U_N(x)] = \sum_{i=0}^{N} c_i\left[\frac{dB_i(x)}{dx} - \int_0^x k(x,t)B_i(t-t)dt\right] L.O.RIDE$$

$$L[U_N(x)] = \sum_{i=0}^{N} c_i\left[\frac{dB_i(x-t)}{dx} - \int_0^x k(x,t)B_i(t)dt\right] L.O.NIDE$$

$$L[U_N(x)] = \sum_{i=0}^{N} c_i \left[\frac{dB_i(x-t_1)}{dx} - \int_0^x k(x,t) B_i(t-t_2)\,dt \right] L.O.MIDE$$

For which we have the residue equation

$$E_N(x) = L[U_N(x)] - f(x) \qquad ...(7)$$

Substituting (6) in (7) we get

$$E_N(x) = L\left(\sum_{i=0}^{N} a_i B_i(x)\right) - f(x) = \sum_{i=0}^{N} a_i L(B_i(x)) - f(x) \qquad ...(8)$$

Obviously the weighting function setting its weighted integral equal to zero

$$\int w_j E_N(x)dx = 0 \qquad ...(9)$$

Inserting (8) in (9)

$$\int w_j \left[\sum_{i=0}^{N} a_i L(B_i(x)) - f(x)\right] = 0$$

$$\sum_{i=0}^{N} a_i \int w_j L(B_i(x)) = \int w_j f(x)$$

Where

$$\left.\begin{aligned}
L(B_j(x)) &= \frac{dB_i(x)}{dx} - \int_0^x k(x,t) B_i(t-t)\,dt \\
L(B_j(x)) &= \frac{dB_i(x-t)}{dx} - \int_0^x k(x,t) B_i(t)\,dt \\
L(B_j(x)) &= \frac{dB_i(x-t_1)}{dx} - \int_0^x k(x,t) B_i(t-t_2) \\
&\text{for all } i = 0,1,..,N.
\end{aligned}\right\} (10)$$

Introducing Matrix K and vector H has

$$K_{ij} = \int w_j L(B_i(x))dx \qquad i = 0,1,...,N$$

$$H_j = \int w_j f(x) \qquad j = 0,1,...,N$$

$$\ldots (11)$$

Least square Method

The Least square method is one of the approximated methods used to solve delay volterra integro differential equations of the second kind In this method the weighting function is chosen as follow

$$w_j = L(B_j(x))$$

This leads to

$$\int L(B_j(x))L(B_i(x))a_i = L(B_i(x))f(x_i) \quad i = 0,1,...,N \qquad \ldots (12)$$

Hence (12) can be seen as a system of (N+1) equations in the (N+1) unknown a_i, i=0,1,…,N

$$\begin{bmatrix} \int L(B_0)L(B_0) & \int L(B_0)L(B_1) & . & . & \int L(B_0)L(B_N)dx \\ \int L(B_1)L(B_0) & \int L(B_1)L(B_1) & . & . & \int L(B_1)L(B_N)dx \\ . & . & . & . & . \\ . & . & . & . & . \\ \int L(B_N)L(B_0) & \int L(B_N)L(B_1) & . & . & \int L(B_N)L(B_N)dx \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ a_N \end{bmatrix} = \begin{bmatrix} \int L(B_0)fdx \\ \int L(B_1)fdx \\ . \\ . \\ \int L(B_N)fdx \end{bmatrix} \qquad \ldots (13)$$

Where

K=

$$\begin{bmatrix} \int L(B_0)L(B_0) & \int L(B_0)L(B_1) & . & . & \int L(B_0)L(B_N)dx \\ \int L(B_1)L(B_0) & \int L(B_1)L(B_1) & . & . & \int L(B_1)L(B_N)dx \\ . & . & . & . & . \\ . & . & . & . & . \\ \int L(B_N)L(B_0) & \int L(B_N)L(B_1) & . & . & \int L(B_N)L(B_N)dx \end{bmatrix}$$

$$A = \begin{bmatrix} a_0 \\ a_1 \\ . \\ . \\ a_N \end{bmatrix}, \quad H = \begin{bmatrix} \int L(B_0)fdx \\ \int L(B_1)fdx \\ . \\ . \\ \int L(B_N)fdx \end{bmatrix}$$

Acomputionally efficient way to calculate the value a_i is by solving the system

KA=H

For the cofficint aj's which satisfies (6) the approximated solution of (1) will be given

Algothim(ABIF)

Step 1 select B_j Bernstien polynomial

Step2 computed $L(B_j) \& L(B_i) \quad i,j=0,1,\dots,N$ using (10)

Step3 computed the Matrix K and H by using (11)

Step4 solve the system (13) for coefficients a_j's

Step5 substitute a_j's intransforming form to obtain the approximated solution of U(x).

Numerical Examples:

Example 1: Consider the Retarded Volterra integro differential equation he second kind:

$$U'(x) = f(x) + \int_0^x k(x,t)U(t-1)dt$$

where

$$f(x) = 1 - \frac{x^4}{3} + \frac{x^2}{2}$$

and let the linear kernel be

$$k(x,y) = xt$$

and the exact solution is taken to be $u(x)=x$.

For $h=0.1$ and $x=x_1=a+ih,\ i=0,1,\dots,10$

The tabulated result is obtained by applying the method involved in this paper i,e the implementation of Bernistein polynomial(ABIF); these numerical results are compared with the exact one in the same table below.

Example 2: Consider the Neutral Volterra integro equation of the second kind

$$u'(x-1) = f(x) + \int_0^x k(x,t)u(t)dt$$

where $f(x)=1-\frac{x^3}{6}$ and $k(x,t)=(x-t)$, and the exact solution is taken to be $u(x)=x$; the step size $h=0.1$. The application of Bernstein polynomial(ABIF)yields the results shown in the table below together with the exact solution at each point x. The the least square error and the consumed time.

Example 3: Consider the Mixed Volterra integro equation of the second kind

$$u'(x-1) = f(x) + \int_0^x k(x,t)u(t-1/2)dt$$

where $f(x)= 2x-2-\frac{x^5}{4}+\frac{x^4}{3}-\frac{x^3}{8}$ and $k(x,t)=(xt)$, and the exact solution is taken to be $u(x)=x^2$;the step size $h=0.1$. The applicationofBernsteinpolynomial(ABIF) yields the results shown in the table below together with the exact solution at each point x. The the least square error and the consumed time.

Conclusion: Bernstein Polynomial is introduced to find the approximate solution of delay Volterra integro differential equation of the second kind. Three numerical examples were submitted to illustrate the given idea with good approximate results were achieved. We conclude that:

1- In general , least square method with aid Bernstein polynomials have been applied to find the solution of linear delay Volterra integro differential equation and have proved their effectiveness from through finding accurate results.

2- A disadvantage of the Bernstein polynomials is its dependence upon a free parameter n that gives the smallest least square error.

References

1. Ali, H.A.; "Approximated Solutions of Linear Delay Integro-Diffeferential Equations, M.Sc. Thesis,College of Applied Science,Univ. of Technology, Iraq, Baghdad,2006.
2. Boor C.D., "A Practical Guide to Splines", New York 1978.
3. Cheney W. and Kincaid D., "Numerical Mathematics and Computing", 4th. ed.An international Thomson pub. co.1999.
4. Chakrabarti A.,Martha S.,"Approximate Solution of Fredholm Integral Equations of the Second Kind" ,Applied Mathematics and ComputationVol211,PP459-466(2009).
5. Lenneth I.J., "Bernstein Polynomials", Univ. of California, Davis 2000.
6. Pahpatie B.,"On Certain Volttera Integral and Integro Differential Equations" Ser. Mathematics Inform .,Vol 23,PP 1-12(2008).

Table (1) Presents acomparison the exact solution and approximate solution (Bernstein polynomial)

x	Exact sol.	ABIF
0.0	0.0000	0.0000
0.1	0.1000	0.1000
0,2	0.2000	0.2000
0.3	0.3000	0.3000
0.4	0.4000	0.4000
0.5	0.5000	0.5000
0.6	0.6000	0.6000
0.7	0.7000	0.7000
0.8	0.8000	0.8000
0.9	0.9000	0.9000
1.0	1.0000	1.0000
L.Sq.E		0.0000
Time		0.1880sec

Table (2) Presents acomparison the exact solution and approximate solution (Bernstein polynomial)

x	Exact sol.	ABIF
0.0	0.0000	0.0000
0.1	0.1000	0.1000
0,2	0.2000	0.2000
0.3	0.3000	0.3000
0.4	0.4000	0.4000
0.5	0.5000	0.5000
0.6	0.6000	0.6000
0.7	0.7000	0.7000
0.8	0.8000	0.8000
0.9	0.9000	0.9000
1.0	1.0000	1.0000
L.Sq.E		0.0000
Time		0.1880sec

Table (3)
Presents acomparison the exact solution and approximate solution (Bernstein polynomial)

x	Exact sol.	ABIF
0.0	0.0000	0.0000
0.1	0.01000	0.01000
0,2	0.04000	0.04000
0.3	0.09000	0.09000
0.4	0.1600	0. 1600
0.5	0.2500	0.2500
0.6	0.3600	0.3600
0.7	0.4900	0.4900
0.8	0.6400	0.6400
0.9	0.8100	0.8100
1.0	1.0000	1.0000
L.Sq.E		0.0000
Time		0.1880sec