Journal of Al-Nahrain University **Vol.14 (3), September, 2011, pp.144-149** **Science**

Easy Numerical Method to Solution a System of Linear Volterra Integral Equations

Ahlam J. Kaleel[*], Huda H. Omran[**] and Salam J. Majeed[***]
[*]Department of Mathematics and Computer Application, College of Science, Al-Nahrain University.
[**]Department of Mathematics, College of Education, Ibn Al-Haitham, University of Baghdad.
[***]Department of Physics, College of Science, University of Thi-Qar.

Abstract

In this paper, we present an adapted method for solving systems of linear Volterra integral equations of the second kind. This method is based on the Simpson's rule. We used two numerical examples to show the accuracy and simple of our method by comparison with known methods.

1-Introduction

Consider the following system of linear Volterra integral equations of the second kind:

$$U(x) = F(x) + \int_a^x K(x,y)U(y)dy, \quad a \le x \le b, \quad \text{....................(1)}$$

where

$$U(x) = [u_1(x), u_2(x), ..., u_n(x)]^T,$$
$$F(x) = [f_1(x), f_2(x), ..., f_n(x)]^T,$$
$$K(x,y) = [k_{i,j}(x,y)], \quad i, j = 1, 2, ..., m.$$

In Eq.(1) the function F and kernel K are given, and U is the solution to be determined. We assume Eq.(1) has a unique solution.

Many problems for physics and other disciplines lead to integral equations. Several analytical and numerical methods for solving integral equations and its systems have been studied by the authors [1–8]. An adapted trapezoidal method presented in [9] for solving Fredholm integral equations and then in [10] for solving Volterra integral equations. Also, Volterra integral equations solved by modified adapted trapezoidal methods in [11] and by adapted Simpson's method in [12].

In this paper we use an adapted Simpson's method to solve the system of linear Volterra integral equations given by Eq.(1). The idea is to approximate the solutions of this equation in even number of equally spaced points. Then in interval [a, a+2h] we have

$$\int_a^{a+2h} k_{i,j}(x,y)u_j(y)dy = \frac{h}{3}\Big(k_{i,j}(x,a)u_j(a) + 4k_{i,j}(x,a+h)u_j(a+h) + k_{i,j}(x,a+2h)u_{i,j}(a+2h)\Big) - \frac{h^5}{90}\Big(k_{i,j}(x,\zeta)u(\zeta)\Big)^5, \quad \zeta \in (a, a+2h).$$

This approximation indicates that the error of integration denoted by E(h) over two segments by Simpson's rule is proportional to h^5 . Also, we note that if the segment width h is halved to h/2, then

$$E(\frac{h}{2}) \approx -2\frac{(h/2)^5}{90}\Big(k_{i,j}(x,\zeta)u(\zeta)\Big)^4 = \frac{1}{16}E(h).$$

2- Adapted Simpson's Method

Consider the ith equation of Eq.(1):

$$u_i(x) = f_i(x) + \int_a^x \sum_{j=1}^m k_{i,j}(x,y)u_j(y)dy, \quad i = 1, 2, ..., m \text{ ....(2)}$$

Let the interval [a, b] be finite and partitioned by 2n equally spaced points

$$a = x_0 < x_1 < \mathbf{L} < x_{2r-1} < x_{2r} < \mathbf{L} < x_{2n} = b.$$

The approximation of Eq.(1) in the even nodes x_{2r} is given by

$$u_i(x_{2r}) = f_i(x_{2r}) + \sum_{j=1}^m \int_a^{x_{2r}} k_{i,j}(x_{2r},y)u_j(y)dy$$
$$= f_i(x_{2r}) + \sum_{j=1}^m \sum_{s=0}^{r-1} \int_{x_{2s}}^{x_{2s+2}} k_{i,j}(x_{2r},y)u_j(y)dy$$

which can be rewritten as

$$u_{i,2r} = f_{i,2r} + \sum_{j=1}^m \sum_{s=0}^{r-1} \int_{x_{2s}}^{x_{2s+2}} k_{i,j}(x_{2r},y)u_j(y)dy.$$

Using Simpson's quadrature rule, the above discrete equation becomes

$$u_{i,2r}=f_{i,2r}+\sum_{j=1}^{m}\sum_{s=0}^{r-1}\frac{h}{3}\Big(k_{i,j,2r,2s}u_{j,2s}+4k_{i,j,2r,2s+1}u_{j,2s+1}+k_{i,j,2r,2s+2}u_{j,2s+2}\Big)$$

where $h=x_{r+1}-x_r=\dfrac{(b-a)}{2n}$,

(i=1, 2,…, m, r=1,2,…,n).

For a smaller step h, an approximation to $u_{i,2r}$ can then be computed by replacing $u_{j,2s+1}$ by the average $\left(u_{j,2s}+u_{j,2s+2}\right)/2$, to get

$$u_{i,2r}=f_{i,2r}+\sum_{j=1}^{m}\left(\sum_{s=0}^{r-1}\frac{h}{3}\Big(k_{i,j,2r,2s}u_{j,2s}+4k_{i,j,2r,2s+1}\frac{u_{j,2s}+u_{j,2s+2}}{2}+k_{i,j,2r,2s+2}u_{j,2s+2}\Big)\right)$$

$$=f_{i,2r}+\sum_{j=1}^{m}\left(\sum_{s=0}^{r-1}\frac{h}{3}\Big(k_{i,j,2r,2s}+2k_{i,j,2r,2s+1}\Big)u_{j,2s}+\sum_{s=0}^{r-1}\frac{h}{3}\Big(2k_{i,j,2r,2s+1}+k_{i,j,2r,2s+2}\Big)u_{j,2s+2}\right)$$

$$=f_{i,2r}+\sum_{j=1}^{m}\left(\sum_{s=0}^{r-1}\frac{h}{3}\Big(k_{i,j,2r,2s}+2k_{i,j,2r,2s+1}\Big)u_{j,2s}+\sum_{s=1}^{r}\frac{h}{3}\Big(2k_{i,j,2r,2s-1}+k_{i,j,2r,2s}\Big)u_{j,2s}\right)$$

$$=f_{i,2r}+\sum_{j=1}^{m}\frac{h}{3}\Big(\Big(k_{i,j,2r,0}+2k_{i,j,2r,1}\Big)u_{j,0}+2\sum_{s=1}^{r-1}\Big(k_{i,j,2r,2s-1}+k_{i,j,2r,2s}+k_{i,j,2r,2s+1}\Big)u_{j,2s}+\Big(2k_{i,j,2r,2r-1}+k_{i,j,2r,2r}\Big)u_{j,2r}\Big),$$

Therefore,

$$\left(u_{i,2r}-\frac{h}{3}\sum_{j=1}^{m}\Big(2k_{i,j,2r,2r-1}+k_{i,j,2r,2r}\Big)u_{j,2r}\right)=$$

$$f_{i,2r}+\sum_{j=1}^{m}\left(\frac{h}{3}\Big(k_{i,j,2r,0}+2k_{i,j,2r,1}\Big)u_{j,0}+\frac{2h}{3}\sum_{s=1}^{r-1}\Big(k_{i,j,2r,2s-1}+k_{i,j,2r,2s}+k_{i,j,2r,2s+1}\Big)u_{j,2s}\right),$$

(i=1, 2,…, m, r=1,2,…,n), ...(3)

where, $u_{i,0}=f_{i,0}$. This system can be solved to find the unknowns $\{u_{i,2r}\}_{i=1,r=1}^{m,n}$ by any suitable method.

3-Numerical examples

In this section, we give two examples to clarify the accuracy of the presented method by compares this method with other methods such as trapezoidal method, Simpson’s methods and Taylor-series expansion method. In these examples, $f_1(x)$ and $f_2(x)$ are chosen such that the exact solution is $u_1(x)$ and $u_2(x)$ and the computations were carried out using matlab 7.6.

Example 1.

Consider the following system of linear volterra integral equations.

$$u_1(x)=f_1(x)+\int_0^x (x-y)^3u_1(y)dy+\int_0^x (x-y)^2u_2(y)dy,$$

$$u_2(x)=f_2(x)+\int_0^x (x-y)^4u_1(y)dy+\int_0^x (x-y)^3u_2(y)dy.$$

where the exact solution is $u_1(x)=1+x^2$, $u_2(x)=1+x-x^3$.

Table 1 and 2 contain the exact solutions and the approximated solutions obtained by trapezoidal, Simpson’s 1/3, Simpson’s 3/8 and adaptive Simpson's methods. In Table 3, we compare the numerical results (error between exact and approximate value of $u_1(x)$, $u_2(x)$) of the problem in Example 1 by expansion method discussed in [5] and adapted Simpson's method.

Example 2.

Consider the following system of linear volterra integral equations.

$$u_1(x)=f_1(x)+\int_0^x \big(\sin(x-y)-1\big)u_1(y)dy+\int_0^x (1-y\cos x)u_2(y)dy,$$

$$u_2(x) = f_2(x) + \int_0^x u_1(y)dy + \int_0^x (x-y)u_2(y)dy.$$

where the exact solution is $u_1(x) = \cos x$, $u_2(x) = \sin x$.

Table (4) and (5) contain the exact solutions and the approximated solutions obtained by trapezoidal, Simpson's 1/3, Simpson's 3/8 and adaptive Simpson's methods. In Table 6, we compare the numerical results (error between exact and approximate value of $u_1(x)$, $u_2(x)$) of the problem in Example 2 by expansion method discussed in [5] and adapted Simpson's method.

Table (1)
Numerical results of $u_1(x)$, for Example 1 .

x	*Exact u1 n=12*	*Trapezoidal method n=12*	*Simpson's 1/3 method n=12*	*Simpson's 3/8 method n=12*	*The proposed method*	
					n=12	*n=24*
0	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.08333	1.00694	1.00705	1.00705	1.00705		1.00694
0.16667	1.02778	1.02800	1.02778	1.02800	1.02778	1.02778
0.25000	1.06250	1.06286	1.06286	1.06250		1.06250
0.33333	1.11111	1.11163	1.11111	1.11163	1.11111	1.11111
0.41667	1.17361	1.17430	1.17430	1.17430		1.17361
0.50000	1.25000	1.25087	1.25000	1.25000	1.24999	1.25000
0.58333	1.34028	1.34136	1.34135	1.34135		1.34028
0.66667	1.44444	1.44575	1.44446	1.44574	1.44443	1.44444
0.75000	1.56250	1.56405	1.56404	1.56253		1.56250
0.83333	1.69444	1.69628	1.69449	1.69626	1.69442	1.69444
0.91667	1.84028	1.84241	1.84236	1.84238		1.84028
1	2.00000	2.00248	2.00011	2.00013	1.99997	2.00000
Absolute Error u_1		0.0131	0.0060	0.0080	8.0794×10^{-5}	8.5377×10^{-6}
Lest Square Error u_1		2.1033×10^{-5}	8.4923×10^{-6}	1.1362×10^{-5}	1.7829×10^{-9}	1.0030×10^{-11}

Table (2)
Numerical results of $u_2(x)$, for Example 1.

x	*Exact n=12*	*Trapezoidal method n=12*	*Simpson's 1/3 method n=12*	*Simpson's 3/8 method n=12*	*The proposed method*	
					n=12	*n=24*
0	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.08333	1.08275	1.08277	1.08277	1.08277		1.08275
0.16667	1.16204	1.16209	1.16204	1.16209	1.16204	1.16204
0.25000	1.23438	1.23451	1.23451	1.23437		1.23438
0.33333	1.29630	1.29655	1.29630	1.29655	1.29630	1.29630
0.41667	1.34433	1.34476	1.34476	1.34476		1.34433
0.50000	1.37500	1.37565	1.37500	1.37500	1.37500	1.37500
0.58333	1.38484	1.38578	1.38577	1.38578		1.38484
0.66667	1.37037	1.37166	1.37038	1.37166	1.37039	1.37038
0.75000	1.32813	1.32985	1.32984	1.32814		1.32814
0.83333	1.25463	1.25687	1.25465	1.25686	1.25471	1.25465
0.91667	1.14641	1.14926	1.14923	1.14924		1.14645
1	1.00000	1.00356	1.00007	1.00009	1.00020	1.00006
Absolute Error u_2		0.0141	0.0062	0.0082	3.0684×10^{-4}	1.4071×10^{-4}
Lest Square Error u_2		3.2001×10^{-5}	1.1987×10^{-5}	1.5784×10^{-5}	4.7925×10^{-8}	5.1984×10^{-9}

Table (3)
Absolute Error of $u_1(x)$ and $u_2(x)$, for Example 1.

x	*Expansion Method*		*Adapted Simpson's Method*			
			n=10		*n=20*	
	Error of u_1	*Error of u_2*	*Error of u_1*	*Error of u_2*	*Error of u_1*	*Error of u_2*
0	0	0	0	0	0	0
0.1	1.37735×10^{-7}	2.11685×10^{-8}	4.16670×10^{-8}	2.32145×10^{-9}	2.60418×10^{-9}	1.02311×10^{-11}
0.2	1.62592×10^{-5}	2.61132×10^{-6}	1.66676×10^{-7}	1.31076×10^{-9}	1.04168×10^{-8}	4.87534×10^{-9}
0.3	1.74095×10^{-4}	4.18979×10^{-5}	3.75066×10^{-7}	1.06355×10^{-7}	2.34345×10^{-8}	4.45526×10^{-8}
0.4	8.93795×10^{-4}	2.86285×10^{-4}	6.66845×10^{-7}	6.23980×10^{-7}	4.16137×10^{-8}	1.98922×10^{-7}
0.5	3.00491×10^{-3}	1.1994×10^{-3}	1.04156×10^{-6}	2.15489×10^{-6}	6.47263×10^{-8}	6.22918×10^{-7}
0.6	7.47528×10^{-3}	3.56141×10^{-3}	1.49698×10^{-6}	5.70269×10^{-6}	9.19828×10^{-8}	1.57161×10^{-6}
0.7	1.40733×10^{-2}	7.74239×10^{-3}	2.02623×10^{-6}	1.27722×10^{-5}	1.21247×10^{-7}	3.42547×10^{-6}
0.8	1.78384×10^{-2}	1.09171×10^{-2}	2.61243×10^{-6}	2.54713×10^{-5}	1.47616×10^{-7}	6.71598×10^{-6}
0.9	4.97756×10^{-3}	2.27326×10^{-3}	3.21974×10^{-6}	4.66154×10^{-5}	1.61090×10^{-7}	1.21518×10^{-5}
1	3.84378×10^{-2}	3.32111×10^{-2}	3.77951×10^{-6}	7.98346×10^{-5}	1.42994×10^{-7}	2.06456×10^{-5}

Table (4)
Numerical results of $u_1(x)$, for Example 2.

x	*Exact n=12*	*Trapezoidal method n=12*	*Simpson's 1/3 method n=12*	*Simpson's 3/8 method n=12*	*The proposed method*	
					n=12	*n=24*
0	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
0.08333	0.99653	0.99648	0.99648	0.99648		0.99657
0.16667	0.98614	0.98603	0.98613	0.98603	0.98642	0.98621
0.25000	0.96891	0.96874	0.96875	0.96889		0.96900
0.33333	0.94496	0.94472	0.94493	0.94473	0.94532	0.94505
0.41667	0.91444	0.91415	0.91418	0.91416		0.91453
0.50000	0.87758	0.87723	0.87752	0.87751	0.87791	0.87767
0.58333	0.83463	0.83423	0.83429	0.83428		0.83470
0.66667	0.78589	0.78545	0.78578	0.78549	0.78608	0.78593
0.75000	0.73169	0.73121	0.73131	0.73154		0.73171
0.83333	0.67241	0.67189	0.67224	0.67198	0.67235	0.67240
0.91667	0.60847	0.60790	0.60806	0.60800		0.60841
1	0.54030	0.53967	0.54002	0.53999	0.53983	0.54018
Absolute Error u_1		0.0043	0.0023	0.0029	0.0017	7.9029×10^{-4}
Lest Square Error u_1		1.8815×10^{-6}	6.4889×10^{-7}	9.6023×10^{-7}	5.8778×10^{-7}	6.2313×10^{-8}

Table (5)
Numerical results of $u_2(x)$, for Example 2.

x	Exact n=12	Trapezoidal method n=12	Simpson's 1/3 method n=12	Simpson's 3/8 method n=12	The proposed method	
					n=12	n=24
0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.08333	0.08324	0.08309	0.08309	0.08309		0.08319
0.16667	0.16590	0.16560	0.16589	0.16560	0.16553	0.16581
0.25000	0.24740	0.24695	0.24696	0.24738		0.24727
0.33333	0.32719	0.32657	0.32716	0.32659	0.32648	0.32702
0.41667	0.40471	0.40392	0.40396	0.40394		0.40449
0.50000	0.47943	0.47846	0.47934	0.47933	0.47836	0.47916
0.58333	0.55081	0.54966	0.54974	0.54971		0.55049
0.66667	0.61837	0.61702	0.61821	0.61710	0.61689	0.61800
0.75000	0.68164	0.68009	0.68025	0.68140		0.68122
0.83333	0.74018	0.73842	0.73991	0.73856	0.73822	0.73969
0.91667	0.79358	0.79160	0.79185	0.79177		0.79302
1	0.84147	0.83927	0.84105	0.84099	0.83893	0.84084
Absolute Error u_2		0.0133	0.0065	0.0085	0.0081	0.0073
Lest Square Error u_2		1.9626×10^{-5}	7.1191×10^{-6}	1.0088×10^{-5}	1.4244×10^{-5}	1.5344×10^{-6}

Table (6)
Absolute Error of $u_1(x)$ and $u_2(x)$, for Example 2.

x	Expansion Method		Adapted Simpson's Method			
			n=10		n=20	
	Error of u_1	Error of u_2	Error of u_1	Error of u_2	Error of u_1	Error of u_2
0	0	0	0	0	0	0
0.1	1.37735×10^{-4}	1.52721×10^{-4}	6.80754×10^{-5}	8.01325×10^{-5}	1.69774×10^{-5}	1.99760×10^{-5}
0.2	9.27188×10^{-4}	1.14715×10^{-3}	1.09331×10^{-4}	1.55591×10^{-4}	2.72752×10^{-5}	3.87962×10^{-5}
0.3	2.67117×10^{-3}	3.71248×10^{-3}	1.29166×10^{-4}	2.29548×10^{-4}	3.22339×10^{-5}	5.72484×10^{-5}
0.4	5.45507×10^{-3}	8.57201×10^{-3}	1.31586×10^{-4}	3.04688×10^{-4}	3.28484×10^{-5}	7.59997×10^{-5}
0.5	9.2267×10^{-3}	1.64416×10^{-2}	1.19219×10^{-4}	3.83376×10^{-4}	2.97711×10^{-5}	9.56380×10^{-5}
0.6	1.38644×10^{-2}	2.78243×10^{-2}	9.33046×10^{-5}	4.67818×10^{-4}	2.33089×10^{-5}	1.16712×10^{-4}
0.7	1.9296×10^{-2}	4.25337×10^{-2}	5.36747×10^{-5}	5.60226×10^{-4}	1.34178×10^{-5}	1.39771×10^{-4}
0.8	2.56349×10^{-2}	5.91212×10^{-2}	1.27395×10^{-6}	6.62979×10^{-4}	3.03187×10^{-7}	1.65407×10^{-4}
0.9	3.31574×10^{-2}	7.48883×10^{-2}	7.46106×10^{-5}	7.78794×10^{-4}	1.86206×10^{-5}	1.94297×10^{-4}
1	4.19808×10^{-2}	8.70896×10^{-2}	1.70912×10^{-4}	9.10901×10^{-4}	4.26763×10^{-5}	2.27243×10^{-4}

4- Conclusions and Recommendations

In this paper we proposed an adaptive Simpson's method for solving systems of linear Volterra integral equations of the second kind with regular kernels. This method can be easily applied and it is efficient and accurate to estimate the solution of this system. Also we note that, when the exact solutions $u_i(x)$ of polynomials type, this method is more efficient.

References

[1] P. Linz, "Analytic and Numerical Solution of Integral Equations for Volterra Equations", SIAM Stud. Appl. Math., 1985.

[2] K.E. Atkinson, " The Numerical Solution of Integral Equations of the Second Kind", Cambridge University Press, 1997.

[3] H. Brunner, "Collocation Methods for Volterra Integral and Related Functional Differential Equations", Cambridge University Press, 2004.

[4] K. Maleknejad b, N. Aghazadeh, M. Rabbani a, " Numerical solution of second kind Fredholm integral equations system by using a Taylor-series expansion method", Applied Mathematics and Computation 175 (2006) 1229–1234.

[5] M. Rabbani a, K. Maleknejad b, N. Aghazadeh, " Numerical computational solution of the Volterra integral equations system of the second kind by using an expansion method", Applied Mathematics and Computation 187 (2007) 1143–1146.

[6] Elcin Yusufoglu (Agadjanov), " A homotopy perturbation algorithm to solve a system of Fredholm–Volterra type integral equations", Mathematical and Computer Modelling 47 (2008) 1099–1107.

[7] Ali Tahmasbi and Omid S. Fard, " A homotopy perturbation algorithm to solve a system of Fredholm–Volterra type integral equations", Applied Mathematics and Computation 201 (2008) 547–552.

[8] M.C. De Bonis and C. Laurita, " Numerical treatment of second kind Fredholm integral equations systems on bounded intervals", Journal of Computational and Applied Mathematics 217 (2008) 64 – 87.

[9] B. Neta and P. Nelson, "An Adaptive Method for the Numerical Solution of Fredholm Integral Equations of the Second Kind. I. Regular Kernels", Appl. Math. Comput., 21(2): 171–184, 1987.

[10] B. Neta and J. H. Gordis, "An Adaptive Method for the Numerical Solution of Volterra Integral Equations", Recent Advances in Applied and Theoretical Mathematics, N.E. Mastorakis, ed., Math. Comput. Sci. Eng., World Sci. Eng. Soc. Press, Athens, pp. 1–7, 2000.

[11] M. Nadir and B. Gagui. "Two Points for the Adaptive Method for the Numerical Solution of Volterra Integral Equations", Int. J., Math. Manuscripts, 1(1): 133-140, 2007.

[12] M. Nadir and A. Rahmoune, "Modified Method for Solving linear Volterra Integral Equations of the Second Kind Using Simpson's Rule", Int. J., Math. Manuscripts, 1(1): 141-146, 2007.

الخلاصة

في هذا البحث نقدم طريقة معدلة لحل انظمة معادلات فولتيرا التكاملية الخطية من النوع الثاني. هذه الطريقة مستند على قاعدة سمبسون. استعملنا مثالين عدديين لمشاهدة الدقة والبساطة في طريقتنا بالمقارنة مع طرق معروفة.