International Journal: Mathematical Manuscripts.
Volume 1 Number 1 (2007), pp. 141–146

http://www.ripublication.com/ijmm.htm

Modified Method for Solving Linear Volterra Integral Equations of the Second Kind Using Simpson's Rule

Mostefa Nadir
Department of Mathematics,
University of M'sila, Algeria

Azedine Rahmoune
Department of Mathematics,
University of Bordj Bou Arréridj, Algeria

Abstract

The aim of the present paper is to introduce a numerical method for solving linear Volterra integral equations of the second kind with regular kernels. The main idea is based on the adaptive Simpson's quadrature rule. The technique is very effective and simple. Thus, for showing efficiency and performance of this method, a number of numerical examples are cited.

AMS Subject Classification:
Keywords: Volterra integral equations, Simpson's quadrature rule, numerical approximation.

1. Introduction

Consider the second kind Volterra integral equations of the form

$$\varphi(x) = g(x) + \int_a^x k(x,t)\varphi(t)\,dt, \quad a \le x \le b, \tag{1.1}$$

where the function $g(x)$ and the regular kernel $k(x,t)$ are given, and $\varphi(x)$ is the unknown function to be determined.

This integral equation is a mathematical model of many evolutionary problems with memory arising from biology, chemistry, physics, engineering. In the recent years there are several numerical techniques using quadratures rules. For example, repeated trapezoidal rule, Newton-Cotes, Clenshaw-Curtis, and Simpson's rule. This is, of course, not the only way to approximate Volterra integral equations. See, for example, K. E. Atkinson [1], Peter Linz [2]. However, the convection-dominated problem in approximation is the spontaneous formation of non-smooth macro-scale features which pose a challenge for high-resolution computations.

In this paper we introduce a modified method which is based on the Simpson's quadrature formula. The idea is to approximate the solution of the above equation in even number of equally spaced points (or a given mesh). Then in the subinterval $[s, s+2h]$ we have

$$\begin{aligned}\int_s^{s+2h} & k(x,t)\varphi(t)\,dt \\ &= \frac{h}{3}\left(k(x,s)\varphi(s) + 4k(x,s+h)\varphi(s+h) + k(x,s+2h)\varphi(s+2h)\right) \\ &\quad -\frac{h^5}{90}(k(x,\zeta)\varphi(\zeta))^{(4)}.\end{aligned}$$

This indicates that the error $E(h)$ of integration over tow 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 $E(\frac{h}{2}) \approx -2\frac{(h/2)^5}{90}(k(x,\zeta)\varphi(\zeta))^{(4)} = \frac{1}{16}E(h)$.

2. Development of the Method

Consider a Volterra integral equation given by

$$\varphi(x) = g(x) + \int_a^x k(x,t)\varphi(t)\,dt, \quad a \le x \le b.$$

Let the interval $[a, b]$ be finite and partitioned by $2n$ equally spaced points (subdivision of smaller step h)

$$x_0 = a < x_1 < \cdots < x_{2j-1} < x_{2j} < \cdots < x_{2n}.$$

We solve the problem by marching in time. The approximation of (1.1) in the even nodes (x_{2j}) is given by

$$\begin{aligned}\varphi(x_{2j}) &= g(x_{2j}) + \int_a^{x_{2j}} k(x_{2j},t)\varphi(t)\,dt \\ &= g(x_{2j}) + \sum_{i=0}^{j-1}\int_{t_{2i}}^{t_{2i+2}} k(x_{2j},t)\varphi(t)\,dt\end{aligned}$$

which can be rewritten as

$$\varphi_{2j} = g_{2j} + \sum_{i=0}^{j-1}\int_{t_{2i}}^{t_{2i+2}} k(x_{2j},t)\varphi(t)\,dt.$$

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

$$\varphi_{2j} = g_{2j} + \sum_{i=0}^{j-1} \frac{h}{3} \left(k_{2j,2i}\varphi_{2i} + 4k_{2j,2i+1}\varphi_{2i+1} + k_{2j,2i+2}\varphi_{2i+2} \right).$$

For a smaller step h, an approximation to φ_{2j} can then be computed by replacing φ_{2i+1} by the average $\dfrac{\varphi_{2i} + \varphi_{2i+2}}{2}$,

$$\begin{aligned}
\varphi_{2j} &= g_{2j} + \sum_{i=0}^{j-1} \frac{h}{3} \left[k_{2j,2i}\varphi_{2i} + 4k_{2j,2i+1}\frac{\varphi_{2i} + \varphi_{2i+2}}{2} + k_{2j,2i+2}\varphi_{2i+2} \right] \\
&= g_{2j} + \sum_{i=0}^{j-1} \frac{h}{3} \left[\left(k_{2j,2i} + 2k_{2j,2i+1} \right) \varphi_{2i} + \left(2k_{2j,2i+1} + k_{2j,2i+2} \right) \varphi_{2i+2} \right] \\
&= g_{2j} + \sum_{i=0}^{j-1} \frac{h}{3} \left(k_{2j,2i} + 2k_{2j,2i+1} \right) \varphi_{2i} + \sum_{i=0}^{j-1} \frac{h}{3} \left(2k_{2j,2i+1} + k_{2j,2i+2} \right) \varphi_{2i+2} \\
&= g_{2j} + \sum_{i=0}^{j-1} \frac{h}{3} \left(k_{2j,2i} + 2k_{2j,2i+1} \right) \varphi_{2i} + \sum_{i=1}^{j} \frac{h}{3} \left(2k_{2j,2i-1} + k_{2j,2i} \right) \varphi_{2i}, \\
\varphi_{2j} &= g_{2j} + \frac{h}{3} \left(k_{2j,0} + 2k_{2j,1} \right) \varphi_0 + \frac{h}{3} \left(2k_{2j,2j-1} + k_{2j,2j} \right) \varphi_{2j} \\
&\quad + \frac{2h}{3} \sum_{i=1}^{j-1} \left(k_{2j,2i-1} + k_{2j,2i} + k_{2j,2i+1} \right) \varphi_{2i}.
\end{aligned}$$

In general, for $j = 1, 2, \ldots, n$

$$\begin{aligned}
\varphi_{2j} \left(1 - \frac{h}{3} \left(2k_{2j,2j-1} + k_{2j,2j} \right) \right) &= g_{2j} + \frac{h}{3} \left(k_{2j,0} + 2k_{2j,1} \right) \varphi_0 \\
&\quad + \frac{2h}{3} \sum_{i=1}^{j-1} \left(k_{2j,2i-1} + k_{2j,2i} + k_{2j,2i+1} \right) \varphi_{2i}.
\end{aligned}$$

Clearly from (1.1), the value of $\varphi(0)$ is $g(0)$, so $\varphi_0 = g_0$.

3. Numerical Experiments

In this section we describe some of the numerical experiments performed in solving the linear Volterra integral equations. In all cases we chose the right-hand side $g(x)$ in such a way that we know the exact solution. This exact solution is used only to show that the numerical solution obtained with our method is correct. Then, in such example, we calculate the absolute errors at some points (even points for example).

Example 3.1. The equation $\varphi(x) = 2 - e^{x+1} + \int_{-1}^{x} e^{x-t}\varphi(t)\,dt$ has exact solution $\varphi_e(x) = 1$. See Table 1.

t	$h = 0.1$	$h = 0.05$	$h = 0.025$
-1	0	0	0
-0.8	1.37577E -07	8.55288E -09	5.32963E -10
-0.6	3.43374E -07	2.13204E -08	1.32786E -09
-0.4	6.51221E -07	4.03833E -08	2.52112E -09
-0.2	1.11172E -06	6.88414E -08	4.29281E -09
0	1.80057E -06	1.11320E -07	6.93944E -09
0.2	2.83099E -06	1.74730E -07	1.08812E -08
0.4	4.37238E -06	2.69410E -07	1.67856E -08
0.6	6.67807E -06	4.10718E -07	2.55695E -08
0.8	1.01271E -05	6.21688E -07	3.86717E -08
1	1.52864E -05	9.36628E -07	5.82840E -08

Table 1. Errors for different steps in Example 1

Example 3.2. The equation $\varphi(x) = 1 + \int_{0}^{x} e^{-(x-t)}\varphi(t)\,dt$ has exact solution $\varphi_e(x) = 1 + x$ [3]. See Table 2.

t	$h = 0.1$	$h = 0.05$	$h = 0.025$
0	0	0	0
0.2	5.66099E -07	3.54109E -08	2.21257E -09
0.4	1.15495E -06	7.22079E -08	4.51245E -09
0.6	1.76488E -06	1.10389E -07	6.90124E -09
0.8	2.39757E -06	1.49965E -07	9.37416E -09
1	3.05245E -06	1.90928E -07	1.19362E -08

Table 2. Errors for different steps in Example 2

Example 3.3. The equation $\varphi(x) = 1 - \int_{0}^{x} (x-t)\varphi(t)\,dt$ has exact solution $\varphi_e(x) = \cos(x)$ [4]. See Table 3.

s	$h = 0.1$	$h = 0.05$	$h = 0.025$
0	0	0	0
0.2	6.58725E -05	1.65337E -05	4.13757E -06
0.4	2.58246E -04	6.48173E -05	1.62204E -05
0.6	5.61704E -04	1.40976E -04	3.52786E -05
0.8	9.51576E -04	2.38811E -04	5.97604E -05
1	1.39542E -03	3.50172E -04	8.76256E -05

Table 3. Errors for different steps in Example 3

Example 3.4. Consider the integral equation $\varphi(x) = \dfrac{(2-x)e^{-x^2}}{2} - \int_0^x xt\varphi(t)\,dt$, $0 \le x \le 1$ [5] with the exact solution $\varphi_e(x) = e^{-x^2}$. The numerical results are shown in Table 4.

s	$h = 0.1$	$h = 0.05$	$h = 0.025$
0	0	0	0
0.2	2.55494E -05	6.29520E -06	1.56817E -06
0.4	1.67386E -04	4.12408E -05	1.02727E -05
0.6	3.88530E -04	9.54955E -05	2.37720E -05
0.8	4.75592E -04	1.16643E -04	2.89688E -05
1	1.09918E -04	4.79273E -05	1.16985E -05

Table 4. Errors for different steps in Example 4

As it can be seen, for the four above examples, in order to test the consistence of the proposed method, we have changed the kernels form, since the solutions behave linearly (Examples **??**), oscillatory (Example 3.3), and exponentially (Example 3.4). The numerical results obtained in different tables shown that, each time, the step size was halved, the numerical error decreasing. For this reason, we suggest to repeat the algorithm with smaller steps to improve the numerical solution.

4. Conclusion

A modified method was developed and tested for the numerical solution of linear Volterra integral equations of the second kind with regular kernel. The computing of this approach is simple and effectively executed using symbolic computing codes on any personal computer.

References

[1] K.E. Atkinson. The Numerical Solution of Integral Equations of the Second Kind, Cambridge Univ. Press, Cambridge, 1997.

[2] P. Linz. Analytical and Numerical Methods for Volterra Equations, *SIAM Stud. Appl. Math.*, 1985.

[3] J. M. Bownds and B. Wood. A Note on Solving Volterra Integral Equations with Convolution Kernels, *Appl. Math. Comput.*, 3:307–315, 1977.

[4] K. Maleknejad and N. Aghazadeh. Numerical Solution of Volterra Integral Equations of the Second Kind with Convolution Kernel by Using Taylor-Series Expansion Method, *Appl. Math. Comput.*, 161:915–922, 2005.

[5] L. M. Delves and J. L. Mohamed. Computational Methods for Integral Equations, Cambridge University Press, Cambridge, 1985.

[6] M. Masajima. Applied Mathematical Methods in Theorical Physics, Wiley-VCH Verlag , GmbH. KGaA, Weinheim, 2005.

[7] P. K. Kyth and M. R. Schäferkotter. Handbook of Computational Methods for Integration, Chapman & Hall. CRC Press, USA, 2005.

[8] M. Nadir and B. Gagui. to appear, Two Points for the Adaptive Method for the Numerical Solution of Volterra Integral Equations, ATMP.