Cubic Spline for SolvingVolterra Integral Equation with Delay

A.M.Ayal[1]Adhra'a M.Muhammad[2]
1E-mail: abdulbasetaial@yahoo.com
2E-mail:adhraa85 @yahoo.com
Department of Mathematics,
College of Basic Education
University ofMisan,
Misan-Iraq

Abstract:
In this paper, we presentCubic spline function is used to solve 2nd kind Volterra integral equations with delay. Numerical examples are presented to illustrate the applications of this method, Moreover, programs for methodis written in MATLAB language.
Keywords: Volterra integral equation with delay,Cubic spline function.

1. Introduction:

Many problems of mathematical physics can be started in the form of integral equations. These equations also occur as reformulations of other mathematical problems such as partial differential equations and ordinary differential equations. Numerical simulation in engineering science and in applied mathematics has become a powerful tool to model the physical phenomena, particularly when analytical solutions are not available then very difficult to obtain. Therefore, the study of integral equations and methods for solving them are very useful in application. In recent years, there has been a growing interest in the Volterra integral equations arising in various fields of physics and engineering [16]

Many researchers studied and discuss the using of non-polynomial spline to solve Volterra integral equation, Erica Jen and R.P.Srivastav[4] in 1981 introduced cubic splines and approximate solution of singular integral equations.Hermann Brunner[8]in 1982 introduced the Non-polynomial spline collocation for Volterraequation with weakly singular kernels. Sarah H.Harbi [17] in 2013 introduced algorithms for solving volterra integral equations using non-polynomial spline functions. Muna M.Mustafa and Sarah H.Harbi[11] in 2014 is used solution of second kind Volterra integral equation using non-polynomial spline function.Sarah H.Harbi, Mohammed A.Murad, Saba N.Majeed[16] in 2015 introduced a solution of second kind volterra integral equations using third order non-polynomial spline function. PeeterOja and DarjaSaveljeva [15] in 2002 studycubic spline collocation for vollterra integral equations.Hesam-EldienDeriliGherjalar and HosseinMohammadikia [7] in 2012 used numerical solution of functional integral and integro-differential equation by using B-Splines.

Also, volterra integral equation with delay.BaruchCahlon and Louis J.Nachman [2] in 1985 introuduced numerical solution of volterra integral equations with asolution dependent delay. BaruchCahlon [1] in 1990 study on the numerical stability of volterra integral equations with delay argument. George Karakostas,I.P.Stavroulakis and Yumiwv [6] in 1993 presented osciliations of volterra integral equation with delay. VilmosHorvat [18] in 1999 solved on collocation methods for volterra integral equation with delay arguments.Daniel Franco and DonaloRegen[3] in 2005 give solution of Volterra integral equation with infinit delay. Muna M. Mustafa and Thekra A. LatiffIbrahem [13] in 2008 studied numerical solution of volterra integral equation with delay using block methods. Ishtiaq Ali, Hermann Brunner and Tao Tang [9] in 2009 presented spectral methods for pantograph type differential and integral equations with multiple delays. M.Avaji, J.S.Hafshejani,S.S.Dehcheshmeh andD.F.Ghahfarokhi [12] in 2012 presented solution of delay volterra integral equation using the vaolational iteration method.Jose R.Moraies and EdixonM.Rojas [10] in 2011 discussed hyers-ulam and hyers-ulam-rassias stability of nonlinear integral equation with delay.ParvizDarania [14] in 2016 studied multistep colloction method for nonlinear delay integral equationFarshdMirzace,SaeedBimesl and EmranTohidi [5] in 2016 presenteda numerical framework for solving high-order pantograph-delay VolterraIntegro-differential equation.In this paper a solution of Volterra Integral Equation with delay are introduced using non-polynomial spline function.

2. Non-polynomial Spline Function Methods: [17]

Consider the partition $\Delta = \{t_0, t_1, t_2, \dots, t_n\}$ of [a, b] $\subset \mathbb{R}$. Let p(Δ) denote the set of piecewise polynomials subinterval $I_i = [t_i, t_{i+1}]$ of partition Δ . Let u (t) be the exact solution, this new method provides an approximation not only for u(t_i) at the knots but also $u^{(n)}(t_i)$, n=1,2,…,at every point in the range of integration. The non-polynomial spline function, obtained by the segment $p_i(t)$, each non-polynomial spline of n order $p_i(t)$ has the form:

$$p_i(t) = a_i \cos k(t - t_i) + b_i \sin k(t - t_i) + \cdots + y_i (t - t_i)^{n-1} + z_i \quad (1)$$

Where $a_i, b_i, \dots, y_i$ and z_i constants k is the frequency of the trigonometric functions which will be raise the accuracy of the method.

Now to introduce different of non-polynomial spline functions, linear non-polynomial spline function, the span of Cubic is x^5.

2.1 Cubic Non-Polynomial Spline Function.

The form of the cubic non-polynomial spline function is:

$$C_i(t) = a_i \cos k(t - t_i) + b_i \sin k(t - t_i) + c_i (t - t_i) + d_i (t - t_i)^2 + e_i (t - t_i)^3 + f_i \quad (2)$$

Where a_i, b_i, c_i, d_i, e_i and f_i are constant to be determined. In order to obtain the value a_i, b_i, c_i, d_i, e_i and f_i we differentiate equation (2) five times with respect to t, and then we get thefollowing:

$$\left.\begin{aligned}
C_i'(t) &= -k a_i \sin k(t - t_i) + k b_i \cos k(t - t_i) + c_i + 2d_i (t - t_i) + 3e_i (t - t_i)^2 \\
C_i''(t) &= -k^2 a_i \cos k(t - t_i) - k^2 b_i \sin k(t - t_i) + 2d_i + 6e_i (t - t_i) \\
C_i^{(3)}(t) &= k^3 a_i \sin k(t - t_i) - k^3 b_i \cos k(t - t_i) + 6e_i \\
C_i^{(4)}(t) &= k^4 a_i \cos k(t - t_i) - k^4 b_i \sin k(t - t_i) \\
C_i^{(5)}(t) &= -k^5 a_i \sin k(t - t_i) + k^5 b_i \cos k(t - t_i)
\end{aligned}\right\} \quad (3)$$

Hence replace t to t_i in the relation (2) and (3) yields:

$$C_i(t_i) = a_i + f_i$$
$$C_i'(t_i) = k b_i + c_i$$
$$C_i''(t_i) = -k^2 a_i + 2d_i$$
$$C_i^{(3)}(t_i) = -k^3 b_i + 6e_i$$
$$C_i^{(4)}(t_i) = k^4 a_i$$
$$C_i^{(5)}(t_i) = k^5 b_i$$

From the above equation, the values of a_i, b_i, c_i, d_i, e_i $and f_i$ are obtained as follows:

$$a_i = \frac{1}{k^4} C_i^{(4)}(t_i) \quad (4)$$

$$b_i = \frac{1}{k^5} C_i^{(5)}(t_i) \quad (5)$$

$$c_i = C_i'(t_i) - \frac{1}{k^4} C_i^{(5)}(t_i) \quad (6)$$

$$d_i = \frac{1}{2\left(C_i''(t_i) + \frac{1}{k^2} C_i^{(4)}(t_i)\right)} \quad (7)$$

$$e_i = \left(\frac{1}{6\left(C_i^{(3)}(t_i)\right)} - \frac{1}{k^2} C_i^{(5)}(t_i) \right) \quad (8)$$

$$f_i = C_i(t_i) - \frac{1}{k^4} C_i^{(4)}(t_i) \quad (9)$$

For i= 0, 1,…, n

3. The Solving Method:

Consider the Volterra integral equationwith delay of the second kind:

$$u(x) = f(x) + \int_0^x k(x,t)u(t-\tau)dt, \quad 0 \le x \le X \quad (10)$$

$$u(x) = \varphi(x), \quad x \in [-\tau, 0)$$

$\varphi(x),$

Where τ positive constant, u (x) is the unknown function and f(x), k(x,t) are givenfunction, this type of integral arises in certain application to impulse theory [2].In order to solve (10), we differentiate (10) five timeswith respect to x, by using Libenze formula, to get:

$$u'(x) = f'(x) + \int_a^x \frac{\partial k(x,t)}{\partial x} u(t-\tau)dt + k(x,x)u(x-\tau) \quad (11)$$

$$u''(x) = f''(x) + \int_a^x \frac{\partial^2 k(x,t)}{\partial x^2} u(t-\tau)dt + \left(\frac{\partial k(x,t)}{\partial x}\right)_{t=x} u(x-\tau) + \frac{\partial k(x,x)}{\partial x} u(x-\tau) + k(x,x)u'(x-\tau)$$

(12)

$$u^{(3)}(x) = f^{(3)}(x) + \int_a^x \frac{\partial^3 k(x,t)}{\partial x^3} u(t-\tau)dt + \left(\frac{\partial^2 k(x,t)}{\partial x^2}\right)_{t=x} u(x-\tau$$

(13)

$$u^{(4)}(x) = f^{(4)}(x) + \int_a^x \frac{\partial^4 k(x,t)}{\partial x^4} u(t-\tau)dt + \left(\frac{\partial^3 k(x,t)}{\partial x^3}\right)_{t=x} u(x-\tau) + \frac{\partial}{\partial x \left(\frac{\partial^2 k(x,t)}{\partial x^2}\right)_{t=x} u(x-\tau)} + \left(\frac{\partial^2 k(}{\partial x}\right.$$

(14)

$$u^{(5)}(x) = f^{(5)}(x) + \int_a^x \frac{\partial^5 k(x,t)}{\partial x^5} u(t-\tau)dt + \left(\frac{\partial^4 k(x,t)}{\partial x^4}\right)_{t=x} u(x-\tau) + \frac{\partial}{\partial x \left(\frac{\partial^3 k(x,t)}{\partial x^3}\right)_{t=x} u(x-\tau)} + \left(\frac{\partial^3 k(}{\partial x^3}\right.$$

(15)

To complete our procedure for solving eq(10), we substitute x=a in eq(10)-(15), then we get:

$u_0 = u(a) = f(a)$ (16)

$u_0' = u'(a) = f'(a) + k(a,a)u(a-\tau)$ (17)

$$u_0'' = u''(a) = f''(a) + \left(\left(\frac{\partial k(x,t)}{\partial x}\right)_{t=x}\right)_{x=a} u(a-\tau) + \left(\frac{\partial k(x,x)}{\partial x}\right)_{x=a} u(a-\tau) + k(a,a)u'(a-\tau)$$ (18)

u↓0↑((3)) = u↑((3)) (a) = f↑((3)) (a) + (((∂↑2 k(x,t))/(∂x↑2))↓(t = x))↓(x = a) u(a − τ) + (∂k(x,x)/∂x)
(19)

u↓0↑((4)) = u↑((4)) (a) = f↑((4)) (a) + (((∂↑3 k(x,t))/(∂x↑3))↓(t = x))↓(x = a) u(a − τ) + (∂/∂x((∂↑2 k(
(20)

u↓0↑((5)) = u↑((5)) (a) = f↑((5)) (a) + (((∂↑4 k(x,t))/(∂x↑4))↓(t = x))↓(x = a) u(a − τ) + (((∂↑3 k(x,t))/
(21)

4. Test Examples:

In this section, we give some of the numerical examples to illustrate the above methods for solving the Volterra integral equation with delay.

The exact solution is known and used to show that the numerical solution obtained with our methods is correct. We used MATLAB v 7.10 to solve the examples.

Example 1: Consider the following Volterra integral equation with delay:

$$u(x) = e^x - (x(e^x - 1))e^{-1} + \int_0^x (x)u(t-\tau)dt$$ (25)

Where $u(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} \quad , x \in [-1,0)$

With the exact solution $u(x) = e^{x}$.

Table (1)

Exact The Absolute Error of Example (1)by using non-polynomial spline function, with n=10 and h=0.1

x	*Exact solution*	*Cubic approximation Error*
0	**1**	**0**
0.1	**1.105170918075648e+000**	**1.352108424392373e-005**
0.2	**1.221402758160170e+000**	**6.015445385040286e-005**
0.3	**1.349858807576003e+000**	**1.470337526621979e-004**
0.4	**1.491824697641270e+000**	**2.786351455896785e-004**
0.5	**1.648721270700128e+000**	**4.567841954952412e-004**
0.6	**1.822118800390509e+000**	**6.807162662299038e-004**
0.7	**2.013752707470477e+000**	**9.471926923492902e-004**
0.8	**2.225540928492468e+000**	**1.250674697945722e-003**
0.9	**2.459603111156950e+000**	**1.583556800592798e-003**
1	**2.718281828459046e+000**	**1.936461207379110e-003**
$\|err\|_{\infty}$	**2.718281828459046e+000**	**1.936461207379110e-003**

Table (2)

Exact The Absolute Error of Example (1) by using **non-***polynomial spline function, with n=20 and h=0.05*

x	*Exact solution*	*Cubic approximation Error*
0	**1**	**0**
0.05	**1.051271096376024e+000**	**3.170296854448917e-006**
0.1	**1.105170918075648e+000**	**1.352108424392373e-005**
0.15	**1.161834242728283e+000**	**3.219087064509019e-005**
0.2	**1.221402758160170e+000**	**6.015445385039842e-005**
0.25	**1.284025416687741e+000**	**9.822114219774303e-005**
0.3	**1.349858807576003e+000**	**1.470337526621934e-004**
0.35	**1.419067548593257e+000**	**2.070684069367279e-004**
0.4	**1.491824697641270e+000**	**2.786351455896696e-004**
0.45	**1.568312185490169e+000**	**3.618793793450514e-004**
0.5	**1.648721270700128e+000**	**4.567841954952456e-004**
0.55	**1.733253017867395e+000**	**5.631735364272150e-004**
0.6	**1.822118800390509e+000**	**6.807162662299127e-004**
0.65	**1.915540829013896e+000**	**8.089311403568389e-004**
0.7	**2.013752707470477e+000**	**9.471926923492768e-004**
0.75	**2.117000016612675e+000**	**1.094738050688591e-003**
0.8	**2.225540928492468e+000**	**1.250674697945704e-003**

0.85	**2.339646851925991e+000**	**1.413989183537718e-003**
0.9	**2.459603111156950e+000**	**1.583556800592767e-003**
0.95	**2.585709659315846e+000**	**1.758152236668917e-003**
1	**2.718281828459046e+000**	**1.936461207379070e-003**
$\|\|err\|\|_{\infty}$	**2.718281828459046e+000**	**1.936461207379070e-003**

Example 2: Consider the following Volterra integral equation with delay:

$$u(x) = \cos(1) - \sin(x-1) + \cos(x) - 2x\sin(x-1) - x\sin(1) + \int_0^x (x+t)u(t-1)dt \quad (26)$$

With

Where the exact solution $u(x) = \cos(x)$.

Table (3)

Exact The Absolute Error of Example (2) by using non-polynomial spline function, with n=10 and h=0.1

x	*Exact solution*	*Cubic approximation Error*
0	**1**	**0**
0.1	**1.105170918075648e+000**	**1.290653103589556e-004**
0.2	**1.221402758160170e+000**	**4.604607864163679e-004**
0.3	**1.349858807576003e+000**	**9.192739291640795e-004**
0.4	**1.491824697641270e+000**	**1.442842235502309e-003**
0.5	**1.648721270700128e+000**	**1.981315942476345e-003**
0.6	**1.822118800390509e+000**	**2.498092750849260e-003**

0.7	**2.013752707470477e+000**	**2.970120184396005e-003**
0.8	**2.225540928492468e+000**	**3.388062563845097e-003**
0.9	**2.459603111156950e+000**	**3.756330927119211e-003**
1	**2.718281828459046e+000**	**4.092975596921729e-003**
$\|\|err\|\|_\infty$	**2.718281828459046e+000**	**4.092975596921729e-003**

Table (4)

Exact The Absolute Error of Example (2) by using non-polynomial spline function, with n=20 and h=0.05

x	*Exact solution*	*Cubic approximation Error*
0	**1**	**0**
0.05	**1.051271096376024e+000**	**3.409876392326483e-005**
0.1	**1.105170918075648e+000**	**1.290653103589734e-004**
0.15	**1.161834242728283e+000**	**2.744334713384278e-004**
0.2	**1.221402758160170e+000**	**4.604607864163857e-004**
0.25	**1.284025416687741e+000**	**6.781508725485896e-004**
0.3	**1.349858807576003e+000**	**9.192739291641150e-004**
0.35	**1.419067548593257e+000**	**1.176385327181121e-003**
0.4	**1.491824697641270e+000**	**1.442842235502362e-003**
0.45	**1.568312185490169e+000**	**1.712818243433661e-003**
0.5	**1.648721270700128e+000**	**1.981315942476416e-003**

0.55	**1.733253017867395e+000**	**2.244177436046260e-003**
0.6	**1.822118800390509e+000**	**2.498092750849349e-003**
0.65	**1.915540829013896e+000**	**2.740606128891834e-003**
0.7	**2.013752707470477e+000**	**2.970120184396093e-003**
0.75	**2.117000016612675e+000**	**3.185897915233319e-003**
0.8	**2.225540928492468e+000**	**3.388062563845168e-003**
0.85	**2.339646851925991e+000**	**3.577595328001973e-003**
0.9	**2.459603111156950e+000**	**3.756330927119336e-003**
0.95	**2.585709659315846e+000**	**3.926951035215255e-003**
1	**2.718281828459046e+000**	**4.092975596921853e-003**
$\|err\|_{\infty}$	**2.718281828459046e+000**	**4.092975596921853e-003**

5. Conclusion:

In this paper, non-polynomial spline function method for solving Volterra integral equations with delay of the second kind is presented successfully. According to the numerical results which obtain from the illustrative example, we conclude that:

- The approximate solutions obtained by MATLAB software show the validity and efficiency of the proposed method.
- The method can be extended and applied to nonlinear Volterra integral equation.
- The method can be extended also for solving nonlinearVolterraintegro equation of nth order.

References:

[1] BaruchCahlon, "On the Numerical Stability of Volterra Integral Equations with Delay Argument";Journal of Computational and Applied Mathematics, 33,97-104, **1990.**

[2] BaruchCahlon and Louis J.Nachman,"Numerical Solution of Volterra Integral Equations with asolution Dependent Delay"; Journal of Mathematical Analysis and Applications, 112,541-562, **1985.**

[3] Daniel Franco and O'Regan, "Solution of Volterra integral equations with infinit Delay ", **2005**.

[4] Erica Jen and R.P.Srivastav, "Cubic Splines and Approximate Solution of Singular Integral Equations ", vol(37),N(156),**1981**.

[5] FarshdMirzace,SaeedBimesl and EmranTohidi,"Anumerical framework for Solving high-order pantograph-delay VolterraIntegro-Differential Equation ";Kuwait,J,Sci.43(1)pp.69-83, **2016**.

[6] George Karakostas, I.P.Stavroulakis and Yumiwn,"Osciliations of Volterra Integral Equation with Delay ";Tohoka. Math.J.583-605, **1993**.

[7] Hesam-EldienDeriliGherjalar and HosseinMohammadikia,"Numerical Solution of Functional Integral and Integro-Differential Equation by Using B-Splines";Applied Mathematics, 3, 1940-1944, **2012**.

[8] Hermann Brunner,"Non-polynomial Spline Colloction for VolterraEquation with Weakly Singular Kernels"; SIAM J-Numer.Anal, 20(6), 1106-1119, **1982.**

[9]Ishtiaq Ali, Hermann Brunner and Tao Tang,"Spectral Methods for Pantagraph-Type Differential and Integral Equation with Multiple Delays";Front. Math. China, 4(1), 49–61, **2009**.

[10] Joser Morales and EdixonM.Rojas; "Hyers-Ulam and Hyers-Ulam-RasstasStabilty of Nonlinear Integral Equation with Delay"; Int. J. Nonlinear Anal. Appl. 2 (2011) No.2, 1-6, **2011**.

[11]Muna M. Mustafa and Sarah H.Harbi;"Solution of Second Kind Volterra Integral Equations Using Non-polynomial Spline Function"; Baghdad Science Journal, Vol.11, No.2, **2014.**

[12]M.Avaji,J.S.Hafshejani,S.S.Dehchesmehand D.F.Ghahfarokhi;"Solution of Delay Volterra Integral Equation Using the VariationalItreationMethod";Journal of Applied Sciences,12(2),196-200,**2012**.

[13] MunaM.Mustafaand ThekraA.LatiffIbrahem, "Numerical solution of Volterra Integral Equations with Delay Using Block Methods", AL-Fatih Journal.No.36, **2008.**

[14]ParvizDarania"Multistep Colloction Method for Nonliear Delay Integral Equation" (SCMA), Vol.3, No.2, 47-65, **2016**.

[15]PeeterOja and DarjaSaveljeva" Cubic Spline Collocation for Vollterra Integral Equations " Computing 69, 319-337, **2002**.

[16]Sarah H.Harbi, Mohammed A.Murad, Saba N.Majeed; "ASolution of Second KindVolterra Integral Equations Using Third Order Non-polynomial Spline Function"; Baghdad Science Journal, Vol.12, No.2, **2015.**

[17] Sarah H.Harbi;"Algorithms for Solving Volterra Integral Equations Using Non-polynomial Spline Functions " Thesis, University of Bahdad,**2013**.

[18] VilmosHorvat,"On Colloction Methods for Volterra Integral Equations with Delay Arguments", Mathematical communications, 4, 93-109, **1999**.