Solution of Second Kind Volterra Integro Equations Using linear Non-Polynomial Spline Function

Saba N. Majeed

College of Education for Pure Scince Ibn-Al Haitham, Baghdad, Addamia. Iraq .

E-mail: snoori39@yahoo.com

Abstract:

This research use linear non-polynomial Spline function to solve second kind Volterra integro equations . An algorithem introduced with numerical examples to illustrate the proposed method.

Keywords : Integro equation, Volterra second kind , Non-Polynomial Spline

1-Introduction:

Spline functions are piecewise polynomials of degree **n** joined together at the break points with **n-1** continuous derivatives. The break points of the splins are called knots [1].A piecewise non-polynomial spline function is a blend of trigonometric, as well as, polynomial basis functions, which form a complete extended Chebyshev space. This approach ensures enhanced accuracy and general form to the existing spline function . A parameter is introduced in the trigonometric part of the splin function of non-polynomial splines compensates for the loss of smoothes inherited by spline function.It is well known that the Bezire basis is a basis for the degree **n** algebraic polynomials $S_n = span\{1, x, x^2, \dots, x^n\}$

A new basis , called c-Bezire basis, is constructed in [2], for the space $\Gamma(n) = span\{1, x, x^2, x^n, \dots, x^{n-2}, \cos x, \sin x\}$

In which x^{n-1} and x^n in (1) replaced by **cos x** and **sin x** . There is wide use to non-polynomial spline functions , see [3,4,5,6,7,8,9].

Integro differential equations occure in various area of engineering , mechanics , physics , chemistry , astronomy , biology , economies , potential theory , electrostatics , ets…[10 , 11] . An integro-differential equation involves one (or more) unkown functions **u(t)** and both at its differential and integral , to solve this equations several analytics and numerical approaches have been proposed [12],[13].

Spline non-polynomial , proposed in this paper to solve second kind Volterra integro differential equation of the form

$$u'(x) + p(x)u(x) = f(x) + \int_a^x k(x,t)u(t)dt \quad \dots\dots\dots\dots\dots\dots.(1)$$
$$u(0) = u_0$$

The rest of this paper is organized as follows : section 2 introduce the non-polynomial spline function , section 3 perform it on Volterra integro equation of the second kind and demonstrate the solving process by discretiation with an essential algorithem ,section 4 offers three examples of second kind Volterra integro equation with non-polynomial spline approximation method , and finally section 5 conclude the paper .

2- Second order non-polynomial spline function :

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 on subinterval $I_i = [t_i, t_{i+1}]$ of partition Δ. In this work, we using second order non-polynomial spline function for finding approximate solution of Volttera integro differential of the second kind. Consider the grid point t_i on the interval [*a,b*] as follows:

$a=t_0 < t_1 < t_2 < \dots < t_n = b$ … (2)

$t_i = t_0 + ih \ ,i=0,1,\dots,n$ … (3)

$h = \frac{b-a}{n}$ … (4)

where n is a positive integer. The form of **n** order non-polynomial spline function is:

$$Q_i(t) = a_i cosk(t-t_i) + b_i sink(t-t_i) + c_i(t-t_i) + \dots + d_i(x-x_i)^{n-1} + m_i \dots (5)$$

where a_i, b_i, c_i, d_i and m_i are constant to be determined, and *k* is the frequency of the trigonometric functions which will be used to raise the accuracy of the method, therefore the linear non- polynomial spline has the form[8]

$$Q_i(t) = a_i cosk(t-t_i) + b_i sink(t-t_i) + c_i(t-t_i) + d_i \dots\dots (6)$$

obtained by the segment *Qi(t)*.

Let us consider the following relations:

$$Q'_i(t_i) = kb_i + c_i = u'(t_i) \approx S'_i(t_i)$$

$$Q''_i(t_i) = -k^2 a_i = u''(t_i) \approx S''_i(t_i)$$

$$Q'''_i(t_i) = -k^3 b_i = u'''(t_i) \approx S'''_i(t_i)$$

We can obtain the values of $a_i, b_i, c_i,$ and d_i as follows:

$$a_i = -\frac{1}{k^2} u''(t_i) \approx -\frac{1}{k^2} S''_i(t_i) \quad \dots (7)$$

$$b_i = -\frac{1}{k^3} u'''(t_i) \approx -\frac{1}{k^3} S'''_i(t_i) \quad \dots (8)$$

$$c_i = u'(t_i) + \frac{1}{k^2} u'''(t_i) \approx S'_i(t_i) + \frac{1}{k^2} S'''_i(t_i) \quad \dots (9)$$

$d_i = u_0 + \frac{1}{k}u''(t_i) \approx u_0 + \frac{1}{k}S_i''(t_i)$ … (10)

for $i=0,1,\dots,n$.

3-The method of solution :

Consider the Volterra integro differential equation of the second kind (1), in order to solve (1), we differentiate (1) two times with respect to x and then put x=a, to get:

$u(a) = u_0$ ………..(11)

$u_0' = u'(a) = -p(a)u(a) + f(a)$ ………..(12)

$u_0'' = u''(a) = -p'(a)u(a) - p(a)u'(a) + f'(a) + k(a,a)u(a)$ .. ………(13)

$u_0''' = u'''(a) =$
$-p(a)u''(a) - p''(a)u(a) - 2p'(a)u'(a) + f''(a) +$
$[(\frac{\partial k(a,a)}{\partial x})|_{t=x}]_{x=a}u(a) + \left[\frac{\partial k(x,x)}{\partial x}\right]|_{x=a}u(a) + k(a,a)u'$ …....(14)

Therefore, we approximate the solution of equation (1) using equation (6) in the following algorithm (IVNPS)=(Integro Volterra Non-Polynomial Spline) :

Algorithm (VIENPS):

To find the approximate solution of (1), first we select positive integer n, and perform the following steps:

Step 1: Set h=(b-a)/n; $t_i = t_0 + ih, i = 0,1,\dots,n, t_0 = a, t_n = b$ and $u_0 = u(0)$

Step 2: Evaluate a_0, b_0, c_0 and d_0 by substituting 11-14 in equations 7-10.

Step 3: Calculate $Q_0'(t)$ using step 2 and equation 6 for *i=0*.

Step 4: Approximate $Q_1' \approx Q_0'(t_1)$.

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

Step6: Evaluate a_i, b_i, c_i and d_i using equations7-10 and replacing $u'(t_i), u''(t_i)$ and $u'''(t_i)$ in $Q_i'(t_i), Q_i''(t_i)$ and $Q_i'''(t_i)$.

Step 8: Approximate $u_{i+1} = Q_i(t_{i+1})$.

4-Numerical Examples:

4.1- Example 1: consider the integro Volterra differential equation of the second kind:

$$\begin{cases} \emptyset'(x) + x\emptyset(x) = x(cosx - 1) - cosx + x^2cosx - xsinx + 2 + \int_0^x (x + t^2)dt \\ \emptyset(0) = 0 \end{cases}$$

The exact solution is $\emptyset(x) = sinx$. Results have been shown in Table 1,where $Q_i(x)$ denote the approximate solution by the proposed method and err=$|\emptyset(x)$ - $Q_i(x)|$.

See Table 1 fig.1.

4.2 - Example 2: consider the integro Volterra deferential equation of the second kind :

$$\begin{cases} y'(x) + y(x) = x^2 - \frac{x^2(7x^2-20x+18)}{12} - 1 + \int_0^x (t+x)y(t)dt \\ y(0) = 1 \end{cases}$$

The exact solution is $y(x) = x^2 - 2x + 1$, Results have been shown in Table 2,where $Q_i(x)$ denote the approximate solution by the proposed method and err=$|y_i(x)$ - $Q_i(x)|$, see table 2 fig.2 .

4.3- Example 3 : consider the integro Volterra deferential equation of the second kind :

$$\begin{cases} u'(x) + u(x) = 2e^x + e^x(x-1) + 1 + \int_0^x -tu(t)dt \\ u(0) = 1 \end{cases}$$

The exact solution is $u(x) = e^x$, Results have been shown in Table 3,where $Q_i(x)$ denote the approximate solution by the proposed method and err=$|u_i(x)$ - $Q_i(x)|$, see table 3 fig.

5.Conclusion:

In this paper, non-polynomial spline function method for solving Volterra integro equations of the second kind results is successful . This new idea based on the use of the VIE =(Volterra Integro Equation) and its derivatives. So it is necessary to mention that this approach can be used when f(x), p(x) and k(x,t) are analytic. The proposed scheme is simple and computationally attractive and their accuracy are high and we can execute this method in a computer simply. The numerical examples support this claim with explanation to the solution as a figures for each example followed after references.

References:

1. Rice,J. R. ,1985,Numerical Method Software and Analysis , macgra Hill.
2. G,Ch. ,2003, Class of Bbezier ,Aided Geom , Design, p.p 20,29,39.
3. Brunner,H. ,1983,Non-Polynomial Spline Collection for Volterra Equation with weakly Singular Kernals , SIAM Jornal on Numerical Analysis , 20(6):1106-1119.
4. Eldanafof,T. S. and Abdel Alaal,F. El. , 2009 ,Non-Polynomial Spline Method for the Solution of the Dissipative Wave Equation , International Jornal of Numerical Methods for Heat and Fluied Flow , 19(8) :950-959.

5. Hossinpour, A. ,2012 , The Solve of Integeral Differential Equation by Non-Polynomial Spline Function and Quadrature Formula , International Conference on Applide Mathematics and Pharmaceutical Science , Jan. 7-8 :595-597.
6. Quaraderoni,A. , Sacco,R. and Saleri , F. ,2000 , Numerical Mathematics ,Springer-Verlag New York , Inc.
7. Rashidinia,J. and Mohammadi, R. , 2009 , Non-polynomial Spline Approximation of Singularity Perturbed Boundery Value Problem , TWMSJ.Pure Apple.Math 1(2.2)251-336.
8. Zarebnia,M.,Hoshyar,M. and Sedahti,M. , 2011 , Non-Polynomial Spline Method for the Solution of Problem in Calculus of Variations . Word Academy Engenering Technology (51): 986-991.
9. THieme,H.R. ,1977, A model for the Spatial of the Epidemic . J. Math. Biol.,4:337-351 .
10. Wang,Sh. , Ji. He. , 2007, Variational Iteration Method for Solving Integro-differential equations . Physics Letters A, 367:188-191.
11. Al-Khalidi,S.,Mustafa,M.M. ,2013, Algorithems for Solving Volterra Integeral Equations Using Non-polynomial Spline Functions, MSC. Thesis, College of Science Women,University of Baghdad .
12. Babolian,E. and Masouri,Z. , 2010 , Numerical Solution of Volterra Integro Differential Equation By Direct Method Via-Block-Pluse Functions , J. Sci. Tarbiat Moallem University , Vol.(9),no. 1.
13. Fariborzi,M. A. and Kazemi, G. Ch. , 2012 , Numerical Solution of Integro Differential Equations Based on Double Exponential Transformation in the Sinc_Collocation Method , App. Math. And Comp. Intel. , Vol.1 , 48-55.

x	**Exact solution** $\emptyset(x)$	$Q_i(x)$	**error**
0.0	0	0	0
0.1	9.983341664682816e-002	9.983341664682816e-002	0
0.2	1.986693307950612e-001	1.986693307950612e-001	0
0.3	2.955202066613396e-001	2.955202066613396e-001	5.551115123125783e-017
0.4	3.894183423086505e-001	3.894183423086504e-001	1.110223024625157e-016
0.5	4.794255386042030e-001	4.794255386042029e-001	1.110223024625157e-016
0.6	5.646424733950354e-001	5.646424733950353e-001	1.110223024625157e-016
0.7	6.442176872376910e-001	6.442176872376909e-001	1.110223024625157e-016
0.8	7.173560908995228e-001	7.173560908995226e-001	2.220446049250313e-016
0.9	7.833269096274834e-001	7.833269096274831e-001	3.330669073875470e-016
1.0	8.414709848078965e-001	8.414709848078961e-001	4.440892098500626e-016

Table 1: Computed Absolute Error of Example (1)

X	Exact solution $y_i(x)$	$Q_i(x)$	error
0.0	1.000000000000000e+000	1.000000000000000e+000	0
0.1	8.100000000000001e-001	8.099916694439484e-001	8.330556051650007e-006
0.2	6.400000000000000e-001	6.398668443175168e-001	1.331556824831770e-004
0.3	4.900000000000000e-001	4.893270217487880e-001	6.729782512120419e-004
0.4	3.600000000000000e-001	3.578780119942295e-001	2.121988005770459e-003
0.5	2.500000000000000e-001	2.448348762192543e-001	5.165123780745740e-003
0.6	1.600000000000000e-001	1.493287701806432e-001	1.067122981935686e-002
0.7	9.000000000000008e-002	7.031562543102310e-002	1.968437456897698e-002
0.8	4.000000000000004e-002	6.586581305669359e-003	3.341341869433068e-002
0.9	1.000000000000001e-002	-4.321993654132861e-002	5.321993654132862e-002
1.0	0	-8.060461173627909e-002	8.060461173627909e-002

Table 2: Computed Absolute Error of Example (2)

x	Exact solution $y_i(x)$	$Q_i(x)$	error
0.0	1.000000000000000e+000	1.000000000000000e+000	0
0.1	1.105170918075648e+000	1.105162418075146e+000	8.500000501676652e-006
0.2	1.221402758160170e+000	1.221264091363697e+000	1.386667964724531e-004
0.3	1.349858807576003e+000	1.349143304213055e+000	7.155033629484553e-004
0.4	1.491824697641270e+000	1.489520663688465e+000	2.304033952805318e-003
0.5	1.648721270700128e+000	1.642991899505425e+000	5.729371194703070e-003
0.6	1.822118800390509e+000	1.810021911695287e+000	1.209688869522152e-002
0.7	2.013752707470477e+000	1.990940125477822e+000	2.281258199265479e-002
0.8	2.225540928492468e+000	2.185937199753314e+000	3.960372873915441e-002
0.9	2.459603111156950e+000	2.395063122101854e+000	6.453998905509595e-002
1.0	2.718281828459046e+000	2.618226709323966e+000	1.000551191350798e-001

Table 3: Computed Absolute Error of Example (3)

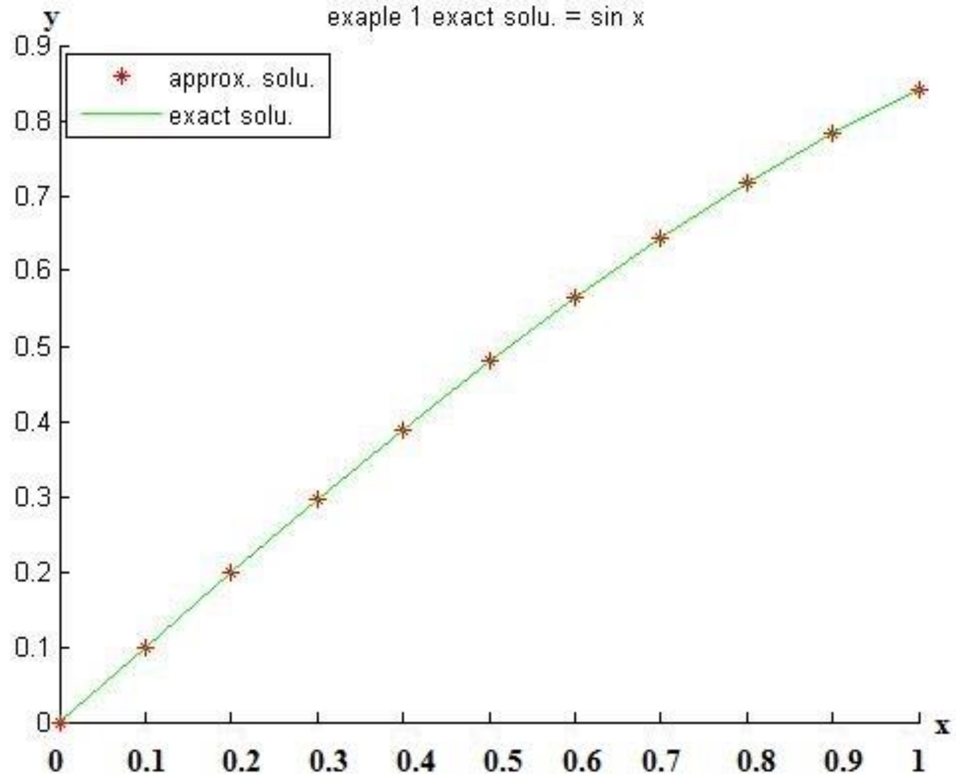

Fig.1 solution results of example 1

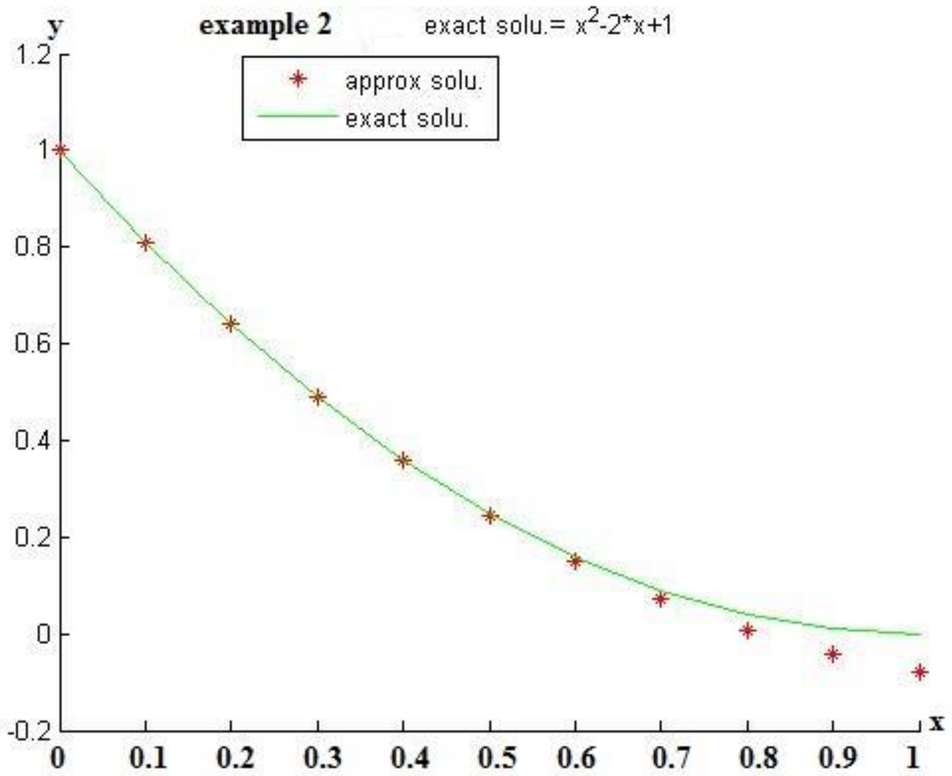

Fig. 2 solution results of example 2

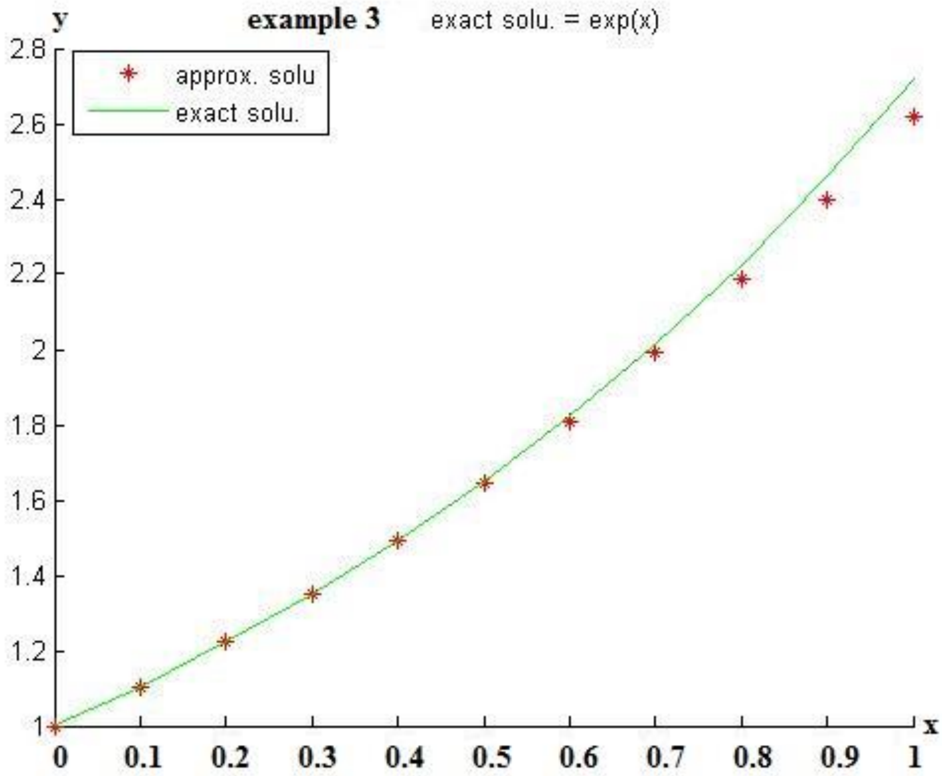

Fig.3 solution results of example 3