International Journal of Industrial Engineering & Production Research (2010) pp. 135-138

March 2010, Volume 20, Number 4

International Journal of Industrial Engineering & Production Research

ISSN: 2008-4889

Journal Website: http://IJIEPR.iust.ac.ir/

Monte Carlo Simulation to Solve the Linear Volterra Integral Equations of The Second Kind

R. Farnoosh* **& M. Ebrahimi**

R. Farnoosh, *School of Mathematics, Iran University of Science and Technology.*
M. Ebrahimi*, School of Mathematics, Iran University of Science and Technology*

KEYWORDS

Monte Carlo method,
Markov chain,
Computer simulation,
Volterra integral
equation of the
second kind

ABSTRACT

This paper is intended to provide a numerical algorithm based on random sampling for solving the linear Volterra integral equations of the second kind. This method is a Monte Carlo (MC) method based on the simulation of a continuous Markov chain. To illustrate the usefulness of this technique we apply it to a test problem. Numerical results are performed in order to show the efficiency and accuracy of the present method.

© 2009 IUST Publication, All rights reserved. Vol. 20, No.4

1. Introduction

Monte Carlo methods for partial differential equations and integral equations are widespread. There are several basic advantages of Monte Carlo algorithms. It is clear that MC algorithms are parallel algorithms. They have high parallel efficiency where parallel computers are used [1-4]. It is well known that MC methods are preferable for solving high dimensional Multiple integrals, such as those arising from approximations of integral equations [5-7]. Therefore such methods are good for solving integral equations, specially they are a classical tools for solving high dimensional integral equations. In a number of papers, MC methods for the computation of integrals depending on a parameter, integral operators and the solution of integral equations were proposed and studied [8].

Recently Farnoosh and Ebrahimi [3], employed MC method based on the simulation of a continuous Markov chain, for solving Fredholm integral equations of the second kind. This method solves the problem without any need to discretization of variables. In this paper we will consider the following linear Volterra integral equation of the second kind:

$$u(x) = f(x) + \lambda \int_0^x k(x,t)u(t)dt,\ \lambda = 1,\ 0 \le x \le 1\ , \tag{1}$$

where the function, $f(x) \in L_2[0,1]$ the kernel $k(x,t) \in L_2([0,1]\times[0,1])$ are given and $u(x) \in L_2[0,1]$ is an unknown function to be determined.

Here we want to propose a numerical method based on random sampling to find an approximation to the unknown function $u(x)$. Associated with integral equation (1), two subproblems can be distinguished. In the first case we seek to approximate the full solution function $u(x)$ in some way (e.g., by solving on a grid and interpolating or by finite difference approximation etc.). We call this the problem of global solution. In the second case we want to approximate the value $u(x_0)$ of the solution in a single point x_0. This is called the local solution problem [9]. While deterministic numerical methods such as Nystrom, Collocation, FDM usually aim at solving the global problem, the classical MC approach is directed to the local solution. MC methods are well understood in this situation and are generally acknowledged to bring advantages over the deterministic approaches. In this paper we show that how the MC method can be used to solve linear Volterra integral equation (1). Our idea for solving integral equation (1) by a MC method is using of continuous Markov chain with state space [0,1], for simulation.

2. Overview of the Method

Equation (1) may be written in the operational form as

$$u(x) = f(x) + (Ku)(x)\ ,$$

Or

$$u = f + Ku$$

*Corresponding author. R. Farnoosh
Email: *rfarnoosh@iust.ac.ir mo.ebrahimi@iust.ac.ir.*
Paper first received Mar. 21. 2008 ,and in revised form July. 26. 2009.

[Downloaded from www.iust.ac.ir on 2025-06-23]

where K is an integral operator for the integral in equation (1) which is defined as follows:

$$(Ku)(x) = \int_0^x k(x,t)u(t)dt.$$

(Ku)(x) is called the first iteration of u with respect to the kernel k. The second iteration is

$$K[(Ku)](x) = (K^2u)(x) = \int_0^x \int_0^t k(x,t)k(t,t_1)u(t_1)dt_1dt\,.$$

Proceeding recursively we obtain the n-th iteration of u with respect to the kernel k as

$$K[(K^{n-1}u)](x) = (K^n u)(x)$$
$$= \int_0^x k(t,t_{n-1})K^{n-1}u(t_{n-1})dt_{n-1}. \tag{2}$$

Let us assume that

$$\|K\| = \sup_{[0,1]} \int_0^x |k(x,t)dt| < 1. \tag{3}$$

Under this assumption we can solve (1) by applying the following recursive equation,

$$u^{(n+1)} = Ku^{(n)} + f, \quad n = 1,2,..., \tag{4}$$

if $u^{(0)} = 0$ and $K^{(0)} = 0$, then from equation (4), we obtain

$$u^{(n+1)} = f + Kf + ... + K^n f = \sum_{m=0}^{n} K^m f\,. \tag{5}$$

It is well known that property (3) is a sufficient condition for convergence, i.e.

$$\lim_{n\to+\infty} u^{(n)} = \lim_{n\to+\infty} \sum_{m=0}^{n} K^m f = u.$$

In our algorithm we compute the iterations $u^{(n)}$ by using Monte Carlo method where n is a finite number.

2-1. Monte Carlo Method

The application of the present Monte Carlo method to find a solution of Volterra integral equation (1), is as follows:

Consider the continuous Markov chain

$$P = \|P(x,y)\|, \tag{6}$$

with state space [0,1], satisfying

$$\int_0^1 P(x,y)dy = 1, \tag{7}$$

and

$$\int_0^1 p(x)dx = 1, \tag{8}$$

where p(x) and P(x,y) are, respectively, the initial and the
transition densities of the Markov chain (6)-(8).

Define the weight function W_m, for the Markov chain (6)-(8)
using recursion formula

$$W_m = W_{m-1}\frac{k(x_{m-1},x_m)}{P(x_{m-1},x_m)}, \quad m = 1,2,..., \tag{9}$$

where $W_0 = 1.$

Now define the following random variable

$$\Gamma_n[h] = \frac{h(x_0)}{p(x_0)}\sum_{m=0}^{n} W_m f(x_m), \tag{10}$$

associated with the sample path

$$x_0 \to x_1 \to x_2 \to ... \to x_n,$$

where n is a given integer number and $h(x)$ is a given real value function.

Definition 1. The inner product of two functions $h(x)$ and $u(x)$ is defined by

$$\langle h,u\rangle = \int_0^1 h(x)u(x)dx. \tag{11}$$

Remark 1. If $h(x) \in L_2[0,1]$ and $u(x) \in L_2[0,1]$ then $|\langle h,u\rangle| < \infty$.

Remark 2. If $u(x) \in L_2[0,1]$ and $k(x,t) \in L_2([0,1]\times[0,1])$ then $(Ku)(x) \in L_2[0,1]$.

Now we consider the problem of finding the inner product $\langle h,u\rangle$ where $h(x)$ is a given real valued function and $u(x)$ is solution of Volterra integral equation (1).

Theorem 1. The mathematical expectation value of the random variable $\Gamma_n[h]$ is equal to the inner product $\langle h,u^{(n+1)}\rangle$, i.e.

$$E(\Gamma_n[h]) = \langle h,u^{(n+1)}\rangle.$$

Proof. Each path $x_0 \to x_1 \to x_2 \to ... \to x_n$, will be realized with probability

[Downloaded from www.iust.ac.ir on 2025-06-23]

$$P(x_0, x_1, ..., x_n) = p(x_0)P(x_0, x_1)P(x_1, x_2)...P(x_{n-1}, x_n).$$

Since the random variable $\Gamma_n[h]$ is defined along path $x_0 \to x_1 \to x_2 \to ... \to x_n$, we have

$$E(\Gamma_n[h]) = \int_0^1\int_0^1 ... \int_0^1 \Gamma_n[h]p(x_0)P(x_0, x_1)...P(x_{n-1}, x_n)dx_0...dx_n,$$
$$= \int_0^1\int_0^1 ... \int_0^1 h(x_0)\sum_{m=0}^{n} \Phi\Psi p(x_m)dx_0dx_1...dx_n,$$

which, together with (9) and (10), gives

$$E(\Gamma_n[h]) = E\left(\frac{h(x_0)}{p(x_0)}\sum_{m=0}^{n} W_m f(x_m)\right), \qquad (12)$$

where

$$\Phi = k(x_0, x_1)k(x_1, x_2)...k(x_{n-1}, x_n),$$

and

$$\Psi = P(x_m, x_{m+1})...P(x_{n-1}, x_n).$$

Using the property $\int_0^1 P(x, y)dy = 1$, the equation (12) can be written as

$$E(\Gamma_n[h]) = \sum_{m=0}^{n}\int_0^1\int_0^1 ... \int_0^1 h(x_0)\Phi f(x_m)dx_0dx_1...dx_n.$$

Thus, taking into account (2), (5) and (11) we deduce

$$E(\Gamma_n[h]) = \langle h(x), \sum_{m=0}^{n}(K^m f)(x)\rangle = \langle h(x), u^{(n+1)}(x)\rangle.$$

In fact, from Theorem 1 we conclude that $\Gamma_n[h]$ is an unbiased estimator of the inner product $\langle h, u^{(n+1)}\rangle$, while $u^{(n+1)}$ is the (n+1)-th iterative solution of (4). In this paper, we estimate $\langle h(x), u^{(n+1)}(x)\rangle$ by using Monte Carlo method based on simulation of a continuous Markov chain.

3. Computational Procedure For Estimating $\langle h, u^{(n+1)}\rangle$

To estimate $\langle h, u^{(n+1)}\rangle$, we consider a continuous Markov chain with transition kernel

$$P(x, y) = \rho(x)\delta(x - y) + (1 - \rho(x))g(y), \qquad (13)$$

where $x, y \in [0,1]$, $\delta(x-y)$ is the Dirac's delta function at x, g(x) is a probability density on [0,1], ρ is a function such that $0 < \rho(x) < 1$ and $\int_0^1 \frac{g(x)}{1-\rho(x)} < \infty$.

Now, we implement the following steps:

Step 1. At the beginning of the computational procedure we choose any integer n and then we simulate N independent random paths of length n

$$x_0^{(s)} \to x_1^{(s)} \to x_2^{(s)} \to ... \to x_n^{(s)}, \quad s = 1(1)N, \qquad (14)$$

from the Markov chain (13). In this work, we proposed an algorithm for generating the Markov chain associated with (14) as follows:

Algorithm 1.

1. Input initial Data:

The number of trajectory N, the length of Markov chain n, the function $\rho(x)$ and the probability density g on [0,1].

2. For s:=1 to N do Step 2.1:

2.1.Perform one trajectory:

2.1.1. Generate $x_0^{(s)}$ from density p(x);

2.1.2. For m:=0 to n-1 do

2.1.2.1. Generate an uniformly distributed random number $\tau \in (0,1)$;

2.1.2.2. If $\rho(x_m^{(s)} > \tau)$ then

2.1.2.2.1. Equate $x_{m+1}^{(s)} = x_m^{(s)}$;

else

2.1.2.2.2. Generate $x_{m+1}^{(s)}$ from probability density;

2.1.3. End of the trajectory.

3. End of the Algorithm 1.

Step 2. We also define the random variable $\Gamma_n[h]$ on the path (14) such that,

$$\Gamma_n[h] = \frac{h(x_0)}{p(x_0)}\sum_{m=0}^{n} W_m^{(s)} f(x_m),$$

Where

$$W_m^{(s)} = \frac{k(x_0^{(s)}, x_1^{(s)})k(x_1^{(s)}, x_2^{(s)})...k(x_{n-1}^{(s)}, x_n^{(s)})}{P(x_0^{(s)}, x_1^{(s)})P(x_1^{(s)}, x_2^{(s)})...P(x_{n-1}^{(s)}, x_n^{(s)})},$$

where $m = 1,2,...$ and $W_0^{(s)} = 1$.

Step 3. Finally evaluate the sample mean

$$\Theta_n[h] = \frac{1}{N}\sum_{s=1}^{N}\Gamma_n^{(s)}[h] \approx \langle h, u^{(n+1)}\rangle.$$

4. Numerical Experiment

It is readily seen that by setting $h(x) = \delta(x - y + \varepsilon)$, where $0 < \varepsilon < y$ and $y \in [0,1]$ and $\delta(.)$ is the Dirac's delta function at $y - \varepsilon$. Thus, we obtain

[Downloaded from www.iust.ac.ir on 2025-06-23]

$$\langle h(x), u^{(n+1)}(x)\rangle = \langle \delta(x-y), \sum_{m=0}^{n}(K^m f)(x)\rangle$$
$$= \int_0^1 \delta(x-y)\left(\sum_{m=0}^{n}(K^m f)(x)\right)dx$$
$$= \sum_{m=0}^{n}(K^m f)(y) = u^{(n+1)}(y), \qquad (15)$$

whene $y \in [0,1]$. Hence from equation (15) and Theorem 1 we immediately obtain an unbiased estimator $\Gamma_n[h]$ of the $u^{(n+1)}$. In this work to obtain u(y) for some given $y \in [0,1]$, we implement the Algorithm 1 when p(x)=h(x)= $\delta(x-y)$ and g is the density of the Beta distribution with parameters α and β, i.e.

$$g \sim B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \quad \alpha > 0, \quad \beta > 0,$$

where $\Gamma(.)$ is the Gama function. To give a clear overview of the present MC method, the following example is considered and the solution of which is to be obtained. We consider $\beta = 1$ and α varies around 2.

Example 1. Consider the following Volterra integral equation of the second kind

$$u(x) = \exp(-x^2) + \int_0^x xtu(t)dt, \qquad 0 \le x \le 1$$

for which the exact solution is

$$u(x) = \exp(-x^2) + \frac{x(1-\exp(-x^2))}{2}.$$

The results obtained for error between exact solution and present Monte Carlo solution are shown in Table 1.

Tab. 1. The error estimation obtained with $\beta = 1$ and $\alpha = 2.02$.

X	MC (n=5 and N=10^3)	MC (n=32 and =10^5)
0.1	1.005×10^{-3}	2.014×10^{-4}
0.2	1.010×10^{-3}	3.207×10^{-5}
0.3	1.027×10^{-3}	3.005×10^{-4}
0.4	1.017×10^{-3}	1.001×10^{-3}
0.5	1.103×10^{-3}	2.417×10^{-4}
0.6	1.200×10^{-3}	3.076×10^{-4}
0.7	1.083×10^{-3}	7.005×10^{-5}
0.8	1.205×10^{-3}	5.019×10^{-4}
0.9	1.220×10^{-3}	3.715×10^{-4}
1.0	1.316×10^{-3}	0.201×10^{-3}

5. Conclusion and Future Works

The present study, successfully applies a stochastic technique involving the Monte Carlo method based on simulation of continuous Markov chain, for solving the linear Volterra integral equations of the second kind. From the numerical example it can be seen that the proposed Monte Carlo method is efficient and accurate to estimate the solution of Volterra integral equations. We also applied other different values of number of independent random paths N such as N=100 and N=5×10^6. The results show that the present Monte Carlo method is also very efficient. Furthermore the results indicate that the present Monte Carlo method is preferable when one needs to have a rough estimation. In future papers we plan to extend the present approach to system of Fredholm integral equations and nonlinear integral equations with high dimensions.

References

[1] Dimov, I.T., Tonev, O., *"Monte Carlo Algorithms: Performance Analysis for Some Computer Architectures"*, Journal of Computational and Applied Mathematics, Vol. 48, 1993, pp. 253-277.

[2] Dimov, I.T., Dimov, T.T., Gurov, T.V., *"A New Iterative Monte Carlo Approach for Inverse Matrix Problem"*, Journal of Computational and Applied Mathematics, Vol. 92, 1998, pp. 15–35.

[3] Farnoosh, R., Ebrahimi, M., "*Monte Carlo Method for Solving Fredholm Integral Equations of the Second Kind*", Applied Mathematics and Computation, Vol. 195, 2008, pp. 309-315.

[4] Farnoosh, R., Ebrahimi, M., *"Monte Carlo Method via a Numerical Algorithm to Solve a Parabolic Problem"*, Applied Mathematics and Computation, Vol. 190, 2007, pp. 1593-1601.

[5] Haselgrove, C.B., *"A method for numerical integration, Mathematics of Computation"*, Vol. 15 No. 76, 1961, pp. 323-337.

[6] Robert, C.P., Casella, G., *"Monte Carlo Statistical Methods"*, Springer, New York, 1999.

[7] Rubinstein, R.Y., Simulation and the Monte Carlo method, Wiley, New York, 1981.

[8] Heinrich, S., *"Monte Carlo Approximation of Weakly Singular Integral Operators"*, Journal of Complexity, Vol. 22 , 2006, pp. 192–219.

[9] Heinrich, S., *"Monte Carlo Complexity of Global Solution of Integral Equations"*, Journal of complexity, Vol. 14, 1998, pp. 151-175.

[Downloaded from www.iust.ac.ir on 2025-06-23]