Thamar University Journal of Natural & Applied Sciences ISSN: 2073-0764
2012 A(5) 121-133

Classic Spline Functions for Approximate Solution of System of Non-Linear Volterra Integral Equations of the Second Kind

Debo M. Al-Rahal

Math Dept, College of Education (Sa'ada), Amran University, (Yemen), P.O. Box: 14034,, +967-713839052(Mobil),Email: deborahal1960@yahoo.com

ABSTRACT

In this paper we consider non-linear system of Volterra integral equations of the second kind (**NSVIEK2**), Three different kinds of classic spline functions (Linear, quadratic and cubic) have been modified and applied for the first time to treat the above system, A comparison between approximate and exact results for two numerical examples depending on the least-square error are given to show the accuracy of the results obtained by using this method, Programs are written in matlab program version 7.0.
Keywords: Three different kinds of classic spline function (Linear, quadratic and cubic), system of nonlinear Volterra integral equations.

1. INTRODUCTION

Splines are functions that are mathematically equivalent to physical spline used by draughtsman, A physical spline is a thin flexible rod held fixed at certain points but is free to move between them, and subject of course to laws of physics [8, 10].

Mathematical spline are piecewise polynomial, where pieces correspond to interval between points that hold physical spline fixed, a set of knots defines the intervals.

Al-Salhi; find numerical solution of non-linear Volterra integral equations of the second kind [3], Al-Kahachi; Approximate method for solving Volterra integral equations of the first kind [2], Abd-Al-Hammeed; Numerical solution of Fredholm integro-differential equations using spline functions [1].

2- CLASSIC SPLINE FUNCTIONS: [9]

Generally, impose a subdivision of the interval [a, b] as; Δ: $a=x_0<x_1<\ldots<x_{n-1}=b$,

And use a linear polynomial on each subinterval $[x_i, x_{i+1}]$; $i=0, 1,\ldots, n-1$,

Let $\Delta x_i=x_{i+1}-x_i$, then $|\Delta|=\max\{\Delta x_i\}$, we control accuracy by choosing $|\Delta|$ as small as is needed, thus ; under the assumption that the local error is mainly concentrated where the grid points are most distant, this of course; is not the entire story, as it also depends on the spectrum of the data itself,

We will define a class of functions; $S_m^k(\Delta)=\{s:s\in C^k[a,b],\ s_{[x_{i-1},x_i]}\in P_m,\ i=1,2,...,n-1\}$ ...(1),

Where m, k≥0 and $S_m^k(\Delta)$ "**classic spline functions of degree m and smoothness class k**" relative to the subdivision Δ, so $S_m^k(\Delta)$ has differentiable stricture of C^k,

The continuity assumption of (1) is that k^{th}, derivative of *s* is continuous everywhere on [a, b],and in particular; at the grid points x_i, *i*=0, 1,…, n-1 of Δ,

If k=m, then $s=S_m^m$ consist of polynomial of degree m and whole interval [a, b], we don't want this, so k<m,

Three types of classic spline functions has been considered, they are in following;

2.1 Linear Classic Spline S^1(x): [6]

Now; we want $s\in S_i^0(\Delta)$, s. t. for *f* on [a, b]; $S(x_i)=S_i^1$ where $S_i^1\in S^1(x_i),\ i=1,2,...,n$,

It will be assumed that the interpolation points concede with grid locations x_i, though that is not necessary, so we take; $S^1(x)=S_i^1+(x-x_i)\dfrac{(S_{i-1}^1-S_i^1)}{(x_{i-1}-x_i)}$,

Therefore; $S^1(x)=\left(\dfrac{x_{i+1}-x}{h}\right)S_i^1+\left(\dfrac{x-x_i}{h}\right)S_{i+1}^1$, for $x_i\le x_{i+1}\le x$, *i*=1, 2,…, n-1, . . . (2),

This shown schematically in figure (1);

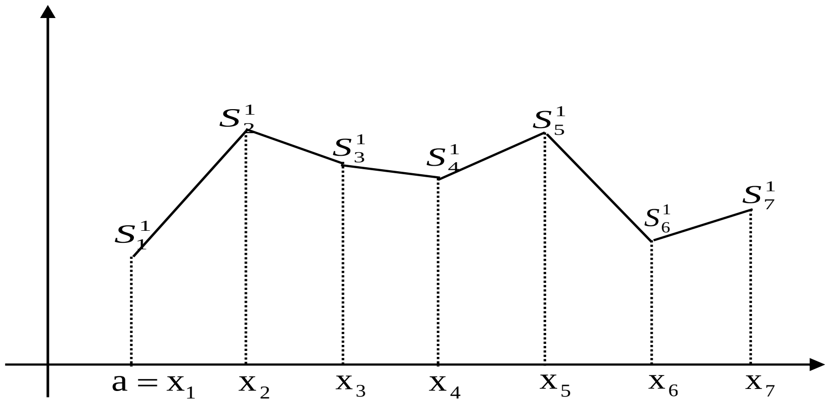

figure (1): Discretization of domain

2.2 Quadratic Classic Spline S^2(x): [9]

Quadratic classic Spline is continuously differentiable pricewise quadratic function which consists of n separated polynomial over each subinterval [x_i, x_{i+1}], i.e. $s\in S_2^1(\Delta)$, s. t. for *f* on [a, b]; $S(x_i)=S_i^2$ where $S_i^2\in S^2(x_i),\ i=1,2,...,n$,

Now; it is assumed that $S_i^2(\text{x}) = \text{a}_i\text{x}^2 + \text{b}_i\text{x} + \text{c}_i$, where $S^2(\text{x})$ is piecewise quadratic function which is continuously differentiable on each subinterval [x_i, x_{i+1}], $0\leq I \leq$n-1,
The formula for $S_i^2(\text{x})$ on [x_i, x_{i+1}] taken from some polynomials P_i of order 3, the i^{th} polynomial piece made to satisfy condition; $P_i(\text{x}_i)=f_i(\text{x}_i)$, $P_i(\text{x}_{i+1})=f_i(\text{x}_{i+1})$ and $P_i^{'}(\text{x}_i) = f_i^{'}(\text{x}_i)$, in order to compute coefficient of i^{th} polynomial piece P_i, Newton form used as; $P_i=P_i(\text{x}_i)+(\text{x}-\text{x}_i)[\text{x}_i, \text{x}_{i+1}]P_i+(\text{x}-\text{x}_i)^2[\text{x}_i, \text{x}_i, \text{x}_{i+1}]P_i$ ... (3),
And divided difference table for P_i given as;

	$P_i(\text{x}_i)$	$[\text{x}_i, \text{x}_{i+1}]P_i$	$[\text{x}_i, \text{x}_i, \text{x}_{i+1}]P_i$
x_i x_i x_i	$f_i(\text{x}_i)$ $f_i(\text{x}_i)$ $f_i(\text{x}_{i+1})$	$P_i^{'}(\text{x}_i) = \dfrac{dS_i^2(\text{x}_i)}{d\,\text{x}}$ $[\text{x}_i, \text{x}_{i+1}]P_i$	$\dfrac{[\text{x}_i, \text{x}_{i+1}]P_i - \dfrac{dS_i^2(\text{x}_i)}{d\,\text{x}}}{(\text{x}_{i+1} - \text{x}_i)}$

For above; if the table substituting in (3), yields;

$$S^2(\text{x}) = \left[1-\left(\frac{\text{x}-\text{x}_i}{h}\right)^2\right]S_i^2(\text{x}_i) + \left(\frac{\text{x}-\text{x}_i}{h}\right)^2\left(\frac{\text{x}-\text{x}_i}{h}\right)S_i^2(\text{x}_{i+1}) + \frac{(\text{x}-\text{x}_i)(\text{x}_{i+1}-\text{x})}{h}\frac{dS_i^2(x_i)}{d\,\text{x}} \quad . . . \ (4),$$

Since $s\in S_2^1(\Delta)$, that is $s\in C^1$[a, b], then we have; $S_2^1(\text{x}_{i+1}) = f(\text{x}_{i+1}),\ i = 0,1,...,n-1,$

Differentiating (4) w. r. t. x, gives; $\dfrac{d}{d\,\text{x}}S^2(\text{x}) = \dfrac{-2(\text{x}-\text{x}_i)}{h^2}S_i^2(\text{x}_i) + \dfrac{2(\text{x}-\text{x}_i)}{h^2}S_i^2(\text{x}_{i+1}) + \dfrac{-(\text{x}-\text{x}_i)+(\text{x}_{i+1}-\text{x})}{h}\dfrac{dS_i^2(x_i)}{d\,\text{x}},$

Putting x=x_{i+1}, yields; $\dfrac{d}{d\,\text{x}}S_i^2(\text{x}_{i+1}) = \dfrac{d}{d\,\text{x}}S_i^2(\text{x}_i) + 2\dfrac{\left(S_i^2(\text{x}_{i+1}) - S_i^2(\text{x}_i)\right)}{h}$, i. e. $\dfrac{dS_{i+1}^2}{d\,\text{x}} = \dfrac{dS_i^2}{d\,\text{x}} + 2\dfrac{\left(S_{i+1}^2 - S_i^2\right)}{h}\cdots$ (5),

2.3 Cubic Classic Spline S^3(x): [8]

Cubic classic Spline are the lowest order polynomial endowed with inflection points, and it is a pricewise technique, which is very popular.

To find a cubic Spline S^3(x) for which $S^3(\text{x})=\text{y}_i$, i=0, 1,…, n, . . . (6),

We begin by investigating how many degrees of freedom are left in the choice of S^3(x), once it satisfies (6), the technique used will not lead directly to practical means for calculating S^3(x), but will furnish additional insight,

Write; $S_i^2(\text{x}) = \text{a}_i + \text{b}_i\text{x} + \text{c}_i\text{x}^2 + \text{d}_i\text{x}^3$, $\text{x}_{i-1}\leq\text{x}\leq\text{x}_i$, i=1, 2,…, n, . . . (7),

There are 4_{n} unknown coefficients {a_i, b_i, c_i, d_i}, the constraints are (6) and continuity restrictions, thus; $\dfrac{d^j}{d\,\text{x}^j}S^3(\text{x}_i+0) = \dfrac{d^j}{d\,\text{x}^j}S^3(\text{x}_i-0)$ . . . (8), together gives; n+1+3(n-1)=4n-2,

Constraints; as compared with 4_{n} unknowns, thus there are at least two degrees of freedom in choosing the coefficient of (7),
Note that nothing has been said about value of S^3(x) on [x_0, x_{n}], it is easy to construct an extension to (-∞, ∞), although it will not be unique,

Now we give method for constructing S^3(x), Introduce notation $M_i = \frac{d^2}{d\,x^2} S^3(x_i),\ i = 0,1,\ldots,n,$,

Since; S^3(x) is cubic on [x_i, x_{i+1}], $\frac{d^2}{d\,x^2} S^3(x_i)$ is linear and thus;

$M_i = \frac{d^2}{d\,x^2} S^3(x_i) = \frac{(x_{i+1} - x)M_i + (x - x_i)M_{i+1}}{h_i},\ i = 0,1,\ldots,n-1,$ . . . (9), where h_i=x_{i+1}-x_i,

with this formula $\frac{d^2}{d\,x^2} S^3(x)$ is continuous on [x_0, x_n],

Integrate twice to get; $S^3(x) = \frac{(x_{i+1} - x)^3 M_i + (x - x_i)M_{i+1}}{6h_i} + C(x_{i+1} - x) + D(x - x_i),$ with C, D arbitrary,

The interpolating condition (6) implies; $C = \frac{y_i}{h_i} - \frac{h_i M_i}{6},\ D = \frac{y_{i+1}}{h_i} - \frac{h_i M_{i+1}}{6}$, and so;

$$S^3(x) = \frac{(x_{i+1} - x)^3 M_i + (x - x_i)M_{i+1}}{6h_i} + \frac{(x_{i+1} - x)y_i + (x - x_i)y_{i+1}}{h_i} - \frac{h_i}{6}\left[(x_{i+1} - x)^3 M_i + (x - x_i)M_{i+1}\right],\ x_i \le x \le x_{i+1},\ 0 \le i \le n-1, \quad ..(10)$$

Thus (10) implies continuity of S^3(x) on [a, b], as well as interpolating condition (6), To determine M_0, M_1,…, M_n we require $\frac{d^2}{d\,x^2} S^3(x)$ to be continuous at x_1, x_2,…, x_{n-1},

$$\lim_{x \to x_i^-} \frac{d^2}{d\,x^2} S^3(x) = \lim_{x \to x_i^+} \frac{d^2}{d\,x^2} S^3(x),\ i = 1,2,\ldots,n-1, \ldots \quad (11),$$

on [x_i, x_{i+1}]; $\frac{d}{d\,x} S^3(x) = \frac{-(x_{i+1} - x)^2 M_i + (x - x_i)^2 M_{i+1}}{2h_i} + \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{6}\left[M_{i+1} - M_i\right],$,

and on[x_{i-1}, x_i]; $\frac{d}{d\,x} S^3(x) = \frac{-(x_i - x)^2 M_{i-1} + (x - x_{i-1})^2 M_i}{2h_{i-1}} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{6}\left[M_i - M_{i-1}\right],$,

Using (11) and some manipulation, yield; $\frac{h_{i-1}}{6} M_{i-1} + \frac{h_i + h_{i-1}}{3} M_i + \frac{h_i}{6} M_{i+1} = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}},$ ... (12),

For i=1, 2,…, n-1, this gives n-1 equations for n+1 unknowns M_0, M_1,…, M_n, we generally specify an endpoint condition at x_0, x_n, to remove two degrees freedom as in (12),

Case1: "The cubic natural interpolating spline",

One particular choice of endpoint condition is to use; M_0=M_n=0 . . . (13),

This may appear, but it is based on the solution of following important problem, Among all functions g(x), which are twice continuously differentiable on [a, b], and satisfy; $g(x_i)=y_i$, i=0, 1,…, n, . . . (14), Choose g minimizes integral; $\int_a^b \left[g^n(x), (x)\right]^2 d\,x$,

The solution of problem above will be an interpolating function to the data in (14), And it should contain a minimum of oscillatory behavior g^n(x) because is small,

The solution is cubic spline S^n(x), which satisfies (12) and (13),

The condition $\frac{d^2 S^3(x_0)}{d\,x^2} = \frac{d^2 S^3(x_n)}{d\,x^2} = 0$, has interpolation that S^3(x) is linear on (-∞, x_0), (x_n, ∞),

This can be given additional physical meaning by using theory of small deformation of thin beam of an elastic material,
With the choice of (13) we have the system of equations; $A\,M = D$ . . . (15), S. t;

$$A=\begin{bmatrix} \frac{h_0+h_1}{3} & \frac{h_2}{6} & 0 & \cdots & 0 \\ \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & \cdots & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ 0 & \cdots & & \frac{h_{n-2}}{6} & \frac{h_{n-1}+h_{n-2}}{3} \end{bmatrix},\quad M=\begin{bmatrix} M_1 \\ M_2 \\ . \\ . \\ . \\ M_{n-1} \end{bmatrix},$$

$$\text{and}\quad D=\left[\frac{h_2-h_1}{h_1}-\frac{h_1-h_0}{h_0},\ldots,\frac{h_n-h_{n-1}}{h_{n-1}}-\frac{h_{n-1}-h_{n-2}}{h_{n-2}}\right]'$$

This matrix is symmetric, positive definite and diagonally dominant, also the associated linear system (15) can be easily and rapidly solved,
Case1: "Endpoint derivative conditions",

To specify M_0, M_n implicitly, impose condition $\frac{d}{d\,x}S^3(x)$ at endpoints;

$$\frac{d}{d\,x}S^3(x_0)=y_0',\ \frac{d}{d\,x}S^3(x_n)=y_n' \quad . . . \ (16),$$

this leads to additional equations;

$$\frac{h_0}{3}M_0+\frac{h_0}{6}M_1=\frac{y_1-y_0}{h_0}-y_0',\ \frac{h_{n-1}}{6}M_{n-1}+\frac{h_{n-1}}{3}M_n=y_n'-\frac{y_n-y_{n-1}}{h_{n-1}} \quad . . . \ (17),$$

The system composed of (13) and (17) is of order n+1, and it has exactly the same properties as the earlier matrix *A* in (15) of case1.

3- THE ERRORS OF CLASSIC SPLINE FUNCTIONS

3-1 Theorem:

Let $x_0, x_1,\ldots, x_n$, be distinct real numbers, and *f* be given real-value function of C^{n+1}, i. e. *f* has n+1-derivative on interval $I_t=\aleph\{t, x_0, x_1,\ldots, x_n\}$, with *t* some given real number, then there exist $\xi\in I_t$, s. t. $f(t)-p_n(t)=\frac{(t-x_0)\ldots(t-x_n)}{(n+1)!}f^{n+1}(\xi)$,

Where $p_n(t)=\sum_{j=0}^{n} f(x_j)\,l_j(t)$, and that $l_j(t)=\prod_{j\neq i}\left(\frac{t-x_j}{x_i-x_j}\right)$, $i=0,1,\ldots,n$,

Proof:

Note that result is trivially true if *t* is any node point, since then both sides of (18) are zero,

Assume *t* does not equal node point, define; $E(x)=f(x)-p_n(x)$, $G(x)=E(x)-\frac{\psi(x)}{\psi(t)}E(t),\ \forall\ x\in I_t$,

With $\psi(x)=(x-x_0)(x-x_1)\ldots(x-x_n)$, so $G(x)$ is n+1time differentiable on I_t, as $E(x)$, $\psi(x)$ are,

Also $G(x_i) = E(x_i) - \frac{\psi(x_i)}{\psi(t)} E(t),\ i = 0,1,\ldots,n$ and $G(t) = E(t) - E(t) = 0$, thus G has n+2 distinct zero in I_t,
when one uses the mean value theorem, G' has n+2distinct zero, inductively $G^{(j)}(x)$ has n+2-j zero in I_t, for j=0, 1,…, n+1,
Let ξ be the zero of $G^{(n+1)}(x)$, $G^{(n+1)}(\xi)=0$, since $E^{(n+1)}(x)=f^{(n+1)}(x)$, $\psi^{(n+1)}(x)=(n+1)!$, we obtain; $\psi^{(n+1)}(x) = f^{(n+1)}(x) - \frac{(n+1)!}{\psi(t)} E(t)$, substituting x=ξ and solving for $E(t)$, $E(t) = \frac{\psi(t)}{(n+1)!} f^{(n+1)}(\xi)$, give the desired result, and this may seem a tricky derivation, but it is commonly used technique for obtaining some error formulas,

4- SOLUTION OF SYSTEM OF NON-LINEAR VIE'S USING CLASSIC SPLINE FUNCTIONS

In this section, we use classic Spline functions "quadratic and cubic" to find the numerical solution of system of non-linear VIE's of the form;

$u_i(x) = f_i(x) + \sum_{j=i}^{m} \int_0^x K_{ij}(x, t, u_j(t))\, dt, \quad x \in I = [a,b] \cdots$(19), where $m \in N$, f_i s. t. i=1, 2,…, n, assumed to be continuous on I, and K_{ij} s. t. i, j=1, 2,…, m, denotes given continuous functions,

4-1 Using Linear Classic Spline Function $S^1(t)$:

A linear interpolation $S^1(t)$ with knots $t_0, t_1, \ldots, t_n$, in interval $[t_i, t_{i+1}]$ is given by; $S^1(t) = A_i(t) S_i^1 + B_i(t) S_{i+1}^1$ . . . (20), where , $A_i(t) = \frac{t_{i+1} - t}{h}$ *and* $B_i(t) = \frac{t - t_i}{h}$, substituting equation (20) into (19) gives; $S_i^1,\ i = 1,2,\ldots n$, with $t = t_r$ and $r = 1,2,\ldots n$, so we get;

$$S_{ir}^1 = f_{ir} + \sum_{j=1}^{m} \sum_{z=0}^{r-1} \left(\int_{x_z}^{x_{z+1}} K_{ij}\left(x_r, t[A_z(t) S_{jz}^1 + B_z(t) S_{j,z+1}^1]\right) dt \right) \quad i = 1,2,\ldots,m \text{ and } r = 1,2,\ldots,n, \ldots(21)$$

Where $S_{ir}^1 = S_i^1(t_r)$ and $f_{ir} = f_i(t_r)$ and the iterated integral above are calculated using Trapezoidal rule, the following algorithm is considered as solve a system of non-linear VIE's using linear classic Spline function as follows;

The algorithm (NSVIELSP);

Step1:

a- Assume $h = \frac{b}{n},\ n \in N$,

b- Set $S_{i0}^1 = f_{i0},\ i = 1,2,\ldots,m$,

Step2:

To compute $S_{i1}^1,\ i = 1,2,\ldots,m$ we use step1, put r=1 in (21) and using algorithm (NLVIETRP),

Step3: In same way as in step2, and using (21) we compute $S_{1r}^1, S_{2r}^1, \ldots, S_{mr}^1,\ r = 2,3,\ldots,n$,

4-2 Using Quadratic Classic Spline Function $S^2(t)$:

Quadratic Classic Spline $S^2(t)$ with knots t_0, t_1,…, t_n, in $[t_i, t_{i+1}]$ can be written as;

$$S^2(t) = A_i(t)S_i^2 + B_i(t)S_{i+1}^2 + D_i(t)\frac{dS_i^2}{dt} \quad . \quad . \quad . \quad (22),$$

where; $A_i(t) = 1 - \left(\frac{t-t_i}{h}\right)^2$, $B_i(t) = 1$ and $D_i(t) = \frac{(t-t_i)(t_{i+1}-t)}{h}$,

Putting (22) into (19) gives; u_i; i=1,2,…,n, with $t=t_r$, r=1,2,…,n, we get;

$$S_{ir}^2 = f_{ir} + \sum_{j=1}^{m}\sum_{z=0}^{r-2}\left\{\int_{x_z}^{x_{z+1}} K_{ij}\left(x_r, t[A_z(t)S_{jz}^2 + B_z(t)S_{j,z+1}^2 + D_z(t)\frac{dS_{jz}^2}{dt}]\right)dt\right\} \quad \ldots(23),$$

$$+\sum_{j=1}^{m}\left\{\int_{x_{r-1}}^{x_r} K_{ij}\left(x_r, t[A_{r-1}(t)S_{j,r-1}^2 + B_{r-1}(t)S_{j,r}^2 + D_r(t)\frac{dS_{j,r-1}^2}{dt}]\right)dt + \right\} \quad i = 1,2,\ldots,m,$$

Where $S_{ir}^2 = S_i^2(t_r)$ and $f_{ir} = f_i(t_r),\ i = 1,2,\ldots,m,$

Now for r=1 we need to calculate $\frac{dS_{j0}^2}{dt}$, $j = 1,2,\ldots,m$, we can find this value by differentiating (19) one-time w. r. t. x, so we get;

$$u_i'(x) = f_i'(x) + \sum_{j=i}^{m}\int_0^x \frac{dK_{ij}(x,t,u_j(t))}{dx}dt + \sum_{j=i}^{m} K_{ij}(x,x,u_j(x)),\ j = 1,2,\ldots,m,$$ putting t=a, we obtain;

$$u_{i0}' = f_{i0}' + \sum_{j=i}^{m} K_{ij}(a,a,u_{j0}),\ j = 1,2,\ldots,m, \ . \ . \ . \ (24),$$

But for r=2,3,…,n we calculate $\frac{dS_{jr}^2}{dt}$ by; $\frac{dS_{jr}^2}{dt} = \frac{dS_{j,r-1}^2}{dt} + \frac{2(S_{jr}^2 - S_{j,r-1}^2)}{h},\ j = 1,2,\ldots,n \ . \ . \ . \ (25),$

The following algorithm is considered to solve a system of non-linear VIE's using quadratic classic Spline functions,

The algorithm (NSVIEQSP);

Step1:

a- Assume $h = \frac{b}{n},\ n \in N$,

b- Set $S_{i0}^2 = f_{i0},\ i = 1,2,\ldots,m$,

Step2:

a- To calculate $\frac{dS_{j0}^2}{dt}$ we use step1, with equation (24),

b- To compute $S_{i1}^2,\ i = 1,2,\ldots,m$ we use step1 and step2-a, put r=1 in (23) and using algorithm (NLVIETRP),

Step3:

a- Using step1, step2, to find $\frac{dS_{j1}^2}{dt}$, by putting i=0 in (25),

b- Putting r=2 in (23) to find $S_{i2}^2,\ i=1,2,...,m$, and using algorithm (NLVIETRP),

Step4:

In same way as in step3, and using (21, 25) we compute;

$\frac{dS_{j2}^2}{dt},\ S_{j3}^2,\ \frac{dS_{j3}^2}{dt}, S_{j4}^2,...$ and so on, $j=1,2,...,m$,

4-3 Using Cubic Classic Spline Function $S^3(t)$:

A cubic Classic Spline $S^3(t)$ with knots $t_0, t_1,\dots, t_n$, in $[t_i, t_{i+1}]$ can be written as;

$$S^3(t) = A_i(t)S_i^3 + B_i(t)S_{i+1}^3 + C_i(t)\frac{dS_i^3}{dt} + D_i(t)\frac{dS_{i+1}^3}{dt} \quad \dots \quad (26), \text{ where;}$$

$$A_i(t) = 1-3\left(\frac{t-t_i}{h}\right)^2 + 2\left(\frac{t-t_i}{h}\right)^3,\ B_i(t) = 1-A_i(t),\ C_i(t) = (t-t_i)\left(\frac{t-t_{i+1}}{h}\right)^2 \text{ and } D_i(t) = (t-t_{i+1})\left(\frac{t-t_i}{h}\right)^2,$$

Substituting (26) in (19) gives; u_i; i=1,2,…,n, with $t=t_r$, r=1,2,…,n, we get;

$$S_{ir}^3 = f_{ir} + \sum_{j=1}^{m}\sum_{z=0}^{r-2}\left\{\int_{x_z}^{x_{z+1}} K_{ij}\left(x_r, t[A_z(t)S_{jz}^3 + B_z(t)S_{j,z+1}^3 + C_z(t)\frac{dS_{jz}^3}{dt} + D_z(t)\frac{dS_{j,z+1}^2}{dt}]\right)dt\right\}$$

$$+\sum_{j=1}^{m}\left\{\int_{x_{r-1}}^{x_r} K_{ij}\left(x_r, t[A_{r-1}(t)S_{j,r-1}^3 + B_{r-1}(t)S_{j,r}^3 + C_z(t)\frac{dS_{j,r-1}^3}{dt} + D_r(t)\frac{dS_{jr}^3}{dt}]\right)dt + \right\}\ i=1,2,...,m, \quad ,..(27)$$

,

Where $S_{ir}^3 = S_i^3(t_r)$ and $f_{ir} = f_i(t_r),\ i=1,2,...,m$, we calculate $\frac{dS_{jr}^3}{dt},\ j=1,2,...,m$, by equation;

$$\frac{dS_{j,r+1}^3}{dt} = -\frac{dS_{j,r-1}^3}{dt} - 4\frac{dS_{jr}^3}{dt} + \frac{3(S_{j,r+1}^3 - S_{j,r-1}^3)}{h},\ j=1,2,...,n \quad \dots \quad (28),$$

The following algorithm is considered to solve a system of non-linear VIE's using Cubic classic Spline functions,

The algorithm (NSVIECSP);

Step1:

Assume $h=\frac{b}{n},\ n\in N$, and set $S_{i0}^3 = f_{i0},\ i=1,2,...,m$,

Step2:

a- To calculate $\frac{dS_{j0}^3}{dt}$ we use step1, with equation (24),

b- To compute $S_{i1}^3,\ i=1,2,...,m$, use step1, 2-a, put r=1 in (27) and use (NLVIETRP),

Step3:

a- Using step1, step2, to find $\frac{dS_{j1}^3}{dt}$, by putting i=0 in (28), b- Putting *r*=2 in (21) to find $S_{i2}^3,\ i=1,2,...,m$, and using algorithm (NLVIETRP),

Step4:

In same way as in step3, and using (27, 28) we compute;

$$\frac{dS_{j2}^3}{dt},\ S_{j3}^3,\ \frac{dS_{j3}^3}{dt}, S_{j4}^3, \text{...and so on},\quad j=1,2,...,m,$$

5- NUMERICAL EXAMPLES

5-1 Example:

Consider problem; $u_1(x)=\frac{1}{4}(1-e^{2x})+\int_0^x (x-t)(u_2(t))^2\,dt$ and $u_2(x)=-(xe^x-2e^{2x}+1)+\int_0^x te^{-2u_1(t)}\,dt$,

Which is system of two non-linear VIE's with; $u_1(x)=-\frac{1}{2}x,\ u_2(x)=e^x$,

Tables (1, 2), present comparison between exact and numerical solution using classic Spline "Quadratic and Cubic", for respectively depending on last square error and running time with *h*=0.1,

Table(1),

X	Exact1	LSP	QSP	CSP
0.0	0	0	0	0
0.1	-5.0000e-002	-5.2151e-002	-5.2151e-002	-5.0351e-002
0.2	-1.0000e-001	-1.0356e-001	-1.0294e-001	-1.0074e-001
0.3	-1.5000e-001	-1.5456e-001	-1.5384e-001	-1.5117e-001
0.4	-2.0000e-001	-2.0642e-001	-2.0503e-001	-2.0165e-001
0.5	-2.5000e-001	-2.5827e-001	-2.5638e-001	-2.5216e-001
0.6	-3.0000e-001	-3.0903e-001	-3.0832e-001	-3.0285e-001
0.7	-3.5000e-001	-3.6983e-001	-3.6043e-001	-3.5315e-001
0.8	-4.0000e-001	-4.4377e-001	-4.1377e-001	-4.0558e-001
0.9	-4.5000e-001	-4.6974e-001	-4.6712e-001	-4.6108e-001
1.0	-5.0000e-001	-5.4390e-001	-5.2290e-001	-5.0261e-001
L. S. E.		6.2481e-002	1.2791e-003	1.8818e-004
R. T.		0.118000	0.141000	0.156000

Table(2),

X	Exact2	LSP	QSP	CSP
0.0	1.0000e+000	1.0000e+000	1.0000e+000	1.0000e+000
0.1	1.1052e+000	1.1053e+000	1.1053e+000	1.1053e+000
0.2	1.2214e+000	1.2207e+000	1.2207e+000	1.2218e+000
0.3	1.3499e+000	1.3484e+000	1.3484e+000	1.3504e+000

0.4	1.4918e+000	1.4889e+000	1.48893e+000	1.4925e+000
0.5	1.6487e+000	1.6449e+000	1.6449e+000	1.6494e+000
0.6	1.8221e+000	1.8159e+000	1.8159e+000	1.8227e+000
0.7	2.0138e+000	2.0067e+000	2.0067e+000	2.0135e+000
0.8	2.2255e+000	2.2154e+000	2.2154e+000	2.2261e+000
0.9	2.4596e+000	2.4488e+000	2.4488e+000	2.4662e+000
1.0	2.7183e+000	2.7052e+000	2.7052e+000	2.7108e+000
L. S. E.		7.1242e-002	5.0352e-003	1.0218e-004
R. T.		0.118000	0.141000	0.156000

Tables (3), presents the error and running time for Cubic classic Spline when *h* is changing and initial value is fixed,

	H=0.1	H=0.05	H=0.02	H=0.01
CSP	1.8818e-004	2.0280e+919	8.3653e+067	NaN
	1.0218e-004	8.8485e+006	2.1605e+003	NaN
R. T.	0.156000	0.421000	2.235000	9.047000

Tables (4), presents the error and running time for Cubic classic Spline when initial value is changing and h is fixed,

I.V.	u_1	u_2	**R.T.**
0.0	1.8818e-004	1.0218e-004	0.156000
0.001	3.0520e-004	3.4644e-005	0.141000
0.01	2.5423e-003	1.2761e-002	0.140000
0.1	1.8072e-001	9.7074e+001	0.156000
0.5	3.2369e+003	5.4064e+002	0.157000

5-2 Example:

Consider problem; $u_1(x) = (x - x^2) + \int_0^x (u_1(t) + u_2(t))dt$ and $u_2(x) = (x - \frac{x^2}{2} - \frac{x^3}{3}) + \int_0^x (u_1^2(t) + u_2(t))dt$,

Which is system of two non-linear VIE's with; $u_1(x) = x,\ u_2(x) = x$,

Tables (5, 6), present comparison between exact and numerical solution using classic Spline functions "Quadratic and Cubic", for $u_1(t)$ and $u_2(t)$ respectively, depending on last square error and running time with *h*=0.1,

Table(5),

t	Exact1	LSP	QSP	CSP
0.0	0	0	0	0
0.1	1.0000e-001	5.4933e-002	7.9233e-002	9.9233e-002
0.2	2.0000e-001	0.9345e-001	1.8245e-001	1.9985e-001

0.3	3.5000e-001	2.4384e-001	2.6754e-001	2.9985e-001
0.4	4.0000e-001	3.5818e-001	3.7858e-001	3.9978e-001
0.5	5.5000e-001	4.3855e-001	4.5185e-001	4.9990e-001
0.6	6.0000e-001	5.6667e-001	5.7617e-001	5.9930e-001
0.7	7.5000e-001	6.0101e-001	6.2291e-001	7.0134e-001
0.8	8.0000e-001	7.5892e-001	7.8202e-001	7.9354e-001
0.9	9.5000e-001	7.6116e-001	7.6346e-001	8.7086e-001
1.0	1.0000e+000	1.0294e+000	1.0200e+000	1.0054e+000
L. S. E.		2.1189e-001	3.0450e-002	9.2268e-004
R. T.		0.097000	0.109000	0.141000

Table(6),

t	Exact2	LSP	QSP	CSP
0.0	0	0	0	0
0.1	1.0000e-001	8.1805e-002	9.1805e-002	9.9805e-002
0.2	2. 0000e-001	0.9999e-001	1.9027e-001	2.0025e-001
0.3	3. 0000e-001	2.1821e-001	2.8477e-001	3.0042e-001
0.4	4. 0000e-001	3.5679e-001	3.8472e-001	4.0052e-001
0.5	5. 0000e-001	4.6881e-001	4.7021e-001	5.0081e-001
0.6	6. 0000e-001	5.7678e-001	5.8190e-001	6.0038e-001
0.7	7. 0000e-001	6.2789e-001	6.4163e-001	7.0259e-001
0.8	8. 0000e-001	7.8119e-001	7.9791e-001	7.9489e-001
0.9	9. 0000e-001	7.7821e-001	7.8992e-001	8.7128e-001
1.0	1. 0000e+000	1.1981e+000	1.1935e+000	1.0046e+000
L. S. E.		7.5391e-001	2.8080e-002	8.8035e-004
R. T.		0.97000	0.109000	0.141000

Tables (7), presents the error and running time for Cubic classic Spline when *h* is changing and initial value is fixed,

	H=0.1	H=0.05	H=0.02	H=0.01
CSP	9.2268e-004	1.0507e-005	NaN	NaN
	8.8035e-004	7.2419e-006	NaN	NaN
R. T.	0.141000	0.359000	1.875000	7.485000

Tables (8), presents the error and running time for Cubic classic Spline when initial value is changing and *h* is fixed,

I.V.	u_1	u_2	**R.T.**
0.0	9.2268e-004	8.8035e-004	0.141000

0.001	9.2269e-003	8.8488e-003	0.141000
0.01	5.7983e-001	2.7313e-000	0.141000
0.1	6.2564e-002	5.8120e+004	0.140000
0.5	4.5803e+004	3.8322e+007	0.141000

6- DISCUSSION

This Paper; introduce numerical methods for approximation solution of system of non-linear VIE's, which are classic Spline "Linear classic Spline, Quadratic classic Spline, Cube classic Spline", we have tried to emphasize some important ideas while maintaining a reasonable level of complexity, for one thing; we have always used a uniform step size,
This method for solving a system of non-linear VIE's has some advantages and disadvantages, the following remarks are concluded;

- The cube classic Spline gives better accuracy than Quadratic classic Spline,
- The quadratic classic Spline gives better accuracy than Linear classic Spline,
- The running time of quadratic classic Spline is less than running time of cubic classic Spline,

REFERENCES

[1] ABD Al-Hammeed, F. T. "**Numerical solution of Fredholm Integro-differential equations using Spline functions**" M.Sc. thesis, University Technology, 2002.

[2] Al-Kahachi; A. M. "**Approximate method for solving Volterra Integral equations of the first kind**" M.Sc. thesis, University Technology, 2000.

[3] Al-Salhi; B. F. "**Numerical solution of Non-Linear Volterra Integral equations of the second kind**" M.Sc. thesis, University Technology, 2000.

[4] Biazer; J., Babolian; E. and Islam; R. "**solution of system of Volttera Integral Equations of the first kind by Adomian Method**" Applied Math. And Computation, Vol. 139, (2003) pp. 249-258.

[5] Develse; L. M. and Walsh; J. "**Numerical solution of Integral Equations"** Claredon Press Oxford, 1974.

[6] Lutterkort; D. "**Computing Linear Envelopes for uniform B-Spline Curves"** Internet, 1999.

[7] Mostafa; M. M. "**Numerical Algorithms for Treating system of Volttera Integral equations using Spline functions**" PH. D. thesis, University of Al-Mustansiriya, College of Science, 2004.

[8] Nielsen; H. B. "**Cubic Spline Technical Report**" Dept. of Math. Modeling Technical, University of Denmark, 1998.

[9] Ravi; P. Agarwal and Donal; O'Regan, "**Integral and Integro-differential equations Theory, Methods and Applications**" University of Ireland, 2000.

[10] Restrepo; J. "**Spline Interpolations**" 2002, http://m707.math.arizonaedu/resterpo/475An_notes/sourcea/node35.html

العنوان بالعربي

اسم المؤلف

عنوانه

ملخص

في هذا العمل تم دراسة منظومة معادلات فولتيرا التكاملية اللاخطية من النوع الثاني، كما تم تطوير واستخدام ثلاث أنواع مختلفة من دوال الثلمة التقليدية "الخطية، التربيعية، التكعيبية"، حيث طورت وطبقت في منحى غير مسبوق لمعالجة المنظومة أعلاه، إضافة للمقارنات بين الحلول التقريبية والمضبوطة من خلال الاعتماد على أخطاء التربيعات الصغرى، وهو ما يمثل تعزيز للنتائج التي تم التوصل اليها باستخدام تلك الطريقة، مع استخدام برنامج " Matlab Version 7.0" لكتابة البرامج الخاصة بهذه الطريقة.