Numerical Solution of Volterra Integral Equation with Delay by Using Non-Polynomial Spline Function

Assist Lecturer .Adhra'a M.Muhammad
College of Basic Education, University of Misan,
E-mail:adhraa85 @yahoo.com

Abstract:
In this paper the non-polynomial spline function which include (first kind and second kind) will be applied to Volterra integral equation with delay. Moreover, programs for each types are written in MATLAB language. A comparison between two types has been made depending on the least squares errors.

Keywords: Volterra integral equation with delay, Non-polynomial spline.

الحل العددي لمعادلات فولتيرا التكاملية التباطؤية بأستخدام
دوال الثلمة الغير متعددة الحدود

المستخلص:
في هذا البحث دوال الثلمة الغير متعددة الحدود التي تتضمن (النوع الاول والنوع الثاني) طبقت على معادلة فولتيرا التكاملية التباطؤية.
علاوة على ذلك، تم كتابة البرامج الخاصة بكل طريقة بأستخدام لغة الماتلاب.
كما و تم اجراء المقارنة بين الطريقتين بأستخدام الاخطاء التربيعية.

1. Introduction:

Volterra integral equation arise in awide variety of Mathematical, scientific and engineering problems [2].

Many researchers studied and discuss the using of non-polynomial spline to solve Volterra integral equation, Hermann Brunner [1] in 1982 introduced the Non-polynomial spline collocation for Volterra equation with weakly singular kernels. Sarah H.Harbi [14] 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 and Saba N. Majeed [12] in 2015 presented a solution of second kind volterra integral equation using third order non-polynomial spline function.

Also, volterra integral equation with delay. Baruch Cahlon [3] in 1990 study on the numerical stability of Volterra integral equations with delay argument. George Karakostas,I.P.Stavroulakis and Yumiwv [4] in 1993 presented osciliations of Volterra integral equation with delay. Vilmos Horvat [5] in 1999 solved on collocation methods for Volterra integral equation with delay arguments. Daniel Franco and Donalo Regen [6] in 2005 give solution of Volterra integral equation with infinit delay. Muna M. Mustafa and Thekra A. Latiff Ibrahem [7] in 2008 studied numerical solution of volterra integral equation with delay using block methods. Ishtiaq Ali, Hermann Brunner and Tao Tang [8] in 2009 presented spectral methods for pantograph type differential and integral equations with multiple delays. M. Avaji, J.S.Hafshejani, S.S.Dehcheshmeh and D.F.Ghahfarokhi [9] in presented solution of delay volterra integral equation using the vaolational iteration method.Jose R.Moraies and Edixon M.Rojas [10] in 2011 discussed hyers-ulam and hyers-ulam-rassias stability of nonlinear integral equation with delay. Parviz Darania [13] in 2016 studied multistep colloction method for nonlinear delay integral 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: [14]

Consider the partition $\Delta=\{t_0, t_1, t_2, \dots, t_n\}$ of [a, b] $\subset \mathbb{R}$. Let S (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:

$$p_i(t) = a_i\, cosk(t - t_i) + b_i\, sink(t - t_i) + \cdots + y_i\,(t - t_i)^{n-1} + z_i \quad (1)$$

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 linear is x^3, quadratic non-polynomial spline functions, the span of linear is x^4 .

2.1 Linear Non-Polynomial Spline Function [14]:

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

$$p_i(t) = a_i\, cosk(t - t_i) + b_i\, sink(t - t_i) + c_i\,(t - t_i) + d_i \qquad , \quad i= 0,\ldots,n \qquad (2)$$

Where $a_i, b_i, c_i, and\ d_i$ are constants to be determined . In order to obtain the value $a_i, b_i, c_i\ and\ d_i$ we differentiate equation (2) three times with respect to t, then we get:

$$\left.\begin{aligned} p_i'(t) &= -ka_i sink(t - t_i) + kb_i cosk(t - t_i) + c_i \\ p_i''(t) &= -k^2 a_i cosk(t - t_i) - k^2 b_i sink(t - t_i) \\ p_i^{(3)}(t) &= k^3 a_i sink(t - t_i) - k^3 b_i cosk(t - t_i) \end{aligned}\right\} \qquad (3)$$

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

$$p_i(t_i) = a_i + d_i$$
$$p_i'(t_i) = kb_i + c_i$$
$$p_i''(t_i) = -k^2 a_i$$
$$p_i^{(3)}(t_i) = -k^3 b_i$$

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

$$a_i = -\frac{1}{k^2}\, p_i''(t_i) \qquad (4)$$

$$b_i = -\frac{1}{k^3}\, p_i^{(3)}(t_i) \qquad (5)$$

$$c_i = p_i'(t) + kb_i \qquad (6)$$

$$d_i = p_i(t_i) + a_i \qquad (7)$$

For i= 0,1,…,n

2.2 Quadratic Non-Polynomial Spline Function [14]:

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

$$Q_i(t) = a_i\, cosk(t - t_i) + b_i\, sink(t - t_i) + c_i\,(t - t_i) + d_i\,(t - t_i)^2 + e_i \qquad (8)$$

Where $a_i, b_i, c_i, d_i\ and\ e_i$ are constants to be determined. In order to obtain the value $a_i, b_i, c_i, d_i, and\ e_i$, we differentiate equation (8) four times with respect to t, and then we get the following:

$$Q_i'(t) = -ka_i sink(t-t_i) + kb_i cosk(t-t_i) + c_i + 2d_i((t-t_i)$$
$$Q_i''(t) = -k^2 a_i cosk(t-t_i) - k^2 b_i sink(t-t_i) + 2d_i$$
$$Q_i^{(3)}(t) = k^3 a_i sink(t-t_i) - k^3 b_i cosk(t-t_i) \quad (9)$$
$$Q_i^{(4)}(t) = k^4 a_i cosk(t-t_i) - k^4 b_i sink(t-t_i)$$

Hence replace t to t_i in the relation (8) and (9) yields:

$$Q_i(t_i) = a_i + e_i$$
$$Q_i'(t_i) = kb_i + c_i$$
$$Q_i''(t_i) = -k^2 a_i + 2d_i$$
$$Q_i^{(3)}(t_i) = -k^3 b_i$$
$$Q_i^{(4)}(t_i) = k^4 a_i$$

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

$$a_i = \frac{1}{k^4} Q_i^{(4)}(t_i)$$
(10)
$$b_i = -\frac{1}{k^3} Q_i^{(3)}(t_i)$$
(11)
$$c_i = Q_i'(t_i) + \frac{1}{k^2} Q_i^{(3)}(t_i)$$
(12)
$$d_i = \frac{1}{2}(Q_i''(t_i) + \frac{1}{k^2} Q_i^{(4)}(t_i))$$
(13)
$$e_i = Q_i(t_i) - \frac{1}{k^4} Q_i^{(4)}(t_i)$$
(14)

For i= 0,1,…,n

3. The Solving Method:

Consider the Volterra integral equation with delay of the second kind:

$$u(x) = f(x) + \int_0^x k(x,t)u(t-\tau)dt \quad , \quad 0 \le x \le X$$
(15)
$$u(x) = \varphi(x) \quad , x \in [-\tau, 0)$$

Where τ positive constant, u (x) is the unknown function and f(x), $\varphi(x)$, k(x,y) are given function, this type of integral arises in certain application to impulse theory

[3]. In order to solve (15), we differentiate (15) four times with respect to x, by using Libenze formula, to get:

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

(16)

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

(17)

$$u^{(3)}(x) = f^{(3)}(x) + \int_a^x \frac{\partial^3 k(x,t)}{\partial x^3} u(t-\tau)dt + (\frac{\partial^2 k(x,t)}{\partial x^2})_{t=x} u(x-\tau) + \frac{\partial}{\partial x}(\frac{\partial k(x,t)}{\partial x})_{t=x} u(x-\tau) + (\frac{\partial k(x,t)}{\partial x})_{t=x} u'(x-\tau) + \frac{\partial^2 k(x,x)}{\partial x^2} u(x-\tau) + 2\frac{\partial k(x,x)}{\partial x} u'(x-\tau) + k(x,x)u''(x-\tau)$$

(18)

$$u^{(4)}(x) = f^{(4)}(x) + \int_a^x \frac{\partial^4 k(x,t)}{\partial x^4} u(t-\tau)dt + (\frac{\partial^3 k(x,t)}{\partial x^3})_{t=x} u(x-\tau) + \frac{\partial}{\partial x}(\frac{\partial^2 k(x,t)}{\partial x^2})_{t=x} u(x-\tau) + (\frac{\partial^2 k(x,t)}{\partial x^2})_{t=x} u'(x-\tau) + \frac{\partial^2}{\partial x^2}(\frac{\partial k(x,t)}{\partial x})_{t=x} u(x-\tau) + 2\frac{\partial}{\partial x}(\frac{\partial k(x,t)}{\partial x})_{t=x} u'(x-\tau) + (\frac{\partial k(x,t)}{\partial x})_{t=x} u''(x-\tau) + \frac{\partial^3 k(x,x)}{\partial x^3} u(x-\tau) + 3\frac{\partial^2 k(x,x)}{\partial x^2} u'(x-\tau) + 3\frac{\partial k(x,x)}{\partial x} u''(x-\tau) + k(x,x)u'''(x-\tau)$$

(19)

To complete our procedure for solving eq(15), we substitute x=a in eq(15)-(19), then we get:

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

(20)

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

(21)

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

(22)

$$u_0^{(3)} = u^{(3)}(a) = f^{(3)}(a) + ((\frac{\partial^2 k(x,t)}{\partial x^2})_{t=x})_{x=a} u(a-\tau) + (\frac{\partial k(x,x)}{\partial x})_{x=a} u(a-\tau) + (\frac{\partial}{\partial x}(\frac{\partial k(x,t)}{\partial x})_{t=x})_{x=a} u(x-\tau) + (\frac{\partial k(x,t)}{\partial x})_{t=x})_{x=a} u'(x-\tau) + (\frac{\partial^2 k(x,x)}{\partial x^2})_{x=a} u(x-\tau) + \quad 2(\frac{\partial k(x,x)}{\partial x})_{x=a} u'(a-\tau) + k(a,a)u''(a-\tau)$$

(23)

$$u_0^{(4)} = u^{(4)}(a) = f^{(4)}(a) + ((\frac{\partial^3 k(x,t)}{\partial x^3})_{t=x})_{x=a} u(a-\tau) + (\frac{\partial}{\partial x}(\frac{\partial^2 k(x,t)}{\partial x^2})_{t=x})_{x=a} u(a-\tau) + (\frac{\partial^2 k(x,t)}{\partial x^2})_{t=x} u'(a-\tau) + (\frac{\partial^2}{\partial x^2}(\frac{\partial k(x,t)}{\partial x})_{t=x})_{x=a} u(a-\tau) + 2(\frac{\partial}{\partial x}(\frac{\partial k(x,t)}{\partial x})_{t=x})_{x=a} u'(a-\tau) + ((\frac{\partial k(x,t)}{\partial x})_{t=x})_{x=a} u''(a-\tau) + ((\frac{\partial^3 k(x,x)}{\partial x^3})_{t=x})_{x=a} u(a-\tau) + 3(\frac{\partial^2 k(x,x)}{\partial x^2})_{x=a} u'(a-\tau) + \quad 3(\frac{\partial k(x,x)}{\partial x})_{x=a} u''(a-\tau) + k(a,a)u'''(a-\tau)$$

(24)

4. Algorithms:

The following algorithms (linear and quadratic non-polynomial spline function) for solving Volterra integral equation with delay:-

4.1 The Algorithm (VIE2NPS1):

Step 1:
Set h= (b-a)/n, t_i=$t_0 + ih$, i=0,…, n,(where $t_0 = a$, t_n= b) and $u_0 = f(a)$.

Step 2:
Evaluate a_0, b_0, c_0 and d_0 by substituting (20)-(24) in equations (4)-(7).

Step 3:
Calculate $p_0(t)$ using step 2 and equation (3).

Step 4:
Approximate $u_1 = p_0(t_1)$.

Step 5:
For i=1 to n-1 do the following steps.

Step 6:
Evaluate a_i, b_i, c_i and d_i by using equations (4)-(7) and replacing. $u(t_i)$, $u'(t_i)$, $u''(t_i)$ and $u'''(t_i)$ by $p_i(t_i)$, $p_i'(t_i)$, $p_i''(t_i)$ and $p_i'''(t_i)$.

Step 7:
Calculate $p_i(t)$ using step 6 and equation (3).

Step 8:
Approximate $u_{i+1} = p_i(t_{i+1})$.
4.1 The Algorithm (VIE2NPS2):
Step 1:
Set h= (b-a)/n, $t_i = t_0 + ih$, i=0,…, n, (where $t_0 = a, t_n =$ b) and $u_0 = f(a)$.
Step 2:
Evaluate a_0, b_0, c_0 , d_0 and e_0 by substituting (20)-(24) in equations (10)-(14).
Step 3:
Calculate $p_0(t)$ using step 2 and equation (8).
Step 4:
Approximate $u_1 = p_0(t_1)$.
Step 5:
For i=1 to n-1 do the following steps.
Step 6:
Evaluate a_i, b_i, c_i, d_i and e_i by using equations (20)-(24) and replacing. $u(t_i)$, $u'(t_i)$, $u''(t_i)$, $u'''(t_i)$ and $u^{(4)}(t_i)$ by $p_i(t_i)$, $p_i'(t_i)$, $p_i''(t_i)$, $p_i'''(t_i)$ and $p_i^{(4)}(t_i)$.
Step 7:
Calculate $p_i(t)$ using step 6 and equation (8).
Step 8:
Approximate $u_{i+1} = p_i(t_{i+1})$.

5. 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 [7]:

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

Where $u(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!}$, $x \in [-1,0)$
With the exact solution $\mathrm{u}(\mathrm{x}) = e^x$.

Table (1): Exact and numerical solution for example (1)
Where $p_i(x)$ denotes the approximate solution using non-polynomial spline function, with h=0.1

x	Exact solution	$p_i(x)$	
		Linear	Quadratic
0	1	1.000000000000000	1.000000000000000
0.1	1.105170918075648	1.105228445332284	1.105141225650530
0.2	1.221402758160170	1.221791517640614	1.221063397768386
0.3	1.349858807576003	1.350918922611075	1.348361704741052
0.4	1.491824697641270	1.493714827271005	1.487423443384161
0.5	1.648721270700128	1.651146827517052	1.638424149199733
0.6	1.822118800390509	1.824036280216817	1.801325842483466
0.7	2.013752707470477	2.013050096482538	1.975877407805384
0.8	2.225540928492468	2.218694078115707	2.161617103072210
0.9	2.459603111156950	2.441307863802668	2.357877173101727
1	2.718281828459046	2.681061535557130	2.563790521611826

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

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

With $u(x) = x - \frac{x^3}{3!} \quad , x \in [-1,0)$

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

Table (2): Exact and numerical solution for example (2)
Where $p_i(x)$ denotes the approximate solution using non-polynomial spline function, with h=0.1

x	Exact solution	$p_i(x)$	
		Linear	Quadratic
0	0	0	0
0.1	0.099833416646828	0.099867357315368	0.099868051528372
0.2	0.198669330795061	0.198777912999978	0.198789009306851
0.3	0.295520206661340	0.295703116745105	0.295759198266039
0.4	0.389418342308651	0.389634255222335	0.389811087556149
0.5	0.479425538604203	0.479592530813702	0.480022957795431
0.6	0.564642473395035	0.564638841433945	0.565528110585558
0.7	0.644217687237691	0.643883163728112	0.645523528275526
0.8	0.717356090899523	0.716493445890723	0.719277897448583
0.9	0.783326909627483	0.781703921252371	0.786138915964148
1	0.841470984807897	0.838822759567154	0.845539810545178

6. 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 nonlinearVolterra integro equation of nth order.

References:

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

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

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

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

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

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

[7] Muna M.Mustafa and Thekra A.Latiff Ibrahem, "Numerical solution of Volterra Integral Equations with Delay Using Block Methods ", AL-Fatih Journal.No.36, **2008.**

[8] 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**.

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

[10] Joser Morales and Edixon M.Rojas; "Hyers-Ulam and Hyers-Ulam-Rasstas Stabilty 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] Sarah H.Harbi, Mohammed A.Murad, Saba N.Majeed; "A Solution of Second Kind Volterra Integral Equations Using Third Order Non-polynomial Spline Function "; Baghdad Science Journal, Vol.12, No.2, **2015.**

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

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